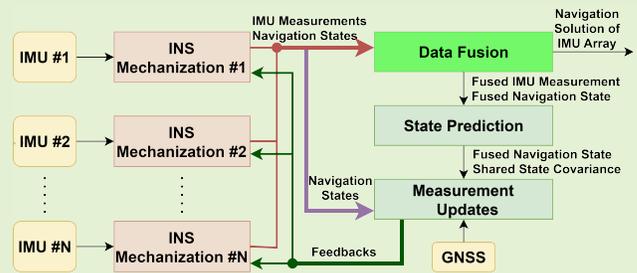


A Robust and Efficient IMU Array/GNSS Data Fusion Algorithm

Tisheng Zhang¹, Man Yuan¹, Liqiang Wang¹, Hailiang Tang¹, and Xiaoji Niu¹

Abstract—The inertial measurement unit (IMU) array, composed of multiple IMUs, has been proven to be able to effectively improve the navigation performance in inertial navigation system (INS)/global navigation satellite system (GNSS) integrated applications. The conventional IMU-level fusion algorithm, using IMU raw measurements, is straightforward and highly efficient but yields poor robustness when the IMU array is not rigidly installed. On the contrary, the classic INS-level fusion algorithm, using navigation results from each IMU, is immune to the nonrigid installation of the IMU array but suffers a heavy computation load. Here, we propose a robust and efficient INS-level fusion algorithm for IMU array/GNSS (eNav-Fusion). Each IMU in the array shares the common state covariance (P matrix) and Kalman gain (K matrix), and the navigation solutions of all IMUs are eventually fused to produce a more accurate solution. The proposed eNav-Fusion was fully evaluated with rigidly and nonrigidly installed IMU arrays. For a rigid 16-IMU array, the processing time of eNav-Fusion was close to that of the IMU-level fusion and only 1.22 \times to that of the INS/GNSS algorithm for a single IMU; and the navigation performance was improved by 2.51 \times , as expected for such scale of array. For a nonrigid 6-IMU array, in which case the traditional IMU-level fusion does not work, eNav-Fusion still maintained the same accuracy as the classic INS-level fusion algorithm, while the computation load is still close to that of the IMU-level fusion. In conclusion, the proposed eNav-Fusion achieves the same robustness as the INS-level fusion, while only consuming comparable computational complexity to the IMU-level fusion.

Index Terms—Data fusion, inertial measurement unit (IMU) array, inertial navigation system (INS)/global navigation satellite system (GNSS), micro-electromechanical system (MEMS) IMU.



I. INTRODUCTION

INERTIAL measurement unit (IMU), composed of a gyroscope and an accelerometer, is used to measure acceleration and angular velocity [1], [2]. Inertial navigation system (INS) is an autonomous navigation system based on Newton's classical mechanics, utilizing IMU measurements to compute high-frequency position, velocity, and attitude [3]. INS is often integrated with global navigation satellite system (GNSS) [4]. Micro-electromechanical system (MEMS) IMU has the advantages of small size, low power consumption, and affordability [5], [6], rendering it widely applicable in

civilian fields [7]. However, due to drawbacks in the design and manufacturing process, MEMS IMUs have the problem of poor parameter stability. The error of the computed navigation solution accumulates rapidly over time, restricting its suitability for scenarios demanding high precision.

Since the 1960s, researchers have been investigating IMU arrays [8]. Utilizing multiple IMUs to construct an IMU array enables the detection of individual IMU failures and the expansion of the dynamic measurement range [9]. In [10], two GPS/INS systems were employed to estimate the deck bending of an aircraft carrier. In [11], multiple IMUs were installed at different locations on an aircraft to measure the flexible deformation of the body. The ability of the IMU array to mitigate random errors is also a subject of considerable interest. According to the random error theory, an IMU array composed of N IMUs with independent measurement errors can reduce the random error by a factor of $(N)^{1/2}$ after data fusion [12]. Therefore, employing multiple MEMS IMUs to form an IMU array and designing a suitable data fusion algorithm can reduce random errors and enhance navigation performance without a significant cost increase. Research on the navigation performance and data fusion algorithm of IMU arrays has attracted widespread attention.

Manuscript received 8 April 2024; revised 25 May 2024; accepted 20 June 2024. Date of publication 9 July 2024; date of current version 15 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42374034 and Grant 41974024. The associate editor coordinating the review of this article and approving it for publication was Dr. Yang Yang. (Corresponding authors: Hailiang Tang; Xiaoji Niu.)

Tisheng Zhang and Xiaoji Niu are with the GNSS Research Center, Wuhan University, Wuhan 430079, China, and also with Hubei Luoqia Laboratory, Wuhan 430079, China (e-mail: zts@whu.edu.cn; xjniu@whu.edu.cn).

Man Yuan, Liqiang Wang, and Hailiang Tang are with the GNSS Research Center, Wuhan University, Wuhan 430079, China (e-mail: yuanman@whu.edu.cn; wlq@whu.edu.cn; thl@whu.edu.cn).

Digital Object Identifier 10.1109/JSEN.2024.3418383

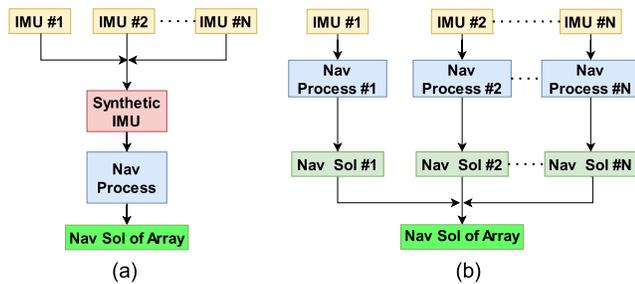


Fig. 1. Two data fusion algorithms for IMU array. Nav represents Navigation; Sol represents solution. (a) IMU level. (b) INS level.

The data fusion algorithm for the IMU array plays a crucial role in improving navigation performance and can be classified into two types based on the object of data fusion: IMU-level fusion algorithm and navigation solution level (INS-level) fusion algorithm, as illustrated in Fig. 1. The IMU-level fusion algorithm integrates measurements from multiple IMUs to produce a more precise measurement. This refined measurement is subsequently employed for integrated navigation, contributing to an enhanced navigation solution. The INS-level fusion algorithm initially conducts integrated navigation on each IMU, such as INS/GNSS integrated navigation. Subsequently, it merges the navigation solutions from all IMUs to yield a more precise solution.

When considering the computational complexity, there is a significant difference between IMU-level and INS-level fusion algorithms. The former typically directly fuses IMU measurements, resulting in computational complexity nearly equivalent to the INS/GNSS integrated navigation algorithm for a single IMU. The latter requires performing integrated navigation on multiple IMUs separately or expanding the state vector, so the computational complexity increases rapidly with the increase in the number of IMUs. However, the INS-level fusion algorithm exhibits greater robustness, allowing for its application in a broader range of scenarios. Particularly when the IMUs are nonrigidly connected, the lack of a definite transformation relationship between measurements of different IMUs theoretically prevents fusion at the IMU level [9]. In such cases, the INS-level fusion algorithm remains applicable, which can still effectively reduce random errors and enhance navigation performance.

Aiming at the limitations of the IMU-level fusion algorithm for nonrigid scenes and the high computational complexity of the navigation solution level fusion algorithm, we propose a robust and efficient navigation solution level IMU array/GNSS data fusion algorithm (eNav-Fusion) based on Kalman filter (KF) [13]. The proposed eNav-Fusion exhibits the same ability to improve navigation performance in both rigid and nonrigid scenarios, proving its robustness. Besides, the computational complexity is nearly equivalent to the integrated navigation algorithm for a single IMU. The fundamental concept of the proposed eNav-Fusion is to share the common operational steps and similar results present in the integration of each IMU with GNSS to achieve substantial computational savings. The proposed eNav-Fusion maximizes computational savings without causing a significant impact on navigation performance.

The main contributions of our work are as follows.

- 1) A robust and efficient IMU array/GNSS data fusion algorithm is proposed by sharing the same operation steps and similar operation results when each IMU in the array is integrated with GNSS. Specifically, each IMU within the array shares the same state covariance and Kalman gain. The navigation solutions from all IMUs are fused to produce a more accurate one.
- 2) The positioning accuracy and computational complexity of the proposed eNav-Fusion are analyzed theoretically, including the qualitative analysis of the navigation performance improvement and the quantitative analysis of computational complexity.
- 3) Rigid and nonrigid IMU arrays are employed for experimental validation on vehicles and robot carriers, respectively. The navigation performance and computational complexity of the proposed eNav-Fusion are quantitatively evaluated.

