Joint Magnetometer-IMU Calibration via Maximum A Posteriori Estimation

Chuan Huang, *Graduate Student Member, IEEE*, Gustaf Hendeby, *Senior Member, IEEE* and Isaac Skog, *Senior Member, IEEE*

***Abstract*—This paper presents a new approach for jointly calibrating magnetometers and inertial measurement units, focusing on improving calibration accuracy and computational efficiency. The proposed method formulates the calibration problem as a maximum a posteriori estimation problem, treating both the calibration parameters and orientation trajectory of the sensors as unknowns. This formulation enables efficient optimization with closed-form derivatives. The performance of the proposed method is compared against that of two state-of-the-art approaches. Simulation results demonstrate that the proposed method achieves lower root mean square error in calibration parameters while maintaining competitive computational efficiency. Further validation through real-world experiments confirms the practical benefits of our approach: it effectively reduces position drift in a magnetic field-aided inertial navigation system by more than a factor of two on most datasets. Moreover, the proposed method calibrated 30 magnetometers in less than 2 minutes. The contributions include a new calibration method, an analysis of existing methods, and a comprehensive empirical evaluation. Datasets and algorithms are made publicly available to promote reproducible research.**

***Index Terms*—inertial sensors, magnetometers, in-situ calibration, MAP estimation, IMU preintegration.**

I. Introduction

MAGNETOMETERS and inertial measurement units (IMUs) are widely used in various applications, such as robotics [1], augmented reality [2], localization and navigation [3], [4]. However, the accuracy of these sensors, particularly commercial-grade ones, is affected by factors such as biases, scale factors, and non-orthogonal sensitivity axes. In addition, magnetometers are highly susceptible to magnetic disturbances, often caused by nearby ferromagnetic materials, further complicating their usability in practical applications. Therefore, one must calibrate these sensors before use to ensure the best possible performance. More importantly, when multiple sensors are used in a system, cross-sensor calibration becomes essential to ensure proper alignment and consistent measurements across all sensors.

When it comes to calibrating low-cost magnetometers and IMUs, in-situ calibration methods, such as those proposed in [5]–[7], are popular because they do not require additional equipment, and thus saves time and cost. Although in-situ calibration methods generally cannot achieve the same accuracy as those that use additional equipment, such as a turntable, they are still sufficient for most use cases involving low- to medium-precision applications. These include pedestrian navigation, smartphone-based localization, and indoor mapping, where sub-degree orientation accuracy and sub-meter positional precision are not strictly required. To that end, in this paper, in-situ calibration methods are discussed.

Several in-situ joint calibration methods of magnetometers and IMUs have been proposed [8]–[14], most of which are based on the assumption that the magnetic field is homogeneous (spatially uniform and stable) and require the sensor platform to rotate to ensure good exposure to different orientations. A key distinction among these methods is whether they account for the rotation dynamics, captured by the gyroscope, across different time instances as the sensor is rotated through various poses. In other words, the distinction is whether the method models the full temporal sequence of sensor measurements within a dynamic system, rather than treating each pose independently. An advantage of dynamic system-based calibration methods is that they allow sensors to be easily rotated by hand, facilitating exposure to a wide range of orientations. In contrast, calibration methods that do not typically require the sensors to be rotated in a more controlled and discrete manner—for example, by placing them in 15 distinct static poses [10]. This process can be cumbersome and time-consuming in practice. Moreover, such methods often result in limited exposure to the full range of orientations compared to dynamic system-based approaches, which continuously incorporate sensor motion and typically achieve broader coverage of the orientations.

Due to their flexibility and ease of implementation, dynamic system-based methods are generally more practical and preferable. Two state-of-the-art methods are proposed in [8] and [9]. In [8], the magnetometer and IMU calibration problem is formulated as a maximum likelihood (ML) problem, where the likelihood is approximated using an extended Kalman filter (EKF) and then maximized. This method achieves good calibration accuracy; however, it is computationally expensive due to the complexity of the likelihood function. As a result, numerical derivatives must be used when maximizing the likelihood, which entails running the EKF multiple times to perform the necessary computations. In [9], the calibration problem is formulated as a maximum a posteriori (MAP) estimation problem, where an EKF is used to compute (approximate) the posterior probability density of the calibration parameter and current sensor orientation. This method is less

This work has been funded by the Swedish Research Council (Vetenskapsrådet) project 2020-04253 ”Tensor-field based localization”.

Chuan Huang is with Dept. of Electrical Engineering and Computer Science, KTH Royal Institute of Technology (e-mail: chuanh@kth.se).

Gustaf Hendeby is with Dept. of Electrical Engineering, Linköping University (e-mail: gustaf.hendeby@liu.se).

Isaac Skog is with Dept. of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, and the Div. of Underwater Technology, Swedish Defence Research Agency (FOI), Kista, Sweden (e-mail: skog@kth.se).

computationally demanding; however, its accuracy depends on the quality of the EKF approximation, which is sensitive to the choice of linearization points.

To address the computational complexity of the method in [8] and the sensitivity to linearization points in [9], we propose a new calibration approach that formulates the problem as a joint MAP estimation problem, in which both the orientation trajectory and calibration parameters are treated as unknowns. This formulation results in a simpler objective function, compared to that in [8], with closed-form derivatives that facilitate efficient optimization. Furthermore, unlike [9], our method allows for dynamic adaptation of linearization points during the optimization process, improving robustness and convergence. In addition, we analyze the two state-of-the-art methods as well as the proposed method in terms of their underlying assumptions, i.e., the approximations to the probability densities, and give asymptotic upper bounds of the algorithms' computational complexity. Finally, we evaluate the proposed approach and the state-of-the-art methods on both simulation and real-world datasets, highlighting their advantages and key differences.

Our contributions are summarized as follows:

1) A new joint magnetometer-imu calibration method is proposed, offering improved computational efficiency and accuracy.
2) The connection between the proposed method and two state-of-the-art methods is analyzed.
3) A comparative empirical study of all three methods is presented, with a focus on computational efficiency and calibration accuracy.

Reproducible research: Both the datasets and algorithms used in this study are available at https://github.com/Huang-Chuan/Mag-IMU-JointCalibration.

II. Problem Formulation

Consider a sensor system consisting of a magnetometer and an IMU. Assume the system is slowly rotated, with a minimum of translational movement, in a homogeneous magnetic field. The dynamics of the corresponding discrete system can be specified as

$$x_{k+1} = f(x_k, \tilde{u}_k, \theta, \Delta T), \tag{1}$$

where the subscript k denotes the time index, x_k denotes the orientation of the IMU frame with respect to the reference frame, $\tilde{u}_k$ denotes the angular velocity measurements, θ denotes the unknown calibration parameter, and ΔT denotes the sampling interval. The measurements from an accelerometer and a magnetometer, as well as the measurements from a gyroscope, can be modeled as

$$y_k \triangleq \begin{bmatrix} \tilde{s}_k \\ \tilde{m}_k \end{bmatrix} = \begin{bmatrix} h^a(x_k;\theta) + e_k^a \\ h^m(x_k;\theta) + e_k^m \end{bmatrix}, \tag{2a}$$

and

$$\tilde{u}_k = h^\omega(u_k;\theta) + e_k^\omega, \tag{2b}$$

respectively. Here, $\tilde{s}_k \in \mathbb{R}^3$ and $\tilde{m}_k \in \mathbb{R}^3$ denote the accelerometer and magnetometer measurement, respectively. Moreover, $u_k \in \mathbb{R}^3$ denotes the true angular velocity, $e_k^{(\cdot)} \in \mathbb{R}^3$ denotes the additive white noise associated with each type of sensor, and its covariance is specified by $\Sigma_{(\cdot)}$. Further, the noise terms are assumed to be independent of each other.

Let $y_{1:T} \triangleq (y_1, y_2, \dots, y_T)$ denote the sequence of accelerometer and magnetometer measurements from time 1 to T, and $\tilde{u}_{0:T-1} \triangleq (\tilde{u}_0, \tilde{u}_1, \dots, \tilde{u}_{T-1})$ denote the sequence of angular velocity measurements from time 0 to $T-1$. The purpose of calibration is to estimate θ given the measurements $y_{1:T}$ and $\tilde{u}_{0:T-1}$.

