Robotica (2022), **40**, pp. 2953–2968
doi:10.1017/S0263574721002034

CAMBRIDGE
UNIVERSITY PRESS

RESEARCH ARTICLE

Optimal robot-world and hand-eye calibration with rotation and translation coupling

Xiao Wang and Hanwen Song*

School of Aerospace Engineering and Applied Mechanics, Tongji University, Shanghai 200092, China
*Corresponding author. E-mail: hwsong@tongji.edu.cn

Received: 4 June 2021; **Accepted:** 29 December 2021; **First published online:** 27 January 2022

Keywords: hand-eye calibration, robot-world calibration, Lie algebra, Jacobian, singularity

Abstract

A classic hand-eye system involves hand-eye calibration and robot-world and hand-eye calibration. Insofar as hand-eye calibration can solve only hand-eye transformation, this study aims to determine the robot-world and hand-eye transformations simultaneously based on the robot-world and hand-eye equation. According to whether the rotation part and the translation part of the equation are decoupled, the methods can be divided into separable solutions and simultaneous solutions. The separable solutions solve the rotation part before solving the translation part, so the estimated errors of the rotation will be transferred to the translation. In this study, a method was proposed for calculation with rotation and translation coupling; a closed-form solution based on Kronecker product and an iterative solution based on the Gauss–Newton algorithm were involved. The feasibility was further tested using simulated data and real data, and the superiority was verified by comparison with the results obtained by the available method. Finally, we improved a method that can solve the singularity problem caused by the parameterization of the rotation matrix, which can be widely used in the robot-world and hand-eye calibration. The results show that the prediction errors of rotation and translation based on the proposed method be reduced to 0.26° and 1.67 mm, respectively.

1. Introduction

With the development of advanced vision technology and robotics [1], the application fields of robotics are becoming more and more extensive [2, 3]. Vision-based robotic technology is widely used in many fields [4]; typical examples include robotic-assisted minimally invasive surgery [5, 6], robotic object catching [7], and on-orbit services [8]. Robot vision systems involve eye-in-hand, eye-to-hand, and hybrid forms [9, 13], in which the robot uses image information as feedback to perform related tasks. Such a system including a robot and a sensor (one or more cameras) equipped at the end of the robot is a typical hand-eye system.

The determination of the robot-world transformation and the hand-eye transformation involved in the hand-eye system is commonly known as hand-eye calibration problems. The mathematical models for the hand-eye calibration problems can all be grouped into the hand-eye equation ($\boldsymbol{AX} = \boldsymbol{XB}$) [14] and the robot-world and hand-eye equation ($\boldsymbol{AX} = \boldsymbol{YB}$) [15]. Despite the hand-eye equation can only determine the hand-eye transformation, the dual equation of hand-eye equation can determine the robot-world transformation [16]. The observation data involved in the robot-world and hand-eye equation consist of absolute poses, while the observation data involved in the dual equations consist of relative motions. Therefore, it is mandatory to compute the robot and camera relative motions before solving hand-eye equation. In other words, simultaneous robot-world and hand-eye calibration based on the robot-world and hand-eye equation ($\boldsymbol{AX} = \boldsymbol{YB}$) can reduce data generation errors.

Hand-eye calibration was proposed by Shiu et al. [14] and Tsai et al. [17] and was pointed out that solving the hand-eye equation needs at least two motions with non-parallel rotation axes. Early research

mainly focused on linear algorithms. The hand-eye equation can be decoupled into the rotation and translation parts. Estimating the rotation and translation parts sequentially is commonly referred to as the separable solutions. The rotation part of the separable solutions was parameterized based on quaternion [18, 20], Euclidean group [21], and Kronecker product [22], and then, the estimated value of rotation was required as a priori information to calculate the translation. However, the separable solutions caused a problem that the error of the rotation is inevitably transferred to the translation part. Obviously, calculating rotation and translation simultaneously can effectively avoid propagation errors. A screw motion can represent rigid body motion based on Chasles theorem, and it is a geometric interpretation that regards rotation and translation as an interdependent entirety [23]. Zhao and Liu [24] employed the screw motion theory to establish a hand-eye equation based on quaternion and yielded a simultaneous result for rotation and translation by solving linear equations. Dual quaternions as the algebraic counterpart of screws facilitated a simultaneous solution for the hand-eye rotation and translation using singular value decomposition (SVD) and were proposed in refs. [25, 26]. Furthermore, a probabilistic method for hand-eye calibration [27] was feasible even without prior knowledge of the correspondence between the data streams obtained via different systems. Since the linear algorithms are sensitive to possible random noise, a lot of research has been carried out on iterative optimization as an effective method to improve accuracy and robustness. Zhuang and Shiu [28] solved hand-eye problem based on Gauss–Newton method, which the rotation parameters formulated in Euler angles. The nonlinear least square method was proposed in ref. [19], where the rotation part was represented by quaternions. Furthermore, Strobl and Hirzinger [29] proposed a novel physically based metric on $SE(3)$ that can select the weighting of the rotation part and the translation part automatically. The global optimal method proposed in ref. [30] did not require initial estimates. The cost function constructed by minimizing the reprojection error was proposed in ref. [31], which was lately developed by considering the hand pose errors [32]. There is an exponential mapping between Lie algebra $so(3)$ and the rotation matrix as it belongs to the special orthogonal group $SO(3)$, which makes it easy to obtain the derivative of the rotation matrix. The Jacobian matrix in the Gauss–Newton algorithm can be constructed with the partial derivative of Lie algebra $so(3)$. Yang and Zhao [33] applied Gauss–Newton algorithm to optimize the estimation, but the estimation of translation was not accurate enough to be used in practice. Hand-eye calibration has been conducted using extensive methods, but it can only determine the hand-eye transformation.

The robot-world and hand-eye calibration was proposed by Zhuang et al. [15], and a separable solution based on quaternion was implemented. A new robot-world and hand-eye calibration algorithm based on Kronecker product, which could separably determine the rotation and translation by applying the SVD approach, was presented in ref. [34]. Typically, separate solution based on Kronecker product has good rotation accuracy; however, the position accuracy is often compromised [35], while the Kronecker product and the dual quaternions can also be applied to solve rotation and translation simultaneously [36]. The probabilistic method was also applied on robot-world and hand-eye calibration [37]. A nonlinear least square method was presented in robot-world and hand-eye calibration [38], where the rotation part was represented by a matrix without parameterization. Different parameterization methods of rotation matrices based on Euler angle, axial angle, or quaternion were compared in ref. [39]. Zhao and Weng proposed a joint method that gives the solutions of the cameras' parameters and the hand-eye parameters simultaneously using nonlinear optimization [40]. The global optimal methods [41, 42] were also found to be instrumental in determining robot-world and hand-eye calibration. Gauss–Newton algorithm in which the Jacobian matrix is constructed by Lie algebra was applied for dual robots [10], hybrid system [9], and other conditions [43]. However, these methods only optimized the rotation part and did not consider rotation and translation simultaneously. In addition, Gauss–Newton algorithm in which the Jacobian matrix is constructed by Lie algebra has not been applied to the classic robot-world and hand-eye calibration. Hence, this study gave a detailed process of optimizing classic robot-world and hand-eye calibration based on the Gauss–Newton algorithm and realized simultaneous estimation of the rotation and the translation.

In this paper, a simultaneous solution for the robot-world and hand-eye equation was proposed. The simultaneous solution includes a closed-form solution that serves as an initial value and an iterative

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

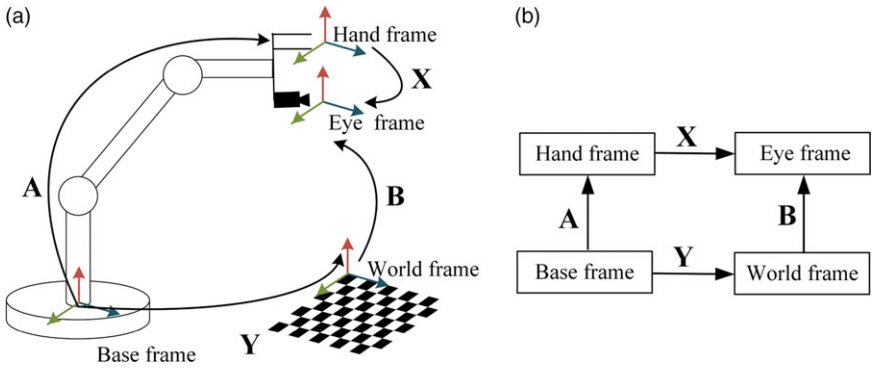

Figure 1. *The robot-world and hand-eye kinematics loop*

solution. The overdetermined linear equation system corresponding to the closed-form solution was constructed based on the Kronecker product, and such problem was converted into linear null space calculations. Applying SVD on the linear equation system led to the closed-form solution. The Gauss–Newton algorithm is based on Lie algebra to calculate the Jacobian matrix to implement the iterative solution. The contributions of this paper can be listed as:

(1) A Jacobian matrix based on Lie algebra that can optimize rotation and translation simultaneously was derived and refined a closed-form solution, thus forming a more complete method.
(2) An improved method to solve the singularity problem caused by the parameterization of the rotation matrix was proposed, which can be widely used in the robot-world and hand-eye calibration.
(3) The feasibility of the Jacobian matrix has been verified that it can also converge to a better accuracy when other linear method is used as the initial value.

