# Hand-Eye Calibration Made Easy through a Closed-Form Two-Stage Method

Soheil Sarabandi, Josep M. Porta, and Federico Thomas

***Abstract*— An analysis of the existing hand-eye calibration methods reveals that most of them are far from trivial. And, what is worse, their intrinsic complexity makes it difficult to elucidate under which circumstances they fail to provide an accurate solution. Thus, although it might seem that hand-eye calibration problem is uninspiring because it is assumed to be well-solved, we show in this paper that there was still room for improvement, both in terms of simplicity and robustness. After reviewing the most representative methods, we analyze the situations in which they fail, and we introduce a simpler closed-form alternative that accurately solves the problem in all the identified critical circumstances. Its performance is evaluated using simulated and real experimental data.**

***Index Terms*— Hand-eye calibration, 3D rigid displacements, axis-angle representation, orthonormalization.**

## I. Introduction

Many applications require attaching a sensor, in particular a camera, to the end effector of a robot. A typical such system is depicted in Fig. 1, where a robot points a camera to a calibration pattern. According to the spatial displacements represented in this figure, we have that

$$\mathbf{Z} = \mathbf{K}_i \mathbf{X} \mathbf{C}_i, \tag{1}$$

where $\mathbf{Z}$, $\mathbf{K}_i$, $\mathbf{X}$, and $\mathbf{C}_i$ for $i = 0, \ldots, n$ are rigid transformations. Although different formalisms can be used to represent such transformations, we can initially assume that they are homogeneous $4\times4$ matrices of the form

$$\begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{pmatrix}, \tag{2}$$

where $\mathbf{R} \in SO(3)$ is a $3\times3$ rotation matrix (*i.e.*, a proper orthogonal matrix) and $\mathbf{t} \in \mathbb{R}^3$, a translation vector. While $\mathbf{K}_i$ can be obtained from the forward kinematics of the robot, $\mathbf{C}_i$ can be calculated using a camera calibration method [1]. The transformations $\mathbf{X}$ and $\mathbf{Z}$, are called the *hand-eye* and the *robot-world* transformations, respectively. The estimation of these transformations from experimental data corresponds to the hand-eye and robot-world calibration problems. Although both calibration problems have been solved simultaneously at least in [2]–[6], we focus here only on the hand-eye calibration part, noting that, once $\mathbf{X}$ is estimated, the computation of $\mathbf{Z}$ directly follows from (1).

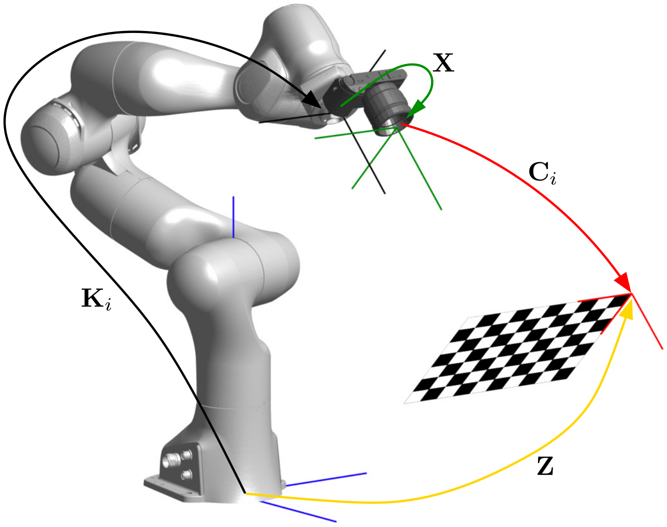

Fig. 1. Rigid transformations in a serial robot with a camera attached to its end effector. Four reference frames are involved: the robot reference frame (in blue), the end-effector reference frame (in black), the camera reference frame (in green), and the world reference frame (in red). The hand-eye calibration problem consists in computing $\mathbf{X}$ from the equations $\mathbf{Z} = \mathbf{K}_i\mathbf{X}\mathbf{C}_i$ for $i = 0 \ldots n$. $\mathbf{K}_i$ and $\mathbf{C}_i$ are given, but $\mathbf{Z}$ is assumed to be unknown, otherwise the problem becomes trivial.

Without loss of generality, we can use the robot and sensor poses at $i = 0$ as a reference. Then, we can write

$$\mathbf{A}_i \mathbf{X} = \mathbf{X} \mathbf{B}_i, \tag{3}$$

where $\mathbf{A}_i = \mathbf{K}_i^{-1}\mathbf{K}_0$ and $\mathbf{B}_i = \mathbf{C}_i\mathbf{C}_0^{-1}$ for $i = 1, \ldots, n$ are, respectively, the *end-effector motion* and the *sensor motion* transformations. It has been proved that, to generate a system with a well-defined solution, one needs to move the robot to, at least, three different configurations to generate two independent equations of the form of (3) [7], [8]. Moreover, while usually $\mathbf{A}_i$ can be accurately obtained from the robot forward kinematics, it is sometimes difficult to obtain accurate values for $\mathbf{B}_i$, depending on the quality of the sensor and its calibration procedure. Thus, to mitigate the effects of noise, more than two equations are typically used to define an over-constrained system whose solution minimizes a certain least squares error.

The matrix equation of the form given in (3) was first studied in [9]. It can be seen as a particular case of the Sylvester equation $\mathbf{AX} + \mathbf{XB} = \mathbf{C}$, which often occurs in system theory [10]. The study of this equation for matrices of the Euclidean group, in the context of hand-eye calibration, dates back to the 1980s when Shiu and Ahmad proposed

This work was partially supported by the Spanish Government through project PID2020-117509GB-I00/AEI/10.13039/50110001103.

The authors are with the Institut de Robòtica i Informàtica Industrial (CSIC-UPC), ETSEIB, Avda. Diagonal, 647, 08028 Barcelona, Spain (e-mail: ssarabandi@iri.upc.edu; porta@iri.upc.edu; f.thomas@csic.es).

This paper has supplementary multimedia material. This material consists of a software package written in MATLAB that permits the reproduction of the results presented in Section IV. Contact with Josep M. Porta for questions about this material.

the first solution method [7], [8]. After all this time, the topic still remains of interest due to its relevance in many applications [11]–[16]. As a consequence, a vast amount of literature connected to this problem is available.

In this paper, we review the most popular hand-eye calibration methods, and we propose a simple method whose main aim is to attain robustness without compromising on either accuracy or computational efficiency. This method is based on a variation of the axis-angle representation of rotations which leads to a closed-form formula involving only the four elementary arithmetic operations. Therefore, it is particularly useful for implementation in embedded microcontrollers with limited computational resources. Moreover, it should be reminded that closed-form formulas are always preferable over iterative procedures, for two main reasons: (1) they permit the analytical analysis of the influence of each variable on the result; and (2) the computational cost, in terms of arithmetic operations, is constant.

