Research on a Distributed Calibration Method Based on Specific Force Measurement

Peng Zhang 1, Yuanting Xue 2, Peng Liu 2,3,* and Mengwei Li 1,2

1 School of Instrumentation and Electronic, North University of China, Taiyuan 030051, China
2 Academy for Advanced Interdisciplinary Research, North University of China, Taiyuan 030051, China
3 North Automatic Control Technology Institute, Taiyuan 030006, China
* Correspondence: pengliu@nuc.edu.cn

Abstract: Micro-electro-mechanical system (MEMS) inertial devices are small volume, lightweight, low cost, and have mass-production characteristics. The development trend of inertial modules is to reduce cost and improve accuracy, and batch calibration of MEMS devices is one of the most feasible solutions to reduce cost. In this paper, we propose a distributed calibration method based on system-level, discrete calibration. The distributed calibration method requires only one or a few rotations of the combined and arranged devices to excite the individual error parameters of the inertial instruments. In this study, the relationship between the error parameters and the navigation error was rewritten using equivalence transformation, and the 24 error parameters of the device were identified by the distributed least-squares estimation using the velocity error as the observed quantity. In simulation experiments, this method could calibrate more MEMS devices simultaneously than the traditional calibration method with the exact accuracy requirement.

Keywords: batch calibration; distributed; least-squares estimation; error parameter

1. Introduction

Inertial devices need to be calibrated before use. Calibration is an estimation of the system error, which is used in system modulation. The aim of calibration and compensation of error is to improve accuracy; this is a feasible solution to improve the accuracy of a device when using software.

Inertial sensor errors include accelerometer and gyroscope errors. The nature of the error can be divided into systematic or random errors, and the error source can be divided into zero-bias error, scale-factor error, nonlinear error, installation error, or random noise. For the calibration of random errors, we usually use the Allan variance method or direct variance statistics.

Here, we mainly studied the calibration of systematic errors. At this stage, there are two main calibration methods, namely, discrete calibration and system-level calibration. Discrete calibration uses specialized laboratory equipment to provide information regarding acceleration and angular velocity, and directly uses the output of the gyro and accelerometer as the observed quantity [1]. This method uses the centralized least-squares algorithm to obtain performance parameters of the component to be calibrated. This method is simple in principle and practical in engineering. Yet, the calibration period of this method is long, and the experimental calibration process is complicated. This method leads to a long calibration time and inaccurate error measurement for mass-produced micro-electro-mechanical system (MEMS) devices, which inevitably increases the calibration cost and reduces calibration efficiency.

System-level calibration means that all error parameters are considered as part of a navigation system. By observing the navigation error, the device error is inverted. Device parameter errors will inevitably lead to navigation system output errors, that is, position...
and velocity errors [2,3]. As the error propagation law in the navigation system is already in place, we can reverse the error equation to deduce the device error. System-level calibration can be summarized into two schemes: (1) Kalman filter-based system-level calibration [4,5]; (2) least-squares fitting-based system-level calibration [6,7]. The Kalman filter-based system-level calibration method requires establishing the error equation of the MEMS device and then compensating for the error. The prerequisite for reliable operation of the Kalman filter is the observability of the system, which requires observability or observability analysis of the error system. Therefore, this method is difficult for path programming and unsuitable for batch MEMS calibration. The system-level calibration method based on the least-squares fitting method must establish the error equation of the MEMS device and use the appropriate ratio measurement method [8]. The key to this fitting method is the design optimization of the rotation arrangement, which is suitable for calibrating medium-precision devices.

Unmanned motion platform such as crewless vehicles, guided munitions, and unmanned systems must use many inexpensive MEMS inertial devices, which require developing new batch calibration techniques. Considering the homogeneity characteristic of scale production and combining the advantages of discrete calibration and system-level calibration, a distributed least-squares method for batch calibration of multiple devices is proposed in this paper. The ideas of the kernel space orthogonal projection matrix and graph theory are utilized with dimensionality reduction and splitting of the measurement equations [9,10]. The high-dimensional measurement equations are reduced to multiple low-dimensional equations. The solutions of the low-dimensional equations communicate with each other in a certain way so that the final solutions converge to the same level. Each device requires only a few simple rotations to obtain the required measurement equations for calibration. This method belongs to the category of system-level calibration. Compared to other calibration methods at this stage, this method has a simple calibration path design, and all error parameters can be excited with only a few simple rotations according to the choreography rules. The calibration speed is increased using the distributed method, which decentralizes the process of solving a single device centrally into a common calculation for multiple sensors. This paper focuses on the calibration of mass-produced multi-sensors, and experiments show that the calibration accuracy of this method is close to that of the conventional method, and that multiple devices can be calibrated at once.