In the following, two related yet distinct approaches are discussed. The first is the proposed MAP estimator based on the joint posterior distribution of states and calibration parameters. The second is a MAP estimator based on the marginal posterior distribution of the calibration parameters, as presented in [8] and [9].

A. MAP Estimation Using the Joint Posterior

A MAP estimator can be used to estimate the calibration parameter θ and the state trajectory $x_{0:T}$ jointly. Formally, the estimate is given by [15, p.15]

$$(\hat{\theta}, \hat{x}_{0:T})^{\text{MAP}} \triangleq \underset{\theta, x_{0:T}}{\arg\max}\, p(\theta, x_{0:T} \,|\, y_{1:T}, \tilde{u}_{0:T-1}), \tag{3a}$$

$$= \underset{\theta, x_{0:T}}{\arg\max} \Big(p(y_{1:T} \,|\, \theta, x_{0:T}, \tilde{u}_{0:T-1}) \times p(x_{0:T}, \theta \,|\, \tilde{u}_{0:T-1}) \Big). \tag{3b}$$

Here, $p(A \,|\, B)$ denotes the conditional probability density function of the random variable A given the random variable B. Equation (3b) follows directly from Bayes' rule applied to (3a), omitting the normalizing factor $p(\tilde{u}_{0:T-1} \,|\, y_{1:T})$. Furthermore, the terms $p(y_{1:T} \,|\, \theta, x_{0:T}, \tilde{u}_{0:T-1})$ and $p(x_{0:T}, \theta \,|\, \tilde{u}_{0:T-1})$ in (3b) can be factorized as

$$p(y_{1:T} \mid \theta, x_{0:T}, \tilde{u}_{0:T-1}) = \prod_{k=1}^{T} p(y_k \mid \theta, x_k), \tag{4a}$$

and

$$p(x_{0:T}, \theta \mid \tilde{u}_{0:T-1}) = p(\theta) p(x_0) \prod_{k=0}^{T-1} p(x_{k+1} | x_k, \theta, \tilde{u}_k), \tag{4b}$$

respectively. Equations (3), (4a), and (4b) form the foundation of the proposed method. How to practically solve the optimization problem in (3) will be discussed in Sec. III.

B. MAP Estimation Using the Marginal Posterior

A MAP estimator can also be used to estimate the calibration parameter *only*. This is done by maximizing the marginal posterior probability density of the calibration parameter θ, which is given by [15, p.18]

$$\hat{\theta}^{\text{MAP}} = \underset{\theta}{\arg\max}\, p(\theta \mid y_{1:T}, \tilde{u}_{0:T-1}), \tag{5a}$$

where

$$p(\theta \mid y_{1:T}, \tilde{u}_{0:T-1}) = \int p(\theta, x_{0:T} \mid y_{1:T}, \tilde{u}_{0:T-1})\, dx_{0:T}. \tag{5b}$$

Note the integrand in (5b) is exactly the probability density that is to be maximized in (3a).

Equation (5a) can be equivalently written as [8]

$$\hat{\theta}^{\text{MAP}} = \arg\max_{\theta} p(\theta) \prod_{k=1}^{T} p(y_k \mid y_{1:k-1}, \theta, \tilde{u}_{0:k-1}), \tag{6a}$$

where

$$p(y_1 \,|\, y_{1:0}, \theta, \tilde{u}_0) \triangleq p(y_1 \,|\, \theta, \tilde{u}_0). \tag{6b}$$

In [8], $p(y_k \,|\, y_{1:k-1}, \theta, \tilde{u}_{0:k-1})$ is approximated by the one-step ahead measurement prediction density in an EKF. In addition, the prior $p(\theta)$ is assumed to be uniform distributed, thus, the estimator used in [8] is equivalent to an ML estimator.

Alternatively, (5a) can also be equivalently expressed as

$$\hat{\theta}^{\text{MAP}} = \arg\max_{\theta} \int p(\theta, x_T \mid y_{1:T}, \tilde{u}_{0:T-1}) dx_T, \tag{7}$$

which reflects the approach proposed in [9]. The authors used an EKF with an augmented state-space model

$$x_{k+1} = f(x_k, \tilde{u}_k, \theta_k, \Delta T), \tag{8a}$$

$$\theta_{k+1} = \theta_k, \tag{8b}$$

to compute an approximation to the integrand in (7). Since $p(\theta, x_T \mid y_{1:T}, \tilde{u}_{0:T-1})$ is approximated as a joint Gaussian distribution density, the marginal posterior $p(\theta \mid y_{1:T}, \tilde{u}_{0:T-1})$ is also a Gaussian distribution density. Further, given the Gaussian distribution, the maximization process in (5a) corresponds to selecting the mean of the marginal posterior.

C. Discussion of the Calibration Methods

The proposed approach offers several advantages, notably its reduced reliance on approximations to the underlying probability distributions and its favorable computational efficiency. First, all probability terms in the formulation, except for the transition term $p(x_{k+1} \,|\, x_k, \theta, \tilde{u}_k)$, can be evaluated exactly when the noise in the system is Gaussian. Meanwhile, the transition term can be well approximated under the assumption of reasonably small gyroscope noise, as discussed in [16]. More importantly, the estimator does not assume a specific joint probability distribution of the calibration parameter and state or rely on the Gaussian approximation that one would use in an EKF. By avoiding such assumptions and approximations, the proposed approach preserves the true posterior structure and can potentially yield more accurate estimates. Furthermore, when the priors $p(\theta)$ and $p(x_0)$ are Gaussian distributed or uninformative, i.e., $p(\cdot) \propto 1$, the MAP estimation problem reduces to a nonlinear least squares problem, which can be efficiently solved using standard solvers such as Gauss–Newton (GN) or Levenberg–Marquardt (LM). Due to the sparsity of the Jacobian matrices associated with the optimization problem, which is similar to those encountered in the exploration tasks during the simultaneous localization and mapping process [17], the computational complexity per iteration is $\mathcal{O}(3T + \dim(\theta))$. The factor 3 corresponds to the three degrees of freedom in the orientation representation for each of the T poses. Assuming convergence in N_{iter} iterations, the total computational complexity is $\mathcal{O}\left(N_{\text{iter}}(3T + \dim(\theta))\right)$.

In contrast, the methods proposed in [8] and [9] rely on the EKF to approximate either the likelihood function or the posterior density. However, it is well-known that the EKF is sensitive to the choice of linearization points, which can degrade estimation accuracy. This issue may be mitigated by using an Iterated Kalman Filter (IKF) [18], which iteratively refines the linearization around updated state estimates. For the method [8], a quasi-Newton optimization solver is used to maximize the likelihood function. The numerical derivatives needed by the optimization program require running an EKF $\dim(\theta)$ times. For an EKF that estimates state vectors of dimension d, the computational complexity per time step is $\mathcal{O}(d^3)$ [19]. Here $d = 3$ because of the three degrees of freedom in the state. Therefore, the total computational complexity is $\mathcal{O}(N_{\text{iter}} \times \dim(\theta) \times T \times 3^3)$, assuming the optimization program converges in N_{iter} iterations. In practice, the number of iterations N_{iter} required for convergence in this approach tends to be significantly larger than that of the proposed MAP-based method. Recent work [20], [21] shows that the computational complexity could be significantly reduced using gradient backpropagation techniques. On the other hand, the method in [9] avoids numerical derivatives by directly running an EKF filter with the augmented state-space model (8). This leads to a computational complexity of $\mathcal{O}\left(T \times (\dim(\theta) + 3)^3\right)$. While computationally more efficient in this respect, the reliance on a Gaussian approximation to the joint posterior may limit estimation quality.

III. Implementation

Next, we present the state-space model employed for joint magnetometer-IMU calibration, along with the implementation details of the proposed method.

A. State-Space Model and Probabilities

The unknown calibration parameter θ is defined as

$$\theta \triangleq (o^a, o^\omega, D^m, o^m, \alpha), \tag{9}$$

where $o^a \in \mathbb{R}^3$, $o^\omega \in \mathbb{R}^3$, and $o^m \in \mathbb{R}^3$ denote the accelerometer, gyroscope, and magnetometer biases, respectively. Moreover, $D^m \in \mathbb{R}^{3\times3}$ denotes the magnetometer distortion matrix, and $\alpha \in \mathbb{R}$ denotes the dip angle of the local magnetic field, as shown in Fig.1. The state x_k is parameterized as a rotation matrix $R_k \in \text{SO}(3)$. In the sequel, x_k is used interchangeably with R_k.

