Journal of Systems Engineering and Electronics
Vol. 35, No. 3, June 2024, pp.769 – 779

Kinematic calibration under the expectation maximization framework for exoskeletal inertial motion capture system

QIN Weiwei, GUO Wenxin*, HU Chen, LIU Gang, and SONG Tainian

School of Nuclear Engineering, Rocket Force University of Engineering, Xi'an 710025, China

Abstract: This study presents a kinematic calibration method for exoskeletal inertial motion capture (EI-MoCap) system with considering the random colored noise such as gyroscopic drift. In this method, the geometric parameters are calibrated by the traditional calibration method at first. Then, in order to calibrate the parameters affected by the random colored noise, the expectation maximization (EM) algorithm is introduced. Through the use of geometric parameters calibrated by the traditional calibration method, the iterations under the EM framework are decreased and the efficiency of the proposed method on embedded system is improved. The performance of the proposed kinematic calibration method is compared to the traditional calibration method. Furthermore, the feasibility of the proposed method is verified on the EI-MoCap system. The simulation and experiment demonstrate that the motion capture precision is significantly improved by 16.79% and 7.16% respectively in comparison to the traditional calibration method.

Keywords: human motion capture, kinematic calibration, exoskeleton, gyroscopic drift, expectation maximization (EM).

DOI: 10.23919/JSEE.2024.000050

1. Introduction

Motion capture (MoCap) technology is widely used in many applications including, but not limited to, telemedicine [1,2], virtual reality [3], and physical training [4,5]. It is a technology which mainly uses camera [6–8] (e.g., infrared, or optical camera), mechanical device [9], and inertial sensor [10,11] to obtain the position and orientation of the whole human body or its parts (e.g., an arm). In the above methods, the inertial MoCap (I-MoCap) technology is applied widely for its advantages such as portability, low cost, and insensitivity toward ambient light conditions [12–14]. However, the drawbacks of the I-MoCap [15] (e.g., gyroscopic drift and not easy to place correctly) greatly promote the development of the exoskeletal I-MoCap (EI-MoCap) technology.

In order to improve the MoCap precision, kinematic calibration is necessary [16]. However, during the process of data acquisition, the measurement noise is affected by many reasons such as the gyroscopic drift [17]. Hence, the traditional calibration method such as the least square (LS) method, which is extensively used in industrial arm, will be a biased estimate [18]. Therefore, kinematic parameters for EI-MoCap system pose a considerable challenge to calibrate as it introduces the random colored noise.

In order to meet this challenge, the maximum likelihood (ML) method is adopted. However, kinematic calibration by the ML method is difficult as it introduces a hybrid calibration problem since the uncalibrated parameters' influences are independently. For solving this problem, the expectation maximization (EM) algorithm was used by Jin et al. to calibrate the kinematic parameters which improves the calibration precision [19]. Based on the research of [19], Zhao et al. applied the variational Bayesian (VB) algorithm to estimate the model parameters [20]. Compared with the EM algorithm, the advantage of the VB algorithm is that the uncertainty can be decreased by providing the full probabilities of parameters. Sun et al. presented a calibration method based on the ML method for inertial navigation system. Compared with the LS method, the parameters calibrated by the ML method are more accurate and stable [21]. Moreover, Zhang et al. adapted the EM algorithm to the extended Kalman filter (EKF) to multiple global navigation satellite system (GNSS) precise point positioning (PPP) and they also compared the EM algorithm with other methods such as the LS method to demonstrate their advantages [22]. As discussed above, the EM algorithm

Manuscript received December 14, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (61503392).

could effectively find the ML estimates of parameters in an iterative way when solving the hybrid calibration problem.

In this study, the ML method is adopted to calibrate the kinematic parameters with considering the random colored noise such as gyroscopic drift. For solving the hybrid calibration problem introduced by the ML problem, the EM algorithm is used. The initial value of the EM algorithm is calculated by the traditional calibration method. By iterating the hidden variable and the posteriori distribution, and maximizing the objective estimation function, the ML problem can be solved. The key contributions of this paper can be summarized as follows:

(i) Unlike the LS method used in [18] to calibrate the kinematic parameters, this study proposes an ML method that considers the random colored noise produced by the gyroscopic drift.

(ii) For solving the hybrid calibration problem introduced by the ML problem, the EM algorithm is used to maximize the objective estimation function.

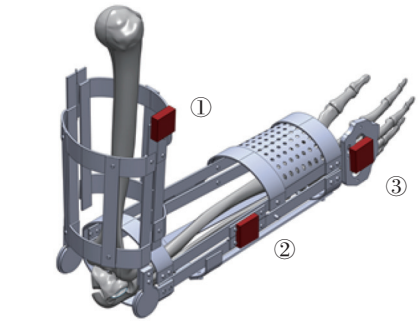
(iii) The effects of the EM algorithm are verified by comparing with the traditional calibration method in the hypothetical object simulation and semi-physical experiment.

The remainder of this paper is organized as follows. Section 2 describes the system model and gives the problem formulation. The geometric error calibration model and the formulation under the EM framework are presented in Section 3. Section 4 demonstrates the effects of the EM algorithm by simulation. The feasibility of the proposed method is verified on the embedded system we design in Section 5. Finally, Section 6 presents the conclusions of this paper.

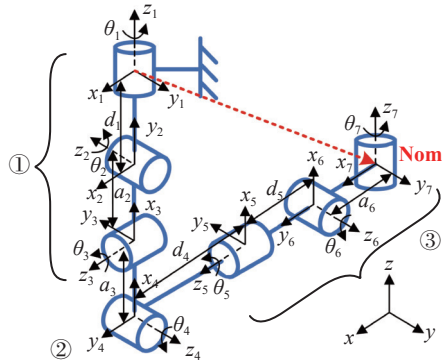
2. Problem formulation

2.1 EI-MoCap system

In order to achieve the goal of improving MoCap precision, some EI-MoCap systems have been developed. The design principles of the EI-MoCap system are portable, easy to wear and high precision. Hence, the structure of the system is mainly based on the configuration of the human upper limb. In this paper, an EI-MoCap system is designed, which is defined as a 7 degrees of freedom (7-DoFs) serial model by referring to a 7-DoFs human upper limb kinematic model [23], as shown in Fig. 1(a). The similar structures can be seen in [24] and [25]. Then, the link frames of the EI-MoCap system are established and their assignments are displayed in Fig. 1(b).



(a) System configuration



(b) System schematic

Fig. 1 3D model of 7-DoFs EI-MoCap system and its coordinate system

The kinematic parameters in Fig. 1(b) are notated as follows [26,27]. a_i represents the distance between the z_i axis and z_{i+1} axis, which is measured along the x_{i+1} axis. θ_i denotes the rotation about the z_i axis, which is needed to rotate the x_i axis to the x_{i+1} axis. d_i denotes the distance between the x_i axis and the x_{i+1} axis, which is measured along the z_i axis. α_i represents the rotation about the x_i axis, which is needed to rotate the z_i axis to the z_{i+1} axis. i denotes the label of joint coordinate system. The kinematic parameters of EI-MoCap system are defined based on the Denavit-Hartenberg (D-H) convention as shown in Table 1.