The rest of the text is as follows. Section 2 of the article starts with constructing the accelerometer and gyroscope error models and then establishes the relationship between device errors and navigation. Section 3 gives the arrangement of the sensors for distributed calibration. Section 4 explains precisely how the algorithm and calibration scheme of the paper were designed. The results of the simulation experiments are given in Section 5. The conclusions are given in Section 6.

2. Construction of Inertial-Navigation Standard Calibration Model

In this section, the error model and calibration error equation of inertial devices are given, which can serve as the basis for the subsequent transformation from centralized calibration to distributed calibration.

2.1. Mathematical Error Model of the Inertial Sensor

The error source of the sensor can be divided into two types: one is the error of the device, and the other is the systematic error generated during navigation. The inertial device’s error is mainly the error of the gyroscope and the accelerometer. A strapdown inertial navigation system mainly consists of three gyroscopes and three accelerometers. Among the errors are the zero offset of the accelerometer and gyroscope, the scale factor,
and the installation error. From this, we define the projection of the error model of the accelerometer and gyroscope under the body system as:

\[
\begin{bmatrix}
\hat{f}_x \\
\hat{f}_y \\
\hat{f}_z
\end{bmatrix} =
\begin{bmatrix}
K_{ax} & E_{axy} & E_{axz} \\
E_{axy} & K_{ay} & E_{ayz} \\
E_{axz} & E_{ayz} & K_{az}
\end{bmatrix}
\begin{bmatrix}
\hat{f}_x \\
\hat{f}_y \\
\hat{f}_z
\end{bmatrix}
+ 
\begin{bmatrix}
D_{bx} \\
D_{by} \\
D_{bz}
\end{bmatrix} 
\] (1)

\[
\begin{bmatrix}
\delta \omega_{bx} \\
\delta \omega_{by} \\
\delta \omega_{bz}
\end{bmatrix} =
\begin{bmatrix}
K_{gx} & E_{gxy} & E_{gxz} \\
E_{gxy} & K_{gy} & E_{gyz} \\
E_{gxz} & E_{gyz} & K_{gz}
\end{bmatrix}
\begin{bmatrix}
\omega_{bx} \\
\omega_{by} \\
\omega_{bz}
\end{bmatrix}
+ 
\begin{bmatrix}
\epsilon_{bx} \\
\epsilon_{by} \\
\epsilon_{bz}
\end{bmatrix} 
\] (2)

where \( K_{ax}, K_{ay}, K_{az} \) denotes the scale factor error of the accelerometer; \( E_{gx}, E_{gy}, E_{gz} \) denotes the scale factor error of the gyroscope. \( E_{axx}, E_{axy}, E_{axz}, E_{ayx}, E_{ayy}, E_{azx}, E_{azy}, E_{azz} \) denotes the installation error of the gyroscope; \( E_{gxx}, E_{gxy}, E_{gyx}, E_{gyy}, E_{gzz} \) denotes the installation error of the accelerometer; \( E_{bx}, E_{by}, E_{bz} \) denotes the installation error of the gyroscope. \( D_{bx}, D_{by}, D_{bz} \) denotes the zero offset of the accelerometer; \( \epsilon_{bx}, \epsilon_{by}, \epsilon_{bz} \) denotes the zero offset of the gyroscope.

2.2. Establishment of Calibration Error Equations

Distributed calibration methods use navigation results as observational measurements, then the inertial sensor error is calibrated. The calibration process adopts the motion arrangement method of “stationary-turning-stationary”, and simultaneous control of turntables with multiple MEMS sensors. The initial alignment of inertial devices is performed first; then, the turntable is controlled to rotate according to the preset position sequence. The data are measured and recorded as the observed quantity after resting.

