PERGAMON

Mechanism and Machine Theory 36 (2001) 1215–1239

MECHANISM AND MACHINE THEORY

www.elsevier.com/locate/mechmt

Local POE model for robot kinematic calibration

I-Ming Chen [a,*], Guilin Yang [b], Chee Tat Tan [a], Song Huat Yeo [a]

[a] *School of Mechanical & Production Engineering, Nanyang Technological University, Nanyang Avenue, Singapore, 639798, Singapore*

[b] *Automation Technology Division, Gintic Institute of Manufacturing Technology, Singapore 638075, Singapore*

Received 25 May 2000; accepted 31 May 2001

Abstract

A robot kinematic calibration method based on the local frame representation of the product-of-exponentials (Local POE) formula is introduced. In this method, the twist coordinates of the joint axes are expressed in their respective local (body) frames. The advantages of this new approach are threefolds: (1) revolute and prismatic joints can be uniformly expressed in the twist coordinates based on the line geometry; (2) the twist coordinates of the joint axes can be set up with simple values because the local frames can be arbitrarily defined on the links; (3) the kinematic parameters described by the twist coordinates vary smoothly that makes the method robust and singularity-free. By assuming that the kinematic errors exist only in the relative initial poses of the consecutive link frames, the kinematic calibration models can be formulated in a simple and elegant way. The calibration process then becomes to re-define a set of new local link frames that are able to reflect the actual kinematics of the robot. This method can be applied to robot manipulators with generic open chain structures (serial or tree-typed). The simulation and experiment results on a 4-DOF SCARA type robot and a 5-DOF tree-typed modular robot have shown that the average positioning accuracy of the end-effector increases significantly after calibration.

Keywords: Kinematic calibration; Product-of-matrix exponentials

1. Introduction

Most robots have their kinematic model implemented, assuming nominal arrangement and description of their coordinate systems. Due to the inherent kinematic errors such as machining and assembly errors, the actual kinematic parameters of a robot will differ from their nominal

* Corresponding author. Tel.: +65-790-6203; fax: +65-791-1859.
E-mail address: michen@ntu.edu.sg (I.-M. Chen).

0094-114X/01/$ - see front matter
PII: S0094-114X(01)00048-9

values as implemented in the robot controller, lowering the positioning accuracy of the robot. Kinematic calibration serves as a solution to improve the absolute positioning accuracy of a robot. It is addressed as an integrated process of modeling, measuring, and numerically identifying the actual characteristics of a robot.

In order to formulate the calibration model, several kinematic modeling methods have been employed such as the Denavit–Hartenberg (D–H) parameterization approach [1–7], the continuous and parametrically complete (CPC) modeling approach [8–10], the zero reference position modeling approach [11–14], and the product-of-exponentials (POE) formulation approach [15–20].

The D–H kinematic formulation method [21] uses a minimum set of parameters to describe the relationship between adjacent joint axes for the formulation of the kinematics. However, being adopted for calibration modeling, this method is not amenable to direct identification as all the parameters in the D–H model are stringently defined and are unique to the particular robot configuration concerned [5]. In addition, it has the singularity problem when neighboring joint axes are nearly parallel. A number of researchers use modified forms of D–H formulation or other modeling techniques to overcome the singularity problem. Hayati and co-workers [1–4] introduce an angular alignment parameter β_i in place of the d_i parameter in the D–H model to represent a small misalignment between two consecutive parallel axes, and an additional linear parameter for handling prismatic joints. Stone [5] uses a six-parameter representation S-model in which two additional parameters are added to the D–H model to allow for arbitrary placement of link frames. Zhuang and co-workers [8–10] propose a six-parameter CPC model. In this model, a singularity-free line representation consisting of four line parameters are adopted to ensure parametric continuous, while two additional parameters are used to allow arbitrary placement of the link coordinate frames to make the model complete. Kazerounian and co-workers [11,12] and Mooring [13,14] develop their calibration model based on a zero reference position analysis method as described in [22,23]. This method describes the manipulator kinematics in terms of the axes directions and locations in the zero reference position. The error parameters considered in this model are redundant, and the elimination of redundant parameters requires assumptions that rotation and translation errors in certain directions to be negligible [14].

The POE representation method can also be considered as a zero reference position method. It describes the joint axes based on line geometry. Hence, it is uniform in modeling manipulators with both revolute and prismatic joints. Significantly, the kinematic parameters in the POE model vary smoothly with changes in joint axes so that the model can handle certain kinematic singularity problems encountered in other kinematic parametrization methods. These features make the POE formula very suitable for robot kinematic calibration. Park and Okamura [16,17] propose a novel calibration model based on the base (global) frame representation of the POE formula for open chain manipulators. This model assumes that the kinematic errors exist in each of the joint axes and the initial pose of the tool frame. Since all the kinematic parameters are expressed in the base frame, the attachment of the local frames to each of the joints (or links) is unnecessary.

Unlike Park and Okamura's model [16,17] in which the base frame representation of POE formula is employed, we propose a calibration model based on the local frame representation of the POE formula, termed as the *Local POE formula*. In the local POE formula, all the joint axes are expressed in their respective *local frames*. The major advantage of this formula is that the local coordinate frames can be arbitrarily assigned onto their corresponding links. Therefore, in the

presence of kinematic errors in the robot structure, one can always assume that the kinematic errors only exist in the initial poses of the consecutive local frames. The calibration algorithm is, according to the measurement data, to find a set of new local coordinate frames in the neighborhood of their original ones that precisely describe the actual kinematics of the robot. Since the calibration algorithm also allows the twist coordinates of the joint axes and the joint displacements to retain their nominal values when expressed in the new local frames, the calibration model can be greatly simplified and the kinematic constraints such as zero-pitch screws need not to be included into the algorithm.

Because of the local POE representation, the proposed calibration algorithm can uniformly deal with general open chain robots regardless of the types of joints, the number of branches, and the number of degrees of freedom. The kinematic parameters in this calibration model vary smoothly with changes in joint axes which makes the model singularity-free. Special descriptions are unnecessary when adjacent joint axes are close to parallel. Since the joint axes are expressed in their local frames, this calibration model is not only applicable to conventional industrial robots, but is especially useful for *modular reconfigurable robotic systems* [24]. The model can be easily modified by adding or removing local frames when modules are added in or removed from the robot system during reconfiguration.

The remaining sections of this paper are organized as follows. Section 2 briefly introduces the local POE formula for robot kinematics. The formulation of the calibration models for a general serial type robot is addressed in Section 3. A computer simulation example for calibrating a 4-DOF SCARA type robot is then presented in Section 4. For validation of the proposed calibration algorithm, the calibration experiment conducted on a 5-DOF tree-type modular robot is described in Section 5. This paper is summarized in Section 6. Finally, the geometric background related to the POE formula is provided in Appendix A.

2. The local POE formula for robot kinematics

Brocket [25] shows that forward kinematic equation of an open chain robot containing either revolute or prismatic joints can be uniformly expressed as a product of matrix exponentials. Because of its compactness, the POE formula has been shown to be a useful modeling tool in robotics [26,27]. Depending on the coordinate frames used for describing the joint axes, the POE formula may have different expressions. For the local frame representation of the POE formula, the twists of joints are expressed in their respective local frames. By introducing the *dyad kinematics* which relates the forward kinematic transformation of each local frame with respect to its preceding frame and concatenating the dyad kinematics from the base to the tool frame, the forward kinematics of an open chain robot can be derived.

2.1. Dyad kinematics

Let links $i-1$ and i be adjacent and connected by joint i, as shown in Fig. 1. Such an assembly is termed a *dyad*. If we denote the body coordinate frame on link i by frame i, then the relative pose (position and orientation) of frame i with respect to frame $i-1$, under a joint displacement q_i can be described by a 4×4 homogeneous matrix, an element of SE(3), such that

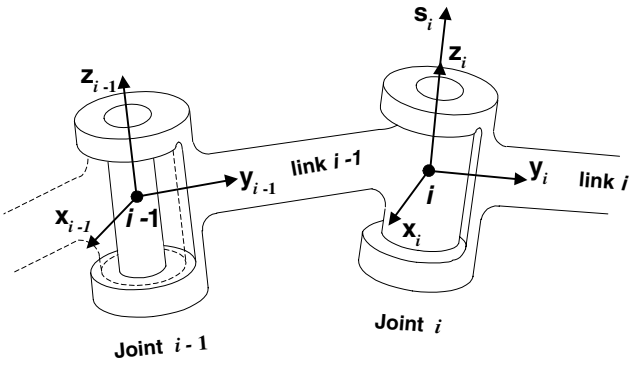

Fig. 1. Two consecutive links: a dyad.

$$T_{i-1,i}(q_i) = T_{i-1,i}(0)\, e^{\hat{s}_i q_i}, \tag{1}$$

where $T_{i-1,i}(0) \in \mathrm{SE}(3)$ is the initial pose of frame i relative to frame $i-1$. $T_{i-1,i}(0) \in \mathrm{SE}(3)$ can be written as

$$T_{i-1,i}(0) = \begin{bmatrix} R_{i-1,i}(0) & d_{i-1,i}(0) \\ 0 & 1 \end{bmatrix}, \tag{2}$$

where $R_{i-1,i}(0) \in \mathrm{SO}(3)$ and $d_{i-1,i}(0) \in \Re^{3\times 1}$ are the initial orientation and position of frame i relative to frame $i-1$, respectively.

Moreover, $\hat{s}_i \in \mathrm{se}(3)$ is the twist associated with joint i. The twist $\hat{s}_i$, also expressed in frame i, can be written as

$$\hat{s}_i = \begin{bmatrix} \hat{w}_i & v_i \\ 0 & 0 \end{bmatrix}, \tag{3}$$

where $v_i = [v_{ix}, v_{iy}, v_{iz}]^{\mathrm{T}} \in \Re^{3\times 1}$, and $\hat{w}_i \in \mathrm{so}(3)$ is the cross-product matrix (a skew-symmetric matrix) of $w_i = [w_{ix}, w_{iy}, w_{iz}]^{\mathrm{T}} \in \Re^{3\times 1}$, and is given by

$$\hat{w}_i = \begin{bmatrix} 0 & -w_{iz} & w_{iy} \\ w_{iz} & 0 & -w_{ix} \\ -w_{iy} & w_{ix} & 0 \end{bmatrix}. \tag{4}$$

The twist $\hat{s}_i$ can also be expressed as a six-dimensional vector through a mapping: $\hat{s}_i \mapsto s = (v_i, w_i) \in \Re^{6\times 1}$, termed the *twist coordinates*.

Geometrically, a twist $s = (v, w)$ is associated with a screw that consists of an axis, a pitch, and a magnitude. The twist represents the line coordinates of the screw axis of a general rigid body motion. Given a screw, there exists a twist $s = (v, w)$ such that the rigid motion associated with the screw is generated by the twist (refer to [26] for more details). In the local POE formula, the twists are expressed in their local frames. Without loss of generality, we always assign the local frame i in a simple way such that the joint axis i passes through the origin of frame i. Hence, $s_i = (0, w_i)$ for revolute joints, where w_i is the unit directional vector of the joint axis i; $s_i = (v_i, 0)$ for prismatic joints, where v_i is the unit vector indicating the direction of prismatic joint i.

