

A Parameter Dimension Reduction-Based Estimation Approach to Enhance the Kinematic Accuracy of a Parallel Hardware-in-the-Loop Docking Simulator

Yan Hu[†], Feng Gao^{†*}, Xianchao Zhao[†], Tianhao Yang[‡], Haoran Shen[‡], Chenkun Qi[†] and Rui Cao[†]

[†]State Key Lab of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, PR China. E-mails: huyanwhu@163.com, fengg@sjtu.edu.cn, xczhao@sjtu.edu.cn, chenkqi@sjtu.edu.cn, caorui@sjtu.edu.cn

[‡]Shanghai Aerospace Equipment Manufacturer Co., Ltd., Shanghai 200245, PR China
E-mails: yangtianhao149@126.com, haoran26@126.com

(Accepted August 12, 2020. First published online: October 20, 2020)

SUMMARY

The hardware-in-the-loop docking simulators are significant ground test equipment for aerospace projects. The fidelity of docking simulation highly depends on the accuracy performance. This paper investigates the kinematic accuracy for the developed docking simulator. A novel kinematic calibration method which can reduce the number of parameters for error modeling is presented. The principle of parameters separation is studied. A simplified error model is derived based on Taylor series. This method can contribute to the simplification of the error model, fewer measurements, and easier convergence during the parameters identification. The calibration experiment validates this method for further accuracy enhancement.

KEYWORDS: Kinematics; Accuracy; Parallel robot; Hardware-in-the-loop; Docking simulator.

1. Introduction

The hardware-in-the-loop (HIL) docking simulators are significant ground test equipment for aerospace projects. On the earth, they are used for simulating the contact process of two spacecrafts, such as the capturing or docking process. The advantages of the HIL simulation over the full numerical simulation¹ and the full physical simulation² are summarized in our previous work.³ The fidelity of the HIL simulation highly depends on the accuracy performance of the docking simulator. This accuracy performance can be divided into the dynamics accuracy and the kinematic accuracy. The dynamics accuracy means the accuracy of the real-time simulation control of the dynamic process. It determines the accuracy of the calculated motion instructions of the docking simulator. The main factors which influence the dynamics accuracy are the time delay and the divergence in the control system. Our previous work was focused on the improvement of dynamics accuracy through divergence compensation.⁴ However, the kinematic accuracy has not been well perfected. As a supplement, this paper investigates the kinematic accuracy based on the developed HIL docking simulator. The kinematic inaccuracy is mainly caused by the large size of the docking simulator. This docking simulator is made up by a novel six degree-of-freedom (6-DOF) perpendicular parallel

* Corresponding author. E-mail: fengg@sjtu.edu.cn

mechanism. The mechanism is about 3.5 m in height, covering an area of nearly 6.5 m in diameter. The huge size brings much difficulty in the manufacturing and assembly process of the parallel robot and has a great influence on the robot accuracy. For the engineering applications of parallel robots, the kinematic accuracy is fundamental and essential, especially for the docking simulators.⁵

It is widely recognized that the kinematic inaccuracy comes from the propagation of geometric errors and non-geometric errors.^{6,7} The non-geometric errors, which are induced by compliance errors, thermal distortions, clearance, and friction, are proved to just account for nearly 10% of the entire positioning error by the early work from Renders and Becquet et al.^{8–10} The geometric errors, which are mainly caused by the unavoidable manufacturing imperfections or assembling errors, play a dominant role in robot inaccuracy.¹¹ So, in this research, we focus on geometric errors. Since the geometric errors influence the pose accuracy of the robot end effector, kinematic calibration is a common and effective approach to enhance the robot accuracy.¹² Kinematic calibration is the process to identify and modify the exact values of the geometric parameters in robot kinematics by the inputs and outputs of the kinematics.¹³ As it is well known, there are three main types of kinematic calibration methods: external calibration methods, constrained calibration methods, and auto/self-calibration methods.^{13,14} By searching the literature, we can find a flourishing development of the robot calibration in the past decades, and a lot of creative approaches are proposed for the robot calibration.^{15–23} In the field of kinematic calibration for docking simulator, Yu et al. did a lot of work for the 6-DOF Stewart-based docking simulator.^{24–28} They used a coordinate measuring machine that made constraint conditions or adopted radial-based function neural network for pose accuracy compensation.²⁹ In our preliminary work, the kinematic accuracy of the docking simulator can only meet the basic accuracy index which requires the positioning errors to be within 1 mm and 0.1°. A further improvement of the kinematic accuracy is crucial to the accuracy performance of the HIL simulation.

After a review of the existing methods, we can classify them into “full-parameter-identification” method, with all the kinematic parameters identified in one model. According to our experience, this type of method may not have a good identification result when facing large geometric errors and measurement noise. In this condition, we have to seek an alternative solution. In this paper, a new calibration method based on parameters separation is presented. Unlike the regular “full-parameter-identification” methods, we sequentially identify the subsets of all the geometric parameters. With this method, some parameters can be sorted out and determined firstly and individually through simple measuring process, and then a simplified error model containing a minimum set of unknowns will be obtained for the remaining parameters. Since the number of parameters is cut down in the error model, the influence of measurement noise, human uncertainty as well as the coupling effect between parameters can decrease. During the parameters identification, the simplified error model will need fewer measurements and will contribute to easier convergence.

In addition, unlike all the existing 6-DOF docking simulators which use Stewart platform,^{30,31} our docking simulator is based on a 3-3-PUS perpendicular parallel robot. The driving and transmission components are mounted on the ground. This will make the robot to have large workspace and dexterity as well as good dynamic response than the Stewart platform.¹⁴ The goal of this study is to further diminish the positioning error to improve the kinematic accuracy of the docking simulator. The content of this paper is listed as follows. Section 2 presents the kinematic model of the 3-3-PUS parallel robot. Section 3 studies the kinematic parameters of the robot and shows how the parameters separation is conducted. Then, a simplified error model is derived. In Section 4, the calibration experiment and the confirmatory experiment are conducted. Then, the results and concluding remarks are presented in Section 5.

2. Kinematic Description of the 3-3-PUS Parallel Mechanism

As seen in Fig. 1 this parallel mechanism has one moving platform and six uniform kinematic chains. Each chain is composed of a spherical (S) joint, a linkage, a universal (U) joint, and a prismatic (P) joint in sequence. So each chain is the PUS kinematic chain. The six prismatic joints are linear driving joints, which are installed on the fixed base. The moving platform is the end effector. According to the Grübler–Kutzbach’s criterion for mobility calculation,³² the DOF of this mechanism can be calculated as follows:

$$M = 6 \times (14 - 18 - 1) + (6 \times 1 + 6 \times 2 + 6 \times 3) = 6$$

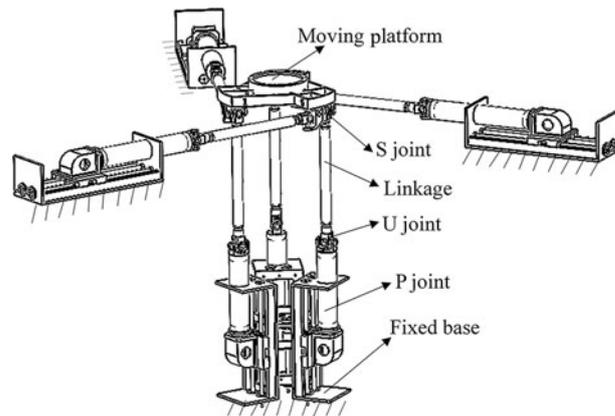


Fig. 1. The 3-3-PUS parallel mechanism.

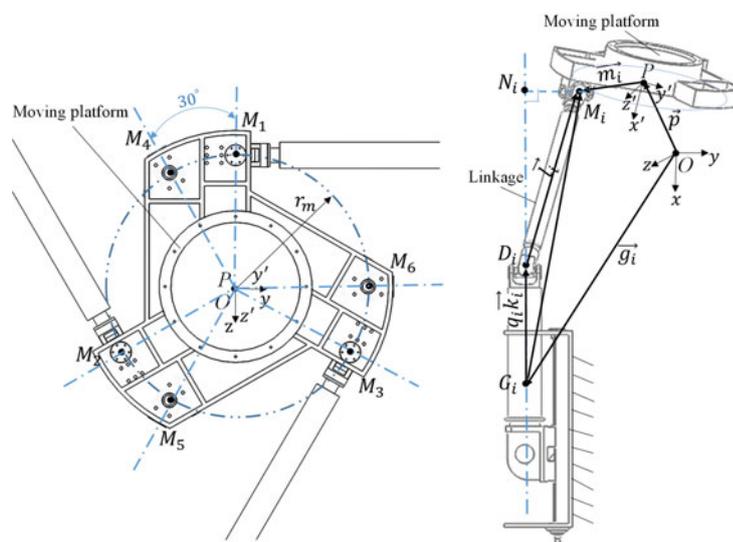


Fig. 2. Kinematics diagram.