Assume that the true angular velocity remains approximately constant between the measurement intervals and the gyroscope measurements are affected by additive biases, i.e., $h^\omega(u_k; \theta) = u_k + o^\omega$. The system dynamics are then described by

$$f(x_k, \tilde{u}_k, \theta, \Delta T) = R_k \text{Exp}\big((\tilde{u}_k - o^\omega - e^\omega_k)\Delta T\big) \tag{10a}$$

where

$$\text{Exp}(v) \triangleq e^{v^\wedge}, \quad v \in \mathbb{R}^3, \tag{10b}$$

$$v^\wedge \triangleq \begin{bmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{bmatrix}, \tag{10c}$$

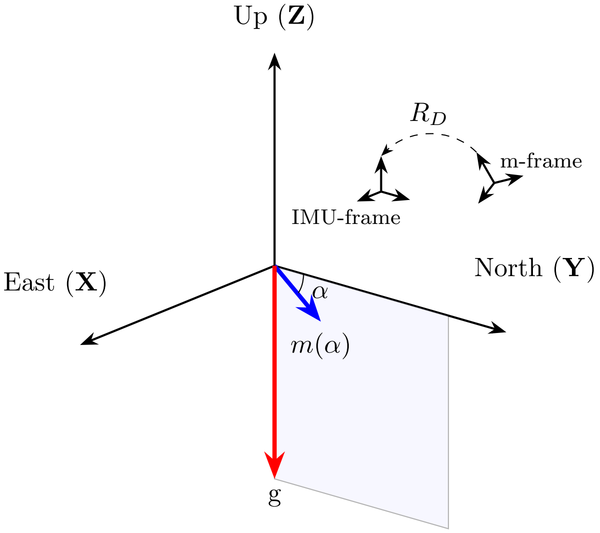

Fig. 1. The reference (big axes sets) and sensors' coordinate frames (small axes sets). The Z-axis in the reference frame is aligned with the local gravity vector, g, the Y–Z plane is parallel to the local magnetic field, $m(\alpha)$, with its horizontal component pointing in the positive Y-axis direction. The dip angle of the magnetic field α is the angle between the magnetic field and the horizontal plane. The misalignment of the magnetometer frame (m-frame) with the inertial sensor frame (IMU-frame) is represented by the rotation matrix R_D.

The sensors' measurement functions in (2a) are [8]

$$h^a(x_k;\theta) = -R_k^\top \mathrm{g}^\mathrm{n} + o^a, \tag{11a}$$
$$h^m(x_k;\theta) = D^m R_k^\top m^\mathrm{n}(\alpha) + o^m. \tag{11b}$$

Here $\mathrm{g}^\mathrm{n} = \begin{pmatrix}0 \; 0 \; \mathrm{g}_0\end{pmatrix}^\top$ and $m^\mathrm{n}(\alpha) = \begin{pmatrix}0 \; \cos(\alpha) \; -\sin(\alpha)\end{pmatrix}^\top$ denote the local gravity and the normalized local magnetic field expressed in the reference frame, respectively. Note that the distortion matrix D^m can be factorized as

$$D^m = D^I R_D, \tag{12}$$

where $D^I \in \mathbb{R}^{3\times 3}$ denotes the intrinsic calibration matrix and $R_D \in \mathrm{SO}(3)$ denotes the rotation matrix that aligns the magnetometer frame with the inertial sensor frame.

The equations in (11) allow us to specify the probability densities in (4a) as

$$p(y_k \mid x_k, \theta) = \mathcal{N}\left(y_k \,\middle|\, \begin{bmatrix} h^a(x_k;\theta) \\ h^m(x_k;\theta) \end{bmatrix}, \begin{bmatrix} \Sigma_a & 0 \\ 0 & \Sigma_m \end{bmatrix} \right). \tag{13}$$

Here $\mathcal{N}(\mu, P)$ denotes a Gaussian distribution with mean μ and covariance P, and $\mathcal{N}(\cdot \mid \mu, P)$ is the corresponding probability density function. Further, (10a) specify the state transition probability densities $p(x_{k+1} \,|\, x_k, \tilde{u}_k, \theta)$ in (4b) as

$$p(x_{k+1} \mid x_k, \tilde{u}_k, \theta) \approx \frac{1}{\sqrt{\det(2\pi\Sigma_w)}} e^{-\frac{1}{2}\|\delta u_k\|^2_{\Sigma_w}}, \tag{14a}$$
$$\delta u_k \triangleq \tilde{u}_k - \frac{1}{\Delta T}\mathrm{Log}(R_k^\top R_{k+1}) - o^\omega, \tag{14b}$$
$$\mathrm{Log}(R) \triangleq a\psi, \; a \in \mathbb{R}^3, \psi \in \mathbb{R}. \tag{14c}$$

Here a, ψ denote the rotation axis and the rotation angle of R, respectively, and the operation $\mathrm{Log}(\cdot)$ can be found in [16]. Further, $\det(\cdot)$ denotes the determinant of a matrix, and $\|\cdot\|^2_\Sigma \triangleq (\cdot)^\top \Sigma^{-1} (\cdot)$. The approximative density function (14a) is the result of applying the theorem in [16], which states if $R \in \mathrm{SO}(3)$ and $\tilde{R} \in \mathrm{SO}(3)$, and they are related via $\tilde{R} = R\mathrm{Exp}(v)$, where $v \sim \mathcal{N}(0, \Sigma)$ and Σ is sufficiently small, then $p(\tilde{R}) \approx \frac{1}{\sqrt{\det(2\pi\Sigma)}} e^{-\frac{1}{2}\|\mathrm{Log}(R^\top \tilde{R})\|^2_\Sigma}$.

B. Cost Functions

Since the logarithm $\ln(\cdot)$ is a monotonic function, the MAP estimator in (3) can be equivalently obtained by minimizing the cost function $L(\theta, x_{0:T})$, i.e.,

$$(\hat{\theta}, \hat{x}_{0:T})^{\text{MAP}} \triangleq \underset{\theta, x_{0:T}}{\arg\min}\, L(\theta, x_{0:T}), \tag{15a}$$

where

$$\begin{aligned} L(\theta, x_{0:T}) = &- \ln(p(\theta)) - \ln(p(x_0)) \\ &+ \sum_k \|\tilde{s}_k + R_k^\top \mathrm{g}^\mathrm{n} - o^a\|^2_{\Sigma_a} \\ &+ \sum_k \|\tilde{m}_k - D^m R_k^\top m(\alpha) - o^m\|^2_{\Sigma_m} \\ &+ \sum_k \|\tilde{u}_k - \frac{1}{\Delta T}\mathrm{Log}(R_k^\top R_{k+1}) - o^\omega\|^2_{\Sigma_\omega}. \end{aligned} \tag{15b}$$

Note that the cost function $L(\theta, x_{0:T})$ is simply the negative logarithm of the joint posterior density in (3). The first two terms are prior terms, and the last three terms correspond to (13) and (14).

In cases where the sample rate of the magnetometer is N times slower than that of the IMU sensor due to hardware limitations or downsampling to reduce computational load, the cost function can be modified accordingly, i.e.,

$$\begin{aligned} L(\theta, x_{0:T}^{(N)}) = &- \ln(p(\theta)) - \ln(p(x_0)) \\ &+ \sum_{k'} \|\tilde{s}_{k'} + R_{k'}^\top \mathrm{g}^\mathrm{n} - o^a\|^2_{\Sigma_a} \\ &+ \sum_{k'} \|\tilde{m}_{k'} - D^m R_{k'}^\top m(\alpha) - o^m\|^2_{\Sigma_m} \\ &+ \sum_{k'} \|\mathrm{Log}(\Delta\tilde{R}_{k',k'+N}^\top R_{k'}^\top R_{k'+N})\|^2_{\Sigma_{\Delta\tilde{R}_{k',k'+N}}}, \end{aligned} \tag{16a}$$

where

$$x_{0:T}^{(N)} = \{x_0, x_{N-1}, x_{2(N-1)}, \ldots\}, \tag{16b}$$
$$k' = (N-1)(l-1), \; l = 1, 2, \ldots, \tag{16c}$$
$$\Delta\tilde{R}_{i,j} = \prod_{k=i}^{j-1} \mathrm{Exp}\big((\tilde{u}_k - o^\omega)\Delta T\big), \tag{16d}$$
$$\Sigma_{\Delta\tilde{R}_{k',k'+N}} = \sum_{k=k'}^{k'+N-1} A_k \Sigma_w A_k^\top, \tag{16e}$$
$$A_k \triangleq \Delta\tilde{R}_{k+1,k'+N-1}^\top J_k^r \Delta T, \tag{16f}$$
$$J_k^r \triangleq J^r\big((\tilde{u}_k - o^\omega)\Delta T\big). \tag{16g}$$

