Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced
Robotic Systems
November-December 2020: 1–13
© The Author(s) 2020
DOI: 10.1177/1729881420977296
journals.sagepub.com/home/arx
SAGE

Hand-eye calibration method with a three-dimensional-vision sensor considering the rotation parameters of the robot pose

Jinsheng Fu[1], Yabin Ding[1], Tian Huang[1,2] and Xianping Liu[2]

Abstract

Hand-eye calibration is a fundamental step for a robot equipped with vision systems. However, this problem usually interacts with robot calibration because robot geometric parameters are not very precise. In this article, a new calibration method considering the rotation parameters of the robot pose is proposed. First, a constraint least square model is established assuming that each spherical center measurement of standard ball is equal in the robot base frame, which provides an initial solution. To further improve the solution accuracy, a nonlinear calibration model in the sensor frame is established. Since it can reduce one error accumulation process, a more accurate reference point can be used for optimization. Then, the rotation parameters of the robot pose whose slight errors cause large disturbance to the solution are selected by analyzing the coefficient matrices of the error items. Finally, the hand-eye transformation parameters are refined together with the rotation parameters in the nonlinear optimization solution. Some comparative simulations are performed between the modified least square method, constrained least square method, and the proposed method. The experiments are conducted on a 5-axis hybrid robot named TriMule to demonstrate the superior accuracy of the proposed method.

Keywords

Machine vision, hand-eye calibration, 3D-vision sensor, machining robot

Date received: 20 July 2020; accepted: 9 November 2020

Topic Area: Vision Systems
Associate Editor: Shengyong Chen
Topic Editor: Antonio Fernandez-Caballero

Introduction

The use of industrial robots is widespread in diverse machining operations, such as drilling, milling, and grinding. Compared with machine tools, machining robots have many advantages of large workspace, high degree of freedom (DOF), cost-effectiveness, and high flexibility. Hence, there have been attempts to use industrial robots for machining processes instead of machine tools. Recently, machining robots are expected to perform more complex tasks, such as low-volume automated production in unstructured environments. It requires the machining robots not only to have good machining performance but should be equipped with relevant measuring system to assist the robots to locate the machining position of workpiece.[1–4] The commonly used measuring system in industry

[1] Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, Tianjin, China
[2] School of Engineering, University of Warwick, Coventry, UK

Corresponding author:
Yabin Ding, Key Laboratory of Mechanism Theory and Equipment Design of Ministry of Education, Tianjin University, No.135 Yaguan Road, Haihe Education Park, Tianjin, 300350, China.
Email: ybding@tju.edu.cn

can be divided into contact measurement and noncontact measurement. Due to the complex and irregular surface profile of the workpiece, the contact measuring system is difficult to meet the needs of industry. The visual measuring system, as one of the widely applied noncontact measuring systems, is characterized by low cost, compact structure, easy integration, and installation and is capable of realizing multiple fields of view measurement when it is integrated into the robotic system and forms robotic visual measuring system. And the problems such as occlusion, shadow, and inefficient measurement data can be overcome.[5] There are many visual measuring systems, such as monocular vision, stereo vision, structured light, and fringe projection. Fringe projection measuring system has been widely used in surface and mold measurement, rapid prototyping, reverse engineering, industrial detection, and other fields due to its characteristics of fast measurement speed, high precision, strong system reliability, and large amount of acquired point cloud. To obtain three-dimensional (3D) point clouds of the workpiece accurately and fast to determine the machining information, such as the position of the workpiece relative to the robot end effector and the machining allowance, a 3D-vision sensor based on fringe projection is adopted into the machining robot.

Two kinds of calibration need to be performed before the robotic visual measuring system is used. One is visual system calibration[6–8] that is used to determine the internal parameters of the camera (called camera calibration) and external parameters between two cameras (called stereo calibration) and describes the mapping relationship between 3D space coordinates and two-dimensional (2D) image coordinates. Usually, the visual measuring system has been calibrated in advance by producer. The other is hand-eye calibration, which is used to determine the transformation between vision sensor and robot end effector. In this article, hand-eye calibration aims to establish cooperation between hand and eye to assist machining robotic positioning. A typical process of measurement-assisted robotic machining is shown in Figure 1. In the measurement process (No. 1), a 3D-vision sensor mounted onto the end effector of a machining robot measures the workpiece to obtain the machining information in the vision sensor frame. Through hand-eye transformation (No. 2), the machining information under the vision sensor frame is transformed into the robot end-effector frame. Finally, according to the current position of the robot end effector and the machining information, the robot end effector is driven to the machining position to complete machining (No. 3). It can be seen that hand-eye calibration is a key step that accurately determine the position and pose relationship between vision sensor and robot end effector to ensure the machining accuracy of the robot.

Essentially, hand-eye calibration is to solve the homogeneous transformation matrix between the vision sensor frame and the robot end-effector frame, also known as the hand-eye transformation matrix. There are many typical algorithms for solving the matrix, including screw theory,[9] quaternions,[10] dual quaternions,[11] Kronecker product,[12,13] and so on. According to the calculating process, hand-eye calibration methods can be divided into three categories. The first belongs to separable closed-form solutions,[14–16] which is to solve rotation and translation separately. The rotation can be solved firstly by linear equations and then the translation is solved with the solved rotation. This kind of method is simple in derivation and convenient in solution, but the error of estimated rotation will propagate into the solution of the translation. It is sensitive to the noise in the collected data. The second belongs to simultaneous solutions,[17–19] which is to solve rotation and translation simultaneously. Obviously, there is no error accumulation problem in this kind of method, but the rotation estimated cannot guarantee the orthogonality,[20] such as Kronecker product method. The rotation can be further optimized to satisfy the orthogonality, but the translation cannot meet the accuracy requirement. The third belongs to iterative solutions,[21–23] which is to solve rotation and translation simultaneously by optimizing iteratively the objective function established. This kind of method usually has high accurate solution but is complex and time-consuming. In addition, the accuracy of solution depends on the initial value for iteration. Choosing wisely, linear method is generally adopted for estimating initial value and followed by iterative optimization for accurate solution.

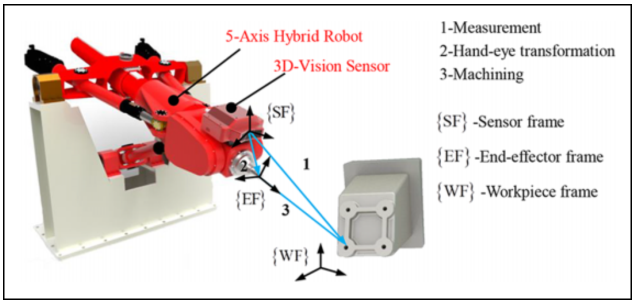

Figure 1. Typical process of measurement-assisted robotic machining.

In summary, various hand-eye calibration methods have been developed and tremendous progress has been made in solving hand-eye transformation matrix. However, most researchers only focus on the solution of the hand-eye matrix by different mathematical tools or how to avoid error accumulation during the solution. The effects of the robot pose error on the solution accuracy are not fully considered in their methods which are applied only in the case of low accuracy requirement. Some researchers have proposed to calibrate the hand-eye and the robot simultaneously,[24,25] which leads to complicated derivation and may arise linear correlation problems in parameter estimation. Especially for parallel or hybrid robots, the error modeling is actually a complicated problem.[26,27]

This article aims to guide the positioning of machining robots using a 3D-vision sensor, which is integrated at the end effector of machining robots. An accurate hand-eye calibration method only considering the rotation parameters of the robot pose is proposed. The hand-eye calibration model is simple and easy to establish. In addition, the linear correlation problems are avoided through the parameter analysis. The main contributions of our work are summarized as follows:

- A constrained least square solution is introduced to provide good initial values for the next step of optimization.
- A nonlinear optimization solution that further improve the solution accuracy is presented simultaneously to optimize the rotation parameters that have a significant impact on the solution accuracy.
- Simulations and experiments are conducted to compare the three different methods and demonstrate the superiority of the proposed method (PM).

The article is structured as follow. The second section introduces robotic visual measuring system, coordinate system and coordinate transformation, and conventional calibration method. In the third section, the PM of hand-eye calibration is presented and parameter analysis is also performed to discuss the effects of different parameters on the solution accuracy. Some simulation and experimental results are detailed in the fourth section, and the article ends with concluding remarks in the fifth section.

Background knowledge

Robotic visual measuring system

As shown in Figure 1, robotic visual measuring system is mainly composed of a 5-axis hybrid robot and a 3D-vision sensor. The 3D-vision sensor is mounted onto the A/C type wrist of the 5-axis hybrid robot, which can be considered as being fixed in spatial position with the end effector of the robot. In the working process of the system, the 5-axis hybrid robot cooperates with the 3D-vision sensor. After the robot moves to the designated position, the sensor performs the measurement function. After the measurement, the robot moves to the next designated position, and the sensor continues the measurement, and so on, until the whole measurement task is completed. Finally, the measured data of multiple robot poses are unified in the same coordinate frame, which generally is the workpiece coordinate frame, and the complex surface profile is reconstructed or the end effector of the robot is adjusted to the specified position for machining according to the measurement information. TCP/IP protocol is adopted to construct data communication between the robot and the vision sensor, which can implement real-time control.