The structure of the rest of this article is as follows. We give a brief literature review in Section II. The proposed eNav-Fusion is presented in Section III. The experiments and results are discussed in Section IV for quantitative evaluation. Finally, we conclude with a summary and prospects for future work in Section V.

II. RELATED WORKS

This section discusses the related works on the performance and data fusion algorithm of the IMU array. The performance includes the static noise performance and the navigation performance. As mentioned above, the data fusion algorithm for the IMU array can be classified into IMU-level fusion algorithm and INS-level fusion algorithm.

A. IMU Array Performance

The static noise performance of the IMU array was first studied by researchers, and subsequent studies in recent years have further investigated this aspect. Skog et al. [9], Greenheck et al. [14], Greenheck [15], Bose et al. [16], and Jing and Zhao [17] designed IMU arrays composed of 14, 16, 16, 32, and 32 MEMS IMUs, respectively, and evaluated the static noise performance of IMU arrays by indicators, such as angular/velocity random walk and bias instability. The results from the studies mentioned above consistently demonstrated that the static noise performance of IMU arrays could achieve a relatively ideal improvement effect. That is, the improvement closely approached the theoretical value $(N)^{1/2}$, where N is the number of IMUs.

While the static noise performance is insufficient to fully reflect the positioning accuracy of the IMU array during navigation, numerous studies have been conducted to explore the navigation performance of the IMU array. Clausenl et al. [18] integrated the measurements from four IMUs, achieving enhanced navigation performance. The IMU array designed by Waegli et al. [19] comprised four IMUs installed on a regular tetrahedron, and the positioning error was reduced by 30%. Guerrier [20] designed a simulation model to establish a precise relationship between the number of sensors in the

array and the improvement of navigation performance. The results indicated that fusing four IMUs could decrease the positioning error by approximately 40%, while fusing ten IMUs could reduce the positioning error by around 60%. Bancroft [21] fused two, three, four, and five IMUs, reducing 25%, 29%, 32%, and 34% in plane positioning errors, respectively. Mi et al. [22] proposed a performance enhancement method for the IMU array based on neural network and geometric constraint. The results showed that the proposed method achieved a substantial 74% enhancement in position accuracy compared to the single IMU. Wang et al. [23] designed four groups of IMU arrays, each composed of 16 IMUs. They precisely calibrated and compensated for the errors of each IMU, including dynamic errors, such as installation angle and cross-axis coupling error, resulting in a 70% reduction in positioning error.

The above studies on navigation performance are based on rigid IMU arrays, and there are also researches focused on nonrigid IMU arrays. Skog et al. [24] combined two foot-mounted zero-velocity-aided INSs; one system on each foot. The navigation performance was significantly improved compared to using two uncoupled foot-mounted systems. In [10] and [11], the IMUs were nonrigidly connected but were not combined to improve navigation performance.

To summarize, the navigation performance of the IMU array can be significantly improved compared to a single IMU, and there are a few studies on the navigation performance of nonrigid IMU arrays.

B. IMU Array Data Fusion Algorithm

IMU-level fusion algorithms directly or indirectly integrate the IMU measurements, with a simple system structure and extensive research. Liu et al. [25] filtered and denoised measurements from eight IMUs and conducted distributed temperature compensation to offer a reliable data source for the backend solution, thereby enhancing navigation performance. Skog et al. [26] employed the mean and median values to fuse measurements from 14 IMUs installed on the feet. Both methods markedly decreased the final position error, with the median method mitigating the impact of some outliers. Skog et al. [27] proposed a maximum likelihood estimation method to integrate IMU array measurements, transforming the fusion process into a parameter estimation problem. Waegli et al. [19] proposed a method that does not directly fuse IMU measurements but adjusts the INS mechanization algorithm to adapt to multiple IMUs. Results indicated improved navigation performance compared to direct IMU fusion, as the systematic errors of each IMU are separately estimated. Wang et al. [28] proposed a weighted fusion method for IMU arrays, which takes into account the performance differences of individual chips and adopts a targeted scheme to allocate weights. The experiments proved that this scheme is significantly superior to the average fusion method.

INS-level fusion algorithms integrate the navigation solutions from IMUs, resulting in a more complex system structure. Bancroft [21] employed a centralized filter to integrate navigation solutions from multiple IMUs, reducing the position drift through relative position constraints.

Skog et al. [24] designed a state-constrained KF, which employs the distance upper bound constraint to associate IMUs distributed on two feet. Wägli [29] proposed a geometric constraint fusion algorithm, computing the navigation solution for each IMU, periodically comparing these solutions, and enabling the estimation of systematic errors for all IMUs. Skog et al. [26] fused navigation solutions from 14 IMUs using the mean value, significantly improving navigation performance. Besides, they compared the navigation performance of IMU-level and INS-level fusion and found a negligible difference between these two methods. However, they only conducted experiments on short trajectories and rigid arrays.

The literature survey above indicates numerous studies on the navigation performance of rigid IMU arrays, and it has been shown that there is a significant improvement compared to a single IMU. However, there are a few studies on the navigation performance of nonrigid (i.e., flexible installed) IMU arrays. Regarding the data fusion algorithm perspective, the IMU-level fusion necessitates a certain transformation relationship between the measurements of different IMUs. This requirement is not applicable to the nonrigid arrays, leading to poor robustness and even degraded navigation performance of the IMU array. The INS-level fusion, while relatively robust in dealing with the uncertain transformation relationships between IMU measurements, faces the challenge of high computational complexity.

III. METHODOLOGY

This section first introduces the framework of the proposed eNav-Fusion and then introduces the INS mechanization algorithm and the INS/GNSS integrated navigation algorithm. On this basis, the proposed eNav-Fusion is elaborated, followed by a theoretical analysis of its navigation performance and computational complexity.

A. System Overview

The system overview of the proposed eNav-Fusion is depicted in the abstract. In the proposed eNav-Fusion, each IMU in the IMU array independently performs INS mechanization. Subsequently, data fusion of the inertial navigation solutions and IMU measurements is executed. Then, the fused inertial navigation solution and measurement are utilized for state prediction, thus making all IMUs share the same state covariance. Subsequently, each IMU is individually updated using the same GNSS position measurement, ensuring the estimation of systematic errors for each IMU. When the GNSS measurement is available, the navigation states of all IMUs with error feedback and correction are fused, and the fused result serves as the navigation solution of the IMU array; otherwise, when the GNSS measurement is not available, the INS solutions of all IMUs are fused and serve as the IMU array's solution.

B. Proposed eNav-Fusion

1) *INS Mechanization Algorithm*: The IMU frame (b frame) is defined as the IMU body frame, i.e., the front-right-down frame. The attitude (represented by \mathbf{q}_b^n or \mathbf{C}_b^n), velocity (represented by \mathbf{v}_{eb}^n), and position are derived in the navigation

frame (n frame, the north-east-down frame). The kinematic model in the n frame can be written as

$$\begin{aligned} \dot{\mathbf{q}}_b^n &= \frac{1}{2} \mathbf{q}_b^n \otimes \begin{bmatrix} 0 \\ \boldsymbol{\omega}_{nb}^b \end{bmatrix} \\ \dot{\mathbf{v}}_{eb}^n &= \mathbf{C}_b^n \mathbf{f}^b - (2\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) \times \mathbf{v}_{eb}^n + \mathbf{g}^n \\ \dot{\varphi} &= \frac{v_N}{R_M + h}, \dot{\lambda} = \frac{v_E}{(R_N + h) \cos \varphi}, \dot{h} = -v_D \end{aligned} \quad (1)$$

where \otimes represents the quaternion product, $(\cdot) \times$ represents the antisymmetric matrix, e frame is the Earth frame, φ , λ , and h are the latitude, longitude, and ellipsoid height, respectively, $\boldsymbol{\omega}_{nb}^b$ is the projection vector of the angular velocity of the b frame with respect to the n frame in the b frame, \mathbf{f}^b is the specific force measured by the accelerometer, \mathbf{g}^n is the local gravity, R_M and R_N represent the meridian curvature radius and the curvature radius of the ellipse, respectively, and v_N , v_E , and v_D represent the northward, eastward, and vertical velocity, respectively. The INS mechanization can be formulated by adopting the kinematic model in (1) to obtain high-frequency inertial navigation solutions.

