

Self-calibration of three-legged modular reconfigurable parallel robots based on leg-end distance errors

Guilin Yang,† I-Ming Chen,* Wee Kiat Lim,* Song Huat Yeo*

† Automation Technology Division, Gintic Institute of Manufacturing Technology, (Singapore) 638075

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

(Received in Final Form: July 3, 2000)

SUMMARY

A class of three-legged modular reconfigurable parallel robots is designed and constructed for precision assembly and light machining tasks by using standard active and passive joint modules in conjunction with custom designed links and mobile platforms. Since kinematic errors, especially the assembly errors, are likely to be introduced, kinematic calibration becomes particularly important to enhance the positioning accuracy of a modular reconfigurable robot. Based on the local frame representation of the Product-Of-Exponentials (Local POE) formula, a self-calibration method is proposed for these three-legged modular reconfigurable parallel robots. In this method, both revolute and prismatic joint axes can be uniformly expressed in twist coordinates by their respective local (body) frames. Since these local frames can be arbitrarily defined on their corresponding links, we are able to calibrate them, and yet retain the nominal local description of their respective joints, i.e., the nominal twist coordinates and nominal joint displacements, to reflect the actual kinematics of the robot. The kinematic calibration thus becomes a procedure of fine-tuning the locations and orientations of the local frames. Using mathematical tools from differential geometry and group theory, an explicit linear calibration model is formulated based on the leg-end distance errors. An iterative least-square algorithm is employed to identify the error parameters. A simulation example of calibrating a three-legged (RRRS) modular parallel robot shows that the robot kinematics can be fully calibrated within two to three iterations.

KEYWORDS: Parallel robots; Self-calibration; Leg-end distance; Local POE.

1. INTRODUCTION

A parallel robot is a closed-loop mechanism in which the mobile platform is connected to the base by at least two serial kinematic chains (legs). Applications of this type of robot can be found in the pilot training simulators and in the high precision surgical tools because of the high force loading capacity and fine motion characteristics of the closed-loop mechanism. Recently, researchers are trying to utilize these advantages to develop parallel-type robot based multi-axis machining tools¹ and precision assembly tools.² However, the design, trajectory planning, and application

development of the parallel robot are difficult and tedious. This is because the closed-loop nature of the mechanism limits the motion of the platform and creates complex kinematic singularities in the workspace of the mobile platform. To overcome this drawback, we employ modular design concept in the development of parallel robots. A modular 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 a complete robot with various configurations having different kinematics and dynamic characteristics. The concept of modularity has been previously used in the design of serial-type robots for flexibility, ease of maintenance, and rapid deployment.^{3–6} A modularly designed reconfigurable parallel robot not only possess the above advantages but can also shorten the development cycle, i.e., the time from design, construction, to deployment. The modular design is able to reduce the complexity of the overall design problem to a manageable level.

One of the main concerns in the modular 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. Factors like machining tolerance, compliance, misalignment of the connected modules, and wear of the connecting mechanism will affect the positioning accuracy of the robot. As a result, the assembly errors of a modular robot are usually larger than those of a robot having fixed configuration. 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.

Based on the closed-loop structure of the parallel robot, the calibration procedure is normally divided into two steps, i.e. self-calibration of the closed-loop mechanism and calibration of the end-effector. The purpose of self-calibration is to calibrate the closed-loop mechanism by using the built-in sensors in the passive joints. The end-effector calibration, on the other hand, is to improve the absolute positioning accuracy of the end-effector by using external measuring equipment. Once the parallel robot is self-calibrated, the subsequent end-effector calibration can be easily performed. In this step, only the kinematic errors in the fixed transformations from the robot world frame to robot base frame and from the mobile platform frame to the end-effector frame need to be identified. 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.^{7–12} A representative work in this approach is presented by Zhuang.⁷ In this work, a self-calibration method is proposed for the conventional six-legged Stewart platform through the installation of redundant sensors in several passive joints and constructing a measurement residues 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. Since this model is based on the linearization of the kinematic constraint equations, it converges rapidly. Wampler, Hollerbach, and Arai¹² presents a unified formulation for the self-calibration of both serial-link robots and robotic mechanisms having kinematically 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, simplifying the treatment of multi-loop mechanisms. Inrascu and Park¹³ develop a unified 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 of works on the calibration of the three-legged manipulators.^{13,14} Notash and Podhorodeski¹⁴ presents a methodology allowing kinematic calibration of three-legged parallel manipulator based on the minimization of the leg-end distance error. The work employs the Lavenberg-Marquardt nonlinear least-square algorithm to identify kinematic parameters of the manipulator.

This paper is focused on the self-calibration of a class of three-legged modular parallel robots. The calibration objective is, though minimizing the leg-end distance errors, to identify the kinematics errors in the robots. A general and effective linear calibration algorithm is developed for modular parallel robots based on the local frame representation of the *Products-Of-Exponentials* (POE) formula. The POE representation method describes the joint axes as twists based on line geometry. It is, therefore, 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 formulation singularity problems that can not be handled by using other kinematic parameterization methods. Significantly, the POE formula has a very explicit differential structure such that it can be easily differentiated with respect to any of its kinematic parameters.^{15–17} Since a modular reconfigurable parallel robot has unlimited configurations, it is necessary for the calibration algorithm to be as generic to robot configurations. For these reasons, the POE formula would be the most appropriate modeling tool for the modular parallel robot calibration.

The remaining sections of this article are organized as follows: The design consideration for robot modules and possible parallel robot configurations are briefly introduced in Section 2. The kinematic modeling issues including the

local POE formula, the forward displacement analysis, and the inverse kinematics are briefly addressed in Section 3. The self-calibration model, based on the local POE formula, is presented in Section 4. Simulation examples are included in Section 5. The summary is in Section 6.