This paper is structured as follows. Section II reviews existing approaches classified in terms of the representation used for spatial displacements. On a first reading this section can be omitted without loss of continuity. The new method is described in Section III whose performance is analyzed in Section IV by comparing it against the methods reviewed in Section II. This analysis employs simulated and experimental datasets. The conclusions are drawn in Section V.

## II. A BRIEF REVIEW OF THE EXISTING METHODS

Representing the spatial displacements in (3) in terms of $4\times4$ homogeneous matrices is a highly redundant representation because it uses twelve parameters to encode six degrees of freedom. As a consequence, a simple but meaningful classification of the existing hand-eye calibration methods can be achieved attending to the alternative representations used for the spatial displacements. Contrarily to the usual practice, we will start by reviewing the methods that rely on the most sophisticated representations to end up with the simplest ones because this leads to a smooth introduction of our method.

### A. Dual quaternions

Screw parameters permit encapsulating both translations and rotations just with six independent parameters. Such parameters can be seen as the elements of a dual quaternion. Thus, the methods using this kind of representation are said to be based on dual quaternions, even though the algebra of dual quaternions is not strictly used in the resolution of the problem.

Chen was the first to use screw parameters to solve the hand-eye calibration problem in [17]. Later, Lu and Chou introduced an eight-space quaternion space to solve the problem [18]. The idea was to represent the translation component also using quaternions, but their formulation is similar to that resulting from using dual quaternions. Other methods using the same representation subsequently appear in the works of Kim [19], Daniilidis [20], and Zhao [21].

### B. Double quaternions

A 3D displacement can be approximated by a 4D rotation [22], which can be represented as a double quaternion. Although the result is an approximation, very recently this idea has been successfully used by Wu *et al.* to solve the hand-eye calibration problem [23]. The numerical stability of this method was later improved in [24].

### C. A rotation matrix and a translation vector

The two approaches mentioned above are usually classified as one-stage methods because they simultaneously recover the rotation and the translation components of the hand-eye displacement. Despite their theoretical appeal, mixing translations and rotations in a single optimization is seldom a good idea because a weight factor scaling one with respect to the other is necessarily introduced (in an explicit [23] or explicit [17] way). As a result, the error in the estimated rotation and translation components depend on each other. As an alternative, the translation and the rotation components can be separated by rewriting (3) as

$$\begin{pmatrix}\mathbf{R}_{A_i} & \mathbf{t}_{A_i}\\ \mathbf{0} & 1\end{pmatrix}\begin{pmatrix}\mathbf{R}_{X} & \mathbf{t}_{X}\\ \mathbf{0} & 1\end{pmatrix}=\begin{pmatrix}\mathbf{R}_{X} & \mathbf{t}_{X}\\ \mathbf{0} & 1\end{pmatrix}\begin{pmatrix}\mathbf{R}_{B_i} & \mathbf{t}_{B_i}\\ \mathbf{0} & 1\end{pmatrix}.$$

That is,

$$\begin{pmatrix}\mathbf{R}_{A_i}\mathbf{R}_{X} & \mathbf{R}_{A_i}\mathbf{t}_{X}+\mathbf{t}_{A_i}\\ \mathbf{0} & 1\end{pmatrix}=\begin{pmatrix}\mathbf{R}_{X}\mathbf{R}_{B_i} & \mathbf{R}_{X}\mathbf{t}_{B_i}+\mathbf{t}_{X}\\ \mathbf{0} & 1\end{pmatrix},$$

and, therefore,

$$\mathbf{R}_{A_i}\mathbf{R}_X=\mathbf{R}_X\mathbf{R}_{B_i}, \tag{4}$$

$$\left(\mathbf{R}_{A_i}-\mathbf{I}_{3\times3}\right)\mathbf{t}_X=\mathbf{R}_X\mathbf{t}_{B_i}-\mathbf{t}_{A_i}, \tag{5}$$

which are the *equation of rotation* and the *equation of translation*, respectively. Using these two equations, the original problem can be solved in two stages: first, the least-squares solution, say $\widehat{\mathbf{R}}_X$, of the overconstrained system formed by $n$ equations in the form of (4) is taken as an estimation of $\mathbf{R}_X$ and then the least-squares solution, say $\widehat{\mathbf{t}}_X$, of the overconstrained system of $n$ equations of the form of (5) is taken as an estimation of $\mathbf{t}_X$.

It is usually argued that, although two-stage methods may be optimal in each stage, the final result may not be globally optimal due to the coupling of the rotation and translation equations. However, as we will see in Section IV, the effects of such coupling may be worse in one-stage methods in some cases.

Assuming that $\mathbf{R}_X$ is known, (5) is rewritten for $n$ constraints as a linear overconstrained system

$$\mathbf{C}\,\mathbf{t}_X=\mathbf{D}, \tag{6}$$

with

$$\mathbf{C}=\begin{pmatrix}\mathbf{R}_{A_1}-\mathbf{I}_{3\times3}\\ \vdots\\ \mathbf{R}_{A_n}-\mathbf{I}_{3\times3}\end{pmatrix} \tag{7}$$

and

$$\mathbf{D}=\begin{pmatrix}\mathbf{R}_X\mathbf{t}_{B_1}-\mathbf{t}_{A_1}\\ \vdots\\ \mathbf{R}_X\mathbf{t}_{B_n}-\mathbf{t}_{A_n}\end{pmatrix}. \tag{8}$$

The least-squares solution of (6) is

$$\widehat{\mathbf{t}}_X = \mathbf{C}^{+}\mathbf{D}, \tag{9}$$

where $\mathbf{C}^{+} = \left(\mathbf{C}^{\top}\mathbf{C}\right)^{-1} \mathbf{C}^{\top}$ is the left Moore-Penrose pseudoinverse of $\mathbf{C}$. Since $\widehat{\mathbf{t}}_X$ is easy to obtain, the problem is essentially reduced to obtain $\widehat{\mathbf{R}}_X$ from (4).

