
Probabilistic Approaches to the $AXB = YCZ$ Calibration Problem in Multi-Robot Systems

**Qianli Ma · Zachariah Goh · Sipu Ruan ·
Gregory S. Chirikjian**

Abstract Interest in multi-robot systems has grown rapidly in recent years. This is due in part to the reduced cost of such systems and in part to the increased difficulty of the tasks that they can address. A multi-robot system is usually composed of several individual robots such as mobile robots or unmanned aerial vehicles. Many problems have been investigated for multi-robot system such as motion planning, collision checking and scheduling. However, not much has been published previously about the calibration problem for multi-robot systems despite the fact that it is the prerequisite for the whole system to operate in a consistent and accurate manner. Compared to the traditional hand-eye & robot-world calibration, a relatively new problem called the $AXB = YCZ$ calibration problem arises in the multi-robot scenario, where A, B, C are time-varying rigid body transformations measured from sensors and X, Y, Z are unknown static transformations to be calibrated. Several solvers have been proposed previously in different application areas that can solve for X, Y and Z simultaneously. However, all of the solvers assume a priori knowledge of the exact temporal correspondence among the data streams $\{A_i\}$, $\{B_i\}$ and $\{C_i\}$. While that assumption

Qianli Ma
Ph.D
Department of Mechanical Engineering
Johns Hopkins University

Zachariah Goh
Senior Engineer
National University of Singapore

Sipu Ruan
Ph.D Candidate
Department of Mechanical Engineering
Johns Hopkins University

Gregory S. Chirikjian
E-mail: gchirik1@jhu.edu
Professor
Department of Mechanical Engineering
Johns Hopkins University

may be justified in some scenarios, in the application domain of multi-robot systems, which may use ad hoc and asynchronous communication protocols, knowledge of this correspondence generally cannot be assumed. Moreover, the existing methods in the literature require good initial estimates that are not always easy or possible to obtain. To address this, we propose two probabilistic approaches that can solve the $AXB = YCZ$ problem without a priori knowledge of the temporal correspondence of the data. In addition, no initial estimates are required for recovering X , Y and Z . These methods are probabilistic in the sense of viewing the sets $\{A_i\}$, $\{B_i\}$, and $\{C_i\}$ as samples drawn from underlying probability density functions. This is what allows these methods to work in the absence of temporal correspondence. However, measurement errors are not explicitly modeled, and so the results are sensitive to the sort of noise that is ubiquitous in real world data. We therefore introduce ways to add robustness to noise, including a hybrid method which combines traditional $AXB = YCZ$ solvers with the probabilistic methodology and an iterative method for refinement to add robustness in the case of noisy experimental data. It is shown that the new algorithm is robust to both noise and the loss of correspondence information in the data. These methods are particularly well suited for multi-robot systems, and also apply to other areas of robotics in which $AXB = YCZ$ arises.

Keywords Calibration · $AXB = YCZ$ · Multi-Robot · Probabilistic Methods

1 Introduction

Many multirobot calibration problems can be formulated using the equation $AXB = YCZ$, where A , B and C are known time-varying homogeneous transformations from sensor readings, and X , Y and Z are unknown static relationships between two target frames. For the dual arm system [22] shown in Fig. (1), the problem becomes the hand-eye (X), robot-robot (Y) and tool-flange (Z) calibration problem where robot 1 holds the camera and robot 2 holds the marker. For a team of mobile robots [8] illustrated in Fig. (2), a triple hand-eye (or camera-marker) calibration problem exists where each robot agent is “looking at” the marker on the next agent. In Fig. (3), the problem of the serial-parallel hybrid robot system [25] is cast as the tool-gripper (X), flange-base (Y) and camera-base (Z) calibrations. The same mathematical modeling also exists in co-robotic ultrasound (US) tomography where two hand-eye and one robot-robot calibrations are needed [1].

Relatively little work has been done on $AXB = YCZ$ calibration. To the best of our knowledge, only Wang et al [22, 24] and Yan et al [25] have proposed several algorithms for solving X , Y and Z simultaneously. Yan *et al* proposed an optimization approach and a Kronecker-product based approach towards calibrating a serial-parallel manipulator whereas Wang *et al* presented an optimization method for the calibration of a cooperative dual-arm system, both of which are formulated as the $AXB = YCZ$ problem.

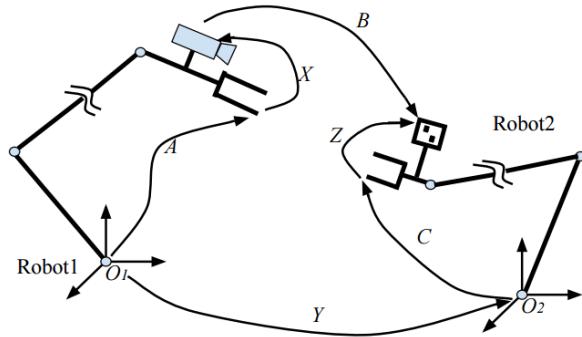


Fig. 1: The Hand-Eye, Robot-Robot, Tool-Flange Calibration of a Dual Arm System

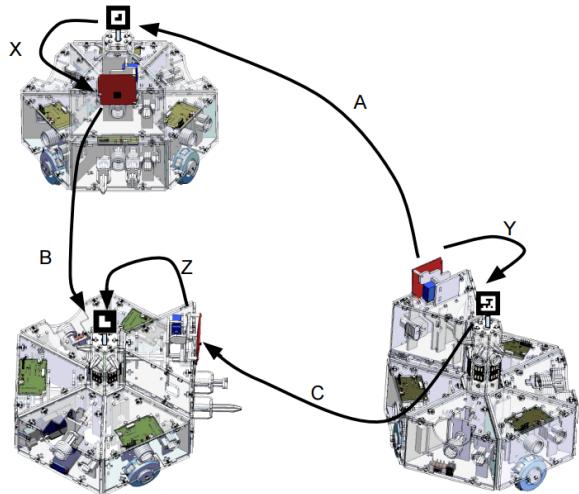


Fig. 2: Triple Hand-Eye Calibration of a Multi-Robot System

1.1 Related Work

The $AXB = YCZ$ problem first originated as an extension of the robot hand-eye problem. The hand-eye calibration problem can be formulated as $AX = XB$, where A and B are homogeneous transformations calculated from sensor readings, and X is the unknown transformation from the mounted sensor (US probe, camera, etc.) to the robot end-effector. Tsai [21] and Shiu [20] were among the first to solve the $AX = XB$ problem. Many other solvers have also been proposed in literature [2, 3, 4, 7, 10, 14, 18, 26]. The hand-eye and robot-world calibration problem is an extension of the hand-eye problem, and is formulated as $AX = YB$ where

X denotes the transformation between the sensor and the end-effector and Y describes the transformation between the robot base frame and the world frame. In this formulation, A and B are the homogeneous transformations measured directly from the sensors. Quite a few $AX = YB$ solvers have been proposed in literature [9, 12, 13, 15, 16, 19, 27]. Most of the existing $AX = XB$ and $AX = YB$ solvers deal with the case where there is an exact correspondence between the data pairs A_i and B_i . However, this is generally not true in real applications due to asynchronous sensors or missing data. [2] and [16] proposed probabilistic approaches for solving the $AX = XB$ and $AX = YB$ problems respectively, and both of them show the superiority of the probabilistic approaches over the traditional solvers when handling data without a priori knowledge of correspondence. Ackerman and Chirikjian[2] handled the data that is completely scrambled whereas Li et al.[16] more focused on recovering the correspondence of shifted data streams using probabilistic methods which can boost the performance of traditional $AX = YB$ solvers. Wang [22, 24] and Yan [25] have proposed several $AXB = YCZ$ algorithms for solving X , Y and Z simultaneously.

1.2 Contribution

This article is a substantially expanded and modified version of the authors' work in the conference paper [17] in which we proposed two "probabilistic" algorithms for solving the $AXB = YCZ$ robot system calibration problem. Due to the different physical properties of robotic systems, these two types of probabilistic $AXB = YCZ$ solvers are built which greatly reduce the need for a priori knowledge of the temporal correspondence between sensor data. We use the word "probabilistic" because the measured datasets $\{A_i\}$, $\{B_i\}$, and $\{C_i\}$ are each replaced with histograms on the space of rigid-body poses, and normalized to be probability densities. That is, while there are no random variables in this problem, the tools of probability and measure theory can still be employed with great benefit. As novel contributions in the present article, we develop a hybrid approach which combines traditional deterministic $AXB = YCZ$ solvers and probabilistic estimates of rigid-body poses, and introduce an iterative post-processing step to add robustness to measurement noise in experimental data. The approach is simple and turns out to be very effective at handling noisy data when there is partial knowledge of the correspondence.

The rest of this article is organized as follows. In Section 2, we introduce some of the fundamental mathematical background. Section 3 describes in detail the formulation of the two probabilistic $AXB = YCZ$ solvers. In Section 4, we perform numerical simulations to compare the probabilistic and traditional $AXB = YCZ$ solvers, and show the effectiveness and robustness of the former. Comparison between the two probabilistic approaches are also performed to show their respective desired application scenarios. Section 5 presents a new hybrid approach which combines traditional solvers with probabilistic methodology to handle noisy data. Section 6 develops a new post-processing step which iterates on the whole set of data and demonstrates greater robustness to measurement noise. In Section 7, physical experiments are performed, and the comparisons between probabilistic and traditional

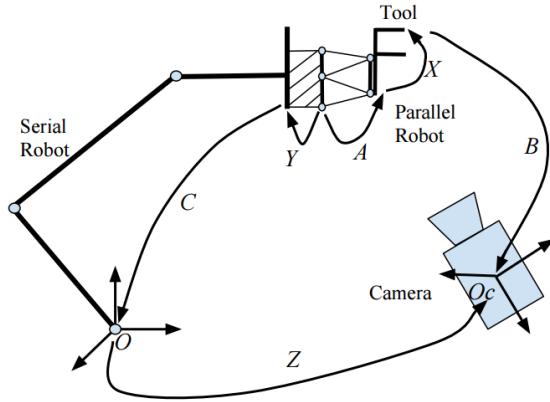


Fig. 3: Flange-Based, Camera-Based and Tool-Gripper Calibration of a Serial-Parallel Manipulator

algorithms show that the former is more robust to the scrambled data and the iterative solver recovers the unknown transformations more effectively. In Section 8, we draw conclusions and point out future directions. Three appendices provide the full mathematical justification for the presented algorithms.

2 Mathematical Background

Before going into the probabilistic solvers for the $AXB = YCZ$ problem, we provide a brief introduction to the concepts of mean, covariance and convolution on the special Euclidean group $SE(3)$.

2.1 Preliminaries on the Special Euclidean Group

The special Euclidean group $SE(3)$ is the space consisting of rigid body transformations of the following form:

$$H(R, t) = \begin{pmatrix} R & t \\ 0^T & 1 \end{pmatrix} \in SE(3), \quad (1)$$

where $t = [t_1, t_2, t_3]^T \in \mathbb{R}^3$ is a translation vector and $R \in SO(3)$, the special orthogonal group consisting of 3×3 rotation matrices with the group operation of matrix multiplication. T denotes the transpose of a vector or matrix and H is the symbol for a group element in $SE(3)$. The properties of the operation of matrix multiplication $H(R_a, t_a)H(R_b, t_b) = H(R_aR_b, R_a t_b + t_a)$ together with the analyticity of the underlying space makes $SE(3)$ a six-dimensional Lie group.

To every Lie group, G , consisting of elements $\{g\}$, there is an associated Lie algebra, \mathfrak{G} , consisting of elements $\{X\}$, which is the tangent space of the Lie group at the

identity element. In the neighborhood of the identity, any $g \in G$ can be parameterized as

$$g = \exp(X) \quad (2)$$

where $X \in \mathfrak{G}$ and \exp is the exponential map, which for matrix Lie groups such as $SO(3)$ and $SE(3)$ is simply matrix exponentiation. For these two groups, the exponential map is surjective, and it is possible to define its inverse (i.e., the logarithm) uniquely except for the set of measure zero corresponding to rotations by an angle of π [5, 6].

The Lie algebra for $SO(3)$ is

$$so(3) \doteq \{\Omega \mid \Omega = -\Omega^T \in \mathbb{R}^{3 \times 3}\}. \quad (3)$$

Specifically,

$$\Omega = \hat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix} \in so(3) \quad (4)$$

where $\omega = [\omega_1, \omega_2, \omega_3]^T \in \mathbb{R}^3$ and $\hat{\cdot}$ is called the “hat” operator that converts a three-dimensional vector into a 3×3 skew symmetric matrix. The inverse “vee” operation \vee does the following

$$\Omega^\vee = \omega. \quad (5)$$

Any rotation matrix can be parameterized as $R = \exp \hat{\omega}$ where $\|\omega\| \leq \pi$, and $\log R$ is uniquely defined as long as $\|\omega\| < \pi$. In the probabilistic formulations that follow, the set of measure zero corresponding to $\|\omega\| = \pi$ is inconsequential and so no distinction is made between $SO(3)$ and $SO_{<}(3)$, which is $SO(3)$ depleted by the set $\{\exp \hat{\omega} \mid \|\omega\| = \pi\}$.

For the Lie algebra of $SE(3)$, the following expression can be evaluated for any $\xi = [\xi_1, \xi_2, \xi_3, \xi_4, \xi_5, \xi_6]^T \in \mathbb{R}^6$:

$$\hat{\xi} = \begin{pmatrix} 0 & -\xi_3 & \xi_2 & \xi_4 \\ \xi_3 & 0 & -\xi_1 & \xi_5 \\ -\xi_2 & \xi_1 & 0 & \xi_6 \\ 0 & 0 & 0 & 0 \end{pmatrix} \in se(3). \quad (6)$$

This defines the “hat” operator for $se(3)$ as $\hat{\cdot}: \mathbb{R}^6 \rightarrow se(3)$. This maps a six-dimensional vector to its corresponding Lie algebra element. The “vee” operator does the reverse. The use of the hat and vee notation to mean two different things for $so(3)$ and $se(3)$ should not be a source of confusion, because the case under consideration is always clear from the object to which they are applied.