2. DESIGN CONSIDERATIONS

2.1 Robot modules

A set of commercial grade and custom designed actuator modules, passive joint modules, rigid link modules (connectors), and mobile platforms are considered as the basic parallel robot modules. Off-the-shelf intelligent mechatronic drives, PowerCube, from Amech GmbH, Germany are selected as actuator modules for rapid deployment. Both prismatic (Fig. 1(a)) and revolute (Fig. 1(b)) actuator modules are considered. Each of the actuator modules is a self-contained drive unit with a built-in motor, a controller, an amplifier, and the communication interface. It has a cubic or double-cube design with multiple connecting sockets so that two actuator modules can be connected in many different orientations. Three types of passive joint modules (without actuators) are designed and fabricated: the rotary joint (Fig. 2(a)), the pivot joint (Fig. 2(b)), and the spherical

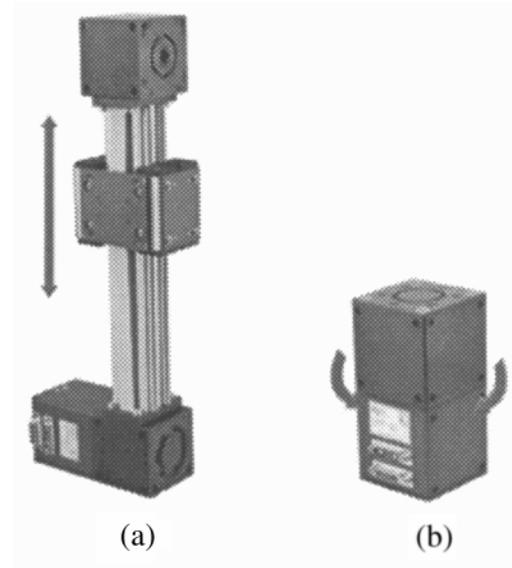


Fig. 1. Actuator modules.

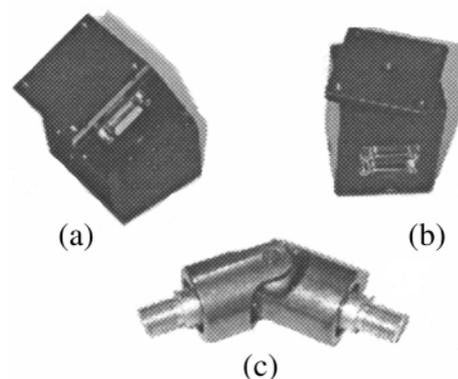


Fig. 2. Passive joint modules.

joint (Fig. 2(c)). To facilitate the forward kinematics analysis, an angular displacement sensor is built into each of the passive rotary and pivot joint modules. A set of rigid links with various geometrical shapes and dimensions has been customly designed for connecting joint modules (Fig. 3). A circular mobile platform has also been designed and fabricated (Fig. 4), which can be used along with parallel robots with various number of legs.

2.2 Possible robot configurations

Based on the module designs, many possible parallel robot configurations can be constructed. Here, however, we mainly focus on the 6-DOF, nonredundant, parallel robot configurations. An enumeration scheme for such parallel robot topological structures is presented in.¹⁸ In this work, a class of three-legged, nonredundant, parallel robots is identified as having simple kinematics and desirable characteristics. Such a 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 fact, all of the

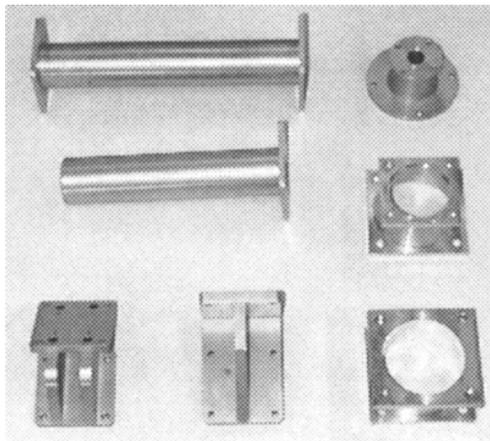


Fig. 3. Custom designed links (connectors).

possible robot configurations can be enumerated (refer to reference [18] for more detail). Figure 5 shows two such possible robot configurations.

3. KINEMATICS OF THREE-LEGGED MODULAR PARALLEL ROBOTS

In order to develop a self-calibration model for the three-legged parallel robots, the kinematic algorithms of both forward and inverse kinematics analysis 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.¹⁹

3.1 The local POE formula

Brockett²⁰ 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

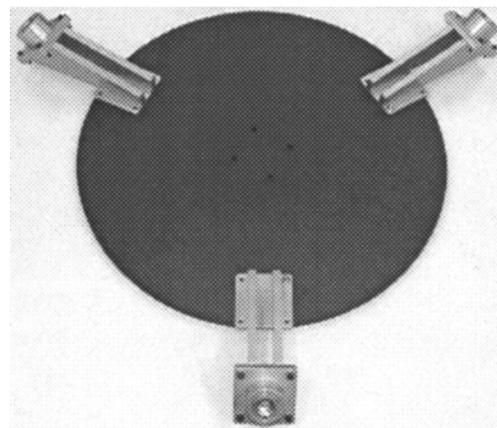


Fig. 4. Custom designed mobile platform.

