ISSN 1392 - 1207. MECHANIKA. 2011. 17(5): 529-534

Kinematic parameter identification of parallel robots for semi-physical simulation platform of space docking mechanism

Dayong Yu*, Weifang Chen, Hongren Li*****
**School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China, E-mail: wy_ydy@yahoo.com.cn*
***School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China, E-mail: chenwfnudt@163.com*
****School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China, E-mail: lihr@hit.edu.cn*

1. Introduction

Parallel robots such as a Stewart platform have some advantage of high rigidity, high accuracy, and high load-carrying capacity over serial robots. However, they have some drawbacks of relatively small workspace and very difficult forward kinematics problems. These robots have found a variety of applications in flight simulators, high-precision machining centers, medical surgery [1, 2], and so on.

In general, accuracy is defined by repeatability and bias. Lack of repeatability is due to random error and quantified by the variance of a number of measurements. Bias is systematic error and determined by the mean value. While it is difficult to compensate for the random error, compensation for the systematic error could be done effectively by means of calibration. It is well known that excellent positioning performance of the manipulator may be achieved based on an accurate kinematic equation. However, parameters of the equation inevitably deviate from their nominal values due to manufacturing and assembly errors. A direct consequence is to reduce the accuracy of the robots, since their control strategy heavily relies on a precise description of the kinematic equation. One way to tackle this problem is to improve the theoretical kinematic equation through kinematic calibration which consists of identifying a more accurate geometrical relationship between the joint sensor/encoder reading and the actual pose of the end-effector. Literatures indicated that the most economical and feasible way of enhancing the manipulator accuracy is through kinematic calibration [3-5]. Kinematics calibration involves the following procedures: set up an appropriate kinematics model; take measurements of the robot pose; identify the actual kinematics parameters to minimize the errors between the poses predicted by the model and the actual measured ones; implement the identified robot kinematics model.

Let us employ the paradigm of literature [6] in stating a unified calibration formulation. First, the principle is to link the unknown kinematic parameters P and the information on the state of the manipulator M, either provided by sensors or through constraints applied on joints or brought by an additional mechanism. Some closed loop equations $f(P, M) = 0$ can be determined; the equations vanish within the measurement error. The simplest way to determine M is by using the internal sensors of the manipulator. Usually, though, as they do not provide redundant information, their number is minimal for controlling the manipulator's degrees of freedom. It is possible to install additional captors on passive joints for self-calibration (with the benefit of simplifying the forward kinematics). In practice, it is not easy to add redundant sensors or constrain. Hence, most calibration methods use external measurements devices to obtain the required information, such as laser trackers, theodolites, cameras, inclinometers or mechanical devices. Many authors use the kinematics to relate the kinematic parameters P to the available information M. Then, the basic calibration methods with external measurements use either the forward kinematics or the inverse kinematics. Those calibration methods may be prone to error. The reason is the difficulty to obtain a closed form for the solutions of the kinematics problem.

2. Kinematic model

2.1. Inverse kinematics

This section describes the parallel robot and its kinematics model. The robot consists of two rigid bodies, the base and the mobile platform, connected by 6 legs. The leg linear actuator provides 6 degree of freedom for the platform pose relative to the base, corresponding to position P and rotation matrix R. A pose $X = [P, R]$ is associated to 6 length variations l_i measured by internal leg sensors, $I = 1, \ldots, 6$.

Each leg is attached to the base by a hook joint and to the platform by a hook joint; so 23 parameters are required to model each leg. But as shown in [6], the principal source of error in positioning is due to limited knowledge of the joint centers and to the fact that part of the length is not given by the sensors. We thus use a simpler model with attachment point's a_i in the mobile frame, b_i in the reference frame, and offset lengths $l_{0,i}$. This gives 7 parameters per leg, therefore 42 overall, denoted by ρ.

The inverse kinematics problem of the parallel robot deals with calculating the leg lengths when the pose is given and the kinematics parameters are known. In effect, it is a mapping from global pose to local leg transducer readings. The inverse kinematics of a parallel robot is simple, yielding a nonlinear closed form solution.