Here $J^r(\cdot) \in \mathbb{R}^3$ is the right Jacobian of SO(3) and it is given by [22, p. 40]

$$J^r(v) = I - \frac{1 - \cos(\|v\|)}{\|v\|^2} v^\wedge + \frac{\|v\| - \sin(\|v\|)}{\|v\|^3} (v^\wedge)^2. \tag{17}$$

Note the terms in the last sum in (16a) are related to the probability density

$$p(x_{k'+N}\,|\,x_{k'},\theta,\tilde{u}_{k'},\ldots,\tilde{u}_{k'+N-1}) = \prod_{k=k'}^{k'+N-1} p(x_{k+1}|x_k,\theta,\tilde{u}_k), \tag{18}$$

which is given by IMU preintegration proposed in [16, Eq. 38].

C. Optimization on Manifolds

When the priors on θ and x_0 are Gaussian or uninformative, the optimization problems in (15a) or (16a) are reduced to nonlinear least squares problems, which can be solved by the GN algorithm [23]. However, since rotation variables live on the manifold $\mathrm{SO}(3)$, special treatment is required. In particular, the optimization problem is solved in the tangent space of $\mathrm{SO}(3)$, and interested readers are referred to [16] for more details. A brief review of the GN algorithm on manifolds and the analytic derivatives required by the solver are given in the Appendix A, B, and C.

D. Initialization

To solve the non-convex optimization problems in the previous sections, the optimization algorithms must have good initial values for the state trajectory $x_{0:T}$ and the calibration parameter θ, as well as the covariance of the noise $\Sigma_{(\cdot)}$ if fast convergence and accuracy are desired. The covariance of the noise can be obtained from the sensor datasheet or estimated using the Allan variance analysis [24]. To initialize the state trajectory, the gyroscope bias is estimated by calculating the mean value from stationary data. This bias is then subtracted from the measurements, and the corrected values are integrated to obtain an initial estimate of the orientation trajectory. The initial orientation is assumed to be the identity, which is approximately correct when the sensor board is held flat with its Z-axis pointing upwards and X-axis aligned with the north using a digital compass on a smartphone. Regarding IMU bias initialization, o^a is initialized with all zeros, and o^ω is initialized with the mean value of the stationary data. Further, the dip angle α is initialized using a geomagnetic calculator based on the most recent world magnetic model [25] and the current location. To initialize the calibration parameters D^m and o^m, intrinsic calibration and extrinsic calibration are applied.

1) Intrinsic Calibration: Intrinsic calibration refers to the calibration of the sensor biases, non-orthogonal sensitivity axes, and scale factors of the magnetometers, using magnetometer measurements only. To this end, (12) is used to substitute D^m in (11b) and R_D is absorbed into a time varying orientation matrix $R^m_k = R_D R_k^\top$, resulting in the following measurement model

$$\tilde{m}_k = h^m(R^m_k;\theta) + e^m_k, \tag{19a}$$

where

$$h^m(R^m_k;\theta) = D^I R^m_k m^{\mathrm{n}'} + o^m. \tag{19b}$$

Here R^m_k denotes the rotation matrix that aligns the magnetometer frame with the reference frame n'. For convenience, the reference frame is chosen such that the positive Z-axis aligns with the magnetic field vector, see Fig. 2. The benefit of this particular choice is that the orientation can be fully described using only the roll (ϕ) and pitch (γ) angles. This is because yaw (ψ) represents a rotation around the magnetic field direction, which does not change the relative inclination of the sensor to the field.

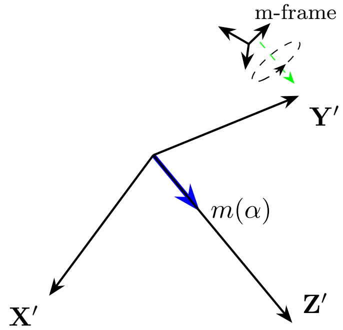

Fig. 2. The reference (big axes sets) and magnetometer's coordinate frames (small axes sets) used in the intrinsic calibration. The Z-axis in the reference frame is aligned with the local magnetic field, m. The orientation of the m-frame can be parameterized with only roll (ϕ) and pitch (γ) angles, since the yaw (ψ) angle represents a rotation around the magnetic field direction (in green), which does not change the relative inclination of the sensor to the field.

Let R^m_k be defined in terms of roll and pitch angles, i.e.,

$$R^m_k \triangleq R^m_k(\phi_k,\gamma_k) = R_x(\phi_k)R_y(\gamma_k), \tag{20a}$$

where

$$R_x(\phi_k) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi_k) & \sin(\phi_k) \\ 0 & -\sin(\phi_k) & \cos(\phi_k) \end{bmatrix}, \tag{20b}$$

$$R_y(\gamma_k) = \begin{bmatrix} \cos(\gamma_k) & 0 & -\sin(\gamma_k) \\ 0 & 1 & 0 \\ \sin(\gamma_k) & 0 & \cos(\gamma_k) \end{bmatrix}. \tag{20c}$$

The ML estimation problem of the parameters in the intrinsic calibration problem can then be formulated

$$\begin{aligned}&\left(\hat{D}^I,\hat{o}^m,\hat{\phi}_{1:T},\hat{\gamma}_{1:T}\right)^{\mathrm{ML}}\\ &= \operatorname*{arg\,max}_{D^I,o^m,\phi_{1:T},\gamma_{1:T}} \; p(\tilde{m}_{1:T};D^I,o^m,\phi_{1:T},\gamma_{1:T}).\end{aligned} \tag{21}$$

The ML estimator can be obtained by solving

$$\begin{aligned}&\left(\hat{D}^I,\hat{o}^m,\hat{\phi}_{1:T},\hat{\gamma}_{1:T}\right)^{\mathrm{ML}}\\ &= \operatorname*{arg\,min}_{D^I,o^m,\phi_{1:T},\gamma_{1:T}} \sum_{k=1}^{T} \|\tilde{m}_k - D^I R^m_k(\phi_k,\theta_k) m^{\mathrm{n}'} - o^m\|^2.\end{aligned} \tag{22}$$

Here $m^{\mathrm{n}'} = (0\;0\;1)^\top$ is the normalized magnetic field vector expressed in the reference frame n'. The solution to (22) is used to initialize the intrinsic calibration matrix D^I and the magnetometer bias o^m.

2) Extrinsic Calibration: Extrinsic calibration refers to the calibration of the rotation matrix R_D that aligns the magnetometer frame with the inertial sensor frame. In this work, we adopt the well-established method proposed in [26], which aligns the magnetometer frame with that of the gyroscope. This approach leverages the fact that, in a homogeneous magnetic field, variations in magnetometer measurements are solely caused by orientation changes, as measured by the gyroscope. Interested readers are referred to [26] for further details.

IV. Evaluation

The methods from Kok et al. [8] and Wu et al. [9] were implemented. The optimization toolboxes Manopt [27] and YALMIP [28] were used to implement both the proposed method and the method from [8]. We evaluated all methods on both simulated and real-world datasets. To ensure fair comparisons, identical termination criteria were applied to the optimization procedures for the proposed method and that in [8]. Specifically, the termination condition was set to either a norm of the update step less than 10^{-6} or reaching the maximum number of iterations.

Regarding performance, the evaluation on the simulated datasets is based on the computation time of each method and root mean square error (RMSE) of the estimated calibration parameters, while the evalutation on the real-world datasets is based on the computation time of each method and position RMSE of the navigation system in [3], which uses the calibrated IMU and the magnetometers' data from all three methods.

A. Simulation Study

To evaluate the performance of all three methods under varying conditions, two sets of experiments were conducted:

1) **Sampling Frequency Variation:** Datasets were generated with different sampling frequencies to assess how computational efficiency and calibration accuracy scale with measurement rate.
2) **Sampling Rate Ratio Variation:** In this experiment, datasets were created with different sample rate ratios between the IMU and the magnetometer, simulating asynchronous sensor setups commonly encountered in practice.