2) *INS/GNSS Integrated Navigation Algorithm*: The INS and GNSS are loosely coupled, and KF is used for integrated navigation. When designing the KF, the system state vector includes the navigation state error and the IMU error, which is expressed as

$$\delta \mathbf{x} = [\delta \mathbf{r}^n, \delta \mathbf{v}^n, \phi, \mathbf{b}_g, \mathbf{b}_a, \mathbf{s}_g, \mathbf{s}_a] \quad (2)$$

where $\delta \mathbf{r}^n$, $\delta \mathbf{v}^n$, and ϕ are the inertial navigation position error, velocity error, and attitude error, respectively; \mathbf{b}_g and \mathbf{b}_a are the gyroscope and accelerometer bias errors, respectively; \mathbf{s}_g and \mathbf{s}_a are the gyroscope and accelerometer scale factor errors, respectively. The bias and scale factors are modeled as the first-order Gauss–Markov processes [30].

The discrete-time system equation is

$$\delta \mathbf{x}_k = \Phi_{k/k-1} \delta \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \quad (3)$$

where the subscript of $\delta \mathbf{x}$ is the corresponding moment; $\Phi_{k/k-1}$ is the state transition matrix from t_{k-1} to t_k ; the specific expression is shown in [30]; \mathbf{w}_{k-1} is the system noise.

In the prediction stage, according to the state transition matrix, the system state and its covariance are updated

$$\hat{\delta \mathbf{x}}_{k/k-1} = \Phi_{k/k-1} \hat{\delta \mathbf{x}}_{k-1} \quad (4)$$

$$\mathbf{P}_{k/k-1} = \Phi_{k/k-1} \mathbf{P}_{k-1} \Phi_{k/k-1}^T + \mathbf{Q}_{k-1} \quad (5)$$

where the hat above $\delta \mathbf{x}$ represents the estimate value; $\hat{\delta \mathbf{x}}_{k/k-1}$ and $\mathbf{P}_{k/k-1}$ are the predicted value of the system state and its covariance at t_k , respectively; $\hat{\delta \mathbf{x}}_{k-1}$ and \mathbf{P}_{k-1} are the optimal estimates of the system state and its covariance at t_{k-1} , respectively; \mathbf{Q}_{k-1} is the covariance matrix of the system noise. Please note that, after the KF measurement update, the error state $\delta \mathbf{x}$ will be fed back and then reset to zero. As a result, the state vector remains zero all the time during the prediction. Therefore, the state prediction in formula (4) has no need to be performed in practice; only the state covariance prediction in formula (5) needs to be executed.

In the measurement update, the GNSS position measurement is the position of the antenna phase center, so the KF

observation vector is computed by subtracting the GNSS position measurement from the GNSS antenna position calculated by INS. The observation equation can be written as

$$\delta \mathbf{z}_k = \mathbf{H}_k \delta \mathbf{x}_k + \mathbf{n}_k \quad (6)$$

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & (\mathbf{C}_{b_k}^n \mathbf{l}) \times & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \quad (7)$$

where $\delta \mathbf{z}_k$ is the observation vector, \mathbf{H}_k is the design matrix, and \mathbf{n}_k is the error of GNSS position measurement, which is modeled as a Gaussian white noise sequence [30], that is, $\mathbf{n}_k \sim N(0, \mathbf{R}_k)$, $\mathbf{R}_k = E[\mathbf{n}_k \mathbf{n}_k^T]$; \mathbf{l} is the lever arm of the GNSS antenna. Therefore, the GNSS position measurement update can be expressed as

$$\mathbf{K}_k = \mathbf{P}_{k/k-1} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_{k/k-1} \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (8)$$

$$\delta \hat{\mathbf{x}}_k = \delta \hat{\mathbf{x}}_{k/k-1} + \mathbf{K}_k (\delta \mathbf{z}_k - \mathbf{H}_k \delta \hat{\mathbf{x}}_{k/k-1}) \quad (9)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k/k-1} (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T \quad (10)$$

where \mathbf{K}_k is the Kalman gain. Since the error state $\delta \hat{\mathbf{x}}_{k/k-1}$ remains zero before the measurement update, formula (9) can be simplified as

$$\delta \hat{\mathbf{x}}_k = \mathbf{K}_k \delta \mathbf{z}_k. \quad (11)$$

Finally, $\delta \hat{\mathbf{x}}_k$ is fed back to the navigation state and IMU error. The corrected navigation state is the system's output, and the corrected IMU error will compensate for IMU measurements of the next epoch.

3) *Proposed Data Fusion Algorithm*: When integrating different IMUs in the IMU array with GNSS, numerous identical operation steps and similar operation results can be shared to reduce considerable computational complexity without significantly impacting navigation performance. Consequently, we conducted an analysis of the simplified operation steps and results and designed the corresponding improved algorithm.

During covariance propagation, according to (5), the update of $\mathbf{P}_{k/k-1}$ is determined by the state transition matrix $\Phi_{k/k-1}$ and the noise \mathbf{Q}_{k-1} involved in the propagation process. As demonstrated in [30], $\Phi_{k/k-1}$ depends on the navigation state and IMU measurement, while \mathbf{Q}_{k-1} is associated with the navigation state and noise performance of the IMU. For the IMU array, the measurement (after correction) and navigation state of each IMU are similar enough, and $\Phi_{k/k-1}$ and \mathbf{Q}_{k-1} during state propagation are essentially identical. Hence, the IMUs within the IMU array can share a common $\mathbf{P}_{k/k-1}$: the shared $\mathbf{P}_{k/k-1}$ is updated using the fused IMU measurement and navigation state. Similarly, when computing the GNSS position design matrix \mathbf{H}_k , the lever arm and navigation state for different IMUs are also close. The fused lever arm and navigation state can be employed to compute \mathbf{H}_k , which can then serve as the design matrix shared by all IMUs, enabling all IMUs to share the same Kalman gain \mathbf{K}_k . The specific procedure of the proposed eNav-Fusion is detailed as follows, as presented in Fig. 2.

First, INS mechanization is performed independently for each IMU in the array. Then, the inertial navigation solutions are fused by projecting them onto the IMU array frame and averaging the projected solutions. IMU measurements

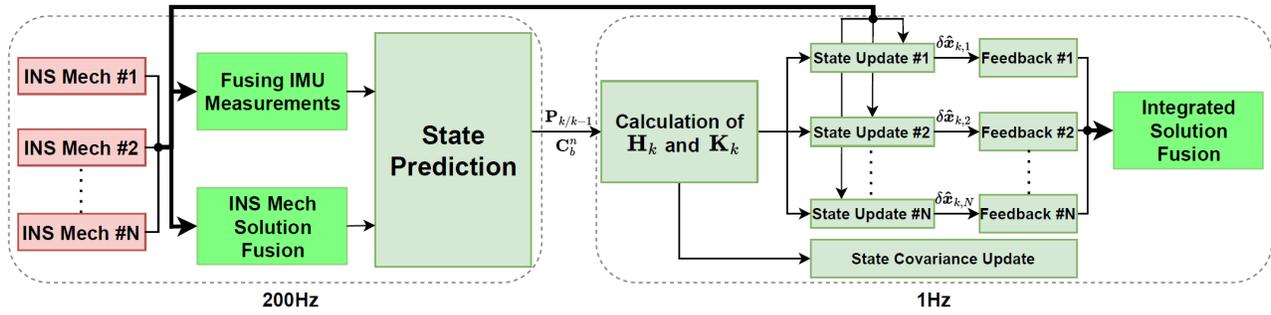


Fig. 2. Flowchart of the proposed eNav-Fusion.

are also fused in the same way. The fused inertial navigation solutions and IMU measurements are then employed for covariance propagation. That is, when updating $\mathbf{P}_{k/k-1}$ through formula (5), the fused inertial navigation solutions and IMU measurements are used to compute $\Phi_{k/k-1}$ and \mathbf{Q}_{k-1} . The obtained $\mathbf{P}_{k/k-1}$ serves the state covariance shared by all IMUs. During the measurement update, the fused navigation state and the lever arm of the GNSS antenna to the IMU array frame are utilized to compute \mathbf{H}_k , which serves as a common design matrix shared by all IMUs. According to formula (8), with the same $\mathbf{P}_{k/k-1}$, \mathbf{H}_k , and \mathbf{R}_k , every IMU shares the same Kalman gain \mathbf{K}_k . Similarly, they share the same \mathbf{P}_k in formula (10). However, the observation vector of each IMU is computed, respectively, using their own navigation state and the GNSS position measurement. Then, $\delta\hat{\mathbf{x}}_k$ in formula (11) is computed individually and fed back to each IMU's navigation solution and systematic error. Finally, the navigation solutions after correction are again fused to acquire the navigation solution of the IMU array. The steps of IMU measurements and navigation states/solutions fusion are as follows.

