Full length article

VQF: Highly accurate IMU orientation estimation with bias estimation and magnetic disturbance rejection

Daniel Laidig [a,*], Thomas Seel [b]

[a] *Control Systems Group, Technische Universität Berlin, 10623, Berlin, Germany*
[b] *Department Artificial Intelligence in Biomedical Engineering, Friedrich-Alexander-Universität Erlangen-Nürnberg, 91052, Erlangen, Germany*

ABSTRACT

The miniaturization of MEMS-based inertial measurement units (IMUs) facilitates their widespread use in a growing number of application domains. The fundamental sensor fusion task of orientation estimation is a prerequisite for most further data processing steps in inertial motion tracking, such as position and velocity estimation, joint angle estimation, and 3D visualization. Errors in the estimated orientations severely affect all further processing steps. Recent systematic comparisons of existing algorithms show that out-of-the-box accuracy is often low and that application-specific tuning is required to obtain high accuracy. In the present work, we propose and extensively evaluate a quaternion-based orientation estimation algorithm that is based on a novel approach of filtering the acceleration measurements in an almost-inertial frame and that includes extensions for gyroscope bias estimation and magnetic disturbance rejection, as well as a variant for offline data processing. In contrast to all existing work, we perform an extensive evaluation, using a large collection of publicly available datasets and eight literature methods for comparison. The proposed method consistently outperforms all eight literature methods and achieves an average RMSE of 2.9°, while the errors obtained with literature methods range from 5.3° to 16.7°. This improved accuracy with respect to the state of the art is observed not only in average but also for each of several different motion characteristics, as well as for gyroscope bias estimation. Since the evaluation was performed with one single fixed parametrization across a very diverse dataset collection, we conclude that the proposed method provides unprecedented out-of-the-box performance for a broad range of motions, sensor hardware, and environmental conditions. This gain in orientation estimation accuracy is expected to advance the field of IMU-based motion analysis and provide performance benefits in numerous applications. The provided open-source implementation makes it easy to employ the proposed method.

1. Introduction

In recent years, MEMS-based inertial measurement units (IMUs) have become small, lightweight, and affordable. New application domains in which they are used include, for example, sports [1], gait analysis [2,3], rehabilitation monitoring [4], rehabilitation robotics [5], autonomous vehicles [6], aerial vehicles [7], and kites [8]. In all these applications, IMUs are used to estimate variables of motion, such as orientations, velocities, and positions, either in real time or via postprocessing of recorded data.

IMUs measure angular rate, specific force (also called proper acceleration), and magnetic field strength, each as a time-dependent 3D vector in an intrinsic sensor coordinate system. Those measurements are processed to determine the motion parameters of interest, e.g., the orientation of an object to which the sensor is attached, the object's

velocity or position, or other application-specific motion parameters [9, 10]. As shown in Fig. 1, determining such motion parameters generally requires the prior estimation of the orientation of the sensor with respect to an inertial frame of reference, a procedure known as *inertial orientation estimation (IOE)*.

Since IOE is such a fundamental step in IMU-based motion analysis and the accuracy of all further parameters of interest depends on the accuracy of the orientation estimate, it is not surprising that abundant prior research has aimed at solving this task. Comprehensive reviews that classify and compare the existing solution approaches are found in [11,12]. The existing methods widely vary with respect to the filter type, the computational complexity, the number of tuning parameters, and the additional features such as gyroscope bias estimation. For
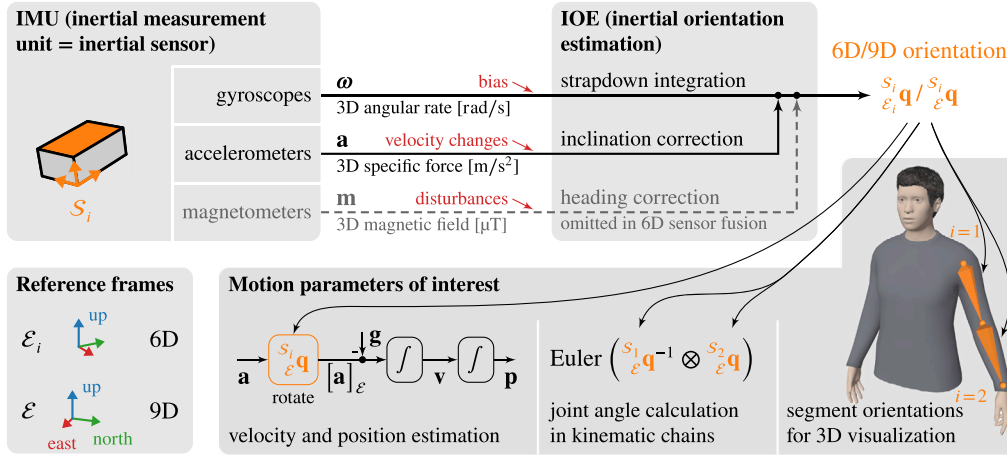
Fig. 1. Orientation estimation of IMUs is achieved by sensor fusion of the gyroscope measurements with the accelerometer and, in 9D sensor fusion, magnetometer measurements. Obtaining an accurate orientation estimate is the prerequisite for fundamental further steps in inertial motion tracking, including velocity and position estimation, joint angle calculation, and 3D visualization.

several algorithms, implementations in C++ or Matlab are available, see e.g. [12–17].

In literature that proposes new IOE algorithms, accuracy is most commonly validated using marker-based optical motion capture (OMC) as a ground truth. However, the employed datasets are almost always non-public and highly application-specific; they vary widely in terms of movement speed, characteristics of the employed motion, magnetic environment, and sensor error characteristics. Therefore, reported performance figures cannot be compared directly. This lack of common datasets and suitable benchmarks has recently been addressed by the publication of the Sassari dataset [18] and the Berlin Robust Orientation Estimation Assessment Dataset (BROAD) [19]. Newly proposed algorithms should be validated using such publicly available datasets in comparison with other state-of-the-art algorithms [11].

The few existing comparative studies show that the out-of-the-box performance of most algorithms is poor and that application-specific algorithm selection, as well as laborious parameter tuning, are necessary to achieve good results [12,19], which represents a severe limitation of the state of the art. Even with optimized parameters, the root-mean-square errors achieved by the best IOE algorithms are in the range of 1° to 3° for slow and smooth motions and as much as 5° to 15° for fast and challenging motions [12,19]. Further improving this accuracy seems highly desirable in view of numerous applications.

In summary, while there is ample work on various IOE algorithms, evaluation of the proposed methods is often limited and cannot be compared across publications. Recent comparative reviews and benchmarks show that there is no one-size-fits-all solution that works out of the box and yields high accuracy for a wide variety of application scenarios. Furthermore, the widespread adoption of novel IOE algorithms is not only driven by accuracy but also depends on the availability of an easy-to-use implementation. In combination, this demonstrates that there is a need for an algorithm that is validated on a very large and diverse set of experimental data, provides accurate out-of-the-box orientation estimates without tuning, and is easy to use and to integrate into existing code projects.

We aim at filling this gap in two steps: We first propose a new feature-rich quaternion-based orientation estimation algorithm and then perform an extensive validation to demonstrate the exceptionally high accuracy that is achieved by this algorithm. With respect to the first step, the key differences of the proposed algorithm with respect to the latest state of the art are best expressed by the following five features:

1. As a novel approach to sensor fusion of gyroscopes and accelerometers, the accelerometer information is low-pass filtered

in an almost-inertial frame, which yields robust rejection of accelerations due to velocity changes.
2. Magnetometer-based heading correction is performed as a modular decoupled step, which eliminates the influence of magnetic disturbances on the inclination and facilitates simultaneous 6D and 9D estimation.
3. The algorithm includes extensions for online gyroscope bias estimation during rest and motion and an optional magnetic disturbance rejection strategy.
4. In contrast to the vast majority of previous approaches, an acausal offline version is available, which further increases the accuracy in situations in which real-time capability is not required.
5. Easy-to-use open-source implementations of the proposed algorithms are provided in C++, Python, and Matlab.

With respect to the second step, the main contributions and results of the extensive accuracy evaluation are:

1. In contrast to most previous work, the proposed method is extensively evaluated using a large collection of publicly available data and in comparison with eight existing IOE algorithms.
2. The results show that the proposed method outperforms all evaluated existing methods, providing a 1.8-fold to 5-fold increase in orientation estimation accuracy.
3. For a large variety of motions, speeds, and disturbed environments, the proposed method works out of the box, and application-specific parameter tuning is not necessary.

2. Proposed method for inertial orientation estimation

We briefly explain the employed terminology and notation and then, building upon preliminary work in [20], propose a modular method for simultaneous 6D and 9D orientation estimation.

2.1. Terminology and notation

As illustrated in Fig. 1, the following measurements are available in IOE: gyroscope readings $\boldsymbol{\omega}(t_k) \in \mathbb{R}^3$, accelerometer readings $\mathbf{a}(t_k) \in \mathbb{R}^3$, and magnetometer readings $\mathbf{m}(t_k) \in \mathbb{R}^3$, sampled at times $t_k = kT_s, k \in \{1, 2, \ldots, N\}, T_s \in \mathbb{R}_{>0}$.

If only gyroscopes and accelerometers are employed, we use the term *6D* IOE, while *9D* IOE additionally uses magnetometers. Therefore, 9D IOE yields the sensor orientation with respect to a fixed inertial reference frame, typically using the east-north-up (ENU) convention
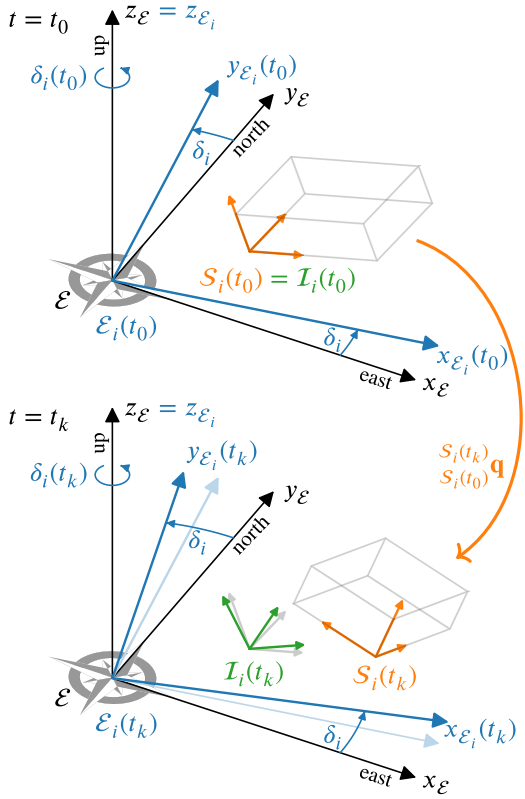
Fig. 2. Illustration of the different coordinate systems used by the proposed method. The aim of IOE is to determine the orientation of the sensor S_i relative to an ENU reference frame $\mathcal{E}$ (in 9D sensor fusion) or relative to a reference frame $\mathcal{E}_i$ with vertical z-axis (in 6D sensor fusion). The angle δ_i describes the slowly drifting heading offset between $\mathcal{E}$ and $\mathcal{E}_i$. Internally, the auxiliary $\mathcal{I}_i$ frame is used to represent the orientation obtained by pure gyroscope strapdown integration and slowly drifts due to the integration of gyroscope bias.

(i.e., z is pointing up and y is pointing north). In contrast, only vertical reference information is available in 6D IOE, and the resulting orientations are thus provided with respect to an almost-inertial reference frame, which has one vertical axis and slowly drifts around this axis (at a rate determined by the gyroscope bias, i.e., typically $\leq 1°/\text{s}$).

We denote the moving sensor frame, i.e., the coordinate system in which the sensor readings are provided, by $S_i(t_k)$. The ENU inertial reference frame used in 9D IOE is denoted $\mathcal{E}$, and the sensor-specific almost-inertial reference frame used in 6D IOE is denoted $\mathcal{E}_i(t_k)$. Anticipating the common application scenario with multiple IMUs on different segments of a kinematic chain, a sensor index i is used for sensor-specific quantities.

We use square brackets to specify the coordinate system in which a vector is expressed, for example, $[\mathbf{a}]_{\mathcal{E}}$ is the accelerometer measurement transformed into frame $\mathcal{E}$. We denote rotations and orientations by unit quaternions [21] in vector notation, i.e., we write the quaternion $w + ix + jy + kz$ as $[w \ x \ y \ z]^\mathsf{T}$. In the context of quaternion multiplication, which we denote by $\otimes$, we implicitly regard 3D vectors as pure quaternions. For example, $[\mathbf{a}]_{\mathcal{E}} = {}^S_{\mathcal{E}}\mathbf{q} \otimes \mathbf{a} \otimes {}^S_{\mathcal{E}}\mathbf{q}^{-1}$. Here, the left upper and lower indices denote the frames between which the quaternion rotates. Quaternions that represent the rotation of an angle $\alpha \in \mathbb{R}$ around the axis $\mathbf{v} \in \mathbb{R}^3$ are written as $(\alpha \ @ \ \mathbf{v}) := \left[\cos\frac{\alpha}{2} \quad \frac{\mathbf{v}^\mathsf{T}}{\|\mathbf{v}\|} \sin\frac{\alpha}{2} \right]^\mathsf{T}$.

2.2. A modular estimation approach

The most fundamental state of any IOE algorithm is the current orientation estimate, which is commonly represented by a single orientation quaternion. For the proposed method, we use a more modular

approach and represent the 6D estimate ${}^{S_i(t_k)}_{\mathcal{E}_i(t_k)}\mathbf{q}$ as the concatenation of an inclination correction quaternion ${}^{\mathcal{I}_i(t_k)}_{\mathcal{E}_i(t_k)}\mathbf{q}$ with a gyroscope strapdown integration quaternion ${}^{S_i(t_k)}_{\mathcal{I}_i(t_k)}\mathbf{q}$, and the 9D estimate ${}^{S_i(t_k)}_{\mathcal{E}}\mathbf{q}$ as the concatenation of a heading correction rotation ${}^{\mathcal{E}_i(t_k)}_{\mathcal{E}}\mathbf{q}$, represented by the scalar heading offset $\delta_i(t_k)$, with the aforementioned 6D estimate, i.e.,

$$
\underbrace{{}^{S_i(t_k)}_{\mathcal{E}}\mathbf{q}}_{\text{9D estimate}} = \underbrace{\overbrace{\left(\delta_i(t_k) \ @ \ [0 \ 0 \ 1]^\mathsf{T} \right)}^{\text{magnetometer correction}}}_{{}^{\mathcal{E}_i(t_k)}_{\mathcal{E}}\mathbf{q}} \otimes \underbrace{\overbrace{{}^{\mathcal{I}_i(t_k)}_{\mathcal{E}_i(t_k)}\mathbf{q}}^{\text{accelerometer correction}} \otimes \overbrace{{}^{S_i(t_k)}_{\mathcal{I}_i(t_k)}\mathbf{q}}^{\text{strapdown integration}}}_{\text{6D estimate}} . \tag{1}
$$

The introduced auxiliary coordinate system $\mathcal{I}_i(t_k)$, with $\mathcal{I}_i(t_0) = S_i(t_0)$, is an almost-inertial frame that slowly drifts around arbitrary axes due to errors in gyroscope strapdown integration. See Fig. 2 for an illustration of the four distinct coordinate systems that are used in (1).

As demonstrated in [14], a drawback of many existing methods is that magnetic disturbances can severely impact the inclination estimates. While previous methods [13,14] have already ensured that the magnetometer correction can only influence the heading but not the inclination, the proposed modular state representation makes this property very explicit by representing the heading offset with a scalar variable $\delta_i(t_k)$ and facilitates simultaneous 6D and 9D orientation estimation.

The chosen approach of separating strapdown integration, inclination correction, and heading correction in the state is also represented in the filter structure as shown in Fig. 3. Unlike conventional methods, the correction steps are decoupled from the previous steps, i.e., there is no feedback loop from the heading correction to the strapdown integration and, therefore, neither to the inclination correction.