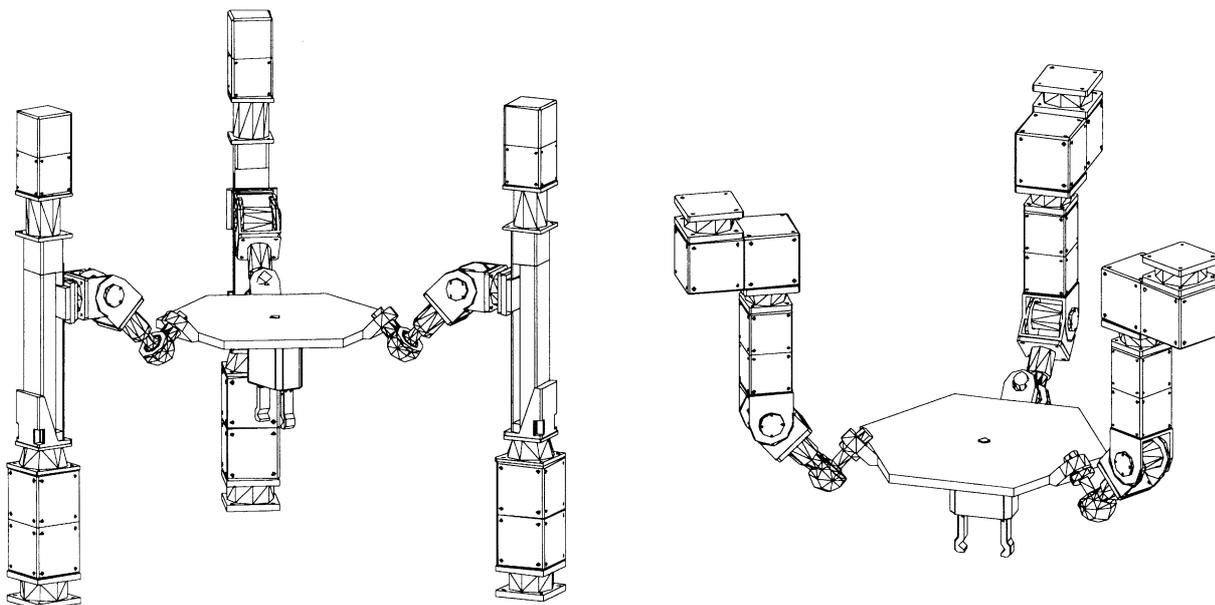


Fig. 5. Two modular 3-leg parallel robot configurations.

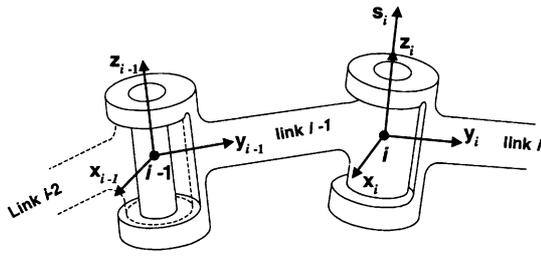


Fig. 6. Two consecutive links: a dyad.

formula has proven to be a useful modeling tool in robot kinematics.^{16,20–23} For our purpose, only the local frame representation of the POE formula is introduced in this article.

3.1.1. Dyad kinematics. Let link $i - 1$ and link i be two adjacent links connected by joint i , as shown in Fig. 6. 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

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

where $\hat{s}_i \in se(3)$ 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 $d_{i-1,i}(0) \in \mathbb{R}^{3 \times 1}$ are the initial orientation and position of frame i relative to frame $i - 1$, respectively.

The twist of joint i can be written as

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

where $v_i (= (v_{ix}, v_{iy}, v_{iz})^T)$ is the positional vector of the joint axis i expressed in frame i , and \hat{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 . \hat{w}_i 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 6-dimensional vector through a mapping: $\hat{s}_i \mapsto s = (v_i, w_i)^T \in \mathbb{R}^{6 \times 1}$, 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 $e^{\hat{s}_i q_i}$ is given in references 20 and 21. For the local POE formula, it can also 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 the displacement of joint i and

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

3.1.2. 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 $0, 1, \dots, n$ (from the base 0 to the end link n). The forward kinematic transformation thus can be given by:

$$T_{0,n}(q_1, q_2, \dots, q_n) = T_{0,1}(q_1)T_{1,2}(q_2) \dots T_{(n-1),n}(q_n) \\ = \prod_{i=1}^n (T_{(i-1),i}(0)e^{\hat{s}_i q_i}). \tag{7}$$

3.2 Forward displacement analysis

We consider a class of modular three-legged (6-DOF) parallel robots as shown in Fig. 7. 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 the end of the leg. We assume that joint ij (\hat{s}_{ij}) is an active joint ($i = 1, 2, 3$; $j = 1, 2$), and joint $i3$ (\hat{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.

3.2.1 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 the 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

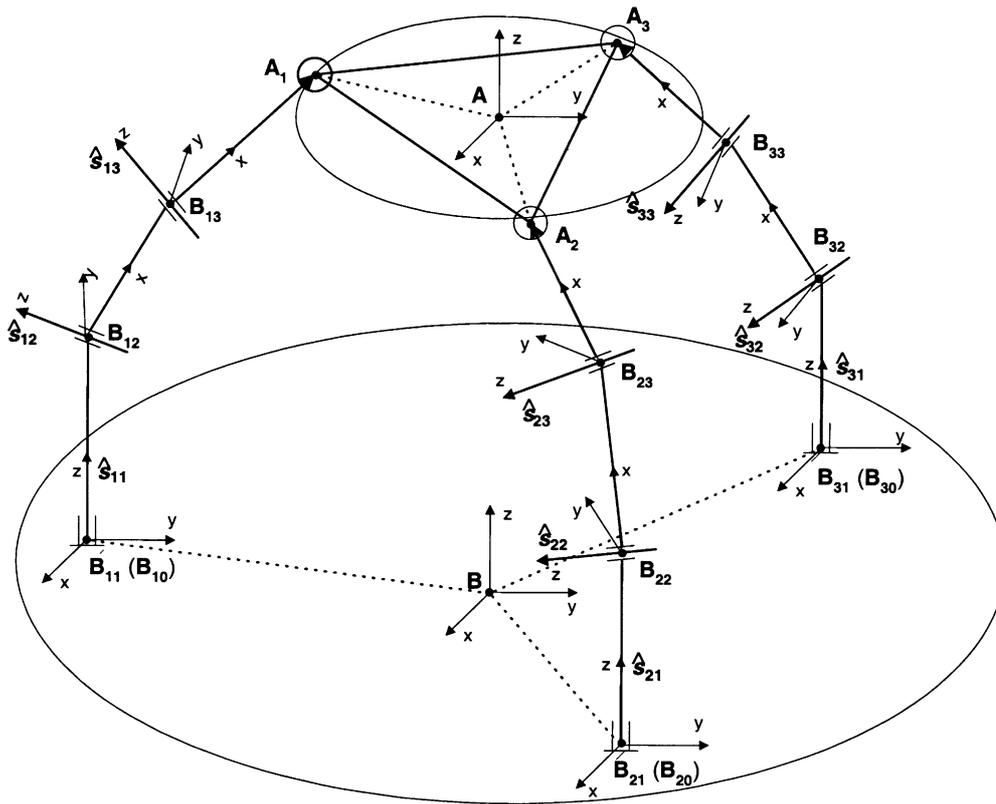


Fig. 7. A 3-leg parallel robot.