Table 1 Standard D-H parameters of EI-MoCap system

Rod	a_i/m	α_i/(°)	d_i/m	θ_i/(°)	Offset/(°)
1	0.04	270	0	$\theta_1(k)$	0
2	0.04	90	0	$\theta_2(k)$	90
3	0.431	90	0	$\theta_3(k)$	0
4	0.244	−90	0	$\theta_4(k)$	0
5	0	90	0	$\theta_5(k)$	0
6	0	90	0	$\theta_6(k)$	90
7	0	90	0	$\theta_7(k)$	0

According to the D-H convention, the homogeneous transformation from the $(i-1)$th joint to the ith joint could be calculated by a rotation around x-axis (α_i), fol-

lowed by two translations along the x- and z-axes (a_i, d_i) and a rotation around the z-axis (θ_i). The specific content is given by

$$\boldsymbol{T}_{i-1}^{i} = \mathbf{Rot}_z(\theta_i)\mathbf{Tran}_z(d_i)\mathbf{Tran}_x(a_i)\mathbf{Rot}_x(\alpha_i) = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

Then, based on the direct kinematics equation, the homogeneous transformation of the end-effector with respect to the base coordinate system is given by

$$\boldsymbol{T}_0^7 = \boldsymbol{T}_0^1\boldsymbol{T}_1^2\boldsymbol{T}_2^3\cdots\boldsymbol{T}_6^7 = \begin{bmatrix} Q_1 & Q_2 & \sin\beta_y\sin\beta_x & p_x \\ Q_3 & Q_4 & -\sin\beta_y\cos\beta_x & p_y \\ \sin\beta_y\sin\beta_z & \sin\beta_y\cos\beta_z & \cos\beta_y & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

where

$$Q_1 = \cos\beta_z\cos\beta_x - \cos\beta_y\sin\beta_x\sin\beta_z,$$

$$Q_2 = -\sin\beta_z\cos\beta_x - \cos\beta_y\sin\beta_x\cos\beta_z,$$

$$Q_3 = \cos\beta_z\sin\beta_x + \cos\beta_y\cos\beta_x\sin\beta_z,$$

$$Q_4 = -\sin\beta_z\sin\beta_x + \cos\beta_y\sin\beta_x\cos\beta_z,$$

and $[p_x, p_y, p_z, \beta_x, \beta_y, \beta_z]^{\mathrm{T}}$ denotes the position and orientation of the end-effector in the Cartesian space.

The calibration experiment is set up based on the designed EI-MoCap system. The actual position and orientation are obtained, which is called the lever effect, by successively measuring palstance and acceleration of inertial measurement unit (IMU) mounted on the end-effector. Then, the kinematic calibration method under the EM framework for EI-MoCap system is proposed and the corresponding flow chart is shown in Fig. 2. The idea is to search for geometric errors $\Delta\boldsymbol{\zeta}$ and random colored noise $\Delta\boldsymbol{\theta}$ that make the **Act** and **Nom** as close as possible, ideally the same value.

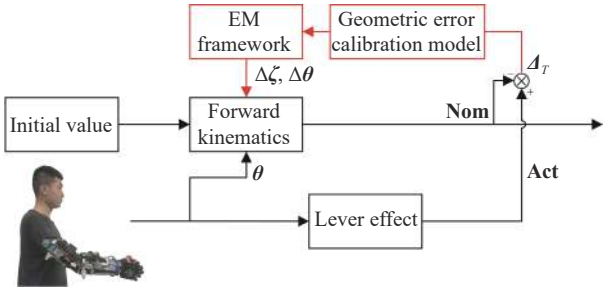


Fig. 2 Flow chart of the kinematic calibration method

2.2 Forward and differential kinematics

Considering the 7-DoFs EI-MoCap system, the position and orientation of the end-effector can be analytically described as a nonlinear mapping:

$$\mathbf{Nom} = f(\boldsymbol{\zeta}, \boldsymbol{\theta}) \quad (3)$$

where $\mathbf{Nom} = [p_x, p_y, p_z, \beta_x, \beta_y, \beta_z]^{\mathrm{T}}$ is called nominal system.

The mapping $\mathbf{Nom}(\cdot)$ provides the connection between the joint angle $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_7]^{\mathrm{T}}$, the geometric parameter $\boldsymbol{\zeta} \in \mathbf{R}^{21\times1}$ including $\boldsymbol{a}$, $\boldsymbol{d}$ and $\boldsymbol{\alpha} \in \mathbf{R}^{7\times1}$ in the joint space and the position and orientation of the end-effector in the Cartesian space. Then, the differential on both sides of (3) can be written as follows:

$$\mathrm{d}\mathbf{Nom} = \frac{\partial f(\boldsymbol{\zeta}, \boldsymbol{\theta})}{\partial \boldsymbol{\zeta}}\mathrm{d}\boldsymbol{\zeta} + \frac{\partial f(\boldsymbol{\zeta}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}}\mathrm{d}\boldsymbol{\theta} = \boldsymbol{J}_1(\boldsymbol{\zeta}, \boldsymbol{\theta})\mathrm{d}\boldsymbol{\zeta} + \boldsymbol{J}_2(\boldsymbol{\zeta}, \boldsymbol{\theta})\mathrm{d}\boldsymbol{\theta} \quad (4)$$

where $\boldsymbol{J}_1(\boldsymbol{\zeta}, \boldsymbol{\theta}) \in \mathbf{R}^{6\times21}$ and $\boldsymbol{J}_2(\boldsymbol{\zeta}, \boldsymbol{\theta}) \in \mathbf{R}^{6\times7}$ represent the Jacobian matrix and usually are abbreviated as $\boldsymbol{J}_1$ and $\boldsymbol{J}_2$.

2.3 Geometric error modeling

Geometric errors are the deviations between the actual and nominal kinematic parameters. The actual system can be defined as the nominal system with geometric errors $\Delta\boldsymbol{\zeta}$ and random colored noise $\Delta\boldsymbol{\theta}$, which can be given by

$$\mathbf{Act} = f(\boldsymbol{\zeta} + \Delta\boldsymbol{\zeta}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}) \quad (5)$$

where geometric errors $\Delta\boldsymbol{\zeta} \in \mathbf{R}^{21\times1}$ include $\Delta\boldsymbol{a}$, $\Delta\boldsymbol{d}$ and $\Delta\boldsymbol{\alpha} \in \mathbf{R}^{7\times1}$. Specifically, $\Delta\boldsymbol{a}$, $\Delta\boldsymbol{d}$, $\Delta\boldsymbol{\alpha}$ and $\Delta\boldsymbol{\theta}$ are defined as

$$\Delta\boldsymbol{a} = [\Delta a_1, \Delta a_2, \cdots, \Delta a_7]^{\mathrm{T}} \in \mathbf{R}^{7\times1},$$

$$\Delta\boldsymbol{d} = [\Delta d_1, \Delta d_2, \cdots, \Delta d_7]^{\mathrm{T}} \in \mathbf{R}^{7\times1},$$

$$\Delta\boldsymbol{\alpha} = [\Delta\alpha_1, \Delta\alpha_2, \cdots, \Delta\alpha_7]^{\mathrm{T}} \in \mathbf{R}^{7\times1},$$

$$\Delta\boldsymbol{\theta} = [\Delta\theta_1, \Delta\theta_2, \cdots, \Delta\theta_7]^{\mathrm{T}} \in \mathbf{R}^{7\times1}.$$

Then, the positional and orientational accuracy of the EI-MoCap system, namely the end-effector errors, is defined as follows:

$$\boldsymbol{\Delta}_T = \|\mathbf{Act} - \mathbf{Nom}\|_2 = \|f(\boldsymbol{\zeta} + \Delta\boldsymbol{\zeta}, \boldsymbol{\theta} + \Delta\boldsymbol{\theta}) - f(\boldsymbol{\zeta}, \boldsymbol{\theta})\|_2 \quad (6)$$

where $\boldsymbol{\Delta}_T = [l_x, l_y, l_z, \delta_x, \delta_y, \delta_z]^{\mathrm{T}}$.

2.4 Gyroscopic drift modeling

Gyroscopic drift is usually defined as a random colored noise and without loss of generality, we define the gyroscopic drift as a first-order Gauss-Markov process for approximation in this paper, which is given by

$$\boldsymbol{y}_{k+1} = \mathrm{e}^{-\Delta t_{k+1}/\tau}\boldsymbol{y}_k + \boldsymbol{\gamma}_k \quad (7)$$

where $\boldsymbol{y}_k = [\Delta\omega_{1k}, \Delta\omega_{2k}, \cdots, \Delta\omega_{7k}]^{\mathrm{T}}$ denotes the palstance error state at the time k. Δt_{k+1} represents the time interval between the time k and $k+1$. τ denotes the correlation time. $\boldsymbol{\gamma}_k$ is a discrete white noise, and the variance

$Q_k = \sigma^2(1-\mathrm{e}^{-\Delta t_{k+1}/\tau})$ (i.e., $\boldsymbol{\gamma}_k \sim \mathrm{N}(\boldsymbol{0}, Q_k, \boldsymbol{I})$).

According to (7), the first-order Gauss-Markov process needs two parameters (i.e., τ and σ) to describe. τ is the length of a past period that affects the present moment, which reflects the memory degree for past information, σ affects the rangeability of the first-order Gauss-Markov process (i.e., the more σ is, the more deviation rangeability of the first-order Gauss-Markov process is).