The fusion of IMU measurements is described first. The frame of the i th IMU is denoted as b_i frame, and the frame of the IMU array is denoted as b_c frame. The corresponding axes of each IMU and the IMU array frame are parallel. First, the measurements of each IMU are projected onto b_c frame

$$\boldsymbol{\omega}_i^{b_c} = \mathbf{C}_{b_i}^{b_c} \boldsymbol{\omega}^{b_i} \quad (12)$$

$$\mathbf{f}_i^{b_c} = \mathbf{f}^{b_i} - \boldsymbol{\omega}^{b_i} \times (\boldsymbol{\omega}^{b_i} \times \mathbf{l}^{b_i}) - \dot{\boldsymbol{\omega}}^{b_i} \times \mathbf{l}^{b_i} \quad (13)$$

where $\boldsymbol{\omega}^{b_i}$ and $\dot{\boldsymbol{\omega}}^{b_i}$ are the angular velocity and angular acceleration of the i th IMU, respectively. Then, the measurements $\boldsymbol{\omega}^{b_c}$ and \mathbf{f}^{b_c} of the IMU array are obtained by averaging

$$\boldsymbol{\omega}^{b_c} = \frac{1}{N} \sum \boldsymbol{\omega}_i^{b_c}, \quad \mathbf{f}^{b_c} = \frac{1}{N} \sum \mathbf{f}_i^{b_c}. \quad (14)$$

Then, the fusion of navigation states/solutions is presented. The navigation states of the i th IMU are represented by the geodetic coordinate \mathbf{r}_i^n , velocity \mathbf{v}_i^n , and attitude $\mathbf{C}_{b_i}^n$. First, the navigation states are projected onto b_c frame

$$\mathbf{r}_{c,i}^n = \mathbf{r}_i^n + \mathbf{D}_{R,i}^{-1} \mathbf{C}_{b_i}^n \mathbf{l}^{b_i} \quad (15)$$

$$\mathbf{v}_{c,i}^n = \mathbf{v}_i^n + \mathbf{C}_{b_i}^n (\boldsymbol{\omega}_{e b_i}^{b_i} \times \mathbf{l}^{b_i}) \quad (16)$$

$$\mathbf{C}_{b_c,i}^n = \mathbf{C}_{b_i}^n \mathbf{C}_{b_c}^{b_i} \quad (17)$$

where $\mathbf{D}_{R,i}^{-1}$ is employed to transform the northward, eastward, and vertical position differences in the n frame into latitude,

longitude, and elevation differences. Before fusion, $\mathbf{C}_{b_c,i}^n$ is converted to Euler angle $\boldsymbol{\alpha}_{nb_c,i}$. Finally, the navigation states of the IMU array are derived by averaging $\mathbf{r}_{c,i}^n$, $\mathbf{v}_{c,i}^n$, and $\boldsymbol{\alpha}_{nb_c,i}$, denoted as \mathbf{r}_c^n , \mathbf{v}_c^n , and $\boldsymbol{\alpha}_{nb_c}$.

C. Theoretical Performance Analysis of eNav-Fusion

1) *Navigation Performance*: The primary distinction between the proposed eNav-Fusion and the classic navigation solution level fusion algorithm (Nav-Fusion, refer to Section IV-B for details) is that each IMU shares the same state covariance $\mathbf{P}_{k/k-1}$ and Kalman gain \mathbf{K}_k . Based on the previous analysis, $\mathbf{P}_{k/k-1}$ is associated with the measurement and navigation state of the IMU. Generally, in the rigid IMU array, the measurement and navigation state of each IMU are similar; hence, $\mathbf{P}_{k/k-1}$ of different IMUs is similar. \mathbf{K}_k depends on $\mathbf{P}_{k/k-1}$ and the reliability of GNSS observation, represented by \mathbf{R}_k . All IMUs employ the same GNSS observation; hence, \mathbf{R}_k remains the same for different IMUs. Therefore, \mathbf{K}_k of different IMUs is similar. Consequently, sharing a common state covariance and Kalman gain for all IMUs in the proposed eNav-Fusion will not significantly impact the navigation performance compared to Nav-Fusion.

When IMUs are nonrigidly connected, there are some differences in the measurements and navigation states among different IMUs, leading to differences in $\mathbf{P}_{k/k-1}$. However, due to the same overall motion of IMUs, the differences in $\mathbf{P}_{k/k-1}$ will not consistently increase. Taking six nonrigidly connected IMUs with a maximum lever arm of about 46 cm as an example, the INS/GNSS integrated navigation algorithm was implemented individually for each IMU. The position error standard deviation, defined as the square root of the first three diagonal elements of $\mathbf{P}_{k/k-1}$, for the six IMUs at the time interval 51–53 s, is illustrated in Fig. 3. Among them, the most noticeable difference lies in the standard deviation of eastward errors. Specifically, all six IMUs had the same initial value. After 1 s of propagation, IMUs 1 and 5 exhibited errors around 0.193 m, whereas other IMUs ranged between 0.155 and 0.17 m, with a maximum gap of approximately 0.038 m. However, after the update of GNSS measurement, the standard deviation of the six IMUs was significantly reduced, ranging between 0.018 and 0.021 m, with the maximum gap not exceeding 0.003 m. The experimental results indicated that the variations in $\mathbf{P}_{k/k-1}$ of different IMUs are similar, and the differences are maintained within a small range. Therefore, despite the nonrigid connection of IMUs,

TABLE I
FLOATING-POINT OPERATION COUNT OF EACH STEP

Operation Step	Addition	Multiplication	SUM	
INS Mechanization	179	134	313	
Prediction	$\Phi_{k/k-1}$	535	732	1267
	Q_{k-1}	32004	33706	65710
	$P_{k/k-1}$	18081	18522	36603
	SUM	50620	52960	103580
Update	H_k	9	6	15
	K_k	2862	3054	5916
	P_k	19656	20223	39879
	δz_k	65	46	111
	$\delta \hat{x}_k$	42	63	105
	Feedback	38	48	86
	SUM	22756	23503	46112
Fusion (each IMU)	IMU Measurement	144	196	340
	Navigation Solution	18	33	51

the navigation performance of the proposed eNav-Fusion will not be significantly diminished compared to Nav-Fusion.

2) *Computational Complexity*: The INS/GNSS integrated navigation algorithm mainly consists of INS mechanization, state prediction, and measurement update. The frequency of mechanization and prediction depends on the data rate of IMU, while the update frequency corresponds to the data rate of GNSS. Generally, the data rate of IMU (e.g., 200 Hz) is much higher than that of GNSS (e.g., 1 Hz). The INS mechanization only involves low-dimensional matrix operations, while the prediction involves multiple high-dimensional matrix operations. Therefore, the computational complexity of the integrated navigation algorithm primarily depends on the prediction. In the proposed eNav-Fusion, different IMUs share the prediction process, resulting in significant savings in computational complexity. A detailed quantitative analysis is conducted as follows.

The number of floating-point operations is used to gauge the computational complexity. Floating-point operations include addition (including subtraction) and multiplication (including division). Modern navigation computer CPUs generally include a hardware multiplier, making the time spent on different floating-point operations comparable. Consequently, no distinction is made when evaluating the computational complexity.

First, the floating-point operation count of each component in the integrated navigation algorithm is analyzed, as shown in Table I. As mentioned in Section III-B, after the measurement update, the error state will be fed back and then reset to zero, mitigating any additional computational complexity arising from the error state prediction in formula (4). The KF prediction consumes the most floating-point operations, followed by the measurement update, and finally, the INS mechanization.

TABLE II
COUNT OF EXECUTIONS PER STEP WITHIN 1 S FOR DIFFERENT ALGORITHMS

Operation Step	Single	Nav-Fusion	eNav-Fusion	
INS Mechanization (200Hz)	200	$200 * N$	$200 * N$	
Prediction (200Hz)	200	$200 * N$	200	
Update (1Hz)	H_k, K_k, P_k	1	N	1
	$\delta z_k, \delta \hat{x}_k$, Feedback	1	N	N
IMU Measurement Fusion	0	0	$200 * N$	
Navigation Solution Fusion	0	$200 * N$	$(200 * 2 + 1) * N$	

Single represents the INS/GNSS integrated navigation algorithm for a single IMU; the data rate of IMU and GNSS is 200Hz and 1Hz, respectively; N is the number of IMUs in the IMU array.