Any $H \in SE(3)$ can be parameterized as $H = \exp(\hat{\xi})$ where the first three entries of ξ are contained in a solid closed ball of radius π and $[\xi_4, \xi_5, \xi_6]^T \in \mathbb{R}^3$. If the discussion is restricted to the open solid ball, then the result of the exponential map is $SE_{<}(3)$, which is $SE(3)$ depleted by a set of measure zero. Technically, the logarithm can be defined uniquely only on $SE_{<}(3)$. But the distinction between $SE(3)$ and $SE_{<}(3)$ can be blurred when integrating over $SE(3)$, and it is acceptable to compute $\xi = \log^\vee(H) \in \mathbb{R}^6$ for all $H \in SE(3)$ by ignoring any contributions from $SE(3) - SE_{<}(3)$ in integrals.

The adjoint for $SE(3)$ is defined as

$$Ad(H(R, t)) = \begin{pmatrix} R & \mathbb{O}_3 \\ \hat{t}R & R \end{pmatrix} \in \mathbb{R}^{6 \times 6} \quad (7)$$

where $\hat{t} \in so(3)$.

If $\xi \in \mathbb{R}^6$ is partitioned as $\xi = [\omega^T, v^T]^T$, then adjoint for $se(3)$ is defined as

$$ad(\hat{\xi}) = \begin{pmatrix} \hat{\omega} & \mathbb{O}_3 \\ \hat{v} & \hat{\omega} \end{pmatrix} \in \mathbb{R}^{6 \times 6}. \quad (8)$$

The relationship between ad and Ad is

$$\exp ad(\hat{\xi}) = Ad(\exp \hat{\xi}). \quad (9)$$

The following identities will be used later:

$$Ad(H^{-1}) = Ad^{-1}(H) = \begin{pmatrix} R^T & \mathbb{O} \\ -\widehat{R^T t R^T} & R^T \end{pmatrix} \quad (10)$$

and

$$Ad(H_1)Ad(H_2) = Ad(H_1 H_2). \quad (11)$$

2.2 Probability Densities on $SE(3)$

There is a unique and correct way to define integration on $SE(3)$ called the ‘‘Haar measure’’ denoted as dH . In particular, if R is expressed in ZXZ Euler angles (α, β, γ) , then the Haar measure can be written as $dH = \sin \beta d\alpha d\beta d\gamma dt_1 dt_2 dt_3$ to within an arbitrary scale factor. In this context, the convolution of two well-behaved functions can be defined as [5, 6]:

$$(f_1 * f_2)(H) = \int_{SE(3)} f_1(K)f_2(K^{-1}H) dK \quad (12)$$

where $K, H \in SE(3)$. The integral over $SE(3)$ can be expressed in various coordinates and here we choose the exponential coordinates. The six-dimensional integral over $SE(3)$ and its measure dK expressed in these coordinates can be found in [23].

As it will be used in later sections, we define a Dirac delta function on $SE(3)$, $\delta(H)$, by the properties

$$(f * \delta)(H) = \int_{SE(3)} f(K)\delta(K^{-1}H) dK = f(H), \quad (13)$$

and

$$\int_{SE(3)} \delta(H)dH = 1. \quad (14)$$

Informally, we can think of $\delta(H)$ as an infinite spike of the form

$$\delta(H) = \begin{cases} +\infty & \text{if } H = \mathbb{I}_4 \\ 0 & \text{if } H \neq \mathbb{I}_4 \end{cases} \quad (15)$$

for any $H \in SE(3)$, and \mathbb{I}_4 denotes a 4 by 4 identity matrix. Given $A \in SE(3)$, the corresponding shifted version of the Dirac delta function can be defined as $\delta_A(H) = \delta(A^{-1}H)$.

In the probabilistic approaches to be introduced, all $SE(3)$ elements will be represented by their mean M and covariance Σ . Given a probability density function (PDF) as $f(H)$, its mean M and covariance Σ are defined to satisfy the following equations [5, 23]:

$$\int_{SE(3)} \log(M^{-1}H) f(H) dH = \mathbb{O}_4 \quad (16a)$$

$$\Sigma = \int_{SE(3)} \log^\vee(M^{-1}H) [\log^\vee(M^{-1}H)]^T f(H) dH \quad (16b)$$

where \mathbb{O}_4 denotes the 4×4 zero matrix.

Given a PDF describing $\{A_i\}$ as $f_A(H)$, where $i = 1, \dots, n$, the corresponding discrete versions of the sample mean and sample covariance are:

$$\sum_{i=1}^n \log(M_A^{-1} A_i) = \mathbb{O}_4 \quad (17a)$$

$$\Sigma_A = \sum_{i=1}^n \log^\vee(M_A^{-1} A_i) [\log^\vee(M_A^{-1} A_i)]^T. \quad (17b)$$

Numerically, given a set of A_i , the mean M_A can be solved in an iterative manner as described in [23]. In the context of $AXB = YCZ$ calibration, M_B and M_C can be computed in a similar fashion, and Σ_A , Σ_B and Σ_C are straightforward to compute given M_A , M_B and M_C .

Given two PDFs f_1 and f_2 on $SE(3)$, the sample mean and sample covariance of their convolution $(f_1 * f_2)(g)$ can be approximated to a high order of accuracy as [23]:

$$M_{1*2} = M_1 M_2 \quad (18a)$$

$$\Sigma_{1*2} = Ad(M_2^{-1}) \Sigma_1 Ad^T(M_2^{-1}) + \Sigma_2 + F(A, B) \quad (18b)$$

where $F(\cdot; \cdot)$ is defined as

$$\begin{aligned} F(A, B) &= \frac{1}{4} \sum_{i,j=1}^6 ad(E_i) B ad(E_j)^T A_{ij} \\ &+ \frac{1}{12} \left\{ \left[\sum_{i,j=1}^6 A'_{ij} \right] B + B^T \left[\sum_{i,j=1}^6 A'_{ij} \right]^T \right\} \\ &+ \frac{1}{12} \left\{ \left[\sum_{i,j=1}^6 B'_{ij} \right] A + A^T \left[\sum_{i,j=1}^6 B'_{ij} \right]^T \right\} \end{aligned} \quad (19)$$

with $A = Ad(M_2^{-1})\Sigma_1 Ad^T(M_2^{-1})$, $B = \Sigma_2$, and

$$A'_{ij} = ad(E_i)ad(E_j)A_{ij} \quad (20)$$

$$B'_{ij} = ad(E_i)ad(E_j)B_{ij} \quad (21)$$

with E_i denoting the $\hat{\cdot}$ of the i^{th} unit basis vector in \mathbb{R}^6 .

The $F(\cdot, \cdot)$ function is bi-linear in the sense that

$$F(\alpha\Sigma_1 + \alpha'\Sigma'_1, \Sigma_2) = \alpha F(\Sigma_1, \Sigma_2) + \alpha' F(\Sigma'_1, \Sigma_2) \quad (22)$$

and

$$F(\Sigma_1, \beta\Sigma_2 + \beta'\Sigma'_2) = \beta F(\Sigma_1, \Sigma_2) + \beta' F(\Sigma_1, \Sigma'_2) \quad (23)$$

for arbitrary constants $\alpha, \alpha', \beta, \beta'$ and arbitrary symmetric matrices $\Sigma_1, \Sigma'_1, \Sigma_2, \Sigma'_2$. In particular, if one of these constants is zero, then

$$F(\mathbb{O}, \Sigma_2) = F(\Sigma_1, \mathbb{O}) = \mathbb{O} \quad (24)$$

and if both $f_1(H)$ and $f_2(H)$ are “highly-focused” in the sense that $\|\Sigma_1\| \ll 1$ and $\|\Sigma_2\| \ll 1$, then $F(\Sigma_1, \Sigma_2) \approx \mathbb{O}$.

For the three-fold convolution, the mean and covariance can be written as:

$$M_{1*2*3} = M_1 M_2 M_3$$

and

$$\begin{aligned} \Sigma_{(1*2)*3} &= Ad^{-1}(M_2 M_3) \Sigma_1 Ad^{-T}(M_2 M_3) + Ad^{-1}(M_3) \Sigma_2 Ad^{-T}(M_3) \\ &\quad + Ad^{-1}(M_3) F(Ad^{-1}(M_2) \Sigma_1 Ad^{-T}(M_2); \Sigma_2) Ad^{-T}(M_3) + \Sigma_3 \\ &\quad + F(Ad^{-1}(M_2 M_3) \Sigma_1 Ad^{-T}(M_2 M_3); \Sigma_3) + \\ &\quad F(Ad^{-1}(M_3) \Sigma_2 Ad^{-T}(M_3); \Sigma_3) \\ &\quad + F(Ad^{-1}(M_3) F(Ad^{-1}(M_2) \Sigma_1 Ad^{-T}(M_2); \Sigma_2) Ad^{-T}(M_3); \Sigma_3). \end{aligned}$$

By the properties of $F(\cdot, \cdot)$, if either $\Sigma_1 = \Sigma_2 = \mathbb{O}$ or $\Sigma_2 = \Sigma_3 = \mathbb{O}$, then all the $F(\cdot, \cdot)$ terms in the covariance Σ_{1*2*3} vanish (please see Appendix B for detailed derivation). This will be useful in our approach to solving the $AXB = YCZ$ problem described in the following section.

3 Problem Formulation

In this section, we formulate the two probabilistic approaches for the $AXB = YCZ$ problem. They share a common theoretical underpinning but are designed for different types of robotic systems. Note that for the three types of robotic systems described in Fig. 1, Fig. 2 and Fig. 3, different types of constraints can be applied onto the datasets $\{A_i\}$, $\{B_i\}$ and $\{C_i\}$. For the multi-mobile robotic system, any two robot agents can remain static with the third agent moving freely. Or equivalently, any one of A , B and C can be fixed without fixing the other two. For the dual-arm and serial-parallel robotic systems, only A or C can be fixed without fixing the other

two. This is because B describes the transformation between the marker frame and the camera frame, while A and C are solely determined using the forward kinematics of the robots. Hence it is very difficult to keep B constant while varying A and C . We employ the above physical properties of the systems to build the fundamental framework for the probabilistic approaches.

3.1 Mathematical Approach

Given a large set of triples $(A_i, B_i, C_i) \in SE(3) \times SE(3) \times SE(3)$ where $i = 1, \dots, n$, the following equation can be obtained:

$$A_i X B_i = Y C_i Z. \quad (25)$$

Using the shifting property of Dirac delta function, we have

$$(\delta_{A_i} * \delta_X * \delta_{B_i})(H) = \delta(B_i^{-1} X^{-1} A_i^{-1} H) \quad (26a)$$

$$(\delta_Y * \delta_{C_i} * \delta_Z)(H) = \delta(Z^{-1} C_i^{-1} Y^{-1} H). \quad (26b)$$

The above two equations can be combined to replace Eq. (25), with

$$(\delta_{A_i} * \delta_X * \delta_{B_i})(H) = (\delta_Y * \delta_{C_i} * \delta_Z)(H). \quad (27)$$

An analogous approach was taken previously in the $AX = XB$ and $AX = YB$ problems [2, 16]. Converting the original equations to functional equations has two advantages. First, it allows them to be summed over all values of i because it always makes sense to add scalar functions, whereas it does not make sense to add group elements. Second, it allows us to use the covariance propagation formulas presented earlier. The difference between the $AXB = YCZ$ problem and the others is that A_i and B_i are on the same side of the equation. Therefore, if data were being recorded in an uncontrolled way, the approximation

$$\frac{1}{n} \sum_i \delta_{A_i} * \delta_X * \delta_{B_i} \approx \left(\frac{1}{n} \sum_i \delta_{A_i} \right) * \delta_X * \left(\frac{1}{n} \sum_i \delta_{B_i} \right)$$

would need to be made on the left-hand side of Eq. (25). Such an approximation can be justified if the covariances Σ_A and Σ_B are small, but there is no need to resort to this approximation, because during the calibration process we have control over A , B , and C , and can choose to hold one of these fixed.

