Probabilistic Calibration and Catheter Tracking with Robotic Systems

by

Qianli Ma

A dissertation submitted to The Johns Hopkins University in conformity with the requirements for the degree of Doctor of Philosophy.

Baltimore, Maryland

July, 2017

Abstract

A significant boost in robotics technology has been observed in recent years and more and more tasks are being automated by robots such as robotic surgery, autonomous driving, package delivery, etc. Not only has the precision of robots been improved, but the number of robots involved in a specific task has also grown in many scenarios. An important part in a robotic automated task involves the relative pose estimation among objects, and this often boils down to calibration and tracking. The dissertation begins with a robotic catheter tracking system and then focuses on calibration of robotic systems.

The presentation first introduces a novel robotic catheter tracking system which uses an embedded active piezoelectric element at the tip of the catheter. Catheter intervention procedure is performed exclusively with X-ray, while ultrasound comes as an alternative modality which is radiation free. However, the catheter tip is usually very small and hard to be differentiated from human tissue in an ultrasound image. Moreover, an ultrasound photographer needs to hold the ultrasound probe during the procedure which can easily last for over an hour. The proposed system can tackle these issues using a robot arm and the active echo signal, and is, to the best knowledge of the author, the first robotic catheter tracking system using ultrasound. It is demonstrated in both the simulation and experiment that a robotic arm holding the ultrasound probe can track the catheter tip without image input.

To better assist the tracking process, other procedures can be automated such as catheter insertion and phantom localization, etc. All these require introducing an extra robot and a precise calibration between robots and targets of interest. Out of many calibration approaches, the most classical one is called the hand-eye calibration problem formulated as $AX = XB$ which takes in data from sensors in different locations to solve for an unknown rigid-body transformation. A generalization of this problem is the $AX = YB$ robot-world and hand-eye calibration, where two unknowns need to be recovered simultaneously. The above two approaches mainly deal with the calibration of a single robot system. For multi-robot systems, a problem cast as the $AXB = YCZ$ formulation arises where three unknowns need to be solved given three sensor data streams. The second portion of the presentation investigates in the probabilistic approaches toward all three problems above. Different methods based on the probabilistic theory on Lie group are developed to show their superior performance over non-probabilistic equivalents when there is partial knowledge of the correspondence among sensor data.

ABSTRACT

Primary Reader and Advisor: Professor Gregory S. Chirikjian

Secondary Reader:

Professor Emad M. Boctor

Professor Noah J. Cowan

Acknowledgments

The past five years has been really valuable and unforgettable for me, and I can't express how much I have grown during this long and hard-working journey. I would like to sincerely thank my adviser Dr. Gregory Chirikjian for his guidance and patience during my studies. I am very grateful for the time and advice he has given me and he has truly set a role model in various aspects of life and research.

I would also like to thank the other committee members, Dr. Emad Boctor and Dr. Noah Cowan for their commitment regarding this thesis. I won't be where I am today without the support of my friends and colleagues at Johns Hopkins University. I would specifically like to thank Joshua Davis, Alexis Cheng, Haiyuan Li, Younsu Kim, Haichong Zhang, Kristopher Reynolds, Yuttana Itsarachalyot, Jin Seob Kim, Gowtham Garimella, Chi Li, Xiang Xiang, Fereshteh Aalamifar, Sipu Ruan, Xiao Li, Ying Xu, Yan Yan, Kendal Ackerman, Taosha Fan, Hans Ma and Yixin Gao for their support and knowledge.

At the end, I would also like to thank my parents for their encouragement,

support and unconditional love for the past five years. Their perseverance and hard work have set up a good model in my early childhood and I am grateful to share both the happiness and sad times with them all along the way.

Dedication

This thesis is dedicated to my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This dissertation presents two main topics. The work begins by presenting a robotic ultrasound system that can track the tip of a catheter. Both the hardware and software frameworks are described in detail and it is demonstrated in both simulation and experiments that the system is able to track the tip of a catheter in a low-cost and computationally efficient manner. In order to fully automate the catheter tracking and insertion processes for real clinical application, calibration techniques become critical for the entire system to operate both consistently and precisely. This leads to the discussion of various calibration techniques for both single robot and multi-robot systems. The calibration solvers presented can be applied to various kinds of robotic systems, and the fundamental mathematical concepts are given in Chapter 2.

1.1 Robotic Assisted Catheter Tracking

Robotic technology has played an important role in surgical applications, and a number of successful platforms have been launched such as da Vinci, CyberKnife, and the Raven. Despite the clinical use of various robotic surgical system, the application of robotic technology in cardiac catheterization is still under development. An important part of this is to automate tracking procedure for the tip of the catheter. One popular approach is to use ultrasound information for catheter tip localization due to the fact that it is radiation free. Most of the proposed systems employ either a 2D or 3D ultrasound probe and use the corresponding 2D or 3D ultrasound images to help locate the position of the catheter tip. However, none of them really use ultrasound to perform tracking tasks on the catheter tip. Chapter 3 presents a robotically assisted catheter tracking system that can track the catheter tip utilizing an embedded active piezoelectric element. The hardware system is consisted of a 6 degrees of freedom robot arm, an ultrasound system and a phantom, and the software framework is built on top of the Robot Operating System. A position controller is designed to track the 2D position of the catheter tip within the phantom based on the active echo signal. It is demonstrated in both simulation and experiment that the system can track the catheter tip in a multi-vein phantom, and a 3D structure of the vein can be built for potential visualization usage.

1.2 Robotic System Calibration

The term *sensor calibration* can mean different things in different contexts, both within the field of robotics, and in other areas of study. In the most general terms, a sensor is a device that produces measurements that provide information about the state of the world. For example, an inertial measurement unit (IMU) consisting of an accelerometer and gyroscope can provide information about the motion of a rigid body. Cameras provide visual information about a scene in the form of pixel intensities.

Note also that the word “calibration” has been used in robotics previously in the context of kinematic calibration [1]. This topic is concerned with obtaining manipulator kinematic parameters that are not known a priori by having the manipulator move around and using the information from encoders and from an external sensor that tracks markers (typically on the end-effector) to obtain the link twists and offsets. A calibration problem that is somewhat analogous to this in spirit is the dynamic calibration problem in which the inertial properties of a manipulator are obtained by swinging it through various motions and measuring the torques on the motors [2] using the linearity of inertial parameters in the nonlinear dynamical equations. This fact is also useful in adaptive control [3–6].

Extrinsic calibration is another important field, which describes the fixed

3D rigid-body transformation either between a sensor and the mounted body frame or between two sensors fixed on the same rigid body. The sensors that can benefit from such calibration techniques include but are not limited to monocular cameras, stereo cameras, ultrasound (US) probes, and IMUs. Once the unknown rigid-body transformations in the system are calibrated, one can determine the pose of the target more accurately. This is beneficial in many ways such as helping the manipulator to better determine the configuration of the object to be grasped, improving the accuracy in constructing the 3D environment and facilitating the localization of a robot. Calibrating the extrinsic parameters of various sensors can also help with the sensor fusion in a system. From the perspective of simulation, employing the calibrated configurations of sensors can enable the simulation to better reflect the implementation in the physical world compared to using artificial sensor frames or those obtained from the spec sheet.

In the fields of robotics and computer vision, sensor calibration problems are often codified using the "$AX = XB$" formulation. Example applications include camera calibration, robot eye-to-hand calibration [7], aerial vehicle sensor calibration [8], image-guided therapy (IGT) sensor calibration and endoscopic surgery [9]. An alternative name that describes this system is "hand-eye" calibration. Several methods exist in the literature that can perform hand-eye calibration without directly dealing with the $AX = XB$ formulation such as [10–16]. Another problem, called the robot-world and hand-eye calibration, deals with the case where an additional calibration of the robot pose with respect to the world frame is needed. This calibration technique is formulated as the "$AX = YB$" problem and multiple methods have been developed such as [17–23]. Furthermore, a hand-eye, tool-flange, and robot-robot calibration problem is formulated as the "$AXB = YCZ$" problem in [24, 25].

In Chapter 4, a comprehensive review on robot & sensor calibration, specifically the hand-eye calibration, is given to provide an overview of the history, state-of-the-art and other related techniques of $AX = XB$ calibration. Some of the mathematical concepts and theorems are also covered which will be heavily used in later chapters. Most calibration algorithms deal with the case where there is exact correspondence among the sensor data. However, this might not be true for asynchronous systems. Therefore, through Chapter 5 to Chapter 7, various probablistic solvers are presented for $AX = XB$, $AX = YB$ and $AXB = YCZ$ calibrations. Chapter 5 discusses different approaches to obtain the mean of a set of $SE(3)$ rigid-body transformation matrices, which can improve the computation accuracy of a probabilistic $AX = XB$ solver. Chapter 6 focuses on using probabilistic formulation of the $AX = YB$ equation to recover the correspondence of shifted data streams and boost the performance of non-probabilistic $AX = YB$ solvers. Chapter 7 looks at the multi-robot calibration scenario and presents two probabilistic approaches in comparison with the state-of-the-art $AXB = YCZ$ solvers. A hybrid approach is also discussed to better handle noisy data.

Chapter 2

Mathematical Background

This chapter presents an overview of the mathematical concepts that will be heavily used in the dissertation.

2.1 Group Theory

A group is a set G with a binary operation $\circ$ where the elements in the group satisfy the following four principles:

- Closure: For all $g_1, g_2 \in G$, $g_1 \circ g_2 \in G$.
- Associativity: For all $g_1, g_2, g_3 \in G$, $(g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$
- Identity element: There exists an identity element e such that, for every $g \in G$, $e \circ g = g \circ e = g$.

- Inverse element: For each $g \in G$, there exists a unique $g^{-1} \in G$, such that $g^{-1} \circ g = e$.

2.1.1 Matrix Lie Group and Lie Algebras

One type of group which is widely used in the area of robotics is the matrix Lie group. A matrix Lie group is a group whose element $g \in G$ is an $N \times N$ dimensional matrix and group operation $\circ$ is matrix multiplication. To every Lie group, there is an associated Lie algebra $\mathfrak{H}$ whose vector space is the tangent space of the Lie group at the identity element:

$$g(t) = \exp(tX) \tag{2.1}$$

where $X \in \mathfrak{H}$ and t close to zero and $\exp$ represents matrix exponentiation. There are two important matrix Lie groups in the field of robotics, special orthogonal group $SO(n)$ and special Euclidean group $SE(n)$.

The special orthogonal group $SO(n)$ is defined as

$$SO(n) \doteq \{R \,|\, RR^T = R^T R = \mathbb{I}, \det(R) = 1, R \in \mathbb{R}^{n\times n}\} \tag{2.2}$$

and the associated Lie algebra $so(n)$ is

$$so(n) \doteq \{\Omega \,|\, R = \exp(\Omega), \Omega \in \mathbb{R}^{n\times n}, R \in SO(n)\} \tag{2.3}$$

Two cases of interest in the dissertation are $n = 2$ and $n = 3$, which relate to planar rigid-body rotation and spatial rigid-body rotation. When $n = 3$,

$$\Omega = \widehat{\omega} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{pmatrix}. \tag{2.4}$$

where $\omega = [\omega_{\mathbf{1}}, \omega_{\mathbf{2}}, \omega_{\mathbf{3}}]$ and $\widehat{}$ is called the "hat" operator that converts a 3×1 vector into a 3×3 skew symmetric matrix. The inverse "vee" operation $\vee$ does the following

$$\Omega^{\vee} = \omega \tag{2.5}$$

The special Euclidean group $SE(n)$ is defined as

$$SE(n) \doteq \{H \,|\, H = \big(R,\ \mathbf{t}; \mathbf{0}^T, 1\big) \in \mathbb{R}^{n+1 \times n+1}, R \in SO(n), \mathbf{t} \in \mathbb{R}^n\} \tag{2.6}$$

whereas the associated Lie algebra is

$$se(n) \doteq \{\Xi \,|\, H = \exp(\Xi), \Xi = \big(\Omega,\ \xi;\ \mathbf{0}^T, 0\big) \in \mathbb{R}^{n+1 \times n+1}, \Omega \in so(n),\ \xi \in \mathbb{R}^n,\ H \in SE(n)\} \tag{2.7}$$

When $n = 2, 3$, $SE(n)$ represents the rigid-body motions in 2D and 3D respectively. Take $n = 3$ for an example,

$$\Xi = \log(H) = \widehat{\mathbf{h}} = \begin{pmatrix} 0 & -h_3 & h_2 & h_4 \\ h_3 & 0 & -h_1 & h_5 \\ -h_2 & -h_1 & 0 & h_6 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{2.8}$$

where $\mathbf{h} = \log^{\vee}(H) \in \mathbb{R}^{6\times 1}$ and $\widehat{\mathbf{h}}$ is the vectorized Lie algebra element, $\widehat{\mathbf{h}} \in se(3)$, such that $H = \exp(\widehat{\mathbf{h}})$. The "hat" operator $\widehat{\ } : \mathbb{R}^{6\times 1} \to se(3)$ maps a 6 by 1 vector to its corresponding Lie algebra which satisfies the exponential map $\exp : se(3) \to SE(3)$.

2.2 Integration on SE(3)

Let $\Sigma = \Sigma^T \in \mathbb{R}^{6\times 6}$ be a positive definite covariance matrix. Assuming the norm $||\Sigma||$ is small, we define a Gaussian probability distribution function on $SE(3)$ as:

$$\rho(H; M, \Sigma) = \frac{1}{(2\pi)^3 |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}F(M^{-1}H)}, \tag{2.9}$$

where $H \in SE(3)$, $M \in SE(3)$ is the mean, $\Sigma \in \mathbb{R}^{6\times 6}$ is the covariance matrix, $||\Sigma||$ denotes the determinant of Σ and

$$F(H) = [\log^{\vee}(H)]^T \Sigma^{-1} [\log^{\vee}(H)]. \tag{2.10}$$

If $Z \in se(3)$ is the corresponding Lie algebra of $H \in SE(3)$, then exp$(Z) = H$, log$H = Z$ and $\log^{\vee}(H) = \mathbf{z}$ where $\mathbf{z} \in \mathbb{R}^{6\times 1}$ is the vectorization of Lie algebra.

$SE(3)$ is a six-dimensional matrix Lie group, and there is a unique and correct way to define integration on $SE(3)$ called the "Haar measure", and is denoted as dH. If $H = H(\mathbf{q})$ where $\mathbf{q} = [q_1, ..., q_6]^T$ is a global set of coordinates, then functions $f : SE(3) \rightarrow \mathbb{R}$ can be integrated as

$$\int_{SE(3)} f(H)dH \doteq \int_{\mathbf{q}\in D} f(H(\mathbf{q}))|J(\mathbf{q})|d\mathbf{q}$$

where D is the domain of integration in the parameter space and $d\mathbf{q} = dq_1 dq_2 \cdots dq_6$. The Jacobian determinant $|J(\mathbf{q})|$ is computed from the Jacobian matrix:

$$J(\mathbf{q}) = \left[\left(H^{-1}\frac{\partial H}{\partial q_1} \right)^{\vee}; \left(H^{-1}\frac{\partial H}{\partial q_2} \right)^{\vee}; \cdots \left(H^{-1}\frac{\partial H}{\partial q_6} \right)^{\vee} \right].$$

For example, if Cartesian coordinates are used for the translation vector and ZXZ Euler angles are used for rotations, then $\mathbf{q} = [x, y, z, \alpha, \beta, \gamma]^T$, $D = \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times [0, 2\pi] \times [0, \pi] \times [0, 2\pi]$ and $|J(\mathbf{q})| = \sin\beta$. While D and $J(\mathbf{q})$ will change depending on which parameterization is used, the value of the integral itself does not as it is a property of the Lie group. The integral depends on the function themselves but not a matter of how the function is expressed or the coordinates used for integration.

$SE(3)$ is unimodular, which means that the integration measure, $dH =$

$|J(\mathbf{q})|d\mathbf{q}$, has the property that for any fixed $H_0 \in SE(3)$ and "well-behaved function[1]" $f : SE(3) \to \mathbb{R}$, [26]

$$\int_{SE(3)} f(H_0 \circ H)dH = \int_{SE(3)} f(H \circ H_0)dH = \int_{SE(3)} f(H)dH. \tag{2.11}$$

In addition, it can be shown that when these conditions hold, so too does

$$\int_{SE(3)} f(H^{-1})dH = \int_{SE(3)} f(H)dH. \tag{2.12}$$

A common source of confusion is that many books on Lie groups are concerned with compact Lie groups, which possess both bi-invariant metrics and bi-invariant integration measures. When discussing the noncompact case, bi-invariant metrics generally do not exist (except for special cases such as products of tori and Euclidean spaces), and they do not exist for $SE(3)$. Though bi-invariant integration measures also do not exist in general, they do exist for a broader class of special noncompact Lie groups, and this includes $SE(3)$.

[1] Here a well-behaved function means a function for which the integral exists, and hence $f \in L^1(SE(3))$, and later that the convolution integral exists, which is guaranteed by further requiring that $f \in L^2(SE(3))$. And so, with the notable exception of the Dirac delta function, the discussion is restricted to $f \in (L^1 \cap L^2)(SE(3))$.

2.3 Convolution on SE(3)

The convolution of two well-behaved functions is defined as [27,28]:

$$(f_1 * f_2)(H) = \int_{SE(3)} f_1(K) f_2(K^{-1}H)\, dK \tag{2.13}$$

where $K, H \in SE(3)$. The integral over $SE(3)$ can be expressed in various coordinates and here we choose the exponential coordinates, where the six-dimensional integral over $SE(3)$ and its measure can be found in [29].

Convolution inherits the associative property from the underlying group, which is written as

$$(f_1 * f_2) * f_3 = f_1 * (f_2 * f_3)$$

where the dependence of these functions on H has been temporarily suppressed. Analogous with the way convolution inherits associativity, it also inherits non-commutativity for general functions, with the exception of special functions called "class functions".

The Dirac delta function on $SE(3)$ can be thought of as follows:

$$\delta(H) = \begin{cases} +\infty, & H = I \\ 0, & H \neq I \end{cases} \tag{2.14}$$

which satisfies the constraint that:

$$\int_{SE(3)} \delta(H)dH = 1. \tag{2.15}$$

A slightly further expansion of allowable functions to include shifted delta functions of the form:

$$\delta_X(H) \doteq \delta(X^{-1}H) = \delta(HX^{-1}). \tag{2.16}$$

The unshifted delta function is an example of a symmetric function, in that $\delta(H) = \delta(H^{-1})$.

If the class of functions is expanded to consider beyond $(L^1 \cap L^2)(SE(3))$ to include Dirac delta functions, then the following is true for every $f \in (L^1 \cap L^2)(SE(3))$:

$$(f * \delta)(H) = (\delta * f)(H) = f(H). \tag{2.17}$$

Eq. (2.15) and Eq. (2.17) constitute the formal definition of delta function on $SE(3)$.

Using the properties of the invariant integral on $SE(3)$, convolving a shifted delta function with an arbitrary function transfers the shift:

$$\begin{aligned}(\delta_X * f)(H) &= \int_{SE(3)} \delta(X^{-1}K)f(K^{-1}H)dK \\ &= \int_{SE(3)} \delta(J)f((XJ)^{-1}H)dK = f(X^{-1}H)\end{aligned} \tag{2.18}$$

where the change of variables $J = X^{-1}K$ and the invariance of integration have been used.

Chapter 3

Robotic Assisted Catheter Tracking

Robotic-assisted catheter insertion is becoming increasingly popular due to its potential applications including cardiac catheterization. Typically, catheters are tracked during insertion procedures to verify the location of the tip relative to anatomy or features of interest, and this procedure is performed exclusively with X-ray. Ultrasound is a good alternative because it is radiation free. However, the catheter tip is usually very small and hard to be differentiated from human tissue in an ultrasound image. Moreover, an operator needs to hold the ultrasound probe for the entire procedure lasting for over an hour which is very physically demanding. As a result, this chapter proposes an ultrasound-enabled robotic catheter tracking system that uses a 2D ultrasound probe and an active piezoelectric element to track the tip of a catheter. This system, to the best of the author's knowledge, is the first robotic catheter tracking system using ultrasound. This approach has the potential to guide catheters from initial insertion, in a vein in the groin, to final placement at a target area inside of the heart. During the tracking process, no information from the ultrasound image is necessary; however, this information can be used to help clinicians to steer the catheter or to perform diagnostic procedures.

This chapter outlines the procedure by first discussing the individual components of the system and then by describing the methodology for tracking the catheter tip. Next, the system is simulated in ROS to test its effectiveness, and finally we experimentally verify that a robotic arm equipped with a 2D ultrasound probe can accurately track a catheter in a multi-vein phantom. Furthermore, the data collected during tracking are used to virtually reconstruct the 3D structure of veins while tracking. The remainder of this chapter is organized in the following manner: Section 3.2 discusses, in detail, the elements and components of the RUSTCAPE system; Section 3.3 develops the control methodology and introduces the ROSAUS system and its corresponding simulation environment; Section 3.4 presents and analyzes the experimental results; and finally Section 3.5 draws the conclusions.

3.1 Introduction

Cardiac catheterization is a common diagnostic and interventional procedure employed by physicians around the world. In fact, according to the United States' Centers for Disease Control and Prevention, in 2010 there were approximately 1 million cardiac catheterizations in the United States alone [30]. Cardiac catheterization uses catheters to monitor a patient's heartbeat, diagnose cardiovascular disease, or place interventional devices such as pacemakers or stents. During these procedures, the catheter is typically inserted through a small incision in the groin region and then into a large blood vessel and artery. Once inserted, the catheter is manually pushed by a physician to the proper location in or around the heart. The placement of the catheter tip is then confirmed by a diagnostic X-ray, which results in the patient, radiologist, and physician being exposed to potentially harmful ionizing radiation [31–33].

The amount and effect of this radiation depends not only on the type of procedure (diagnostic or interventional) [31], but also the location of exposure [34] and the age of the person exposed [35]. In general, repeated exposure to X-rays or ionizing radiation above normal levels has been linked to an increased risk of cancer. This risk is pronounced in children and adolescents especially in patients with repeated and cumulative exposures [36]. The risk of radiation exposure is even further increased with diagnostic procedures such as

X-ray fluoroscopy where a dye is injected in the patient to monitor heart flow and movement. Therefore, a need exists to develop methods and systems that reduce radiation exposure during cardiac catheterization and subsequent diagnostic or interventional procedures.

3.1.1 Robotic Catheter Systems

One approach that is increasingly gaining the attention of researchers is the use of robotic systems for catheter insertion. These systems seek to reduce or possibly eliminate radiation exposure to radiologists and physicians by providing teleoperative or autonomous capabilities for insertion. One such system, developed by Jayender *et al.* [37], uses visual feedback from some imaging modality to autonomously insert the catheter into the patient . More recently, a second robotic system was described by Loschak *et al.* [38] that can precisely control the tip and insertion of an ultrasound (US) imaging catheter; however, the autonomous use of this system has yet to be shown . The main drawback of these systems is that they still require the patient to undergo X-rays to ensure proper placement of the catheter tip. On the commercial platform side, Hansen Medical developed two robotic catheterization systems called Sensei ® X and Magellan ™ respectively, which don't provide radiation-free tracking technology. A more complete review on the robotic catheterization systems can be seen as in [39].

As a result, other researchers have developed systems to track catheters using modalities without ionizing radiation, which effectively decrease exposure to patients, radiologists, and physicians, such as electromagnetic (EM) and bioelectric navigation. In fact, Condino *et al.* [40] showed that EM navigation and tracking of a catheter tip had no significant difference from its X-ray fluoroscopy counterpart. However, there are some cases where US is preferred over EM tracking, as it is also non-ionizing, unobtrusive, and can provide additional diagnostic or interventional capabilities. Fuerst *et al.* [41] proposed a bioelectric navigation technique to determine the relative position of the catheter within the vessel tree, which is inspired by the electrolocalization behavior of weakly electric fish.

Currently, US is being used in a variety of robotic procedures including needle biopsy, high-intensity focused US ablation, thrombosis, and catheterization [42–44]. In many of these procedures, the main control method is US visual-servoing (USVS), where a particular feature is used to inform the control strategy to reposition the ultrasound probe so that a desired cross-section image is reached. Specifically, Mebarki *et al.* [45] used USVS to actuate a US probe to reach a desired cross-sectional image of an object of interest. However, the proposed visual servoing technique is devoted for motionless objects and not applicable for catheter tracking.

3.1.2 The RUSTCAPE System

As a solution to some of these drawbacks, a new system on Robotic UltraSound Tracking of CAtheter with Piezoelectric Element (RUSTCAPE) is proposed which utilizes an embedded active piezoelectric (active echo (AE)) element [46,47]. The system consists of an AE element embedded near the tip of a catheter and a robotic manipulator (UR5) equipped with a US probe at the end-effector. The system also includes the Robot Operating System (ROS) [48] based Active echo-UltraSound system (ROSAUS) which connects the US system and UR5, and fulfils the control and visualization functions. Autonomous tracking of the catheter is then accomplished by using data obtained from the AUSPIS system to localize the AE element in reference to the US probe and hence the robotic manipulator. As the catheter is inserted by the physician, the resulting change in location and data sent from the AE element is used to provide an appropriate update and subsequent motion of the robotic arm. In addition to autonomous tracking, if the pattern injection functionality of the AE element is utilized as outlined in [46], the catheter tip can be more easily identified in the US image by clinicians improving the accuracy and effectiveness of diagnostic and interventional procedures. Therefore, by utilizing concepts outlined in [49] and by developing additional methodology, one can realize a fully autonomous catheter tracking system that reduces or eliminates the need for ionizing radiation and opens the possibility for further diagnostic

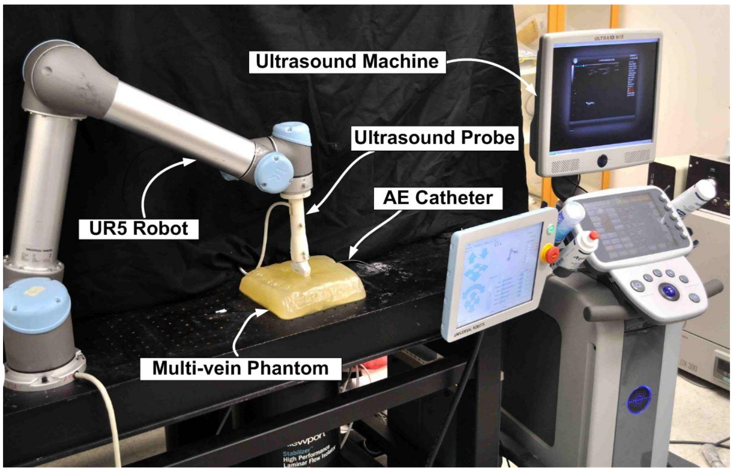


Figure 3.1: The RUSTCAPE system contains the phantom, Active Echo (AE) catheter, US machine, UR5 robot, US probe, teaching pendant, and AUSPIS system as displayed in Fig. 3.2. A workstation is also needed, which is not shown.

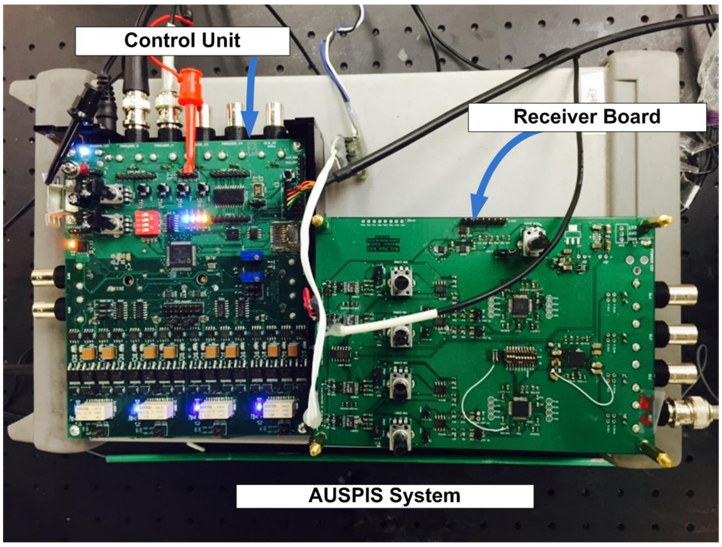


Figure 3.2: Active Ultrasound Pattern Injection System (AUSPIS) with the control unit and receiver board.

capabilities.

3.2 System Description

As shown in Fig. 3.1 and Fig. 3.2, the hardware of the RUSTCAPE system is comprised of five parts: a multi-vein phantom, a Sonix CEP system (US machine), the Active Ultrasound Pattern Injection System (AUSPIS) and active echo (AE) catheter, a workstation, and a UR5 robot arm. The software of the system, ROSAUS, includes the commercial Ultrasonix software, ROS packages, and the Linux-based Universal Robots graphical programming environment (URGPE) provided by the Universal Robotics company. In the rest of this section, a detailed description of the RUSTCAPE system will be provided.

3.2.1 The Multi-vein Phantom

The multi-vein phantom is formed from a mixture of plastisol and plastic softener (used for mimicking the stiffness and elasticity of human tissue) that is poured into a mold with multiple veins. The phantom has a footprint of $203 \times 203 \times 51\ mm^3$. The radius of the veins are 3.57 mm which is slightly smaller than the average measured radius, of approximately 5.5 mm, of pulmonary veins in adults [50]. We designed the veins of the phantom to have a tree-like structure as shown in Fig. 3.3. Note that all the veins are inside the phantom and invisible to human beings, and the depth of the branches are variable to better mimic human arteries. Therefore, light violet lines are overlaid on the picture of the phantom to show the approximate shape of the tree-like branches. The hole on the right side of phantom is the entrance where the physician, or the experimental operator, inserts the catheter with the AE element located at its tip. Depending on how the operator maneuvers, the catheter can traverse any of the branches and emerge from one of the six exits on the left side of the phantom. The material of the phantom is less dense than water, therefore it is impossible to submerge the phantom in a water bath without the use of a fixture. Every branch in the phantom was designed to terminate on the surface of the phantom with the actual veins traveling approximately 3 cm below the surface to permit the veins to hold fluid. Two holes are also present on the top surface which are connected to the six branches of the phantom. It is only necessary to fill one of these holes to completely fill all of the channels up to their respective entrances and exits, which enables transmission of the US wave through the entire phantom. Under this condition, the US probe has to be in contact with the top surface of the phantom to capture a US image and activate the AE element.

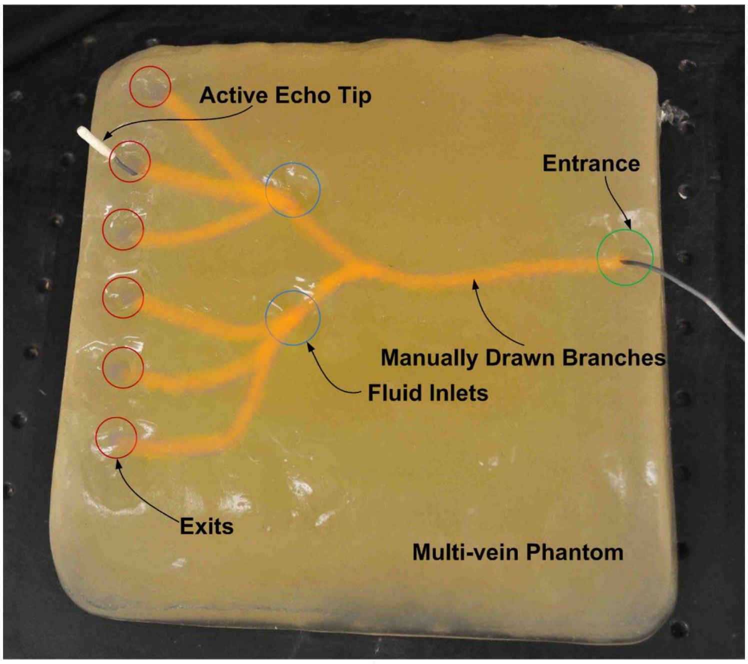


Figure 3.3: The multi-vein phantom with tree-like channels and the active echo catheter. Note that the six channels inside the phantom are invisible to the naked-eye and that the six orange branches are manually drawn for illustration purposes only.