The vector chain in Fig. 1 can be expressed as

$$l_i = Ra_i + P - b_i \tag{1}$$

The length of leg i can then be determined by taking the magnitude of Eq. (1)

$$\lambda_i = \|l_i\| = \|Ra_i + P - b_i\| \tag{2}$$

and the leg length sensor reading can be obtained by

$$s_i = \lambda_i - l_{0,i} \tag{3}$$

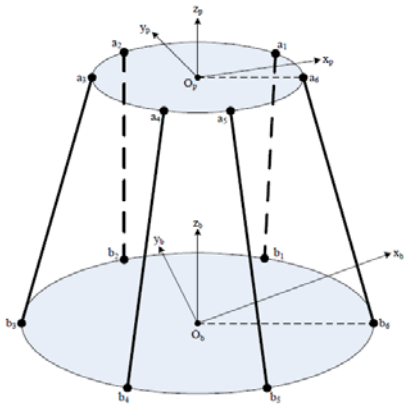

Fig. 1 Schematic representation of the parallel robot

2.2. Forward kinematics

For the parallel robots, the forward kinematics is difficult to compute since it consists in solving Eq. (1) for P and R given l_i and ρ. Define the vector function to describe the difference between the estimated sensor reading (s_i) and the actual sensor reading ($\hat{s}_i$).

$$f = \begin{bmatrix} f_1 \\ \vdots \\ f_6 \end{bmatrix} = \begin{bmatrix} s_1^2 - \hat{s}_1^2 \\ \vdots \\ s_6^2 - \hat{s}_6^2 \end{bmatrix} \tag{4}$$

The Newton-Raphson algorithm can be stated as:

1. measure $\hat{s}$ and select an initial guess for the pose X;
2. compute s based on X_0;
3. form f;
4. if $X^T X <$ tolerance$_1$, exit with X as the solution;
5. compute the partial derivative matrix $J = \partial f / \partial X$ such that $J_{i,j} = \partial f_i / \partial X_j$;
6. solve for the update δX from $J\delta X = -f$;
7. if $\delta X^T \delta X <$ tolerance$_2$, exit with X as the solution;
8. update X by $X = X + \delta X$ and go to step 2.

In step one, an initial pose vector X must be guessed. This is usually taken as the last pose of the mobile platform. In step two, the estimated length can be computed with the inverse kinematics. Step three and four are straightforward, with f formed through Eq. (4) and tolerance being the allowed error in the pose calculation. The partial derivatives required in step five can be computed. Step six involves a 6 by 6 matrix inversion to calculate δX, and then in step seven, the norm of δX is tested to see if the update is significant. If the update is considered significant, then the algorithm repeats from step two with the update pose vector.

3. Error model

3.1. Pose errors description of a joint-link chain

Due to manufacturing tolerances and assembly errors, all hook joints are imperfect-their axes neither intersect nor are perpendicular to one another. As such, joint centers in actuality do not exist and the axes of the actuators are skewed to joint axes. An error model that accommodates these error sources is needed in order to develop a calibration method that greatly enhances the parallel robot accuracy performance.

A joint-link chain is defined as a set of consecutive structure elements starting from the center of the base and going to the center of the mobile platform through one of the links. Kinematically, a joint-link chain can be modeled by a set of consecutive transformations from the coordinate frame O-XYZ to the coordinate frame o-xyz, as illustrated in Fig. 2. Seven transformations are used to model each joint-link chain in order to express the pose of the end-effector with respect to the base coordinate frame

$$T_p^b = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 T_7^6 \tag{5}$$

The end-effector pose errors of each chain can be expressed by applying the general relations used in precision modeling of open chains

$$\Delta p_i = J_i \left[\delta X_{Bi} \; \delta Y_{Bi} \; \delta Z_{Bi} \; \delta\gamma_{xi} \; \delta\gamma_{yi} \; \delta L_{0i} \; \delta l_i \; \delta\varphi_{xi} \times \right.$$
$$\left. \times \delta\varphi_{yi} \; \delta\varphi_{zi} \; \delta x_{Ai} \; \delta y_{Ai} \; \delta z_{Ai} \right]^T = J_i \delta\upsilon_i \tag{6}$$

$$J_i = \left[J_{XBi} \; J_{YBi} \; J_{ZBi} \; J_{\gamma xi} \; J_{\gamma yi} \; J_{L0i} \; J_{li} \; J_{\varphi xi} \; J_{\varphi yi} \times \right.$$
$$\left. \times J_{\varphi zi} \; J_{xAi} \; J_{yAi} \; J_{zAi} \right]^T \tag{7}$$

where each column of the Jacobian J_i describes the influence of one error source on the end-effector errors.