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

where $T_{B,i0}$ is the fixed kinematic transformation from base frame B to local frame $i0$ 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 Eqn. (8). Once the position vector p_i ($i = 1, 2, 3$) is computed, we can determine the pose of the mobile platform.¹⁹

3.2.2 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 measurable. Hence, 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.*¹⁹ based on the POE formula. This model describes the differential relationship between the leg-end distance and the passive joint displacement in a manner:

$$da = J_f dq, \tag{9}$$

where da and dq represents the differential changes of the leg-end distance and the passive joint displacement respectively and J_f is the Jacobian matrix of the forward

displacement analysis. Eqn. (9) can be written as an iterative form, i.e.,

$$q^{(k+1)} = q^{(k)} + (J_f^{-1} da)^{(k)}, \tag{10}$$

where k represents the number of iterations. Based on the standard iterative form of Eqn. (10), 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.

3.3 Inverse Kinematic Analysis

The purpose of the inverse kinematic analysis for parallel modular robots is to determine the active joint displacements when the desired pose of the mobile platform is given. As shown in Fig. 7, when the pose of mobile platform frame with respect to the base frame $T_{B,A}$ is given, the position vector of point A_i with respect to the base frame, i.e., p_i ($i = 1, 2, 3$), can be directly determined through a simple kinematic transformation such that

$$\begin{bmatrix} p_i \\ 1 \end{bmatrix} = T_{B,A} \begin{bmatrix} p''_i \\ 1 \end{bmatrix}, \tag{11}$$

where p''_i is the position vector of point A_i with respect to the mobile platform frame A .

Therefore, each of the three legs can be treated as an independent serial open chain. Because of the simple kinematic structures of the legs, most of the inverse kinematic algorithms for the serial manipulators can be

implemented here for the solutions. It should be noted that the inverse solution of the parallel manipulator exists if and only if the inverse solutions exist for every individual leg. Following the POE approach, we employ the Paden-Kahan's method²² for the inverse kinematic analysis.

4. THE CALIBRATION MODEL

4.1 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. Traditionally, the self-calibration model of a parallel robot is formulated through constructing a measurement residual with measured values and the computed values of the readable passive joints. Although the calibration idea is simple, the model formulation is complicated. Besides, during the parameter identification process, the passive joint displacements need to be updated at each of the iterations by using the numerical forward displacement analysis algorithm, which lowers the computational efficiency.

Based on the local POE formula, a simple self-calibration model is formulated for a class of three-legged parallel robots in this section. The identification objective function is defined as the leg-end distance errors because they are very sensitive to the variations of the kinematic parameter so as to speed up the identification process. In order to simplify the calibration model, we assume that (1) the kinematic errors in a dyad exist only in the relative initial pose, (2) the joint twist coordinate and the joint angle offset in a dyad have no kinematic error and remain in their nominal values. Since all the readable passive joints can be considered as the active joints, they will always take their measured values during the identification process and the forward displacement analysis at each iteration becomes unnecessary. Note that in this formulation, we also assume that the 3-DOF spherical joint is kinematically perfect such that its three axes intersect at one point.

4.2 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

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

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 $\hat{t} \in se(3)$ for a given

$T \in SE(3)$, such that $e^{\hat{t}} = T$. Hence, for the initial pose $T_{i(j-1),ij}(0)$, it is sufficient to let $e^{\hat{t}^{ij}} = T_{i(j-1),ij}(0)$ (with $e^{\hat{t}^{i1}} = T_{B,i1}(0)$), where $\hat{t}_{ij} \in se(3)$ ($i, j = 1, 2, 3$). Eqn. (12) can be rewritten as

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

For each of the legs, we assume that the kinematic errors occur only in the initial pose $T_{i(j-1),ij}(0)$ (hence in \hat{t}_{ij}) and the position vector p'_i . Let the kinematic errors in \hat{t}_{ij} be expressed in the local frame $i(j-1)$, denoted by $\delta\hat{t}_{ij}$. Since $\hat{t}_{ij} \in se(3)$, $\delta\hat{t}_{ij}$ also belongs to $se(3)$. Geometrically, $\delta e^{\hat{t}_{ij}} = \delta\hat{t}_{ij}e^{\hat{t}_{ij}}$. Linearizing Eqn. (12) with respect to \hat{t}_{ij} and p'_i , we have

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

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

$$\begin{aligned} \begin{bmatrix} \delta p_i \\ 0 \end{bmatrix} &= \delta\hat{t}_{i1} \begin{bmatrix} p'_{B,i} \\ 1 \end{bmatrix} + T_{B,i1}\delta\hat{t}_{i2} \begin{bmatrix} p'_{i1,i} \\ 1 \end{bmatrix} \\ &+ T_{B,i2}\delta\hat{t}_{i3} \begin{bmatrix} p'_{i2,i} \\ 1 \end{bmatrix} + T_{B,i3} \begin{bmatrix} \delta p'_i \\ 0 \end{bmatrix}, \quad (15) \end{aligned}$$

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

Eqn. (15) is actually a differential equation. It describes the gross kinematic error of the leg-end position vector p_i

resulting from the kinematic errors in the initial pose $T_{i(j-1),ij}(0)$ and the position vector p'_i . However, Eqn. (15) appears to have a nonlinear form, which is undesirable in robot calibration. With some modification, Eqn. (15) can be converted into a clear linear equation as described as follows.

Let $\delta\hat{t}$ be an element of $se(3)$ such that $\delta\hat{t} = \begin{bmatrix} \delta\hat{w} & \delta v \\ 0 & 0 \end{bmatrix}$

and $p \in \mathfrak{R}^{3 \times 1}$ be a positional vector. We have

$$\begin{aligned} \delta\hat{t} \begin{bmatrix} p \\ 1 \end{bmatrix} &= \begin{bmatrix} \delta\hat{w} & \delta v \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} \delta v + \delta\hat{w} p \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \delta v - \hat{p} & \delta w \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & -\hat{p} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \delta v \\ \delta w \end{bmatrix}. \end{aligned} \quad (16)$$

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

$$\delta\hat{t} \begin{bmatrix} p \\ 1 \end{bmatrix} = \begin{bmatrix} T_p \\ 0 \end{bmatrix} \delta t, \quad (17)$$

