공학석사 학위논문

레이져 포인터를 이용한 Product-of-Exponentials 기반 직렬로봇 기구학적 보정 알고리즘

A Product-of-Exponentials Kinematic Calibration Algorithm for Serial Robots Using a Laser Pointer

2019년 8월

서울대학교 대학원

기계항공공학부

Pubest Detdee

ABSTRACT

A Product-of-Exponentials Kinematic Calibration Algorithm for Serial Robots Using a Laser Pointer

by

Pubest Detdee

Department of Mechanical and Aerospace Engineering
Seoul National University

This thesis proposes a kinematic calibration algorithm for serial robots based on a minimal product of exponentials (POE) forward kinematic model. Generally, robot calibration requires the measurement of the end-effector frame (position and orientation), which typically requires special measurement equipment. To avoid using complex measurement devices and to make the calibration easy to implement for even the most general serial robots, in our approach we attach a laser pointer to the end-effector, which is then aimed at a set of fixed known reference points in the plane. Treating the laser as a prismatic joint and the reference point as the

tip, kinematic calibration is then performed by minimizing the Cartesian position difference between the measured and estimated Cartesian tip position of the robot. Our method is validated via simulations and experiments involving a seven-dof industrial robot arm.

Keywords: Kinematic calibration, product of exponentials formula, laser pointer.

Student Number: 2017-24281

Contents

List of Tables

List of Figures

1

Introduction

Because of errors in the manufacturing and assembly process, the actual kinematic parameters of a robot usually deviate from its nominal values. While directly measuring a robot's kinematic parameters is difficult, one can instead measure the pose (*e.g.,* position and orientation) of the end-effector to estimate the actual kinematic parameter values; this process is referred to as kinematic calibration.

A typical kinematic calibration procedure begins by constructing a model of the forward kinematics of a robot, which can be expressed as a function of the form $x = f(\theta, p)$, where θ are the joint variables and p denote the kinematic parameters, and x represents the position and orientation of the end-effector. Calibration typically proceeds by linearizing the forward kinematics as

$$dx = \frac{\partial f(\theta, p)}{\partial \theta} d\theta + \frac{\partial f(\theta, p)}{\partial p} dp,$$

where the terms dx, $d\theta$, and dp can be viewed as errors between the actual and

predicted parameter values:

$$dx \triangleq x_{actual} - x_{predicted},$$
$$d\theta \triangleq \theta_{actual} - \theta_{input},$$
$$dp \triangleq p_{actual} - p_{nominal}.$$

Calibration involves taking measurements of the position and orientation of the end-effector in various configurations, and then determining the optimal $d\theta$ and dp that minimizes a suitable criterion. The most common criterion is a least-squares criterion of the form

$$\min_{dp,d\theta} \sum_{i=1}^{n} \left\| dx_i - \frac{\partial f(\theta_i, p)}{\partial \theta} d\theta - \frac{\partial f(\theta_i, p)}{\partial p} dp \right\|^2,$$

which is minimized with respect to dp and $d\theta$.

It should be noted that x here is a local coordinate representation of the end-effector position and orientation. More generally, an end-effector's position and orientation is represented as a homogeneous transformation matrix, also known as the Special Euclidean group of rigid-body motions, denoted SE(3).

1.1 Existing Methods

The Denavit-Hartenberg (D-H) parameters [4] are widely used for modeling the kinematics of a robot due to its minimal set of parameters in describing the robot kinematics. However, the D-H parameters are singular when neighboring joint axes are nearly parallel, which makes its error model discontinuous. Several modified D-H parameters have been proposed to solve the singularity problem, such as the Hayati model [5], the Veitschegger model [6], and CPC model [7]. These models contain redundant parameters that are defined in an *ad hoc* fashion, and obtaining

a closed-form expression for $\frac{\partial f}{\partial p}$ is not straightforward. It has further been pointed out that these redundant parameters may limit the application of these models to calibration [8].

Instead of focusing on the D-H parameters, Brockett [9] presents an alternative way to describe the forward kinematics based on a modern Lie group representation of classical screw theory, in which the forward kinematics can be expressed as a product of exponentials (POE). Unlike the D-H parameters, kinematic parameters in the POE formula vary smoothly with changes in the joint axes, leading to a singularity-free model. Okamura and Park [8] are the first to employ the POE formula to serial robot kinematic calibration. They present a closed-form set of equations for the linearization and also derive an iterative least-squares algorithm for calibration. A more explicit form of the linearization is presented in He *et al.* [10].

Generally, the calibration algorithm based on the POE formula requires the operations of normalization and orthogonalization, which is used to adjust the updated screw axes to satisfy the magnitude and pitch constraints. Yang *et al.* [3] propose a minimal POE-based model, eliminates these constraints, and also show that the identification process based on their model converges more rapidly than the original algorithm proposed in [8].

For the measurement of the errors of the end-effector pose, there are generally two methods. The open-loop method utilizes an external tool such as a ball-bar [11], laser tracking systems [12], or coordinate measurement machines (CMMs) [13] to measure the end-effector pose. On the other hand, the closed-loop method utilizes constraints on the end-effector and then uses the joint angle measurements alone to measure the errors of the end-effector pose. This closed-loop method is simpler to implement compared to the open-loop method.

Other approaches have imposed a physical constraint to the end-effector. For example, Zhuang *et al.* [14] use a plane constraint on the end-effector, while Meggiolaro *et al.* [15] impose a single endpoint contact constraint on the end-effector. However, such physical constraints lead to contact forces that can cause errors in the joint measurements.

In order to avoid the effect from the forces, some other researchers choose alternative processes such as adding visual tools or attaching sensors to the end-effector. Gong *et al.* [16] use an optical sensor mounted on the end-effector to measure the point on a calibration plate. Hu *et al.* [17] attach a laser pointer to the end-effector and use a stationary camera to observe the laser's position. Meng *et al.* [18] propose a vision-based measurement method by attaching a camera to the end-effector. However, these methods need prior calibration between the added tools and the end-effector, which makes the calibration process more complicated.

Gatla *et al.* [19] propose a virtual closed kinematic chain, in which a laser pointer attached to the end-effector aimed at a fixed point in a plane creates a virtual closed kinematic chain. In their algorithm, kinematic parameters of the laser pointer are also added into the error model. Hence, both of the kinematic parameters of the robot and the laser pointer can be calibrated at the same time.

1.2 Contributions of This Thesis

In this thesis, we propose a new kinematic calibration algorithm for serial robots that exploits the advantages of the minimal POE-based model [3] and the virtual closed kinematic chain approach described in [19]. The main concept of our approach is to attach a laser pointer to the end-effector, where the laser is then aimed at a set of fixed known reference points in the plane. Treating the laser as

a prismatic joint and the reference point as the tip, the forward kinematics is then constructed by using the minimal POE formula. This method is relatively easy for implementation and is suitable for any serial robots.

The thesis is organized as follows. In Chapter 2, we provide necessary background about Lie group formulations and a brief introduction of the POE formula, and the kinematic calibration based on the POE formula. In Chapter 3, we describe our kinematic calibration method. The experimental results and discussions are presented in Chapter 4. In Chapter 5, we summarize our method and also discuss the advantages and limitations.

2

Kinematics Preliminaries

In this Chapter, we provide the necessary geometric background for the robot kinematics based on Lie group formulations. A brief introduction about the POE formula, the minimal POE formula and the linearization of the forward kinematics equations are also included in this chapter.

2.1 Geometric Background

In this section, we describe the motion of a rigid body (*e.g.,* position, orientation), referring to Lie group formulations. Details of this section can be referred to [2].

2.1.1 The Lie Group Formulations

In the robotics literature, we use the Special Euclidean Group SE(3) or also known as the homogeneous transformation matrix, denoted T, to describe the orientation and position of a rigid body in three-dimensional space, where $T \in \mathrm{SE}(3)$ is the

set of all 4x4 real matrices of the form

$$\begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, \tag{2.1.1}$$

where $R \in \mathrm{SO}(3)$ and $p \in \mathbb{R}^3$ are represented the orientation and position of the rigid body, respectively. Here SO(3) refers to the group of 3x3 rotation matrices and is defined as

$$\mathrm{SO}(3) = \left\{ R \in \mathbb{R}^{3\times 3} \;\middle|\; R^T R = I, \;\; \det R = 1 \right\}. \tag{2.1.2}$$

It should be noted that the rotation of the rigid body can be described by the rotation around some unit axis $\omega \in \mathbb{R}^3$ (*i.e.,* $\|\omega\| = 1$) with some angle $\theta \in \mathbb{R}$. Thus, the rotation matrix $R \in \mathrm{SO}(3)$ can be expressed as

$$R = e^{[\omega]\theta} = I + \sin\theta[\omega] + (1 - \cos\theta)[\omega]^2, \tag{2.1.3}$$

where $[\omega] \in \mathrm{so}(3)$. Here so(3) is 3x3 skew-symmetric which is known as the Lie algebra of the Lie group SO(3), and is expressed as

$$[\omega] = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \tag{2.1.4}$$

Similarly, given a vector $\mathcal{S} = (\omega, v) \in \mathbb{R}^6$ and $\theta \in \mathbb{R}$, where $\omega \in \mathbb{R}^3$ is a unit vector (*i.e.,* $\|\omega\| = 1$) and $v \in \mathbb{R}^3$. The homogeneous transformation matrix $T \in \mathrm{SE}(3)$ can be described as

$$T = e^{[\mathcal{S}]\theta} = \begin{bmatrix} e^{[\omega]\theta} & G(\theta)v \\ 0 & 1 \end{bmatrix}, \tag{2.1.5}$$

where $[\mathcal{S}] \in \mathrm{se}(3)$ and $G(\theta)$ is given by

$$G(\theta) = I\theta + (1 - \cos\theta)[\omega] + (\theta - \sin\theta)[\omega]^2. \tag{2.1.6}$$

Here se(3) is the Lie algebra of the Lie group SE (3) and is expressed as

$$[\mathcal{S}] = \begin{bmatrix} [\omega] & v \\ 0 & 0 \end{bmatrix}, \tag{2.1.7}$$

where $[\omega] \in \mathrm{so}(3)$ and $v \in \mathbb{R}^3$.