When defining the east-north-up coordinate system as the navigation coordinate system, the output equation of the strapdown inertial navigation system can be expressed as:

\[
V_{et}^f = T_{b}^i f^b_i - (2\omega_{et}^i + \omega_{et}^i) \times V_{et}^i + g^i 
\] (3)

In the actual calculations, there are effects such as measurement errors and systematic errors. This results in the calculation and the actual calculation being different. Therefore, the following arises:

\[
V_{et}^i = T_{b}^i f^b_i - (2\omega_{et}^i + \omega_{et}^i) \times V_{et}^i + g^i 
\] (4)

where \( T_{b}^i \) denotes the attitude matrix after alignment. \( T_{b}^i = T_{b}^i T_{b}^i = (I - \phi \times)T_{b}^i \), \( \phi \times \) represents the skew-symmetric matrix of \( \phi \), \( f^b_i \) denotes the specific force measurement, and \( \omega_{et}^i \) denotes the angular velocity of the rotation of the Earth in the Earth’s coordinate system. \( \omega_{et}^i \) denotes the angular velocity of rotation of the geosystem relative to the Earth’s coordinate system, \( V_{et}^i \) denotes the velocity of motion relative to the Earth’s coordinate system, and \( g^i \) denotes the acceleration of gravity.

Accordingly, \( \delta V_{et}^i = V_{et}^i - V_{et}^i, \delta \omega_{et}^i = \omega_{et}^i - \omega_{et}^i, \delta \omega_{et}^i = \omega_{et}^i - \omega_{et}^i \). At \( V_{et}^i = 0 \), neglecting the convective acceleration, \( 2\omega_{et}^i + \omega_{et}^i \) yields the approximate error equation as:

\[
\hat{V}_{et}^i \approx \delta T_{b}^i f^b_i + T_{b}^i \delta f^b
\] (5)

The specific force measurement of the position of the sensor before rotation is 1. The specific force measurement at the position after rotation is 2. From this, it is possible to write out specific force measurement equations for the two positions, namely:

\[
(\delta V_{et}^i)_1 = (T_{b}^i \cdot f^b_i)_1 - \phi_1 \times (T_{b}^i \cdot f^b_i)_1 + (T_{b}^i \cdot \delta f^b)_1 
\] (6)

\[
(\delta V_{et}^i)_2 = (T_{b}^i \cdot f^b_i)_2 - \phi_2 \times (T_{b}^i \cdot f^b_i)_2 + (T_{b}^i \cdot \delta f^b)_2 
\] (7)
where \( \phi_1 \) denotes the initial error alignment angle, and \( \phi_2 \) denotes the navigation error angle after rotation.

The observed quantity defined using the system-level calibration method is obtained by re-differentiating the velocity error and decomposing it into the X, Y, and Z axes. Therefore, according to the literature [11], specific force measurements can be written as vertical and horizontal components. In this case, the available vertical components are:

\[
(\delta V^{v}_{et})_{1U} = \mathbf{g} + [(T^f_b \cdot \delta f^f_b)_1]_u \tag{8}
\]

The horizontal component is:

\[
(\delta V^{h}_{et})_{1H} = [\phi_1 \times \mathbf{g} + (T^f_b \cdot \delta f^f_b)_1]_H \tag{9}
\]

The specific force of each component before and after rotation is measured and integrated as:

\[
(\delta V^{v}_{et})_{2U} - (\delta V^{v}_{et})_{1U} = [(T^f_b \cdot \delta f^f_b)_2]_u - [(T^f_b \cdot \delta f^f_b)_1]_u \tag{10}
\]

\[
(\delta V^{h}_{et})_{2H} - (\delta V^{h}_{et})_{1H} = [(\phi_2 - \phi_1) \times \mathbf{g}]_H + [(T^f_b \cdot \delta f^f_b)_2]_u - [(T^f_b \cdot \delta f^f_b)_1]_u \tag{11}
\]

At this point, the error equation for distributed scaling is established. Below, the calibration path orchestration and calibration process are designed.

### 3. Calibration Path Orchestration

The traditional calibration method is time-consuming; this article uses a batch-processing approach to shorten the duration. Compared with the traditional method, this method differs in the scheme design. The batch-processing method groups the same batch of sensors to be calibrated and then loads the sensors onto four identical three-axis turntables. The turntables are connected to the central computer with a bus for unified control; each sensor rotates five or six times in a choreographed pattern, with four turntables processed in parallel, and the data before and after the rotation are recorded and analyzed to obtain specific force observations.