Liang and Mao in [25] directly reformulated (4) using $3\times3$ rotation matrices in terms of Kronecker products, or tensor products, and the vectorization operator *vec*[1], as

$$\begin{pmatrix} \mathbf{R}_{A_1} \otimes \mathbf{I}_{3\times3} - \mathbf{I}_{3\times3} \otimes \mathbf{R}_{B_1}^{\top} \\ \vdots \\ \mathbf{R}_{A_n} \otimes \mathbf{I}_{3\times3} - \mathbf{I}_{3\times3} \otimes \mathbf{R}_{B_n}^{\top} \end{pmatrix} \mathit{vec}\, \mathbf{R}_X = \mathbf{0}_{9n\times1}. \tag{10}$$

Then, $\widehat{\mathbf{R}}_X$ is obtained by computing two singular value decompositions (SVDs), one to solve (10) depending on the number of robot and camera poses and another of constant size to orthonormalize the result. As we will see in Section IV, this method leads to excellent results, the only disadvantage being its high computational cost, especially when $n$ is large. To reduce this computational complexity, other authors explored the use of alternative representations for the rotations.

### D. Axis-angle representation of the rotations

According to Euler's rotation theorem, any composition of rotations in three-dimensional space, say $\mathbf{R}_{A_i}$, is equivalent to a pure rotation of size $\alpha_i$ about an axis, $\bar{\mathbf{a}}_i$, called the equivalent axis of rotation [26]. $\mathbf{R}_{A_i}$, $\alpha_i$, and $\mathbf{a}_i$ are related through the expression:

$$\bar{\mathbf{a}}_i = \frac{1}{2\sin\alpha_i}\,\mathbf{a}_i, \tag{11}$$

where

$$\mathbf{a}_i = \begin{pmatrix} \mathbf{R}_{A_i}(3,2) - \mathbf{R}_{A_i}(2,3) \\ \mathbf{R}_{A_i}(1,3) - \mathbf{R}_{A_i}(3,1) \\ \mathbf{R}_{A_i}(2,1) - \mathbf{R}_{A_i}(1,2) \end{pmatrix} \tag{12}$$

is the vector of Euler parameters which involves all the elements out of the diagonal of $\mathbf{R}_{A_i}$ since $\mathbf{R}_{A_i}(i,j)$ refers to the entry in the $i$-th row and $j$-th column of $\mathbf{R}_{A_i}$. Equation (11) does not work if $\sin\alpha_i = 0$. That is, if $\alpha_i{=}k\pi$, for $k \in \mathbb{Z}$. When $k$ is odd, $\mathbf{a}_i$ can still be computed by considering the elements in the diagonal of $\mathbf{R}_{A_i}$, otherwise $\bar{\mathbf{a}}_i$ is undefined. In particular, it is undefined when the rotation is the identity and it is numerically ill-conditioned for small rotations. The same happens for $\bar{\mathbf{b}}_i$, defined from $\mathbf{R}_{B_i}$ in the same way as $\bar{\mathbf{a}}_i$ is defined from $\mathbf{R}_{A_i}$. Despite this drawback, this representation of rotations has dominated the resolution of the hand-eye calibration problem since the seminal work of Shiu and Ahmad [7], [8]. One additional disadvantage of their method is the large size of the generated system of equations because they treat the sines and cosines of the rotated angles as independent terms. Almost simultaneously, Tsai and Lenz [27] solved the problem using a more efficient approach where the number of unknowns stays the same no matter how many measurements are considered. The axis angle representation is also used in [28] and [29] which, as we will see in Section IV are more robust than the previous methods in some situations.

A simple way to circumvent the issue caused by small rotations consists directly using $\mathbf{a}_i$ and $\mathbf{b}_i$ instead on $\bar{\mathbf{a}}_i$ and $\bar{\mathbf{b}}_i$. This is an essential point in the new method presented in the next section.

Since Euler parameters can be arranged as the elements of a quaternion, the methods using this representation are usually called quaternion-based methods, despite the algebra of quaternions is not needed. Chou and Kamel are credited to be the first to present an algorithm based on quaternions [30]. In their approach, a system of nonlinear equations is iteratively solved using a Newton-Raphson procedure. They later improved their method by transforming the rotational component into a system of linear equations, which was solved using a SVD [31]. Although quaternions were also used later by Zhuang and Roth in [32] to come up with a non-iterative approach, the results were presented without referring to quaternions, resulting in an algorithm that is easy to understand and implement. This work was later extended in [33] to simultaneously solve the hand-eye and the robot-world calibration problems.

## III. The proposed method

The proposed method arises from rewriting (4) as

$$\mathbf{R}_{A_i} = \mathbf{R}_X \mathbf{R}_{B_i} \mathbf{R}_X^{-1}, \tag{13}$$

which emphasizes that $\mathbf{R}_X$ can be interpreted as a reference change: the rotation $\mathbf{R}_{A_i}$ can be obtained by first applying the reference change given by $\mathbf{R}_X$, then applying the rotation $\mathbf{R}_{B_i}$ and, finally undoing the reference change. Thus, the equivalent axes of rotations corresponding to $\mathbf{R}_{A_i}$ and $\mathbf{R}_{B_i}$ are related through $\mathbf{R}_X$ by the equation

$$\mathbf{a}_i = \mathbf{R}_X \mathbf{b}_i. \tag{14}$$

For a formal proof of this fact, see Theorem 3 and Lemma 4 in [7].

In contrast to previous approaches, which solve (14) by expressing the rotation described by $\mathbf{R}_X$ with a minimal set of parameters, we generate enough equations in the form of (14) to directly identify all the entries of $\mathbf{R}_X$. The advantage of this approach is that the constraints remain linear and, thus, they can be solved using standard linear algebra tools. Next, we describe this approach for the case of two constraints, which, as already mentioned, is the minimal set of constraints that permits determining $\widehat{\mathbf{R}}_X$. Then, we will generalize the result to deal with $n > 2$ constraints.

For two constraints, we have that

$$(\mathbf{a}_1,\, \mathbf{a}_2) = \mathbf{R}_X\ (\mathbf{b}_1,\, \mathbf{b}_2)\,. \tag{15}$$

Since if $\mathbf{a}_1{=}\mathbf{R}_X\mathbf{b}_1$ and $\mathbf{a}_2{=}\mathbf{R}_X\mathbf{b}_2$, then we have that $\mathbf{a}_1{\times}\mathbf{a}_2{=}\mathbf{R}_X(\mathbf{b}_1{\times}\mathbf{b}_2)$ and, hence,

$$(\mathbf{a}_1,\, \mathbf{a}_2,\, \mathbf{a}_1{\times}\mathbf{a}_2) = \mathbf{R}_X\ (\mathbf{b}_1,\, \mathbf{b}_2,\, \mathbf{b}_1{\times}\mathbf{b}_2)\,. \tag{16}$$

[1] The vectorization of a matrix is a linear transformation which converts the matrix into a column vector by stacking the columns of the matrix on top of one another.