2.1.2 Screw Motions

For serial robots consist of n joint and $n+1$ links, the rotation from the revolute joint or the translation from the prismatic joint can be described by the screw motion as

$$T = e^{[\mathcal{S}]\theta} \in \mathrm{SE}(3), \tag{2.1.8}$$

where $[\mathcal{S}] \in \mathrm{se}(3)$ and $\theta \in \mathbb{R}$ is a joint angle for a revolute joint or a translation distance for a prismatic joint. Here $\mathcal{S} \in \mathbb{R}^6$ denotes the screw axis.

For a given reference frame, the screw axis $\mathcal{S}$ can be expressed in the frame as

$$\mathcal{S} = \begin{bmatrix} \omega \\ v \end{bmatrix} \in \mathbb{R}^6, \tag{2.1.9}$$

where $\omega \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$.

To apply the screw motions to each joint in a robot, the screw axis has to satisfy the constraints below.

• **For a revolute joint**, $\omega \in \mathbb{R}^3$ is a unit vector in the positive direction of the joint axis (*i.e.,* positive rotation one defined in the right-hand sense, $\|\omega\| = 1$) and $v = -\omega \times q$, where $q \in \mathbb{R}^3$ is any point on the joint axis.

• **For a prismatic joint**, $\omega = 0$ and $v \in \mathbb{R}^3$ is a unit direction in a positive translation of the joint axis (*i.e.,* $\|v\| = 1$).

2.1.3 Adjoint Representation

We usually describe a vector in the space as expressed in some reference frame. Given a reference frame {a} and {b}, we denote the screw axis $\mathcal{S} \in \mathbb{R}^6$ as expressed in frame {a} and {b} as $\mathcal{S}_a$ and $\mathcal{S}_b$, respectively. The relationship between $\mathcal{S}_a$ and $\mathcal{S}_b$ can be expressed as

$$\mathcal{S}_b = [\mathrm{Ad}_{T_{ba}}]\mathcal{S}_a, \tag{2.1.10}$$

where $[\mathrm{Ad}_{T_{ba}}]$ is the adjoint representation and is expressed as

$$[\mathrm{Ad}_{T_{ba}}] = \begin{bmatrix} R_{ba} & 0 \\ [p_{ba}]R_{ba} & R_{ba} \end{bmatrix} \in \mathbb{R}^{6\times 6}, \tag{2.1.11}$$

where $R_{ba} \in \mathrm{SO}(3)$ and $p_{ba} \in \mathbb{R}^3$ are the orientation and position of frame {a} expressed in frame {b}.

2.2 Forward Kinematics

Forward kinematics refers to the use of kinematic parameters of a robot and its joint values to calculate the position and orientation of the end-effector frame. In this section, we describe the forward kinematics using the POE and the minimal POE formula. More details of these formulations can be found in [2], [3], respectively.

2.2.1 The Product of Exponentials Formula

To use the POE formula, only two coordinate frames need to be assigned. First, is a fixed reference frame {s} which is usually attached to the robot's base. Another one is an end-effector frame {b} which is attached to the end-effector of the robot.

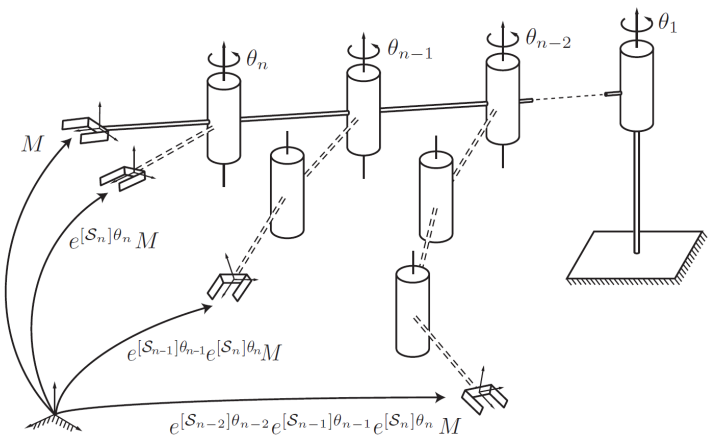


Figure 2.1: An n-dof serial robot [2].

Let $M \in \mathrm{SE}(3)$ denote the position and orientation of the end-effector frame relative to the fixed base frame when the robot is in its zero position (*i.e.*, all the joint variables are equal to zero).

Consider Figure 2.1; if we assume joint n is rotated with some angle θ_n, the end-effector frame {b} can be expressed as

$$T = e^{[\mathcal{S}_n]\theta_n} M, \tag{2.2.12}$$

where $T \in \mathrm{SE}(3)$ represents the new configuration of the end-effector frame and $\mathcal{S}_n$ is the screw axis of joint n as expressed in frame {s}.

Next, if we assume that joint $n-1$ is also rotated with some angle θ_{n-1}, the end-effector frame {b} can be then expressed as

$$T = e^{[\mathcal{S}_{n-1}]\theta_{n-1}} \left(e^{[\mathcal{S}_n]\theta_n} M \right). \tag{2.2.13}$$

Repeatedly applying these steps until the joint 1 is rotated with θ_1, the configuration of the end-effector frame can be expressed as

$$T(\theta) = e^{[\mathcal{S}_1]\theta_1} \cdots e^{[\mathcal{S}_{n-1}]\theta_{n-1}} e^{[\mathcal{S}_n]\theta_n} M. \tag{2.2.14}$$

Equation (2.2.14) is the product of exponentials formula describing the forward kinematics of an n-dof serial robot.

2.2.2 The Minimal Product of Exponentials Formula

During the identification process, all kinematic parameters have to be updated at the end of each iteration step. The screw axis of each joint in the POE formula has to be adjusted to satisfy the joint constraints. This may decrease the performance of the identification process [3]. According to [3], these constraints can be easily eliminated by considering the screw axis in the link frame which is quite similar to the D-H parameters. In this section, we will explain how to construct the link frame into each link and also describe the forward kinematics using the minimal POE formula. For more details, refer to [3].

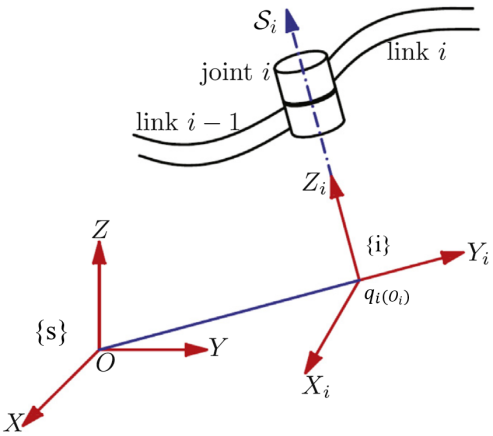


Figure 2.2: Illustration of a link frame {i} [3].

2.2.2.1 Link Frame Establishment

The concept of this method is to attach a link frame into each link and then consider the screw axis in its link frame, as shown in Figure 2.2. The link frame can be constructed by following the rules below.

For a revolute joint: given $\mathcal{S}_i = (\boldsymbol{\omega}_i, \boldsymbol{v}_i) \in \mathbb{R}^6$ is the screw axis of the joint i as expressed in the base frame {s}.

• The unit direction of the joint axis $\boldsymbol{\omega}_i$ is set as the z_i-axis of a link frame {i}.

• If the z_i-axis does not pass through the origin of the base frame (O) (see Figure 2.2), we choose the origin of the link frame {i}, denoted $\boldsymbol{q}_i$, to be the intersection point between the z_i-axis and a plane which is perpendicular to the z_i-axis and passes through O.

By setting the unit vector along q_i be the y_i-axis, the x_i-axis can be calculated by $-\boldsymbol{\omega}_i \times \boldsymbol{q}_i / \|\boldsymbol{q}_i\|$. This vector can be regarded as a unit vector along $\boldsymbol{v}_i$. Hence, the homogeneous transformation matrix of frame {i} relative to the base frame {s} is given by

$$T_{si} = \begin{bmatrix} \frac{\boldsymbol{v}_i}{\|\boldsymbol{v}_i\|} & \frac{\boldsymbol{q}_i}{\|\boldsymbol{q}_i\|} & \boldsymbol{\omega}_i & \boldsymbol{q}_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2.2.15}$$

• If the z_i-axis passes through the origin of the base frame (O) (see Figure 2.3), we set $\boldsymbol{q}_i$ be the same point as O. A link frame {i} can be then obtained by rotating frame {s} about $\boldsymbol{e}$-axis with an angle α, where $\boldsymbol{e} := [0,0,1]^T \times \boldsymbol{\omega}_i$ and $\alpha := \arccos([0,0,1]\boldsymbol{\omega}_i)$. Hence, T_{si} can be expressed as

$$T_{si} = \begin{bmatrix} I + \sin\alpha\,[\boldsymbol{e}] + (1-\cos\alpha)[\boldsymbol{e}]^2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{2.2.16}$$

For a prismatic joint: given $\mathcal{S}_i = (0, \boldsymbol{v}_i)$ is the screw axis of the joint i as expressed in the base frame {s}.