The 3D-vision sensor consists of three sensors, which are two industrial charge coupled device (CCD) cameras and a digital light processing projector, as shown in Figure 2. The measurement principle is based on fringe projection profilometry. First, multifrequency fringes are projected onto the object, which is deformed by the depth change of the object and modulates the depth information into the fringes. At the same time, two CCD cameras on the left and right capture the deformed fringe images and transmit the images to data processing system. And then through phase analysis and phase unwrapping, the same imaging points of object in the left and right camera planes can be matched. Finally, 3D information of the object is recovered according to the system configuration parameters.

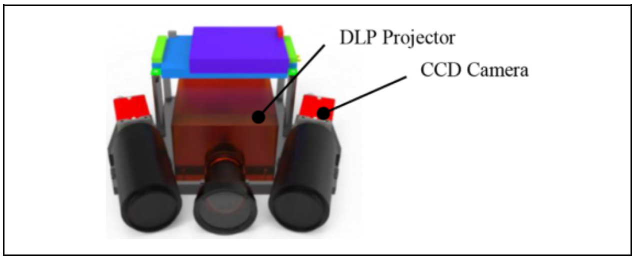

Figure 2. Schematic diagram of 3D-vision sensor. 3D: three dimensional.

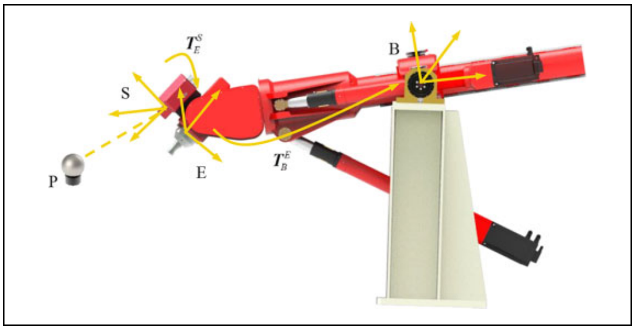

Figure 3. Schematic diagram of hand-eye calibration.

Coordinate system and coordinate transformation

As shown in Figure 3, hand-eye calibration mainly involves three coordinate frames: 3D-vision sensor frame (called sensor frame for short [SF]), robot end-effector frame (called end-effector frame for short [EF]), and robot base frame (called base frame for short [BF]). The origins of these coordinate frames are S, E, and B, respectively.$\boldsymbol{T}_E^S$ is hand-eye transformation matrix from the sensor frame to the end-effector frame and is an unknown quantity to be solved.$\boldsymbol{T}_B^E$ is the homogeneous transformation matrix from the end-effector frame to the base frame and is a known quantity. For any homogeneous transformation matrix $\boldsymbol{T}$, it includes a rotation matrix $\boldsymbol{R}$ and a translation vector $\boldsymbol{t}$, and it can be expressed in the form of homogeneous matrix as follows

$$\boldsymbol{T}=\begin{bmatrix}\boldsymbol{R} & \boldsymbol{t}\\ \boldsymbol{0}^{\mathrm{T}} & 1\end{bmatrix} \tag{1}$$

where the rotation matrix $\boldsymbol{R}$ is an orthogonal matrix with nine elements, consisting of three orthogonal column vectors. It can be expressed as

$$\boldsymbol{R}=[\boldsymbol{r}_1 \quad \boldsymbol{r}_2 \quad \boldsymbol{r}_3]=\begin{bmatrix} r_{11} & r_{12} & r_{13}\\ r_{21} & r_{22} & r_{23}\\ r_{31} & r_{32} & r_{33}\end{bmatrix} \tag{2}$$

A standard ball is placed in the robot workspace for calibration, the position of which is guaranteed to be fixed. P is the spherical center.

After the robot moves to a certain position, the sensor scans the standard ball and obtains the coordinate of spherical center. The relationship that the coordinate of the spherical center is transformed from the sensor frame $\{\mathrm{SF}\}$ to the base frame $\{\mathrm{BF}\}$ is established

$$\boldsymbol{p}^B=\boldsymbol{T}_B^E\boldsymbol{T}_E^S\boldsymbol{p}^S \tag{3}$$

where $\boldsymbol{p}^B$ represents the homogeneous coordinate of the spherical center P in the base frame, $\boldsymbol{p}^S$ represents the homogeneous coordinate of the spherical center P in the sensor frame.

Conventional calibration method

When the robot moves to different positions and the sensor scans the standard sphere simultaneous, we can obtain the following equations[24]

$$\begin{cases}\boldsymbol{p}^B={}_1\boldsymbol{T}_B^E\boldsymbol{T}_E^S\boldsymbol{p}_1^S\\ \boldsymbol{p}^B={}_2\boldsymbol{T}_B^E\boldsymbol{T}_E^S\boldsymbol{p}_2^S\\ \quad\vdots\\ \boldsymbol{p}^B={}_n\boldsymbol{T}_B^E\boldsymbol{T}_E^S\boldsymbol{p}_n^S\end{cases} \tag{4}$$

Since the position of the standard ball relative to the robot base remains invariable, without loss of generality, the first measurement of the spherical center in the base frame is regarded as the reference point, and the following constraint equations can be established

$${}_1\boldsymbol{T}_B^E\boldsymbol{T}_E^S\boldsymbol{p}_1^S={}_i\boldsymbol{T}_B^E\mathrm{T}_E^S\boldsymbol{p}_i^S \; i=2,3...n \tag{5}$$

By handling equation (5), we can obtain

$$\boldsymbol{A}_i\boldsymbol{x}=\boldsymbol{b}_i \tag{6}$$

where $\boldsymbol{A}_i=[\boldsymbol{p}_{s1}^{\mathrm{T}}\otimes\boldsymbol{R}_1-\boldsymbol{p}_{si}^{\mathrm{T}}\otimes\boldsymbol{R}_i \quad \boldsymbol{R}_1-\boldsymbol{R}_i]\; i=2,3\dots n$, $\boldsymbol{x}=\begin{pmatrix}\mathrm{vec}(\boldsymbol{R}_h)\\ \boldsymbol{t}_h\end{pmatrix}$, $\boldsymbol{b}_i=\boldsymbol{t}_i-\boldsymbol{t}_1$, $\boldsymbol{p}_{s1}$, and $\boldsymbol{p}_{si}$ are the corresponding nonhomogeneous coordinates of $\boldsymbol{p}_1^S$ and $\boldsymbol{p}_i^S$, respectively. $\boldsymbol{R}_1$ and $\boldsymbol{R}_i$ are the corresponding rotation matrices of ${}_1\boldsymbol{T}_B^E$ and ${}_i\boldsymbol{T}_B^E$, respectively. $\boldsymbol{t}_1$ and $\boldsymbol{t}_i$ are the corresponding translation vectors of ${}_1\boldsymbol{T}_B^E$ and ${}_i\boldsymbol{T}_B^E$, respectively.$\boldsymbol{R}_h$ and $\boldsymbol{t}_h$ are, respectively, the rotation matrix and translation vector of the hand-eye transformation matrix $\boldsymbol{T}_E^S$.vec() represents vectorization operation. And $\mathrm{vec}(\boldsymbol{R}_h)=(\boldsymbol{r}_{h1}^{\mathrm{T}} \quad \boldsymbol{r}_{h2}^{\mathrm{T}} \quad \boldsymbol{r}_{h3}^{\mathrm{T}})^{\mathrm{T}}$,$\boldsymbol{R}_h=[\boldsymbol{r}_{h1} \quad \boldsymbol{r}_{h2} \quad \boldsymbol{r}_{h3}]$.$\otimes$ denotes Kronecker operation. Superscript T is transpose symbol. Assuming $\boldsymbol{C}1$ is a $\mathrm{n}_1\times\mathrm{n}_2$ matrix, $\boldsymbol{C}2$ is a $\mathrm{n}_3\times\mathrm{n}_4$ matrix, then $\boldsymbol{C}1\otimes\boldsymbol{C}2$ is a $\mathrm{n}_1\mathrm{n}_3\times\mathrm{n}_2\mathrm{n}_4$ matrix, and the following equation can be obtained

$$\boldsymbol{C}1\otimes\boldsymbol{C}2=\begin{bmatrix} c_{11}\boldsymbol{C}2 & \cdots & c_{1\mathrm{n}_2}\boldsymbol{C}2\\ \vdots & \ddots & \vdots\\ c_{\mathrm{n}_1 1}\boldsymbol{C}2 & \cdots & c_{\mathrm{n}_1\mathrm{n}_2}\boldsymbol{C}2\end{bmatrix} \tag{7}$$

According to equation (4), multiple equations similar to equation (6) can be obtained. The least square solution of hand-eye transformation parameters obtained is as follows

$$\tilde{\boldsymbol{x}}_{\mathrm{LS}}=\left(\sum_{i=2}^{n}\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{A}_i\right)^{-1}\left(\sum_{i=2}^{n}\boldsymbol{A}_i^{\mathrm{T}}\boldsymbol{b}_i\right) \tag{8}$$