The relationship in Eq. (27) obviously still holds if we fix A_i to be a constant transformation A :

$$(\delta_A * \delta_X * \delta_{B_i})(H) = (\delta_Y * \delta_{C_i} * \delta_Z)(H). \quad (28)$$

Fixing A (or B or C) while letting the other two sensor streams take various values is practical by employing the physical properties of the robotic systems as described above. We only provide the derivation for fixed A , as the derivations for fixed B or

C follow in a similar way, and the results for all three cases are summarized later in Table 1. Next, define the PDF of $\{K_i\}$ as:

$$f_K(H) = \frac{1}{n} \sum_{i=1}^n \delta_{K_i}(H) \quad (29)$$

where $K \in \{A, B, C\}$. If we use the bi-linearity of convolution, add n instances of Eq. (??) and substitute Eq. (29) into the summation, the Eq. (32) can be achieved with intermediate steps shown from Eq. (30) to Eq. (31):

$$\frac{1}{n} \sum_{i=1}^n (\delta_A * \delta_X * \delta_{B_i})(H) = \frac{1}{n} \sum_{i=1}^n (\delta_Y * \delta_{C_i} * \delta_Z)(H). \quad (30)$$

$$\left(\delta_A * \delta_X * \frac{1}{n} \sum_{i=1}^n \delta_{B_i} \right) (H) = \left(\delta_Y * \frac{1}{n} \sum_{i=1}^n \delta_{C_i} * \delta_Z \right) (H). \quad (31)$$

$$(\delta_A * \delta_X * f_B)(H) = (\delta_Y * f_C * \delta_Z)(H). \quad (32)$$

Then by employing Eq. (18a) twice, we get the mean equation of $AXB = YCZ$ as

$$M_A M_X M_B = M_Y M_C M_Z. \quad (33)$$

Because X, Y, Z and A are all single elements of $SE(3)$, $M_X = X$, $M_Y = Y$, $M_Z = Z$, $M_A = A$ and $\Sigma_X = \mathbb{O}$, $\Sigma_Y = \mathbb{O}$, $\Sigma_Z = \mathbb{O}$, $\Sigma_A = \mathbb{O}$. Eq. (33) then becomes

$$AXM_B = YM_C Z \quad (34)$$

The covariance equation is obtained by first computing Σ_{A*X} and then Σ_{A*X*B} as:

$$\Sigma_{A*X*B} = Ad(B^{-1})Ad(X^{-1})\Sigma_A Ad^T(X^{-1})Ad^T(B^{-1}) + \Sigma_B \quad (35a)$$

$$= \Sigma_B \quad (35b)$$

Similarly, Σ_{Y*C*Z} can be obtained as:

$$\Sigma_{Y*C*Z} = Ad(Z^{-1})\Sigma_C Ad^T(Z^{-1}) \quad (36)$$

Therefore, by equating Eq. (35a) and Eq. (36), the covariance equation for $AXB = YCZ$ with A fixed becomes

$$\Sigma_B = Ad(Z^{-1})\Sigma_C Ad^T(Z^{-1}) \quad (37)$$

Eq. (34) and Eq. (37) are the fundamental tools to solve for the unknowns without a priori knowledge of the correspondence among sensor measurements.

In order to decompose Eq. (37) into sub-equations, define the covariance matrix as

$$\Sigma_H = \begin{pmatrix} \Sigma_H^1 & \Sigma_H^2 \\ \Sigma_H^3 & \Sigma_H^4 \end{pmatrix} \in \mathbb{R}^{6 \times 6} \quad (38)$$

No.	Representation	Fixing	Simplified Sig-Rot
1	$AXB = YCZ$	A	$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z$
2	$A^{-1}YC = XBZ^{-1}$		
3	$BZ^{-1}C^{-1} = X^{-1}A^{-1}Y$	B	$R_C \Sigma_C^1 R_C^T = R_Y^T R_A \Sigma_A^1 R_A^T R_Y$
4	$B^{-1}X^{-1}A^{-1} = Z^{-1}C^{-1}Y^{-1}$		
5	$C^{-1}Y^{-1}A = ZB^{-1}X^{-1}$	C	$R_B \Sigma_B^1 R_B^T = R_X^T \Sigma_A^1 R_X$
6	$CZB^{-1} = Y^{-1}AX$		

Table 1: The simplified Sig-Rot equations after fixing A , B or C

where $H = A, B, C$ and $\Sigma_H^i \in \mathbb{R}^{3 \times 3}$. To simplify the notation, we define $U = \hat{t}$. Substitute Eq. (38) into Eq. (37) and one gets the upper left block as

$$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z, \quad (39)$$

and the upper right block as

$$\Sigma_B^2 = R_Z^T \Sigma_C^1 R_Z U_Z^T + R_Z^T \Sigma_C^2 R_Z \quad (40)$$

For convenience, we call Eq. (39) the Sig-Rot equation and Eq. (40) the Sig-Trans equation. Sig-Rot equation contains only the rotational information from the unknown matrices while Sig-Trans equation contains both the unknown rotations and translations. These two equations are not sufficient to solve the problem since Eq. (39) contains only R_X and R_Z , and Eq. (40) contains only t_X and t_Z in addition to the above two rotations, whereas Y is “lost” in the covariance equation. However, it turns out that by rearranging the order of X, Y, Z and fixing different sensor data (A, B, C), similar equations to Eq. (39) and Eq. (40) can be obtained to solve for the unknown transformations.

There are a total of six variations of $AXB = YCZ$ formulations. If we write $AXB = YCZ$ as $AXBZ^{-1}C^{-1}Y^{-1} = \mathbb{I}$, premultiply it by A^{-1} and postmultiply it by A on both sides of the equation, we have $XBX^{-1}C^{-1}Y^{-1}A = \mathbb{I}$ which “moves” A from the left to the right. The same operation can be done in turn for X, B, Z^{-1}, C^{-1} and Y^{-1} and these give a total of six variations as shown in the “Representation” column of Table 1. For simplicity, we only list the Sig-Rot equations and leave out the Sig-Trans equations. In the next part, we present the frameworks for solving the calibration problem for each of these two types of systems.

3.2 Two Versions of $AXB = YCZ$ Calibration

Before presenting the two versions, we present Theorem 1 to simplify the computation of the mean and covariance of H^{-1} .

Theorem 1 *If the mean and covariance are M and Σ for a PDF $f(H)$, then the mean and covariance for $f(H^{-1})$ are M^{-1} and $Ad(M)\Sigma Ad^T(M)$ respectively.*

Please refer to the Appendix A for the proof. Theorem 1 provides a simple way to calculate the mean and covariance of $f(H^{-1})$, which is very useful due to the frequent calculations of PDFs on the inverses of $\{A, B, C\}$. Another equation extracted from Eq. (81) is

$$\Sigma_{K^{-1}}^1 = R_K \Sigma_K^1 R_K^T. \quad (41)$$

which is useful when converting the Sig-Rot equations into the simplified versions shown in the last column of Table 1.

3.2.1 Version 1

For the dual-arm and serial-parallel systems, we show that X , Y and Z can be recovered without a priori knowledge of the correspondence between the data. This is achieved by fixing A and C to give datasets I and II respectively. When A is fixed, or equivalently $A = A_I$, datasets $\{B_{Ii}\}$ and $\{C_{Ij}\}$ can be measured where $i, j = 1, \dots, n$. According to Table 1, we have

$$\Sigma_{B_I}^1 = R_Z^T \Sigma_{C_I}^1 R_Z. \quad (42)$$

Remember that the above equation is achieved using $\Sigma_{A_I} = \mathbb{O}$. However, this zero constraint on Σ_{A_I} applies to neither Rep.3 nor Rep.6, where Rep.3 and Rep.6 denote the No.3 and No.6 Representation equations in Table 1 respectively. When A is fixed to A_I , the right hand side of Rep.6, namely $Y^{-1} A_I X$, becomes a single “point” on $SE(3)$, whereas both C_{Ii} and B_{Ij}^{-1} are PDFs on $SE(3)$. The corresponding convolution equation of Rep.6 becomes

$$(f_{C_I} * \delta_Z * f_{B_I^{-1}})(H) = (\delta_{Y^{-1}} * \delta_{A_I} * \delta_X)(H). \quad (43)$$

which does not hold because the convolution of PDFs is a general PDF instead of a Dirac delta function. Therefore, the underlying constraint of every convolution equation is that there should be at least one non-trivial PDF on both sides of the equation, and we call it the balanced-PDF constraint. This constraint simply means that there is a sufficiently rich set of calibration motions so that those covariances corresponding to varying quantities are full rank, and that the choice of quantities being fixed during each segment of the calibration process are complementary. For example, if A is fixed and B and C are allowed to vary, this does not provide independent information as compared to holding B fixed and allowing A and C to vary. We either hold A or B fixed, and then hold C fixed. Other solvers also require a rich enough set of input data to function.

As shown in [2], Σ_B^1 and Σ_C^1 have the same eigenvalues due to the fact that Eq. (42) is a similarity transformation between Σ_B^1 and Σ_C^1 . Calculate the eigendecomposition of Σ_B and Σ_C as $\Sigma_B^1 = Q_B \Lambda Q_B^T$ and $\Sigma_C^1 = Q_C \Lambda Q_C^T$ where Λ denotes the diagonal matrix. Substitute these two equations into Eq. (42), and we have

$$\Lambda = \underbrace{Q_B^T R_Z^T Q_C}_{\mathcal{Q}} \Lambda Q_C^T R_Z Q_B = \mathcal{Q} \Lambda \mathcal{Q}^T. \quad (44)$$

According to [2], the special structure of Eq. (44) gives four solutions for \mathcal{Q} . Thus, we also get four candidates of R_Z as:

$$R_Z = Q_{C_I} \mathcal{Q} Q_{B_I}^T. \quad (45)$$

For the translation part t_Z , Eq. (40) can be simplified as

$$\Sigma_{B_I}^2 = R_Z^T \Sigma_{C_I}^1 R_Z U_Z^T + R_Z^T \Sigma_{C_I}^2 R_Z \quad (46)$$

and $t_Z = U_Z^\vee$ can be solved directly.

Similarly, when fixing $C \equiv C_{II}$, $\Sigma_{C_{II}} = \Sigma_{C_{II}^{-1}} = \mathbb{O}$, the Sig-Rot equation of Rep.6 is

$$\Sigma_{B_{II}^{-1}}^1 = R_X^T \Sigma_{A_{II}}^1 R_X. \quad (47)$$

Recall that this leads to an equation with structure similar to Eq. (44) and so \mathcal{Q} has four possibilities, and the four candidates of R_X can be calculated as

$$R_X = Q_{A_{II}} \mathcal{Q} Q_{B_{II}^{-1}}^T \quad (48)$$

There are two possible methods to recover Y . One method is to apply $\Sigma_C^1 = \mathbb{O}$ to Sig-Rot.2 to get

$$R_C^T R_Y^T \Sigma_{A^{-1}}^1 R_Y R_C = R_{Z^{-1}}^T \Sigma_B^1 R_{Z^{-1}}, \quad (49)$$

and hence we obtain a total of sixteen candidates of R_Y that are based on the candidates of \mathcal{Q} and R_Z :

$$R_Y = Q_{A^{-1}} \mathcal{Q} Q_B^T R_Z^T R_C^T. \quad (50)$$

The other method is to employ the mean equations to recover Y using the candidates of X and Z as

$$Y = A_I X M_{B_I} Z^{-1} M_{C_I}^{-1} \quad (51)$$

and

$$Y = M_{A_{II}} X M_{B_{II}} Z^{-1} C_{II}^{-1} \quad (52)$$

Hence the second approach gives a total of $16 + 16 = 32$ candidates of Y . When numerically simulating the two approaches above, the second approach is better in terms of generating candidates of Y that are closer to the ground truth, whereas the first one is more likely to result in candidates far from the true Y .

The solution for t_Z and t_X becomes trivial once R_Z and R_X are known. Using the second approach to compute Y , we will have a total of $4 \times 4 \times 32 = 512$ combinations of $\{X, Y, Z\}$. In order to filter out the best combination among the 512 choices, the two datasets can be used to minimize an objective function. For simplicity, let $M_{L_I} = A_I X_i M_{B_I}$, $M_{R_I} = Y_j M_{C_I} Z_k$, $M_{L_{II}} = M_{A_{II}} X_i M_{B_{II}}$ and $M_{R_{II}} = Y_j C_{II} Z_k$.

It turns out that the objective function is critical in getting an optimal X, Y, Z consistently out of the possible 512 candidates. We tried a few functions and found that this function

$$\begin{aligned} \min & \| \log^{\vee}(R_{M_{L_I}}^T R_{M_{R_I}}) \|_2 + \| \log^{\vee}(R_{M_{L_{II}}}^T R_{M_{R_{II}}}) \|_2 \\ & w \cdot \| t_{M_{L_I}} - t_{M_{R_I}} \|_2 + w \cdot \| t_{M_{L_{II}}} - t_{M_{R_{II}}} \|_2 \end{aligned} \quad (53)$$

where $i = 1, \dots, 4$, $j = 1, \dots, 4$, $k = 1, \dots, 32$ has the highest success rate of picking the optimal X, Y, Z . Here w is the weighting factor and can be varied depending on the precision requirement on rotation and translation. Different X, Y, Z will be selected given different w , and we settled on $w = 1.5$ for the simulation.

3.2.2 Version 2

For the multi-robot hand-eye calibration problem, a less restrictive approach exists to solve for X, Y and Z . In addition to fixing A or C , we can also fix B , and this will produce three datasets that are labeled as follows.

Dataset I: $A = A_I$ with $\{B_{Ii}\}$ and $\{C_{Ij}\}$

$$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z. \quad (54)$$

Dataset II: $B = B_{II}$ with $\{A_{IIi}\}$ and $\{C_{IIj}\}$

$$\Sigma_{C^{-1}}^1 = R_Y^T \Sigma_{A^{-1}}^1 R_Y. \quad (55)$$

Dataset III: $C = C_{III}$ with $\{A_{IIIi}\}$ and $\{B_{IIIj}\}$

$$\Sigma_{B^{-1}}^1 = R_X^T \Sigma_A^1 R_X. \quad (56)$$

Under this situation, X, Y and Z are solved independently and there are a total of $4 \times 4 \times 4 = 64$ combinations of solutions. By letting $M_{L_{III}} = M_{A_{III}} X_i M_{B_{III}}$ and $M_{R_{III}} = Y_j C_{III} Z_k$, we can form the following objective function using all 3 datasets:

$$\begin{aligned} \min & \| \log^{\vee}(R_{M_{L_I}}^T R_{M_{R_I}}) \|_2 + \| \log^{\vee}(R_{M_{L_{II}}}^T R_{M_{R_{II}}}) \|_2 \\ & \| \log^{\vee}(R_{M_{L_{III}}}^T R_{M_{R_{III}}}) \|_2 + w \cdot \| t_{M_{L_I}} - t_{M_{R_I}} \|_2 + \\ & w \cdot \| t_{M_{L_{II}}} - t_{M_{R_{II}}} \|_2 + w \cdot \| t_{M_{L_{III}}} - t_{M_{R_{III}}} \|_2 \end{aligned} \quad (57)$$

where $i = 1, \dots, 4$, $j = 1, \dots, 4$ and $k = 1, \dots, 4$.