So, the DOF of this parallel mechanism is 6. This 6-DOF parallel mechanism has three horizontally arranged prismatic joints and three vertically arranged prismatic joints. The prismatic joints are all guide slider mechanisms. The three horizontal prismatic joints are arranged to be coplanar. The three vertical prismatic joints are arranged to be parallel. The direction of each universal joint is arbitrary.

In the previous design of this robot, the optimum configurations for the six kinematic chains are determined as follows: the distribution of all the spherical joints is on a circle with radius r_m theoretically. The center of this circle is point P . Three of the S joints (denoted by $M_1 M_2 M_3$) are distributed symmetrically on this circle as well as the other three S joints (denoted by $M_4 M_5 M_6$). And each central angle of the adjacent S joints (denoted by $\angle M_1 P M_4, \angle M_2 P M_5, \angle M_3 P M_6$) is 30° , as shown in Fig. 2. Theoretically, in the initial position of the robot, for each kinematic chain, the linkage is collinear with the P joint and is tangent to the circle mentioned above. This configuration can realize singularity-free in the demanded workspace. The dimensions and workspace of this robot will be introduced later.

For parallel robots, the inverse kinematics is easier to obtain than forward kinematics, so we will take advantages of the inverse kinematics to deduce the calibration model. The next is a concise inverse kinematics description. The moving coordinate frame $P\{x', y', z'\}$ and the fixed coordinate frame $O\{x, y, z\}$ are shown in Fig. 2. The origin of the moving frame is point P and the origin of the fixed frame is point O . In the initial position of the robot, point P theoretically coincides with point O . The illustration for the inverse kinematics is based on any one of the kinematic chains of the robot. We use the subscript i to denote the i th kinematic chain. Here, i is from 1 to 6.

As shown in Fig. 2, point M_i is the theoretical center of the i th S joint. Point D_i is the theoretical rotation center of the i th U joint. The initial position of point D_i is marked by point G_i . In the fixed frame $O\{x, y, z\}$, the position vector of the point G_i is represented by \mathbf{g}_i . Point N_i is the projection point of M_i onto the direction vector of the i th P joint. In the fixed frame $O\{x, y, z\}$, the vector $\overrightarrow{PM_i}$ is represented by \mathbf{m}_i , while it is denoted by \mathbf{m}_i' in the moving frame $P\{x', y', z'\}$. The vector $\overrightarrow{D_iM_i}$ is denoted by \mathbf{l}_i with respect to the fixed frame $O\{x, y, z\}$. The norm of \mathbf{l}_i represents the length of the i th linkage, which is denoted by L_i . The unit vector specifying the direction of the i th P joint is represented by \mathbf{k}_i relative to the fixed frame $O\{x, y, z\}$. The actuating displacement of the i th P joint is denoted by q_i . The position vector of point P is represent by \mathbf{p} , which describes the position of the moving platform relative to the fixed frame $O\{x, y, z\}$.

The pose of the moving platform can be described by a six-dimensional vector $\mathbf{X} = [p_x, p_y, p_z, \alpha, \beta, \gamma]^T$. The first three variables p_x, p_y, p_z of the pose make up \mathbf{p} . The last three variables α, β, γ of the pose are the orientation description of the moving platform. α, β, γ are the roll-pitch-yaw angles, and they determine the rotation matrix of the end effector relative to the fixed frame $O\{x, y, z\}$. The rotation matrix is denoted by \mathbf{R} . Easily, we can build the equation of q_i from the Fig. 3:

$$q_i = \left| \overrightarrow{G_iN_i} \right| - \left| \overrightarrow{D_iN_i} \right| \tag{1}$$

Furthermore, we can write

$$\left| \overrightarrow{G_iN_i} \right| = \left| \overrightarrow{G_iM_i} \cdot \mathbf{k}_i \right| \tag{2}$$

$$\left| \overrightarrow{D_iN_i} \right| = \sqrt{\left| \overrightarrow{D_iM_i} \right|^2 - \left| \overrightarrow{N_iM_i} \right|^2} = \sqrt{L_i^2 - \left| \overrightarrow{G_iM_i} \times \mathbf{k}_i \right|^2} \tag{3}$$

According to the geometric relations of the vectors, we can write the following equation:

$$\overrightarrow{G_iM_i} = \mathbf{p} + \mathbf{m}_i - \mathbf{g}_i = \mathbf{p} + \mathbf{R}\mathbf{m}_i' - \mathbf{g}_i, \quad (i = 1, 2, \dots, 6) \tag{4}$$

Then combined with Eq. (1), we can solve q_i as follows:

$$q_i = \left| \overrightarrow{G_iM_i} \cdot \mathbf{k}_i \right| - \sqrt{L_i^2 - \left| \overrightarrow{G_iM_i} \times \mathbf{k}_i \right|^2}, \quad (i = 1, 2, \dots, 6) \tag{5}$$

This is the inverse kinematic solution of the parallel robot.

3. Kinematic Calibration Method

According to the kinematics description introduced in Section 2, we can find that the key parameters of kinematics are $\mathbf{m}_i', \mathbf{g}_i, \mathbf{k}_i$, and L_i . These parameters are determined by the geometry of the robot at the design time. They directly influence the accuracy of the kinematics of the robot. However, the manufacturing imperfections and assembling errors will bring about inevitable geometric errors to these parameters. The geometric errors can be denoted by $\Delta\mathbf{m}_i', \Delta\mathbf{g}_i, \Delta\mathbf{k}_i$, and ΔL_i , respectively.

There are the following equations:

$$\begin{cases} \Delta\mathbf{m}_i' = (\Delta m_{xi}', \Delta m_{yi}', \Delta m_{zi}')^T \\ \Delta\mathbf{g}_i = (\Delta g_{xi}, \Delta g_{yi}, \Delta g_{zi})^T \\ \Delta\mathbf{k}_i = (\Delta k_{xi}, \Delta k_{yi}, \Delta k_{zi})^T \\ \Delta L_i \end{cases}, \quad (i = 1, 2, \dots, 6) \tag{6}$$

where Δ is the symbol of error operator. For each kinematic chain, the geometric errors have 10 unknowns. Since the robot has six kinematic chains, there are total 60 unknowns. To identify these unknowns is our calibration task.

The relation between the geometric errors and the robot positioning accuracy can be deduced as follows. Firstly, several symbols are defined. The vector $\boldsymbol{\theta}$ represents the angle coordinates of the

moving platform. Unit vector \mathbf{n}_i represents the direction of the i th linkage. So, there are the following equations:

$$\begin{aligned} \boldsymbol{\theta} &= [\alpha, \beta, \gamma]^T \\ \mathbf{l}_i &= L_i \mathbf{n}_i, \quad (i = 1, 2, \dots, 6) \end{aligned} \tag{7}$$

From the kinematics diagram in Fig. 2, we can get

$$\mathbf{p} + \mathbf{R}\mathbf{m}_i' = \mathbf{g}_i + q_i \mathbf{k}_i + \mathbf{l}_i = \mathbf{g}_i + q_i \mathbf{k}_i + L_i \mathbf{n}_i, \quad (i = 1, 2, \dots, 6) \tag{8}$$

Taking the derivative of both sides of Eq. (8), there is

$$\delta \mathbf{p} + \delta \mathbf{R}\mathbf{m}_i' + \mathbf{R}\delta \mathbf{m}_i' = \delta \mathbf{g}_i + q_i \delta \mathbf{k}_i + \delta L_i \mathbf{n}_i + L_i \delta \mathbf{n}_i, \quad (i = 1, 2, \dots, 6) \tag{9}$$

where δ is the symbol of differential operator.