The rest of this article is organized as follows. In Section 2, we derived the left perturbation model of the rotation matrix and constructed the Jacobian matrix based on this model, forming a complete solution including the initial value and the iterative solution. Experiments with simulated data and real data are performed and the results of which are discussed in Section 3. Conclusions are given in Section 4.

2. Proposed method

2.1. Problem formulation

The classic robot-world and hand-eye calibration has the setup similar to the one described in Fig. 1(a), and the corresponding kinematic loop illustrated in Fig. 1(b) can be mapped to the robot-world and hand-eye equation:

$$\boldsymbol{AX} = \boldsymbol{YB}. \tag{1}$$

where $\boldsymbol{A}$ represents the absolute pose of the robot; $\boldsymbol{B}$ represents the absolute pose of the camera; $\boldsymbol{X}$ represents the hand-eye transformation; and $\boldsymbol{Y}$ represents the robot-world transformation. The camera moves around the calibration checkerboard and collects n images of the checkerboard with the corresponding robot poses, thus leading to the calibration equation system.

2.2. Closed-form solution

The Kronecker product [44], usually denoted by $\otimes$, can construct a linear equation system for the nonlinear robot-world and hand-eye equation (Eq. (1)). Using an $m \times n$ matrix $\boldsymbol{A}$ and a $p \times q$ matrix

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

$\boldsymbol{B}$, a larger matrix with the dimension of $mp \times nq$ can be defined as:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \left[a_{ij}\boldsymbol{B} \right] = \begin{bmatrix} a_{11}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix}. \tag{2}$$

In the linear equation system, another linear operator, denoted by *vec*, can vectorize a matrix by reordering the coefficients into a single column vector as:

$$vec(\boldsymbol{A}) = [a_{11}, \cdots, a_{1n}, a_{21}, \cdots, a_{mn}]^T. \tag{3}$$

According to the fundamental principle of the Kronecker product [44]:

$$\begin{aligned} vec(\boldsymbol{AX}) &= (\boldsymbol{A} \otimes \boldsymbol{I}_4) vec(\boldsymbol{X}); \\ vec(\boldsymbol{YB}) &= (\boldsymbol{I}_4 \otimes \boldsymbol{B}^T) vec(\boldsymbol{Y}), \end{aligned} \tag{4}$$

Eq. (1) can be represented as the following linear equation system:

$$\begin{bmatrix} \boldsymbol{A} \otimes \boldsymbol{I}_4 & -\boldsymbol{I}_4 \otimes \boldsymbol{B}^T \end{bmatrix}_{16\times 32} \begin{Bmatrix} vec(\boldsymbol{X}) \\ vec(\boldsymbol{Y}) \end{Bmatrix}_{32\times 1} = \boldsymbol{0}_{16\times 1}. \tag{5}$$

Each set of data constructs a linear equation of the form $\boldsymbol{Kx} = \boldsymbol{0}$, and stacking n linear equations can obtain an overdetermined equation system as follows:

$$\tilde{\boldsymbol{K}}\boldsymbol{x} = \boldsymbol{0}, \tag{6}$$

in which

$$\tilde{\boldsymbol{K}} = [\boldsymbol{K}_1^T, \boldsymbol{K}_2^T, \ldots, \boldsymbol{K}_n^T]^T|_{16n\times 32}. \tag{7}$$

The robot-world and hand-eye calibration problem is transformed into a linear null space calculation, which generates a unique solution based on the constraints of the orthogonal matrix. Applying SVD on the matrix $\tilde{\boldsymbol{K}}$ leads to the right singular vector $\boldsymbol{v}_K$, which is corresponding to the minimum singular value. Consequently, the matrices $\boldsymbol{V}_X$ and $\boldsymbol{V}_Y$ extracted from the null space of $\tilde{\boldsymbol{K}}$ are proportional to the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, which can be obtained as

$$\begin{aligned} \tilde{\boldsymbol{X}} &= \varphi \boldsymbol{V}_X; \\ \tilde{\boldsymbol{Y}} &= \varphi \boldsymbol{V}_Y. \end{aligned} \tag{8}$$

where $\boldsymbol{V}_X = vec^{-1}(\boldsymbol{v}_K(1{:}16))$ and $\boldsymbol{V}_Y = vec^{-1}(\boldsymbol{v}_K(17{:}32))$. The proportionality factor φ of the solution $\boldsymbol{x}$ is

$$\varphi = \frac{sign(det(\boldsymbol{V}_X(1{:}3, 1{:}3)))}{\sqrt[3]{|det(\boldsymbol{V}_X(1{:}3, 1{:}3))|}} \tag{9}$$

since φ is determined by the orthogonal constraint: $det(\boldsymbol{R}_X) = 1$.

Meanwhile, it is necessary to go through a re-orthogonalization procedure to guarantee the orthogonality of solutions $\boldsymbol{R}_X$ and $\boldsymbol{R}_Y$ as the presence of noise. The best re-orthogonalization can be obtained by applying SVD on the approximate matrix as:

$$\begin{aligned} \tilde{\boldsymbol{R}}_i &= \boldsymbol{USV}^T; \\ \boldsymbol{R}_i &= sign\,(det\,(\boldsymbol{S}))\,\boldsymbol{UV}^T, \end{aligned} \tag{10}$$

and i stands for the unknown $\boldsymbol{X}$, $\boldsymbol{Y}$.

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

2.3. Iterative solution

The closed-form solution obtained above can be used as the initial value of the Gauss–Newton algorithm iteration. In turn, the Gauss–Newton algorithm can improve the accuracy and robustness of the closed-form solution. The Gauss–Newton algorithm proposed in this study aims to optimize the rotation and translation parts simultaneously.

Since a rotation matrix belongs to the special orthogonal group $SO(3)$ [45], there is an exponential mapping from the 'axis-angle' representation of the rotation vector to the rotation matrix as:

$$\boldsymbol{R} = exp\left([\boldsymbol{\omega}]^{\wedge}\right) = \boldsymbol{I}_3 + \frac{[\boldsymbol{\omega}]^{\wedge}}{\|\boldsymbol{\omega}\|} sin(\|\boldsymbol{\omega}\|) + \frac{[\boldsymbol{\omega}]^{\wedge 2}}{\|\boldsymbol{\omega}\|^2}(1 - cos(\|\boldsymbol{\omega}\|)). \tag{11}$$

where $\boldsymbol{I}_3$ is a three order identify matrix; $[\boldsymbol{\omega}]^{\wedge}$ is the skew-symmetric matrix corresponding to the vector $\boldsymbol{\omega}$ and is given by:

$$[\boldsymbol{\omega}]^{\wedge} = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}. \tag{12}$$

$\boldsymbol{\omega}$ is the three independent variables of the rotation matrix based on the 'axis-angle' representation and $\boldsymbol{\omega} = \boldsymbol{k}\theta$, where $\boldsymbol{k}$ is the normalized unit axis of the rotation matrix and θ is the angle of the rotation matrix.