This basic filter structure is extended by an optional gyroscope bias estimation algorithm and an algorithm for magnetic disturbance detection and rejection. The bias estimation algorithm includes a rest detection and automatically adjusts to whether the IMU is at rest or in motion. The extended filter structure is shown in Fig. 4. Note that it is also possible, and supported by the reference implementation (Section 2.7), to independently enable or disable rest bias estimation, motion bias estimation, and magnetic disturbance rejection.

In the following, we call the extended algorithm *VQF* (Versatile Quaternion-based Filter) and the basic version *BasicVQF*. Furthermore, we introduce an acausal implementation called *OfflineVQF* in Section 2.6.

2.3. Fusion of gyroscope, accelerometer, and magnetometer measurements

The basic filter update consists of gyroscope-based prediction, followed by accelerometer correction and, in 9D IOE, by magnetometer correction. The algorithm is given in Algorithm 1, and details on each step are given in Appendix A. Gyroscope prediction is performed via strapdown integration of the measured angular rate. Errors due to gyroscope bias, noise, and other measurement errors lead to slow drift of the $\mathcal{I}_i$ frame.

To obtain a vertical reference, we transform the measured accelerations into the almost-inertial frame $\mathcal{I}_i$ and then apply a second-order Butterworth low-pass filter to each component. This low-pass filter effectively averages the accelerometer measurements and allows for short-term acceleration and deceleration to cancel out, as illustrated in Fig. 5. The inclination of the orientation estimate is then corrected so that the filtered acceleration points in upward direction. In contrast to conventional IOE algorithms, which typically regard each single accelerometer sample as a 3D vector and perform a nonlinear correction step based on the comparison of this vector to a vertical reference vector, the use of a linear low-pass filter in the $\mathcal{I}_i$ frame more effectively and robustly separates the gravitational acceleration component from the acceleration caused by velocity changes.

Conventional filter structure:
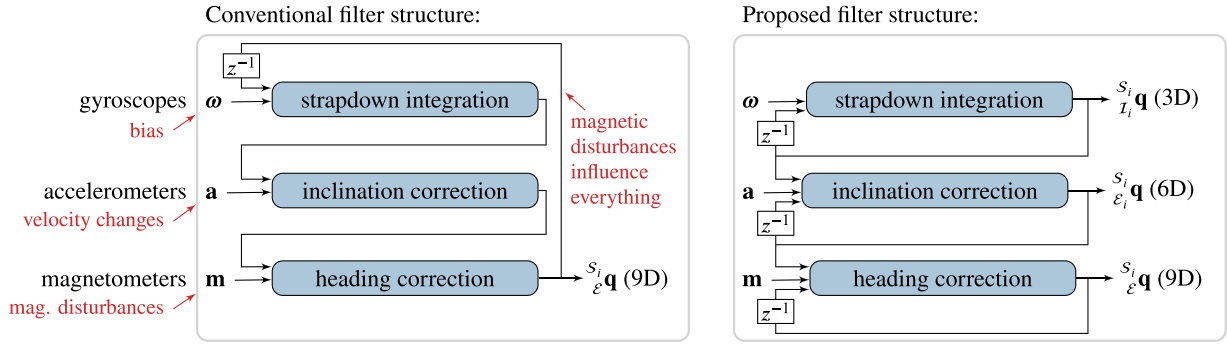
Proposed filter structure:

Fig. 3. Illustration of conventional and proposed filter structures (z^{-1} denotes the unit delay). The proposed filter structure avoids the feedback of the 9D estimate on the strapdown integration. It thereby enables simultaneous 6D and 9D orientation estimation and ensures that the inclination cannot be influenced by magnetic disturbances.
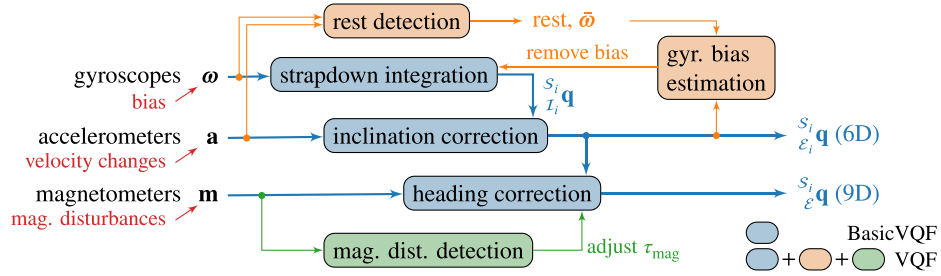

Fig. 4. Variants of the proposed algorithm. *BasicVQF* consists of strapdown integration, inclination correction, and heading correction. The full version *VQF* additionally includes rest detection, gyroscope bias estimation, and magnetic disturbance rejection (which can be enabled or disabled independently).
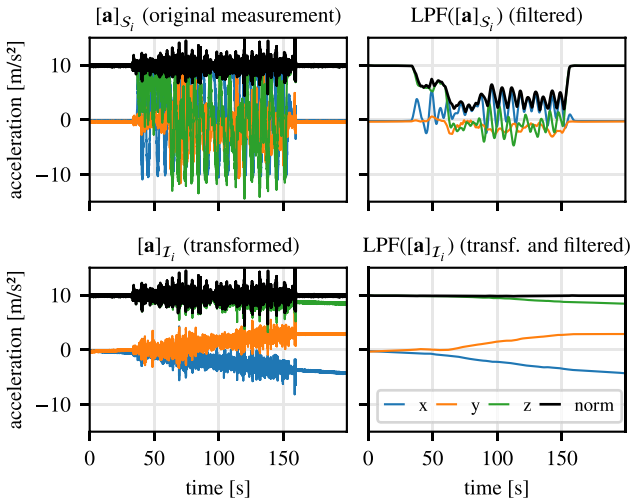

Fig. 5. Example of unfiltered and low-pass filtered accelerations in the original sensor frame S_i and the almost-inertial frame $\mathcal{I}_i$. Applying a low-pass filter with a low cutoff frequency to each component in the sensor frame does not give a meaningful output. In the $\mathcal{I}_i$ frame, low-pass filtering each component of the acceleration effectively averages the measurement, allowing for acceleration and deceleration to cancel out, and the result shows the drift of the $\mathcal{I}_i$ frame due to errors in gyroscope integration.

If magnetometer measurements are given, a heading offset is derived from the projection of the magnetic field vector into the horizontal plane and tracked via an exponential filter.

As it is common in IOE algorithms, the behavior can be influenced by fusion weights that balance between rejecting gyroscope drift and rejecting disturbances in the accelerometer and magnetometer measurements. In Algorithm 1, those parameters are the cut-off frequency $f_{c,acc}$ and the magnetometer correction gain k_{mag}. Like for many existing methods, the meaning of the values assigned to those parameters is

hard to interpret, depends on the sampling time (for k_{mag}), and does not allow for a comparison between the trust assigned to the accelerometer and the trust assigned to the magnetometer. To provide a more intuitive parametrization, we replace those parameters with time constants τ_{acc} and τ_{mag} that can be changed by the user to influence the algorithm behavior. A small time constant leads to fast correction and indicates high trust in the accelerometer or magnetometer measurements, while large time constants indicate trust in the gyroscope measurements.

Those time constants map to the internal values as follows (see Appendix A and Fig. 17 for more information). For the magnetometer correction first-order exponential filter, we use the time constant that is commonly used to characterize first-order systems and corresponds to the time needed for the step response to reach $1 - e^{-1} \approx 63.2\%$ of its final value. To ensure similar behavior for the second-order Butterworth filter of the accelerometer correction, we use a time constant that corresponds to the undamped part of the step response. This leads to the mapping

$$f_{c,acc} = \frac{\sqrt{2}}{2\pi\tau_{acc}}, \quad k_{mag} = 1 - \exp\left(-\frac{T_s}{\tau_{mag}}\right), \quad (2)$$

which allows us to derive the internal parameters $f_{c,acc}$ and k_{mag} from the user-specified time constants τ_{acc} and τ_{mag}. The same parametrization via time constants is also used for the other first-order and second-order filters introduced in the following subsections. Note that we will later determine default values for τ_{acc} and τ_{mag} that yield excellent out-of-the-box accuracy for a large range of application scenarios, and manual tuning by adjusting these time constants is therefore only required in rare edge cases.

2.4. Gyroscope bias estimation

To ensure high accuracy in the presence of gyroscope bias, we extend the BasicVQF algorithm from Section 2.3 by a method to estimate and compensate such bias. In existing IOE algorithms, this is commonly realized by integral action [14,16,17]. However, this approach requires feedback of both accelerometer and magnetometer
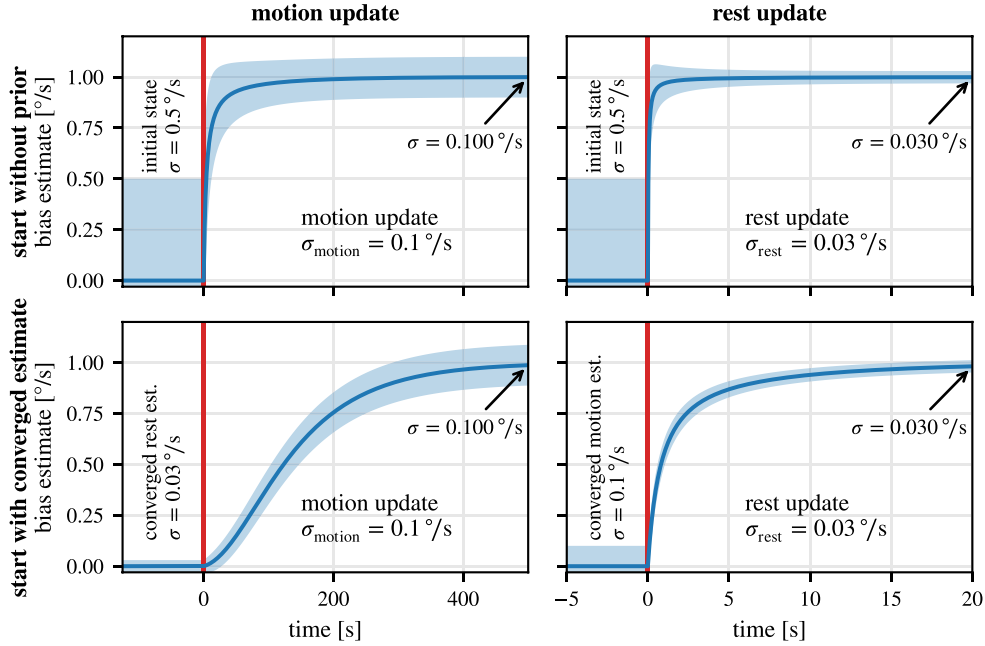
Fig. 6. Step response for the Kalman filter with the proposed parametrization. The bands show the standard deviation σ of the estimate. At $t < 0$, the Kalman filter either starts in the initial state ($\sigma = 0.5°/s$) or in a converged state of either the motion or rest update with an estimate of $0°/s$. At $t = 0$, the measurement changes to $1°/s$. Directly after initialization, the filter converges much faster than in cases where a (contradictory) previous estimate was already obtained. In general, the motion update converges much slower than the rest update.

Algorithm 1 BasicVQF

1: **procedure** INITIALIZEFILTER
2:     $_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q} \leftarrow [1\ 0\ 0\ 0]^\intercal$ ▷ Gyroscope strapdown integration quaternion
3:     $_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q} \leftarrow [1\ 0\ 0\ 0]^\intercal$ ▷ Accelerometer correction quaternion
4:     $\delta_i \leftarrow 0$ ▷ Magnetometer correction angle
5:     initialize low-pass filter state
6: **end procedure**
7: **procedure** FILTERUPDATE($\boldsymbol{\omega} = \boldsymbol{\omega}(t_k), \mathbf{a} = \mathbf{a}(t_k), \mathbf{m} = \mathbf{m}(t_k), f_{c,acc}, k_{mag}, T_s$)
8:     $_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q} \leftarrow {}_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q} \otimes \left(T_s \|\boldsymbol{\omega}\| \ @ \ \boldsymbol{\omega}\right)$ ▷ Perform gyroscope strapdown integration
9:     $[\mathbf{a}]_{\mathcal{I}_i} \leftarrow {}_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q} \otimes \mathbf{a} \otimes {}_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q}^{-1}$ ▷ Transform acceleration to $\mathcal{I}_i$ frame
10:     $[\mathbf{a}_{LP}]_{\mathcal{I}_i} \leftarrow \text{lpfStep}([\mathbf{a}]_{\mathcal{I}_i}, f_c = f_{c,acc})$ ▷ Apply low-pass filter
11:     $[\mathbf{a}_{LP}]_{\mathcal{E}_i} \leftarrow {}_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q} \otimes [\mathbf{a}_{LP}]_{\mathcal{I}_i} \otimes {}_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q}^{-1}$ ▷ Transform to $\mathcal{E}_i$ frame
12:     $[a_x\ a_y\ a_z]^\intercal \leftarrow \frac{[\mathbf{a}_{LP}]_{\mathcal{E}_i}}{\|[\mathbf{a}_{LP}]_{\mathcal{E}_i}\|}$ ▷ Normalize
13:     $q_w \leftarrow \sqrt{\frac{a_z+1}{2}}$
14:     $_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q} \leftarrow [q_w\ \frac{a_y}{2q_w}\ \frac{-a_x}{2q_w}\ 0]^\intercal \otimes {}_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q}$ ▷ Update correction quaternion
15:     $_{\mathcal{E}_i}^{\mathcal{S}_i}\mathbf{q} \leftarrow {}_{\mathcal{E}_i}^{\mathcal{I}_i}\mathbf{q} \otimes {}_{\mathcal{I}_i}^{\mathcal{S}_i}\mathbf{q}$ ▷ Calculate 6D orientation estimate
16:     **if** $\mathbf{m}$ is given **then**
17:         $[m_x\ m_y\ m_z]^\intercal \leftarrow {}_{\mathcal{E}_i}^{\mathcal{S}_i}\mathbf{q} \otimes \mathbf{m} \otimes {}_{\mathcal{E}_i}^{\mathcal{S}_i}\mathbf{q}^{-1}$ ▷ Transform mag. sample to $\mathcal{E}_i$ frame
18:         $\delta_{mag} \leftarrow \text{atan2}(m_x, m_y)$ ▷ Calculate heading offset from mag. sample
19:         $\delta_i \leftarrow \delta_i + k_{mag}\,\text{wrapToPi}(\delta_{mag} - \delta_i)$ ▷ Update correction angle
20:     **end if**
21:     $_{\mathcal{E}}^{\mathcal{S}_i}\mathbf{q} \leftarrow [\cos\frac{\delta_i}{2}\ 0\ 0\ \sin\frac{\delta_i}{2}]^\intercal \otimes {}_{\mathcal{E}_i}^{\mathcal{S}_i}\mathbf{q}$ ▷ Calculate 9D orientation estimate
22:     **return** $_{\mathcal{E}_i}^{\mathcal{S}_i}\mathbf{q}, {}_{\mathcal{E}}^{\mathcal{S}_i}\mathbf{q}$ ▷ Provide 6D and 9D orientation estimate
23: **end procedure**

lpfStep: update step of second-order Butterworth low-pass filter with cutoff frequency f_c

wrapToPi: bring angle into the interval $[-\pi, \pi]$ by adding integer multiples of 2π

---

correction [17], making the gyroscope bias estimate susceptible to magnetic disturbances. To prevent this, the proposed method for gyroscope bias estimation avoids using any information from the magnetometer correction. Instead, the bias is estimated solely from the disagreement between strapdown integration and accelerometer measurements during motion.

Moreover, note that, in many application scenarios, the IMU will occasionally be at rest for several seconds, and those phases can be detected reliably. This is leveraged by the proposed bias estimation algorithm, which determines the bias directly from low-pass-filtered gyroscope measurements whenever it detects a period of rest.

