Self-Calibration of Three-Legged Modular Reconfigurable Parallel Robots Based on Measurement Residues

I-MING CHEN* GUILIN YANG+ WEE KIAT LIM* SONG HUAT YEO*

* *School of Mechanical & Production Engineering*
Nanyang Technological University, Singapore 639798

+*Automation Technology Division*
Gintic Institute of Manufacturing Technology, Singapore 638075

Abstract

A modular reconfigurable parallel robot system consists of a collection of standardized active and passive joint modules and custom-designed links and mobile platforms that can be assembled into many possible parallel robot configurations. Since kinematic errors, especially the assembly errors, are likely to be introduced into such a modular parallel robot, kinematic calibration becomes essential to enhance its positioning accuracy. Using the local frame representation of the Product-Of-Exponential (Local POE) formula, a linear self-calibration model is proposed for a class of three-legged modular reconfigurable parallel robots. In the local POE formula, each link has a local frame and its corresponding joint adopts the local frame representation. Since all of the local frames can be arbitrarily assigned, we are able to calibrate (relocate) them, and yet retain the nominal (local) descriptions of their respective joints to reflect the actual kinematics of the robot. The calibration criteria is defined as the measurement residues of the passive-joint displacements. The kinematic calibration thus simply becomes a procedure of fine-tuning the poses of the local frames to minimize the measurement residues. To identify the error parameters, an iterative least-square algorithm is employed. The simulation example of calibrating a three-legged (RRRS) modular parallel robot shows that the robot kinematics can be fully calibrated within three to four iterations.

1 Introduction

Parallel robots are complex kinematic systems that require extensive development effort. A modular reconfigurable parallel robot system consists of a set of independently designed modules, such as actuators, passive joints, rigid links (connectors), mobile platforms, and end-effectors, that can be rapidly assembled into various robot configurations having different kinematic and dynamic characteristics. Employing modular design to the parallel robots can shorten the development cycle, i.e., the time from design to deployment, and therefore reduce the complexity of the overall design problem to a manageable level.

In this work, a class of three-legged, nonredundant, parallel robot structures [1] has been identified for our modular robot system. Such a modular parallel robot consists of three legs. Each leg has two active joints, one passive 1-DOF (revolute) joint, and one passive 3-DOF (spherical) joint which is placed at the end of the leg. Based on this topological structure, all of the possible robot configurations can be enumerated [2]. Fig. 1 shows two such possible robot configurations that have been constructed.

One of the main concerns in using such a reconfigurable robot system is the positioning accuracy of the robot end-effector. A set of robot modules are joined together to form a complete parallel robot assembly. Machining tolerance, compliance, misalignment of the connected modules, and

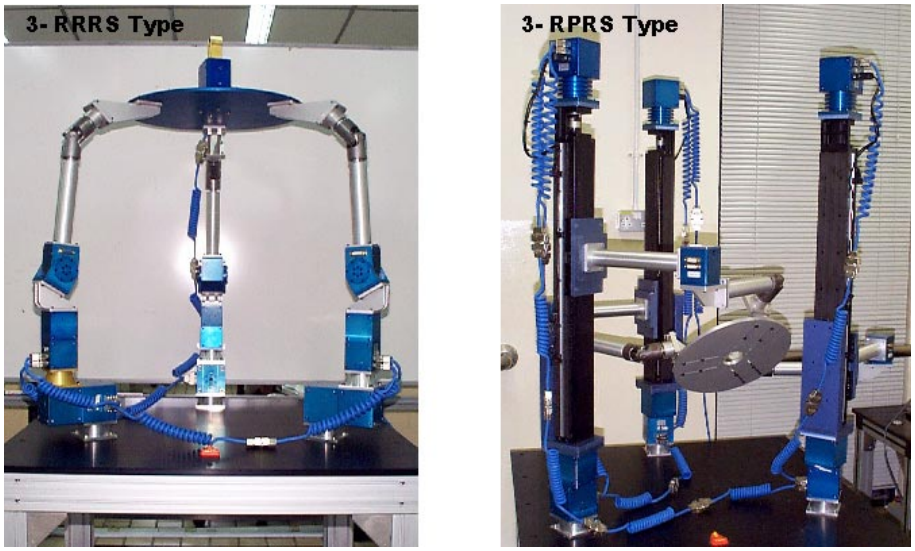


Figure 1: Two modular three-legged parallel robot configurations

wear of the connecting mechanism will affect the positioning accuracy of the robot. Hence, identifying the critical kinematic parameters to improve the positioning accuracy of the robot end-effector becomes a very important issue for modular reconfigurable parallel robots.

Because of the closed-loop structure of a parallel robot, its kinematic calibration is normally divided into two consecutive procedures: 1) *self-calibration*, and 2) *base and tool calibration*. The purpose of self-calibration is to calibrate the closed-loop mechanism itself by using the built-in sensors in the passive joints. After a robot is self-calibrated, the kinematic transformation from the robot base frame to the mobile platform frame can be computed with sufficient accuracy. The base and tool calibration, on the other hand, is to identify the fixed kinematic transformations from the world frame to the robot base frame and from the mobile platform frame to the end-effector frame by using external measuring equipment. It can be conducted only if the self-calibration has been completed. Because of the importance of the self-calibration step, past research efforts on calibration of parallel robots have been concentrated on the self-calibration techniques [3, 4, 5, 6]. A representative work in this approach was presented by Zhuang [3]. In this work, a self-calibration method was proposed for the conventional six-legged Stewart platform through the installation of redundant sensors in several passive joints and construction of a measurement residue with measured values and the computed values of these readable passive joint angles. When these passive joint angles are recorded at a sufficient number of measurement configurations, the actual kinematic parameters can be estimated by minimizing the measurement residues. Wampler, Hollerbach, and Arai [6] presented a unified formulation for the self-calibration of both serial-type robots and robotic mechanisms having closed loops by using the implicit loop method. In this method, the kinematic errors are allocated to the joints such that the loop equations are exactly satisfied. Inrascu and Park [7] developed a geometric framework for the calibration of kinematic chains containing closed loops. Both joint encoder readings and end-effector pose measurements can be uniformly included into this frame work. As a result, the kinematic calibration is cast as a nonlinear constrained optimization problem. There is only a handful works on the calibration of the three-legged parallel robots

[7, 8]. Notash and Podhorodeski [8] presented a methodology allowing kinematic calibration of three-legged parallel robot based on the minimization of the leg-end distance error. The work employs the Levenberg-Marquardt nonlinear least-square algorithm to identify the actual kinematic parameters.