The frame transformations and absolute poses involved in the equations (Eq. (1)) have the same mathematical representation, that is, a homogeneous matrix including a rotation matrix and a translation vector. Therefore, Eq. (1) can decouple the rotation part and the translation part, and each can be expressed as a function as follows:

$$\begin{aligned} \boldsymbol{F}_1(\boldsymbol{\omega}_X, \boldsymbol{\omega}_Y) &= \boldsymbol{R}_A exp\left([\boldsymbol{\omega}_X]^{\wedge}\right) - exp\left([\boldsymbol{\omega}_Y]^{\wedge}\right)\boldsymbol{R}_B; \\ \boldsymbol{F}_2(\boldsymbol{\omega}_Y, \boldsymbol{t}_X, \boldsymbol{t}_Y) &= exp\left([\boldsymbol{\omega}_Y]^{\wedge}\right)\boldsymbol{t}_B + \boldsymbol{t}_Y - \boldsymbol{R}_A\boldsymbol{t}_X - \boldsymbol{t}_A. \end{aligned} \tag{13}$$

The basic method of the theory of Lie groups, which makes it possible to obtain deep results with striking simplicity, consists in reducing questions concerning Lie groups to certain problems of linear algebra [46]. It is not convenient to take the derivative of a rotation matrix because the perturbation of a rotation matrix is a multiplication operation. Therefore, taking the derivative of the vector corresponding to the rotation matrix can simplify the problem. The Jacobian matrix of the Gauss–Newton algorithm can be easily calculated by the addition operation of Lie algebra. We use the symbol $\boldsymbol{J}$ to represent the Jacobian matrix, and the iterative formula is $\boldsymbol{J}\Delta\boldsymbol{x} = \boldsymbol{f}$. Let $(\,)_i$ represents the i^{th} column vector in the bracket, and the Jacobian matrix can be determined by the equation as follows:

$$\boldsymbol{J} = \begin{bmatrix} \begin{bmatrix} \frac{\partial(\boldsymbol{F}_1)_1}{\partial\boldsymbol{\omega}_X} \\ \frac{\partial(\boldsymbol{F}_1)_2}{\partial\boldsymbol{\omega}_X} \\ \frac{\partial(\boldsymbol{F}_1)_3}{\partial\boldsymbol{\omega}_X} \end{bmatrix} & \begin{bmatrix} \frac{\partial(\boldsymbol{F}_1)_1}{\partial\boldsymbol{\omega}_Y} \\ \frac{\partial(\boldsymbol{F}_1)_2}{\partial\boldsymbol{\omega}_Y} \\ \frac{\partial(\boldsymbol{F}_1)_3}{\partial\boldsymbol{\omega}_Y} \end{bmatrix} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\partial\boldsymbol{F}_2}{\partial\boldsymbol{\omega}_Y} & \frac{\partial\boldsymbol{F}_2}{\partial\boldsymbol{t}_X} & \frac{\partial\boldsymbol{F}_2}{\partial\boldsymbol{t}_Y} \end{bmatrix}. \tag{14}$$

The key task of obtaining the Jacobian matrix is to calculate the left perturbation model of the rotation matrix. Taking the first-order approximation of the exponential function, we have

$$\partial\boldsymbol{R} = exp([\partial\boldsymbol{\omega}]^{\wedge}) \approx \boldsymbol{I}_3 + [\partial\boldsymbol{\omega}]^{\wedge}. \tag{15}$$

And the principle of cross product: $[\boldsymbol{a}]^{\wedge}\boldsymbol{b} = \boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a} = -[\boldsymbol{b}]^{\wedge}\boldsymbol{a}$, which is also instrumental in the calculation of the left perturbation model of the rotation matrix.

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

The left perturbation model of the rotation matrix in the help of the operation of Lie algebra is constructed by the following processes:

$$\begin{aligned}\frac{\partial(\boldsymbol{R}_A\boldsymbol{R}_X)}{\partial\boldsymbol{\omega}_X} &= \lim_{\partial\boldsymbol{\omega}_X\to 0}\frac{\boldsymbol{R}_A\partial\boldsymbol{R}_X\boldsymbol{R}_X-\boldsymbol{R}_A\boldsymbol{R}_X}{\partial\boldsymbol{\omega}_X}\\ &= \lim_{\partial\boldsymbol{\omega}_X\to 0}\frac{\boldsymbol{R}_A[\partial\boldsymbol{\omega}_X]^\wedge\boldsymbol{R}_X}{\partial\boldsymbol{\omega}_X}\\ &= \lim_{\partial\boldsymbol{\omega}_X\to 0}\frac{\boldsymbol{R}_A[\partial\boldsymbol{\omega}_X]^\wedge[(\boldsymbol{R}_X)_1,(\boldsymbol{R}_X)_2,(\boldsymbol{R}_X)_3]}{\partial\boldsymbol{\omega}_X}\\ &= \lim_{\partial\boldsymbol{\omega}_X\to 0}\frac{-\boldsymbol{R}_A[[(\boldsymbol{R}_X)_1]^\wedge\partial\boldsymbol{\omega}_X,[(\boldsymbol{R}_X)_2]^\wedge\partial\boldsymbol{\omega}_X,[(\boldsymbol{R}_X)_3]^\wedge\partial\boldsymbol{\omega}_X]}{\partial\boldsymbol{\omega}_X}.\end{aligned} \tag{16}$$

According to the left perturbation model of the rotation matrix, a complete and concrete expression of the Jacobian iteration formula $\boldsymbol{J}\Delta\boldsymbol{x}=\boldsymbol{f}$ can be obtained as:

$$\begin{bmatrix}\begin{bmatrix}-\boldsymbol{R}_A[(\boldsymbol{R}_X)_1]^\wedge\\-\boldsymbol{R}_A[(\boldsymbol{R}_X)_2]^\wedge\\-\boldsymbol{R}_A[(\boldsymbol{R}_X)_3]^\wedge\end{bmatrix} & \begin{bmatrix}[(\boldsymbol{R}_Y\boldsymbol{R}_B)_1]^\wedge\\[(\boldsymbol{R}_Y\boldsymbol{R}_B)_2]^\wedge\\[(\boldsymbol{R}_Y\boldsymbol{R}_B)_3]^\wedge\end{bmatrix} & \boldsymbol{0}_{9\times3} & \boldsymbol{0}_{9\times3}\\ \boldsymbol{0}_{3\times3} & -[\boldsymbol{R}_Y\boldsymbol{t}_B]^\wedge & -\boldsymbol{R}_A & \boldsymbol{I}_3\end{bmatrix}\begin{Bmatrix}\partial\boldsymbol{\omega}_X\\\partial\boldsymbol{\omega}_Y\\\partial\boldsymbol{t}_X\\\partial\boldsymbol{t}_Y\end{Bmatrix}=-\begin{Bmatrix}(\boldsymbol{R}_A\boldsymbol{R}_X-\boldsymbol{R}_Y\boldsymbol{R}_B)_1\\(\boldsymbol{R}_A\boldsymbol{R}_X-\boldsymbol{R}_Y\boldsymbol{R}_B)_2\\(\boldsymbol{R}_A\boldsymbol{R}_X-\boldsymbol{R}_Y\boldsymbol{R}_B)_3\\\boldsymbol{R}_Y\boldsymbol{t}_B+\boldsymbol{t}_Y-\boldsymbol{R}_A\boldsymbol{t}_X-\boldsymbol{t}_A\end{Bmatrix}. \tag{17}$$

Stacking n sets of Eq. (17) leads to

$$\tilde{\boldsymbol{J}}\Delta\boldsymbol{x}=\tilde{\boldsymbol{f}}, \tag{18}$$

in which

$$\tilde{\boldsymbol{J}}=[\boldsymbol{J}_1^T,\boldsymbol{J}_2^T,\ldots,\boldsymbol{J}_n^T]^T|_{12n\times12}; \tag{19}$$

$$\tilde{\boldsymbol{f}}=[\boldsymbol{f}_1^T,\boldsymbol{f}_2^T,\ldots,\boldsymbol{f}_n^T]^T|_{12n\times1}. \tag{20}$$

Then, the solution can be obtained as:

$$\Delta\boldsymbol{x}=(\tilde{\boldsymbol{J}}^T\tilde{\boldsymbol{J}})^{-1}\tilde{\boldsymbol{J}}^T\tilde{\boldsymbol{f}}. \tag{21}$$

$\Delta\boldsymbol{x}$ is subsequently used to update $\tilde{\boldsymbol{J}}$ and $\tilde{\boldsymbol{f}}$, and the process should be repeated until the limit of $\Delta\boldsymbol{x}$ is reached.