As further detailed in Appendix C and Algorithm 2, the bias estimation is realized via a single Kalman filter that performs different updates based on whether the IMU is currently at rest or in motion. This Kalman filter is parametrized in a way that is independent of the sampling rate and that gives much larger trust to the rest-based updates than to the motion-based updates. In each sampling step, the current Kalman filter estimate of the bias is subtracted from the gyroscope measurement before strapdown integration.

Fig. 6 shows how the Kalman filter behaves with the proposed parametrization. The relation between the uncertainty of the measurement (large during motion, small during rest) and the uncertainty of the current estimate determines how fast the bias estimation converges. After initialization without prior knowledge, the estimation uncertainty is large, which leads to fast convergence. From a converged estimate during motion (i.e., with medium uncertainty), a contradicting but much more reliable observation during rest is adopted within several seconds. From a converged rest estimate, a contradicting observation during motion is adopted much slower due to the larger uncertainty of the measurement.

2.5. Magnetic disturbance rejection

We extend the proposed IOE algorithm with a set of methods that enable adaptive filtering of the magnetometer measurements with the aim of reducing the influence of temporary magnetic disturbances. The employed strategy is composed of three parts: magnetic disturbance

Table 1
Inertial orientation estimation algorithms used in the evaluation.

Algorithm		6D	9D	Bias	Source
VQF	Proposed method	✓	✓	✓	https://github.com/dlaidig/vqf
MAH	Mahony et al. [16]	✓	✓	✓	https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/
MAD	Madgwick [17]	✓	✓	✗	https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/
VAC	Valenti et al. [13]	✓	✓	✓	https://wiki.ros.org/imu_complementary_filter
FKF	Guo et al. [22]	✗	✓	✗	https://github.com/zarathustr/FKF
SEL	Seel and Ruppin [14]	✓	✓	✓	https://github.com/dlaidig/qmt, qmt.oriEstIMU
MKF	Matlab	✓	✓	✓	Matlab R2021b (The MathWorks Inc., Natick, MA, USA), imufilter/ahrsfilter
KOK	Kok and Schön [15]	✗	✓	✓	https://github.com/manonkok/fastRobustOriEst
RIANN	Weber et al. [23]	✓	✗	✗	https://github.com/daniel-om-weber/riann

detection, magnetic disturbance *rejection*, and new magnetic field *acceptance*. All three parts are briefly explained below, and the complete algorithm is described in detail in Algorithm 3 in Appendix F.

The magnetic disturbance *detection* uses a user-defined or automatically determined reference for the norm and dip angle of the local magnetic field. The magnetometer measurements are considered to be undisturbed only if they have been close to the reference for at least $0.5\,\text{s}$ – and disturbed otherwise. Whenever the magnetic field is considered undisturbed, the reference values are slowly updated to track slow changes in the norm and dip angle.

The magnetic disturbance *rejection* adjusts the speed of the first-order filter used for heading correction if magnetic disturbances are detected. For disturbances of up to $60\,\text{s}$, the magnetometer update is fully disabled. For longer periods that are considered to be disturbed, updates are performed but at a lower speed.

Finally, the new magnetic field *acceptance* is used to deal with sudden changes in the environment, e.g., after changing the terrain from outdoor to indoor or moving to a different indoor room with a different local magnetic field. Whenever the magnetic field is considered to be disturbed but seems homogeneous (for a sufficiently long time during which the IMU was not stationary), the norm and dip angle of the current magnetic field is used as the new reference.

2.6. Acausal filtering for offline data processing

In application scenarios in which the complete time series of recorded data is available (offline data processing), we can employ acausal signal processing methods to further improve accuracy. As commonly done in signal processing for zero-phase filtering [24], we first run the filtering steps forward and then again backward in time. This allows us to leverage the existing real-time implementation to create the offline variant detailed in Algorithm 4 in Appendix G.

2.7. Open-source implementation

Implementations of the proposed orientation estimation algorithm are available at https://github.com/dlaidig/vqf under the MIT license. Native implementations are provided in C++, Python, and Matlab. Furthermore, the fast C++ implementation can easily be used from Python code. The Python package is available at https://pypi.org/project/vqf/ and can be installed via pip, and documentation is available at https://vqf.readthedocs.io/.

3. Evaluation

In this section, we evaluate the performance of the proposed IOE algorithm on six publicly available datasets and compare the results obtained with the proposed method to results obtained with eight state-of-the-art IOE algorithms.

3.1. Algorithms and datasets

Table 1 lists the algorithms used for the evaluation. This table also shows whether the algorithm can be used for 6D and/or 9D orientation estimation, whether it supports gyroscope bias estimation, and where the implementation is available.

To evaluate the accuracy of IOE, we consider publicly available datasets consisting of IMU measurements and a ground truth orientation obtained from marker-based optical motion capture (OMC). In [19], we presented the Berlin Robust Orientation Estimation Assessment Dataset (BROAD) and briefly reviewed other existing datasets that contain trial data that is suitable for IOE accuracy evaluation. In the present evaluation, we use all of these datasets, i.e.,

- BROAD [19]: 39 trials (23 undisturbed trials with different motion types and speeds and 16 trials with various deliberate disturbances)
- Sassari [18]: 18 trials (3 speeds, 3 IMU models, and 2 IMUs of each model)
- RepoIMU [25]: 21 trials (*T-Stick* only; test 5, test 6 trial 1, test 10 excluded due to artifacts, as explained in [19])
- OxIOD [26]: 71 trials (only *handbag, handheld, pocket, running, slow walking, trolley* trials)
- TUM VI [27]: 6 trials (*room* only; no magnetometer data)
- EuRoC MAV [28]: 6 trials (*Vicon room* only; no magnetometer data).

Combined, the collection of evaluation data consists of 161 trials with a total duration of $12.9\,\text{h}$. The data includes motions of handheld IMUs (various combinations of fast and slow rotations and translations), walking and running, as well as flight data from a micro aerial vehicle, and contains data from eight different IMU models, recorded at sampling rates ranging from $100\,\text{Hz}$ to $286\,\text{Hz}$. For more information about the datasets, refer to the respective publications and the summary provided in [19]. This large collection of experimental data allows us to evaluate the robustness of the proposed method for different motion characteristics, different sensor hardware, and different sampling rates.

3.2. Algorithm parametrization

Before assessing the IOE accuracy, we need to determine suitable tuning parameters for each algorithm. In [19], we introduced a metric to assess the performance of an IOE algorithm, the trial-agnostic generalized performance (TAGP). This metric is defined as the smallest possible root mean square error (RMSE), averaged over all 39 trials of the BROAD dataset, that can be obtained with a common algorithm parametrization for all trials. The associated parametrization can then be expected to provide good results for a wide variety of motions and disturbance scenarios.

However, one limitation of the BROAD dataset is that all trials are performed with the same IMU model. To find parameters that are not only robust against movement speed, type of motion, and various disturbances, but also work well for different sensor characteristics, we define an extended TAGP metric, the TAGP_x.
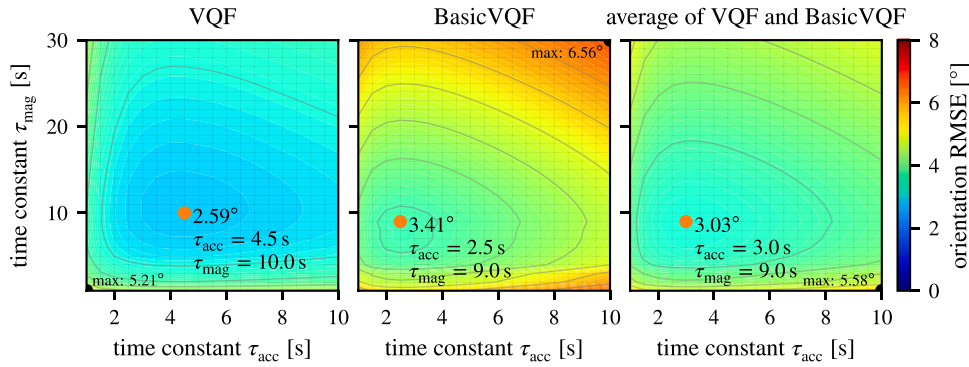
Fig. 7. RMSE (weighted average over all datasets) achieved with the proposed VQF algorithm and with the reduced BasicVQF variant, for different values of the tuning parameters. The default algorithm parameters are chosen such that the mean of both errors is minimized, i.e., $\tau_{acc} = 3\,s$ and $\tau_{mag} = 9\,s$.
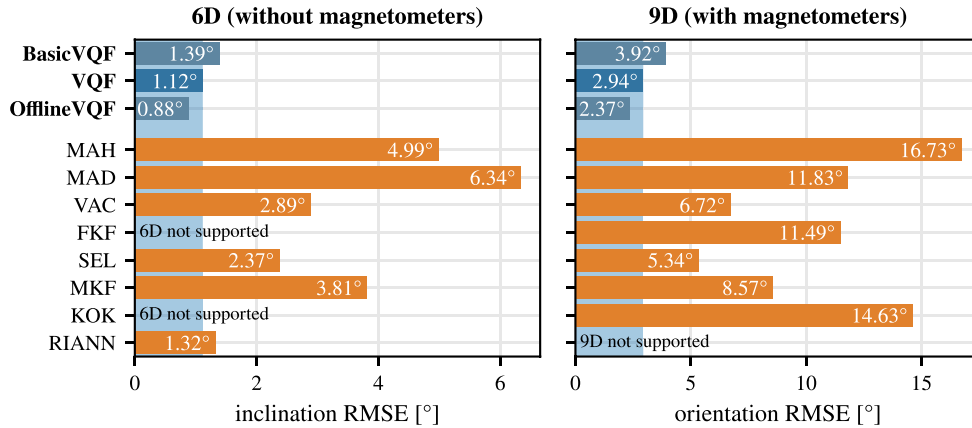

Fig. 8. RMSE (weighted average over all datasets) for the proposed VQF and all state-of-the-art algorithms. VQF outperforms all eight literature methods, and only RIANN provides similar 6D performance. Even the errors obtained with the simple BasicVQF variant are clearly lower than for the other seven algorithms. When real-time capability is not required, using the offline variant is advised to further increase the accuracy of the orientation estimates.

Similar to the TAGP, the TAGP$_x$ is the smallest possible RMSE, averaged over the aforementioned trials for each of the six datasets. The errors are first averaged by dataset, where the dedicated benchmark dataset BROAD is given a five times larger weight than all other datasets, which are weighted equally. For trials without magnetometer data (and for RIANN, which does not support magnetometers), the inclination error, as defined in [19], is used instead of the orientation error.

Fig. 7 shows how the weighted error defined above depends on the tuning parameters for the default and the basic variant of the proposed VQF algorithm. The TAGP$_x$ is the minimum value of this error, i.e., 2.59° for VQF and 3.41° for BasicVQF, which shows that gyroscope bias estimation and magnetic disturbance rejection lead to improved accuracy. However, the optimal values for the time constants τ_{acc} and τ_{mag} are similar for both variants. To avoid specifying different default parameters for VQF and BasicVQF, we simply use the average of the errors obtained with both variants to determine the default values $\tau_{acc} = 3\,s$ and $\tau_{acc} = 9\,s$.

To provide a fair comparison between the proposed and the state-of-the-art methods, we also optimize the parameters for all other algorithms according to the TAGP$_x$, i.e., we find the parameters that allow each algorithm to provide the best possible performance across all datasets. Details on the employed search strategy as well as the resulting parameters are given in Appendix H. In the following, we always use those optimal parameters to evaluate and compare algorithm performance.

3.3. Orientation estimation accuracy

To assess the performance of all algorithms, we apply each to the data of all trials of all datasets. In the case of 6D (magnetometer-free) sensor fusion, we determine the inclination error [19], and for 9D sensor fusion the orientation error, in each case between the IMU-based estimate and the OMC ground truth. We then calculate the RMSE while only considering the motion phases. Since the TUM VI and EuRoC MAV datasets do not include magnetometer data, they are only considered for the 6D results. We average the errors by dataset analogously to the definition of the TAGP$_x$ in Section 3.2. The resulting values are presented for all algorithms in Fig. 8.

One main observation from this figure is that the proposed method VQF consistently provides considerably lower errors than the existing orientation estimation algorithms. For 9D IOE, there is a 1.8-fold to 5-fold increase in accuracy, while for 6D IOE, the increase is 17% for RIANN and between 2.1-fold and 5-fold for the other methods. It should be noted that the only algorithm that achieves similar inclination errors, the neural network RIANN, cannot perform 9D sensor fusion and was trained on some of the datasets that are used in this evaluation (see [23] for details).

Fig. 8 also allows us to compare the variants of the proposed method, cf. Fig. 4. Unsurprisingly, the errors obtained with the BasicVQF variant (no bias estimation and no magnetic disturbance rejection) are slightly larger. However, with the aforementioned exception of RIANN, even the 6D and 9D errors of BasicVQF are still clearly lower than the corresponding errors obtained with any of the existing algorithms. The low 6D errors of BasicVQF can directly be attributed to the novel approach for inclination correction, while the comparatively
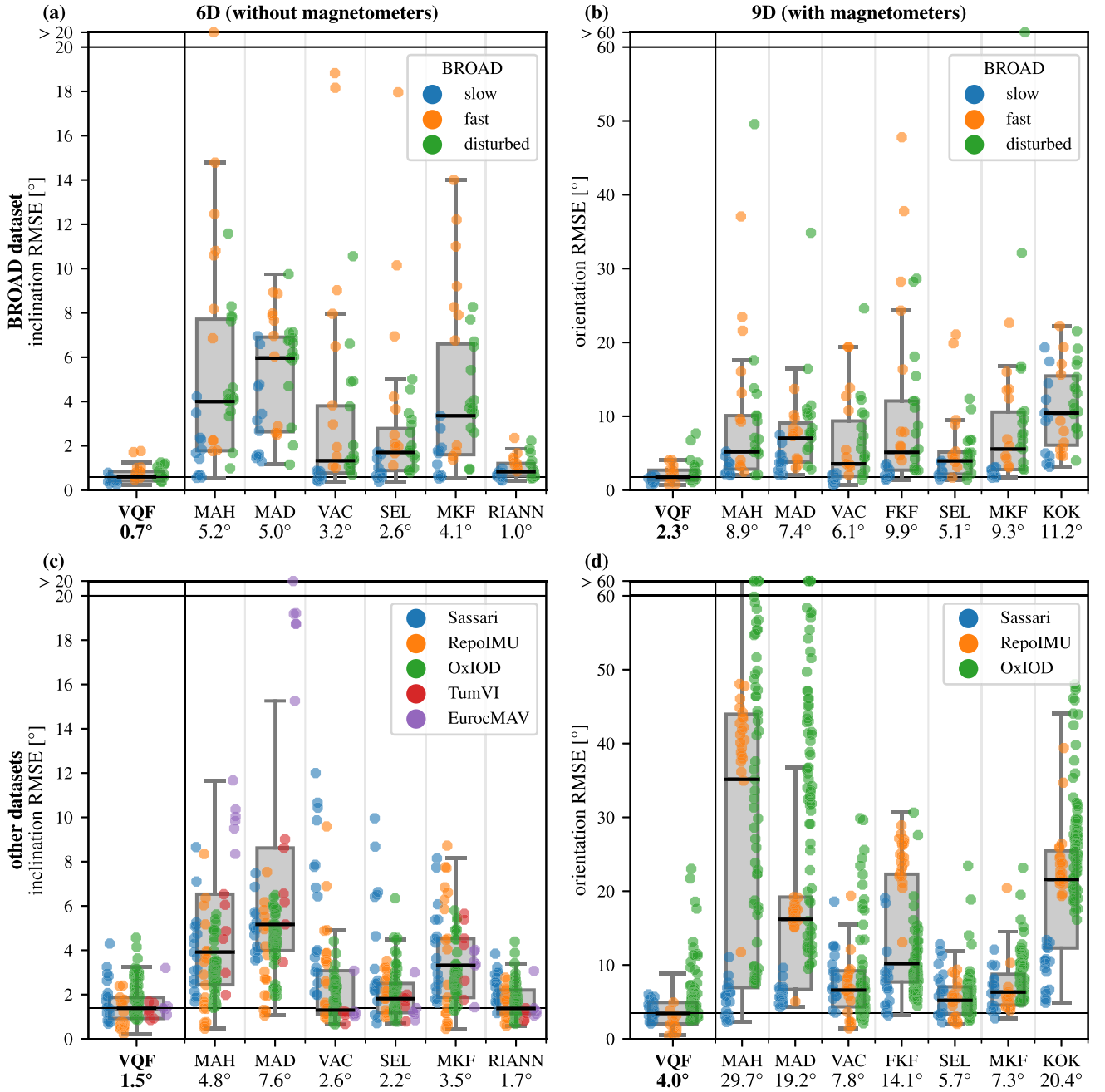
Fig. 9. Orientation estimation errors for all evaluated algorithms and for all trials of the **(a,b)** BROAD dataset and **(c,d)** the five other datasets and for **(a,c)** 6D and **(b,d)** 9D sensor fusion. The numbers below the algorithm names indicate the RMSE averaged over all trials. For **(b,d)**, the boxplots and average values are weighted to give each dataset the same weight regardless of the number of trials. The proposed VQF algorithm consistently provides the best performance.

small difference between 6D and 9D errors can be attributed to the decoupled filter structure and modular state representation. Compared to the real-time-capable implementation, the offline variant OfflineVQF is able to further increase the estimation accuracy by another 20 percent. Therefore, employing this variant is advisable when analyzing recorded data.