4 Numerical Comparison between Probabilistic and Traditional Methods

In this section, we compared our probabilistic approaches numerically with the existing methods in the literature. For convenience, we called the two versions of the probabilistic method presented previously *Prob1* and *Prob2* respectively. In [25], two approaches were proposed for solving the $AXB = YCZ$ problem: one is called the *DK* method while the other is the *PN* method. In [22], a simultaneous $AXB = YCZ$ solver was introduced and we call it *Wang* in this article. Note that all of the three

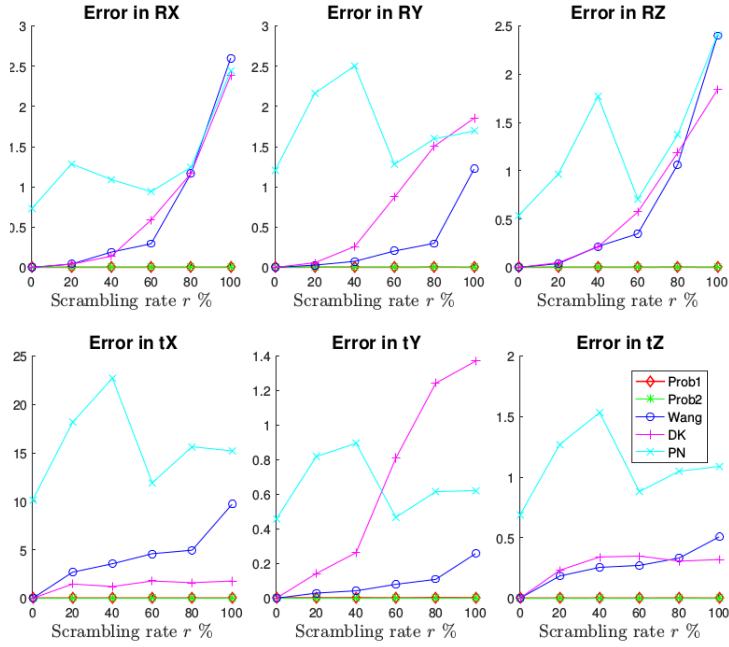


Fig. 4: Rotation/Translation Errors in X, Y, Z v.s. scrambling rate for 10 trials and 100 measurements

methods in the literature require a priori knowledge of the exact correspondence between the datasets $\{A_i\}$, $\{B_i\}$ and $\{C_i\}$, and in this section we refer to them as the “traditional methods”. We performed numerical simulations on both the traditional and probabilistic methods to show that: 1) probabilistic approaches showed superior performance when dealing with data that has little or no correspondence compared to traditional solvers; 2) *Prob2* performed better than *Prob1* when the former had complete datasets.

There are several things to pay attention to when comparing the probabilistic approaches with the traditional methods. Firstly, *PN* is an unconstrained nonlinear optimization algorithm which requires multiple initial guesses of X , Y and Z to achieve an almost global minimum solution. Secondly, *Wang* is a least-squares-based search algorithm that requires good initial guesses of at least two of R_X , R_Y and R_Z . Thirdly, both of them are simultaneous approaches meaning that none of A , B or C needs to be fixed during the calibration process. Lastly, *DK* is a separable method requiring A or C to be fixed during the calibration process. However, no initial guesses are needed to obtain the final result.

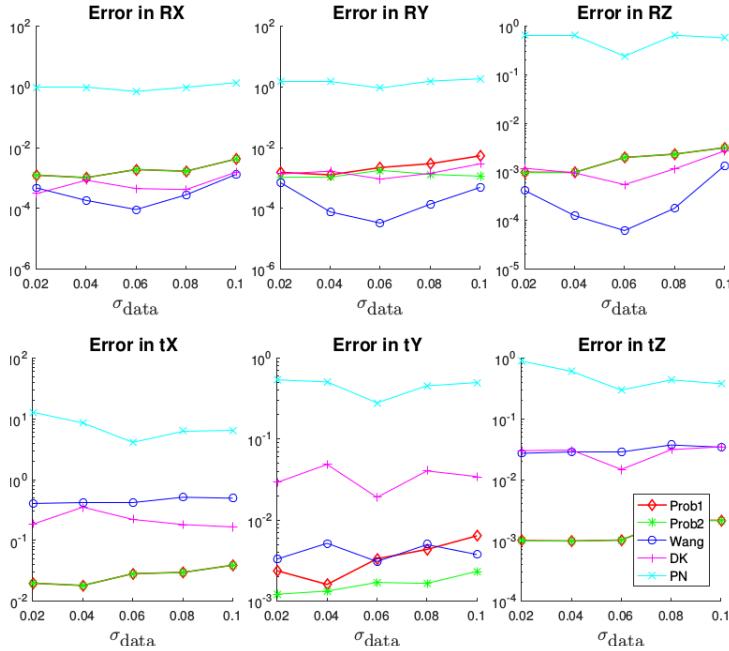


Fig. 5: Rotation/Translation errors v.s. standard deviation of measurement data for $r = 1\%$ and $\sigma_{\text{noise}} = 0$

4.1 Data Generation and Error Metrics

In order to compare all of the five $AXB = YCZ$ solvers together, we generated the simulated datasets as follows. First, we fixed A such that $A = A_I$, and $\{B_{Ii}\}$ are given by

$$B_{Ii} = \exp(\hat{\delta}_i)B_{I0} \quad (58a)$$

$$\delta_i \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \quad (58b)$$

where the mean $\mu = \mathbf{0} \in se(3)$, the covariance matrix $\Sigma = \sigma_{\text{data}}\mathbb{I}_6 \in \mathbb{R}^{6 \times 6}$ and $i = 1, 2, \dots, 100$. The hat operator $\hat{\cdot}$ converts a 6 by 1 vector into its corresponding Lie algebra in $se(3)$. Given the ground truth of X , Y and Z , $\{C_{Ii}\}$ is generated by

$$C_{Ii} = Y^{-1}A_I X B_{Ii} Z^{-1}, \quad (59)$$

and we call this dataset I.

Then, we generated dataset II where we fixed C such that $C = C_{II}$, and generated B_{IIi} and A_{IIi} in a similar fashion:

$$B_{IIi} = \exp(\hat{\delta}_i)B_{II0} \quad (60a)$$

$$A_{IIi} = Y C_{II} Z B_{IIi}^{-1} X^{-1} \quad (60b)$$

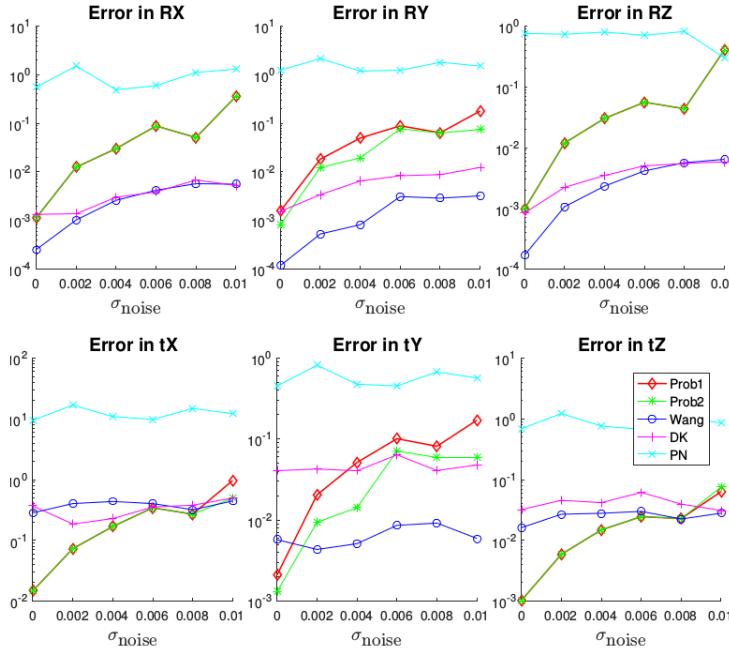


Fig. 6: Rotation/Translation errors v.s. standard deviation of noise applied to the data for $r = 1\%$ and $\sigma_{\text{data}} = 0.02$

Lastly, dataset III was obtained by fixing B such that $B = B_{III}$, and $\{A_{IIIi}\}$, $\{C_{IIIi}\}$ were given by

$$A_{IIIi} = \exp(\hat{\delta}_i) A_{III0} \quad (61a)$$

$$C_{IIIi} = Y^{-1} A_{III} X B_{IIIi} Z^{-1}. \quad (61b)$$

In each dataset, the number of measurement data for A, B, C is 100, i.e. $i = 1, \dots, 100$. Note that there were a total of three datasets but only the first two could be applied on *DK* and *Prob1* methods, but all three sets could be used by *PN*, *Wang* and *Prob2* methods. In order to compare the methods, the datasets being passed into each method are indicated by checkmarks in Table 2. The recovered X, Y and Z were compared with the actual transformations using the following metrics for the errors in rotation and translation:

$$\text{Error}(R_H) = \| \log^V (R_{H_{\text{solved}}}^T R_{H_{\text{true}}}) \| \quad (62a)$$

$$\text{Error}(t_H) = \| t_{H_{\text{solved}}} - t_{H_{\text{true}}} \| / \| t_{H_{\text{true}}} \| \quad (62b)$$

where $H = X, Y, Z$.

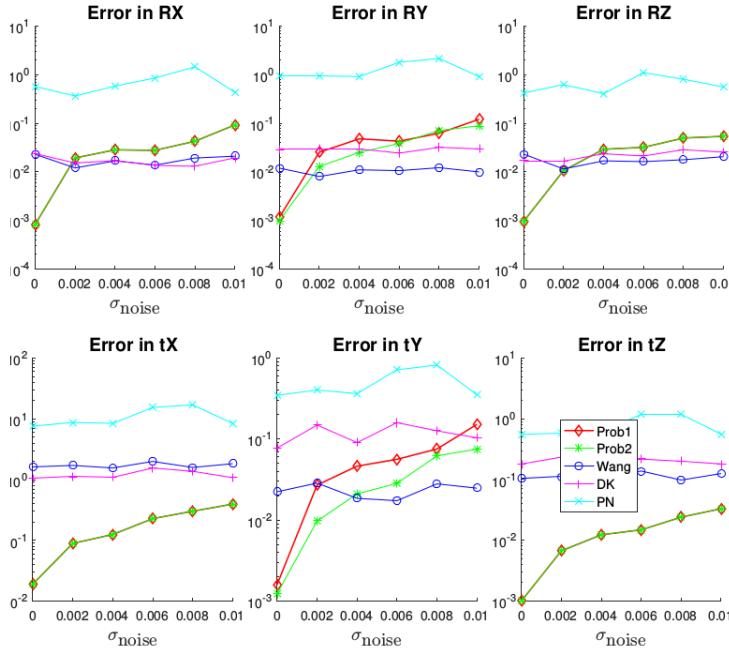


Fig. 7: Rotation/Translation errors v.s. standard deviation of noise applied to the data for $r = 10\%$ and $\sigma_{\text{data}} = 0.02$

Dataset	Prob1	Prob2	Wang	PN	DK
I $A_I, \{B_{Ii}\}, \{C_{Ii}\}$	✓	✓	✓	✓	✓
II $C_{II}, \{B_{IIIi}\}, \{A_{IIIi}\}$	✓	✓	✓	✓	✓
III $B_{III}, \{A_{IIIi}\}, \{C_{IIIi}\}$	✗	✓	✓	✓	✗

Table 2: Datasets used on each method

4.2 Simulation and Discussion

To compare all the five algorithms comprehensively, we performed numerical simulations by varying

1. the scrambling rate r ,
2. standard deviation σ_{data} for generating the measurement data,
3. noise level σ_{noise} .

For each set of conditions, we ran 10 trials and plotted the average error of the computed X, Y, Z from the true values. For experiments (2) and (3) the range of the errors were very big across all the methods. Hence we used the logarithm scale for the vertical axis in Fig. (5), Fig. (6) and Fig. (7).

First, given the three sets of $\{A\}$, $\{B\}$ and $\{C\}$, we scrambled r percent of the dataset $B_{Ii}, A_{IIi}, C_{IIIi}$ where $r = 0\%, 20\%, 40\%, 60\%, 80\%, 100\%$. This means that r percent of the elements in each set have been scrambled by a random permuta-

tion. Since these permutations are random and include the possibility of leaving some of the r percent of the data in the correct temporal correspondence, they changed the order of up to r percent (rather than the full r percent) as compared to the original unscrambled data sequence. This is why the error plots for the method that assumes correspondence are not monotonically increasing. This is just one way of generating the data that doesn't have the complete correspondence information. There are also other situations such as missing data and asynchronous data that are not introduced here. We set $\sigma_{\text{data}} = 0.02$ to generate the original datasets A , B and C . 10 trials were run for each algorithm at each scrambling rate r , and the average errors in rotation and translation were plotted as in Fig. (4). It can be seen that the rotation and translation errors of both *Prob1* and *Prob2* remain close to zero despite the scrambling rate r increasing, while the errors of *DK*, *PN* and *Wang* either diverged quickly or blew up in the beginning. This showed the superior performance of the probabilistic approaches when dealing with data streams that had missing correspondence information. In addition, no initial estimates of any kind were needed to calculate X , Y and Z .