2.4. Singular value processing

For an arbitrary rotation matrix $\boldsymbol{R}$, rotation axis $\boldsymbol{k}$ is not unique. The results in this study were normalized based on the fact that rotation about an axis $\boldsymbol{k}$ by an angle of $\theta(0\le\theta\le\pi)$ is equivalent to rotation about the axis $-\boldsymbol{k}$ by the rotation angle of $(2\pi-\theta)$. Therefore, if the rotation angle satisfies $0\le\theta\le\pi$, then $\boldsymbol{\omega}=\boldsymbol{k}\theta$, and if $\pi\le\theta\le2\pi$, then $\boldsymbol{\omega}=-\boldsymbol{k}(2\pi-\theta)$. A more detailed method for extracting the rotation axis $\boldsymbol{k}$ from rotation matrix $\boldsymbol{R}$ is presented in refs. [45, 47]. But the problem is that the rotation axis $\boldsymbol{k}$ cannot be extracted when θ is equal to 0 or π. In another words, when the unknown rotation angle of $\boldsymbol{R}_X$ or $\boldsymbol{R}_Y$ is close to 0 or π, singular phenomenon occurs.

Song et al. [48] proposed a method to solve the singularity for the closed-form solutions of the hand-eye calibration, which expressed the unknown hand-eye rotation matrix by two non-singular rotation matrices. However, the singularity of robot-world and hand-eye calibration has not been discussed yet.

The singularity in robot-world and hand-eye calibration is more complicated because both unknown matrices have possible singular values. This study improves a method to solve the singularity problem, which can be widely used for simultaneous robot-world and hand-eye calibration. The specific process is as follows:

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

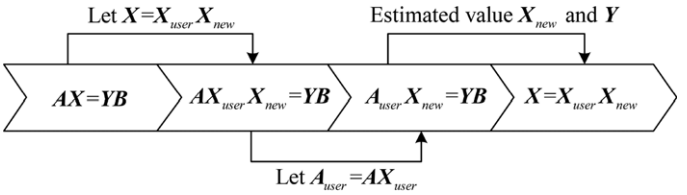

Figure 2. *Singular value processing flowchart: only $\boldsymbol{R}_X$ is singular value.*

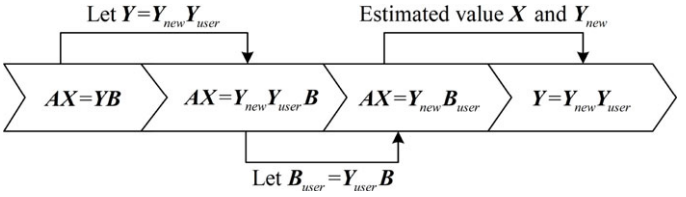

Figure 3. *Singular value processing flowchart: only $\boldsymbol{R}_Y$ is singular value.*

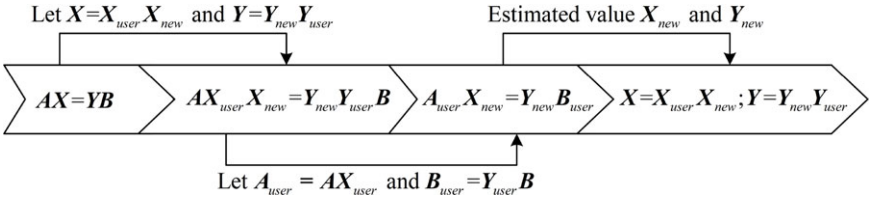

Figure 4. *Singular value processing flowchart: both $\boldsymbol{R}_X$ and $\boldsymbol{R}_Y$ are singular values.*

(1) Firstly, determine whether the angle of the linear estimated rotation matrices $\boldsymbol{R}_X$ and $\boldsymbol{R}_Y$ is close to the singular value or not. Singularity can be divided into three types of results: only $\boldsymbol{R}_X$ is singular value; only $\boldsymbol{R}_Y$ is singular value; both $\boldsymbol{R}_X$ and $\boldsymbol{R}_Y$ are singular values.

(2) According to the result of the first step, select the corresponding processing method.

Only $\boldsymbol{R}_X$ is singular value, and the singular value processing process is shown in Fig. 2.
Only $\boldsymbol{R}_Y$ is singular value, the singular value processing is shown in Fig. 3.
Both $\boldsymbol{R}_X$ and $\boldsymbol{R}_Y$ are singular values, and the singular value processing is shown in Fig. 4.
Note that the angle of the rotation of $\boldsymbol{X}_{\text{user}}$ or $\boldsymbol{Y}_{\text{user}}$ is between 0 and $2\pi/3$ according to ref. [48].

2.5. Error metrics

For the calculated values $\hat{\boldsymbol{R}}_X$ and $\hat{\boldsymbol{R}}_Y$, the error metric is defined as:

$$\theta_{\text{error}} = Rodrigues(\hat{\boldsymbol{R}}_i^{-1}\boldsymbol{R}_i), i = X, Y. \tag{22}$$

where $\boldsymbol{R}_i$ is the ground truth. $Rodrigues(\bullet)$ means to extract the 'axis-angle' from the homogeneous matrix, more detailed method in ref. [45].

For the calculated values $\hat{\boldsymbol{t}}_X$ and $\hat{\boldsymbol{t}}_Y$, the error metrics are defined as:

$$\boldsymbol{t}_{\text{error}} = \left\|\hat{\boldsymbol{t}}_i - \boldsymbol{t}_i\right\|_2, i = X, Y. \tag{23}$$

In order to compare the accuracy of all methods, root mean square was suggested to use,

$$\begin{aligned} \text{Error of } \boldsymbol{R} &= RMS(\boldsymbol{\theta}^1_{\text{error}}, \boldsymbol{\theta}^1_{\text{error}}, \ldots, \boldsymbol{\theta}^N_{\text{error}}); \\ \text{Error of } \boldsymbol{t} &= RMS(\boldsymbol{t}^1_{\text{error}}, \boldsymbol{t}^1_{\text{error}}, \ldots, \boldsymbol{t}^N_{\text{error}}). \end{aligned} \tag{24}$$

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

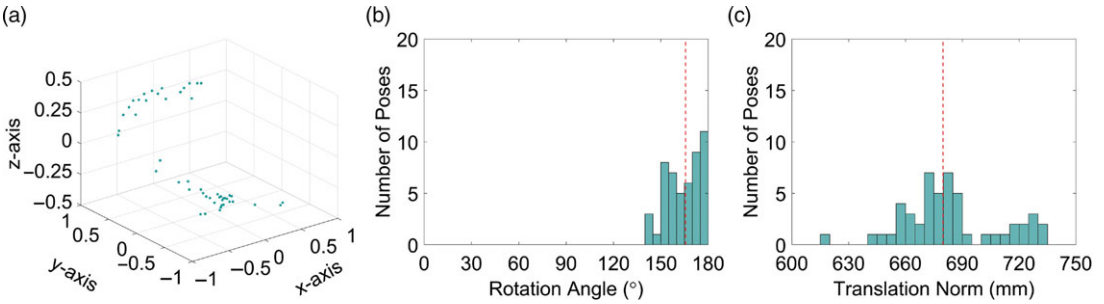

Figure 5. *Distribution of robot configurations.*

where N refers to the number of simulations in the simulation test and the number of verification datasets in the real experimental test.

3. Verification and discussions

In this section, comprehensive tests have been conducted to verify the feasibility and effectiveness of the closed-form solution and the iterative solution. Li et al. [36] constructed a linear equation system for robot-world and hand-eye calibration based on Kronecker product, which converted such problem into least square calculations. Same as the closed-form solution proposed in this study, Li's method can solve the rotations and translations problem simultaneously and serve as an initial value for the iterative solution. On the one hand, the Li's method can verify the feasibility of the closed-form solution as proposed in this study. On the other hand, the iterative solution requires an initial value as a nonlinear method, and the Li's method can be used to test the convergence efficiency of the iterative solution. The mnemonics 'Li' and 'I-Li', respectively, indicate the Li's method and the iterative method with the Li's method as the initial value. The closed-form solution and the iterative solution proposed in this study, respectively, denote as 'closed-form' and 'iterative'. In order to further improve the credibility of the method proposed in this study, two methods based on dual quaternion are added as reference in the real experiment. One method [26] (denotes as 'DQ') solves the rotation and translation problems simultaneously, and the other method [20] (denotes as 'IDQ') is to solve the rotation and translation problems separately.

An industrial robot (GSK RB03) with 6-degree-of-freedom (6-DOF) and a CMOS camera (Photonfocus $MV1 - D2048 * 1088 - 240 - CL$) with an 8 mm lens installed at the end of the robot have been employed for experimentation. The same settings were simulated in the synthesis experiment to closely integrate the simulated and real test results.