If $\mathbf{b}_1$ and $\mathbf{b}_2$ are not collinear, $(\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_1 \times \mathbf{b}_2)$ is a full rank $3\times3$ matrix. Therefore, the solution of (16) is

$$\widetilde{\mathbf{R}}_X = (\mathbf{a}_1,\, \mathbf{a}_2,\, \mathbf{a}_1 \times \mathbf{a}_2)\,(\mathbf{b}_1,\, \mathbf{b}_2,\, \mathbf{b}_1 \times \mathbf{b}_2)^{-1}. \tag{17}$$

For the general case, that in which $n > 2$, we can define the overconstrained linear system

$$\mathbf{A} = \mathbf{R}_X\, \mathbf{B}, \tag{18}$$

where

$$\mathbf{A} = (\mathbf{a}_1, \cdots \mathbf{a}_n, \mathbf{a}_{1,1}, \cdots \mathbf{a}_{i,j}, \cdots \mathbf{a}_{n,n}), \tag{19}$$

and

$$\mathbf{B} = (\mathbf{b}_1, \cdots \mathbf{b}_n, \mathbf{b}_{1,1}, \cdots \mathbf{b}_{i,j}, \cdots \mathbf{b}_{n,n}), \tag{20}$$

with $\mathbf{a}_{i,j}{=}\mathbf{a}_i \times \mathbf{a}_j$ and $\mathbf{b}_{i,j}{=}\mathbf{b}_i \times \mathbf{b}_j$, $i, j \in \{1, \ldots, n\}$, $i < j$.

Now, the least-squares solution of (18) is

$$\widetilde{\mathbf{R}}_X = \mathbf{A}\mathbf{B}^+, \tag{21}$$

where $\mathbf{B}^+ = \mathbf{B}^\top \left(\mathbf{B}\mathbf{B}^\top\right)^{-1}$ denotes the right Moore-Penrose pseudoinverse of $\mathbf{B}$. Observe that $\mathbf{B}^+ = \mathbf{B}^{-1}$ if $\mathbf{B}$ is a $3\times3$ full rank matrix. Thus, (21) generalizes (17). However, as we will see in Section IV, for large values of $n$, the incorporation in the formulation of all the possible cross products may result in marginal accuracy gains, but in a noticeable computational cost increment. Thus, a trade-off arises with respect to these two factors. Nevertheless, if $\mathbf{B}\mathbf{B}^\top$ has a low condition number without considering the cross products, clearly there is no reason to include them.

Due to noisy measurements, $\widetilde{\mathbf{R}}_X$ is not necessarily proper orthogonal, i.e., its determinant is not exactly equal to 1, and its rows, or columns, are not exactly orthogonal to each other. Several methods can be used to restore the orthogonality of noisy rotation matrices [34]. For instance, the nearest rotation matrix, in Frobenius norm, to $\widetilde{\mathbf{R}}_X$ can be approximated by iteratively applying the following updating rule

$$\widetilde{\mathbf{R}}_X \leftarrow \widetilde{\mathbf{R}}_X \left(3\,\mathbf{I}_{3\times3} + \widetilde{\mathbf{R}}_X^\top \widetilde{\mathbf{R}}_X\right)\left(\mathbf{I}_{3\times3} + 3\,\widetilde{\mathbf{R}}_X^\top \widetilde{\mathbf{R}}_X\right)^{-1}. \tag{22}$$

This iterative procedure was introduced in [35]. Although other updating rules exist, (22) is advantageous due to its cubic convergence to the solution [36]. As we will see in the next section, for most practical applications, only two iterations of (22) are enough to obtain estimated rotation matrices with negligible orthogonality errors. This permits expressing the proposed method as a closed-form formula.

Finally, we can summarize the proposed method in the following four simple steps:

1) Apply (12) to the elements in the input sets $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ to obtain the corresponding $\mathbf{a}_i$ and $\mathbf{b}_i$ vectors. These vectors, and possibly their cross products, form the columns of two matrices, $\mathbf{A}$ and $\mathbf{B}$, respectively.
2) Use (21) to determine $\widetilde{\mathbf{R}}_X$, which can be seen as a noisy rotation matrix.
3) Use two iterations of (22) to orthonormalize $\widetilde{\mathbf{R}}_X$ thus obtaining $\widehat{\mathbf{R}}_X$, the solution to the equation of rotation (4).
4) Finally, complete the hand-eye transform $\mathbf{X}$ using (9) to find the solution of the equation of translation (5).

## IV. Performance analysis

The methods appearing in the first column of Table I have been implemented and evaluated using simulated and experimental data. For each method, the first letter denotes the type of representation used according to the subsections of Section II. Thus, methods A1 and A2 rely on dual quaternions, B1 on double quaternions, C1 use rotation matrices and translation vectors, and D1 to D7 the axis-angle representation. A1, A2, and B1 are single-stage methods and the rest are two-stage methods. D7 is the method proposed in Section III. This paper has supplementary multimedia material consisting of a set of MATLAB functions implementing these methods and the scripts used to obtain the results presented in this section. The reported results have been obtained on a desktop computer with a 3.7 GHz Intel Core i7 processor using double-precision floating-point format.

### A. Performance analysis using simulated data

First of all, we have to verify that all methods recover the original displacement for noiseless data. To this end, we apply the following procedure:

1) Randomly generate $\mathbf{X}$, and $\mathbf{B}_i$, for $i = 1, \ldots, 10$. In both cases, the rotational component is obtained by computing uniformly distributed points on $S^3 {\subset} \mathbb{R}^4$ and the translation component by computing points uniformly distributed in $[-5, 5]^3 \subset \mathbb{R}^3$.
2) Compute $\mathbf{A}_i = \mathbf{X}\mathbf{B}_i\mathbf{X}^{-1}$.
3) Taking $\{\mathbf{A}_i\}$ and $\{\mathbf{B}_i\}$ as inputs, obtain $\widehat{\mathbf{R}}_X$ and $\widehat{\mathbf{t}}_X$ with each of the tested methods, and compute:
    a) The rotation error, $E_R$, as the Frobenius norm of $\widehat{\mathbf{R}}_X{-}\mathbf{R}_X$, which has the geometric interpretation given in [34].
    b) The orthogonality error, $E_O$, as $\left|\det(\widehat{\mathbf{R}}_X){-}1\right|$.
    c) The translational error, $E_T$, as $\left\|\widehat{\mathbf{t}}_X{-}\mathbf{t}_X\right\|$.

The second column of Table I summarizes the results averaged over 1000 repetitions of this procedure. According to these results, the errors of method B1 are the largest ones. This is because this method obtains an approximate result. In contrast, the level of error for the rest of the methods, including the one proposed here, can be considered as negligible in practice. Thus, despite its undeniable theoretical interest, B1 should not be the method of choice, in general.