We focus on the self-calibration of a class of three-legged modular parallel robots in this work. A general and effective calibration algorithm is developed for modular parallel robots base on the local frame representation of the *Products-Of-Exponentials* (POE) formula. The POE formulation method describes the joint axes based on line geometry. It is uniform in modeling manipulators with both revolute and prismatic joints. The kinematic parameters in the POE model vary smoothly with changes in joint axes so that the model can cope with certain kinematic singularity problems that can not be handled by using the conventional D-H parameterization method. Significantly, the POE formula has a well-defined differential structure such that it can be easily differentiated with respect to any of its kinematic parameters. These features make the POE formula very suitable for robot kinematic calibration [9, 10].

The remaining sections of this article are organized as follows. The kinematic modeling issues including the local POE formula, the forward displacement analysis, and the inverse kinematics are briefly addressed in Section 2. The the self-calibration model, based on the local POE formula, is formulated in Section 3. The simulation examples are then presented in Section 4. This article is summarized in Section 5.

2 Kinematics of Three-legged Modular Parallel Robots

In order to develop a self-calibration model for the three-legged modular parallel robots, two forward displacement analysis algorithms, i.e., the sensor based method and the numerical method, are briefly introduced in this section. These algorithms, based on the POE formula, are general enough to deal with the three-legged modular parallel robots having different assembly configurations. For more detail, please refer to our previous paper [11].

2.1 The Local POE Formula

In [12], Brocket 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 compact representation and its connection with Lie groups and Lie algebras, the POE formula has proven to be a useful modeling tool in robot kinematics [13, 14, 15, 16]. For our purpose, only the local frame representation of the POE formula is introduced in this article.

2.1.1 Dyad kinematics

Let link $i-1$ and link i be two adjacent links connected by joint i, as shown in Fig. 2. Link i and joint i are termed as *link assembly i*. If we denote the body coordinate frame on link assembly 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

$$T_{i-1,i}(q_i) = T_{i-1,i}(0)e^{\widehat{s_i} q_i}, \tag{1}$$

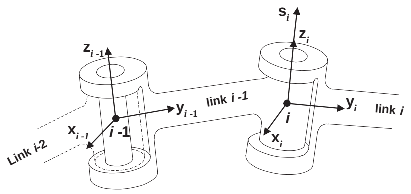


Figure 2: Two consecutive links: a dyad

where $\widehat{s}_i \in se(3)$ is the twist of joint i expressed in frame i, and $T_{i-1,i}(0) \in SE(3)$ is the initial pose of frame i relative to frame $i-1$.

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

where $R_{i-1,i}(0) \in SO(3)$ 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.

The twist of joint i can be written as

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

where $v_i = (v_{ix}, v_{iy}, v_{iz})^T$ is the positional vector of the joint axis i expressed in frame i, and $\widehat{w}_i$ is a skew-symmetric matrix related to $w_i = (w_{ix}, w_{iy}, w_{iz})^T$, which is the directional vector of joint axis i expressed in frame i. $\widehat{w}_i$ is given by

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

The twist, $\widehat{s}_i$, can also be expressed as a 6-dimensional vector through a mapping: $\widehat{s}_i \mapsto s = (v_i, w_i)^T \in \Re^{6\times 1}$, termed as *twist coordinates*. 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 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 and $\|w_i\| = 1$; $s_i = (v_i, 0)$ for prismatic joints, where v_i is the unit directional vector of the joint axis i and $\|v_i\| = 1$.

An explicit formula for the computation of $e^{\widehat{s_i q_i}}$, is given in [13, 14]. For the local POE formula, the computation formula can also be simplified as:

$$e^{\widehat{s_i q_i}} = \begin{bmatrix} e^{\widehat{w_i q_i}} & v_i q_i \\ 0 & 1 \end{bmatrix}, \tag{5}$$

where q_i is the displacement of joint i and

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

2.1.2 The Local POE formula for Open Chains

Based on the *dyad kinematics*, the forward kinematic transformation for an open kinematic chain can be easily derived. Consider an open kinematic chain with $n+1$ links, sequentially numbered as 0 , 1, ..., n (from the base 0 to the end link n). The forward kinematic transformation thus can be given by:

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

2.2 Forward Displacement Analysis

The kinematic structure of a modular three-legged (6-DOF) parallel robot is shown in Fig. 3. Each leg contains four joint modules, i.e., two actuator modules, one passive revolute (rotary or pivot) joint module, and one passive spherical joint module which is at end of the leg. It is assumed that joint ij ($\widehat{s}_{ij}$) is an active joint ($i = 1, 2, 3; j = 1, 2$), and joint $i3$ ($\widehat{s}_{i3}$) is a passive joint ($i = 1, 2, 3$). Define frame A as the local frame attached to the mobile platform and frame B as the base frame. The forward displacement analysis becomes to determine the pose of frame A with respect to the base frame B when the joint displacements of the six active joints, q_{ij} ($i = 1, 2, 3; j = 1, 2$), are known.

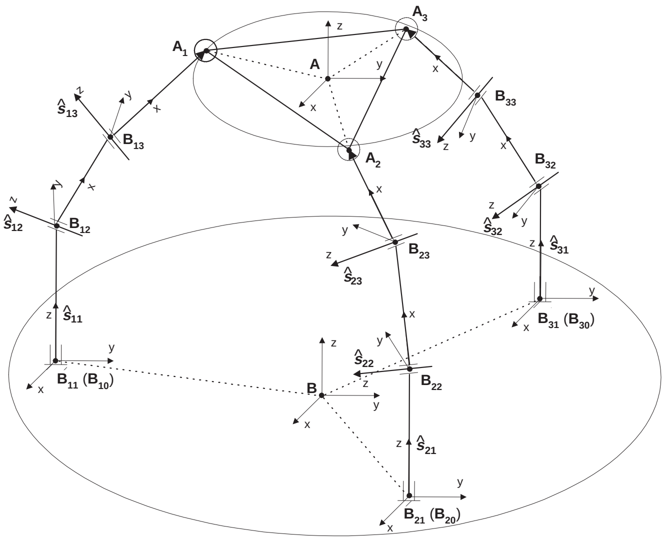


Figure 3: Kinematic structure of a three-legged parallel robot

2.2.1 Sensor-based solution approach

The sensor-based method is a simple and practical approach for the forward displacement analysis of parallel robots. The basic idea is to install a sensor in each of the passive joint modules to measure its corresponding joint displacement. In this case, the position vector of point $A_i (i = 1, 2, 3)$ with respect to the base frame B can be directly determined. It is a function of both active and passive joint displacements in leg i. Based on the local POE formula (Eqn. (7)),