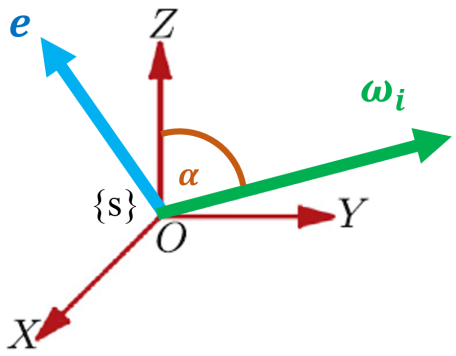


Figure 2.3: Illustration of the rotation of frame {s} [3].

- $\boldsymbol{v}_i$ is set as the z_i-axis of a link frame {i}.
- $\boldsymbol{q}_i$ is chosen at the base origin O. Then, T_{si} can be calculated in the same way as a revolute joint that the z_i-axis passes through O, but replacing $\boldsymbol{\omega}_i$ with $\boldsymbol{v}_i$.

2.2.2.2 Forward Kinematics Using Minimal POE Formula

Let $\mathcal{S}'_i$ be the screw axis of the joint i as expressed in the link frame {i}. Using the adjoint representation, the screw axis of joint i can be expressed in frame {s} as

$$\mathcal{S}_i = [\mathrm{Ad}_{T_{si}}]\mathcal{S}'_i. \tag{2.2.17}$$

According to [3], we can choose a set of independent parameters $\boldsymbol{\eta}_i = [v'_{xi}, v'_{yi}]^T$ and $\boldsymbol{\eta}_i = [\omega'_{xi}, \omega'_{yi}, q'_{xi}, q'_{yi}]^T$ to describe the screw axis $\mathcal{S}'_i$ of a prismatic joint and a revolute joint, respectively. Note that, according to the link frame establishment rules, the nominal parameters in $\boldsymbol{\eta}_i$ are all equal to zero.

Besides, it will be more convenient to write $\mathcal{S}_i$ in a function of $\boldsymbol{\eta}_i$ as follows.

For a revolute joint:

$$\mathcal{S}_i = \mathcal{S}_i(\boldsymbol{\eta}_i) = [\mathrm{Ad}_{T_{si}}]\mathcal{S}'_i = [\mathrm{Ad}_{T_{si}}] \begin{bmatrix} \begin{bmatrix} \omega'_{xi} \\ \omega'_{yi} \\ \sqrt{1-\omega'^2_{xi}-\omega'^2_{yi}} \end{bmatrix} \\ -\begin{bmatrix} \omega'_{xi} \\ \omega'_{yi} \\ \sqrt{1-\omega'^2_{xi}-\omega'^2_{yi}} \end{bmatrix} \times \begin{bmatrix} q'_{xi} \\ q'_{yi} \\ 0 \end{bmatrix} \end{bmatrix}. \quad (2.2.18)$$

For a prismatic joint:

$$\mathcal{S}_i = \mathcal{S}_i(\boldsymbol{\eta}_i) = [\mathrm{Ad}_{T_{si}}]\mathcal{S}'_i = [\mathrm{Ad}_{T_{si}}] \begin{bmatrix} 0 \\ 0 \\ 0 \\ v'_{xi} \\ v'_{yi} \\ \sqrt{1-v'^2_{xi}-v'^2_{yi}} \end{bmatrix}. \quad (2.2.19)$$

Substituting Equations (2.2.18) and (2.2.19) into (2.2.14). Hence, the forward kinematics of an n-dof serial robot can be expressed by using the minimal POE formula as

$$T(\theta) = e^{[\mathcal{S}_1(\eta_1)]\theta_1} \cdots e^{[\mathcal{S}_{n-1}(\eta_{n-1})]\theta_{n-1}} e^{[\mathcal{S}_n(\eta_n)]\theta_n} M. \quad (2.2.20)$$

2.3 Kinematic Error Model

In this Section, we describe the error model based on the linearization of the forward kinematics in Equation (2.2.20). It is noted that there is no necessity to obtain the error model, using this method. Some of the researchers also apply other methods to create their error models, *e.g.,* [19].

2.3.1 Linearizing the Forward Kinematics

Note that M can be expressed using the screw motion as $M = e^{[\mathcal{S}_M]}$, where $\mathcal{S}_M = (\omega_M, v_M) \in \mathbb{R}^6$ is denoted the initial screw axis, where $\omega_M \in \mathbb{R}^3$ and $v_M \in \mathbb{R}^3$. Equation (2.2.20) can be then rewritten as

$$T(\theta) = e^{[\mathcal{S}_1(\eta_1)]\theta_1} \cdots e^{[\mathcal{S}_{n-1}(\eta_{n-1})]\theta_{n-1}} e^{[\mathcal{S}_n(\eta_n)]\theta_n} e^{[\mathcal{S}_M]}. \tag{2.3.21}$$

By differentiating Equation (2.3.21) and right multiplying with T^{-1}, the error model can be expressed as

$$dT\,T^{-1} = \left(\frac{\partial T}{\partial \mathcal{S}} \frac{\partial \mathcal{S}}{\partial \boldsymbol{\eta}} d\boldsymbol{\eta} + \frac{\partial T}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} + \frac{\partial T}{\partial \mathcal{S}_M} d\mathcal{S}_M \right) T^{-1}. \tag{2.3.22}$$

First, we consider the left-hand side of Equation (2.3.22) $dT\,T^{-1} \in \mathrm{se}(3)$. Let T_a and T_n be the actual and nominal end-effector frames, respectively, where T_a is obtained from measurement data and T_n is calculated using the nominal parameter values. Then $dT\,T^{-1} = (T_a - T_n)T_n^{-1} = T_a T_n^{-1} - I$. If the deviation between T_a and T_n are sufficiently small, then $T_a T_n^{-1} = I + \boldsymbol{\beta} + \boldsymbol{\beta}^2/2! + \cdots$, where $\boldsymbol{\beta} = \log(T_a T_n^{-1})$. Then, with the first-order approximation, we have

$$dT\,T^{-1} = \boldsymbol{\beta} = \log(T_a T_n^{-1}). \tag{2.3.23}$$

By taking several measurements of the end-effector pose, the kinematic parameters can be then determined by

$$\min_{d\eta, d\theta, d\mathcal{S}_M} \left\| dT\,T^{-1} - \left(\frac{\partial T}{\partial \mathcal{S}} \frac{\partial \mathcal{S}}{\partial \boldsymbol{\eta}} d\boldsymbol{\eta} + \frac{\partial T}{\partial \boldsymbol{\theta}} d\boldsymbol{\theta} + \frac{\partial T}{\partial \mathcal{S}_M} d\mathcal{S}_M \right) T^{-1} \right\|^2. \tag{2.3.24}$$

Next, we consider the right-hand side of Equation (2.3.22). Here, we define the operator $\vee$ which maps : $\mathrm{se}(3) \to \mathbb{R}^6$. According to [10] and [3], the explicit form

of $dT\,T^{-1}$ is given by

$$\begin{aligned}\left[dT\,T^{-1}\right]^{\vee} =&[(de^{[\mathcal{S}_1]\theta_1})e^{[-\mathcal{S}_1]\theta_1}]^{\vee} + [\mathrm{Ad}_{e^{[\mathcal{S}_1]\theta_1}}][(de^{[\mathcal{S}_2]\theta_2})e^{[-\mathcal{S}_2]\theta_2}]^{\vee}\\ &+\cdots+[\mathrm{Ad}_{e^{[\mathcal{S}_1]\theta_1}\cdots e^{[\mathcal{S}_{n-1}]\theta_{n-1}}}][(de^{[\mathcal{S}_n]\theta_n})e^{[-\mathcal{S}_n]\theta_n}]^{\vee}\\ &+[\mathrm{Ad}_{e^{[\mathcal{S}_1]\theta_1}\cdots e^{[\mathcal{S}_n]\theta_n}}][(de^{[\mathcal{S}_M]})e^{[-\mathcal{S}_M]}]^{\vee},\end{aligned} \tag{2.3.25}$$

where

$$[(de^{[\mathcal{S}_i]\theta_i})e^{[-\mathcal{S}_i]\theta_i}]^{\vee} = \mathbf{A}_i\mathbf{B}_i d\boldsymbol{\eta}_i + \mathcal{S}_i d\boldsymbol{\theta}_i, \tag{2.3.26}$$

where for a revolute joint,

$$\begin{aligned}\mathbf{A}_i = \theta_i I_6 &+ \frac{4-\alpha_i\sin\alpha_i-4\cos\alpha_i}{2\|\boldsymbol{\omega}_i\|^2}\boldsymbol{\Omega}_i + \frac{4\alpha_i-5\sin\alpha_i+\alpha_i\cos\alpha_i}{2\|\boldsymbol{\omega}_i\|^3}{\boldsymbol{\Omega}_i}^2\\ &+\frac{2-\alpha_i\sin\alpha_i-2\cos\alpha_i}{2\|\boldsymbol{\omega}_i\|^4}{\boldsymbol{\Omega}_i}^3 + \frac{2\alpha_i-3\sin\alpha_i+\alpha_i\cos\alpha_i}{2\|\boldsymbol{\omega}_i\|^5}{\boldsymbol{\Omega}_i}^4,\end{aligned}$$

and

$$\boldsymbol{\Omega}_i = \begin{bmatrix}[\boldsymbol{\omega}_i] & 0\\ [\boldsymbol{v}_i] & [\omega_i]\end{bmatrix}, \|\boldsymbol{\omega}_i\| = (\omega_{xi}^2+\omega_{yi}^2+\omega_{zi}^2)^{1/2}, \alpha_i = \|\boldsymbol{\omega}_i\|\theta_i,$$

and

$$\mathbf{B}_i = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&1\\0&0&-1&0\\0&0&0&0\end{bmatrix},$$

and for a prismatic joint,

$$\mathbf{A}_i = \theta_i I_6,\ \mathbf{B}_i = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