Since no constraint is imposed on the rotation of the hand-eye transformation matrix for solution, the rotation estimated is not orthogonal. The direct method is to rearrange the rotation $\tilde{\boldsymbol{x}}_{\mathrm{LS}}(1:9)$ of the least square solution $\tilde{\boldsymbol{x}}_{\mathrm{LS}}$ into the form of a rotation matrix

$$\tilde{\boldsymbol{R}}_{\mathrm{LS}}=\mathrm{matrix}\left(\tilde{\boldsymbol{x}}_{\mathrm{LS}}(1:9),3,3\right) \tag{9}$$

where $\mathrm{matrix}\left(\boldsymbol{x}(1:9),3,3\right)$ is matrixing operation and is inverse operation of $\mathrm{vec}(\boldsymbol{R}_h)$. Assuming $\boldsymbol{c}$ is a $\mathrm{m}_1\mathrm{m}_2$ vector, then $\mathrm{matrix}(\boldsymbol{c},\mathrm{m}_1,\mathrm{m}_2)$ means to rearrange $\boldsymbol{c}$ as a $\mathrm{m}_1\times\mathrm{m}_2$ matrix. $\tilde{\boldsymbol{x}}_{\mathrm{LS}}(1:9)$ represents the first nine elements in the vector $\tilde{\boldsymbol{x}}_{\mathrm{LS}}$. The singular value decomposition of the rotation matrix obtained $\tilde{\boldsymbol{R}}_{LS}$ is computed in the form

$$\tilde{\boldsymbol{R}}_{\mathrm{LS}}=\boldsymbol{U\Lambda V}^{\mathrm{T}} \tag{10}$$

The $\tilde{\boldsymbol{R}}_{\mathrm{LS}}$ is corrected as follows

$$\tilde{\boldsymbol{R}}_{\mathrm{MLS}}=\boldsymbol{UV}^{\mathrm{T}} \tag{11}$$

The results with corrected rotation matrix are referred as modified least square solution $\tilde{\boldsymbol{x}}_{\mathrm{MLS}}$ as follows:

$$\tilde{\boldsymbol{x}}_{\mathrm{MLS}}=\begin{pmatrix}\mathrm{vec}(\tilde{\boldsymbol{R}}_{\mathrm{MLS}})\\ \tilde{\boldsymbol{x}}_{\mathrm{LS}}(10:12)\end{pmatrix} \tag{12}$$

Proposed method

Constraint least square solution

The translation $\tilde{\boldsymbol{x}}_{\mathrm{LS}}(10:12)$ of the modified least square solution is not corrected synchronously with the rotation matrix $\tilde{\boldsymbol{R}}_{LS}$. And $\tilde{\boldsymbol{x}}_{MLS}$ is no longer optimal in the sense of least squares. Considering the orthogonality of the rotation matrix, a constraint least square model is established

$$\min\sum_{i=2}^{n}||\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}_i||^2$$
$$s.t.\begin{cases}\boldsymbol{r}_{h1}^{\mathrm{T}}\cdot\boldsymbol{r}_{h1}-1=0\\ \boldsymbol{r}_{h2}^{\mathrm{T}}\cdot\boldsymbol{r}_{h2}-1=0\\ \mathrm{r}_{h3}^{\mathrm{T}}\cdot\boldsymbol{r}_{h3}-1=0\\ \boldsymbol{r}_{h1}^{\mathrm{T}}\cdot\boldsymbol{r}_{h2}=0\\ \boldsymbol{r}_{h2}^{\mathrm{T}}\cdot\boldsymbol{r}_{h3}=0\\ \boldsymbol{r}_{h1}^{\mathrm{T}}\cdot\boldsymbol{r}_{h3}=0\end{cases}\tag{13}$$

where |||| denotes the Frobenius norm. This model is solved by introducing Lagrange multipliers[28] and the constraint least square solution is given as

$$(\tilde{\boldsymbol{x}}_{\mathrm{CLS}},\tilde{\lambda}_j)=\arg\min_{x,\lambda_j}\sum_{i=2}^{n}||\boldsymbol{A}_i\boldsymbol{x}-\boldsymbol{b}_i||^2+\sum_{j=1}^{3}\lambda_j(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}_j\boldsymbol{x}-1)+\sum_{j=4}^{6}\lambda_j(\boldsymbol{x}^{\mathrm{T}}\boldsymbol{G}_j\boldsymbol{x})\tag{14}$$

where λ_i is a Lagrange multiplier. G_i is a coefficient matrix corresponding to the constraint in equation (13). The obtained rotation in equation (14) is considered as being orthogonal.

Nonlinear solution considering the rotation parameters of the robot pose

According to equation (5), the accuracy of the reference point in the base frame is affected by both the measurement error of the sensor and the robot pose error. To obtain a more accurate reference point, the first measurement of the spherical center in the sensor frame can be referred as the reference point. As the accuracy of the reference point is only affected by the measurement error and one error accumulation process is reduced, the accuracy of the reference point in the sensor frame is higher than that in the base frame, which provides the possibility to make a better estimation for the hand-eye transformation parameters. However, the measurement by the sensor transformed from the sensor frame under other robot poses to the sensor frame where the reference point is located suffers from more error propagation and error accumulation. Therefore, additional parameters are introduced to reduce the effects of the errors on the solution accuracy. In addition, excessive additional parameters will lead to linear correlation between the parameters and reduce the accuracy of the solution. It is necessary to analyze the effects of all the candidate parameters on the solution accuracy. Through the parameter analysis in the "Parameter analysis" section, we observed that slight errors of the rotation parameters can cause large amount of error. Hence, the rotation parameters of the robot pose are regarded as the additional parameters and are optimized in a nonlinear calibration model together with the hand-eye transformation parameters. Based on the reference point in the sensor frame, the nonlinear calibration model is established as follows

$$\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\boldsymbol{R}_h,\boldsymbol{t}_h)=\boldsymbol{p}_{s1i}-\boldsymbol{p}_{s1},i=2,...,n\tag{15}$$

$$\boldsymbol{p}_{s1i}=(\boldsymbol{R}_1\boldsymbol{R}_h)^{\mathrm{T}}\boldsymbol{R}_i\boldsymbol{R}_h\boldsymbol{p}_{si}+(\boldsymbol{R}_h^{\mathrm{T}}\boldsymbol{R}_1^{\mathrm{T}}\boldsymbol{R}_i-\boldsymbol{R}_h^{\mathrm{T}})\boldsymbol{t}_h+(\boldsymbol{R}_1\boldsymbol{R}_h)^{\mathrm{T}}(\boldsymbol{t}_i-\boldsymbol{t}_1)\tag{16}$$

where $\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\boldsymbol{R}_h,\boldsymbol{t}_h)$ means the ith measurement error in the sensor frame when the first measurement of the spherical center is as the reference point. $\boldsymbol{p}_{s1i}$ represents the coordinates of the spherical center transformed from the sensor frame with the ith robot pose to the sensor frame with the first robot pose. In addition, to ensure the orthogonality of the rotation matrix calculated in the optimization process, the rotation matrices $\boldsymbol{R}_i,\boldsymbol{R}_h,\boldsymbol{R}_1$ are parameterized by 3D vectors $\boldsymbol{\omega}_i,\boldsymbol{\omega}_h,\boldsymbol{\omega}_1$, respectively. The parameters of the nonlinear model include $\boldsymbol{R}_i,\boldsymbol{R}_h,\boldsymbol{R}_1,\boldsymbol{t}_h$. Estimating the parameters of the nonlinear model in equation (15) is equivalent to minimizing the following objective function

$$\sum_{i=2}^{n}\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\boldsymbol{R}_h,\boldsymbol{t}_h)^{\mathrm{T}}\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\boldsymbol{R}_h,\boldsymbol{t}_h)\tag{17}$$

This is a nonlinear optimization problem, which is solved using Levenberg-Marquardt (LM) algorithm.[29] Nonlinear optimization requires a good initial value to ensure the accuracy of the solution. Here, the constraint least square solution in equation (14) is used as the initial values. We can obtain

$$(\tilde{\boldsymbol{R}}_i,\tilde{\boldsymbol{R}}_1,\tilde{\boldsymbol{R}}_{h,\mathrm{PM}},\tilde{\boldsymbol{t}}_{h,\mathrm{PM}})=\underset{\boldsymbol{R}_i,\boldsymbol{R}_1,\boldsymbol{R}_h,\boldsymbol{t}_h}{\mathrm{argmin}}\sum_{i=2}^{n}\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\tilde{\boldsymbol{R}}_{h,\mathrm{CLS}},\tilde{\boldsymbol{t}}_{h,\mathrm{CLS}})^{\mathrm{T}}\boldsymbol{f}_i(\boldsymbol{R}_i,\boldsymbol{R}_1,\tilde{\boldsymbol{R}}_{h,\mathrm{CLS}},\tilde{\boldsymbol{t}}_{h,\mathrm{CLS}})\tag{18}$$

where $\tilde{\boldsymbol{R}}_{h,\mathrm{CLS}},\tilde{\boldsymbol{t}}_{h,\mathrm{CLS}}$ represents the hand-eye transformation parameters solved by equation (14). The central step of LM algorithm is the solution of the normal equations that is given as follows

