

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Kinematic Parameters Auto-Calibration of Redundant Planar 2-Dof Parallel Manipulator

Shuang Cong<sup>1)</sup>, Chunshi Feng<sup>1)</sup>, Yaoxin Zhang<sup>1)</sup>,  
Zexiang Li<sup>2)</sup> and Shilon Jiang<sup>3)</sup>

<sup>1)</sup> University of Science and Technology of China

<sup>2)</sup> Hong Kong University of Science and Technology

<sup>3)</sup> Googol Technology (Shenzhen) Limited  
P. R. China

## 1. Introduction

Parallel manipulators have the advantage of high speed and high precision in the theory of mechanisms. This has opened up broad possibilities for the use of parallel manipulators in many fields. But in real applications, due to the inevitable manufacturing tolerances and assembling errors, the actual kinematic parameters of parallel manipulators are always unequal to the nominal values and calibration procedures have to be implemented to compensate the kinematic parameter errors between them.

According to the metrology devices adopted, calibration methods of parallel manipulators can be classified into two categories, the external calibration methods and the auto-calibration methods. External calibration methods rely on the precise external 3D measuring devices, such as laser tracking systems (Koseki et al., 1998; Vincze et al., 1994), mechanical devices (Jeong et al., 1999) and camera systems (Zou & Notash, 2001; Renaud et al., 2006). With these external devices, one can measure the end-effector position of parallel manipulators and calibrate the kinematic parameters by minimizing either the errors between the measured end-effector positions and the estimated end-effector positions (Masory et al., 1993), or the errors between the measured joint positions and the estimated joint positions (Zhuang et al., 1995; Zhuang et al., 1998). The auto-calibration methods rely on the redundant joint sensors of parallel manipulators, which can be achieved by adding extra sensors to the uninstrumented joints (Baron & Angeles, 1998; Zhuang, 1997; Wampler et al., 1995; Patel & Ehmman, 2000), or by constraining the motion of end-effector or some joints (Khalil & Besnard, 1999; Wang & Masory, 1993). With the redundant joint sensors, extra information can be obtained for the sampled configurations without employing any external measuring devices (Hollerbach & Wampler, 1996; Yiu et al., 2003c; Chiu & Perng, 2004), and the auto-calibration procedure is usually implemented by minimizing a function of closed-loop constraint errors. Obviously, it is more convenient to measure the sampled configurations by the redundant joint sensors than the external 3D measuring devices. especially for the parallel manipulator with inherent redundant joint sensors. But it is

Source: Parallel Manipulators, New Developments, Book edited by: Jee-Hwan Ryu, ISBN 978-3-902613-20-2, pp. 498, April 2008, I-Tech Education and Publishing, Vienna, Austria

usually difficult to minimize the closed-loop constraint error function, and the calibration results of the auto-calibration methods are usually dependent on the error function adopted in the calibration.

In this chapter, we will calibrate the kinematic parameters of a planar 2-dof parallel manipulator. In the literatures, this type of parallel manipulator has been studied from different aspects. Yiu and Zhang studied the kinematics of the parallel manipulator (Yiu & Li, 2003b; Zhang, 2006). Liu studied the singularities of the parallel manipulator with a geometric method (Liu et al., 2001a; Liu et al., 2003). Furthermore, the dynamics and controller design problem of the parallel manipulator were studied by Liu (Liu et al., 2001a; Liu et al., 2001b; Liu & Li, 2002), Kock (Kock & Schumacher, 2000a; Kock & Schumacher, 2000b), Yiu (Yiu & Li, 2001; Yiu & Li, 2003a), Cheng (Cheng et al., 2003), Shen (Shen et al., 2003) and Zhang (Zhang & Cong, 2005). In this chapter, we will study the calibration and solve three problems. In the second part of the chapter, based on the study of the relationship between the projected tracking error of the joint angles and the error of the sensor zero positions, we propose a projected tracking error function for the calibration of the sensor zero positions of a planar 2-dof parallel manipulator with a redundant joint sensor. With a simple searching strategy for the minimal value of the error function, an auto-calibration procedure is designed, and the validity of the calibration procedure is verified through actual experiments on a real redundant planar 2-dof parallel manipulator. In the third part of the chapter, by eliminating the passive joint positions, we derive another type of error function with only the variables of the active joint positions. Moreover, by decoupling the products items of the kinematic parameters in the error function into the linear combinations of a group of new variables, the error function minimization process is simplified and the calibration precision can be improved further. Based on two error functions proposed in this section, an auto-calibration method and design procedure is given, and the validity of the auto-calibration method is studied with stepwise simulations. Under the assumption that only one coordinate is known accurately beforehand, the other 11 kinematic parameters of the parallel manipulator including 3 sensor zero positions, 6 link lengths and 2 base coordinates can be calibrated precisely. In order to obtain the global optimum and auto-calibrate all parameters of the parallel manipulator, in the fourth part of the chapter, three stochastic optimization algorithms including genetic algorithm (Holland, 1975), particle swarm optimization (Kennedy & Eberhart, 1995) and differential evolution (Storn & Price, 1995) are applied to minimize the error functions proposed in the third part of the chapter, respectively. In the applications, the performances of the applied algorithms on the problem are compared under the different methods. Finally, actual calibration is carried out based on differential evolution algorithms, and the results demonstrate that all of the 12 parameters of the parallel manipulator are calibrated with high accuracy. We'll end the chapter with the conclusions.

## **2. Auto-calibration of sensor zero positions based on the projected tracking Error**

### **2.1 Calibration problem of the sensor zero positions**

The structure of the planar 2-dof parallel manipulator to be calibrated is shown in Fig. 1, in which the parallel manipulator consists of 6 links with the same lengths, and 3 active joints located at  $A_1$ ,  $A_2$ ,  $A_3$ , and 3 passive joints located at  $B_1$ ,  $B_2$ ,  $B_3$ . The end-effector of the parallel manipulator coincides with  $O$ . According to Fig. 1, a reference frame is established in the

workspace of the parallel manipulator. The zero positions of the joint angles are all defined as the positive direction of the  $X$  axis of the reference frame, and the positive directions of the angles are all defined as the anticlockwise direction.