Besides,

$$[(de^{[\mathcal{S}_M]})e^{[-\mathcal{S}_M]}]^{\vee} = \mathbf{A}_M d\mathcal{S}_M, \tag{2.3.27}$$

where

$$\begin{aligned}\mathbf{A}_M = I_6 &+ \frac{4-\alpha_M \sin\alpha_M - 4\cos\alpha_M}{2\alpha_M^2}\mathbf{\Omega}_M + \frac{4\alpha_M - 5\sin\alpha_M + \alpha_M\cos\alpha_M}{2\alpha_M^3}\mathbf{\Omega}_M^2 \\ &+ \frac{2-\alpha_M\sin\alpha_M - 2\cos\alpha_M}{2\alpha_M^4}\mathbf{\Omega}_M^3 + \frac{2\alpha_M - 3\sin\alpha_M + \alpha_M\cos\alpha_M}{2\alpha_M^5}\mathbf{\Omega}_M^4,\end{aligned}$$

and

$$\mathbf{\Omega}_M = \begin{bmatrix} [\boldsymbol{\omega}_M] & 0 \\ [\boldsymbol{v}_M] & [\boldsymbol{\omega}_M] \end{bmatrix}, \alpha_M = (\omega_{Mx}^2 + \omega_{My}^2 + \omega_{Mz}^2)^{1/2}.$$

For simplicity, letting

$$\begin{aligned}\mathbf{J}_i &= \begin{cases} [\mathbf{A}_1\mathbf{B}_1, \mathcal{S}_1] & \text{for } i = 1 \\ \left[\mathrm{Ad}_{\left(\prod_{k=1}^{i-1} e^{[\mathcal{S}_k]\theta_k}\right)}\right][\mathbf{A}_i\mathbf{B}_i, \mathcal{S}_i] & \text{for } 1 < i \le n \end{cases} \\ \mathbf{J}_M &= \left[\mathrm{Ad}_{\left(\prod_{k=1}^{n} e^{[\mathcal{S}_k]\theta_k}\right)}\right]\mathbf{A}_M.\end{aligned} \tag{2.3.28}$$

Hence, Equation (2.3.25) can be expressed as

$$\mathbf{y} = \mathbf{J}\mathbf{x}, \tag{2.3.29}$$

where

$$\mathbf{y} = [dT\, T^{-1}]^{\vee} \in \mathbb{R}^6, \tag{2.3.30}$$

$$\mathbf{J} = [\mathbf{J}_1, \mathbf{J}_2, \cdots, \mathbf{J}_n, \mathbf{J}_M] \in \mathbb{R}^{6\times(5r+3p+6)}, \tag{2.3.31}$$

$$\mathbf{x} = [d\boldsymbol{\eta}_1^T, d\boldsymbol{\theta}_1, \cdots, d\boldsymbol{\eta}_n^T, d\boldsymbol{\theta}_n, d\mathcal{S}_M^T]^T \in \mathbb{R}^{5r+3p+6}, \tag{2.3.32}$$

where r and p are the number of revolute joints and prismatic joints, respectively. Equation (2.3.29) is the error model for an n-dof serial robot based on the minimal POE formula.

3

Calibration Methodology

In this chapter, we describe our kinematic calibration in this thesis in detail. The chapter begins with the concept of our method, followed by the calibration algorithm.

3.1 The Concept of the Method

Our method merges the advantages of the minimal POE-based model [3] and the virtual closed kinematic chain approach described in [19]. The minimal POE-based model is used for representing the forward kinematics. The idea of using a laser pointer in a virtual closed kinematic chain approach is used for calibration. In this session, we will explain these processes in detail.

3.1.1 Forward Kinematics of a Robot With a Laser Pointer

The kinematic calibration process starts by constructing the forward kinematics. Figure 3.1 shows that the laser pointer attached to the end-effector is pointing to

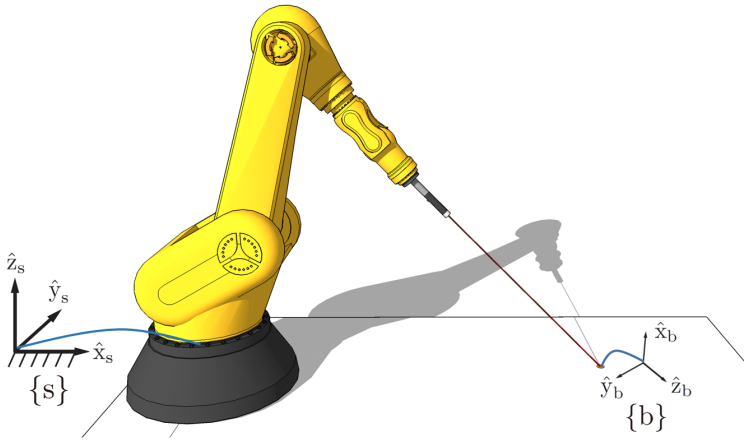


Figure 3.1: A fixed reference frame and an end-effector frame of the robot.

the fixed reference point in the plane. In order to identify the location of the laser spot in the plane, we attach a fixed reference frame {s} at the robot's base and attach an end-effector frame {b} at the fixed reference point. Treating the laser as a prismatic joint and the reference point as the tip, the forward kinematics for an n-dof serial robot can be expressed as

$$T(\theta) = e^{[\mathcal{S}_1(\eta_1)]\theta_1} \cdots e^{[\mathcal{S}_n(\eta_n)]\theta_n} e^{[\mathcal{S}_p(\eta_p)]\theta_p} M, \tag{3.1.1}$$

where $\theta_p \in \mathbb{R}$ is referred to the length of the laser and $\mathcal{S}_p = (0, v_p) \in \mathbb{R}^6$ denotes the screw axis of the laser (*i.e.,* a prismatic joint), where the nominal values of v_p is set as the same as the direction of the joint axis n. Here $T(\theta)$ is represented the orientation and the position of the laser spot in the plane.

3.1.2 The Error Model for Calibration

The error model in Section 2.3.1 requires both orientation and position data as input for calibration algorithm. However, our method can only measure the position

of the reference point. Therefore, we adopt the concept of the calibration model in [20], which need only the position data into our method.
For simplicity we denote

$$h(\theta) = e^{[\mathcal{S}_1(\eta_1)]\theta_1} \cdots e^{[\mathcal{S}_n(\eta_n)]\theta_n} e^{[\mathcal{S}_p(\eta_p)]\theta_p} = \begin{bmatrix} \boldsymbol{R}_h & \boldsymbol{p}_h \\ 0 & 1 \end{bmatrix}, \tag{3.1.2}$$

and $M \in \mathrm{SE}(3)$ is expressed as

$$M = \begin{bmatrix} \boldsymbol{R}_M & \boldsymbol{p}_M \\ 0 & 1 \end{bmatrix}, \tag{3.1.3}$$

where $\boldsymbol{R}_M$ and $\boldsymbol{p}_M$ are the orientation and position of the end-effector frame {b} expressed in frame {s}, respectively.

Let $\boldsymbol{p}_t$ be the tip's position as expressed in frame {s}. Using the forward kinematics in Equation (3.1.1), $\boldsymbol{p}_t$ is calculated by

$$\begin{aligned} \begin{bmatrix} \boldsymbol{p}_t \\ 1 \end{bmatrix} &= h \begin{bmatrix} \boldsymbol{R}_M & \boldsymbol{p}_M \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ &= h \begin{bmatrix} \boldsymbol{p}_M \\ 1 \end{bmatrix}. \end{aligned} \tag{3.1.4}$$

Differentiating (3.1.4):

$$\begin{aligned} \begin{bmatrix} d\boldsymbol{p}_t \\ 0 \end{bmatrix} &= (dh\, h^{-1}) h \begin{bmatrix} \boldsymbol{p}_M \\ 1 \end{bmatrix} + h \begin{bmatrix} d\boldsymbol{p}_M \\ 0 \end{bmatrix} \\ &= (dh\, h^{-1}) \begin{bmatrix} \boldsymbol{p}_t \\ 1 \end{bmatrix} + h \begin{bmatrix} d\boldsymbol{p}_M \\ 0 \end{bmatrix} \end{aligned} \tag{3.1.5}$$

$$i.e.,\ \ d\boldsymbol{p}_t = [\, -[\boldsymbol{p}_t], \boldsymbol{I}_3\,](dh\, h^{-1})^{\vee} + \boldsymbol{R}_h d\boldsymbol{p}_M, \tag{3.1.6}$$

where $[\boldsymbol{p}_t]$ is the skew-symmetric of $\boldsymbol{p}_t$. Here, $d\boldsymbol{p}_t \in \mathbb{R}^3$ is the Cartesian position difference between the actual and nominal tip's position. Note that in our approach, we treat the reference point as the tip. Hence, $d\boldsymbol{p}_t$ is expressed as

$$d\boldsymbol{p}_t = \boldsymbol{p}_t^a - \boldsymbol{p}_t^n \in \mathbb{R}^3, \tag{3.1.7}$$

where $\boldsymbol{p}_t^a$ is the measured position of the references point, and $\boldsymbol{p}_t^n$ is the calculated position of the references point using nominal parameter values.