While Fig. 8 shows a clear improvement in accuracy with respect to the state of the art when looking at errors averaged over a large number of trials, a comprehensive comparison should also include a closer look at individual trials. Fig. 9 differentiates the errors by dataset and shows a marker for each single trial. Comparing medians, the interquartile ranges, the lengths of the whiskers, or the distributions of outliers yields the same conclusion: The proposed method not only performs better on

average but consistently and robustly provides lower errors than the existing methods.

It is noticeable that unusually large errors of ∼ 23° are observed for some trials of the OxIOD dataset. As those large errors are observed across all methods, the most likely cause of those errors are irregularities in the measurement or ground truth data that have previously been noted in [19].

Beyond the level of comparing performance on different datasets, the BROAD dataset facilitates the investigation of algorithm performance for seven different motion characteristics and six different disturbance characteristics. Fig. 10 compares the average RMSE across different groups of trials and compares the proposed VQF method in each case with the best of the other 9D-capable algorithms, i.e., the
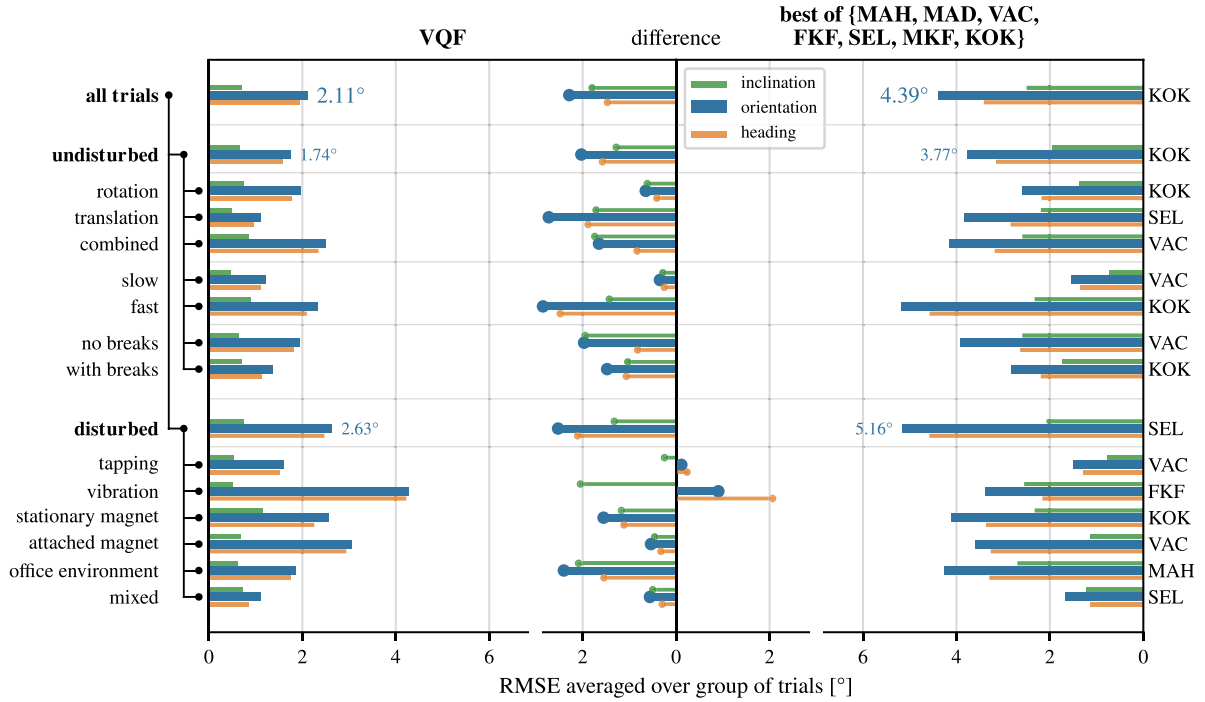
Fig. 10. Averaged RMSE errors for various groups of trials of the BROAD dataset. The proposed algorithm VQF is compared with the best of the seven other evaluated algorithms, i.e., the algorithm that provides the lowest orientation error for the respective group of trials. The lines originating from the center highlight the difference between the errors. For all groups except for the *tapping* and *vibration* trials, the proposed algorithm outperforms even the best-performing literature method.
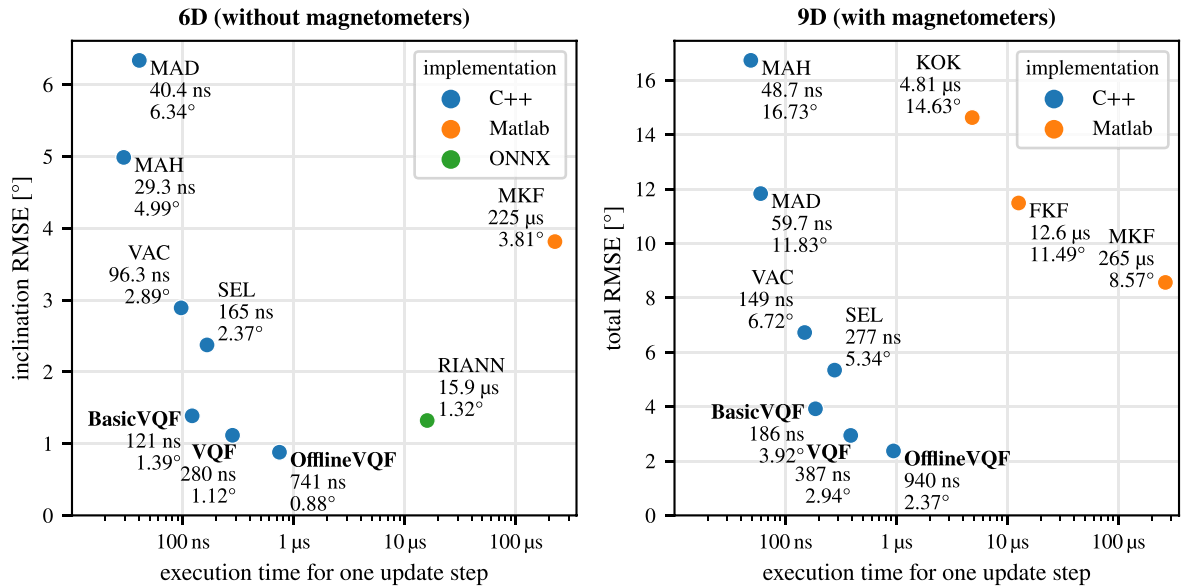

Fig. 11. Execution time for one update step on an AMD Ryzen 5 3600 CPU vs. orientation estimation RMSE (weighted average over all datasets) for the proposed VQF and all state-of-the-art algorithms. Accuracies vary largely, and execution times mainly depend on the programming language used for implementing the algorithm. The execution time of VQF is in the same order of magnitude as for the other algorithms implemented in C++, while the errors are clearly smaller.

algorithm that achieves the lowest errors for the respective group of trials. Except for the *tapping* and *vibration* groups, VQF always achieves lower errors than even the best of the other algorithms. For the *vibration* group, it is worth noticing that, while the orientation error is slightly larger than the error obtained with FKF, the inclination errors obtained with VQF are clearly lower.

In summary, compared with eight other IOE algorithms and using a collection of six publicly available datasets that cover a wide range of motions, speeds, disturbances, and different sensor hardware, the proposed method VQF consistently provides the best IOE accuracy, both for 6D and 9D orientation estimation.

3.4. Algorithm execution time

In addition to accuracy, the execution time of an IOE algorithm is often relevant, especially in real-time applications or when the algorithm is running on low-powered microcontrollers directly on the IMU. To compare the execution times, we repeatedly process the entire BROAD dataset with all algorithms on an AMD Ryzen 5 3600 CPU while measuring the execution time. Fig. 11 shows the average execution time for one update step in combination with the orientation estimation error for the respective algorithm. The results show that execution time mostly depends on the programming language used

	BROAD	true bias: 0.332 °/s	BROAD (cut)	true bias: 0.332 °/s	Sassari	true bias: 0.855 °/s	Parameters
VQF	90 %	0.033 °/s	39 %	0.204 °/s	89 %	0.092 °/s	default
OfflineVQF	97 %	0.008 °/s	58 %	0.138 °/s	96 %	0.037 °/s	default
MAH	7 %	0.309 °/s	1 %	0.329 °/s	30 %	0.601 °/s	optimized
MAD	17 %	0.276 °/s	3 %	0.321 °/s	33 %	0.572 °/s	optimized
VAC	79 %	0.069 °/s	18 %	0.274 °/s	39 %	0.522 °/s	optimized
SEL	51 %	0.162 °/s	6 %	0.312 °/s	67 %	0.280 °/s	optimized
MKF	0 %	1.021 °/s	0 %	1.094 °/s	10 %	0.773 °/s	$TAGP_x$
KOK	15 %	0.283 °/s	3 %	0.323 °/s	34 %	0.567 °/s	optimized
	bias reduction	residual bias	bias reduction	residual bias	bias reduction	residual bias	

Fig. 12. Bias estimation results for different IOE algorithms. The bars and percentages indicate the reduction of the bias norm with respect to the true bias contained in the measurement data. For all datasets, VQF with default parameters surpasses the best possible performance of the existing algorithms. Using the OfflineVQF variant further improves accuracy.
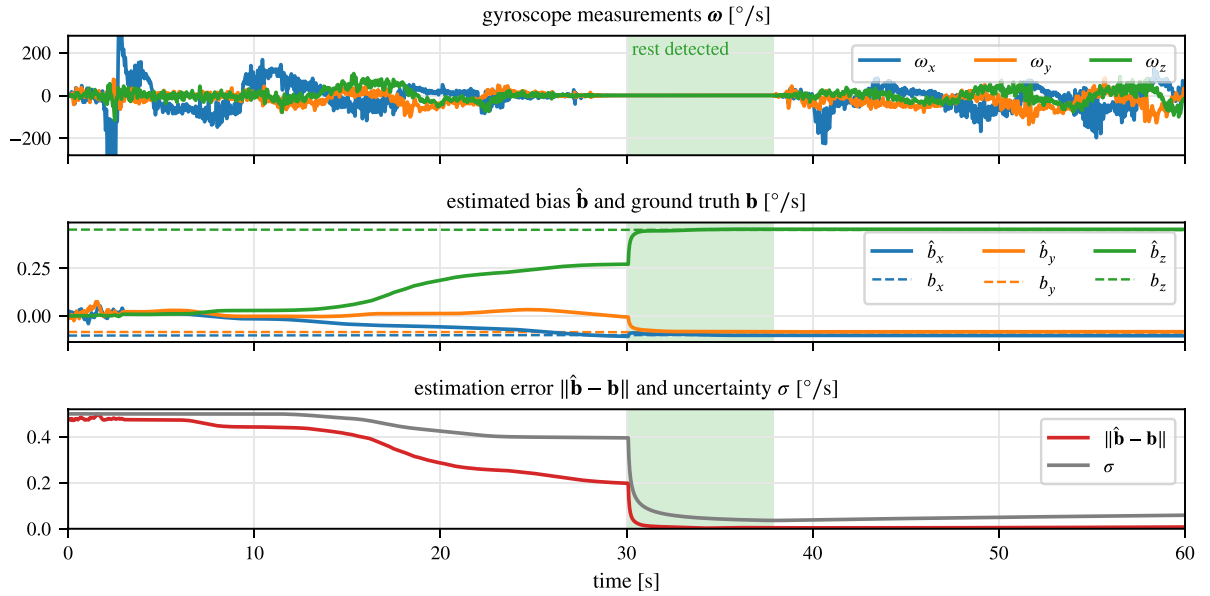

Fig. 13. Behavior of the bias estimation method for the first 60 s of trial 4 of the BROAD dataset (undisturbed slow rotation with breaks, initial rest cut). During motion, the estimated bias slowly converges toward the true value. Once rest is automatically detected by Algorithm 2, the convergence speed increases, and the error and the estimation uncertainty suddenly drop.

for the implementation and that the algorithms written in C++ are considerably faster than the algorithms written in Matlab or using the ONNX machine learning runtime. While the VQF algorithms achieve clearly higher accuracy, the execution times are in the same order of magnitude as for the existing state-of-the-art methods with a C++ implementation. VQF is fast enough for use on microcontrollers, which we verified by integrating it into an IMU firmware running on a Cortex M4 at a comparatively high sampling rate of 1600 Hz.

3.5. Gyroscope bias estimation

We now investigate the performance of the gyroscope bias estimation method. Besides the proposed algorithm, we also evaluate the performance of the five other algorithms that are able to estimate gyroscope bias.

For the BROAD and Sassari datasets, we derive a ground truth for the gyroscope bias by averaging the gyroscope measurements during the rest phases at the beginning and at the end of each trial and linearly interpolating in between to account for slow bias instability. For each IOE algorithm, we calculate the root-mean-square over time of the residual bias norm, i.e., of the norm of the difference between the estimated bias and the true bias. Fig. 12 shows the achieved relative reduction of the gyroscope bias, the true bias norm, and the

residual bias norm, averaged over all trials for each dataset. Since the proposed VQF and the literature method VAC use rest detection for bias estimation, while the other algorithms do not, we also test the performance of all algorithms on a cut version of the BROAD dataset, in which the initial and final rest phases were removed.

For the proposed algorithms VQF and OfflineVQF, we present the bias estimates obtained with the default parameters. For the existing methods, we found that parameters that yield the best orientation estimation results often do not yield the best bias estimates. We therefore optimized, separately for each dataset, all parameters of all literature methods (except MKF) across the search grid presented in Table 2, such that the bias estimation error is minimized. Despite this disparity, the proposed method clearly outperforms the bias estimation methods of all other IOE algorithms. Even though the results obtained with the cut BROAD dataset are worse than for the datasets with long rest phases, the proposed method is still able to reduce the bias by 39 %, while the best literature method only achieves a reduction of 18 %. As with orientation estimation, using the OfflineVQF variant further improves the accuracy in comparison to the real-time capable VQF algorithm. MKF only achieves a slight reduction of gyroscope bias for the Sassari dataset, while for BROAD the bias norm increases compared to the original bias found in the measurement data.

