2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Coordinate Calibration of a Dual-Arm Robot System by Visual Tool Tracking

Junlei Hu, Dominic Jones, *Member, IEEE*, Pietro Valdastri, *Fellow, IEEE*

***Abstract*— The calibration of a vision-guided dual-arm robotic system, including the robot-robot and hand-eye calibration, requires the tracked positions of markers in different postures. However, in many cases, using markers to calibrate is impractical. Only some markerless features can be obtained rather than the rigid transform matrix; for example, the shaft of a markerless robotic tool can be tracked. Therefore, we proposed a Kronecker-Product-based method to calibrate the dual-arm system with a tracked robotic tool by decoupling the translation and rotation. The simulation and experiment results on a *da Vinci* Research Kit show that the proposed method is robust and accurate under different noise levels and various sample robot movements, compared with two state-of-the-art methods for dual-arm calibration with complete homogeneous transformations.**

I. Introduction

Robotic assisted surgical systems rely heavily on co-manipulation between multiple robots to perform highly dexterous tasks. Currently, a surgeon teleoperates the end-effectors with an innate sense of proprioception, allowing for effective bimanual control based on vision alone. As the systems tend to become more autonomous, the operating surgeon is removed from the control loop, requiring new calibration methods to understand the coordinate relationship between the manipulators by utilising the available visual data. In a typical dual-arm system, the sensor (e.g. a camera) is fixed at the end-effector of Robot 1, with fiducial markers placed on Robot 2. This leaves three calibrations that must be explored: Hand 2 to eye and robot to robot (Fig. 1).

The dual-arm or multi-arm calibration can be roughly classified into two main types: simultaneous and non-simultaneous. The non-simultaneous method is solved sequentially with two discrete steps. First, Robot 1 is fixed while Robot 2 is moved around the visible workspace, allowing for the hand-eye problem to be solved in an $\mathbf{AX} = \mathbf{XB}$ manner. Second, the state of the robots is inverted to solve the robot-world and tool-flange calibration in the $\mathbf{AX} = \mathbf{YB}$ manner [1].

*This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under grants number EP/R045291/1 and EP/V047914/1, by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation program (grant agreement No. 818045) and by the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No. 952118 (AUTOCAPSULE). Any opinions, findings, and conclusions, or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the EPSRC, ERC or the EC.

The authors are all with the STORM Lab UK, School of Electronic and Electrical Engineering, University of Leeds, UK. {eljh, d.p.jones, p.valdastri}@leeds.ac.uk

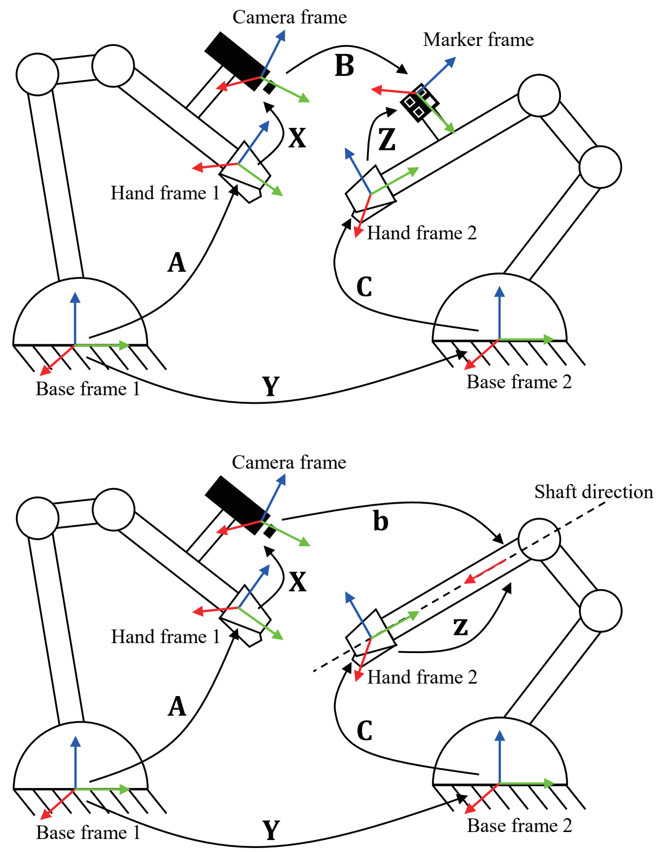

Fig. 1. The illustration of dual-arm calibration problem. Upper: $\mathbf{AXB} = \mathbf{YCZ}$ problem, lower: $\mathbf{AXb} = \mathbf{YCz}$ problem. Both of them are calculating the hand-eye ($\mathbf{X}$), robot-robot ($\mathbf{Y}$) calibration. $\mathbf{A}$ and $\mathbf{C}$ are forward kinematics of both two robots in both configurations. $\mathbf{B}$ in the upper configuration is the transform matrix representing the pose of the marker under the camera frame. $\mathbf{b}$ in lower configuration is the direction of the robotic tool in camera coordinate. $\mathbf{Z}$ (maker to hand transformation) and $\mathbf{z}$(shaft direction to hand vector) are unknown parameter.