$$(\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}+\beta\boldsymbol{E}_{3n+6})\Delta\boldsymbol{u}=-\boldsymbol{J}^{\mathrm{T}}\boldsymbol{f}\tag{19}$$

where

$$\boldsymbol{J}=[\boldsymbol{J}_2^{\mathrm{T}}\quad\boldsymbol{J}_3^{\mathrm{T}}\quad\cdots\quad\boldsymbol{J}_n^{\mathrm{T}}]^{\mathrm{T}},$$
$$\boldsymbol{J}_i=\left[\frac{\partial\boldsymbol{f}_i}{\partial\boldsymbol{\omega}_2}\quad\cdots\quad\frac{\partial\boldsymbol{f}_i}{\partial\boldsymbol{\omega}_n}\quad\frac{\partial\boldsymbol{f}_i}{\partial\boldsymbol{\omega}_1}\quad\frac{\partial\boldsymbol{f}_i}{\partial\boldsymbol{\omega}_h}\quad\frac{\partial\boldsymbol{f}_i}{\partial\boldsymbol{t}_h}\right],i=2,..,n\tag{20}$$

$$\boldsymbol{f}=[\boldsymbol{f}_2^{\mathrm{T}}\quad\boldsymbol{f}_3^{\mathrm{T}}\quad\cdots\quad\boldsymbol{f}_n^{\mathrm{T}}]^{\mathrm{T}}\tag{21}$$

$$\Delta\boldsymbol{u}=[\,\Delta\boldsymbol{\omega}_2^{\mathrm{T}}\quad\cdots\quad\Delta\boldsymbol{\omega}_n^{\mathrm{T}}\quad\Delta\boldsymbol{\omega}_1^{\mathrm{T}}\quad\Delta\boldsymbol{\omega}_h^{\mathrm{T}}\quad\Delta\boldsymbol{t}_h^{\mathrm{T}}\,]^{\mathrm{T}}\tag{22}$$

$\boldsymbol{J}$ is the Jacobian matrix of$\boldsymbol{f}$. The parameter β varies from iteration to iteration.[29] $\boldsymbol{E}_{3n+6}$ is the $(3n+6)\times(3n+6)$

identity matrix. The iterative value $\Delta \boldsymbol{u}$ can be obtained by solving the normal equations.

$$\Delta \boldsymbol{u} = -\left(\boldsymbol{J}^T\boldsymbol{J} + \beta \boldsymbol{E}_{3n+6}\right)^{-1}\boldsymbol{J}^T\boldsymbol{f} \tag{23}$$

The obtained $\Delta \boldsymbol{u}$ is subsequently used to update $\boldsymbol{R}_i, \boldsymbol{R}_1, \boldsymbol{R}_h, \boldsymbol{t}_h$ by

$$\begin{cases} \boldsymbol{R}_{i,\text{new}} = \boldsymbol{R}_{i,\text{old}}\exp([\Delta \boldsymbol{\omega}_i]^{\wedge}), i = 1, 2, ..., n, h \\ \boldsymbol{t}_{h,\text{new}} = \boldsymbol{t}_{h,\text{old}} + \Delta \boldsymbol{t}_h \end{cases} \tag{24}$$

The iterative process does not terminate until the variation of $||\boldsymbol{f}||$ falls below a set threshold, which indicates that the parameters converge to the optimal value.

The PM is concluded in Algorithm 1:

Algorithm 1. Nonlinear optimization method considering the rotation parameters

```
input: measurement data $\boldsymbol{p}_s$ , robot pose data $\boldsymbol{T}_B^E$
output: hand-eye transformation parameters $\boldsymbol{R}_h, \boldsymbol{t}_h$ and the rotation parameters $\boldsymbol{R}_i, \boldsymbol{R}_1$
initialization: initial value $\boldsymbol{R}_h, \boldsymbol{t}_h$ ←constraint least square solution in Eq. (14);
        set $\beta$=0.0001,$\varsigma$=0.00001,$\|\boldsymbol{f}\| \leftarrow +\infty$ ; $\|\boldsymbol{f}_0\|$ ←compute the norm of Eq. (21);
while $\big|\|\boldsymbol{f}\| - \|\boldsymbol{f}_0\|\big| > \varsigma$ do
    while $\|\boldsymbol{f}\| > \|\boldsymbol{f}_0\|$ do
        $\boldsymbol{f}_i \leftarrow \boldsymbol{p}_{s1i} - \boldsymbol{p}_{s1}$
        $\boldsymbol{f} \leftarrow \left[\boldsymbol{f}_2^{\mathrm{T}} \quad \boldsymbol{f}_3^{\mathrm{T}} \quad \cdots \quad \boldsymbol{f}_{\mathrm{n}}^{\mathrm{T}}\right]^{\mathrm{T}}$;
        $\boldsymbol{J}_i \leftarrow \left[\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_2} \quad \cdots \quad \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_{\mathrm{n}}} \quad \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1} \quad \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_h} \quad \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_h}\right], i = 2,..,\mathrm{n}$;
        $\boldsymbol{J} \leftarrow \left[\boldsymbol{J}_2^{\mathrm{T}} \quad \boldsymbol{J}_3^{\mathrm{T}} \quad \cdots \quad \boldsymbol{J}_{\mathrm{n}}^{\mathrm{T}}\right]^{\mathrm{T}}$;
        $\Delta \boldsymbol{u} \leftarrow -\left(\boldsymbol{J}^T\boldsymbol{J} + \beta \boldsymbol{E}_{3\mathrm{n}+6}\right)^{-1} \boldsymbol{J}^T \boldsymbol{f}$;
        $\tilde{\boldsymbol{R}}_i \leftarrow \boldsymbol{R}_i \exp\left(\left[\Delta \boldsymbol{\omega}_i\right]^{\wedge}\right), i = 1, 2, ..., \mathrm{n}, h$;
        $\tilde{\boldsymbol{t}}_h \leftarrow \boldsymbol{t}_h + \Delta \boldsymbol{t}_h$;
        $\|\boldsymbol{f}\|$ ←compute the norm of $\boldsymbol{f}$ ; $\beta \leftarrow 10\beta$ ;
    end
    update: $\boldsymbol{R}_i \leftarrow \tilde{\boldsymbol{R}}_i, i = 1, 2, ..., \mathrm{n}, h$;
        $\boldsymbol{t}_h \leftarrow \tilde{\boldsymbol{t}}_h$; $\beta \leftarrow \beta/10$; $\|\boldsymbol{f}\| \leftarrow \|\boldsymbol{f}_0\|$;
        $\|\boldsymbol{f}_0\|$ ←compute the norm of $\boldsymbol{f}$ ;
end
```

Parameter analysis

In this section, parameter analysis is carried on to select the key parameters that have a great impact on the solution accuracy of hand-eye transformation parameters. The key parameters will be as additional parameters and optimized together with the hand-eye transformation parameters to improve the solution accuracy.

It has been proved[30] that the solution accuracy is related to the condition number of the Jacobian matrix $\boldsymbol{J}$. Matrices with small condition number are said to be well-conditioned. In other words, the smaller the condition number of $\boldsymbol{J}$, the better the solution. To understand the impact of data errors on the solution, the total differential of function $\boldsymbol{f}_i(\boldsymbol{R}_i, \boldsymbol{R}_1, \boldsymbol{R}_h, \boldsymbol{t}_h)$ is given as follows

$$\begin{aligned}\Delta \boldsymbol{f}_i = & \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{p}_{si}} \Delta \boldsymbol{p}_{si} + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_i} \Delta \boldsymbol{t}_i - \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_1} \Delta \boldsymbol{t}_1 + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_i} \boldsymbol{\Delta \omega}_i \\ & + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1} \Delta \boldsymbol{\omega}_1 + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_h} \Delta \boldsymbol{\omega}_h + \frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_h} \Delta \boldsymbol{t}_h\end{aligned} \tag{25}$$

where $\boldsymbol{f}_i$ is short for $\boldsymbol{f}_i(\boldsymbol{R}_i, \boldsymbol{R}_1, \boldsymbol{R}_h, \boldsymbol{t}_h)$. And $\Delta \boldsymbol{p}_{si}$, $\Delta \boldsymbol{t}_i$, $\Delta \boldsymbol{t}_1$, $\Delta \boldsymbol{\omega}_i$, $\Delta \boldsymbol{\omega}_1$ denote the data error items, the coefficient matrices of which need to be analyzed. The derivatives of $\boldsymbol{f}_i$ with respect to $\boldsymbol{p}_{si}, \boldsymbol{t}_i, \boldsymbol{t}_1, \boldsymbol{\omega}_i, \boldsymbol{\omega}_1$ are given as follows

$$\begin{cases} \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{p}_{si}} = (\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \mathrm{R}_i \boldsymbol{R}_h \\ \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_i} = (\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \\ \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_1} = (\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \\ \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_i} = -\,(\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \boldsymbol{R}_i [\boldsymbol{R}_h \boldsymbol{p}_{si} + \boldsymbol{t}_h]^{\wedge} \\ \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1} = \boldsymbol{R}_h^{\mathrm{T}} [\boldsymbol{R}_1^{\mathrm{T}} (\boldsymbol{R}_i \boldsymbol{R}_h \boldsymbol{p}_{si} + \boldsymbol{R}_i \boldsymbol{t}_h + \boldsymbol{t}_i - \boldsymbol{t}_1)]^{\wedge} \end{cases} \tag{26}$$