Combined with the idea of a batch calibration algorithm, the Savage orchestration method (as shown in Table 1) is recombinantly designed.

| Sensor Ordinal | Rotation | State before Rotation |
|----------------|----------|-----------------------|
|                | Degree of Rotation | Rotary Axis | X-Axis | Y-Axis | Z-Axis |
| 1              | +360      | Y                     | North  | East   | Down   |
| 2              | +360      | X                     | East   | South  | Down   |
| 3              | +90 +360 -90 | Y Z Y                | North  | East   | Down   |
| 4              | +180 +90 +180 -90 | Y Z X Z            | North  | East   | Down   |
| 5              | +180 +90 +180 -90 | X Z Y Z             | East   | South  | Down   |
| 6              | +90 +90 -90 -90  | Y Z X Z             | North  | East   | Down   |
| 7              | +180      | Y                     | North  | East   | Down   |
| 8              | +180      | Y                     | South  | East   | Up     |
| 9              | +90       | X                     | East   | South  | Down   |
| 10             | +90       | X                     | East   | Down   | North  |
| 11             | +180      | Z                     | Down   | East   | South  |
| 12             | +180      | X                     | East   | South  | Down   |
| 13             | +180      | X                     | East   | North  | Up     |
| 14             | +90       | Y                     | North  | East   | Down   |
| 15             | +90       | Y                     | Up     | East   | North  |
| 16             | +180      | Z                     | East   | Down   | North  |
| 17             | +180      | X                     | East   | Up     | South  |
| 18             | +180      | X                     | East   | Down   | North  |
| 19             | +90       | Z                     | East   | Up     | South  |
| 20             | +90       | Z                     | Up     | West   | South  |
| 21             | +180      | X                     | East   | South  | Down   |
According to the above arrangement method, the arrangement scheme is designed as follows: four sensors calibrated as a group and installed in four turntables, respectively. The turntable is connected by the bus to the central computer using a parallel control mode. The first turntable rotates according to the arrangement from ordinal numbers 1 to 5 in Table 1. The second turntable rotates according to the arrangement from ordinal numbers 6 to 10 in Table 1. The third turntable rotates according to the arrangement from ordinal numbers 11 to 15 in Table 1. Finally, the fourth turntable rotates according to the arrangement from ordinal numbers 16 to 21 in Table 1.

4. Calibration Scheme Design Based on the Distributed Algorithm

The purpose of the distributed calibration algorithm is to establish a mathematical model between the static navigation speed error and inertial device error parameters. In combination with this model, all error parameters are estimated using the distributed algorithm.

For traditional calibration, the sensor needs to be rotated in all programmed positions to obtain the observation equation. This method is computationally complex due to the large observation of equations; therefore, a distributed approach is proposed. All turntables containing sensors yield four sets of observations after rotation according to Table 1. The four sets of observations are calculated independently of each other, thus greatly reducing computational complexity. The distributed algorithm uses this observation as a node in the sensor network, with each node existing independently and holding local information. The final estimate is obtained using a distributed algorithm that combines the local information of each node, which is the estimated value of the distributed calibration.

The method uses a central computer to control the turntables, each of which is rotated five or six times. The local measurement of each node is obtained. This measurement is the ratio observation. There are a total of four sensors to be calibrated here, and after the above method of orchestration and rotation has been applied, 63 sets of equations can be obtained. The parameters to be calibrated are used as state quantities \( X = [X_a, X_g]^T \); based on these 63 sets of data, the measurement equation is obtained as:

\[
Z = HX + V
\]  

(12)

where \( V \) is the measurement noise with mean 0 and covariance \( R \), \( H \) is the measurement matrix, and the measurement equation is expanded as:

\[
Z = \begin{bmatrix} a_1 & b_1 & c_1 & a_2 & b_2 & c_2 & \cdots & a_{20} & b_{20} & c_{20} \end{bmatrix}^T
\]  

(13)

\[
X = [X_a, X_g]^T
\]  

(14)

where:

\[
X_a = [K_{ax}, E_{axy}, E_{axz}, K_{ay}, E_{ayz}, E_{axy}, E_{axy}, K_{az}, D_{bx}, D_{by}, D_{bz}]^T
\]

\[
X_g = [K_{gx}, E_{gxy}, E_{gxz}, K_{gy}, E_{gyz}, E_{gxy}, E_{gzy}, E_{gxy}, K_{gz}, \varepsilon_{bx}, \varepsilon_{by}, \varepsilon_{bz}]^T
\]

After obtaining the measurement equation, the error parameters are estimated using the distributed least-squares method. As obtained from (12):

\[
Ax = b
\]  

(15)

The sensor is rotated to obtain a set of corresponding observations \( b_i \) and the measurement matrix \( A_i \). Combining the above arrangement scheme, four corresponding sets of equations can be obtained \([12,13]\):
The distributed algorithm synthesizes the solutions of multiple nodes to obtain the optimal solution for the measurement equation [14,15]. Mou and colleagues first proposed the distributed algorithm to resolve Equation (14) in references [16–19]. We changed the computational process and applied it to the sensor calibration. This can perform distributed computation on a network composed of multiple sensors. Each node in the sensor network only needs to master some of the elements in \([A \ b]\).