The $\mathbf{AX} = \mathbf{XB}$ is a well-recognised problem for the eye-in-hand configuration, where $\mathbf{A}$ and $\mathbf{B}$ are the rigid transformation matrices of end-effector and camera movements, respectively, and $\mathbf{X}$ is the unknown hand-eye relationship. It has been well studied since Shiu *et al.* first proposed the method [2]. Various subsequent solutions were proposed: a closed-form solution [3], geometrical interpretation [4], orthogonal dual tensor [5] and probabilistic method [6], [7]. The $\mathbf{AX} = \mathbf{YB}$ problem is extension of $\mathbf{AX} = \mathbf{XB}$, where $\mathbf{Y}$ is the robot-world transformation. Similar solutions of the problem exist, such as probabilistic method [8], dual Lie algebra [9], Kronecker product [10], and nonlinear optimization [11], [12]. Therefore, Fu *et al.* proposed a dual-quaternion-based method [13] which calculates the eye-hand matrix first.

In simultaneous methods, both robots move simultaneously, with all unknown relationships being calculated at once. Compared to non-simultaneous methods [13], simultaneous solutions [14] offer a more efficient solution utilising the dual-quaternion method. Ma *et al.* proposed a probabilistic approach to solve the problem without temporal correspondence of the data streams [15]. Wang *et al.* solved the problem simultaneously using an iterative method, whereby an initial estimation of the unknown matrices is made before refining the estimate to the optimal value [16]. In contrast, a closed-form solution based on the Kronecker product provides superior accuracy and efficiency in comparison to conventional methods [17].

In robot-assisted minimally invasive surgery, the camera and end-effectors are deployed within the body across the abdominal wall. This generates a constrained working area of the tools, prohibiting the use of visual markers for inference of the eye-hand relationship. Richter *et al.* [18] propose a solution by visually identifying the tool-tip transformation through visual assessment of the tool-tip of a *da Vinci* patient side manipulator. Using visual feed from the endoscope, the pose of the central axis of the tool shaft is identified before identifying the jaw pose through analysis of key-point positions on the jaw.

In this work, we propose a method of coordinate calibration for a dual-arm robot system by visual tracking of a robotic tool, including robot-robot and hand-eye calibration. The contributions of this work may be summarised as follows:

- A marker-less calibration problem for the dual-arm robot by a tracked robotic tool is modelled and formulated as an $\mathbf{AXb} = \mathbf{YCz}$ problem for the first time (to the best of our knowledge).
- A simultaneous method by decoupling the rotation and translation to solve the calibration problem.
- A comparative study with two state-of-the-art methods for dual-arm calibration with full rigid transformations.

The paper is organized in the following manner: Section II models and formulates the calibration problem, followed by the proposed solutions, mathematical interpretations and algorithms. We further compare the simulation and experiment with other methods, discuss the results in Section III, and draw final conclusions in Section IV.

II. Methods

A. Problem Formulation

The scenario for the calibration problem is presented in Fig.1. Robot 1 is equipped with a camera, and Robot 2 is equipped with a fiducial marker, where all system configurations ensure the marker is located within the camera's visual field. To fully define the transformation between the robot end-effectors, the relationships between the robot base frames and between the end-effector and the camera must be calculated to enable accurate cooperation of the manipulators. The end effector position is varied on both robots, while kinematic position data and the camera's inferred pose are recorded. The problem is defined as follows:

$$\mathbf{AXb} = \mathbf{YCz} \tag{1}$$

where $\mathbf{A}$, $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{C}$ are 4×4 rigid body transform matrices, and $\mathbf{b}$ and $\mathbf{z}$ are 4×1 homogeneous vectors. (Note: $\mathbf{b}$ and $\mathbf{z}$ are not the translational part, in the following sections $\mathbf{b}$ and $\mathbf{z}$ are simplified as the 3×1 vectors with the first three elements) Given m measurements, $\mathcal{A} = \{\mathbf{A}_i\}$, $\mathcal{B} = \{\mathbf{b}_i\}$, $\mathcal{C} = \{\mathbf{C}_i\}$, $i = 1, 2, \ldots, m$, to compute the unknowns $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{z}$.

B. Solving for Rotation

The Kronecker product of an $m \times n$ matrix $\mathbf{A}$, whose ijth element is a_{ij} for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, with a $p \times q$ matrix $\mathbf{B}$, is defined as the $mp \times nq$ matrix

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \in \mathbb{R}^{mp \times nq}, \tag{2}$$

and $\operatorname{vec}(\mathbf{A})$ is defined to be the mn-vector formed by stacking the columns of the $m \times n$ matrix $\mathbf{A}$.

The rotational part of the linear system in Eq.(1) can be expressed in Kronecker Product form and written in the following manner:

$$(\mathbf{b}^\intercal \otimes \mathbf{R}_A) \operatorname{vec}(\mathbf{R}_X) = (\mathbf{z}^\intercal \otimes \mathbf{R}_Y) \operatorname{vec}(\mathbf{R}_C) \tag{3}$$

where $\mathbf{R}_A$, $\mathbf{R}_C$, $\mathbf{R}_X$ and $\mathbf{R}_Y$ are the rotational part of $\mathbf{A}$, $\mathbf{C}$, $\mathbf{X}$ and $\mathbf{Y}$. In order to construct a linear system and decouple the known and unknown variables, the right side of Eq.(3) can be expressed as

$$(\mathbf{z}^\intercal \otimes \mathbf{R}_Y) \operatorname{vec}(\mathbf{R}_C) = (\operatorname{vec}(\mathbf{R}_C) \otimes \mathbf{I}_3)^\intercal \operatorname{vec}(\mathbf{z}^\intercal \otimes \mathbf{R}_Y) \tag{4}$$

where $\mathbf{I}_9$ is the 9×9 identity matrix. Hence, the Kronecker-product form equation is

$$(\mathbf{b}^\intercal \otimes \mathbf{R}_A) \operatorname{vec}(\mathbf{R}_X) = (\operatorname{vec}(\mathbf{R}_C) \otimes \mathbf{I}_3)^\intercal \operatorname{vec}(\mathbf{z}^\intercal \otimes \mathbf{R}_Y). \tag{5}$$