3.1. Experiments with simulated data

This section presents a detailed account of the number of datasets and the noise effect on the calibration accuracy. The accuracy of the robot-world and hand-eye calibration is affected by the robot configurations distribution [43], and some data selection methods were elaborated in refs. [49, 50]. Therefore, the robot configurations are selected within the field of view of the monocular camera to maintain consistency with the real experiment.

$\boldsymbol{A}_i$ $(i = 1, 2, \ldots, 50)$ came from 50 different robot configurations, which were obtained from the robot control system. Figure 5 shows the distribution of $\boldsymbol{A}_i$. Figure 5(a) is the distribution of the normalized rotation axis. The histograms of the rotation angle and translation norm are shown in Fig. 5(b) and (c), respectively, where the red line refers to the median. Assume that the ground truth of the robot-world and hand-eye transformations are shown in Table I. $\boldsymbol{B}_i$ $(i = 1, 2, \ldots, 50)$ were obtained through equation:

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

Table I. *The ground truth for the robot-world and hand-eye transformations*

Parameter	$\boldsymbol{R}_Y$	$\boldsymbol{R}_X$	$\boldsymbol{t}_Y$ **(mm)**	$\boldsymbol{t}_X$ **(mm)**
Value	$\boldsymbol{k}_Y = [1, 0, 0]^T$ $\theta_Y = \pi/4$	$\boldsymbol{k}_X = [1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}]^T$ $\theta_X = \pi/3$	$[200, 200, 200]^T$	$[100, 100, 100]^T$

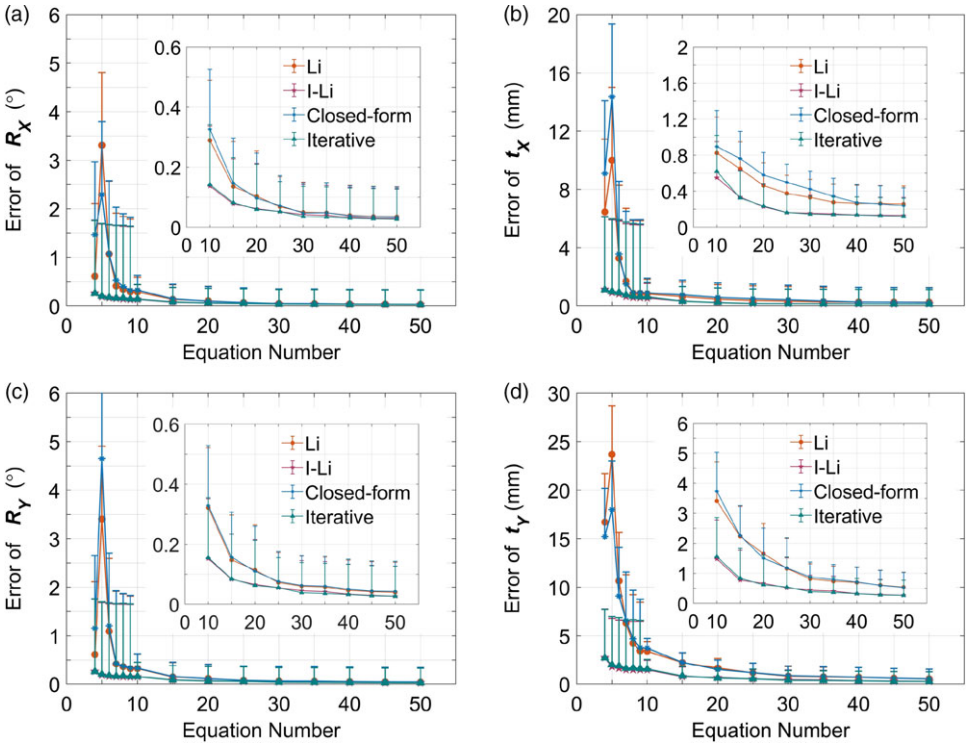

Figure 6. *Accuracy comparison based on different number of datasets.*

$\boldsymbol{B}_i = \boldsymbol{Y}^{-1}\boldsymbol{A}_i\boldsymbol{X}$. All methods were implemented based on MATLAB2019a. One hundred simulations were performed for each method.

Gaussian noise was added to all simulation data as the following equations:

$$\begin{aligned} \boldsymbol{R}_m^{noise} &= Rot\,(\boldsymbol{k}, (\theta_m + \theta_{noise}))\,; \\ \boldsymbol{t}_m^{noise} &= \boldsymbol{t}_m + \boldsymbol{t}_{noise}, \end{aligned} \tag{25}$$

where θ_{noise} and $\boldsymbol{t}_{noise}$ were the amplitude of the rotational noise and the translational noise, both were zero-mean-value. $\boldsymbol{t}_{noise}$ was created by adding noises with the same standard deviation on three directions, respectively. The subscript m denoted $\boldsymbol{A}_i$ and $\boldsymbol{B}_j$. The synthetic experiment verifies the superiority of the method proposed in this study in terms of accuracy convergence and noise sensitivity. In the accuracy convergence test part, the standard deviation of the rotation noise was $\sigma_{\theta_{noise}} = 0.1(°)$ and the noise standard deviation of the translation noise was $\sigma_{t_{noise}} = [0.1, 0.1, 0.1]^T$(mm). In the noise sensitivity test part, the noise standard deviation $\sigma_{\theta_{noise}}$ and $\sigma_{t_{noise}}$ were set with a maximum amplitude of 2(°) in rotation and $[2, 2, 2]^T$(mm) in translation and were evenly divided into 10 levels.

It is noted that the dataset number for the accuracy convergence test ranges from 4 to 50, since the overdetermined system is formed when the dataset number is greater than 3. Figure 6 shows the error curves corresponding to the four methods with different number of datasets, where the errors of $\boldsymbol{R}_X$, $\boldsymbol{t}_X$, $\boldsymbol{R}_Y$, and $\boldsymbol{t}_Y$ are illustrated in Fig. 6(a), (b), (c), and (d), respectively. Note that the error curves have similar