This method needs to satisfy two conditions. The first is that the estimated value of each node \(i\) has to satisfy the set of equations consisting of the measurement matrix \(A_i\) and the observations \(b_i\) for the sensor. The second is that the estimates obtained using all nodes must finally converge to the exact solution of \(Ax = b\), satisfying the consistency principle.

Condition one requires the selection of a point as the starting point of the algorithm. This point is one of the solutions of the measurement matrix. That is, the point satisfies \(A_i x_i(1) = b_i\). As can be seen from [20], it is updated iteratively as follows:

\[
x_i(k + 1) = x_i(k) + K_i u_i(k)
\]  

(17)

where \(K_i\) is the base matrix on the kernel space of the measurement matrix \(A_i\), so as long as \(x_i(1)\) satisfies \(A_i x_i = b_i\), then (24) always satisfies \(A_i x_i(k + 1) = b_i\).

According to the consistency of the distributed algorithm, condition two needs to satisfy the following expression:

\[
x_i(k + 1) = \frac{1}{m_i(k)} \sum_{j \in N_i(k)} x_j(k)
\]  

(18)

But conditions one and two cannot be met at the same time because:

\[
-x_i(k) + \frac{1}{m_i(k)} \sum_{j \in N_i(k)} x_j(k) \notin \text{image}(K_i)
\]  

(19)

We use the least-squares approach to solve the problem:

\[
\min \frac{1}{2} \left| x_i(k) + K_i u_i(k) - \frac{1}{m_i(k)} \sum_{j \in N_i(k)} x_j(k) \right|^2
\]  

(20)

The network can be represented as:

\[
x_i(k + 1) = x_i(k) - \frac{1}{m_i(k)} P_i(m_i(k) x_i(k) - \sum_{j \in N_i(k)} x_j(k))
\]  

(21)

where \(m_i\) represents the sum of all sensors in the same batch as sensor \(i\), \(P_i\) represents the measurement matrix \(A_i\) quadrature projection matrix on the core space, and \(N_i\) represents the set of sensors \(i\) in the same batch of sensors to be calibrated.

The calibration using the distributed algorithm requires the measurement of the ratio measurements before and after rotation of the inertial guide. This value was projected onto the navigation system as the observation required for calibration. Let each set of observations be \(b_i\), the measurement matrix \(A_i\) corresponding to the observations be obtainable by calculation, and the solution of \(A_i x_i = b_i\) be obtainable according to (15). Here, a specific initial point \(q\) is selected and the closest point to \(q\) in the solution of \(A_i x_i = b_i\) is found for the iteration. This method can speed up the iteration process.
Inspired by reference [21], \( x_{\text{min}}^q = \arg\min_{Ax=b} \frac{1}{2} |x - q|^2 \) is obtained, where \( |\cdot| \) denotes the Euclidean norm. The text makes a simple change to \( x_{\text{min}}^q = \arg\min_{Ax=b} \frac{1}{2} |x - q|^2 \) to achieve a specific solution \( x_{\text{min}}^q \) with respect to the local minimum of \( \frac{1}{2} |x - q|^2 \) depending on \( A_i x = b_i \). That is:

\[
x_i(1) = \arg\min_{A_i x=b_i} \frac{1}{2} |x - q|^2
\]

(22)

When \( Ax = b \) has multiple solutions, it is not yet clear which one is to be achieved. Equation (22) is the first step for the initialization.

When \( q \) is 0, the unique solution is the minimum norm solution of the measurement equation.

The main steps are as follows:

1. Set the initial point \( x_i(1) \), which is required to meet \( \arg\min_{A_i x=b_i} \frac{1}{2} |x - q|^2 \), where \( i \in \{1, 2, 3, \ldots, m\} \);
2. Set the error value \( \sigma > 0 \), meaning when \( \sum_{i=1}^{m} |Ax_i(k) - b|^2 < \sigma \), the iteration stops;
3. If \( \sum_{i=1}^{m} |Ax_i(k) - b|^2 > \sigma \), the estimate is updated: \( x_i(k+1) = x_i(k) - \frac{1}{|P_i(k)|} P_i(m_i(k)x_i(k) - \sum_{j \in N_i(k)} x_j(k)) \);
4. Assign the value of \( x_i(k+1) \) to \( x_i(k) \) and continue to iterate until the algorithm terminates.