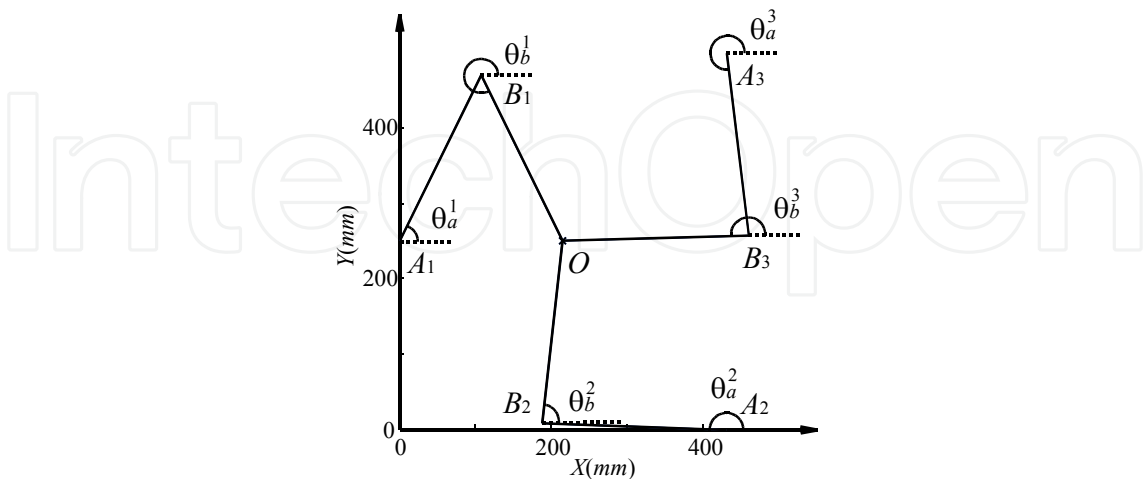


Fig. 1 Structure of the planar 2-dof parallel manipulator

The kinematics of the parallel manipulator has been studied by Yiu and Zhang (Yiu & Li, 2003b; Zhang, 2006), and the end-effector coordinate can be calculated from the active joint angles through the kinematics. For the active joints located at  $A_1$ ,  $A_2$ ,  $A_3$ , each is attached with a position sensor, through which the active joint angles can be measured. But due to the assembly errors, there is always some bias angle between each sensor zero position and the zero position of the active joint. So active joint angles can be formulated as the sum of the sensors readings and the sensor zero positions.

$$\theta_a^i = \theta_m^i + \theta_z^i, i = 1,2,3 \tag{1}$$

in which symbols  $\theta_a^i, i = 1,2,3$  refer to the active joint angles, symbols  $\theta_m^i, i = 1,2,3$  refer to the sensor readings and symbols  $\theta_z^i, i = 1,2,3$  refer to the estimations of the sensor zero positions. Obviously the precision of  $\theta_z^i, i = 1,2,3$  determines the precision of the parallel manipulator in real applications. But it is difficult to measure the sensor zero positions directly. So calibration procedure has to be implemented to estimate their actual values.

Usually, the sensor zero positions are estimated through the following manual procedure. First move the end-effector to a predefined position in the workspace manually, for which the corresponding active joint angles  $\theta_a^i, i = 1,2,3$  have been known accurately. Then record the sensor readings  $\theta_m^i, i = 1,2,3$  and one can get the estimations of the sensor zero positions  $\theta_z^i, i = 1,2,3$  by subtracting the sensor readings from the active joint angles.

The calibration procedure mentioned above is convenient to be implemented, but the precision of the calibration results of the sensor zero positions is usually limited, since it is difficult to move the end-effector to the predefined position precisely, there is always several millimeter error between the real position of the end-effector and the predefined position. To solve this problem, Yiu proposed two iterative algorithms to calibrate the parallel manipulator (Yiu et al., 2003c), but the robustness of Yiu's method was not proved. Here we propose a new calibration method based on the projected tracking error of the active joint angles of the parallel manipulator.

## 2. 2 Error function based on the projected tracking error

To formulate the tracking error of the parallel manipulator, we define the tracking error of the end-effector  $dx, dy$  and the tracking error of the joint angles  $d\theta_a^i, d\theta_b^i, i = 1, 2, 3$  as follows:

$$\begin{aligned} dx &= x - \hat{x}, dy = y - \hat{y} \\ d\theta_a^i &= \theta_a^i - \hat{\theta}_a^i, d\theta_b^i = \theta_b^i - \hat{\theta}_b^i, i = 1, 2, 3 \end{aligned} \quad (2)$$

in which symbols  $(\hat{x}, \hat{y})$ ,  $\hat{\theta}_a^i, i = 1, 2, 3$  and  $\hat{\theta}_b^i, i = 1, 2, 3$  refer to the end-effector coordinate, active joint angles and passive joint angles of the desired path respectively, while symbols  $(x, y)$ ,  $\theta_a^i, i = 1, 2, 3$  and  $\theta_b^i, i = 1, 2, 3$  are the counterparts of the real path respectively. According to (1), the tracking error of the active joint angles  $d\theta_a^i, i = 1, 2, 3$  can be formulated as the sum of the tracking error of the sensor readings  $d\theta_m^i, i = 1, 2, 3$  and the error of the sensor zero positions  $d\theta_z^i, i = 1, 2, 3$ . So one can have:

$$d\theta_a^i = d\theta_m^i + d\theta_z^i, i = 1, 2, 3 \quad (3)$$

in which symbols  $d\theta_m^i, d\theta_z^i, i = 1, 2, 3$  are defined by

$$\begin{aligned} d\theta_m^i &= \theta_m^i - \hat{\theta}_m^i \\ d\theta_z^i &= \theta_z^i - \hat{\theta}_z^i, i = 1, 2, 3 \end{aligned} \quad (4)$$

where symbols  $\hat{\theta}_m^i, i = 1, 2, 3$  and  $\theta_m^i, i = 1, 2, 3$  refer to the desired sensor readings and real sensor readings respectively, symbols  $\hat{\theta}_z^i, i = 1, 2, 3$  and  $\theta_z^i, i = 1, 2, 3$  refer to the actual value and estimated value of the sensor zero positions respectively. According to (4), the tracking error of the sensor readings  $d\theta_m^i$  is defined as the difference between the actual value and the desired value of the sensor readings, and the error of the sensor zero positions  $d\theta_z^i$  is defined as the difference between the estimated value and the actual value of the sensor zero positions.

Denote the link length of the parallel manipulator by symbol  $l$ , and the coordinates of the three bases by  $(x_a^i, y_a^i), i = 1, 2, 3$ . Then according to the kinematics of the parallel manipulator, one can express end-effector coordinate of the desired path  $(\hat{x}, \hat{y})$  through following equations:

$$\begin{aligned} \hat{x} &= x_a^i + l \cdot \cos(\hat{\theta}_a^i) + l \cdot \cos(\hat{\theta}_b^i) \\ \hat{y} &= y_a^i + l \cdot \sin(\hat{\theta}_a^i) + l \cdot \sin(\hat{\theta}_b^i), i = 1, 2, 3 \end{aligned} \quad (5)$$

As a result of trajectory tracking, the real path of the end-effector  $(x, y)$  can be expressed as follows:

$$\begin{aligned} x &= x_a^i + l \cdot \cos(\theta_a^i) + l \cdot \cos(\theta_b^i) \\ y &= y_a^i + l \cdot \sin(\theta_a^i) + l \cdot \sin(\theta_b^i) \quad i = 1, 2, 3 \end{aligned} \quad (6)$$

Suppose that the estimated values of the sensor zero positions  $\theta_z^i, i = 1, 2, 3$  are accurate and equal to the actual values  $\hat{\theta}_z^i, i = 1, 2, 3$  exactly, so one can have

$$\theta_z^i = \hat{\theta}_z^i, i = 1, 2, 3 \quad (7)$$

According to (7) and (1), the active joint angles can be measured accurately, and both of the desired path and the real path lie in the configuration space of the parallel manipulator accurately. By subtracting (5) from (6), one can formulate the tracking error of the parallel manipulator as following equations:

$$\begin{aligned} x - \hat{x} &= l \cdot \cos(\theta_a^i) + l \cdot \cos(\theta_b^i) - l \cdot \cos(\hat{\theta}_a^i) - l \cdot \cos(\hat{\theta}_b^i) \\ y - \hat{y} &= l \cdot \sin(\theta_a^i) + l \cdot \sin(\theta_b^i) - l \cdot \sin(\hat{\theta}_a^i) - l \cdot \sin(\hat{\theta}_b^i) \end{aligned}, i = 1, 2, 3 \quad (8)$$