p_i - the positional vector of point A_i, can be given by

$$\begin{bmatrix} p_i \\ 1 \end{bmatrix} = T_{B,i0} T_{i0,i1}(0) e^{\widehat{s_{i1}} q_{i1}} T_{i1,i2}(0) e^{\widehat{s_{i2}} q_{i2}} T_{i2,i3}(0) e^{\widehat{s_{i3}} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix}, \tag{8}$$

where $T_{B,i0}$ is the fixed pose of local frame $i0$ with respect to the base frame B and p'_i is the position vector of point A_i with respect to local frame $i3$ $(i = 1, 2, 3)$. Note that the homogeneous coordinate representation is employed in Eq. (8). Once the position vector p_i $(i = 1, 2, 3)$ is computed, we can readily determine the pose of the mobile platform such that (refer to [11] for more details):

$$T_{B,A} = \begin{bmatrix} p_1 & p_2 & p_3 & (p_2 - p_1) \times (p_3 - p_2) \\ 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} p''_1 & p''_2 & p''_3 & (p''_2 - p''_1) \times (p''_3 - p''_2) \\ 1 & 1 & 1 & 0 \end{bmatrix}^{-1}, \tag{9}$$

where p''_i is the position vector of point A_i relative to frame A.

2.2.2 Numerical solution approach

The limitation of the sensor-based algorithm is that it can only be implemented on the actual parallel robot in which each of the passive joints is sensible. In situations where the passive joint displacements are unable to be obtained, e.g., off-line computations and simulations, the iterative numerical solution method will be more practical. To mathematically derive the passive joint displacements, a differential kinematic model has been formulated by Yang et al [11] based on the POE formula. This model describes the differential relationship between the leg-end distance and the passive joint displacement in a manner:

$$da = \mathcal{J} dq, \tag{10}$$

where da and dq represents the differential changes of the leg-end distance and the passive joint displacement respectively, and $\mathcal{J}$ is the Jocabian matrix. Eqn. (10) can be written as an iterative form, i.e.,

$$q^{(k+1)} = q^{(k)} + (\mathcal{J}^{-1} da)^{(k)}, \tag{11}$$

where k represents the number of iterations. Based on the standard iterative form of Eqn. (11), the Newton Raphson method is employed to derive the numerical solution of the passive joint displacements. After the passive joint displacement q is derived, the pose of the mobile platform can be easily determined by using the rest of the sensor-based algorithm.

3 The Calibration Model

3.1 Basic Considerations

Due to the closed-loop structure of a parallel robot, the forward kinematic transformations of its legs are coupled together through the spherical joints and the unique mobile platform. The overall kinematic errors of a parallel robot are contributed by the kinematic errors in each of the legs and those in the mobile platform in a coupled manner. Based on the local POE formula, a linear self-calibration model is formulated for a class of three-legged parallel robots. The identification objective function is defined as the measurement residues, i.e., the difference between the measured values and the nominal (computed) values of the readable passive joints. The measurement residue can fully reflect the overall kinematic errors of a parallel robot. If a robot does not have any kinematic error, the measurement residue will be definitely equal to zero. Based on the local POE formula, we assume that:

- the kinematic errors in a dyad exist only in the relative initial pose;
- the joint twist coordinate and the joint angle in a dyad have no kinematic error and retain their nominal values.

Note that for the passive joints, the measurement residues, resulting from the kinematic errors in the robot, are not constant and vary with the changes of robot configurations. Hence, they are different from the constant passive-joint angle offset which, in our formulation, is assumed to be equal to zero. In this formulation, we also assume that the 3-DOF spherical joint is kinematically perfect such that its three axes intersect at one point.

3.2 Kinematic Errors in an Individual leg

Now let us first consider the kinematic errors of an individual leg. Due to the kinematic errors in the leg assembly, the actual leg-end position will be different from its nominal value. From Eqn. (8), the forward kinematic transformation of leg i can also be given by

$$\begin{bmatrix} p_i \\ 1 \end{bmatrix} = T_{B,i1}(0) e^{\widehat{s}_{i1} q_{i1}} T_{i1,i2}(0) e^{\widehat{s}_{i2} q_{i2}} T_{i2,i3}(0) e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix} \tag{12}$$

where $T_{B,i1}(0)$ is the fixed kinematic transformation from the base frame B to the initial pose of frame $i1$, $T_{B,i1}(0) = T_{B,i0} T_{i0,i1}(0)$.

According to the definition of matrix logarithm defined on $SE(3)$, there exists at least a $\widehat{t} \in se(3)$ for a given $T \in SE(3)$, such that $e^{\widehat{t}} = T$. Hence, for the initial pose $T_{i(j-1),ij}(0)$, it is sufficient to let $e^{\widehat{t}_{ij}} = T_{i(j-1),ij}(0)$ (with $e^{\widehat{t}_{i1}} = T_{B,i1}(0)$), where $\widehat{t}_{ij} \in se(3)$ $(i, j = 1, 2, 3)$. Eqn. (12) can be rewritten as

$$\begin{bmatrix} p_i \\ 1 \end{bmatrix} = e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix}. \tag{13}$$

We assume that the kinematic errors occur only in the initial pose $T_{i(j-1),ij}(0)$ (hence in $\widehat{t}_{ij}$) and the position vector p'_i. Let the kinematic errors in $\widehat{t}_{ij}$ be expressed in the local frame $i(j-1)$, denoted by $\delta\widehat{t}_{ij}$. Since $\widehat{t}_{ij} \in se(3)$, $\delta\widehat{t}_{ij}$ will also belong to $se(3)$ and $\delta e^{\widehat{t}_{ij}} = \delta\widehat{t}_{ij}\, e^{\widehat{t}_{ij}}$. In addition, the measurement residue of passive joint $i3$, δq_{i3}, will also be treated as the kinematic error when computing the positioning error of the leg-end point A_i. Linearizing Eqn. (12) with respect to $\widehat{t}_{ij}$, p'_i, and δq_{i3}, we have

$$\begin{aligned} \begin{bmatrix} \delta p_i \\ 0 \end{bmatrix} &= \delta\widehat{t}_{i1} e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix} + e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} \delta\widehat{t}_{i2} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix} \\ &+ e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} \delta\widehat{t}_{i3} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} p'_i \\ 1 \end{bmatrix} + e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \begin{bmatrix} \delta p'_i \\ 0 \end{bmatrix} \\ &+ e^{\widehat{t}_{i1}} e^{\widehat{s}_{i1} q_{i1}} e^{\widehat{t}_{i2}} e^{\widehat{s}_{i2} q_{i2}} e^{\widehat{t}_{i3}} e^{\widehat{s}_{i3} q_{i3}} \widehat{s}_{i3} \begin{bmatrix} p'_i \\ 1 \end{bmatrix} \delta q_{i3} \end{aligned} \tag{14}$$