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

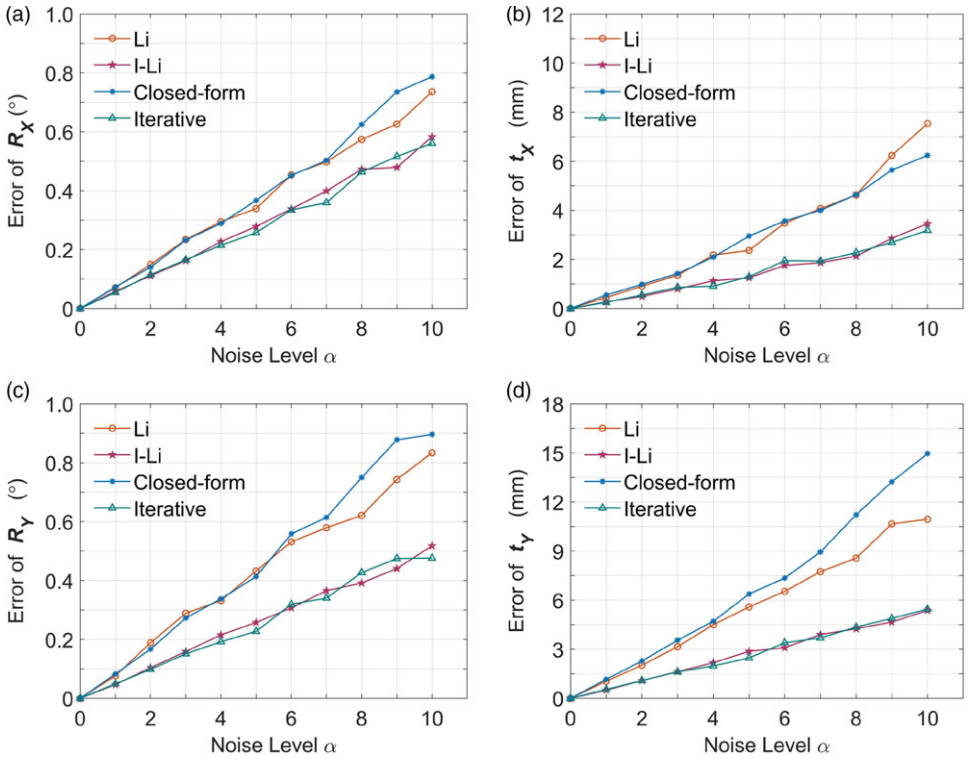

Figure 7. *Accuracy comparison based on different noise level.*

trends, and the accuracy of four methods grows with the number of datasets. However, the numerical value of accuracy and the speed of accuracy convergence are different. The results from 'closed-form' and 'Li' methods have similar trends, while the results from 'I-Li' and 'iterative' methods have almost the same trends. This means that the iterative method proposed in this study is suitable for the case where two linear solutions are used as initial values. Figure 6(a) shows that the accuracy of the $\boldsymbol{R}_X$ estimated by four methods converges to a stable value when the number of datasets exceeds 30. When the number of datasets is less than 10, the accuracy of two iterative methods converges faster. In other words, the iterative process can obtain a high-precision solution with data from limited number of robot configurations. Moreover, the accuracy of the iterative methods is better than that of the linear methods. Figure 6(c) shows the error curves of $\boldsymbol{R}_Y$ estimated by four methods, and the results are mostly the same as $\boldsymbol{R}_X$. Figure 6(b) shows the error curves of $\boldsymbol{t}_X$ estimated by four methods. The accuracy of the four methods converges a stable value as the number of datasets increases. When the number of datasets is less than 10, the accuracy of two iterative methods is superior to that of two linear methods. Moreover, the accuracy superiority of the iterative methods has not disappeared as the number of datasets increases. Figure 6(d) shows the error curves of $\boldsymbol{t}_Y$ estimated by the four methods, and the downward trends of error curves are similar to that of $\boldsymbol{t}_X$, but the numerical values are different, which is related to the different amplitudes of $\boldsymbol{t}_X$ and $\boldsymbol{t}_Y$ [43]. It can be seen that the iterative methods had faster convergence rate and higher accuracy compared with the linear methods.

All 50 sets of data were used in the noise sensitivity test part. Figure 7 shows the error curves corresponding to the four methods with different noise level, where the errors of $\boldsymbol{R}_X$, $\boldsymbol{t}_X$, $\boldsymbol{R}_Y$, and $\boldsymbol{t}_Y$ are illustrated in Fig. 7(a), (b), (c), and (d), respectively. Generally, the solution errors of the four methods are roughly linearly correlated with the noise level. Figure 7(a) shows the error curves of $\boldsymbol{R}_X$, and the iterative methods yield a better performance in robustness. The result of $\boldsymbol{R}_Y$ illustrated in Fig. 7(c) has similar trends and amplitudes with the result of $\boldsymbol{R}_X$. The results of $\boldsymbol{t}_X$ illustrated in Fig. 7(b) are slightly

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

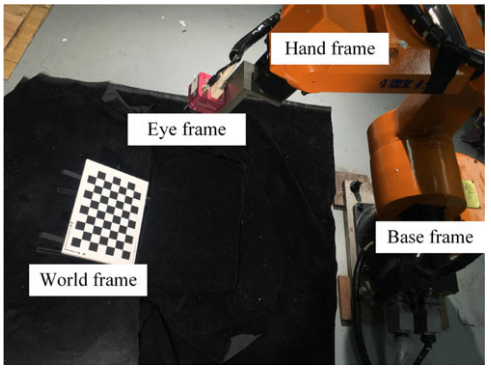

Figure 8. *Experimental setup.*

different from the result of $\boldsymbol{t}_X$ illustrated in Fig. 7(d) in amplitude of errors, which is still related to the different amplitudes of $\boldsymbol{t}_X$ and $\boldsymbol{t}_Y$ [43]. In addition, the slight perturbation of the rotation matrices will cause a larger error in the translation part according to the translation part of Eq. (2.3). Therefore, robot calibration may be an effective method to improve the accuracy of translation estimation. The above results show that iterative optimization can effectively reduce the influence of random noise.

3.2. Experiments with real data

The experimental setup corresponding to the robot-world and hand-eye calibration, as shown in Fig. 8, including an industrial robot (GSK RB03) with 6-DOF and a CMOS camera with a 8 mm lens installed at the end of the robot. The intrinsic parameters of the camera were obtained by the Camera Calibration Toolbox for MATLAB [51] and Zhang's method [52], which was achieved by a checkerboard with 7×10 calibration grid (each calibration grid is a 25 mm $\times$ 25 mm square).

The experiment process is carried out through the following steps:

(1) Camera calibration. Keep the robot fixed and move the checkerboard, and select the poses of the checkerboard in the field of view of the camera based on Zhang's method [52]. Enter the images into Camera Calibration Toolbox for MATLAB, and the intrinsic parameters of camera are listed in Table II.

(2) Measurement and calibration. Keep the checkerboard fixed and move the robot to obtain the n configurations for calibration, and the N ($N = 100$) configurations for verification. $\boldsymbol{A}_i$ ($i = 1, 2, \ldots, n$) were obtained from the robot control system. $\boldsymbol{B}_i$ ($i = 1, 2, \ldots, n$) were calculated by Zhang's method [52]. The verification data $\boldsymbol{A}_i$ ($i = 1, 2, \ldots, N$) and $\boldsymbol{B}_i$ ($i = 1, 2, \ldots, N$) were obtained in the same way.
In order to solve the robot-world and hand-eye calibration, at least three different poses are required [34]. Consistent with the simulated experiment, take n ($n = 4, 5, \ldots, 10$) datasets to form an overdetermined system for calibration. Run the algorithms and get the calibration results based on all methods.

(3) Verify the results. Since there is no ground truth in the real experiment, the calibration accuracy can be verified in an indirect way. Calibration accuracy was quantitatively assessed by comparing $\boldsymbol{A}_i$ ($i = 1, 2, \ldots, N$), using the following equation:

$$\hat{\boldsymbol{A}}_i = \boldsymbol{Y}\boldsymbol{B}_i\boldsymbol{X}^{-1}. \tag{26}$$

The $\hat{\boldsymbol{A}}_i$ and $\boldsymbol{A}_i$ ($i = 1, 2, \ldots, N$) were compared and the errors of all poses (from 4 to 10) in rotation and translation were computed.

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

Table II. *Intrinsic parameters of the camera*

Intrinsic	**Value (pixels)**
Image size	960×1680
Focal length	$[1.4935e+03, 1.4944e+03]$
Principal point	[445.6785,828.2813]
Mean reprojection error	0.1152

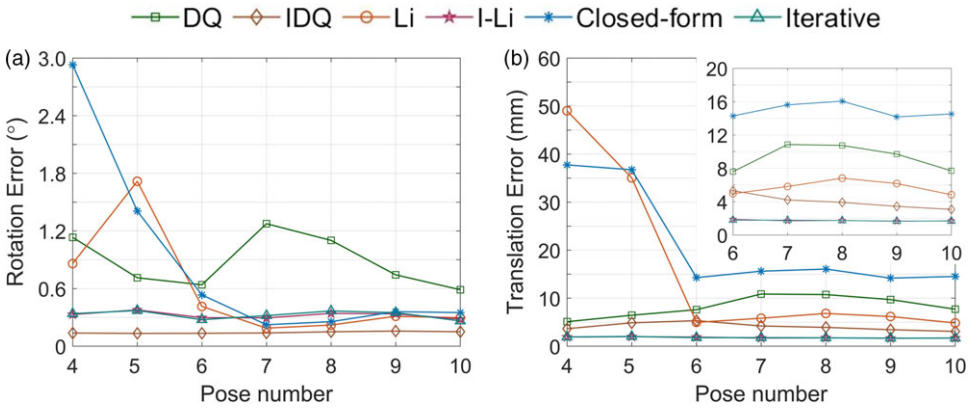

Figure 9. *Real experiment results: (a) rotation error; (b) translation error.*

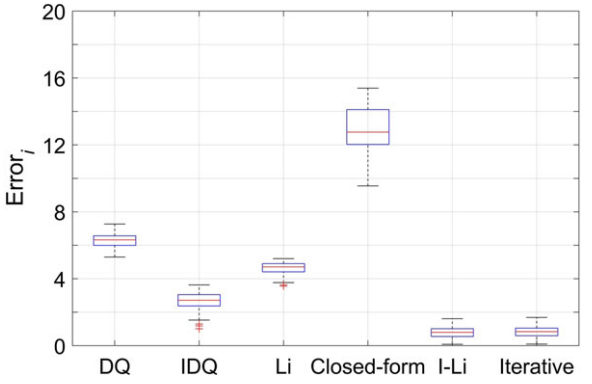

Figure 10. *Real experiment results: Error distributions.*

The results are shown in Fig. 9. It can be seen that the solutions of Li and closed-form as the initial value of the iterative solution have a faster convergence rate after optimization, which is consistent with the simulation results. In other words, the proposed method can effectively reduce the negative influence of random noise, especially when the measurement information is limited. IDQ method obtains an optimal rotation estimate. This may be due to avoiding the negative effect of translation noise, but the propagation error caused by rotation cannot be ignored. Especially, Fig. 9(b) shows that the translational accuracy estimated by iterative methods outperforms that of the linear methods, and this advantage may be due to the fact that optimizing robot-world and hand-eye calibration with rotation and translation coupling can effectively reduce the propagation error. The proposed method combines the best performance and minimal errors in both rotation and translation using 10 poses. The error is 0.2632° in rotation and 1.6887 mm in translation. The calibration error still exists because of the error in robot and vision measuring system.