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

$$\begin{aligned} \begin{bmatrix} \delta p_i \\ 0 \end{bmatrix} &= \begin{bmatrix} T_{p'_{b,i}} \\ 0 \end{bmatrix} \delta t_{i1} + T_{B,il} \begin{bmatrix} T_{p'_{i1,i}} \\ 0 \end{bmatrix} \delta t_{i2} \\ &+ T_{B,i2} \begin{bmatrix} T_{p'_{i2,i}} \\ 0 \end{bmatrix} \delta t_{i3} + T_{B,i3} \begin{bmatrix} \delta p'_i \\ 0 \end{bmatrix}. \end{aligned} \quad (18)$$

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

$$\begin{aligned} \delta p_i &= T_{p'_{b,i}} \delta t_{i1} + R_{B,il} T_{p'_{i1,i}} \delta t_{i2} + R_{B,i2} T_{p'_{i2,i}} \delta t_{i3} + R_{B,i3} \delta p'_i \\ &= J_i \delta t_i, \end{aligned} \quad (19)$$

where

$$J_i = [T_{p'_{b,i}} \quad R_{B,il} T_{p'_{i1,i}} \quad R_{B,i2} T_{p'_{i2,i}} \quad R_{B,i3}] \in \mathfrak{R}^{3 \times 21},$$

$$\delta t_i = (\delta t_{i1}, \delta t_{i2}, \delta t_{i3}, \delta p'_i)^T \in \mathfrak{R}^{21 \times 1}.$$

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

4.3 Linear calibration model

Now let us consider the differential change of the leg-end distance resulting from the kinematic errors. Without loss of generality, we first consider the leg-end distance between leg 1 and leg 2 denoted by a_{12} such that

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

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

$$\begin{aligned} \delta a_{12} &= \frac{(p_2 - p_1)^T}{a_{12}} (\delta p_2 - \delta p_1) \\ &= \frac{(p_2 - p_1)^T}{\sqrt{(p_2 - p_1)^T (p_2 - p_1)}} (J_2 \delta t_2 - J_1 \delta t_1). \end{aligned} \quad (21)$$

Eqn. (21) describes the differential change of a_{12} resulting from the kinematic errors in leg 1 and leg 2. Similarly, for the leg-end distance between leg 2 and leg 3: a_{23} and the leg-end distance between leg 3 and leg 1: a_{31} , we have

$$\delta a_{23} = \frac{(p_3 - p_2)^T}{\sqrt{(p_3 - p_2)^T (p_3 - p_2)}} (J_3 \delta t_3 - J_2 \delta t_2); \quad (22)$$

$$\delta a_{31} = \frac{(p_1 - p_3)^T}{\sqrt{(p_1 - p_3)^T (p_1 - p_3)}} (J_1 \delta t_1 - J_3 \delta t_3). \quad (23)$$

If we know the actual values of a_{12} , a_{23} , and a_{31} denoted by a_{12}^a , a_{23}^a , and a_{31}^a , respectively, the differential change of the leg-end distance can be given by

$$\delta a = \begin{bmatrix} \delta a_{12} \\ \delta a_{23} \\ \delta a_{31} \end{bmatrix} = \begin{bmatrix} a_{12}^a - a_{12} \\ a_{23}^a - a_{23} \\ a_{31}^a - a_{31} \end{bmatrix}, \quad (24)$$

where a_{12} , a_{23} , and a_{31} represents their computed nominal values determined by the sensor readings of the passive joint displacements. Therefore, they can be given by $\frac{\sqrt{(p_2 - p_1)^T (p_2 - p_1)}}{a_{12}^a}$, $\frac{\sqrt{(p_3 - p_2)^T (p_3 - p_2)}}{a_{23}^a}$, and $\frac{\sqrt{(p_1 - p_3)^T (p_1 - p_3)}}{a_{31}^a}$, respectively. Note that a_{12} , a_{23} , and a_{31} are different from their original theoretical values. Arranging Eqn. (21), (22), and (23) into a matrix form, a linear calibration model can be obtained:

$$\delta a = J \delta t, \quad (25)$$

where

$$\delta a = \begin{bmatrix} a_{12}^a - \sqrt{(p_2 - p_1)^T(p_2 - p_1)} \\ a_{23}^a - \sqrt{(p_3 - p_2)^T(p_3 - p_2)} \\ a_{31}^a - \sqrt{(p_1 - p_3)^T(p_1 - p_3)} \end{bmatrix} \in \mathfrak{R}^{3 \times 1},$$

$$J = \begin{bmatrix} -\frac{(p_2 - p_1)^T}{\sqrt{(p_2 - p_1)^T(p_2 - p_1)}} J_1 & \frac{(p_2 - p_1)^T}{\sqrt{(p_2 - p_1)^T(p_2 - p_1)}} J_2 \\ 0 & -\frac{(p_3 - p_2)^T}{\sqrt{(p_3 - p_2)^T(p_3 - p_2)}} J_2 \\ \frac{(p_1 - p_3)^T}{\sqrt{(p_1 - p_3)^T(p_1 - p_3)}} J_1 & 0 \\ 0 & \frac{(p_3 - p_2)^T}{\sqrt{(p_3 - p_2)^T(p_3 - p_2)}} J_3 \\ \frac{(p_3 - p_2)^T}{\sqrt{(p_3 - p_2)^T(p_3 - p_2)}} J_3 & \\ -\frac{(p_1 - p_3)^T}{\sqrt{(p_1 - p_3)^T(p_1 - p_3)}} J_3 & \end{bmatrix} \in \mathfrak{R}^{3 \times 63},$$

$$\delta t = \begin{bmatrix} \delta t_1 \\ \delta t_2 \\ \delta t_3 \end{bmatrix} \in \mathfrak{R}^{3 \times 1}.$$

This linear calibration model is based on the assumption that all of the actual leg-end distances, i.e., a_{12}^a, a_{23}^a , and a_{31}^a , are known. Such an assumption, however, is not always true. In many cases, we are unable to know the actual leg-end distances without external measuring equipment. A modified calibration model is therefore developed, in which the actual leg-end distances are also considered as the kinematic parameters to be identified. Denote the original theoretical values of the leg-end distances as a_{12}^0, a_{23}^0 , and a_{31}^0 , respectively. The actual leg-end distances, which are in the neighbourhood of their original theoretical values, can be given by

$$\begin{aligned} a_{12}^a &= a_{12}^0 + \delta a_{12}^*, \\ a_{23}^a &= a_{23}^0 + \delta a_{23}^*, \\ a_{31}^a &= a_{31}^0 + \delta a_{31}^*, \end{aligned} \tag{26}$$