An explicit formula for computing $e^{\hat{s}_i q_i}$ is given in [26,28] (refer to Appendix A for more detail). For the local POE formula (with $s_i = (0, w_i)$ or $s_i = (v_i, 0)$), the formula can be simplified as

$$e^{\hat{s}_i q_i} = \begin{bmatrix} e^{\hat{w}_i q_i} & v_i q_i \\ 0 & 1 \end{bmatrix}, \tag{5}$$

where q_i is displacement of joint i and

$$e^{\hat{w}_i q_i} = I + \hat{w}_i \sin q_i + \hat{w}_i^2 (1 - \cos q_i). \tag{6}$$

is the rotation matrix about an axis, parallel to w_i, of an angle q_i.

2.2. Local POE formula for open chains

Based on the dyad kinematics, the forward kinematic transformation for an open kinematic chain can be derived as follows. Consider a serial kinematic chain with $n + 1$ links (Fig. 2), sequentially numbered as $0, 1, \ldots, n$ (from the base 0 to the last link n). The forward kinematic transformation is the concatenation of the dyad kinematics of the consecutive dyads such that

$$\begin{aligned} T_{0,n}(q_1, q_2, \ldots, q_n) &= T_{0,1}(q_1) T_{1,2}(q_2) \cdots T_{(n-1),n}(q_n) \\ &= T_{0,1}(0) e^{\hat{s}_1 q_1} T_{1,2}(0) e^{\hat{s}_2 q_2} \cdots T_{(n-1),n}(0) e^{\hat{s}_n q_n}. \end{aligned} \tag{7}$$

Adding a fixed transformation $T_{n,n+1}$ to represent the tool frame $n + 1$ with respect to the last link frame n, the forward kinematic transformation becomes

$$T_{0,n+1}(q_1, q_2, \ldots, q_n) = T_{0,1}(0) e^{\hat{s}_1 q_1} T_{1,2}(0) e^{\hat{s}_2 q_2} \cdots T_{(n-1),n}(0) e^{\hat{s}_n q_n} T_{n,n+1}. \tag{8}$$

Eq. (8) is termed the local POE formula for a serial kinematic chain. For a tree-structured robot, the forward kinematic transformation can be derived in a similar manner by treating each branch as a serial chain.

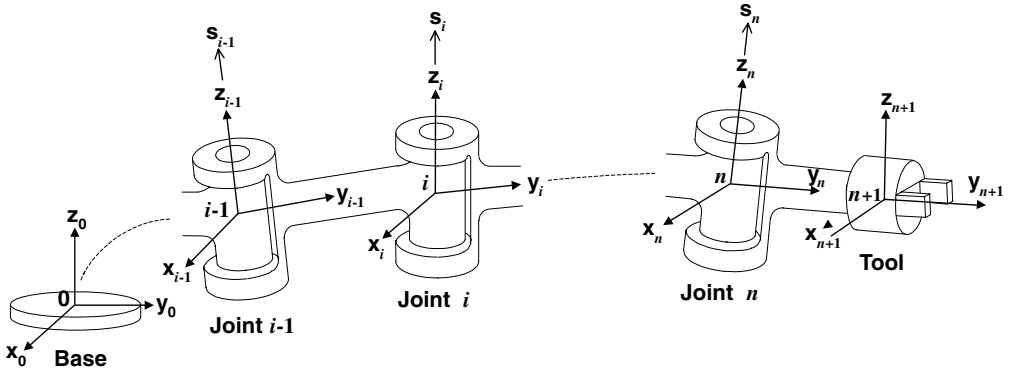

Fig. 2. Forward kinematic transformation of an n-DOF serial robot.

3. The calibration model – local POE model

3.1. Basic considerations

The nominal forward kinematics $T_{0,n+1}(q_1, q_2, \ldots, q_n)$ defined by Eq. (8), denoted as T in brief, is a function of the local frame initial poses $T(0)$, joint twists s, and joint displacements q, where $T(0) = [T_{0,1}(0), T_{1,2}(0), \ldots, T_{n-1,n}(0), T_{n,n+1}]^{\mathrm{T}}$, $s = [s_1, s_2, \ldots, s_n]^{\mathrm{T}}$, and $q = [q_1, q_2, \ldots, q_n]^{\mathrm{T}}$. Mathematically,

$$T = f(T(0), s, q). \tag{9}$$

The calibration model can be obtained by linearizing the forward kinematic equation such that

$$\delta T\, T^{-1} = \left(\frac{\partial f}{\partial T(0)} \delta T(0) + \frac{\partial f}{\partial s} \delta s + \frac{\partial f}{\partial q} \delta q \right) T^{-1}, \tag{10}$$

where $\delta T\, T^{-1}$ denotes the pose error at the tool frame expressed in the base frame, as a differential term, resulting from the kinematic errors in the $T(0)$, q, and s at each dyad. This pose error, on the other hand, represents the deviation between the pose computed from the nominal kinematic model with that from the actual measurement data. The purpose of kinematic calibration is, based on the measurement data made at various robot postures, to identify the optimal values for $\delta T(0)$, δq, and δs such that the least square pose error is minimized, i.e.,

$$\mathrm{Min}\left(\sum \left\| \delta T\, T^{-1} - \left(\frac{\partial f}{\partial T(0)} \delta T(0) + \frac{\partial f}{\partial s} \delta s + \frac{\partial f}{\partial q} \delta q \right) T^{-1} \right\| \right). \tag{11}$$

According to Eq. (8), the actual characteristics of the robot can be recovered if the actual initial poses of the local frames, joint twists, and joint displacements are identified. To formulate such a calibration model, however, 13 error parameters should be identified for a dyad, i.e., six for describing the actual local frame pose, six for describing the actual joint twist and one for describing the actual joint displacement. This will make the overall calibration model highly redundant and complex. In addition, the direct identification of the actual joint twists is difficult and may result in complicated computations.

Fig. 3 shows the dyad with kinematic errors. We denote the actual and calibrated quantities by the superscript 'a' and 'c', respectively, while those without superscripts are the nominal quantities. Due to kinematic errors, the nominal frame i ($T_{i-1,i}(0)$), the twist (s_i), and the joint displacement (q_i) of joint i are shifted to the actual frame i^{a} ($T^{\mathrm{a}}_{i-1,i}(0)$), the actual twist ($s_i^{\mathrm{a}}$), and the actual joint displacement (q_i^{a}), respectively. By taking the advantage of the local POE formula that the local frames can be arbitrarily assigned, we propose a new modeling scheme for kinematic calibration. We make two assumptions with regard to the kinematics errors:

- kinematic errors only exist in the initial poses of the local frames $T(0)$;
- the joint displacements q and the joint twists $\hat{s}$ retain their nominal values throughout the calibration analysis.

Based on the assumptions, we are able to define a new local frame on each of the link assemblies, termed as the calibrated local frame i^{c} ($T^{\mathrm{c}}_{i-1,i}(0)$), such that the calibrated joint displacement q_i^{c} and twist s_i^{c} in the dyad admit the values of their nominal counterpart q_i and $\hat{s}_i$, respectively. In other

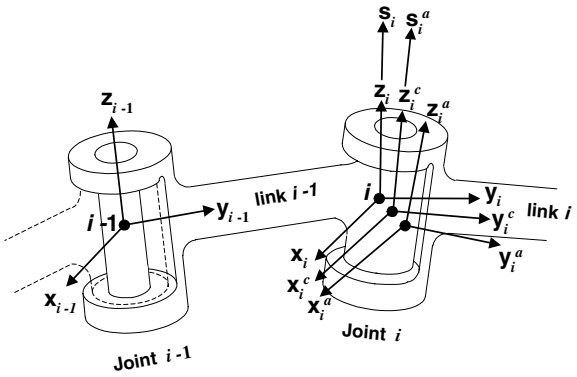

Fig. 3. Coordinate frames in a dyad.

words, instead of simultaneously identifying the actual initial poses $T(0)^{\mathrm{a}}$, joint twists s^{a}, and joint displacement q^{a}, we only have a set of newly defined calibrated frame $T(0)^{\mathrm{c}}$ to be identified. By doing so, the kinematic calibration model can be greatly simplified. Without loss of generality, we also assume that the nominal and actual base frames coincide with the world frame. Therefore, Eqs. (10) and (11) can be simplified as

$$\delta T\, T^{-1} = \frac{\partial f}{\partial T(0)}\,\delta T(0) T^{-1}, \tag{12}$$

$$\mathrm{Min}\left(\sum \left\| \delta T\, T^{-1} - \frac{\partial f}{\partial T(0)}\,\delta T(0) T^{-1} \right\| \right). \tag{13}$$

3.2. Calibration models

According to the definition of matrix logarithm defined on SE(3), there exists at least an element of $\hat{p} \in \mathrm{se}(3)$ for a given $T \in \mathrm{SE}(3)$ such that $e^{\hat{p}} = T$. Hence, it is sufficient to let $e^{\hat{p}_i} = T_{i-1,i}(0)$ (with $e^{\hat{p}_{n+1}} = T_{n,n+1}$), where $\hat{p}_i \in \mathrm{se}(3)$ $(i = 1, 2, \ldots, n+1)$. In this case, Eq. (8) can be rewritten as

$$T_{0,n+1} = e^{\hat{p}_1} e^{\hat{s}_1 q_1} e^{\hat{p}_2} e^{\hat{s}_2 q_2} \cdots e^{\hat{p}_n} e^{\hat{s}_n q_n} e^{\hat{p}_{n+1}}. \tag{14}$$

Assume that the kinematic errors occur only in $\hat{p}_i$, hence in $T_{i-1,i}(0)$ $(i = 1, 2, \ldots, n)$ and $T_{n,n+1}$. By linearizing the forward kinematic equation, Eq. (14) with respect to $\hat{p}_i$, the error models for robot calibration can be derived. Because the kinematic errors in $\hat{p}_i$ can be expressed in either frame i or frame $i-1$, the following two basic calibration models can be formulated accordingly.