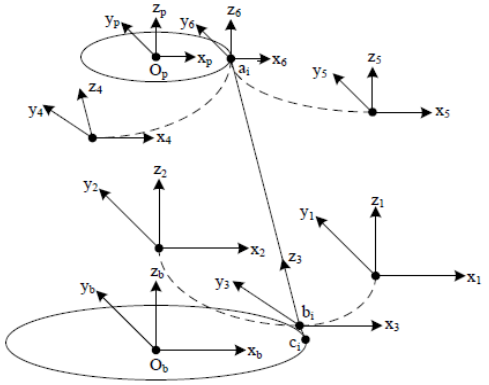

Fig. 2 The coordinate frame of the ith joint-link chain

3.2. Displacement error model of the joint

The joint displacements in the passive joints A_i and B_i are dependent displacements; as a result, the displacements errors in these joints are also dependent errors. Knowing that the end-effector errors are identical for all 6 joint-link chains, all dependent errors can be expressed starting from the following equations

$$J_1\delta\upsilon_1 = J_2\delta\upsilon_2 = J_3\delta\upsilon_3 = J_4\delta\upsilon_4 = J_5\delta\upsilon_5 = J_6\delta\upsilon_6 \tag{8}$$

From Eq. (8) 5 independent matrix equations result; separating the dependent term from the independent ones, we obtain

$$\left[J_{\gamma x1} \; J_{\gamma y1} \; J_{\varphi x1} \; J_{\varphi y1} \; J_{\varphi z1} \right] \delta v_1^{dep} - \left[J_{\gamma xk} \; J_{\gamma yk} \; J_{\varphi xk} \; J_{\varphi yk} \; J_{\varphi zk} \right] \times$$
$$\times \delta\upsilon_k^{dep} = \left[J_{XBk} \; J_{YBk} \; J_{ZBk} \; J_{L0k} \; J_{lk} \; J_{xAk} \; J_{yAk} \; J_{zAk} \right] \delta\upsilon_k^{\text{int}} -$$
$$- \left[J_{XB1} \; J_{YB1} \; J_{ZB1} \; J_{L01} \; J_{l1} \; J_{xA1} \; J_{yA1} \; J_{zA1} \right] \delta\upsilon_1^{\text{int}} \quad k = 2,\ldots,6 \tag{9}$$

where $\delta\upsilon_i^{dep} = \left[\delta\gamma_{xi}\ \delta\gamma_{yi}\ \delta\varphi_{xi}\ \delta\varphi_{yi}\ \delta\varphi_{zi}\right]^T$, $\delta\upsilon_i^{int} = \left[\delta X_{Bi}\right.$ $\left.\delta Y_{Bi}\ \delta Z_{Bi}\ \delta L_{0i}\ \delta l_i\ \delta x_{Ai}\ \delta y_{Ai}\ \delta z_{Ai}\right]^T$.

Next, the 5 equations (4) are assembled in one matrix equation

$$J^{dep}\delta\upsilon^{dep} = J^{ind}\delta\upsilon^{ind} \tag{10}$$

where $\delta\upsilon^{dep} = \left[\delta\gamma_{xi}\ \delta\gamma_{yi}\ \delta\phi_{xi}\ \delta\phi_{yi}\ \delta\phi_{zi}\cdots\delta\gamma_{xi}\ \delta\gamma_{yi}\ \times\right.$ $\left.\times\delta\phi_{xi}\ \delta\phi_{yi}\ \delta\phi_{zi}\right]^T$ is the global vector of dependent errors, and

$$J^{dep} = \begin{bmatrix} J_1^{dep} & -J_2^{dep} & O & O & O & O \\ J_1^{dep} & O & -J_3^{dep} & O & O & O \\ J_1^{dep} & O & O & -J_4^{dep} & O & O \\ J_1^{dep} & O & O & O & -J_5^{dep} & O \\ J_1^{dep} & O & O & O & O & -J_6^{dep} \end{bmatrix},$$

$$J^{ind} = \begin{bmatrix} -J_1^{ind} & J_2^{ind} & O & O & O & O \\ -J_1^{ind} & O & J_3^{ind} & O & O & O \\ -J_1^{ind} & O & O & J_4^{ind} & O & O \\ -J_1^{ind} & O & O & O & J_5^{ind} & O \\ -J_1^{ind} & O & O & O & O & J_6^{ind} \end{bmatrix},$$