According to [20, 10], it points out that $d\boldsymbol{\theta}$ and $d\boldsymbol{\mathcal{S}}_M$ cannot be identified in the same error model. In our approach, we assume that there are only $d\boldsymbol{\mathcal{S}}$ and $d\boldsymbol{\mathcal{S}}_M$ exist. Thus, $d\boldsymbol{\theta}$ can be excluded from the error model. Hence, $dh\,h^{-1}$ can be expressed as

$$dh\,h^{-1} = \left(\frac{\partial T}{\partial \mathcal{S}}\frac{\partial \mathcal{S}}{\partial \boldsymbol{\eta}}d\boldsymbol{\eta}\right)h^{-1}. \tag{3.1.8}$$

By converting $dh\,h^{-1}$ to an equivalent form as expressed in Section 2.3.1. The explicit form of $d\boldsymbol{p}_t$ can be expressed as

$$\mathbf{z} = \mathbf{K}_1 d\boldsymbol{\eta}_1 + \cdots + \mathbf{K}_n d\boldsymbol{\eta}_n + \mathbf{K}_p d\boldsymbol{\eta}_p + \mathbf{K}_M d\boldsymbol{p}_M, \tag{3.1.9}$$

where

$$\mathbf{z} = d\boldsymbol{p}_t = \boldsymbol{p}_t^a - \boldsymbol{p}_t^n, \tag{3.1.10}$$

$$\mathbf{K}_i = \begin{cases} [\,-[\boldsymbol{p}_t], \boldsymbol{I}_3\,][\mathbf{A}_1\mathbf{B}_1] & \text{for } i = 1 \\ [\,-[\boldsymbol{p}_t], \boldsymbol{I}_3\,]\Big[\,\mathrm{Ad}_{\left(\prod_{k=1}^{i-1} e^{[\mathcal{S}_k]\theta_k}\right)}\Big][\mathbf{A}_i\mathbf{B}_i] & \text{for } i = 2, 3, \cdots, n, p \end{cases} \tag{3.1.11}$$

$$\mathbf{K}_M = \boldsymbol{R}_h. \tag{3.1.12}$$

For simplicity, letting

$$\mathbf{K} = [\mathbf{K}_1, \cdots, \mathbf{K}_n, \mathbf{K}_p, \mathbf{K}_M] \in \mathbb{R}^{3\times(4r+2(p+1)+3)}, \tag{3.1.13}$$

$$\mathbf{x} = [d\boldsymbol{\eta}_1^T, \cdots, d\boldsymbol{\eta}_n^T, d\boldsymbol{\eta}_p^T, d\boldsymbol{p}_M^T]^T \in \mathbb{R}^{4r+2(p+1)+3}. \tag{3.1.14}$$

Hence, Equation (3.1.9) can be expressed as

$$\mathbf{z} = \mathbf{K}\mathbf{x}. \tag{3.1.15}$$

Calibration process is operated by aiming the laser at a fixed known reference point in the plane with several m configurations and combining the error vector $\mathbf{z}_i$ and the Jacobians $\mathbf{K}_i$ into a single equation:

$$\begin{bmatrix} \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_m \end{bmatrix} = \begin{bmatrix} \mathbf{K}_1 \\ \vdots \\ \mathbf{K}_m \end{bmatrix} \mathbf{x}, \tag{3.1.16}$$

or more compactly,

$$\tilde{\mathbf{z}} = \tilde{\mathbf{K}}\mathbf{x}. \tag{3.1.17}$$

The least-squares solution for $\mathbf{x}$ is

$$\mathbf{x} = (\tilde{\mathbf{K}}^T\tilde{\mathbf{K}})^{-1}\tilde{\mathbf{K}}^T\tilde{\mathbf{z}}. \tag{3.1.18}$$

The details about the identification process are discussed in the next section.

3.2 Calibration Algorithm

To solve $\mathbf{x}$ from the equation (3.1.18), we need all joint measurement data as input for calibration algorithm. $\theta_1 - \theta_n$ can be obtained directly from the controller. For θ_p, we may need an external tool or advance laser pointer to measure the length.

In order to simplify the calibration process, we propose the method to estimate the length of the laser in this section. We also describe in detail on the identification process for our calibration algorithm.

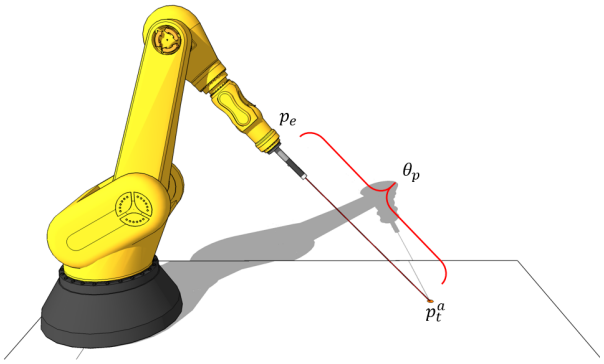


Figure 3.2: Estimation method of the length of the laser.

3.2.1 The Estimation Method of the Length of the Laser

According to Figure 3.2, the length of the laser (θ_p) can be estimated by calculating the distance between $\boldsymbol{p}_t^a$ and $\boldsymbol{p}_e$ as

$$\theta_p \approx \|\boldsymbol{p}_t^a - \boldsymbol{p}_e\|, \tag{3.2.19}$$

where

- $\boldsymbol{p}_t^a$ is the position of the reference point which can be obtained from the measurement.
- $\boldsymbol{p}_e$ is the position of the end-effector frame without the laser pointer (*i.e.*, $\theta_p = 0$) which can be calculated by the forward kinematics using $\theta_1 - \theta_n$.

However, $\boldsymbol{p}_e$ and θ_p are the unknown parameters in the calibration algorithm. After testing the idea, we find out that our algorithm cannot identify these two parameters at the same time. In order to solve this problem, the element of the end-effector position that follows the direction of the laser line has to be fixed as a constant during the identification process.

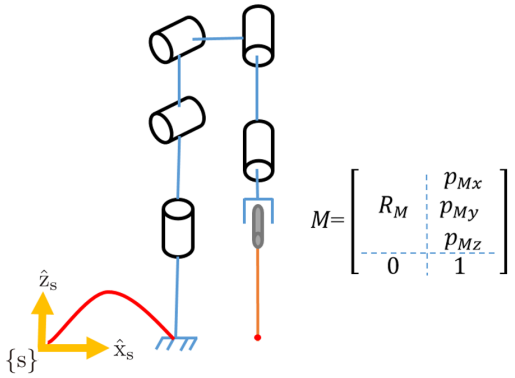


Figure 3.3: Robot at its zero position.

As shown in Figure 3.3, the direction of the laser line follows the z-axis when the robot is at zero position. In this case, the $\boldsymbol{p}_{Mz}$ has to be measured and fixed as a constant during the identification process.

3.2.2 Identification Process

By using the estimated length of the laser in Equation (3.2.19), $\mathbf{x}$ can be solved by the least-squares iteration, as shown in Figure 3.4.

At the end of each iteration step, the kinematic parameters are updated by $\mathbf{x}$. For the next iteration, T_{si} and also θ_p are recalculated according to the updated $\mathcal{S}_i$. $\boldsymbol{\eta}_i$ is reset to its nominal values which is all parameters in $\boldsymbol{\eta}_i$ are equal to zero. The iteration process is terminated when $\|\mathbf{x}\|$ becomes sufficiently small.

However, using the estimated length of the laser instead of its actual value may decrease the performance on the identification process.

Treating the error of the length of the laser line as a joint offset error ($d\boldsymbol{\theta}_p$). According to [20, 3], it points out that the joint offset errors ($d\boldsymbol{\theta}$) can be converted into $d\mathcal{S}$ and $d\mathcal{S}_M$ (see Appendix A). This means that the performance of

our algorithm depends on the accuracy of the estimated length of the laser.

It should be noted that this algorithm is just an estimation method. The more accurate measurement data are used, the more accurate results are obtained. In order to earn better accuracy result, the number of joint configurations should be equal or over the number of the identified parameters in the algorithm which is $4r + 2(p + 1) + 3$.

Additionally, the reference point locations in the plane should cover most of the robot's workspace. We suggest that the reference points should be located on the right, left, and front of the robot's workspace as shown in Figure 3.5.

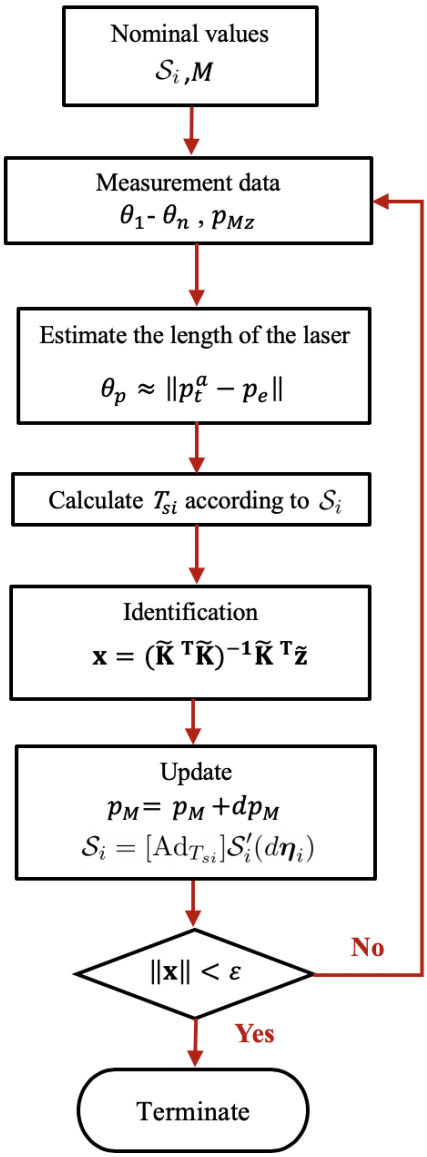


Figure 3.4: Calibration algorithm.

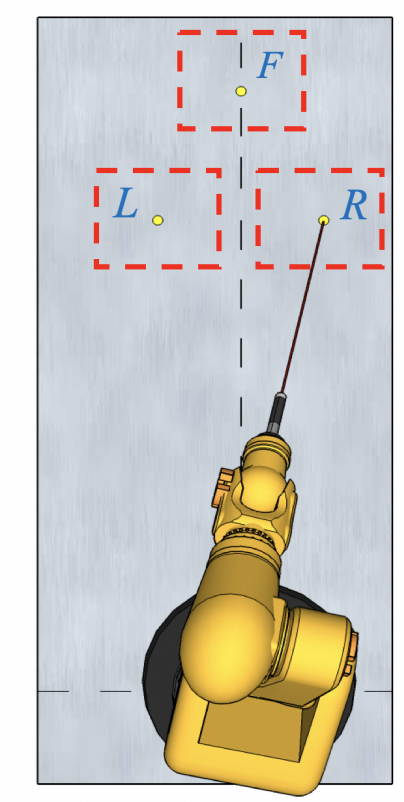


Figure 3.5: 3 Fixed point locations in a plane.