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

Table III. *Comparison of calculational efficiency*

Method	Run time (s)
DQ	0.2582
IDQ	0.1987
Li	0.0574
I-Li	0.1033
Closed-form	0.0267
Iterative	0.0771

The calibration accuracy can also be estimated by a global perspective. According to Chasles's theorem [23], a rigid transformation can be modeled as a rotation with the same angle about an axis not through the origin and a translation along this axis. A screw represented by a vector specified by six independent parameters is exactly equal to the number of degrees of freedom of a rigid body in 3D Euclidean space, which is the most compact representation of a rigid body motion. Therefore, the screw is a better global error evaluation index. The error is denoted as

$$\text{Error}_i = \left\| Screw((\boldsymbol{A}_i\boldsymbol{X})^{-1}\boldsymbol{Y}\boldsymbol{B}_i) \right\|_2, \tag{27}$$

where $i = 1, 2, \ldots, 100$; $Screw(\bullet)$ means conversion from homogeneous matrix to screw, more detailed method in ref. [23]. The box plots (Fig. 10) show the error distributions. The iterative methods outperform the linear methods in terms of the median error. Meanwhile, the iterative method based on the Li method as the initial value is better, which may be due to the fact that a better initial value can converge to a better iteration result.

In this study, MATLAB R2019a was used to the program implementation of the above four methods, while the hardware used featured Intel(R) Core(TM) i5-10210U CPU @1.60GHz, 8G RAM. Ten poses were used in this test. Table III lists the running time values of the four methods and the number of iterations of the two iterative methods. The runtime data presented in Table III indicate that the iterative solution with the closed-form solution as the initial value is faster than the iterative solution with the reference solution as the initial value.

4. Conclusion

This paper proposes a method to solve rotation and translation simultaneously for robot-world and hand-eye calibration. A closed-form initial estimation is originally derived based on Kronecker product for the simultaneous method. Then, the Gauss–Newton algorithm in which the Jacobian matrix is constructed by Lie algebra is applied to the closed-form solution. Additionally, a method to the problem of singularity that can be widely applied to robot-world and hand-eye calibration is elaborated. A comparative analysis with the reference method shows that the proposed method outperforms the later ones in terms of accuracy and robustness, and the proposed method can significantly reduce the propagation error and obtain a higher-precision translation estimation.

CRediT authorship contribution statement. Xiao Wang: conceptualization, formal analysis, investigation, methodology, software, validation, roles/writing – original draft, writing – review and & editing. Hanwen Song: funding acquisition, project administration, resources, and supervision.

Conflicts of interests. The authors declare none.

Acknowledgements. This research was supported by National Natural Science Foundation of China (Grant No. 11872047).

References

[1] J. Huang, K. Zhou, W. Chen and H. Song, "A pre-processing method for digital image correlation on rotating structures," *Mech. Syst. Signal Proc.* **152**(12), 107494 (2021).

[2] X. Ding and F. Yang, "Study on hexapod robot manipulation using legs," *Robotica* **34**(2), 468–481 (2016).

[3] S. Peng, X. Ding, F. Yang and K. Xu, "Motion planning and implementation for the self-recovery of an overturned multi-legged robot," *Robotica* **35**(5), 1107–1120 (2017).

[4] J. Wu, Y. Sun, M. Wang and M. Liu, "Hand-Eye calibration: 4D procrustes analysis approach," *IEEE Trans. Instrum. Meas.* **69**(6), 2966–2981 (2020).

[5] K. Pachtrachai, F. Vasconcelos, G. Dwyer, S. Hailes and D. Stoyanov, "Hand-eye calibration with a remote centre of motion," *IEEE Rob. Autom. Lett.* **4**(4), 3121–3128 (2019).

[6] O. Özgüner, T. Shkurti, S. Huang, R. Hao, R. C. Jackson, W. S. Newman and M. C. Çavuşoğlu, "Camera-robot calibration for the Da Vinci robotic surgery system," *IEEE Trans. Autom. Sci. Eng.* **17**(4), 2154–2161 (2020).

[7] S. Kansal and S. Mukherjee, "Vision-Based kinematic analysis of the delta robot for object catching," *Robotica*, 1–21 (2021).

[8] G. Ma, Z. Jiang, H. Li, J. Gao, Z. Yu, X. Chen, Y.-H. Liu and Q. Huang, "Hand-eye servo and impedance control for manipulator arm to capture target satellite safely," *Robotica* **33**(4), 848–864 (2015).

[9] S. Ren, X. Yang, Y. Song, H. Qiao, L. Wu, J. Xu and K. Chen, "A Simultaneous Hand-Eye Calibration Method for Hybrid Eye-in-Hand/Eye-to-Hand System," 2017 *IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)* (2017) pp. 568–573.

[10] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren and M. Q.-H. Meng, "Simultaneous hand-eye, tool-flange, and robot-robot calibration for comanipulation by solving the AXB = YCZ problem," *IEEE Trans. Robot.* **32**(2), 413–428 (2016).

[11] Z. Fu, J. Pan, E. Spyrakos-Papastavridis, X. Chen and M. Li, "A dual quaternion-based approach for coordinate calibration of dual robots in collaborative motion," *IEEE Robot. Autom. Lett.* **5**(3), 4086–4093 (2020).

[12] G. Wang, W.-L. Li, C. Jiang, D.-H. Zhu, H. Xie, X.-J. Liu and H. Ding, "Simultaneous calibration of multicoordinates for a dual-robot system by solving the axb = ycz problem," *IEEE Trans. Rob.* **37**(4), 1172–1185 (2021).

[13] J. Su, "Convergence analysis for the uncalibrated robotic hand–eye coordination based on the unmodeled dynamics observer," *Robotica* **28**(4), 597–605 (2010).

[14] Y. C. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form AX = XB," *IEEE Trans. Robot. Autom.* **5**(1), 16–29 (1989).

[15] H. Zhuang, Z. S. Roth and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form AX = YB," *IEEE Trans. Robot. Autom.* **10**(4), 549–554 (1994).

[16] X. Wang, J. Huang and H. Song, "Simultaneous robot–world and hand–eye calibration based on a pair of dual equations," *Measurement.* **181**(3), 109623 (2021).

[17] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Rob. Autom.* **5**(3), 345–358 (1989).

[18] J. C. K. Chou and M. Kamel, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *Int. J. Robot. Res.* **10**(3), 240–254 (1991).

[19] R. Horaud and F. Dornaika, "Hand-eye calibration," *Int. J. Rob. Res.* **14**(3), 195–210 (1995).

[20] A. Malti and J. P. Barreto, "Robust hand-eye calibration for computer aided medical endoscopy," *IEEE International Conference on Robotics and Automation,* ICRA 2010, Anchorage(2010) pp. 5543–5549.

[21] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX = XB on the Euclidean group," *IEEE Trans. Rob. Autom.* **10**(5), 717–721 (1994).

[22] N. Andreff, R. Horaud and B. Espiau, "Robot hand-eye calibration using structure-from-motion," *Int. J. Rob. Res.* **20**(3), 228–248 (2001).

[23] H. Chen, "A Screw Motion Approach to Uniqueness Analysis of Head-Eye Geometry," *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA (IEEE Computer Society, 1991) pp. 145–151.

[24] Z. Zhao and Y. Liu, "A hand–eye calibration algorithm based on screw motions," *Robotica* **27**(2), 217–223 (2009).

[25] K. Daniilidis and E. Bayro-Corrochano, "The Dual Quaternion Approach to Hand-Eye Calibration," *Proceedings of 13th International Conference on Pattern Recognition*, vol. **1** (1996) pp. 318–322.

[26] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Rob. Res.* **18**(3), 286–298 (1998).

[27] Q. Ma, H. Li and G. S. Chirikjian, "New Probabilistic Approaches to the AX = XB Hand-Eye Calibration without Correspondence," 2016 IEEE International Conference on Robotics and Automation (ICRA) (2016) pp. 4365–4371.