In addition, $\Delta \boldsymbol{\omega}_h, \Delta \boldsymbol{t}_h$ denote the parameter items that need to be estimated. The derivatives of $\boldsymbol{f}_i$ with respect to $\boldsymbol{\omega}_h, \boldsymbol{t}_h$ are given as follows:

$$\begin{cases} \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_h} = [(\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \left(\boldsymbol{R}_i \boldsymbol{R}_h \boldsymbol{p}_{si} + (\boldsymbol{R}_i - \boldsymbol{R}_1) \mathrm{t}_h + \boldsymbol{t}_i - \boldsymbol{t}_1\right)]^{\wedge} - (\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \boldsymbol{R}_i \boldsymbol{R}_h [\boldsymbol{p}_{si}]^{\wedge} \\ \dfrac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_h} = (\boldsymbol{R}_1 \boldsymbol{R}_h)^{\mathrm{T}} \boldsymbol{R}_i - \boldsymbol{R}_h^{\mathrm{T}} \end{cases} \tag{27}$$

where $[\,]^{\wedge}$ denotes Lie bracket.[31] It can be seen that $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_i}, i = 2, ..., n$ remains unchanged and is equal to $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{t}_1}$. If $\boldsymbol{t}_i, \boldsymbol{t}_1$ are optimized as additional parameters, it will cause the linear correlation of the columns of $\boldsymbol{J}$, which will seriously increase the condition number of $\boldsymbol{J}$. Note that $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{p}_{si}}$ is an orthogonal matrix and does not change the norm of the error $\Delta \boldsymbol{p}_{si}$, and the measurement accuracy of the sensor is relatively high (The measurement accuracy is given in the "Experiments" section), the effect of the measurement error caused by the sensor is ignored.

Observe the constitution of $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_i}$ and $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1}$, we can find that $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_i}$ and $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1}$ can amplify the norm of $\Delta \boldsymbol{\omega}_i$ and $\Delta \boldsymbol{\omega}_1$, respectively, due to these items $\boldsymbol{p}_{si}, \boldsymbol{t}_h, \boldsymbol{t}_i - \boldsymbol{t}_1$. This means that slight errors $\Delta \boldsymbol{\omega}_i$ and $\Delta \boldsymbol{\omega}_1$ can cause large error items $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_i} \Delta \boldsymbol{\omega}_i$ and $\frac{\partial \boldsymbol{f}_i}{\partial \boldsymbol{\omega}_1} \Delta \boldsymbol{\omega}_1$. As these errors are shared by the hand-eye transformation parameters $\boldsymbol{R}_h, \boldsymbol{t}_h$, the solution will suffer from a large disturbance. The rotation parameters of the robot pose should be used as the additional parameters that significantly impact the solution accuracy and are optimized together with the hand-eye transformation parameters.

Simulations and experiments

Some simulations and experiments are carried out to verify the effectiveness of the PM. The PM is compared with the modified least square method (MLS) and the constraint least square method (CLS) under different aspects.

Simulations

The simulations are performed to compare the performance of the three methods described in this article from noise level and data bulk. The true values of the hand-eye transformation parameters $\bar{\boldsymbol{\omega}}_h, \bar{\boldsymbol{t}}_h$ are given. The TriMule robot[32] is used for the simulations. $m (m = 10, 20 \ldots 100)$ robot poses $\{\bar{\boldsymbol{\omega}}_i, \bar{\boldsymbol{t}}_i\}_{i=1,\ldots,m}$ are randomly generated in the robot workspace, and one measurement data $\boldsymbol{p}_{si}$ by the sensor under some robot pose is randomly generated simultaneously. According to equation (5), the remaining $m - 1$ measurement data $\{\boldsymbol{p}_{si}\}_{i=1,\ldots,m-1}$ can be calculated. Different levels of noise are added to the generated data as follows: $\boldsymbol{p}_{si} \leftarrow \bar{\boldsymbol{p}}_{si} + \Delta \boldsymbol{p}_{si}, \boldsymbol{\omega}_i \leftarrow \bar{\boldsymbol{\omega}}_i + \Delta \boldsymbol{\omega}_i, \boldsymbol{t}_i \leftarrow \bar{\boldsymbol{t}}_i + \Delta \boldsymbol{t}_i$

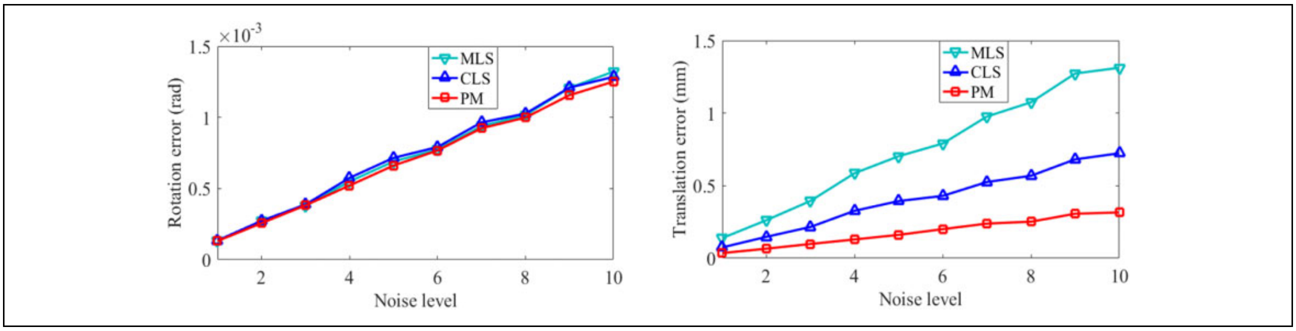

Figure 4. Rotation error and translation error at different noise levels.

$$\begin{cases} \Delta \boldsymbol{p}_{si} \sim \mathcal{N}\left(\boldsymbol{0}, \kappa \mathrm{diag}(\sigma_{px}^2, \sigma_{py}^2, \sigma_{pz}^2)\right) \\ \Delta \boldsymbol{\omega}_i \sim \mathcal{N}\left(\boldsymbol{0}, \kappa \mathrm{diag}(\sigma_{\omega x}^2, \sigma_{\omega y}^2, \sigma_{\omega z}^2)\right) \\ \Delta \boldsymbol{t}_i \sim \mathcal{N}\left(\boldsymbol{0}, \kappa \mathrm{diag}(\sigma_{tx}^2, \sigma_{ty}^2, \sigma_{tz}^2)\right) \end{cases} \tag{28}$$

where $\Delta \boldsymbol{p}_{si}, \Delta \boldsymbol{\omega}_i, \Delta \boldsymbol{t}_i$ are normally distributed noise. $\kappa \in [1, 10]$ is noise level. $\sigma_i^2, i = px, \ldots, tz$ is variance. The rotation error er_{ro} and translation error er_{tr} are defined in the following equations

$$er_{\mathrm{ro}} = \frac{1}{N} \sum_{i=1}^{N} ||\tilde{\boldsymbol{\omega}}_{h,i} - \bar{\boldsymbol{\omega}}_h|| \tag{29}$$

$$er_{\mathrm{tr}} = \frac{1}{N} \sum_{i=1}^{N} ||\tilde{\boldsymbol{t}}_{h,i} - \bar{\boldsymbol{t}}_h|| \tag{30}$$

where $\tilde{\square}$ and $\bar{\square}$ strand for the estimated value and the true value, respectively. The rotation matrices of the hand-eye transformation matrix $\tilde{\boldsymbol{R}}_h, \bar{\boldsymbol{R}}_h$ are parameterized by 3D vectors $\tilde{\omega}_h, \bar{\omega}_h$. The rotation matrix of the robot pose $\bar{\boldsymbol{R}}_i$ generated is also parameterized by 3D vector $\bar{\omega}_i$. N is total number of trials.

(a) Noise level

We set 10 noise levels and randomly select 20 sets of data to estimate the hand-eye transformation matrix with the three methods described above under each noise level; 500 trials are carried out under each noise level to get the average error of rotation and translation. The simulation results are shown in Figure 4. It shows the rotation error and translation error of the three methods at different noise levels.

It can be seen that with the increase of noise level, the rotation error and the translation error of the three methods increase. In terms of solving the rotation matrix, the accuracy difference of the three methods is not obvious, while in solving the translation vector, the accuracy of the PM is significantly better than the other two methods.

(b) Data bulk

We select different amounts of data ($m = 10, 20 \ldots 100$) to estimate the hand-eye transformation matrix using the three methods under low, medium, and high noise levels. Similarly, 500 trials are carried out under each noise level and the same amount of data to get the average error of rotation and translation. The simulation results are shown in Figure 5. Figure 5(a) to (c) shows the rotation error and translation error of the three methods with different amounts of data under low, medium, and high noise levels, respectively.

The simulation results indicate that the rotation error and the translation error of the three methods decrease with the amount of data increasing under each noise level. The accuracy improvement of the three methods is the most obvious when the amount of data becomes from 10 to 20. When about 40 sets of data, the accuracy improvement rate slows down significantly.