To illustrate how the bias estimation method works, Fig. 13 shows the estimated bias, the estimation error, and the estimation uncertainty
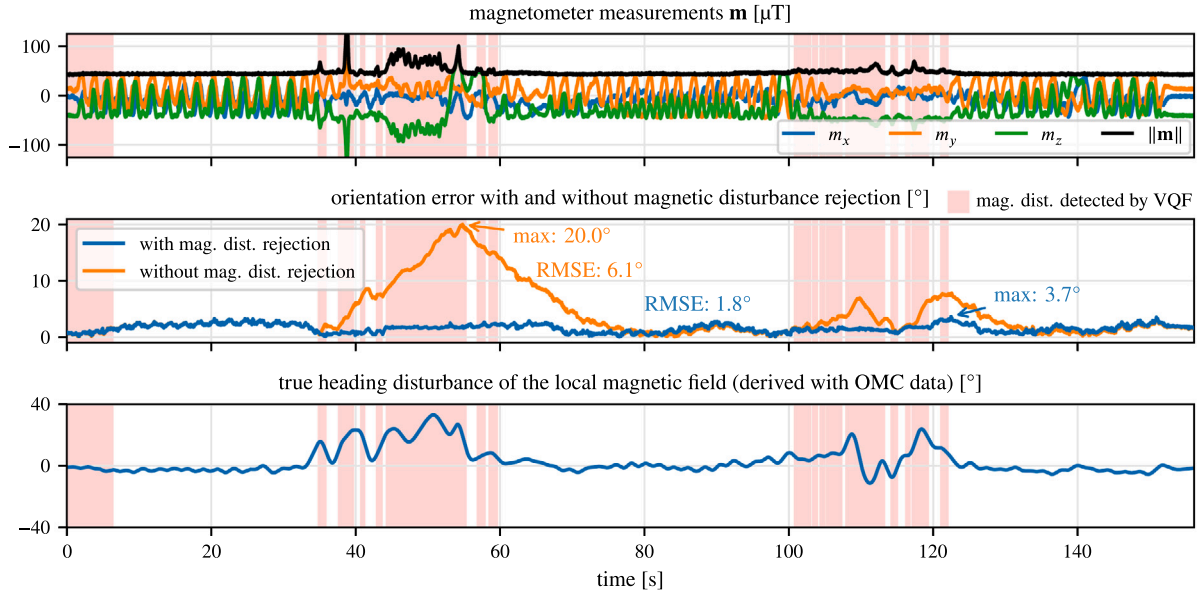
Fig. 14. Performance of the magnetic disturbance rejection method for trial 37 of the BROAD dataset (disturbed office environment). Whenever magnetic disturbances are detected by Algorithm 3 (light red background), the magnetometer-based correction is automatically disabled. In comparison to the same algorithm with disabled magnetic disturbance rejection, the RMSE is considerably reduced, and the large error peak of 20° is avoided.

for an exemplary trial of the BROAD dataset. It can be seen that the bias estimation works as intended: During the initial motion phase, the estimated bias slowly converges to the true bias. Once rest is automatically detected, the alternative update is used, causing the estimate to rapidly converge.

In summary, both in a systematic comparison, as well as in an exemplary case study, the proposed gyroscope bias estimation method was found to work reliably and clearly outperformed existing bias estimation approaches.

3.6. Magnetic disturbance rejection

To further illustrate the difference in performance between the proposed BasicVQF and VQF algorithms, we now take a brief look at the performance of the magnetic disturbance detection and rejection method. Fig. 14 shows the behavior of this extension on an example of the BROAD dataset. For the sake of evaluation, OMC data was used to determine the true disturbance of the local magnetic field caused by ferromagnetic material and electric devices in the office environment. This ground truth information shows that the detection is triggered whenever disturbances are present. Without magnetic disturbance rejection, the orientation estimation error reaches a maximum of 20.0°, and the RMSE is 6.1°. In contrast, enabling magnetic disturbance rejection reduces the maximum error to just 3.7° and the RMSE to 1.8°, which translates to an at least three times better accuracy. This example demonstrates how the optional magnetic disturbance rejection can improve the reliability of 9D IOE in real-world scenarios inside buildings, near ferromagnetic material and electric devices.

3.7. Summary of the results

The proposed VQF algorithm achieved an average RMSE of 2.9° for 9D IOE, while the average errors obtained with state-of-the-art methods range from 5.3° to 16.7°. For 6D IOE, VQF attained an average RMSE of 1.1°, compared to 2.4°–6.3° obtained with existing methods, and it achieved even 17% lower errors than a neural network that was trained on large portions of the benchmark data. For the 13 characteristic trial groups of the BROAD dataset, the proposed method outperforms even the best-performing literature method for all seven motion characteristics and for four out of six disturbance characteristics. Furthermore,

the gyroscope bias estimation of VQF clearly outperformed all existing state-of-the-art literature methods and compensated $\sim 90\%$ of the bias. Even for the challenging case without rest phases, the bias could still be reduced by ~ 40–60%, while the existing algorithms barely achieved any bias reduction. For an exemplary case in a simulated office environment, the magnetic disturbance rejection algorithm was shown to achieve a five-fold reduction of the maximum orientation error.

Even the variant BasicVQF, without bias estimation and magnetic disturbance rejection, was shown to provide clearly more accurate 9D orientation estimates than all state-of-the-art methods. For applications in which real-time capability is not required, the variant OfflineVQF can be used to further increase accuracy.

4. Conclusions

We proposed a novel IOE algorithm that simultaneously performs 6D and 9D sensor fusion, estimates gyroscope bias, and performs magnetic disturbance detection and rejection. An open-source implementation is provided in C++, Python, and Matlab, making it easy to use the algorithm. We compared the proposed algorithm VQF with eight other IOE algorithms and using a collection of six publicly available datasets that cover a wide range of motions, speeds, disturbances, and different sensor hardware. As summarized in Section 3.7, VQF consistently provided the best performance, both for 6D and 9D orientation estimation, as well as for gyroscope bias estimation, and it proved capable of magnetic disturbance rejection.

The proposed method provides a highly accurate out-of-the-box performance, which means that – unlike existing literature methods – VQF requires no parameter tuning for a vast range of motions and application scenarios. For rare edge cases, the proposed method facilitates easy and intuitive tuning via the time constants τ_{acc} and τ_{mag}.

The achieved improvements in ease of use and in orientation estimation accuracy are expected to change the way we use IOE algorithms in practice and, thereby, to advance the broad field of inertial motion tracking since it enables more accurate IMU-based position and velocity estimation, joint angle estimation, and 3D visualization. This, in turn, leads to improved performance in many existing application areas of miniature inertial sensor technology, and it likewise facilitates the applicability in novel application domains with increased accuracy demands.

Future work will focus on integrating continuous and automatic magnetometer calibration as well as on employing the VQF algorithm in various applications.

CRediT authorship contribution statement

Daniel Laidig: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Thomas Seel:** Conceptualization, Methodology, Formal analysis, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Details on the basic update step

The basic filter update given in Algorithm 1 consists of a prediction of the next orientation based on the gyroscope measurement, followed by a correction of the inclination based on the accelerometer measurement, and an (optional) correction of the heading based on the magnetometer measurement.

A.1. Gyroscope prediction

Gyroscope prediction via strapdown integration is performed by multiplying the previous estimate with a quaternion based on the norm and direction of the measured angular rate:

$$\substack{S_i(t_k) \\ I_i(t_k)}\mathbf{q} = \substack{S_i(t_{k-1}) \\ I_i(t_{k-1})}\mathbf{q} \otimes \left(T_s \|\boldsymbol{\omega}\| @ \boldsymbol{\omega} \right). \tag{3}$$

The rotation due to the gyroscope measurement is composed of the true change of sensor orientation and an error, due to gyroscope bias, noise, and other measurement errors (e.g., scaling errors, nonlinearity, misalignment, and clipping). This error can be regarded as a small drift in the I_i frame, i.e., as $\substack{I_i(t_{k-1}) \\ I_i(t_k)}\mathbf{q}$, which can be shown via quaternion algebra (cf. Appendix B).

A.2. Accelerometer correction

The accelerometer measurements consist of the gravitational acceleration, change of velocity, as well as noise, bias, and other measurement errors. Most existing methods [14,16,17] interpret each single accelerometer sample as a 3D vector and use the angle between this vector and the expected vertical direction to derive the correction step. In contrast, to better separate the gravitational acceleration from the other components of the measurement, we transform the measured accelerations into the almost-inertial frame I_i and then apply a linear low-pass filter to each component. The resulting signal provides a vertical reference in the I_i frame, which slowly drifts due to errors in gyroscope integration, as shown in Fig. 5. As a low-pass filter, we use a second-order Butterworth filter. The cutoff frequency $f_{c,acc}$ of this filter defines the weight between gyroscope prediction and accelerometer correction. To ensure a fast and robust convergence when the algorithm is initialized, we calculate the arithmetic mean for the first few samples instead of using the Butterworth filter. Then, the filter state is initialized based on this mean value.

As illustrated in Fig. 15, we can use this vertical reference to correct the inclination estimate. If $[a_x \; a_y \; a_z]^\intercal$ denotes the filtered and normalized acceleration measurement in the $\mathcal{E}_i$ frame, the shortest
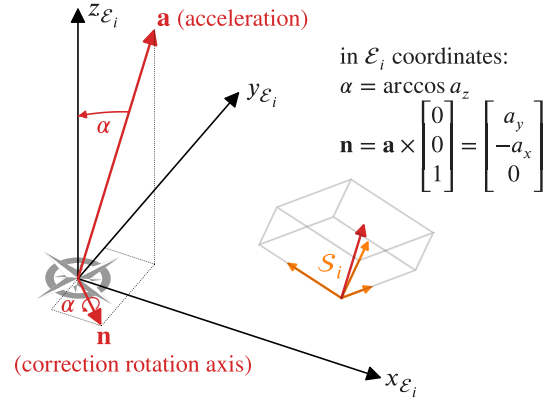

Fig. 15. Illustration of the inclination correction step based on the filtered accelerometer measurement. In global $\mathcal{E}_i$ coordinates, the filtered and normalized acceleration $\mathbf{a} = [a_x \; a_y \; a_z]^\intercal$ is expected to point in positive z-direction. This can be achieved by a correction rotation with angle $\arccos a_z$ and axis $[a_y \; -a_x \; 0]^\intercal$.
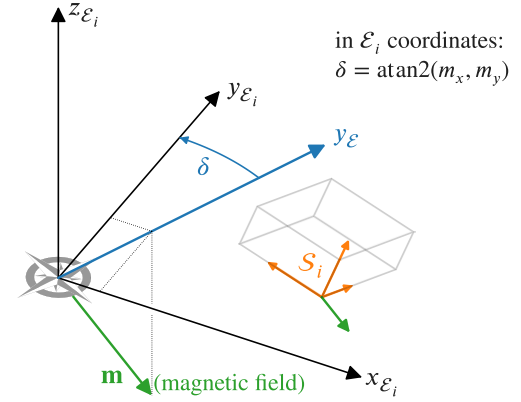
Fig. 16. Illustration of the heading correction step based on the magnetometer measurement. In many parts of the world, Earth's magnetic field is dominated by a vertical component, e.g., in Berlin, the magnetic field is pointing down with a dip angle of 68°. An estimate of the magnetic north direction can be obtained by projecting the measured magnetic field into the horizontal plane.

rotation for inclination correction has an angle of $\arccos a_z$ around the axis $[a_x \; a_y \; a_z]^\intercal \times [0 \; 0 \; 1]^\intercal = [a_y \; -a_x \; 0]^\intercal$. The quaternion $\mathbf{q}_{corr}$ that corresponds to this rotation can be expressed without trigonometric functions as

$$q_w = \cos\left(\frac{\arccos a_z}{2}\right) = \sqrt{\frac{a_z + 1}{2}} \tag{4}$$

$$\mathbf{q}_{corr} = \begin{bmatrix} q_w & \dfrac{a_y}{2q_w} & \dfrac{-a_x}{2q_w} & 0 \end{bmatrix}^\intercal. \tag{5}$$

This quaternion is used to correct the estimate of the $\substack{I_i \\ \mathcal{E}_i}\mathbf{q}$ quaternion, i.e.,

$$\substack{I_i(t_k) \\ \mathcal{E}_i(t_k)}\mathbf{q} = \mathbf{q}_{corr}(t_k) \otimes \substack{I_i(t_{k-1}) \\ \mathcal{E}_i(t_{k-1})}\mathbf{q}. \tag{6}$$

After the correction step, the low-pass filtered acceleration will perfectly point in upward direction, i.e., in z-direction of the $\mathcal{E}_i$ and $\mathcal{E}$ frames.

A.3. Magnetometer correction

If magnetometer measurements are given, we use them to correct the heading estimate. As shown in Eq. (1), the heading is tracked via a scalar state $\delta_i(t_k)$ that represents the vertical rotation from the

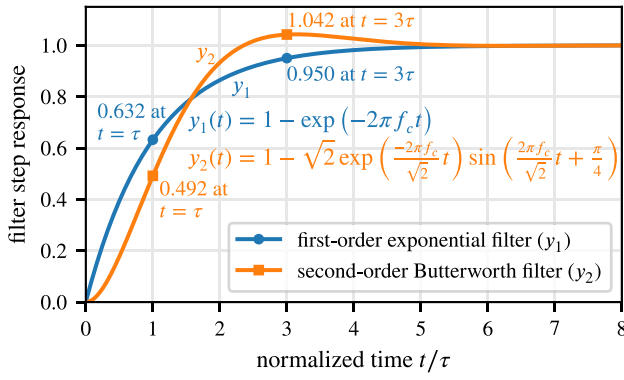**Fig. 17.** Step response for first- and second-order low-pass filters. The time axis is normalized by dividing by the time constant τ of the proposed parametrization. Both filter outputs are roughly close to 0.5 at $t = \tau$ and converge up to a deviation of $\leq 5\,\%$ at $t = 3\tau$.

global $\mathcal{E}$ frame (with the y-axis pointing north) to the $\mathcal{E}_i$ frame. As illustrated in Fig. 16, we can use the current magnetometer sample to derive a measurement $\delta_{\mathrm{mag}}(t_k)$ for this state by projecting the magnetic field vector into the horizontal plane. The state is then corrected by a fixed fraction k_{mag} of the deviation between state and measurement, which corresponds to a first-order low-pass filter with exponential convergence.

The parameter k_{mag} defines the fusion weight between gyroscope prediction and magnetometer correction. To ensure robust and fast convergence when the filter is initialized with the default value $\delta_i(t_0) = 0$, we average the first measurements by choosing the filter weight k_{mag} as $1, \frac{1}{2}, \frac{1}{3}, \ldots$ during the first $\frac{1}{k_{\mathrm{mag}}}$ steps.

A.4. Definition of intuitive fusion weights

The behavior of the algorithm can be influenced by two fusion weights $f_{\mathrm{c,acc}}$ and k_{mag}, which we will now replace with a more intuitive parametrization by defining intuitive time constants τ_{acc} and τ_{mag} that allow the user to influence the fusion weights between gyroscope prediction and accelerometer correction and between gyroscope prediction and magnetometer correction, respectively. The parameters $f_{\mathrm{c,acc}}$ and k_{mag} that are internally used in the filter update step (Algorithm 1) are automatically derived from those time constants.