Rewrite it into the form

$$\mathbf{W}_{AbC}\mathbf{v}_{XYz} = \mathbf{0} \tag{6}$$

where

$$\mathbf{W}_{AbC} = \left[\; (\mathbf{b}^\intercal \otimes \mathbf{R}_A) \quad -(\operatorname{vec}(\mathbf{R}_C) \otimes \mathbf{I}_3)^\intercal \;\right] \in \mathbb{R}^{3 \times 36}, \tag{7}$$

$$\mathbf{v}_{XYZ} = \begin{bmatrix} \operatorname{vec}(\mathbf{R}_X) \\ \operatorname{vec}(\mathbf{z}^\intercal \otimes \mathbf{R}_Y) \end{bmatrix} \in \mathbb{R}^{36}. \tag{8}$$

If m different independent measurements are provided, these equations can be combined as an over-determined equation:

$$\tilde{\mathbf{W}}_{AbC}\mathbf{v}_{XYz} = \mathbf{0} \tag{9}$$

where

$$\tilde{\mathbf{W}}_{AbC} = \left[\mathbf{W}_{AbC1}^\intercal \quad \mathbf{W}_{AbC2}^\intercal \quad \cdots \quad \mathbf{W}_{AbCm}^\intercal\right]^\intercal \in \mathbb{R}^{3m \times 36}. \tag{10}$$

According to the min-max theorem and the fact that the norm of $\mathbf{v}_{XYz}$ is constant ($\|\mathbf{v}_{XYz}\| = \sqrt{6}$, see Appendix), the solution $\mathbf{v}_{XYz}$ that minimizes (9) is parallel to eigenvector $\hat{\mathbf{v}}_{\text{eig}}$ of $\tilde{\mathbf{W}}_{AbC}^{\intercal}\tilde{\mathbf{W}}_{AbC}$ associated with the minimum eigenvalue. Thus, the relationship between $\hat{\mathbf{v}}_{\text{eig}}$ and $\hat{\mathbf{R}}_X$ is as following:

$$\hat{\mathbf{v}}_{\text{eig}} = \frac{1}{\sqrt{6}} \begin{bmatrix} \text{vec}\left(\hat{\mathbf{R}}_X\right) \\ \text{vec}\left(\hat{\mathbf{z}} \otimes \hat{\mathbf{R}}_Y^{\intercal}\right) \end{bmatrix}. \tag{11}$$

Since $\mathbf{R}_X \in \text{SO}(3)$, its determinant is $+1$. Although the sign of $\hat{\mathbf{v}}_{\text{eig}}$ is ambiguous, the $\mathbf{R}_X$ is unique. Generally, the solution $\hat{\mathbf{R}}_X$ is not a rotation matrix because of the existence of noise. There are procedures for projecting $\mathbf{R}_X$ back into the group $SO(3)$ to result in $\mathbf{R}_X$:

$$\mathbf{R}_X = \text{sgn}\left(\det\left(\mathbf{U}_X\mathbf{V}_X^{\intercal}\right)\right)\mathbf{U}_X\mathbf{V}_X^{\intercal} \tag{12}$$

where $\mathbf{U}_X$ and $\mathbf{V}_X$ are unitary matrices in singular value decomposition (SVD):

$$\hat{\mathbf{R}}_X = \mathbf{U}_X\mathbf{\Sigma}_X\mathbf{V}_X^{\intercal}. \tag{13}$$

The estimated $\hat{\mathbf{z}}$ is

$$\hat{\mathbf{z}} = \sqrt{2}\left[\|\left.\hat{\mathbf{v}}_{\text{eig}}\right|_{(10:18)}\| \quad \|\left.\hat{\mathbf{v}}_{\text{eig}}\right|_{(19:27)}\| \quad \|\left.\hat{\mathbf{v}}_{\text{eig}}\right|_{(28:36)}\|\right] \tag{14}$$

where $\left.\hat{\mathbf{v}}_{\text{eig}}\right|_{(i:j)}$ is a vector whose elements consist of the i-th through j-th elements of vector $\hat{\mathbf{v}}_{\text{eig}}$. The calculated $\mathbf{z}$ should be normalized as

$$\mathbf{z} = \frac{\hat{\mathbf{z}}}{\|\hat{\mathbf{z}}\|}. \tag{15}$$

The estimated

$$\hat{\mathbf{R}}_Y = \frac{\sqrt{6}}{\left.\mathbf{z}\right|_1}\text{vec}^{-1}\left(\left.\hat{\mathbf{v}}_{\text{eig}}\right|_{(10:18)}\right) \tag{16}$$

where $\left.\mathbf{z}\right|_1$ is the first element of the vector $\mathbf{z}$ and $\text{vec}^{-1}(\cdot)$ is the inverse operator of $\text{vec}(\cdot)$ should be orthogonalized with SVD method like $\mathbf{R}_X$ in (12) and (13).