Assuming IMU and GNSS data rates are 200 and 1 Hz, respectively, and the number of IMUs is denoted by N , the count of executions per step within 1 s for INS/GNSS integrated navigation algorithm of a single IMU, Nav-Fusion, and eNav-Fusion is analyzed, as listed in Table II. Nav-Fusion repeats each step of INS/GNSS integrated navigation algorithm of a single IMU N times and increases the computational complexity of fusing navigation solutions. Therefore, its computational complexity exceeds N times that of the single IMU algorithm. For the proposed eNav-Fusion, the count of high-dimensional matrix operations is independent of the number of IMUs in the array, significantly improving computational efficiency. Besides, the proposed eNav-Fusion introduces additional computational complexity to data fusion beyond the single IMU algorithm. Specifically, the prediction process fuses the navigation solutions and IMU measurements of the previous moment (200 Hz), and the update process fuses the uncorrected navigation solutions of the current moment (1 Hz). The output process fuses the navigation solutions of the current moment (200 Hz).

The floating-point operation counts for INS mechanization, prediction, and update are denoted as N_{InsMech} , N_{Predict} , and N_{Update} , respectively; the sum of the floating-point operation counts for calculating δz_k and $\delta \hat{x}_k$ and performing feedback is denoted as $N_{\text{UpdateEach}}$; the floating-point operation counts for fusing the IMU measurement and navigation solution of each IMU are denoted as N_{IMU} and N_{Nav} , respectively. According to Table II, the floating-point operation count of the integrated navigation algorithm for a single IMU in 1 s is given by $200(N_{\text{InsMech}} + N_{\text{Predict}}) + N_{\text{Update}}$, while the floating-point operation count of the proposed eNav-Fusion is given by $200[(N(N_{\text{InsMech}} + 2N_{\text{NavState}} + N_{\text{IMU}}) + N_{\text{Predict}})] + N_{\text{Update}} + (N - 1)N_{\text{UpdateEach}} + N * N_{\text{NavState}}$. Substituting the values in Table I, the floating-point operation count of the integrated navigation algorithm for a single IMU in 1 s is 20 824 712, while the floating-point operation count of the proposed eNav-Fusion is $20761810 + 209442 * N$. The increase in floating-point operation count introduced by the proposed eNav-Fusion for each additional IMU is only 1.01% of the floating-point operation count of the single IMU algorithm. For N equal to 6, 16, and 100, the floating-point operation count of

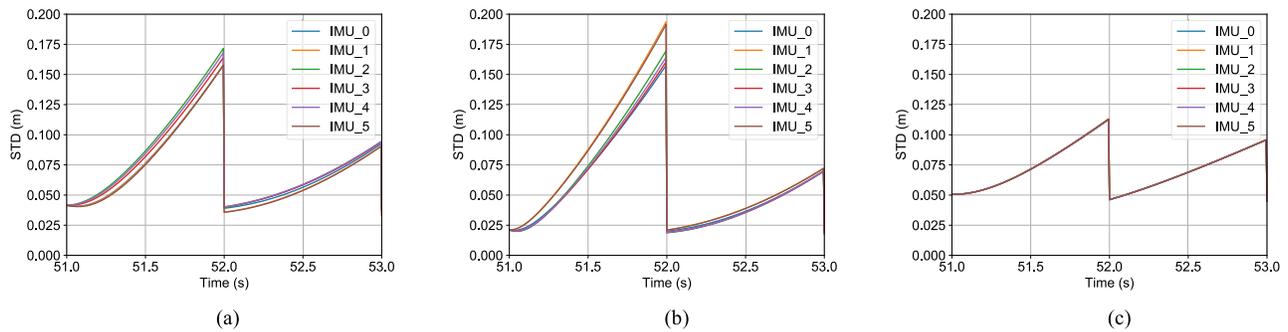


Fig. 3. Standard deviation of IMU position in array. (a) North. (b) East. (c) Vertical.

the proposed eNav-Fusion is $1.06\times$, $1.16\times$, and $2.00\times$ that of the single IMU algorithm. Thus, the computational complexity of the proposed eNav-Fusion exhibits no significant increase compared to the single IMU algorithm, making a significant improvement over the current Nav-Fusion algorithm.

IV. EXPERIMENTS AND RESULTS

To accurately evaluate the performance of the proposed eNav-Fusion, we designed four groups of rigid arrays, each consisting of 16 rigidly connected IMUs, along with one flexible array comprising six nonrigidly connected IMUs. Sufficient tests were conducted on both vehicles and wheeled robots. This section first verified the navigation performance of the proposed eNav-Fusion through comparative analysis. Then, the optimization design of the proposed eNav-Fusion was tested and analyzed. Finally, we conducted a discussion on the navigation performance and computational complexity of the proposed eNav-Fusion.

A. Experimental Description

1) *Rigid IMU Array*: We utilized the experimental equipment and data from our previous work [23] to verify the performance of the proposed data fusion algorithm. In [23], we used 16 pieces of MEMS IMU ICM20602 to form an IMU array, which was welded on a PCB board, ensuring rigid installation. To guarantee accurate testing of the IMU array's performance, we built four sets of such IMU arrays. A photograph of the rigid IMU arrays is shown in Fig. 4. The vehicle experiment was conducted under open-sky conditions to obtain high-precision reference results. Please note that, in the data processing of this article, we intentionally used raw IMU measurements without turntable calibration, so as to keep the proposed solution feasible and cost down.

2) *Nonrigid IMU Array*: The nonrigid array comprised six pieces of ICM20602 modules, mounted on three flexible steel rulers, as illustrated in Fig. 5. This nonrigid array was mounted on a wheeled robot with a maximum speed of 1.5 m/s. The integrated navigation solution of a navigation-grade IMU Ledor-A15 and the high-precision GNSS positioning result obtained by postprocessed kinematic (PPK) served as the reference truth. The position accuracy is 0.02 m, and the attitude accuracy is 0.01° .

The nonrigid array experiment was conducted on an open playground, and the test trajectory is illustrated in Fig. 6, with the green and red trajectories representing tests A and

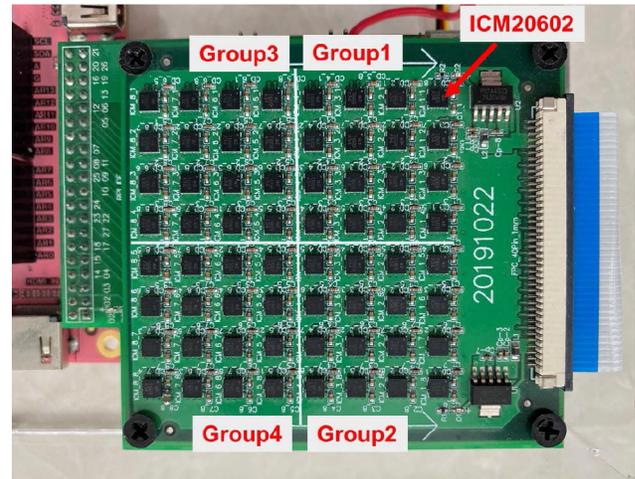


Fig. 4. Photograph of the rigid IMU arrays.

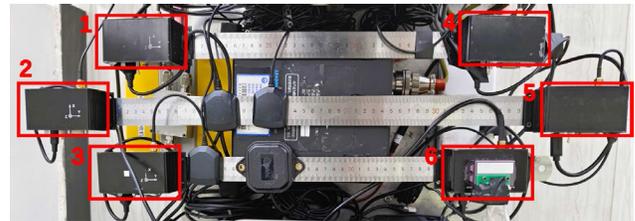


Fig. 5. Installation of the nonrigid IMU array (top view, IMUs hold by steel rulers).

B, respectively. Test A was conducted on the uneven and bumpy lawn at the center of the playground, resulting in significant vibrations as the robot drove on the lawn. This led to diverse motion perceptions among different IMUs, indicating a nonrigid installation. Test B was carried out on the playground runway, with six books of varying thicknesses placed to mimic the ground obstacles in real-world scenarios. As depicted in Fig. 7, the robot encountered the books during its movement, causing it to generate noticeable vibrations, which resulted in a substantial angular velocity along the x -axis of the gyroscope and a significant acceleration along the y - and z -axes of the accelerometer.

B. Data Processing

We employed INS/GNSS integrated navigation algorithm for a single IMU to compute the positioning results of each IMU. Additionally, we utilized the classic IMU array

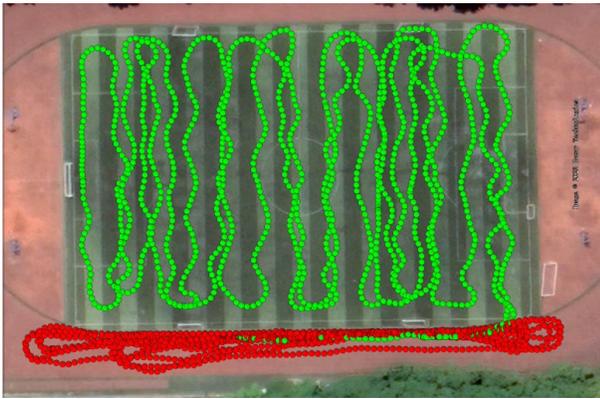


Fig. 6. Trajectory of nonrigid array tests.