Implement the Taylor series expansion on (8) and ignore the high-order items of the tracking errors, one can have:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} -l \cdot \sin(\hat{\theta}_a^i) & -l \cdot \sin(\hat{\theta}_b^i) \\ l \cdot \cos(\hat{\theta}_a^i) & l \cdot \cos(\hat{\theta}_b^i) \end{bmatrix} \begin{bmatrix} d\theta_a^i \\ d\theta_b^i \end{bmatrix}, i = 1, 2, 3 \quad (9)$$

Solve (9), one can express the tracking error of the joint angles  $d\theta_a^i, d\theta_b^i, i = 1, 2, 3$  by the tracking error of the end-effector  $dx, dy$  as following equations:

$$\begin{bmatrix} d\theta_a^i \\ d\theta_b^i \end{bmatrix} = \frac{1}{l \cdot \sin(\hat{\theta}_a^i - \hat{\theta}_b^i)} \begin{bmatrix} -\cos(\hat{\theta}_b^i) & -\sin(\hat{\theta}_b^i) \\ \cos(\hat{\theta}_a^i) & \sin(\hat{\theta}_a^i) \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix}, i = 1, 2, 3 \quad (10)$$

And the tracking error of the active joint angles  $d\theta_a^i, i = 1, 2, 3$  can be expressed by the tracking error of the end-effector as follows:

$$\begin{bmatrix} d\theta_a^1 \\ d\theta_a^2 \\ d\theta_a^3 \end{bmatrix} = - \begin{bmatrix} \frac{\cos(\hat{\theta}_b^1)}{l \cdot \sin(\hat{\theta}_a^1 - \hat{\theta}_b^1)} & \frac{\sin(\hat{\theta}_b^1)}{l \cdot \sin(\hat{\theta}_a^1 - \hat{\theta}_b^1)} \\ \frac{\cos(\hat{\theta}_b^2)}{l \cdot \sin(\hat{\theta}_a^2 - \hat{\theta}_b^2)} & \frac{\sin(\hat{\theta}_b^2)}{l \cdot \sin(\hat{\theta}_a^2 - \hat{\theta}_b^2)} \\ \frac{\cos(\hat{\theta}_b^3)}{l \cdot \sin(\hat{\theta}_a^3 - \hat{\theta}_b^3)} & \frac{\sin(\hat{\theta}_b^3)}{l \cdot \sin(\hat{\theta}_a^3 - \hat{\theta}_b^3)} \end{bmatrix} \begin{bmatrix} dx \\ dy \end{bmatrix} \quad (11)$$

Define the tracking error vector of the active joint angles by  $d\theta_a = [d\theta_a^1 \ d\theta_a^2 \ d\theta_a^3]^T$  and the tracking error vector of the end-effector by  $dxy = [dx \ dy]^T$ . Then one can express (11) by:

$$d\theta_a = J \cdot dxy \quad (12)$$

where the symbol  $J$  refers to the Jacobian matrix of the active joint angles with the end-effector coordinate.

With the assumption expressed in (7), one can have  $d\theta_a^i = d\theta_m^i, i = 1, 2, 3$ . So (12) can be formulated as follows:



$$d\theta_m = J \bullet dxy \quad (13)$$

where the symbol  $d\theta_m$  is defined by  $d\theta_m = [d\theta_m^1 \ d\theta_m^2 \ d\theta_m^3]^T$ .

Define the image space of the Jacobian matrix  $J$  as the feasible subspace, and the orthogonal complement of the feasible subspace as the infeasible subspace. According to (13), the tracking error vector of the sensor readings  $d\theta_m$  lies in the feasible subspace and the projection of  $d\theta_m$  into the infeasible subspace is zero, which can be formulated by

$$Pd\theta_m = 0 \quad (14)$$

where symbol  $P$  is the linear projection that can project the vector into the infeasible subspace. The matrix representation of  $P$  can be formulated as

$$P = I - J(J^T J)^{-1} J^T \quad (15)$$

where symbol  $I$  is the identity matrix and symbol  $J$  is the Jacobian matrix of the active joint angles with the end-effector coordinate. Obviously, the matrix  $P$  is symmetric and idempotent, so one can have following equations:

$$P = P^T$$

$$P = P^2 \quad (16)$$

With the linear projection  $P$ , one can project the vector of the tracking error  $d\theta_a$  into the infeasible subspace, and define the projected tracking error of the active joint angles by

$$d\theta_e = Pd\theta_a \quad (17)$$

where the symbol  $d\theta_e = [d\theta_e^1 \ d\theta_e^2 \ d\theta_e^3]^T$  refers to the projected tracking error of the active joint angles and equals to the infeasible component of  $d\theta_a$ .

Let  $d\theta_z = [d\theta_z^1 \ d\theta_z^2 \ d\theta_z^3]^T$  be the vector of the error of the sensor zero positions. With (3), one can have that  $d\theta_a$  equals to the sum of  $d\theta_m$  and  $d\theta_z$ . So the infeasible component of  $d\theta_a$  equals to the sum of the infeasible component of  $d\theta_m$  and  $d\theta_z$ . According to (14), the tracking error vector of the sensor readings  $d\theta_m$  can be viewed as a feasible component of  $d\theta_a$ , and its infeasible component always vanishes. Then for the error of the sensor zero positions  $d\theta_z$ , which usually has nonvanishing feasible component and infeasible component simultaneously, its infeasible component equals to the infeasible component of  $d\theta_a$ . At last through (3) and (14) one can have following equation for the projected tracking error  $d\theta_e$ :

$$d\theta_e = Pd\theta_a = P(d\theta_m + d\theta_z) = Pd\theta_m + Pd\theta_z = Pd\theta_z \quad (18)$$

According to (18), the projected tracking error of the active joint angles  $d\theta_e$  equals to the projection of the vector  $d\theta_z$  into the infeasible subspace. So by minimizing the following function of the projected tracking error  $d\theta_e$ , the value of  $d\theta_z$  can be minimized and the sensor zero positions can be calibrated:

$$E = \frac{1}{n} \sum_{j=1}^n d\theta_{ej}^T \bullet d\theta_{ej} \quad (19)$$

in which the symbol  $d\theta_{ej}$  is the  $j$ th sampled projected tracking errors and the symbol  $n$  refers to the number of the sampled projected tracking errors.

### 2.3 Analysis of robustness of the error function

In real application, the readings of sensor are usually inaccurate, so the precision of the calibration results may be limited by the accuracy of the sensor readings. In this section, we will prove that the error function expressed in (19) is robust to the measurement error of the sensor readings, so accurate calibration results can be obtained by minimizing the error function.