$J_i^{dep} = \left[J_{\gamma xi}\ J_{\gamma yi}\ J_{\varphi xi}\ J_{\varphi yi}\ J_{\varphi zi}\right]$, $J_i^{ind} = \left[J_{XBi}\ J_{YBi}\ J_{ZBi}\right.$, $\left.J_{L0i}\ J_{li}\ J_{xAi}\ J_{yAi}\ J_{zAi}\right]$, $i = 1,\ldots,6$.

Finally, the dependent errors can be calculates by the following equation

$$\delta\upsilon^{dep} = \left(J^{dep}\right)^{-1} J^{ind}\delta\upsilon^{ind} = J^{*}\delta\upsilon^{ind}. \tag{11}$$

3.3. Pose errors determination of the parallel robot

The matrix J^* is a 30×48 size matrix, which can be subdivided into 6 sub matrices of 5 rows; each sub matrix represent the error Jacobian of dependent errors from one joint-link chain related to the deviation $\delta\upsilon^{ind}$. Considering one of the open chains, e.g. chain 1, the end-effector pose errors are expressed as

$$\begin{aligned} \Delta p &= J_1\delta\upsilon_1 = J_1^{ind}\delta\upsilon_1^{ind} + J_1^{dep}\delta\upsilon_1^{dep} = \\ &= \begin{bmatrix} J_1^{ind} & J_1^{dep} \end{bmatrix}\begin{bmatrix} \delta\upsilon_1^{ind} \\ \delta\upsilon_1^{dep} \end{bmatrix} = J_1 J_{G1}\delta\upsilon^{ind} \end{aligned} \tag{12}$$

where J_{G1} is a special matrix, which relates the vector $\delta\upsilon_1$ to the vector $\delta\upsilon^{ind}$.

The influence of kinematic paramer deviations on the end-effector pose errors is described by a linear model, where the error Jacobian J_G is the matrix of the following equation

$$\Delta p = J_G\delta\upsilon^{ind} \tag{13}$$

Finally, considering Eqs. (12) and (13), the error Jacobian J_G can be obtained though the following relations

$$J_G = J_1 J_{G1} = J_2 J_{G2} = J_3 J_{G3} = J_4 J_{G4} = J_5 J_{G5} = J_6 J_{G6}. \tag{14}$$

4. Kinematic calibration

The error model given in Eq. (13) can be used for calibration to estimate the kinematic parameter errors based on pose measurement. It can also be used for tolerance analysis to examine the effect on the pose accuracy.

There are eight independent parameters in each joint-link chain as shown in Eq. (9) for a 6-DOF parallel robot, so in total there are 48 parameters to be calibrated. As mentioned earlier in this paper, there parameters are independent, and must be determined individually. For calibration of 48 parameters, it requires at least eight pose measurements. A pose measurement collects six data, three position, and three orientations. Eight or more measurements will collect 48 or more data, so Eq. (13) will become determinate or redundant. With adequate measurement at hand, these parameters can be estimated by the least squares algorithm as

$$\delta\hat{\upsilon}^{ind} = \left(\overline{J}_G^T\overline{J}_G\right)^{-1}\overline{J}_G^T\Delta\overline{p} \tag{15}$$

where the hat ($\wedge$) sign indicates the estimated value

$$\overline{J}_G = \left[\tilde{J}_{G1}^T\ \tilde{J}_{G2}^T\ \tilde{J}_{G3}^T\ \tilde{J}_{G4}^T\ \tilde{J}_{G5}^T\ \tilde{J}_{G6}^T\ \tilde{J}_{G7}^T\ \tilde{J}_{G8}^T\right]^T \tag{16}$$

is the augmented vector of the error Jacobian matrix, J_G, for all eight poses and

$$\Delta\overline{p} = \left[\ \Delta p_1^T\ \Delta p_2^T\ \Delta p_3^T\ \Delta p_4^T\ \Delta p_5^T\ \Delta p_6^T\ \Delta p_7^T\ \Delta p_8^T\ \right] \tag{17}$$

is the augmented vector of the pose error vector Δp for all eight pose measurement.