Next, we varied the datasets $\sigma_{\text{data}} = \{0.02, 0.04, 0.06, 0.08, 0.1\}$ and $r = 1\%$. As shown in Fig. (5), as the standard deviation σ_{data} increased, both the rotation and translation errors increased. This was consistent with the assumption on the datasets that they should be “highly focused”. Moreover, *Prob2* gave smaller rotation and translation errors compared to *Prob1*, when all three datasets were available. This meant that although *Prob1* can be applied to a broader scope of robotic systems, *Prob2* is preferable if the system allowed the acquisition of complete datasets. This is important because candidates of Y can affect the picking of both X and Z . If no Y is close to its ground truth, it is possible to pick the wrong X and Z even when there are some candidates very close to their ground truths. Besides, the performance of the probabilistic methods in general are comparable to or better than the traditional methods even when the scrambling rate is as low as 1%.

In the real world, data gathered from experiments are usually noisy, and it is interesting to see how the five algorithms perform with noisy and scrambled data. In Fig. (6) and Fig. (7), we fixed the standard deviation for generating the datasets as $\sigma_{\text{data}} = 0.02$. For a homogeneous matrix H , we apply noise with zero mean and standard deviation $\sigma_{\text{noise}} = \{0, 0.002, 0.004, 0.006, 0.008, 0.01\}$ to get a noisy $H_{\text{noise}} = H \exp(\hat{\delta})$, $\hat{\delta} \in \mathcal{N}(\mathbf{0}; \Sigma)$, where the covariance matrix $\Sigma = \sigma_{\text{noise}} \mathbb{I}_6 \in \mathbb{R}^{6 \times 6}$. The scrambling rate in Fig. (6) is $r = 1\%$ and the scrambling rate in Fig. (7) is $r = 10\%$. There are several observations from these two figures.

1. Probabilistic methods deteriorate relatively faster than the traditional methods when the scrambling rate is very low, in this case $r = 1\%$.
2. The probabilistic methods become closer or much better than the traditional methods when the scrambling rate increases from 1% to 10%, despite the effects of noise.
3. For traditional methods, the scrambling rate was the dominant factor on the errors of the solved X , Y and Z when it is large enough. As in Fig. (7), when $r = 10\%$, the performance of the traditional methods only fluctuated within a small range despite the increasing noise.

5 A Hybrid Approach to Handle Noise and Lack of Correspondence

5.1 Algorithm Formulation

As can be observed in Fig. 6 and Fig. 7, the two probabilistic approaches deteriorate quickly as the noises on the sensor measurements increase. This is reasonable because the applied noises becomes more and more comparable to or even on the same scale of the sensor measurements themselves. In practice, one has to keep the noise level really low in order for *Prob1* and *Prob2* to work well. To solve this problem, we seek to introduce multiple clouds of sensor measurement cluttered around different means instead of just one. This is logical because the perturbation of the noise on the means of the data cloud is much smaller compared to its influence on each sample point of the cloud. To avoid the confusion of notation, we use \mathcal{A}_{ijk} to represent the new sets of data measurements. i is the index of the sequence of all the point clouds where $i = 1, \dots, n_s$, $j = I, II, III$ indicates the types of motion constraints when gathering data and k represents the k_{th} individual data in the corresponding point cloud. For consistency, we use the Arabic numbers for the j index instead and one has $j = 1, 2, 3$. The algorithm is described as the following work flow:

$$\begin{aligned}
M_{A11}XM_{B11} &= YM_{C11}Z \\
M_{A12}XM_{B12} &= YM_{C12}Z \\
M_{A13}XM_{B13} &= YM_{C13}Z \\
&\vdots \\
M_{An_s1}XM_{Bn_s1} &= YM_{Cn_s1}Z \\
M_{An_s2}XM_{Bn_s2} &= YM_{Cn_s2}Z \\
M_{An_s3}XM_{Bn_s3} &= YM_{Cn_s3}Z \\
&\downarrow \\
\text{Traditional } AXB = YCZ \text{ Solver}
\end{aligned}$$

where M_{Aij} , M_{Bij} and M_{Cij} are obtained as in Eq. (17a), denoting the mean of the i_{th} cloud under the j_{th} motion constraint. These $3n_s$ equations can be fed into *Wang* or *PN* method to solve for X , Y and Z in a simultaneous manner. The overall formulation of the solver is very simple and straightforward, but it turns out to be very effective in handling both the noise and incorrespondence in the data.

5.2 Numerical Comparison

In this section, we picked *Wang* method as the traditional solver for the hybrid method and compared it with *Prob1* and *Prob2* methods. For a comprehensive study, we compared these three solvers against σ_{data} , σ_{noise} and n_s . We adopted the same way to generate the data cloud as in Section 4.1 except that a number of n_s clouds are needed instead of one for each case of motion constraint. In Eq. (64), B_{i10} denotes

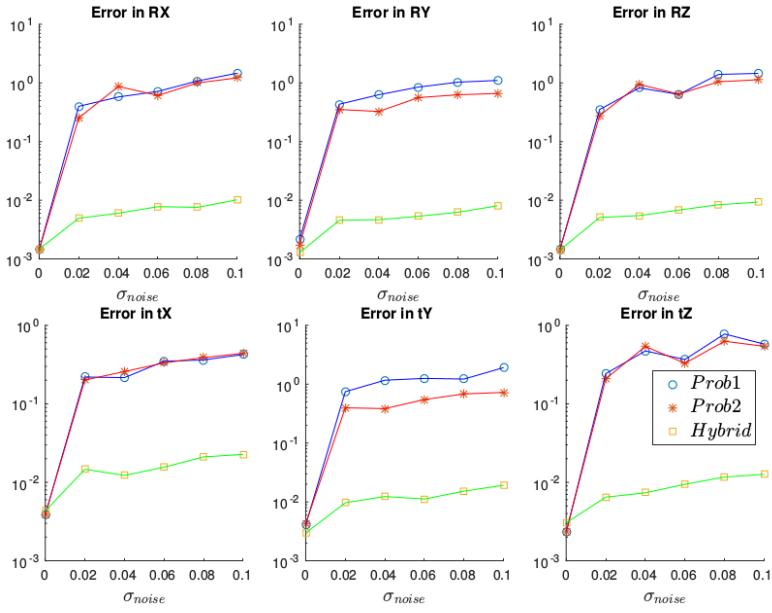


Fig. 8: Rotation/Translation errors in X , Y , Z v.s. σ_{noise} for 20 trials and 50 measurements with $n_s = 4$ and $\sigma_{data} = 0.02$

the initial B for generating the i_{th} data cloud under case *I*. Data scrambling within each point cloud is not needed because none of the probabilistic or hybrid methods use this information for solving the unknowns. To better reflect the noise level, the data noise is set as $\sigma_{noise} = r_{noise} * \sigma_{data}$ where r_{noise} is the percentage of noise on σ_{data} . The same error metrics were used for evaluating the errors of the solved X , Y and Z .

$$B_{i1j} = \exp(\hat{\delta}_j) B_{i10} \quad (64a)$$

$$\hat{\delta}_j \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \quad (64b)$$

The numerical simulation results can be seen as in Fig. (8 - 11). Note that the y axes in Fig. (8) and Fig. (9) are in logarithm due to the large range of the errors. In Fig. 8, the noise level r_{noise} is $r_{noise} = \{0\%, 2\%, \dots, 10\%\}$, the number of sets of data clouds is $n_s = 4$, and the standard deviation used for generating each no-noise cloud is $\sigma_{data} = 0.02$. A total of 50 points were chosen for each data cloud. 20 trials were run for each combination of parameters and the averaged errors were plotted. It can be seen that *Prob1* and *Prob2* methods diverged quickly as the r_{noise} went up while the hybrid method maintained a relatively low level of errors. Fig. (9) showed that the errors of hybrid method decreased very quickly as the number of data set n_s incremented. The hybrid method is competent or even better than *Prob1* and *Prob2* when $\sigma_{data} = 0.1$ and $r_{noise} = 0$. This behavior is a reflection of the property of the traditional $AXB = YCZ$ solver which demands more data with

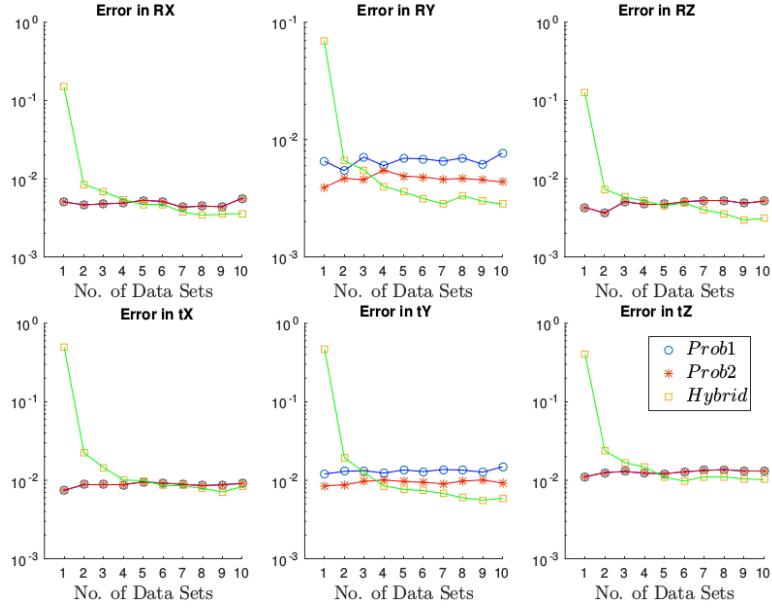


Fig. 9: Rotation/Translation errors in X, Y, Z v.s. n_s for 20 trials and 50 measurements with $\sigma_{data} = 0.1$ and $r_{noise} = 0\%$

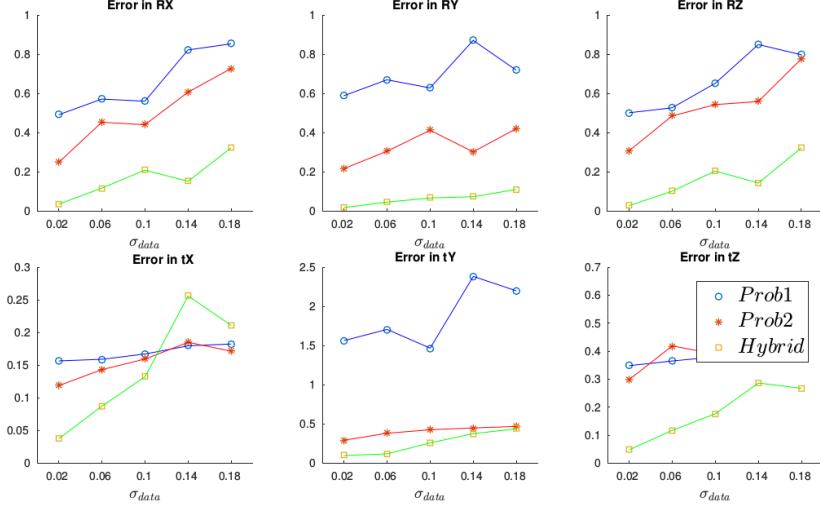


Fig. 10: Rotation/Translation errors in X, Y, Z v.s. standard deviation of the original data cloud for 20 trials and 50 measurements with $r_{noise} = 1\%$ and $n_s = 3$

correspondence. However, the sets of means of A, B, C fed into the solver do not have the exact correspondence, since they are the “average” of the corresponding data cloud. Fig. (10) and Fig. (11) showed the performances of the three solvers

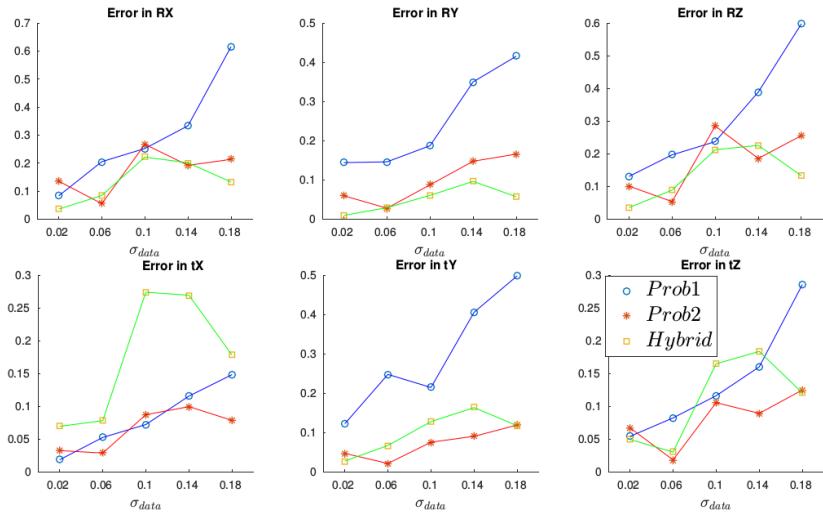


Fig. 11: Rotation/Translation errors in X , Y , Z v.s. standard deviation of the original data cloud for 20 trials and 50 measurements with $r_{noise} = 0\%$ and $n_s = 3$

when σ_{data} increased. It can be seen that all of the solvers share similar sensitiveness towards σ_{data} . The hybrid method performed better when there is noise and *Prob2* gave more accurate results when the data is noise free.

5.3 Discussion

The proposed hybrid method is better at handling noise compared to the probabilistic approaches. This is achieved by providing extra amount of data clouds which cluttered around different means, and this mitigated the affect of noise on the entire data sets. However, the pure probabilistic methods gave better results when the data is noise free and require much less complicated data gathering process. In analogy to combining deterministic and probabilistic approaches in the area of robot path planning, the hybrid method presented here is a tentative approach to apply the same methodology onto the field of robot-sensor calibration. The authors think this is a very empty and open field and worth the further pursue of the community.

6 Iterative Updates for Combined Dataset

The two versions of the algorithm proposed in the Section 3 solve the system of equations by fixing each known transformations only once, which affects the robustness of the procedure and might be sensitive to noise. In order to cope with such problem, a combined dataset with various configurations of the fixed frames should be used. For this reason, we proposed an iterative process to solve for the same equations, where the input is a combined dataset of different configurations of fixed frames.