Suppose that the number of the sampled sensor readings is  $n$ , and denote the measurements error of the  $j$ th sensor readings by the symbol  $\delta_{mj} = [\delta_{mj}^1 \ \delta_{mj}^2 \ \delta_{mj}^3]^T$ . According to (3), the tracking error of the active joint can be formulated by

$$d\theta_a = \delta_{mj} + d\theta_{mj} + d\theta_z, \quad j = 1, \dots, n \quad (20)$$

where symbols  $d\theta_{aj}, j = 1, \dots, n$  is the tracking error of the active joint angles and  $d\theta_{mj}, j = 1, \dots, n$  is the tracking error of the sensor readings. Then following two propositions can be proved for the robustness of the error function to the measurement error  $\delta_{mj}, j = 1, \dots, n$ .

**Proposition 1:** Suppose that the measurements error vector  $\delta_{mj}, j = 1, \dots, n$  lies in the feasible subspace. The value of the error function  $E$  will not be affected by the measurement error and accurate calibration results of the sensor zero positions can be obtained by minimizing the error function  $E$ .

**Proof:** For the assumption that the measurement error  $\delta_{mj}, j = 1, \dots, n$  lies in the feasible subspace, one can have

$$P_j \delta_{mj} = 0, \quad j = 1, \dots, n \quad (21)$$

where the symbol  $P_j$  refer to the linear projection of the  $j$ th desired configuration. Then following equation can be obtained for the projected tracking error of the active joint angles  $d\theta_{ej}, j = 1, \dots, n$ :

$$d\theta_{ej} = P_j d\theta_{aj} = P_j (\delta_{mj} + d\theta_{mj} + d\theta_z) = P_j \delta_{mj} + P_j d\theta_{mj} + P_j d\theta_z = P_j d\theta_z, \quad j = 1, \dots, n \quad (22)$$

According to (22), although the sensor readings are not accurate, the projected tracking error of the active joint angles  $d\theta_{ej}, j = 1, \dots, n$  still equals to the projection of the vector  $d\theta_z$  into the infeasible subspace. Substitute (22) into (19), one can find that the value of the error function  $E$  will not be affected by the measure error  $\delta_{mj}, j = 1, \dots, n$ . By minimizing the error function, the error vector  $d\theta_z$  can be minimized, and accurate calibration results can be obtained for the sensor zero positions. So the robustness of the error function is proved.

**Proposition 2:** Suppose that the measurement error vector  $\delta_{mj}, j = 1, \dots, n$  lies in the infeasible subspace. Denote the mean value of  $\delta_{mj}, j = 1, \dots, n$ . by  $\bar{\delta}_m$ , the variance of  $\delta_{mj}, j = 1, \dots, n$ . By  $\sigma^2$ . If the mean value  $\bar{\delta}_m$  equals to zero, then accurate calibration results of the sensor zero positions still can be obtained by minimizing the error function  $E$ .

**Proof:** For the assumption that the measurement error  $\delta_{mj}, j = 1, \dots, n$  lies in the infeasible subspace, one can have

$$P_j \delta_{mj} = \delta_{mj}, \quad j = 1, \dots, n \quad (23)$$

Then following equations can be obtained for the projected tracking error of the active joint angles  $d\theta_{ej}, j = 1, \dots, n$ :

$$d\theta_{ej} = P_j d\theta_{aj} = P_j (\delta_{mj} + d\theta_{mj} + d\theta_z) = P_j \delta_{mj} + P_j d\theta_{mj} + P_j d\theta_z = \delta_{mj} + P_j d\theta_z, \quad j = 1, \dots, n \quad (24)$$



Substitute (24) into (19), one can calculate the value of the error function  $E$  as follows:

$$\begin{aligned}
 E &= \frac{1}{n} \sum_{j=1}^n d\theta_{ej}^T \cdot d\theta_{ej} = \frac{1}{n} \sum_{j=1}^n (\delta_{mj} + P_j d\theta_z)^T \cdot (\delta_{mj} + P_j d\theta_z) \\
 &= \frac{1}{n} \sum_{j=1}^n (\delta_{mj}^T \delta_{mj} + \delta_{mj}^T P_j d\theta_z + d\theta_z^T P_j^T \delta_{mj} + d\theta_z^T P_j^T P_j d\theta_z) \\
 &= \frac{1}{n} \sum_{j=1}^n \delta_{mj}^T \delta_{mj} + \frac{2}{n} \sum_{j=1}^n d\theta_z^T P_j^T \delta_{mj} + \frac{1}{n} \sum_{j=1}^n d\theta_z^T P_j^T P_j d\theta_z
 \end{aligned} \tag{25}$$

With (16) and (23), equation (25) can be simplified as follows:

$$E = \frac{1}{n} \sum_{j=1}^n \delta_{mj}^T \delta_{mj} + \frac{2}{n} d\theta_z^T \sum_{j=1}^n \delta_{mj} + \frac{1}{n} \sum_{j=1}^n d\theta_z^T P_j^T P_j d\theta_z \tag{26}$$

Consider the assumption of the mean value and variance of the measurement error, one can have

$$\begin{aligned}
 \bar{\delta}_m &= \frac{1}{n} \sum_{j=1}^n \delta_{mj} = 0 \\
 \sigma^2 &= \frac{1}{n} \sum_{j=1}^n \delta_{mj}^T \delta_{mj}
 \end{aligned} \tag{27}$$

Then with (27), the value of the error function  $E$  can be formulated as follows:

$$E = \sigma^2 + \frac{1}{n} \sum_{j=1}^n d\theta_z^T P_j^T P_j d\theta_z \tag{28}$$

One can see that, with the measurement error lie in the infeasible subspace, the value of the error function  $E$  equals to the norm of the projected tracking error plus the variance of the measurement error, which is a constant. So by minimizing the error function  $E$ , accurate calibration results of the sensor zero positions still can be obtained, and the robustness of the error function is proved.

## 2.4 Auto-calibration procedure based on the error function

In this subsection, with a simple searching strategy for the minimal value of error function proposed above, we will design an auto-calibration procedure for the sensor zero positions of the parallel manipulator. The auto-calibration procedure based on the simple searching strategy is proposed as follows:

Step 1: Move the end-effector manually to a predefined reference point  $O$ , for which the active joint angles have been known accurately beforehand. Denote the coordinate of  $O$  by symbol  $(x_o, y_o)$ , and denote the actual coordinate of the end-effector position by symbol  $(x_r, y_r)$ . Here we call  $(x_r, y_r)$  as the initial estimation of the reference point  $O$