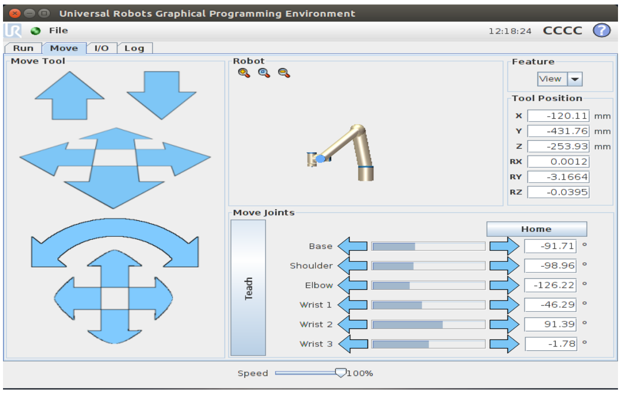
3.2.2 The AUSPIS System and Active Echo Element

The AUSPIS system was previously developed by Guo *et al.* [46] for accurate tool visualization and US mid-plane detection. The system achieved a maximal accuracy of ± 0.3 mm in placing an active echo element (AE element) within the US image mid-plane at a depth of over 8.5 cm which is beyond traditional US imaging resolution. The small US transducer (AE element) is a customized tube made of PZT5-H material with an outer diameter of 2.08 mm, an inner diameter of 1.47 mm, and a length of 2 mm. One embodiment of AUSPIS consists of a catheter with an AE element very close to the tip, a customized controller board, and a signal receiver. In this project, the output obtained from the controller unit will serve as the control input into ROS to manipulate the UR5.

Instead of using any type of visual servoing, the signal generated by the active echo element under the effect of US waves is used for tracking the tip of the catheter. The active echo element is integrated into the tip of a catheter shown as the white tip in Fig. 3.3. The small size of the AE element makes it possible to insert the catheter into the phantom without much effort. Once inside the detection range of the US probe, the element receives transmission beams from the US probe and counts how many of the acoustic waves transmitted by each probe element in a single image frame surpass a predetermined energy threshold. This count number is correlated with how far away the AE element is from the US image plane; however, the count number cannot be used to determine on which side the AE tip is located with respect to (w.r.t.) the US mid-plane. For in-plane position, the automatic segmentation of the point can be achieved by using the beacon delay method as described in [47]. This will be discussed in detail in Section 3.3.

3.2.3 The Workstation and UR5 Robot Arm

The UR5 is a lightweight industrial robot arm designed by Universal Robots [51]. The UR5's low cost and collaborative/safety features make it a potential



X -120.11 mm
Y -431.76 mm
Z -253.93 mm
RX 0.0012
RY -3.1664
RZ -0.0395

Base -91.71 °
Shoulder -98.96 °
Elbow -126.22 °
Wrist 1 -46.29 °
Wrist 2 91.39 °
Wrist 3 -1.78 °

Speed 100%

Figure 3.4: GUI of the Universal Robots Graphical Programming Environment (URGPE) used for simulation.

option for applications in clinical environments. In our experiment, the collaborative feature is necessary because a clinician or experimenter manually inserts the catheter alongside the robot. The UR5 is designed such that the force exerted by the robot on any surface (including an operator) is limited below the threshold for personal injury effectively guaranteeing an operator's safety. In addition, with 6 degrees of freedom and an end-effector positioning repeatability of $\pm 100\ \mu\mathrm{m}$, the UR5 is suitable for this AE element tracking task. The actual repeatability might differ depending on the usage time of the robot. A 58.5mm L14-5W US probe (Ultrasonix. Inc) is attached to the end-effector of the UR5 by a specially designed 3D printed holder.

A workstation is connected to the controller of the UR5 through an Ethernet cable and to the AUSPIS controller unit via an FTDI cable. The workstation sends interpolated time-stamped joint trajectories to the controller given the input received from the controller unit as in Fig.3.6. Our communication and

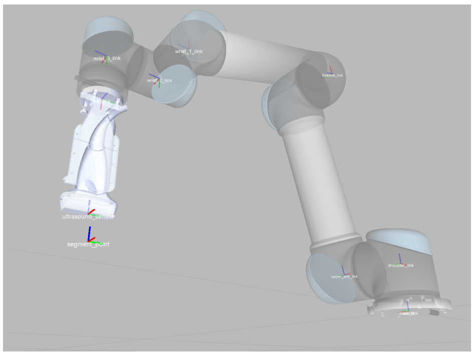


Figure 3.5: The UR5 with a US probe attached at the end-effector in Rviz. The reference frame below the US probe displays the reference frame of the detected AE element.

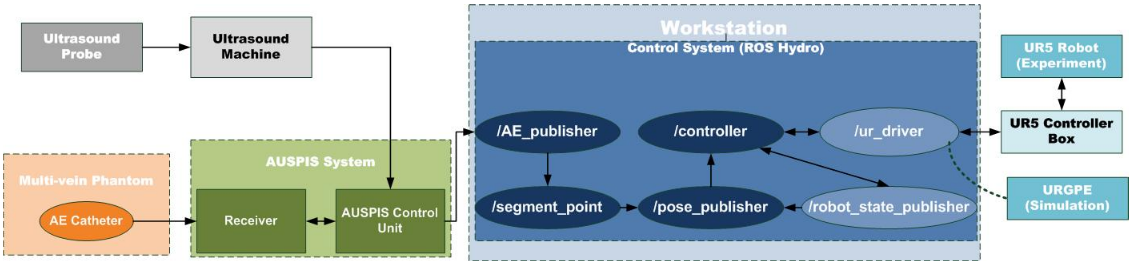


Figure 3.6: Diagram of the RUSTCAPE and core ROSAUS framework. The arrows denote the data flow of the system and the control system is illustrated in the blue boxes on the right. Ellipses within the blue box denote ROS nodes (Only the core ROS nodes are plotted here for simplicity. Dark blues ellipses indicate custom-made nodes for this project and the light blue ellipses indicate nodes available online).

control platform was built within ROS Hydro, and the online meta-package `universal_robot` developed under the ROS-industrial program [52] is used as the communication interface between ROS and the UR5.

3.2.4 Simulation and Experimental Environment

Because the operator will be inserting the catheter alongside the robot, simulation is still very necessary despite the safety features of the UR5. The

URGPE as shown in Fig. 3.4 is a Linux-based software that can simulate the behavior and conditions of the real robot. It resembles the interface of the teach pendant and is very helpful in building up the ROS-based control system in a simulation environment before directly applying the system to actual hardware. For real experiments, the workstation is directly connected to the controller of the UR5 and all the processes are monitored in ROS terminals, and visualized in a built-in GUI, Rviz, as shown in Fig. 3.5.

3.3 Methodology

In this section, the methodology is discussed in detail, including the framework of the system and the control strategy. A brief simulation is shown in a video clip to show the effectiveness of the system [53]. The complete RUST-CAPE system diagram is illustrated in Fig. 3.6. The arrows depict the major data flow within the system and the different colors in the diagram are used to separate the different functional parts of the system.

Note that though the US machine is connected to the control unit of the AUSPIS system, no information from the US image is transmitted during the process. Only the trigger information from the US machine is used to compute the in-plane position of the catheter tip.

3.3.1 Tip Position of the Catheter

As briefly discussed in Section 3.2.2, the beacon delay method can be used to obtain the in-plane position of the segmented AE element. This is achieved by sending the line trigger and frame trigger into the AUSPIS system for computation. The time of flight (TOF t_{tof}) of the US wave can be used to calculate the axial position of the AE element. n_{tof} denotes the transmitted acoustic waves from each probe element that surpass a pre-determined energy threshold. The counting of n_{tof} will start when the frame trigger is received and stop when the next frame trigger is received, and it is used for obtaining the lateral position. If the image frame of the US probe is defined as in Fig. 3.7, then the in-plane position denoted by (y, z) is obtained as:

$$y = n_{tof} * \frac{L_{probe}}{n_{total}} \tag{3.1a}$$

$$z = -t_{tof} * v_{sound} \tag{3.1b}$$

where $t_{tof} = t_{delay}/s_{rate}$, L_{probe} is the array length, n_{total} is the total number of line triggers in each frame, s_{rate} is the sampling rate, t_{delay} is the delay between a line trigger and the corresponding beacon signal received by the AE element, and v_{sound} is the speed of sound in the media.

The absolute value of the off-plane displacement $|x|$ is indicated by the count

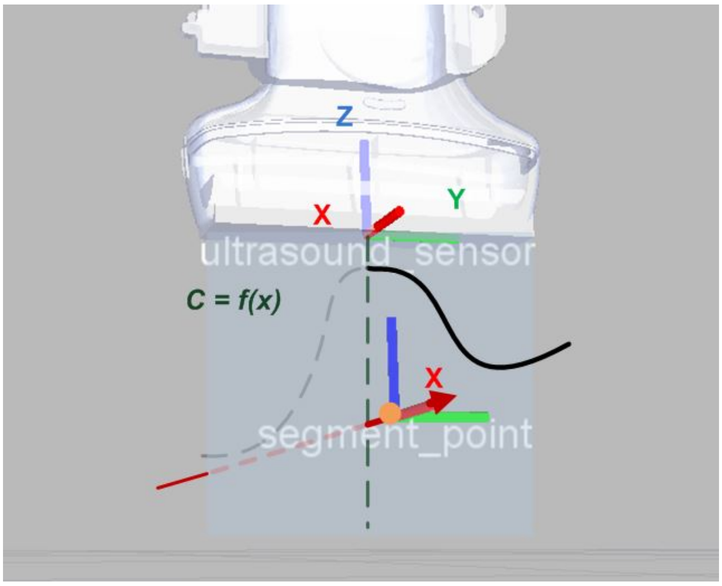
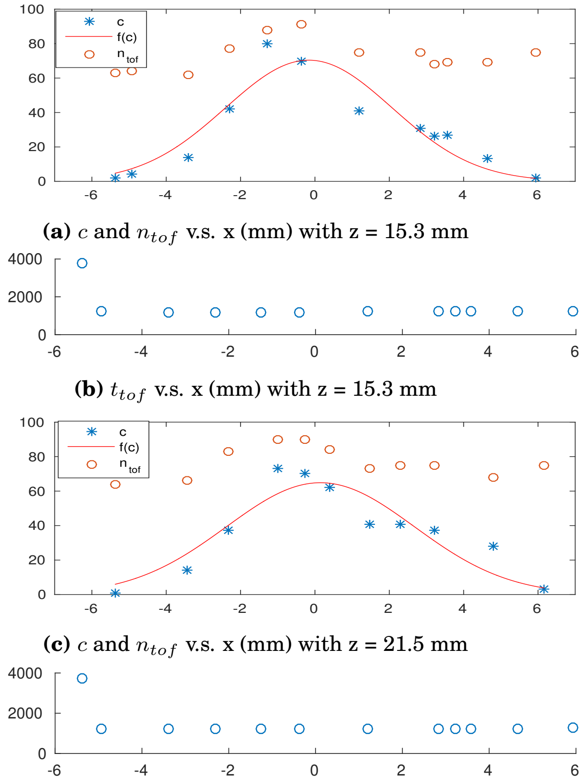


Figure 3.7: A close-up of the detected catheter tip (AE element) in the reference frame of the US probe. The orange dot denotes the physical position of the catheter tip. The count number c in terms of the off-plane position x has a Gaussian-like distribution if the environmental parameters are fixed.

number c which corresponds to the US beacon intensity detected by the AE element. It has been noted in [49] that $c = f(x)$ has a distribution similar to a Gaussian function whose amplitude is affected by the focus point of the US machine, the US beacon intensity emitted by the US probe, and the in-plane axial position z. Fig. 3.7 illustrates the reference frames of both the US probe and the detected AE element. In the water tank, data samples of c, t_{tof}, and n_{tof} are obtained for different x and z positions of the AE element as shown in Fig. 3.8. It can be seen that: 1) the amplitude of the fitting Gaussian distribution $f(c)$ decreases as the depth z increases; 2) the t_{tof} signal can be very noisy when the AE element is close to the limit of the detection range, which means t_{tof} becomes unreliable when c is very small (≤ 3); 3) n_{tof} might



(a) c and n_{tof} v.s. x (mm) with z = 15.3 mm

(b) t_{tof} v.s. x (mm) with z = 15.3 mm

(c) c and n_{tof} v.s. x (mm) with z = 21.5 mm

(d) t_{tof} v.s. x (mm) with z = 21.5 mm

Figure 3.8: Trigger count c, n_{tof}, and time of flight t_{tof} v.s. off mid-plane position x with different depths in z. $f(c)$ denotes the fitted curve of c into the Gaussian distribution on x. (Data collected in water tank)

fluctuate but is very reliable even when the AE element is about to go out of the detection range along the x axis. The relationship between the AE signals and positions above will be used as a benchmark for use in the phantom, because the current phantom has specific tree-like branches and it is not possible to choose x and z freely.

3.3.2 Tracking Strategy and Simulation

Because the surface of the phantom is almost flat and the US probe has to be kept in contact with it, the motion of the robot along the normal direction of the phantom surface (or equivalently the axial direction of the US probe) is disabled, and the problem is converted into a 2D tracking problem, while the virtual vein construction problem is still 3D. However, this is not a simple 2D position feedback control problem because in the phantom, the count number c can provide neither the exact relative position nor the relative direction of the AE element relative to the US mid-plane. When tracking the catheter, it is assumed that the catheter will initially move forward along the direction of insertion. If the catheter is moved backward, the system is designed to stay at the last point of forward motion. To facilitate the position adjustment and backward tracking, a GUI is built on top of the ROS package `dynamic_reconfigure` to enable/disable tracking functionality as well as selecting the tracking direction (forward/backward) and speed.

The control diagram is provided in Fig. 3.9, which shows how n_{tof}, t_{tof} and c are used to track the catheter and construct the virtual channels of the phantom. First, n_{tof} and t_{tof} are taken for segmentation to generate the in-plane position Y_{active} and Z_{active}. With the assumption that the AE element is in the mid-plane while detected, the x-axis position of the robot end-effector $X_{probe}(i)$ is used to "recover" the 3D position of the AE element along with Y_{active} and

Z_{active}, where i stands for the current pose and $i + 1$ denotes the next desired pose for tracking the catheter. This data is stored in P_{record} which is then plotted in Rviz to construct the virtual veins in real time. Count number c is used to calculate the moving step dX of the robot along the x-axis, so that the next desired x position is obtained by $X_{probe}+dX$. Given the Gaussian distribution as in Fig. 3.8, one can inversely calculate the absolute value of x as $x = finv(c) \geq 0$ when c is greater than a lower bound c_{low}, which marks the bound of reliable detection range. Since the phantom is flat, there is no motion along the z axis of the robot, so the current $Z_{probe}(i)$ is passed directly to the next $Z_{probe}(i + 1)$ in $P_{probe}(i + 1)$. Y_{active} is used for updating $Y_{probe}(i + 1)$ and the next position of the probe $P_{probe}(i + 1)$ is complete. Next, the position of the end-effector $P_{robot}(i + 1)$ is obtained and fed into the inverse kinematics of the UR5 to obtain the corresponding joint angles $q_1(i + 1), ..., q_6(i + 1)$. After the interpolation between the current joint angles and next joint angles, a joint trajectory is generated and sent to the controller of the UR5 for low-level PID control.

Back to the ROS system as shown in Fig. 3.6, the `AE_publisher` node reads the data socket from the AUSPIS control unit through the FTDI cable and parses it into n_{tof}, t_{tof} and c. It publishes these three variables to the rostopic named `active_echo_data` which is subscribed to by `segment_point`. The `pose_publisher` node subscribes to the estimated position of the AE element in the US image plane published by the node `segment_point`, and outputs

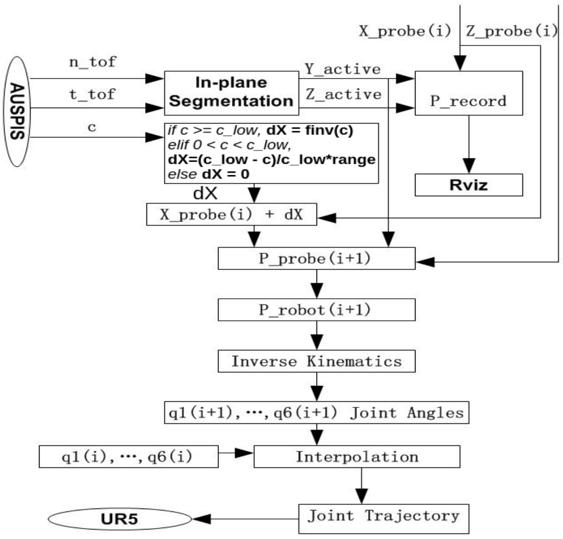
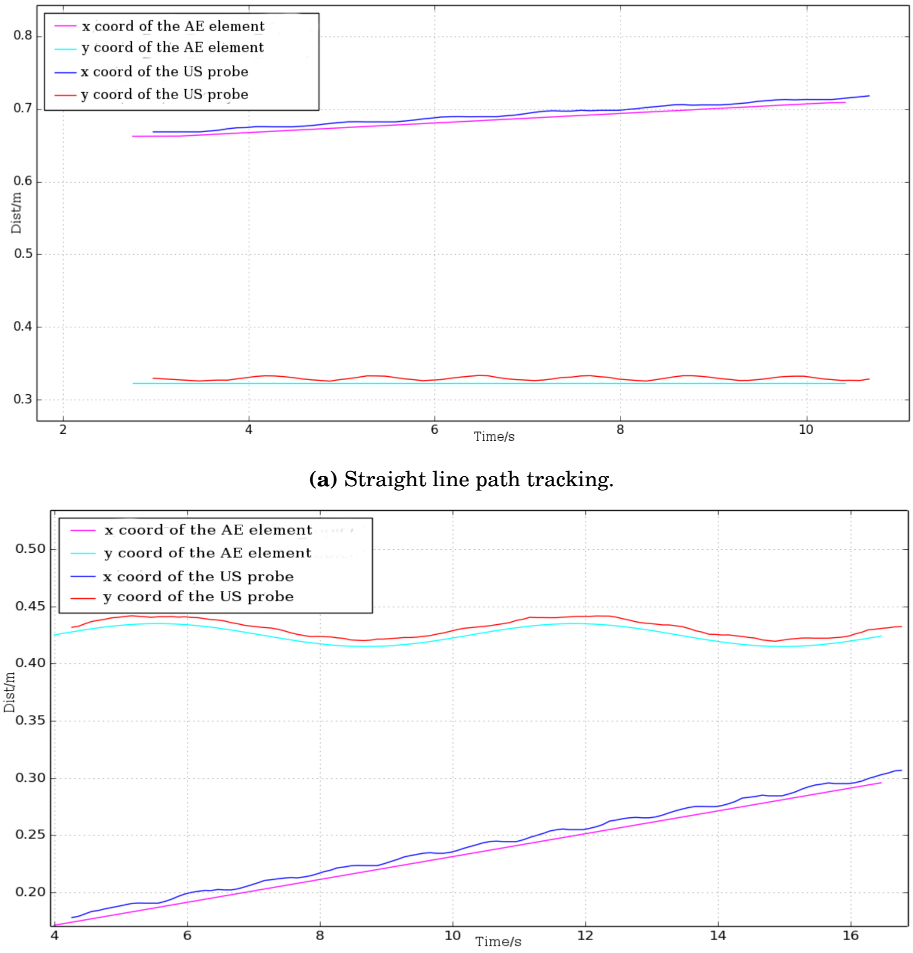


Figure 3.9: Control diagram and data flow in ROSAUS.

the next desired pose of the end-effector (`setpose`) of the UR5 for tracking the AE tip. Based on the current configuration of the robot, the `controller` node reads the newly published rostopic `setpose` and calculates the corresponding joint angles for the next step using Jacobian-based inverse kinematics. Then using the current joint angles, a first-order Taylor series update step is completed to generate the next joint angles in the trajectory. Next, the joint trajectory is time-stamped and sent to either the UR5 controller or the URGPE UR5 simulation software through the `ur_driver` node depending on whether one is performing an experiment or a simulation.

Two simple tracking simulations are performed in the ROSAUS system, one tracks an AE element moving along a straight line at a constant speed, and the other tracks an AE element moving along a sine wave path at a con-



(a) Straight line path tracking.

(b) Sine wave path tracking.

Figure 3.10: Straight line (a) and sine wave path (b) tracking of the AE element in ROSAUS. The horizontal axis denotes time (second) and the vertical axis denotes distance (meter) of the targets in the world frame.

stant speed. Fig. 3.10a and Fig. 3.10b are obtained directly from the `rqt_plot` of ROS in real time. The purple and cyan lines are the x/y position of the AE element in the world frame, while the blue and red lines are the x/y position of the US frame. It can be seen that the UR5 is able to maintain a very close separation between the ultrasound probe and the AE element so that it falls into the narrow 1 cm detection range along the x-axis and the 5cm detection range

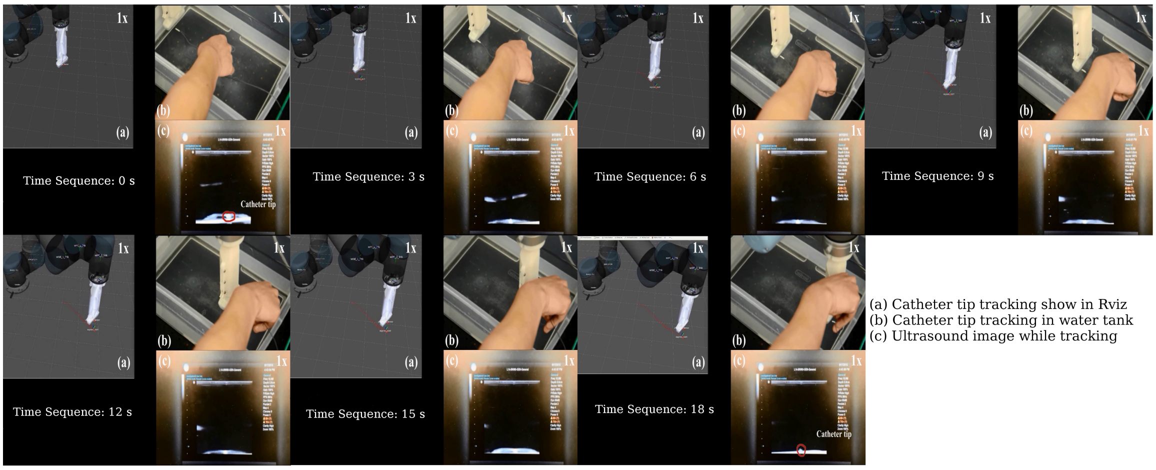
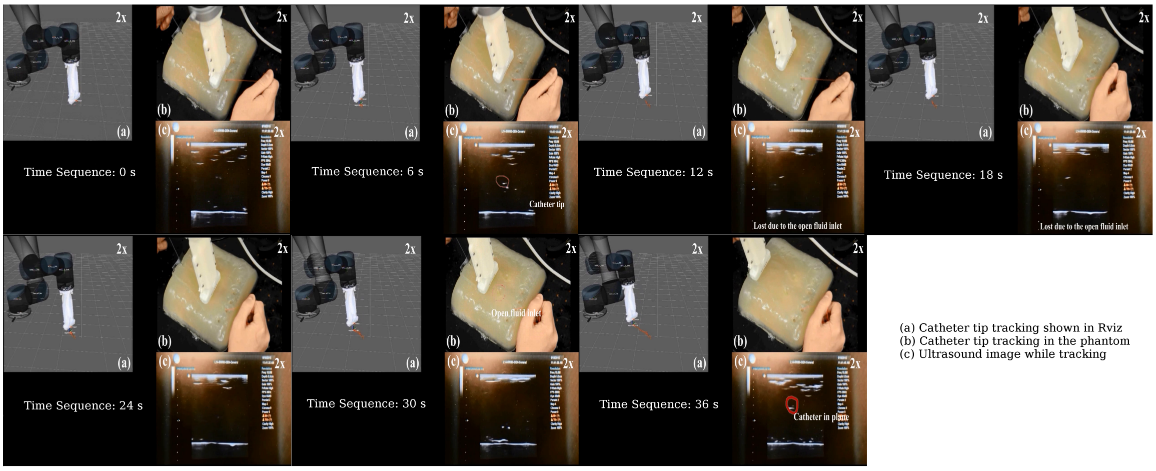


Figure 3.11: Time elapsed screen shots of catheter tracking experiment in the water tank. Seven pictures are taken over the course of 18 seconds.

along the y axis. Theoretically, the AE element should fall into the US mid-plane when c achieves the maximum; however, there is usually a small shift between the US mid-plane and the c_{max} position, and this partially contributes to the small shifts in Fig. 3.10.

3.4 Experiment Results

In this section, experiments are performed using the RUSTCAPE system, and both the effectiveness of the system and problems encountered are discussed. Experimental results are illustrated in combination with a video [53] to validate the performance of the system in action. Time elapsed screen shots for the experiments for both water tank and phantom can be seen as in Fig. 3.11 and Fig. 3.12. A time elapsed picture of the robot arm tracking the catheter can be seen in Fig. 3.13.



(a) Catheter tip tracking shown in Rviz
(b) Catheter tip tracking in the phantom
(c) Ultrasound image while tracking

Figure 3.12: Time elapsed screen shots of catheter tracking experiment in the phantom. Seven pictures are taken over the course of 36 seconds.

3.4.1 Experiments

Three operators are needed to use the system in a safe manner. To begin with, one operator inserts the tip of the catheter into the entrance of the phantom so that the AE element is submerged in water. Then the UR5 robot is manually operated to place the US probe over the entrance of the phantom and the AE element. The position of the probe can be further adjusted to detect the AE element until the count number c is larger than zero. The same operator inserts the catheter alongside the robot and a second operator holds the UR5 control panel in case a situation arises where the emergency stop button must be pressed. A third operator monitors the ROS terminals and sends commands to either start/stop or forward/reverse tracking.

In the video, it is demonstrated that the US probe can track the catheter tip from near the entrance of the phantom up to the vicinity of the exit au-

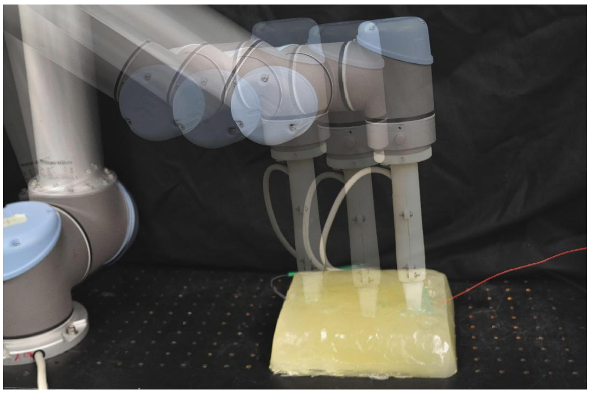


Figure 3.13: Robot in action while tracking the tip of the catheter maneuvered by a human being. Three poses are selected for illustration.

tonomously. The positions of the detected AE element are also recorded in Rviz to build a virtual structure of the vein in real time. Even though the channels are invisible, by reconstructing the virtual channels the system can gradually provide a more and more complete picture of the phantom, which can help with the maneuvering of the catheter in the future. Experiments show that one can traverse all of the six branches of the phantom and construct the virtual channel in Rviz for each of them. A screen shot of the constructed phantom channels during tracking is included in Fig. 3.14 which contains four out of the six branches of the phantom. The red dots are the raw data obtained from the AUSPIS system. The outliers as seen in Fig. 3.14 are caused by the shift of t_{tof} as designed in the AUSPIS system and can be easily filtered by post-processing in software such as Matlab. In addition, it is also demonstrated the US probe tracking the catheter tip inside the water tank and the vein path can be constructed in real time.

3.4.2 Discussion

Several observations are made after performing the experiments. First, the count number c plays a critical role in helping track the off-plane position x of the catheter tip. Its value depends on both its physical off-plane distance and the adjustable parameters of the US machine. In the water tank, the relationship between c and x is relatively stable and it mainly depends on the depth of the AE element, the type of ultrasound probe that is used, and the tuning position of the sensitivity knob on the receiver board of the AUSPIS system. In the phantom, however, the maximum of c can range from 10 to 60 depending on where the catheter is located. For example, the amplitude of $c = f(x)$ will be larger if the focus of the US probe is closer to where the AE tip is located. However, this condition is difficult to achieve because of the curved shapes of the channels inside of the flat phantom. The AE tip will be very close to the top surface of the phantom near the entrance and exits while close to the bottom in the middle, which can also cause c to fluctuate. This issue may be fixed in an autonomous manner by hacking into the US machine via the available API and adjusting the focus based on the current axial position z of the AE tip. In addition, the amplitude can also experience a noticeable increase if the probe contacts the phantom with greater force. As the amplitude increases, the detection range of the AE element is also broadened. This will be very useful since the off-plane detection range of the AE element varies from several millimeters to one centimeter. The length of the search step of the robot is also in the scale of millimeters, so even a slight increase in the detection range can enable the robot to better track the catheter. However, too much contact between the phantom and probe can deform the channel and increase friction to a point where the catheter will get stuck.

Second, the tracking speed of the catheter in the water is faster than that in the phantom. In the experiments, as shown in the video, it took approximately 17 seconds for the catheter to traverse an 46 cm long water tank, whereas it took 38 seconds to the go through an 20 inch long phantom. The detection range of the AE element is relatively larger in the water tank, and the reliable range is 8 - 10 mm as shown in Fig. 3.8. However, in the phantom, the entire detection range of the AE element varies from several millimeters to one centimeter depending on how tight the probe is in contact with the phantom. Therefore, both the catheter and the robot have to move at a lower speed so that the catheter will not move out of the detection range in the phantom.

Thirdly, the robot arm and hence the US probe can cease to move due to the contact force and friction between the phantom and the probe. The probe is usually placed deeper onto the phantom to achieve a better detection range of the AE element; however, the slow speed of the robot results in moving steps as small as 1 mm which ceases robot motion due to large forces generated by the local unevenness of the phantom. This issue is currently resolved by setting

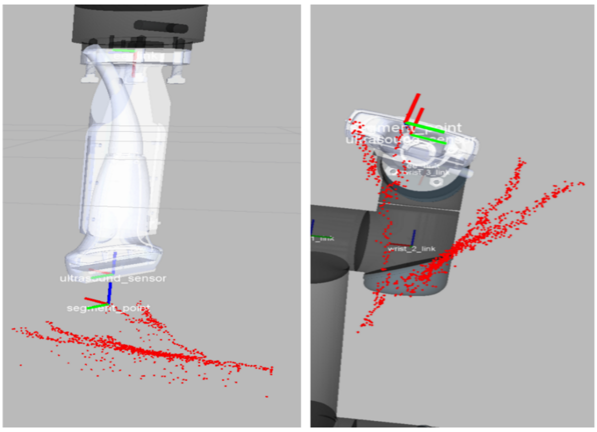


Figure 3.14: Side view and bottom view for the four out of a total of six branches are constructed in Rviz by tracking the AE tip of the catheter.

a larger step length in the beginning of the process and applying ultrasound gel periodically. An ideal solution would be to incorporate force feedback to automatically adjust the probe contact force with the phantom.