Then, integrating both sides of (7) and the following expression can be obtained:

$$\boldsymbol{x}_{k+1} = \mathrm{e}^{-\Delta t_{k+1}/\tau}\boldsymbol{x}_k + \boldsymbol{\rho}_k \tag{8}$$

where $\boldsymbol{x}_k = [\Delta\theta_{1k}, \Delta\theta_{2k}, \cdots, \Delta\theta_{7k}]^{\mathrm{T}}$ denotes the joint angle error parameters (kinematic parameters to be calibrated) at the time k. $\boldsymbol{\rho}_k \sim \mathrm{N}(\boldsymbol{0}, \Delta t_{k+1}^2 \cdot Q_k, \boldsymbol{I})$ is a discrete white noise.

According to (8), one can see that the random colored noise $\Delta\boldsymbol{\theta}$ (joint angle error parameters) follows the first-order Gauss-Markov process.

2.5 Nonlinear inequality constraints

Clearly, the EI-MoCap system is fixed on the human upper limb and the kinematic parameters of it should satisfy some fundamental biological characteristics. It is worth noting that those biological characteristics could be converted to some nonlinear inequality constraints for the EI-MoCap system. Specifically, $\boldsymbol{\theta}$ should be constrained by the reasonable joint rotation angle. Furthemore, $\boldsymbol{a}$, $\boldsymbol{d}$ and $\boldsymbol{\alpha}$ should be constrained by the anatomical scale.

According to the SAE J833 standard, we obtain the geometric parameter ranges as Table 2 shows. Then, by analyzing the rotation ranges of human upper limb based on the anatomical principles, we obtain the joint rotation angle ranges as shown in Table 3.

Table 2 Geometric parameter range of each joint m

Parameter	Min	Max
a_1	0.02	0.06
a_2	0.02	0.06
a_3	0.377	0.485
a_4	0.211	0.267

Table 3 Rotation range of each joint (°)

Parameter	Initial	Final
θ_1	−90	90
θ_2	−120	30
θ_3	−10	100
θ_4	0	150
θ_5	30	−20
θ_6	40	−30
θ_7	−90	90

According to Table 2 and Table 3, one can see that the nonlinear inequality constraints are given by

$$\begin{cases}\zeta_{i\min} \leqslant \zeta_i \leqslant \zeta_{i\max} \\ \theta_{i\min} \leqslant \theta_i \leqslant \theta_{i\max}\end{cases} \tag{9}$$

where $i = 1, 2, \cdots, 7$.

2.6 Calibration problem formulation

The main purpose of this study is to improve the EI-MoCap precision. In order to finish this purpose, the geometric errors $\Delta\boldsymbol{\zeta}$ and the random colored noise $\Delta\boldsymbol{\theta}$ need to be calibrated. As discussed in Subsection 2.4, the random colored noise $\Delta\boldsymbol{\theta}$ is defined as a first-order Gauss-Markov process. Hence, it will be a biased estimate for $\Delta\boldsymbol{\zeta}$ and $\Delta\boldsymbol{\theta}$ when the traditional LS method is used for calibration. Based on this, we calibrate those kinematic parameters under the EM framework for filtering. In order to reduce the iteration steps of the EM algorithm, the LS method can be used for giving an initial value. The object function of the optimization is given by

$$\min_{\Delta\zeta, \Delta\theta} y = \sum_{k=1}^{K} \|\mathbf{Act}_k - \mathbf{Nom}_k\|_2, \quad k = 1, 2, \cdots, K$$
$$\text{s.t.} \quad (9) \tag{10}$$

where K represents the total number of data pairs (indexed by k).

By using $\boldsymbol{\zeta}' = \boldsymbol{\zeta} + \Delta\boldsymbol{\zeta}$ to replace $\boldsymbol{\zeta}$ and $\boldsymbol{\theta}' = \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$ to replace $\boldsymbol{\theta}$ in (3), $\boldsymbol{\Delta}_T$ will be decreased. Then, the distribution of $\Delta\theta_i$ is defined as a Gauss distribution, i.e., $\Delta\theta_i \sim \mathrm{N}(\mu_i, \delta_i)$, and calculated based on the ML estimation. The likelihood function is given as follows:

$$\ln p(z) = \sum_{k=1}^{K} \ln \sum_{i=1}^{7} \pi_i \mathrm{N}(z_k | \mu_{k,i} \kappa_{k,i}, \Sigma_{k,i}) \tag{11}$$

where π_i denotes mixing coefficient, and it must satisfy $0 \leqslant \pi_i \leqslant 1$. z_k represents the 1st row of the kth end-effector error $\boldsymbol{\Delta}_T$. $\mu_{k,i}$ denotes the expectation of $\Delta\theta_i$. $\kappa_{k,i}$ denotes the ith column element of $\boldsymbol{J}_2$. $\Sigma_{k,i}$ denotes the variance of $\Delta\theta_i$.

According to (11), one can see that it no longer has a closed-form analytical solution, due to the presence of the summation over i inside the logarithm. Hence, the parameters cannot be estimated by taking the derivative of the likelihood function. Alternatively, we can employ the EM algorithm to obtain a numerical solution. The optimization problem given in (10) and the ML problem given in (11) will be solved by the LS method and the EM algorithm in Section 3.

3. Kinematic calibration based on LS method and EM algorithm

In this section, we establish the geometric error calibra-

tion model at first. Then, we calibrate the geometric errors $\Delta\boldsymbol{\zeta}$ and the random colored noise $\Delta\boldsymbol{\theta}$ by the LS method. Furthermore, we give the derivation process of $\Delta\boldsymbol{\theta}$ under the EM framework.

3.1 Geometric error calibration model

The relationship of the end-effector errors $\boldsymbol{\Delta}_T$, the random colored noise $\Delta\boldsymbol{\theta}$, and the geometric errors $\Delta\boldsymbol{\zeta}$ is presented in (6). For calculating $\Delta\boldsymbol{\theta}$ and $\Delta\boldsymbol{\zeta}$, the geometric error calibration model is established from the kinematics part and the differential part of this equation. The kinematics part reflects the relationship between the deviation of each joint and the deviation of the end-effector. In addition, the differential part reflects the relationship between the geometric errors $\Delta\boldsymbol{\zeta}$, the random colored noise $\Delta\boldsymbol{\theta}$, and the deviation of each joint. The derivation process is given by the following two parts.

(i) The kinematics part

The first-order differential approximation of the transformation matrix is given by

$$\mathrm{d}\boldsymbol{T}_{i-1}^{i}=\boldsymbol{T}_{i-1}^{i\mathrm{A}}-\boldsymbol{T}_{i-1}^{i\mathrm{N}}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\cdot\delta\boldsymbol{T}_{i-1}^{i} \tag{12}$$

where $\boldsymbol{T}_{i-1}^{i\mathrm{A}}$ denotes the actual transformation matrix, $\boldsymbol{T}_{i-1}^{i\mathrm{N}}$ denotes the nominal transformation matrix, and $\delta\boldsymbol{T}_{i-1}^{i}$ represents the motion error matrix.

According to (12), $\delta\boldsymbol{T}_{i-1}^{i}$ is given by

$$\delta\boldsymbol{T}_{i-1}^{i}=\left(\boldsymbol{T}_{i-1}^{i\mathrm{N}}\right)^{-1}\cdot\left(\boldsymbol{T}_{i-1}^{i\mathrm{A}}-\boldsymbol{T}_{i-1}^{i\mathrm{N}}\right)=\begin{bmatrix}0&-\delta_{zi}&\delta_{yi}&l_{xi}\\ \delta_{zi}&0&-\delta_{xi}&l_{yi}\\ -\delta_{yi}&\delta_{xi}&0&l_{zi}\\ 0&0&0&0\end{bmatrix} \tag{13}$$

where $[\delta_{xi},\delta_{yi},\delta_{zi}]^{\mathrm{T}}$ is the orientation error vector, and $[l_{xi},l_{yi},l_{zi}]^{\mathrm{T}}$ is the position error vector.