1. Let the kinematic errors in $\hat{p}_i$ be expressed in module frame i, denoted by $\delta\hat{p}_{i,i}$ or simply $\delta\hat{p}_i$. Since $\hat{p}_i \in \mathrm{se}(3)$, the error $\delta\hat{p}_i$ is also an element of se(3). Geometrically, $\delta\hat{p}_i = e^{-\hat{p}_i}\delta(e^{\hat{p}_i})$ so that $\delta(e^{\hat{p}_i}) = e^{\hat{p}_i}\delta\hat{p}_i$. Linearizing Eq. (14) with respect to $\hat{p}_i$, we have

$$\begin{aligned}\delta T_{0,n+1} &= \delta(e^{\hat{p}_1})e^{\hat{s}_1 q_1}e^{\hat{p}_2}e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + e^{\hat{p}_1}e^{\hat{s}_1 q_1}\delta(e^{\hat{p}_2})e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + \cdots \\ &\quad + e^{\hat{p}_1}e^{\hat{s}_1 q_1}\cdots e^{\hat{p}_{n-1}}e^{\hat{s}_{n-1}q_{n-1}}\delta(e^{\hat{p}_n})e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + e^{\hat{p}_1}e^{\hat{s}_1 q_1}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}\delta(e^{\hat{p}_{n+1}}) \\ &= e^{\hat{p}_1}\delta\hat{p}_1 e^{\hat{s}_1 q_1}e^{\hat{p}_2}e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + e^{\hat{p}_1}e^{\hat{s}_1 q_1}e^{\hat{p}_2}\delta\hat{p}_2 e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + \cdots \\ &\quad + e^{\hat{p}_1}e^{\hat{s}_1 q_1}\cdots e^{\hat{p}_{n-1}}e^{\hat{s}_{n-1}q_{n-1}}e^{\hat{p}_n}\delta\hat{p}_n e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}} + e^{\hat{p}_1}e^{\hat{s}_1 q_1}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}e^{\hat{p}_{n+1}}\delta\hat{p}_{n+1}. \end{aligned} \tag{15}$$

Right-multiplying both sides of the resulting equation by $T_{0,n+1}^{-1}$ and using the *adjoint representation*, the following error model can be shown:

$$\delta T_{0,n+1}T_{0,n+1}^{-1} = \mathrm{Ad}_{e^{\hat{p}_1}}\delta\hat{p}_1 + \mathrm{Ad}_{e^{\hat{p}_1}e^{\hat{s}_1 q_1}}\mathrm{Ad}_{e^{\hat{p}_2}}\delta\hat{p}_2 + \cdots + \mathrm{Ad}_{e^{\hat{p}_1}e^{\hat{s}_1 q_1}e^{\hat{p}_2}e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_n}e^{\hat{s}_n q_n}}\mathrm{Ad}_{e^{\hat{p}_{n+1}}}\delta\hat{p}_{n+1}, \tag{16}$$

where $\delta T_{0,n+1}T_{0,n+1}^{-1} \in \mathrm{se}(3)$ represents the gross kinematic errors of a complete robot with respect to the base frame. According to the formulation of matrix logarithm defined on SE(3) [29], $\delta T_{0,n+1}T_{0,n+1}^{-1}$ can be given by

$$\delta T_{0,n+1}T_{0,n+1}^{-1} = \log\left(T_{0,n+1}^{\mathrm{a}}T_{0,n+1}^{-1}\right), \tag{17}$$

where $T_{0,n+1}^{\mathrm{a}}$ represents the actual (measured) pose of the tool frame with respect to the base frame. Since elements of se(3) can also be identified by six-dimensional vectors through: $\delta\hat{p}_i \mapsto \delta p_i \in \Re^{6\times 1}$ and $\log(T_{0,n+1}^{\mathrm{a}}T_{0,n+1}^{-1}) \mapsto \log(T_{0,n+1}^{\mathrm{a}}T_{0,n+1}^{-1})^{\vee} \in \Re^{6\times 1}$, Eq. (16) can be expressed as

$$\log\left(T_{0,n+1}^{\mathrm{a}}T_{0,n+1}^{-1}\right)^{\vee} = \mathrm{Ad}_{T_{0,1}(0)}\delta p_1 + \mathrm{Ad}_{T_{0,1}}\mathrm{Ad}_{T_{1,2}(0)}\delta p_2 + \cdots + \mathrm{Ad}_{T_{0,n}}\mathrm{Ad}_{T_{n,n+1}(0)}\delta p_{n+1}, \tag{18}$$

where $T_{0,i} = e^{\hat{p}_1}e^{\hat{s}_1 q_1}e^{\hat{p}_2}e^{\hat{s}_2 q_2}\cdots e^{\hat{p}_i}e^{\hat{s}_i q_i}$, representing the forward kinematic transformation from the base frame 0 to module frame i. After δp_i is identified, the calibrated initial pose of module frame i with respect to module frame $i-1$, $T_{i-1,i}^{\mathrm{c}}(0)$, can be given by

$$T_{i-1,i}^{\mathrm{c}}(0) = e^{\hat{p}_i}e^{\delta\hat{p}_i} = T_{i-1,i}(0)e^{\delta\hat{p}_i}. \tag{19}$$

2. Let the kinematic errors in $\hat{p}_i$ be expressed in the module frame $i-1$, denoted by $\delta\hat{p}_{i-1,i}$. Then $\delta\hat{p}_{i-1,i} \in \mathrm{se}(3)$ and $\delta\hat{p}_{i-1,i} = \delta(e^{\hat{p}_i})e^{-\hat{p}_i} = \mathrm{Ad}_{T_{i-1,1}(0)}\delta\hat{p}_i$. Eq. (18) can be rewritten as

$$\log\left(T_{0,n+1}^{\mathrm{a}}T_{0,n+1}^{-1}\right)^{\vee} = \delta p_{0,1} + \mathrm{Ad}_{T_{0,1}}\delta p_{1,2} + \cdots + \mathrm{Ad}_{T_{0,n}}\delta p_{n,n+1}, \tag{20}$$

where $\delta p_{i-1,i} \in \Re^{6\times 1}$. After $\delta p_{i-1,i}$ is identified, $T_{i-1,i}^{\mathrm{c}}(0)$ – the calibrated initial pose of module frame i with respect to module frame $i-1$ can be given by

$$T_{i-1,i}^{\mathrm{c}}(0) = e^{\delta\hat{p}_i}e^{\hat{p}_i} = e^{\delta\hat{p}_i}T_{i-1,i}(0). \tag{21}$$

The basic error models of Eqs. (18) and (21) are derived by observing the gross kinematic errors of a complete robot with respect to the base frame. Alternatively, the gross kinematic errors of the robot can also be expressed with respect to the tool frame. Left-multiplying both sides of Eqs. (18) and (21) by $\mathrm{Ad}_{T_{0,n+1}^{-1}}$, the following two equations can be easily shown:

$$\log\left(T_{0,n+1}^{-1}T_{0,n+1}^{\mathrm{a}}\right)^{\vee} = \mathrm{Ad}_{T_{0,n+1}^{-1}}(\mathrm{Ad}_{T_{0,1}(0)}\delta p_1 + \mathrm{Ad}_{T_{0,1}}\mathrm{Ad}_{T_{1,2}(0)}\delta p_2 + \cdots + \mathrm{Ad}_{T_{0,n}}\mathrm{Ad}_{T_{n,n+1}(0)}\delta\hat{p}_{n+1}), \tag{22}$$

$$\log\left(T_{0,n+1}^{-1}T_{0,n+1}^{a}\right)^{\vee} = \mathrm{Ad}_{T_{0,n+1}^{-1}}(\delta p_{0,1} + \mathrm{Ad}_{T_{0,1}}\delta p_{1,2} + \cdots + \mathrm{Ad}_{T_{0,n}}\delta p_{n,n+1}), \tag{23}$$

where $\log(T_{0,n+1}^{-1}T_{0,n+1}^{a})^{\vee}$ is a six-dimensional vector associated with $\log(T_{0,n+1}^{-1}T_{0,n+1}^{a}) \in \mathrm{se}(3)$, representing the gross kinematic errors of the robot with respect to the tool frame, and $\mathrm{Ad}_{T_{0,n+1}^{-1}}\log(T_{0,n+1}^{a}T_{0,n+1}^{-1}) = \log(T_{0,n+1}^{-1}T_{0,n+1}^{a}) = T_{0,n+1}^{-1}dT_{0,n+1}$.

By expressing the local and gross kinematic errors of a robot with respect to different reference frames, four compact calibration models are formulated, i.e., Eqs. (18) and (21)–(23). For the purpose of kinematic calibration, these models are equivalent. From the computational point of view, however, these four models are slightly different. For example, the term $\log(T_{0,n+1}^{-1}T_{0,n+1}^{a})^{\vee}$ in both Eqs. (22) and (23) can be calculated in a simple and efficient way that

$$\log\left(T_{0,n+1}^{-1}T_{0,n+1}^{a}\right)^{\vee} = \begin{bmatrix} (R - R^{T})/2 & R_{0,n+1}^{T}\left(P_{0,n+1}^{a} - P_{0,n+1}\right) \\ 0 & 0 \end{bmatrix}, \tag{24}$$

where $T_{0,n+1}^{a}$ and $T_{0,n+1}$ are given by $(P_{0,n+1}^{a}, R_{0,n+1}^{a})$ and $(P_{0,n+1}, R_{0,n+1})$, respectively. In Eq. (24), $R = R_{0,n+1}^{T}R_{0,n+1}^{a}$.

All the calibration models mentioned above are based on the local POE formula. They can be easily modified to suit the calibration of a tree-structured modular robot. Since the branches in a tree-structured modular robot may contain several common modules (or dyads), all these branches are to be calibrated simultaneously in order to get a unique calibrated initial pose for each of the common dyads.

3.3. An iterative least-squares algorithm

Based on the calibration models, an iterative least-squares algorithm is employed for the calibration solution. This algorithm is suitable for each of the calibration models. For simplicity, we will introduce this algorithm according to the error model of Eq. (18).

Eq. (18) can also be expressed as a linear equation of the form

$$y = Ax, \tag{25}$$

where

$$y = \log\left(T_{0,n+1}^{a}T_{0,n+1}^{-1}\right)^{\vee} \in \mathfrak{R}^{6\times 1},$$

$$x = [\delta p_1, \delta p_2, \ldots, \delta p_{n+1}]^{T} \in \mathfrak{R}^{6(n+1)\times 1},$$

$$A = \left[\mathrm{Ad}_{T_{0,1}(0)}, \mathrm{Ad}_{T_{0,1}}\mathrm{Ad}_{T_{1,2}(0)}, \ldots, \mathrm{Ad}_{T_{0,n}}\mathrm{Ad}_{T_{n,n+1}(0)}\right] \in \mathfrak{R}^{6\times 6(n+1)}.$$

$T_{0,n+1}^{-1}$ and A can be obtained from the nominal model, while $T_{0,n+1}^{a}$ can be obtained from the actual measurement data. Hence, the pose error $y = \log(T_{0,n+1}^{a}T_{0,n+1}^{-1})^{\vee}$ can be computed. x is the kinematic errors in the robot to be identified.

The identification of the kinematic errors requires comparing the difference between the robot's actual (measured) and nominal poses. To improve the calibration accuracy, we usually need to measure the tool frame in many different robot postures. Suppose we need to take m measured pose data. For ith measurement, we obtain a pose error y_i. The corresponding adjoint map matrix

A_i can be determined from the nominal kinematic model. After m measurements, we can stack y_i and A_i to form the following equation:

$$\widetilde{Y} = \widetilde{A}x, \tag{26}$$

where

$$\widetilde{Y} = [y_1, y_2, \ldots, y_m]^{\mathrm{T}} \in \Re^{6m\times 1},$$
$$x = [\delta p_1, \delta p_2, \ldots, \delta p_{n+1}]^{\mathrm{T}} \in \Re^{6(n+1)\times 1},$$
$$\widetilde{A} = \mathrm{column}[A_1, A_2, \ldots, A_m] \in \Re^{6m\times 6(n+1)}.$$

It is seen that Eq. (26) is a linear calibration model. Since the model consists of $6m$ linear equations with $6(n+1)$ variables ($m > (n+1)$), the linear least-squares algorithm is employed for the parameter identification. The least-squares solution of x is given by

$$x = \left(\widetilde{A}^{\mathrm{T}}\widetilde{A}\right)^{-1}\widetilde{A}^{\mathrm{T}}\widetilde{y}, \tag{27}$$

where $(\widetilde{A}^{\mathrm{T}}\widetilde{A})^{-1}\widetilde{A}^{\mathrm{T}}$ is the pseudoinverse of $\widetilde{A}$.