For each sampling frequency and sample rate ratio, 10 independent datasets were generated. In each dataset, the sensors were rotated around 6 fixed (with minor deviation) rotation axes, with an angular velocity of approximately 7 deg/s, consecutively. The duration of each dataset is approximately 5 minutes. The sensor data was generated using the sensor models in (11a) and (11b). Further, the calibration parameters were drawn from Gaussian distributions. In particular, the distortion matrix was constructed as

$$D^m = D_{\text{diag}} D_{\text{skew}} R_D, \tag{23a}$$

where

$$D_{\text{diag}} \triangleq \begin{bmatrix} D_{1,1} & 0 & 0 \\ 0 & D_{2,2} & 0 \\ 0 & 0 & D_{3,3} \end{bmatrix}, \tag{23b}$$

and

$$D_{\text{skew}} \triangleq \begin{bmatrix} 1 & 0 & 0 \\ \sin(\zeta) & \cos(\zeta) & 0 \\ -\sin(\eta) & \cos(\eta)\sin(\rho) & \cos(\eta)\cos(\rho) \end{bmatrix}. \tag{23c}$$

Here, $D_{1,1}, D_{2,2}, D_{3,3}$ denote the different scale factors on the three sensitivity axes, respectively. Further, ζ, η, ρ denote the different non-orthogonality angles, respectively. Lastly, R_D is defined by a set of Euler angles $(\phi_D, \gamma_D, \psi_D)$. The calibration parameter and noise settings are summarized in the TABLE I.

TABLE I
Calibration parameters and noise settings used in the simulations

Calibration Parameter	Distribution
D_{diag}	$D_{1,1}, D_{2,2}, D_{3,3} \sim \mathcal{U}(0.9, 1.1)$
D_{skew}	$\zeta, \eta, \rho \sim \mathcal{U}(-10^\circ, 10^\circ)$
R_D	$\phi_D, \gamma_D, \psi_D \sim \mathcal{U}(-5^\circ, 5^\circ)$
o^a	$o_1^a, o_2^a, o_3^a \sim \mathcal{U}(-0.5, 0.5)$ (m/s^2)
o^ω	$o_1^\omega, o_2^\omega, o_3^\omega \sim \mathcal{U}(0.47, 0.67)$ (°/s)
o^m	$o_1^m, o_2^m, o_3^m \sim \mathcal{U}(-2, 2)$ (μT)
α	$\mathcal{U}(67^\circ, 77^\circ)$
Sensor Type	Noise Density (per $\sqrt{\text{Hz}}$)
Accelerometer	0.02 m/s^2
Gyroscope	0.05 °/s
Magnetometer	0.003 μT

[1] $\mathcal{U}(a, b)$ denotes uniform distribution on (a, b).

First, the evaluation was performed on the datasets with different sampling frequencies, and the results are shown in Fig. 3. It can be seen from Fig. 3a that the computation time of all three methods increases almost linearly with the sampling frequency, which implies that the processing time increases linearly with the number of sensor measurements, which is consistent with the computational complexity upper bounds discussed in Section II-C. Furthermore, the method by Wu et al. is the fastest, followed by the proposed method and the method by Kok et al.. Note that parallel computing with 20 CPU cores is used in computing the numerical Jacobian matrix for the method by Kok et al.; computing with a single CPU core takes a much longer time. Regarding the magnetometer calibration results shown in Fig. 3b and Fig. 3c, the proposed method has the lowest RMSE in both bias and distortion matrix estimation. Meanwhile, the method by Kok et al. has a lower RMSE in bias than the method by Wu et al., while they share similar performance in distortion matrix estimation. In terms of accelerometer and gyroscope calibration results shown in Fig. 3d and Fig. 3e, the proposed method still has the lowest RMSE. However, the method by Wu et al. performs better than the method by Kok et al. Another interesting observation is that the RMSEs of the calibration parameters for all methods remain approximately constant as the sampling frequency increases. A possible explanation is that the sensor's motion, specifically its rotational dynamics, has limited bandwidth due to the relatively slow movement, making lower sampling frequencies sufficient to capture the necessary information.

Next, the three methods were evaluated on datasets where the magnetometer sampling rate was several times slower

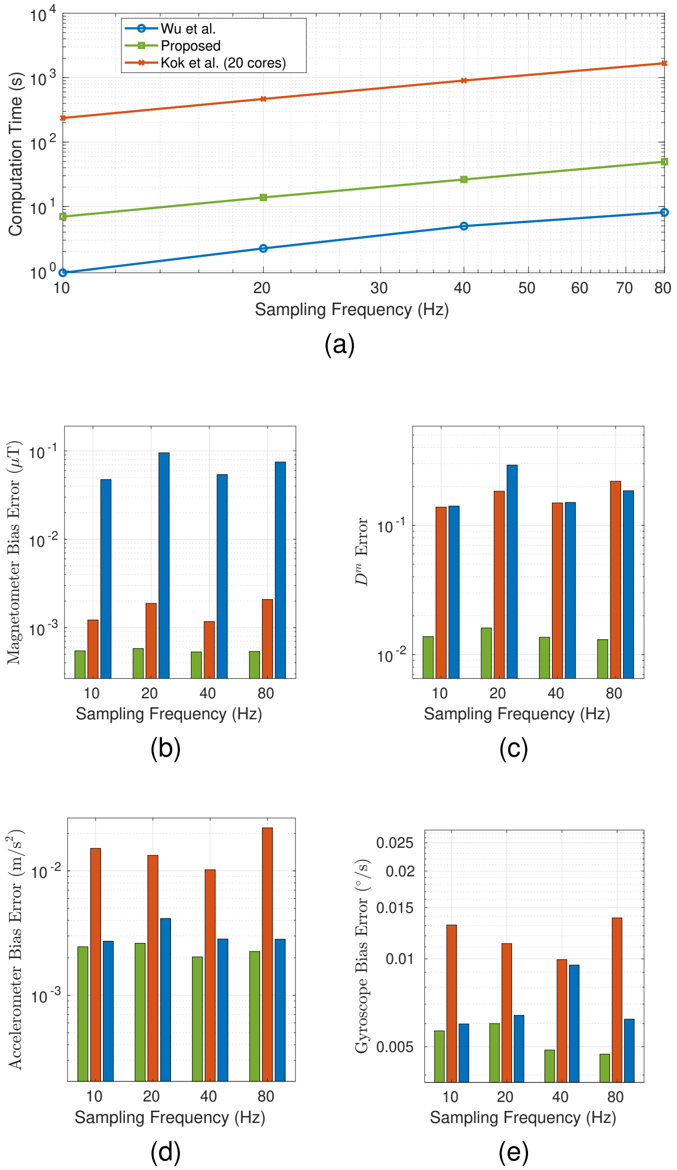

Fig. 3. Comparison of the computation time and RMSE of the estimated calibration parameters on datasets with different sampling frequencies.