In addition, for the phantom, there are currently two fluid inlets on its top surface, and this can affect the interaction between the US probe and the AE element when the probe is passing over the holes. The received signals from the AUSPIS system can become unstable at the inlets, but this can be fixed by designing a phantom that has no holes on top.

3.5 Conclusion

In this chapter, a new US-assisted robotic system for catheter tracking is proposed. The active piezoelectric element is installed at the tip of the catheter for tracking purposes. A phantom is specially built to mimic human tissues and arteries/veins. Both simulations and real experiments are performed to demonstrate that the 2D US probe is able to track the catheter by detecting the AE element at the tip, and the tree-like structure of the channels inside the phantom can be reconstructed from the experimental data. The above features show that the system is low-cost and efficient in terms of computational intensity. Currently, the position of the phantom is unknown to the robot and the probe is manually moved to the vicinity of the phantom for locating the initial position of the tip catheter. The catheter is also manually inserted by an operator instead of using a 2nd robot arm. However, if one wants to further automate some of the procedures such as locating the phantom autonomously and inserting the catheter using a 2nd robot arm, then calibrations between the phantom and the robots become very necessary. The later chapters discuss the calibration techniques that can be potentially applied for augmenting the existing system. However, both of the reviewed and novel calibration techniques that will be covered can be used in various kinds of robotic systems where the same problems arise.

Chapter 4

Robot and Sensor Calibration

The calibration of a robotic system is the prerequisite for the whole system to function wtih high accuracy. This is especially true when there are multiple robots and sensors in the system. Out of various types of calibrations, the external calibration of the relative transformation between two fixed rigid bodies becomes more and more useful. In the context of a robotic system, the two rigid bodies can be a robot and a sensor, a robot and a robot, or a robot and a marker. The very original problem deals with the calibration between a robot and a sensor, which is also known as the robot hand-eye calibration. This problem is usually cast as the $AX = XB$ problem. Many methods have been proposed to solve X given data streams of A and B under different scenarios. This review chapter presents a complete picture of the $AX = XB$ solvers up to date. In Section 4.1, a brief overview of the various important sensor calibration techniques is given and problems of interest are highlighted. In Section 4.2, a detailed review of the various $AX = XB$ algorithms is presented. The notations used in the literature are very different from each other and this can cause confusion for readers. A major effort is spent on unifying the notations for all of the classic and state-of-the-art solvers to clearly reflect their inner connections. In Section 4.3, the criterion for data selection and various error metrics are introduced, which are of critical importance for evaluating the performance of $AX = XB$ solvers. Note that some of the techniques and concepts are directly applied in the novel calibration solvers to be covered in later chapters.

4.1 $AX = XB$ Calibration

In the $AX = XB$ formulation A, X, and B are each homogeneous transformations with each pair of measurements (A, B) coming from sensors such as cameras, US probes, optical, or electromagnetic pose tracking systems, among others. X is the unknown rigid-body transformation that is found as a result of solving $AX = XB$. It is well known that it is not possible to solve for a unique X from a single pair of exact (A, B), but if there are two instances of independent exact measurements, (A_1, B_1) and (A_2, B_2) satisfying certain constraints, then the problem can be solved for a unique, fixed X. However, in practice, sensor noise is always present, and an exact solution is not possible. The goal, therefore, becomes one of finding an X with the least error given corresponding noisy pairs of data (A_i, B_i) for $i = 1, 2, \ldots, n$.

The Hand-eye calibration problem, or the $AX = XB$ problem dates back to the 1980s when Shiu and Ahmad [54] first proposed a solution. Various $AX = XB$ solvers have been proposed so far for either same or different scenarios and applications. Tsai [7] proposed a method as well as several principles on data selection in 1989. A geometric view of $AX = XB$ is given by Fassi and Legnani [55] and the over-constrained and singular conditions are also discussed. Many other $AX = XB$ methods include but are not limited to the Euclidean Group method [56], the quaternion method [57, 58], the dual quaternion method [59], the Kronecker method [60], and the screw motion method [61]. Several new optimization methods emerged recently such as the convex optimization method [62] and the global optimization methods [22]. Most of the above methods are designed for off-line usage where a complete list of data pairs (A_i, B_i) with $i = 1, 2, ..., n$ has to be provided to recover X whereas online methods are more preferable in real time applications [60, 63]. A common feature of these methods is that they demand the exact correspondence between the data streams $\{A_i\}$ and $\{B_i\}$. This might not always hold because of asynchronizicity of the sensors or missing data, etc. Therefore, a probabilistic method called the “Batch method” that deals with data pairs without correspondence was first brought up in [64]. An information-theoretic approach is proposed in [65] by viewing the problem as distributions on the special Euclidean group $SE(3)$ and minimizing the Kullback-Leibler divergence on the distributions.

4.1.1 The Mathematical Formulation

Given:

$$AX = XB \tag{4.1}$$

where $A, B, X \in SE(3)$, it is well known that, in non-degenerate cases, there are two unspecified degrees of freedom to the problem for a single pair of sensor measurements, (A, B). This situation is rectified by considering two pairs of exact measurements of the form in Eq. (4.1), i.e., $A_{12}X = XB_{12}$ and $A_{23}X = XB_{23}$, provided that some mild conditions are observed for the selection of the pairs (A_{12}, B_{12}) and (A_{23}, B_{23}) [56, 66, 67]. Note that (A_{ij}, B_{ij}) here are relative transformation data pairs, and (A_i, B_i) & (A_j, B_j) are absolute transformation data pairs. For simplicity of notation, we will use (A_i, B_i) to denote the relative transformation data pair, which is also the desired input data for any $AX = XB$ solver. So in real experiments at least 3 absolute transformation data pairs have to be measured. Additionally, if there is sensor measurement error, then it may not be possible to find compatible pairs that reproduce the exact value of

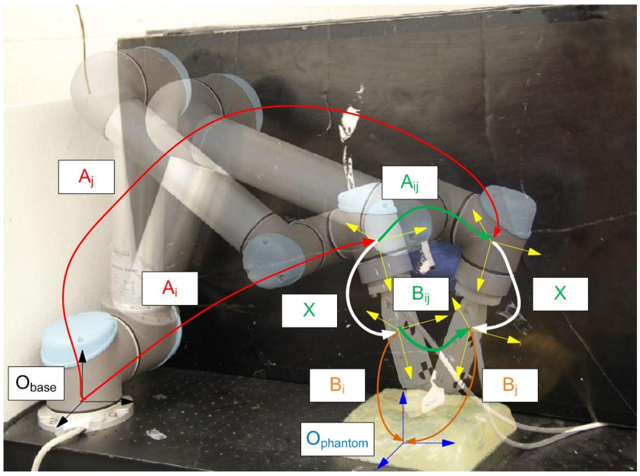


Figure 4.1: Application of $AX = XB$ in ultrasound sensor calibration: an ultrasound probe is attached to the end-effector and the calibration phantom is used for ultrasound probe calibration.

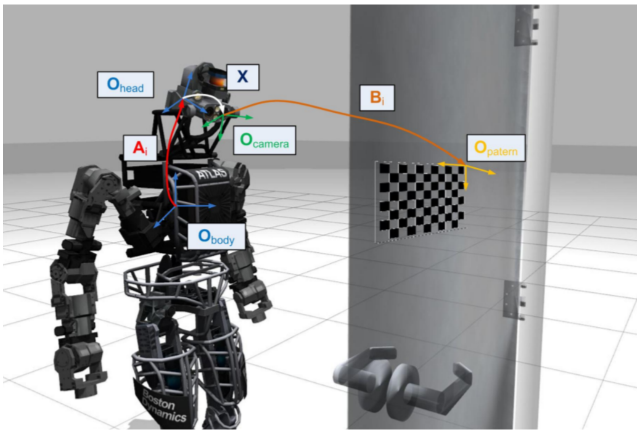


Figure 4.2: Application of $AX = XB$ in humanoid camera calibration: The humanoid robot named Atlas designed by *Boston Dynamics* requires head-body calibration before turning the knob. Different pairs of (A_i, B_i) are measured by changing the head pose of the humanoid robot.

X. For this reason, minimization and least squares approaches are often taken over large sets of $A's$ and $B's$.

The following two equations can be obtained by separating out the rotational and translational parts in Eq. (4.1):

$$R_A R_X = R_X R_B \tag{4.2a}$$

$$R_A \mathbf{t}_X + \mathbf{t}_A = R_X \mathbf{t}_B + \mathbf{t}_X. \tag{4.2b}$$

There are in general two approaches to solve the unknown rotation and translation. One way solves Eq. (4.2a) for only rotation and then rearranges Eq. (4.2b) to obtain the solutions of $\mathbf{t}_X$ satisfying $(R_A - \mathbb{I}_3)\mathbf{t}_X = R_X \mathbf{t}_B - \mathbf{t}_A$. The other way is to solve for R_X and $\mathbf{t}_X$ simultaneously based on some cost function $f(R_X, \mathbf{t}_X)$ or reformulating the matrix equation into other representations such as the dual quaternion. However, more than two exact sensor measurements are needed for the first approach. As pointed out in [56, 67], in nondegenerate cases, there is a one-parameter set of solutions to Eq. (4.2a), and the matrix $R_A - \mathbb{I}_3$ in general has a rank of 2. Hence, there are two unspecified degrees of freedom on the whole problem, which makes it impossible to calculate a unique solution unless additional measurements are taken.

Additionally, minimization approaches are often taken on a cost function if there is sensor error, because it may not be possible to find compatible pairs that reproduce the exact value of X. The cost falls in the following form

$$C(X) = \sum_{i=1}^{n} w_i \, d^2(A_i X, X B_i) \tag{4.3}$$

where $n > 2$, $d(\cdot, \cdot)$ is some distance metric on $SE(3)$ and $\{w_i\}$ is a set of weights which can be taken as a partition of unity.

4.2 Existing $AX = XB$ Solvers

The problem of solving Eq. (4.1) for X when multiple corresponding pairs of As and Bs are present has a history that stretches back more than a quarter of a century [56, 67, 68], with the earliest proposed method introduced by Shiu [54, 67] and Tsai [7]. Applications involving this problem still remain active today [14, 69]. Shah [70] overviewed several different $AX = XB$ methods qualitatively. Fassi and Legnani [55] gave a geometric view of $AX = XB$ and discussed the over-constrained and singular conditions. A more complete list of the traditional $AX = XB$ solvers includes: the Shiu method [67], the screw motion method [61, 66], the Euclidean group method [56, 71], the quaternion method [57, 58, 68], the dual quaternion method [59, 72], and the Kronecker method [60, 73]. Several new optimization methods emerged recently such as the convex optimization method [62], some global optimization methods [22, 73, 74] and a structure from motion (SfM) approach [75]. The SfM method deals with the case where a calibration target is not applicable and a scaling factor needs to be calibrated in addition to rotation and translation. The methods mentioned previously are all off-line methods where X can be calculated given a complete list of data pairs. However, several online methods have been proposed that are more preferable to real-time applications [60,63,76]. All the methods above assume $\{A_i, B_i\}$ data pairs with correspondence. Probabilistic methods that deal with data pairs without correspondence are introduced in [64, 65]. In the following sections, representatives of the above $AX = XB$ solvers will be reviewed in detail.

4.2.1 Shiu and Ahmad

Shiu and Ahmad [54, 67] use two data pairs (A_i, B_i) to solve for X. The necessary condition for the uniqueness of X is that the rotation axes of R_{A_1} and R_{A_2} are neither parallel nor anti-parallel, and the angles of rotation are neither 0 nor π. Though this method shows tolerance to noise to a certain extent, it is specifically designed to solve for the case where only two sets of (A, B) are given. The rotation matrix R_X is solved for first and the translation is obtained using a least squares technique given a known R_X.

The closed form expression for R_X is:

$$R_X = \exp(\hat{\mathbf{n}}_A \beta) R_{X_P} \tag{4.4}$$

where

$$R_{X_p} = \exp(\widehat{\mathbf{n}}_X \theta_X)$$

$$\mathbf{n}_X = \mathbf{n}_B \times \mathbf{n}_A$$

$$\theta_X = \operatorname{atan2}(|\mathbf{n}_\mathrm{B} \times \mathbf{n}_\mathrm{A}|, \mathbf{n}_\mathrm{B} \cdot \mathbf{n}_\mathrm{A})$$

and β is an arbitrary angle. Given a vector $\mathbf{n} = (n_1, n_2, n_3)^T \in \mathbb{R}^3$, $\widehat{\mathbf{n}}$ is a skew-symmetric matrix defined as below:

$$\widehat{\mathbf{n}} = \begin{pmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{pmatrix}. \tag{4.5}$$

Equation (4.4) shows that R_x has one degree of freedom which is determined by the angle β. Therefore, two relative arm motions are needed to generate two (A_i, B_i) data pairs in order to calculate the unique solution of X. Given two pairs of As and Bs, two equations can be obtained as:

$$A_1 X = X B_1 \tag{4.6a}$$

$$A_2 X = X B_2. \tag{4.6b}$$

Instead of giving a closed-form solution, R_X is calculated by solving for β in Eq. (4.7), which is formulated as a system of linear equations obtained by equating two instances of Eq. (4.4) obtained by substituting in (A_1, B_1) and (A_2, B_2):

$$CY = D. \tag{4.7}$$

In Eq. (4.7), $Y = \begin{pmatrix} \cos(\beta_1), & \sin(\beta_1), & \cos(\beta_2), & \sin(\beta_2) \end{pmatrix}^T$ where β_1, β_2 correspond to Eq. (4.6a) and Eq. (4.6b) respectively, $C \in \mathbb{R}^{9\times 4}$ and $D \in \mathbb{R}^{9\times 1}$ are determined from $\mathbf{n}_{A_1}$, $\mathbf{n}_{B_1}$, $\mathbf{n}_{A_2}$, and $\mathbf{n}_{B_2}$. The explicit expressions for C and D are given in Eq. (44) of [67].

Similarly, with known R_X, $\mathbf{t}_x$ can be calculated using the least squares method:

$$\begin{pmatrix} R_{A_1} - \mathbb{I}_3 \\ \\ R_{A_2} - \mathbb{I}_3 \end{pmatrix} \mathbf{t}_X = \begin{pmatrix} R_X \mathbf{t}_{B_1} - \mathbf{t}_{A_1} \\ \\ R_X \mathbf{t}_{B_2} - \mathbf{t}_{A_2} \end{pmatrix}. \tag{4.8}$$

This is not a closed-form solution and it is constrained to the case where only two data pairs are provided.

4.2.2 Lie Group Method

The Lie Group method [56] by Park and Martin is the first method to solve the $AX = XB$ problem from the perspective of Lie groups. It uses the axes of rotation of A_i and B_i to construct R_X and gives both the closed-form solution for the no-noise case and the numerical solution for multiple noisy (A_i, B_i) pairs.

4.2.2.1 Closed-Form Solution with Two Exact Pairs

The closed-form solution for R_X is as follows:

$$R_X = \mathcal{A}\mathcal{B}^{-1} \tag{4.9}$$

where

$$\mathcal{A} = (\mathbf{n}_{A_1}, \mathbf{n}_{A_2}, \mathbf{n}_{A_1} \times \mathbf{n}_{A_2}) \in \mathbb{R}^{3\times 3}$$

$$\mathcal{B} = (\mathbf{n}_{B_1}, \mathbf{n}_{B_2}, \mathbf{n}_{B_1} \times \mathbf{n}_{B_2}) \in \mathbb{R}^{3\times 3}$$

$$\widehat{\mathbf{n}}_{A_1} = \log(R_{A_i})/\| \log^{\vee}(R_{A_i})\|$$

$$\widehat{\mathbf{n}}_{B_1} = \log(R_{B_i})/\| \log^{\vee}(R_{B_i})\|$$

The solution for R_X is uniquely determined given two pairs of (A_i, B_i), and the solution for $\mathbf{t}_X$ can be obtained using Eq. (4.8) once R_X is obtained.

4.2.2.2 Estimation of X Using Multiple Pairs with Noise

When there are multiple pairs of (A_i, B_i) with noise, rotation matrix R_X is solved for first and then the translation vector $\mathbf{t}_x$ is obtained using a least squares method given known R_X. The closed-form expression for R_X is as follows:

$$R_X = (M^T M)^{-\frac{1}{2}} M^T \tag{4.10}$$

where

$$M = \textstyle\sum_{i=1}^{n} \mathbf{n}_{B_i} \mathbf{n}_{A_i}^T.$$

Note that $i \geq 3$ is a necessary condition for M to be a non-singular matrix, but it does not guarantee M to be nonsingular. Theoretically, the Lie group method is more likely to fail when the number of data pairs is small, i.e., close to 3 pairs, while in real application, failure is rarely seen as long as the data pairs are not specially chosen such that M is degenerate. Given known R_X, $\mathbf{t}_X$ can be calculated using the least squares method shown in Eq. (4.11):

$$\mathbf{t}_x = (C^T C)^{-1} C^T d \tag{4.11}$$

where

$$C = \begin{pmatrix} \mathbb{I}_3 - R_{A_1} \\ \mathbb{I}_3 - R_{A_2} \\ . \\ . \\ . \\ \mathbb{I}_3 - R_{A_n} \end{pmatrix} \quad d = \begin{pmatrix} \mathbf{t}_{A_1} - R_X \mathbf{t}_{B_1} \\ \mathbf{t}_{A_2} - R_X \mathbf{t}_{B_2} \\ . \\ . \\ . \\ \mathbf{t}_{A_n} - R_X \mathbf{t}_{B_n} \end{pmatrix}.$$

4.2.3 Quaternion Method

The Quaternion method proposed by Chou and Kamel [57, 68] uses unit quaternions to transform the rotation parts of $A_i X = X B_i$ (i = 1,2) into two linear systems. Then a singular value decomposition (SVD) is performed to obtain a closed-form solution for R_X. In order to estimate X given multiple pairs of (A_i, B_i) with noise, Horaud and Dornaika [58] cast the problem into a nonlinear optimization one. Two different approaches are discussed: (1) estimate the rotation matrix R_X by minimizing an objective function, and solve for the translation $\mathbf{t}_X$ using a least squares method separately and (2) estimate R_X and $\mathbf{t}_X$ simultaneously by minimizing an objective function that incorporates both the rotational and translational information. Method (1) turns out to have a closed-form solution for the unit quaternion q_X representing the rotation R_X, while method (2) is a nonlinear optimization problem which requires an initial guess and will be discussed in Section 4.2.6.

4.2.3.1 Closed Form Solution with Two Exact Pairs

First, the rotation equation in Eq. (4.2) is transformed into the equation of quaternion multiplication as below:

$$R_A R_X = R_X R_B \quad \Leftrightarrow \quad q_A \odot q_X = q_X \odot q_B \tag{4.12}$$

where q_A, q_B and q_X are unit quaternions that represent the rotation parts of matrices A, B and X, and $\odot$ denotes quaternion multiplication.

Given two quaternions $q_\alpha = \left(\alpha_0, \alpha^{\mathbf{T}}\right)^T = (\alpha_0, \alpha_1, \alpha_2, \alpha_3)^T$ and $q_\beta = \left(\beta_0, \beta^{\mathbf{T}}\right)^T =$

$(\beta_0, \beta_1, \beta_2, \beta_3)^T$, quaternion multiplication $\odot$ is defined as:

$$q_\alpha \odot q_\beta = \begin{pmatrix} \alpha_0\beta_0 - \alpha^{\mathbf{T}}\beta \\ \alpha_0\beta + \beta_{\mathbf{0}}\alpha + \tilde{\alpha}\beta \end{pmatrix}. \tag{4.13}$$

α_0 is called the scalar component and α is the vector component of the quaternion q_α. In order to solve for q_X, the quaternion equation is transformed into:

$$E\mathbf{q}_X = \mathbf{0} \tag{4.14}$$

where $E \in \mathbb{R}^{4\times4}$ is obtained by grouping q_A and q_B together, and $\mathbf{q}_X \in \mathbb{R}^4$ is the vector representation of the unit quaternion q_X.

It turns out that the unit quaternion q_X which represents the rotation part of X can be written as:

$$\mathbf{q}_X = V_2\mathbf{y_2}. \tag{4.15}$$

To obtain the matrix V_2 and vector y_2, E is first written as $E = \sin(\theta_{A|B}/2)E_0$ with $\theta_{A|B} = \theta_A = \theta_B$, which is the constraint that the corresponding transformations A_i and B_i should have the same angle of rotation. Next, the SVD of M is computed as $E_0 = U\Sigma V^T$ where $V = (V_1, V_2)$, $V_1 \in \mathbb{R}^{4\times2}$, $V_2 \in \mathbb{R}^{4\times2}$, $U \in \mathbb{R}^{4\times4}$ and Σ is a diagonal matrix. Vector $\mathbf{y_2}$ is obtained by calculating $\mathbf{y} = V^T\mathbf{q}_x$ where $\mathbf{y} = (\mathbf{y_1^T}, \mathbf{y_2^T})^{\mathbf{T}}$, $\mathbf{y_1} \in \mathbb{R}^{2\times1}$ and $\mathbf{y_2} \in \mathbb{R}^{2\times1}$. Their expressions are also outlined as follows:

$$\begin{aligned}
E &= \sin(\theta_{A|B}/2)E_0 \\
M &= U\Sigma V^T \\
V &= (V_1, V_2) \\
\mathbf{y} &= V^T\mathbf{q}_x \\
\mathbf{y} &= (\mathbf{y_1^T}, \mathbf{y_2^T})^{\mathbf{T}}.
\end{aligned}$$

The translation vector $\mathbf{t}_x$ satisfies the following equation:

$$\left(\cot(\frac{\theta_A}{2})\hat{\mathbf{n}}_A(R_A - \mathbb{I}_3) + R_A + \mathbb{I}_3\right)\mathbf{t}_x = \mathbf{n}_A z \tag{4.16}$$

where $z \in \mathbb{R}$ is arbitrary. A unique solution can be calculated using Eq. (4.15) and Eq. (4.16) given two nondegenerate pairs of (A_i, B_i).

4.2.3.2 Estimation of X Using Multiple Pairs With Noise

As shown in [58], in the case where there are n pairs of (A_i, B_i), the problem of recovering R_X is converted into minimizing the following error objective function:

$$f(R_X) = \textstyle\sum_{i=1}^{n} ||n_{A_i} - q_X \odot n_{Bi} \odot \bar{q}_X||^2 \tag{4.17}$$

$$= \mathbf{q}_X{}^T\tilde{K}\mathbf{q}_X$$

where $n_{A_i} = \left(0, {\mathbf{n}_{A_i}}^T\right)^T$ and $n_{B_i} = \left(0, {\mathbf{n}_{B_i}}^T\right)^T$. $\tilde{K} = \sum_{i=1}^{n} \tilde{K}_i$ and $\tilde{K}_i \in \mathbb{R}^{4\times4}$ is a symmetric positive definite matrix determined by $\mathbf{n}_{A_i}$ and $\mathbf{n}_{B_i}$; $\bar{q}_X$ is the conjugate of q_X where $q_X \odot \bar{q}_X = 1$.

To minimize Eq. (4.17) under the constraint that $\mathbf{q}_X$ is a unit quaternion, the Lagrangian multiplier is introduced as:

$$\min_q f = \min_q ({\mathbf{q}_X}^T \tilde{K} \mathbf{q}_X + \lambda(1 - {\mathbf{q}_X}^T \mathbf{q}_X)). \tag{4.18}$$

Differentiating the error function with respect to $\mathbf{q}_X$, the 1st order necessary optimality condition is obtained as:

$$\tilde{K}\mathbf{q}_X = \lambda \mathbf{q}_X. \tag{4.19}$$

It can be shown that the unit quaternion $\mathbf{q}_X$ that minimizes f is the eigenvector of $\tilde{K}$ associated with its smallest positive eigenvalue. After recovering $\mathbf{q}_X$ (or equivalently R_X), $\mathbf{t}_X$ can be recovered using the least square techniques introduced in previous methods.

4.2.4 Dual Quaternion Method

The dual quaternion method proposed by Daniilidis and Bayro-Corrochano [72] (Daniilidis [59]) treats the rotation and translation parts of the matrix X in a unified way and facilitates a new simultaneous solution of X using SVD. To begin with, Eq. (4.1) is transformed into an equation of dual quaternions:

$$AX = XB \quad \Leftrightarrow \quad \check{a} = \check{q}_X \hat{\odot} \check{b} \hat{\odot} \bar{\check{q}}_X \tag{4.20}$$

where $\check{a}$, $\check{b}$ and $\check{q}$ are the dual quaternions that represent matrices A, B and X, and $\bar{\check{q}}$ is the conjugate of $\check{q}$.

The dual quaternion that corresponds to a 4×4 rigid transformation matrix is defined as follows:

$$\check{q}_X = \begin{pmatrix} \cos(\frac{\theta+\epsilon d}{2}) \\ \\ \sin(\frac{\theta+\epsilon d}{2})(\mathbf{l} + \epsilon \mathbf{m}) \end{pmatrix} \tag{4.21}$$

where θ, d, $\mathbf{l}$ and $\mathbf{m}$ are screw parameters and $\epsilon^2 = 0$. θ is the rotation angle, d is the pitch, $\vec{\mathbf{l}}$ is the direction of the screw, and $\mathbf{m} = \mathbf{p} \times \mathbf{l}$ is the line moment where $\mathbf{p}$ is a point on the line. The six tuple $(\mathbf{l}, \mathbf{m})$ defines a line in 3-D space. Furthermore, by expanding the dual terms in $\check{q}_X$, Eq. (4.21) can also be written as:

$$\check{q}_X = q_X + \epsilon q'_X. \tag{4.22}$$

Both q and q' are quaternions satisfying the following constraints:

$$\mathbf{q}_X^T \mathbf{q}_X = 1 \;\text{ and }\; \mathbf{q}_X^T \mathbf{q}'_X = 0 \tag{4.23}$$

where $\mathbf{q}_X$ and $\mathbf{q}'_X$ are the vector representations of q_X and q'_X. Then equation $A_iX = XB_i$ can be converted into the form below:

$$S_i \underbrace{\begin{pmatrix} \mathbf{q}_X \\ \mathbf{q}'_X \end{pmatrix}}_{\mathbf{x}} = \mathbf{0} \tag{4.24}$$

where

$$S_i = \begin{pmatrix} \mathbf{a}-\mathbf{b} & (\mathbf{a}+\mathbf{b})^\wedge & \mathbf{0_{3\times1}} & \mathbf{0_{3\times3}} \\ \mathbf{a}'-\mathbf{b}' & (\mathbf{a}'+\mathbf{b}')^\wedge & \mathbf{a}-\mathbf{b} & (\mathbf{a}+\mathbf{b})^\wedge \end{pmatrix} \in \mathbb{R}^{6\times8}. \tag{4.25}$$

The notation for x here will be also used in Section 4.2.5. To maintain the consistency of notation throughout the chapter as well as preserve the original notation in [72], for a vector $\mathrm{v} \in \mathbb{R}^{3\times1}$, $\mathrm{v}^\wedge$ is the same as $\tilde{\mathrm{v}}$ which maps a vector into the corresponding skew-symmetric matrix. $\mathrm{a}' = \frac{1}{2}\mathbf{t}_X \times \mathrm{a}$ and a is the vector part of q_X. Similarly, b is the vector part of q'. After concatenating S_i, the following matrix T can be obtained and used to solve for X:

$$T = \begin{pmatrix} S_1^T \, S_2^T \; ... \; S_n^T \end{pmatrix}^T. \tag{4.26}$$

By calculating the SVD of $T = U\Sigma V^T$, the dual quaternion for matrix X can be expressed as a linear combination of the last two right-singular vectors (v_7, v_8)

of matrix T, which are the last two columns of matrix V, as shown below:

$$\begin{pmatrix} \mathbf{q}_X \\ \mathbf{q}'_X \end{pmatrix} = \lambda_1 \mathbf{v_7} + \lambda_2 \mathbf{v_8} \in \mathbb{R}^8 \text{ where } \lambda_1, \lambda_2 \in \mathbb{R}. \tag{4.27}$$

Different from the quaternion method in 4.2.3.1, the dual quaternion method solves the rotational part and translational part in a united way, and it contains all the information to reconstruct matrix X. However, it does not use all of the available information, only the imaginary parts of $\check{a}$ and $\check{b}$. Despite the advantages of the dual quaternion method, its major drawback is the need to filter the data pairs to ensure appropriate solutions when there is noise on A_i and B_i.

4.2.5 Kronecker Product Method

Inspired by the well known Sylvester equation ($AX + XB = C$) in linear systems, Andreff *et al.* [60] proposed the Kronecker method which converts Eq. (4.1) into the form of Kronecker products [60]:

$$AX = XB \Leftrightarrow$$
$$\underbrace{\begin{pmatrix} \mathbb{I}_9 - R_B \otimes R_A & 0_{9\times 3} \\ \mathbf{t}_B^T \otimes \mathbb{I}_3 & \mathbb{I}_3 - R_A \end{pmatrix}}_{C} \underbrace{\begin{pmatrix} vec(R_X) \\ \mathbf{t}_X \end{pmatrix}}_{\mathbf{x}} = \underbrace{\begin{pmatrix} 0_9 \\ \mathbf{t}_A \end{pmatrix}}_{d}. \tag{4.28}$$

where $C, d,$ x will be referred to in Section 4.2.6. Given multiple pairs of As and Bs with noise, the Kronecker product is reformulated as:

$$\begin{pmatrix} \mathbb{I}_9 - R_{B_1} \otimes R_{A_1} \\ \mathbb{I}_9 - R_{B_2} \otimes R_{A_2} \\ \vdots \\ I_9 - R_{B_n} \otimes R_{A_n} \end{pmatrix} vec(R_X) = \mathbf{0}_{9n\times 1}, \tag{4.29}$$

and

$$\begin{pmatrix} \mathbb{I}_3 - R_{A_1} \\ \mathbb{I}_3 - R_{A_2} \\ \vdots \\ \mathbb{I}_3 - R_{A_n} \end{pmatrix} \mathbf{t}_X = \begin{pmatrix} \mathbf{t}_{A_1} - R_X \mathbf{t}_{B_1} \\ \mathbf{t}_{A_2} - R_X \mathbf{t}_{B_2} \\ \vdots \\ \mathbf{t}_{A_n} - R_X \mathbf{t}_{B_n} \end{pmatrix}. \tag{4.30}$$

The vectorized version of R_X obtained from Eq. (4.29) is not an element of $SO(3)$ and orthogonalization on R_X is required to obtain a rotation matrix [77] :

$$R_{X_e} = R_X(R_X^T R_X)^{-1/2} \tag{4.31}$$

where R_{X_e} denotes the orthogonalized R_X.