The solution of Eq. (27) can be further improved through iterative substitution as shown in Fig. 4. Let $T(0)$ denote the vector of the initial poses of the dyads and initialize $T(0)$ using the nominal value, i.e.,

$$T(0) = [T_{0,1}(0), T_{1,2}(0), \ldots, T_{n-1,n}(0), T_{n,n+1}]^{\mathrm{T}}.$$

Based on the measured data and Eq. (27), we can obtain the kinematic error parameter vector, x. The vector $T(0)$ is updated by substituting x into Eq. (19). The same procedure is repeated until the norm of the error vector, $\|x\|$, approaches zero and the vector $T(0)$ converges to some stable values. Then the final $T(0)$ represents the calibrated initial poses of the dyads, i.e.,

$$T(0) = [T_{0,1}(0)^{\mathrm{c}}, T_{1,2}(0)^{\mathrm{c}}, \ldots, T_{n,n+1}(0)^{\mathrm{c}}]^{\mathrm{T}}.$$

Note that the kinematic error vector, x, will no longer represent the actual kinematic errors in the dyads after iterations. However, the actual kinematic errors in each of the dyads can be extracted by comparing $T_{i-1,i}(0)$ with $T_{i-1,i}(0)^{\mathrm{c}}$. Based on the matrix logarithm, Eq. (19) can also be written as

$$\delta p_i = \log\left(T_{i-1,i}^{-1}(0)T_{i-1,i}^{\mathrm{c}}(0)\right)^{\vee}. \tag{28}$$

After calibration, the forward kinematic equation becomes

$$T_{0,n+1} = T_{0,1}^{\mathrm{c}}(0)e^{\hat{s}_1 q_1}T_{1,2}^{\mathrm{c}}(0)e^{\hat{s}_2 q_2}\cdots T_{n-1,n}^{\mathrm{c}}(0)e^{\hat{s}_n q_n}T_{n,n+1}^{\mathrm{c}}. \tag{29}$$

In order to evaluate the calibration result, we can define a deviation metric between the measured (actual) 'a' and calibrated 'c' tool frames mathematically as

$$\Delta T = \frac{1}{m}\sum_{i=1}^{m}\left\|\log(T_{\mathrm{a}i}^{-1}T_{\mathrm{c}i})^{\vee}\right\|, \tag{30}$$

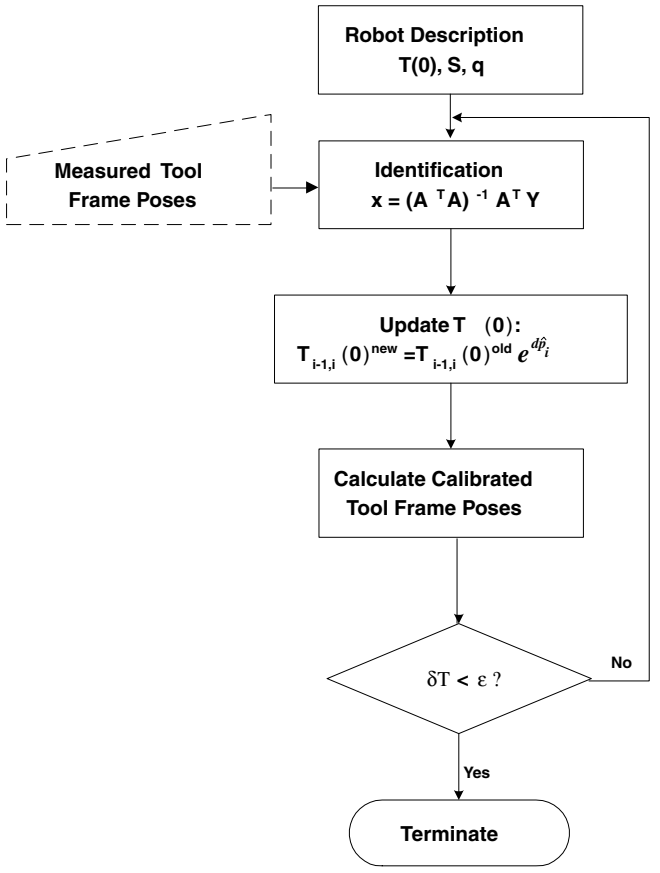

Fig. 4. Calibration algorithm.

where ΔT denote the average quantified deviation between the calibrated and actual poses and m is the number of measured poses. Since ΔT cannot clearly indicate the positional and orientational deviations, we define

$$\Delta R = \frac{1}{m}\sum_{i=1}^{m} \left\| \log(R_{ai}^{-1} R_{ci})^{\vee} \right\|, \tag{31}$$

$$\Delta P = \frac{1}{m}\sum_{i=1}^{m} \|P_{ai} - P_{ci}\|, \tag{32}$$

where ΔR and ΔP denotes the average quantified orientational and positional deviations between the calibrated and actual poses, respectively.

3.4. Computational complexity of the calibration models

The calibration models presented in this section require the identification of six kinematic error parameters to calibrate the initial pose of each dyad. An n-DOF modular robot will have $6(n+1)$

error parameters to be identified. This number is slightly greater than that of the conventional calibration model in which the number of error parameters is equal to $4R + 2P$, where R and P represent the number of revolute and prismatic joints, respectively [30]. Consequently, the dimension of the adjoint map matrix $\widetilde{A}$ in Eq. (26) is slightly larger than identification Jacobian matrix of the convention model thus requires heavier computation. However, the calibration computation is usually an off-line computation scheme and the adjoint map matrix $\widetilde{A}$ is a $6m \times 6(n+1)$ matrix which, in general, is not very large. Therefore, the computation complexity of the calibration models is not so critical.

4. Computer simulation

In this section, a simulation example for calibrating a 4-DOF SCARA type robot (Fig. 5) is given to demonstrate the effectiveness of the calibration algorithm. This example is presented because

1. it allows the verification of the assumptions made during modeling;
2. it demonstrates the capability of the calibration algorithm in handling a robot with different joint types (revolute and prismatic) and the robustness in dealing with the singularity problems inherent in the conventional D–H representation;
3. it illustrates the effect of the measurement noise and robot repeatability on the calibration result.

As shown in Fig. 5, the given kinematic parameters for the 4-DOF (RRPR) SCARA type robot are as follows:

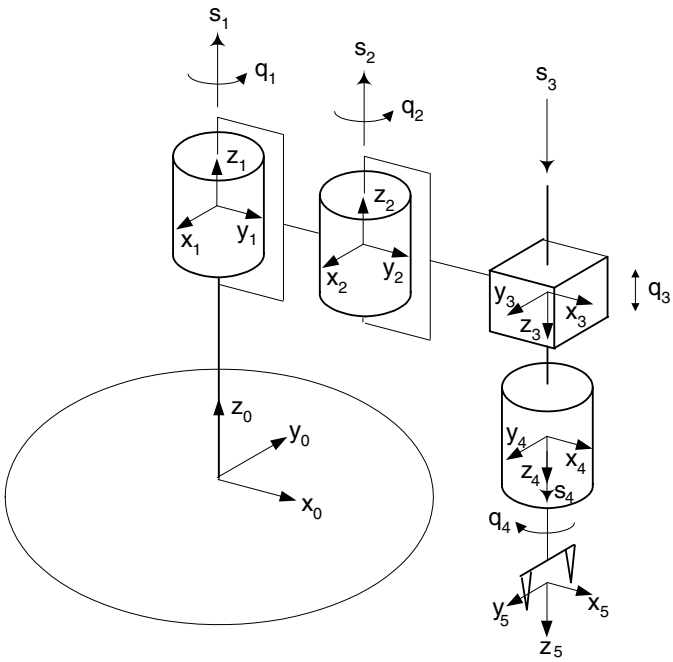

Fig. 5. A 4-DOF SCARA type robot.

$$T_{0,1}(0)=\begin{bmatrix}0&1&0&0\\-1&0&0&0\\0&0&1&0.75\\0&0&0&1\end{bmatrix},\quad T_{1,2}(0)=\begin{bmatrix}1&0&0&0\\0&1&0&0.25\\0&0&1&0\\0&0&0&1\end{bmatrix},$$

$$T_{2,3}(0)=\begin{bmatrix}0&1&0&0\\1&0&0&0.22\\0&0&-1&0\\0&0&0&1\end{bmatrix},\quad T_{3,4}(0)=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0.15\\0&0&0&1\end{bmatrix},$$

$$T_{4,5}(0)=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&0.10\\0&0&0&1\end{bmatrix},$$

$$s_1=(0,0,0,0,0,1)^{\mathrm{T}},\quad s_2=(0,0,0,0,0,1)^{\mathrm{T}},\quad s_3=(0,0,1,0,0,0)^{\mathrm{T}},$$

$$s_4=(0,0,0,0,0,1)^{\mathrm{T}},\quad s_5=(0,0,0,0,0,0)^{\mathrm{T}}.$$

The SI units are employed for the kinematic parameters – *radians* for angular (orientational) parameters and *meters* for linear (positional) parameters. The entire simulation process is shown in Fig. 4. In this example, we simulate the calibration process under an ideal experimental condition where there is no measurement noise. To obtain the actual tool frame poses, the following procedures are employed:

1. Assign kinematic errors at each of the dyads $\delta p_i'$, $\delta q_i'$, and $\delta s_i'$ $(i=1,2,\ldots,5)$ (listed in Table 1).
2. Find the actual initial poses of frame i in the dyad of link $(i-1)$ and i $T_{i-1,i}^{\mathrm{a}}(0)=T_{i-1,i}(0)e^{\delta\hat{p}_i'}$ $(i=1,2,\ldots,5)$.
3. Calculate the actual tool frame poses at different robot postures.

$$T_{0,5}^{\mathrm{a}}=\prod_{i=1}^{5}\left(T_{i-1,i}^{\mathrm{a}}(0)e^{(\hat{s}_i+\delta\hat{s}_i')(q_i+\delta q_i')}\right).$$

Since each joint is treated as an ideal 1-DOF joint, the conditions to be considered in assigning the errors in each joint twist are $\|w_i+\delta w_i'\|=1$ and $(w_i+\delta w_i')^{\mathrm{T}}(v_i+\delta v_i')=0$, where $s_i=(v_i,w_i)^{\mathrm{T}}$ and $\delta s_i'=(\delta v_i',\delta w_i')^{\mathrm{T}}$.