According to the mathematical definition of the rotation matrix \mathbf{R} , there is the following equation:

$$\delta \mathbf{R} = \begin{bmatrix} 0 & -\delta\gamma & \delta\beta \\ \delta\gamma & 0 & -\delta\alpha \\ -\delta\beta & \delta\alpha & 0 \end{bmatrix} \mathbf{R} = [\delta\boldsymbol{\theta} \times] \mathbf{R} \tag{10}$$

where $[\delta\boldsymbol{\theta} \times]$ represents the skew-symmetric matrix in this equation. And $\delta\boldsymbol{\theta}$ is the differential of the vector $\boldsymbol{\theta}$. Here, $[\delta\boldsymbol{\theta} \times]$ maps the vector $\delta\boldsymbol{\theta}$ to its skew-symmetric matrix. It should be noted that in the linear algebra, the cross-product of two vectors is equivalent to the skew-symmetric matrix of the first vector multiplying the second vector. So, we can take advantage of this property to relate the skew-symmetric matrix with the cross-product of vectors. Then, substituting Eq. (10) into Eq. (9), we can get

$$\delta \mathbf{p} + \delta\boldsymbol{\theta} \times (\mathbf{R}\mathbf{m}_i') = -\mathbf{R}\delta \mathbf{m}_i' + \delta \mathbf{g}_i + q_i \delta \mathbf{k}_i + \delta L_i \mathbf{n}_i + L_i \delta \mathbf{n}_i, \quad (i = 1, 2, \dots, 6) \tag{11}$$

Since \mathbf{n}_i is a unit vector, there is the relation $\mathbf{n}_i \cdot \mathbf{n}_i = 1$. Taking the dot product by \mathbf{n}_i for both sides of Eq. (11), we can obtain

$$\mathbf{n}_i^T \delta \mathbf{p} + (\mathbf{R}\mathbf{m}_i' \times \mathbf{n}_i)^T \delta \boldsymbol{\theta} = -\mathbf{n}_i^T \mathbf{R} \delta \mathbf{m}_i' + \mathbf{n}_i^T \delta \mathbf{g}_i + q_i \mathbf{n}_i^T \delta \mathbf{k}_i + \delta L_i + L_i \mathbf{n}_i^T \delta \mathbf{n}_i, \quad (i = 1, 2, \dots, 6) \tag{12}$$

During the simplification for Eq. (12), we can take advantage of the relation $\mathbf{n}_i \cdot \delta \mathbf{n}_i = 0$. Because that $\delta \mathbf{n}_i$ represents angular velocity of the linkage and that the linkage cannot rotate about its length direction due to the PUS structure of the kinematic chain, the angular velocity is right perpendicular to the length direction of the linkage. So, the last term $L_i \mathbf{n}_i^T \delta \mathbf{n}_i$ on the right-hand side of Eq. (12) equals zero and can be eliminated.

Rewriting Eq. (12) in matrix form, we can get

$$\begin{bmatrix} \mathbf{n}_1^T & (\mathbf{R}\mathbf{m}_1' \times \mathbf{n}_1)^T \\ \vdots & \vdots \\ \mathbf{n}_6^T & (\mathbf{R}\mathbf{m}_6' \times \mathbf{n}_6)^T \end{bmatrix}_{6 \times 6} \begin{bmatrix} \delta \mathbf{p} \\ \delta \boldsymbol{\theta} \end{bmatrix} = \begin{bmatrix} -\mathbf{n}_1^T \mathbf{R} \mathbf{n}_1^T & q_1 \mathbf{n}_1^T & 1 & \dots & 0 \\ 0 & \dots & -\mathbf{n}_2^T \mathbf{R} \mathbf{n}_2^T & q_2 \mathbf{n}_2^T & 1 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & -\mathbf{n}_6^T \mathbf{R} & \mathbf{n}_6^T & q_6 \mathbf{n}_6^T & 1 \end{bmatrix}_{6 \times 60} \begin{bmatrix} \delta \mathbf{m}_1' \\ \delta \mathbf{g}_1 \\ \delta \mathbf{k}_1 \\ \delta L_1 \\ \vdots \\ \delta \mathbf{m}_6' \\ \delta \mathbf{g}_6 \\ \delta \mathbf{k}_6 \\ \delta L_6 \end{bmatrix}_{60 \times 1} \tag{13}$$

Supposing that the geometric errors are small, the differential operator δ could be regarded as Δ . Then, the relationship between the geometric errors and the end effector pose errors is deduced as:

$$\Delta X = E_1^{-1} E_2 \begin{bmatrix} \Delta m_1' \\ \Delta g_1 \\ \Delta k_1 \\ \Delta L_1 \\ \vdots \\ \Delta m_6' \\ \Delta g_6 \\ \Delta k_6 \\ \Delta L_6 \end{bmatrix}_{60 \times 1} \tag{14}$$

where $E_1 = \begin{bmatrix} n_1^T & (Rm_1' \times n_1)^T \\ \vdots & \vdots \\ n_6^T & (Rm_6' \times n_6)^T \end{bmatrix}_{6 \times 6}$, $E_2 = \begin{bmatrix} -n_1^T R n_1^T & q_1 n_1^T & 1 & \cdots & 0 \\ 0 & \cdots & -n_2^T R n_2^T & 1 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & \cdots & \cdots \\ 0 & \cdots & -n_6^T R n_6^T & q_6 n_6^T & 1 & \cdots \end{bmatrix}_{6 \times 60}$ and their

elements can be computed through the kinematics solution.

To approximately quantify the influence of the geometric errors on the robot pose accuracy, much numerical calculation is conducted. The possible ranges of the geometric errors are experientially estimated according to the manufacturing and assembling results. By traversing the whole workspace, the maximum pose errors can be preliminary calculated according to the error ranges. Table I shows the details of the geometric parameters and their effects on the robot positioning accuracy. Since k_i is a non-dimensional vector, its unit is 1. The values listed in the last column include the effect on position and orientation coordinates.

Table I. The details of geometric parameters.

Geometric parameters	Geometric errors	Source of errors	Possible ranges	Effect on accuracy
m_i'	$\Delta m_i'$	Manufacture and assembly	± 3 mm	(± 3.5 mm, 0.008°)
g_i	Δg_i	Assembly	± 6 mm	(± 6.5 mm, 0.015°)
k_i	Δk_i	Arrangement of the guide rail	± 0.005	(± 0.6 mm, 0.002°)
L_i	ΔL_i	Manufacture	± 3 mm	(± 3.8 mm, 0.010°)

Kinematic calibration treats all the geometric/kinematic parameters as an unknown system and then to identify this system through its inputs and outputs. For parallel robots, this system usually is a strongly coupled system. The identification of the parameters in this system is easily influenced by the measurement noise and human operation errors, especially when there are large numbers of unknowns. Moreover, to this huge parallel robot, the large body brings much difficulty in the manufacturing and installing process. The geometric errors may show big order of magnitude. Meanwhile, the electromagnetic environment of the docking simulator causes considerable measurement noise to the measuring apparatus. All these factors can lead to the inaccuracy of the calibration results. To reduce the influence of these unavoidable factors, we seek solutions by reducing the number of parameters for error modeling. If the number of parameters can be cut down in the error model, the influence of the measurement noise, human uncertainty as well as the coupling effect between parameters can decrease. In addition, during the parameters identification, the simplified error model will need fewer measurements and will contribute to easier convergence. A kernel step to reduce the dimension of the parameters is the parameters separation. Next, we will present the process of parameters separation for this parallel robot.