For the EI-MoCap system, the MoCap precision is affected by the kinematic parameters of each joint. Hence, the homogeneous matrix $\boldsymbol{T}_0^{7\mathrm{A}}$ in the Cartesian coordinate system can be expressed as

$$\boldsymbol{T}_0^{7\mathrm{A}}=\boldsymbol{T}_0^{7\mathrm{N}}+\mathrm{d}\boldsymbol{T}_0^{7}=\prod_{i=1}^{7}\left(\boldsymbol{T}_{i-1}^{i\mathrm{N}}+d\boldsymbol{T}_{i-1}^{i}\right)=$$

$$\boldsymbol{T}_0^{7\mathrm{N}}+\sum_{i=1}^{7}\left(\boldsymbol{T}_0^1\cdot\boldsymbol{T}_1^2\cdots\boldsymbol{T}_{i-2}^{i-1}\cdot\mathrm{d}\boldsymbol{T}_{i-1}^{i}\cdot\boldsymbol{T}_i^{i+1}\cdots\boldsymbol{T}_6^7\right). \tag{14}$$

According to (12), (14) can be converted as follows:

$$\boldsymbol{T}_0^{7\mathrm{A}}=\boldsymbol{T}_0^{7\mathrm{N}}+\sum_{i=1}^{7}\left(\boldsymbol{T}_0^1\cdot\boldsymbol{T}_1^2\cdots\boldsymbol{T}_{i-2}^{i-1}\cdot\mathrm{d}\boldsymbol{T}_{i-1}^{i}\cdot\boldsymbol{T}_i^{i+1}\cdots\boldsymbol{T}_6^7\right)=$$

$$\boldsymbol{T}_0^{7\mathrm{N}}+\sum_{i=1}^{7}\left(\boldsymbol{T}_0^{7\mathrm{N}}\cdot\left(\boldsymbol{T}_i^{i+1}\cdots\boldsymbol{T}_6^7\right)^{-1}\cdot\delta\boldsymbol{T}_{i-1}^{i}\cdot\boldsymbol{T}_i^{i+1}\cdots\boldsymbol{T}_6^7\right). \tag{15}$$

From (14) and (15), one can see that

$$\mathrm{d}\boldsymbol{T}_0^7=\boldsymbol{T}_0^{7\mathrm{N}}\sum_{i=1}^{7}\left(\boldsymbol{U}_{i+1}^{-1}\cdot\delta\boldsymbol{T}_{i-1}^{i}\cdot\boldsymbol{U}_{i+1}\right) \tag{16}$$

where $\boldsymbol{U}_i=\boldsymbol{T}_{i-1}^{i}\boldsymbol{T}_i^{i+1}\cdots\boldsymbol{T}_6^7=\begin{bmatrix}\boldsymbol{f}_i&\boldsymbol{o}_i&\boldsymbol{c}_i&\boldsymbol{p}_i\\ 0&0&0&1\end{bmatrix}$, and $\boldsymbol{U}_8=\boldsymbol{I}_4$ [28].

By combining (12) and (16), $\delta\boldsymbol{T}_0^7$ can be described as follows:

$$\delta\boldsymbol{T}_0^7=\sum_{i=1}^{7}\left(\boldsymbol{U}_{i+1}^{-1}\cdot\delta\boldsymbol{T}_{i-1}^{i}\cdot\boldsymbol{U}_{i+1}\right)=\begin{bmatrix}0&-\delta_z&\delta_y&l_x\\ \delta_z&0&-\delta_x&l_y\\ -\delta_y&\delta_x&0&l_z\\ 0&0&0&0\end{bmatrix}. \tag{17}$$

Then, the end-effector position and orientation error vector $\boldsymbol{\Delta}_T$ can be written as

$$\boldsymbol{\Delta}_T=\begin{bmatrix}l_x\\ l_y\\ l_z\\ \delta_x\\ \delta_y\\ \delta_z\end{bmatrix}=\sum_{i=1}^{7}\left(\begin{bmatrix}\boldsymbol{f}_{i+1}^{\mathrm{T}}&\boldsymbol{F}_1^{\mathrm{T}}\\ \boldsymbol{o}_{i+1}^{\mathrm{T}}&\boldsymbol{F}_2^{\mathrm{T}}\\ \boldsymbol{c}_{i+1}^{\mathrm{T}}&\boldsymbol{F}_3^{\mathrm{T}}\\ \boldsymbol{0}_{(1\times3)}&\boldsymbol{F}_4^{\mathrm{T}}\\ \boldsymbol{0}_{(1\times3)}&\boldsymbol{F}_5^{\mathrm{T}}\\ \boldsymbol{0}_{(1\times3)}&\boldsymbol{F}_6^{\mathrm{T}}\end{bmatrix}\cdot\begin{bmatrix}l_i\\ \delta_i\end{bmatrix}\right) \tag{18}$$

where $\boldsymbol{l}_i=[l_{xi},l_{yi},l_{zi}]^{\mathrm{T}}$, $\boldsymbol{\delta}_i=[\delta_{xi},\delta_{yi},\delta_{zi}]^{\mathrm{T}}$, $\boldsymbol{F}_1=\boldsymbol{p}_{i+1}\times\boldsymbol{f}_{i+1}$, $\boldsymbol{F}_2=\boldsymbol{p}_{i+1}\times\boldsymbol{o}_{i+1}$, $\boldsymbol{F}_3=\boldsymbol{p}_{i+1}\times\boldsymbol{c}_{i+1}$, $\boldsymbol{F}_4=\boldsymbol{o}_{i+1}\times\boldsymbol{c}_{i+1}$, $\boldsymbol{F}_5=\boldsymbol{c}_{i+1}\times\boldsymbol{f}_{i+1}$, $\boldsymbol{F}_6=\boldsymbol{f}_{i+1}\times\boldsymbol{o}_{i+1}$.

(ii) The differential part

We construct the first-order differential approximation of the transformation matrix from the differential part as follows:

$$\mathrm{d}\boldsymbol{T}_{i-1}^{i}=\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial a_i}\Delta a_i+\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial d_i}\Delta d_i+\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial\alpha_i}\Delta\alpha_i+\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial\theta_i}\Delta\theta_i. \tag{19}$$

Let $\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial a_i}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\cdot\boldsymbol{C}_{ai}$, $\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial d_i}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\cdot\boldsymbol{C}_{di}$, $\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial\alpha_i}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\cdot\boldsymbol{C}_{\alpha i}$, $\frac{\partial\boldsymbol{T}_{i-1}^{i}}{\partial\theta_i}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\cdot\boldsymbol{C}_{\theta i}$, then by combining them with (1), $\boldsymbol{C}_{ai}$, $\boldsymbol{C}_{di}$, $\boldsymbol{C}_{\alpha i}$, and $\boldsymbol{C}_{\theta i}$ can be expressed as

$$\begin{cases}\boldsymbol{C}_{ai}=\begin{bmatrix}0&0&0&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\end{bmatrix}\\ \boldsymbol{C}_{di}=\begin{bmatrix}0&0&0&0\\ 0&0&0&\sin\alpha_i\\ 0&0&0&\cos\alpha_i\\ 0&0&0&0\end{bmatrix}\\ \boldsymbol{C}_{\theta i}=\begin{bmatrix}0&-\cos\alpha_i&\sin\alpha_i&0\\ \cos\alpha_i&0&0&a_i\cos\alpha_i\\ -\sin\alpha_i&0&0&-a_i\sin\alpha_i\\ 0&0&0&0\end{bmatrix}\\ \boldsymbol{C}_{\alpha i}=\begin{bmatrix}0&0&0&0\\ 0&0&-1&0\\ 0&1&0&0\\ 0&0&0&0\end{bmatrix}\end{cases}. \tag{20}$$

By substituting (20) into (19), $\mathrm{d}\boldsymbol{T}_{i-1}^{i}$ can be reconsti-

tuted as

$$\mathrm{d}\boldsymbol{T}_{i-1}^{i}=\boldsymbol{T}_{i-1}^{i\mathrm{N}}\left(\boldsymbol{C}_{ai}\Delta a_i+\boldsymbol{C}_{di}\Delta d_i+\boldsymbol{C}_{\alpha i}\Delta\alpha_i+\boldsymbol{C}_{\theta i}\Delta\theta_i\right). \tag{21}$$

According to (12), one can see that

$$\delta\boldsymbol{T}_{i-1}^{i}=\boldsymbol{C}_{ai}\Delta a_i+\boldsymbol{C}_{di}\Delta d_i+\boldsymbol{C}_{\alpha i}\Delta\alpha_i+\boldsymbol{C}_{\theta i}\Delta\theta_i. \tag{22}$$

By combining (20) and (22), $\delta\boldsymbol{T}_{i-1}^{i}$ is given by

$$\delta\boldsymbol{T}_{i-1}^{i}=\begin{bmatrix}0 & -\cos\alpha_i\Delta\theta_i & \sin\alpha_i\Delta\theta_i & \Delta a_i\\ \cos\alpha_i\Delta\theta_i & 0 & -\Delta\alpha_i & M_1\\ -\sin\alpha_i\Delta\theta_i & \Delta\alpha_i & 0 & M_2\\ 0 & 0 & 0 & 0\end{bmatrix} \tag{23}$$

where $M_1=\sin\alpha_i\Delta d_i+a_i\cos\alpha_i\Delta\theta_i$ and $M_2=\cos\alpha_i\Delta d_i-a_i\sin\alpha_i\Delta\theta_i$.