Now, if the same experiment is repeated with the only difference that one of the elements of the set $\{\mathbf{B}_i\}$, say $\mathbf{B}_n$, is equal to the identity, the results in third column of Table I are obtained. Methods D1, D2, D3, D4, and D5 fail because the computation of the equivalent axis of rotation is ill-defined for small rotations. If, instead, $\mathbf{B}_n$ is equal to a rotation of $\pi$ radians about the $x-$axis, the results in the fourth column of Table I are obtained. In this case, methods A1 and A2 also fail and D1 to D5 fail again. Although the lack of robustness of these methods can be remedied by discarding the measurements whose rotational components are close to the identified singularities, the situation is more complicated because some singularities depend on the spatial

TABLE I

PERFORMANCE COMPARISON OF THE DIFFERENT METHODS WITH NOISELESS DATA AND $n = 10$.

| **Method** | **Random set** $\{\mathbf{B}_i\}$ | | | $\mathbf{B}_n = \mathbf{I}$ | | | $\mathbf{B}_n = \mathbf{R_x}(\pi)$ | | | $\mathbf{R}_X = \mathbf{I}$ | | | $\mathbf{R}_X = \mathbf{R_x}(\pi)$ | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $E_R$ | $E_O$ | $E_T$ | $E_R$ | $E_O$ | $E_T$ | $E_R$ | $E_O$ | $E_T$ | $E_R$ | $E_O$ | $E_T$ | $E_R$ | $E_O$ | $E_T$ |
| A1, Lu-Chou, 1995 [18] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| A2, Daniilidis, 1999 [20] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| B1, Wu *et al.*, 2021 [23], [24] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| C1, Liang-Mao, 2008 [25] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| D1, Tsai-Lenz, 1989 [27] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| D2, Shiu-Ahmad, 1989 [8] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ |
| D3, Park-Martin, 1989 [29] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✗ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| D4, Wang, 1992 [37] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ | ✓ | ✗ |
| D5, Horaud-Dornaika, 1995 [28] | ✓ | ✓ | ✓ | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| D6, Chou-Kamel, 1991 [31] | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| D7, This paper | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

✓: $E < 10^{-8}$ ✓: $E \in [10^{-8}, 10^{-4}]$ ✗: $E > 10^{-4}$ or Failure

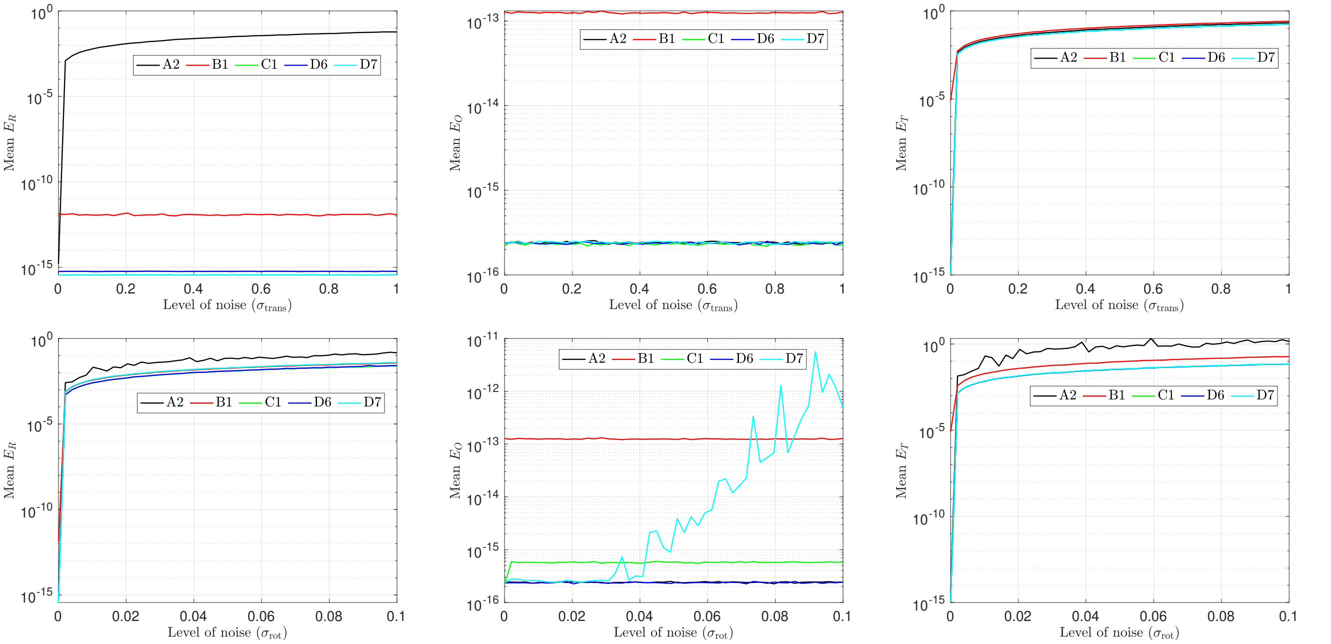

Fig. 2. Errors of methods A2, B1, C1, D6, and D7 with translational (top row) and rotational noise (bottom row) in the elements of $\{\mathbf{A}_i\}$. Left column: mean rotational error of the estimated rotation matrix. Middle column: mean orthogonality error. Right column: mean translation error.

displacement to be estimated. For example, if the same experiment is repeated for $\mathbf{R}_X$ equal to the identity, the results in the fifth column of Table I are obtained. In this case, D3, D4, and D5 generate accurate results, but D1 and D2 still fail. If $\mathbf{R}_X$ is made equal to a rotation of $\pi$ radians about the $x-$axis, the results in the last column of Table I are obtained. Now, A1, D1, and D4 fail.

Considering the results with noiseless input data, the most robust methods for each type of representation, i.e., methods with less fails and with lower error in all the tested situations, are A2, B1, C1, D6 and the proposed method, D7. To evaluate the performance of these methods under the presence of noise, we can perturb the ten elements in $\{\mathbf{A}_i\}$ as follows:

$$\mathbf{A}_i \leftarrow \mathbf{A}_i \, \delta\mathbf{T}, \tag{23}$$

where

$$\delta\mathbf{T} = \begin{pmatrix} \mathbf{R_x}(\delta\theta_x)\mathbf{R_y}(\delta\theta_y)\mathbf{R_z}(\delta\theta_z) & \begin{matrix}\delta_x \\ \delta_y \\ \delta_z\end{matrix} \\ \begin{matrix} 0 & 0 & 0 \end{matrix} & 1 \end{pmatrix}. \tag{24}$$

If we perturb the translation component of $\mathbf{A}_i$, then $\delta_x$, $\delta_y$, and $\delta_z$ are treated as uncorrelated uniformly distributed random variables in the internal $[-\sigma_{\text{trans}}, \sigma_{\text{trans}}]$, and $\delta\theta_x$=$\delta\theta_y$=$\delta\theta_z$=0. The plots obtained for $\sigma_{\text{trans}} \in [0, 1]$ appear in Fig. 2(top row).