The previous paragraphs have studied the relevant parameters that influence the robot accuracy, which are listed in Table I. These parameters are independent parameters at the design stage of the robot. Their nominal values are determined before the robot is manufactured. The purpose of

parameters separation is to find certain parameters which could be determined firstly through simple measuring process. We get to analyze the parameters one by one to see if it can be determined firstly. Firstly, we consider m_i' . The six S joints are on the moving platform, and they move relative to the ground as the moving platform moves. So, their coordinates relative to the ground are influenced by the pose of the moving platform. To obtain the coordinates, we have to get the moving platform pose first. Furthermore, point M_i is actually a virtual central point. We have to take indirect ways to obtain the coordinates. This can be achieved by indirect vector computations, but it would bring about solving nonlinear equations with high orders. That will increase inconvenience and complexities for the acquisitions of m_i' . Therefore, this parameter is not suitable to be determined firstly. Then, we consider g_i . Although point G_i is stationary relative to the ground, the acquisition of its coordinates is difficult. The reason lies in the fact that point G_i is also virtual. We have to take indirect ways to obtain its coordinates. Since G_i is the center of the U joint. An easy way to obtain its coordinates is by taking advantage of its definition. This should cooperate with the whole kinematic chains and might need the chain disassembly. Other ways by establishing closed-loop vector equations will need to solve for systems of nonlinear multivariable equations. So, this parameter is also not suitable to be determined firstly. Next, we consider k_i . It is stationary relative to the ground. Since the P joint is the linear driving joint with guide slider mechanism, we can take advantage of this to directly measure several coordinates of a point fixed on the slider by letting the slider move along the driving line. Then through the most fundamental linear fitting, these coordinate will construct a straight line whose unit direction vector is exactly k_i . This is a directly measuring approach to obtain this parameter. This approach is easy to implement and is also convenient. Therefore, this parameter is suitable to be determined firstly. Lastly, we consider L_i . This parameter is the distance between point M_i and point D_i . From the above analysis, we know that the points are virtual and impractical to be directly measured. Moreover, the linkage is not stationary relative to the ground. The indirect ways for the acquisition of this parameter need either chain disassembly or complicated equations. It is hard to find an intuitive way to obtain L_i directly. So, this parameter is also not suitable to be determined firstly.

From the above analysis of the geometric parameters, we can divide the parameters into two groups. Group 1 contains k_i , while Group 2 contains m_i' , g_i , and L_i . Group 1 can be identified firstly through simple measuring process. We do not need to establish extra identification model. Group 2 can be identified by a simplified model with a minimum set of unknowns. Summarily, we can extract the principle of parameters separation as follows:

- a. Independence: the parameter has to be independent parameter at the design stage of the robot.
- b. Immovability: the parameter has to be stationary relative to the ground. The mathematical description is $d({}^0\xi_i)/dt = \mathbf{0}$, where ${}^0\xi_i$ is the description of a parameter relative to the ground.
- c. Measurability: the parameter has to be easy to measure in practice.
- d. Integrity: the measuring process of the parameter has to sustain the integrity of the kinematic chain without chain disassembly.

These four rules are the criterions to judge whether a parameter can be sorted out and determined firstly through simple measuring process.

Although the rules are developed from the proposed docking simulator, the principle of parameters separation is applicable to other type of robots. Theoretically, the robots whose parameters meet the presented separation rules can adopt this method for kinematics calibration. This class of robots usually have rotary joint or prismatic joint directly connected to the rack, such as the Hexaglide, I4, and Orthoglide mechanisms introduced in ref. [20]. To show the universality and to better explain how to use the proposed separation rules, we will take a quick case of the 3-DOF Orthoglide mechanism next. The kinematics description of the mechanism is explained in detail in ref. [33]. This mechanism has three identical chains, so we choose one of them for clarity. Firstly, we must find out the kinematic parameters of the Orthoglide mechanism: the parallelogram length L_i , the orientation of each linear actuated joint n_i , and the position of the linear actuator A_i . Secondly, all of these parameters are the independent design parameters which meet the first rule. Thirdly, since the frame of the parallelogram keeps moving when the mechanism works, the parameter L_i does not meet the second rule. However, n_i and A_i meet this rule, because the position and orientation of the linear actuator have been fixed. Fourthly, since the position of the linear actuator is an inside point which may need chain disassembly for direct measurement, the parameter A_i does not meet the fourth rule. For the parameter n_i , it is easy to obtain if we move the linear actuator and select a point on the

actuator to track its trajectory. The trajectory can fit a line for \mathbf{n}_i . So, only \mathbf{n}_i meets all the four rules and can be separated. This example shows the general steps to use the presented rules to judge if one parameter can be separated. For the calibration of this 6-DOF docking simulator, the result of parameters separation is listed in Table II according to the analyzation in previous paragraphs. This table shows that the total parameters are separated into two groups. Group 1 has 18 unknowns, while Group 2 has 42 unknowns. Group 1 is determined firstly, while Group 2 is identified subsequently.

Table II. The result of parameters separation.

Group	Geometric parameters	Their errors to be determined	Number of unknowns
Group 1	\mathbf{k}_i ($i = 1, 2, \dots, 6$)	$\Delta \mathbf{k}_i = (\Delta k_{xi}, \Delta k_{yi}, \Delta k_{zi})^T$ ($i = 1, 2, \dots, 6$)	$3 \times 6 = 18$
Group 2	$\mathbf{m}'_i, \mathbf{g}_i$, and L_i ($i = 1, 2, \dots, 6$)	$\Delta \mathbf{m}'_i = (\Delta m'_{xi}, \Delta m'_{yi}, \Delta m'_{zi})^T$ $\Delta \mathbf{g}_i = (\Delta g_{xi}, \Delta g_{yi}, \Delta g_{zi})^T$ ΔL_i ($i = 1, 2, \dots, 6$)	$7 \times 6 = 42$

The next is how to identify the parameters of each group. For Group 1, it is a basic mathematical problem for linear fitting. And the corresponding toolbox is embedded in the supporting software of most measuring apparatus. The information of the constructed space line can be displayed as soon as the sample points are measured. That is quite convenient and efficient. For Group 2, the rest parameters which contain a reduced number of unknowns need to be identified by certain error model. Here, we present an error model by taking advantage of the inverse kinematics of the robot.

According to the kinematics diagram in Fig. 2, there is the following relation:

$$L_i^2 = |\mathbf{p} + \mathbf{R}\mathbf{m}'_i - \mathbf{g}_i - q_i\mathbf{k}_i|, \quad (i = 1, 2, \dots, 6) \tag{15}$$

After measuring an actual pose of the robot, the components of \mathbf{p} and \mathbf{R} can be determined and then be written as follows:

$$\begin{cases} \mathbf{p} = [p_x, p_y, p_z]^T \\ \mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}, \quad (i = 1, 2, \dots, 6) \end{cases} \tag{16}$$

where the nine components of the rotation matrix \mathbf{R} are calculated according to the roll-pitch-yaw angles. The coordinates of the vectors $\mathbf{m}'_i, \mathbf{g}_i$ and \mathbf{k}_i can be denoted by:

$$\begin{cases} \mathbf{m}'_i = (m'_{xi}, m'_{yi}, m'_{zi})^T \\ \mathbf{g}_i = (g_{xi}, g_{yi}, g_{zi})^T \\ \mathbf{k}_i = (k_{xi}, k_{yi}, k_{zi})^T \end{cases}, \quad (i = 1, 2, \dots, 6) \tag{17}$$

Substitute Eqs. (16) and (17) into Eq. (15) and then yield

$$L_i^2 = \chi_1^2 + \chi_2^2 + \chi_3^2, \quad (i = 1, 2, \dots, 6) \tag{18}$$

where

$$\begin{cases} \chi_1 = p_x + r_{11}m'_{xi} + r_{12}m'_{yi} + r_{13}m'_{zi} - g_{xi} - q_ik_{xi} \\ \chi_2 = p_y + r_{21}m'_{xi} + r_{22}m'_{yi} + r_{23}m'_{zi} - g_{yi} - q_ik_{yi}, \quad (i = 1, 2, \dots, 6) \\ \chi_3 = p_z + r_{31}m'_{xi} + r_{32}m'_{yi} + r_{33}m'_{zi} - g_{zi} - q_ik_{zi} \end{cases} \tag{19}$$

Let $\Theta_i = [m_{xi}', m_{yi}', m_{zi}', g_{xi}, g_{yi}, g_{zi}, L_i]^T$ be the unknown variable of the i th kinematic chain, and $\Theta_{i0} = [m_{xi0}', m_{yi0}', m_{zi0}', g_{xi0}, g_{yi0}, g_{zi0}, L_{i0}]^T$ be the initial value of the unknown variable. The function of the robot inputs and outputs is set up as follows:

$$f(\Theta_i) = \chi_1^2 + \chi_2^2 + \chi_3^2 - L_i^2 = 0, \quad (i = 1, 2, \dots, 6) \tag{20}$$

