Toward Simultaneous Coordinate Calibrations of AX=YB Problem by the LMI-SDP Optimization

Jiabin Pan, Zhongtao Fu, Hengtao Yue, Xiaoyu Lei, Miao Li, *Senior Member, IEEE*, and Xubing Chen, *Member, IEEE*

***Abstract*— Accurate calibration of the robot hand-eye (X) and robot-world (Y) relationships is extremely important for visually-guided robotic systems, and is usually symbolized by the AX=YB equation. The existing methodologies always calibrate the X and Y matrices using the separation of the rotational and translational components, causing the error propagation and accumulation. While the simultaneous calibration solves the derived linear matrix equation by the *SVD* (Singular Value Decomposition) based approach, which produces unreliable results depending on the smallest singular value of the regression matrix. To this end, the work contained herein proposes a novel and generic calibration methodology for solving the AX=YB problem using the LMI-SDP (Linear Matrix Inequality and Semi-definite Programming) optimization. In this approach, the linear form of the calibration equation is retrieved by means of the Kronecker product, and formulated as an optimization problem involving the unknown variable matrices X and Y with convex constraints, in which the simultaneous solution is obtained via the LMI-SDP techniques. The results procured via the simulation analysis, accounting for the presence of noise levels and different data pairs, as well as the calibration experiments, are compared to those produced using the classical iterative method and *DQ* (Dual Quaternion)-based approach, thereby verifying the accuracy and efficacy of the proposed method.**

***Note to Practitioners*—The motivation behind this work stems from the simultaneous calibration issues pertaining to robot-eye and robot-workpiece coordinate relationships, that are present in vision-guided robotic systems. Considering the inaccuracy and robustness deficiencies of the existing methodologies, due to the separated calibration of the rotational and translational components, this paper proposes a generic and efficient calibration methodology to deal with the AX=YB problem, using the Kronecker product and the LMI-SDP optimization. Simulation analysis reveals that the proposed algorithms exhibit robustness under different noise levels and data pairs. Moreover, the practicability of the algorithm has also been verified via practical experiments. The average errors with 16 sets of calibration data can reach 0.0056rad in the rotational component, and 0.2529mm in the translational component. The proposed methodology can be extended to the practical applications of the coordinate calibration involving the multi-robot systems with visual sensors.**

***Index Terms*— AX = YB, simultaneous coordinate calibration, Kronecker product, LMI-SDP.**

I. Introduction

THE visual camera can provide perceptive information of the various tasks in a working environment, for robotic operations, such as grasping, welding, and assembly [1]. In practical applications, the camera is usually fixed at the robot end-effector or independently atop the robotic device to form a robot hand-eye system [2]. In general, the hand-eye and robot-world calibration constitutes a more effective method than the classical robot hand-eye calibration approach, in terms of practical scenarios, and is usually formulated as a solution to the equation **AX=YB**, as shown in Fig.1 (red line), in which the known homogeneous transformation matrices (HTM) **A** and **B** are obtained by the robot controllers and visual sensors, while the two unknown matrices **X** and **Y**, represent the following homogeneous transformations: **X** is located between the sensor frame and the robot end-effector frame; **Y** is located between the robot base frame and the world frame. However, there exist some uncertainties, such as variation of the robot base frame and replacement of various tools and/or sensors in different scenarios, and the general mechanical methods utilize calibration poles to cumbersomely obtaion the **A** and **B** matrices, which inevitably increase the calibration workload and inaccuracy. Therefore, in order to achieve the robotic manipulation with high accuracy, and simplify the calibration procedures, research efforts have been directed towards the development of various algorithms for the coordinate calibration of the hand-eye and robot-world relationships, which is a necessity for the achievement of the accurate, visually-guided robot motion.

A. Related Work

The calibration equation, **AX=YB**, of hand-eye and robot-world relationships, originates from the extension of the

Manuscript received 27 May 2022; revised 26 July 2022; accepted 15 September 2022. This article was recommended for publication by Associate Editor B. Zhang and Editor Q. Zhao upon evaluation of the reviewers' comments. This work was supported in part by the Natural Science Foundation of China under Grant 51805380 and Grant 51875415, in part by the Key Research and Development Project of Hubei under Grant 2021BCA138, in part by the Foundation of State Key Laboratory of Digital Manufacturing Equipment and Technology under Grant DMETKF2022019, and in part by the Central Guidance on Local Science and Technology Development Foundation of Hubei Province under Grant 2019ZYYD010. *(Corresponding author: Zhongtao Fu.)*

Jiabin Pan is with the School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan 430205, China, and also with the School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200444, China.

Zhongtao Fu, Hengtao Yue, Xiaoyu Lei, and Xubing Chen are with the School of Mechanical and Electrical Engineering, Wuhan Institute of Technology, Wuhan 430205, China (e-mail: hustfzt@gmail.com).

Miao Li is with the Institute of Technological Sciences, Wuhan University, Wuhan 430072, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TASE.2022.3207771.

Digital Object Identifier 10.1109/TASE.2022.3207771

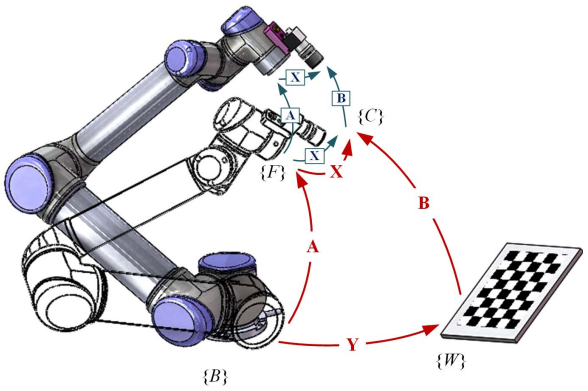

Fig. 1. The formulation of the **AX=XB** and **AX=YB** problems.

classical robot hand-eye problem **AX=XB** [3], as depicted in Fig.1 (blue line). Initial work on hand-eye problem was carried out by Shiu *et al.* [4], who transformed **AX=XB** into a linearized form, and pointed out that the optimized solution was eventually required, at least three different configurations. Chou *et al.* [5] established the quaternion-based symbolic equations of the hand-eye problem and obtained a linear solution by means of the *SVD* method. Subsequently, numerous mathematical tools were employed to formulate calibration equation, such as rotation axis and angle [6], *DQ* [7], Euclidean group [8], Clifford algebra [9] and screw theory [10]. The associated solving procedures for obtaining the final solutions mainly comprise two form, i.e., iterative and analytical solutions. The iterative-based method typically transforms **AX=XB** into convex optimization problem, such as second-order cone program [11], to obtain optimal solution, while analytical methods can provide the solution's expression immediately. Ma [12] and Hu [13] presented a probabilistic approach to derive a closed-form solution of **X**, which aimed to deal with simple data without prior knowledge of time correspondence, and was not applicable to high-noise scenarios [14]. For this issue, Ref. [15] suggested a hybrid method that solved the calibration equation using the Kronecker product and *SVD*-based algorithm to obtain approximate solution, which was finally optimized via an iterative algorithm with epipolar and spatial distance constraints to improve accuracy.