C. Solving for Translation

In order to compute the translation of the matrices $\mathbf{X}$ and $\mathbf{Y}$, the translational parts of $\mathbf{b}$ and $\mathbf{z}$ ($\mathbf{t}_B$ and $\mathbf{t}_z$) would be added. (Note: $\mathbf{t}_B$ and $\mathbf{t}_z$ are introduced solely for computational purposes, and their meaning within robotic systems may not be clearly defined.) According to the spatial relationship, we have the translation relation

$$\mathbf{R}_A\mathbf{R}_X\mathbf{t}_B + \mathbf{R}_A\mathbf{t}_X + \mathbf{t}_A = \mathbf{R}_Y\mathbf{R}_C\mathbf{t}_z + \mathbf{R}_Y\mathbf{t}_C + \mathbf{t}_Y \tag{17}$$

Note that the vectors $\mathbf{t}_X$ and $\mathbf{t}_Y$, the translational part of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$, are unknown constant, and $\mathbf{t}_B$ and $\mathbf{t}_Z$ are unknown variables. Then the Eq.(17) can be reformatted as:

$$\mathbf{t}_B = \mathbf{K}\mathbf{t}_{XYZ} + \mathbf{m} \tag{18}$$

where

$$\mathbf{K} = \mathbf{R}_X^{-1}\mathbf{R}_A^{-1}\left[-\mathbf{R}_A \quad \mathbf{I}_3 \quad \mathbf{R}_Y\mathbf{R}_C\right], \tag{19}$$

$$\mathbf{t}_{XYZ} = \left[\mathbf{t}_X^{\intercal} \quad \mathbf{t}_Y^{\intercal} \quad \mathbf{t}_Z^{\intercal}\right]^{\intercal} \in \mathbb{R}^9, \tag{20}$$

and

$$\mathbf{m} = \mathbf{R}_X^{-1}\mathbf{R}_A^{-1}\left(\mathbf{R}_Y\mathbf{t}_C - \mathbf{t}_A\right). \tag{21}$$

The distance from the point $\mathbf{t}_B$ to the line $\mathbf{Pt} = \mathbf{q}$ in 3D space can be expressed as:

$$d = \sqrt{\left(\mathbf{q} - \mathbf{P}\mathbf{t}_B\right)^{\intercal}\left(\mathbf{P}\mathbf{P}^{\intercal}\right)\left(\mathbf{q} - \mathbf{P}\mathbf{t}_B\right)}. \tag{22}$$

With the aim to minimize the sum of d^2 of m different independent measurements, the objective function is defined as follows:

$$\phi\left(\mathbf{t}_{XYZ}\right) = \frac{1}{2}\sum_{i=0}^{m} d_i^2\left(\mathbf{A_i}, \mathbf{C}_i, \mathbf{R}_{Xi}, \mathbf{R}_{Yi}, \mathbf{t}_{XYZ}\right) \tag{23}$$

The Jacobian matrix is $\mathbf{J}_{d^2}\left(\mathbf{t}_{XYZ}\right) \in \mathbb{R}^{9\times m}$, whose i-th element is:

$$-2\left(\mathbf{P}\mathbf{K}_i\right)^{\intercal}\mathbf{P}\mathbf{P}^{\intercal}\left(\mathbf{q} - \mathbf{P}\mathbf{t}_B\right) \tag{24}$$

where $\mathbf{K}_i$ is the i-th column of $\mathbf{K}$.

Therefore, the non-linear objective function can be minimized with the in k-th iteration:

$$\Delta\mathbf{t}_{XYZk} = \left(\mathbf{J}_{d^2}^{\mathrm{T}}\mathbf{J}_{d^2}\right)^{-1}\mathbf{J}_{d^2}^{\mathrm{T}}d\left(\mathbf{t}_{XYZk}\right), \tag{25}$$

$$\mathbf{t}_{XYZk+1} = \mathbf{t}_{XYZk} + \Delta\mathbf{t}_{XYZk}. \tag{26}$$

When the 2D norm of $\mathbf{t}_{XYZk}$ is less than the threshold ϵ, the iteration stops. The $\hat{\mathbf{t}}_X$ is the vector contains the first three elements of $\mathbf{t}_{XYZ}$, and the $\hat{\mathbf{t}}_Y$ is the vector contains the fourth, fifth and sixth elements of $\mathbf{t}_{XYZ}$.

In the Gaussian-Newton method, the initial guess of the iterations is essential. Thus, we provide a method to estimate the initial seed of the iteration.

$$\mathbf{p}_i - k_i\mathbf{v}_i = \mathbf{t}_B \tag{27}$$

where $\mathbf{p}$ is on the line, and the v is the normal of line.

$$\mathbf{K}_i^{+}\mathbf{t}^{+} = \mathbf{n}_i \tag{28}$$

where $\mathbf{K}_i^{+}$ is an extension of $\mathbf{K}_i$ in Eq.(19) with a $3\times m$ sparse matrix $\mathbf{K}_{\mathbf{ex}i}$:

$$\mathbf{K}_i^{+} = \left[\mathbf{K}_i \quad \mathbf{K}_{\mathbf{ex}i}\right]. \tag{29}$$

Only the i-th column of $\mathbf{K}_{\mathbf{ex}i}$ is non-zero, equalling $\mathbf{v}_i$.

$$\mathbf{t}^{+} = \left[\mathbf{t}_{XYZ}^{\intercal} \quad k_1 \quad k_2 \quad \ldots \quad k_m\right]^{\intercal} \tag{30}$$

is a $(9+m)\times 1$ vector. And $\mathbf{n}_i = \mathbf{p}_i - \mathbf{m}_i$.