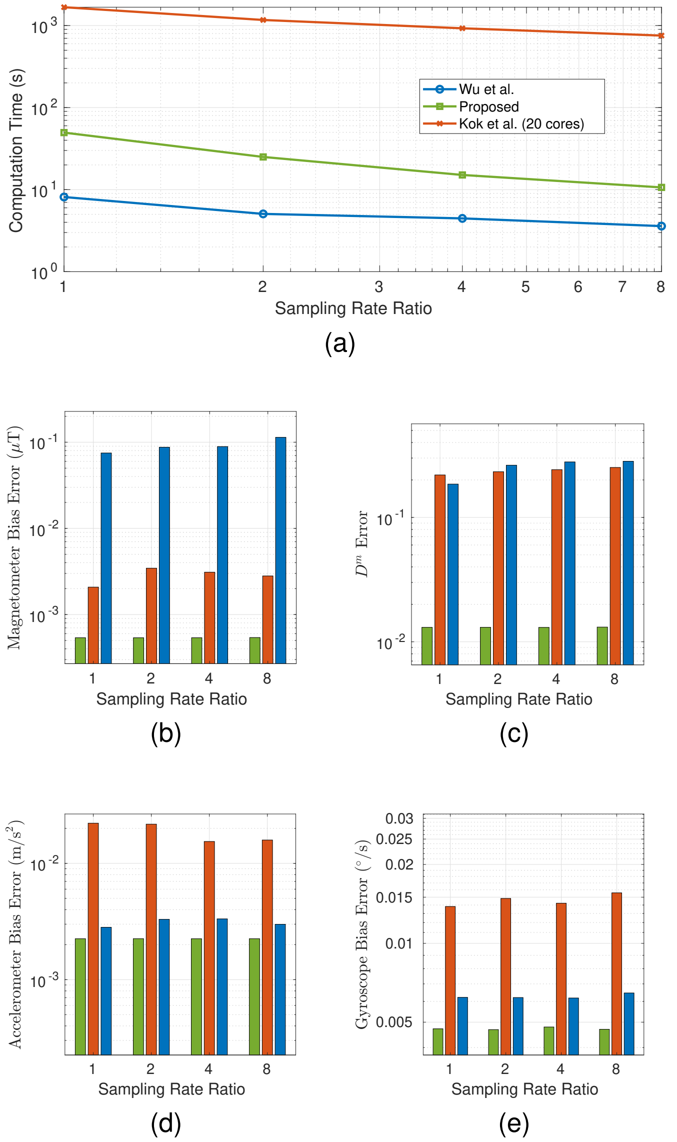

Fig. 4. Comparison of the computation time and RMSE of the estimated calibration parameters on datasets with different sampling rate ratios. The horizontal axis is the ratio of the IMU sampling rate to the magnetometer sampling rate.

(specifically 2, 4, and 8 times) than that of the IMU (80 Hz). These ratios define the sampling rate ratio, which refers to the ratio between the IMU and magnetometer sampling rates. For the proposed method, the cost function was constructed according to (16a). The other two methods performed EKF updates only when magnetometer and accelerometer measurements were received simultaneously. In this way, all three methods process the same number of measurements. The results are shown in Fig. 4. It can be seen that the computation time of all three methods decreases as the sampling rate ratio increases, while the RMSE of the estimated calibration parameters remains approximately constant. This indicates that one can have a lower magnetometer sampling rate than the IMU sampling rate to save computation time without sacrificing the accuracy of the calibration parameters. Moreover, Figs. 4b to 4e show that the relative RMSE magnitudes ranking across all three methods remains consistent as in the previous experiment.

B. Real-world Datasets Study

In real-world experiments, the three methods were used to calibrate a magnetic and inertial sensor array (see Fig. 5), which consists of 1 IMU and 30 magnetometers. The calibration data was collected on June 1st, 2024 in Linköping, Sweden. The magnetometer and IMU's sample rate was 62.5 Hz, and the duration of the calibration data collection was approximately 5 minutes. During calibration, the magnetometer and accelerometer data were downsampled by a factor of 3 to save computation cost. The joint magnetometer-IMU calibration procedures were done for each magnetometer-IMU pair. To evaluate the performance of the three methods, the estimated calibration parameters were used to calibrate the magnetometer data used in [3], and the calibrated data were fed into

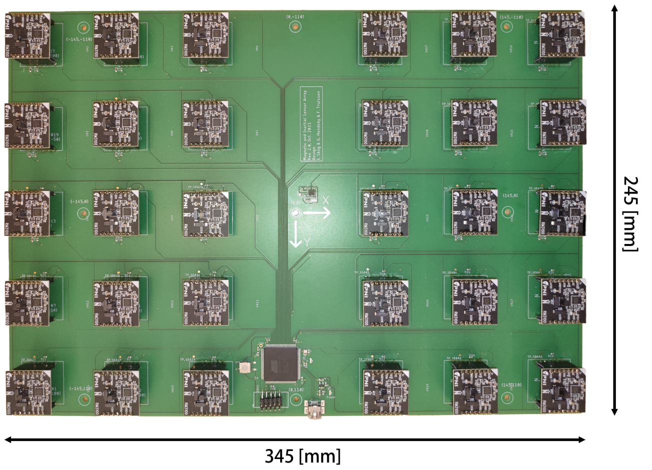

Fig. 5. The sensor board used in the experiment. It has 30 PNI RM3100 magnetometers and an Osmium MIMU 4844 IMU mounted on the bottom side.

the magnetic field-aided inertial navigation system (MAINS) proposed in [3]. The computation time of the three calibration methods, the resulting position RMSE of the MAINS, and an example of output trajectories are shown in Fig. 6. As a first observation, the ranking of the methods by calibration computation time matches the simulation results. Second, joint magnetometer-IMU calibration is clearly essential for MAINS to achieve good performance, as uncalibrated data leads to significant position drift shown in Fig. 6b. All three calibration methods effectively reduce this drift. Furthermore, the position RMSEs of MAINS using calibrated data from the proposed method and the method by Kok et al. are very similar in Figs. 6b and 6c, indicating comparable performance. Although the proposed method results in a slightly higher position RMSE, it offers a significant advantage in computational efficiency. This slight reduction in accuracy may be due to the method by Kok et al. being more robust to unmodeled sensor errors. Finally, the RMSE of MAINS using data calibrated by the method of Wu et al. is noticeably higher than those of the other two methods, which is consistent with the simulation results.

V. Conclusion

In this paper, we proposed a new method for joint magnetometer-IMU calibration, which jointly estimates the calibration parameters and orientation trajectory. The proposed method was compared with two state-of-the-art methods in terms of computational efficiency and calibration accuracy. Simulation results showed that the proposed method achieves lower RMSE in calibration parameters compared to the other methods, while maintaining competitive computational efficiency. Real-world experiments demonstrated that the proposed method can effectively calibrate an inertial-magnetometer array, thereby reducing position drift in a magnetic field-aided inertial navigation system. These results suggest that the proposed method is an attractive alternative to existing state-of-the-art methods, offering fast and accurate calibration with sustained performance.

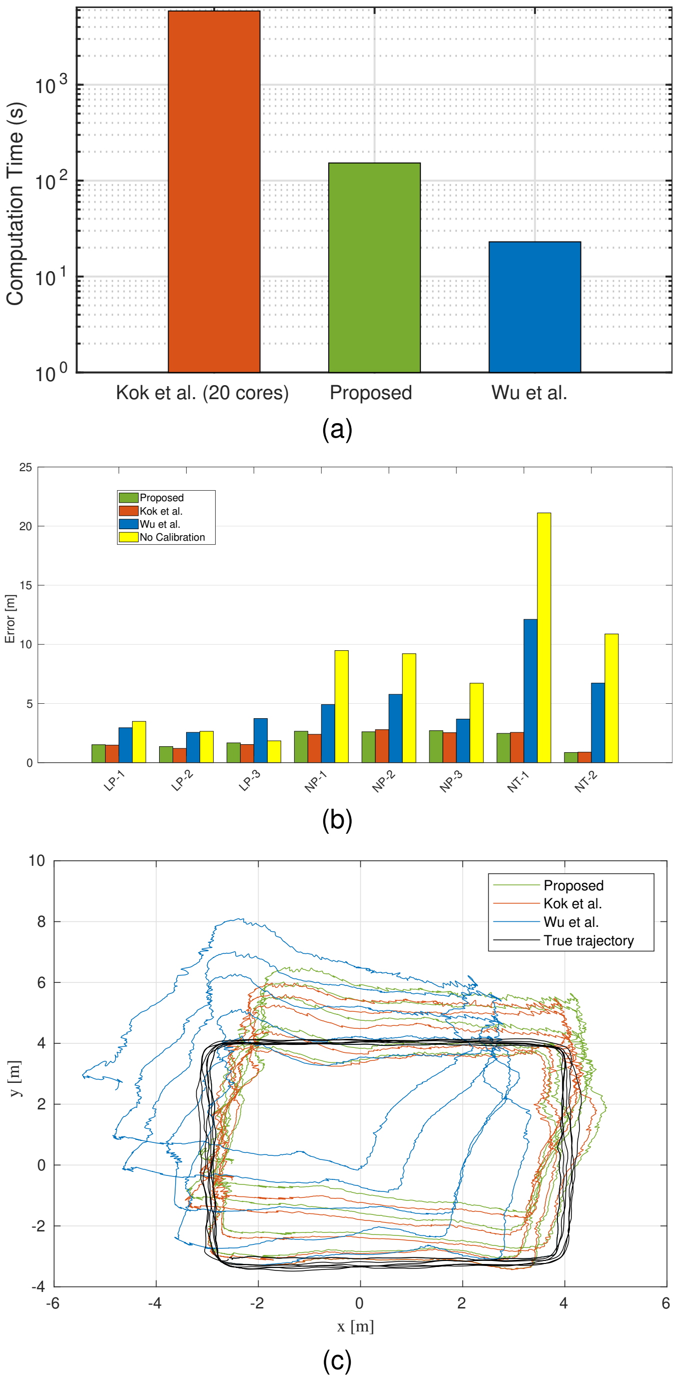

Fig. 6. Performance comparison on real-world data. (a) The computation time of the calibration algorithms. (b) The position RMSE of a magnetic field-aided inertial navigation system that uses the uncalibrated and calibrated magnetometer data. (c) The horizontal trajectories (LP-2) estimated by the magnetic field-aided inertial navigation system v.s. the true trajectory.