where $\delta\widehat{t}_{ij} \in se(3)$ is the kinematic errors in $\widehat{t}_{ij}$ expressed in module frame $i(j-1)$, $\delta p_i' \in \Re^{3\times 1}$ is the kinematic error of position vector p_i' with respect to frame $i3$, and δq_{i3} is the measurement residue of passive joint $i3$. Based on the fact that $e^{\widehat{t}_{ij}} = T_{i(j-1),ij}(0)$, Eqn. (14) can also be simplified as:

$$\begin{aligned} \begin{bmatrix} \delta p_i \\ 0 \end{bmatrix} &= \delta\widehat{t}_{i1} \begin{bmatrix} p_{B,i}' \\ 1 \end{bmatrix} + T_{B,i1}\delta\widehat{t}_{i2} \begin{bmatrix} p_{i1,i}' \\ 1 \end{bmatrix} + T_{B,i2}\delta\widehat{t}_{i3} \begin{bmatrix} p_{i2,i}' \\ 1 \end{bmatrix} \\ &+ T_{B,i3} \begin{bmatrix} \delta p_i' \\ 0 \end{bmatrix} + T_{B,i3}\widehat{s}_{i3} \begin{bmatrix} p_i' \\ 1 \end{bmatrix} \delta q_{i3}, \end{aligned} \tag{15}$$

where $T_{B,ij}(j = 1,2,3)$ represents the forward kinematic transformation from frame B to frame ij and $p_{ij,i}' \in \Re^{3\times 1}$ (with $p_{i0,i}' = p_{B,i}'$) represents the position vector of point A_i with respect to frame ij.

Eqn. (15) describes the gross kinematic error of the leg-end position vector p_i that results from the kinematic errors in the initial pose $T_{i(j-1),ij}(0)$, the position vector p_i', and the measurement residue δq_{i3}. However, Eqn. (15) appears to have a nonlinear form, which is undesirable for robot calibration. With some modification, Eqn. (15) can be converted into a clear linear equation described as follows.

Let $\delta\widehat{t}$ be an element of $se(3)$ such that $\delta\widehat{t} = \begin{bmatrix} \delta\widehat{w} & \delta v \\ 0 & 0 \end{bmatrix}$ and $p \in \Re^{3\times 1}$ be a positional vector. We have

$$\delta\widehat{t} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \delta\widehat{w} & \delta v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \delta v + \delta\widehat{w}\, p \\ 0 \end{bmatrix} = \begin{bmatrix} \delta v - \widehat{p}\, \delta w \\ 0 \end{bmatrix} = \begin{bmatrix} I_{3\times 3} & -\widehat{p} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta w \end{bmatrix}. \tag{16}$$

In Eqn. (16), the matrix $[I_{3\times 3} \quad -\widehat{p}] \in \Re^{3\times 6}$ can be considered as the transition matrix related to the positional vector p. We term such a matrix the *position transition matrix* and denote it by T_p. Therefore, Eqn. (16) can be rewritten as

$$\delta\widehat{t} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} T_p \\ 0 \end{bmatrix} \delta t, \tag{17}$$

where $\delta t = (\delta w \quad \delta v)^T \in \Re^{6\times 1}$ is a $6-$dimensional vector representation of $\delta\widehat{t}$. Substituting Eqn. (17) into Eqn. (15), we have

$$\begin{aligned} \begin{bmatrix} \delta p_i \\ 0 \end{bmatrix} &= \begin{bmatrix} T_{p_{B,i}'} \\ 0 \end{bmatrix} \delta t_{i1} + T_{B,i1} \begin{bmatrix} T_{p_{i1,i}'} \\ 0 \end{bmatrix} \delta t_{i2} + T_{B,i2} \begin{bmatrix} T_{p_{i2,i}'} \\ 0 \end{bmatrix} \delta t_{i3} \\ &+ T_{B,i3} \begin{bmatrix} \delta p_i' \\ 0 \end{bmatrix} + T_{B,i3}\widehat{s}_{i3} \begin{bmatrix} p_i' \\ 1 \end{bmatrix} \delta q_{i3}. \end{aligned} \tag{18}$$

Since $T_{B,ij} = \begin{bmatrix} R_{B,ij} & p_{B,ij} \\ 0 & 1 \end{bmatrix}$, in which $R_{B,ij}$ and $p_{B,ij}$ $(i,j = 1,2,3)$ represent the orientation and position of frame ij with respect to the base frame B respectively, Eqn. (18) can also be further simplified as

$$\begin{aligned} \delta p_i &= T_{p_{B,i}'}\delta t_{i1} + R_{B,i1}T_{p_{i1,i}'}\delta t_{i2} + R_{B,i2}T_{p_{i2,i}'}\delta t_{i3} + R_{B,i3}\delta p_i' + R_{B,i3}T_{p_i'}s_{i3}\delta q_{i3} \\ &= \mathcal{J}_i\delta t_i + \mathcal{J}_{q_{i3}}\delta q_{i3}, \end{aligned} \tag{19}$$

where
$\mathcal{J}_i = \begin{bmatrix} T_{p_{B,i}'} & R_{B,i1}T_{p_{i1,i}'} & R_{B,i2}T_{p_{i2,i}'} & R_{B,i3} \end{bmatrix} \in \Re^{3\times 21}$,

$\delta t_i = [\,\delta t_{i1} \quad \delta t_{i2} \quad \delta t_{i3} \quad \delta p_i'\,]^T \in \Re^{21\times 1}$,
$\mathcal{J}_{q_{i3}} = R_{B,i3} T_{p_i'} s_{i3} \in \Re^{3\times 1}$, $s_{i3} = (v_{i3}, w_{i3})^T \in \Re^{6\times 1}$.

Apparently, Eqn. (19) is a liner equation with respect to the kinematic errors and the measurement residue. Based on this equation, a linear self-calibration model for the three-legged modular parallel robots can be formulated.