The **AX=YB** calibration problem is even more complicated than the classical robot hand-eye equation, since higher dimensions of the unknown variables need to be acquired. Shah [16] constructed a separable closed-form solution of robot-world (**Y**) and hand-eye (**X**) by means of an *SVD*-based algorithm. However, the orthogonality of the calculated matrices $\mathbf{R}_X$ and $\mathbf{R}_Y$ cannot be guaranteed due to the noise factors. A common way is to introduce orthogonalization constrain on the rotation components in the optimization problem, while it is difficult to achieve, particularly in linear programming methods [17], due to the nonlinearity of the constraints. Based on this, the conventional procedure involved performing an orthogonality operation on the rotational matrix of the calibrated results. Ref. [18] proposed a *QR15* method, which assumed that the unknown rotation matrix was a 3×3 non-orthogonal matrix, and the desired rotation matrix was approximated via orthogonalization process. Wang *et al.* [19] established an objective equation with a penalty function of a multiplier to update the rotation matrix, so that it would be unitary and orthogonal. However, the subsequent orthogonalization of the rotation matrix may have been detrimental to accuracy, especially when the calibration solution was inaccurate.

In recent years, many researches works revolve around techniques for collaborative calibration of dual robot coordinates, **AXB=YCZ**, such as probabilistic approache [20], iterative methods [21], [22], and analytical method [23], most of which tend to include an element of simultaneous calibration of **X**, **Y** and **Z** matrices, and rotational and translational components for each unknown **X**, **Y** and **Z** variables [24]. Generally, the simultaneous methods exhibit higher accuracy, which is mainly due to the fact that separate (non-simultaneous) calibration methods entail multiple steps, the error generated in the first calibration step will affect the accuracy of the subsequent step. The iterative algorithm proposed by Wang *et al.* [22] obtains simultaneous solution of three unknown variables in the same equation, while the rotation matrix and the translation vector are solved separately. Our previous work based on the *DQ* technique, achieves simultaneous calibration of rotational and translational components [23], which allow **X**, **Y**, and **Z** to be solved separately, i.e., **AXB=YCZ** is decomposed into a two-step calibration problem involving **AX=XB** and **AX=YB**. Nevertheless, the accuracy of the solution in the second step is severely affected by the smallest singular value of the regression matrix, when *SVD*-based algorithms are employed.

B. Proposed Method and Contributions

The coordinate calibration of hand-eye and robot-world relationships, **AX=YB**, has a mathematical equation similar to **AX=XB** and **AXB=YCZ**, which reveals that the proposed methodology can be extended to deal with these different problems. In this work, the **AX=YB** equation was rewritten in SDP form with an LMI constraint, in which the error upper limit variable is firstly introduced to convert the calibration equation **AX=YB** into a convex optimization problem with a positive definite matrix constraint; hence, the solution in the convex set can be obtained via global optimization. Based on this, the specific contributions offered by this work are the following:

1) The proposed method based on LMI-SDP[1] technology, utilizes convex constraints to limit the solution to a very small feasibility space to find the optimal simultaneous solution, avoiding error propagation and accumulation.

2) A iterative method existing in [22] was rewritten for the **AX=YB** problem, combined with the closed-form solution method [23], the performance analysis of the proposed algorithm was given an in-depth explanation.

3) The proposed method provides excellent robustness and solvability, which has been fully proved in simulation and experiment, and the relevant code is provided on Github.[2]

[1]Linear matrix inequalities (LMI) and semi-definite programming (SDP) symbolic tools for Python at https://pypi.org/project/PyLMI-SDP/

[2]Experimental and simulation code of Fanuc LR Mate 200iD robot are available online at https://github.com/GilbertPan97/RobCalibration_AXYB

The remainder of this letter commences with Section II, where the calibration problem is modeled and formulated, and then three approaches, proposed LMI-SDP-based method, closed form solution and the iterative method are presented. Section III provides analysis of three algorithms, including robustness analysis and calibration consumption analysis. Furthermore, Section IV presents the experiment verification results and analysis, and the conclusions of this letter are given in Section V.

II. Problem Description and Modeling

One rigid-body transformation in 3D Euclidean space can be viewed as one Lie group element, which is defined via the homogeneous transformation matrix (HTM) as:

$$\mathbf{g} := \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \in SE(3) \tag{1}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{p} \in \mathbb{R}^3$ denote the orientation and position of the rigid-body respectively.

As depicted in Fig.1, the kinematic chain, **AX=YB**, pertaining to the robot base frame $\{B\}$ and the camera frame $\{C\}$, is established by expanding Eq.(1) as follows:

$$\begin{bmatrix} \mathbf{R}_A & \mathbf{t}_A \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_X & \mathbf{t}_X \\ \mathbf{0}^T & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_Y & \mathbf{t}_Y \\ \mathbf{0}^T & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{R}_B & \mathbf{t}_B \\ \mathbf{0}^T & 1 \end{bmatrix} \tag{2}$$

The following expressions are then obtained as:

$$\mathbf{R}_A\mathbf{R}_X = \mathbf{R}_Y\mathbf{R}_B \tag{3}$$

$$\mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_Y\mathbf{t}_B + \mathbf{t}_Y \tag{4}$$

where $\mathbf{R}_A$, $\mathbf{R}_B$ and $\mathbf{t}_A$, $\mathbf{t}_B$ represent the known rotation matrices and translation vectors respectively, while $\mathbf{R}_X$, $\mathbf{R}_Y$ and $\mathbf{t}_X$, $\mathbf{t}_Y$ represent the unknown rotation matrices and translation vectors, respectively, which will be solved in Section III.

III. Problem Solution

In this section, the proposed LMI-SDP-based methodology with Kronecker product is introduced to solve Eq.(2), and compared with the iterative and *DQ*-based algorithms, for the purpose of demonstrating the superiority of the proposed approach. Moreover, the solvability of the associated algorithm and error analysis are investigated.

A. Kronecker Product and LMI-SDP-Based Algorithm

Considering the matrix equation **AXB=C**, where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\mathbf{B} \in \mathbb{R}^{p\times q}$, and $\mathbf{C} \in \mathbb{R}^{m\times q}$ are given known matrices and $\mathbf{X} \in \mathbb{R}^{n\times p}$ is an unknown matrix. **AXB=C** can be rewritten as one linear equation by means of the Kronecker product as follows:

$$(\mathbf{B}^T \otimes \mathbf{A}) \cdot vec(\mathbf{X}) = vec(\mathbf{AXB}) \tag{5}$$

where the symbol $\otimes$ signifies the Kronecker operation, and $vec(\cdot)$ represents a column vector constituted of the matrix elements arranged in column order.

Thus, Eqs.(3) and (4) can be derived as one linear equation based on the identity relation in Eq.(5), with the following form:

$$(\mathbf{R}_B \otimes \mathbf{R}_A) \cdot vec(\mathbf{R}_X) = = vec(\mathbf{R}_Y) \tag{6}$$

$$\mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = (\mathbf{t}_B^T \otimes \mathbf{I}_{3\times3}) \cdot vec(\mathbf{R}_Y) + \mathbf{t}_Y \tag{7}$$

Stacking Eqs.(6) and (7) into matrix form, the linear equation, pertaining to the problem **AX=YB**, is obtained as:

$$\tilde{\mathbf{H}} \cdot \boldsymbol{\beta} = \tilde{\boldsymbol{\omega}} \tag{8}$$

where $\tilde{\boldsymbol{\omega}}$ is stacked by $\boldsymbol{\omega}_i = \begin{bmatrix} \mathbf{0}_{9\times1} \\ \mathbf{t}_{A_i} \end{bmatrix} \in \mathbb{R}^{12}$, $\tilde{\mathbf{H}}$ is stacked by $\mathbf{H}_i = \begin{bmatrix} \mathbf{R}_{B_i} \otimes \mathbf{R}_{A_i} & -\mathbf{I}_{9\times9} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{9\times3} & \mathbf{t}_{B_i}^T \otimes \mathbf{I}_{3\times3} & -\mathbf{R}_{A_i} & \mathbf{I}_{3\times3} \end{bmatrix} \in \mathbb{R}^{12\times24}$, and $\boldsymbol{\beta} = \begin{bmatrix} vec(\mathbf{R}_X) \\ vec(\mathbf{R}_Y) \\ \mathbf{t}_X \\ \mathbf{t}_Y \end{bmatrix} \in \mathbb{R}^{24}$; $\tilde{\boldsymbol{\omega}}$, $\tilde{\mathbf{H}}$ and $\boldsymbol{\beta}$ represent the constant vector, regression matrix and unknown variable vector, respectively.