Then, we can write the Taylor expansion of $f(\Theta_i)$ at Θ_{i0} by ignoring the second-order higher-order small quantities as follows:

$$\begin{aligned} f(\Theta_i) = & f(\Theta_{i0}) + (m_{xi}' - m_{xi0}') \frac{\partial f(\Theta_{i0})}{\partial m_{xi}'} + (m_{yi}' - m_{yi0}') \frac{\partial f(\Theta_{i0})}{\partial m_{yi}'} + (m_{zi}' - m_{zi0}') \frac{\partial f(\Theta_{i0})}{\partial m_{zi}'} \\ & - (g_{xi} - g_{xi0}) \frac{\partial f(\Theta_{i0})}{\partial g_{xi}} - (g_{yi} - g_{yi0}) \frac{\partial f(\Theta_{i0})}{\partial g_{yi}} - (g_{zi} - g_{zi0}) \frac{\partial f(\Theta_{i0})}{\partial g_{zi}} \\ & - (L_i - L_{i0}) \frac{\partial f(\Theta_{i0})}{\partial L_i} \end{aligned} \tag{21}$$

Then, we can get the following equation:

$$\begin{aligned} f(\Theta_i) = & f(\Theta_{i0}) + 2(\chi_1 r_{11} + \chi_2 r_{21} + \chi_3 r_{31}) (m_{xi}' - m_{xi0}') \\ & + 2(\chi_1 r_{12} + \chi_2 r_{22} + \chi_3 r_{32}) (m_{yi}' - m_{yi0}') \\ & + 2(\chi_1 r_{13} + \chi_2 r_{23} + \chi_3 r_{33}) (m_{zi}' - m_{zi0}') - 2\chi_1 (g_{xi} - g_{xi0}) - 2\chi_2 (g_{yi} - g_{yi0}) \\ & - 2\chi_3 (g_{zi} - g_{zi0}) - 2L_i (L_i - L_{i0}) \\ = & 0 \end{aligned} \tag{22}$$

Divided by 2, Eq. (22) can be written as follows:

$$\begin{aligned} \frac{f(\Theta_{i0})}{2} + & (\chi_1 r_{11} + \chi_2 r_{21} + \chi_3 r_{31}) (m_{xi}' - m_{xi0}') + (\chi_1 r_{12} + \chi_2 r_{22} + \chi_3 r_{32}) (m_{yi}' - m_{yi0}') \\ & + (\chi_1 r_{13} + \chi_2 r_{23} + \chi_3 r_{33}) (m_{zi}' - m_{zi0}') - \chi_1 (g_{xi} - g_{xi0}) - \chi_2 (g_{yi} - g_{yi0}) \\ & - \chi_3 (g_{zi} - g_{zi0}) - L_i (L_i - L_{i0}) = 0 \end{aligned} \tag{23}$$

The purpose of Taylor expansion is to obtain the linearized error model thus to use the linear least squares algorithms for solution. Although more high-order terms of the Taylor expansion can realize better precision of Eq. (21), they also bring about more non-linearization. So, according to our experience, we choose the first-order Taylor expansion which can achieve enough accuracy as well as the linearization. Using the matrix form, the above equation could be rewritten as follows:

$$\eta_i^T \Delta \Theta_i = \frac{f(\Theta_{i0})}{2}, \quad (i = 1, 2, \dots, 6) \tag{24}$$

where $\Delta \Theta_i = \Theta_i - \Theta_{i0}$

$$\eta_i = [-\chi_1 r_{11} - \chi_2 r_{21} - \chi_3 r_{31}, -\chi_1 r_{12} - \chi_2 r_{22} - \chi_3 r_{32}, -\chi_1 r_{13} - \chi_2 r_{23} - \chi_3 r_{33}, \chi_1, \chi_2, \chi_3, L_i]^T.$$

For each kinematic chain, there are seven unknowns to solve, so we need at least seven equations for the solution. For one group of the measured inputs (actuating displacement of the six P joints) and outputs (actual pose of the moving platform), we can establish one equation for each kinematic chain according to Eq. (24). In order to minimize the influence of noise and uncertainty, we will obtain greater number of measured inputs and outputs and adopt iterative least square algorithm. Assuming

that there are n groups of measured inputs and outputs, we will get the following equation for the i th kinematic chain:

$$\begin{bmatrix} \eta_{i,1}^T \\ \eta_{i,2}^T \\ \vdots \\ \eta_{i,n}^T \end{bmatrix}_{n \times 7} \Delta \Theta_i = \begin{bmatrix} \frac{f_1(\Theta_{i0})}{2} \\ \frac{f_2(\Theta_{i0})}{2} \\ \vdots \\ \frac{f_n(\Theta_{i0})}{2} \end{bmatrix}_{n \times 1}, \quad (i = 1, 2, \dots, 6), (n = 1, 2, \dots, 7, \dots, n) \quad (25)$$

Then, we can write the least squares solution of the Eq. (25) as follows:

$$\Delta \Theta_i = (\Phi^T \Phi)^{-1} \Phi^T \begin{bmatrix} \frac{f_1(\Theta_{i0})}{2} \\ \frac{f_2(\Theta_{i0})}{2} \\ \vdots \\ \frac{f_n(\Theta_{i0})}{2} \end{bmatrix}_{n \times 1}, \quad (i = 1, 2, \dots, 6), (n = 1, 2, \dots, 7, \dots, n) \quad (26)$$

where $\Phi = [\eta_{i,1} \ \eta_{i,2} \ \eta_{i,3} \ \dots \ \eta_{i,n}]^T$.

The flowchart of the iterative process is shown in Fig. 3. The nominal values of the kinematic parameters are substituted into Θ_i to start the calculation. When the solution $\Delta \Theta_i$ approaches zero, the calculation ends. After each circulation, the kinematic parameters are updated for the next circulation until the iterative process ends. The iterative processes of each kinematic chain can run simultaneously, which can improve the numerical efficiency.

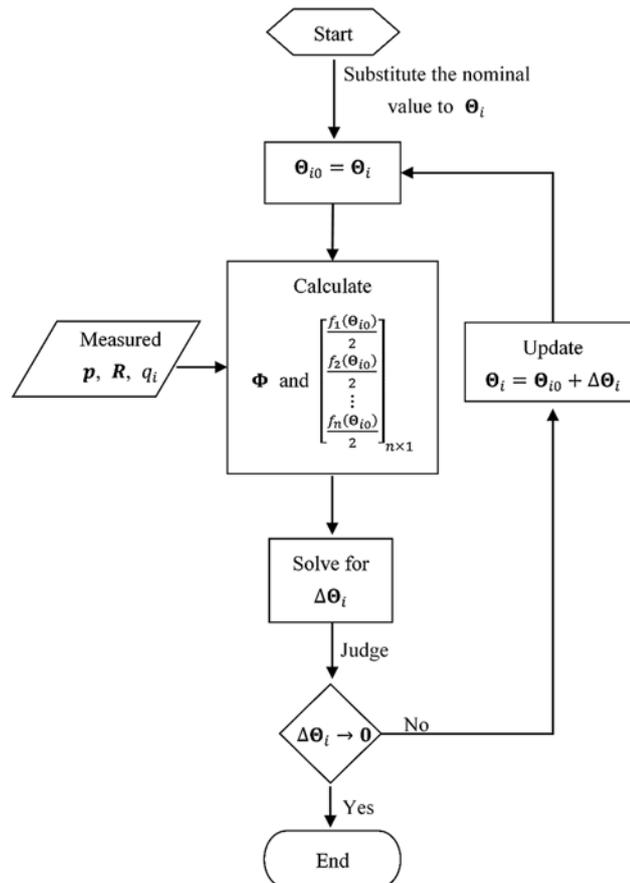


Fig. 3. The flowchart of the iterative process.