With m groups of data measured, we can similarly concatenate them together and have

$$\tilde{\mathbf{K}}^{+}\mathbf{t}^{+} = \tilde{\mathbf{n}} \tag{31}$$

where

$$\tilde{\mathbf{K}}^{+} = \left[\mathbf{K}_1^{\intercal} \quad \mathbf{K}_2^{\intercal} \quad \cdots \quad \mathbf{K}_m^{\intercal}\right]^{\intercal} \in \mathbb{R}^{3m\times 9}, \tag{32}$$

$$\tilde{\mathbf{n}} = \left[\mathbf{n}_1^{\intercal} \quad \mathbf{n}_2^{\intercal} \quad \cdots \quad \mathbf{n}_m^{\intercal}\right]^{\intercal} \in \mathbb{R}^{3m}. \tag{33}$$

Hence, $\mathbf{t}$ can be simply solved by

$$\mathbf{t}^{+} = \left(\tilde{\mathbf{J}}^{\intercal}\tilde{\mathbf{J}}\right)^{-1}\tilde{\mathbf{J}}^{\intercal}\tilde{\mathbf{p}}, \tag{34}$$

and $\mathbf{t}_{XYZ}$ is the first 9 elements of the vector $\mathbf{t}^{+}$.

The solution of $\mathbf{AXb} = \mathbf{YCz}$ can be summarized in Algorithm 1.

Algorithm 1 Solving for $\mathbf{AXb} = \mathbf{YCz}$ problem

Require: $\mathcal{A} = \{\mathbf{A}_i\}, \mathcal{B} = \{\mathbf{b}_i\}, \mathcal{C} = \{\mathbf{C}_i\}, i = 1, 2, \ldots, m$
Ensure: $\mathbf{A}_i\mathbf{X}\mathbf{b}_i = \mathbf{Y}\mathbf{C}_i\mathbf{z}, i = 1, 2, \ldots, m$

```
(R_X, R_Y, z) ← ClosedForm({R_Ai}, {b_i}, {R_Ci})          ▷ Eq. (3) - (16)
t_XYZ ← InitiateGuess({A_i}, {b_i}, {C_i}, R_X, R_Y, z)     ▷ Eq. (27) - (34)
while ||Δt_XYZ|| ≤ ε do                                     ▷ ε is the threshold
    J_d² ← Jacobian({A_i}, {b_i}, {C_i}, R_X, R_Y, t_XYZ)   ▷ Eq. (18) - (24)
    Δt_XYZ ← −(J_d²^T J_d²)^(−1) J_d²^T d(t_XYZ)
    t_XYZ ← t_XYZ + Δt_XYZ
end while
(t_X, t_Y) ← VectorElements(t_XYZ)
```

TABLE I
THE DISTURBANCE OF THE NOISE INJECTED INTO THE DATA SET

Noise Level	Rotational (rad)	Translational (mm)
High	$\mathcal{N}(-0.02, 0.02)$	$\mathcal{N}(-2.0, 2.0)$
Medium	$\mathcal{N}(-0.01, 0.01)$	$\mathcal{N}(-0.5, 0.5)$
Low	$\mathcal{N}(-0.005, 0.005)$	$\mathcal{N}(-0.1, 0.1)$

III. Verification and Discussion

A. Simulation Analysis

The simulated analysis of the proposed calibration method entails two distinct scenarios. The first scenario considers introducing different levels of artificial noise, while the second involves different simulation data sets. Since there is little research on this problem, two different simultaneous methods for solving the $\mathbf{AXB} = \mathbf{YCZ}$ are employed for comparison, namely G. Wang's [17] and J. Wang's [16]. J. Wang's method requires a proper initial guess, for which we use the result of G. Wang's method. The data set, including homogeneous transform matrices, is grouped to $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$. To validate our method for $\mathbf{AXb} = \mathbf{YCz}$, only the first column of $\mathbf{B}$ is used. The threshold ϵ of 0.001 is used in rotation determination. 100 random measurements (each contains the $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ trios) were generated. To investigate the influence of the number of data sets, we randomly selected $12, 17, 22, \ldots, 97$ trios (where 12 measurements is the minimum sample requirement for G. Wang's method).

Three different levels of noise were injected into the simulation data set to investigate the performance of the proposed methods under different levels of disturbance, as shown in Table I. In order to ensure the homogeneity of the noise-injected transforms, the rotation part with noise is defined as:

$$\mathbf{R}_i' = \mathbf{R}_i \exp\left(\mathbf{v}_{noise}\right) \tag{35}$$

Where $i = \mathbf{X}, \mathbf{Y}$, and $\exp(\cdot) : \mathfrak{so}(3) \to SO(3)$ is the exponential map form Lie algebra to Lie group (3D rotation group). Every element in the 3×1 vector $\mathbf{v}_{noise}$ is normally distributed. The translational part has noise added directly to the ground truth data.