In general, for the calibration dataset $(\mathbf{A}_i, \mathbf{B}_i)$ acquired via practical experiments, the accompanying noise signals are inevitably introduced. Herein, the error vector, $\boldsymbol{\epsilon} \in \mathbb{R}^{24}$, is employed to describe the error characteristics of Eq.(8) as:

$$\boldsymbol{\epsilon} = \tilde{\mathbf{H}} \cdot \boldsymbol{\beta} - \tilde{\boldsymbol{\omega}} \tag{9}$$

The regression matrix, $\tilde{\mathbf{H}}$, is sparse and possesses a larger dimension for the numerous calibration datasets. The *QR* decomposition algorithm is unitized to deal with this problem for the optimization method [25], [26], and the matrix $\tilde{\mathbf{H}}$ can be expressed as:

$$\tilde{\mathbf{H}} = \mathbf{QR} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \tag{10}$$

The error vector can be further formulated as:

$$\|\boldsymbol{\epsilon}\|^2 = (\mathbf{Q}^T\boldsymbol{\epsilon})(\mathbf{Q}^T\boldsymbol{\epsilon})^T = \|\mathbf{Q}^T\boldsymbol{\epsilon}\|^2 \tag{11}$$

Substituting Eqs.(9) and (10) into Eq.(11), the following equation is derived:

$$\|\boldsymbol{\epsilon}\|^2 = \left\| \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix} \boldsymbol{\beta} - \mathbf{Q}^T\tilde{\boldsymbol{\omega}} \right\|^2 = \|\mathbf{R}_1\boldsymbol{\beta} - \mathbf{Q}_1^T\tilde{\boldsymbol{\omega}}\|^2 + \|\mathbf{Q}_2^T\tilde{\boldsymbol{\omega}}\|^2 \tag{12}$$

By means of the minimal objective $u \geq \|\boldsymbol{\epsilon}\|$, Eq.(12) can be transformed into one linear matrix inequality (LMI) as:

$$u - \|\mathbf{Q}_2^T\tilde{\boldsymbol{\omega}}\|^2 \geq \|\mathbf{R}_1\boldsymbol{\beta} - \mathbf{Q}_1^T\tilde{\boldsymbol{\omega}}\|^2 \tag{13}$$

Therefore, Eq.(13) can be solved using the following SDP form:

$$\begin{cases} \underset{(u,\boldsymbol{\beta})}{\text{minimize}} \ u \\ s.t \begin{bmatrix} u - \left\|\mathbf{Q}_2{}^T\tilde{\boldsymbol{\omega}}\right\|^2 & \left(\mathbf{R}_1\cdot\boldsymbol{\beta} - \mathbf{Q}_1{}^T\tilde{\boldsymbol{\omega}}\right)^T \\ \mathbf{R}_1\cdot\boldsymbol{\beta} - \mathbf{Q}_1{}^T\tilde{\boldsymbol{\omega}} & 1 \end{bmatrix} \succ 0 \end{cases} \tag{14}$$

The symbolic toolbox, named PyLMI-SDP, is utilized to solve the optimization problem expressed as Eq.(14). Once the optimized solution, $\boldsymbol{\beta}$, is retrieved, the constraint of the rotation matrix needs to be constructed via the Schmidt orthogonalization.

B. Iterative Algorithm

Based on the work of Ref. [22], an iterative method for solving **AX=YB**, will be introduced and derived, in which the solving process comprising the rotation matrices, $\mathbf{R}_X$, $\mathbf{R}_Y$, and translation vectors, $\mathbf{t}_X$, $\mathbf{t}_Y$, is implemented separately.

1) Solve for Rotation Matrices, $\mathbf{R}_X$ and $\mathbf{R}_Y$: By introducing the true value and initial value of the rotation matrix, $\mathbf{R}_{q,true}$ and $\mathbf{R}_{q,init}$ $(q = \mathbf{X}, \mathbf{Y})$ for the iterative algorithm, the rotation equation in Eq.(3) can be rewritten as:

$$\mathbf{R}_A(\mathbf{R}_{X,true} \cdot \mathbf{R}_{X,init}^{-1})\mathbf{R}_{X,init} = (\mathbf{R}_{Y,true} \cdot \mathbf{R}_{Y,init}^{-1})\mathbf{R}_{Y,init}\mathbf{R}_B \tag{15}$$

By means of the Lie algebra of the rotation matrix $\mathbf{R}_{q,true} \cdot \mathbf{R}_{q,init}^{-1}$, the approximation expression $\mathbf{R}_{q,true} \cdot \mathbf{R}_{q,init}^{-1} \approx \mathbf{I} + \hat{\boldsymbol{\delta}}_{R,q}$ is obtained, in which $\hat{\boldsymbol{\delta}}_{R,q} \in so(3)$ is the skew-symmetric matrix of the column vector $\boldsymbol{\delta}_{R,q} = \begin{bmatrix}\delta_1 & \delta_2 & \delta_3\end{bmatrix}^T$. Eq.(15) is thus obtained as:

$$\mathbf{R}_A\hat{\boldsymbol{\delta}}_{Rx}\mathbf{R}_{X,init} - \hat{\boldsymbol{\delta}}_{Ry}\mathbf{R}_{Y,init}\mathbf{R}_B = \mathbf{R}_{Y,init}\mathbf{R}_B - \mathbf{R}_A\mathbf{R}_{X,init} \tag{16}$$

with the identity, $\hat{\mathbf{a}} \cdot \mathbf{b} = -\hat{\mathbf{b}} \cdot \mathbf{a}$, Eq.(16) can be further rewritten in the linear form:

$$\mathbf{F} \cdot \boldsymbol{\delta} = \mathbf{q} \tag{17}$$

where

$$\mathbf{F} = \begin{bmatrix} -\mathbf{R}_A \cdot [vec_1(\mathbf{R}_{X,init})]\widehat{\ } & [vec_1(\mathbf{R}_{Y,init} \cdot \mathbf{R}_B)]\widehat{\ } \\ -\mathbf{R}_A \cdot [vec_2(\mathbf{R}_{X,init})]\widehat{\ } & [vec_2(\mathbf{R}_{Y,init} \cdot \mathbf{R}_B)]\widehat{\ } \\ -\mathbf{R}_A \cdot [vec_3(\mathbf{R}_{X,init})]\widehat{\ } & [vec_3(\mathbf{R}_{Y,init} \cdot \mathbf{R}_B)]\widehat{\ } \end{bmatrix} \in \mathbb{R}^{9\times 6},$$

$\mathbf{q} = vec(\mathbf{R}_{Y,init} \cdot \mathbf{R}_B - \mathbf{R}_A \cdot \mathbf{R}_{X,init}) \in \mathbb{R}^9$, $\boldsymbol{\delta} = \begin{bmatrix}\boldsymbol{\delta}_{R,X}^T & \boldsymbol{\delta}_{R,Y}^T\end{bmatrix} \in \mathbb{R}^6$ For the acquired calibration datasets $(\mathbf{A}_i, \mathbf{B}_i)$, Eq.(17) can be stacked linearly as $\tilde{\mathbf{F}} \cdot \boldsymbol{\delta} = \tilde{\mathbf{q}}$, and the initial solution of $\boldsymbol{\delta}$, can be calculated via the pseudo-inverse operation as:

$$\boldsymbol{\delta} = (\tilde{\mathbf{F}}^T\tilde{\mathbf{F}})^{-1}\tilde{\mathbf{F}}^T\tilde{\mathbf{q}} \tag{18}$$

$\boldsymbol{\delta}$ is then used to update the initial value of $\mathbf{R}_{q,init}$ via the following relationship:

$$\mathbf{R}_{q,init}^{i+1} = \exp(\hat{\boldsymbol{\delta}}_{R,q}) \cdot \mathbf{R}_{q,init}^{i} = (\mathbf{I} + \sum_{n=1}^{\infty}(\hat{\boldsymbol{\delta}}_{R,q})^n/n!) \cdot \mathbf{R}_{q,init}^{i} \tag{19}$$

where i is the iterative number of the rotation matrix, and the iterative process is stopped when the rotational error satisfies the desired values.