The final solution for the kinematic parameter errors $\hat{\upsilon}^{ind}$ is iteratively calculated until the error converges. In each iteration, the geometry is updated to reflect the previously calculated kinematic parameter errors and an updated augmented error Jacobian matrix $\overline{J}_G$ is generated to reflect the new geometry. This new matrix is then applied to the measured pose error, according to Eq. (15). This iterative process continues until the kinematic error converges.

The new calibration geometry is then updated according to the following equation:

$$\upsilon^{ind} = \upsilon^{ind} + \delta\hat{\upsilon}^{ind} \tag{18}$$

To ensure optimal calibration, it is proposed to select these poses from the least sensitive area within the workspace. As discussed by Nahvi and Hollerbach [7], several indices have been suggested to quantify the goodness of pose selection. The noise amplification index was shown to have the greatest sensitivity to calibration error, and is thus chosen over indexes. The noise amplification index is defined by taking the singular value decomposition of the error Jacobian given in Eq. (14)

$$J_G = U\Sigma V^T \tag{19}$$

where U and V are orthogonal matrices, and Σ is a matrix with the singular values of J_G along its main diago-

nal. Let the singular values be given by σ_i and ordered from largest to smallest so that $\sigma_1 \geq \sigma_p$. Define the noise amplification index as:

$$O = \sigma_p^2 / \sigma_1 \tag{20}$$

The larger this index, the better the calibration accuracy should be. The noise amplification index should only be used as a guide for pose selection, since a pose set with a larger index does not necessarily guarantee a better calibration. Instead, it only guarantees a better worst-case calibration. The preference for this index over other proposed indexes is justified by Nahvi and Hollerbach [7].

5. Simulations and experiments

5.1. Simulations

Calibrations based on the parallel robot geometry were simulated with various parameter deviations, noise levels, and pose sets. The nominal parameters and the workspace limits of the parallel robot are listed in [8]. Three parameter sets were simulated, with the parameter deviations obtained from normal distributions with variances of 0.01 mm (set I), 0.1 mm (set II), and 1.0 mm (set III). Gaussian noise with variances of 0.0001, 0.001, 0.01, and 0.1 mm was added to the measurements to simulate measurement noise. Pose set 1 contains 64 poses, based on a full factorial exploration of the six pose variable limits. This pose set has a noise amplification index of 7.68×10^{-6}. Pose set 2 has a noise amplification index of 6.71×10^{-5} and contains 100 poses selected from the workspace using a coordinate exchange algorithm. Pose set 3 has a noise amplification index of 3.26×10^{-6} and contains 100 random poses. Pose set 4 contains 200 poses selected by a coordinate exchange algorithm, and has a noise amplification index of 9.57×10^{-5}.

Tables 1-3 give the simulated calibration results. Note that the first row in the tables corresponds to initial conditions of the parameter set. The estimation error is calculated as the 2-norm of the difference between the actual deviations and the estimated deviations. To determine the resulting error improvement, the pose error was simulated before and after calibration.

Table 1
Calibration simulations with parameter set I

Pose Set	Noise	Estimation error		Position error		Orientation error	
		rms	reduct	rms	reduct	rms	reduct
	-	0.0853	-	0.3060	-	0.0177	-
1	0.0001	0.0162	81%	0.0398	87%	0.0018	90%
1	0.001	0.0384	55%	0.1040	66%	0.0041	77%
1	0.01	0.0691	19%	0.2356	23%	0.0101	43%
1	0.1	0.0836	2%	0.2693	12%	0.0150	15%
2	0.0001	0.0111	87%	0.0245	92%	0.0012	93%
2	0.001	0.0350	59%	0.1469	52%	0.0051	71%
2	0.01	0.0640	25%	0.2417	21%	0.0085	52%
2	0.1	0.0742	13%	0.2785	9%	0.0135	24%
3	0.0001	0.0247	71%	0.0765	75%	0.0021	88%
3	0.001	0.0580	32%	0.1928	37%	0.0050	72%
3	0.01	0.0776	9%	0.2723	11%	0.0106	40%
3	0.1	0.0845	0.9%	0.3121	-2%	0.0165	7%
4	0.0001	0.0085	90%	0.0214	93%	0.0009	95%
4	0.001	0.0316	63%	0.1377	55%	0.0055	69%
4	0.01	0.0589	31%	0.2540	17%	0.0094	47%
4	0.1	0.0708	17%	0.2876	6%	0.0122	31%