4

Experiments

4.1 Simulation 1: 6-Dof Robot With Precise Data

In order to verify our calibration algorithm, we calibrate a 6-dof Puma 560 robot to compare the robot accuracy before and after calibration. For this verifying process, we set the parameters identically to the second simulation in [1].

We also assume that there is no error in the joint offset ($d\boldsymbol{\theta}$) and the rotation part of the end-effector frame ($d\boldsymbol{R}_M$). The nominal and actual kinematic parameters of the robot can be referred to as Table 4.1.

In this simulation, we select 3 fixed reference points in the plane which are P1(400, 150, 0) mm, P2(500, 0, 0) mm, P3(400, -150, 0) mm with respect to the robot's base. The $\boldsymbol{p}_{Mz}$ is needed to be fixed constantly to its actual value. The actual kinematic parameters are used to calculate joint configurations in order to aim the laser at the fixed reference points. 30 joint configurations per each fixed point will be collected and used in the calibration. We aim to collect 90 joint configurations in total. The results are shown in Table 4.2.

Table 4.1: Kinematic parameters of Puma 560 robot (unit: mm)

	Nominal Values	**Actual Values**
$\mathcal{S}_1$	$[0, 0, 1, 0, 0, 0]^T$	$[0.04, -0.02, 0.999, 0.02, 0.04, 0]^T$
$\mathcal{S}_2$	$[0, -1, 0, 0, 0, 0]^T$	$[0, -1, 0, -0.02, 0, 0.05]^T$
$\mathcal{S}_3$	$[0, -1, 0, 0, 0, -100]^T$	$[0.178, -0.984, -0.01, -0.0842, 0.0874, -101]^T$
$\mathcal{S}_4$	$[0, 0, -1, -50, 250, 0]^T$	$[0.062, 0.013, -0.0998, -51, 249, 0.0752]^T$
$\mathcal{S}_5$	$[0, -1, 0, -20, 0, -250]^T$	$[0.001, -1, 0, -20.6, -0.0206, -249]^T$
$\mathcal{S}_6$	$[0, 0, -1, -50, 250, 0]^T$	$[0.095, 0.031, -0.0995, -51.27, 248.91, 2.86]^T$
$\mathcal{S}_p$	$[0, 0, 0, 0, 0, -1]^T$	$[0, 0, 0, 0.095, 0.031, -0.995]^T$
M	$\begin{bmatrix} 1 & 0 & 0 & 250 \\ 0 & 1 & 0 & 50 \\ 0 & 0 & 1 & -20 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 249 \\ 0 & 1 & 0 & 51 \\ 0 & 0 & 1 & -20.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

To test the performance of our algorithm, we need to compare the estimated length of the laser to its actual value in each configuration. Additionally, we also need to compare the absolute position errors of the end-effector frames from the calculation of the identified parameters and the actual parameters.

According to Figure 4.1, after 500 iterations, the results show only small differences between the estimated length of the laser to its actual value. They also show that the position errors of the end-effector frames between the identified parameters and the actual parameters are pretty much the same.

As there are no significant differences between the results of the end-effector

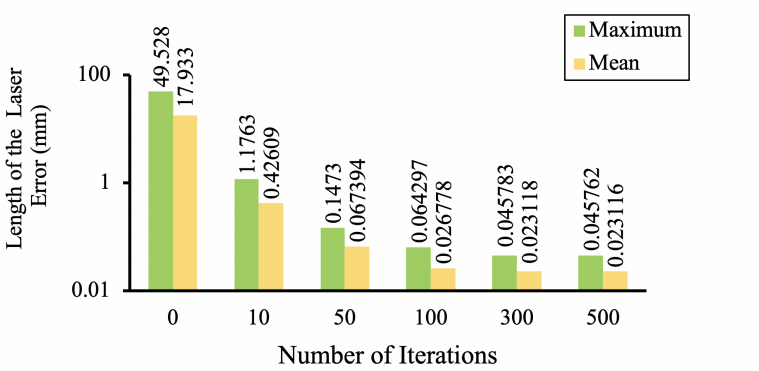
frame position with the actual and identified parameters, we can then use the estimated length of the laser line to calculate in our calibration algorithm.

Table 4.2: Identified parameters of Puma 560 robot (unit: mm)

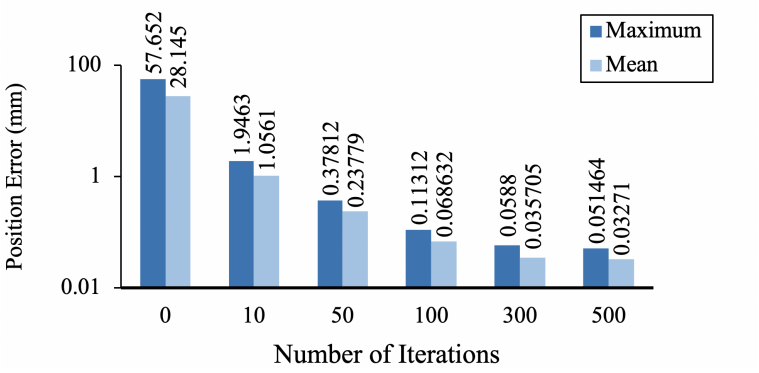
	Identified Values
$\mathcal{S}_1$	$[0.03996, -0.0199, 0.999, 0.04128, 0.0610, -0.00043]^T$
$\mathcal{S}_2$	$[0, -1, 0, -0.0243, 0, 0.0706]^T$
$\mathcal{S}_3$	$[0.17795, -0.98404, -0.000974, -0.0877, 0.0841, -101]^T$
$\mathcal{S}_4$	$[0.0620, 0.0129, -0.9979, -51.01, 249, 0.067884]^T$
$\mathcal{S}_5$	$[0.000949, -1, 0, -20.598, -0.02612, -249]^T$
$\mathcal{S}_6$	$[0.09502, 0.030983, -0.99499, -51.275, 248.92, 2.8542]^T$
$\mathcal{S}_p$	$[0, 0, 0, 0.095017, 0.030972, -0.99499]^T$
M	$\begin{bmatrix} 1 & 0 & 0 & 249 \\ 0 & 1 & 0 & 51.003 \\ 0 & 0 & 1 & -20.6 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

4.2 Simulation 2: 6-Dof Robot With Noisy Data

In reality, it will be quite difficult to manually aim the laser to point at the center of the fixed reference point precisely. In order to simulate that situation, we added Gaussian noise with a specific variance to the fixed reference points every time before calculating the joint configurations.



(a)



(b)

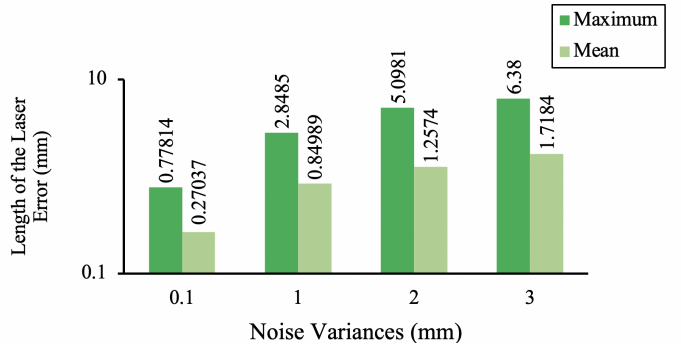
Figure 4.1: Experimental errors in simulation 1 with (a) length of the laser measurement and (b) position measurement.

In this section, we conduct 4 simulations with different noise variances (0.1 mm, 1 mm, 2 mm and 3 mm). We apply these 4 simulations to our calibration algorithm and all the cases successfully converged.

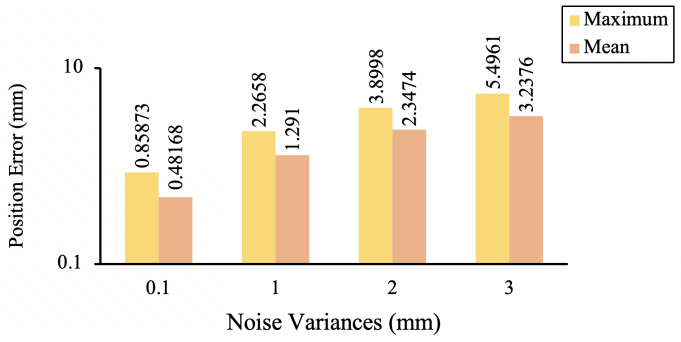
We also test this with other 100 joint configurations with the same process in

the previous session to ensure our calibration algorithm is working properly.

According to Figure 4.2, we find that the added noises also affect the error of the estimated length of the laser and the end-effector frame position. However, the average of the errors in the length of the laser and also the end-effector frame position are not significantly different when compared to the added noise. This means that even though there is some noise on the measurement process, this method still provides a relatively good performance.



(a)



(b)

Figure 4.2: Experimental errors in simulation 2 with (a) length of the laser measurement and (b) position measurement.

4.3 Experiments on a 7-Dof Robot

In this experiment, we calibrate a 7-dof Panda robot with our method as follows:

(i) Attach a laser pointer to the end-effector and ensure that it is aligned with the joint axis 7 as much as possible.

(ii) Measure the position of $\boldsymbol{p}_{Mz}$ at the zero position.