4. Experiments

According to the proposed calibration method, the corresponding experiments can be designed. To calibrate this huge parallel robot, the API T3 laser tracker measuring system is used. Its large measuring range and the accuracy for 6D-full-pose measurement can meet our requirements. Its supporting software (SpatialAnalyzer) contains multiple toolboxes for post-processing. In addition, the actuating displacements of the six P joints are measured by the linear encoder installed on each guide rail. The accuracy of this type of linear encoders is $5\ \mu\text{m}$. The coordinates measuring accuracy of the laser tracker is less than $10\ \mu\text{m}$ and the orientation measuring accuracy is 0.008° . The identification for parameters in Group 1 needs the 3D coordinate measurements which can be achieved by the reflecting target of the laser tracker. Figure 4 shows the way of the measurements acquisition. The reflecting target is fixed on the slider of the P joint. When the slider moves along the driving line, a series of 3D coordinates of the reflecting target can be obtained. After the acquisition of the coordinates, the fitting line can be obtained using the toolbox of the supporting software immediately. So the direction vector is firstly determined. The laser tracker should be placed in a suitable location thus to cover the measurements of all the kinematic chains. When k_i is determined, we will go on to determine the remaining parameters of Group 2. On the basis of the error model in the last section, we need to measure the moving platform poses. This job can be achieved by the 6D-full-pose sensor of the laser tracker. The 6D-full-pose sensor is installed on the moving platform which is shown in Fig. 4. With this sensor, the moving platform poses can be directly measured. Meanwhile, the actuating displacement of the six P joints can be obtained by the corresponding linear encoders. As analyzed in Section 3, at least seven groups of measured inputs and outputs are needed for the Group 2 parameters identification. In this experiment, 49 groups of measured inputs and outputs are used. Considering the isotropy, the 49 poses are uniformly distributed inside the workspace of this robot and are randomly chosen. During the experiment, the poses are input to the robot controller and the moving platform follows the input pose one by one. Meanwhile, the actual poses and the relevant actuating displacements are measured. After all the measurements are acquired, the parameters in Group 2 can be solved.

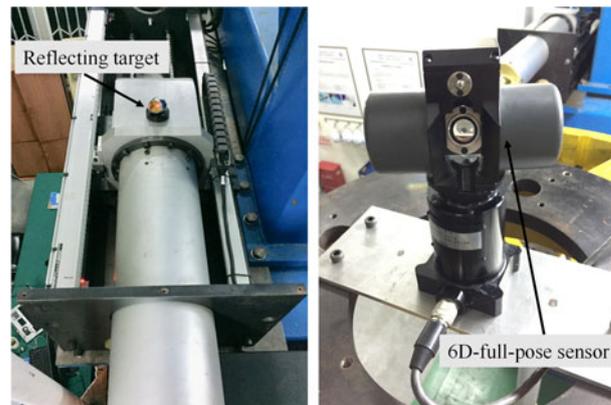


Fig. 4. The acquisition for point coordinates and moving platform poses.

After the identification of all the kinematic parameters, the kinematic model of this robot can be modified. To validate the effectiveness of the calibration, confirmatory experiment should be conducted. We will randomly choose several additional poses inside the robot workspace for the confirmatory experiment. We command the robot to follow the poses. Then, we measure the actual poses to evaluate the pose errors. Each pose will be conducted five times for reducing the influence of measuring noise and uncertainty. So, the major steps of the calibration experiment are designed as follows:

1. Warm up the measuring apparatus and properly arrange the accessories. The reflecting target is pasted on the slider of each P joint. The 6D-full-pose sensor is fixed on the moving platform.
2. Start with one of the six kinematic chains. Command the slider of the kinematic chain to move along the guide rail step by step. During this process, measure the coordinates of the reflecting

- target. It is recommended to collect at least five different coordinates that cover the driving line. Use the toolbox of the measuring software to do the linear fitting to obtain the acquired line.
3. Command the robot to follow the certain amount of given poses one by one. Measure the actual poses by the 6D-full-pose sensor. Meanwhile, obtain the corresponding actuating displacement of the six P joints each time.
 4. Using the above measurements, solve the geometric errors and modify the parameters of the robot kinematics.
 5. For the confirmatory experiment, command the robot to follow the selected poses one by one. Measure the actual poses and then compare with the commanded poses to obtain the pose errors.

When finishing step 2, the parameters in Group 1 which contains k_i with 18 unknowns are identified. When finishing step 4, the parameters in Group 2 which contains m_i' , g_i , and L_i are determined. It should be noted that all the measurements should be converted into the fixed coordinate frame $O\{x, y, z\}$.

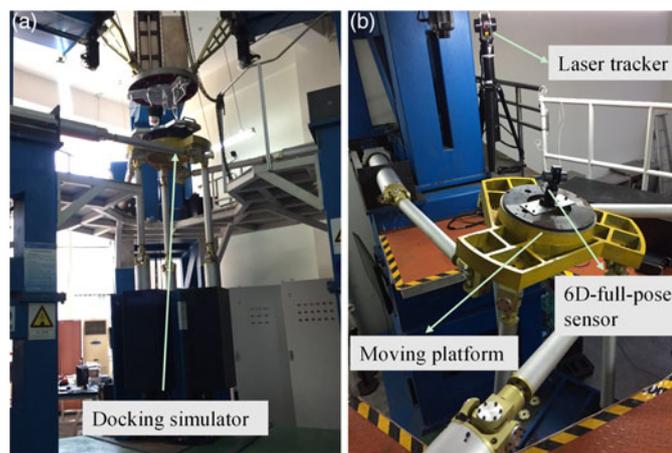


Fig. 5. Photographs of the experiment.

Figure 5 shows some photographs of the experiment. The full size prototype of the docking simulator is shown in the subgraph (a). The workspace of the docking simulator is like a cylinder which is 500 mm in diameter and 400 mm in height. The six S joints are distributed on a circle of diameter 900 mm. The six linkages share the same dimension of the 1200 mm in length. The pose measuring process is shown in the subgraph (b). During the experiment, the laser tracker is placed on a fixed spot. The installation of the 6D-full-pose sensor is suggested to stay close to the center of the moving platform, which will be beneficial to the reduction of the angular measurement error.

5. Results and Conclusions

For the calibration experiment, pose errors are used to evaluate the robot positioning accuracy. The pose error represented by $\Delta X = [\Delta p_x, \Delta p_y, \Delta p_z, \Delta \alpha, \Delta \beta, \Delta \gamma]^T$ is the deviation between the measured pose and the commanded pose. We can see from Figs. 6 and 7 that after calibration, the pose errors are at least reduced by an order of magnitude. The pose errors before calibration nearly reach the range of ± 4 mm and $\pm 0.4^\circ$. After calibration, the poses errors are less than ± 0.3 mm and $\pm 0.035^\circ$. This indicates a significant reduction of the pose errors. In addition, in Table III, the accuracy indices (mean error with three times of standard deviation) of the 6D coordinates of the pose errors are calculated and listed. The accuracy index after calibration is much smaller than that before calibration. We can see that the improvement of the positioning accuracy of the robot is up to 90%.

Table III. The accuracy indices before and after calibration.

Coordinates	Mean error with three times standard deviation		Improved by
	Before calibration	After calibration	
Δp_x	-0.5492 ± 2.5218 mm	0.0336 ± 0.0859 mm	96.60%
Δp_y	-0.2489 ± 3.5589 mm	-0.0173 ± 0.3809 mm	89.30%
Δp_z	-0.2347 ± 6.2213 mm	0.0258 ± 0.3868 mm	93.78%
$\Delta \alpha$	$-0.0108 \pm 0.5010^\circ$	$0.0081 \pm 0.0229^\circ$	95.43%
$\Delta \beta$	$-0.0038 \pm 0.2942^\circ$	$-0.0046 \pm 0.0211^\circ$	92.82%
$\Delta \gamma$	$0.0295 \pm 0.3071^\circ$	$0.0054 \pm 0.0298^\circ$	90.30%

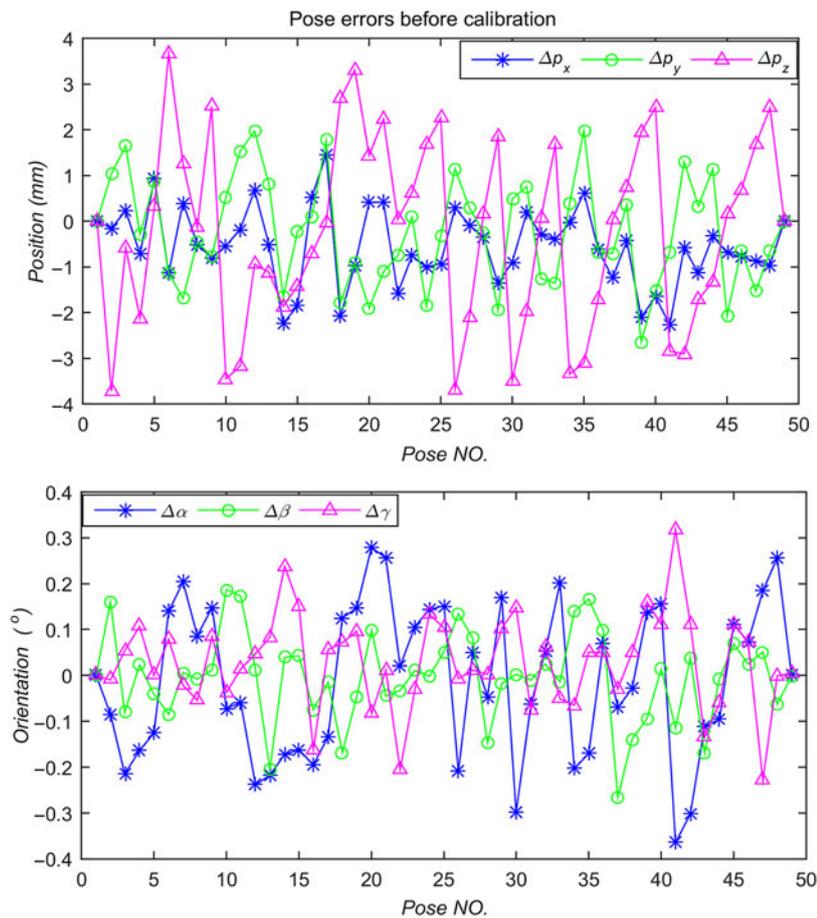


Fig. 6. The pose errors before calibration.