In this simulation example, the number of measured poses are set to 6 [15]. The initial poses of the calibrated local frames as well as the identified kinematic errors in each of the dyads are listed in Table 2. Since the preseted and identified kinematic errors do not have the same

Table 1
Preseted kinematic errors

Dyad	$\delta p_i'$	$\delta s_i'$	$\delta q_i'$
0–1	$(0,\ 0.002,\ 0.002,\ -0.02,\ 0,\ 0)^{\mathrm{T}}$	$(0,\ 0,\ 0,\ 0,\ \sin(0.02),\ -1+\cos(0.02))^{\mathrm{T}}$	0
1–2	$(0,\ 0.002,\ 0.002,\ 0.02,\ 0,\ 0)^{\mathrm{T}}$	$(0,\ 0,\ 0,\ 0,\ -\sin(0.02),-1+\cos(0.02))^{\mathrm{T}}$	0.02
2–3	$(0.002,\ 0.002,\ 0.002,\ 0.02,0.02,\ 0.02)^{\mathrm{T}}$	$(\sin(0.02),\ 0,\ -1+\cos(0.02),\ 0,\ 0,\ 0)^{\mathrm{T}}$	0.002
3–4	$(0.002,\ 0.002,\ 0.002,\ 0.02,\ 0.02,\ 0.02)^{\mathrm{T}}$	$(0.002,\ 0,\ 0,\ 0,\ \sin(0.02),\ -1+\cos(0.02))^{\mathrm{T}}$	0.02
4–5	$(0.002,\ 0.002,\ 0.002,\ 0.02,\ 0.02,\ 0.02)^{\mathrm{T}}$	$(0,\ 0,\ 0,\ 0,\ 0,\ 0)^{\mathrm{T}}$	0

Table 2
Identified kinematic errors and calibrated local frame poses

Dyad	δp_i	$T^c_{i-1,i}(0)$
0–1	$(0.000011,\ 0.001979,\ -0.000706,\ -0.04022,\ -0.00019,\ -0.01137)^T$	$\begin{bmatrix} -0.01139 & 0.99914 & 0.03999 & 0.001891 \\ -0.99994 & -0.01140 & 0 & 0 \\ 0.000456 & -0.03999 & 0.99920 & 0.749273 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
1–2	$(-0.002881,\ 0.002476,\ -0.000804,\ 0.06013,\ 0.00106,\ -0.01209)^T$	$\begin{bmatrix} 0.99993 & 0.01214 & 0.00068 & -0.002878 \\ -0.01208 & 0.99813 & -0.05996 & 0.252523 \\ -0.00141 & 0.05995 & 0.99820 & -0.000714 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2–3	$(0.000225,\ -0.001790,\ 0.000734,\ 0.02080,\ 0.02095,\ 0.01223)^T$	$\begin{bmatrix} 0.01236 & 0.99972 & -0.02006 & -0.001862 \\ 0.99973 & -0.01197 & 0.01954 & 0.220439 \\ 0.01929 & -0.02030 & -0.99961 & -0.000691 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3–4	$(0.000236,\ -0.001796,\ 0.000704,\ -0.00092,\ -0.00143,\ 0.01241)^T$	$\begin{bmatrix} 0.99992 & -0.01243 & 0.00001 & 0.000436 \\ 0.01243 & 0.99992 & 0.00040 & -0.001869 \\ -0.00002 & -0.00040 & 1 & 0.150712 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
4–5	$(0.001923,\ -0.002038,\ 0.00077,\ 0.03993,\ 0.01996,\ 0.01203)^T$	$\begin{bmatrix} 0.99973 & -0.01167 & 0.02008 & 0.001983 \\ 0.01246 & 0.99913 & -0.03974 & -0.0020656 \\ -0.01959 & 0.03998 & 0.99901 & 0.100708 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

physical meaning and are not of one-to-one correspondence, the preseted kinematic errors at each of the dyads are not fully recovered. However, the success of the calibration simulation can be deduced from the results shown in Table 3, where the nominal, actual, and calibrated tool frame poses representing two arbitrarily selected robot postures are listed. The results show that under the calibrated parameters description, the actual (measured) poses can be precisely

Table 3
Tool frame poses before and after calibration

Joint angle	$q = (0.58780,\ 0.64131,\ 0.093684,\ 1.65940)^T$	$q = (1.83054,\ 1.89216,\ 0.067035,\ 2.22327)^T$
Nominal pose	$\begin{bmatrix} 0.90885 & -0.41713 & 0 & 0.281757 \\ -0.41713 & -0.90885 & 0 & 0.345915 \\ 0 & 0 & -1 & 0.406316 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.071308 & 0.997454 & 0 & -0.248097 \\ 0.997454 & -0.071308 & 0 & 0.120845 \\ 0 & 0 & -1 & 0.432965 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Measured pose	$\begin{bmatrix} 0.88159 & -0.47073 & 0.03470 & 0.289488 \\ -0.47191 & -0.87759 & 0.08443 & 0.368679 \\ -0.00930 & -0.09081 & -0.99583 & 0.402706 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.13368 & 0.98612 & -0.09843 & -0.269843 \\ 0.99013 & -0.12868 & 0.05555 & 0.136173 \\ 0.04211 & -0.10489 & -0.99359 & 0.436307 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Calibrated pose	$\begin{bmatrix} 0.88159 & -0.47073 & 0.03470 & 0.289488 \\ -0.47192 & -0.87759 & 0.08443 & 0.368679 \\ -0.00930 & -0.09081 & -0.99583 & 0.402706 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 0.13368 & 0.98612 & -0.09843 & -0.269843 \\ 0.99013 & -0.12868 & 0.05555 & 0.136173 \\ 0.04211 & -0.10489 & -0.99359 & 0.436307 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

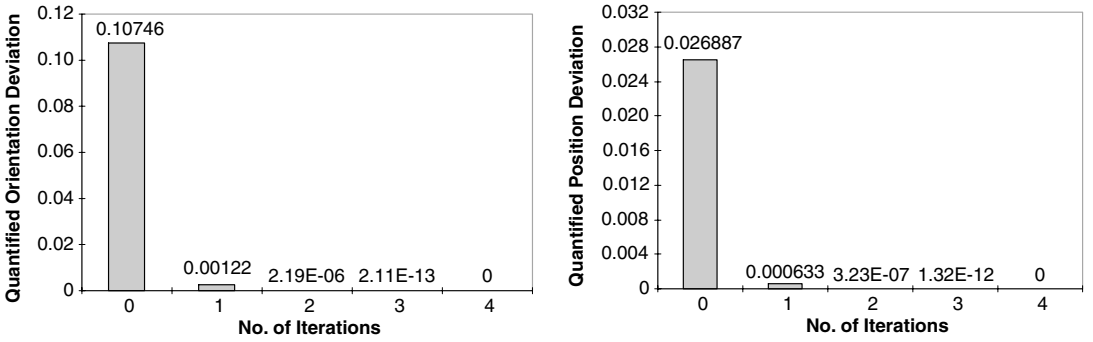

Fig. 6. Calibration convergence.

described. In other words, the calibration algorithm is successful in improving the pose accuracy of the tool frame. A further re-enforcement of the calibration results is from Fig. 6 which shows that the simulation yields a full recovery of kinematic errors at the tool frame. Both the quantified orientation and position deviations, as in Eqs. (31) and (32), are driven from the initial magnitude of 0.10746 and 0.026887 to values approximately zero. Since a SCARA type robot is made up of both prismatic and revolute joints, the simulation example shows that the proposed local POE (calibration) model is uniform for robot kinematic calibration regardless of the types of joints.

The calibration result also verifies the assumptions made for the calibration analysis. First, the actual joint twists expressed in the calibrated local frames really take their nominal values. According to the preset errors in the joint twists, the actual twists expressed in the local actual module frames are given by:

$$s_1^{\mathrm{a}} = s_1 + \delta s_1' = (0,0,0,0,\sin(0.02),\cos(0.02))^{\mathrm{T}},$$
$$s_2^{\mathrm{a}} = s_2 + \delta s_2' = (0,0,0,0,-\sin(0.02),\cos(0.02))^{\mathrm{T}},$$
$$s_3^{\mathrm{a}} = s_3 + \delta s_3' = (\sin(0.02),0,\cos(0.02),0,0,0)^{\mathrm{T}},$$
$$s_4^{\mathrm{a}} = s_4 + \delta s_4' = (0.002,0,0,0,\sin(0.02),\cos(0.02))^{\mathrm{T}}.$$

In the presence of the preseted kinematic errors in the local frames, the initial poses of the actual local frames with respect to the base frame are

$$T_{0,1}^{\mathrm{a}}(0) = \begin{bmatrix} 0 & 0.99980 & 0.01999 & 0.002 \\ -1 & 0 & 0 & 0 \\ 0 & -0.01999 & 0.99980 & 0.752 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{0,2}^{\mathrm{a}}(0) = \begin{bmatrix} 0.01999 & 0.99980 & 0 & 0.25399 \\ -0.99980 & 0.01999 & 0.00040 & 0 \\ 0.00040 & 0 & 1 & 0.74896 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T^{a}_{0,3}(0) = \begin{bmatrix} 0.99980 & 0.000040 & 0.02000 & 0.476065 \\ 0 & -0.99981 & 0.01960 & 0.002478 \\ 0.02000 & -0.01959 & -0.99961 & 0.744962 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T^{a}_{0,4}(0) = \begin{bmatrix} 0.99842 & -0.03877 & 0.04077 & 0.481150 \\ -0.04037 & -0.99842 & 0.03917 & 0.003460 \\ 0.03919 & -0.04075 & -0.99840 & 0.593020 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

According to the calibrated results, the initial poses of the calibrated local frames with respect to the base frame are given by

$$T^{c}_{0,1}(0) = \begin{bmatrix} -0.01139 & 0.99914 & 0.03999 & 0.001891 \\ -0.99994 & -0.01140 & 0 & 0 \\ 0.000456 & -0.03999 & 0.99920 & 0.749273 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T^{c}_{0,2}(0) = \begin{bmatrix} -0.02351 & 0.99952 & -0.02000 & 0.254200 \\ -0.99972 & -0.02352 & 0 & 0 \\ -0.00047 & 0.01999 & 0.99980 & 0.738461 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T^{c}_{0,3}(0) = \begin{bmatrix} 0.99859 & -0.03506 & 0.03999 & 0.474591 \\ -0.03586 & -0.99916 & 0.01960 & -0.003322 \\ 0.03927 & -0.02100 & -0.99901 & 0.742178 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T^{c}_{0,4}(0) = \begin{bmatrix} 0.99807 & -0.04749 & 0.03999 & 0.481119 \\ -0.04829 & -0.99865 & 0.01920 & 0.001483 \\ 0.03902 & -0.02109 & -0.99902 & 0.591672 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Based on the definition of the *adjoint presentation*, the actual twists expressed in their local calibrated frames, denoted by $s'a_i$ $(i = 1, 2, 3, 4)$, can be computed by

$$s'^{a}_{i} = \mathrm{Ad}^{-1}_{T^{c}_{0,i}(0)} \mathrm{Ad}_{T^{a}_{0,i}(0)} s^{a}_{i}. \tag{33}$$

Hence, we can calculate s'^{a}_{i} as

$$s'^{a}_{1} = (0,0,0,0,0,1)^{T} = s_1, \quad s'^{a}_{2} = (0,0,0,0,0,1)^{T} = s_2, \quad s'^{a}_{3} = (0,0,1,0,0,0)^{T} = s_3,$$
$$s'^{a}_{4} = (0,0,0,0,0,1)^{T} = s_4,$$

which really take their nominal values, respectively.