- Step 2: Denote the current real position of the end-effector by symbol  $R$ . Take  $R$  as the estimation of  $O$ , and record the sensor readings corresponding to  $R$ . One can get the estimation of the sensor zero positions by subtracting the sensor readings of  $R$  from the active joint angles of  $O$ .
- Step 3: Drive the end-effector of the parallel manipulator to track a predefined circular trajectory, for example, the circular trajectory with center  $O$ . Record the sensor readings of the real trajectory corresponding to the interpolation point of the desired trajectory. Then calculate the tracking errors of the active joint angles  $d\theta_a$  corresponding to each interpolation point of the desired trajectory by subtracting the estimated sensor zero positions and the sensor readings from the desired active joint angles.
- Step 4: Calculate the projected tracking errors  $d\theta_e$  through (18) and the value of the error function  $E$  through (19).
- Step 5: Take the initial estimation  $(x_r, y_r)$  of  $O$  as the center of the searching region, and the predefined value  $d$  as the scope of the searching region. Then drive the end-effector to the points with coordinates  $(x_r+d, y_r)$ ,  $(x_r+d, y_r+d)$ ,  $(x_r, y_r+d)$ ,  $(x_r-d, y_r+d)$ ,  $(x_r-d, y_r)$ ,  $(x_r-d, y_r-d)$ ,  $(x_r, y_r-d)$ ,  $(x_r+d, y_r-d)$  in order. Take these points as estimations of the reference point  $O$  and repeat the operations described in step 2 to step 4 for each of these points. Then one can get an estimation of the sensor zero positions and a value of the error function  $E$  for each of these points.
- Step 6: Find out the coordinate of the end-effector corresponding to the minimal  $E$  among the 9 estimations of the reference point  $O$ , and denote it by symbol  $(x_m, y_m)$ . If the coordinate  $(x_m, y_m)$  equals to the initial estimation  $(x_r, y_r)$ , then go to step 7. If not, take the coordinate  $(x_m, y_m)$  as a new initial estimation of the reference point  $O$ , and go to step 2.
- Step 7: If the value of  $d$  is smaller than the predefined lower limit, then the calibration procedure comes to the end and the sensor zero positions corresponding to the point  $(x_r, y_r)$  is the calibration result. If not, divide the variable  $d$  by 2 and go to step 2.

## 2.5 Experiments on a real planar parallel manipulator

With the calibration procedure proposed above, we will calibrate the sensor zero positions of Googol Tech Ltd's GPM2002, which is a planar 2-dof parallel manipulator with a redundant joint sensor. The mechanisms of the parallel manipulator GPM2002 is shown in Fig.2, in which GPM2002 is composed of 6 links and 6 joints. Similar to Fig.1, a reference frame is established in the workspace of GPM2002. Under the reference frame, the coordinate of the 3 bases are (0,250), (433,0) and (433,500) respectively, and the coordinate of the home position of the end-effector is (216.5,250). The lengths of all the 6 links equal to 244mm. The 3 joints located at the bases are actuated by an AC servo motor respectively, while the other 3 joints are unactuated. Each of the AC servo motor is embedded with an internal absolute encoder, with which the active joint angles can be measured.

Based on the auto-calibration procedure proposed in subsection 2.4, an auto-calibration program is realized with VC++. The reference point  $O$  of the auto-calibration procedure is defined as home position of the end-effector (216.5,250), and the desired trajectory to be tracked is defined as a circle with center (216.5,250). The initial value of  $d$  is defined as 4mm and the lower limit of  $d$  0.125mm. Then calibration experiments are implemented and the bias angles between the sensor zero positions and the active joint zero positions of GPM2002

are calibrated. The calibration experiments were implemented for 3 times, each time with a different initial estimation of the reference point (216.5, 250), and so different initial estimation of the sensor zero positions. Experiment results are shown in Table 1.

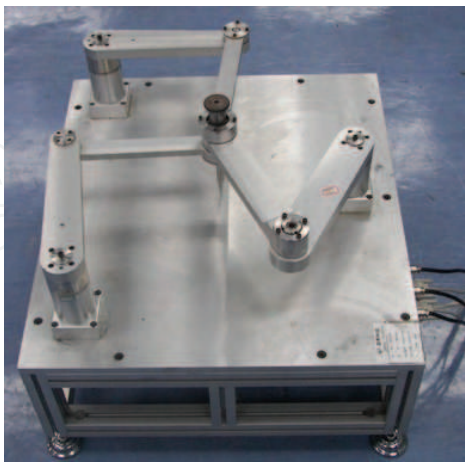


Fig. 2. Structure of GPM2002

	Initial Estimation ( <i>rad</i> )			Calibration Results ( <i>rad</i> )		
	Active Joint1	Active Joint2	Active Joint3	Active Joint1	Active Joint2	Active Joint3
Experiment1	1.0325	3.1045	4.8055	1.0661	3.0797	4.7974
Experiment2	1.0362	3.0796	4.8413	1.0652	3.0798	4.7971
Experiment3	1.0888	3.0651	4.7880	1.0663	3.0800	4.7966

Table 1. Calibration results of the sensor zero positions

As shown in Table 1, from different initial estimated values, the calibrated sensor zero positions will converge to an identical value with the decrease of the projected tracking errors in the end.

**3. Kinematic parameters calibration of redundant planar 2-dof parallel manipulator with a new error function**

In the former section, the three sensor zero positions have been calibrated by optimizing the error function we proposed. In this section, we will further calibrate the other parameters of the parallel manipulator. Based on the minimization of the closed-loop constraint errors, Yiu proposed an auto-calibration procedure for the planar 2-dof parallel manipulator (Yiu et al., 2003c). But for the parallel manipulator, the difficulty of minimizing the closed-loop constraint error function increases as the number of kinematic parameters to be calibrated increases. Among the 12 independent kinematic parameters, only 3 sensor zero positions were calibrated successfully in Yiu’s paper and also in the second section of present chapter, while the other 9 parameters were supposed to be known beforehand. By eliminating the passive joint variables of the closed-loop constraint equations, we will simplify the

formulation of the closed-loop constraint equations, and propose a new error function to calibrate not only the sensor zero positions but also other kinematic parameters of the parallel manipulator. Compared to Yiu's error function, which involves both active joint positions and passive joint positions as variables, our error function involves only the active joint positions as variables. Besides, by decoupling the product item of the kinematic parameters of the error function into linear combinations of a group of new variables, we simplify the minimization process and improve the calibration precision further.

### 3.1 A new error function

To formulate the kinematics of GPM2002, we denote the coordinate of the end-effector  $O$  by  $(x, y)$ , the active joint angles by  $\theta_a^i$ ,  $i = 1, 2, 3$ , the passive joint angles by  $\theta_b^i$ ,  $i = 1, 2, 3$ , the lengths of the links connected to the active joints by  $l_a^i$ ,  $i = 1, 2, 3$ , the lengths of the links connected to the passive joints by  $l_b^i$ ,  $i = 1, 2, 3$ , and the coordinates of  $A_1, A_2, A_3$  by  $(x_a^i, y_a^i)$ ,  $i = 1, 2, 3$ . While the nominal values of the link lengths  $l_a^i, l_b^i$ ,  $i = 1, 2, 3$  are all  $244\text{mm}$ , the nominal values of the base coordinates  $(x_a^i, y_a^i)$ ,  $i = 1, 2, 3$  are  $(0, 250)$ ,  $(433, 0)$ ,  $(433, 500)$  respectively. Then from the joint angles, the coordinate of the end-effector can be calculated through following equations:

$$\begin{aligned} x &= x_a^i + l_a^i \cdot \cos(\theta_a^i) + l_b^i \cdot \cos(\theta_b^i) \\ y &= y_a^i + l_a^i \cdot \sin(\theta_a^i) + l_b^i \cdot \sin(\theta_b^i) \quad i = 1, 2, 3 \end{aligned} \quad (29)$$