2) Solve for Translation Vectors, $\mathbf{t}_X$ and $\mathbf{t}_Y$: Once the rotation matrices, $\mathbf{R}_X$ and $\mathbf{R}_Y$, are retrieved and substituted into Eq.(4), the linear form can be obtained as:

$$\mathbf{J} \cdot \mathbf{t} = \mathbf{p} \tag{20}$$

where $\mathbf{J} = \begin{bmatrix}\mathbf{R}_A & -\mathbf{I}\end{bmatrix}$, and $\mathbf{p} = \mathbf{R}_Y \cdot \mathbf{t}_B - \mathbf{t}_A$.

Similarly, the full linear equation, $\tilde{\mathbf{J}} \cdot \mathbf{t} = \tilde{\mathbf{p}}$, is grouped into Eq.(20) with the same dataset $(\mathbf{A}_i, \mathbf{B}_i)$. the solution of the translation vectors, $\mathbf{t}_X, \mathbf{t}_Y$, is obtained as follows:

$$\mathbf{t} = (\tilde{\mathbf{J}}^T\tilde{\mathbf{J}})^{-1}\tilde{\mathbf{J}}^T\tilde{\mathbf{p}} \tag{21}$$

C. DQ-Based Algorithm

The *DQ*-based algorithm is firstly applied to the hand-eye calibration problem [7], **AX= XB**, and extended to the coordinate calibration problem, **AXB=YCZ**, of the dual-robot collaborative motion in our previous work [23]. The solving process relating to **AXB=YCZ** comprised two steps: 1) the *DQ*-based equation **AX=XB** is derived to get the solution of **X** by letting one robot move freely, while keeping another robot in a fixed configuration; 2) the simultaneous calibration of **Y** and **Z** by means of a *DQ*-based equation **AZ=YB** in the similar way. Based on Step (2), the equation **AX=YB** can be also rewritten into *DQ* form as:

$$\hat{\mathbf{q}}_A\hat{\mathbf{q}}_X = \hat{\mathbf{q}}_Y\hat{\mathbf{q}}_B \tag{22}$$

where $\hat{\mathbf{q}}_{(\cdot)}$ is an extension to the quaternion combined with a dual number,

$$\hat{\mathbf{q}}_{(\cdot)} = \mathbf{q}_{(\cdot)} + \varepsilon\mathbf{q}'_{(\cdot)} \tag{23}$$

Utilizing the *DQ* multiplication rule, Eq.(22) can be derived as a linear equation, as shown below:

$$\mathbf{D} \cdot \Delta = \mathbf{0} \tag{24}$$

where $\mathbf{D} = \begin{bmatrix} Q_L(\mathbf{q}_A) & \mathbf{0}_{4\times4} & -Q_R(\mathbf{q}_B) & \mathbf{0}_{4\times4} \\ Q_L(\mathbf{q}'_A) & Q_L(\mathbf{q}_A) & -Q_R(\mathbf{q}'_B) & -Q_R(\mathbf{q}_B) \end{bmatrix} \in \mathbb{R}^{8\times16}$, and the unknown column vector $\Delta = \begin{bmatrix}\mathbf{q}_X & \mathbf{q}'_X & \mathbf{q}_Y & \mathbf{q}'_Y\end{bmatrix}^T \in \mathbb{R}^{16}$; the symbols, $Q_L(\cdot)$ and $Q_R(\cdot)$ represent the left and right matrices of the quaternion multiplication, and are defined as:

$$\mathbf{q}_1 \cdot \mathbf{q}_2 = Q_L(\mathbf{q}_1) \cdot \mathbf{q}_2 = Q_R(\mathbf{q}_2) \cdot \mathbf{q}_1 \tag{25}$$

with $Q_L(\mathbf{q}) = \begin{bmatrix} q_0 & \vec{q}^T \\ \vec{q} & q_0\mathbf{I} + \Omega(\vec{q}) \end{bmatrix}$, $Q_R(\mathbf{q}) = \begin{bmatrix} q_0 & -\vec{q}^T \\ \vec{q} & q_0\mathbf{I} - \Omega(\vec{q}) \end{bmatrix}$, where $\Omega(\vec{v})$ is the skew-symmetric matrix of the vector $\vec{v} = (v_1, v_2, v_3)^T$.

The solving process pertaining to Eq.(24) is carried out via the *SVD* algorithm as: $\tilde{\mathbf{D}} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\tilde{\mathbf{D}}$ is a $8n\times16$ matrix, stacked by the $\mathbf{D}$ of the given n datasets ($n \geq 2$). $\mathbf{U}$, Σ and $\mathbf{V}$ denotes an $8n \times 8n$ unitary matrix, an $8n \times 16$ singular-value matrix, and a 16×16 unitary matrix, respectively. The basic solution of Eq.(24) was constructed based on the last two columns, $\mathbf{v}_{15}$ and $\mathbf{v}_{16}$, in $\mathbf{V}$, with the unknown coefficients, μ_1, μ_2 as demonstrated below:

$$\Delta = \begin{bmatrix}\mathbf{q}_X & \mathbf{q}'_X & \mathbf{q}_Y & \mathbf{q}'_Y\end{bmatrix}^T = \mu_1\mathbf{v}_{15} + \mu_2\mathbf{v}_{16} \tag{26}$$

The unique solution in Eq.(26) can be obtained by applying the inherent constraints, $\|\mathbf{q}_{(\cdot)}\| = 1$ and $\|\mathbf{q}'_{(\cdot)}\| = 0$ on Eq.(23).

D. Feasible Solution and Error Analysis

A feasible solution of the proposed LMI-SDP-based method for the calibration equation **AX=YB** can be guaranteed by the following two properties: 1) the left-hand and right-hand sides of **AX=YB** form closed-loop kinematic chains, in which the data pair $(\mathbf{A}_i, \mathbf{B}_i)$, and the calibrated variables **X**, **Y** must satisfy the equation **AX=YB** in Fig.1; there should be a feasible solution of calibrated variables **X** and **Y**. 2) The solving process of the PyLMI-SDP toolbox is performed by

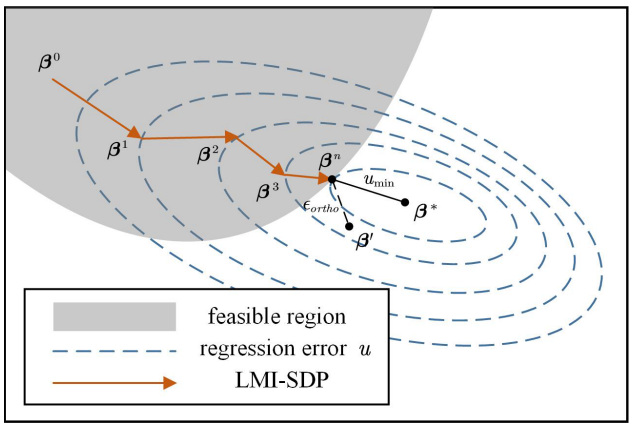

Fig. 2. An illustration of the iterative process of SDP optimization.

the *DSDP5* solver for the optimization problem in Eq.(14), and its solvability mainly depends on the convergence of the *DSDP5* algorithm and the selection of the initial feasibility points; a detailed illustration is available in [27]. What is more, the solving process of the *DSDP5* algorithm is essentially an iterative algorithm, in which the error barrier u narrows the feasible region with the iteration of the inner point once the initial point is selected to lie within the feasible region in Fig.2. When the feasible region is sufficiently small, the internal point can represent the optimal solution, $\boldsymbol{\beta}^n$, in Eq.(14).