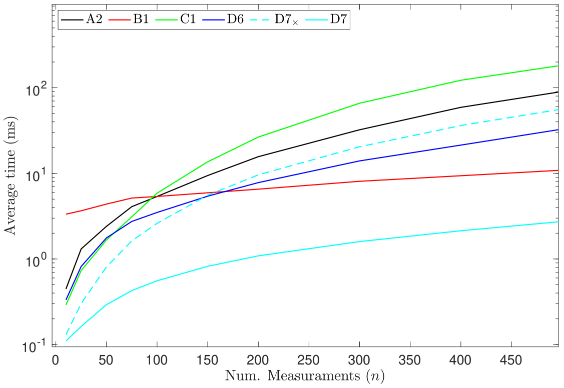

Fig. 3. Evolution of the execution time in milliseconds for methods A2, B1, C1, D6, D7$_\times$ and D7 as a function of the number of considered measurements ($n$).

If we perturb the rotational component of $\mathbf{A}_i$, then $\delta\theta_x$, $\delta\theta_y$, and $\delta\theta_z$ are treated as uncorrelated uniformly distributed random variables in the internal $[-\sigma_{\text{rot}}, \sigma_{\text{rot}}]$, and $\delta_x{=}\delta_y{=}\delta_z{=}0$. The plots obtained for $\sigma_{\text{rot}} \in [0, 0.1]$ radians appear in Fig. 2(bottom row).

From these plots, we conclude that method B1 leads to higher levels of error in some cases (which is not surprising because it is an approximate method), and method A2 couples rotations and translations in such a way that errors in translation have a strong influence on the rotation estimation. This is an expected effect for all one-stage methods, but the particular bad behavior of method A2 under the presence of noise makes it less interesting for practical applications. In contrast, two-stage methods lead to negligible levels of error in all cases. The only relevant effect is the increment in $E_O$ for method D7 for large values of $\sigma_{\text{rot}}$. This error can be canceled by applying the update rule in (22) more times. However, the error is small enough in all cases for practical purposes and this refinement would result in larger execution times. Figure 3 shows the average execution time with $n$ varying from 10 to 500 for methods A1, B1, C1, D6, and D7. Concerning method D7, we give the results with and without incorporating the cross products denoted, respectively, as D7$_\times$ and D7. The computational cost of B1 is dominated by the computation of the cofactors of a constant-size matrix. This operation is not included in the original proposal [23], but it is necessary to avoid some singularities. Despite this fact, this method is more efficient than methods A2, C1, and D6 for large numbers of measurements. Concerning methods D7$_\times$ and D7, Figure 3 reflects that the incorporation of cross products obviously results in a computational time increment. Nevertheless, D7$_\times$ is only worse than alternative methods for more than 150 measurements, a number rarely attained in practical applications. Without the incorporation of cross products, the proposed method is faster than all other methods even for a number of measurements as high as 500.

The consideration of the methods A1 and D1 to D5 do not modify the conclusions drawn from Figs. 2 and 3. This can be verified with the code accompanying this paper.

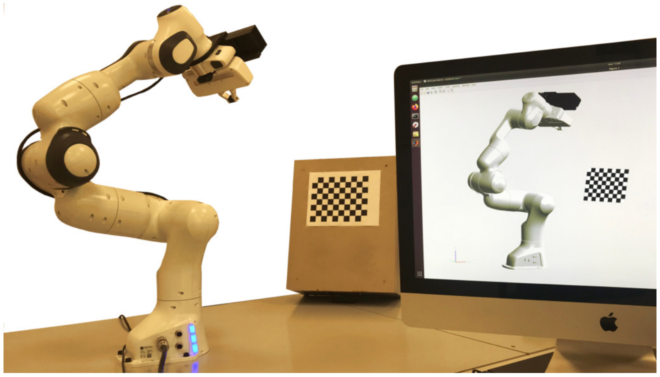

Fig. 4. Experimental setting used to evaluate the proposed approach.

### B. Performance analysis using experimental data

The experimental setup to validate the proposed method appears in Fig. 4. It consists of a 7-DOF Panda serial robot manipulator from Franka Emika [38] equipped with a Microsoft Kinect v2 camera [39] mounted on its end effector using a custom 3D printed flange.

The robot has been moved to 13 different configurations taken at random, but pointing to a checkered pattern, as shown in Fig. 5. The poses of the camera with respect to this pattern have been obtained using the Matlab camera calibration toolbox, which implements the method described in [1]. We obtained an average reprojection error of 0.22 pixels (with a maximum of 0.35 pixels).

Using these 13 configurations, up to $n = 12$ independent constraints in the form of (3) can be defined. However, we first evaluate the methods identified as robust (*i.e.*, methods A2, B1, C2, D6, and D7) using the 66 possible subsets of two constraints. This illustrates the performance of such methods when using the minimum information from which the hand-eye calibration problem can be solved. Table II summarizes the results averaged over the 66 combinations. Since we do not have the ground truth, the errors have to be computed with respect to the input data. Thus, we compute the rotation error as

$$\widehat{E}_R = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{R}_{A_i}\widehat{\mathbf{R}}_X - \widehat{\mathbf{R}}_X\mathbf{R}_{B_i}\right\|, \tag{25}$$

the orthogonality error as

$$\widehat{E}_O = \left|\det\left(\widehat{\mathbf{R}}_X\right) - 1\right|, \tag{26}$$

and the translation error as

$$\widehat{E}_T = \frac{1}{n}\sum_{i=1}^{n}\left\|\left(\mathbf{R}_{A_i} - \mathbf{I}_{3\times3}\right)\widehat{\mathbf{t}}_X - \widehat{\mathbf{R}}_X\mathbf{t}_{B_i} + \mathbf{t}_{A_i}\right\|. \tag{27}$$

The results show that, in this case, two-stage methods provide more accurate estimations in translation than one-stage methods. Thus, at least in this case, the rotational error does not negatively affect the translational error, which is an argument often used to criticize two-stage methods. Although error for all the tested methods are low enough for most practical cases, the different two-stage methods deliver