The orthogonalized matrix R_{X_e} is further normalized as:

$$R_{X_n} = \frac{sign(\det(R_{X_e}))}{|\det(R_{X_e})|^{\frac{1}{3}}} R_{X_e} \tag{4.32}$$

where R_{X_n} is the normalized matrix of R_{X_e}. After getting the estimation of R_X, a least squares method is implemented on Eq. (4.30) to recover t_X. One advantage of the Kronecker product method is its capability of dealing with small motions, because the associated orthogonal matrix R_{X_n} is always defined, while the rotation can be ill-defined when using the axis-angle representation. Moreover, the linear system can also be used to analyze the recoverable information in X based on the available type and number of motions. Details are included in Section 4.3. In addition, an on-line hand-eye calibration method is developed for an unknown scene based on the above algorithm, where the camera translations are estimated up to a scaling factor.

4.2.6 Optimization Methods

Different optimization methods have been proposed in the literature and most of them are built upon the various parameterizations of the $AX = XB$ equations mentioned previously.

4.2.6.1 Quaternion Based Simultaneous Approach

When trying to solve for R_X and t_X simultaneously, it is impossible to find a closed-form solution. Horaud and Dornaika [58] presented an objective function for minimization, which is a sum of squares of nonlinear functions as:

$$
\begin{aligned}
f(q_X, \mathbf{t}_X) = \lambda_1 \sum_{i=1}^{n} ||vec(q_X \odot t_{B_i} \odot \bar{q}_X) - (R_{A_i} - \mathbb{I})\mathbf{t}_X - \mathbf{t}_{A_i}||^2 \\
+\lambda_2 \sum_{i=1}^{n} ||n_{A_i} - q_X \odot n_{B_i} \odot \bar{q}_X||^2 + \lambda_3(1 - \mathbf{q}_X^T \mathbf{q}_X)^2
\end{aligned}
\tag{4.33}
$$

where $\lambda_1, \lambda_2, \lambda_3 \in \mathbb{R}$. Note that $vec()$ here represents the vector part of a quaternion such that $vec(q) \in \mathbb{R}^3$. The third term is a penalty function where the modulus of q will approach 1 when λ_3 becomes large. This is a non-convex optimization problem which requires a good initial guess due to the existence of multiple local minima. However, the result can be more accurate than the rest of the solvers for certain motion pairs when the initial guess is “good”.

4.2.6.2 Polynomial Global Optimization

Heller *et al.* [22] developed a polynomial global optimization method which does not require an initial estimate and is also globally optimal in the L_2-norm sense. Defining a certain parameterization of $X \in SE(3)$ as $P(X)$, the previous minimization problem Eq. (4.3) is formulated as:

$$
\begin{aligned}
&\min_{X \in SE(3)} \sum_{i=1}^{n} ||A_i X - X B_i||^2 \Leftrightarrow \\
&\text{minimize} \quad f(P(X)) \\
&\text{subject to} \;\; \mathbf{c}(P(X)) \geq \mathbf{0}
\end{aligned}
\tag{4.34}
$$

where $f(X)$ is the converted multivariate polynomial function using the convex linear matrix inequality (LMI) relaxations technique [78].

When R_X is parameterized using the orthonormal basis as $R_X(\mathbf{u},\mathbf{v}) = (\mathbf{u},\mathbf{v},\mathbf{u}\times\mathbf{v})$ where $\mathbf{v},\mathbf{u} \in \mathbb{R}^3$, then Eq. (4.34) becomes:

$$\begin{aligned} \text{minimize} \quad & f_1(\mathbf{u},\mathbf{v},\mathbf{t}_X) = \\ & \sum_{i=1}^{n} ||A_i X(\mathbf{u},\mathbf{v},\mathbf{t}_X) - X(\mathbf{u},\mathbf{v},\mathbf{t}_X)B_i||^2 \\ \text{subject to} \quad & \mathbf{u}^T\mathbf{u} = 1, \mathbf{v}^T\mathbf{v} = 1, \mathbf{u}^T\mathbf{v} = 0. \end{aligned} \tag{4.35}$$

Similarly, using the quaternion representation of R_X, Eq. (4.34) becomes:

$$\begin{aligned} \text{minimize} \quad & f_2(\mathbf{q}_X,\mathbf{t}_X) = \\ & \sum_{i=1}^{n} ||A_i X(\mathbf{q}_X,\mathbf{t}_X) - X(\mathbf{q}_X,\mathbf{t}_X)B_i||^2 \\ \text{subject to} \quad & \mathbf{q}_X^T\mathbf{q}_X = 1, \;\; q_{X1} \geq 0. \end{aligned} \tag{4.36}$$

If A, B and X are parameterized using dual the quaternion representation, then

$$\begin{aligned} \text{minimize} \quad & f_3(\check{q}_X) = \\ & \sum_{i=1}^{n} ||\check{q}_A \hat{\odot} \check{q}_X - \check{q}_X \hat{\odot} \check{q}_B||^2 \\ \text{subject to} \quad & \mathbf{q}_X^T\mathbf{q}_X = 1, \;\; q_{X1} \geq 0 \\ & q_{X1}q_{X5} + q_{X2}q_{X6} + q_{X3}q_{X7} + q_{X4}q_{X8} = 0. \end{aligned} \tag{4.37}$$

As pointed out in [22], the polynomial global optimization method as described in Eq. (4.37) can give better solutions than Park [56], Eq. (4.33), Eq. (4.35) and Eq. (4.36). For those who are also interested in robot-world and hand-eye calibration, several solvers for $AX = YB$ using the LMI technique are also given; however, they fail to give better results than the traditional methods.

4.2.6.3 Convex Optimization

Based on different ways of formulating the rotation part of the rigid body transformation, Zhao [62] gives two formulations that use a L_∞ optimization technique. L_∞ optimization is the minimax problem:

$$\min_{\mathbf{x}} \ \max_{i} f_i(\mathrm{x}) \ \ \mathbf{i = 1, 2, ..., n} \tag{4.38}$$

where x represents all the unknown transformation parameters and $f_i(\mathrm{x})$ is the error function corresponding to (A_i, B_i). Eq. (4.38) can be converted into a convex optimization problem if $f_i(\mathrm{x})$ is a quasi-convex function on a convex domain on which it is to be minimized.

Using the Kronecker formulation as in Eq. (4.28) and introducing an additional variable δ, the equivalent form of the L_∞ optimization problem is:

$$\begin{aligned}&\min_{\delta,\mathbf{x}} \;\; \delta \\ &\text{subject to} \;\; ||C_i\mathbf{x} - \mathbf{d}_i||_2 \leq \delta \\ &\text{where} \;\; i = 1, 2, ..., n.\end{aligned} \tag{4.39}$$

The matrix $\mathbf{C}_i$, vector x, and $\mathbf{d}_i$ correspond to those in Eq. (4.28). Above is a convex optimization problem that can be solved using a *second-order cone program*, which can be solved using toolboxes available online.

When the $AX = XB$ problem is formulated as a dual quaternion representation as in Eq. (4.24), the equivalent L_∞ optimization problem can be written as:

$$\begin{aligned}&\min_{\delta,\mathbf{x}} \;\; \delta \\ &\text{subject to} \;\; ||S_i\mathbf{x}||_2 \leq \delta \\ &\text{where} \;\; i = 1, 2, ..., n \;\; \text{with} \;\; \mathbf{Dx} \geq \mathbf{f}.\end{aligned} \tag{4.40}$$

The matrix S_i and vector x correspond to those in Eq. (4.24). The inequality constraint Dx $\geq$ f is added manually in order to prevent x from reaching zero, which is a meaningless solution for the program. x must also satisfy two additional constraints: $\|\mathrm{q}_X\| = 1$ and $\|\mathrm{q}'_X\| = 1$.

The proposed methods need no initial guess and are less time consuming compared to the simultaneous optimization method given in Eq. (4.33). However, the errors for both of the convex optimization methods are larger than the latter.

4.2.7 Gradient Descent Method

Except for the Kronecker product method, all the methods mentioned above are only able to solve for matrix X offline, which means $\{A_i, B_i\}$ data pairs should be fully collected before being put into the algorithm. The gradient descent method [76] by Ackerman *et al.* is an online sensor calibration method which uses a gradient descent optimization on the Euclidean group (SE(3)) given an appropriate cost function.

To begin, define $\mathcal{X} \in se(3)$ as the Lie algebra corresponding to $G = SE(3)$, and let $f : G \to \mathbb{R}$ be an analytic function and $g \in G$. As defined in [26], the concept of directional derivatives in $\mathbb{R}^n$ is extended to functions on a Lie group as:

$$(\hat{\mathcal{X}}^r f)(g) \doteq \frac{d}{dt} f(g \circ \exp(t\mathcal{X}))\bigg|_{t=0}. \tag{4.41}$$

Note that t is just a scalar denoting the time, while t represents the translation part of a homogeneous transformation. Eq. (4.41) is the "right" Lie derivative and the "left" Lie derivative can be defined in a similar form. A gradient on $SE(3)$ is then defined using the right Lie derivative with the "natural" basis of

Lie algebra $\{E_i\}$ where $i = 1, 2, ..., 6$. Therefore, the gradient of the function on Lie group $f(g)$ is as follows:

$$\nabla f(g) = \begin{pmatrix} \frac{d}{dt} f(g \circ \exp(tE_1))|_{t=0} \\ \frac{d}{dt} f(g \circ \exp(tE_2))|_{t=0} \\ \vdots \\ \frac{d}{dt} f(g \circ \exp(tE_6))|_{t=0} \end{pmatrix}. \tag{4.42}$$

In order to update g, the rigid body velocity is introduced where $V_g^r = g^{-1}\dot{g}$, and the update law is written as below:

$$g_{s+1} = g_s \exp(\Delta t V_g^r) \tag{4.43}$$

where $t_{s+1} = t_s + \Delta t$ is the discrete time step corresponding to g_{s+1}. To prevent steps from becoming too small, the update law of g_s is modified by defining V_g^r as:

$$V_g^r = g^{-1}\dot{g} = -\alpha \widehat{\nabla f(g)}. \tag{4.44}$$

where α is a scaling factor. After choosing the cost function as:

$$C(X) = \sum_{i=1}^{n} ||A_i X - X B_i|| \tag{4.45}$$

X can be optimized using Eq. (4.44). The gradient descent method updates the calibrations parameters online based on new incoming data. The initial guess of X will converge to the true X; however, the rate of convergence depends on how "good" the initial guesses are.

4.2.8 Batch Method

This section presents a probabilistic method [64] by Ackerman *et al.* to solve for X in the absence of a priori knowledge of the correspondence between the exact sets of measurements $A = \{A_i\}$ and $B = \{B_j\}$. In other words, the sets A and B each can be given as unordered "batches" without knowing how each A_i matches to a specific B_j.

A commonality of all the methods in the previous sections is that exact knowledge of the correspondence between $\{A_i\}$ and $\{B_j\}$ is assumed; however, this is not always the case. There are many instances in the literature when the sensor data used in calibration becomes "unsynchronized". Different attempts have been implemented to solve this problem, such as time stamping the data, developing dedicated software modules for syncing the data, and analyzing components of the sensor data stream to determine a correlation [8], to varying effects. The Batch method bypasses these issues without recomputing, correspondence. By modeling the set of A's and B's as probability distributions on $SE(3)$, the data can be taken as an unordered, uncorrelated "batch" and a solution for X can be generated.

In this section, part of the mathematical framework for Batch method is introduced, some of which will be revised and used for other calibration solvers in later chapters. In the Batch method, two key equations, or the equivalent probabilistic equations, are obtained for solving X. The first equation describes the relationship among the means of $\{A_i\}$, $\{B_j\}$ and X. The second equation reflects the relationship among the covariance matrices of $\{A_i\}$ & $\{B_j\}$ and X.

4.2.8.1 Mathematical Framework

Given a set of data pairs $(A_i, B_i) \in SE(3) \times SE(3)$ with correspondence, the following is true:

$$A_i X = X B_i, \tag{4.46}$$

where $i = 1, 2, ..., n$. If we use the probability theory on $SE(3)$, Eq. (4.46) can be converted into:

$$(\delta_{A_i} * \delta_X)(H) = (\delta_X * \delta_{B_i})(H). \tag{4.47}$$

Note that convolution is a linear operation on functions, therefore n instances of Eq.(4.47) can be added into a single equation as:

$$(f_A * \delta_X)(H) = (\delta_X * f_B)(H), \tag{4.48}$$

where

$$f_A(H) = \frac{1}{n}\sum_{i=1}^{n}\delta(A_i^{-1}H) \text{ and } f_B(H) = \frac{1}{n}\sum_{i=1}^{n}\delta(B_i^{-1}H). \tag{4.49}$$

The above functions can be normalized to be probability densities:

$$\int_{SE(3)} f_A(H)dH = \int_{SE(3)} f_B(H)dH = 1. \tag{4.50}$$

If we let the mean $M \in SE(3)$ and covariance $\Sigma \in \mathbb{R}^{6\times 6}$ of a probability density $f(H)$ satisfy:

$$\int_{SE(3)} \log(M^{-1}H)f(H)dH = \mathbb{O} \tag{4.51a}$$

$$\Sigma = \int_{SE(3)} \log^{\vee}(M^{-1}H)[\log^{\vee}(M^{-1}H)]^T f(H)dH. \tag{4.51b}$$

where $\mathbb{O}$ is a 6 by 6 matrix with all elements equal to zero. Then for a PDF $f_A(H)$ as given in Eq.(4.49), the discrete version of the mean M_A and covariance Σ_A will be:

$$\sum_{i=1}^{n} \log(M_A^{-1}A_i) = \mathbb{O} \tag{4.52a}$$

$$\Sigma_A = \frac{1}{n}\sum_{i=1}^{n} \log^{\vee}(M_A^{-1}A_i)[\log^{\vee}(M_A^{-1}A_i)]^T. \tag{4.52b}$$

In [29], f_1 and f_2 are assumed to be highly focused functions so that the mean and covariance for the convolution can be closely approximated as:

$$M_{1*2} = M_1 \, M_2 \tag{4.53a}$$

$$\Sigma_{1*2} = Ad(M_2^{-1}) \, \Sigma_1 \, Ad^T(M_2^{-1}) + \Sigma_2, \tag{4.53b}$$

where

$$Ad(H) = \begin{pmatrix} R & \mathbb{O} \\ \widehat{\mathbf{x}}R & R \end{pmatrix}. \tag{4.54}$$

Employing the fact that the mean of $\delta_X(H)$ is $M_X = X$ and the covariance $\Sigma_X = \mathbb{O}_{6\times 6}$ is a zero matrix, Eq. (4.53a) and Eq. (4.53b) give two key equations in the Batch method as:

$$M_A \, X = X \, M_B \tag{4.55}$$

and

$$Ad(X^{-1}) \, \Sigma_A \, Ad^T(X^{-1}) = \Sigma_B. \tag{4.56}$$

[65] notes that there are two degrees of freedom in Eq. (4.55), so Σ_A and Σ_B are decomposed to provide the required constraints:

$$\Sigma_i = \begin{pmatrix} \Sigma_i^1 & \Sigma_i^2 \\ \Sigma_i^3 & \Sigma_i^4 \end{pmatrix}, \tag{4.57}$$

where $\Sigma_i^3 = (\Sigma_i^2)^T$ and the subscript i can be either A or B. We can extract the first two blocks of Eq. (4.56) and get:

$$\Sigma^1_{M_B} = R_X^T \Sigma^1_{M_A} R_X \tag{4.58a}$$

$$\Sigma^2_{M_B} = R_X^T \Sigma^1_{M_A} R_X (\widehat{R_X^T t_x}) + R_X^T \Sigma^2_{M_A} R_X. \tag{4.58b}$$

As described in [65], a unique R_X can be obtained by calculating the eigendecomposition of $\Sigma^1_{M_A}$ and $\Sigma^1_{M_B}$ as $\Sigma_i = Q_i \Lambda Q_i^T$, where Q_i is the square matrix whose ith column is the eigenvector of Σ_i and Λ is the diagonal matrix with corresponding eigenvalues as diagonal entries. After further derivations, the mean of the rotation component of X can be written as:

$$R_X = Q_{M_A} Q Q^T_{M_B}, \tag{4.59}$$

where Q is a diagonal matrix and there exist four candidates of Q in total. The optimal solution of R_X can be picked by minimizing a cost function that contains the constraint information from Eq. (4.55). The translation component t_X can be uniquely determined using Eq. (4.58b)

4.3 Data Selection and Error Metrics

In this section, a review is given on the existing techniques for data selection for $AX = XB$ problem. When performing hand-eye calibration in experiments, the accuracy of the calibrated X is highly dependent on the data that is obtained in the process. Tsai and Lenz [7] proposed several principles on designing the movement of the robot. This is useful when the motion planning under these constraints is practical. However, data selection has to be considered when such movement is not applicable, such as when there is a lack of free space or a hand-held sensor is used. In this section, several principles and methods for data selection are reviewed. In addition, different error metrics are also discussed to give a more complete picture of the $AX = XB$ problem. Some of the data selection techniques and error metrics will be used in the novel calibration solvers to be presented in later chapters.

4.3.1 Data Selection

Selection of well defined (A_i, B_i) is very important for the $AX = XB$ solvers. Data selection methods for off-line application have been proposed in [79–81], and corresponding selection techniques for on-line solvers are introduced in [82, 83]. For probabilistic methods, data sets $\{A_i\}$ and $\{B_i\}$ must be highly concentrated, which means small and relative motions are preferable [64]. This is opposite to the data selection criterion of other non-probabilistic approaches.

To determine the hand-eye transformation, at least 2 non-parallel rotation axes from the data pairs are needed (which is also referred at the non-parallelism criterion), and further data selection algorithms are all built on top of this.

In the error analysis of [7], four observations are given to show the relationships between the errors in rotation and translation and the features of the robot motions. In addition, seven steps are suggested to improve the calibration accuracy. Shi *et al.* [82] developed a motion selection algorithm based on three out of the four observations in [7]. However, the thresholds in [82] are chosen in a heuristic manner. To fix this problem, Zhang *et al.* [83] proposed an adaptive selection method which can update the thresholds online. All of the above approaches share some common standards such that small relative rotations between (A_i, B_i) and (A_{i+1}, B_{i+1}) should be avoided, and the rotation angles for both $\{A_i\}$ and $\{B_i\}$ should be large enough to avoid the singularity of the representation. The Kronecker product method, however, has a tolerance on small robot motions as pointed out in Section 4.2.5. It also offers an algebraic analysis to show what information of X can be obtained using certain types and numbers of motions. Interesting results are: (1) three independent pure translations can fully define R_X but not t_X; (2) with two or more independent pure rotations, both R_X and t_X can be recovered. A more detailed summary can be seen in Table 1 of [60]. Schmidt *et al.* [9] discussed data selection for the dual quaternion hand-eye calibration algorithm based on a RANSAC approach for filtering, which shows that the dual quaternion method can yield a better X after data selection; without filtering the method can either fail or perform worse. However, the computation of all possible relative movements from the data set results in a long computational time.

For a set of continuous robot motions, a sequence of images or sensor recorded information will be obtained. But due to the small differences between consecutive images or sensor recordings, it is often undesirable to process the data in the temporal order because this can yield high errors. Schmidt *et al.* [79] then proposed a vector quantization based data selection technique which selects a globally consistent set of motions that optimizes the non-parallelism criterion. The main idea is to select a subset of the given rotation axes of (A_i, B_i) using clustering algorithms. It is shown that compared to the above two approaches, the algorithm presented is both fast and accurate. Ackerman *et al.* [81] also proposed a method which uses the Euclidean group invariants in the structure of $\{A_i\}$ and $\{B_i\}$ to realign asynchronous data streams.

Take the ultrasound sensor calibration in Fig. (4.1) as an example. The data stream $\{A_i\}$ is calculated from the joint angles of the robot arm, which is very accurate due to the high sensitivity of the encoder on each motor. However, due to the sensing accuracy of 2D US probe, $\{B_i\}$ obtained from the ultrasound sensor might not be as accurate and sometimes one can get B_k that is far from the "correct" measurement. In order to get X that is closer to the ground truth, the above data selection algorithms can be used to detect the mismatched data pair (A_k, B_k) caused by the big measurement noise on B_k.

4.3.2 Calibration Verification

Normally, the verification of a calibration algorithm consists of two parts: synthetic data and real experiments. For the verification with synthetic data, ground truth X_{true} is usually given in advance. The data stream $\{A_i\}$ (or $\{B_i\}$) then is generated according to a predetermined trajectory or distribution on $SE(3)$ and the other stream is obtained by $B_i = X_{true}^{-1} A_i X_{true}$ (or $A_i = X_{true} B_i X_{true}^{-1}$). Till this point, both $\{A_i\}$ and $\{B_i\}$ are noise-free which can be regarded as the exact sensor measurements in an ideal experiment. However, different types and levels of noise exist in real experiments, and noise is usually artificially applied onto either one or both of the data streams. There are a lot of ways to apply noise onto the data stream and one can choose based on the properties of the sensor measurement in the specific experiment setup. At last, noisy data streams are fed into the calibration solvers to compute X_{calc}, which will be compared with X_{true} using error metrics. The verification of calibration solver with synthetic data is relatively consistent in the literature, because one can artificially provide the ground truth X_{true}. However, in the real experiment, it is quite often impossible to get the ground truth of X and the validation process usually uses the calibrated X_{calc} for other tasks, e.g. 3D reconstruction, to evaluate its accuracy. The validation procedure in real experiments varies from platform to platform.

4.3.3 Error Metrics

There are multiple ways to define the errors of rigid body transformations, and some methods rely heavily on the metric that is chosen [84]. One approach is to measure the errors of R_x and $\mathbf{t}_X$ simultaneously which is rarely seen in more recent literature. The other approach is to measure the rotation error and translation error separately.

4.3.3.1 Metrics for Rotation and Translation Errors

Various metrics for rotation error have been used in the literature. In [7] and [58], the matrix error metric is defined as:

$$E_{rot} \doteq \|R_{X_{true}} - R_{X_{calc}}\|. \tag{4.60}$$

However, this is less preferable because rotation matrices lie in $SO(3)$ and the deduction operation is not defined for $SO(3)$. In [59] and [60], the quaternion error metric is used as:

$$E_{rot} \doteq \|\mathbf{q}_{X_{true}} - \mathbf{q}_{X_{calc}}\|. \tag{4.61}$$

As noted in [72], $\|R_{X_{true}} - R_{X_{calc}}\|_F$ is $2\sqrt{2}$ times larger than $\|\mathbf{q}_{X_{true}} - \mathbf{q}_{X_{calc}}\|$ so it is important to maintain a consistent error metric especially for result comparisons. Another rotational error metric is to calculate the norm of the relative rotation between X_{true} and X_{calc} as:

$$E_{rot} \doteq \|R_{X_{true}}^{T} R_{X_{calc}}\|. \tag{4.62}$$

In [64], the Lie algebra error metric is defined for the relative rotation $R_{X_{true}}^{T} R_{X_{calc}}$ as:

$$E_{rot} \doteq \| \log^{\vee}(R_{X_{true}}^{T} R_{X_{calc}})\|. \tag{4.63}$$

This is also the metric for measuring the rotation errors in later chapters.

Metrics for the translation error are relatively simple because translation lies in Euclidean space. A common method is to use the relative translation error to eliminate the influence of the translation unit:

$$E_{tran} \doteq \frac{\|\mathbf{t}_{X_{true}} - \mathbf{t}_{X_{calc}}\|_2}{\|\mathbf{t}_{X_{true}}\|_2}. \tag{4.64}$$

Conventionally, multiple trials are performed at each fixed noise level in numerical simulation, and the “averaged” errors in rotation and translation are defined in [58] as:

$$e_{rot} \doteq \sqrt{\frac{1}{N}\sum_{i=1}^{N} E_{rot}^2}, \tag{4.65}$$

$$e_{tran} \doteq \sqrt{\frac{1}{N}\sum_{i=1}^{N} E_{tran}^2}. \tag{4.66}$$

It should be noted due to the diversity of rotation error metrics, it is yet to be seen which metric is better or whether different metrics make a difference at all.

4.4 Conclusion

In this chapter, the $AX = XB$ formulation of the sensor calibration problem is examined, which is widely used in head-eye calibration. A review of some of the most influential and effective methods was presented and their positive and negative traits were discussed. For the various $AX = XB$ solvers, the focus is put on the case where there is noise on the incoming sensor data, and therefore multiple sensor readings are needed. It was clear that each algorithm has strengths and weaknesses in different contexts, and it is important to use the appropriate method for different circumstances.

In addition to measurement error contributing to noise, it was emphasized that the sensor data streams containing the A's and B's may be present at different sampling rates, may be asynchronous, and/or each stream may contain gaps in information. Therefore, a probabilistic method is reviewed in detail which is used for calculating the calibration transformation that works for data without any a priori knowledge of the correspondence between the A's and B's. Data selection is of critical importance to $AX = XB$ solvers. Depending on the quality of the data pairs, the usage of data selection techniques can either greatly improve the final result or prevent the solver from failing.

Chapter 5

Probabilistic Approaches Towards $AX = XB$ Hand-eye Calibration

Many $AX = XB$ solvers have been reviewed in Chapter 4 in terms of both the traditional methods and probabilistic methods. Similar to the fact that the performance of traditional solvers can be influenced by the trajectory of the robot motions, it is presented in this chapter that the probabilistic approach, namely the Batch method, can perform up to different levels of accuracy given $\{A\}$ and $\{B\}$ samples with different distributions. There are several ways of defining the mean and covariance for a set of $SE(3)$ matrices, and they play an important role for probabilistic approaches in general. In this chapter, two new means are defined for $SE(3)$ based on the $1st$ order and $2nd$ order approximations of Eq. 4.51a. It is shown that instead of being just the approximations of original mean, the new mean definitions can boost the accuracy of the Batch method when solving for the unknown X. The rest of the chapter is organized as follows. In Section 5.1, a brief introduction is given to the $AX = XB$ problem. In Section 5.2, the motivation for deriving the $1st$ and $2nd$ order mean approximations is given. Section 5.3 and Section 5.4 present the detailed mathematical formulations for the $1st$ order mean and $2nd$ order mean respectively. In Section 5.5, simulation experiments are performed among the Kronecker product method, Batch method, and two new Batch methods to show both the effectiveness and the accuracy of the latter. Finally, conclusions are given in Section 5.6.

5.1 Introduction

Fig. (5.1) shows that a UR5 robot uses a phantom to calibrate the relative transformation between the ultrasound probe and the end-effector of the robot. X is the unknown rigid body transformation calculated by solving $AX = XB$. It is well known that for non-probabilistic methods, at least two exact data pairs (A_1, B_1) and (B_2, B_2) are required to recover a unique X. For probabilistic approaches such as the Batch method, no correspondence between A and B

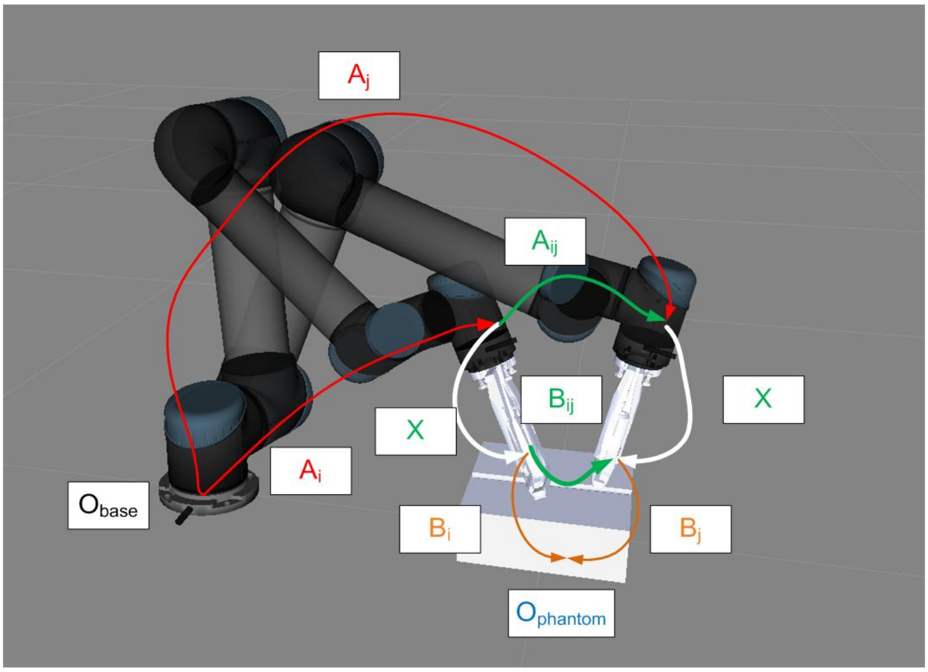


Figure 5.1: Application of the $AX = XB$ problem in the extrinsic calibration of the ultrasound probe with respect to the UR5 robot arm. The white squarish object in the middle represents the ultrasound calibration phantom.

is needed and a comparison between the Batch method and Kronecker product method in [64] shows its extraordinary capability of handling data sets without correspondence. However, as claimed in [64], the Batch method has restrictions on the data set that can be used which limits its scope of application. In addition, the definition of mean on $SE(3)$ plays an important role in the Batch method, and the current definition can't provide the optimal "average" of a set rigid body transformations for the use of $AX = XB$ problem. In this chapter, it is shown that this restriction can be alleviated and by using new definitions, the accuracy of the calibrated X can be improved significantly.

5.2 Motivation

Analogous to non-probabilistic approaches which require data selection to filter out ill-conditioned data pairs, the Batch method has restrictions on the data set $\{A_i\}$ and $\{B_j\}$ that can be used. We propose two new probabilistic approaches built on top of the Batch method by giving new definitions of the mean on $SE(3)$, which alleviate the restrictions on the data set and significantly improve the calibration accuracy of X.

To determine the mean and covariance of the convolution of two highly focused PDFs as in [29], the Baker-Campbell-Hausdorff (BCH) formula is used:

$$\log(e^X e^Y) = \underbrace{X + Y}_{\text{0th order}} + \underbrace{\frac{1}{2}[X, Y]}_{\text{1st order}} + h.o.t. \tag{5.1}$$

where $h.o.t.$ stands for the 2nd and higher order terms. If X and Y are further constrained to be small so that $\|X\| \ll 1$ and $\|Y\| \ll 1$, then the first order approximation of Eq. (5.1) can be used to derive Eq. (4.53a) and Eq. (4.53b). In the $AX = XB$ context, e^X and e^Y represent A_i and B_i respectively. This derivation constrains the distribution function f_A and f_B to be highly focused. Moreover, the current definition of mean on $SE(3)$ as in Eq.(4.52a) can't always reflect the desired M_A and M_B in the probabilistic $AX = XB$ context. It will be shown that two new ways of defining the mean can be used to augment the existing Batch method which can significantly improve the accuracy of X

depending on the distributions of $\{A_i\}$ and $\{B_j\}$. In later sections, we will call the original Batch method as the "Batch method" and the two augmented Batch methods as the "Batch1" and "Batch2" methods.

Then two new definitions of the mean on $SE(3)$ will be given based on the 1st order and 2nd order approximation of of Eq. (4.51a). The new means for $\{A_i\}$ and $\{B_j\}$, and the corresponding covariance matrices Σ_A and Σ_B, can be directly incorporated into the Batch method to form the Batch1 and Batch2 methods.