The error of the proposed method is mainly attributed to the Schmidt orthogonalization, when the optimal solution is found. The rotational matrix is an identity matrix, with its inherent orthogonal constraints in nonlinear form, the constraint is acquired via Schmidt orthogonalization, which may result in a slight increase, $\boldsymbol{\epsilon}_{ortho}$, of the rotational errors of $\mathbf{X}$ and $\mathbf{Y}$.

The error of the iterative algorithm is mainly attributed to the approximative estimation, $\mathbf{R}_{q,true} \cdot \mathbf{R}_{q,init}^{-1} \approx \mathbf{I} + \hat{\boldsymbol{\delta}}_{R,q}$, in Eq.(16), which results in a truncation error as:

$$\begin{aligned}\boldsymbol{\epsilon}_{tr} = \mathbf{R}_A \cdot \left(\sum_{i=1}^{\infty} \left(\hat{\boldsymbol{\delta}}_{\mathbf{R}_X} \right)^n \Big/ n \right) \cdot \mathbf{R}_{X,init} \\ - \left(\sum_{i=1}^{\infty} \left(\hat{\boldsymbol{\delta}}_{\mathbf{R}_Y} \right)^n \Big/ n \right) \cdot \mathbf{R}_{Y,init} \cdot \mathbf{R}_B \end{aligned} \tag{27}$$

In addition, the step-by-step calibration of the rotational and translational components in the iterative algorithm will produce error accumulation and propagation. The error, $\hat{\boldsymbol{\delta}}_{R,q}$, is assumed to lie between the calculated and the true values of the rotation matrix in the first step as:

$$\mathbf{R}_{q,cal} = \mathbf{R}_{q,true} \cdot \left(\mathbf{I} + \hat{\boldsymbol{\delta}}_{R,q} \right) \tag{28}$$

And the translational error, $\boldsymbol{\epsilon}_{acc}$, caused by the error accumulation of the rotational error component, is expressed as:

$$\boldsymbol{\epsilon}_{acc} = \begin{bmatrix} \mathbf{R}_A^3 + \mathbf{R}_A \\ \mathbf{R}_A^2 \end{bmatrix} \cdot \mathbf{R}_{Y,true} \cdot \hat{\boldsymbol{\delta}}_{\mathbf{R}_Y} \cdot \mathbf{t}_B \tag{29}$$

The solving accuracy of the *DQ*-based algorithm depends on the smallest singular value of the regression matrix when *SVD*-based algorithm is employed on Eq.(24). When the noise levels are not considered, the solution of Eq.(24) corresponds to the null space of the coefficient matrix $\tilde{\mathbf{D}}$, and can be obtained via linear combination of the last two columns, $\mathbf{v}_{15}$ and $\mathbf{v}_{16}$, in $\mathbf{V}$. Accordingly, the minimum singular values of the matrix Σ are equal to zero, i.e., rank($\tilde{\mathbf{D}}$) = **14**. However, when noise is injected into the dataset, the minimum singular values are not equal to zero, and this results in an erroneous solution produced by $\mathbf{v}_{15}$ and $\mathbf{v}_{16}$.

IV. Verification and Discussions

To validate the robustness and efficacy of the proposed simultaneous calibration methodology for the hand-eye and robot-world transformation relationship, the results obtained using the proposed Kronecker product and LMI-SDP-based methodology, are compared to those produced using two existing iterative and *DQ*-based algorithms in Section III, respectively, different noise levels and simulated datasets are also accounted for in this comparative analysis. An experimental analysis is subsequently conducted, for the purpose of acquiring the calibration results pertaining to the practical **AX=YB** problem. The accompanying computer programs have been developed in MATLAB 9.5, to facilitate an analysis of the calibration algorithm results.

A. Simulation Analysis

The simulation analysis mainly entails two scenarios, i.e., calibration under different noise levels and different datasets. In the simulation procedure, the calibration datasets are generated using the ground values of camera-tool ($\mathbf{X}$), and world-robot ($\mathbf{Y}$), which are given as follows:

$$\mathbf{X}_{true} = \begin{bmatrix} 0.0162 & 0.9995 & 0.0289 & 22 \\ -0.9996 & 0.0168 & -0.0207 & -3 \\ -0.0211 & -0.0285 & 0.9994 & -60 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{Y}_{true} = \begin{bmatrix} -0.9990 & -0.0327 & 0.0279 & 165 \\ 0.0274 & 0.0155 & 0.9995 & 300 \\ -0.0331 & 0.9994 & -0.0146 & -962 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

And the sets $(\mathbf{A}, \mathbf{B})$ with noise level are generated via the ground values for purpose of the simulation, to verify the robustness and accuracy of the proposed method by calculating the error between ground values ($\mathbf{X}_{true}$, $\mathbf{Y}_{true}$) and the calibrated values ($\mathbf{X}_{cal}$, $\mathbf{Y}_{cal}$). The solving of $\mathbf{X}_{cal}$ and $\mathbf{Y}_{cal}$ is achieved via the proposed method, the iterative method, and the *DQ*-based method according to the following steps:

Step 1: Generate calibration dataset $(\mathbf{A}_i, \mathbf{B}_i)$: the matrix $\mathbf{A}_i$ is generated via robot kinematics using the given random joint values, and the matrix $\mathbf{B}_i$ is calculated via the formula $\mathbf{B} = \mathbf{Y}_{true}^{-1} \cdot \mathbf{A} \cdot \mathbf{X}_{true}$;

Step 2: Introduce the dataset noise: different levels of data noise are injected into the $(\mathbf{A}_i, \mathbf{B}_i)$ and illustrated for the analysis of different noise levels and data pairs;

Step 3: Obtain the calibrated values of $\mathbf{X}_{cal}$ and $\mathbf{Y}_{cal}$: the comparisons are performed by the three methods in Section III using generated dataset $(\mathbf{A}_i, \mathbf{B}_i)$.