In terms of solving the rotation matrix, the PM is slightly better in accuracy than the other two methods before 20 sets of data. After 30 sets of data, MLS becomes the best in accuracy, and the PM is almost the same as CLS in accuracy.

In terms of solving the translation vector, the PM shows excellent advantages in accuracy. The accuracy difference between CLS and MLS gradually becomes smaller.

According to the simulation results of noise level and data bulk, the PM outperforms in the accuracy of the hand-eye transformation matrix than MLS and CLS. Balancing accuracy and efficiency in practical applications, it is recommended to select about 20 sets of data to estimate hand-eye transformation matrix.

Experiments

(a) Experimental setup

The experimental setup mainly consists of a 5-axis hybrid robot called TriMule (averaged repeatability 0.02 mm), a 3D-vision sensor developed by the research team of Tianjin University (field of view 170 mm × 170 mm), a designed

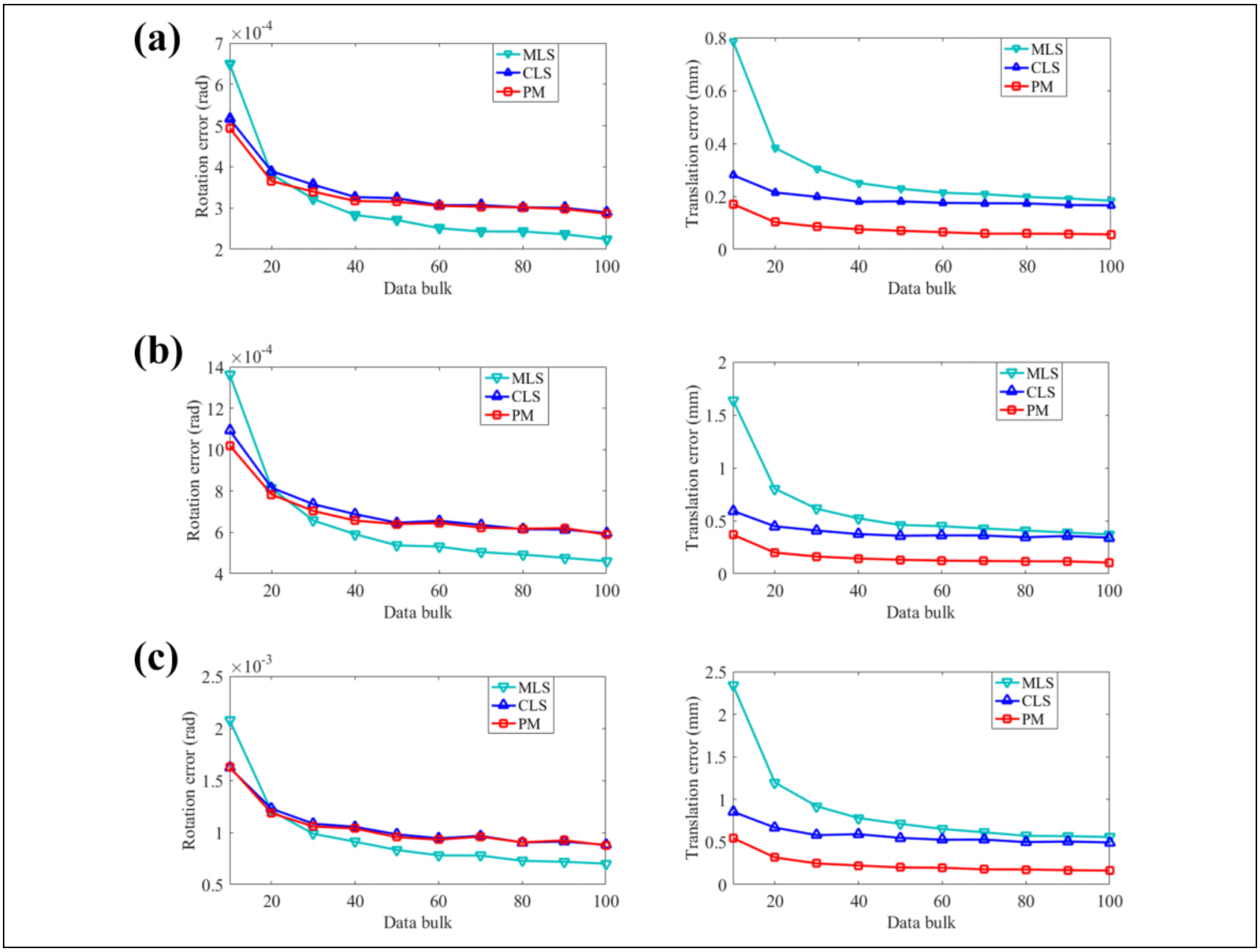

Figure 5. Rotation error and translation error with different amounts of data: (a) low noise level, (b) medium noise level, and (c) high noise level.

artifact with known dimensions and data processing system, as shown in Figure 6. The artifact has two standard balls, ball A with the diameter of 19.9886 mm, ball B with the diameter of 19.9878 mm, whose center distance between the two balls is 59.8573 mm. The data processing system monitors the measurement state in real time and guarantees the measurement quality of the 3D-vision sensor. It also provides images and 3D-point cloud display.

(b) 3D-vision sensor accuracy test

Before verifying the calibration method, the measurement accuracy of the sensor should be tested. The artifact is measured by the sensor from different positions, and the 3D-point cloud of the artifact is obtained. The diameter of ball A and ball B and the center distance between the two balls are calculated by fitting the point cloud measured. By comparing the calculated value with the standard value of the artifact, we can evaluate the measurement accuracy of the sensor. The measurement experiment of the artifact is shown Figure 7. The measurement results are listed in Table 1. Mean absolute error (MAE) represents the deviation between the measured value of the sensor and the truth value. Standard deviation (SD) represents the dispersion of the measurement data. MAE of ball A diameter, ball B diameter and the center distance is 0.0150 mm, 0.0101 mm, and 0.0058 mm, and SD is 0.0041 mm, 0.0094 mm, and 0.0045 mm, respectively, indicating that the accuracy of the sensor is enough high for the hand-eye calibration.

(c) Measurement-assisted robotic positioning

To compare the accuracy of the three methods, the hand-eye calibration experiment is carried out with the ball A of the artifact, and the hand-eye transformation matrix is calculated using the three methods. An important application of hand-eye calibration is to assist robotic positioning with vision sensors. Under the same conditions, the accuracy of auxiliary robotic positioning reflects the accuracy of hand-eye calibration. Hence, an analog experiment is designed to

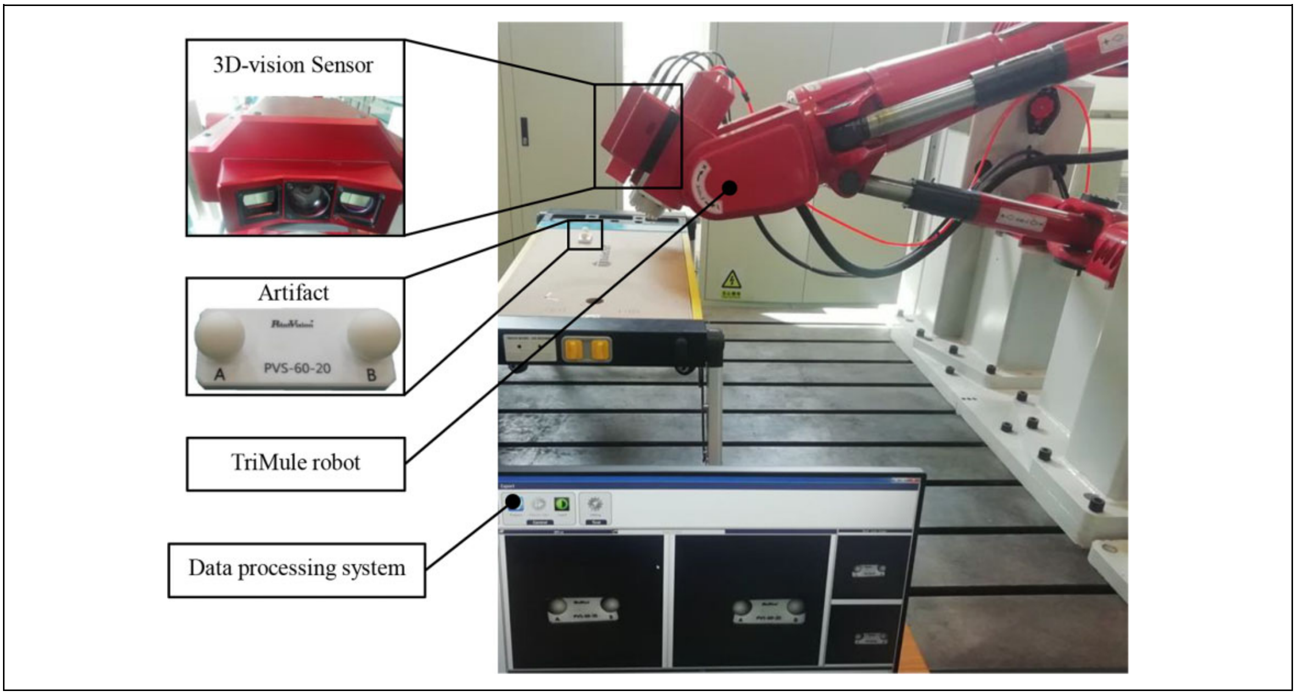

Figure 6. Experimental setup.