5.2.1 Conditions for the Key Equations

It is shown in [16] that $\{A_i\}$ and $\{B_j\}$ don't have to highly concentrated in order for the key equations to hold. This result broadens the types of data sets $\{A_i\}$ and $\{B_j\}$ that can be used for the Batch method. However, the mean definition as in Eq.(4.51a) doesn't always provide the desired M_A and M_B in terms of minimizing $C_1(R_X) = ||R_{M_A}R_X - R_XR_{M_B}||$ or $C_2(\mathbf{t}_X) = ||R_{M_A}\mathbf{t}_X + \mathbf{t}_{M_A} - R_{M_X}\mathbf{t}_{M_B} - \mathbf{t}_{M_B}||$, which are simply the error metrics for the rotational and translational components of Eq. (4.55). To find a better definition of mean under the "$AX = XB$" context, we start by assuming $M^{-1}H$ is small such that $||M^{-1}H - \mathbb{I}|| \ll 1$. This is different from the assumption of the Batch method that H is small enough such that $||\log(H)|| \ll 1$. From there, the 1st order and 2nd order approximations of the mean in Eq.(4.51a) can be achieved and new definitions are proposed accordingly. As will be shown in section IV, the new definitions are more than just the approximations of the original mean.

Note that though Eq. (4.51a) and Eq. (4.51b) have a similar form, they are not necessarily bounded together. Equivalently, Eq. (4.51b) will be valid as long as the mean M is given, which does not have to be defined as in Eq. (4.51a). Therefore, it is appropriate to use a different mean while using the same definition of the covariance. One thing to be noted is that the proof of Eq. (4.53b) does not depend on the definition of mean, but only on the 1st order approximation of BCH. Therefore, Eq. (4.56) will still be valid when we use the new definitions of mean. As for Eq. (4.55), the 1st or 2nd order approximations can be made on $\log(XM_B^{-1}X^{-1}K)$ in the previous proof by treating $XM_B^{-1}X^{-1}$ as a whole. The same result can be obtained by following the similar procedure in the definitions as below, which will not be mentioned in detail here.

Batch1 and Batch2 methods simply replace the old means of $\{A_i\}$ and $\{B_j\}$ with new ones, obtain the corresponding covariances in terms of the new means, and use the two key equations to solve for X. For the two different kinds of distributions of $\{A_i\}$ and $\{B_j\}$ that are tested, Batch1 and Batch2 methods are able to recover a much more accurate R_X in a consistent manner, and recover t_X to a certain level of accuracy depending on the type of distribution.

5.3 Mean Based on the 1st Order Approximation

Consider the Taylor expansion of the matrix logarithm described as:

$$\log(\mathbb{I} + X) = X - \frac{1}{2}X^2 + \frac{1}{3}X^3 - ... \tag{5.2}$$

If $\|M^{-1}H - \mathbb{I}\| \ll 1$ and we retain the 1st order approximation of Eq.(5.2), it can be written as:

$$\log(M^{-1}H) = \log(\mathbb{I} + (M^{-1}H - \mathbb{I})) \approx \left(M^{-1}H - \mathbb{I}\right). \tag{5.3}$$

The first order approximation of Eq.(4.51a) is:

$$\int_{SE(3)} (M^{-1}H - \mathbb{I}) f(H) dH \approx \mathbb{O}. \tag{5.4}$$

Note that $f(H)$ is a normalized probability density such that $\int_{SE(3)} f(H)dH = 1$, so:

$$M^{-1} \int_{SE(3)} H f(H) dH \approx \mathbb{I}. \tag{5.5}$$

Define the first order approximation of $M \in \mathbb{R}^{4\times 4}$ as $\widehat{M}$:

$$\widehat{M} \doteq \int_{SE(3)} Hf(H)dH \notin SE(3). \tag{5.6}$$

Take $\{A_i\}$ for an example, the corresponding discrete version of $\widehat{M}$ will be:

$$\widehat{M_A} \doteq \sum_{i=1}^{n} A_i \left(\frac{1}{n} \sum_{j=1}^{n} \delta(A_j^{-1} A_i) \right) = \frac{1}{n} \sum_{i=1}^{n} A_i \notin SE(3). \tag{5.7}$$

Note that though Eq.(5.6) and Eq.(5.7) are obtained based on the 1st order approximation of the matrix logarithm, $\widehat{M}$ is not the 1st order approximation of the mean $M \in SE(3)$ because it is not necessary a group element in $SE(3)$ and elements in $SE(3)$ do not add. Therefore, we define a new mean based on Eq.(5.7) by projecting $\widehat{M}$ onto $SE(3)$, where singular value decomposition is performed on the "rotation component" $\widehat{R}_{\widehat{M_A}}$ of $\widehat{M_A}$:

$$\widehat{R}_{\widehat{M_A}} = U\Sigma V^T. \tag{5.8}$$

The rotation component $R_{M_A} \in SO(3)$ can be obtained according to [85] as:

$$R_{M_A} = UV^T. \tag{5.9}$$

After recovering the rotation component, the new mean based on the 1st order approximation becomes:

$$M_A^1 = \begin{pmatrix} R_{M_A} & \frac{1}{n}\sum_{i=1}^{n}\mathbf{t}_{A_i} \\ \mathbf{0}^T & 1 \end{pmatrix}. \tag{5.10}$$

Using this definition of mean, M_A^1 and M_B^1 can be calculated in a straight way. Σ_A^1 and Σ_B^1 can be obtained by Eq. (4.52b) afterwards. Batch1 method uses the above means and covariances for recovering X, and all the other procedures strictly follow the procedure described as in Eqs. (4.55-4.59).

5.4 Mean Based on the 2nd Order Approximation

In this section, we further define a mean on $SE(3)$ based on the second order approximation of Eq.(4.51a). Under the assumption that $\|M^{-1}H - \mathbb{I}\| \ll 1$ and using the Eq, (5.3), the 2nd order approximation of Eq.(4.51a) can be written as:

$$\int_{SE(3)} \left((M^{-1}H - \mathbb{I}) - \frac{1}{2}(M^{-1}H - \mathbb{I})^2 \right) f(H)dH \approx \mathbb{O}. \tag{5.11}$$

Expand Eq.(5.11) and multiply M on both sides of Eq.(5.11):

$$\int_{SE(3)} \left(2H - \frac{1}{2}HM^{-1}H - \frac{3}{2}M \right) f(H)dH \approx \mathbb{O}. \tag{5.12}$$

Substituting Eq. (5.6) into Eq.(5.12), we have:

$$2\widehat{M} - \frac{1}{2}\int_{SE(3)} HM^{-1}Hf(H)dH - \frac{3}{2}M \approx \mathbb{O}. \tag{5.13}$$

Let the 2nd order approximation of $M \in \mathbb{R}^{4\times 4}$ be denoted by $\overline{M}$:

$$2\widehat{M} - \frac{1}{2}\int_{SE(3)} H\overline{M}^{-1}Hf(H)dH - \frac{3}{2}\overline{M} = \mathbb{O}. \tag{5.14}$$

Take the data set $\{A_i\}$ for an example, the discrete version of Eq.(5.14) is:

$$\frac{2}{n}\sum_{i=1}^{n} A_i - \frac{1}{2n}\sum_{i=1}^{n} A_i\overline{M}_A^{-1}A_i - \frac{3}{2}\overline{M}_A = \mathbb{O}. \tag{5.15}$$

Similar to the 1st order case, $\overline{M}_A$ is the 2nd order approximation of $M_A \in \mathbb{R}^{4\times 4}$, because the candidates that satisfy Eq.(5.15) are not typically elements of $SE(3)$. A straight forward way of calculating $\overline{M}_A$ is by using nonlinear optimization technique given an initial guess. It is possible to solve for $\overline{M}_A \in \mathbb{R}^{4\times 4}$ and then project $\overline{M}_A$ back to $M_A^2 \in SE(3)$ as shown in [86], where M_A^2 denotes the new mean based on the 2nd order approximation. However, this highly nonlinear problem can be linearized by defining the update law as below:

$$M_A[j+1] = M_A[j]\exp(\Omega_A) \approx M_A[j](\mathbb{I} + \Omega_A), \tag{5.16}$$

where $M_A[j]$ is the value of $\overline{M}_A$ at the jth step, $\Omega_A \in se(3)$ is a Lie algebra element such that $\|\Omega_A\| \ll 1$. Under this assumption, the inverse of M_A can be updated as:

$$M_A^{-1}[j+1] = \exp^{-1}(\Omega_A) M_A^{-1}[j] \approx (\mathbb{I} - \Omega_A) M_A^{-1}[j]. \tag{5.17}$$

Substituting Eq. (5.16) and Eq. (5.17) into Eq. (5.15), the previous nonlinear problem on $\overline{M}_A$ is then converted into a linear one on Ω_A, where only Ω_A needs to be solved for given a known $M_A[j]$:

$$J \cdot vec(\Omega_A) = \mathbf{b}, \tag{5.18}$$

where

$$J = \frac{1}{2n} \sum_{i=1}^{n} \big((M_A[j]A_i)^T \otimes A_i\big) - \frac{3}{2}(\mathbb{I} \otimes M_A[j]) \tag{5.19a}$$

$$\mathbf{b} = vec\left[-\frac{2}{n}\sum_{i=1}^{n} A_i + \frac{1}{2n}\sum_{i=1}^{n}(A_i M_A[j] A_i) + \frac{3}{2}M_A[j]\right]. \tag{5.19b}$$

$\otimes$ is the symbol for Kronecker product and $vec()$ denotes the vectorization of a matrix formed by stacking its columns into a single column vector. .

To start with, we use the result M_A^1 obtained by Batch1 as the initial guess such that $M_A[0] = M_A^1$. We keep solving for Ω_A and update $M_A[j+1]$ and $M_A^{-1}[j+1]$ until the matrix norm of left hand side of Eq. (5.15) falls below a small threshold. It is observed that $M_A[j]$ can converge quickly in approximately 4 steps to obtain $\overline{M}_A$, which will be projected onto $SE(3)$ to get M_A^2. Similar to the Batch1 method, Batch2 method uses the new means (M_A^2, M_B^2) and the corresponding covariances (Σ_A^2, Σ_B^2) to recover X.

5.5 Numerical Simulations

In this section, numerical simulations are performed to show the advantages of using M^1 and M^2 as the means for $\{A_i\}$ and $\{B_j\}$ when solving for X. These Batch methods are compared with each other to show the accuracy in terms of recovering R_X and $\mathbf{t}_X$. They are also compared with the Kronecker product to show the effectiveness of probabilistic methods when dealing with data streams without correspondence.

5.5.1 Generation of $\{A_i\}$ and $\{B_j\}$ Using Different Distributions

First, we generate (A_i, B_i) data pairs which have correspondence given the ground truth of X_{true}. Then we scrambled the data in $\{A_i\}$ and $\{B_i\}$ up to a certain percentage rate to get $\{A_i\}$ and $\{B_j\}$ which have only partial or no correspondence. Finally, the scrambled data streams will be used in Batch,

Batch1, Batch2 and Kronecker product methods to calculate the R_X and $\mathbf{t}_X$ which will be compared with $R_{X_{\mathbf{true}}}$ and $\mathbf{t}_{X_{\mathbf{true}}}$.

Eq. (5.20) and Eq. (5.21) show two ways to generate (A_i, B_i) data pairs for numerical experiments, both of which sample on B_i first and then given X_{true} obtain the corresponding A_i:

$$B_i = B_0 \exp(\widehat{\delta_i}) \exp(\widehat{\gamma_i}) \tag{5.20a}$$

$$\delta_i = (\mathbf{0}_{3\times1}^T, \sigma\mathbf{n}_1^T)^T \in \mathbb{R}^{6\times1} \tag{5.20b}$$

$$\gamma_i = (\sigma\mathbf{n}_2^T / \|\mathbf{n}_2\|, \mathbf{0}_{3\times1}^T)^T \in \mathbb{R}^{6\times1} \tag{5.20c}$$

$$\sigma \in \mathbb{R}, \;\; \mathbf{n}_1, \mathbf{n}_2 \in \mathcal{N}^{3\times1}(0,1) \in \mathbb{R}^{3\times1}, \tag{5.20d}$$

and

$$B_i = B_0 \mathbf{exp}(\widehat{\delta_i}) \tag{5.21a}$$

$$\delta_i \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \tag{5.21b}$$

$$\Sigma = \sigma\mathbb{I}_{6\times6}, \sigma \in \mathbb{R}, \tag{5.21c}$$

where $B_0 \in SE(3)$ is an arbitrary "baseline" for generating the cloud of B_i, $\mathcal{N}^{3\times1}(0,1)$ denotes a 3 by 1 vector where each vector element follows a standard normal distribution $\mathcal{N}(0,1)$. $\mathcal{N}(\mathbf{0};\Sigma)$ is a zero mean multivariate Gaussian distribution. The major difference between Eq. (5.20) and Eq. (5.21) is that when perturbing B_0 to get $\{B_i\}$, the former samples on the rotational and translational component of $se(3)$ separately while the latter samples on each component simultaneously. After generating $\{B_i\}$, $\{A_i\}$ can be easily obtained as:

$$A_i = X_{true} B_i X_{true}^{-1}. \tag{5.22}$$

It turns out that the way of generating the data streams influences the effectiveness of different means. Equivalently speaking, each type of mean might be better at representing certain transformations in the $AX = XB$ context.

Next, we compare the performances of all the four methods in numerical simulation. Given an arbitrary B_0, 50 instances of B_i are generated using Eq. (5.20) and Eq. (5.21) respectively, with $\sigma = 0.9$. If we provide the ground truth X_{true}, 50 corresponding A_i can be calculated as in Eq. (5.22). In addition, different sets of $\{B_j\}$ are obtained by permuting the elements in $\{B_i\}$ by a percentage rate $r = 0, 10, 20, \ldots, 100$. Then the calculated R_X and $\mathbf{t}_X$ are compared with the ground truth X_{true} using the following error metrics:

$$error_R = || \log^\vee(R_X^T R_{X_{\mathbf{true}}}) ||_2 \tag{5.23}$$

$$error_{\mathbf{t}} = \frac{||t_X - t_{X_{\mathbf{true}}}||_2}{||t_{X_{\mathbf{true}}}||_2}. \tag{5.24}$$

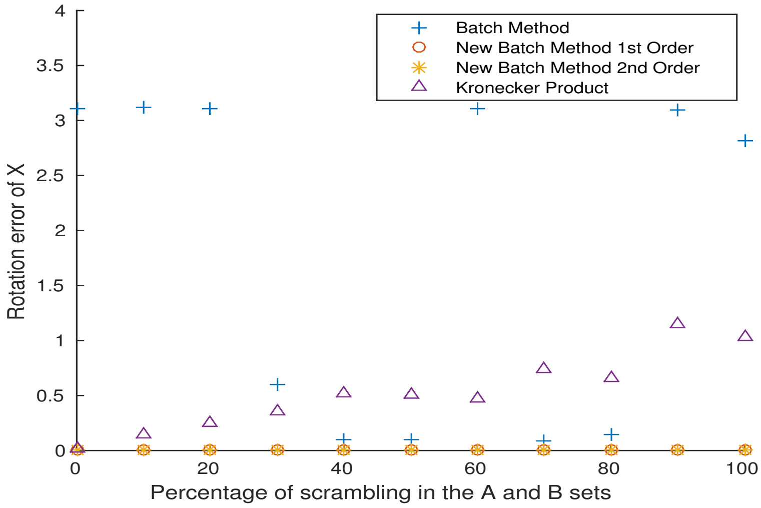


Figure 5.2: Rotation error vs. percentage of scrambling in $\{A_i\}$ and $\{B_j\}$ for the Batch, Batch1, Batch2 and Kronecker product methods where $\{B_i\}$ is generated using Eq. (5.20).

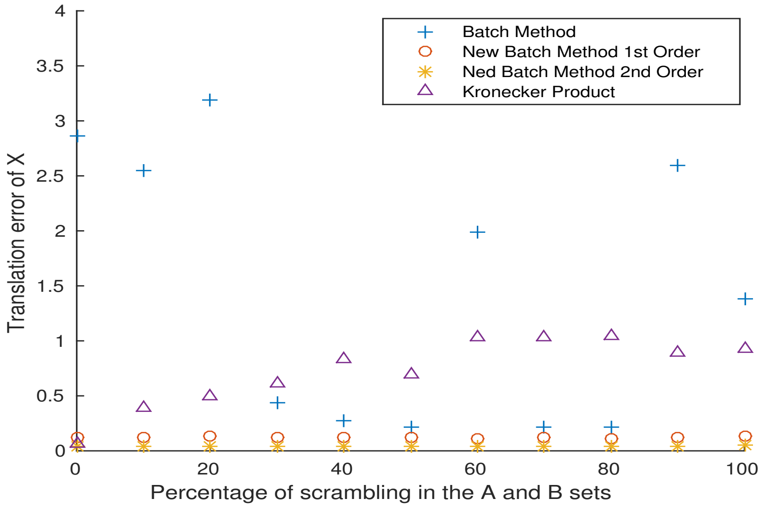


Figure 5.3: Translation error vs. percentage of scrambling in $\{A_i\}$ and $\{B_j\}$ for the Batch, Batch1, Batch2 and Kronecker product methods where $\{B_i\}$ is generated using Eq. (5.20)

Eq. (5.24) has the advantage of eliminating the influence of using different units for the translation. 70 trials are performed for each scrambling rate r and the averages of $error_R$ and $error_{\mathbf{t}}$ are calculated for comparison.

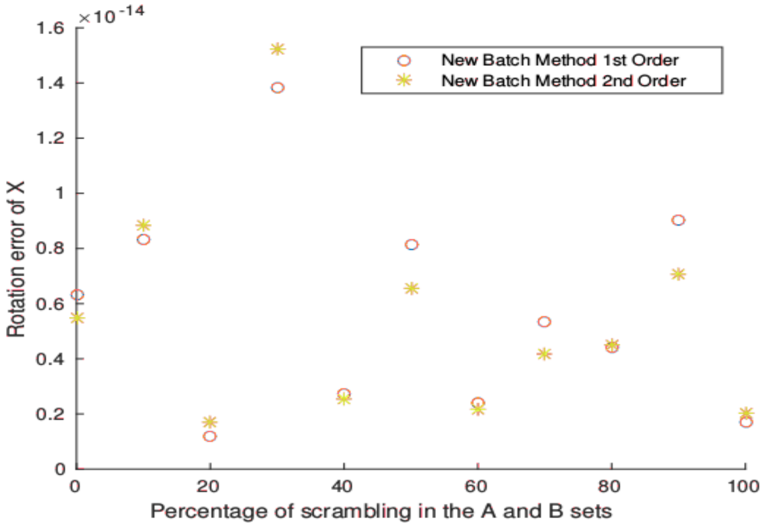


Figure 5.4: A close look at Fig. (5.2) for the Batch1 and Batch2 methods which shows that the rotation errors from Batch1 and Batch2 methods are at the magnitude of 10^{-15}.

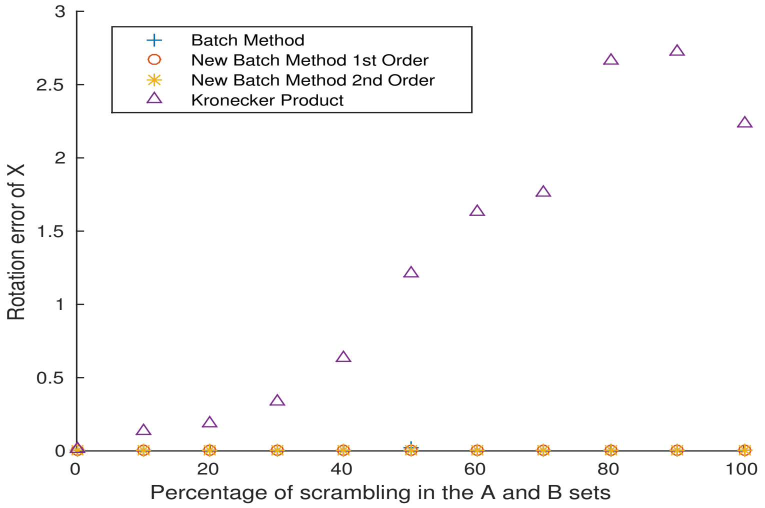


Figure 5.5: Rotation error vs. percentage of scrambling in $\{A_i\}$ and $\{B_j\}$ for the Batch, Batch1, Batch2 and Kronecker product methods where $\{B_i\}$ is generated using Eq. (5.21). Results of Batch method are overlapped by Batch1 & Batch2 and a close look can be found in Fig. (5.7)

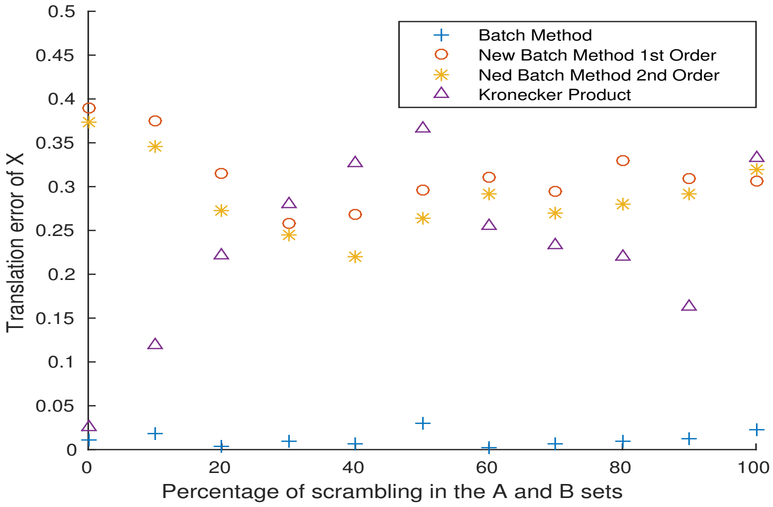


Figure 5.6: Translation error vs. percentage of scrambling in $\{A_i\}$ and $\{B_j\}$ for the Batch, Batch1, Batch2 and Kronecker product methods where $\{B_i\}$ is generated using Eq. (5.21)

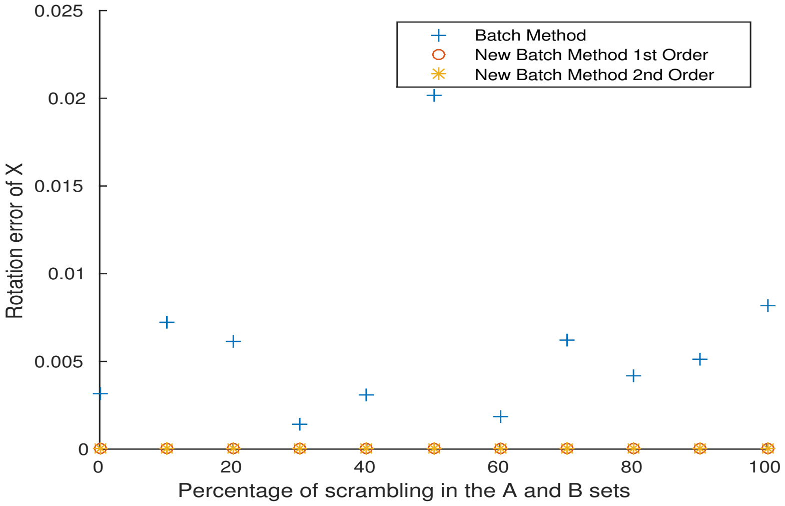


Figure 5.7: A close look at Fig. 5.5 for the Batch, Batch1 and Batch2 methods. It shows that the rotation errors from Batch1 and Batch2 methods are at the magnitude of 10^{-15} which is significantly smaller than that of the Batch method.

5.5.2 Numerical Simulation Results

All the simulation results are plotted as in Figs. 5.2-5.7 where Fig. 5.2-5.4 use the data sampled from Eq. (5.20) and Figs. 5.5-5.7 use the data sampled from Eq. (5.21). In addition, Fig 5.4 and Fig. 5.7 are closer looks at the Batch methods in Fig. 5.2 and Fig. 5.4 respectively. Several observations can be made from the figures. First, Figs. 5.2-5.3 and Figs. 5.5-5.6 show that all the Batch methods are invariant on the scrambling rate r whereas the results of the Kronecker product deteriorate quickly as r increases. Second, no matter which data sampling method is used, Batch1 and Batch2 methods show significant improvements over the Batch method on recovering R_X. The rotation error magnitude of Batch1 and Batch2 ranges from 10^{-14} to 10^{-15}, while the rotation error from Batch is at the level of 10^{-1} in Fig. 5.2 and 10^{-4} in Fig. 5.7. Moreover, the rotation error of Batch method can go beyond reasonable ranges occasionally as can be seen in Fig. 5.2. Third, Fig. 5.3 shows that the translational component t_X obtained by Batch1 and Batch2 are also better than Batch. However, Fig. 5.6 shows that the performance of Batch1 and Batch2 is not as good as that of Batch if Eq. (5.21) is used for data sampling. Lastly, in Fig. 5.3 and Fig. 5.6, the translational error from Batch2 is smaller than that of Batch1, which reflects the necessity of using the new mean M^2 based on the 2nd order approximation to solve for X.

In summary, for data sets $\{A_i\}$ and $\{B_j\}$ without correspondence, Batch1 and Batch2 are extremely good at recovering R_X in a precise and consistent manner no matter which sampling method is chosen. The accuracy of the calculated t_X is dependent on the sampled data, which is true for both the old and new Batch methods. One can be better than another if a different type of data samples is used. However, the Batch2 method is always better than Batch1 method in terms of getting a more accurate $\mathbf{t}_X$.

5.6 Conclusions

In this chapter, two new probabilistic methods are presented which can handle the $AX = XB$ problem without a priori knowledge of the correspondence between $\{A_i\}$ and $\{B_j\}$. The new approaches are built on top of the Batch method, which shows that an appropriate definition of mean for a set of rigid body transformations in $SE(3)$ can affect the effectiveness of the probabilistic methods to a large extent. The new definitions of means are derived from the $1st$ order and $2nd$ order approximations of the original definition. Rather than simple approximations, they reflect the "average" of a set of rigid body transformations from different perspectives, which significantly improves the accuracy and consistency for the calibration of X given data sets without correspondence. Numerical simulations are performed to show the superiority of the Batch1 and Batch2 methods.

Chapter 6

Probabilistic Approaches Towards $AX = YB$ Hand-Eye and Robot-World Calibration

The $AX = XB$ hand-eye calibration is widely used in calibration of a robot-sensor system, and a natural extension is the $AX = YB$ hand-eye and robot-world calibration. In the case of the $AX = YB$ problem, the relative hand-eye (X) and robot-world (Y) transformations must be determined to provide accurate data for use in control. As an added difficulty, the exact correspondence between the streams of sensor data (A's and B's) is typically unknown due to asynchrony in sampling rates and processing time. One common scenario is a constant shift between the two data streams.

The rest of the chapter is organized as follows. In Section 6.1, a literature review is given on $AX = YB$ problem. In Section 6.2, a novel probabilistic method is presented to solve for eight candidates of X and Y. In Section 6.3, an algorithm involving both a temporal correlation calculation and Euclidean group invariants is proposed to recover the correspondence between $\{A_i\}$ and $\{B_j\}$, which is used to select the optimal solution among the candidates. In Section 6.4, the simulation results obtained by taking noisy data without correspondence are illustrated. In Section 6.5, we briefly discuss the case where one can obtain (X, Y) without recovering the correspondence between the data sets. In Section 6.6, conclusions are drawn based on the numerical results.

6.1 Introduction

A variation of this problem is the $AX = YB$ problem, where both the hand-eye transformation and the pose of the robot base with respect to the world frame need to be calibrated. In a typical environment, the relationships between the sensor frame, robot frame, and world frame are variant and uncertainties exist. Therefore, simultaneous coordinate calibrations have to be determined frequently in order to enable the robot to respond to dynamic environments.

In the $AX = YB$ problem, data streams of A's and B's can be respectively obtained via different sensors. The data streams may arrive in an asynchronous fashion due to the different working frequencies of the sensors. This asynchrony causes a shift between the two data streams which can obscure the correspondence between the A's and B's. Moreover, data loss can destroy information about correspondence altogether. In this chapter, a novel method is presented to solve for X and Y without a priori knowledge of the correspondence between the A's and B's.

The hand-eye calibration problem can be modeled as $AX = XB$, where A and B are the homogeneous transformation matrices describing the relative motions of the end-effector and the sensor respectively. As shown in Fig. 6.1 part (2), $A = A^i(A^{i+1})^{-1}$ and $B = (B^i)^{-1}B^{i+1}$. Given multiple pairs of (A_i, B_i) with correspondence (note that (A_i, B_i) are the relative transformations obtained from the raw data), many deterministic methods have been proposed to solve for X.

Simultaneous estimation of the hand-eye and robot-world transformations has been viewed as the $AX = YB$ problem. As shown in Fig. 6.1 part (1), Y is the transformation from the robot base to the world frame, A denotes the pose of the sensor in the world frame and B is the transformation from the end-effector to its fixed base. Other concrete examples are the camera-IMU calibration in Fig. (6.2) and IR camera & drone camera calibrations in Fig. (6.3). The A and B in $AX = YB$ are different from those in $AX = XB$

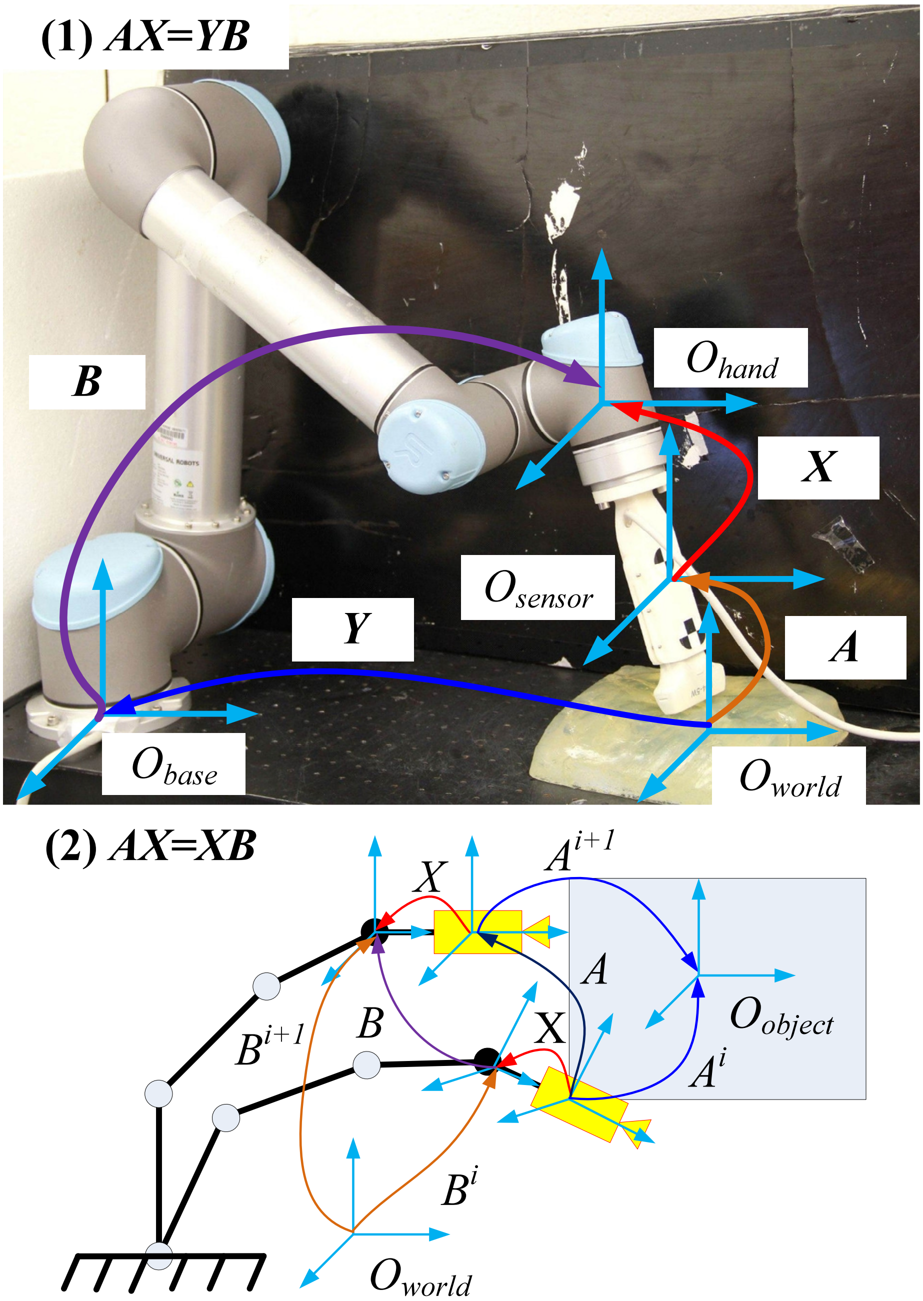


Figure 6.1: (1) The hand-eye and robot-world calibration problem formulated as $AX = YB$. (2) The hand-eye calibration problem formulated as $AX = XB$. Note: matrices A and B above have different physical meanings in the $AX = XB$ and $AX = YB$ problems.