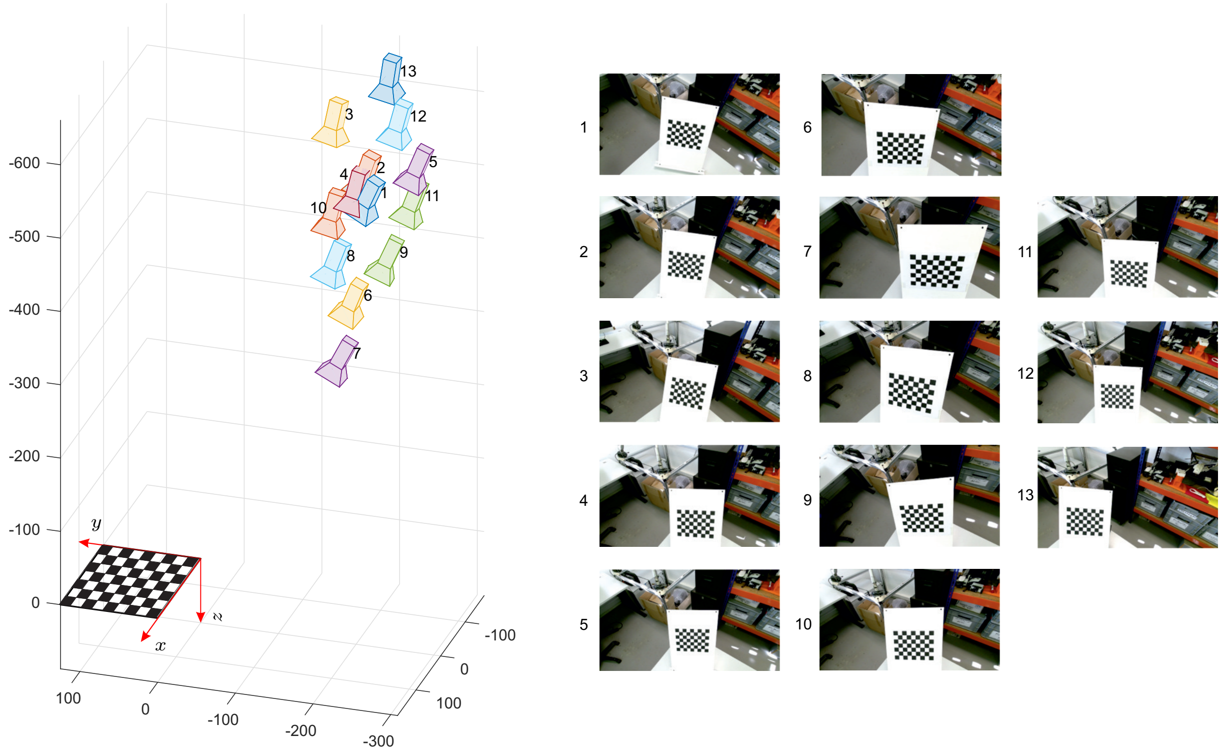

Fig. 5. To evaluate the proposed method, the pose of the camera with respect to a fixed external pattern has been obtained for 13 different randomly selected robot configurations using the experimental setup in Fig 1. Left: Poses of the camera with respect to the pattern. The displacements are given in millimeters. Right: Images of the pattern obtained in the different poses.

TABLE II

Average performance of the compared methods on experimental data considering two constraints at a time.

| Method | $\widehat{E}_R$ | $\widehat{E}_O$ | $\widehat{E}_T$ [mm] |
|---|---|---|---|
| A2 | $7.68{\cdot}10^{-2}$ | $2.01{\cdot}10^{-16}$ | 11.44 |
| B1 | $0.85{\cdot}10^{-2}$ | $1.26{\cdot}10^{-10}$ | 4.24 |
| C1 | $0.85{\cdot}10^{-2}$ | $6.19{\cdot}10^{-16}$ | 3.14 |
| D6 | $0.85{\cdot}10^{-2}$ | $2.50{\cdot}10^{-16}$ | 3.14 |
| D7$_\times$ | $1.05{\cdot}10^{-2}$ | $1.76{\cdot}10^{-12}$ | 2.86 |

TABLE III

Performance of the compared methods on experimental data considering all the possible constraints simultaneously.

| Method | $\widehat{E}_R$ | $\widehat{E}_O$ | $\widehat{E}_T$ [mm] |
|---|---|---|---|
| A2 | $1.44{\cdot}10^{-2}$ | $2.22{\cdot}10^{-16}$ | 1.98 |
| B1 | $1.00{\cdot}10^{-2}$ | $1.05{\cdot}10^{-10}$ | 4.29 |
| C1 | $1.00{\cdot}10^{-2}$ | 0 | 3.36 |
| D6 | $1.00{\cdot}10^{-2}$ | 0 | 3.36 |
| D7$_\times$ | $1.01{\cdot}10^{-2}$ | $4.44{\cdot}10^{-16}$ | 3.14 |
| D7 | $1.18{\cdot}10^{-2}$ | $8.54{\cdot}10^{-15}$ | 2.44 |

different compromise solutions with respect to the rotational and translational errors. When considering all the possible constraints simultaneously, this compromise still remains, as shown in Table III. In this latter case, the condition number of $\mathbf{B}\mathbf{B}^\top$ in (21) is as low as $5.76$. Thus, the equivalent axes of rotation derived from the input data already provide enough information for calibration and, consequently, the incorporation of cross products have a minor influence: a moderate rotational error decrease and a slightly translational error increment.

It is interesting to observe that, when considering all the constraints, method B1 is the one with larger orthogonal and translational errors. This is because it is an approximated method. Method A2 is the one with the lower translational error, but we have seen that this is not the case when the number of used constraints is low. In contrast, the one-stage methods (C1, D6, and D7) have good performance independently of the number of measurements. Nevertheless, while D7 does not rely on any iterative numerical method, C1 requires the computation of the SVD of a $9n{\times}9$ and a $3{\times}3$ matrix, and D6, the SVD of a $4n{\times}4$ matrix, which explains their higher execution times.

## V. Conclusion

It is usually argued that the two-stage methods for the hand-eye calibration problem (those that first estimate the orientation and then the translation) have the important drawback that the orientation estimation error propagate to the translation estimation. To provide a remedy to this problem, some authors have proposed one-stage methods based on dual quaternions (*e.g.*, methods A1 and A2), or on double quaternions (*e.g.*, method B1). Nevertheless, the experiments reported in this paper do not support their superiority in all cases. Moreover, using a closed-form method, such as the one proposed in this paper, one can always derive expressions for both the rotation and the translation directly in terms of the input data, effectively preventing the error propagation from rotations to translations.