3.3 Linear Calibration Model

Now consider the three leg-end distance equations. Without loss of generality, we first consider the leg-end distance between leg 1 and leg 2 denoted by a_{12} such that

$$a_{12}^2 = (p_2 - p_1)^T (p_2 - p_1). \tag{20}$$

Differentiate Eqn. (20) with respect to p_1 and p_2, we have

$$\begin{aligned} a_{12}\delta a_{12} &= (p_2 - p_1)^T (\delta p_2 - \delta p_1) \\ &= (p_2 - p_1)^T (\mathcal{J}_2 \delta t_2 - \mathcal{J}_1 \delta t_1 + \mathcal{J}_{q_{23}} \delta q_{23} - \mathcal{J}_{q_{13}} \delta q_{13}). \end{aligned} \tag{21}$$

Similarly, for leg-end distance between leg 2 and leg 3 – a_{23} and leg-end distance between leg 3 and leg 1 – a_{31}, we have

$$a_{23}\delta a_{23} = (p_3 - p_2)^T (\mathcal{J}_3 \delta t_3 - \mathcal{J}_2 \delta t_2 + \mathcal{J}_{q_{33}} \delta q_{33} - \mathcal{J}_{q_{23}} \delta q_{23}); \tag{22}$$

$$a_{31}\delta a_{31} = (p_1 - p_3)^T (\mathcal{J}_1 \delta t_1 - \mathcal{J}_3 \delta t_3 + \mathcal{J}_{q_{13}} \delta q_{13} - \mathcal{J}_{q_{33}} \delta q_{33}). \tag{23}$$

Arranging Eqn. (20), (22) and (23) into a matrix form, a linear calibration model can be obtained

$$\mathcal{J}_{q_3} \delta q_3 = \mathcal{J} \delta t, \tag{24}$$

where
$\delta q_3 = (\,\delta q_{13} \quad \delta q_{23} \quad \delta q_{33}\,)^T \in \Re^{3\times 1}$,
$\delta t = (\,\delta t_1 \quad \delta t_2 \quad \delta t_3 \quad \delta a_{12} \quad \delta a_{23} \quad \delta a_{31}\,)^T \in \Re^{66\times 1}$,
$$\mathcal{J}_{q_3} = \begin{bmatrix} (p_2 - p_1)^T \mathcal{J}_{q_{13}} & -(p_2 - p_1)^T \mathcal{J}_{q_{23}} & 0 \\ 0 & (p_3 - p_2)^T \mathcal{J}_{q_{23}} & -(p_3 - p_2)^T \mathcal{J}_{q_{33}} \\ -(p_1 - p_3)^T \mathcal{J}_{q_{13}} & 0 & (p_1 - p_3)^T \mathcal{J}_{q_{33}} \end{bmatrix} \in \Re^{3\times 3},$$
$$\mathcal{J} = \begin{bmatrix} -(p_2 - p_1)^T \mathcal{J}_1 & (p_2 - p_1)^T \mathcal{J}_2 & 0 & -a_{12} & 0 & 0 \\ 0 & -(p_3 - p_2)^T \mathcal{J}_2 & (p_3 - p_2)^T \mathcal{J}_3 & 0 & -a_{23} & 0 \\ (p_1 - p_3)^T \mathcal{J}_1 & 0 & -(p_1 - p_3)^T \mathcal{J}_3 & 0 & 0 & -a_{31} \end{bmatrix} \in \Re^{3\times 66},$$
$a_{12} = \sqrt{(p_2 - p_1)^T (p_2 - p_1)}$,
$a_{23} = \sqrt{(p_3 - p_2)^T (p_3 - p_2)}$,
$a_{31} = \sqrt{(p_1 - p_3)^T (p_1 - p_3)}$.

Eqn. (24) can also be further simplified as the following linear calibration model:

$$Y = \mathcal{J} X, \tag{25}$$

where
$Y = \mathcal{J}_{q_3} \delta q_3 \in \Re^{3\times 1}$,
$X = \delta t \in \Re^{66\times 1}$.

δq_3 is termed as the measurement residue of passive joint angles and $\delta q_3 = q_{i3}^a - q_{i3}$, where q_{i3}^a and q_{i3} $(i = 1, 2, 3)$ represent the measured and nominal passive joint displacements respectively. The nominal passive joint displacement q_{i3} can be determined by using the numerical forward kinematic algorithm as mentioned in Section 3. In this calibration model, we have altogether 66 error parameters to be identified, which reflect the kinematic errors of a three-legged modular parallel robots.

3.4 An Iterative Least-Square Algorithm

Based on the calibration model Eqn. (25), an iterative least-square algorithm is employed for the calibration solution. To improve the calibration accuracy, we need to measure the passive joint displacements in many different robot postures. Suppose we need to take m sets of measured data. For i^{th} measurement, we can obtain a $Y_{[i]}$ as well as an identification Jacobian matrix $\mathcal{J}_{[i]}$. After m measurements, we can stack $Y_{[i]}$ and $A_{[i]}$ to form the following equation:

$$\widetilde{Y} = \widetilde{\mathcal{J}} X, \tag{26}$$

where
$\widetilde{Y} = (\, Y_{[1]}, \quad Y_{[2]}, \quad \ldots \quad , Y_{[m]}\,)^T \in \Re^{3m \times 1}$,
$X = (\, \delta t_1, \quad \delta t_2, \quad \delta t_3, \quad \delta a_{12}, \quad \delta a_{23}, \quad \delta a_{31}\,)^T \in \Re^{66 \times 1}$,
$\widetilde{\mathcal{J}} = Row[\mathcal{J}_{[1]}, \mathcal{J}_{[2]}, \ldots, \mathcal{J}_{[m]}] \in \Re^{3m \times 66}$.

Since Eqn. (26) consists of $3m$ linear equations with 66 variables (normally $m > 22$), the linear least-squares algorithm is employed for the parameter identification. The least-square solution of X is given by

$$X \quad = \quad (\widetilde{\mathcal{J}}^T \widetilde{\mathcal{J}})^{-1} \widetilde{\mathcal{J}}^T \widetilde{Y}, \tag{27}$$

where $(\widetilde{\mathcal{J}}^T \widetilde{\mathcal{J}})^{-1} \widetilde{\mathcal{J}}^T$ is the pseudoinverse of $\widetilde{\mathcal{J}}$. Due to the closed loop structure of the parallel robot, the determinant of $\widetilde{\mathcal{J}}^T \widetilde{\mathcal{J}}$ is normally very small. To avoid the computational difficulty, the Singularity Value Decomposition (SVD) method can be employed to derive the pseudoinverse of $\widetilde{\mathcal{J}}$.