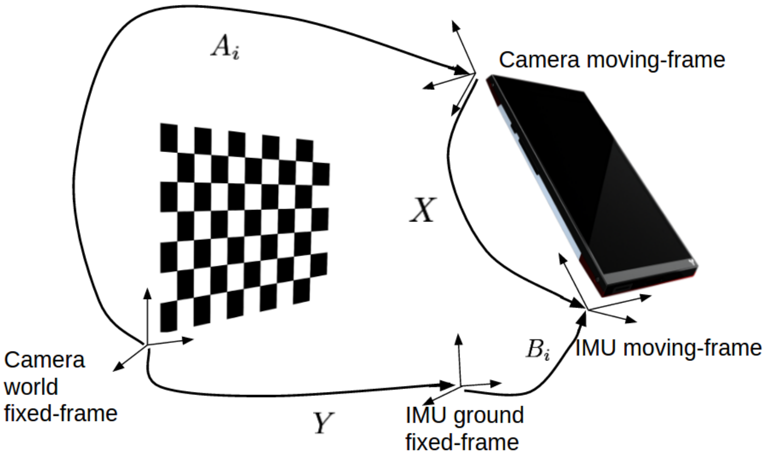


Figure 6.2: Application of $AX = YB$ in IMU-camera calibration of a mobile phone using checkerboard.

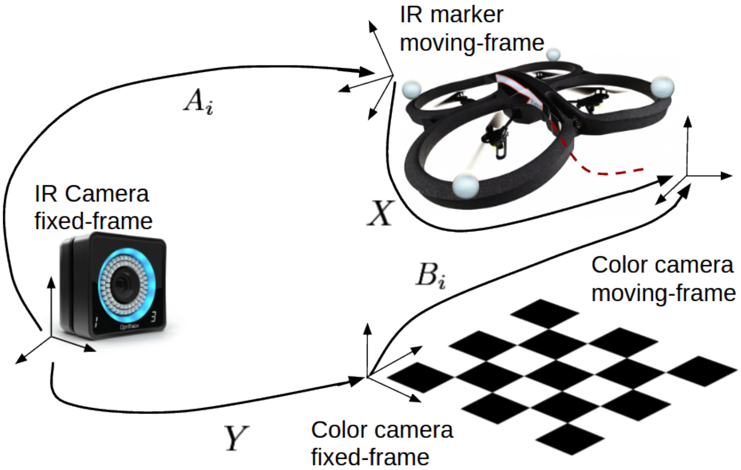


Figure 6.3: Application of $AX = YB$ in IR camera and drone camera calibrations.

where the former uses absolute transformations and the latter uses relative transformations. This problem has been solved by many different methods such as the Kronecker product, quaternion, dual quaternion, and nonlinear optimization methods [17–22, 84, 87]. Simultaneous calibration of X and Y can be problematic in that all the methods above assume exact correspondence between $\{A_i\}$ and $\{B_j\}$, which is not the case in the real world, and this is why a simultaneous solution for X and Y in the $AX = YB$ problem can be a challenging issue. In the above methods, the correspondence between A and B is known a priori. In this chapter, we focus on one case of the $AX = YB$ problem where there is no a priori knowledge of the correspondence between the data streams.

6.2 Solving $AX = YB$ Using a Probabilistic Method on Motion Groups

In this section, the probabilistic representation of $AX = YB$ is derived based on the mathematical concepts and formulations introduced in Chapter 2 and Chapter 4.

6.2.1 Fundamental Mathematical Framework

Given

$$A_i X = Y B_i. \tag{6.1}$$

where $i = 1, \ldots, n$, after employing the property of Delta function of $SE(3)$ on both sides of the equation separately, one can get

$$(\delta_{A_i} * \delta_X)(H) = \delta(A_i^{-1} H X^{-1}) \tag{6.2a}$$

$$(\delta_Y * \delta_{B_i})(H) = \delta(Y^{-1} H B_i^{-1}). \tag{6.2b}$$

Using Eq. (2.14) and Eq. (6.1), the above two equations can be combined into a single equation as:

$$(\delta_{A_i} * \delta_X)(H) = (\delta_Y * \delta_{B_i})(H). \tag{6.3}$$

Defining the PDF of $\{A_i\}$ and $\{B_i\}$ as:

$$f_A(H) = \frac{1}{n}\sum_{i=1}^{n} \delta_{A_i}(H) \tag{6.4a}$$

$$f_B(H) = \frac{1}{n}\sum_{i=1}^{n} \delta_{B_i}(H), \tag{6.4b}$$

then by using the bi-linearity of convolution, add n instances of Eq. (6.3), and substitute Eq. (6.4) into the summation, and we will have:

$$(f_A * \delta_X)(H) = (\delta_Y * f_B)(H). \tag{6.5}$$

The mean and covariance of f_A and f_B are defined as in Eq. (4.51a) and Eq. (4.51b). Given $\{A_i\}$ with the cloud of frames A_i clustering around M_A, an iterative formula can be used for computing M_A [29] as:

$$^{k+1}M_A =^k M_A \circ \exp\left[\frac{1}{n}\sum_{i=1}^{n}\log(^kM_A^{-1} \circ A_i)\right]. \tag{6.6}$$

An initial estimate of the iterative procedure can be chosen as:

$$^0M_A = \exp\left(\frac{1}{n}\sum_{i=1}^{n}\log(A_i)\right).$$

Alternatively, M_A can be obtained by solving a nonlinear optimization problem with the cost function

$$C_1(M_A) = \left\|\sum_{i=1}^{n}\log(M_A^{-1}A_i)\right\|^2.$$

Note, however, that mathematically this is not the same as minimizing

$$C_2(M_A) = \sum_{i=1}^{n}\left\|\log(M_A^{-1}A_i)\right\|^2,$$

though in practice they often are minimized by very close values of M_A.

A similar procedure can be used to compute M_B. Σ_A and Σ_B are then straightforward to compute once M_A and M_B are known. Because X and Y

are constant, their corresponding PDF will be $\delta_X(g)$ and $\delta_Y(g)$, of which the mean and covariance are $M_X = X$, $\Sigma_X = \mathbb{O}_6$ and $M_Y = Y$, $\Sigma_Y = \mathbb{O}_6$, respectively. If we further assume that $f_A(H)$ and $f_B(H)$ are two highly focused PDFs, then Eq. (4.53a) and Eq. (4.53b) can be used to the propagation of mean and covariance, and the following equations can be obtained:

$$M_A X = Y M_B \tag{6.7a}$$

$$Ad(X^{-1})\Sigma_A Ad^T(X^{-1}) = \Sigma_B. \tag{6.7b}$$

This is a nonparametric result, meaning that the underlying probability density functions $f_A(H)$ and $f_B(H)$ need not be Gaussian or belong to any other family of parametric distributions.

6.2.2 Generalization Beyond Highly Focused Data

Moreover, it can be shown that in the context of $AX = YB$, Eq. (6.7a) and Eq. (6.7b) don't require f_A and f_B to be highly concentrated. Starting with Eq. (6.5), performing a convolution on both of the left sides of the equation with $\delta_{Y^{-1}}(H)$, and using the associativity of convolution, we will have:

$$(\delta_{Y^{-1}} * f_A * \delta_X)(H) = f_B(H). \tag{6.8}$$

Use the definition of mean as in Eq. (4.53a) along with Eq. (6.8), one can get

$$\int_{SE(3)} \log(M_B^{-1}H) f_B(H) dH =$$

$$\int_{SE(3)} \log(M_B^{-1}H)(\delta_{X^{-1}} * f_A * \delta_X)(H) dH =$$

$$\int_{SE(3)} \log(M_B^{-1}H) f_A(YHX^{-1}) dH = \mathbb{O}_4. \tag{6.9}$$

Change the variable as $K = YHX^{-1}$ and use the invariance of integration [27], then Eq. (6.9) becomes:

$$\int_{SE(3)} \log(M_B^{-1}Y^{-1}KX) f_A(K) dK = \mathbb{O}_4. \tag{6.10}$$

which falls into the form of the mean definition of $\{A_i\}$. If we further multiply X and X^{-1} on the left and right of both sides of Eq. (6.10), then $X[\log(M_B^{-1}Y^{-1}KX)]$ $X^{-1} = \log(M_A^{-1}K)$. Knowing that $X[\log(M_B^{-1}Y^{-1}KX)]X^{-1} = \log(XM_B^{-1}Y^{-1}K)$, and we have $XM_B^{-1}Y^{-1} = M_A^{-1}$ which is equivalent to Eq. (6.7a). Eq.(6.7b) follows from [29] because the $F(A,B)$ term as defined in [29] has products of covariances of the functions being convolved, and delta functions have zero covariance so the $F(A,B)$ term is zero, which results in Eq. (6.7b).

6.2.3 Candidates of (X, Y) and Screw Invariants

The problem of solving the above equations, Eq. (6.7a) is decomposed into a rotational equation and a translational equation as follows:

$$R_{M_A} R_X = R_Y R_{M_B} \tag{6.11a}$$

$$R_{M_A} t_X + t_{M_A} = R_Y t_{M_B} + t_Y. \tag{6.11b}$$

Σ_A and Σ_B can be decomposed into blocks as $\begin{pmatrix} \Sigma_A^1 & \Sigma_A^2 \\ \Sigma_A^3 & \Sigma_A^4 \end{pmatrix}$ and $\begin{pmatrix} \Sigma_B^1 & \Sigma_B^2 \\ \Sigma_B^3 & \Sigma_B^4 \end{pmatrix}$, respectively. Knowing that $X^{-1} = \begin{pmatrix} R_X^T & -R_X^T t_X \\ 0 & 1 \end{pmatrix}$, then the first two blocks of Eq. (6.7b) can be written as follows:

$$\Sigma_{M_B}^1 = R_X^T \Sigma_{M_A}^1 R_X \tag{6.12a}$$

$$\Sigma_{M_B}^2 = R_X^T \Sigma_{M_A}^1 R_X (\widehat{R_X^T t_X}) + R_X^T \Sigma_{M_A}^2 R_X. \tag{6.12b}$$

Because Eq. (6.12a) is a similarity transformation between $\Sigma_{M_B}^1$ and $\Sigma_{M_A}^1$, they share the same eigenvalues and can be eigendecomposed into $\Sigma_{M_A}^1 = Q_{M_A} \Lambda Q_{M_A}^T$ and $\Sigma_{M_B}^1 = Q_{M_B} \Lambda Q_{M_B}^T$ where Λ is a diagonal matrix whose diagonal elements are the eigenvalues of $\Sigma^1_{M_A}$ (or $\Sigma^1_{M_B}$), and Q_{M_A} (or Q_{M_B}) is a square matrix whose columns are the corresponding eigenvectors. The following equation is obtained after substituting $\Sigma^1_{M_B}$ and $\Sigma^1_{M_A}$ into Eq. (6.12a):

$$\Lambda = (Q^T_{M_A} R^T_X Q_{M_B})\Lambda(Q^T_{M_B} R_X Q_{M_A}) = P\Lambda P^T \tag{6.13}$$

where $P = Q^T_{M_A} R_X Q_{M_B}$. Since Q_{M_A} and Q_{M_B} are further constrained to be rotation matrices, the orthogonal matrix P satisfies Eq. (6.14).

$$\begin{cases} P^T = P^{-1} \\ \det(P) = \pm 1. \end{cases} \tag{6.14}$$

Combing Eq. (6.13) and Eq. (6.14), then an orthogonal matrix P can be one of $\mathcal{P}$ or $-\mathcal{P}$:

$$\begin{aligned} \mathcal{P} = \Bigg\{ &\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ &\begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \Bigg\}. \end{aligned} \tag{6.15}$$

Therefore, there are eight candidates for R_X which can be calculated via

$R_X = Q_{M_A} P Q_{M_B}^T$, and the corresponding t_X can be obtained from Eq. (6.12b). Given known X, Y can be solved from $Y = M_A X M_B^{-1}$. At last, eight candidate pairs of $\{X_k, Y_k\}$ can be obtained as:

$$X_k = \begin{pmatrix} R_{X_k} & t_{X_k} \\ 0^T & 1 \end{pmatrix}, \quad Y_k = \begin{pmatrix} R_{Y_k} & t_{Y_k} \\ 0^T & 1 \end{pmatrix} \tag{6.16}$$

where $k = 1, 2, ..., 8$.

The problem then becomes selecting the best pair of $\{X_k, Y_k\}$ from the eight candidates. Based on screw theory, it is known that a homogeneous transformation H can be expressed by the four screw parameters $(\theta, d, \mathbf{n}, \mathbf{p})$ as:

$$H = \begin{pmatrix} e^{\theta\hat{n}} & (I_3 - e^{\theta\hat{n}})p + dn \\ 0^T & 1 \end{pmatrix} \tag{6.17}$$

where θ is the angle of rotation, d is the translation along the rotation axis, n is the unit vector representing the axis of rotation and p is the position of a point on the line relative to the origin of a space-fixed reference frame with $p \cdot n = 0$.

Moreover, $AX_k = Y_k B$ can be written as $AX_k = X_k(X_k^{-1} Y_k B)$. Defining $B^k = X_k^{-1} Y_k B$, we have $AX_k = X_k B^k$. As discussed in [81], for $AX = XB$ problem, there exist two Euclidean-group invariant relationships for each pair of $(A_i, B_i^k)(i = 1, \cdots, n; k = 1, \ldots, 8)$ as follows:

$$\theta_{A_i} = \theta_{B_i^k}, d_{A_i} = d_{B_i^k}. \tag{6.18}$$

Among the eight pairs (X_k, Y_k), one can find an optimal solution which minimizes the cost function defined as:

$$(X, Y) = \underset{(X_k, Y_k)}{argmin} \frac{1}{n} \sum_{i=1}^{n} (\| \theta_{A_i} - \theta_{B_i^k} \| + \| d_{A_i} - d_{B_i^k} \|). \tag{6.19}$$

In summary, eight candidates of (X_k, Y_k) can be calculated without knowing the correspondence between A_i and B_j. However, the correspondence needs to be recovered to pick the optimal (X_k, Y_k). Note that the Euclidean-group invariant relationships in the context of $AX = YB$ problem are still unknown. Therefore, $AX = YB$ is converted into $AX = XB$ problem to recover the correspondence of data using screw invariants.

6.3 Solution with Unknown Correspondence between A_i and B_j^k

In most cases, the two sets of homogeneous transformations $\{A_i\}$ and $\{B_j\}$ are calculated based on the data obtained from different sensors. Due to the asynchronous timing of the sensor readings, the correspondence between $\{A_i\}$ and $\{B_j\}$ is usually unknown. This section deals with the case where there is a shift between $\{A_i\}$ and $\{B_j\}$, and the Euclidean-group invariants are used to recover the correspondence between the data streams. The advantage of the above probabilistic solution lies in that X and Y can be calculated even if there is no a priori knowledge of the correspondence. However, there are still eight possible candidates of (X_k, Y_k) to choose from and by using Euclidean-group invariants, it is straightforward to determine which is the optimal pair if the correspondence between A_i and B_j^k can be known.

The Discrete Fourier Transform (DFT) decomposes a time-domain signal into its constituent frequencies. The input is a finite list of equally spaced samples of a function. Given a discrete signal consisting of a sequence of N complex numbers $x_0, x_1, \cdots, x_{N-1}$, the DFT is denoted by $X_\kappa = \mathcal{F}(x_n)$ as:

$$X_\kappa = \sum_{n=0}^{N-1} x_n \cdot exp(-i\frac{2\pi}{N}n\kappa). \tag{6.20}$$

where i here is the imaginary unit.

The Inverse Discrete Fourier Transform (IDFT) is denoted as:

$$x_n = \frac{1}{N}\sum_{n=0}^{N-1} X_\kappa \cdot exp(i\frac{2\pi}{N}n\kappa). \tag{6.21}$$

The discrete convolution of two sequences f_n and g_n is defined as:

$$(f * g)(\tau) = \sum_{j=0}^{N} f(t_j)g(t_j - \tau). \tag{6.22}$$

In the convolution theorem, the Fourier transform of a convolution is the product of the Fourier transforms, namely:

$$f * g = \mathcal{F}^{-1}[\mathcal{F}(f) \cdot \mathcal{F}(g)]. \tag{6.23}$$

The correlation theorem indicates that the correlation function, $Corr(f, g)$, will be larger for a shift vector where the two sequences f_n and g_n share more similar features. The correlation can be obtained based on the convolution theorem. The DFT of $Corr(f, g)$ is equal to the product of the DFT of f_n and the complex conjugate $\mathcal{F}^*$ of the DFT of g_n:

$$Corr(f, g) = f \star g = \mathcal{F}^{-1}[\mathcal{F}(f) \cdot \mathcal{F}^*(g)]. \tag{6.24}$$

Compared to the standard time-domain convolution algorithm, the complexity of the convolution by multiplication in the frequency domain is significantly reduced with the help of the convolution theorem and the Fast Fourier Transform (FFT).

Given two sequences $\{\theta_{A_i}\}$ and $\{\theta_{B_j^k}\}$ corresponding to $\{A_i\}$ and $\{B_j^k\}$, the shift that is needed to recover the data correspondence is obtained as below. Firstly, θ_{Ai} and $\theta_{B_j^k}$ are normalized as:

$$\theta_{1,k} = \frac{(\theta_{A_i} - \mu_A)}{\sigma_A}, \theta_{2,k} = \frac{(\theta_{B_j^k} - \mu_{B^k})}{\sigma_{B^k}} \tag{6.25}$$

where $\mu_A(\mu_{B^k})$ is the mean of $\theta_{A_i}(\theta_{B_i^k})$ and $\sigma_A(\sigma_{B^k})$ is the standard deviation.

Here, the correlation function $Corr(\theta_{1,k}, \theta_{2,k})$ is the function of the time sequence index n which describes the probability of these two sequences being separated by this particular index. The index corresponding to the maximum of $Corr(\theta_{1,k}, \theta_{2,k})$ indicates the amount of shift τ_{shift} between $\{\theta_{A_i}\}$ and $\{\theta_{B_j^k}\}$.

$$\tau_{shift} = \underset{index}{argmax}(Corr(\theta_{1,k}, \theta_{2,k})) \tag{6.26}$$

Therefore, the correspondence between the two sequences can be found. The data of θ_{A_i} or d_{A_i} are shifted by $-\tau_{shift}$ to obtain a sequence of new pairs $(\theta_{A_i}(i + \tau_{shift}), \theta_{B_i^k})$ and $(d_{A_i}(i + \tau_{shift}), d_{B_i^k})$, where $max(0, \tau_{shift}) \leq i \leq min(n, n + \tau_{shift})$. The data stream can be shifted back to regain correspondence to synchronize the data streams once the shift is computed, and the optimal solution of X and Y can also be recovered by minimizing the cost function in Eq. (6.19) using the Euclidean-group invariants.

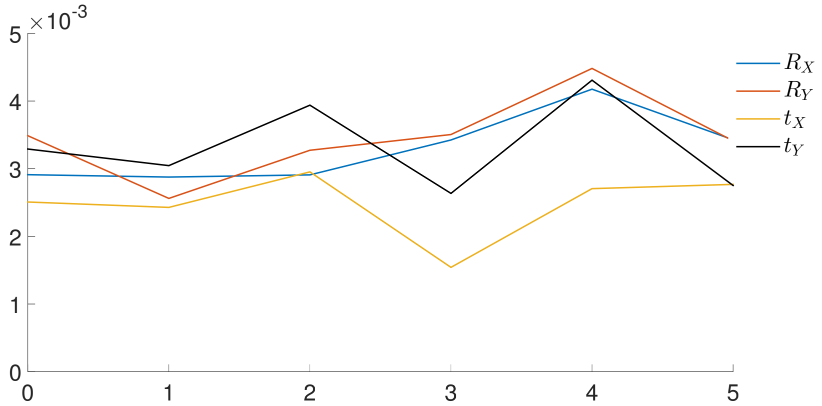


Figure 6.4: The translational and rotational errors versus the shift between data streams $\{A_i\}$ and $\{B_i\}$.

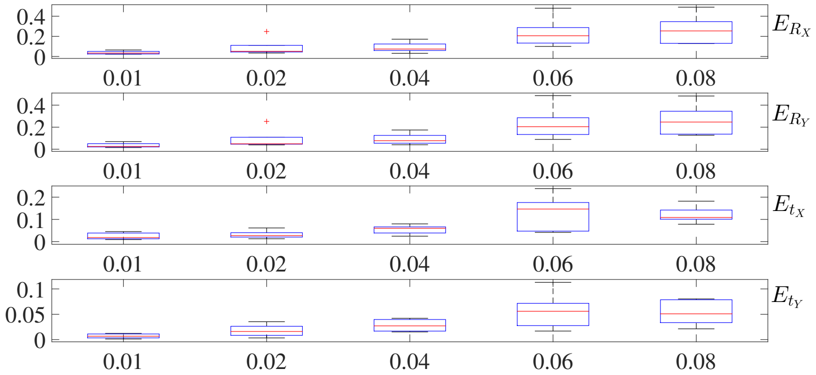
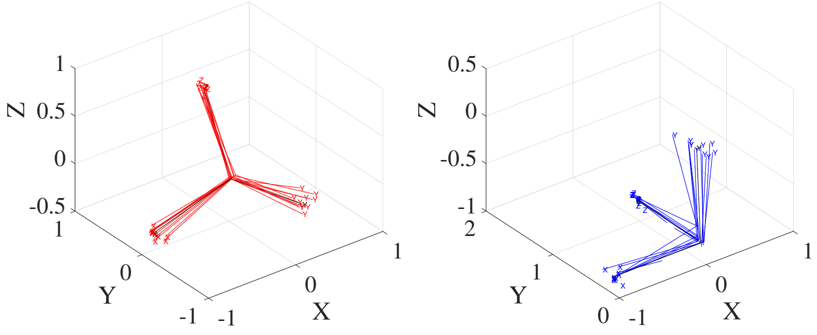


Figure 6.5: Box-and-whisker plots of translational and rotational errors versus the covariance noise on data stream $\{B_i\}$.



(a) X_{solved} (b) Y_{solved}

Figure 6.6: (a) The solved X (in red) and the actual X (in black) for 10 simulation trials with covariance noise of 0.05 and shift of 2. (b) The solved Y (in blue) and the actual Y (in black) for 10 simulation trials with covariance noise of 0.05 and shift of 2.

6.4 Numerical Simulations

For the numerical experiments in this section, the rotational and translational errors for X and Y are measured as

$$Error(R_X) = \parallel log^{\vee}(R^T_{X_{Solved}} R_{X_{true}}) \parallel \tag{6.27a}$$

$$Error(t_X) = \parallel t_{X_{Solved}} - t_{X_{true}} \parallel \tag{6.27b}$$

$$Error(R_Y) = \parallel log^{\vee}(R^T_{Y_{Solved}} R_{Y_{true}}) \parallel \tag{6.27c}$$

$$Error(t_Y) = \parallel t_{Y_{Solved}} - t_{Y_{true}} \parallel \tag{6.27d}$$

There are multiple ways of generating the data streams $\{A_i\}$ and $\{B_i\}$. One way is to first generate $\{B_i\}$ and then map it to $\{A_i\}$ using $A = YBX^{-1}$. $\{B_i\}$ can be obtained by randomly sampling on the Lie algebra of B from a zero mean multivariate Gaussian distribution as follows:

$$\delta_i \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \tag{6.28a}$$

$$B_i = \text{exp}(\hat{\delta}_i)\text{exp}(\mu) \tag{6.28b}$$

where the mean $\mu = \mathbf{0} \in se(3)$ and the covariance matrix $\Sigma \in \mathbb{R}^{6\times 6}$ is a diagonal matrix with same diagonal elements σ. The hat operator $\hat{\delta}$ converts a 6 by 1 vector into its corresponding Lie algebra. The data stream $\{A_i\}$ can be easily obtained as described above. After employing the proposed probabilistic method, 8 sets of sequences $(\theta_{A_i}, \theta_{B_i^k})$ and $(d_{A_i}, d_{B_i^k})$ can be obtained respectively where $i = 1, \cdots, 100$ and $k = 1, \ldots, 8$.

If the data stream $\{A_i\}$ is shifted by m units relative to $\{B_i\}$, then the maximum of the cross correlation can be used to recover the shift. After that, we can shift the data stream $\{A_i\}$ back to its original position to recover the correct correspondence with $\{B_i\}$, which will be used to find a correct solution satisfying the Euclidean-group invariants as defined in Eq. (6.18). Therefore, a unique pair of (X_k, Y_k) ($k = 1, \cdots, 8$) can be selected to minimize the cost function. In Fig. 6.4, because the shift between $\{A_i\}$ and $\{B_i\}$ is calculated accurately, the translational and rotational errors fluctuate by only a small amount compared to the errors of the no-shift data streams.

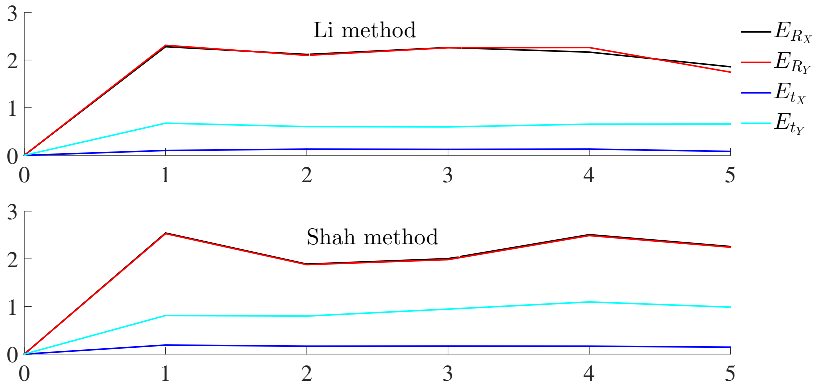


Figure 6.7: Orientation and translation errors of X and Y versus shift using Li's and Shah's methods without correspondence.

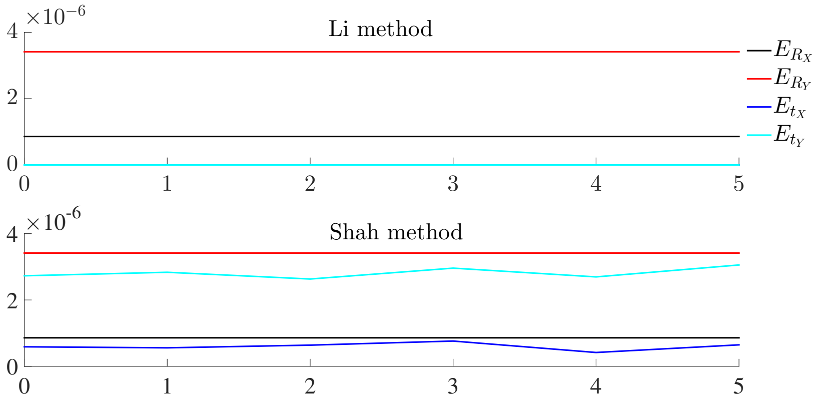


Figure 6.8: Orientation and translation errors of X and Y versus shift using Li's and Shah's methods with correspondence.

To test the robustness of the proposed method, noises are exerted onto $\{B_i\}$ by employing $B_i^{noise} = B_i\text{exp}(\widehat{\mathbf{x}}_{noise})$, where each element of the Lie Algebra $\mathbf{x}_{noise}$ belongs to the Gaussian distribution defined as $N \sim (\mu_{noise}, \sigma_{noise})$. In Fig. 6.5, as the covariance noise σ_{noise} increments from 0.01 to 0.08, the errors of R_X, R_Y, t_X, and t_Y increase as shown in the box-and-whisker plot. There are several outliers outside the whiskers, while the median is calculated as the final solved X and Y. Fig. 6.6 shows the solved (X,Y)s in red and blue with the actual (X,Y) in black with covariance noise of $\sigma = 0.05$ and shift $n = 2$.

The probabilistic method can recover the correspondence between shifted data streams, which is useful for other sensor calibration methods. In the $AX = YB$ problem, there have been many calibration methods developed for solving X and Y given data streams with correspondence. However, few of them considered the cases without correspondence. When data streams of A and B are shifted or asynchronous, most of these methods fail to give a valid solution. To further test the effectiveness of our method, we shift the data sequence of $\{A_i\}$ by $n = 0, 1, 2, 3, 4, 5$ with respect to the data sequence of $\{B_i\}$ such that A_{k+n} "matches" B_k where $k = 1, 2, \cdots, m - n$ and $i = 1, 2, \cdots, m$. We augment other $AX = YB$ solvers with our probabilistic approach by recovering the correspondence between shift data sets. In Li's method [20], X and Y are solved for at the same time, while Shah [87] solved for X and Y in a separate way. As shown in Fig. 6.7, when dealing with the shifted data streams $\{A_{k+n}, B_k\}$, the errors on both rotations and translations are significant. After recovering the correspondence between data streams by using the probabilistic method, Li and Shah's methods achieve the same level of performance as shown in Fig. 6.8.

6.5 A Brief Case Study with Completely Scrambled Data

In this section, we will briefly discuss the case where $\{A_i\}$ and $\{B_j\}$ are completely scrambled. Unlike the case of shifted data, it is extremely hard to recover the correspondence between two completely scrambled data sets $\{A_i\}$ and $\{B_j\}$. The correlation theorem can't be applied because there is no shift in the scrambled data sets. Euclidean group invariants are not practical because given $\{A_i\}$ and $\{B_j\}$ both of which have the size of m, there are $m! = m \times m-1 \times \cdots 1$ combinations between the data sets, and it is extremely computationally intensive to test all the combinations. Without recovering the correspondence between the data sets, it is impossible to choose the optimal solution from the eight candidates of $\{X_k, Y_k\}$.

In the above approach, we used Eq. (6.7b) to calculate X_k and Eq. (6.7a) to obtain the corresponding Y_k. However, we now show that one can calculate the eight candidates of Y independently and employ Eq. (6.7a) as a constraint to filter out the optimal $\{X, Y\}$ pair. Given the equation $AX = YB$, apply an inverse on both sides of the equation and we will have $B^{-1}Y^{-1} = X^{-1}A^{-1}$.