The method proposed in this paper relies on two observations on the hand-eye calibration problem: (a) only the equivalent axes of rotations of the pose measurements are relevant as the rotated angles about such axes play no role; and (b) only under the presence of noise, the estimated rotation matrix using a pseudoinverse departs from being proper orthogonal. The result of exploiting these two facts is a closed-form solution method which can be implemented in few lines of code. We have shown that it outperforms the analyzed previously-proposed methods in terms of computational cost without introducing any significant degradation in the quality of the results.

## References

[1] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[2] F. Dornaika and R. Horaud, "Simultaneous robot-world and hand-eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 617–622, 1998.

[3] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product," *International Journal of the Physical Sciences*, vol. 5, no. 10, 2010.

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

[5] 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, 2016.

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

[7] Y. C. Shiu and S. Ahmad, "Finding the mounting position of a sensor by solving a homogeneous transform equation of the form $AX = XB$," in *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, vol. 4, 1987, pp. 1666–1671.

[8] ——, "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.

[9] J. Sylvester, "Sur l'equations en matrices $px = xq$," *Comptes Rendus de l'Académie des Sciences Paris*, vol. 99, no. 2, p. 67–71 and 115–116, 1884.

[10] R. H. Bartels and G. W. Stewart, "Solution of the matrix equation $AX + XB = C$," *Communications of the ACM*, vol. 15, no. 9, pp. 820–826, 1972.

[11] G. H. DeSouza, A. H. Jones, and A. C. Kak, "An world-independent approach for the calibration of mobile robotics active stereo heads," in *IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 4, 2002, pp. 3336–3341.

[12] J. Schmidt, F. Vogt, and H. Niemann, "Robust hand–eye calibration of an endoscopic surgery robot using dual quaternions," in *Pattern Recognition. DAGM 2003. Lecture Notes in Computer Science, vol 2781, Springer, Berlin, Heidelberg*, 2003, pp. 548–556.

[13] M. Zhou, M. Hamad, J. Weiss, A. Eslami, K. Huang, M. Maier, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasseri, "Towards robotic eye surgery: Marker-free, online hand-eye calibration using optical coherence tomography images," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3944–3951, 2018.

[14] K. Koide and E. Menegatti, "General hand–eye calibration based on reprojection error minimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1021–1028, 2019.

[15] Z. Li, *New Methods of Eye-to-Eye Calibration and its Application to Cooperative Localization*. PhD dissertation, Technische Universität Regelungsmethoden und Robotik, 2020.

[16] F. Zhong, Z. Wang, W. Chen, K. He, Y. Wang, and Y.-H. Liu, "Hand-eye calibration of surgical instrument for robotic surgery using interactive manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1540–1547, 2020.

[17] H. H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," in *Proceedings of the 1991 IEEE Conference on Computer Vision and Pattern Recognition*, 1991, pp. 145–146.

[18] Y.-C. Lu and J. C. K. Chou, "Eight-space quaternion approach for robotic hand-eye calibration," in *IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, vol. 4, 1995, pp. 3316–3321.

[19] D. Kim, "Dual quaternion application to kinematic calibration of wrist-mounted camera," *Journal of Robotic Systems*, vol. 13, no. 3, pp. 153–162, 1996.

[20] K. Daniilidis, "Hand-eye calibration using dual quaternions," *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[21] Z. Zhao and Y. Liu, "Hand-eye calibration based on screw motions," in *International Conference on Pattern Recognition*, vol. 3, 2006, pp. 1022–1026.

[22] F. Thomas, "Approaching dual quaternions from matrix algebra," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1037–1048, 2014.

[23] J. Wu, Y. Sun, M. Wang, and M. Liu, "Hand-eye calibration: 4D procrustes analysis approach," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 6, pp. 2966–2981, 2020.

[24] J. Wu, S. Sarabandi, J. M. Porta, M. Liu, and F. Thomas, "Yet a better closed-form formula for the 3D nearest rotation matrix problem," IRI, Tech. Rep. 01-21, 2021.

[25] R.-H. Liang and J.-F. Mao, "Hand-eye calibration with a new linear decomposition algorithm," *Journal of Zhejiang University - SCIENCE A*, vol. 9, no. 10, pp. 1363–1368, 2008.

[26] S. Sarabandi and F. Thomas, "A survey on the computation of quaternions from rotation matrices," *ASME Journal of Mechanisms and Robotics*, vol. 11, no. 2, p. 021006, 2019.

[27] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, 1989.

[28] R. Horaud and F. Dornaika, "Hand-eye calibration," *The International Journal of Robotics Research*, vol. 14, no. 3, pp. 195–210, 1995.

[29] F. C. Park and B. J. 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.

[30] J. C. K. Chou and M. Kamel, "Quaternions approach to solve the kinematic equation of rotation, $A_a A_x = A_x A_b$, of a sensor-mounted robotic manipulator," in *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, 1988, pp. 656–662.

[31] ——, "Finding the position and orientation of a sensor on a robot manipulator using quaternions," *The International Journal of Robotics Research*, vol. 10, no. 3, pp. 240–254, 1991.

[32] H. Zhuang and Z. S. Roth, "Comments on 'Calibration of wrist-mounted robotic sensors by solving homogeneous transform equations of the form $AX = XB$'," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 6, pp. 877–878, 1991.

[33] H. Zhuang, Z. S. Roth, and R. Sudhakar, "Simultaneous robot/world and tool/flange calibration by solving homogeneous transformation equations of the form $AX = YB$," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 4, pp. 549–554, 1994.

[34] S. Sarabandi, A. Shabani, J. M. Porta, and F. Thomas, "On closed-form formulas for the 3D nearest rotation matrix problem," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1333–1339, 2020.

[35] M. Hasan, "Square-root free orthogonalization algorithms," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3173–3176, 2009.

[36] S. Sarabandi, A. Shabani, and F. Thomas, "A survey on the orthonormalization of noisy rotation matrices," *submitted for publication*, 2021.

[37] C.-C. Wang, "Extrinsic calibration of a vision sensor mounted on a robot," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 2, pp. 161–175, 1992.

[38] Franka Emika, "The Panda Robot Description," https://www.franka.de/robot-system, last accessed: 2021-7-6.

[39] E. Lachat, H. Macher, T. Landes, and P. Grussenmeyer, "Assessment and calibration of a RGB-D camera (Kinect v2 sensor) towards a potential use for close-range 3D modeling," *Remote Sensing*, vol. 7, no. 10, pp. 13 070–13 097, 2015.