$\delta a_{12}^*, \delta a_{23}^*$, and δa_{31}^* , in Eqn. (26) represent the differential change of the leg-end distances with respect to their original theoretical values. They are different from $\delta a_{12}, \delta a_{23}$, and δa_{31} which represent the differential change of the leg-end distances with respect to their computed nominal values. Substituting Eqn. (26) into Eqn. (25), we can obtain a modified linear calibration model such that

$$y = \mathcal{A}x \tag{27}$$

where

$$y = \begin{bmatrix} a_{12}^0 - \sqrt{(p_2 - p_1)^T(p_2 - p_1)} \\ a_{23}^0 - \sqrt{(p_3 - p_2)^T(p_3 - p_2)} \\ a_{31}^0 - \sqrt{(p_1 - p_3)^T(p_1 - p_3)} \end{bmatrix} \in \mathfrak{R}^{3 \times 1},$$

$$\mathcal{A} = [J \quad -I_{3 \times 3}] \in \mathfrak{R}^{3 \times 66},$$

$$x = (\delta t_1, \delta t_2, \delta t_3, \delta a_{12}^*, \delta a_{23}^*, \delta a_{31}^*)^T \in \mathfrak{R}^{66 \times 1}.$$

In the modified calibration model, we have altogether 66 error parameters to be identified, which reflect the kinematic errors of a three-legged modular parallel robot.

4.4 An iterative least-squares algorithm

Based on the calibration model Eqn. (27), 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{A}_i . After m measurements, we can stick y_i and \mathcal{A}_i to form the following equation:

$$\tilde{Y} = \tilde{\mathcal{A}}x, \tag{28}$$

where

$$\tilde{Y} = [y_1 \ y_2 \ \dots \ y_m]^T \in \mathfrak{R}^{3m \times 1},$$

$$\tilde{\mathcal{A}} = \text{column}[\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_m] \in \mathfrak{R}^{3m \times 66}.$$

Since the model Eqn. (28) 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

$$x = (\tilde{\mathcal{A}}^T \tilde{\mathcal{A}})^{-1} \tilde{\mathcal{A}}^T \tilde{y}, \tag{29}$$

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

The solution of Eqn. (29) can be further improved through iterative substitution as shown in Fig. 8. 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 actual leg-distances are updated by substituting x into the following equations:

$$\begin{aligned} T_{i(j-1),ij}(0)^{new} &= e^{\delta t_{ij}} T_{i(j-1),ij}(0)^{old}, \\ p_i'^{new} &= p_i'^{old} + \delta p_i', \\ a_{12}^{0\ new} &= a_{12}^{0\ old} + \delta a_{12}^*, \\ a_{23}^{0\ new} &= a_{23}^{0\ old} + \delta a_{23}^*, \\ a_{31}^{0\ new} &= a_{31}^{0\ old} + \delta a_{31}^*. \end{aligned} \tag{30}$$

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}^0 , a_{23}^0 , and a_{31}^0 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 .

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 leg-end distance error, as

$$\delta a = \sqrt{\frac{1}{3m} \sum_{i=1}^m ((a_{12}^c - a_{12}^{(i)})^2 + (a_{23}^c - a_{23}^{(i)})^2 + (a_{31}^c - a_{31}^{(i)})^2)}. \tag{31}$$

In Eqn. (31), $a_{12}^{(i)}$, $a_{23}^{(i)}$, and $a_{31}^{(i)}$ are the three leg-end distances computed according to the calibration results.

5. SIMULATION RESULTS

In this section, a simulation example of calibrating a three-legged (6-DOF, RRRS) parallel robot in Fig. 5(b) is given to demonstrate the effectiveness of the calibration algorithm. As shown in the kinematic diagram in Fig. 7, 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, $i = 1, 2,$ and $3,$ and P''_i represents nominal leg-end positions with respect to the mobile platform frame. Hence, we can compute the original theoretical values of the leg-end distances as:

$$a_{12}^0 = \sqrt{(p''_2 - (p''_1)^T)(p''_2 - p''_1)} = 410\sqrt{3},$$

$$a_{23}^0 = \sqrt{(p''_3 - (p''_2)^T)(p''_3 - p''_2)} = 410\sqrt{3},$$

$$a_{31}^0 = \sqrt{(p''_1 - (p''_3)^T)(p''_1 - p''_3)} = 410\sqrt{3},$$

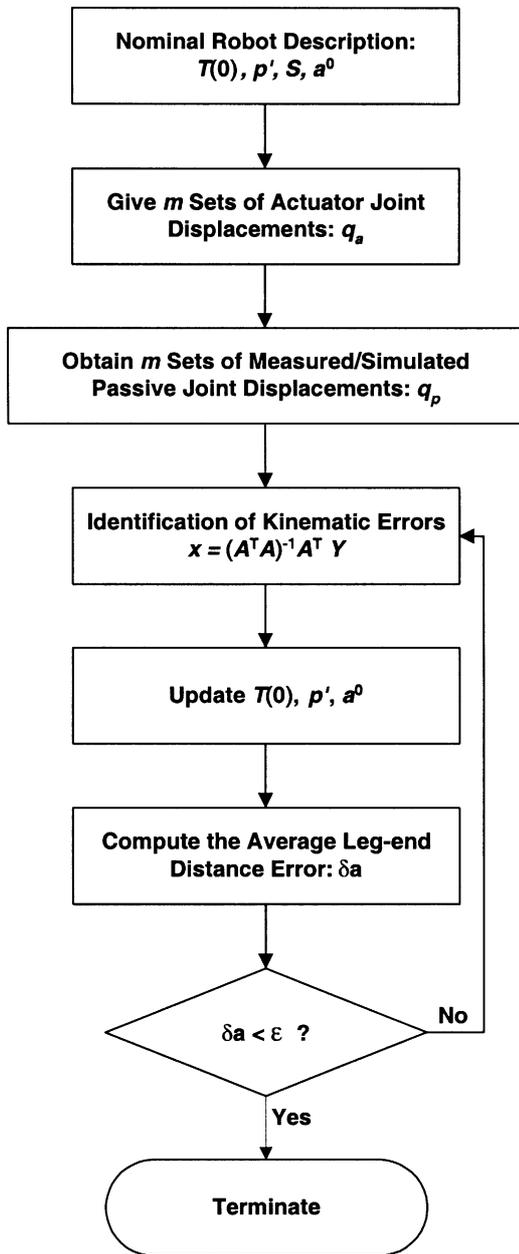


Fig. 8. Iterative calibration loop.