The solution of Eqn. (27) can be further improved through iterative substitution as shown in Fig 4. Once the kinematic error parameter vector, X is identified, the initial pose $T_{i(j-1),ij}(0)$, the position vector p_i', and the leg-end distances a_{12}, a_{23}, and a_{31} are updated by substituting X into the following equations.

$$\begin{aligned}
T_{i(j-1),ij}(0)^{new} &= e^{\widehat{\delta t_{ij}}} T_{i(j-1),ij}(0)^{old}, \\
p_i'^{\;new} &= p_i'^{\;old} + \delta p_i', \\
a_{12}{}^{new} &= a_{12}{}^{old} + \delta a_{12}, \\
a_{23}{}^{new} &= a_{23}{}^{old} + \delta a_{23}, \\
a_{31}{}^{new} &= a_{31}{}^{old} + \delta a_{31}.
\end{aligned} \tag{28}$$

The same procedure is repeated until the norm of the error vector, $||X||$, approaches zero and the actual leg-end distances converge to some stable values. Then the final $T_{i(j-1),ij}(0)$, p_i', a_{12}, a_{23}, and a_{31} represent the calibrated kinematic parameters of robots, denoted by $T_{i(j-1),ij}^c(0)$, $p'_i{}^c$, a_{12}^c, a_{23}^c, and a_{31}^c respectively.

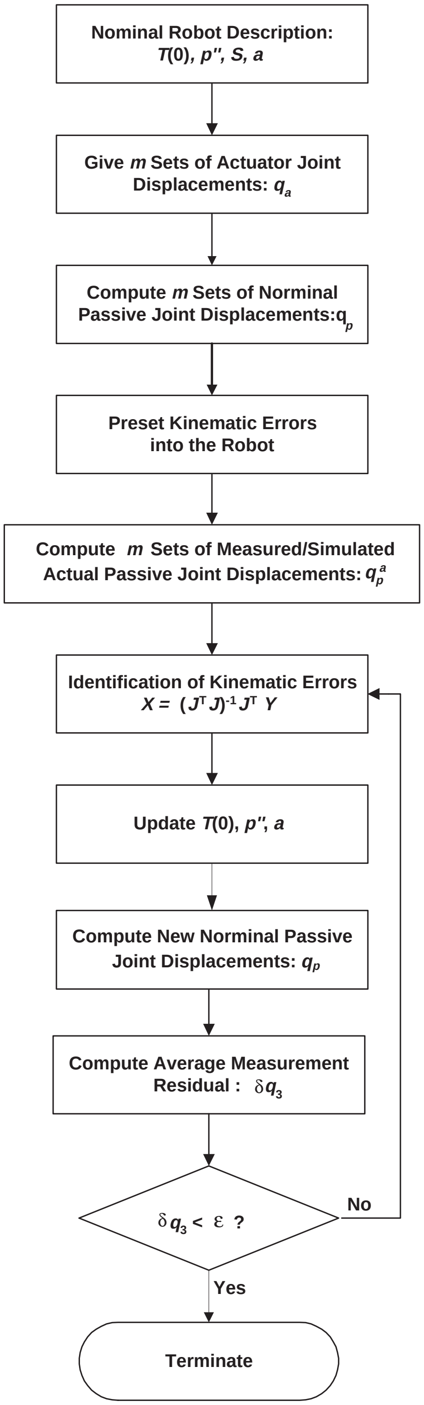

Nominal Robot Description:
T(0), p'', S, a
Give m Sets of Actuator Joint Displacements: q_a
Compute m Sets of Norminal Passive Joint Displacements:q_p
Preset Kinematic Errors into the Robot
Compute m Sets of Measured/Simulated Actual Passive Joint Displacements: q_p^a
Identification of Kinematic Errors
X = (J^T J)^-1 J^T Y
Update T(0), p'', a
Compute New Norminal Passive Joint Displacements: q_p
Compute Average Measurement Residual : δq_3
δq_3 < ε ?
No
Yes
Terminate

Figure 4: Iterative calibration loop

Note that the kinematic error vector, X, will no longer represent the actual kinematic errors after iterations. However, the actual kinematic errors can be extracted by comparing the calibrated kinematic parameters with their nominal values.

In order to evaluate the calibration result, we define a deviation metric, i.e., the average measurement residue of passive joint angles as

$$\delta q_3 = \sqrt{\frac{1}{3m}\sum_{i=1}^{m}(\delta q_{13}{}^2 + \delta q_{23}{}^2 + \delta q_{33}{}^2)}. \tag{29}$$

4 Simulation Example

In this section, a simulation example of calibrating a three-legged (6-DOF,RRRS) parallel robot is given to demonstrate the effectiveness of the calibration algorithm. As shown in the kinematic diagram in Fig. 3, the nominal kinematic parameters of the 6-DOF modular parallel robot (RRRS) are given as follows.

$$T_{B,10} = \begin{bmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 0 & -250 \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & -250\sqrt{3} \\ 0 & 0 & 1 & 90 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ T_{B,20} = \begin{bmatrix} -1 & 0 & 0 & 500 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 90 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ T_{B,30} = \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} & 0 & -250 \\ -\frac{\sqrt{3}}{2} & \frac{1}{2} & 0 & 250\sqrt{3} \\ 0 & 0 & 1 & 90 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$T_{i0,i1} = \begin{bmatrix} 0 & 0 & 1 & 90 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ T_{i1,i2} = \begin{bmatrix} 0 & 0 & -1 & -330 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};\ T_{i2,i3} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};$$

$$p''_1 = \begin{bmatrix} -205 \\ -205\sqrt{3} \\ 0 \end{bmatrix};\ p''_2 = \begin{bmatrix} 410 \\ 0 \\ 0 \end{bmatrix};\ p''_3 = \begin{bmatrix} -205 \\ 205\sqrt{3} \\ 0 \end{bmatrix};\ p'_i = \begin{bmatrix} 0 \\ -330 \\ 0 \end{bmatrix};$$

$s_{i1} = s_{i2} = s_{i3} = (0, 0, 0, 0, 0, 1)$.

Here, p''_i $(i = 1, 2, 3)$ represents nominal leg-end positions with respect to the mobile platform frame. Hence, we can compute the nominal values of the leg-end distances as:

$a_{12} = \sqrt{(p''_2 - p''_1)^T(p''_2 - p''_1)} = 410\sqrt{3}$,
$a_{23} = \sqrt{(p''_3 - p''_2)^T(p''_3 - p''_2)} = 410\sqrt{3}$,
$a_{31} = \sqrt{(p''_1 - p''_3)^T(p''_1 - p''_3)} = 410\sqrt{3}$.

Note that the units of the kinematic parameters are in radians and millimeters. The following procedures are employed for the simulation of the calibration algorithm.