Fig. 7. Robot passed through the ground obstacle in test B.

IMU-level fusion algorithm (IMU-Fusion), navigation solution level fusion algorithm (Nav-Fusion), and the proposed eNav-Fusion to compute the positioning results of IMU arrays. IMU-Fusion and Nav-Fusion fuse IMU measurements or navigation solutions by averaging, as demonstrated in [26]. During the initialization stage of INS, the gyroscope measurements obtained when the vehicle/robot was stationary were performed as the initial bias of the gyroscope, so that the navigation state converged quickly.

To evaluate the navigation performance of the IMU array, GNSS outage occurred after the vehicle or robot moved for a while. In the rigid array experiment and nonrigid array test A, the outage length and interval were 30 and 90 s, respectively. In nonrigid array test B, GNSS outages were manually set without periodicity, with each outage lasting for 30 s. The commencement of each outage coincided with the moment the front wheel of the robot rolled over the book for the first time during each round trip, which aimed to comprehensively demonstrate the navigation performance of the nonrigid array during structural motion deformation.

Next, the positioning results of the IMU array solved by IMU-Fusion, Nav-Fusion, and the proposed eNav-Fusion were compared with the reference truth, yielding positioning error sequences for the IMU array. The maximum position drift error during all GNSS outage periods was then extracted, and the rms value of the maximum position drift errors across all outages was calculated, serving as the positioning error

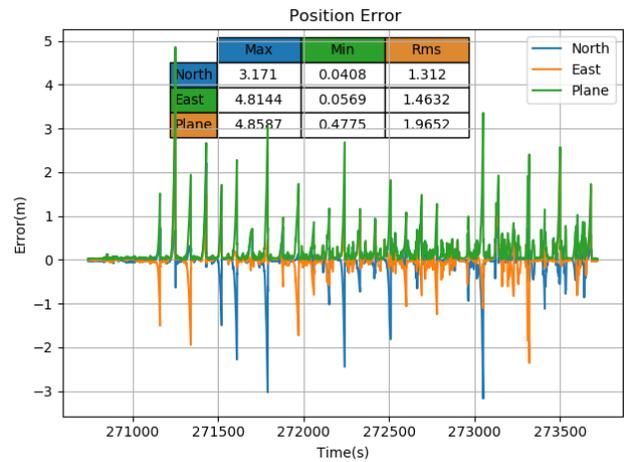


Fig. 8. Position error of IMU array (Group1 as example).

of the IMU array [31]. This approach was also employed to assess the positioning error of each IMU, with the rms value of the positioning errors for all IMUs serving as the positioning error of a single IMU. It is worth mentioning that the MEMS IMU, ICM20602 produced low-frequency noise in the z -axis measurement of the accelerometer in a static state, which affects the performance along the IMU array's z -axis [23]. Therefore, we exclusively evaluated the horizontal positioning error. After fusion, the random error of each axis is theoretically reduced by $(N)^{1/2}$ times, where N is the number of IMUs. Therefore, the theoretical reduction factor for the horizontal positioning error should also be $(N)^{1/2}$ times.

Finally, the positioning errors of the IMU array and a single IMU were compared to evaluate the navigation performance of the proposed eNav-Fusion. The evaluation was conducted in comparison with IMU-Fusion and Nav-Fusion.

C. Results of Rigid Array Experiment

1) *Navigation Performance*: The positioning errors of the four rigid arrays were computed using the data processing method mentioned above. The error drift curve of Group1 solved by the proposed eNav-Fusion is illustrated in Fig. 8, with a total of 29 GNSS outage test samples, ensuring the accuracy and persuasiveness of the evaluation. The horizontal position errors of a single IMU and the arrays are presented in Table III. The ratio of the horizontal error of a single IMU to the horizontal error of the IMU array was calculated, serving as the improvement effect on the navigation performance of the IMU array.

Due to the differences in the individual IMUs and the random errors during testing, including GNSS positioning errors and differences in vehicle dynamics, there were differences in the improvement effects among different IMU arrays. For instance, when using Nav-Fusion, Group1's navigation performance was improved by $2.88\times$ (rms), while the worst improvement effect of Group3 was only $2.11\times$ (rms).

Despite variations in results among different IMU array tests due to random errors, the three algorithms reduced the horizontal position error of the IMU array by $2.53\times$, $2.47\times$, and $2.51\times$ (rms) compared to a single IMU, achieving 63.3%,

TABLE III
30-S GNSS OUTAGE HORIZONTAL POSITION ERROR OF FOUR RIGID IMU ARRAYS

IMU Array	Single IMU Error	IMU Array					
		IMU-Fusion		Nav-Fusion		eNav-Fusion	
		Error	Ratio	Error	Ratio	Error	Ratio
Group1	5.68m	1.98m	2.87	1.97m	2.88	2.05m	2.77
Group2	4.23m	1.66m	2.55	1.71m	2.47	1.63m	2.60
Group3	3.94m	1.74m	2.26	1.87m	2.11	1.72m	2.29
Group4	5.42m	2.27m	2.39	2.31m	2.35	2.31m	2.35
RMS	4.87m	1.93m	2.53	1.98m	2.47	1.95m	2.51

The ratio is the error of Single IMU over that of the IMU Array.

61.8%, and 62.8% of the theoretical value (i.e., four times), respectively. The results indicated no significant difference in terms of the navigation performance among the three algorithms. It is clear that for a rigid array, both fusing IMU measurements and fusing navigation solutions can effectively reduce random errors, and their ability to reduce random errors is essentially equivalent. Besides, the simplification of computational complexity designed by the proposed eNav-Fusion has no significant impact on the navigation performance for the rigid IMU arrays.

We also observed a certain gap between the navigation performance improvement effect of the three algorithms and the theoretical factor of 4. The reasons primarily stem from two factors. One is that errors such as cross-axis coupling of MEMS IMU are not compensated, and the other is the common-mode circuit noise, both of which result in the measurement errors of IMUs not being completely independent.

2) Computational Efficiency: The three data fusion algorithms were implemented on the desktop PC (AMD R7-3700X, Ubuntu 20.04.6 LTS) and employed to solve the four arrays. The duration of each array's data was 1000 s. The running time for each algorithm was recorded. To reduce randomness, each computation was repeated three times, and the rms values of the three computation times are presented in Table IV.

According to Table IV, the running time of IMU-Fusion was essentially identical to that of the INS/GNSS integrated navigation algorithm for a single IMU, as it only increased the computational complexity of fusing IMU measurements, which can be ignored. The running time of Nav-Fusion was $16.08\times$ that of the algorithm for a single IMU, closely aligning with the number of IMUs, given that each IMU requires separate solving for integrated navigation. The proposed eNav-Fusion exhibited a running time of only $1.22\times$ that of the algorithm for a single IMU, consistent with theoretical analysis ($1.16\times$). It is noteworthy that the running time is not only related to the number of floating-point operations but also affected by various factors, such as space complexity and memory access times. Therefore, the ratio of the running time and the ratio of the number of floating-point operations among different algorithms are not exactly the same.

Processing 1 s of the integrated navigation data, the average times for the integrated navigation algorithm for a single IMU, Nav-Fusion, and the proposed eNav-Fusion were 4.67, 75, and 5.71 ms, respectively. Compared with the algorithm for a sin-

gle IMU, the running time of the proposed eNav-Fusion only increased by 1.04 ms (22.27%). On the other hand, the running time of the proposed eNav-Fusion was only 7.61% of the Nav-Fusion. In summary, the proposed eNav-Fusion demonstrates navigation performance similar to the Nav-Fusion without imposing a substantial computational load compared to the integrated navigation algorithm for a single IMU.

D. Results of Nonrigid Array Experiment

1) Navigation Performance: The horizontal positioning errors of nonrigid array tests A and B are detailed in Table V. The horizontal errors of a single IMU for the two tests were 8.05 and 8.41 m, respectively, which were markedly higher than 4.87 m of the rigid array. This difference was primarily attributed to severe vibration in these two tests, which led to continuous changes in the lever arm of the GNSS antenna, thereby increasing the positioning error. The deteriorated positioning errors of such single IMU should not be used as the benchmark to evaluate the improved navigation performance of the nonrigid IMU array. Therefore, we used the single IMU positioning error (i.e., 4.87 m) of the rigid array experiment to calculate the improvement ratio of the navigation performance.