Step 4: Evaluate the calibrated results: the rotation error, $err_{\mathbf{R},k}$, and translation error, $err_{\mathbf{t},k}$, of the calibrated results,

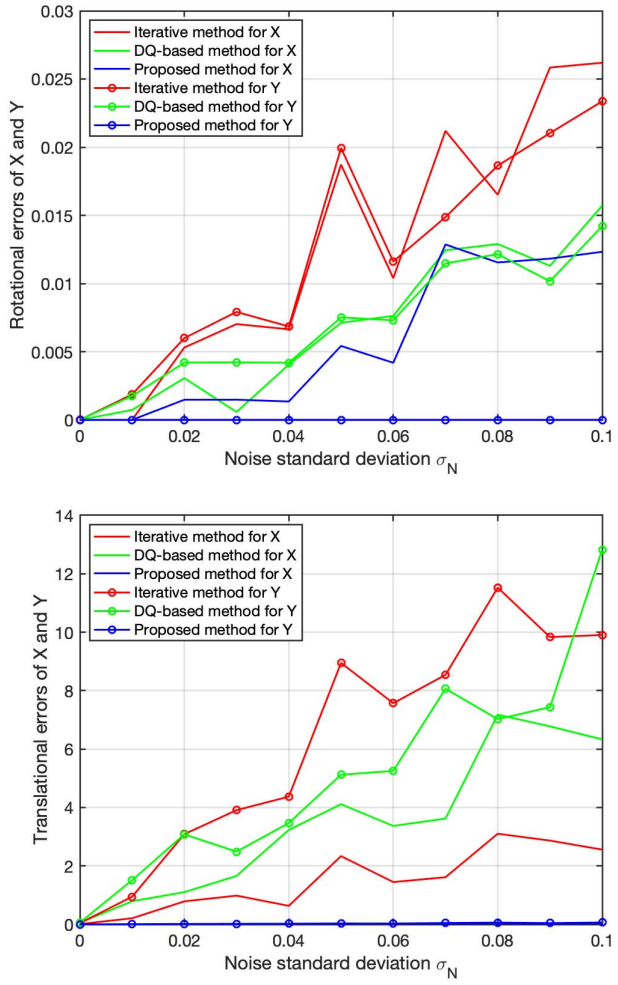

Fig. 3. Rotational and translational errors of calibrated results under different noise levels.

$\mathbf{X}$ and $\mathbf{Y}$ by means of Eqs.(30) and (31).

$$err_{\mathbf{R},k} = \theta(\mathbf{R}_{cal,k} \cdot \mathbf{R}_{true,k}^{-1}) \tag{30}$$

$$err_{\mathbf{t},k} = \|\mathbf{t}_{cal,k} - \mathbf{t}_{true,k}\| \tag{31}$$

where $k = X, Y$.

To analyze the effects of different noise levels on the calibrated values, the m data pair $(\mathbf{A}_i, \mathbf{B}_i)$ is injected with the different noise levels of variance $\sigma_N \sim u(0, 0.1)$, with 0.01 intervals. The calibrated values, $\mathbf{X}$, $\mathbf{Y}$, are calculated respectively by means of the proposed method, the iterative method, and the *DQ*-based method. It is noted that the calibration errors are produced at each noise level, and the mean errors are computed using Eqs.(30) and (31), based on the solutions generated over the five simulation cycles. The mean rotational errors and translational errors of the three methods are plotted in Fig.3, under different noise levels. Fig.3 shows that the calibrated variables, $\mathbf{X}$, $\mathbf{Y}$, can be ultimately obtained, although they are very sensitive to the noise levels. The rotational and translational errors of the calibrated values increase as a function of the noise levels, and remain close to zero in the absence of noise. Based on the comparison of the computed errors, the proposed method exhibits improved performance over, and higher accuracy than iterative and *DQ*-based methods, especially for increased noise magnitudes. Besides, the rotational and translational errors of the variable $\mathbf{Y}$, as well as the translational errors of the variable $\mathbf{X}$, are insensitive to the noise levels and remain close to zero. The phenomena observed in the simulation results, according to the error analysis in Section III D, can be summarized as follows: 1) The approximation in Eq.(15) and the separable solving of rotational and translational components for the variables $\mathbf{X}$, $\mathbf{Y}$, are adapted using the iterative method, which inevitably cause certain inaccuracy; 2) The *DQ*-based approach calibrates $\mathbf{X}$ and $\mathbf{Y}$ simultaneously with the *SVD* technique, which is greatly affected by the smallest singular value of the coefficient matrix $\tilde{\mathbf{D}}$ in Eq.(24).

TABLE I
AVERAGE COMPUTATIONAL TIMES RECORDED AT EACH LOOP

Methods	Different noise level	Different datasets
Proposed method	0.451s	0.496s
Iterative method	0.011s	0.021s
DQ-based method	0.010s	0.012s

Furthermore, to investigate the effects of the data pairs on the calibration accuracy of the proposed method, the calibrated errors are calculated in an analogous manner. The total number of data pairs is set to m, with m lying in the range of $10 \sim 100$ with 10 intervals. A noise level of $\sigma_N = 0.05$ is injected into the dataset $(\mathbf{A}_i, \mathbf{B}_i)$. The calibrated results of the unknown variables, $\mathbf{X}$, $\mathbf{Y}$ are obtained, and subsequently compared to the iterative and *DQ*-based methods. The mean rotational and translational errors of the unknown variables, $\mathbf{X}$, $\mathbf{Y}$, are computed respectively via the three methods and are plotted in Fig.3. It is clear from Fig.3 that all three methods are capable of finding the calibrated $\mathbf{X}$, $\mathbf{Y}$ values, using different data pairs, where the rotational and translational errors are inversely proportional to the number of data pairs. When the amount of data pairs increases, the proposed approach exhibits higher performance levels as compared to the other two methods, in terms of the rotational and translational components accuracy. Moreover, SDP is in essence an iterative algorithm for the solving process, and $\mathbf{Y}$ converges to the ground value more rapidly; the effect of noise levels on the calibrated results mainly relates to $\mathbf{X}$, as the calibrated error of $\mathbf{Y}$ is extremely small.

The respective computational times of the three methods with various noise levels and data sets are given in Table I. It is seen that the solving speed using the *DQ*-based method is significantly faster, since the *DQ*-based method revolves around an analytical solving process, and the solution can be acquired immediately by symbolic calculation. Contrarily, the proposed LMI-SDP-based and iterative approaches are iterative solving processes that require multiple loops to find the optimal solutions. Moreover, the proposed LMI-SDP algorithm with the PyLMI-SDP toolbox is implemented in Python, in combination with additional packages, and this significantly increases the execution time required for finding the solutions.

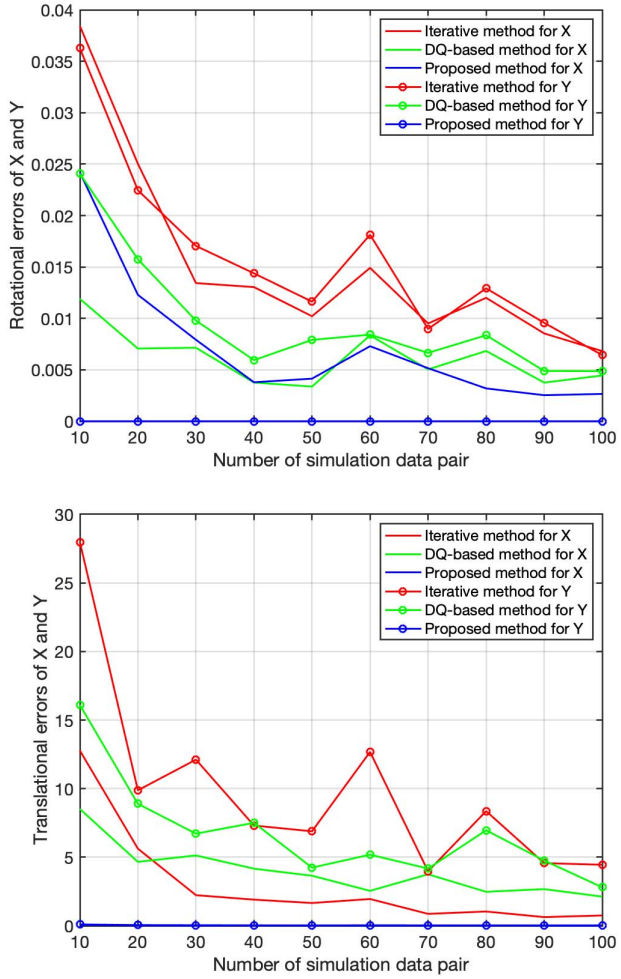

Fig. 4. Rotational and translational errors of calibrated results under different number of data pairs.