Table 2
Calibration simulations with parameter set II

Pose Set	Noise	Estimation error		Position error		Orientation error	
		rms	reduct	rms	reduct	rms	reduct
	-	0.3137	-	1.1730	-	0.0746	-
1	0.0001	0.0063	98%	0.0352	97%	0.0015	98%
1	0.001	0.0847	73%	0.2229	81%	0.0149	80%
1	0.01	0.1914	39%	0.7390	37%	0.0612	18%
1	0.1	0.2761	12%	0.9853	16%	0.0880	-18%
2	0.0001	0.0094	97%	0.0235	98%	0.0007	99%
2	0.001	0.0471	85%	0.1408	88%	0.0127	83%
2	0.01	0.1663	47%	0.4105	65%	0.0395	47%
2	0.1	0.2478	21%	0.8328	29%	0.0664	11%
3	0.0001	0.0282	91%	0.0704	94%	0.0060	92%
3	0.001	0.1349	57%	0.4809	59%	0.0157	79%
3	0.01	0.2259	28%	0.8680	26%	0.0380	49%
3	0.1	0.2917	7%	1.0205	13%	0.0656	12%
4	0.0001	0.0157	95%	0.1525	87%	0.0045	94%
4	0.001	0.0596	81%	0.0821	93%	0.0119	84%
4	0.01	0.1725	45%	0.5161	56%	0.0545	27%
4	0.1	0.2416	23%	0.7859	33%	0.0634	15%

The reduction of the parameter estimation error by itself is not the goal of the calibration. Ultimately, the resulting errors of the parallel robot should be reduced. Fig. 3 shows that by estimating the model parameters well, the overall accuracy of the parallel robot can be improved. This verifies that better parameter estimates will result in improved robot accuracy. The plot shows good correlation between the parameter estimation error reduction percentage and the pose error reduction percentage.

Table 3
Calibration simulations with parameter set III

Pose Set	Noise	Estimation error		Position error		Orientation error	
		rms	reduct	rms	reduct	rms	reduct
	-	1.0019	-	1.6196	-	0.1246	-
1	0.0001	0.0040	99.6%	0.0146	99.1%	0.0009	99.3%
1	0.001	0.0601	94%	0.0486	97%	0.0025	98%
1	0.01	0.2405	76%	0.2915	82%	0.0199	84%
1	0.1	0.4108	59%	0.9880	39%	0.0511	59%
2	0.0001	0.0030	99.7%	0.0049	99.7%	0.0006	99.5%
2	0.001	0.0501	95%	0.0324	98%	0.0012	99%
2	0.01	0.2104	79%	0.2429	85%	0.0137	89%
2	0.1	0.3707	63%	0.7612	53%	0.0374	70%
3	0.0001	0.0080	99.2%	0.0211	98.7%	0.0012	99%
3	0.001	0.1102	89%	0.0810	95%	0.0112	91%
3	0.01	0.3306	67%	0.4859	70%	0.0436	65%
3	0.1	0.5210	48%	1.1823	27%	0.0586	53%
4	0.0001	0.0020	99.8%	0.0032	99.8%	0.0004	99.7%
4	0.001	0.0301	97%	0.0162	99%	0.0050	96%
4	0.01	0.1603	84%	0.1782	89%	0.0237	81%
4	0.1	0.2805	72%	0.8746	46%	0.0523	58%