Table I. 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 II. Identified Kinematic Errors.

Dyad	Kinematic errors	$T_{i(j-1), ij}^c(0)$
0-1 (Leg1)	$(-7.698, 7.941, -0.160, 0.01763, 0.02284, -0.00001)^T$	$\begin{bmatrix} -0.02284 & -0.86569 & 0.50005 & -210.664 \\ 0.01763 & 0.49976 & 0.86599 & -348.699 \\ -0.99958 & 0.02859 & 0.00384 & 88.383 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
1-2 (Leg1)	$(1.36772, 8.42726, 3.04695, 0.00027, 0.01880, 0.02956)^T$	$\begin{bmatrix} -0.01880 & 0.02956 & -0.99939 & -328.526 \\ -0.00001 & -0.99956 & -0.02956 & -1.30877 \\ -0.99982 & -0.00054 & 0.01879 & 9.237 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
2-3 (Leg1)	$(2.022, -0.097, -1.296, 0.05999, 0.00135, -0.00058)^T$	$\begin{bmatrix} -0.00133 & 0.99999 & -0.00062 & 2.021 \\ 0.05995 & -0.99820 & -0.99820 & -0.059 \\ -0.99820 & -0.00136 & -0.05995 & -1.300 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
0-1 (Leg2)	$(-2.570, -10.644, -0.171, -0.02861, -0.00384, -0.00013)^T$	$\begin{bmatrix} 0.00384 & -0.00018 & -0.99999 & 407.081 \\ -0.02860 & -0.99959 & 0.00007 & -8.100 \\ -0.99958 & 0.02860 & -0.00384 & 91.514 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
1-2 (Leg2)	$(1.368, 8.427, 2.581, 0.00027, 0.01880, 0.02955)^T$	$\begin{bmatrix} -0.01879 & 0.02954 & -0.99939 & -328.530 \\ -0.00001 & -0.99956 & -0.02955 & -1.304 \\ -0.99982 & -0.00054 & 0.01879 & 8.771 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
2-3 (Leg2)	$(2.022, -0.096, -1.301, 0.05999, 0.00127, -0.00058)^T$	$\begin{bmatrix} -0.00125 & 0.99999 & -0.00062 & 2.021 \\ 0.05995 & -0.00054 & -0.99820 & -0.057 \\ -0.99820 & -0.00124 & -0.05995 & -1.304 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
0-1 (Leg3)	$(10.271, 2.697, -0.086, 0.01097, -0.01899, -0.00007)^T$	$\begin{bmatrix} 0.01899 & 0.86585 & 0.49994 & -196.414 \\ 0.01097 & 0.49982 & -0.86606 & 356.794 \\ -0.99976 & 0.02194 & -0.00001 & 90.006 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
1-2 (Leg3)	$(1.377, 8.436, 2.776, 0.01033, 0.01891, 0.02278)^T$	$\begin{bmatrix} -0.01903 & 0.02268 & -0.99956 & -328.548 \\ 0.01011 & -0.99969 & -0.02288 & 0.888 \\ -0.99977 & -0.01055 & 0.01879 & 9.008 \\ 0. & 0. & 0. & 1. \end{bmatrix}$
2-3 (Leg3)	$(2.022, -0.080, -1.316, 0.05999, 0.00106, -0.01057)^T$	$\begin{bmatrix} -0.00075 & 0.99994 & -0.01060 & 2.021 \\ 0.05996 & -0.01053 & -0.99815 & -0.052 \\ -0.99820 & -0.00138 & -0.05995 & -1.31884 \\ 0. & 0. & 0. & 1. \end{bmatrix}$

Table II. Continued.

Position vector	Kinematic errors	P_i^c
p'_1	$(-4.000, 4.598, 0.057)^T$	$(-4.000, -325.402, 0.057)^T$
p'_2	$(-4.025, 4.598, 0.058)^T$	$(-4.025, -325.402, 0.058)^T$
p'_3	$(-3.994, 4.598, 0.084)^T$	$(-3.994, -325.402, 0.084)^T$
Leg-end distance	Kinematic errors	$(a_{12}^c, a_{23}^c, a_{31}^c)^T$
(1-2, 2-3, 3-1)	$(-3.599, -3.599, -3.599)^T$	$(706.542, 706.542, 706.542)^T$

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. Generate 50 robot poses as well as the corresponding 50 sets of nominal passive joint angles by using the numerical forward kinematic algorithm;
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 I);
3. Find the actual initial poses in each dyad:
 $T_{i(j-1),i}^a(0) = T_{i(j-1),i}(0)e^{dt_{ij}}$, the actual position vectors of each leg end with respect to frame i : $p'^a_i = p'_i + dp'_i$ and the actual leg-end distances: $a^a_{12} = a^0_{12} + da_{12}$, $a^a_{23} = a^0_{23} + da_{23}$, $a^a_{31} = a^0_{31} + da_{31}$;
4. Determine the simulated actual passive joint displacements for the 50 poses using the numerical forward kinematics algorithm.
5. Identify the kinematic errors by using the iterative calibration algorithm.

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 be satisfied such that $\|w_i + dw_i\| = 1$ 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 of the 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 II. 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. 9, where the average leg-end distance error (combined for the 50 poses) is reduced from about 15mm to nearly 0 within 3 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.

6. CONCLUSION

In this paper, a linear local POE model is proposed for the kinematic calibration of a class of three-legged modular parallel robot based on the leg-end distance errors. 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 refining the 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 recovery of the kinematic errors by at least two orders. Future work will be focused on experimental study of the proposed self-calibration algorithm.