Acknowledgments

The authors would like to thank Manon Kok (Delft University of Technology) for inspiring the development of the ideas presented in this work, and Yuanxin Wu (Shanghai Jiao Tong University) for valuable discussions on the implementation details of [9].

APPENDIX A
GAUSS NEWTON METHOD ON MANIFOLDS

A general nonlinear least squares problem may be expressed as

$$\hat{x} = \arg\min_{x} \sum_{k} \|g_k(x)\|^2, \tag{24}$$

where $x \in \mathcal{M}$, $\mathcal{M}$ is a manifold, and $g_k(x)$ is a vector-valued function. To solve (24) with the GN method, one must first specify a operation that allows the Jacobian of $g_k(x)$ to be computed as well as the increment $\delta x \in \mathbb{R}^{\dim(\mathcal{M})}$ to be applied to x to obtain a new point on the manifold.

In (15a) and (16a), $x \in \mathcal{M} = \mathbb{R}^{\dim(\theta)} \times \mathrm{SO}(3)^M$, where M is the number of rotation matrices. The operation associated with the manifold $\mathcal{M}$ is given by

$$\theta \oplus \delta\theta \triangleq \theta + \delta\theta, \tag{25a}$$

$$R_i \oplus \delta R_i \triangleq R_i \mathrm{Exp}(\delta R_i). \tag{25b}$$

Here $\delta R_i \in \mathbb{R}^3$ is the increment in the tangent space of SO(3). Furthermore, the Jacobian of $g_k(x)$ is given by

$$J_k(x) = \begin{bmatrix} \frac{\partial g_k}{\partial \theta} & \frac{\partial g_k}{\partial R_0} & \cdots & \frac{\partial g_k}{\partial R_{M-1}} \end{bmatrix}, \tag{26}$$

where

$$\frac{\partial g_k}{\partial \theta} \triangleq \left. \frac{\partial g_k\big(\theta \oplus \delta\theta, \{R_i\}_{i=0}^{M-1}\big)}{\partial \delta\theta} \right|_{\delta\theta=0}, \tag{27}$$

$$\frac{\partial g_k}{\partial R_i} \triangleq \left. \frac{\partial g_k\big(\theta, \cdots, R_i \oplus \delta R_i, \cdots\big)}{\partial \delta R_i} \right|_{\delta R_i=0}. \tag{28}$$

With these definitions, the set of linear equations (normal equations) in each iteration of the GN algorithm can be constructed, and the increment obtained from solving it can be used to update the current estimate using (25).

APPENDIX B
DERIVATIVES FOR (15a)

For simplicity, the priors are assumed to be uniform, i.e., $\ln(p(\theta)), \ln(p(x_0))$ are constants, therefore, the corresponding terms can be removed from the cost function. The derivatives required by the optimization program are the derivatives of each individual residual term with respect to the calibration parameter θ and state R_k. Let the residual terms be defined as

$$r_k^a \triangleq L_a^\top (\tilde{s}_k + R_k^\top \mathrm{g}^{\mathrm{n}} - o^a), \tag{29}$$

$$r_k^m \triangleq L_m^\top \left(\tilde{m}_k - D^m R_k^\top m(\alpha) - o^m\right) \tag{30}$$

$$r_k^\omega \triangleq L_\omega^\top \left(\tilde{\omega}_k - \frac{1}{\Delta T}\mathrm{Log}(R_k^\top R_{k+1}) - o^\omega\right) \tag{31}$$

where $L_{(\cdot)}$ is the lower triangular matrix from Cholesky decomposition of the covariance matrix $\Sigma_{(\cdot)}^{-1}$.

The derivatives of r_k^a with respect to R_k and o^a are given by

$$\frac{\partial r_k^a}{\partial R_k} = L_a^\top [R_k^\top \mathrm{g}^{\mathrm{n}}]^\wedge R_k^\top, \tag{32a}$$

$$\frac{\partial r_k^a}{\partial o^a} = -L_a^\top, \tag{32b}$$

respectively. The derivatives of r_k^m with respect to R_k, D^m, α and o^m are given by

$$\frac{\partial r_k^m}{\partial R_k} = -L_m^\top D^m [R_k^\top m(\alpha)]^\wedge R_k^\top, \tag{33a}$$

$$\frac{\partial r_k^m}{\partial D_{\mathrm{vec}}^m} = -L_m^\top \left(R_k^\top m(\alpha)\right)^\top \otimes_K I_3, \tag{33b}$$

$$\frac{\partial r_k^m}{\partial \alpha} = L_m^\top D^m R_k^\top \left(0 \ \sin(\alpha) \ \cos(\alpha)\right)^\top, \tag{33c}$$

$$\frac{\partial r_k^m}{\partial o^m} = -L_m^\top, \tag{33d}$$

respectively. Here $D_{\mathrm{vec}}^m \in \mathbb{R}^9$ denotes the vectorized form of D^m and $\otimes_K$ denotes the Kronecker product. The derivatives of r_k^ω with respect to R_k, R_{k+1}, and o^ω are given by

$$\frac{\partial r_k^\omega}{\partial R_k} = \frac{1}{\Delta T} L_\omega^\top J^{-r}\left(\mathrm{Log}(R_k^\top R_{k+1})\right) R_{k+1}^\top, \tag{34a}$$

$$\frac{\partial r_k^\omega}{\partial R_{k+1}} = -\frac{1}{\Delta T} L_\omega^\top J^{-r}\left(\mathrm{Log}(R_k^\top R_{k+1})\right) R_{k+1}^\top, \tag{34b}$$

$$\frac{\partial r_k^\omega}{\partial o^\omega} = -L_\omega^\top, \tag{34c}$$

respectively. Here $J^{-r}(\cdot) \triangleq \left(J^r(\cdot)\right)^{-1}$.

APPENDIX C
DERIVATIVES FOR (16a)

As in Appendix A.1, we assume uniform priors, allowing us to omit the corresponding terms from the cost function. The derivatives of the accelerometer and magnetometer terms are the same as in Appendix A.1. To simplify the derivatives of the IMU preintegration term and reduce the computational load, two assumptions are made:

1) The gyroscope noise is assumed to be uncorrelated and identically distributed, i.e., $\Sigma_\omega = \sigma_\omega^2 I_3$.
2) The nominal gyroscope bias, $\bar{o}^\omega \in \mathbb{R}^3$, is assumed to be known. In practice, the nominal gyroscope bias can be obtained from averaging stationary data. The true gyroscope bias is then modeled as $o^\omega = \bar{o}^\omega + \Delta o^\omega$, where $\|\Delta o^\omega\|$ is small.

The first assumption is useful in simplifying the covariance of the preintegrated measurements, while the second assumption is to avoid directly evaluating preintegration measurements, $\Delta\tilde{R}_{i,j}$, due to changes to the bias estimate during optimization [16]. Instead of evaluating (16d), the preintegration measurements are linearized at $\bar{o}^\omega$ and approximated in each iteration as

$$\Delta\tilde{R}_{k',k'+N} \approx \Delta\bar{R}_{k',k'+N} \mathrm{Exp}\left(\left. \frac{\partial \Delta\tilde{R}_{k',k'+N}}{\partial o^\omega} \right|_{o^\omega=\bar{o}^\omega} \Delta o^\omega \right), \tag{35}$$

where $\Delta\bar{R}_{k',k'+N} \triangleq \Delta\tilde{R}_{k',k'+N}\big|_{o^\omega=\bar{o}^\omega}$.