According to (23), one can see that the position errors $\boldsymbol{l}_i$ and the orientation errors $\boldsymbol{\delta}_i$ between the i joint and the $i+1$ joint are given by

$$\boldsymbol{l}_i=\begin{bmatrix}\Delta a_i\\ \sin\alpha_i\Delta d_i+a_i\cos\alpha_i\Delta\theta_i\\ \cos\alpha_i\Delta d_i-a_i\sin\alpha_i\Delta\theta_i\end{bmatrix}=$$

$$\begin{bmatrix}1\\0\\0\end{bmatrix}\Delta a_i+\begin{bmatrix}0\\\sin\alpha_i\\\cos\alpha_i\end{bmatrix}\Delta d_i+\begin{bmatrix}0\\a_i\cos\alpha_i\\-a_i\sin\alpha_i\end{bmatrix}\Delta\theta_i, \tag{24}$$

$$\boldsymbol{\delta}_i=\begin{bmatrix}\Delta\alpha_i\\\sin\alpha_i\Delta\theta_i\\\cos\alpha_i\Delta\theta_i\end{bmatrix}=\begin{bmatrix}1\\0\\0\end{bmatrix}\Delta\alpha_i+\begin{bmatrix}0\\\sin\alpha_i\\\cos\alpha_i\end{bmatrix}\Delta\theta_i. \tag{25}$$

Let $\boldsymbol{k}_{i,1}=[1,0,0]^{\mathrm{T}}$, $\boldsymbol{k}_{i,2}=[0,\sin\alpha_i,\cos\alpha_i]^{\mathrm{T}}$, $\boldsymbol{k}_{i,3}=[0,a_i\cos\alpha_i,-a_i\sin\alpha_i]^{\mathrm{T}}$, $\boldsymbol{k}_{i,4}=[1,0,0]^{\mathrm{T}}$, $\boldsymbol{k}_{i,5}=[0,\sin\alpha_i,\cos\alpha_i]^{\mathrm{T}}$, and (24) and (25) can be simplified as follows:

$$\begin{cases}\boldsymbol{l}_i=\boldsymbol{k}_{i,1}\Delta a_i+\boldsymbol{k}_{i,2}\Delta d_i+\boldsymbol{k}_{i,3}\Delta\theta_i\\ \boldsymbol{\delta}_i=\boldsymbol{k}_{i,4}\Delta\alpha_i+\boldsymbol{k}_{i,5}\Delta\theta_i\end{cases}. \tag{26}$$

By submitting (26) into (18), we can obtain

$$\boldsymbol{\Lambda}_T=\sum_{i=1}^{7}\left(\begin{bmatrix}\boldsymbol{f}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,1} & \boldsymbol{f}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,2} & \boldsymbol{F}_1^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{f}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,3}+\boldsymbol{F}_1^{\mathrm{T}}\boldsymbol{k}_{i,5}\\ \boldsymbol{o}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,1} & \boldsymbol{o}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,2} & \boldsymbol{F}_2^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{o}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,3}+\boldsymbol{F}_2^{\mathrm{T}}\boldsymbol{k}_{i,5}\\ \boldsymbol{c}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,1} & \boldsymbol{c}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,2} & \boldsymbol{F}_3^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{c}_{i+1}^{\mathrm{T}}\boldsymbol{k}_{i,3}+\boldsymbol{F}_3^{\mathrm{T}}\boldsymbol{k}_{i,5}\\ 0 & 0 & \boldsymbol{F}_4^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{F}_4^{\mathrm{T}}\boldsymbol{k}_{i,5}\\ 0 & 0 & \boldsymbol{F}_5^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{F}_5^{\mathrm{T}}\boldsymbol{k}_{i,5}\\ 0 & 0 & \boldsymbol{F}_6^{\mathrm{T}}\boldsymbol{k}_{i,4} & \boldsymbol{F}_6^{\mathrm{T}}\boldsymbol{k}_{i,5}\end{bmatrix}\cdot\begin{bmatrix}\Delta a_i\\\Delta d_i\\\Delta\alpha_i\\\Delta\theta_i\end{bmatrix}\right)=\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 & \boldsymbol{A}_4\\ \boldsymbol{0}_{(3\times7)} & \boldsymbol{0}_{(3\times7)} & \boldsymbol{A}_5 & \boldsymbol{A}_6\end{bmatrix}\cdot\begin{bmatrix}\Delta\boldsymbol{a}\\\Delta\boldsymbol{d}\\\Delta\boldsymbol{\alpha}\\\Delta\boldsymbol{\theta}\end{bmatrix}. \tag{27}$$

Then, the geometric errors $\Delta\boldsymbol{\zeta}$ and the random colored noise $\Delta\boldsymbol{\theta}$ can be obtained by the LS method, according to (27). The results are given by

$$\left[\Delta\boldsymbol{\zeta}^{\mathrm{T}},\Delta\boldsymbol{\theta}^{\mathrm{T}}\right]^{\mathrm{T}}=\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Lambda}_T \tag{28}$$

where $\boldsymbol{\Phi}=\begin{bmatrix}\boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 & \boldsymbol{A}_4\\ \boldsymbol{0}_{(3\times7)} & \boldsymbol{0}_{(3\times7)} & \boldsymbol{A}_5 & \boldsymbol{A}_6\end{bmatrix}$.

3.2 Formulation under EM framework

3.2.1 EM method revisited

The EM algorithm is a general method for finding ML solutions for probabilistic models having latent variables. Consider a probabilistic model in which we collectively denote all of the observed variables by Z and all of the hidden variables by M. The joint distribution $p(Z,M|\Theta)$ is governed by a set of parameters denoted by Θ. The goal is to maximize the ML function that is given by

$$p(Z|\Theta)=\sum_{M}p(Z,M|\Theta). \tag{29}$$

We shall suppose that direct optimization of $p(Z|\Theta)$ is difficult, but that optimization of the ML function $p(Z,M|\Theta)$ is significantly easier. Then we introduce a distribution $q(M)$ defined over the latent variables, and we observe that, for any choice of $q(M)$, the following decomposition holds:

$$\ln p(Z|\Theta)=L(M,\Theta)+\mathrm{KL}(q\|p) \tag{30}$$

where

$$L(M,\Theta)=\sum_{M}q(M)\ln\left\{\frac{p(Z,M|\Theta)}{q(M)}\right\}, \tag{31}$$

$$\mathrm{KL}(q\|p)=-\sum_{M}q(M)\ln\left\{\frac{p(M|Z,\Theta)}{q(M)}\right\}. \tag{32}$$

Note that $L(M,\Theta)$ is a functional of the distribution $q(M)$ and parameter Θ. From (32), we see that $\mathrm{KL}(q\|p)$ is the Kullback-Leibler divergence between $q(M)$ and posterior distribution $p(M|Z,\Theta)$.

The EM algorithm is a two-stage iterative optimization method for finding ML solutions. The decomposition (30) can be used to define the EM algorithm and to demonstrate that it does indeed maximize the log likelihood.