To evaluate the accuracy of these solutions, the metrics for the errors both in rotation and translation are defined as follows:

$$E_{\mathbf{R}_i} = \| \log\left(\mathbf{R}_{i\text{solved}}^{\mathsf{T}} \mathbf{R}_{i\text{true}}\right) \|, \tag{36}$$

$$E_{\mathbf{t}_i} = \| \mathbf{t}_{i\text{solved}} - \mathbf{t}_{i\text{true}} \| \tag{37}$$

where $i = \mathbf{X}, \mathbf{Y}$, and $\log(\cdot) : SO(3) \to \mathfrak{so}(3)$ is the logarithmic map form Lie group to Lie algebra. The error of $\mathbf{z}$ is defined as

$$E_{\mathbf{z}} = |\cos^{-1}(\mathbf{z}_{\text{true}}^{\mathsf{T}} \mathbf{z}_{\text{solved}})|. \tag{38}$$

As shown in Fig. 2, the average translational and rotational errors are calculated under different noise levels with three methods. Our method can find the proper robot-robot and hand-eye transform matrices if the number of trios in a data set is larger than 30, and the errors decrease as the number of trios increases. According to the results, the number of measurements should be increased to more than 60 to improve the calibration further. The error of our method is observed to be higher than that of the solution for $\mathbf{AXB} = \mathbf{YCZ}$ in some cases. This is because the dimensions of linear equations $\mathbf{AXB} = \mathbf{YCZ}$ are higher than that in $\mathbf{AXb} = \mathbf{YCz}$, which means less information was provided in our method. However, our method may still be considered a suitable solution compared with the noise. The reason why the rotational error of $E_{\mathbf{R}_z}$ is much less than the errors of $E_{\mathbf{R}_X}$ and $E_{\mathbf{R}_Y}$ is that $E_{\mathbf{R}_z}$ represents the rotation angle between true and solved $\mathbf{z}$.

It is evident from the data that the errors decrease when the level of noise decreases. In addition, all methods or observations become increasingly robust to noise as the number of samples increases. Both the translational and rotational errors are less than the mean perturbation in noise.

B. Experimental Analysis

Real-world experiments were conducted to validate our method in the case of a robotic surgical system. For this, we used the da Vinci Surgical System (Intuitive Surgical, CA, USA, Fig. 3) controlled by a da Vinci Research Kit (dVRK) [19]. We utilised two patient-side manipulators (PSM), and one endoscopic camera manipulator (ECM) mounted on three setup joints (SUJ). The PSM is a 7-degree-of-freedom (DOF) cable-driven robot, and the ECM is a 6-DOF robot. The matrices $\mathbf{A}$ and $\mathbf{B}$ in (1) are calculated via their forward kinematics, representing the transformation from the base of PSM/ECM to their respective hand frame. The Panasonic binocular endoscope system records images of the tools, and OpenCV [20], is used to detect the shaft of the surgical instruments and infer their directions in 3D ($\mathbf{b}$). The matrices $\mathbf{X}$, $\mathbf{Y}$ are transformed from endoscope to hand, from ECM base to PSM base. The vector $\mathbf{z}$ is the direction of the shaft under the frame of the PSM end-effector.

In this experiment, two PSMs were employed to investigate the performance of our method. The base frame of each PSM was set to 10 for different configurations by moving the corresponding SUJ. To collect the data, both the ECM and the PSM were programmed to move to a set of 100