Second, the assumption that the joint angles assuming nominal values under the calibrated local frame description is also fulfilled. This can be deduced from the results listed in Table 3, i.e. the computed end frame poses are exactly equal to the corresponding simulated measured poses under the various sets of nominal joint angles.

This example also shows the capability of the model in handling parameters changes at singularity configurations of the robot. As indicated in Fig. 7, the kinematic errors are deliberately preset in such a way that the first two parallel axes of the robot configuration are misaligned to intersect at a point. We understand that with this 'actual' twist geometry, the identification of the error parameters by the traditional D–H model would be difficult due to the abrupt changes in the kinematic parameters at this singularity configuration. However, our modeling technique allows the smooth change of the kinematic parameters which is able to handle the singularity problem.

In the same example, we finally illustrate the effect of the measurement noise and robot repeatability on the calibration result. Form the viewpoint of computer simulation, both of the measurement noise and robot repeatability can be simulated in a uniform way by adding uniformly distributed noise on each of the actual end-effector's poses. The resulting actual end-effector's poses $T_{0,5}^{a*}$ can be given by

$$T_{0,5}^{a*} = T_{0,5}^{a} e^{\hat{\delta p}_i^*}, \tag{34}$$

where

$$\delta p_i^* = \left(\delta x_i^*, \delta y_i^*, \delta z_i^*, \delta \alpha_i^*, \delta \beta_i^*, \delta \gamma_i^*\right)^{\mathrm{T}}$$

represents the noise injected. δx_i^*, δy_i^*, and $\delta z_i^* \in \mathbf{U}[0.0001, 0.0001]$ m; $\delta \alpha_i^*$, $\delta \beta_i^*$, and $\delta \gamma_i^* \in \mathbf{U}[0.001, 0.001]$ rad.

Extensive simulations have been done to study the effect of the measurement noise and robot repeatability on the calibration results. In each simulation, two groups of 50 simulated end-effector's poses are employed. One group is used to calibrate the robot, while the other group is used to verify the result of the calibration, i.e., to derive the quantified orientation and position deviations. From these simulations, it can be found that the calibrated initial poses of the module frames and the quantified orientation and position deviations become stable when the number of

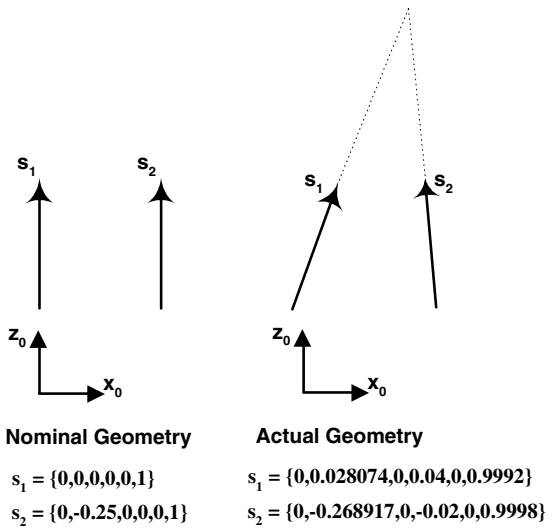

Fig. 7. A singularity configuration.

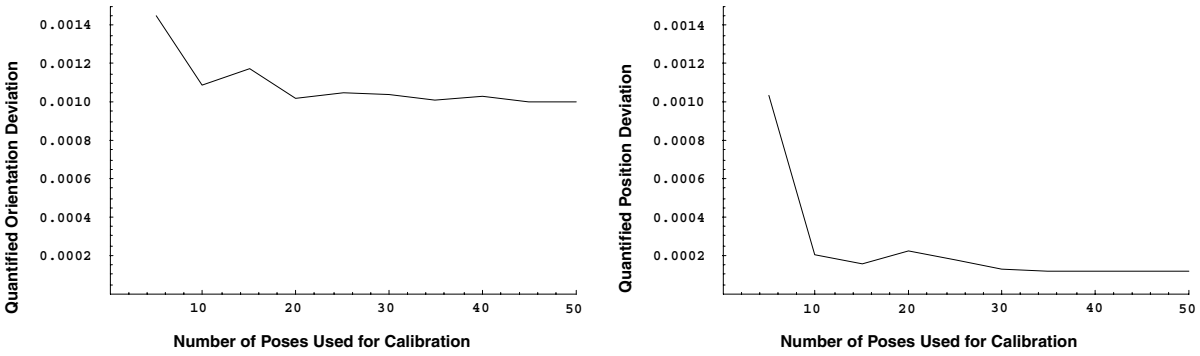

Fig. 8. Verification of the calibration result.

poses used for calibration is greater than 30. A typical simulation result is given in Fig. 8, in which the stable quantified orientation and position deviations are 0.001 rad and 0.0001 m, respectively. These values are in the same order of magnitude as that of the injected noise.

5. Calibration experiment

To further validate the effectiveness and generality of the calibration algorithm for industrial applications, a calibration experiment is also conducted. The experimental setup consists of an articulated type coordinate measuring machine, SpinArm, by Mitutoyo of Japan, and a tree-type modular robot (to be calibrated) from Amtec GmbH of Germany, shown in Fig. 9.

The Mitutoyo SpinArm is a 6-DOF articulated coordinate measuring machine. The accuracy for positioning measurement is ± 0.00008 m. The positioning accuracy of Amtec modular robots average at ± 0.01 m experimentally, which is significantly less than that of the SpinArm. The working envelope of SpinArm covers a hemisphere of 2.4 m in diameter. A simple cubic fixture with very high precision is attached to the robot end-effector for the collection of both the position

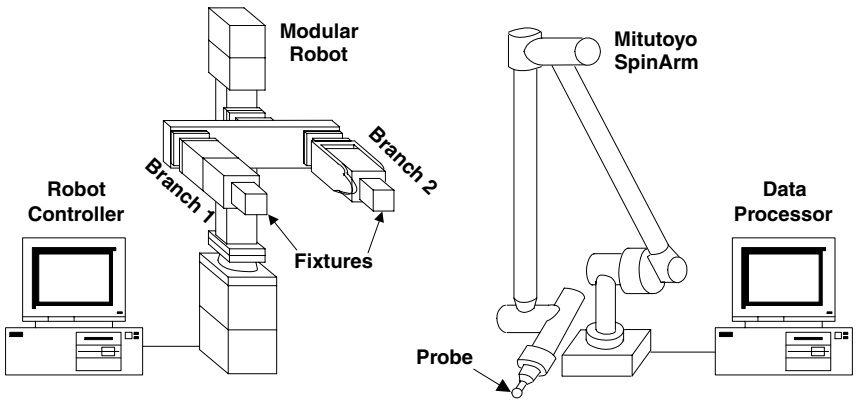

Fig. 9. Experimental setup for robot calibration.

and orientation description of the end-effector frame using the data acquisition software provided. The setup of the system is easy due to its portability and ease of establishing the coordinate systems.

As shown in Fig. 10, the tree-type modular robot has two branches: Branch 1 consists of 3 (RPR) joints and Branch 2 consists of 4 (RPRR) joints. In total the robot has 5 DOFs because joints 1 and 2 are the two common joints in branches 1 and 2. We will calibrate these two branches simultaneously by using the local POE calibration model. Fig. 10 shows the local frame assignment of the tree-type modular robot. Note that the two tool frames, frame 6 and frame 7, normally coincide with their corresponding last link frames, frame 3 and frame 5, respectively. Based on Eq. (25) and the connecting sequence of the links and joints, the kinematic calibration model for this tree-structured robot can be written as

$$Y = Ax,$$

where

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \log(T^{a}_{0,6} T^{-1}_{0,6})^{\vee} \\ \log\left(T^{a}_{0,7} T^{-1}_{0,7}\right)^{\vee} \end{bmatrix} \in \Re^{12\times 1},$$

$$x = [\delta p_1, \delta p_2, \ldots, \delta p_7]^{T} \in \Re^{42\times 1},$$

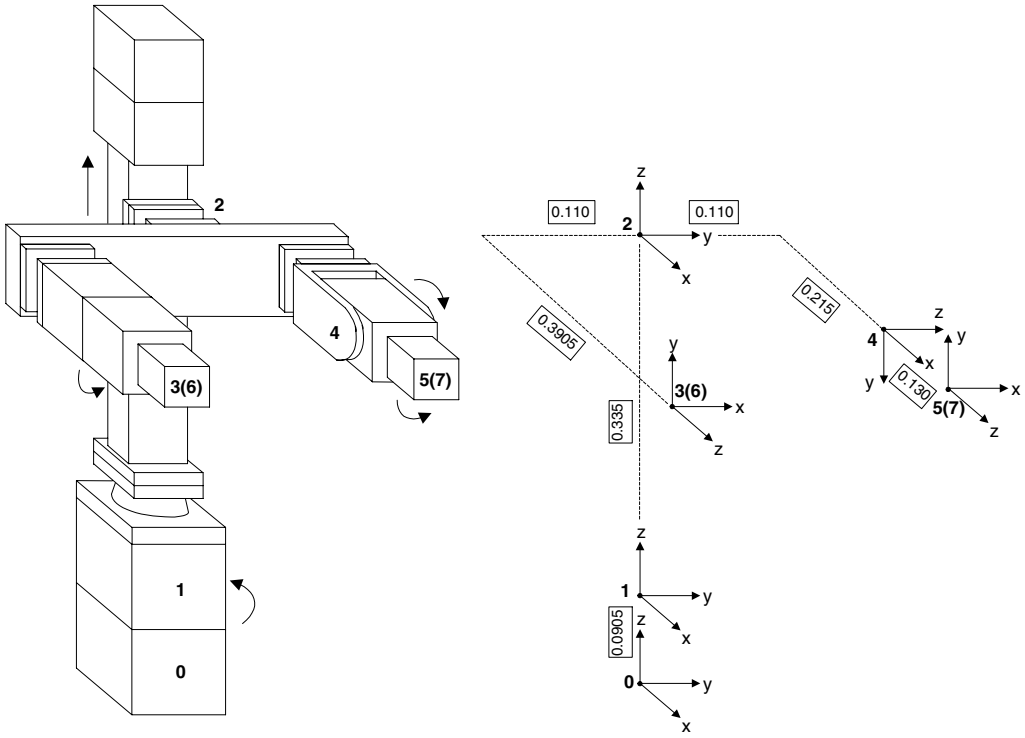

Fig. 10. Tree-type modular robot.

$$A = \begin{bmatrix} A0_{0,1} & A_{0,1}A0_{1,2} & A_{0,2}A0_{2,3} & 0 & 0 & A_{0,3}A_{3,6} & 0 \\ A0_{0,1} & A_{0,1}A0_{1,2} & 0 & A_{0,2}A0_{2,4} & A_{0,4}A0_{4,5} & 0 & A_{0,5}A_{5,7} \end{bmatrix} \in \Re^{12\times 42},$$

in which $A0_{i,j}$ and $A_{i,j}$ represent $\mathrm{Ad}_{T_{i,j}(0)}$ and $\mathrm{Ad}_{T_{i,j}}$, respectively.