In order to specify the gain k_{mag} of the magnetometer correction first-order exponential filter, we use the time constant $\tau = \frac{1}{2\pi f_c}$ that is commonly used to characterize first-order systems and corresponds to the time needed for the step response to reach $1 - e^{-1} \approx 63.2\,\%$ of its final value. The filter weight for the proportional update can be derived from this time constant as

$$k_{\mathrm{mag}} = 1 - \exp\left(-\frac{T_{\mathrm{s}}}{\tau_{\mathrm{mag}}}\right). \tag{7}$$

The second-order Butterworth filter used for accelerometer correction is characterized by the cutoff frequency $f_{\mathrm{c,acc}}$. In order to obtain a parametrization that is similar to the parametrization of the magnetometer correction, we use a time constant that corresponds to the undampened part of the step response, i.e., $\tau_{\mathrm{acc}} = \frac{\sqrt{2}}{2\pi f_{\mathrm{c,acc}}}$. The cutoff frequency used to determine the Butterworth filter coefficients is then given as

$$f_{\mathrm{c,acc}} = \frac{\sqrt{2}}{2\pi\tau_{\mathrm{acc}}}. \tag{8}$$

See Fig. 17 for a comparison of the step responses of both filter types in relation to the time constant τ.

This mapping is used to derive the internal parameters $f_{\mathrm{c,acc}}$ and k_{mag} from the user-specified time constants τ_{acc} and τ_{mag}.

Appendix B. Gyroscope integration errors cause drift of almost-inertial frame

Every gyroscope prediction step causes a small error, due to gyroscope bias, noise, and other measurement errors. Without correction by accelerometers and magnetometers, those errors will add up and lead to drift in the orientation estimates. We can show that this error can be regarded as a small drift in the $\mathcal{I}_i$ frame, i.e., all rotation that happens in the gyroscope prediction and that is not the true change of sensor orientation can mathematically be expressed as a rotation $^{\mathcal{I}_i(t_{k-1})}_{\mathcal{I}_i(t_k)}\mathbf{q}$.

The gyroscope prediction step consists of multiplication of the previous estimate with an update quaternion based on the measured angular rate:

$$^{S_i(t_k)}_{\mathcal{I}_i(t_k)}\mathbf{q} = {}^{S_i(t_{k-1})}_{\mathcal{I}_i(t_{k-1})}\mathbf{q} \otimes \left(T_s\|\boldsymbol{\omega}\| @ \boldsymbol{\omega}\right). \tag{9}$$

This update quaternion can be expressed as the true change in sensor orientation multiplied with a small error quaternion $\mathbf{q}_e$, i.e.,

$$\left(T_s\|\boldsymbol{\omega}\| @ \boldsymbol{\omega}\right) = {}^{S_i(t_k)}_{S_i(t_{k-1})}\mathbf{q} \otimes \mathbf{q}_e. \tag{10}$$

We can transform the error rotation $\mathbf{q}_e$ to any frame, here $\mathcal{I}_i(t_{k-1})$:

$$\mathbf{q}_e = {}^{\mathcal{I}_i(t_{k-1})}_{S_i(t_k)}\mathbf{q} \otimes \left[\mathbf{q}_e\right]_{\mathcal{I}_i(t_{k-1})} \otimes {}^{\mathcal{I}_i(t_{k-1})}_{S_i(t_k)}\mathbf{q}^{-1}. \tag{11}$$

When putting this into the prediction step, we obtain

$$^{S_i(t_k)}_{\mathcal{I}_i(t_k)}\mathbf{q} = {}^{S_i(t_{k-1})}_{\mathcal{I}_i(t_{k-1})}\mathbf{q} \otimes \left(T_s\|\boldsymbol{\omega}\| @ \boldsymbol{\omega}\right) \tag{12}$$

$$= {}^{S_i(t_{k-1})}_{\mathcal{I}_i(t_{k-1})}\mathbf{q} \otimes {}^{S_i(t_k)}_{S_i(t_{k-1})}\mathbf{q} \otimes \mathbf{q}_e \tag{13}$$

$$= {}^{S_i(t_k)}_{\mathcal{I}_i(t_{k-1})}\mathbf{q} \otimes {}^{\mathcal{I}_i(t_{k-1})}_{S_i(t_k)}\mathbf{q} \otimes \left[\mathbf{q}_e\right]_{\mathcal{I}_i(t_{k-1})} \otimes {}^{\mathcal{I}_i(t_{k-1})}_{S_i(t_k)}\mathbf{q}^{-1} \tag{14}$$

$$= \underbrace{\left[\mathbf{q}_e\right]_{\mathcal{I}_i(t_{k-1})} \otimes {}^{S_i(t_k)}_{\mathcal{I}_i(t_{k-1})}\mathbf{q}}_{^{\mathcal{I}_i(t_{k-1})}_{\mathcal{I}_i(t_k)}\mathbf{q}} \tag{15}$$

Therefore, expressed in the almost-inertial frame $\mathcal{I}_i(t_{k-1})$, the gyroscope prediction error quaternion $\mathbf{q}_e$ corresponds to the drift rotation $^{\mathcal{I}_i(t_{k-1})}_{\mathcal{I}_i(t_k)}\mathbf{q}$ of the almost-inertial frame $\mathcal{I}_i$.

Appendix C. Details on gyroscope bias estimation

The full algorithm for gyroscope bias estimation is given in Algorithm 2.

To detect whether the IMU is at rest (procedure RestDetection in Algorithm 2), we first filter each component of the gyroscope and accelerometer measurements with a second-order Butterworth filter and a time constant of $\tau = 0.5\,\mathrm{s}$. Note that, unlike the low-pass filter for the acceleration used for inclination correction, we apply the filter directly in the sensor frame. We then calculate the Euclidean norm of the deviation between the current measurement and the filtered measurement. Rest is detected if, in the last $1.5\,\mathrm{s}$, the gyroscope and accelerometer deviations are always less than $2°/\mathrm{s}$ and $0.5\,\mathrm{m/s^2}$, respectively. Note that we deliberately do not use magnetometer measurements for rest detection since we found the rest detection to be very reliable when only using gyroscope and accelerometer measurements, whereas using magnetometers did not add additional value.

To estimate the gyroscope bias during rest and motion, we employ a Kalman filter [29,30], with the gyroscope bias as state and a time-dependent output matrix:

$$\mathbf{b}(t_k) = \mathbf{b}(t_{k-1}) + \mathbf{v}(t_k), \qquad \mathbf{v}(t_k) \sim \mathcal{N}(0, \mathbf{V}) \tag{16}$$

$$\mathbf{y}(t_k) = \mathbf{C}(t_k)\mathbf{b}(t_k) + \mathbf{w}(t_k), \qquad \mathbf{w}(t_k) \sim \mathcal{N}(0, \mathbf{W}(t_k)). \tag{17}$$

During rest, we use the low-pass filtered gyroscope readings $\boldsymbol{\omega}_{\mathrm{LP}}$ (which we calculated in the rest detection procedure) as a direct

Algorithm 2 Gyroscope Bias Estimation

1: **procedure** RestDetection($\omega, \mathbf{a}$)
2:     $T_{\text{rest}} \leftarrow T_{\text{rest}} + T_{\text{s}}$
3:     **for** each measurement $\omega, \mathbf{a}$ **do**
4:         low-pass filter the measurement with $\tau = 0.5\,\text{s}$
5:         **if** absolute difference between current and filtered measurement value is above threshold for any component **then**
6:             $T_{\text{rest}} \leftarrow 0$
7:         **end if**
8:     **end for**
9:     **if** $T_{\text{rest}} \geq 1.5\,\text{s}$ **then**
10:         rest detected
11:     **else**
12:         movement detected
13:     **end if**
14: **end procedure**
15: **procedure** InitializeKalmanFilter
16:     $\hat{\mathbf{b}} \leftarrow [0\ 0\ 0]^{\mathsf{T}}$ ▷ Gyroscope bias estimate
17:     $\mathbf{P} \leftarrow (0.5\,°/\text{s})^2 \mathbf{I}_{3\times3}$ ▷ Covariance matrix
18:     $v \leftarrow (0.1\,°/\text{s})^2 T_{\text{s}}(100\,\text{s})^{-1}$ ▷ System noise
19:     $w_{\text{motion}} \leftarrow (0.1\,°/\text{s})^4 v^{-1} + (0.1\,°/\text{s})^2$
                                               ▷ Motion update variance
20:     $w_{\text{rest}} \leftarrow (0.03\,°/\text{s})^4 v^{-1} + (0.03\,°/\text{s})^2$
                                               ▷ Rest update variance
21: **end procedure**
22: **procedure** BiasEstimationStep($^{I_i}_{\mathcal{E}_i}\mathbf{q}, a_x, a_y, a_z$)
23:     $\mathbf{R} \leftarrow$ rotation matrix corresponding to $^{I_i}_{\mathcal{E}_i}\mathbf{q}$
24:     $\mathbf{R}_{\text{LP}} \leftarrow$ low-pass filter $\mathbf{R}$ with $\tau = \tau_{\text{acc}}$
25:     $\hat{\mathbf{b}}_{\mathcal{E}_i,\text{LP}} \leftarrow$ low-pass filter $\mathbf{R}\hat{\mathbf{b}}$ with $\tau = \tau_{\text{acc}}$
26:     **if** rest detected **then**
27:         $\mathbf{y} \leftarrow \hat{\mathbf{b}}$
28:         $\mathbf{C} \leftarrow \mathbf{I}_{3\times3}$
29:         $\mathbf{W} \leftarrow w_{\text{rest}}[1\ 1\ 1]^{\mathsf{T}}$
30:     **else**
31:         $\mathbf{y} \leftarrow T_{\text{s}}^{-1}[a_y\ -a_x\ 0]^{\mathsf{T}} + \text{diag}(1,1,0)\hat{\mathbf{b}}_{\mathcal{E}_i,\text{LP}}$
32:         $\mathbf{C} \leftarrow \mathbf{R}_{\text{LP}}$
33:         $\mathbf{W} \leftarrow w_{\text{motion}}[1\ 1\ \frac{1}{0.0001}]^{\mathsf{T}}$
34:     **end if**
35:     $\mathbf{P} \leftarrow \mathbf{P} + v[1\ 1\ 1]^{\mathsf{T}}$ ▷ Kalman filter update
36:     $\mathbf{K} \leftarrow \mathbf{P}\mathbf{C}^{\mathsf{T}}(\mathbf{W} + \mathbf{C}\mathbf{P}\mathbf{C}^{\mathsf{T}})^{-1}$
37:     $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{b}} + \mathbf{K}\,\text{clip}(\mathbf{y} - \mathbf{C}\hat{\mathbf{b}}, -2\,°/\text{s}, 2\,°/\text{s})$
                                 ▷ Limit disagreement to $2\,°/\text{s}$
38:     $\mathbf{P} \leftarrow \mathbf{P} - \mathbf{K}\mathbf{C}\mathbf{P}$
39:     $\hat{\mathbf{b}} \leftarrow \text{clip}(\hat{\mathbf{b}}, -2\,°/\text{s}, 2\,°/\text{s})$
                               ▷ Limit bias estimate to $2\,°/\text{s}$
40: **end procedure**

Fig. 18. Illustration of the principle behind gyroscope bias estimation from the inclination correction step. In the steady state, the correction angular rate ω_{corr} that corresponds to the accelerometer-based correction is the (negative) horizontal projection of the remaining gyroscope bias $\mathbf{b} - \hat{\mathbf{b}}$.

With $\mathbf{R}$ being the rotation matrix corresponding to $^{s_i}_{\mathcal{E}_i}\mathbf{q}$, we can express this as

$$\mathbf{R}(\mathbf{b} - \hat{\mathbf{b}}) = -\frac{1}{T_{\text{s}}}\begin{bmatrix} c_x \\ c_y \\ * \end{bmatrix}, \tag{18}$$

where the star indicates that the gyroscope bias in the current vertical direction is not observable by accelerometer measurements.

To achieve slow forgetting of the bias for the special case in which the same axis is vertical for a long time, we set the corresponding measurement to zero, with a significantly larger variance to slow down convergence.

With an output matrix of $\mathbf{C}_k = \mathbf{R} =: (r_{ij})$ and the correction vector $[a_y\ -a_x\ 0]^{\mathsf{T}}$, we obtain the following measurement equation:

$$\mathbf{y}_k = \begin{bmatrix} -\frac{1}{T_{\text{s}}}a_y + r_{11}\hat{b}_x + r_{12}\hat{b}_y + r_{13}\hat{b}_z \\ \frac{1}{T_{\text{s}}}a_x + r_{21}\hat{b}_x + r_{22}\hat{b}_y + r_{23}\hat{b}_z \\ 0 \end{bmatrix}. \tag{19}$$

This measurement equation does not take into account that the measured accelerations are low-pass filtered in the $\mathcal{E}_i$ frame. The accuracy and robustness of the bias estimation can be significantly improved by low-pass filtering the components of $\mathbf{R}$ and $\mathbf{R}\hat{\mathbf{b}}$ with the same low-pass filter used for the accelerometer measurements in the $\mathcal{I}_i$ frame. See Appendix D for a detailed derivation of this full measurement equation.

The tuning parameters of a Kalman filter are the initial covariance, the covariance of the system noise, and the covariance of the measurement noise. Since the real values of those covariances are hard to obtain and would furthermore depend on the sampling rate, we employ a parametrization that ensures the following properties:

1. The initial estimation uncertainty is $\sigma_{\text{init}} = 0.5°/\text{s}$.
2. During motion, the uncertainty converges to $\sigma_{\text{motion}} = 0.1°/\text{s}$ (for the non-vertical axes).
3. During rest, the uncertainty converges to $\sigma_{\text{rest}} = 0.03°/\text{s}$.
4. Without updates, the estimation uncertainty increases from 0 to $0.1°/\text{s}$ in the forgetting time $t_{\text{forget}} = 100\,\text{s}$.

How those parameters translate to the covariances internally used by the Kalman filter is specified in Algorithm 2 and further explained in Appendix E. Note that the absolute values of the provided parameters are arbitrary and chosen to facilitate an intuitive understanding of the estimation uncertainty. For the behavior of the Kalman filter, only the relation between the parameters is relevant.

measurement of the bias, i.e., $\mathbf{C}(t_k) = \mathbf{I}_{3\times3}$ and $\mathbf{y}(t_k) = \omega_{\text{LP}}(t_k)$. Because the IMU is at rest and gyroscope readings are already filtered, we can assign a comparatively large weight (i.e., a small covariance) to this measurement update and achieve fast convergence for the bias estimate $\hat{\mathbf{b}}(t_k)$.

During motion, we estimate the gyroscope bias from the inclination correction steps. At every time step, the new error due to gyroscope bias is a local rotation (i.e., in S_i) with the rotation vector $T_s(\mathbf{b} - \hat{\mathbf{b}})$, and the inclination correction is a global rotation (i.e., in $\mathcal{E}_i$) with a horizontal rotation vector $\mathbf{c} = [c_x\ c_y\ 0]^{\mathsf{T}}$. In ideal conditions (i.e., in a steady state and without noise or other errors), the correction rotation will exactly compensate the inclination portion of the bias rotation. As illustrated in Fig. 18, in this case, the correction is the inverse of the horizontal projection of the bias rotation.
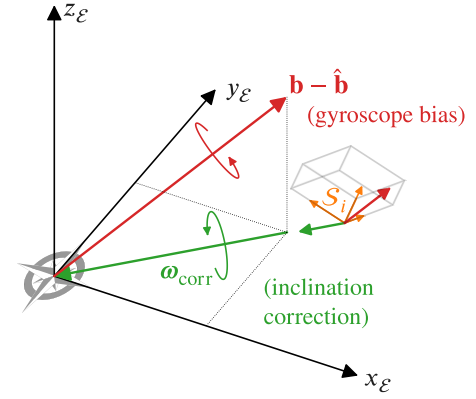
Appendix D. Measurement for motion bias estimation

The approach for gyroscope bias estimation during motion, as shown in Fig. 18, is based on the assumption that the current inclination correction step corresponds to the gyroscope bias that is transformed to the global frame using the current sensor orientation. This works well if the sensor orientation does not change much. In reality, due to the low-pass filter used for the acceleration, the inclination correction corresponds to the bias-induced rotations from the last few seconds, taking the respective sensor orientations into account. In the following, we show that this effect is well-described by applying the same low-pass filter to the elements of the rotation matrices $\mathbf{R}$ corresponding to $_{\mathcal{I}_i}^{S_i}\mathbf{q}$, and also to the rotated bias estimates $\mathbf{R}\hat{\mathbf{b}}$.