Acknowledgement

The authors thank Prof. Frank Chongwoo Park of Seoul National University, Korea, for his valuable advice on the formulation of the calibration model. This work is supported by Gintic Institute of Manufacturing Technology, Singapore, under Upstream project U-97-A006 and Ministry of

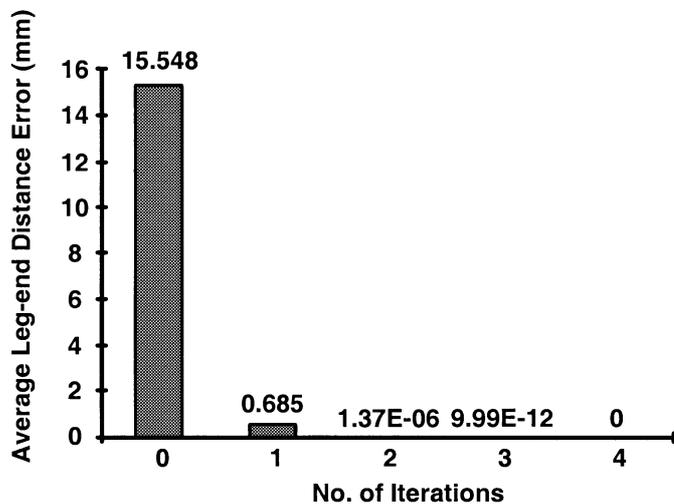


Fig. 9. Leg-end distance errors before and after calibration.

Education, Singapore, under academic research fund RG 29/99.

References

1. S.J. Ryu, Jinwood Kim, F.C. Park and Jongwon Kim, "Design and performance analysis of a parallel mechanism-based universal machining center", *Proceedings of 3rd International Conference on Advanced Mechatronics*, Okayama, Japan (1998) pp. 359–363.
2. F. Badano et al., "Evaluation of exploration strategies in robotic assembly," *Proceedings of 4th IFAC Symposium on Robot Control* (1994) pp. 63–68.
3. I.M. Chen, "Theory and Applications of Modular Reconfigurable Robotic Systems," *PhD Thesis* (California Institute of Technology, CA, USA, 1994).
4. I.M. Chen and G. Yang, "Configuration independent kinematics for modular robots," *Proceedings of IEEE Conference on Robotics and Automation* Minneapolis, Minnesota (April, 1996) pp. 1440–1445.
5. R. Cohen, M.G. Lipton, M.Q. Dai and B. Benhabib, "Conceptual design of a modular robot," *ASME Journal of Mechanical Design* **114**, 117–125 (March, 1992).
6. D. Schmitz, P. Khosla and T. Kanada, "The CMU reconfigurable modular manipulator system," *Technical Report CMU-RI-TR-88-7* (Carnegie-Mellon University, 1988).
7. H. Zhuang, "Self-calibration of parallel mechanisms with a case study on Stewart platform," *IEEE Transactions on Robotics and Automation* **13**(3), 387–397 (1997).
8. H. Zhuang and L. Liu, "Self calibration of a class of parallel manipulators," *Proceedings of IEEE Conference on Robotics and Automation*, Minneapolis, Minnesota (April, 1996) pp. 994–999.
9. A. Nahvi, J.M. Hollreback and V. Hayward, "Calibration of a parallel robot using multiple kinematic closed loops," *Proceedings of IEEE Conference on Robotics and Automation* (1994) pp. 407–412.
10. D.J. Bennett and J.M. Hollerbach, "Self-calibration of single-loop, closed kinematic chains formed by dual or redundant manipulators," *Proceedings of IEEE Conference on Decision and Control*, Athens, Greece Dec, 1988) pp. 627–631.
11. D.J. Bennett and J.M. Hollerbach, "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 (1991).
12. C.W. Wampler, J.M. Hollerbach and T. Arai, "An implicit loop method for kinematic calibration and its application to closed-chain mechanisms," *IEEE Transactions on Robotics and Automation* **11**(5), 710–724 (1995).
13. C.C. Iurascu and F.C. Park, "Geometric algorithms for closed chain kinematic calibration," *Proceedings of IEEE Conference on Robotics and Automation*, Detroit, USA (May, 1999) pp. 1752–1757.
14. L. Notash and R.P. Podhorodeski, "Kinematic calibration of parallel manipulators," *IEEE Transaction on Systems, Man, and Cybernetics* 3310–3315 (1995).
15. F.C. Park and K. Okamura, "Kinematic calibration and the product of exponential formula," *Advances in Robot Kinematics and Computational Geometry* (MIT Press, Cambridge, 1994) pp. 119–128.
16. I.M. Chen and G. Yang, "Kinematic calibration of modular reconfigurable robots using product-of-exponentials formula," *J. Robotic Systems* **14**(11), 807–821 (1997).
17. G. Yang and I.M. Chen, "A novel kinematic calibration algorithm for reconfigurable robotic systems," *Proceedings of IEEE Conference on Robotics and Automation*, Albuquerque, New Mexico (April, 1997) pp. 3197–3202.
18. R.P. Podhorodeski and K.H. Pittens, "A class of parallel manipulators based on kinematically simple branches," *ASME Journal of Mechanical Design* **116**, 908–914 (1994).
19. G. Yang, I.M. Chen, W.K. Lim and S.H. Yeo, "Design and kinematic analysis of modular reconfigurable parallel robots," *Proceedings of IEEE Conference on Robotics and Automation* (May, 1999) pp. 2501–2506.
20. F.C. Park, "Computational aspect of manipulators via product of exponential formula for robot kinematics," *IEEE Transactions on Automatic Control* **39**(9), 643–647 (1994).
21. R. Murray, Z. Li and S.S. Sastry, *A Mathematical Introduction to Robotic Manipulation* (CRC Press, 1994).
22. I.M. Chen and G. Yang and I.G. Kang, "Numerical inverse kinematics for modular reconfigurable robots," *J. Robotic Systems* **16**, 213–225 (1999).
23. G. Yang, "Kinematics, dynamics, calibration, and configuration optimization of modular reconfigurable robots," *PhD Thesis* (Nanyang Technological University, Singapore, 1999).
24. R. Brockett, "Robotic manipulators and the product of exponential formula," *International Symposium in Math. Theory of Network and Systems*, Beer Sheva, Israel (1983) pp. 120–129.