Following the same derivations from Eq. (6.1) to Eq. (6.7b), we have:

$$M_{B^{-1}}Y^{-1} = X^{-1}M_{A^{-1}} \tag{6.29a}$$

$$Ad(Y)\Sigma_{B^{-1}}Ad^T(Y) = \Sigma_{A^{-1}}. \tag{6.29b}$$

Similarly, Eq. (6.29b) can give eight candidates of Y^{-1}, or equivalently, the eight candidates of Y. Let X_{k1} where $k1 = 1, 2, \cdots, 8$ denote the Xs obtained from Eq. (6.7b) and Y_{k2} where $k2 = 1, 2, \cdots, 8$ denote the Ys obtained from Eq. (6.29b), and we can use Eq. (6.7a) and Eq. (6.29a) to form a minimization problem as:

$$\underset{k1,k2}{min}||M_A X_{k1} - Y_{k2} M_B||_F + ||M_B^{-1} Y_{k2}^{-1} - X_{k1}^{-1} M_A^{-1}||_F \tag{6.30}$$

which can give the optimal (X_{k1}, Y_{k2}) pair. We call this approach the *prob* method, and compare it with Li's method for testing its effectiveness of handling scrambled data sets. For simplicity, we use Eq. (6.29a) and Eq. (6.29b) to generate $\{B_i\}$, whereas compute $\{A_i\}$ using $A_i = X^{-1}YB_i$ without exerting noise on B_i. Then $\{A_i\}$ is scrambled at each percentage from 0% up to 100%. 50 times of simulations are performed for each percentage rate and the same error metrics are used as in Section 6.6. As shown in Fig. (6.9) and Fig. (6.10),

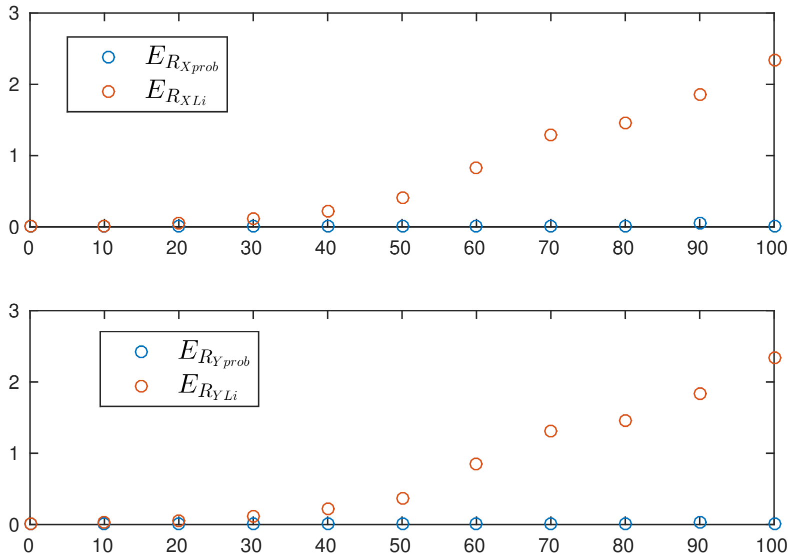


Figure 6.9: Rotation error in X and Y v.s. scrambling rate for the $prob$ and Li's methods

as the percentage of scrambled data goes up, the errors in rotation and translation for Li's method gradually diverge, while the errors for the $prob$ method are very stable and small. This shows the significant advantage of the probabilistic method in handling disordered data sets. However, Li's method is still more accurate when the exact correspondence is known between $\{A_i\}$ and $\{B_i\}$.

6.6 Conclusions

In this section, a probabilistic approach is developed to recover the correspondence between shifted data streams to augment traditional $AX = YB$

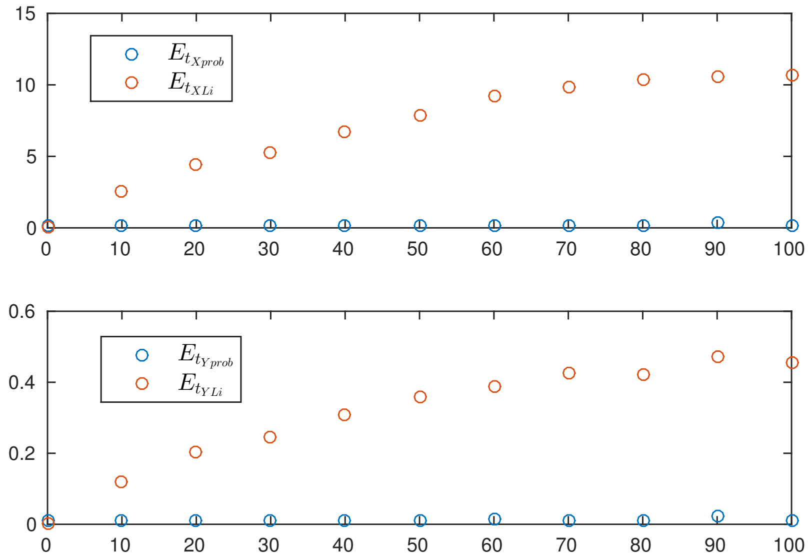


Figure 6.10: Translation error in X and Y v.s. scrambling rate for the $prob$ and Li's methods

solvers. Without a priori knowledge of the correspondence between $\{A_i\}$ and $\{B_j\}$, the proposed probabilistic method on Lie groups is used to constrain the possible solutions of X and Y to eight pairs of candidates. Given shifted data streams of $\{A_{i+s}\}$ and $\{B_i\}$, using the correlation theorem with Euclidean-group invariants, the correspondence is recovered to determine the correct solution among the eight candidates. In the numerical simulation, the method performs well with different sets of data samples. In addition, the shifted-back data streams are further validated by the Li and Shah solvers. Lastly, a new approach is presented to deal with completely disordered data sets and briefly show its effectiveness in simulation.

Chapter 7

Probabilistic Approaches Towards $AXB = YCZ$ of Robotic System Calibration

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 system 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 rigid body transformations measured from sensors and X, Y, Z are unknown 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 $\{C_i\}$. Moreover, the existing methods in the literature require good initial estimates that are not always easy or possible to obtain. To address this, two probabilistic approaches are presented that can solve the $AXB = YCZ$ problem without a priori knowledge of the correspondence of the data. In addition, no initial estimates are required for recovering X, Y and Z. However, noise is ubiquitous in the real world and the above two methods are sensitive to the noisy data. In addition, a hybrid method is presented which combines traditional $AXB = YCZ$ solvers with the probabilistic methodology. It is shown that the new algorithm is robust to both the 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.

The rest of the chapter is organized as follows. In section 7.1, we introduce the background of multi-robot calibration. Section 7.2 describes in detail the formulation of the two probabilistic $AXB = YCZ$ solvers. In section 7.3, 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 7.4 presents a new hybrid approach which combines traditional solvers with probabilistic methodology to handle noisy data. In section 7.5, conclusions are given.

7.1 Introduction

Many multirobot calibration problems can be formulated using the equation $AXB = YCZ$, where A, B and C are known homogeneous transformations from sensor readings, and X, Y and Z are unknown relationships between two target frames. For the dual arm system [24] shown in Fig. (7.1), 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 [88] illustrated in Fig. (7.2), 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. (7.3), the problem of the serial-parallel hybrid robot system [89] is cast as the tool-gripper (X), flange-base (Y) and camera-base (Z)

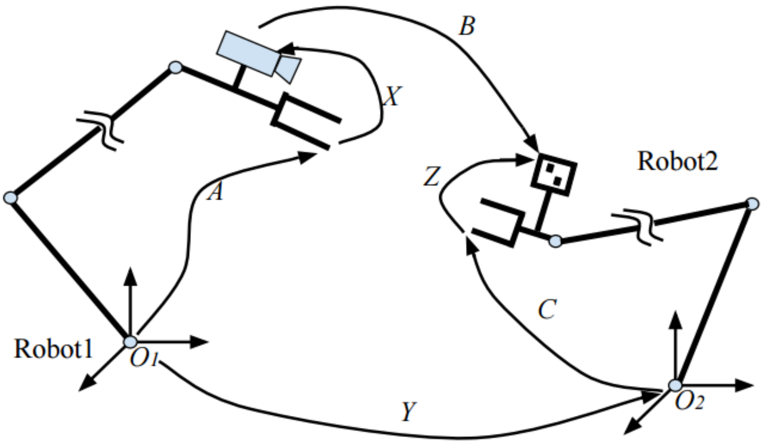


Figure 7.1: The Hand-Eye, Robot-Robot, Tool-Flange Calibration of a Dual Arm System

calibrations. The same mathematical modeling also exists in co-robotic ultrasound (US) tomography where two hand-eye and one robot-robot calibrations are needed [90]. However, relatively little work has been done on $AXB = YCZ$ calibration. To the best of our knowledge, only Wang [24,91] and Yan [89] have proposed several algorithms for solving X, Y and Z simultaneously.

In this chapter, we proposed two "probabilistic" frameworks for solving the $AXB = YCZ$ robot system calibration problem. Due to the different physical properties of robotic systems, two types of probabilistic $AXB = YCZ$ solvers are built which greatly reduce the need for a priori knowledge of the 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

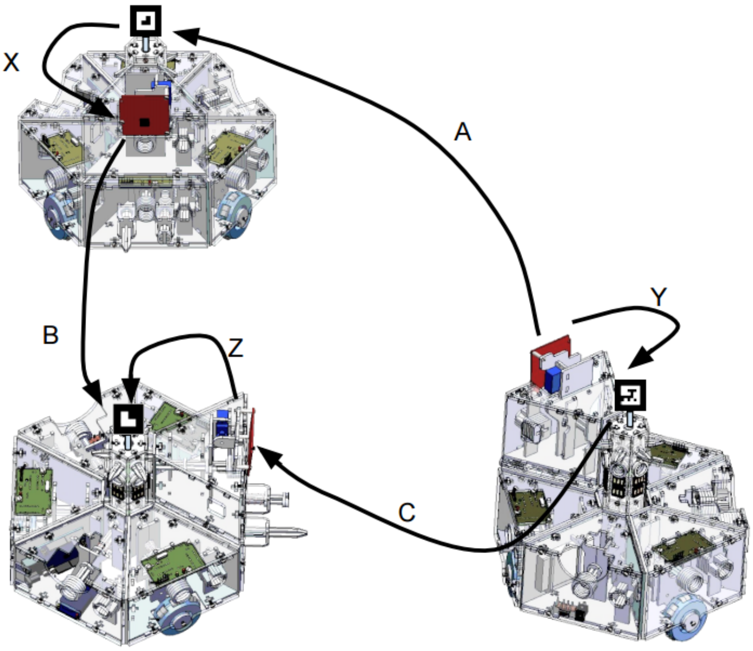


Figure 7.2: Triple Hand-Eye Calibration of a Multi-Robot System

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.In addition, we presented a hybrid approach which combines traditional deterministic $AXB = YCZ$ solvers and probabilistic estimates of rigid-body poses. The framework is simple and turns out to be very effective at handling noisy data when there is partial knowledge of the correspondence.

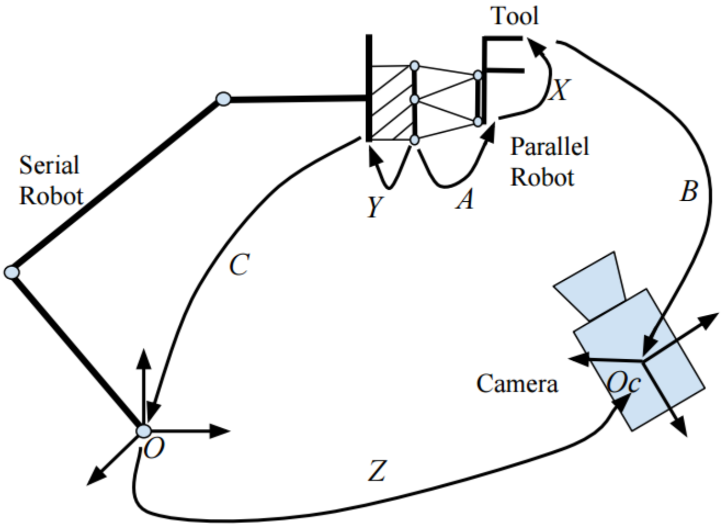


Figure 7.3: Flange-Base, Camera-Base and Tool-Gripper Calibration of a Serial-Parallel Manipulator

7.2 Problem Formulation

In this section, we derive the mathematical frameworks of the two probabilistic approaches for the $AXB = YCZ$ problem. They share a common theoretical framework but are designed for different types of robotic systems. Note that for the three types of robotic systems described in Fig. 7.1, Fig. 7.2 and Fig. 7.3, 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.

7.2.1 Fundamental Framework

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

$$A_i X B_i = Y C_i Z. \tag{7.1}$$

Using the shifting property of Dirac delta function, we have

$$(\delta_{A_i} * \delta_X * \delta_{B_i})(H) = \delta(B_i^{-1} X^{-1} A_i^{-1} H) \tag{7.2a}$$

$$(\delta_Y * \delta_{C_i} * \delta_Z)(H) = \delta(Z^{-1} C_i^{-1} Y^{-1} H). \tag{7.2b}$$

Using Eq. (2.14) and Eq. (7.1), the above two equations can be combined into a single equation as:

$$(\delta_{A_i} * \delta_X * \delta_{B_i})(H) = (\delta_Y * \delta_{C_i} * \delta_Z)(H). \tag{7.3}$$

The relationship still holds if we fix A_i to be a constant transformation A:

$$(\delta_A * \delta_X * \delta_{B_i})(H) = (\delta_Y * \delta_{C_i} * \delta_Z)(H). \tag{7.4}$$

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 $\{G_i\}$ as:

$$f_G(H) = \frac{1}{n}\sum_{i=1}^{n} \delta_{G_i}(H) \tag{7.5}$$

where $G \in \{A, B, C\}$. If we use the bi-linearity of convolution, add n instances of Eq. (7.4) and substitute Eq. (7.5) into the summation, the following equation can be achieved:

$$(\delta_A * \delta_X * f_B)(H) = (\delta_Y * f_C * \delta_Z)(H). \tag{7.6}$$

Then by employing Eq. (4.53a) twice, we get the mean equation of $AXB =$

YCZ as

$$M_A M_X M_B = M_Y M_C M_Z. \tag{7.7}$$

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. (7.7) then becomes

$$\boxed{AXM_B = YM_CZ} \tag{7.8}$$

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

$$\Sigma_{A*X*B} = Ad(B^{-1})Ad(X^{-1})\Sigma_A Ad^T(X^{-1})Ad^T(B^{-1}) + \Sigma_B \tag{7.9a}$$

$$= \Sigma_B \tag{7.9b}$$

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

$$\Sigma_{Y*C*Z} = Ad(Z^{-1})\Sigma_C Ad^T(Z^{-1}) \tag{7.10}$$

Therefore, by equating Eq. (7.9a) and Eq. (7.10), the covariance equation for $AXB = YCZ$ with A fixed becomes

$$\boxed{\Sigma_B = Ad(Z^{-1})\Sigma_C Ad^T(Z^{-1})} \tag{7.11}$$

In order to decompose Eq. (7.11) into sub-equations, define the covariance matrix as

$$\Sigma_H = \begin{pmatrix} \Sigma_H^1 & \Sigma_H^2 \\ \Sigma_H^3 & \Sigma_H^4 \end{pmatrix} \in \mathbb{R}^{6\times 6} \tag{7.12}$$

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. (7.12) into Eq. (7.11) and one gets the upper left block as

$$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z, \tag{7.13}$$

and the lower right block as

$$\Sigma_B^2 = R_Z^T \Sigma_C^1 R_Z U_Z^T + R_Z^T \Sigma_C^2 R_Z \tag{7.14}$$

where $W_{XB} := U_B R_B^T R_X^T + R_B^T U_X R_X^T$. For convenience, we call Eq. (7.13) the Sig-Rot equation and Eq. (7.14) 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. (7.13) contains only

R_X and R_Z, and Eq. (7.14) 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. (7.13) and Eq. (7.14) 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 it postmultiply by A on both sides of the equation, we have $XBZ^{-1}C^{-1}Y^{-1}A = \mathbb{I}$ 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 7.1. 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.

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

Table 7.1: The simplified Sig-Rot equations after fixing A, B or C

7.2.2 Two Frameworks for $AXB = YCZ$ Calibration

7.2.2.1 Framework 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 $A = A_{\mathrm{I}}$, datasets $\{B_{\mathrm{I}i}\}$ and $\{C_{\mathrm{I}j}\}$ can be measured where $i, j = 1, \cdots, n$. In addition, with the zero covariance constraints, $\Sigma_{A_{\mathrm{I}}} = \mathbb{O}$ and $\Sigma_{A_{\mathrm{I}}^{-1}} = R_{A_{\mathrm{I}}} \Sigma_A R_{A_{\mathrm{I}}}^T = \mathbb{O}$, we can simplify Eq. (7.13) to the form

$$\Sigma^1_{B_{\mathrm{I}}} = R_Z^T \Sigma^1_{C_{\mathrm{I}}} R_Z. \tag{7.15}$$

However, note that the zero constraint on $\Sigma_{A_{\mathrm{I}}}$ 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 7.1 respectively. When A is fixed to A_{I}, the right hand side of Rep.6, namely $Y^{-1}A_{\mathrm{I}}X$, becomes a single "point" on $SE(3)$, whereas both $C_{\mathrm{I}i}$ and $B_{\mathrm{I}j}^{-1}$ are PDFs on $SE(3)$. The corresponding convolution equation of Rep.6 becomes

$$(f_{C_{\mathrm{I}}} * \delta_Z * f_{B_{\mathrm{I}}^{-1}})(H) \approx (\delta_{Y^{-1}} * \delta_{A_{\mathrm{I}}} * \delta_X)(H). \tag{7.16}$$

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. The zero covariance constraint can only be applied to the Sig-Rot equation whose corresponding convolution equation satisfies the balanced-PDF constraint.

As shown in [92], Σ_B^1 and Σ_C^1 have the same eigenvalues due to the fact that Eq. (7.15) 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. (7.15), and we have

$$\Lambda = \underbrace{Q_B^T R_Z^T Q_C}_{\mathcal{Q}} \Lambda Q_C^T R_Z Q_B = \mathcal{Q} \Lambda \mathcal{Q}^T. \tag{7.17}$$

According to [92], the special structure of Eq. (7.17) gives four solutions for $\mathcal{Q}$. Thus, we also get four candidates of R_Z as:

$$\boxed{R_Z = Q_{C_{\text{I}}} \mathcal{Q} Q_{B_{\text{I}}}^T.} \tag{7.18}$$

For the translation part t_Z, Eq. (7.14) can be simplified as

$$\Sigma_{B_{\text{I}}}^2 = R_Z^T \Sigma_{C_{\text{I}}}^1 R_Z U_Z^T + R_Z^T \Sigma_{C_{\text{I}}}^2 R_Z \tag{7.19}$$

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

Similarly, when fixing $C \equiv C_{\text{II}}$, $\Sigma_{C_{\text{II}}} = \Sigma_{C_{\text{II}}^{-1}} = \mathbb{O}$, the Sig-Rot equation for Rep.6 (denoted as Sig-Rot.6) becomes

$$\Sigma_{B_{\text{II}}^{-1}}^1 = R_X^T \Sigma_{A_{\text{II}}}^1 R_X. \tag{7.20}$$

Recall that this leads to an equation with structure similar to Eq. (7.17) and so $\mathcal{Q}$ has four possibilities, and the four candidates of R_X can be calculated as

$$\boxed{R_X = Q_{A_{\text{II}}} \mathcal{Q} Q_{B_{\text{II}}^{-1}}^T} \tag{7.21}$$

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

to Sig-Rot.2 to get

$$R_C^T R_Y^T \Sigma_{A^{-1}}^1 R_Y R_C = R_{Z^{-1}}^T \Sigma_B^1 R_{Z^{-1}}, \tag{7.22}$$

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

$$R_Y = Q_{A^{-1}} \mathcal{Q} Q_B^T R_Z^T R_C^T. \tag{7.23}$$

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

$$\boxed{Y = A_{\mathrm{I}} X M_{B_{\mathrm{I}}} Z^{-1} M_{C_{\mathrm{I}}}^{-1}} \tag{7.24}$$

and

$$\boxed{Y = M_{A_{\mathrm{II}}} X M_{B_{\mathrm{II}}} Z^{-1} C_{\mathrm{II}}^{-1}} \tag{7.25}$$

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 $M_{L_{\mathrm{I}}} = A_{\mathrm{I}} X_i M_{B_{\mathrm{I}}}$, $M_{R_{\mathrm{I}}} = Y_j M_{C_{\mathrm{I}}} Z_k$, $M_{L_{\mathrm{II}}} = M_{A_{\mathrm{II}}} X_i M_{B_{\mathrm{II}}}$ and $M_{R_{\mathrm{II}}} = Y_j C_{\mathrm{II}} Z_k$.

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

$$\begin{aligned} \min \| \log^{\vee}(R^T_{M_{L_{\mathrm{I}}}} R_{M_{R_{\mathrm{I}}}})\|_2 + \| \log^{\vee}(R^T_{M_{L_{\mathrm{II}}}} R_{M_{R_{\mathrm{II}}}})\|_2 \\ w \cdot \|t_{M_{L_{\mathrm{I}}}} - t_{M_{R_{\mathrm{I}}}}\|_2 + w \cdot \|t_{M_{L_{\mathrm{II}}}} - t_{M_{R_{\mathrm{II}}}}\|_2 \end{aligned} \tag{7.26}$$

where $i = 1, \ldots, 4,\ j = 1, \ldots, 4, k = 1, \ldots, 32$ 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.

7.2.2.2 Framework 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: $A = A_{\mathrm{I}}$ with $\{B_{\mathrm{I}i}\}$ and $\{C_{\mathrm{I}j}\}$

$$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z. \tag{7.27}$$

Dataset II: $B = B_{\mathrm{II}}$ with $\{A_{\mathrm{II}i}\}$ and $\{C_{\mathrm{II}j}\}$

$$\Sigma_{C^{-1}}^1 = R_Y^T \Sigma_{A^{-1}}^1 R_Y. \tag{7.28}$$

Dataset III: $C = C_{\mathrm{III}}$ with $\{A_{\mathrm{III}i}\}$ and $\{B_{\mathrm{III}j}\}$

$$\Sigma_{B^{-1}}^1 = R_X^T \Sigma_A^1 R_X. \tag{7.29}$$

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 $M_{L_{\mathrm{III}}} = M_{A_{\mathrm{III}}} X_i M_{B_{\mathrm{III}}}$ and $M_{R_{\mathrm{III}}} = Y_j C_{\mathrm{III}} Z_k$, we can form the following objective function using all 3

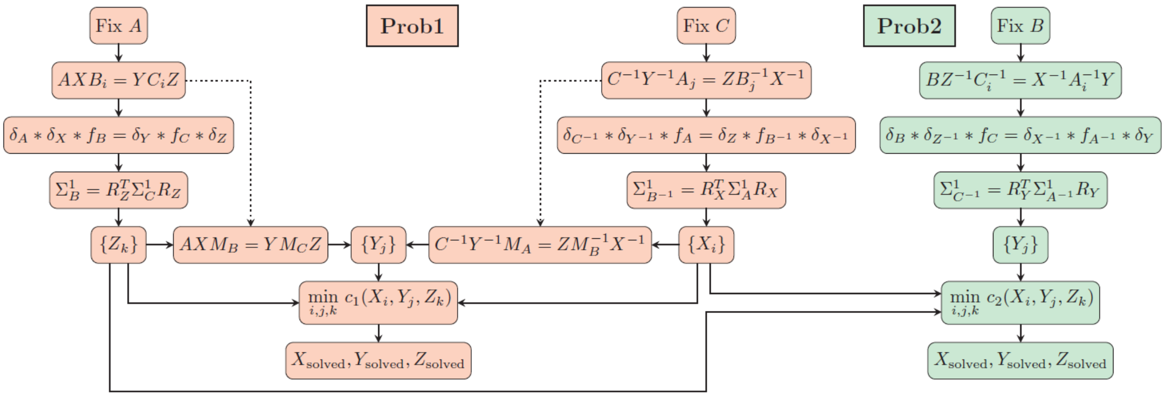

Fix A
Prob1
Fix C
Prob2
Fix B
$AXB_i = YC_iZ$
$C^{-1}Y^{-1}A_j = ZB_j^{-1}X^{-1}$
$BZ^{-1}C_i^{-1} = X^{-1}A_i^{-1}Y$
$\delta_A * \delta_X * f_B = \delta_Y * f_C * \delta_Z$
$\delta_{C^{-1}} * \delta_{Y^{-1}} * f_A = \delta_Z * f_{B^{-1}} * \delta_{X^{-1}}$
$\delta_B * \delta_{Z^{-1}} * f_C = \delta_{X^{-1}} * f_{A^{-1}} * \delta_Y$
$\Sigma_B^1 = R_Z^T \Sigma_C^1 R_Z$
$\Sigma_{B^{-1}}^1 = R_X^T \Sigma_A^1 R_X$
$\Sigma_{C^{-1}}^1 = R_Y^T \Sigma_{A^{-1}}^1 R_Y$
$\{Z_k\}$
$AXM_B = YM_CZ$
$\{Y_j\}$
$C^{-1}Y^{-1}M_A = ZM_B^{-1}X^{-1}$
$\{X_i\}$
$\{Y_j\}$
$\min_{i,j,k} c_1(X_i, Y_j, Z_k)$
$\min_{i,j,k} c_2(X_i, Y_j, Z_k)$
$X_{\text{solved}}, Y_{\text{solved}}, Z_{\text{solved}}$
$X_{\text{solved}}, Y_{\text{solved}}, Z_{\text{solved}}$

Figure 7.4: Two Probabilistic Frameworks for Multi-robot Calibration

datasets:

$$
\begin{aligned}
\min || \log^\vee (R_{M_{L_\mathrm{I}}}^T R_{M_{R_\mathrm{I}}})||_2 + || \log^\vee (R_{M_{L_\mathrm{II}}}^T R_{M_{R_\mathrm{II}}})||_2 \\
|| \log^\vee (R_{M_{L_\mathrm{III}}}^T R_{M_{R_\mathrm{III}}})||_2 + w \cdot ||t_{M_{L_\mathrm{I}}} - t_{M_{R_\mathrm{I}}}||_2 + \\
w \cdot ||t_{M_{L_\mathrm{II}}} - t_{M_{R_\mathrm{II}}}||_2 + w \cdot ||t_{M_{L_\mathrm{III}}} - t_{M_{R_\mathrm{III}}}||_2 \\
\text{where } i = 1, \cdots, 4,\ j = 1, \cdots, 4 \text{ and } k = 1, \cdots, 4.
\end{aligned}
\tag{7.30}
$$

A complete diagram illustrating both the calibration frameworks can be seen as in Fig. (7.4) where "Prob1" denotes the calibration problem for system 1 and "Prob2" denotes the calibration problem for system 2.

7.3 Numerical Simulation

In this section, we compared our probabilistic approaches numerically with the existing methods in the literature. For convenience, we called the probabilistic methods presented in Framework 1 and Framework 2, *Prob1* and *Prob2* respectively. In [89], two approaches were proposed for solving the $AXB = YCZ$ problem: one is called the *DK* method while the other is the *PN* method. In [24], a simultaneous $AXB = YCZ$ solver was introduced and we call it *Wang* in this chapter. Note that all of the three 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.

7.3.1 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 $A = A_{\mathrm{I}}$, and $\{B_{\mathrm{I}i}\}$ are given by

$$B_{\mathrm{I}i} = \exp(\hat{\delta}_i)B_{\mathrm{I}0} \tag{7.31a}$$

$$\delta_i \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \tag{7.31b}$$

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 $i = 1, 2, \ldots, 100$. The hat operator ^ converts a 6 by 1 vector into its corresponding Lie algebra in $se(3)$. Given the ground truth of X, Y and Z, $\{C_{\mathrm{I}i}\}$ is generated by

$$C_{\mathrm{I}i} = Y^{-1}A_{\mathrm{I}}XB_{\mathrm{I}i}Z^{-1}, \tag{7.32}$$

and we call this dataset I.

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

$$B_{\mathrm{II}i} = \exp(\hat{\delta}_i) B_{\mathrm{II}0} \tag{7.33a}$$

$$A_{\mathrm{II}i} = Y C_{\mathrm{II}} Z B_{\mathrm{II}i}^{-1} X^{-1} \tag{7.33b}$$

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

$$A_{\mathrm{III}i} = \exp(\hat{\delta}_i) A_{\mathrm{III}0} \tag{7.34a}$$

$$C_{\mathrm{III}i} = Y^{-1} A_{\mathrm{III}} X B_{\mathrm{II}i} Z^{-1}. \tag{7.34b}$$

In each dataset, the number of measurement data for A, B, C is 100, i.e. $i = 1, \ldots, 100$. 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 7.2. The recovered X, Y and Z were compared with the actual transformations using the following metrics for the errors in rotation and translation:

$$Error(R_H) = \| \log^{\vee}(R_{H_{\text{solved}}}^T R_{H_{\text{true}}}) \| \tag{7.35a}$$

$$Error(t_H) = \| t_{H_{\text{solved}}} - t_{H_{\text{true}}} \| / \| t_{H_{\text{true}}} \| \tag{7.35b}$$

where $H = X, Y, Z$.

	Dataset	*Prob1*	*Prob2*	*Wang*	*PN*	*DK*
I	$A_{\text{I}}, \{B_{\text{I}i}\}, \{C_{\text{I}i}\}$	✓	✓	✓	✓	✓
II	$C_{\text{II}}, \{B_{\text{II}i}\}, \{A_{\text{II}i}\}$	✓	✓	✓	✓	✓
III	$B_{\text{III}}, \{A_{\text{III}i}\}, \{C_{\text{III}i}\}$	×	✓	✓	✓	×

Table 7.2: Datasets used on each method

7.3.2 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. (7.6), Fig. (7.7) and Fig. (7.8).