More specifically, we are interested in the case when frame A or frame C is fixed at different poses, indexed by i and j respectively, and the other two frames are measured for each fixed value of A_i or C_j .

The equations we seek to solve can be picked from Table 1 with the representations and the corresponding covariance, where the full covariance matrices are used instead of the ‘‘Sig-Rot’’ part only. This is because we are trying to recover the rotation and translation parts of the unknown matrices simultaneously.

Explicitly, when A_i is fixed, we pick Representation 1 and the corresponding equation for the covariance matrices as

$$\begin{cases} A_i X M_{B_i} = Y M_{C_i} Z \\ \Sigma_{B_i} = Ad(Z^{-1}) \Sigma_{C_i} Ad^T(Z^{-1}), \end{cases} \quad (65)$$

where A_i is the i^{th} pose when frame A is fixed, M_{B_i} and M_{C_i} are the mean of the sets $\{B_i\}$ and $\{C_i\}$ when A_i is fixed at the i^{th} pose, Σ_{B_i} and Σ_{C_i} are the covariance of $\{B_i\}$ and $\{C_i\}$, $Ad(Z^{-1})$ is the adjoint of Z^{-1} .

As C_j is fixed, on the other hand, we are interested in solving for Representation 6 and its corresponding equation for the covariance matrices as

$$\begin{cases} C_j Z M_{B_j}^{-1} = Y^{-1} M_{A_j} X \\ \Sigma_{B_j^{-1}} = Ad(X^{-1}) \Sigma_{A_j} Ad^T(X^{-1}), \end{cases} \quad (66)$$

where C_j is the j^{th} pose when frame C is fixed, M_{A_j} and M_{B_j} are the mean of the sets $\{A_j\}$ and $\{B_j\}$ when C_j is fixed at the j^{th} pose, Σ_{A_j} and $\Sigma_{B_j^{-1}}$ are the covariance of $\{A_j\}$ and $\{B_j^{-1}\}$, $Ad(X^{-1})$ is the adjoint of X^{-1} .

We solve for this system of equations (65) and (66) at the same time by adding small variations ξ_X, ξ_Y and ξ_Z to X, Y and Z in the space of Lie algebra respectively. Using the first-order Taylor series approximation, the updates can be written as

$$\begin{aligned} X_{k+1} &= X_k(\mathbb{I} + \hat{\xi}_{X_k}) \\ Y_{k+1} &= X_k(\mathbb{I} + \hat{\xi}_{Y_k}) \\ Z_{k+1} &= X_k(\mathbb{I} + \hat{\xi}_{Z_k}), \end{aligned} \quad (67)$$

where k is the number of iterations.

For each iteration, we can expand the equations by the updates of the variables and get a linear system of equations

$$P_k \xi_k = \begin{pmatrix} P_{k1} \\ P_{k2} \\ P_{k3} \\ P_{k4} \end{pmatrix} \begin{pmatrix} \xi_{X_k} \\ \xi_{Y_k} \\ \xi_{Z_k} \end{pmatrix} = \mathbf{b}_k = \begin{pmatrix} \mathbf{b}_{k1} \\ \mathbf{b}_{k2} \\ \mathbf{b}_{k3} \\ \mathbf{b}_{k4} \end{pmatrix}, \quad (68)$$

where $\xi_{X_k}, \xi_{Y_k}, \xi_{Z_k} \in \mathbb{R}^{6 \times 1}$ are variables to be solved in each iteration; and the calculation of $P_{1k} \in \mathbb{R}^{12 \times 18}, P_{2k} \in \mathbb{R}^{36 \times 18}, P_{3k} \in \mathbb{R}^{12 \times 18}$ and $P_{4k} \in \mathbb{R}^{36 \times 18}$, and the corresponding $\mathbf{b}_{1k} \in \mathbb{R}^{12 \times 1}, \mathbf{b}_{2k} \in \mathbb{R}^{36 \times 1}, \mathbf{b}_{3k} \in \mathbb{R}^{12 \times 1}$ and $\mathbf{b}_{4k} \in \mathbb{R}^{36 \times 1}$ are shown in the Appendix C.

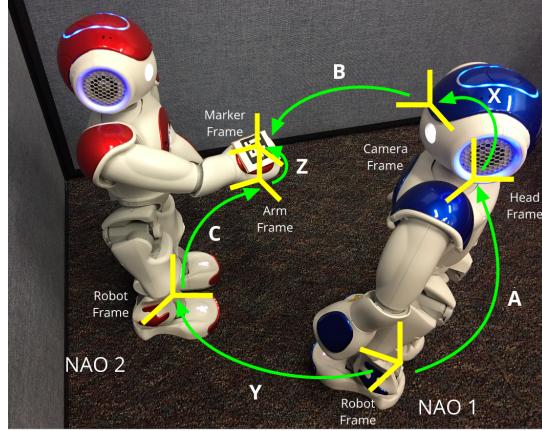


Fig. 12: Real-world experiment settings for calibration using two NAO robots

Once the variable ξ_k is solved by inverting P_k (the pseudo-inverse should be used here) to the right hand side, the matrices to be calibrated can be updated. This process will converge to satisfy all the four equations where in the ideal case $\xi_k = \mathbf{0}$.

7 Experimental Validation

In this section, real-world experiments are performed to further verify the extendability of the probabilistic methods, especially the robustness of the iterative approach, compared with the deterministic ones. The platform is composed of two NAO robots, where one is moving its arm and the other retrieves the transformation of the hand by the camera. The data collected is put into both Wang's [22] approach and the proposed probabilistic approaches for comparisons.

7.1 Experiment Settings

As is shown in Fig. 12, the two NAO robots are set to stand and face to each other, and the description of transformation matrices are summarized in Table 3.

Transformation	Starting Frame	Ending Frame
A	Robot Frame of NAO 1	Head Frame of NAO 1
B	Camera Frame of NAO 1	Marker Frame on NAO 2
C	Robot Frame of NAO 2	Arm Frame of NAO 2
X	Head Frame of NAO 1	Camera Frame of NAO 1
Y	Robot Frame of NAO 1	Robot Frame of NAO 2
Z	Arm Frame of NAO 2	Marker Frame on NAO 2

Table 3: Summary of transformations that are measured and to be calibrated.

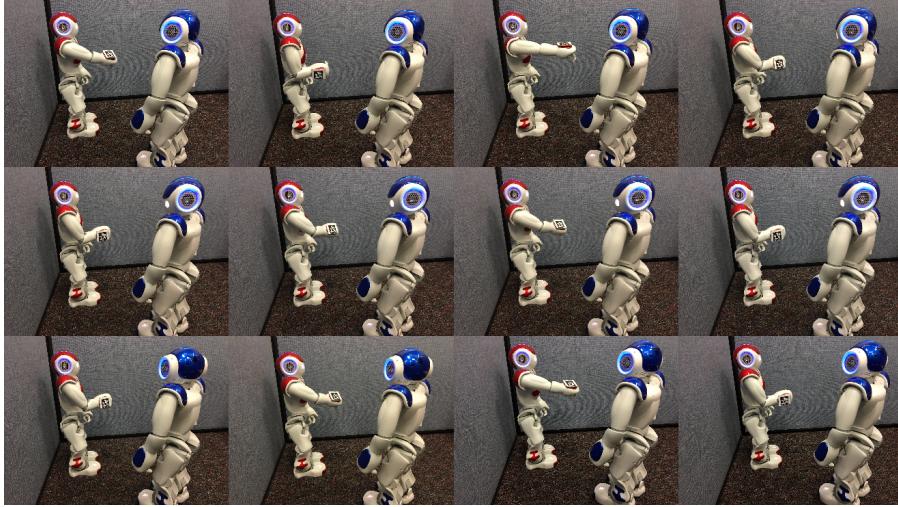


Fig. 13: Moving sequence of the experiment.

The transformations A and C can be measured using the subroutine in the SDK for programming provided by the company, and the transformation information of B is retrieved by attaching a marker on the right hand of NAO 2 and using the *ArUco* library [11].

The experiments are performed by first fixing the head of NAO 1 to a pose, and moving the arm of NAO 2, which we call it a *trial* of data, followed by changing the head frame into several different poses and moving the arm of NAO 2 again. This gives several trials of data with fixed A and varying B and C . Trials of data with fixed C can be obtained by fixing the arm of NAO 2 to several different poses and moving the head of NAO 1. The current experimental data includes: (1) 3 different trials with fixed frame A and 50 sets of changing frames B and C ; (2) 3 different trials with fixed frame C and 50 sets of changing frames A and B . In total, there are 300 pairs of measured data $\{A, B, C\}$ that are with correspondence. But since the camera cannot always detect and measure the transform of the *ArUco* marker, only 298 pairs of the data are valid without missing frames. Fig. 13 shows the sequence of moving the arm of NAO 2 while the head of NAO 1 is fixed in different poses.

The data are then labeled as $\{A_i, B_i, C_i\}$ where $i = 1, 2, 3$ for 3 trials of data with fixed A_i , and $\{A_j, B_j, C_j\}$ where $j = 1, 2, 3$ for 3 trials of data with fixed C_j ,

7.2 Error Metric for Experimental Validation

Since for experiments, we do *not* have ground truths for comparison of calibrated transformations, the metrics for simulated data in Eq. (62) are no longer ideal for evaluating the experimental data. Instead, we can compute the closeness between the left and right hand side of the basic equation $AXB = YCZ$, and evaluate the mean of the accumulated error for all the pairs of data. In other words, the error metric for

Algorithm	Data Combinations
Wang	Combination of the whole dataset $\{A, B, C\}$;
Iterative Refinement	Combined data: $\{A_i, B_i, C_i\}$ and $\{A_j, B_j, C_j\}$, where $i, j = \{1, 2, 3\}$.

Table 4: Summary of the data combination as inputs of each algorithm.

experimental data can be defined as

$$Error = \frac{1}{N} \sum_{m=1}^N \|A_m X_{solved} B_m - Y_{solved} C_m Z_{solved}\|_F, \quad (69)$$

where the pair $\{A_m, B_m, C_m\}$ is the data with correspondence and N is the number of the pairs of data in the whole dataset.

7.3 Data Processing

We compared two different algorithms: *Wang*'s method and the *iterative* refinement. The other two versions of probabilistic method are not tested because: when running the iterative refinement, one can treat the results from *Prob 1* as an initial guess, and since the refinement uses the whole dataset to minimize the error iteratively, the final results should be much better than *Prob 1* (the choice of the initial guess will be discussed through the experimental results); also, for *Prob 2*, it is required to fix B matrix also, but in the experiment it is a nontrivial task to fix the transformation between camera and the marker, so we did not consider this scenario in our experiment. We therefore process the data in the following ways according to the requirements and assumptions of each algorithm:

- (1) for Wang's method, we stack all the data pairs together as $\{A, B, C\}$ where each of the matrices contains the whole sets of the transformations;
- (2) for the iterative refinement, we separate trials of data for fixed A and fixed C , and treat as two inputs to the algorithm.

Table 4 summarizes the combination of data that are put into those algorithms according to their requirements and assumptions.

To verify that the probabilistic methods can deal with the data without correspondence, we also scrambled the order of the data and compare the errors of the methods. Note that the forms and sizes of inputs are the same, but the orders are not; and the error is still calculated using the data with correct correspondence. Also, for reference, the algorithms are tested using the simulated data, where 5 trials for fixed A and C respectively and each trial has 100 pairs of data.

7.4 Results and Analysis

The results for algorithm comparisons using simulated and experimental data with respect to the scramble rate are shown in Fig. 14 and 15 respectively, where the three plots show the error as the scrambling rate increases with different initial guesses.

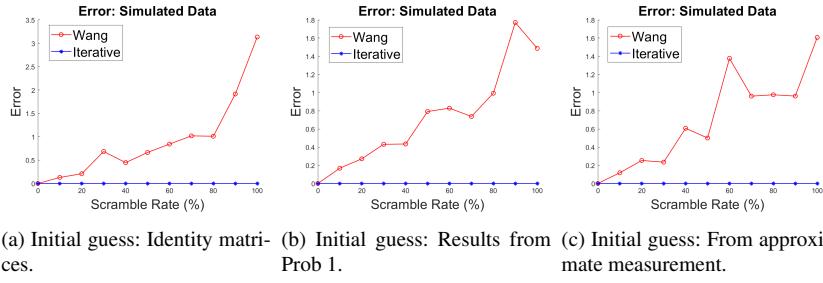


Fig. 14: Error v.s. scrambling rate on simulated data with different initial guesses.

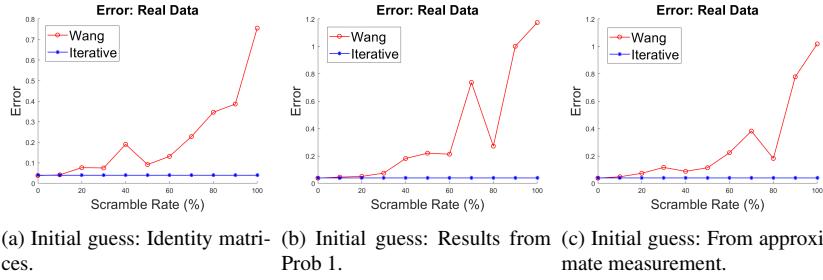


Fig. 15: Error v.s. scrambling rate on simulated data with different initial guesses.

As can be seen from the results, the errors using the probabilistic approach are invariant to the correspondence of the data, while using the traditional method the error increases significantly if the data is scrambled. In this aspect, once the collected data has some missing parts or the order is not correct by accident, the iterative refinement can still get a solution that is close to correct. Further, if the data is *rich* enough, the covariance of the dataset will become even more robust to recover the unknown transformations. Here rich means that the number of data collected is large, and the degrees of freedom for the moving part of both robots are high, which make all the measured transformations vary on a larger space so that the distribution can be approximated closer to a Gaussian.