Meanwhile, the identification for the remaining parameters in the simplified error model shows a good convergence during the iteration process. The norms of the solved geometric errors of the six kinematic chains ($|\Delta \Theta_i|$) in each iterative circulation are shown in Fig. 8. It shows that $|\Delta \Theta_i|$ ($i = 1, 2, \dots, 6$) converges quickly in the first three iteration steps and eventually approaches zero within five steps. With a computer which has an Intel Core i3 M350 processor at 2.27 GHz and a 4GB RAM, the average computing time of the whole solving process is less than 1 s by MATLAB, which is quite short time consumption.

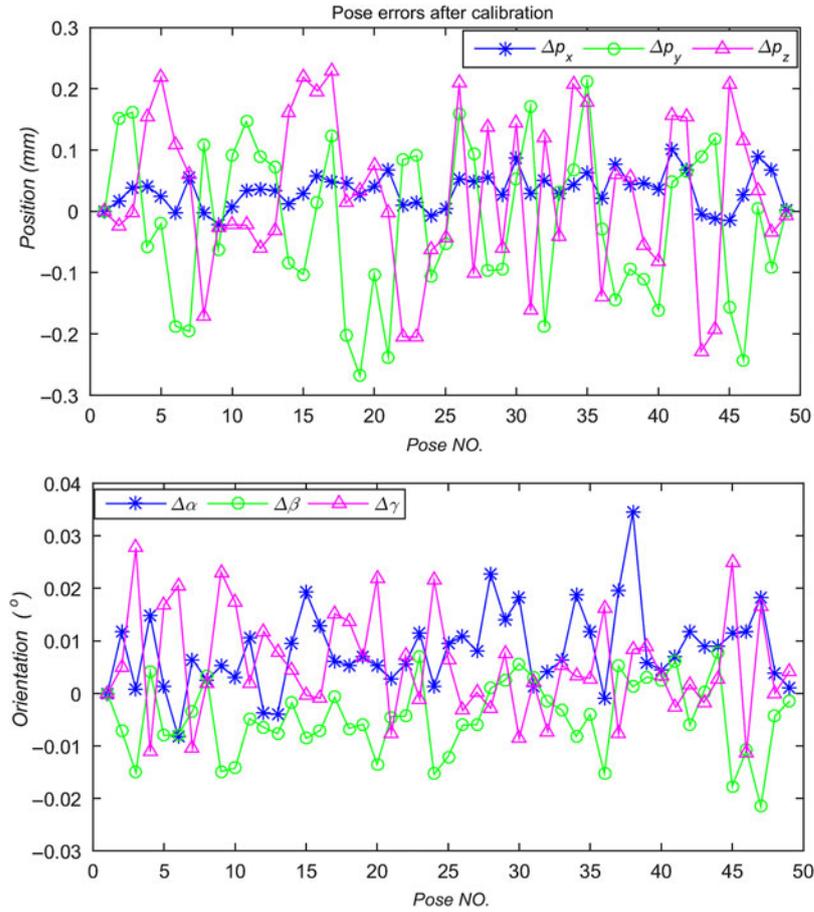


Fig. 7. The pose errors after calibration.

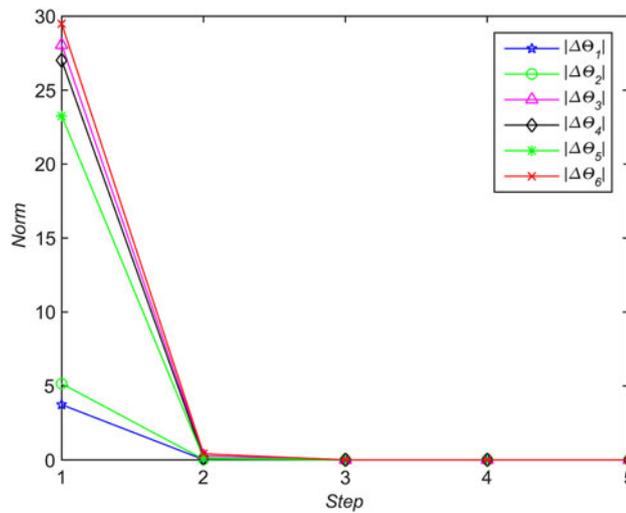


Fig. 8. The convergence of the iteration process.

And lastly, for the confirmatory experiment, we randomly choose another 13 poses inside the workspace for docking. These poses are chosen according to the translational and rotational range of the moving platform during the docking task. The mean error of each pose is recorded in Table IV. As seen from the table, the pose errors of the selected poses are less than 0.25 mm and 0.02° along each coordinate axis. Because the workspace of docking task is less than the robot workspace, these

pose errors are smaller. It should be noted that the x -coordinates of most selected poses are negative because the docking task is taking place above the yOz plane of the fixed coordinate frame $O\{x, y, z\}$.

Table IV. Result of the confirmatory experiment.

No.	Command pose (mm/°)	Pose error (mm/°)
	$X = [p_x, p_y, p_z, \alpha, \beta, \gamma]^T$	$\Delta X = [\Delta p_x, \Delta p_y, \Delta p_z, \Delta \alpha, \Delta \beta, \Delta \gamma]^T$
1	$[10, 0, 130, 13, -14, 14]^T$	$[0.0062, -0.0552, 0.0645, 0.0013, 0.0054, 0.0039]^T$
2	$[10, 130, 0, -13, 14, -14]^T$	$[0.0056, 0.2338, -0.0697, 0.0107, 0.0076, 0.0103]^T$
3	$[10, -91, -91, 13, -14, 14]^T$	$[0.0216, -0.0753, -0.0161, -0.0030, 0.0018, 0.0074]^T$
4	$[-80, 0, 230, 6, -5, 5]^T$	$[0.0738, -0.0708, 0.1839, -0.0085, 0.0025, -0.0047]^T$
5	$[-80, 230, 0, -6, 5, -5]^T$	$[0.0405, 0.2476, 0.0048, -0.0019, 0.0027, 0.0094]^T$
6	$[-80, -162, -162, 6, -5, 5]^T$	$[0.0478, -0.1467, 0.0556, -0.0032, -0.0105, -0.0016]^T$
7	$[-80, 0, 130, 13, -14, 14]^T$	$[0.0228, -0.0990, 0.0931, -0.0016, 0.0044, 0.0037]^T$
8	$[-80, 130, 0, -13, 14, -14]^T$	$[0.0214, 0.2442, -0.0703, 0.0081, 0.0075, 0.0144]^T$
9	$[-80, -91, -91, 13, -14, 14]^T$	$[0.0241, -0.1228, 0.0135, -0.0046, 0.0007, 0.0077]^T$
10	$[-80, 0, 0, 30, 0, 0]^T$	$[-0.0283, -0.0780, -0.0355, -0.0124, 0.0090, -0.0055]^T$
11	$[-160, 0, 90, 13, -6, 6]^T$	$[0.0286, -0.1073, 0.0865, -0.0052, 0.0060, -0.0022]^T$
12	$[-160, 90, 0, -13, 6, -6]^T$	$[-0.0143, -0.0016, -0.0058, -0.0028, 0.0026, -0.0051]^T$
13	$[-160, 64, 64, 13, -6, 6]^T$	$[0.0089, -0.1055, 0.1226, -0.0044, -0.0014, -0.0053]^T$

In conclusion, this paper investigates the kinematic accuracy of a parallel HIL docking simulator and presents a parameter dimension reduction-based kinematic calibration method. Unlike the regular “full-parameter-identification” methods, the proposed method aims at reducing the number of parameters for error modeling. This method starts by separating the relevant parameters into subsets. Then, the parameter subsets are sequentially identified. The principle for parameters separation is summarized in Section 3. With this method, some parameters can be sorted out firstly and then be determined individually through simple measuring process. After this, a simplified error model containing a minimum set of unknowns will be obtained for the remaining parameters. The experimental study based on the proposed method shows good convergence of the parameters identification and great improvement of the robot accuracy. Lastly, the confirmatory experiment validates the effectiveness of this method for the kinematic accuracy enhancement of this docking simulator.