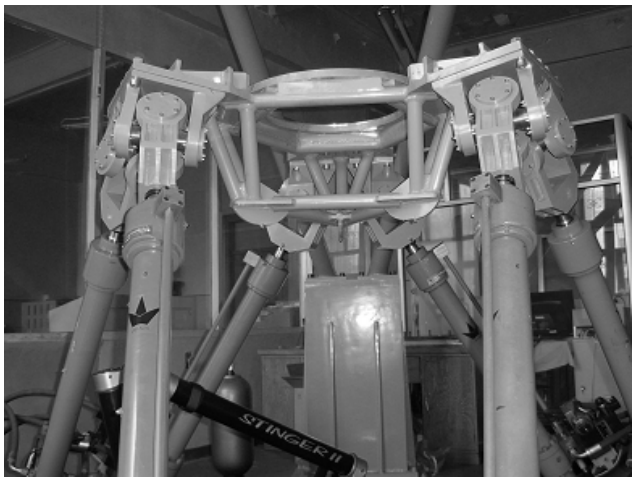

Fig. 3 Experimental setup

5.2. Experiments

The important elements of the experimental setup depicted in Fig. 3 are the coordinate measuring machine, tooling balls and the parallel robots. After the coordinate measuring machine has been calibrated to measure the parallel robot pose with respect to the reference frame, the parallel robot is commanded to 120 different well-spaced poses within the robot workspace, which have been determined to cover the range of motion of all the legs. Note that at least 8 measurements are needed to estimate 48 parameters. The greater number of measurement would contribute to the convergence of the algorithm and reduce the effect of measurement noise. A good initial guess helps a least square estimation algorithm to converge quickly without experiencing any numerical singularities. Therefore, the nominal values of the parameters are taken as the initial values for the parameters while implementing nonlinear least square algorithm.

The estimated technique mentioned above has been implemented using a program prepared in MatLab toolbox. The developed program can perform the calibration procedure considering any combination of the parameters used in the kinematic model. The RMS pose errors with the nominal parameters and with 48 and 42 estimated parameters determined separately from the estimated technique are provided in Tables 4 and 5, respectively. The pose errors of the parallel robot with the nominal parameters, 42, 48 estimated parameters are depicted in the first, second and the third columns of Fig. 4, respectively.

By calibration based on 42-parameter model an accuracy improvement of a factor 6.6 for the parallel robot could be gained on the summation of RMS of position whereas by calibration based on 48-parameter model the predication of the position of the parallel robot improved by a factor of 7.8 for the summation of RMS.

Table 4

The RMS pose error with 48 parameters

	RMS_p	$\sum RMS_p$	RMS_o	$\sum RMS_o$
Normal parameters	0.4444 1.5349 2.2120	4.1913	0.1061 0.2254 0.2750	0.6065
Estimated parameters	0.1718 0.2226 0.1399	0.5343	0.0159 0.0498 0.0504	0.1161
Improvement %	61.34 85.50 93.68	87.25	85.01 77.90 81.67	80.86

Table 5

The RMS pose error with 42 parameters

	RMS_p	$\sum RMS_p$	RMS_o	$\sum RMS_o$
Normal parameters	0.4444 1.5349 2.2120	4.1913	0.1061 0.2254 0.2750	0.6065
Estimated parameters	0.2023 0.2658 0.1625	0.6306	0.0211 0.0585 0.0643	0.1439
Improvement %	54.48 82.68 92.65	84.95	80.11 74.05 76.62	76.27

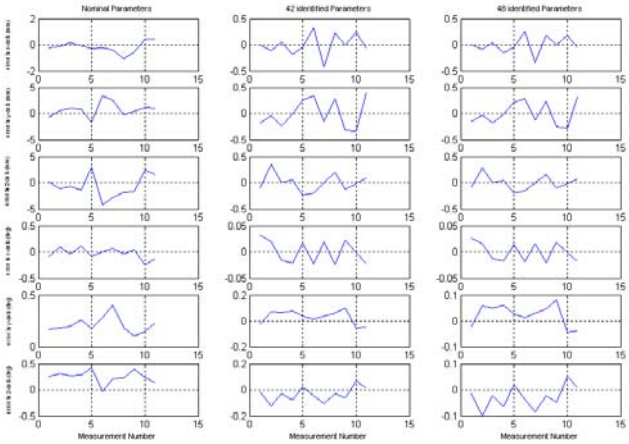

Fig. 4 The pose errors with the nominal, 42 and 48 identified parameters

6. Conclusions

This paper presents a kinematic parameter identification algorithm and some calibration results based on a coordinate measurement technique and a more complete kinematic model including sensor errors for parallel robot. In the iteration algorithm, it is important to select initial values. It has tightly relationship with the convergence and local minima. The identification results based on inverse kinematic calibration model are set as the initial values of nonlinear least squares algorithm. It not only solves the problem mentioned above, but also decreases the iteration steps greatly for the initial values to be closer to the real values. For the ill-condition matrix problem caused by noise, the Levenberg-Marquardt algorithm is adopted. In addition selection of calibration pose sets is briefly discussed, as well as the influence of measurement noise on calibration accuracy. Finally, based on above analysis, kinematic calibration experiments are carried out. The position error RMS of a parallel robot is reduced from 4.19 to 0.53 mm and the orientation error RMS is also reduced from 0.61 to 0.12°.