Define  $x_b^i, y_b^i$ ,  $i = 1, 2, 3$  as follows:

$$\begin{aligned} x_b^i &= x_a^i + l_a^i \cdot \cos(\theta_a^i) \\ y_b^i &= y_a^i + l_a^i \cdot \sin(\theta_a^i) \quad i = 1, 2, 3 \end{aligned} \quad (30)$$

Substitute (30) into (29), one can reformulate (29) into the following quadratic equations:

$$(x - x_b^i)^2 + (y - y_b^i)^2 = l_b^{i2}, \quad i = 1, 2, 3 \quad (31)$$

From (29), (30) and (31), we can have the following equation:

$$\begin{aligned} 2(x_b^2 - x_b^1)x + 2(y_b^2 - y_b^1)y &= d_2 - d_1 \\ 2(x_b^3 - x_b^1)x + 2(y_b^3 - y_b^1)y &= d_3 - d_1 \end{aligned} \quad (32)$$

with  $d_i$ ,  $i = 1, 2, 3$  defined as  $d_i = x_b^{i2} + y_b^{i2} - l_b^{i2}$ ,  $i = 1, 2, 3$ .

Then the coordinates of the end-effector  $(x, y)$  can be solved from (32), and expressed as following equations:

$$\begin{aligned} x &= [d_1(y_b^2 - y_b^3) + d_2(y_b^3 - y_b^1) + d_3(y_b^1 - y_b^2)] / 2[x_b^1(y_b^2 - y_b^3) + x_b^2(y_b^3 - y_b^1) + x_b^3(y_b^1 - y_b^2)] \\ y &= [d_1(x_b^3 - x_b^2) + d_2(x_b^2 - x_b^1) + d_3(x_b^1 - x_b^3)] / 2[x_b^1(y_b^2 - y_b^3) + x_b^2(y_b^3 - y_b^1) + x_b^3(y_b^1 - y_b^2)] \end{aligned} \quad (33)$$

Furthermore, the passive joint angles corresponding to the coordinates of the end-effector can be calculated as follows:

$$\theta_b^i = \arctan((y - y_b^i) / (x - x_b^i)) \in (-\pi, \pi], \quad i = 1, 2, 3 \quad (34)$$

Besides the 6 link lengths  $l_a^i, l_b^i, i = 1,2,3$  and the 6 base coordinates  $(x_a^i, y_a^i), i = 1,2,3$ , 3 more parameters are included in the kinematics to compensate the undetermined bias angles between the actual zero positions of the joint sensors and the predefined zero positions of the joint angles. Here, home position of the end-effector (216.5,250) is defined as zero positions. Denote the bias angles by  $\Delta\theta_a^i, i = 1,2,3$ , and the readings of the encoders by  $\tilde{\theta}_a^i, i = 1,2,3$ , then one can express the active joint angles as follows:

$$\theta_a^i = \tilde{\theta}_a^i + \Delta\theta_a^i, i = 1,2,3 \quad (35)$$

Among the 6 base coordinates, 3 of them must be set to their nominal values to establish the coordinate frame before calibration. If not, the manipulator can move freely in the plane, and infinite solutions can be obtained through calibration and from these solutions, it is impossible to tell which solution is the actual one. With 3 coordinates being predefined, there are altogether 12 kinematic parameters to be calibrated. Without losing generality, we would suppose that the base coordinates  $x_a^1, y_a^1, y_a^2$  are equal to their nominal values, and regard them as constants for the calibration of GPM2002. Thus the kinematic parameters to be calibrated for GPM2002 include 3 sensor zero positions  $\Delta\theta_a^i, i = 1,2,3$ , 6 link lengths  $l_a^i, l_b^i, i = 1,2,3$  and 3 base coordinates  $x_a^2, x_a^3, y_a^3$ . For the calibration of the parallel manipulator, the kinematic parameter errors can be represented by the closed-loop constrained equations (Yiu et al., 2003c):

$$E_1 = \begin{bmatrix} x - x_a^1 - l_a^1 \cdot \cos(\tilde{\theta}_a^1 + \Delta\theta_a^1) - l_b^1 \cdot \cos(\theta_b^1) \\ y - y_a^1 - l_a^1 \cdot \sin(\tilde{\theta}_a^1 + \Delta\theta_a^1) - l_b^1 \cdot \sin(\theta_b^1) \\ x - x_a^2 - l_a^2 \cdot \cos(\tilde{\theta}_a^2 + \Delta\theta_a^2) - l_b^2 \cdot \cos(\theta_b^2) \\ y - y_a^2 - l_a^2 \cdot \sin(\tilde{\theta}_a^2 + \Delta\theta_a^2) - l_b^2 \cdot \sin(\theta_b^2) \\ x - x_a^3 - l_a^3 \cdot \cos(\tilde{\theta}_a^3 + \Delta\theta_a^3) - l_b^3 \cdot \cos(\theta_b^3) \\ y - y_a^3 - l_a^3 \cdot \sin(\tilde{\theta}_a^3 + \Delta\theta_a^3) - l_b^3 \cdot \sin(\theta_b^3) \end{bmatrix} = 0 \quad (36)$$

Then for  $n$  sampled configurations  $\tilde{\theta}_{aj}^1, \tilde{\theta}_{aj}^2, \tilde{\theta}_{aj}^3, j = 1, \dots, n$ ,  $6n$  equations and  $5n+12$  variables can be obtained based on the closed-loop constrained equation (34). Among the variables,  $2n$  variables are end-effector coordinates  $(x_j, y_j), j = 1, \dots, n$ ,  $3n$  variables are passive joint angles  $\theta_{bj}^1, \theta_{bj}^2, \theta_{bj}^3, j = 1, \dots, n$ , and the remaining 12 variables are kinematic parameters to be calibrated. Obviously, if 12 sampled configurations are chosen, then 72 equations and 72 variables can be obtained, and the kinematic parameters can be calculated by solving the equations. If more configurations are sampled, then the number of the equations will exceeds the number of the variables, and the parallel manipulator can be calibrated by minimizing the norm of the vector  $E_1$  corresponding to the sampled configurations. For example, for  $n$  sampled configurations, one can implement the calibration by minimizing the following function  $J_1$  (Yiu et al., 2003c):

$$J_1 = \sum_{j=1}^n E_{1j}^T E_{1j} = \sum_{j=1}^n \|E_{1j}\|^2 \quad (37)$$

By eliminating the items involving passive joint angles, the closed-loop constrained equations can also be expressed as follows:

$$E_2 = \begin{bmatrix} \left( x - x_a^1 - l_a^1 \cdot \cos(\tilde{\theta}_a^1 + \Delta\theta_a^1) \right)^2 + \left( y - y_a^1 - l_a^1 \cdot \sin(\tilde{\theta}_a^1 + \Delta\theta_a^1) \right)^2 - (l_b^1)^2 \\ \left( x - x_a^2 - l_a^2 \cdot \cos(\tilde{\theta}_a^2 + \Delta\theta_a^2) \right)^2 + \left( y - y_a^2 - l_a^2 \cdot \sin(\tilde{\theta}_a^2 + \Delta\theta_a^2) \right)^2 - (l_b^2)^2 \\ \left( x - x_a^3 - l_a^3 \cdot \cos(\tilde{\theta}_a^3 + \Delta\theta_a^3) \right)^2 + \left( y - y_a^3 - l_a^3 \cdot \sin(\tilde{\theta}_a^3 + \Delta\theta_a^3) \right)^2 - (l_b^3)^2 \end{bmatrix} = 0 \quad (38)$$