This method is useful to reduce the parameter dimension in kinematic calibration, which will contribute to the simplification of the error model and the reduction of the identification difficulty. Although it is developed from the proposed docking simulator, this method may be applicable to other type of robots which have rotary joint or prismatic joint directly connected to the rack. For instance, the parameters separation can be used on the Hexaglide, the I4, and the Orthoglide mechanisms introduced in ref. [20]. In our future work, we will specialize in the quantitative research of the influences from the dynamics accuracy and the kinematic accuracy on the accuracy performance of the HIL simulation.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 51335007 and No. 51675328).

References

1. A. Yaskевич, “Real time math simulation of contact interaction during spacecraft docking and berthing,” *J. Mech. Eng. Auto.* **4**, 1–15 (2014).
2. N. Uyama, Y. Fujii, K. Nagaoka and K. Yoshida, “Experimental Evaluation of Contact-Impact Dynamics Between a Space Robot with a Compliant Wrist and a Free-Flying Object,” *In: International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Turin, Italy (2012).
3. C. K. Qi, X. C. Zhao, F. Gao, A. Y. Ren and Y. Hu, “Low-order model based divergence compensation for hardware-in-the-loop space discrete contact,” *J. Intell. Robot. Syst.* **86**(1), 81–93 (2017).
4. C. K. Qi, X. C. Zhao, F. Gao, A. Y. Ren and Y. Hu, “Divergence compensation for hardware-in-the-loop simulation of stiffness-varying discrete contact in space,” *Acta Astronaut.* **128**, 295–303 (2016).
5. D. Daney, “Kinematic calibration of the Gough Platform,” *Robotica* **21**(6), 677–690 (2003).

6. C. H. Gong, J. X. Yuan and J. Ni, "Nongeometric error identification and compensation for robotic system by inverse calibration," *Int. J. Mach. Tools Manuf.* **40**(14), 2119–2137 (2000).
7. H. Zhang, G. Cheng, X. Shan and F. Guo, "Kinematic accuracy research of 2(3HUS S) parallel manipulator for simulation of hip joint motion," *Robotica* **36**(9), 1386–1401 (2018).
8. J. M. Renders, E. Rossignol, M. Becquet and R. Hanus, "Kinematic calibration and geometrical parameter identification for robots," *IEEE Trans. Robot. Automat.* **7**(6), 721–732 (1992).
9. J. G. Chen and L. M. Chao, "Positioning error analysis for robot manipulators with all rotary joints," *Proceedings of International Conference on Robotics and Automation*, vol. 3(6) (1986) pp. 1011–1016.
10. M. Becquet, "Analysis of flexibility sources in robot structures," *Proceedings of IMACS/IFAC International Symposium Modelling and Simulation Distributed Parameter System* (1987) pp. 419–424.
11. C. C. Lurascu and F. C. Park, "Geometric algorithms for kinematic calibration of robots containing closed loops," *J. Mech. Des.* **125**(1), 23–32 (2003).
12. P. Huang, J. S. Wang, L. P. Wang and R. Yao, "Kinematic calibration of a hybrid machine tool with regularization method," *Int. J. Mach. Tools Manuf.* **51**(3), 210–220 (2010).
13. B. W. Mooring, *Fundamentals of Manipulator Calibration* (J. Wiley, New York, 1991).
14. J. P. Merlet, *Parallel Robots* (Springer, New York, 2006).
15. W. Khalil, M. Gautier and C. Enguehard, "Identifiable parameters and optimum configurations for robots calibration," *Robot.* **9**(1), 63–70 (1991).
16. H. Q. Zhuang, "Self-calibration of parallel mechanisms with a case study on Stewart platforms," *IEEE Trans. Robot. Automat.* **13**(3), 387–397 (1997).
17. A. J. Patel and K. F. Ehmann, "Calibration of a hexapod machine tool using a redundant leg," *Int. J. Mach. Tools Manuf.* **40**(4), 489–512 (2000).
18. Y. Takeda, G. Shen and H. Funabashi, "A DBB-based kinematic calibration method for in-parallel actuated mechanisms using a Fourier series," *J. Mech. Des.* **126**(5), 856–865 (2004).
19. T. Huang, D. G. Chetwynd, D. J. Whithouse and J. S. Wang, "A general and novel approach for parameter identification of 6-DOF parallel kinematic machines," *Mech. Mach. Theory* **40**(2), 219–239 (2005).
20. P. Renaud, N. Andreff, P. Martinet and G. Gogu, "Kinematic calibration of parallel mechanisms: A novel approach using legs observation," *IEEE Trans. Robot.* **21**(4), 529–538 (2005).
21. X. Ren, Z. Feng and C. Su, "A new calibration method for parallel kinematics machine tools using orientation constraint," *Int. J. Mach. Tools Manuf.* **49**(9), 708–721 (2009).
22. R. He, Y. Zhao, S. Yang and S. Yang, "Kinematic-parameter identification for serial-robot calibration based on POE formula," *IEEE Trans. Robot.* **26**(3), 411–423 (2010).
23. G. Legnani and M. Tiboni, "Optimal design and application of a low-cost wire-sensor system for the kinematic calibration of industrial manipulators," *Mech. Mach. Theory* **73**(2), 25–48 (2014).
24. D. Y. Yu, H. R. Li and W.F. Chen, "Kinematic calibration of parallel robots for docking mechanism motion simulation," *Int. J. Adv. Robot. Syst.* **8**(4), 1079–1084 (2011).
25. X.G. Zhang, H. W. Zou and W. D. Chen, "The research of kinematic calibration method of parallel 6-DOF docking platform," *J. Astronaut.* **28**(2), 273–277 (2007).
26. D. Y. Yu, "Kinematic calibration of super-length journey parallel robots for spacecraft docking motion simulation platform," *Recent Pat. Mech. Eng.* **10** (2017), 296–303.
27. D. Y. Yu, D. C. Cong and J. W. Han, "Accuracy analysis of motion simulator of docking mechanism," *J. Harbin Inst. Technol.* **40**(2), 264–268 (2008).
28. X. Dai, Q. Huang, J. Han and H. Li, "Accuracy synthesis of Stewart platform used in testing system for spacecraft docking mechanism," *Int. Conf. Meas. Technol. Mechatron. Automat.* **3**, 7–10 (2009).
29. D. Y. Yu, "Pose Accuracy Compensation of Parallel Robots Using RBF Neural Network," *In: Control Decision Conference* (2008) pp. 1857–1861. doi: [10.1109/CCDC.2008.4597645](https://doi.org/10.1109/CCDC.2008.4597645)
30. Q. T. Huang, H. Z. Jiang, S. Y. Zhang and J. W. Han, "Spacecraft docking simulation using hardware-in-the-loop simulator with Stewart platform" *Chin. J. Mech. Eng.* **18**(3) (2005).
31. J. D. Mitchell, S. P. Cryan, K. Baker and C.H. Chien, "Integrated docking simulation and testing with the Johnson space center six-degree-of-freedom dynamic test system," *Proceedings of Space Technology and Applications International Forum* (2008) pp. 709–716.
32. G. Grigore, Mobility of mechanisms: A critical review. *Mech. Mach. Theory* **40**(9), 1068–1097 (2005).
33. D. Chablat and P. Wenger, Architecture optimization of a 3-DOF translational parallel mechanism for machining applications, the Orthoglide. *IEEE Trans. Robot. Automat.* **19**(3), 403–410 (2003).