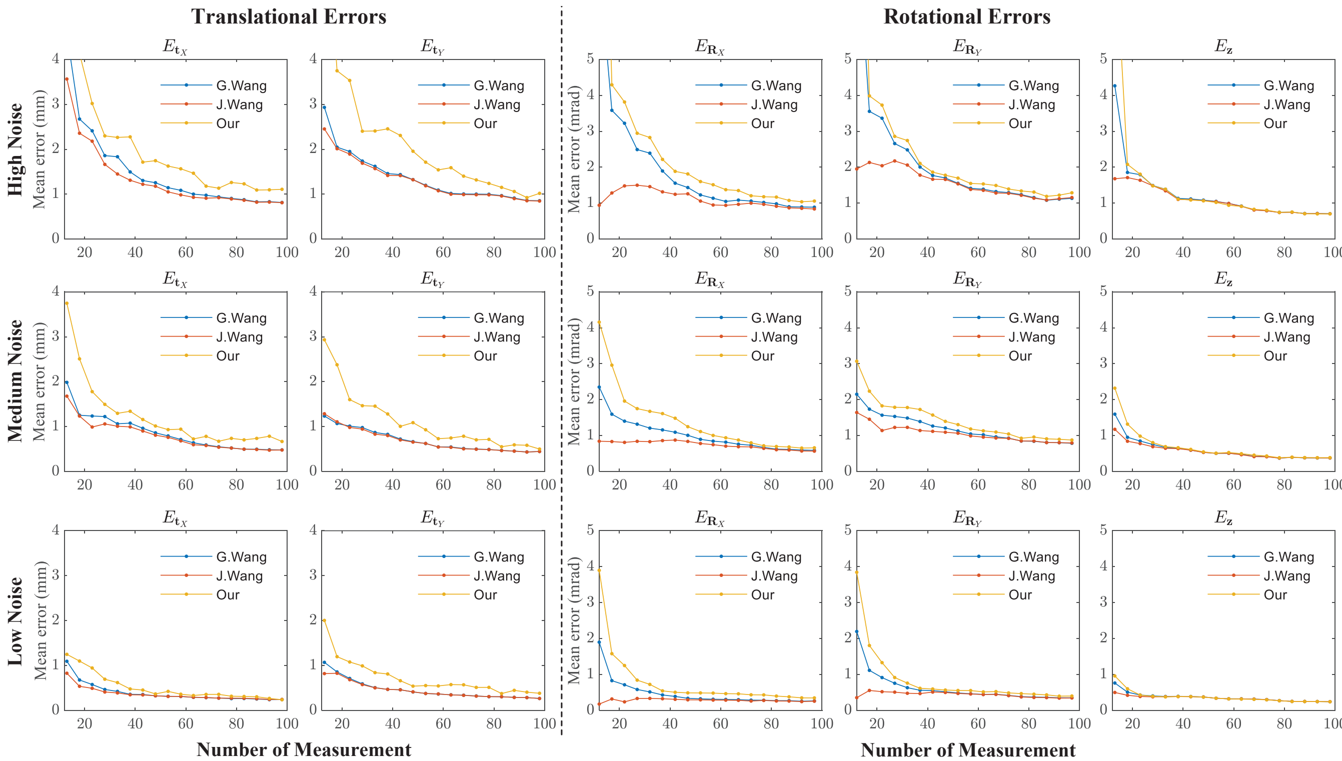

Fig. 2. Translational (first two columns) and rotational (last three columns) errors of calibrated results for three methods with different levels of noise. The methods of G. Wang and J. Wang solve the $\mathbf{AXB} = \mathbf{YBC}$ problem, while our method is for $\mathbf{AXb} = \mathbf{YCz}$.

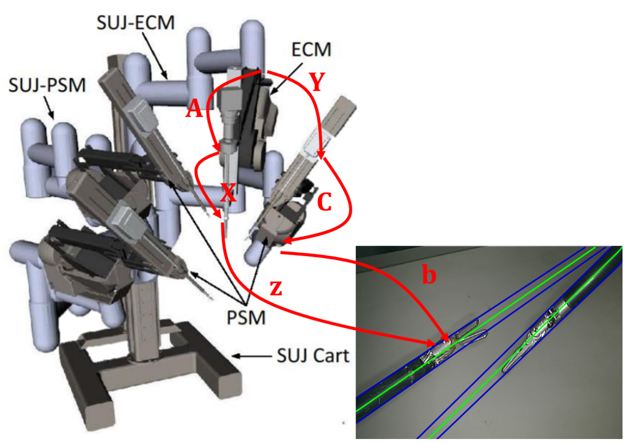

Fig. 3. The setup of the real-world experiment on *da Vinci* Surgical System, and the illustration of the matrices. The endoscope on the ECM captured the image of PSMs. The cyan lines are the detected direction of the surgical instrument.

random positions. Once the target positions were reached, the direction of the surgical instrument was inferred from the endoscopic image. In order to obtain the ground truths for $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{z}$, a bar-code marker is fixed on the surgical instrument (as shown in Fig. 4) from which the central axis may be directly inferred. $\mathbf{z}$ is the first column of $\mathbf{Z}$ if the x-axis of marker's frame is parallel to the shaft. In each case, the 100 trios were employed for the computation. The metrics for the errors have the same definition as those in the simulation.

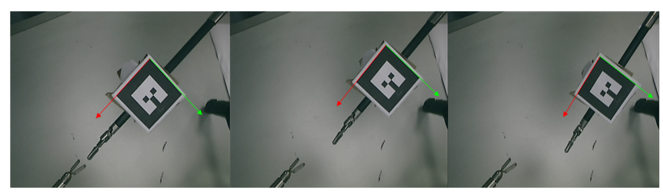

Fig. 4. Using the fiducial marker to detect the 4×4 posture of the surgical instrument. The surgical instrument is in different poses.

The rotational and translational errors are depicted in Fig. 5. Our methods' mean rotational and translational error with calibrated value $\mathbf{X}$ is 0.0239 ± 0.0170 rad and 2.49 ± 0.88 mm, respectively. For the calibration of $\mathbf{Y}$, the mean rotational and translational error calculated is 0.0325 ± 0.0108 rad and 2.78 ± 0.34 mm. The mean angle between $\mathbf{z}$ and the ground truth is 0.0124 ± 0.0112 rad. There is no significant difference between the results of the two PSMs ($P > 0.05$), and the influence of the configuration could be ignored. The proposed method shows acceptable accuracy for some autonomous control o the *da Vinci* Surgical System.

The measurement error from camera imaging, robotic tool detection, fiducial marker tracking, and robotic positioning may influence the validation of the accuracy. As we can see from the error in Fig. 2 and 5, the errors in physical experiments are slightly larger than those in simulations, which is contributed by these measurement errors. Besides, the initial estimation of the $\mathbf{t}_{XYZ}$ may lead to a locally optimal solution in rotation determination. But according to

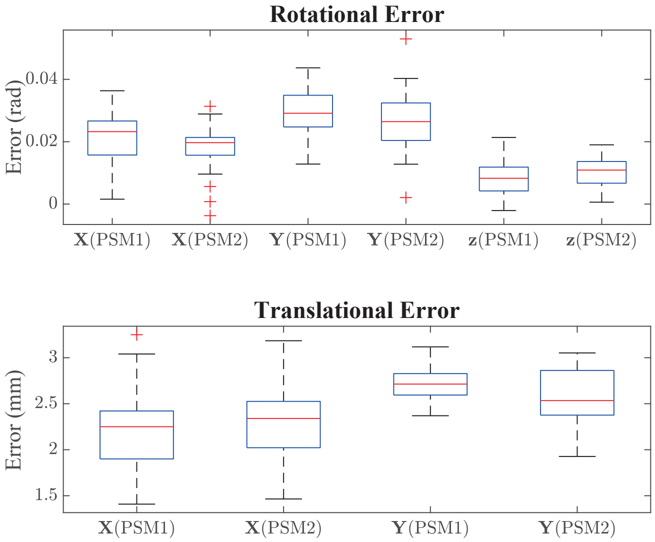

Fig. 5. The disturbance of the rotational and translational errors of PSM1 and PSM2 in 10 different configurations.

the simulation and experiment results, the initial estimations are reliable for optimization.