The initial guess also plays a role on the efficiency of the two algorithms. As shown in Fig. 16, the number of iterations differs with the changes of the initial guesses. The results from *Prob 1* can be a starting point to the iterative refinement, which gives faster convergence than an arbitrary guess, e.g. identity matrix. Further, if we can manually approximate a measurement of the transformations, for instance, from the kinematic data of the robot, the algorithms can perform even more efficiently.

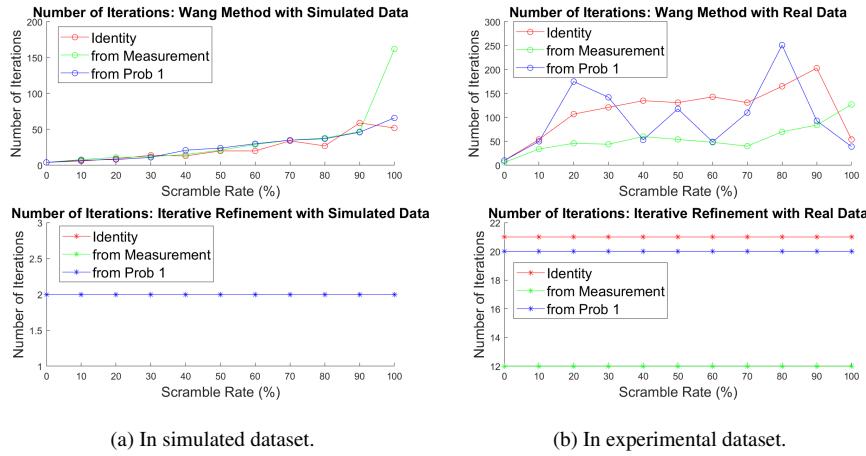


Fig. 16: Number of iterations v.s. scrambling rate on simulated and experimental data with different initial guesses.

8 Conclusion

Motivated by problems that arise in multi-robot systems, in this article we proposed two probabilistic approaches to solve the $AXB = YCZ$ calibration problem for the case where partial or all correspondence information between the datasets was lost. Numerical simulations were performed to show the superior performance of the probabilistic approaches over the traditional $AXB = YCZ$ solvers that demand exact correspondence among the datasets. In addition, the probabilistic approaches did not require initial estimates which made the calibration process easier. We compared the performance between the two probabilistic approaches and showed that given complete datasets, *Prob2* gave better R_Y and t_Y . However, *Prob1* required fewer datasets and had wider applications.

Though the probabilistic methods can handle scrambled data very well, they are very sensitive to the noise. As a tentative action to solve this problem, we also proposed a hybrid approach which combines traditional $AXB = YCZ$ solvers with probabilistic methodology. The probabilistic methods deteriorated quickly as the noise level went up, because the desired data clouds are highly concentrated and even small noise can perturb the data a lot. However, by introducing multiple data clouds cluttered around different centers, the influence of noise on the data is mitigated. It is shown that the hybrid method converged quickly as the number of data clouds incremented, and its performance beat the pure probabilistic methods when the noise level is high. However, when the data is noise free, the probabilistic approaches are still better than the hybrid method, and it also requires much fewer sets of data. A new iterative refinement to the current probabilistic methods is proposed to mitigate the effect of the noise. It is shown, with simulated dataset as well as the physical experiments using a system with two humanoid robots, that it not only preserves the nice properties of the pure probabilistic method, but also can deal with the real-world

data which contains noise. The experiments also show that the refinement reliably converges to the correct answer with different initial guesses of the transformations to be calibrated, and the efficiency improves with a good choice.

A Mean and Covariance of the Distribution of the Inverse Transformation

Proof Let $f'(H) = f(H^{-1})$ whose mean and covariance are M' and Σ' satisfying the following equations

$$\int_{SE(3)} \log(M'^{-1}H)f'(H)dH = \mathbb{O} \quad (70)$$

and

$$\Sigma' = \int_{SE(3)} \log^\vee(M'^{-1}H)[\log^\vee(M'^{-1}H)]^T f'(H)dH. \quad (71)$$

After a simple substitution, Eq.(70) becomes

$$\int_{SE(3)} \log(M'^{-1}H)f(H^{-1})dH = \mathbb{O}. \quad (72)$$

Next, set $K = H^{-1}$ and use the invariance of integration under inversion, Eq. (72) becomes:

$$\int_{SE(3)} \log(M'^{-1}K^{-1})f(K)dK = \mathbb{O}. \quad (73)$$

Premultiply M' and postmultiply M'^{-1} on both sides of the equation to get

$$\int_{SE(3)} \log(K^{-1}M'^{-1})f(K)dK = \mathbb{O} \quad (74)$$

by using the property

$$H_1(\log H_2)H_1^{-1} = \log(H_1H_2H_1^{-1}). \quad (75)$$

Eq.(74) can be further written as

$$\int_{SE(3)} \log(M'K)f(K)dK = \mathbb{O}, \quad (76)$$

given $\log(H^{-1}) = -\log H$. This shows that $M' = M^{-1}$.

By definition, covariance Σ' will be

$$\Sigma' = \int_{SE(3)} \log^\vee(M'^{-1}H)[\log^\vee(M'^{-1}H)]^T f(H^{-1})dH \quad (77)$$

which becomes

$$\Sigma' = \int_{SE(3)} \log^\vee(MK^{-1})[\log^\vee(MK^{-1})]^T f(K)dK \quad (78)$$

under a change of variables and substitution of $M' = M^{-1}$. Knowing that

$$\begin{aligned} Ad(H_1)\log^\vee(H_2) &= \log^\vee(H_1H_2H_1^{-1}) \\ \log^\vee(K^{-1}M) &= -\log^\vee(M^{-1}K) \end{aligned} \quad (79)$$

$$\begin{aligned} Ad(M^{-1})\Sigma'Ad^T(M^{-1}) &= \\ \int_{SE(3)} \log^\vee(K^{-1}M)[\log^\vee(K^{-1}M)]^T f(K)dK & \end{aligned} \quad (80)$$

so that

$$Ad(M^{-1})\Sigma'Ad^T(M^{-1}) = \Sigma \quad (81)$$

which, after inversion of the Ad matrices, completes the proof.

B Second-order Approximation of the Convolution of three PDFs

From the associativity of the group operation, we have

$$((f_1 * f_2) * f_3)(g) = (f_1 * (f_2 * f_3))(g) = (f_1 * f_2 * f_3)(g) \quad (82)$$

and

$$\Sigma_{(1*2)*3} = \Sigma_{1*(2*3)} = \Sigma_{1*2*3}. \quad (83)$$

Let us now evaluate the three fold convolution using both of the above equalities and the properties of $F(\cdot, \cdot)$ in Eq. (24):

$$\Sigma_{(1*2)*3} = Ad(M_3^{-1})\Sigma_{1*2}Ad^T(M_3^{-1}) + \Sigma_3 + F\left(Ad(M_3^{-1})\Sigma_{1*2}Ad^T(M_3^{-1}); \Sigma_3\right), \quad (84)$$

where $\Sigma_{1*2} = Ad(M_2^{-1})\Sigma_1Ad^T(M_2^{-1}) + \Sigma_2 + F\left(Ad(M_2^{-1})\Sigma_1Ad^T(M_2^{-1}); \Sigma_2\right)$.

If $\Sigma_1 = \Sigma_2 = \emptyset$,

$$\Sigma_{1*2*3} = \Sigma_3; \quad (85)$$

and if $\Sigma_2 = \Sigma_3 = \emptyset$,

$$\Sigma_{1*2*3} = Ad_{M_3}^{-1}Ad_{M_2}^{-1}\Sigma_1Ad_{M_2}^{-T}Ad_{M_3}^{-T}. \quad (86)$$

Hence both of the covariances do not depend on the second-order term.

C Solutions for the Iterative Refinement

The equations we need to solve are

$$A_i X M_{B_i} = Y M_{C_i} Z \quad (87a)$$

$$\Sigma_{B_i} = Ad(Z^{-1})\Sigma_{C_i}Ad^T(Z^{-1}) \quad (87b)$$

$$C_j Z M_{B_j}^{-1} = Y^{-1} M_{A_j} X \quad (87c)$$

$$\Sigma_{B_j^{-1}} = Ad(X^{-1})\Sigma_{A_j}Ad^T(X^{-1}) \quad (87d)$$

for i, j are the number of trials of dataset for fixing A_i and C_j respectively.

A small perturbation of the initial X, Y and Z can be expressed as

$$\begin{aligned} X_{k+1} &= X_k(\mathbb{I} + \hat{\xi}_{X_k}) \\ Y_{k+1} &= Y_k(\mathbb{I} + \hat{\xi}_{Y_k}) \\ Z_{k+1} &= Z_k(\mathbb{I} + \hat{\xi}_{Z_k}) \end{aligned} \quad (88)$$

where, $\xi_k = [\omega_k, \mathbf{v}_k]^T$.

The following gives a detailed derivation of the explicit forms of each part of the matrices P_k and \mathbf{b}_k

C.1 Construction of P_{1k} and \mathbf{b}_{1k}

Substituting Eq. (88) back into Eq. (87a) and eliminating quadratic terms gives

$$A_i X_k \hat{\xi}_{X_k} M_{B_i} - Y_k \hat{\xi}_{Y_k} M_{C_i} Z_k - Y_k M_{C_i} Z_k \hat{\xi}_{Z_k} = -A_i X_k M_{B_i} + Y_k M_{C_i} Z_k. \quad (89)$$

Separating the rotation and translation parts gives,

(1) Rotation part:

$$R_{A_i} R_{X_k} \hat{\omega}_X R_{M_{B_i}} - R_{Y_k} R_{M_{C_i}} R_{Z_k} \hat{\omega}_Z - R_{Y_k} \hat{\omega}_Y R_{M_{C_i}} R_{Z_k} = -R_{A_i} R_{X_k} R_{M_{B_i}} + R_{Y_k} R_{M_{C_i}} R_{Z_k} \quad (90)$$

Using the fact that $\mathbf{a} \times \mathbf{b} = \hat{a}b - \hat{b}a$ and treating each column separately gives

$$\begin{aligned} & -R_{A_i} R_{X_k} (\widehat{R}_{M_{B_i}})_m \omega_{X_k} + R_{Y_k} R_{M_{C_i}} R_{Z_k} \widehat{\mathbf{e}}_m \omega_{Z_k} + R_{Y_k} (\widehat{R}_{M_{C_i}} \widehat{R}_{Z_k})_m \omega_{Y_k} \\ & = \text{Rot}(-A_i X_k M_{B_i} + Y_k M_{C_i} Z_k)_m \end{aligned} \quad (91)$$

where $m = 1, 2, 3$ is the index of columns of the rotation matrices, and $\text{Rot}(\cdot)$ is the rotation part of a rigid body transformation matrix.

(2) Translation part:

$$\begin{aligned} & -R_{A_i} R_{X_k} \widehat{\mathbf{t}}_{M_{B_i}} \omega_{X_k} + R_{Y_k} (\widehat{R}_{M_{C_i}} \widehat{\mathbf{t}}_{Z_k} + \mathbf{t}_{M_{C_i}}) \omega_{Y_k} + R_{A_i} R_{X_k} \mathbf{v}_{X_k} \\ & -R_{Y_k} \mathbf{v}_{Y_k} - R_{Y_k} R_{M_{C_i}} R_{Z_k} \mathbf{v}_{Z_k} \\ & = \text{Trans}(-A_i X_k M_{B_i} + Y_k M_{C_i} Z_k) \end{aligned} \quad (92)$$

where $\text{Trans}(\cdot)$ is the translation part of a rigid body transformation matrix.