1. Use the numerical forward kinematic algorithm to randomly generate 50 robot poses as well as the corresponding 50 sets of nominal passive joint angles;

2. Assign errors at the kinematic parameters such as dt_{ij}, dq_{ij}, dp'_i, and $da = [da_{12}, da_{23}, da_{31}]^T$ $(i = 1, 2, 3)$ (listed in Table 1);

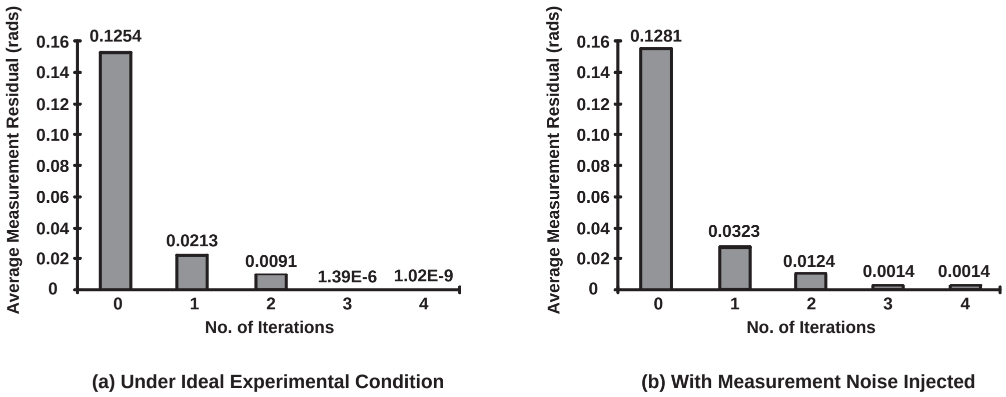
3. Find the actual initial poses in each dyad – $T^a_{i(j-1),i}(0) = e^{d\hat{t}_{ij}} T_{i(j-1),i1}(0)$, the actual position vectors of each leg end with respect to frame $i3$ – $p'^a_i = p'{}_i + dp'_i$, and the actual leg-end distances – $a^a_{12} = a_{12} + da_{12}$, $a^a_{23} = a_{23} + da_{23}$, $a^a_{31} = a_{31} + da_{31}$;

4. Determine the (simulated) actual passive joint displacements for each of the 50 poses using the numerical forward kinematics algorithm;

5. Employ iterative calibration algorithm is then employed to identify the kinematic errors.

Since each of the actuator and passive joints are assumed to be a true 1-DOF joint, the condition for the assignment of errors in each of the joint twists must satisfy $\|w_i + dw_i\| = 1$ and $(w_i + dw_i)^T(v_i + dv_i) = 0$, where $s_i = (v_i, w_i)^T$ and $ds_i = (dv_i, dw_i)^T$. Moreover, in the actual calibration experiment, all actual joint displacements, including both active and passive joints, can be directly obtained from the joints encoder readings.

The calibrated initial local frame poses as well as the kinematic errors are listed in Table 2. Since the preset and identified errors do not have the same physical meaning and are not one-to-one correspondence, the preset kinematic errors are not fully recovered. Note that the calibration solution is not unique and not necessarily identical to the actual robot. However, the success of the calibration simulation can be deduced from the results shown in Fig. 5(a), where the average measurement residue (combined for the 50 poses) is reduced from about 0.125 radians to nearly 0 radians within 4 iterations. This result shows that under the calibrated parameters description, we can directly employ the nominal joint twist coordinates and the joint displacements from both actuator and passive joint encoder readings to compute the actual kinematics of the parallel robot. In other words, the parallel robot itself is precisely calibrated.



(a) Under Ideal Experimental Condition

(b) With Measurement Noise Injected

Figure 5: Measurement residue before and after calibration

To verify the robustness of the proposed self-calibration algorithm, we have also conducted a lot of computer simulations where the measurement noise was injected. The simulation results show that the calibrated measurement residues of the passive-joint angles can always converge to the noise level. A typical simulation result is shown in Fig. 5(b), where the measurement noise of the passive joint angles is subjected to a uniform distribution in the range of $[-0.002, 0.002]rad$. Under such a noise level, the effective number of measurement poses is around fifty. With the increase in the number of measurement poses beyond fifty, there is not much significant improvement in the calibration results.

Table 1: Preset Kinematic Errors

Parameter	Preset errors	Parameter	Preset errors
dt_{ij}	$(2, 2, 2, 0.02, 0.02, 0.02)^T$	$dq_{ij}(j \neq 3)$	0.02
ds_{ij}	$(0, 0, 0, 0, \sin(0.02), -1 + \cos(0.02))^T$	$dq_{ij}(j = 3)$	0
$dp''_{\ i}$	$(2, 2, 2)^T$	da	$(2, 2, 2)^T$