[28] H. Zhuang and Y. C. Shiu, "A Noise Tolerant Algorithm For Wrist-mounted Robotic Sensor Calibration With Or Without Sensor Orientation Measurement," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems* (1992) pp. 1095–1100.

[29] K. H. Strobl and G. Hirzinger, "Optimal Hand-Eye Calibration," 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (2006) pp. 4647–4653.

[30] J. Heller, M. Havlena and T. Pajdla, "Globally optimal hand-eye calibration using branch-and-bound," *IEEE Trans.* Pattern Anal. *Mach. Intell.* **38**(5), 1027–1033 (2016).

[31] A. Malti, "Hand-eye calibration with epipolar constraints: Application to endoscopy," *Robot. Auton. Syst.* **61**(2), 161–169 (2013).

[32] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robot. Autom. Lett.* **4**(2), 1021–1028 (2019).

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

[33] G. Yang and L. Zhao, "Optimal Hand–Eye Calibration of IMU and Camera," 2017 Chinese Automation Congress (CAC) (2017) pp. 1023–1028.
[34] M. Shah, "Solving the robot-world/hand-eye calibration problem using the Kronecker product," *J. Mech. Rob.* **5**(3), 031007 (2013).
[35] I. Ali, S. Olli, A. Gotchev and E. R. Morales, "Methods for simultaneous robot-world-hand–eye calibration: A comparative study," *Sensors* **19**, 2837 (2019).
[36] A. Li, L. Wang and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product," *Int. J. Phys. Sci.* **5**(10), 1530–1536 (2010).
[37] H. Li, Q. Ma, T. Wang and G. S. Chirikjian, "Simultaneous hand-eye and robot-world calibration by solving the AX = YB problem without correspondence," *IEEE Robot. Autom. Lett.* **1**(1), 145–152 (2016).
[38] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Trans. Robot. Autom.* **14**(4), 617–622 (1998).
[39] A. Tabb and K. M. Ahmad Yousef, "Solving the robot-world hand-eye(s) calibration problem with iterative methods," *Mach. Vis. Appl.* **28**(5–6), 569–590 (2017).
[40] Z. Zhao and Y. Weng, "A flexible method combining camera calibration and hand–eye calibration," *Robotica* **31**(5), 747–756 (2013).
[41] J. Heller, D. Henrion and T. Pajdla, "Hand-Eye and Robot-World Calibration by Global Polynomial Optimization," 2014 IEEE International Conference on Robotics and Automation (ICRA) (2014) pp. 3157–3164.
[42] Z. Zhao, "Simultaneous robot-world and hand-eye calibration by the alternative linear programming," *Pattern Recognit. Lett.* **127**, 174–180 (2018).
[43] L. Wu and H. Ren, "Finding the kinematic base frame of a robot by hand-eye calibration using 3D position Data," *IEEE Trans. Autom. Sci. Eng.* **14**(1), 314–324 (2017).
[44] J. Brewer, "Kronecker products and matrix calculus in system theory," *IEEE Trans. Circ. Syst.* **25**(9), 772–781 (1978).
[45] R. Murray, Z. Li and S. Sastry, *A Mathematical Introduction to Robot Manipulation*, (1994).
[46] V. Gorbatsevich, A. L. Onishchik and E. B. Vinberg, *Structure of Lie Groups and Lie Algebras*, (1990).
[47] Z. Fu, J. Dai, Y. Kun, X. Chen and P. Lopez-Custodio, "Analysis of unified error model and simulated parameters calibration for robotic machining based on lie theory," *Rob. Comput. Integr. Manuf.* **61**, 101855 (2019).
[48] H. Song, Z. Du, W. Wang and L. Sun, "Singularity analysis for the existing closed-form solutions of the hand-eye calibration," *IEEE Access*. **6**, 75407–75421 (2018).
[49] J. Schmidt and H. Niemann, "Data-selection for hand-eye calibration a vector quantization approach," *Int. J. Rob. Res.* **27**(9), 1027–1053 (2008).
[50] J. Zhang, F. Shi and Y. Liu, "Adaptive motion selection for online hand-eye calibration," *Robotica* **25**(5), 529–536 (2007).
[51] J. Y. Bouguet, Camera Calibration Toolbox for MATLAB.
[52] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Trans.* Pattern Anal. *Mach. Intell.* **22**(11), 1330–1334 (2000).

A. Appendix

According to the left perturbation model of the rotation matrix (Eq. (16)), the Jacobian matrix represented by Eq. (14) is derived to Eq. (17). The partial derivative of $\boldsymbol{F}_1$ with respect to $\boldsymbol{\omega}_X$ can be represented as

$$\begin{bmatrix} \dfrac{\partial(\boldsymbol{F}_1)_1}{\partial \boldsymbol{\omega}_X} \\ \dfrac{\partial(\boldsymbol{F}_1)_2}{\partial \boldsymbol{\omega}_X} \\ \dfrac{\partial(\boldsymbol{F}_1)_3}{\partial \boldsymbol{\omega}_X} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{R}_A[(\boldsymbol{R}_X)_1]^\wedge \\ -\boldsymbol{R}_A[(\boldsymbol{R}_X)_2]^\wedge \\ -\boldsymbol{R}_A[(\boldsymbol{R}_X)_3]^\wedge \end{bmatrix}. \tag{A1}$$

Similarly, the partial derivative of $\boldsymbol{F}_1$ with respect to $\boldsymbol{\omega}_Y$ can be obtained as

$$\begin{bmatrix} \dfrac{\partial(\boldsymbol{F}_1)_1}{\partial \boldsymbol{\omega}_Y} \\ \dfrac{\partial(\boldsymbol{F}_1)_2}{\partial \boldsymbol{\omega}_Y} \\ \dfrac{\partial(\boldsymbol{F}_1)_3}{\partial \boldsymbol{\omega}_Y} \end{bmatrix} = \begin{bmatrix} [(\boldsymbol{R}_Y\boldsymbol{R}_B)_1]^\wedge \\ [(\boldsymbol{R}_Y\boldsymbol{R}_B)_2]^\wedge \\ [(\boldsymbol{R}_Y\boldsymbol{R}_B)_3]^\wedge \end{bmatrix}. \tag{A2}$$

https://doi.org/10.1017/S0263574721002034 Published online by Cambridge University Press

$\partial \boldsymbol{F}_2/\partial \boldsymbol{\omega}_Y$ can be calculated easily based on the left perturbation model as

$$\begin{aligned}\frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{\omega}_Y} &= \frac{\partial(\boldsymbol{R}_Y \boldsymbol{t}_B)}{\partial \boldsymbol{\omega}_Y} \\ &= \lim_{\partial \boldsymbol{\omega}_Y \to 0} \frac{\partial \boldsymbol{R}_Y \boldsymbol{R}_Y \boldsymbol{t}_B - \boldsymbol{R}_Y \boldsymbol{t}_B}{\boldsymbol{\omega}_Y} \\ &= \lim_{\partial \boldsymbol{\omega}_Y \to 0} \frac{[\partial \boldsymbol{\omega}_Y]^{\wedge} \boldsymbol{R}_Y \boldsymbol{t}_B}{\boldsymbol{\omega}_Y} \\ &= \lim_{\partial \boldsymbol{\omega}_Y \to 0} \frac{-[\partial \boldsymbol{R}_Y \boldsymbol{t}_B]^{\wedge} \boldsymbol{\omega}_Y}{\boldsymbol{\omega}_Y} \\ &= \lim_{\partial \boldsymbol{\omega}_Y \to 0} -[\partial \boldsymbol{R}_Y \boldsymbol{t}_B]^{\wedge}.\end{aligned} \tag{A3}$$

Meanwhile, $\partial \boldsymbol{F}_2/\partial \boldsymbol{t}_X$ and $\partial \boldsymbol{F}_2/\partial \boldsymbol{t}_Y$ can be calculated as

$$\begin{aligned}\frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{t}_X} &= -\boldsymbol{R}_A; \\ \frac{\partial \boldsymbol{F}_2}{\partial \boldsymbol{t}_Y} &= \boldsymbol{I}_3.\end{aligned} \tag{A4}$$

Cite this article: X. Wang and H. Song (2022). "Optimal robot-world and hand-eye calibration with rotation and translation coupling", Robotica **40**, 2953–2968. https://doi.org/10.1017/S0263574721002034