Combining the rotation and translation parts and stacking ω_k and \mathbf{v}_k together to form a linear system of equations, we obtain the corresponding matrices P_{1k} and \mathbf{b}_{1k} as

$$P_{1k} = \begin{bmatrix} P_{1m1} & 0_{3 \times 3} & P_{1m3} & 0_{3 \times 3} & P_{1m5} & 0_{3 \times 3} \\ -R_{A_i} R_{X_k} \widehat{\mathbf{t}}_{M_{B_i}} & R_{A_i} R_{X_k} & R_{Y_k} (\widehat{R}_{M_{C_i}} \widehat{\mathbf{t}}_{Z_k} + \mathbf{t}_{M_{C_i}}) & -R_{Y_k} & 0_{3 \times 1} & -R_{Y_k} R_{M_{C_i}} R_{Z_k} \end{bmatrix} \quad (93)$$

where,

$$P_{1m1} = -R_{A_i} R_{X_k} (\widehat{R}_{M_{B_i}})_m, P_{1m3} = R_{Y_k} (\widehat{R}_{M_{C_i}} \widehat{R}_{Z_k})_m, P_{1m5} = R_{Y_k} R_{M_{C_i}} R_{Z_k} \widehat{\mathbf{e}}_m.$$

And

$$\mathbf{b}_{1k} = \begin{bmatrix} \text{Rot}(-A_i X_k M_{B_i} + Y_k M_{C_i} Z_k)_m \\ \text{Trans}(-A_i X_k M_{B_i} + Y_k M_{C_i} Z_k) \end{bmatrix}. \quad (94)$$

C.2 Construction of P_{2k} and \mathbf{b}_{2k}

Using the similar idea for Eq. (87b), we have

$$\Sigma_{B_i} = \text{Ad}(Z_{k+1}^{-1}) \Sigma_{C_i} \text{Ad}^T(Z_{k+1}^{-1}) \quad (95)$$

By using the identity of adjoint in Eq. (9) and (11), we get

$$\Sigma_{B_i} + \text{ad}(\hat{\xi}_Z) \Sigma_{B_i} + \Sigma_{B_i} \text{ad}^T(\hat{\xi}_Z) = \text{Ad}^{-1}(Z_k) \Sigma_{C_i} \text{Ad}^{-T}(Z_k) \quad (96)$$

Splitting the covariance and adjoint matrices into 4 blocks and applying matrix multiplications give,

$$\begin{bmatrix} \hat{\omega}_Z & \mathbb{O} \\ \hat{v}_Z & \hat{\omega}_Z \end{bmatrix} \begin{bmatrix} \Sigma_{B_i}^1 & \Sigma_{B_i}^2 \\ \Sigma_{B_i}^3 & \Sigma_{B_i}^4 \end{bmatrix} + \begin{bmatrix} \Sigma_{B_i}^1 & \Sigma_{B_i}^2 \\ \Sigma_{B_i}^3 & \Sigma_{B_i}^4 \end{bmatrix} \begin{bmatrix} \hat{\omega}_Z^T & \hat{v}_Z^T \\ \mathbb{O} & \hat{\omega}_Z^T \end{bmatrix} = \text{Ad}^{-1}(Z_k) \Sigma_{C_i} \text{Ad}^{-T}(Z_k) - \Sigma_{B_i} \quad (97a)$$

Applying block matrix multiplications and for each block, we extract ω_Z and \mathbf{v}_Z by treating the columns separately as

$$\begin{aligned} & \left[-(\widehat{\Sigma}_{B_i}^1)_m + \Sigma_{B_i}^1 \widehat{\mathbf{e}}_m \right] \omega_Z = (\text{RHS})_m^1 \\ & \left[-(\widehat{\Sigma}_{B_i}^2)_m + \Sigma_{B_i}^2 \widehat{\mathbf{e}}_m \right] \omega_Z + \Sigma_{B_i}^1 \widehat{\mathbf{e}}_m \mathbf{v}_Z = (\text{RHS})_m^2 \\ & \left[-(\widehat{\Sigma}_{B_i}^3)_m + \Sigma_{B_i}^3 \widehat{\mathbf{e}}_m \right] \omega_Z - (\widehat{\Sigma}_{B_i}^1)_m \mathbf{v}_Z = (\text{RHS})_m^3 \\ & \left[-(\widehat{\Sigma}_{B_i}^4)_m + \Sigma_{B_i}^4 \widehat{\mathbf{e}}_m \right] \omega_Z + \left[-(\widehat{\Sigma}_{B_i}^2)_m + \Sigma_{B_i}^3 \widehat{\mathbf{e}}_m \right] \mathbf{v}_Z = (\text{RHS})_m^4 \end{aligned} \quad (98)$$

where $\text{RHS} = Ad^{-1}(Z_k) \Sigma_{C_i} Ad^{-T}(Z_k) - \Sigma_{B_i}$, $m = 1, 2, 3$ denotes the index of the column on each matrix or block of matrix, and the right superscripts denotes the index of the block in the matrix.

In this case, the P_{2k} and \mathbf{b}_{2k} matrices are

$$P_{2k} = \begin{bmatrix} 0_{3 \times 12} - (\widehat{\Sigma_{B_i}^1})_m + \Sigma_{B_i}^1 \widehat{\mathbf{e}}_m & 0_{3 \times 3} \\ 0_{3 \times 12} - (\widehat{\Sigma_{B_i}^2})_m + \Sigma_{B_i}^2 \widehat{\mathbf{e}}_m & \Sigma_{B_i}^1 \widehat{\mathbf{e}}_m \\ 0_{3 \times 12} - (\widehat{\Sigma_{B_i}^3})_m + \Sigma_{B_i}^3 \widehat{\mathbf{e}}_m & -(\widehat{\Sigma_{B_i}^1})_m \\ 0_{3 \times 12} - (\widehat{\Sigma_{B_i}^4})_m + \Sigma_{B_i}^4 \widehat{\mathbf{e}}_m & -(\widehat{\Sigma_{B_i}^2})_m + \Sigma_{B_i}^3 \widehat{\mathbf{e}}_m \end{bmatrix}, \quad (99)$$

and

$$\mathbf{b}_{2k} = \begin{bmatrix} (Ad^{-1}(Z_k) \Sigma_{C_i} Ad^{-T}(Z_k) - \Sigma_{B_i})_m^1 \\ (Ad^{-1}(Z_k) \Sigma_{C_i} Ad^{-T}(Z_k) - \Sigma_{B_i})_m^2 \\ (Ad^{-1}(Z_k) \Sigma_{C_i} Ad^{-T}(Z_k) - \Sigma_{B_i})_m^3 \\ (Ad^{-1}(Z_k) \Sigma_{C_i} Ad^{-T}(Z_k) - \Sigma_{B_i})_m^4 \end{bmatrix}. \quad (100)$$

C.3 Construction of P_{3k} and \mathbf{b}_{3k}

The matrix P_{3k} and \mathbf{b}_{3k} can be obtained explicitly, using the same ideas above, as

$$P_{3k} = \begin{bmatrix} R_{Y_k^{-1}} R_{M_{A_j}} R_{X_k} \widehat{\mathbf{e}}_m & 0_{3 \times 3} & -(R_{Y_k^{-1}} \widehat{R_{M_{A_j}}} R_{X_k})_m & 0_{3 \times 3} & -R_{C_j} R_{Z_k} (\widehat{R_{M_{B_j}^{-1}}})_m & 0_{3 \times 3} \\ 0_{3 \times 3} & P_{342} & P_{343} & \mathbb{I}_{3 \times 3} & P_{345} & P_{346} \end{bmatrix} \quad (101)$$

where,

$$P_{342} = -R_{Y_k^{-1}} R_{M_{A_j}} R_{X_k}, P_{343} = -(R_{Y_k^{-1}} R_{M_{A_j}} t_{X_k} \widehat{+} \widehat{R_{Y_k^{-1}} t_{M_{A_j}}} + t_{Y_k^{-1}}), P_{345} = -R_{C_j} R_{Z_k} \widehat{t_{M_{B_j}^{-1}}}, \\ P_{346} = R_{C_j} R_{Z_k}.$$

And

$$\mathbf{b}_{3k} = \begin{bmatrix} Rot(-C_j Z_k M_{B_j}^{-1} + Y_k^{-1} M_{A_j} X_k)_m \\ Trans(-C_j Z_k M_{B_j}^{-1} + Y_k^{-1} M_{A_j} X_k) \end{bmatrix} \quad (102)$$

C.4 Construction of P_{4k} and \mathbf{b}_{4k}

For Eq. (87d), we have

$$\Sigma_{B_j^{-1}} = Ad(X_{k+1}^{-1}) \Sigma_{A_j} Ad^T(X_{k+1}^{-1}) \quad (103)$$

Note that when actually calculating the inverse of Σ_B , it is better to use $\Sigma_{B_j^{-1}} = Ad(B) \Sigma_B Ad^T(B)$.

Using the same methodology with previous derivations, the explicit forms of P_{4k} and \mathbf{b}_{4k} can be written as

$$P_{4k} = \begin{bmatrix} -((\widehat{\Sigma_{B_j^{-1}}^1})_m + (\Sigma_{B_j^{-1}}^1)^1 \widehat{\mathbf{e}}_m) & 0_{3 \times 3} & 0_{3 \times 12} \\ -((\widehat{\Sigma_{B_j^{-1}}^2})_m + (\Sigma_{B_j^{-1}}^2)^2 \widehat{\mathbf{e}}_m) & (\Sigma_{B_j^{-1}}^1)^1 \widehat{\mathbf{e}}_m & 0_{3 \times 12} \\ -((\widehat{\Sigma_{B_j^{-1}}^3})_m + (\Sigma_{B_j^{-1}}^3)^3 \widehat{\mathbf{e}}_m) & -((\widehat{\Sigma_{B_j^{-1}}^1})_m) & 0_{3 \times 12} \\ -((\widehat{\Sigma_{B_j^{-1}}^4})_m + (\Sigma_{B_j^{-1}}^4)^4 \widehat{\mathbf{e}}_m) & -((\widehat{\Sigma_{B_j^{-1}}^2})_m + (\Sigma_{B_j^{-1}}^2)^3 \widehat{\mathbf{e}}_m & 0_{3 \times 12} \end{bmatrix}, \quad (104)$$

and

$$\mathbf{b}_{4k} = \begin{bmatrix} (Ad^{-1}(X_k) \Sigma_{A_j} Ad^{-T}(X_k) - \Sigma_{B_j^{-1}})_m^1 \\ (Ad^{-1}(X_k) \Sigma_{A_j} Ad^{-T}(X_k) - \Sigma_{B_j^{-1}})_m^2 \\ (Ad^{-1}(X_k) \Sigma_{A_j} Ad^{-T}(X_k) - \Sigma_{B_j^{-1}})_m^3 \\ (Ad^{-1}(X_k) \Sigma_{A_j} Ad^{-T}(X_k) - \Sigma_{B_j^{-1}})_m^4 \end{bmatrix}. \quad (105)$$

In the end, the P_k and \mathbf{b}_k matrices can be obtained by concatenating the four parts calculated above.

Acknowledgements This work was performed under Office of Naval Research Award N00014-17-1-2142 and National Science Foundation grant IIS-1619050.

References

1. F. Alamifar, D. Jiang, H.K. Zhang, A. Cheng, X. Guo, R. Khurana, I. Iordachita, and E. M. Boctor. Co-robotic ultrasound tomography: dual arm setup and error analysis. In *SPIE Medical Imaging*, pages 94190N–94190N. International Society for Optics and Photonics, 2015.
2. M. K. Ackerman and G. S. Chirikjian. A probabilistic solution to the $AX = XB$ problem: Sensor calibration without correspondence. In *Geometric Science of Information*, pages 693–701. Springer, 2013.
3. M. K. Ackerman, A. Cheng, E. M. Boctor, and G.S. Chirikjian. Online ultrasound sensor calibration using gradient descent on the euclidean group. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4900–4905. IEEE, 2014.
4. M. K. Ackerman, A. Cheng, and G.S. Chirikjian. An information-theoretic approach to the correspondence-free $AX = XB$ sensor calibration problem. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4893–4899. IEEE, 2014.
5. G. S. Chirikjian. *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*, volume 2. Springer Science & Business Media, 2011.
6. G.S. Chirikjian and A.B. Kyatkin. *Harmonic Analysis for Engineers and Applied Scientists*. Dover, 2016.
7. K. Daniilidis. Hand-eye calibration using dual quaternions. *The International Journal of Robotics Research*, 18(3):286–298, 1999.
8. J. D. Davis, Y. Sevimli, M. K. Ackerman, and G. S. Chirikjian. A robot capable of autonomous robotic team repair: The Hex-DMR II system. In *Advances in Reconfigurable Mechanisms and Robots II*, pages 619–631. Springer, 2016.
9. F. Ernst, L. Richter, L. Matthäus, V. Martens, R. Bruder, A. Schlaefer, and A. Schweikard. Non-orthogonal tool/flange and robot/world calibration. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 8(4):407–420, 2012.
10. I. Fassi and G. Legnani. Hand to sensor calibration: A geometrical interpretation of the matrix equation $AX = XB$. *Journal of Robotic Systems*, 22(9):497–506, 2005.
11. S. Garrido-Jurado, R. Mu noz Salinas, F.J. Madrid-Cuevas, and M.J. Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47 (6):2280 – 2292, 2014. ISSN 0031-3203. doi: <http://dx.doi.org/10.1016/j.patcog.2014.01.005>. URL <http://www.sciencedirect.com/science/article/pii/S0031320314000235>.
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*, 2016.
13. R.L. Hirsh, G.N. DeSouza, and A.C. Kak. An iterative approach to the hand-eye and base-world calibration problem. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2171–2176. IEEE, 2001.
14. R. Horaud and F. Dornaika. Hand-eye calibration. *The International Journal of Robotics Research*, 14(3):195–210, 1995.
15. A. Li, L. Wang, and D. Wu. Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product. *The International Journal of Physical Sciences*, 5(10):1530–1536, 2010.
16. 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 (RA-L)*, 2016.
17. Qianli Ma, Zachariah Goh, and Gregory S Chirikjian. Probabilistic approaches to the $axb = ycz$ calibration problem in multi-robot systems. 2016.
18. F.C. Park and B.J. Martin. Robot sensor calibration: solving $AX = XB$ on the euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5), 1994.

19. M. Shah. Solving the robot-world/hand-eye calibration problem using the Kronecker product. *Journal of Mechanisms and Robotics*, 5(3):031007, 2013.
20. Y.C. 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*, 5(1): 16–29, 1989.
21. 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*, 5(3):345–358, 1989.
22. J. Wang, L. Wu, M. Q-H Meng, and H. Ren. Towards simultaneous coordinate calibrations for cooperative multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 410–415. IEEE, 2014.
23. Y. Wang and G.S. Chirikjian. Nonparametric second-order theory of error propagation on motion groups. *The International Journal of Robotics Research*, 27(11-12):1258–1273, 2008.
24. L. Wu, J. Wang, L. Qi, K. Wu, H. Ren, and M. Q. H. Meng. Simultaneous hand-eye, tool-flange, and robot-robot calibration for comanipulation by solving the $AXB = YCZ$ problem. *IEEE Transactions on Robotics*, 32(2):413–428, April 2016. ISSN 1552-3098. doi: 10.1109/TRO.2016.2530079.
25. S.J. Yan, S.K. Ong, and A.Y.C. Nee. Registration of a hybrid robot using the degradation-Kronecker method and a purely nonlinear method. *Robotica*, pages 1–12, 12 2015.
26. Z. Zhao. Hand-eye calibration using convex optimization. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2947–2952. IEEE, 2011.
27. 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*, 10(4):549–554, 1994.