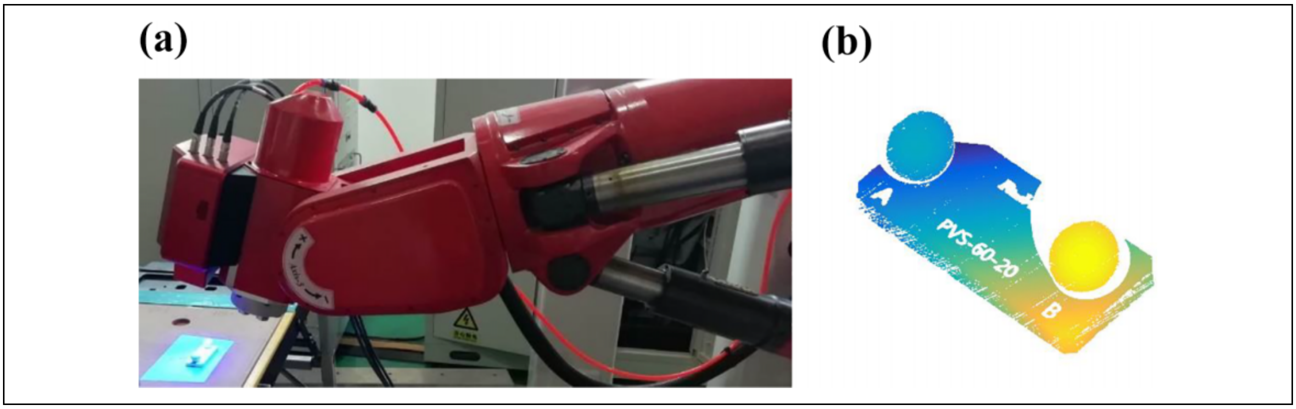

Figure 7. Measurement experiment: (a) measurement of the artifact and (b) 3D-point cloud of the artifact. 3D: three dimensional.

simulate the robotic positioning process of assisted by the 3D-vision sensor. By comparing the positioning accuracy, the accuracy of the three methods in solving the hand-eye transformation matrix can be indirectly evaluated.

The artifact is placed at a position in the robot workspace. The ball A of the artifact is measured by the sensor from one robot pose, and the ball B is measured from the other robot pose. The spherical center coordinates of ball A and ball B are transformed from the sensor frame to the base frame, respectively. The spherical center coordinates of ball A in the base frame are regarded as the starting point, and the center coordinates of ball B are regarded as the target point for positioning. The positioning error er_{po} is defined as follows

$$er_{\text{po}} = |d_{\text{est}} - d_s| \tag{31}$$

$$d_{\text{est}} = ||{}_1\boldsymbol{T}_B^E \tilde{\boldsymbol{T}}_E^S \boldsymbol{p}_a^S - {}_2\boldsymbol{T}_B^E \tilde{\boldsymbol{T}}_E^S \boldsymbol{p}_b^S|| \tag{32}$$

where d_{est}, d_s represent the estimated value and standard value of the ball center distance. $\boldsymbol{p}_a^S, \boldsymbol{p}_b^S$ represent the homogeneous coordinates of the ball A's center in the sensor frame at one robot pose ${}_1\boldsymbol{T}_B^E$ and ball B's center at the other robot pose ${}_2\boldsymbol{T}_B^E$, respectively.

The artifact is placed at 20 different positions in the robot workspace. The positioning errors are shown in Figure 8. The maximum/mean positioning errors are 0.1133/0.0367 mm (MLS), 0.0961/0.0278 mm (CLS), and 0.0703/0.0244 mm (PM), respectively. It can be seen that the

Table 1. Measurement results of the artifact.

Position	Ball A diameter (mm)	Ball B diameter (mm)	Center distance (mm)	Ball A diameter error (mm)	Ball B diameter error (mm)	Center distance error (mm)
1	20.0075	19.9738	59.8696	0.0189	−0.0140	0.0123
2	19.9954	19.9737	59.8506	0.0068	−0.0141	−0.0067
3	20.0036	20.0001	59.8626	0.0150	0.0123	0.0053
4	20.0063	19.9895	59.8628	0.0177	0.0017	0.0055
5	20.0025	19.9790	59.8609	0.0139	−0.0088	0.0036
6	20.0057	19.9730	59.8610	0.0171	−0.0148	0.0037
7	20.0080	19.9903	59.8637	0.0194	−0.0148	0.0037
8	20.0082	19.9863	59.8638	0.0196	−0.0015	0.0065
9	19.9986	19.9706	59.8621	0.0010	−0.0172	0.0048
10	20.0007	19.9733	59.8608	0.0121	−0.0145	0.0035
MAE (mm)				0.0150	0.0101	0.0058
SD (mm)				0.0041	0.0094	0.0045

SD: standard deviation; MAE: mean absolute error.

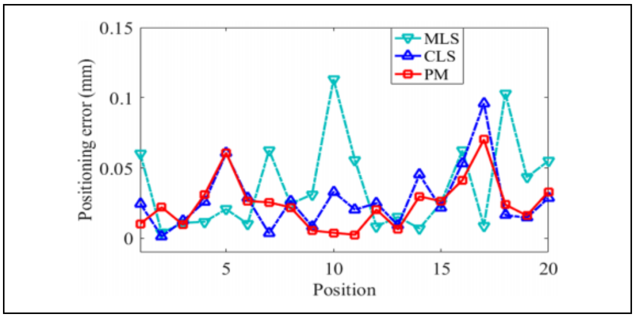

Figure 8. Positioning error at different positions of the artifact.

positioning errors with the PM get lower than that with the other two methods.

(d) Point cloud registration

Another important application of hand-eye calibration is to associate point clouds of the workpiece measured from different robot poses into a common coordinate frame to reconstruct the 3D model of the workpiece. Similarly, the accuracy of the three methods can be indirectly compared through the accuracy of the 3D model reconstructed.

The artifact is placed at a position in the robot workspace and measured by the sensor from two different robot poses. Both sets of point clouds are transformed from the sensor frame to the base frame and are stitched together to get the stitched model. The diameter of ball A and ball B of the stitched model can be calculated and be compared with the standard value of the artifact to get the diameter errors. The point cloud registration process is shown in Figure 9.

The artifact is placed at 20 different positions in the robot workspace. The diameter errors of ball A and ball B are shown in Figure 10. The experimental results of point cloud registration are summarized in Table 2 and further demonstrate the superiority of the PM.

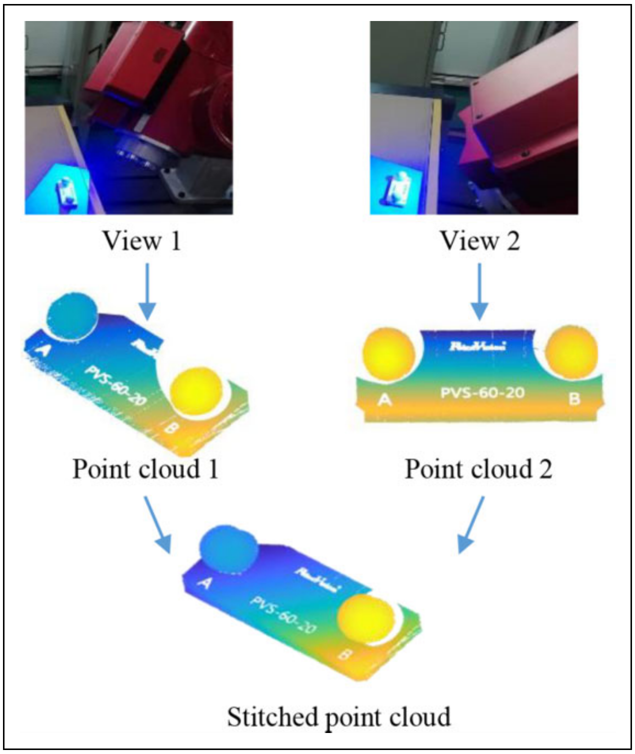

Figure 9. Point cloud registration process.