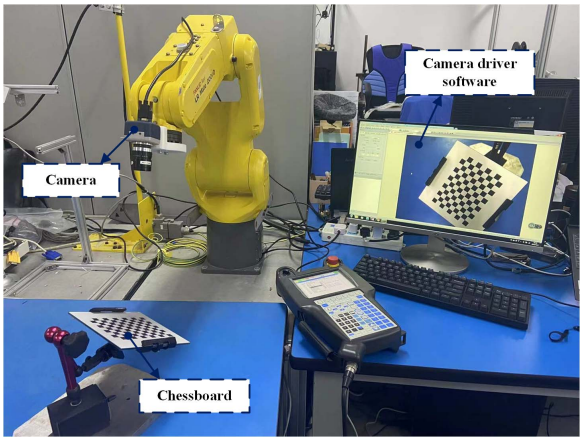

Fig. 5. Experimental setup.

B. Experimental Analysis

In order to further verify the effectiveness of the proposed methodology, the calibration experiment is implemented and analyzed. Fig.4 shows the experimental setup, which involves a Fanuc LR Mate 200iD robot, a SUNTAE 2D camera and a 12 × 9 chessboard with a square size of 10mm. The camera

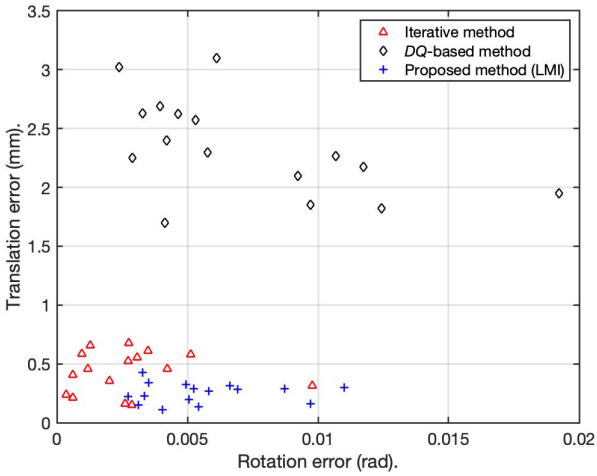

Fig. 6. Rotational and translational errors of equation in calibration experiment.

is mounted onto the robot's end-flange, while the chessboard is fixed onto the workstation at a certain inclination. The data acquisition and calibration code are running on an Intel Core i7-10700@2.9 GHz computer. The whole calibration process follows the procedures listed below:

Step 1: The Mate 200iD robot equipped with the camera is moved to one configuration such that the chessboard is about 500mm away from the camera, and is within the camera's field of view (FOV). Meanwhile, the configuration data of the robot's end-flange and the chessboard images are recorded accordingly.

Step 2: Change the robot's configuration to acquire chessboard images from different viewpoints, collect data according to *Step 1*, and repeat the above process until 16 sets of data are collected.

On the basis of the data collected in *Step 2*, the acquired 16 images will be transformed into a HTM of the camera frame $\{C\}$, with respect to the chessboard frame $\{W\}$, by means of the MATLAB toolbox named Camera Calibrator. The generated 16 sets of data pairs $(\mathbf{A}_i, \mathbf{B}_i)$ will be adapted to calibrate the relationships of the camera-tool $(\mathbf{X})$ and world-robot $(\mathbf{Y})$ via three methods in Section III. The errors for each set of data pairs, $(\mathbf{A}_i, \mathbf{B}_i)$, are calculated using the formula:

$$err_{\mathbf{R}_i} = \theta(\mathbf{R}_{\mathbf{A}_i\mathbf{X}_{cal}} \cdot \mathbf{R}^{-1}_{\mathbf{Y}_{cal}\mathbf{B}_i}) \tag{32}$$

$$err_{\mathbf{t}_i} = \|\mathbf{t}_{\mathbf{A}_i\mathbf{X}_{cal}} - \mathbf{t}_{\mathbf{Y}_{cal}\mathbf{B}_i}\| \tag{33}$$

where $\mathbf{X}_{cal}$ and $\mathbf{Y}_{cal}$ are the calibrated values, while $\mathbf{R}(\cdot)$ and $\mathbf{t}(\cdot)$ represent rotational and translational components.

The error distribution of the calibration experiment is plotted in Fig.6. In accordance with the calibrated results, the rotational errors $Err_{\mathbf{R}}$ of the iterative method, *DQ*-based method and the proposed method are 0.0027rad, 0.0072rad, and 0.0056rad, respectively, and the corresponding translational errors, $Err_{\mathbf{t}}$, are 0.4330mm, 2.3398mm and 0.2529mm, respectively. It is clearly demonstrated that the proposed method, and the iterative one, are superior in terms of the calibration accuracy of the camera-tool $(\mathbf{X})$ and world-robot $(\mathbf{Y})$. Meanwhile, the proposed method is superior to the iterative one in terms of the accuracy of the translational component,

while the iterative one performs better with respect to the rotational component.

The following phenomenon is observed: the simultaneous calibration of the translational and rotational components for the camera-tool ($\mathbf{X}$) and world-robot ($\mathbf{Y}$), occur in the *DQ*-based and proposed methods, in which the error accumulation and propagation can be avoided, while the iterative method deal with this separately. Furthermore, the *DQ*-based method utilized *SVD* to solve the calibration equation directly, in which the solving accuracy is affected dramatically by smallest singular value of the coefficient matrix $\tilde{\mathbf{D}}$ in Eq.(24). The proposed method, i.e., the LMI-SDP-based algorithm, is essentially a nonlinear process, whose optimal solution is found in a small feasible space via continuous minimization of the target variable (error variable u) in Eq.(14), which means that the accuracy can remain increased even in higher dimensions of feasible solution. However, Schmidt orthogonalization of the rotational component on the optimal solution is utilized, and this diminishes the rotational component's accuracy, as compared to the iterative method.

V. Conclusion

This work presents a novel and generic algorithm using an optimized technique, i.e., LMI-SDP, for the transformation calibration of the hand-eye ($\mathbf{X}$) and world-robot ($\mathbf{Y}$) relationships involved in the $\mathbf{AX{=}YB}$ equation. The effectiveness has been verified by means of simulation and experimental analysis. The main contribution of this work can be summarized as follows:

(1) The LMI-SDP technique with Kronecker product is applied initially to the coordinate calibration of the $\mathbf{AX{=}YB}$ problem, which realizes the simultaneous calibration of $\mathbf{X}$ and $\mathbf{Y}$, as well as their rotational and translational components.

(2) Simulation analysis reveals that the proposed LMI-SDP-based method, compared with the iterative and *DQ*-based algorithms, can provide higher accuracy and robustness performance, even at high-noise levels.

(3) The effectiveness of the proposed algorithm has been verified on a practical visually-guided Fanuc LR Mate 200iD robot system. The average errors can reach 0.0056rad in the rotational component, and 0.2529mm in the translational component.

In addition, the combined Kronecker product and LMI-SDP-based algorithm exhibits good performance overall, as far as single robots with visual sensors are concerned. However, the coordinate calibration of multi-robot systems via heterogeneous transformation matrix [28], as well as the weighting of the rotational and translational component errors, need to be studied in our future work.