Turntable one is used as an example for combining the alignment scheme. The specific force measurements before and after the rotation of the inertial guidance system are projected onto the navigation coordinate system to obtain the specific force observations. Each set of observations and the measurement equation form a point in the sensor network, and the 63 sets of observations obtained at the end of all rotations together form the sensor network. Each point in the network combines its observations and the measurement matrix to obtain a feasible solution as the starting point of the distributed calibration algorithm. Each point in the network is updated using Equation (3) to obtain the estimated value iteratively.

5. Analysis of Experimental Results

The random drift of the strapdown inertial navigation system gyroscope in the same batch are 1, 2, and 3 degrees/h, and the zero-skew stability of the accelerometer are 1, 2, and 3 mg, respectively. The installation errors of the gyroscope and accelerometer are 110", 110", 150", 150", 190", and 190". The scale factor error of the accelerometer is set to \( 1 \times 10^{-3} \text{diag}(1 \ 1 \ 1) \), and the scale factor error of the gyroscope is set to \( 1 \times 10^{-2} \text{diag}(1 \ 1 \ 1) \). The rotation speed of the turntable is set at 10 degrees/s, and the rotation is carried out according to the set position to stimulate each error term. The accelerometer and gyroscope data are collected and estimated with the distributed least-squares algorithm. To verify the consistency of the distributed algorithm, several simulation tests were conducted to obtain a total of 63 sets of estimated values. Data from the first, 30th, and 60th sets were selected for consistency analysis. Each node in the sensor network tended to be consistent at the end of the iteration. Comparison charts are shown in Figures 1–3. Comparison between the distributed and traditional calibrations is shown in Figures 4–6. The distributed calibration approximates the traditional calibration via lower-dimensional computation. The calibration experiment results are shown in Table 2.
5. Analysis of Experimental Results

The random drift of the strapdown inertial navigation system gyroscope in the same batch are 1, 2, and 3 degrees/h, and the zero-skew stability of the accelerometer are 1, 2, and 3 mg, respectively. The installation errors of the gyroscope and accelerometer are 110″, 110″, 150″, 150″, 190″, and 190″. The scale factor error of the accelerometer is set to $(1 - 10 \times 10^{-3})$, and the scale factor error of the gyroscope is set to $(2 - 10 \times 10^{-3})$. The rotation speed of the turntable is set at 10 degrees/s, and the rotation is carried out according to the set position to stimulate each error term. The accelerometer and gyroscope data are collected and estimated with the distributed least-square algorithm.

To verify the consistency of the distributed algorithm, several simulation tests were conducted to obtain a total of 63 sets of estimated values. Data from the first, 30th, and 60th sets were selected for consistency analysis. Each node in the sensor network tended to be consistent at the end of the iteration. Comparison charts are shown in Figure 1–3. Comparison between the distributed and traditional calibration is shown in Figures 4–6. The distributed calibration approximates the traditional calibration via lower-dimensional computation. The calibration experiment results are shown in Table 2.

Figure 1. Gyroscope installation−error estimation results.

Figure 2. Gyroscope meter−scal−factor estimation results.

Figure 3. Gyroscope zero−bias estimation results.

The second column in Table 2 estimates the installation error, scale factor error, and zero-bias error of the accelerometer from the conventional system-level calibration. The third column estimates each error value of the accelerometer from the distributed method. The fifth column estimates the installation error, scale factor error, and zero-bias error of the gyroscope using the conventional method. The sixth column estimates the error values of the gyroscope using the distributed method. The zero-bias errors of the gyroscope and accelerometer are analyzed. We give the random drift of the gyroscope as 1, 2, and 3 degrees/h, and the zero-bias stability of the accelerometer as 1, 2, and 3 mg, respectively. The zero-bias error parameters of the accelerometer obtained using the traditional method are 0.9993, 1.9997, and 3.0007, while the parameters obtained using the distributed algorithm are 0.9965, 1.9904, and 3.0132. The zero-bias errors of the gyroscope obtained using the conventional method are 0.9967, 2.0007, and 2.9994. The parameters obtained using the distributed algorithm are 1.0668, 2.1194, and 3.1909.
Figure 2. Gyroscope meter−scal−factor estimation results.

Figure 3. Gyroscope zero−bias estimation results.

Figure 4. Accelerometer installation−error estimation results.

Figure 5. Accelerometer scale−factor−error estimation results.

Figure 3. Gyroscope zero−bias estimation results.

Figure 4. Accelerometer installation−error estimation results.
Figure 4. Accelerometer installation−error estimation results.