To simplify the notation, we introduce the rotation operator:

$$\mathrm{rot}\left(\mathbf{q},\mathbf{v}\right) := \mathbf{q} \otimes \mathbf{v} \otimes \mathbf{q}^{-1}. \tag{20}$$

Assume that the accelerometer measurements are always a perfect vertical vector ($[\mathbf{a}(t_k)]_{\mathcal{E}_{\mathrm{true}}} = [\mathbf{v}]_{\mathcal{E}_{\mathrm{true}}} = [0\ 0\ 1]^\mathsf{T}$), i.e. there are no disturbances or measurement errors (unit length is only used to simplify the notation). In the orientation estimation update step, those accelerometer measurements are transformed into the $\mathcal{I}_i(t_k)$ frame and then low-pass filtered with an infinite impulse response (IIR) filter with impulse response b_n, i.e.,

$$\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{I}_i(t_k)} = \sum_{n=0}^{\infty} b_n \left[\mathrm{rot}\left(_{\mathcal{I}_i(t_{k-n})}^{\mathcal{E}_{\mathrm{true}}}\mathbf{q}, [\mathbf{v}]_{\mathcal{E}_{\mathrm{true}}}\right)\right] \tag{21}$$

$$= \sum_{n=0}^{\infty} b_n \left[\mathbf{v}\right]_{\mathcal{I}_i(t_{k-n})}. \tag{22}$$

One time step earlier, the filter output is

$$\left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{I}_i(t_{k-1})} = \sum_{n=0}^{\infty} b_n \left[\mathbf{v}\right]_{\mathcal{I}_i(t_{k-1-n})} \tag{23}$$

$$= \sum_{n=0}^{\infty} b_n \,\mathrm{rot}\left(_{\mathcal{I}_i(t_{k-1-n})}^{\mathcal{I}_i(t_{k-n})}\mathbf{q}, [\mathbf{v}]_{\mathcal{I}_i(t_{k-n})}\right). \tag{24}$$

From Rodrigues' rotation formula follows that for a small rotation θ around axis $\mathbf{k}$,

$$\mathbf{p}_{rot} \approx \mathbf{p} + \theta \mathbf{k} \times \mathbf{p}. \tag{25}$$

We use this approximation for the rotation of the $\mathcal{I}_i$ frame due to gyroscope bias in one sample step $_{\mathcal{I}_i(t_{k-n})}^{\mathcal{I}_i(t_{k-1-n})}\mathbf{q}$ (cf. Appendix B). Note that this rotation is only caused by the uncorrected part of the gyroscope bias, i.e., by $\mathbf{b}'(t_k) := \mathbf{b} - \hat{\mathbf{b}}(t_k)$. Therefore, we obtain

$$\left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{I}_i(t_{k-1})} \tag{26}$$

$$\approx \sum_{n=0}^{\infty} b_n \left([\mathbf{v}]_{\mathcal{I}_i(t_{k-n})} - T_s \left[\mathbf{b}'(t_{k-n})\right]_{\mathcal{I}_i(t_{k-n})} \times [\mathbf{v}]_{\mathcal{I}_i(t_{k-n})}\right). \tag{27}$$

The difference between two consecutive filtered accelerations is then

$$\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{I}_i(t_k)} - \left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{I}_i(t_{k-1})} \tag{28}$$

$$\approx T_s \sum_{n=0}^{\infty} b_n \left([\mathbf{b}'(t_{k-n})]_{\mathcal{I}_i(t_{k-n})} \times [\mathbf{v}]_{\mathcal{I}_i(t_{k-n})}\right) \tag{29}$$

$$= T_s \sum_{n=0}^{\infty} b_n \left(\mathrm{rot}\left(_{\mathcal{I}_i(t_{k-n})}^{S_i(t_{k-n})}\mathbf{q}, \mathbf{b}'(t_{k-n})\right) \times [\mathbf{v}]_{\mathcal{I}_i(t_{k-n})}\right). \tag{30}$$

Now, we express this difference in the frame $\mathcal{E}_i(t_{k-1})$ that is used to perform the inclination correction step, assuming that $\mathcal{E}_i$ and $\mathcal{I}_i$ do not change much over the duration that is relevant for the filter (since the influence of bias and the inclination correction is limited during short time spans), and assuming that the true vertical axis is approximately $[0\ 0\ 1]^\mathsf{T}$ in the $\mathcal{E}_i(t_{k-1})$ frame.

$$\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{E}_i(t_{k-1})} - \left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{E}_i(t_{k-1})} \tag{31}$$

$$\approx T_s \sum_{n=0}^{\infty} b_n \left[\mathrm{rot}\left(_{\mathcal{E}_i(t_{k-1-n})}^{S_i(t_{k-n})}\mathbf{q}, \mathbf{b}'(t_{k-n})\right) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right] \tag{32}$$

$$= T_s \left[\sum_{n=0}^{\infty} b_n \,\mathrm{rot}\left(_{\mathcal{E}_i(t_{k-1-n})}^{S_i(t_{k-n})}\mathbf{q}, \mathbf{b}'(t_{k-n})\right)\right] \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{33}$$

Expressing the rotation by $_{\mathcal{E}_i(t_{k-n-1})}^{S_i(t_{k-n})}\mathbf{q}$ with a rotation matrix $\mathbf{R}(t_{k-n})$ and introducing an LPF operator to simplify the notation of the low-pass filter yields

$$\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{E}_i(t_{k-1})} - \left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{E}_i(t_{k-1})} \tag{34}$$

$$= T_s \left[\sum_{n=0}^{\infty} b_n \left(\mathbf{R}(t_{k-n})\mathbf{b}'(t_{k-n})\right)\right] \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{35}$$

$$= T_s \,\mathrm{LPF}\left(\mathbf{R}(t_k)\mathbf{b}'(t_k)\right) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{36}$$

Since $\left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{E}_i(t_{k-1})}$ is always $[0\ 0\ 1]^\mathsf{T}$ as the result of the previous inclination correction step, we get

$$\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{E}_i(t_{k-1})} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + T_s \,\mathrm{LPF}\left(\mathbf{R}(t_k)\mathbf{b}'(t_k)\right) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}. \tag{37}$$

The inclination correction rotation vector $\mathbf{c}(t_k)$, i.e., expressing the correction quaternion $\mathbf{q}_{corr}$ from Eq. (5) as a rotation vector, is (neglecting the small change of the norm of $\left[\mathbf{a}_{\mathrm{LP}}(t_k)\right]_{\mathcal{E}_i(t_{k-1})}$)

$$\mathbf{c}(t_k) = \left[\mathbf{a}_{\mathrm{LP}}(t_{k-1})\right]_{\mathcal{E}_i(t_k)} \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{38}$$

$$= \left(\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} + T_s \,\mathrm{LPF}\left(\mathbf{R}(t_k)\mathbf{b}'(t_k)\right) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}\right) \times \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{39}$$

$$= -\mathrm{diag}(1,1,0)T_s \,\mathrm{LPF}\left(\mathbf{R}(t_k)\mathbf{b}'(t_k)\right) \tag{40}$$

$$= -\mathrm{diag}(1,1,0)T_s \left(\mathrm{LPF}(\mathbf{R}(t_k))\mathbf{b} - \mathrm{LPF}(\mathbf{R}(t_k)\hat{\mathbf{b}}(t_k))\right). \tag{41}$$

Therefore, when choosing the system output matrix as $\mathbf{C}(t_k) = \mathrm{LPF}(\mathbf{R}(t_k))$, we obtain the following measurement in the horizontal plane:

$$\mathrm{diag}(1,1,0)\mathbf{y}(t_k) = -\frac{1}{T_s}\mathbf{c}(t_k) + \mathrm{diag}(1,1,0)\,\mathrm{LPF}(\mathbf{R}(t_k)\hat{\mathbf{b}}(t_k)). \tag{42}$$

Appendix E. Parametrization of the bias estimation method

Gyroscope bias is estimated using the system model

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \mathbf{v}_k, \qquad\qquad \mathbf{v}_k \sim \mathcal{N}(0,\mathbf{V}), \tag{43}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{b}_k + \mathbf{w}_k, \qquad\qquad \mathbf{w}_k \sim \mathcal{N}(0,\mathbf{W}_k), \tag{44}$$

where the index k denotes sampling at t_k, and the standard Kalman filter update equations for the estimated state $\hat{\mathbf{b}}_k$

$$\mathbf{P}_k^- = \mathbf{P}_{k-1}^+ + \mathbf{V} \tag{45}$$

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{C}_k^\mathsf{T}(\mathbf{W}_k + \mathbf{C}_k\mathbf{P}_k^-\mathbf{C}_k^\mathsf{T})^{-1} \tag{46}$$

$$\hat{\mathbf{b}}_k = \hat{\mathbf{b}}_{k-1} + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\hat{\mathbf{b}}_{k-1}) \tag{47}$$

$$\mathbf{P}_k^+ = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{C}_k\mathbf{P}_k^-. \tag{48}$$

The tuning parameters are the initial covariance $\mathbf{P}_0^+$, the variance of the system noise $\mathbf{V}$, and the variance of measurement noise $\mathbf{W}(t_k)$. In the following, we derive an intuitive parametrization for those values that is independent of the sampling frequency. Note that scaling all

parameters of the Kalman filter with the same value will not change the system behavior. For heuristically determined parameters, the actual quantities are therefore arbitrary. However, a good parametrization still helps to make the behavior of the algorithm understandable and facilitates tuning.

A fixed amount of variance, the variance of the system noise $\mathbf{V}$, is added to the covariance matrix in every update step. To be independent of the sampling frequency, scaling with the sampling time T_s is necessary. To facilitate interpretation of the value as a *forgetting time*, we parametrize the system noise by the time needed for the standard deviation of the estimation uncertainty to increase from 0 to $0.1°/s$ in the absence of measurements, i.e.,

$$\mathbf{V} = (0.1°/s)^2 \frac{T_s}{t_{\text{forget}}} \mathbf{I}_{3 \times 3}. \tag{49}$$

We use the initial estimation uncertainty, i.e., the standard deviation σ_{init}, to initialize the covariance matrix:

$$\mathbf{P}_0^+ = \sigma_{\text{init}}^2 \mathbf{I}_{3 \times 3}. \tag{50}$$

For the rest and motion updates, we provide the uncertainties σ_{rest} and σ_{motion} to which the estimate will eventually converge when the respective filter update is active instead of specifying the variance of the motion and rest update measurements directly. This ensures independence of the sampling rates and makes the parameters easy to compare to the initial standard deviation. The relation to the measurement variance $w_{\text{rest/motion}}$ is given by

$$w_{\text{rest/motion}} = \frac{\sigma_{\text{rest/motion}}^4}{v} + \sigma_{\text{rest/motion}}^2. \tag{51}$$

To derive this, consider a simplified case of a Kalman filter for a system with one constant state ($x_k = x_{k-1}$) and direct measurement of the state ($C = 1$):

$$p_k^- = p_{k-1}^+ + v \tag{52}$$

$$k_k = \frac{p_k^-}{w + p_k^-} \tag{53}$$

$$\hat{x}_k = \hat{x}_{k-1} + k_k(y_k - \hat{x}_{k-1}) \tag{54}$$

$$p_k^+ = p_k^- - k_k p_k^-. \tag{55}$$

In the converged state, $k_k = k_{k-1}$ and $p_k^+ = p_{k-1}^+$. From Eq. (55) follows

$$p_k^+ = p_k^+ + v - \frac{p_k^+ + v}{p_k^+ + v + w}(p_k^+ + v) \tag{56}$$

$$v = \frac{(p_k^+ + v)^2}{p_k^+ + v + w} \tag{57}$$

$$w = \frac{(p_k^+ + v)^2}{v} - p_k^+ - v = \frac{(p_k^+)^2}{v} + p_k^+. \tag{58}$$

The relation given in Eq. (51) is then obtained by replacing the variance p_k with σ^2.

Note that, for the 3-dimensional bias estimate, the 3×3 covariance matrix might not be close to a diagonal matrix, especially if the same sensor axis is vertical for a long time. The uncertainty σ of the bias estimate (in the worst-case direction) can be derived from the largest eigenvalue of the covariance matrix $\mathbf{P}$. To avoid calculating eigenvalues, the Gershgorin circle theorem can be leveraged to obtain an upper bound estimate via the largest absolute row sum of $\mathbf{P}$.

Appendix F. Magnetic disturbance rejection algorithm

The full algorithm for the magnetic disturbance detection and rejection extension as described in Section 2.5 is given in Algorithm 3.

Algorithm 3 Magnetic Disturbance Rejection

1: **procedure** MagDistDetection($\mathbf{m}, {}_{\mathcal{E}_i}^{S_i}\mathbf{q}$)
2:      $n \leftarrow \|\mathbf{m}\|$ ▷ Norm of magnetic field
3:      $\theta \leftarrow -\arcsin\left([0\ 0\ 1]\left({}_{\mathcal{E}_i}^{S_i}\mathbf{q} \otimes \mathbf{m} \otimes {}_{\mathcal{E}_i}^{S_i}\mathbf{q}^{-1}\right) n^{-1}\right)$
                               ▷ Dip angle
4:      low-pass filter n and θ with $\tau = 0.05\,s$
5:      **if** $|n - n_{\text{ref}}| < 0.1 n_{\text{ref}}$ and $|\theta - \theta_{\text{ref}}| < 10°$ **then**
6:          $T_{\text{undist}} \leftarrow T_{\text{undist}} + T_s$
7:          **if** $T_{\text{undist}} \geq 0.5\,s$ **then**
8:              disturbed $\leftarrow$ false
9:              $n_{\text{ref}} \leftarrow k_{\text{ref}}\left(n - n_{\text{ref}}\right)$
10:             $\theta_{\text{ref}} \leftarrow k_{\text{ref}}\left(\theta - \theta_{\text{ref}}\right)$
                       ▷ Track slow changes of norm and dip
11:          **else**
12:              disturbed $\leftarrow$ true
13:              $T_{\text{undist}} \leftarrow 0$
14:          **end if**
15:      **end if**
16: **end procedure**
17: **procedure** NewMagFieldAcceptance($n, \theta, |\boldsymbol{\omega}|$)
18:      **if** $|n - n_{\text{cand}}| < 0.1 n_{\text{cand}}$ and $|\theta - \theta_{\text{cand}}| < 10°$ **then**
19:          **if** $\|\boldsymbol{\omega}\| \geq 20°/s$ **then**
                     ▷ Only count the time if there is movement
20:              $T_{\text{cand}} \leftarrow T_{\text{cand}} + T_s$
21:          **end if**
22:          $n_{\text{cand}} \leftarrow k_{\text{ref}}\left(n - n_{\text{cand}}\right)$
23:          $\theta_{\text{cand}} \leftarrow k_{\text{ref}}\left(\theta - \theta_{\text{cand}}\right)$
24:          **if** disturbed and $T_{\text{cand}} \geq 20\,s$ **then**
                     ▷ Accept candidate as new reference
25:              disturbed $\leftarrow$ false
26:              $n_{\text{ref}} \leftarrow n_{\text{cand}}$
27:              $\theta_{\text{ref}} \leftarrow \theta_{\text{cand}}$
28:          **end if**
29:      **else**   ▷ Reset candidate to current value
30:          $T_{\text{cand}} \leftarrow 0$
31:          $n_{\text{cand}} \leftarrow n$
32:          $\theta_{\text{cand}} \leftarrow \theta$
33:      **end if**
34: **end procedure**
35: **procedure** MagDistRejection
36:      **if** disturbed **then**
37:          **if** $T_{\text{reject}} < 60\,s$ **then**
38:              $T_{\text{reject}} \leftarrow T_{\text{reject}} + T_s$
39:              do not perform heading correction
40:          **else**
41:              perform heading correction with $\frac{1}{2}k_{\text{mag}}$
42:          **end if**
43:      **else**
44:          $T_{\text{reject}} \leftarrow \max(T_{\text{reject}} - 2T_s, 0)$
45:          perform heading correction with k_{mag}
46:      **end if**
47: **end procedure**

Appendix G. Offline orientation estimation algorithm

The full algorithm for the acausal offline algorithm variant OfflineVQF as described in Section 2.6 is given in Algorithm 4.