In test A, the horizontal position errors of the IMU array in IMU-Fusion degraded compared to that of a single IMU (with a ratio of 0.61), instead of any improvement. This outcome could be attributed to the continuous and severe vibration in test A, resulting in a high degree of nonrigidity. Nav-Fusion and the proposed eNav-Fusion, which fuse navigation solutions, can still bring significant performance improvements for the nonrigid IMU array, enhancing the navigation performance by $1.56\times$ and $1.63\times$, respectively.

In test B, IMU-Fusion improved the navigation performance of the IMU array by only $1.16\times$, still yielding some performance improvements but very limited. This was attributed to the fact that the evident nonrigid connection between the IMUs happened only for part of the GNSS outage time, while for the other part, close to rigid connection, fusing the IMU measurements effectively reduced the positioning error. Nevertheless, Nav-Fusion and the proposed eNav-Fusion exhibited higher navigation performance, reducing horizontal error by $1.48\times$ and $1.51\times$, respectively.

Tests A and B proved that for nonrigidly connected IMUs, due to the failure of the IMU-Fusion to assume rigid installation between IMUs, the rationality of fusing the raw measurements of IMUs is lost. Consequently, IMU-Fusion is unable

TABLE IV
RUNNING TIME OF 1000-S RIGID IMU ARRAY DATA

IMU Array	Single IMU Time	IMU Array					
		IMU-Fusion		Nav-Fusion		eNav-Fusion	
		Time	Ratio	Time	Ratio	Time	Ratio
Group1	4.67s	4.99s	1.07	74.91s	16.04	5.71s	1.22
Group2	4.67s	4.98s	1.07	75.04s	16.07	5.71s	1.22
Group3	4.66s	5.00s	1.07	75.20s	16.14	5.72s	1.23
Group4	4.66s	5.04s	1.08	74.85s	16.06	5.70s	1.22
RMS	4.67s	5.00s	1.07	75.00s	16.08	5.71s	1.22

The ratio is the running time of the IMU Array over that of Single IMU.

TABLE V
30-S GNSS OUTAGE HORIZONTAL POSITION ERROR OF THE NONRIGID IMU ARRAY

Test Group	Single IMU Error	IMU Array					
		IMU-Fusion		Nav-Fusion		eNav-Fusion	
		Error	Ratio	Error	Ratio	Error	Ratio
A	8.05m(4.87m*)	8.04m	0.61	3.12m	1.56	2.99m	1.63
B	8.41m(4.87m*)	4.21m	1.16	3.28m	1.48	3.23m	1.51

* Here use the error of Single IMU in the rigid array (i.e. 4.87m) as the benchmark to calculate the ratio. The ratio is the error of Single IMU over that of the IMU Array.

to effectively improve the navigation performance of the IMU arrays, and there is even a risk of degrading the positioning accuracy. In contrast, the approach of fusing the navigation solutions of each IMU, as adopted in Nav-Fusion and the proposed eNav-Fusion, remains effective in significantly reducing the positioning error and demonstrating superior navigation performance compared to a single IMU. Besides, there was no significant difference in the improvement effect on the navigation performance of these two algorithms, which proved that for a nonrigid array, the algorithm simplification employed by the proposed eNav-Fusion has no significant impact on its navigation performance.

The average improvement of the IMU array navigation performance for the proposed eNav-Fusion in tests A and B was $1.57\times$, which achieved 64.1% of the theoretical value ($(6)^{1/2} \approx 2.45$), while the improvement effect in the rigid array experiment achieved 62.8% of the theoretical value. It is indicated that the proposed eNav-Fusion exhibits the same ability to improve navigation performance in both rigid and nonrigid scenarios, proving its robustness.

2) Computational Efficiency: The algorithms were implemented on the same computer hardware and operating system as the rigid array experiment. Similarly, for the nonrigid array, the time required to solve 1000 s data was recorded and the rms values of the three calculation times are calculated.

The running time of IMU-Fusion was essentially identical to that of the integrated navigation algorithm for a single IMU. The running time of Nav-Fusion was $6.05\times$ that of the integrated navigation algorithm for a single IMU, which was nearly proportional to the number of IMUs. The running time of the proposed eNav-Fusion was $1.09\times$ that of the integrated navigation algorithm for a single IMU, which was consistent with the theoretical analysis of $1.06\times$. This result again proved that the proposed eNav-Fusion significantly improves computational efficiency compared with Nav-Fusion without compromising navigation performance.

E. Discussions

The improved robustness (dealing with nonrigid IMU arrays) of the proposed eNav-Fusion benefits from the fusion of navigation solutions instead of IMU measurements, while the improvement in the computational efficiency, compared with Nav-Fusion, is derived from the simplifications in the KF prediction and measurement update. In addition, there is potential for simplification in the INS mechanization of the navigation solution level fusion. To further validate the rationality of the proposed eNav-Fusion, we conducted the following dedicated tests.

- 1) On Nav-Fusion, simplify the KF prediction and measurement update (i.e., the proposed eNav-Fusion).
- 2) On the above basis, continue to simplify the INS mechanization.

The specific approach to simplify the INS mechanization is to employ the fused navigation state to calculate the relevant intermediate variables for each IMU. Consequently, the intermediate variables related to the navigation state only need to be calculated once instead of N times.

Obviously, for rigid IMU arrays, the navigation states of different IMUs are similar, and the simplification of INS mechanization is unlikely to have a significant impact on navigation performance. However, for nonrigid arrays, the navigation states of different IMUs are somewhat different, and it is difficult to judge how much the simplification in INS mechanization will affect the navigation performance. Therefore, the data from nonrigid array test A were used to evaluate the simplified algorithms. The horizontal position errors of the simplified algorithms are presented in Table VI.

The horizontal error (rms) of the simplification 1) changed from 3.12 to 2.99 m compared with Nav-Fusion, demonstrating similar navigation performance, which proved that the simplification of the KF prediction and measurement update proposed by eNav-Fusion does not compromise the

TABLE VI
HORIZONTAL ERROR OF DIFFERENT SIMPLIFIED ALGORITHMS

Nav-Fusion	30s GNSS Outage Horizontal Error	
	Simplified Algorithm	
	i(eNav-Fusion)	ii
3.12m	2.99m	4.64m

navigation performance of the IMU array. However, the positioning error of simplification 2) was significantly larger than that of the simplification 1). The reason is the significant differences in some navigation states of the nonrigidly connected IMUs, particularly for velocity and attitude. In the nonrigid array test, severe vibration resulted in significant differences in the Y - and Z -axes velocity and the roll angle of different IMUs. Therefore, the corresponding fused navigation states should not be feedback to the INS mechanization of each IMU; otherwise, it would lead to serious accuracy loss. The above results indicate that the computational complexity simplification designed by the proposed eNav-Fusion has no significant impact on navigation performance, and there is limited space for further improvement.

The above experimental results and discussions show that compared with a single IMU, the proposed eNav-Fusion can yield significant navigation performance improvement without substantially increasing computational complexity. Compared with Nav-Fusion, the proposed eNav-Fusion exhibits the lowest computational complexity while maintaining the navigation performance of the IMU array. Additionally, under nonrigid IMU installation conditions where IMU-Fusion is not applicable, the proposed eNav-Fusion still improves the navigation performance as expected. Therefore, the proposed eNav-Fusion exhibits compatibility with navigation performance robustness and computational efficiency.

V. CONCLUSION

This article proposes a navigation solution level data fusion algorithm (eNav-Fusion) for the IMU array that is integrated with GNSS, applicable to both rigidly and nonrigidly installed IMU arrays and computation saving. The proposed eNav-Fusion uses the fused navigation state to calculate intermediate variables in the KF prediction and update processes to save most of the computational complexity without degrading the navigation performance. We conduct comprehensive real-world experiments based on both rigid 16-IMU array and nonrigid 6-IMU array. The results indicate that for rigid and nonrigid IMU arrays, the improvement factor of navigation performance achieved 62.8% and 64.1% of the theoretical value, respectively. Regarding the computational efficiency, the processing time of eNav-Fusion for the 6-IMU array and 16-IMU array is only 1.09 \times and 1.22 \times of the integrated navigation algorithm for a single IMU, respectively.

The proposed eNav-Fusion algorithm was evaluated on consumer-grade IMU arrays in this article. Future work will try the algorithm for higher grade IMU arrays and with different extents of nonrigid installations.