Based on (38),  $3n$  equations and  $2n+12$  variable can be obtained with  $n$  sampled configurations.  $2n$  variables are end-effector coordinates  $(x^j, y^j)$ ,  $j = 1, \dots, n$ , and the remaining other 12 variables are kinematic parameters of the parallel manipulator. And also, with enough sampled configurations, one can calibrate the unknown kinematic parameters of the parallel manipulator by minimizing the following function  $J_2$ :

$$J_2 = \sum_{j=1}^n E_{2j}^T E_{2j} = \sum_{j=1}^n \|E_{2j}\|^2 \quad (39)$$

Therefore, the calibration problem can be converted into a minimization problem, in which either error function  $J_1$  in (37) or our new proposed the error function  $J_2$  in (39) can be used as the error function.

### 3.2 Calibration procedure based on the new error function

The calibration procedure based on the minimization of the error function  $J_2$  proposed in (39) are as follows:

1. Choose the kinematic parameters to be calibrated from the set of the 12 kinematic parameters mentioned above. Evaluate other kinematic parameters by other means and take them as constants for the calibration. Then choose  $n$  sampled configurations of the parallel manipulator and record the readings of the encoders  $\tilde{\theta}_{aj}^1, \tilde{\theta}_{aj}^2, \tilde{\theta}_{aj}^3, j = 1, \dots, n$ . Make sure that the number of the sampled configurations exceeds the number of calibrated kinematic parameters.
2. Choose  $J_2$  in (39) as error function.
3. Choose suitable initial values for variables. For the geometric parameters, the nominal value of the parameter can be used as the initial value. For the sensor zero positions, suitable initial value can be obtained through following procedures: move the end-effector manually to a reference point, e.g. the home position (216.5, 250), and record the readings of the absolute encoder  $\tilde{\theta}_{a0}^i, i = 1, 2, 3$ , then the initial value of the sensor zero positions can be calculated through equation  $\Delta\hat{\theta}_a^i = \theta_{a0}^i - \tilde{\theta}_{a0}^i$ . Here symbols  $\theta_{a0}^i, i = 1, 2, 3$  refer to the active joint angles of the reference point, which can be calculated through inverse kinematic transformation.
4. Calculate the estimations of joint angles and end-effector coordinates of the sampled configurations. With the estimated value of the sensor zero positions, the estimated active joint angles can be calculated through  $\hat{\theta}_{aj}^i = \tilde{\theta}_{aj}^i + \Delta\hat{\theta}_{aj}^i, i = 1, 2, 3, j = 1, \dots, n$ . Then the



estimations of the end-effector coordinates  $\hat{x}_j, \hat{y}_j, j=1, \dots, n$  and the estimations of the passive joint angles  $\hat{\theta}_{bj}^i, i=1,2,3, j=1, \dots, n$  can be calculated through (33) and (34) respectively.

5. Solve the minimizing problem with the initial values of the variables obtained in step 3. For the minimization, the estimations of joint angles and end-effector coordinates can be calculated as in step 4, and the values of the variables corresponding to the minimal value of the error function can be regarded as calibration results.

### 3.3 Experimental results

To verify the validity of the calibration procedure and the error function proposed in subsection 3.1, simulation experiments are implemented in this subsection. In the experiments, a predefined 'actual value' is set for every kinematic parameter, and the encoder readings of the sampled configurations are calculated from the sampled end-effector coordinates with these 'actual values'. Then according to the calibration procedure proposed in 3.2, we can calibrate the parameters involved. We implement the simulation experiments by Matlab™ program, and adopt its optimizing function 'fmincon' to solve the problems. Furthermore, we adopt a stepwise strategy for the experiments, and in each step we calibrate only a part of the kinematic parameters with the assumption that the remaining other parameters have been known accurately. Then by decreasing the number of parameters supposed to be known and increasing the number of parameters to be calibrated step by step, we try to calibrate as many parameters as possible with the calibration procedure. For the purpose of the calibration accuracy comparison, we do each experiment by using of both error functions  $J_1$  and  $J_2$ .

The results of experiments can be examined by two means. The first one is to compare the calibrated results of the kinematic parameters with the predefined 'actual value'. The other one is to compare the calibrated end-effector coordinates  $(\hat{x}_j, \hat{y}_j), j=1, \dots, n$  with the 'actual end-effector coordinates'  $(x_j, y_j), j=1, \dots, n$  of the sampled configurations, through the following 'kinematics model root mean square error' (Yiu et al., 2003c):

$$krmrse = \sqrt{\frac{1}{n} \sum_{j=1}^n \left( (x_j - \hat{x}_j)^2 + (y_j - \hat{y}_j)^2 \right)} \quad (40)$$

#### A. Calibration of Sensor Zero Positions

Suppose that all of the link lengths and the base coordinates have been measured accurately, and take the 3 unknown sensor zero positions  $\Delta\theta_a^i, i=1,2,3$  as variables, we can calibrate the variables by solving these problems:

$$\min J_1(\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3)$$

$$\min J_2(\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3)$$

The point (210,245) is taken as the estimation of the home point, and the initial estimations of  $\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3$  are calculated by subtracting the encoder readings corresponding to point (210,245) from the active joint angles corresponding to the home point (216.5,250). 3 sampled configurations and 9 sampled configurations are chosen respectively for the simulations. Results are shown in Table 2, from which we can see that the precision of calibration results

can be improved by increasing the number of the sampled configurations, as well as that more precise calibration results can be obtained by adopting  $J_2$  as the error function.

Calibrated variable	Actual value	Initial estimation	Results calibrated with $J_1$		Results calibrated with $J_2$	
			3 samples	9 samples	3 samples	9 samples
$\Delta\theta_a^1 (rad)$	1.00000	1.00913	1.00078	1.00000	0.99999	1.00000
$\Delta\theta_a^2 (rad)$	1.00000	0.97689	0.99948	1.00000	1.00001	1.00000
$\Delta\theta_a^3 (rad)$	1.00000	1.02748	0.99976	1.00000	1.00000	1.00000
$kmrmse$ of samples(mm)			1.47188e-1	7.85886e-4	1.49488e-3	1.26624e-5
Number of iterations			12	9	14	13

Table 2. Calibration results of sensor zero positions

B. Calibration of Sensor Zero Positions and Link Lengths

In this subsection, only the base coordinates are supposed to be known, while all of the sensor zero positions and the link lengths are chosen as variables. Then the calibration problem can be converted into the following problems:

$$\min J_1(\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3, l_a^1, l_b^1, l_a^2, l_b^2, l_a^3, l_b^3)$$
$$\min J_2(\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3, l_a^1, l_b^1, l_a^2, l_b^2, l_a^3, l_b^3)$$