The known kinematic parameters for this robot are:

$$T_{0,1}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.0905 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{1,2}(0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.355 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{2,3}(0) = \begin{bmatrix} 0 & 0 & 1 & 0.3905 \\ 1 & 0 & 0 & -0.11 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{2,4}(0) = \begin{bmatrix} 1 & 0 & 0 & 0.215 \\ 0 & 0 & 1 & 0.11 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$T_{4,5}(0) = \begin{bmatrix} 0 & 0 & 1 & 0.13 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad T_{3,6} = T_{5,7} = I_{4\times 4},$$

$$s_1 = s_3 = s_4 = s_5 = (0,0,0,0,0,1)^{\mathrm{T}}, \quad s_2 = (0,0,1,0,0,0)^{\mathrm{T}}.$$

A large number of robot poses are measured in the experiment for identifying the kinematic parameters and verifying the result of the calibration. In this tree-type robot, the two tool frames attached to the two branches are measured. The calibrated local frame poses as well as the identified kinematic parameters are listed in Table 4. Fig. 11 shows that the quantified orientation and position accuracy at both the end-effectors has an improvement of more than one order of magnitude. For branch 1, the accuracy of orientation and position has changed from 0.0346 rad and 0.012536 m initially to 0.0026 rad and 0.000276 m after calibration. Similarly for branch 2, the set of values has been improved from 0.0941 rad and 0.012086 m to 0.0030 rad and 0.000374 m. A repeatability test conducted on the robots has shown that the orientation accuracy falls in the range 0.002–0.003 rad and the position accuracy falls between 0.0002 and 0.0003 m due to the inherent poor rigidity and other nongeometric errors of the robot. The results show that the calibrated accuracy is improved to this level, hence, is only limited by the repeatability of the robot as well as the accuracy of the measuring system. The speed of processing is reflected by the fast convergence rate. The number of iterations required for calibration process averages at 2. Fig. 12 shows that the optimal number of measured poses required for the calibration is around 15. This example shows that the proposed calibration model is capable of calibrating robots with tree-structured geometry possessed by most of the multi-arm robots. Hence, the local POE calibration model is a completely general approach.

6. Summary

A robot kinematic calibration method based on local representation of the POE formula is proposed in this article. By taking advantage of the local POE formula where the local coordinate

Table 4
Identified kinematic errors and calibrated local frame poses

Dyad	δp_i	$T^c_{i-1,i}(0)$
0–1	$(-0.000127,\ 0.000331,\ -0.000462,\ 0.00146,\ -0.00072,\ -0.01184)^{\mathrm{T}}$	$\begin{bmatrix} 0.99993 & 0.011839 & -0.00073 & -0.000125 \\ -0.01184 & 0.99993 & -0.00146 & 0.000332 \\ 0.00072 & 0.00147 & 0.99999 & 0.090038 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
1–2	$(0.000050,\ -0.001060,\ -0.000462,\ -0.00015,\ 0.00217,\ -0.01184)^{\mathrm{T}}$	$\begin{bmatrix} 0.99993 & 0.01184 & 0.00217 & 0.000044 \\ -0.01184 & 0.99993 & 0.00014 & -0.001060 \\ -0.00217 & -0.00016 & 0.99999 & 0.354538 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2–3	$(-0.001049,\ 0.000070,\ -0.000609,\ 0.00357,\ -0.00631,\ -0.00910)^{\mathrm{T}}$	$\begin{bmatrix} 0.00630 & 0.00360 & 0.99997 & 0.389888 \\ 0.99994 & 0.00909 & -0.00633 & -0.111046 \\ -0.00911 & 0.99995 & -0.00354 & 0.000076 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
2–4	$(0.000662,\ 0.000530,\ -0.000009,\ -.00905,\ 0.00491,\ 0.02406)^{\mathrm{T}}$	$\begin{bmatrix} 0.99970 & -0.02408 & 0.00480 & 0.215656 \\ -0.00502 & -0.00899 & 0.99995 & 0.109987 \\ -0.02403 & -0.99967 & -0.00911 & -0.000538 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
4–5	$(-0.000073,\ -0.003011,\ -0.000241,\ 0.02367,\ 0.00090,\ 0.03981)^{\mathrm{T}}$	$\begin{bmatrix} -0.00043 & 0.02368 & 0.99972 & 0.129723 \\ -0.03981 & -0.99893 & 0.02364 & 0.003008 \\ 0.99921 & -0.03979 & 0.00137 & -0.000013 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
3–6	$(-0.000310,\ -0.000239,\ -0.000612,\ 0.00134,\ 0,\ -0.00911)^{\mathrm{T}}$	$\begin{bmatrix} 0.99996 & 0.00911 & -0.00001 & -0.000311 \\ -0.00911 & 0.99996 & -0.00134 & -0.000238 \\ -0.00001 & 0.00134 & 0.99999 & -0.000612 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
5–7	$(0.000349,\ -0.000010,\ -0.000277,\ 0.00046,\ 0.00099,\ 0.03980)^{\mathrm{T}}$	$\begin{bmatrix} 0.99921 & -0.03979 & 0.00100 & 0.000349 \\ 0.03979 & 0.99921 & -0.00044 & 0 \\ -0.00098 & 0.00048 & 0.99999 & -0.000277 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

frames can be arbitrarily assigned, the kinematic calibration becomes a process of redefining a set of new local coordinate frames to reflect the robot actual geometrical characteristics. Identification of the kinematic parameters is achieved by an iterative least squares algorithm. The advantages of this new approach are:

- it is easy to set up because the twist coordinates of the joints and the joint displacements remain in their nominal values with properly defined calibrated local frames;
- it can express different types of joints in the twist coordinates;
- it is singularity-free because the kinematic parameters described by the twist coordinates vary smoothly.

Simulation study on a 4-DOF SCARA type robot shows that the results exhibit a fully recovery of the kinematic errors at the robot tool frame under ideal experimental conditions. In addition, we have verified, through the simulation example that the modeling assumptions are tenable and the local POE model is able to handle the singularity problem. The experimental results conducted on

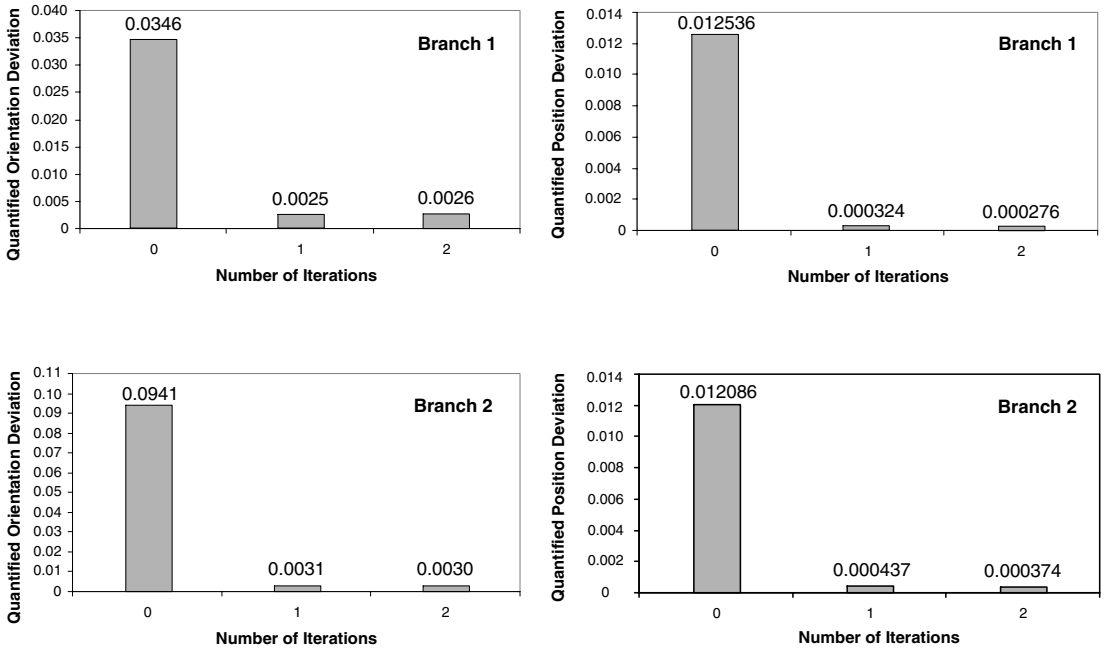

Fig. 11. Calibration convergence.

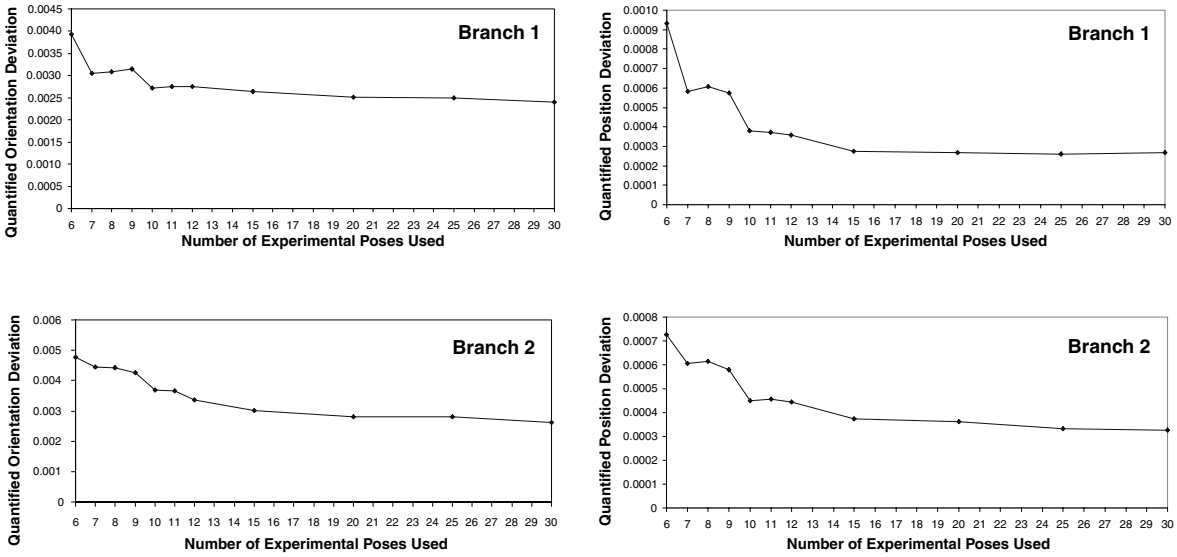

Fig. 12. Quantified deviation versus number of measurement postures used.

a tree-type modular robot also shows an improvement of pose accuracy to the order of the robot repeatability. Hence, the proposed local POE model is a generic kinematic calibration method for open chain robots with serial or tree-typed geometries. This Local POE calibration model also has been successfully implemented in an object-oriented modular robot/industrial robot simulation

software, termed SEMORS, jointly developed by Nanyang Technological University and Gintic Institute of Manufacturing Technology for rapid deployable manufacturing systems.