References

[1] S. Shirmohammadi and A. Ferrero, "Camera as the instrument: The rising trend of vision based measurement," *IEEE Instrum. Meas. Mag.*, vol. 17, no. 3, pp. 41–47, Jun. 2014.

[2] I. Enebuse, M. Foo, B. S. K. K. Ibrahim, H. Ahmed, F. Supmak, and O. S. Eyobu, "A comparative review of hand-eye calibration techniques for vision guided robots," *IEEE Access*, vol. 9, pp. 113143–113155, 2021.

[3] X. Zhi and S. Schwertfeger, "Simultaneous hand-eye calibration and reconstruction," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1470–1477.

[4] 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.*, vol. 5, no. 1, pp. 16–29, Dec. 1989.

[5] 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.*, vol. 10, no. 3, pp. 240–254, Jun. 1991.

[6] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans. Robot. Autom.*, vol. 5, no. 3, pp. 345–358, Jun. 1989.

[7] K. Daniilidis, "Hand-eye calibration using dual quaternions," *Int. J. Robot. Res.*, vol. 18, no. 3, pp. 286–298, Mar. 1999.

[8] F. C. Park and B. J. Martin, "Robot sensor calibration: Solving AX=XB on the Euclidean group," *IEEE Trans. Robot. Autom.*, vol. 10, no. 5, pp. 717–721, Oct. 1994.

[9] E. Bayro-Corrochano, K. Daniilidis, and G. Sommer, "Motor algebra for 3D kinematics: The case of the hand-eye calibration," *J. Math. Imag. Vis.*, vol. 13, no. 2, pp. 79–100, 2000.

[10] Z. Zhao and Y. Liu, "A hand-eye calibration algorithm based on screw motions," *Robotica*, vol. 27, no. 2, pp. 217–223, Mar. 2009.

[11] Z. Zhao, "Hand-eye calibration using convex optimization," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 2947–2952.

[12] Q. Ma, H. Li, and G. S. Chirikjian, "New probabilistic approaches to the AX =XB hand-eye calibration without correspondence," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Jun. 2016, pp. 4365–4371.

[13] X. Hu, D. Olesen, and K. Per, "A novel robust approach for correspondence-free extrinsic calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1–6.

[14] G. Wang *et al.*, "Simultaneous calibration of multicoordinates for a dual-robot system by solving the AXB=YCZ problem," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1172–1185, Aug. 2021.

[15] Z. Liu, X. Liu, G. Duan, and J. Tan, "Precise hand-eye calibration method based on spatial distance and epipolar constraints," *Robot. Auton. Syst.*, vol. 145, Nov. 2021, Art. no. 103868.

[16] M. Shah, "Solving the robot-world/hand-eye calibration problem using the Kronecker product," *J. Mech. Robot.*, vol. 5, no. 3, Aug. 2013, Art. no. 031007.

[17] Z. Zhao, "Simultaneous robot-world and hand-eye calibration by the alternative linear programming," *Pattern Recog. Lett.*, vol. 127, pp. 174–180, Nov. 2019.

[18] F. Ernst *et al.*, "Non-orthogonal tool/flange and robot/world calibration," *Int. J. Med. Robot. Comput. Assist. Surg.*, vol. 8, no. 4, pp. 407–420, 2012.

[19] W. Wang, F. Liu, and C. Yun, "Calibration method of robot base frame using unit quaternion form," *Precis. Eng.*, vol. 41, pp. 47–54, Jul. 2015.

[20] Q. Ma, Z. Goh, S. Ruan, and G. S. Chirikjian, "Probabilistic approaches to the AXB= YCZ calibration problem in multi-robot systems," *Auton. Robot.*, vol. 42, no. 7, pp. 1497–1520, Apr. 2018.

[21] L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q. Meng, "Simultaneous hand-eye, tool-flange, and robot-robot calibration for comanipulation by solving the $AXB = YCZ$ problem," *IEEE Trans. Robot.*, vol. 32, no. 2, pp. 413–428, Mar. 2016.

[22] J. Wang, L. Wu, M. Q.-H. Meng, and H. Ren, "Towards simultaneous coordinate calibrations for cooperative multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2014, pp. 410–415.

[23] 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. Let.*, vol. 5, no. 3, pp. 4086–4093, Apr. 2020.

[24] I. Ali, O. Suominen, A. Gotchev, and E. R. Morales, "Methods for simultaneous robot-world–hand-eye calibration: A comparative study," *Sensors*, vol. 19, no. 12, p. 2837, Jun. 2019.

[25] C. D. Sousa and R. Cortesão, "Physical feasibility of robot base inertial parameter identification: A linear matrix inequality approach," *Int. J. Robot. Res.*, vol. 33, no. 6, pp. 931–944, 2014.

[26] Z. Fu *et al.*, "A lie-theory-based dynamic parameter identification methodology for serial manipulators," *IEEE/ASME Trans. Mechatronics*, vol. 26, no. 5, pp. 2688–2699, Oct. 2021.

[27] S. J. Benson and Y. Ye, "Algorithm 875: DSDP5—Software for semidefinite programming," *ACM Trans. Math. Softw.*, vol. 34, no. 3, pp. 1–20, May 2008.

[28] B. Wang, W. Chen, B. Zhang, P. Shi, and H. Zhang, "A nonlinear observer-based approach to robust cooperative tracking for heterogeneous spacecraft attitude control and formation applications," *IEEE Trans. Autom. Control*, early access, Jan. 13, 2022, doi: 10.1109/TAC.2022.3143082.

Jiabin Pan received the B.Sc. degree in mechanical engineering from the Wuhan Institute of Technology (WIT) in June 2019. He is currently pursing the M.Sc. degree in mechatronics engineering and automation with Shanghai University (SHU). His main research is in the fields of robotics and control, including robot kinematics and dynamics, nonlinear control systems, robotic interaction control, and compliant parallel mechanism.

Zhongtao Fu received the B.Sc. degree in mechanical engineering from the Henan University of Science and Technology (HAUST) in June 2010 and the Ph.D. degree in mechatronic engineering from the Huazhong University of Science and Technology (HUST) in March 2016. He worked as a Visiting Research Fellow with the Department of Engineering, King's College London (KCL), from December 2018 to December 2020. He is currently an Associate Professor with the School of Mechanical and Electrical Engineering, Wuhan Institute of Technology (WIT). His research interests include kinematics and dynamics, flexible-joint systems, multirobot collaboration, and ultrasound scanning.

Hengtao Yue received the B.Sc. degree in mechanical engineering from the Xian University of Architecture and Technology (XAUAT) in June 2019. He is currently pursuing the M.Sc. degree in mechatronics engineering with the Wuhan Institute of Technology (WIT). His research interests include multi-sensor fusion and robot control.

Xiaoyu Lei received the B.Sc. degree in mechanical engineering from the Wuhan Institute of Technology (WIT) in June 2020, where he is currently pursuing the M.Sc. degree in mechatronics engineering. His research interests include robot kinematics, path planning, and ROS-based software.

Miao Li (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology (HUST), Wuhan, China, in 2008 and 2011, respectively, and the Ph.D. degree from the Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland, in 2016. His research interests include robotic grasping and manipulation, robot learning, tactile sensing, and medical robots.

Xubing Chen (Member, IEEE) received the B.Sc. and Ph.D. degrees in material science and engineering from the Huazhong University of Science and Technology (HUST) in 1995 and 2000, respectively. He is currently a Professor with the School of Mechanical and Electrical Engineering, Wuhan Institute of Technology (WIT). His research interests include trajectory generation, multi-robot collaboration, and manufacturing.