(iii) Select 3 fixed reference points in the plane; P1(521, 200, -8) mm, P2(436, 0, -8) mm, P3(521, -200, -8) mm with respect to the robot's base.

(iv) Aim the laser at the fixed reference points with 15 configurations per fixed point and collect their joint configurations. (In case the robot is not applicable for manual control by the operator, the teach-pendant process is needed for this step).

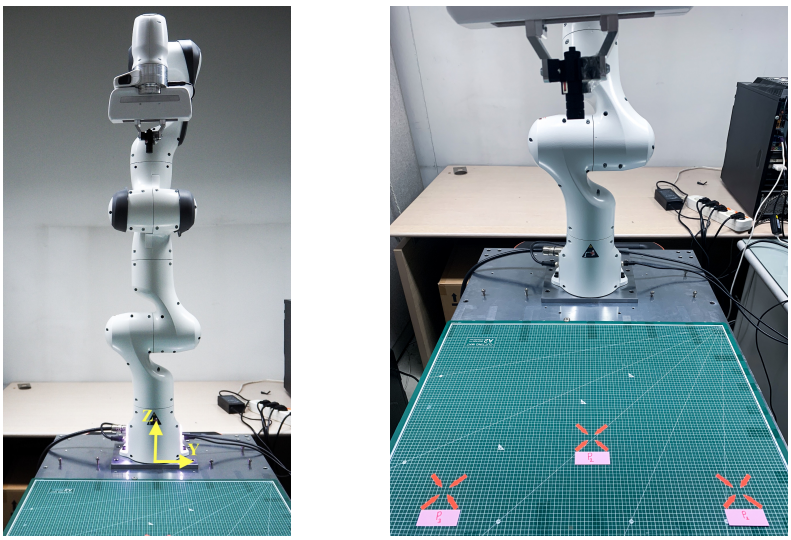
The processes are shown in Figure 4.3. The results of this experiment are also shown in Table 4.3.

To verify the accuracy of the robot, we randomly pick P4(441, 160, -8) mm as a fixed reference point in the plane. We calculate 60 joint configurations from the nominal parameters and other 60 joint configurations from the identified parameters.

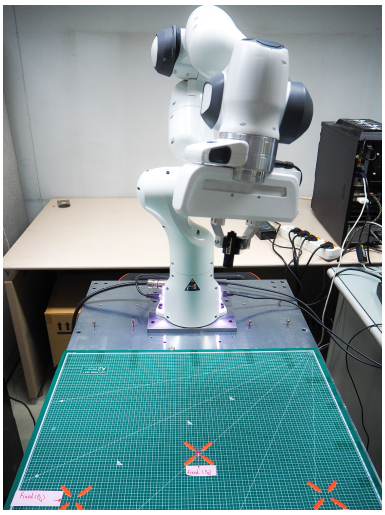
Although all configurations are calculated to aim the laser at the same fixed point in the plane, the results show that the laser spots are landed in various locations in the plane. We take top-view photos of each laser spot and input them to the software called "ImageJ" to estimate the locations of these laser spots. The processes are shown in Figure 4.4. The estimated locations of the laser spots are plotted in MATLAB to demonstrate the scattering of these lasers. The results are shown in Figure 4.5.

After calibration, the mean of the radius of laser scattering is improved from 6.86 mm to 1.11 mm while the standard deviation is shifted from 5.016 mm to 1.00 mm.

We also calculate those laser positions with a fixed point to identify the absolute position error. We find that the maximum error is improved from 26.993 to 4.92 mm. The mean error is also shifted from 9.76 to 1.057 mm. The results are shown in Figure 4.6.



(a) (b)



(c)

Figure 4.3: Measurement process: (a) robot with a laser pointer at zero position ($\boldsymbol{p}_{Mz} = 825$ mm), (b) 3 fixed reference point locations in the plane, and (c) aiming the laser at the fixed reference point.

Table 4.3: Kinematic parameters of 7-dof Panda robot (unit: mm)

	Nominal Values	**Identified Values**
$\mathcal{S}_1$	$[0, 0, 1, 0, 0, 0]^T$	$[0.0046, 0.00126, 0.9999, 0.54439, -0.1662, -0.00229]^T$
$\mathcal{S}_2$	$[0, 1, 0, -333, 0, 0]^T$	$[-0.00768, 0.9999, 0.00027, -334.33, -2.5714, 2.1275]^T$
$\mathcal{S}_3$	$[0, 0, 1, 0, 0, 0]^T$	$[0.000705, 0.00132, 1, -0.20847, -0.42849, 0.00071403]^T$
$\mathcal{S}_4$	$[0, -1, 0, 649, 0, -82.5]^T$	$[0.01345, -0.99991, 0.001986, 651.46, 8.5964, -85.454]^T$
$\mathcal{S}_5$	$[0, 0, 1, 0, 0, 0]^T$	$[-0.00128, 0.00278, 1, -1.4658, -6.2373, 0.015641]^T$
$\mathcal{S}_6$	$[0, -1, 0, 1033, 0, 0]^T$	$[0.017924, -0.99982, 0.005839, 1032.7, 18.492, -3.8108]^T$
$\mathcal{S}_7$	$[0, 0, -1, 0, 88, 0]^T$	$[-0.008532, -0.001502, -0.9999, -0.591, 82.83, -0.1194^T$
$\mathcal{S}_p$	$[0, 0, 0, 0, 0, -1]^T$	$[0, 0, 0, 0.2478, 0.006871, -0.99967]^T$
M	$\begin{bmatrix} 1 & 0 & 0 & 88 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 825 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 1 & 0 & 0 & 89.506 \\ 0 & 1 & 0 & 0.19809 \\ 0 & 0 & 1 & 825 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

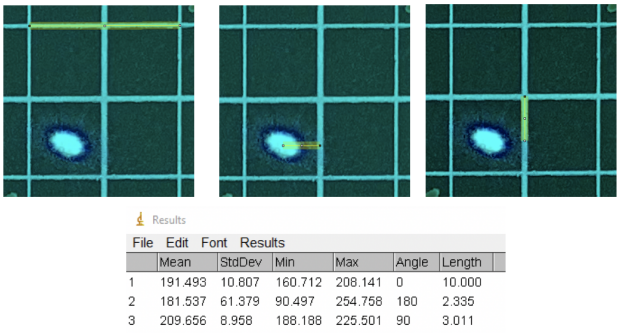


Figure 4.4: Estimated location of the laser spot using ImageJ software.

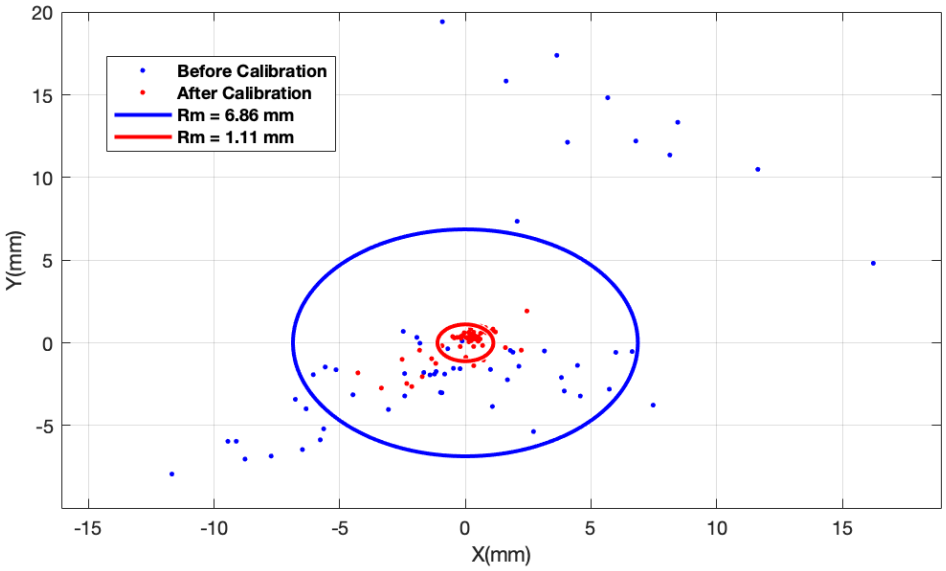


Figure 4.5: The scattering of the laser spot in the plane.

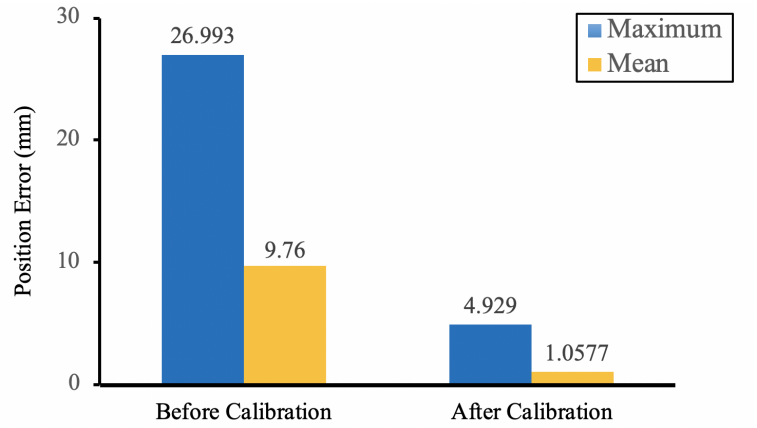


Figure 4.6: Maximum and mean position errors of the laser spot.

5

Conclusion

This thesis proposes a new way of a kinematic calibration algorithm for serial robots that merges the advantages of the minimal POE-based model and the approach of the virtual closed kinematic chain method. To calibrate a robot with our method, we attached a laser pointer to the end-effector. The pointer is set to aim at three fixed known reference points in the plane with various configurations. The results of the joint configurations are applied to our calibration algorithm. Treating the laser as a prismatic joint and the reference point as the tip, we can identify the optimal kinematic parameters by minimizing the Cartesian position difference between the measured and estimated Cartesian tip position of the robot.