The covariance of the preintegration measurements $\Sigma_{\Delta \tilde{R}_{k',k'+N}}$ can be approximated as

$$\Sigma_{\Delta \tilde{R}_{k',k'+N}} = \sum_{k=k'}^{k'+N-1} A_k \Sigma_w A_k^\top = \sum_{k=k'}^{k'+N-1} A_k \sigma_\omega^2 I_3 A_k^\top \tag{36a}$$

$$= \sigma_\omega^2 \sum_{k=k'}^{k'+N-1} A_k A_k^\top \tag{36b}$$

$$\approx \sigma_\omega^2 \Delta T^2 \sum_{k=k'}^{k'+N-1} \Delta \tilde{R}_{k+1,k'+N-1}^\top \Delta \tilde{R}_{k+1,k'+N-1} \tag{36c}$$

$$= N \sigma_\omega^2 \Delta T^2 I_3. \tag{36d}$$

Here $A_k = \Delta \tilde{R}_{k+1,k'+N-1}^\top J_k^r \Delta T \approx \Delta \tilde{R}_{k+1,k'+N-1}^\top \Delta T$ is used, since $J_k^r(\cdot) \approx I_3$ for small argument, $(\tilde{u}_k - \bar{o}^\omega)\Delta T$, which can be interpreted as the small angle the sensor turned between sampling interval, ignoring the noise and residual bias. Let the preintegration residual term be defined as

$$r_{k'}^{p.} = L_{p.}^\top \mathrm{Log}(\underbrace{\Delta \tilde{R}_{k',k'+N}^\top R_{k'}^\top R_{k'+N}}_{\triangleq Q}), \tag{37}$$

where $L_{p.}$ is the lower triangular matrix from Cholesky decomposing $\Sigma_{\Delta \tilde{R}_{k',k'+N}}^{-1}$.

The derivatives of $r_{k'}^{p.}$ with respect to R_k', $R_{k'+N}$, and Δo^ω are given by

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial R_{k'}} = -L_{\mathrm{p.}}^\top J^{-r}\big(\mathrm{Log}(Q)\big) R_{k'+N}^\top, \tag{38a}$$

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial R_{k'+N}} = L_{\mathrm{p.}}^\top J^{-r}\big(\mathrm{Log}(Q)\big) R_{k'+N}^\top, \tag{38b}$$

$$\frac{\partial r_{k'}^{\mathrm{p.}}}{\partial \Delta o^\omega} = -L_{\mathrm{p.}}^\top J^{-r}\big(\mathrm{Log}(Q)\big) Q^\top \times J^r \left(\left. \frac{\partial \Delta \tilde{R}_{k',k'+N}}{\partial o^\omega} \right|_{o^\omega = \bar{o}^\omega} \Delta o^\omega \right) \times \left. \frac{\partial \Delta \tilde{R}_{k',k'+N}}{\partial o^\omega} \right|_{o^\omega = \bar{o}^\omega}, \tag{38c}$$

where

$$\left. \frac{\partial \Delta \tilde{R}_{k',k'+N}}{\partial o^\omega} \right|_{o^\omega = \bar{o}^\omega} = - \sum_{i=k'}^{k'+N-1} \Delta \tilde{R}_{i+1,k'+N}^\top J_i^r \Delta T. \tag{38d}$$

REFERENCES

[1] N. Pavlasek, C. C. Cossette, D. Roy-Guay, and J. R. Forbes, "Magnetic navigation using attitude-invariant magnetic field information for loop closure detection," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, USA, 2023, pp. 5251–5257.

[2] H. Liu, H. Xue, L. Zhao, D. Chen, Z. Peng, and G. Zhang, "Magloc-ar: Magnetic-based localization for visual-free augmented reality in large-scale indoor environments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 11, pp. 4383–4393, 2023.

[3] C. Huang, G. Hendeby, H. Fourati, C. Prieur, and I. Skog, "MAINS: A magnetic-field-aided inertial navigation system for indoor positioning," *IEEE Sensors Journal*, vol. 24, no. 9, pp. 15 156–15 166, 2024.

[4] M. Zmitri, H. Fourati, and C. Prieur, "Improving inertial velocity estimation through magnetic field gradient-based extended Kalman filter," in *Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Pisa, Italy, Sep. 2019, pp. 1–7.

[5] H. Al Jlailaty, A. Celik, M. M. Mansour, and A. M. Eltawil, "IMU hand calibration for low-cost mems inertial sensors," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–16, 2023.

[6] Z. Zou, L. Li, X. Hu, Y. Zhu, B. Xue, J. Wu, and M. Liu, "Robust equipment-free calibration of low-cost inertial measurement units," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1–12, 2024.

[7] H. Yu, L. Ye, Y. Guo, and S. Su, "An effective in-field calibration method for triaxial magnetometers based on local magnetic inclination," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[8] M. Kok and T. B. Schön, "Magnetometer calibration using inertial sensors," *IEEE Sensors Journal*, vol. 16, no. 14, pp. 5679–5689, 2016.

[9] Y. Wu, D. Zou, P. Liu, and W. Yu, "Dynamic magnetometer calibration and alignment to inertial sensors by kalman filtering," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 2, pp. 716–723, 2018.

[10] K. Papafotis and P. P. Sotiriadis, "Mag.i.c.al.–a unified methodology for magnetic and inertial sensors calibration and alignment," *IEEE Sensors Journal*, vol. 19, no. 18, pp. 8241–8251, 2019.

[11] M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge, "Calibration of a magnetometer in combination with inertial sensors," in *2012 15th International Conference on Information Fusion*, 2012, pp. 787–793.

[12] E. Dorveaux, D. Vissière, A.-P. Martin, and N. Petit, "Iterative calibration method for inertial and magnetic sensors," in *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 2009, pp. 8296–8303.

[13] J. Liu, X. Li, Y. Zhang, A. Du, X. Zeng, and Y. Yang, "A gyro-aided strapdown triaxial magnetometer calibration method robust to gyro bias," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–12, 2021.

[14] X. Li and Z. Li, "Vector-aided in-field calibration method for low-end mems gyros in attitude and heading reference systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 11, pp. 2675–2681, 2014.

[15] S. Särkkä, *Bayesian Filtering and Smoothing*, ser. Institute of Mathematical Statistics Textbooks. Cambridge University Press, 2013.

[16] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[17] M. Kaess, A. Ranganathan, and F. Dellaert, "isam: Incremental smoothing and mapping," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1365–1378, 2008.

[18] B. Bell and F. Cathey, "The iterated Kalman filter update as a Gauss-Newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[19] F. Daum, "Nonlinear filters: beyond the Kalman filter," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 8, pp. 57–69, 2005.

[20] C. Parellier, A. Barrau, and S. Bonnabel, "Speeding-up backpropagation of gradients through the Kalman filter via closed-form expressions," *IEEE Transactions on Automatic Control*, vol. 68, no. 12, pp. 8171–8177, 2023.

[21] C. Parellier, C. Chapdelaine, A. Barrau, and S. Bonnabel, "Maximum likelihood estimation of the extended Kalman filter's parameters with natural gradient," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*, 2024, pp. 723–728.

[22] G. S. Chirikjian, *Stochastic models, information theory, and Lie groups, volume 2: Analytic methods and modern applications*. Springer Science & Business Media, 2011, vol. 2.

[23] A. Martin and H. Anders, *Optimization Problems*. John Wiley & Sons, Ltd, 2023, ch. 5, pp. 94–117.

[24] J. A. Farrell, F. O. Silva, F. Rahman, and J. Wendel, "Inertial measurement unit error modeling tutorial: Inertial navigation system state estimation with real-time sensor calibration," *IEEE Control Systems Magazine*, vol. 42, no. 6, pp. 40–66, 2022.

[25] NOAA National Centers for Environmental Information, "Geomagnetic calculator," 2025, accessed: 31-Mar-2025. [Online]. Available: https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml#igrfwmm

[26] Y. Wu and S. Luo, "On misalignment between magnetometer and inertial sensors," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6288–6297, 2016.

[27] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a Matlab toolbox for optimization on manifolds," *Journal of Machine Learning Research*, vol. 15, no. 42, pp. 1455–1459, 2014. [Online]. Available: https://www.manopt.org

[28] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in Matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.