Acknowledgements

This project is funded by the Nanyang Technological University, Singapore, under Applied Research Grants RG64/96 and RG29/99, and the Gintic Institute of Manufacturing Technology, Singapore, under University Collaborative Project Grant, U97-A-006.

Appendix A. Geometric background

Special orthogonal group – SO(3).

$$\mathrm{SO}(3) = \{R \in \Re^{3\times 3} : RR^{\mathrm{T}} = I,\ \det R = 1\}, \tag{A.1}$$

SO(3) is also referred as the rotation group on $\Re^{3\times 1}$. Every rigid body rotation about a fixed axis can be identified with an $R \in \mathrm{SO}(3)$.

Lie Algebra of SO(3) – so(3). The Lie algebra of SO(3), denoted by so(3), is a vector space of 3×3 skew-symmetric matrices, such that

$$\mathrm{so}(3) = \{\hat{w} \in \Re^{3\times 3} : \hat{w}^{\mathrm{T}} = -\hat{w}\},$$
$$\hat{w} = \begin{bmatrix} 0 & -w_z & w_y \\ w_z & 0 & -w_x \\ -w_y & w_x & 0 \end{bmatrix}, \tag{A.2}$$

where $w = (w_x, w_y, w_z)^{\mathrm{T}} \in \Re^{3\times 1}$. $\hat{w}$ corresponds to the axis of a rigid body rotation. In exponential form, $R = \mathrm{e}^{\hat{w}q}$, where $q \in \Re$ is the angle of rotation.

Special Euclidean group – SE(3)

$$\mathrm{SE}(3) = \left\{ g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} : R \in \mathrm{SO}(3),\ p \in \Re^{3\times 1} \right\}, \tag{A.3}$$

where R is interpreted as a rigid body rotation and p as a rigid body translation. SE(3) represents the group of general rigid body motions including rotation and translation. It can also be written as an ordered pair, $g = (p, R)$.

Lie algebra of SE(3) – se(3). The Lie algebra of SE(3), denoted by se(3), can be defined as

$$\mathrm{se}(3) = \left\{ \hat{s} = \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} : v \in \Re^{3\times 1}, \hat{w} \in \mathrm{so}(3) \right\}, \tag{A.4}$$

where $\hat{s}$ admits a six-dimensional vector presentation: $s = (v, w)$, termed a twist. The twist s denotes the line coordinate of the *screw* axis of a general rigid body motion (both rotation and translation). w is the unit directional vector of the axis; v is the position of the axis relative to the origin. If $p \in \Re^{3\times 1}$ is a point on the axis and h is the ratio of translational displacement to the rotary displacement, then $v = -p \times w + hw$. In exponential form, $g = \mathrm{e}^{\hat{s}q} \in \mathrm{SE}(3)$, where $q \in \Re$ is the angle of rotation.

Adjoint representation. An element of a Lie group SE(3) can be identified with a linear mapping between its Lie algebra via the *adjoint map*. The adjoint map of $g = (p, R) \in \mathrm{SE}(3)$ on $\hat{s} \in \mathrm{se}(3)$ is

$$\mathrm{Ad}_g(\hat{s}) = g\hat{s}g^{-1} = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{w} & v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}^{-1}. \tag{A.5}$$

When the twist $\hat{s}$ admits a six-dimensional vector representation denoted by $s = (v, w) \in \Re^{6\times 1}$, the adjoint map of g on s is

$$\mathrm{Ad}_g(s) = \begin{bmatrix} R & \hat{p}R \\ 0 & R \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix}. \tag{A.6}$$

Exponential of se(3). An important connection between a Lie group, SE(3), and its Lie algebra, se(3), is the exponential mapping, defined on each Lie algebra. Let $\hat{s} \in \mathrm{se}(3)$ $(s = (v, w))$ and $\|w\|^2 = w_x^2 + w_y^2 + w_z^2$. Then

$$e^{\hat{s}} = \begin{bmatrix} e^{\hat{w}} & Av \\ 0 & 0 \end{bmatrix} \in \mathrm{SE}(3), \tag{A.7}$$

where

$$e^{\hat{w}} = I + \frac{\sin\|w\|}{\|w\|}\hat{w} + \frac{1-\cos\|w\|}{\|w\|^2}\hat{w}^2, \tag{A.8}$$

$$A = I + \frac{1-\cos\|w\|}{\|w\|^2}\hat{w} + \frac{\|w\| - \sin\|w\|}{\|w\|^3}\hat{w}^2. \tag{A.9}$$

Logarithm of SE(3). Matrix logarithm also establishes a connection between a Lie group and its Lie algebra while the Lie group is in the neighborhood of the identity. Let $R \in \mathrm{SO}(3)$ such that $\mathrm{trace}(R) \neq -1$ and $1 + 2\cos\phi = \mathrm{trace}(R)$, $\|\phi\| < \pi$. Then

$$\log \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \hat{w} & A^{-1}p \\ 0 & 0 \end{bmatrix} \in \mathrm{se}(3), \tag{A.10}$$

where

$$\hat{w} = \log[R] = \frac{\phi}{2\sin\phi}(R - R^{\mathrm{T}}), \tag{A.11}$$

$$A^{-1} = I - \frac{1}{2}\hat{w} + \frac{2\sin\|w\| - \|w\|(1+\cos\|w\|)}{2\|w\|^2\sin\|w\|}\hat{w}^2. \tag{A.12}$$

If ϕ is very small, $\hat{w} = (R - R^{\mathrm{T}})/2$.

References

[1] S.A. Hayati, Robot arm geometric parameter estimation, in: Proceedings of IEEE Conference on Decision and Control, 1983, pp. 1477–1483.

[2] S.A. Hayati, M. Mirmirani, Improving the absolute positioning accuracy of robot manipulators, Journal of Robotic Systems 112 (2) (1985) 397–413.

[3] S.A. Hayati, G.P. Roston, Inverse kinematic solution for near-simple robots and its application to robot calibration, in: Recent Trends in Robotics: Modeling, Control, and Education, Elsevier Science, Amsterdam, 1986, pp. 41–50.

[4] S.A. Hayati, G.P. Roston, Robot geometry calibration, in: Proceedings of IEEE Conference on Robotics and Automation, Philadelphia, PA, 1988, pp. 947–951.
[5] H.W. Stone, Kinematic Modeling, Identification, and Control of Robotic Manipulators, Kluwer Academic Publishers, Dordrecht, 1987.
[6] D.E. Whitney, C.A. Lozinski, J.M. Rourke, Industrial robot forward calibration method and results, Journal of Dynamic Systems 108 (1986) 1–8.
[7] C. Wu, The kinematic error model for the design of robot manipulators, in: Proceedings of the American Control Conference, 1983, pp 497–502.
[8] H. Zhuang, Z.S. Roth, Robot calibration using the CPC error model, Journal of Robotics and Computer-integrated Manufacturing 9 (3) (1992) 227–237.
[9] H. Zhuang, Z.S. Roth, A linear solution to the kinematic parameter identification of robot manipulator, IEEE Transactions on Robotics and Automation 9 (2) (1993) 174–185.
[10] H. Zhuang, Z.S. Roth, H. Fumio, A complete and parametrically continuous kinematic model for robot manipulators, IEEE Transactions on Robotics and Automation 8 (4) (1992) 451–463.
[11] K. Kazerounian, G.Z. Qian, Kinematic calibration of robotics manipulators, in: ASME Design Technology Conferences: Biennial Mechanisms Conference, 1988, pp. 261–266.
[12] K. Kazerounian, G.Z. Qian, Kinematic calibration of robotic manipulator, ASME Journal of Mechanisms, Transmission, and Automation in Design 111 (1989) 482–487.
[13] B.W. Mooring, The effect of joint axis misalignment on robot positioning accuracy, in: Proceedings of Computers in Engineering Conference and Exhibit, 1983, pp. 51–155.
[14] B.W. Mooring, An improved method for identifying the kinematic parameters in a six-axis robot, in: Proceedings of Computers in Engineering Conference and Exhibit, 1984, pp. 79–84.
[15] I.M. Chen, G. Yang, Kinematic calibration of modular reconfigurable robots using product-of-exponentials formula, Journal of Robotic Systems 14 (11) (1997) 807–821.
[16] K. Okamura, F.C. Park, Kinematic calibration and the product of exponential formula, Robotica 14 (1996) 415–421.
[17] F.C. Park, K. Okamura, Kinematic calibration and the product of exponential formula, in: Advances in Robot Kinematics and Computational Geometry, Cambridge, MIT Press, 1994, pp. 119–128.
[18] C.T. Tan, G. Yang, I.M. Chen, A modular kinematic calibration algorithm for industrial robots: simulation and experiment, in: Proceedings of the Sixth ISATED International Conference on Robotics and Manufacturing, 1998, pp. 231–234.
[19] G. Yang, I.M. Chen, Kinematic calibration of modular reconfigurable robots, in: Proceedings of International Conference on Control, Automation, Robotics, and Vision, Singapore, 1996, pp. 1845–1849.
[20] G. Yang, I.M. Chen, A novel kinematic calibration algorithm for reconfigurable robotic systems, in: Proceedings of IEEE Conference on Robotics and Automation, 1997, pp. 3197–3202.
[21] J. Denavit, R.S. Hartenberg, Kinematic notation for lower-pair mechanisms based on matrices, ASME Journal of Applied Mechanics 2 (1955) 211–215.
[22] K.C. Gupta, Kinematic analysis of manipulator using the zero reference position description, in: J.M. McCarthy (Ed.), The Kinematics of Robot Manipulators, MIT Press, Cambridge, MA, 1987, pp. 3–11.
[23] C.H. Suh, C.W. Radcliffe, Kinematics and Mechanisms Design, Wiley, Chichester, 1978.
[24] I.M. Chen, G. Yang, Configuration-independent kinematics for modular robots, in: Proceedings of IEEE Conference on Robotics and Automation, 1996, pp. 1440–1445.
[25] R. Brockett, Robotic manipulators and the product of exponential formula, in: P.A. Fuhrman (Ed.), Mathematical Theory of Network and Systems, Spriner, New York, 1984, pp. 120–129.
[26] R. Murray, Z. Li, S.S. Sastry, A Mathematical Introduction to Robotic Manipulation, CRC Press, Boca Raton, 1994.
[27] F.C. Park, R.W. Brockett, Kinematic dexterity of robotic mechanisms, International Journal of Robotics Research 13 (1) (1994) 1–15.
[28] F.C. Park, Computational aspect of manipulators via product of exponential formula for robot kinematics, IEEE Transactions on Automatic Control 39 (9) (1994) 643–647.
[29] M.L. Curtis, Matrix Groups, second ed., Springer, New York, USA, 1984.
[30] B.W. Mooring, Z.S. Roth, M.R. Driels, Fundamentals of Manipulator Calibration, Wiley, Chichester, 1991.