Figure 5. Accelerometer scale−factor−error estimation results.

Figure 6. Accelerometer zero−bias−error estimation results.

According to the set error parameters, the centralized estimation results were compared with the distributed ones in combination with Table 2. The results showed that the distributed algorithm was more accurate than the centralized one in estimating some parameters, for example, \( E_{axz} \), \( E_{azy} \), \( E_{gyz} \), etc. However, the rest of the estimation results were equal or close to the centralized one. The reason for this is that the simple Gaussian model cannot solve complex optimization problems well.
Table 2. Comparison of traditional and distributed calibration results.

| Parameter | Traditional Calibration Method | Distributed Calibration | Parameter | Traditional Calibration Method | Distributed Calibration |
|-----------|--------------------------------|-------------------------|-----------|--------------------------------|-------------------------|
| $E_{axy}$ | 0.0305                         | 0.0326                  | $E_{gxy}$ | 0.0337                         | 0.0334                  |
| $E_{axz}$ | 0.0445                         | 0.0321                  | $E_{gxz}$ | 0.0472                         | 0.0272                  |
| $E_{ayx}$ | 0.0416                         | 0.0377                  | $E_{gyx}$ | 0.0444                         | 0.0246                  |
| $E_{azy}$ | 0.0529                         | 0.0470                  | $E_{gyz}$ | 0.0558                         | 0.0416                  |
| $E_{azx}$ | 0.0444                         | 0.0576                  | $E_{gzx}$ | 0.0472                         | 0.0373                  |
| $K_{ax}$  | $-6.20 \times 10^{-5}$         | $5.97 \times 10^{-4}$  | $K_{gx}$  | $-2.08 \times 10^{-4}$         | $-4.3 \times 10^{-3}$  |
| $K_{ay}$  | $-1.28 \times 10^{-5}$         | $7.11 \times 10^{-4}$  | $K_{gy}$  | $-1.31 \times 10^{-4}$         | $-1.1 \times 10^{-3}$  |
| $K_{az}$  | $7.48 \times 10^{-5}$          | $2.91 \times 10^{-4}$  | $K_{gz}$  | $3.86 \times 10^{-4}$          | $1.8 \times 10^{-3}$   |
| $D_{bx}$  | 0.9993                         | 0.9965                  | $\epsilon_{bx}$ | 0.9967                         | 1.0668                  |
| $D_{by}$  | 1.9997                         | 1.9904                  | $\epsilon_{by}$ | 2.0007                         | 2.1194                  |
| $D_{bz}$  | 3.0007                         | 3.0132                  | $\epsilon_{bz}$ | 2.9994                         | 3.1909                  |

As far as calibration results are concerned, the distributed algorithm achieves a high accuracy for the calibration of sensors. For the same batch of sensors to be calibrated, the distributed algorithm can calibrate several sensors at the same time. Therefore, distributed calibration is a good choice for medium- and high-accuracy sensors.

6. Conclusions

The traditional calibration method of inertial devices is only suitable for calibrating a single device, and the number of calibrations per unit of time is limited for mass-produced inertial sensors. This paper proposed a batch calibration method to calibrate multiple inertial sensors simultaneously to address the above issues. Firstly, multiple sensors were arranged and combined using position arrangement. Then, the output in the navigation state was obtained using five or six rotations as the observed quantity, and the measurement equation related to calibrating error was established. Finally, the distributed algorithm was used to estimate the equation with error, and the estimated values of multiple inertial sensors were obtained. The experiments showed that the distributed calibration method can calibrate several sensors simultaneously. In terms of results, the calibration results for multiple sensors were similar. The calibration accuracy of this calibration method was as good as the centralized one. It can be used to calibrate MEMS inertial devices such as smartphones and crewless vehicles, which do not require high sensor accuracy and have low cost.

Author Contributions: Conceptualization, P.Z. and Y.X.; methodology, P.L.; software, Y.X.; validation, P.Z. and Y.X.; formal analysis, Y.X.; writing—original draft preparation, Y.X.; and writing—review and editing, P.Z., P.L. and M.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Defence Fund (2021-JJ-0726) and test instrument research projects (2005CS037).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank all of the authors cited and the anonymous reviewers in this article for their helpful suggestions.

Conflicts of Interest: The authors declare no conflict of interest.
References

1. Wang, Z.; Qin, F.; Zha, F.; Tong, L.; Huang, C. A fast and non-directional calibration method for fiber optic gyro inertial navigation system. In Proceedings of the 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), Xi’an, China, 21–22 September 2019; pp. 137–142.