Appendix H. Parameter tuning for the evaluated IOE algorithms

To provide a fair comparison for all evaluated algorithms, all algorithm parameters were tuned to provide the smallest possible TAGP_x values. To determine the parameters, a grid search was performed,

Algorithm 4 Offline Orientation Estimation Algorithm

1: **procedure** OFFLINEVQF($\boldsymbol{\omega}[t_{1:N}], \mathbf{a}[t_{1:N}], \mathbf{m}[t_{1:N}]$)
2:     $\ldots, \text{dist}_1, \hat{\mathbf{b}}_1, \mathbf{P}_1 \leftarrow \text{VQF}(\boldsymbol{\omega}[t_{1:N}], \mathbf{a}[t_{1:N}], \mathbf{m}[t_{1:N}])$ ▷ Run real-time filter in forward direction
3:     $\ldots, \text{dist}_2, \hat{\mathbf{b}}_2, \mathbf{P}_2 \leftarrow \text{VQF}(-\boldsymbol{\omega}[t_{N:1}], \mathbf{a}[t_{N:1}], \mathbf{m}[t_{N:1}])$ ▷ Run real-time filter in backward direction
4:     $\text{dist}[t_{1:N}] \leftarrow \text{dist}_1[t_{1:N}] \wedge \text{dist}_2[t_{1:N}]$ ▷ Regard magnetic field as disturbed if both runs detected disturbances
5:     $\hat{\mathbf{b}}[t_{1:N}] \leftarrow (\mathbf{P}_1^{-1} + \mathbf{P}_2^{-1})^{-1}(\mathbf{P}_1^{-1}\hat{\mathbf{b}}_1 - \mathbf{P}_2^{-1}\hat{\mathbf{b}}_2)$ ▷ Average bias estimates of both filter runs via covariance
6:     $^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}] \leftarrow \text{integrateGyr}(\boldsymbol{\omega}[t_{1:N}] - \hat{\mathbf{b}}[t_{1:N}])$ ▷ Perform gyroscope strapdown integration
7:     $[\mathbf{a}]_{\mathcal{I}_i}[t_{1:N}] \leftarrow {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}] \otimes \mathbf{a}[t_{1:N}] \otimes {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}]^{-1}$ ▷ Transform acceleration into $\mathcal{I}_i$ frame
8:     $[\mathbf{a}_{\text{LP}}]_{\mathcal{I}_i}[t_{1:N}] \leftarrow \text{filtfiltLPF}([\mathbf{a}]_{\mathcal{I}_i}[t_{1:N}], \tau = \tau_{\text{acc}})$ ▷ Forward–backward low-pass filtering
9:     $^{\mathcal{I}_i}_{\mathcal{E}_i}\mathbf{q}[t_{1:N}] \leftarrow \text{perform inclination correction based on } [\mathbf{a}_{\text{LP}}]_{\mathcal{I}_i}[t_{1:N}]$
10:    $[m_x\ m_y\ m_z]^{\mathsf{T}}[t_{1:N}] \leftarrow \left({}^{\mathcal{I}_i}_{\mathcal{E}_i}\mathbf{q}[t_{1:N}] \otimes {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}]\right) \otimes \mathbf{m}[t_{1:N}] \otimes \left({}^{\mathcal{I}_i}_{\mathcal{E}_i}\mathbf{q}[t_{1:N}] \otimes {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}]\right)^{-1}$
11:    $\delta_{\text{mag}}[t_{1:N}] \leftarrow \text{atan2}(m_x[t_{1:N}], m_y[t_{1:N}])$
12:    $\delta_i[t_{1:N}] \leftarrow \text{run heading correction filter with magnetic disturbance rejection on } \delta_{\text{mag}}[t_{1:N}]$
13:    $\delta_i[t_{1:N}] \leftarrow \text{run heading correction filter with magnetic disturbance rejection on } \delta_i[t_{N:1}]$
14:    **return** ${}^{\mathcal{I}_i}_{\mathcal{E}_i}\mathbf{q}[t_{1:N}] \otimes {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}],$ ▷ 6D sensor orientation ${}^{S_i}_{\mathcal{E}_i}\mathbf{q}$
15:      $[\cos\frac{\delta_i[t_{1:N}]}{2}\ 0\ 0\ \sin\frac{\delta_i[t_{1:N}]}{2}]^{\mathsf{T}} \otimes {}^{\mathcal{I}_i}_{\mathcal{E}_i}\mathbf{q}[t_{1:N}] \otimes {}^{S_i}_{\mathcal{I}_i}\mathbf{q}[t_{1:N}]$ ▷ 9D sensor orientation ${}^{S_i}_{\mathcal{E}}\mathbf{q}$
16: **end procedure**

VQF: real-time implementation, returns magnetic disturbance state, bias estimate, and bias estimation covariance
integrateGyr: gyroscope strapdown integration by Equation (3)
filtfiltLPF: forward–backward filtering with second-order Butterworth low-pass filter

Table 2
Results of TAGP$_x$-based parameter tuning for all IOE algorithms used in the evaluation.

Algorithm	TAGP$_x$	Parameter	Value	Search grid (start : step : end)
VQF	2.64°	τ_{acc}	3	1 : 0.5 : 10
		τ_{mag}	9	1 : 1 : 30
MAH [16]	14.83°	K_P	1.44	0.02 : 0.02 : 4
		K_I	0.0027	0 : 0.0001 : 0.004
MAD [17]	12.01°	β	0.29	0.01 : 0.01 : 1
		ζ_{bias}	0	0 : 0.00001 : 0.001
VAC [13]	5.63°	α_{acc}	0.00085	0.0001 : 0.00005 : 0.001
		β_{mag}	0.0005	0.0001 : 0.00005 : 0.001
		α_{bias}	0.00055	0.0001 : 0.00005 : 0.001
		b_{est}	True	{False, True}
		α_{adapt}	False	{False, True}
FKF [22]	9.19°	σ^2_{gyr}	0.001	0.001 : 0.001 : 0.001
		σ^2_{acc}	0.002	0.001 : 0.0001 : 0.005
		σ^2_{mag}	0.0033	0.001 : 0.0001 : 0.005
SEL [14]	4.58°	τ_{acc}	3.2	1 : 0.2 : 5
		τ_{mag}	10	1 : 1 : 20
		ζ_{bias}	5	0 : 1 : 10
		r_{acc}	2	0 : 1 : 10
MKF	7.58°	σ^2_{acc}	0.00028171	MKF parameters were iteratively
		σ^2_{mag}	14.55188	determined with line searches
		σ^2_{gyr}	0.15625	instead of a grid search.
		$\sigma^2_{\text{gyrdrift}}$	3×10^{-21}	
		σ^2_{linacc}	0.49128	
		d_{linacc}	0.81297	
		$\sigma^2_{\text{magdist}}$	0.12329	
		d_{magdist}	0.51005	
KOK [15]	11.70°	σ_{gyr}	0.185	0.01 : 0.005 : 0.5
		ζ_{bias}	0	0 : 0.00001 : 0.001
		m_{est}	True	{False, True}
RIANN [23]	1.32°	No parameters		

TAGP$_x$ value not comparable to other algorithms because the inclination RMSE is used instead of the orientation RMSE.

i.e., the algorithm performance was evaluated on a grid defined by the Cartesian product of the linearly spaced parameter sets presented in Table 2. This search grid was iteratively adjusted to ensure that the distance between parameter values is sufficiently small and that the TAGP$_x$ parameters do not lie at the border of the grid. The resulting averaged error, the associated parameters, and the parameter search range are presented in Table 2.

Due to the high dimensionality of the search space and the slow implementation, the parameters for MKF were only evaluated using a line search, i.e., only one parameter was changed while the other

parameters are kept at the previously found minimum. The search range was also iteratively adjusted until it converged to a stable minimum.

References

[1] R.S. McGinnis, N.C. Perkins, A highly miniaturized, wireless inertial measurement unit for characterizing the dynamics of pitched baseballs and softballs, Sensors 12 (2012) 11933–11945, http://dx.doi.org/10.3390/s120911933.

[2] D. Laidig, A.J. Jocham, B. Guggenberger, K. Adamer, M. Fischer, T. Seel, Calibration-free gait assessment by foot-worn inertial sensors, Front. Digit. Health 3 (2021) 147, http://dx.doi.org/10.3389/fdgth.2021.736418.

[3] Z. Wang, H. Zhao, S. Qiu, Q. Gao, Stance-phase detection for ZUPT-aided foot-mounted pedestrian navigation system, IEEE/ASME Trans. Mechatronics 20 (2015) 3170–3181, http://dx.doi.org/10.1109/TMECH.2015.2430357.

[4] H. Nguyen, K. Lebel, S. Bogard, E. Goubault, P. Boissy, C. Duval, Using inertial sensors to automatically detect and segment activities of daily living in people with Parkinson's disease, IEEE Trans. Neural Syst. Rehabil. Eng. 26 (2018) 197–204, http://dx.doi.org/10.1109/TNSRE.2017.2745418.

[5] A. Passon, T. Schauer, T. Seel, Inertial-robotic motion tracking in end-effector-based rehabilitation robots, Front. Robot. AI 7 (2020) http://dx.doi.org/10.3389/frobt.2020.554639.

[6] V. Rodrigo Marco, J. Kalkkuhl, J. Raisch, W.J. Scholte, H. Nijmeijer, T. Seel, Multi-modal sensor fusion for highly accurate vehicle motion state estimation, Control Eng. Pract. 100 (2020) 104409, http://dx.doi.org/10.1016/j.conengprac.2020.104409.

[7] H. Chao, Y. Cao, Y. Chen, Autopilots for small unmanned aerial vehicles: A survey, Int. J. Control Autom. Syst. 8 (2010) 36–44, http://dx.doi.org/10.1007/s12555-010-0105-z.

[8] J.H. Freter, T. Seel, C. Elfert, D. Göhlich, Motion estimation for tethered airfoils with tether sag, in: 2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), 2020, pp. 114–120, http://dx.doi.org/10.1109/MFI49285.2020.9235235.

[9] O.J. Woodman, An Introduction to Inertial Navigation, Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, 2007, http://dx.doi.org/10.48456/tr-696.

[10] M. Kok, J.D. Hol, T.B. Schön, Using inertial sensors for position and orientation estimation, Found. Trends Signal Process. 11 (2017) 1–153, http://dx.doi.org/10.1561/2000000094, arXiv:1704.06053.

[11] M. Nazarahari, H. Rouhani, 40 Years of sensor fusion for orientation tracking via magnetic and inertial measurement units: Methods, lessons learned, and future challenges, Inf. Fusion (2020) http://dx.doi.org/10.1016/j.inffus.2020.10.018.

[12] M. Caruso, A.M. Sabatini, D. Laidig, T. Seel, M. Knaflitz, U. Della Croce, A. Cereatti, Analysis of the accuracy of ten algorithms for orientation estimation using inertial and magnetic sensing under optimal conditions: One size does not fit all, Sensors 21 (2543) (2021) http://dx.doi.org/10.3390/s21072543.

[13] R.G. Valenti, I. Dryanovski, J. Xiao, Keeping a good attitude: A quaternion-based orientation filter for IMUs and MARGs, Sensors 15 (2015) 19302–19330, http://dx.doi.org/10.3390/s150819302.

[14] T. Seel, S. Ruppin, Eliminating the effect of magnetic disturbances on the inclination estimates of inertial sensors, IFAC-PapersOnLine 50 (2017) 8798–8803, http://dx.doi.org/10.1016/j.ifacol.2017.08.1534.

[15] M. Kok, T. Schön, A fast and robust algorithm for orientation estimation using inertial sensors, IEEE Signal Process. Lett. PP (2019) 1, http://dx.doi.org/10.1109/LSP.2019.294399.

[16] R. Mahony, T. Hamel, J.M. Pflimlin, Nonlinear complementary filters on the special orthogonal group, IEEE Trans. Automat. Control 53 (2008) 1203–1218, http://dx.doi.org/10.1109/TAC.2008.923738.

[17] S.O.H. Madgwick, An efficient orientation filter for inertial and inertial/magnetic sensor arrays, 32 (2010).

[18] M. Caruso, A.M. Sabatini, M. Knaflitz, M. Gazzoni, U.D. Croce, A. Cereatti, Orientation estimation through magneto-inertial sensor fusion: A heuristic approach for suboptimal parameters tuning, IEEE Sens. J. 21 (2021) 3408–3419, http://dx.doi.org/10.1109/JSEN.2020.3024806.

[19] D. Laidig, M. Caruso, A. Cereatti, T. Seel, BROAD—A benchmark for robust inertial orientation estimation, Data 6 (2021) 72, http://dx.doi.org/10.3390/data6070072.

[20] D. Laidig, I. Weygers, S. Bachhuber, T. Seel, VQF: A milestone in accuracy and versatility of 6D and 9D inertial orientation estimation, in: 2022 25th International Conference on Information Fusion (FUSION), 2022, pp. 1–6, http://dx.doi.org/10.23919/FUSION49751.2022.9841356.

[21] J.B. Kuipers, Quaternions and rotation sequences, in: Proceedings of the International Conference on Geometry, Integrability and Quantization, Coral Press Scientific Publishing, Varna, Bulgaria, 1999, pp. 127–143, http://dx.doi.org/10.7546/giq-1-2000-127-143.

[22] S. Guo, J. Wu, Z. Wang, J. Qian, Novel MARG-sensor orientation estimation algorithm using fast Kalman filter, J. Sensors 2017 (2017) e8542153, http://dx.doi.org/10.1155/2017/8542153.

[23] D. Weber, C. Gühmann, T. Seel, RIANN—A robust neural network outperforms attitude estimation filters, AI 2 (2021) 444–463, http://dx.doi.org/10.3390/ai2030028.

[24] F. Gustafsson, Determining the initial states in forward–backward filtering, IEEE Trans. Signal Process. 44 (1996) 988–992, http://dx.doi.org/10.1109/78.492552.

[25] A. Szczęsna, P. Skurowski, P. Pruszowski, D. Pęszor, M. Paszkuta, K. Wojciechowski, Reference data set for accuracy evaluation of orientation estimation algorithms for inertial motion capture systems, in: L.J. Chmielewski, A. Datta, R. Kozera, K. Wojciechowski (Eds.), Computer Vision and Graphics, Springer International Publishing, Cham, 2016, pp. 509–520, http://dx.doi.org/10.1007/978-3-319-46418-3_45.

[26] C. Chen, P. Zhao, C.X. Lu, W. Wang, A. Markham, N. Trigoni, OxIOD: The dataset for deep inertial odometry, 2018, arXiv:1809.07491 [cs]. arXiv:1809.07491.

[27] D. Schubert, T. Goll, N. Demmel, V. Usenko, J. Stückler, D. Cremers, The TUM VI benchmark for evaluating visual-inertial odometry, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1680–1687, http://dx.doi.org/10.1109/IROS.2018.8593419.

[28] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M.W. Achtelik, R. Siegwart, The EuRoC micro aerial vehicle datasets, Int. J. Robot. Res. 35 (2016) 1157–1163, http://dx.doi.org/10.1177/0278364915620033.

[29] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Basic Eng. 82 (1960) 35–45, http://dx.doi.org/10.1115/1.3662552.

[30] G. Welch, G. Bishop, An Introduction to the Kalman Filter, Technical Report TR 95-041, University of North Carolina at Chapel Hill, 2006.