Table 2: Identified Kinematic Errors

Dyad	Kinematic errors	$T^c_{i(j-1),ij}(0)$
$0-1$ (Leg1)	$\left(\begin{array}{c} -8.531, 6.599, -0.138 \\ 0.01758, 0.02277, -0.00001 \end{array}\right)^T$	$\begin{bmatrix} -0.02277 & -0.86570 & 0.50005 & -211.664 \\ 0.01758 & 0.49975 & 0.86599 & -350.039 \\ -0.99959 & 0.02851 & 0.00384 & 88.406 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$1-2$ (Leg1)	$\left(\begin{array}{c} 0.075, 8.472, 1.516, \\ 0.00031, 0.018800, 0.02967 \end{array}\right)^T$	$\begin{bmatrix} -0.01880 & 0.02966 & -0.99938 & -329.833 \\ 0.00003 & -0.99956 & -0.02967 & -1.31847 \\ -0.99982 & -0.00059 & 0.01879 & 7.719 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$2-3$ (Leg1)	$\left(\begin{array}{c} 2.037, -0.060, -0.092, \\ 0.059989, 0.00170, -0.00052 \end{array}\right)^T$	$\begin{bmatrix} -0.00169 & 0.99999 & -0.000576 & 2.037 \\ 0.05995 & -0.00047 & -0.99820 & -0.057 \\ -0.99820 & -0.00172 & -0.05995 & -0.096 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$0-1$ (Leg2)	$\left(\begin{array}{c} -1.030, -10.721, -0.144, \\ -0.02856, -0.00382, -0.00012 \end{array}\right)^T$	$\begin{bmatrix} 0.00381 & -0.00018 & -0.99999 & 408.624 \\ -0.02855 & -0.99959 & 0.00007 & -8.180 \\ -0.99959 & 0.02855 & -0.00382 & 91.536 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$1-2$ (Leg2)	$\left(\begin{array}{c} 0.097, 8.472, 1.038, \\ 0.00032, 0.01880, 0.02959 \end{array}\right)^T$	$\begin{bmatrix} -0.01880 & 0.02958 & -0.99939 & -329.815 \\ 0.00004 & -0.99956 & -0.02959 & -1.292 \\ -0.99982 & -0.00060 & 0.01879 & 7.240 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$2-3$ (Leg2)	$\left(\begin{array}{c} 2.037, -0.097, -0.076, \\ 0.05999, 0.00154, -0.00053 \end{array}\right)^T$	$\begin{bmatrix} -0.00152 & 0.99999 & -0.00057 & 2.037 \\ 0.05995 & -0.00048 & -0.99820 & -0.095 \\ -0.99820 & -0.00155 & -0.05996 & -0.080 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$0-1$ (Leg3)	$\left(\begin{array}{c} 9.536, 4.057, -0.061, \\ 0.01096, -0.01894, -0.00006 \end{array}\right)^T$	$\begin{bmatrix} 0.01894 & 0.86585 & 0.49995 & -197.146 \\ 0.01096 & 0.49982 & -0.86606 & 358.154 \\ -0.99976 & 0.02188 & -0.00002 & 90.039 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$1-2$ (Leg3)	$\left(\begin{array}{c} 0.180, 8.479, 1.242, \\ 0.01030, 0.01891, 0.02282 \end{array}\right)^T$	$\begin{bmatrix} -0.01902 & 0.02272 & -0.99956 & -329.759 \\ 0.01008 & -0.99969 & -0.02292 & 0.911 \\ -0.99977 & -0.01051 & 0.01879 & 7.486 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
$2-3$ (Leg3)	$\left(\begin{array}{c} 2.037, -0.054, -0.017, \\ 0.05999, 0.00130, -0.01060 \end{array}\right)^T$	$\begin{bmatrix} -0.00098 & 0.99994 & -0.01063 & 2.036 \\ 0.05997 & -0.01055 & -0.99815 & -0.064 \\ -0.99820 & -0.00162 & -0.05996 & -0.020 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
Position vector	Kinematic errors	$p'^{\,c}_i$
p'_1	$(-3.914, 2.099, 0.059)^T$	$(-3.914, -327.901, 0.059)^T$
p'_2	$(-3.968, 2.100, 0.022)^T$	$(-3.968, -327.900, 0.022)^T$
p'_3	$(-3.947, 2.100, 0.073)^T$	$(-3.947, -327.900, 0.073)^T$
Leg-end distance	Kinematic errors	$(a^c_{12}, a^c_{23}, a^c_{31})^T$
(1-2,2-3,3-1)	$(-1.822, -1.822, -1.822)^T$	$(711.963, 711.963, 711.963)^T$

5 Conclusion

In this paper, a self-calibration model is proposed for the kinematic calibration of a class of three-legged modular parallel robots. The identification objective is defined as the measurement residues of the passive joint displacements. By taking advantage of the local POE formula where the local coordinate can be arbitrarily assigned, the kinematic calibration is modeled as a process of redefining a set of new local coordinate frames to reflect the robot actual geometrical characteristics. Since the calibrated local frames are defined in such a way that makes the twist of the joints and the joint displacements remain in their nominal values, the resulting calibration model is greatly simplified. Simulation studies on a 6-DOF (RRRS) modular parallel robot shows that the results exhibit full recovery of the kinematic errors. Future work will be focused on experimental study of the proposed self-calibration algorithm.

Acknowledgment

This work is supported by both the Nanyang Technological University, Singapore, under Applied Research Grant RG29/99 and Gintic Institute of Manufacturing Technology, Singapore, under University Collaborative Project Grant, U97-A-006.

References

[1] Podhorodeski R.P. and Pittens K.H. 1994, A class of parallel manipulators based on kinematically simple branches. *ASME Journal of Mechanical Design*, 116:908–914.

[2] Yang G., Chen I.M., Lim W.K., and Yeo S.H. 2001, Kinematic design of modular reconfigurable in-parallel parallel robots. *Autonomous Robots*, 10:83–89).

[3] Zhuang, H. 1997, Self-calibration of parallel mechanisms with a case study on stewart platform. *IEEE Transactions on Robotics and Automation*, 13(3):387–397.

[4] Zhuang H. and Liu L. 1996, Self calibration of a class of parallel manipulators. In *Proceedings of IEEE Conference on Robotics and Automation*, Minneapolis, Minnesota, April, pages 994–999.

[5] Bennett D.J. and Hollerbach J.M. 1991, Autonomous calibration of single-loop closed kinematic chains formed by manipulators with passive endpoint constraints. *IEEE Transactions on Robotics and Automation*, 7(5):597–606.

[6] Wampler C.W., Hollerbach J.M., and Arai T. 1995, An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Transactions on Robotics and Automation*, 11(5):710–724,.

[7] Iurascu C.C. and Park F.C. 1999, Geometric algorithms for closed chain kinematic calibration. In *Proceedings of IEEE Conference on Robotics and Automation*, Detroit, USA, May, pages 1752–1757.

[8] L. Notash and R.P. Podhorodeski. 1995, Kinematic calibration of parallel manipulators. In *Proceedings of IEEE Conference on Systems, Man, and Cybernetics*, pages 3310–3315.

[9] Park F.C. and Okamura. K. 1994, Kinematic calibration and the product of exponential formula. In *Advances in Robot Kinematics and Computational Geometry*, Cambridge, MIT Press, pages 119–128..

[10] Chen I.M. and Yang. G. 1997, Kinematic calibration of modular reconfigurable robots using product-of-exponentials formula. *Journal of Robotic Systems*, 14(11):807–821.

[11] Yang G., Chen I.M., Lim W.K., and Yeo S.H. 1999, Kinematic design of modular reconfigurable in-parallel robots. In *Proceedings of IEEE Conference on Robotics and Automation*, May, pages 2501–2506.

[12] Brockett. R. 1983, Robotic manipulators and the product of exponential formula. In *International Symposium in Math. Theory of Network and Systems*, Beer Sheba, Israel, pages 120–129.

[13] Murray R., Li Z. , and Sastry. S.S., 1994, *A Mathematical Introduction to Robotic Manipulation.* CRC Press.

[14] Park. F.C. 1994, Computational aspect of manipulators via product of exponential formula for robot kinematics. *IEEE Transactions on Automatic Control*, 39(9):643–647.

[15] Chen I.M. and Yang G. ang Kang I.G. 1999, Numerical inverse kinematics for modular reconfigurable robots. *Journal of Robotic Systems*, 16:213–225.

[16] Yang G. 1999, *Kinematics, dynamics, calibration, and configuration optimization of modular reconfigurable robots.* PhD thesis, School of Mechanical and Production Engineering, Nanyang Technological University, Singapore.