2. Yang, X.; Huang, Y. A system-level calibration method for laser jet-linked inertial navigation system. *Chin. J. Inert. Technol.* 2008, 1, 1–7.

3. Jia, J.; Qin, Y.; Zhang, B.; Wu, X. A new method for rapid external field calibration of laser gyro-jet-linked inertial navigation system. *Chin. J. Inert. Technol.* 2014, 22, 23–25.

4. Wang, Z.; Xie, Y.; Yu, X.; Fan, H.; Wei, G.; Wang, L.; Luo, H. A system-level calibration method including temperature-related error coefficients for a strapdown inertial navigation system. *Meas. Sci. Technol.* 2021, 32, 115117. [CrossRef]

5. Zhang, J.; Li, W.; Zhang, J.; Nie, P.; Zhang, C. Research on MIMU Attitude Solving Algorithm Based on Kalman Filter. *Comput. Meas. Control* 2020, 28, 28233–28237, 28242.

6. Zhao, G.; Tan, M.; Guo, Q.; Wu, C. An Improved System-Level Calibration Method of Strapdown Inertial Navigation System Based on Matrix Factorization. *IEEE Sens. J.* 2022, 22, 14986–14996. [CrossRef]

7. Lu, L.; Wang, L.; Wang, W.; Ban, J. A self-calibration method for error of photoelectric encoder based on gyro in rotational inertial navigation system. *Microsyst. Technol.* 2019, 25, 2145–2152. [CrossRef]

8. Ghaderpour, E.; Spiros, D.P.; Quazi, K.H. A survey on change detection and time series analysis with applications. *Appl. Sci.* 2021, 11, 6141. [CrossRef]

9. Zhang, Z.; Cai, Q.; Fu, M. A Fast Distributed Solver for Linear Systems Under Generalized Diagonal Dominance. *IEEE Trans. Autom. Control* 2020, 66, 2423–2429. [CrossRef]

10. Shi, G.; Anderson, B.D.O.; Helmke, U. Network flows that solve linear equations. *IEEE Trans. Autom. Control* 2016, 62, 2659–2674. [CrossRef]

11. Jiang, H. Multi-Position System-Level Calibration Method of Optical Fiber Gyro-Jet-Linked Inertial Conduction; Harbin Engineering University: Harbin, China, 2019.

12. Cihan, O. Rapid solution of linear equations with distributed algorithms over networks. *IFAC-PapersOnLine* 2019, 52, 467–471. [CrossRef]

13. Alaviani, S.S.; Elia, N. A distributed algorithm for solving linear algebraic equations over random networks. *IEEE Trans. Autom. Control* 2020, 66, 2399–2406. [CrossRef]

14. Yang, T.; George, J.; Qin, J.; Yi, X.; Wu, J. Distributed least-squares solver for network linear equations. *Automatic* 2020, 113, 108798. [CrossRef]

15. Lei, J.; Yi, P.; Shi, G.; Anderson, B.D.O. Distributed algorithms with finite data rates that solve linear equations. *SIAM J. Optim.* 2020, 30, 1191–1222. [CrossRef]

16. Wang, P.; Gao, Y.; Yu, N.; Ren, W.; Lian, J.; Wu, D. Communication-efficient distributed solutions to a system of linear equations with Laplacian sparse structure. In Proceedings of the 2018 IEEE Conference on Decision and Control (CDC), Miami, FL, USA, 17–19 December 2018; pp. 3367–3372.

17. Wang, P.; Mou, S.; Lian, J.; Ren, W. Solving a system of linear equations: From centralized to distributed algorithms. *Annu. Rev. Control* 2019, 47, 306–322. [CrossRef]

18. Liu, J.; Mou, S.; Morse, A.S. Asynchronous distributed algorithms for solving linear algebraic equations. *IEEE Trans. Autom. Control* 2017, 63, 372–385. [CrossRef]

19. Mou, S.; Liu, J.; Morse, A.S. A distributed algorithm for solving a linear algebraic equation. *IEEE Trans. Autom. Control* 2015, 60, 2863–2878. [CrossRef]

20. Marelli, D.L.; Fu, M. Distributed weighted least-squares estimation with fast convergence for large-scale systems. *Automatica* 2015, 51, 27–39. [CrossRef] [PubMed]

21. Wang, X.; Mou, S.; Sun, D. Improvement of a distributed algorithm for solving linear equations. *IEEE Trans. Ind. Electron.* 2016, 64, 3113–3117. [CrossRef]