Conclusions

To guide the positioning of the machining robots, a 3D-vision sensor is integrated at the end effector of machining robots and form the robotic visual measuring system for positioning and measurement. We have proposed an accurate hand-eye calibration method to determine the transformation between the sensor and the robot end effector. A hand-eye calibration model is established in the sensor frame and is capable of providing the more accurate reference point for the solution of hand-eye transformation

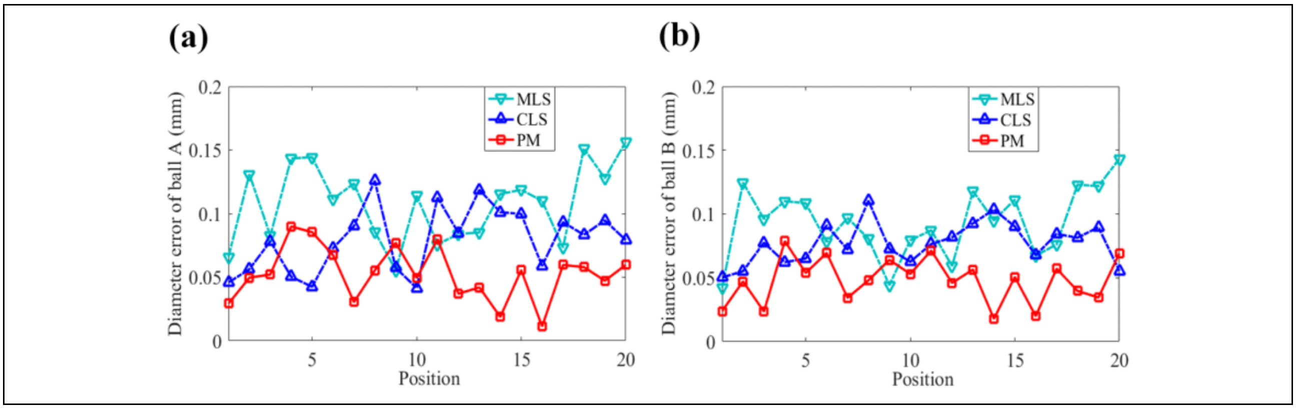

Figure 10. Diameter error of stitched point cloud at different positions of the artifact: (a) ball A and (b) ball B.

Table 2. Experimental results of point cloud registration.

Method	Maximum diameter error of ball A (mm)	Mean diameter error of ball A (mm)	Maximum diameter error of ball B (mm)	Mean diameter error of ball B (mm)
MLS	0.1565	0.1078	0.1433	0.0932
CLS	0.1258	0.0793	0.1105	0.0771
PM	0.0899	0.0528	0.0788	0.0480

parameters. A nonlinear optimization algorithm has been proposed. It is implemented with the consideration of the effects of the robot pose error, where it should be necessary to consider in high precision robotic machining. To avoid the linear correlation problems, parameter analysis is performed and the rotation parameters of the robot pose having a significant impact on the solution accuracy are selected as additional parameters to be optimized. A constraint least square solution has been given and provides good initial values for optimization. The simulation results on noise level and data bulk demonstrate that the PM has higher accuracy in solving hand-eye transformation matrix than MLS and CLS. Positioning and measurement experiments are designed and are carried out on a 5-axis hybrid robot named TriMule. The maximum/mean positioning errors have been reduced from 0.1133/0.0367 mm (MLS), 0.0961/0.0278 mm (CLS) to 0.0703/0.0244 mm (PM) in the positioning experiment, and the maximum diameter errors of ball A and ball B have been reduced from 0.1565/0.1433 mm (MLS), 0.1258/0.1105 mm (CLS) to 0.0899/0.0788 mm (PM) in the point cloud registration experiment. The experimental results further prove the superiority of the PM over MLS and CLS in terms of positioning accuracy and point cloud stitching accuracy. Additionally, the measurement accuracy of the 3D-vision sensor is also tested.

In the future work, we will fix the machining robot on a mobile platform and develop a mobile robotic machining system for large complex components. The calibration method will be investigated to ensure global positioning of the mobile robot.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work is supported by National Key Research and Development Program (Grant No. 2017YFE0111300), National Natural Science Foundation of China (Grant No. 51721003 and No. 51775376), and EU H2020 project (Grant No. 734272).

ORCID iD

Jinsheng Fu https://orcid.org/0000-0002-7500-0692

References

1. Kim SH, Nam E, Ha TI, et al. Robotic machining: a review of recent progress. *Int J Precis Eng Manuf* 2019; 20: 1629–1642.
2. Savio E, De Chiffre L, Carmignato S, et al. Economic benefits of metrology in manufacturing. *CIRP Ann-Manuf Techn* 2016; 65(1): 495–498.
3. Moeller C, Schmidt HC, Koch P, et al. Real time pose control of an industrial robotic system for machining of large scale components in aerospace industry using laser tracker system. *SAE Int J Aerosp* 2017; 10(2): 100–108.
4. Verl A, Valente A, Melkote S, et al. Robots in machining. *CIRP Ann-Manuf Techn* 2019; 68: 799–822.

5. Li J, Zhu J, Guo Y, et al. Calibration of a portable laser 3-D scanner used by a robot and its use in measurement. *Opt Eng* 2008; 47(1): 017202.
6. Liu T, Yin S, Guo Y, et al. Rapid global calibration technology for hybrid visual inspection system. *Sensors* 2017; 17(6): 1440.
7. Zhang Z. A flexible new technique for camera calibration. *IEEE T Pattern Anal* 2000; 22(11): 1330–1334.
8. Wang X, Zhao Y, and Yang F. Camera calibration method based on Pascal's theorem. *Int J Adv Robot Syst* 2019; 16(3): 1–10.
9. Zhao Z and Liu Y. A hand–eye calibration algorithm based on screw motions. *Robotica* 2009; 27: 217–223.
10. Horaud R and Dornaika F. Hand-eye calibration. *Int J Robot Res* 1995; 14(3): 195–210.
11. Daniilidis K. Hand-eye calibration using dual quaternions. *Int J Robot Res* 1999; 18(3): 286–298.
12. Shah M. Solving the robot-world/hand-eye calibration problem using the Kronecker product. *J Mech Robot* 2013; 5(3): 031007.
13. Zhang Q and Gao G. Hand-eye calibration and grasping pose calculation with motion error compensation and vertical-component correction for 4-R (2-SS) parallel robot. *Int J Adv Robot Syst* 2020; 17(2): 1–14.
14. Ren Y, Yin S, and Zhu J. Calibration technology in application of robot-laser scanning system. *Opt Eng* 2012; 51(11): 114204.
15. Chen W, Du J, Xiong W, et al. A noise-tolerant algorithm for robot-sensor calibration using a planar disk of arbitrary 3-D orientation. *IEEE T Autom Sci Eng* 2018; 15(1): 251–263.
16. Tsai RY and Lenz R K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE T Robot Autom* 1989; 5(3): 345–358.
17. Li A, Wang L, and Wu D. Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product. *Int J Phys Sci* 2010; 5(10): 1530–1536.
18. Ulrich M, Heider A, and Steger C.Hand-eye calibration of SCARA robots. In: *9th Open German-Russian workshop on pattern recognition & image understanding*, Koblenz, Germany, 1–5 December 2014.
19. Bayrocorrochano E, Daniilidis K, Sommer G, et al. Motor algebra for 3D kinematics: the case of the hand-eye calibration. *J Math Imaging Vis* 2000; 13(2): 79–100.
20. Shah M, Eastman RD, and Hong T. An overview of robot-sensor calibration methods for evaluation of perception systems. In: *Proceedings of the Workshop on Performance Metrics for Intelligent Systems* (ed R Madhavan), PerMIS'12, College Park, MD USA, 20 March 2012, pp. 15–20. Maryland: ACM.
21. Ulrich M and Steger C. Hand-eye calibration of SCARA robots using dual quaternions. *Pattern Recog Image Anal* 2016; 26(1): 231–239.
22. Zuang H and Shiu YC. A noise-tolerant algorithm for robotic hand-eye calibration with or without sensor orientation measurement. *IEEE T Syst Man Cy* 1993; 23(4): 1168–1175.
23. Tabb AL and Yousef KM. Solving the robot-world hand-eye(s) calibration problem with iterative methods. *Mach Vision Appl* 2017; 28(5-6): 569–590.
24. Yin S, Ren Y, Guo Y, et al. Development and calibration of an integrated 3D scanning system for high-accuracy large-scale metrology. *Measurement* 2014; 54: 65–76.
25. Li W, Xie H, Zhang G, et al. Hand-eye calibration in visually-guided robot grinding. *IEEE Trans Cybern* 2016; 46(11): 2634–2642.
26. Wu J, Wang J, and You Z. An overview of dynamic parameter identification of robots. *Robot Cim-Int Manuf* 2010; 26(5): 414–419.
27. Wu J, Yu G, Gao Y, et al. Mechatronics modeling and vibration analysis of a 2-DOF parallel manipulator in a 5-DOF hybrid machine tool. *Mech Mach Theory* 2018; 121: 430–445.
28. Quo F and Ho KC. A quadratic constraint solution method for TDOA and FDOA localization. In: *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (ed M Tomer), Prague, Czech Republic, 22–27 May 2011, pp. 2588–2591. IEEE.
29. More J. The Levenberg-Marquardt algorithm, implementation and theory. In: Watson GA (ed) *Numerical analysis, lecture notes in mathematics*, Berlin: Springer-Verlag. 1977.
30. Golub GH and Loan CFV. *Matrix computations*. Baltimore: Johns Hopkins University Press, 1983.
31. Murray R, Li Z, and Sastry S. *A mathematical introduction to robotic manipulation*. Boca Raton: CRC press, 1994.
32. Liu Q and Huang T. Inverse kinematics of a 5-axis hybrid robot with non-singular tool path generation. *Robot Cim-Int Manuf*, 2019; 56: 140–148.