Suppose that Θ^{old} denotes the current value of the parameter vector. In the E-step, the lower bound $L(M,\Theta)$ is maximized with respect to $q(M)$ while holding Θ^{old} fixed. In the subsequent M-step, the distribution $q(M)$ is held fixed and the lower bound $L(M,\Theta)$ is maximized with respect to Θ for giving some new value Θ^{new}. We have seen that both the E and the M steps of the EM algorithm are increasing the value of the log likelihood function. The EM algorithm will converge when the log likelihood function is already at a maximum or the parameters remain unchanged.

3.2.2 Kinematic calibration under EM framework

The EM algorithm introduces free joint distribution $\Delta\theta_i\sim\mathrm{N}(\mu_i,\delta_i)$, also named as prior distribution, as an approximation of the distribution of the parameter $\Delta\theta_i$ to

calculate the log likelihood function as shown in (11). We first choose some initial values for the means, variances, and mixing coefficients based on the results of the LS problem. Then we alternate between the following two updates that should be called the *E*-step and the *M*-step, to increase the log likelihood function.

(i) EM *E*-step

In the expectation step, or *E*-step, we use the current values for the parameters to evaluate the posterior probabilities, or responsibilities, which can be given by

$$\gamma(m_{k,i}) = \frac{\pi_i \mathrm{N}(z_k|\mu_{k,i}\kappa_{k,i}, \Sigma_{k,i})}{\sum_{j=1}^{7} \pi_j \mathrm{N}(z_k|\mu_{k,i}\kappa_{k,i}, \Sigma_{k,i})}. \tag{33}$$

(ii) EM *M*-step

Then we use these responsibilities in the maximization step, or *M*-step, to reestimate the means, variances, and mixing coefficients using the current responsibilities:

$$\mu_i^{\text{new}} = \frac{1}{N_i} \sum_{k=1}^{K} \gamma(m_{k,i}) z_k / \boldsymbol{\kappa}_{k,i}, \tag{34}$$

$$\Sigma_i^{\text{new}} = \frac{1}{N_i} \sum_{k=1}^{K} \gamma(m_{k,i}) \boldsymbol{E}\boldsymbol{E}^{\mathrm{T}}, \tag{35}$$

$$\pi_i^{\text{new}} = \frac{N_i}{K}, \tag{36}$$

where $\boldsymbol{E} = (z_k - \mu_i^{\text{new}} \boldsymbol{\kappa}_{k,i})$ and $N_i = \sum_{k=1}^{K} \gamma(m_{k,i})$.

In practice, the EM algorithm is deemed to have converged when the change in the log likelihood function, or alternatively in the parameters, falls below some threshold.

4. Simulation study

In this section, the kinematic calibration simulation is conducted based on the LS method and the EM algorithm respectively. Consider the following kinematic parameters.

In Table 4, $\Delta\theta(k)$ follows the first-order Gauss-Markov process as discussed in Subsection 2.4. The parameters in the process are defined as $\Delta_{t_{k+1}} = 0.01$ s, $\tau = 1$ s, and $\sigma = 1$. We can see that there are some geometric errors between Table 1 and Table 4. Then, the hypothetical system is defined as the actual system and the standard system is defined as the nominal system, the geometric errors are calibrated based on the LS method and the EM algorithm respectively in the following Subsection 4.1 and Subsection 4.2.

Table 4 Hypothetical D-H parameters of the EI-MoCap system

Rod	a_i/m	α_i/(°)	d_i/m	$\theta_i + \Delta\theta_i$/(°)	Offset/(°)
1	0.03	270	0.03	$\theta_1(k) + \Delta\theta_1(k)$	0
2	0.05	93	0.02	$\theta_2(k) + \Delta\theta_2(k)$	90
3	0.401	90	0	$\theta_3(k) + \Delta\theta_3(k)$	0
4	0.264	−95	0	$\theta_4(k) + \Delta\theta_4(k)$	0
5	0.02	90	0.01	$\theta_5(k) + \Delta\theta_5(k)$	0
6	0.01	92	0	$\theta_6(k) + \Delta\theta_6(k)$	90
7	0	90	0	$\theta_7(k) + \Delta\theta_7(k)$	0

4.1 Calibration under single LS method

By following a random trajectory, considering the end-effector errors $\boldsymbol{\Delta}_T$, we get the calibration result based on (28) as shown in Table 5.

Table 5 Calibration result under the single LS method

Rod	Δa_i/m	$\Delta\alpha_i$/rad	Δd_i/m	$\Delta\theta_i$/rad
1	−0.0102	0.0005	0.0302	0.0004
2	0.0080	0.0514	0.0183	0.0010
3	−0.0280	0.0002	−0.0003	−0.0010
4	0.0180	−0.0856	−0.0021	−0.0013
5	0.0203	0.0010	0.0104	−0.0005
6	0.0100	0.0336	−0.0010	0.0004
7	0.0008	−0.0001	−0.0014	−0.0016

In addition, the calibration effect by drawing the motion trajectory of the actual system, the nominal system, and the nominal system with calibration results are demonstrated in Fig. 3.

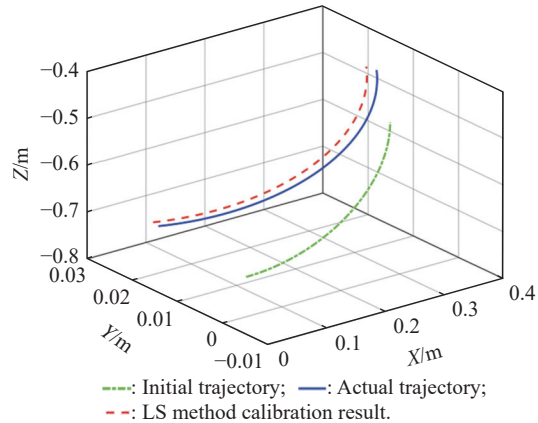


Fig. 3 Effect of the single LS method

In Fig. 3, it can be seen that the trajectory after calibration is closer to the actual trajectory. The objective value y between the actual and initial trajectory is 7.8153 and

between the actual and calibrated trajectory is 0.621 1. The objective value is reduced by 92.05% when the LS method is applied. Hence, the geometric error calibration model as shown in (27) can effectively apply in the EI-MoCap system and improve the MoCap precision.

4.2 Calibration under EM framework

Table 5 displays the parameter estimation result and Fig. 3 displays the calibration effect under the single LS method. By inserting the calibration result $\Delta\boldsymbol{\zeta}$ into (3) for compensating, the end-effector errors can be converted as follows:

$$\boldsymbol{\Delta}_T = \|f(\boldsymbol{\zeta}+\Delta\boldsymbol{\zeta},\boldsymbol{\theta}+\Delta\boldsymbol{\theta})-f(\boldsymbol{\zeta}+\Delta\boldsymbol{\zeta},\boldsymbol{\theta})\|. \tag{37}$$

Compared with (6), one can see that $\boldsymbol{\Delta}_T$ is decreased in (37). Then, in this subsection, we will calibrate $\Delta\boldsymbol{\theta}$ under the EM framework. Let $\Delta\boldsymbol{\theta}$ calibrated by the LS method be the initial expectation of the prior distribution. We get the calibration result under the EM framework as shown in Table 6.

Table 6 Calibration result under the EM framework

Parameter	$\Delta\theta_1$	$\Delta\theta_2$	$\Delta\theta_3$	$\Delta\theta_4$	$\Delta\theta_5$	$\Delta\theta_6$	$\Delta\theta_7$
Mean value	−0.087 7	−0.021 9	−0.864 4	0.571 9	−0.164 4	−0.136 9	−0.215 5
Variance	0.000 4	0.000 5	0.026 1	0.011 4	0.000 1	0.001 5	0.000 6
Responsibility	0.032 1	0.013 4	0.000 0	0.000 0	0.536 6	0.000 1	0.417 9

Table 6 displays the parameter estimation result. As the simulation results suggest, the EM algorithm has better parameter estimation accuracy than the single LS method for giving the confidence interval of the parameters. Then, let the expectation of the posteriori distribution times the corresponding responsibility as the estimation value, the calibration effect is demonstrated by giving the motion trajectory in Fig. 4.