First, given the three sets of $\{A\}$, $\{B\}$ and $\{C\}$, we scrambled $B_{\mathrm{I}i}$, $A_{\mathrm{II}i}$, $C_{\mathrm{III}i}$ up to a certain percentage r where $r = 0\%, 20\%, 40\%, 60\%, 80\%, 100\%$. 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. (7.5). 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 outstanding 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 with different values of σ_{data} where $\sigma_{\text{data}} = \{0.02, 0.04, 0.06, 0.08, 0.1\}$ and $r = 1\%$. As shown in Fig. (7.6), 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

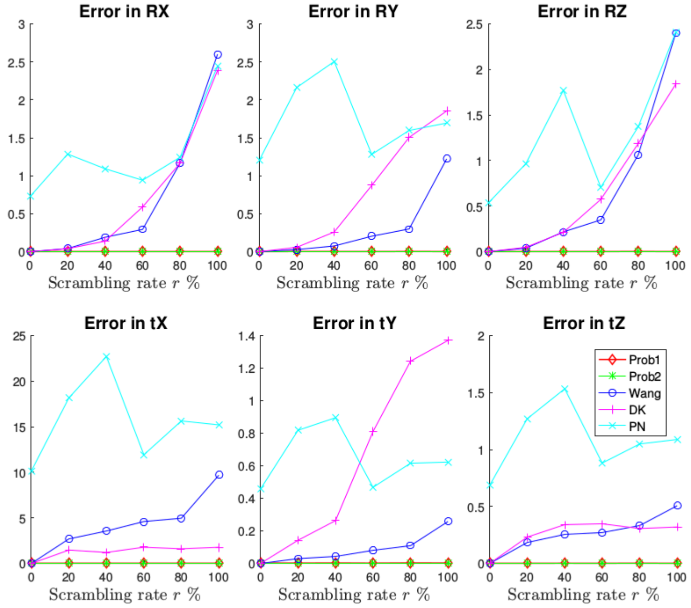


Figure 7.5: Rotation/Translation Errors in X, Y, Z v.s. scrambling rate for 10 trials and 100 measurements

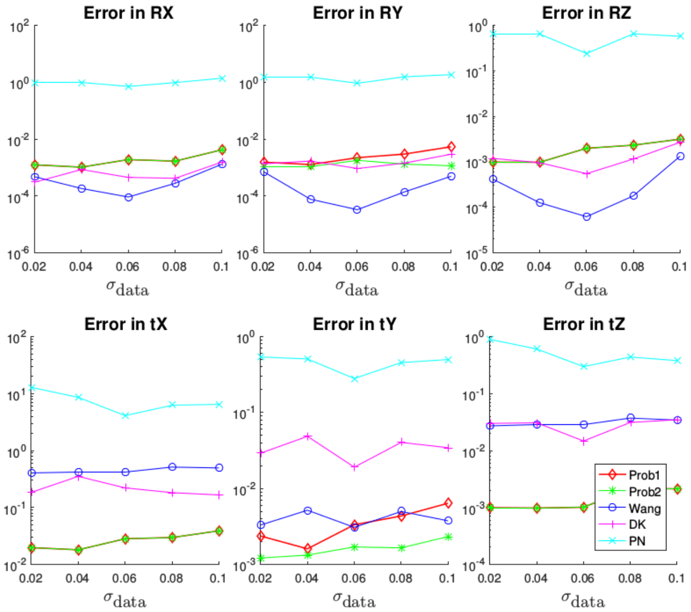


Figure 7.6: Rotation/Translation errors v.s. standard deviation of measurement data for $r = 1\%$ and $\sigma_{\text{noise}} = 0$

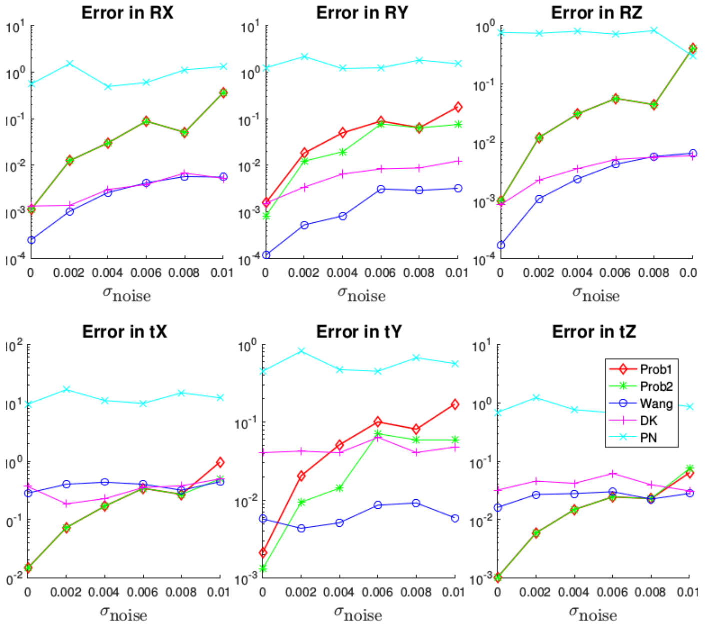


Figure 7.7: Rotation/Translation errors v.s. standard deviation of noise applied to the data for $r = 1\%$ and $\sigma_{\text{data}} = 0.02$

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. (7.7) and Fig. (7.8), we fixed the standard deviation for generating the datasets as $\sigma_{\text{data}} = 0.02$. For a homogeneous matrix H, we apply noise with

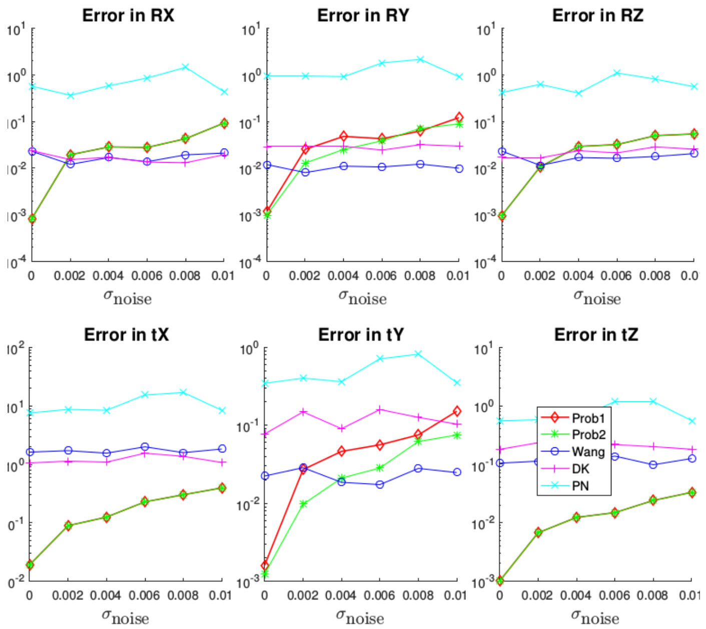


Figure 7.8: Rotation/Translation errors v.s. standard deviation of noise applied to the data for $r = 10\%$ and $\sigma_{\text{data}} = 0.02$

zero mean and standard deviation $\sigma_{\text{noise}} = \{0, 0.002, 0.004, 0.006, 0.008, 0.01\}$ to get a noisy $H_{\text{noise}} = H\exp(\hat{\delta})$, $\delta \in \mathcal{N}(\mathbf{0}; \Sigma)$, where the covariance matrix $\Sigma = \sigma_{\text{noise}}\mathbb{I}_6 \in \mathbb{R}^{6\times 6}$. The scrambling rate in Fig. (7.7) is $r = 1\%$ and the scrambling rate in Fig. (7.8) 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%, de-

spite 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. (7.8), when $r = 10\%$, the performance of the traditional methods only fluctuated within a small range despite the increasing noise.

7.4 A Hybrid Approach towards Handling Noise and Lack of Correspondence

7.4.1 Algorithm Formulation

As can be observed in Fig. 7.7 and Fig. 7.8, the two probabilistic approaches deteriorate quickly as the noises on the sensor measurements increase. This is reasonable because the nature of the probabilistic approaches demand that $\{A\}$, $\{B\}$ and $\{C\}$ be highly concentrated, and the applied noises are comparable to or 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 clattered 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 $i = 1, \ldots, n_s$, $j = \mathrm{I}, \mathrm{II}, \mathrm{III}$ 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:

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

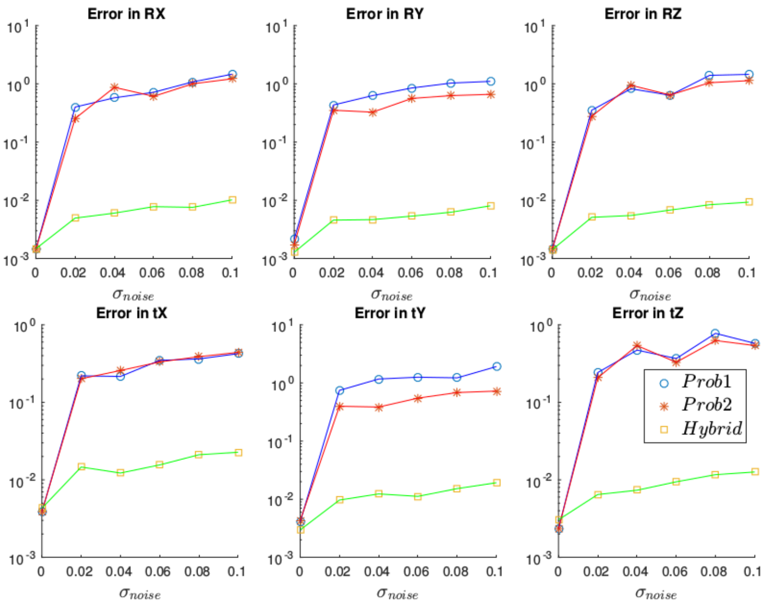


Figure 7.9: Rotation/Translation errors in X, Y, Z v.s. σ_{noise} for 20 trials and 50 measurements with $n_s = 4$ and $\sigma_{data} = 0.02$

where M_{Aij}, M_{Bij} and M_{Cij} are obtained as in Eq. (4.52a), 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.

7.4.2 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 com-

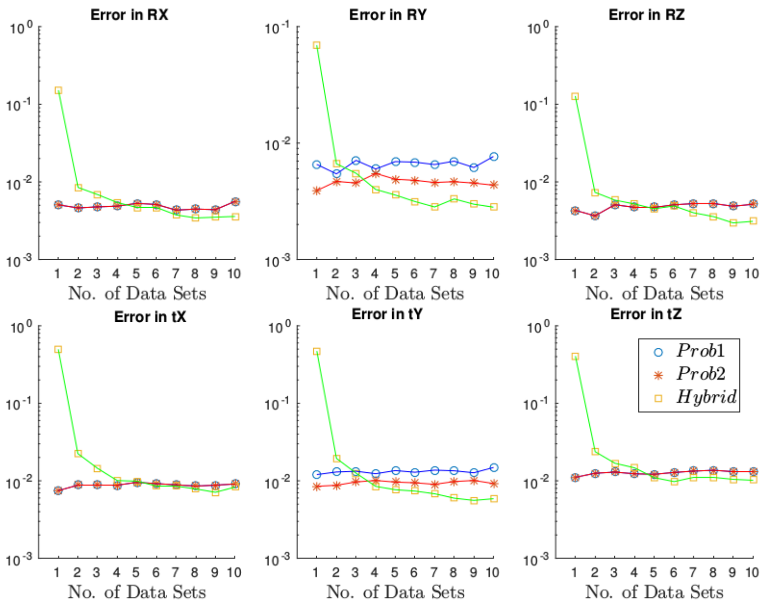


Figure 7.10: Rotation/Translation errors in X, Y, Z v.s. n_s for 20 trials and 50 measurements with $\sigma_{data} = 0.1$ and $r_{noise} = 0\%$

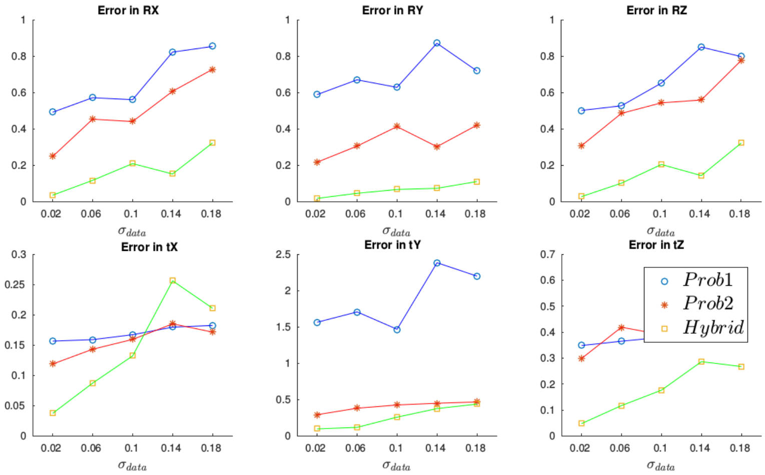


Figure 7.11: Rotation/Translation errors in X, Y, Z v.s. standard deviation of the original data cloud for 20 trials and 50 measurements with $r_{noise} = 1\%$ and $n_s = 3$

prehensive 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 7.3.1 except

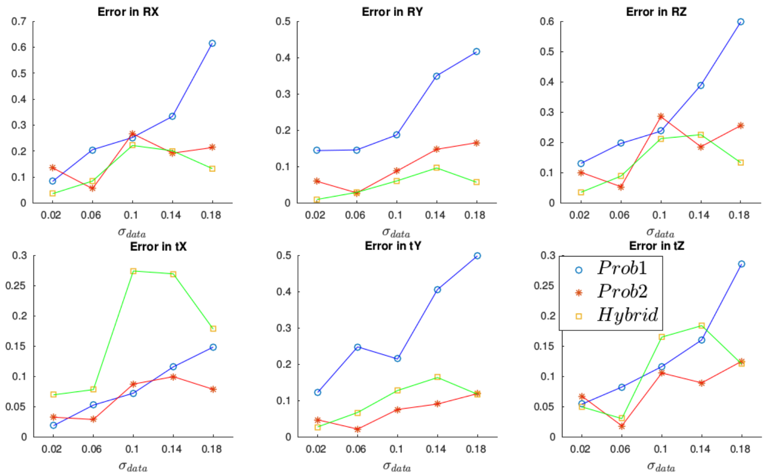


Figure 7.12: Rotation/Translation errors in X, Y, Z v.s. standard deviation of the original data cloud for 20 trials and 50 measurements with $r_{noise} = 0\%$ and $n_s = 3$

that a number of n_s clouds are needed instead of one for each case of motion constraint. In Eq. (7.37), B_{i10} denotes 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.

$$B_{i1j} = \exp(\hat{\delta}_j) B_{i10} \tag{7.37a}$$

$$\delta_j \in \mathcal{N}(\mathbf{0}; \Sigma) \subset \mathbb{R}^6 \tag{7.37b}$$

The numerical simulation results can be seen as in Fig. (7.9 - 7.12). Note that the y axes in Fig. (7.9) and Fig. (7.10) are in logarithm due to the large range of the errors. In Fig. 7.9, the noise level r is $r = \{0\%, 2\%...., 10\%\}$, 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. (7.10) 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 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. (7.11) and Fig. (7.12) showed the performances of the three solvers 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.

7.4.3 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. We think this is a very empty and open field and worth the further pursue of the community.

7.5 Conclusion

Motivated by problems that arise in multi-robot systems, in this chapter, 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 outstanding 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 perturbate the data a lot. However, by introducing multiple data clouds clattered around different centers, the influence of noise on the data is mitigated. It was 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.

Chapter 8

Conclusion

This dissertation demonstrated new methodologies in robotic assisted catheter tracking and robotic system calibration in general. The first part presented a novel robotic ultrasound system built on top of the middleware Robot Operating System. The system avoided the intensive computation in common catheter tracking systems by utilizing an embedded piezoelectric element. The element was attached at the tip of the catheter, and the vibration signal generated between the element and the ultrasound probe was converted into the relative position of the tip with respect to the ultrasound plane. Then a position feedback controller is employed to guide the robot arm to track the catheter tip in simulation, a water tank, and a customized multi-vein phantom. It is demonstrated that the system is able to track the catheter tip without any information of the ultrasound image. In the current setting, the catheter is inserted by a human being. In the more automated setup, one might want to use another robot arm to automate the catheter insertion procedure. In this case, a comprehensive and precise calibration of the whole system becomes very necessary. This inspired the second topic as robotic system calibration. First, a detailed review was given on the most classical robot hand-eye calibration problem. A major effort was put on unifying the mathematical notations across various $AX = XB$ solvers to reveal their interconnection. Then several probabilistic approaches were presented on $AX = XB$, $AX = YB$ and $AXB = YCZ$ calibration problems, which deal with the case where there is partial or no information of the prior knowledge of correspondence among sensor data. For the $AX = XB$ problem, new ways of calculating the mean of a set of $SE(3)$ matrices were proposed, which improved the accuracy of the probabilistic solver dramatically. For the $AX = YB$ problem, a probabilistic approach was derived in combination of group invariants to recover the correspondence of shifted data streams. It was illustrated that this approach can be used to facilitate existing non-probabilistic $AX = YB$ solvers. For the $AXB = YCZ$ problem, two probabilistic solvers were presented that can solve for X, Y and Z without a prior knowledge of the correspondence among sensor data. The new solvers were compared with three state-of-the-art non-probabilistic solvers and demonstrated their superior performance. In addition, to strike a balance between handling loss of correspondence information and the noise in the data, a hybrid approach was also derived, which was more robust to noise but required more sensor data gathering.

Bibliography

[1] J. M. Hollerbach and C. W. Wampler, “The calibration index and taxonomy for robot kinematic calibration methods,” *The International Journal of Robotics Research*, vol. 15, no. 6, pp. 573–591, 1996.

[2] C. G. Atkeson, C. H. An, and J. M. Hollerbach, “Estimation of inertial parameters of manipulator loads and links,” *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986.

[3] J. J. Craig, P. Hsu, and S. S. Sastry, “Adaptive control of mechanical manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 16–28, 1987.

[4] J.-J. E. Slotine and W. Li, “On the adaptive control of robot manipulators,” *The International Journal of Robotics Research*, vol. 6, no. 3, pp. 49–59, 1987.

[5] R. Ortega and M. W. Spong, “Adaptive motion control of rigid robots: A tutorial,” *Automatica*, vol. 25, no. 6, pp. 877–888, 1989.

[6] L. L. Whitcomb, A. A. Rizzi, and D. E. Koditschek, “Comparative experiments with a new adaptive controller for robot arms,” *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 59–70, 1993.

[7] R. Tsai and R. 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.

[8] E. Mair, M. Fleps, M. Suppa, and D. Burschka, “Spatio-temporal initialization for IMU to camera registration,” in *IEEE International Conference on Robotics and Biomimetics*. IEEE, 2011, pp. 557–564.

[9] J. Schmidt, F. Vogt, and H. Niemann, “Robust hand–eye calibration of an endoscopic surgery robot using dual quaternions,” in *Pattern Recognition*. Springer, 2003, pp. 548–556.

[10] H. Malm and A. Heyden, “A new approach to hand-eye calibration,” in *International Conference on Pattern Recognition*, vol. 1. IEEE, 2000, pp. 525–529.

[11] J. Heller, M. Havlena, and T. Pajdla, “A branch-and-bound algorithm for

globally optimal hand-eye calibration," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1608–1615.

[12] T. Ruland, T. Pajdla, and L. Kruger, "Globally optimal hand-eye calibration," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1035–1042.

[13] H. Wu, W. Tizzano, T. T. Andersen, N. A. Andersen, and O. Ravn, "Hand-eye calibration and inverse kinematics of robot arm using neural network," in *Robot Intelligence Technology and Applications 2*. Springer, 2014, pp. 581–591.

[14] S.-J. Kim, M. Jeong, J. Lee, J. Lee, K. Kim, B. You, and S. Oh, "Robot head-eye calibration using the minimum variance method," 2010, pp. 1446 – 1451.

[15] Y. Liu, Q. Wang, and Y. Li, "Calibration of a robot hand-eye system with a concentric circles target," in *International Bhurban Conference on Applied Sciences and Technology*. IEEE, 2015, pp. 204–209.

[16] Q. Ma, H. Li, and G. S. Chirikjian, "New probabilistic approaches to the AX = XB hand-eye calibration without correspondence," in *IEEE Interna-*

tional Conference on Robotics and Automation. IEEE, 2016, pp. 4365–4371.

[17] 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.

[18] 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.

[19] 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*, vol. 3. IEEE, 2001, pp. 2171–2176.

[20] A. Li, L. Wang, and D. Wu, "Simultaneous robot-world and hand-eye calibration using dual-quaternions and Kronecker product," *International Journal of Physical Science*, vol. 5, no. 10, pp. 1530–1536, 2010.

[21] F. Ernst, L. Richter, L. Matthäus, V. Martens, R. Bruder, A. Schlaefer, and A. Schweikard, "Non-orthogonal tool/flange and robot/world calibra-

tion," *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 8, no. 4, pp. 407–420, 2012.

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

[23] H. Li, Q. Ma, T. Wang, and G. S. Chirikjian, "Simultaneous hand-eye and robot-world calibration by solving the problem without correspondence," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 145–152, 2016.

[24] 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*. IEEE, 2014, pp. 410–415.

[25] Q. Ma, Z. Goh, and G. S. Chirikjian, "Probabilistic approaches to the AXB = YCZ calibration problem in multi-robot systems," in *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[26] G. S. Chirikjian and A. B. Kyatkin, *Engineering applications of non-commutative harmonic analysis: with emphasis on rotation and motion groups*. CRC Press, 2001.

[27] G. S. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[28] G. Chirikjian and A. Kyatkin, *Harmonic Analysis for Engineers and Applied Scientists*. Dover, 2016.

[29] Y. Wang and G. Chirikjian, “Nonparametric second-order theory of error propagation on motion groups,” *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1258–1273, 2008.

[30] C. for Disease Control, Prevention *et al.*, “National hospital discharge survey: 2010 table, procedures by selected patient characteristics–number by procedure category and age,” 2013.

[31] D. M. Bakalyar, M. D. Castellani, and R. D. Safian, “Radiation exposure to patients undergoing diagnostic and interventional cardiac catheterization procedures,” *Catheterization and Cardiovascular Diagnosis*, vol. 42, no. 2, pp. 121–125, 1997.

[32] M. Ingwersen, A. Drabik, U. Kulka, U. Oestreicher, S. Fricke, H. Krankenberg, C. Schwencke, and D. Mathey, “Physicians’ radiation exposure in the

catheterization lab: does the type of procedure matter?" *JACC: Cardiovascular Interventions*, vol. 6, no. 10, pp. 1095–1102, 2013.

[33] K. Chida, Y. Kaga, Y. Haga, N. Kataoka, E. Kumasaka, T. Meguro, and M. Zuguchi, "Occupational dose in interventional radiology procedures," *American Journal of Roentgenology*, vol. 200, no. 1, pp. 138–141, 2013.

[34] M. Whitby and C. Martin, "Radiation doses to the legs of radiologists performing interventional procedures: are they a cause for concern?" *The British Journal of Radiology*, 2014.

[35] H. Baysson, J. L. Réhel, Y. Boudjemline, J. Petit, B. Girodon, B. Aubert, D. Laurier, D. Bonnet, and M.-O. Bernier, "Risk of cancer associated with cardiac catheterization procedures during childhood: a cohort study in france," *BMC Public Health*, vol. 13, no. 1, p. 266, 2013.

[36] D. Onnasch, F. Schröder, G. Fischer, and H. Kramer, "Diagnostic reference levels and effective dose in paediatric cardiac catheterization," *The British Journal of Radiology*, 2014.

[37] J. Jayender, M. Azizian, and R. V. Patel, "Autonomous image-guided robot-assisted active catheter insertion," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 858–871, 2008.

[38] P. M. Loschak, A. Degirmenci, Y. Tenzer, C. M. Tschabrunn, E. Anter, and R. D. Howe, “A four degree of freedom robot for positioning ultrasound imaging catheters,” *Journal of mechanisms and robotics*, vol. 8, no. 5, p. 051016, 2016.

[39] H. Rafii-Tari, C. J. Payne, and G.-Z. Yang, “Current and emerging robot-assisted endovascular catheterization technologies: a review,” *Annals of biomedical engineering*, vol. 42, no. 4, p. 697, 2014.

[40] S. Condino, E. Calabrò, A. Alberti, S. Parrini, R. Cioni, R. Berchiolli, M. Gesi, V. Ferrari, and M. Ferrari, “Simultaneous tracking of catheters and guidewires: comparison to standard fluoroscopic guidance for arterial cannulation,” *European Journal of Vascular and Endovascular Surgery*, vol. 47, no. 1, pp. 53–60, 2014.

[41] B. Fuerst, E. E. Sutton, R. Ghotbi, N. J. Cowan, and N. Navab, “Bioelectric navigation: A new paradigm for intravascular device guidance,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2016, pp. 474–481.

[42] A. M. Priester, S. Natarajan, and M. O. Culjat, “Robotic ultrasound systems in medicine,” *IEEE Transactions on Utrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 3, pp. 507–523, 2013.

[43] S. B. Kesner and R. D. Howe, “Robotic catheter cardiac ablation combining ultrasound guidance and force control,” *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 631–644, 2014.

[44] N. Kucher, P. Boekstegers, O. Müller, C. Kupatt, J. Beyer-Westendorf, T. Heitzer, U. Tebbe, J. Horstkotte, R. Müller, E. Blessing *et al.*, “Randomized controlled trial of ultrasound-assisted catheter-directed thrombolysis for acute intermediate-risk pulmonary embolism,” *Circulation*, 2013.

[45] R. Mebarki, A. Krupa, and F. Chaumette, “2-d ultrasound probe complete guidance by visual servoing using image moments,” *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 296–306, 2010.

[46] X. Guo, H.-J. Kang, R. Etienne-Cummings, and E. M. Boctor, “Active ultrasound pattern injection system (auspis) for interventional tool guidance,” *PloS one*, vol. 9, no. 10, p. e104262, 2014.

[47] X. Guo, A. Cheng, H. K. Zhang, H.-J. Kang, R. Etienne-Cummings, and E. M. Boctor, “Active echo: a new paradigm for ultrasound calibration,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2014, pp. 397–404.

[48] R. O. System, “http://www.ros.org,” accessed: 2017-7-11.

[49] X. Guo, B. Tavakoli, H.-J. Kang, J. U. Kang, R. Etienne-Cummings, and E. M. Boctor, "Photoacoustic active ultrasound element for catheter tracking," in *SPIE BiOS*. International Society for Optics and Photonics, 2014, pp. 89 435M–89 435M.

[50] Y.-H. Kim, E. M. Marom, J. E. Herndon, and H. P. McAdams, "Pulmonary vein diameter, cross-sectional area, and shape: Ct analysis 1," *Radiology*, vol. 235, no. 1, pp. 43–49, 2005.

[51] U. Robots, "http://www.universal-robots.com," accessed: 2017-7-11.

[52] ROS-Industrial, "http://www.rosindustrial.org," accessed: 2017-7-11.

[53] e. a. Qianli Ma. Robotic ultrasound system for catheter tracking. Youtube. [Online]. Available: https://www.youtube.com/watch?v=BqTu4wx2sSk

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

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

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

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

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

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

[60] N. Andreff, R. Horaud, and B. Espiau, "On-line hand-eye calibration," in *International Conference on 3D Digital Imaging and Modeling*. IEEE, 1999, pp. 430–436.

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

[62] Z. Zhao, "Hand-eye calibration using convex optimization," in *IEEE In-*

ternational Conference on Robotics and Automation. IEEE, 2011, pp. 2947–2952.

[63] J. Angeles, G. Soucy, and F. P. Ferrie, "The online solution of the hand-eye problem," *IEEE Transactions on Robotics and Automation*, vol. 16, no. 6, pp. 720–731, 2000.

[64] M. K. Ackerman and G. S. Chirikjian, "A probabilistic solution to the AX = XB problem: Sensor calibration without correspondence," in *Geometric Science of Information.* Springer, 2013, pp. 693–701.

[65] M. K. Ackerman, A. Cheng, and G. Chirikjian, "An information-theoretic approach to the correspondence-free AX = XB sensor calibration problem," in *IEEE International Conference on Robotics and Automation.* IEEE, 2014, pp. 4893–4899.

[66] H. Chen, "A screw motion approach to uniqueness analysis of head-eye geometry," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 145–151, 1991.

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

[68] J. Chou and M. Kamel, “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.

[69] Y. Dai, J. Trumpf, H. Li, N. Barnes, and R. Hartley, “Rotation averaging with application to camera-rig calibration,” in *Asian Conference on Computer Vision*. Springer, 2009, pp. 335–346.

[70] M. Shah, R. Eastman, and T. Hong, “An overview of robot-sensor calibration methods for evaluation of perception systems,” in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*. ACM, 2012, pp. 15–20.

[71] S. Gwak, J. Kim, and F. C. Park, “Numerical optimization on the Euclidean group with applications to camera calibration,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 65–74, 2003.

[72] K. Daniilidis and E. Bayro-Corrochano, “The dual quaternion approach to hand-eye calibration,” in *International Conference on Pattern Recognitoin*, vol. 1. IEEE, 1996, pp. 318–322.

[73] N. Andreff, R. Horaud, and B. Espiau, “Robot hand-eye calibration using

structure-from-motion," *The International Journal of Robotics Research*, vol. 20, no. 3, pp. 228–248, 2001.

[74] Y. Seo, Y.-J. Choi, and S. W. Lee, "A branch-and-bound algorithm for globally optimal calibration of a camera-and-rotation-sensor system," in *International Conference on Computer Vision*. IEEE, 2009, pp. 1173–1178.

[75] J. Schmidt, F. Vogt, and H. Niemann, "Calibration–free hand–eye calibration: a structure–from–motion approach," in *Pattern Recognition*. Springer, 2005, pp. 67–74.

[76] M. K. Ackerman, A. Cheng, E. Boctor, and G. Chirikjian, "Online ultrasound sensor calibration using gradient descent on the Euclidean group," in *IEEE International Conference on Robotics and Automation*. IEEE, 2014, pp. 4900–4905.

[77] B. Horn, "Robot vision," 1986.

[78] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o (n) solution to the pnp problem," *International Journal of Computer Vision*, vol. 81, no. 2, pp. 155–166, 2009.

[79] J. S. F. Vogt and H. Niemann, "Vector quantization based data selection

for hand-eye calibration," in *Vision, Modeling, and Visualization*. IOS Press, 2004, pp. 21–28.

[80] J. Schmidt and H. Niemann, "Data selection for hand-eye calibration: a vector quantization approach," *The International Journal of Robotics Research*, vol. 27, no. 9, pp. 1027–1053, 2008.

[81] M. Ackerman, A. Cheng, B. Shiffman, E. Boctor, and G. S. Chirikjian, "Sensor calibration with unknown correspondence: Solving AX=XB using Euclidean-group invariants," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1308–1313.

[82] F. Shi, J. Wang, and Y. Liu, "An approach to improve online hand-eye calibration," in *Pattern Recognition and Image Analysis*. Springer, 2005, pp. 647–655.

[83] J. Zhang, F. Shi, and Y. Liu, "An adaptive selection of motion for online hand-eye calibration," in *Advances in Artificial Intelligence*. Springer, 2005, pp. 520–529.

[84] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006, pp. 4647–4653.

[85] 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.

[86] C. Belta and V. Kumar, "Euclidean metrics for motion generation on se (3)," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 216, no. 1, pp. 47–60, 2002.

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

[88] 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*. Springer, 2016, pp. 619–631.

[89] S. Yan, S.-K. Ong, and A. Y. Nee, "Registration of a hybrid robot using the degradation-Kronecker method and a purely nonlinear method," *Robotica*, vol. 34, no. 12, pp. 2729–2740, 2016.

[90] F. Aalamifar, D. Jiang, H. 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*. International Society for Optics and Photonics, 2015, pp. 94 190N–94 190N.

[91] 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*, vol. 32, no. 2, pp. 413–428, April 2016.

[92] M. K. Ackerman and G. S. Chirikjian, "A probabilistic solution to the AX = XB problem: Sensor calibration without correspondence," in *Geometric Science of Information*. Springer, 2013, pp. 693–701.

Vita

Qianli Ma was born in Harbin, China in 1989. He received a Bachelor in Mechanical Engineering from Tianjin University, China in 2012, and the Master degree in Robotics from Johns Hopkins University in 2014. He enrolled in the Mechanical Engineering Ph.D program at Johns Hopkins University in 2012.

Qianli Ma has served as the assistant to the editor-in-chief to the prestigious international journal of Robotica from 2014 to 2017. He has also been recognized with several distinctions including two National Scholarships at Tianjin University in 2009 and 2011; Cheng Family Fellowship at Johns Hopkins University in 2012; and the Best Conference Paper Finalist and Best Student Paper Finalist for the conference of Robotics: Science and Systems in 2016.