The simulations result in chapter 4 also shows that we can use the estimated length of the laser in our calibration algorithm. Additionally, after testing our calibration algorithm with the real robot, the experimental results demonstrate the significant improvement of the robot's accuracy. However, this method may contain some errors such as the human-errors and the errors from the estimated length of the laser line. These errors may affect the performances of our method.

Accordingly to the experiment with the real robot in chapter 4, we find that the robot's position accuracy rate after calibration is ranged between 1-5 mm. As a result, if the robots have the position accuracy rate under 5 mm, they may not receive significant accuracy improvement with this calibration method.

The kinematic calibration algorithm in this thesis is mainly focused on the errors that are caused by geometrical effects. The errors from non-geometrical effects discussed in [21] such as gear transmission, backlash, and joint compliance can also be integrated into our algorithm to deliver more accurate results.

In conclusion, despite some minor errors, our method is easy to implement as the process only requires the use of a laser pointer and a simple observation plane to operate. Consequently, our method is also considered as a cost-efficient way to deliver effective results for robots' performance accuracy.

A

Appendix

A.1 Conversion From dq to $d\mathcal{S}$ and $d\mathcal{S}_M$ [1]

Assume there exists the joint offset errors for each joint in the forward kinematics, denoted $d\theta_i$. Then Equation (2.2.14) can be written as

$$\begin{aligned} T(\theta) &= e^{[\mathcal{S}_1](\theta_1+d\theta_1)}e^{[\mathcal{S}_2](\theta_2+d\theta_2)}\cdots e^{[\mathcal{S}_n](\theta_n+d\theta_n)}e^{[\mathcal{S}_M]} \\ &= e^{[\mathcal{S}_1]\theta_1}e^{[\mathcal{S}_1]d\theta_1}e^{[\mathcal{S}_2]\theta_2}e^{[\mathcal{S}_2]d\theta_2}\cdots e^{[\mathcal{S}_n]\theta_n}e^{[\mathcal{S}_n]d\theta_n}e^{[\mathcal{S}_M]}. \end{aligned} \tag{A.1.1}$$

According to the fact that if $M \in SE(3)$ and $p \in se(3)$, then $Me^pM^{-1} = e^{MpM^{-1}}$ which can be regarded as the identity $Me^p = e^{MpM^{-1}}M$. Therefore, Equation (A.1.1) can be rewritten as

$$T(\theta) = e^{[\mathcal{S}]_1\theta_1}e^{(e^{[\mathcal{S}_1]d\theta_1}[\mathcal{S}_2]e^{-[\mathcal{S}_1]d\theta_1})\theta_2}e^{[\mathcal{S}_1]d\theta_1}e^{[\mathcal{S}_2]d\theta_2}\cdots e^{[\mathcal{S}_n]\theta_n}e^{[\mathcal{S}_n]d\theta_n}e^{[\mathcal{S}_M]}. \tag{A.1.2}$$

By repeatedly applying the identity $Me^p = e^{MpM^{-1}}M$ and using adjoint representation, we have

$$T(\theta) = e^{[\mathcal{S}'_1]\theta_1}e^{[\mathcal{S}'_2]\theta_2}\cdots e^{[\mathcal{S}'_n]\theta_n}e^{[\mathcal{S}'_M]}, \tag{A.1.3}$$

where

$$[\mathcal{S}_i'] = \begin{cases} [\mathcal{S}_1] & \text{for } i = 1 \\ \left[\left[\operatorname{Ad}_{\left(\prod_{k=1}^{i-1} e^{[\mathcal{S}_k]\theta_k}\right)}\right]\mathcal{S}_i\right] & \text{for } 1 < i \leq n \end{cases} \tag{A.1.4}$$

$$[\mathcal{S}_M'] = \log\left(\left(\prod_{k=1}^{n} e^{[\mathcal{S}_k]d\theta_k}\right) e^{[\mathcal{S}_M]}\right). \tag{A.1.5}$$

If we denote $\mathcal{S}_i' = \mathcal{S}_i + d\mathcal{S}_i$ and $\mathcal{S}_M' = \mathcal{S}_M + d\mathcal{S}_M$, then Equation (A.1.1) is equivalent to

$$T(\theta) = e^{[\mathcal{S}_1+d\mathcal{S}_1]\theta_1} e^{[\mathcal{S}_2+d\mathcal{S}_2]\theta_2} \cdots e^{[\mathcal{S}_n+d\mathcal{S}_n]\theta_n} e^{[\mathcal{S}_M+d\mathcal{S}_M]}. \tag{A.1.6}$$

Bibliography

[1] Liao Wu, Xiangdong Yang, Ken Chen, and Hongliang Ren. A minimal poe-based model for robotic kinematic calibration with only position measurements. *IEEE Transactions on Automation Science and Engineering*, 12(2):758–763, 2014.

[2] Kevin M.. Lynch and Frank Chongwoo Park. *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press, 2017.

[3] Xiangdong Yang, Liao Wu, Jinquan Li, and Ken Chen. A minimal kinematic model for serial robot calibration using poe formula. *Robotics and Computer-Integrated Manufacturing*, 30(3):326–334, 2014.

[4] Jacques Denavit. A kinematic notation for low pair mechanisms based on matrices. *ASME J. Appl. Mech.*, 22:215–221, 1955.

[5] Samad A Hayati. Robot arm geometric link parameter estimation. In *The 22nd IEEE Conference on Decision and Control*, pages 1477–1483. IEEE, 1983.

[6] W Veitschegger and Chi-Haur Wu. Robot accuracy analysis based on kinematics. *IEEE Journal on Robotics and Automation*, 2(3):171–179, 1986.

[7] Hanqi Zhuang, Zvi S Roth, and Fumio Hamano. A complete and parametrically continuous kinematic model for robot manipulators. *IEEE Transactions on Robotics and Automation*, 8(4):451–463, 1992.

[8] Koichiro Okamura and Frank C Park. Kinematic calibration using the product of exponentials formula. *Robotica*, 14(4):415–421, 1996.

[9] Roger W Brockett. Robotic manipulators and the product of exponentials formula. In *Mathematical theory of networks and systems*, pages 120–129. Springer, 1984.

[10] Ruibo He, Yingjun Zhao, Shunian Yang, and Shuzi Yang. Kinematic-parameter identification for serial-robot calibration based on poe formula. *IEEE Transactions on Robotics*, 26(3):411–423, 2010.

[11] Ambarish Goswami, Arthur Quaid, and Michael Peshkin. Identifying robot parameters using partial pose information. *IEEE Control Systems Magazine*, 13(5):6–14, 1993.

[12] Johann P Prenninger. Contactless position and orientation measurement of robot end-effectors. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, pages 180–185. IEEE, 1993.

[13] Morris R Driels, W Swayze, and S Potter. Full-pose calibration of a robot manipulator using a coordinate-measuring machine. *The International Journal of Advanced Manufacturing Technology*, 8(1):34–41, 1993.

[14] Hanqi Zhuang, Shui H Motaghedi, and Zvi S Roth. Robot calibration with planar constraints. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 805–810. IEEE, 1999.

[15] Marco A Meggiolaro, Guglielmo Scriffignano, and Steven Dubowsky. Manipulator calibration using a single endpoint contact constraint. In *Proceedings of ASME Design Engineering Technical Conference, Baltimore, USA*, 2000.

[16] Chunhe Gong, Jingxia Yuan, and Jun Ni. A self-calibration method for robotic measurement system. *Journal of manufacturing science and engineering*, 122(1):174–181, 2000.

[17] Jwu-Sheng Hu, Jyun-Ji Wang, and Yung-Jung Chang. Kinematic calibration of manipulator using single laser pointer. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 426–430. IEEE, 2012.

[18] Yan Meng and Hanqi Zhuang. Autonomous robot calibration using vision technology. *Robotics and Computer-Integrated Manufacturing*, 23(4):436–446, 2007.

[19] Chandra Sekhar Gatla, Ron Lumia, John Wood, and Greg Starr. An automated method to calibrate industrial robots using a virtual closed kinematic chain. *IEEE Transactions on Robotics*, 23(6):1105–1116, 2007.

[20] Yunjiang Lou, Tieniu Chen, Yuanqing Wu, Zhibin Li, and Shilong Jiang. Improved and modified geometric formulation of poe based kinematic calibration of serial robots. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5261–5266. IEEE, 2009.

[21] JM Hollerbach. A survey of kinematic calibration the robotics review 1 ed o khatib, jj craig and t lozano, 1989.

국문초록

위 논문은 Minimal POE (product of exponentials) 정기구학 모델에 기반한 직렬로봇 캘리브레이션 알고리즘을 제안한다. 일반적으로 로봇 캘리브레이션은 엔드이펙터 프레임의 위치와 방향을 측정하는 작업을 수행해야 하는데, 이는 특별한 측정장비를 필요로 한다. 복잡한 측정장비의 사용 회피와 다양한 형태의 직렬로봇에 쉽게 응용하기 위해, 이번 논문에서는 엔드이펙터에 레이저포인터를 부착하여 평면 위의 위치가 알려진 참조점들을 추적하여 캘리브레이션을 수행하는 방법을 제시한다. 캘리브레이션은 레이저포인터와 참조점을 각각 선형조인트와 팁으로 생각하여 로봇 팁 위치의 측정값과 추정값의 차이를 최소화하는 과정으로 진행된다. 7자유도 산업용 로봇 팔에 대해 시뮬레이션과 실제 공간에서의 실험을 통해 캘리브레이션 방식을 검증했다.

주요어: 키네마틱 캘리브레이션, POE, 레이져 포인터.

학번: 2017-24281