References

1. **Rat, N.R.**; **Neagoe, M.**; **Diaconescu, D.**; **Stan, S.D**. 2011. Dynamic simulations regarding the influence of the load-rigidity correlation on the working accuracy of a medical Triglide parallel robot, Mechanika 17(2): 178-181.
2. **Stan, S.D.**; **Balan, R.**; **Maties, V.**; **Rad, C**. 2009. Kinematics and fuzzy control of ISOGLIDE3 medical parallel robot, Mechanika 1(751): 62-66.
3. **Abtahi, M.**; **Pendar, H.**; **Alasty, A.**; **Vossough,i G.H**. 2009. Calibration of parallel kinematic machine tools using mobility constraint on the tool center point, International Journal of Advanced Manufacturing Technology 45(5-6): 531-539.
4. **Pashkevich, A.**; **Chablat, D.**; **Wenger, P**. 2009. Kinematic calibration of Orthoglide-type mechanisms from observation of parallel leg motions, Mechatronics 19(4): 478-488.
5. **Wang, L.P.**; **Xie, F.G.**; **Liu, X.J.**; **Wang, L.S**. 2011. Kinematic calibration of the 3-DOF parallel module of a 5-axis hybrid milling machine, Robotica 29(4): 535-546.

6. **Daney, D**. 2003. Kinematic calibration of the Gough platform, Robotica 21(6): 677-690.
7. **Nahvi, A.**; **Hollerbach, J.M**. 1996. Noise amplification index for optimal selection in robot calibration, Proceedings of the IEEE International Conference on Robotics and Automation 1: 647-654.
8. **Yu, D.Y**.; **Sun, X.W.**; **Wang, Y**. 2007. Kinematic calibration of parallel robots based on total least squares algorithm, Proceedings of the IEEE International Conference on Mechatronics and Automation, 1: 789-794.

Dayong Yu, Weifang Chen, Hongren Li

LYGIAGREČIOJO ROBOTO SKIRTO PUSIAU FIZINIO PRISIJUNGIMO MECHANIZMO KOSMINĖJE ERDVĖJE IMITAVIMUI KINEMATINIŲ PARAMETRŲ NUSTATYMAS

R e z i u m ė

Pusiau fizinė imitacinė platforma naudota siekiant nustatyti lygiagrečių robotų kinematinių parametrų paklaidas, reikalingas norint padidinti padėties tikslumą. Šiame straipsnyje taikomas metodinis kinematinių parametrų nustatymo būdas. Tam reikalinga visų kojų ir padėčių matavimo mašina gauta informacija. Kinematiniams parametrams nustatyti yra pritaikytas nelinijinis mažųjų kvadratų algoritmas. Kojų jutiklių matavimo paklaidos nustatomos kinematiniu modeliavimu ir kinematinių parametrų identifikavimu. Stewarto lygiagrečiojo roboto, sukurto *Electro-hydraulic Servo Simulation and Test System* centre Harbino technologijos institute, imitavimo ir eksperimentiniai tyrimai parodė, kad jo parametrų identifikavimo algoritmas ir kalibravimo metodai tinka lygiagretiesiems robotams.

Dayong Yu, Weifang Chen, Hongren Li

KINEMATIC PARAMETER IDENTIFICATION OF PARALLEL ROBOTS FOR SEMI-PHYSICAL SIMULATION PLATFORM OF SPACE DOCKING MECHANISM

S u m m a r y

Because of errors in the kinematic parameters of parallel robots, it is necessary to identify them to improve the pose accuracy for accurate task performance of semi-physical simulation platform. In this paper, a methodical way of kinematic parameter identification is introduced. It requires measurements of all legs and the pose information provided by a coordinate measuring machine. Nonlinear least squares algorithm is employed to determine the kinematic parameters. The measurement errors in the leg sensors are considered during kinematic modeling and kinematic parameter identification. Simulations and experimental studies on a Stewart parallel robot built in the Institute of Electro-hydraulic Servo Simulation & Test System of Harbin Institute of Technology reveal the convenience and effectiveness of the proposed robot parameter identification algorithm and calibration method for parallel robots.

Received March 25, 2011
Accepted September21, 2011