IV. Conclusion

This paper proposed an efficient dual-arm robot system coordinate calibration approach by visual tool tracking. The problem is formulated as solving a matrix equation $\mathbf{AXb} = \mathbf{YCz}$. The method decoupled the translational and rotational parts in the equations, which can efficiently solve the rotation matrix by the Kronecker-Product-based method and accurately converge to the true solution of the translational part. Simulations and experiments on a *da Vinci* Research Kit are carried out to demonstrate the proposed method's robustness, compared with the state-of-the-art solution of $\mathbf{AXB} = \mathbf{YCZ}$, in terms of average accuracy and stability against noise.

The accuracy is slightly lower than the solutions for $\mathbf{AXB} = \mathbf{YCZ}$ because of less information in the feedback. In the future, we plan to detect all degrees of freedom in the robotic tool, which may improve the feedback of homogeneous transform matrices.

APPENDIX

The 2-Norm has no relation with the vector's shape, and the orthogonal matrix's norm is 3. Thus:

$$\begin{aligned}\|\mathbf{v}_{XYz}\|^2 &= \|\operatorname{vec}(\mathbf{R}_X)\|^2 + \|\operatorname{vec}(\mathbf{z}^\intercal \otimes \mathbf{R}_Y)\|^2 \\ &= \|\mathbf{R}_X\|^2 + \|\mathbf{z}^\intercal \otimes \mathbf{R}_Y\|^2 \\ &= \|\mathbf{R}_X\|^2 + (z_1 + z_2 + z_3)\|\mathbf{R}_Y\|^2 \\ &= 6\end{aligned} \tag{39}$$

where $z_i\,(i = 1, 2, 3)$ are the elements of vector $\mathbf{z}$ and $\sum_i^3 z_i = 1$.

References

[1] 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 Transactions on Robotics*, vol. 32, no. 2, pp. 413–428, 2016.

[2] Y. Shiu and S. Ahmad, "Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form ax=xb," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 1, pp. 16–29, 1989.

[3] F. Park and B. Martin, "Robot sensor calibration: solving ax=xb on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[4] I. Fassi and G. Legnani, "Hand to sensor calibration: A geometrical interpretation of the matrix equation $\mathbf{AX} = \mathbf{XB}$," *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[5] D. Condurache and A. Burlacu, "Orthogonal dual tensor method for solving the $\mathbf{AX} = \mathbf{XB}$ sensor calibration problem," *Mechanism and Machine Theory*, vol. 104, pp. 382–404, 2016.

[6] M. K. Ackerman and G. S. Chirikjian, "A probabilistic solution to the $\mathbf{AX} = \mathbf{XB}$ problem: Sensor calibration without correspondence," in *International Conference on Geometric Science of Information*. Springer, 2013, pp. 693–701.

[7] Q. Ma, H. Li, and G. S. Chirikjian, "New probabilistic approaches to the $\mathbf{AX} = \mathbf{XB}$ hand-eye calibration without correspondence," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4365–4371.

[8] H. Li, Q. Ma, T. Wang, and G. S. Chirikjian, "Simultaneous hand-eye and robot-world calibration by solving the $ax = yb$ problem without correspondence," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 145–152, 2015.

[9] D. Condurache and I.-A. Ciureanu, "A novel solution for $\mathbf{AX} = \mathbf{YB}$ sensor calibration problem using dual lie algebra," in *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*. IEEE, 2019, pp. 302–307.

[10] M. Shah, "Solving the robot-world/hand-eye calibration problem using the kronecker product," *Journal of Mechanisms and Robotics*, vol. 5, no. 3, p. 031007, 2013.

[11] J. Heller, D. Henrion, and T. Pajdla, "Hand-eye and robot-world calibration by global polynomial optimization," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 3157–3164.

[12] J. Ha, D. Kang, and F. C. Park, "A stochastic global optimization algorithm for the two-frame sensor calibration problem," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2434–2446, 2015.

[13] 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 Robotics and Automation Letters*, vol. 5, no. 3, pp. 4086–4093, 2020.

[14] Y. Qin, P. Geng, B. Lv, Y. Meng, Z. Song, and J. Han, "Simultaneous calibration of the hand-eye, flange-tool and robot-robot relationship in dual-robot collaboration systems," *Sensors*, vol. 22, no. 5, p. 1861, 2022.

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

[16] J. Wang, L. Wu, M. Q.-H. Meng, and H. Ren, "Towards simultaneous coordinate calibrations for cooperative multiple robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 410–415.

[17] G. Wang, W.-l. Li, C. Jiang, D.-h. Zhu, H. Xie, X.-j. Liu, and H. Ding, "Simultaneous calibration of multicoordinates for a dual-robot system by solving the axb= ycz problem," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1172–1185, 2021.

[18] F. Richter, J. Lu, R. K. Orosco, and M. C. Yip, "Robotic tool tracking under partially visible kinematic chain: A unified approach," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1653–1670, 2022.

[19] P. Kazanzides, Z. Chen, A. Deguet, G. S. Fischer, R. H. Taylor, and S. P. DiMaio, "An open-source research kit for the da vinci® surgical system," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 6434–6439.

[20] OpenCV, "Open source computer vision library," 2015.