REFERENCES

- [1] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*. London, U.K.: Institution of Engineering and Technology, 2004.
- [2] H. Tang, X. Niu, T. Zhang, Y. Li, and J. Liu, "OdoNet: Untethered speed aiding for vehicle navigation without hardware wheeled odometer," *IEEE Sensors J.*, vol. 22, no. 12, pp. 12197–12208, Jun. 2022.
- [3] H. Tang, T. Zhang, X. Niu, J. Fan, and J. Liu, "Impact of the Earth rotation compensation on MEMS-IMU preintegration of factor graph optimization," *IEEE Sensors J.*, vol. 22, no. 17, pp. 17194–17204, Sep. 2022.
- [4] T. Fan, T. Zhang, H. Zhang, J. Mo, and X. Niu, "A double sideband combined tracking method for Galileo E5 AltBOC signals," *Satell. Navigat.*, vol. 4, no. 1, p. 27, Dec. 2023.
- [5] N. El-Sheimy and A. Youssef, "Inertial sensors technologies for navigation applications: State of the art and future trends," *Satell. Navigat.*, vol. 1, no. 1, pp. 1–21, Dec. 2020.
- [6] G. Zhanshe, C. Fucheng, L. Boyu, C. Le, L. Chao, and S. Ke, "Research development of silicon MEMS gyroscopes: A review," *Microsyst. Technol.*, vol. 21, no. 10, pp. 2053–2066, Oct. 2015.
- [7] N. Ahmad, R. A. R. Ghazilla, N. M. Khairi, and V. Kasi, "Reviews on various inertial measurement unit (IMU) sensor applications," *Int. J. Signal Process. Syst.*, vol. 1, no. 2, pp. 256–262, 2013.
- [8] J.-O. Nilsson and I. Skog, "Inertial sensor arrays—A literature review," in *Proc. Eur. Navigat. Conf. (ENC)*, Helsinki, Finland, May 2016, pp. 1–10.
- [9] I. Skog, J.-O. Nilsson, and P. Händel, "An open-source multi inertial measurement unit (MIMU) platform," in *Proc. Int. Symp. Inertial Sensors Syst. (ISISS)*, Laguna Beach, CA, USA, Feb. 2014, pp. 1–4.
- [10] M. G. Petovello, G. Lachapelle, and M. E. Cannon, "Using GPS and GPS/INS systems to assess relative antenna motion onboard an aircraft carrier for shipboard relative GPS," in *Proc. Nat. Tech. Meeting Inst. Navigat.*, 2005, pp. 219–229.
- [11] J. Kaiser, G. Beck, and S. Berning, "Vital advanced inertial network," in *Proc. IEEE Position Location Navigat. Symp.*, Palm Springs, CA, USA, Apr. 1996, pp. 61–68.
- [12] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering With MATLAB Exercises*, 4th ed. Hoboken, NJ, USA: Wiley, 2012.
- [13] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina Chapel Hill, Chapel Hill, NC, USA, Tech. Rep. TR95-041, 1997.
- [14] D. R. Greenheck, R. H. Bishop, E. M. Jonardi, and J. A. Christian, "Design and testing of a low-cost MEMS IMU cluster for smallsat applications," in *Proc. Small Satell. Conf.*, 2014, pp. 1–10.
- [15] D. R. Greenheck, "Design and characterization of a low cost MEMS IMU cluster for precision navigation," M.S. thesis, Dept. Elect. Eng., Marquette Univ., Milwaukee, WI, USA, 2015.
- [16] S. Bose, A. K. Gupta, and P. Handel, "On the noise and power performance of a shoe-mounted multi-IMU inertial positioning system," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8.
- [17] T. Jing and H. Zhao, "Research on array MEMS inertial sensor and noise suppression," *Chin. J. Sens. Actuators*, vol. 33, no. 6, pp. 779–784, 2020.
- [18] P. Clausen, J. Skaloudl, H. Perakis, and V. Gikas, "Position accuracy with redundant MEMS IMU for road applications," *Eur. J. Navigat.*, vol. 13, no. 2, pp. 4–12, 2015.
- [19] A. Waegli, S. Guerrier, and J. Skaloud, "Redundant MEMS-IMU integrated with GPS for performance assessment in sports," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Monterey, CA, USA, May 2008, pp. 1260–1268.
- [20] S. Guerrier, "Improving accuracy with multiple sensors: Study of redundant MEMS-IMU/GPS configurations," in *Proc. 22nd Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2009, pp. 3114–3121.
- [21] J. Bancroft, "Multiple IMU integration for vehicular navigation," in *Proc. 22nd Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2009, pp. 1828–1840.
- [22] J. Mi, Q. Wang, P. Liu, and X. Han, "A performance enhancement method for redundant IMU based on neural network and geometric constraint," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–11, 2024.
- [23] L. Wang, H. Tang, T. Zhang, Q. Chen, J. Shi, and X. Niu, "Improving the navigation performance of the MEMS IMU array by precise calibration," *IEEE Sensors J.*, vol. 21, no. 22, pp. 26050–26058, Nov. 2021.
- [24] I. Skog, J.-O. Nilsson, D. Zachariah, and P. Händel, "Fusing the information from two navigation systems using an upper bound on their maximum spatial separation," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sydney, NSW, Australia, Nov. 2012, pp. 1–5.

- [25] H. Liu, Q. Li, C. Li, and H. Zhao, "Application research of an array distributed IMU optimization processing method in personal positioning in large span blind environment," *IEEE Access*, vol. 8, pp. 48163–48176, 2020.
- [26] I. Skog, J.-O. Nilsson, and P. Händel, "Pedestrian tracking using an IMU array," in *Proc. IEEE Int. Conf. Electron., Comput. Commun. Technol. (CONECCT)*, Bengaluru, India, Jan. 2014, pp. 1–4.
- [27] I. Skog, J.-O. Nilsson, P. Händel, and A. Nehorai, "Inertial sensor arrays, maximum likelihood, and Cramér–Rao bound," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4218–4227, Aug. 2016.
- [28] T. Wang, K. Li, H. Luo, and S. Zhong, "Improving the measurement accuracy of the MEMS IMU array by a new calibration and fusion technology," *IEEE Sensors J.*, vol. 24, no. 8, pp. 13279–13292, Mar. 2024.
- [29] A. Wägli, "Trajectory determination and analysis in sports by satellite and inertial navigation," Ph.D. dissertation, Doctoral School Inf. Commun. Sci., EPFL, Lausanne, Switzerland, 2009.
- [30] E.-H. Shin, "Estimation techniques for low-cost inertial navigation," Ph.D. dissertation, Dept. Geomatics Eng., Univ. Calgary, Calgary, AB, Canada, 2005.
- [31] X. Niu, H. Zhang, C. Shi, K.-W. Chiang, and N. El-Sheimy, "A proposed evaluation standard for the navigation results of MEMS INS/GPS integrated systems," in *Proc. Int. Symp. GPS/GNSS*, 2010, pp. 1–5.



Man Yuan received the B.E. (Hons.) degree in communication engineering from Wuhan University, Wuhan, China, in 2023, where he is currently pursuing the master's degree in navigation, guidance, and control with the GNSS Research Center.

His primary research interests include GNSS/INS integrations and LiDAR-based navigation.



Liqiang Wang received the B.E. (Hons.) and M.E. degrees from Wuhan University, Wuhan, China, in 2020 and 2023, respectively, where he is currently pursuing the Ph.D. degree with the GNSS Research Center.

His primary research interests include GNSS/INS integrations and visual-based navigation.



Hailiang Tang received the M.E. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2020 and 2023, respectively.

He is currently a Postdoctoral Fellow with the GNSS Research Center, Wuhan University. His current research interests include visual and LiDAR SLAM, autonomous robotics systems, GNSS/INS integration technology, and deep learning.



Xiaoji Niu received the bachelor's and Ph.D. degrees from the Department of Precision Instruments, Tsinghua University, Beijing, China, in 1997 and 2002, respectively.

He was a Postdoctoral Researcher with the University of Calgary, Calgary, AB, Canada, and worked as a Senior Scientist with SIRF Technology Inc., Shanghai, China. He is currently a Professor with the GNSS Research Center, Wuhan University, Wuhan, China. He has authored or coauthored more than 90 academic

articles and own 28 patents. He leads a multisensor navigation group focusing on GNSS/INS integration, low-cost navigation sensor fusion, and its new applications.



Tisheng Zhang received the B.Sc. and Ph.D. degrees in communication and information systems from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

From 2018 to 2019, he was a Postdoctoral Researcher with The Hong Kong Polytechnic University, Hong Kong. He is an Associate Professor with the GNSS Researches Center, Wuhan University. His research interests include GNSS receiver and multisensor deep integration.