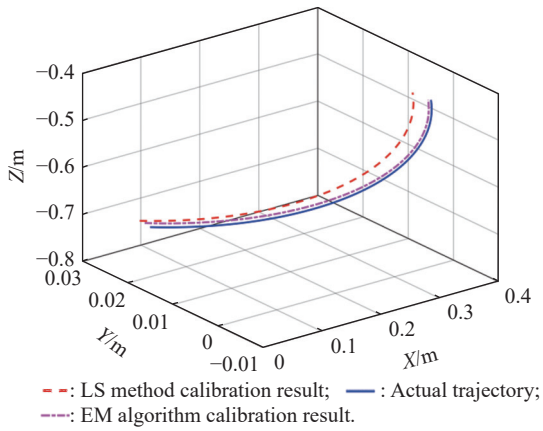


Fig. 4 Effect of the EM algorithm

In Fig. 4, one can see that the EM algorithm can significantly improve the motion capture accuracy. The objective value y between the actual and calibrated trajectory under the LS method is 0.621 1 and between the actual and calibrated trajectory under the EM algorithm is 0.516 8. One can see that the EM algorithm is better than the LS method. The capture precision significantly improves by 16.79% in comparison to the single LS method. In order to verify the convergence of the ML problem shown in (11) under the EM framework, we calculate the value of the likelihood function in the solution process, as shown in Fig. 5.

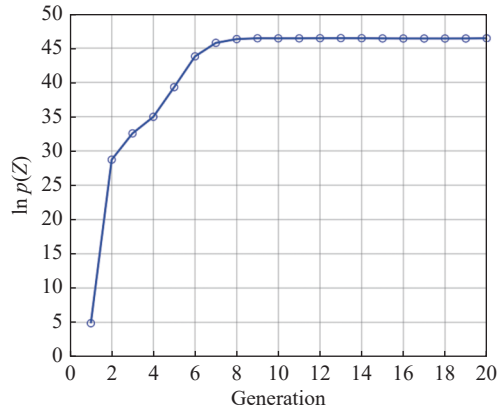


Fig. 5 EM algorithm convergence verification

It can be seen from Fig. 5 that as the generation of the EM algorithm increases, the likelihood function value gradually increases and converges to a fixed value. Therefore, it can be verified that using the EM algorithm to solve the ML problem is convergent. In Section 5, the feasibility of the proposed method is verified on the designed EI-MoCap system.

5. Experimental evaluation

The kinematic calibration experiment is conducted under the EM framework on the designed EI-MoCap system (see Fig. 6) in this section. The EI-MoCap system includes three 9-DoFs IMU (with three axis digital accelerometers, three gyroscopes, and three magnetometers). The dynamic accuracy of the IMU is 2° and the static accuracy is up to 0.7°. The central processing unit (CPU) consists of a K60DN512 with an ARM Cortex-M4 kernel and the frequency of the CPU is 100 MHz.

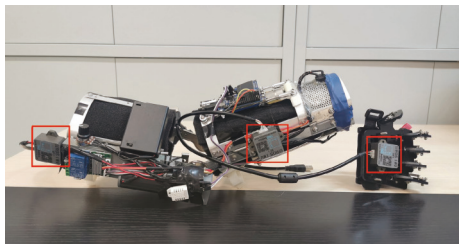


Fig. 6 EI-MoCap system

The specific calibration procedures are as follows.

Step 1 Wearing the EI-MoCap system to complete a series of random actions.

Step 2 Calculating the joint angular velocity $\dot{\theta}$ according to the gyroscope output.

Step 3 Obtaining the joint angle θ, according to the combination of gyroscopes, accelerometers and magnetometers.

Step 4 Submitting joint angle θ into the initial D-H parameters (see Table 1) and obtaining the terminal nominal transformation matrix $\boldsymbol{T}_0^{7\mathrm{N}}$ combined with (2).

Step 5 According to the lever effect, the terminal actual transformation matrix $\boldsymbol{T}_0^{7\mathrm{A}}$ can be obtained.

Step 6 Calculating the end-effector position and orientation error vector $\boldsymbol{\Delta}_T$ according to $\boldsymbol{T}_0^{7\mathrm{N}}$ and $\boldsymbol{T}_0^{7\mathrm{A}}$.

Step 7 submitting $\boldsymbol{\Delta}_T$ into (28) to obtain $\Delta\boldsymbol{\zeta}$ and $\Delta\boldsymbol{\theta}$.

Step 8 Introducing the EM algorithm as Subsection 3.2.2 shows to calibrate the random colored noise $\Delta\boldsymbol{\theta}$.

According to the above process, the results of the experiments are shown in Table 7, Table 8, Fig. 7, and Fig. 8.

Table 7 Calibration result under the LS method

Rod	Δa_i/m	$\Delta\alpha_i$/rad	Δd_i/m	$\Delta\theta_i$/rad
1	−0.0099	−0.0004	0.0300	−0.0002
2	0.0106	0.0518	0.0199	0.0004
3	−0.0304	0.0006	0.0001	−0.0006
4	0.0183	−0.0862	−0.0008	0.0001
5	0.0203	0.0003	0.0093	0.0005
6	0.0091	0.0351	−0.0002	−0.0005
7	−0.0004	0.0001	0.0101	0.0005

Table 8 Kinematic parameters calibrated by the EM algorithm

Parameter	Mean value	Variance	Responsibility
$\Delta\theta_1$	0.0396	0.0001	0.3409
$\Delta\theta_2$	0.2752	0.0002	0.0001
$\Delta\theta_3$	−1.1635	0.0444	0.0000
$\Delta\theta_4$	−0.2356	0.0093	0.0000
$\Delta\theta_5$	0.3317	0.0159	0.0000
$\Delta\theta_6$	0.0787	0.0016	0.6258
$\Delta\theta_7$	−0.3303	0.0002	0.0332

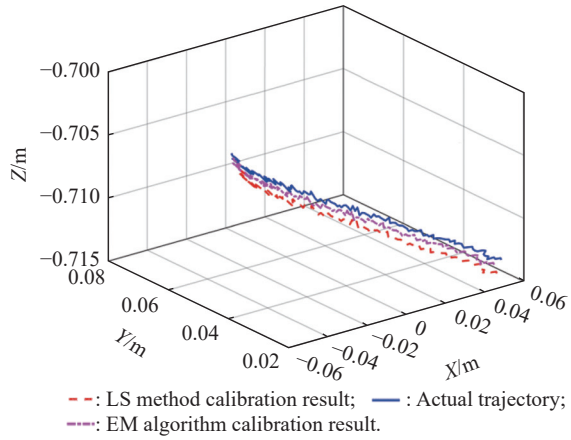


Fig. 7 Effect of calibration under the EM framework

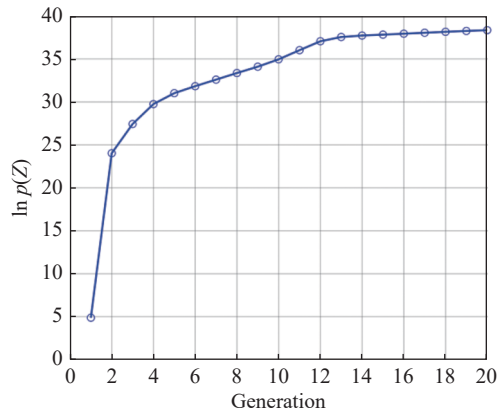


Fig. 8 Convergence verification under the EM framework

Let the standard kinematic parameters (see Table 1) be the initial value, we implement kinematic calibration based on the proposed method. By following a random trajectory, the EI-MoCap system calibrates the kinematic parameters and outputs the accurate trajectory. The results are shown in Table 7,Table 8, and Fig. 7.

Let the calibration result of the LS method which is displayed in Table 7 be the initial values of the EM algorithm at first. Then we can get the calibration result under the EM framework as shown in Table 8.

Then we demonstrate the calibration effect by giving the motion trajectory as shown in Fig. 7.

In Fig. 7, one can see that the kinematic calibration under the EM framework can calibrate the kinematic parameters effectively. The objective value y between the actual and calibrated trajectory under the LS method is 0.3507 and between the actual and calibrated trajectory under the EM algorithm is 0.3256.

One can see that the EM algorithm is better than the LS method. The capture precision is significantly improved by 7.16% in comparison to the LS method. Then we calculate the value of the likelihood function in the solution process (see Fig. 8). Therefore, it can be verified that the

solution is convergent on the semi-physical experiment. Hence, the proposed method can be better used in the embedded system than the LS method.

6. Conclusions

This paper proposes a kinematic calibration method for EI-MoCap system based on the EM algorithm. The proposed method mainly includes two parts. First, the LS method is used to calculate an initial calibration result which is used to reduce the iterations of the EM algorithm. Second, the EM algorithm is proposed to calibrate the parameters influenced by the random colored noise. In comparison to the single LS method, the kinematic calibration effects are improved. The LS method is a biased estimation under the condition of the random colored noise. The simulation and experiment demonstrate the motion capture precision significantly improved by 16.79% and 7.16% respectively in comparison to the LS method. In future studies, the proposed kinematic calibration method could be generalized to the industrial arm.

References

[1] NASER A, LOTFI A, MWANJE M D, et al. Privacy-preserving, thermal vision with human in the loop fall detection alert system. IEEE Trans. on Human-Machine Systems, 2023, 53(1): 164–175.

[2] GAO L, ZHANG G F, YU B, et al. Wearable human motion posture capture and medical health monitoring based on wireless sensor networks. Measurement, 2020, 166: 108252.

[3] AMAT A Z, ADIANI D, TAUSEEF M, et al. Design of a desktop virtual reality-based collaborative activities simulator to support teamwork in workplace settings for autistic adults. IEEE Trans. on Neural Systems and Rehabilitation Engineering, 2023, 31: 2184–2194.

[4] DING Z Y, LOO J Y, NURZAMAN S G, et al. A zero-shot soft sensor modeling approach using adversarial learning for robustness against sensor fault. IEEE Trans. on Industrial Informatics, 2023, 19(4): 5891–5901.

[5] DONG Z H, LUCES J V S, RAVANKAR A A, et al. A performance evaluation of overground gait training with a mobile body weight support system using wearable sensors. IEEE Sensors Journal, 2023, 23(11): 12209–12223.

[6] SCOTT B, CHADWICK E, MCINNES M, et al. Assessing single camera markerless motion capture during upper limb activities of daily living. Gait & Posture, 2023, 106(1): S184.

[7] FAREWIK L G, OLSSON F, DESTRO A, et al. Markerless motion capture using iPad pro with LiDAR camera adjusted with artificial neural networks. Gait & Posture, 2022, 97(1): S72–S73.

[8] USLU T, GEZGIN E, OZBEK S, et al. Utilization of low cost motion capture cameras for virtual navigation procedures: performance evaluation for surgical navigation. Measurement, 2021, 181: 109624.

[9] PARK K W, CHOI J, KONG K. Data-driven modeling for gait phase recognition in a wearable exoskeleton using estimated forces. IEEE Trans. on Robotics, 2023, 39(4): 3072–3086.

[10] WANG W T, LI R, DIEKEL Z M, et al. Controlling object hand-over in human-robot collaboration via natural wearable sensing. IEEE Trans. on Human-Machine Systems, 2019, 49(1): 59–71.

[11] El-GOHARY M, MCNAMES J. Shoulder and elbow joint angle tracking with inertial sensors. IEEE Trans. on Biomedical Engineering, 2012, 59(9): 2635–2641.

[12] LU C H, DAI Z Y, JING L. Measurement of hand joint angle using inertial-based motion capture system. IEEE Trans. on Instrumentation and Measurement, 2023, 72: 9503211.

[13] WANG Z T, GAO F, WU Z H, et al. A method for calculating lower extremity anatomical landmark trajectories based on inertial motion capture data. IEEE Trans. on Neural Systems and Rehabilitation Engineering, 2023, 31: 2734–2746.

[14] LAMBRECHT J M, KIRSCH R F. Miniature low-power inertial sensors: promising technology for implantable motion capture systems. IEEE Trans. on Neural Systems and Rehabilitation Engineering, 2014, 22(6): 1138–1147.

[15] QIU S, WANG Z L, ZHAO H Y, et al. Using distributed wearable sensors to measure and evaluate human lower limb motions. IEEE Trans. on Instrumentation and Measurement, 2016, 65(4): 939–950.

[16] ROBERT-LACHAINE X, PARENT G, FUENTES A, et al. Inertial motion capture validation of 3D knee kinematics at various gait speed on the treadmill with a double-pose calibration. Gait & Posture, 2020, 77: 132–137.

[17] MIEZAL M, TAETZ B, BLESER G. On inertial body tracking in the presence of model calibration errors. Sensors, 2016, 16(7): 1132.

[18] KIM M, LEE D. Wearable inertial sensor based parametric calibration of lower-limb kinematics. Sensors and Actuators A: Physical, 2017, 265(1): 280–296.

[19] JIN X, HUANG B, SHOOK D S. Multiple model LPV approach to nonlinear process identification with EM algorithm. Journal of Process Control, 2011, 21(1): 182–193.

[20] ZHAO Y J, FATEHI A, HUANG B. Robust estimation of ARX models with time varying time delays using variational bayesian approach. IEEE Trans. on Cybernetics, 2018, 48(2): 532–542.

[21] SUN Y F, XU X. Calibration of MEMS triaxial accelerometers based on the maximum likelihood estimation method. Mathematical Problems in Engineering, 2020, 2020: 4617365.

[22] ZHANG X G, LI P, TU R, et al. Automatic calibration of process noise matrix and measurement noise covariance for multi-GNSS precise point positioning. Mathematics, 2020, 8(4): 502.

[23] DENG Y H, CHANG J Y. Human-like posture correction for seven degree of freedom robotic arm. Journal of Mechanisms and Robotics, 2022, 14(2): 024501.

[24] SHINTEMIROV A, TAUNYAZOV T, OMARALI B, et al. An open-source 7-DoF wireless human arm motion-tracking system for use in robotics research. Sensors, 2020, 20(11): 3082.

[25] TAUNYAZOV T, OMARALI B, SHINTEMIROV A. A novel low-cost 4-DoF wireless human arm motion tracker. Proc. of the IEEE International Conference on Biomedical Robotics and Biomechatronics, 2016: 157−162.

[26] ZAHEDI A, WANG Y S, LAU N, et al. A bamboo-inspired exoskeleton based on carbon fiber for shoulder and elbow joints. IEEE Trans. on Medical Robotics and Bionics, 2023,

5(2): 375–386.

[27] WANG Z Y, LIU Y, WANG X Y, et al. Field calibration method for industrial robots based on single position sensitive device. IEEE Trans. on Instrumentation and Measurement, 2023, 72: 7506112.

[28] WU C. A kinematic cad tool for the design and control of a robot manipulator. International Journal of Robotics Research, 1984, 3(1): 58–67.

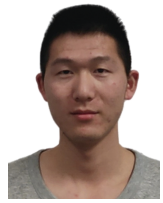
Biographies

QIN Weiwei was born in 1982. He received his M.S. degree in guidance, navigation, and control and Ph.D. degree in control science and engineering from the Rocket Force University of Engineering, in 2008 and 2012, respectively, where he is currently an associate professor with the First College. His research interests include model predictive control, robust active control, and human motion capture.
E-mail: qww_1982@163.com

GUO Wenxin was born in 1997. He received his M.S. degree in guidance, navigation, and control in control science and engineering from the Rocket Force University of Engineering in 2021, where he is currently a research assistant with the First College. His research interests include human motion capture, kinematic calibration, model predictive control, and robust active control.
E-mail: gwx_97@163.com

HU Chen was born in 1989. He received his B.S. degree in automation from the Department of Automation, Xiamen University, Xiamen, China, in 2011, and M.S. and Ph.D. degrees in control science and engineering from the Rocket Force University of Engineering, Xi'an, China, in 2014 and 2018, respectively. He is currently a lecturer with the Rocket Force University of Engineering. His research interests include multiagent systems, distributed state estimation, and machine learning.
E-mail: chenh628@hotmail.com

LIU Gang was born in 1964. He received his M.S. degree in guidance, navigation, and control and Ph.D. degree in control science and engineering from Northwestern Polytechnical University, China, in 1995 and 1999, respectively. He is currently a professor with the Rocket Force University of Engineering. His research interests include guidance, navigation and control, and hypersonic flight vehicle.
E-mail: 1813923249@qq.com



SONG Tainian was born in 1997. He received his M.S. degree in guidance, navigation, and control from the Rocket Force University of Engineering in 2021, where he is currently a research assistant with the First College. His research interests include human motion capture and target detection.
E-mail: 873512274@qq.com