As we'll demonstrate that error function  $J_2$  is easier than  $J_1$  and more precise results can be obtained through minimizing  $J_2$ . Before we move on, take a look at  $J_2$ , one can find that there are 2 product items of the variables in  $J_2$ , which can be expressed as follows:

$$l_a^i \bullet \cos(\tilde{\theta}_{aj}^i + \Delta\theta_a^i) = (l_a^i \bullet \cos(\Delta\theta_a^i)) \bullet \cos(\tilde{\theta}_{aj}^i) - (l_a^i \bullet \sin(\Delta\theta_a^i)) \bullet \sin(\tilde{\theta}_{aj}^i)$$
$$l_a^i \bullet \sin(\tilde{\theta}_{ai}^j + \Delta\theta_a^i) = (l_a^i \bullet \sin(\Delta\theta_a^i)) \bullet \cos(\tilde{\theta}_{aj}^i) + (l_a^i \bullet \cos(\Delta\theta_a^i)) \bullet \sin(\tilde{\theta}_{aj}^i)$$

(41)

in which the product items  $l_a^i \bullet \cos(\tilde{\theta}_{aj}^i + \Delta\theta_a^i)$  and  $l_a^i \bullet \sin(\tilde{\theta}_{ai}^j + \Delta\theta_a^i)$  are decoupled into linear combinations of  $l_a^i \bullet \cos(\Delta\theta_a^i)$  and  $l_a^i \bullet \sin(\Delta\theta_a^i)$ . So the product items in the error function can be eliminated by choosing  $l_a^i \bullet \cos(\Delta\theta_a^i)$  and  $l_a^i \bullet \sin(\Delta\theta_a^i)$  as calibrated variables. Let  $l_{ac}^i = l_a^i \bullet \cos(\Delta\theta_a^i)$ ,  $l_{as}^i = l_a^i \bullet \sin(\Delta\theta_a^i)$ ,  $i = 1, 2, 3$ , then we have:

$$E_3 = \left[ \begin{array}{l} \left( x - x_a^1 - \left( l_{ac}^1 \cdot \cos(\tilde{\theta}_a^1) - l_{as}^1 \cdot \sin(\tilde{\theta}_a^1) \right) \right)^2 + \dots \left( y - y_a^1 - \left( l_{ac}^1 \cdot \sin(\tilde{\theta}_a^1) + l_{as}^1 \cdot \cos(\tilde{\theta}_a^1) \right) \right)^2 - (l_b^1)^2 \\ \left( x - x_a^2 - \left( l_{ac}^2 \cdot \cos(\tilde{\theta}_a^2) - l_{as}^2 \cdot \sin(\tilde{\theta}_a^2) \right) \right)^2 + \dots \left( y - y_a^2 - \left( l_{ac}^2 \cdot \sin(\tilde{\theta}_a^2) + l_{as}^2 \cdot \cos(\tilde{\theta}_a^2) \right) \right)^2 - (l_b^2)^2 \\ \left( x - x_a^3 - \left( l_{ac}^3 \cdot \cos(\tilde{\theta}_a^3) - l_{as}^3 \cdot \sin(\tilde{\theta}_a^3) \right) \right)^2 + \dots \left( y - y_a^3 - \left( l_{ac}^3 \cdot \sin(\tilde{\theta}_a^3) + l_{as}^3 \cdot \cos(\tilde{\theta}_a^3) \right) \right)^2 - (l_b^3)^2 \end{array} \right] = 0$$

(42)

$$J_3 = \sum_{j=1}^n E_{3j}^T E_{3j} = \sum_{j=1}^n \|E_{3j}\|^2$$

(43)

Thus we can also calibrate through another new error function that:

$$\min J_3(l_{ac}^1, l_{as}^1, l_{ac}^2, l_{as}^2, l_{ac}^3, l_{as}^3, l_b^1, l_b^2, l_b^3)$$

When the calibration results are obtained by minimizing  $J_3$ , we can calculate the parameters of the manipulator by following equations:

$$l_a^i = (l_{ac}^i 2 + l_{as}^i 2)^{0.5}$$
$$\Delta\theta_a^i = arctg(l_{as}^i / l_{ac}^i) \quad i = 1, 2, 3 \tag{44}$$

Now we have three error functions. We take the same initial estimations of  $\Delta\theta_a^1, \Delta\theta_a^2, \Delta\theta_a^3$  as used in the calibration of the sensor zero positions. The initial estimations for  $l_a^i, l_b^i, i = 1, 2, 3$  are taken as the nominal values. 21 sampled configurations are taken. The sensor zero positions and the link lengths of the parallel manipulator are calibrated by solving the minimizing problems. Results are shown in Table 3, from which one can find that, with the same sampled configurations and the same initial estimations of the kinematic parameters, the precision of the calibration results are improved by adopting  $l_{ac}^i, l_{as}^i, i = 1, 2, 3$  as calibrated variables. And also, with  $J_3$  as the error function, iterations cost gets lower to find the solution.

Calibrated variable	Actual value	Initial estimation	Results calibrated with function $J_1$	Results calibrated with function $J_2$	Results calibrated with function $J_3$
$\Delta\theta_a^1 (rad)$	1.00000	1.01052	0.99996	1.00001	1.00000
$\Delta\theta_a^2 (rad)$	1.00000	0.97614	1.00004	1.00000	1.00000
$\Delta\theta_a^3 (rad)$	1.00000	1.02659	1.00008	1.00001	1.00000
$l_a^1 (mm)$	244.1	244	244.08300	244.10012	244.09994
$l_a^2 (mm)$	244.2	244	244.19647	244.19818	244.19999
$l_a^3 (mm)$	243.5	244	244.49562	244.50038	244.49998
$l_b^1 (mm)$	243.8	244	243.78108	243.80393	243.79992
$l_b^2 (mm)$	244.2	244	244.21085	244.19883	244.20004
$l_b^3 (mm)$	244.2	244	244.22065	244.20262	244.20006
<i>krmrse</i> of samples(mm)			3.21669e-3	1.56383e-3	1.33563e-5
Number of iterations			46	47	33

Table 3. Calibration results of sensor zero positions and link lengths

C. Calibration of Sensor Zero Positions, Link Lengths and Base Coordinates

The sensor zero positions and link lengths have been calibrated successfully in last subsection, next we will calibrate all of the kinematic parameters. Since the superiority of the error function  $J_3$  has been demonstrated in the last subsection, we use only  $J_3$  as the error function to calibrate the parameters here. Choose  $l_{ac}^i, l_{as}^i, l_b^i, i = 1, 2, 3$  and the base coordinates  $x_a^2, x_a^3, y_a^3$  as variables. Minimize the following function  $J_3$ :

$$\min J_3(l_{acr}^1, l_{asf}^1, l_{acr}^2, l_{asf}^2, l_{acr}^3, l_{asf}^3, l_b^1, l_b^2, l_b^3, x_a^2, x_a^3, y_a^3)$$

31 sample configurations are taken and calibration results are shown in Table 4, from which one can see that there are several millimeter errors between the calibrated results and the predefined ‘actual values’ of the kinematic parameters. Further research reveals that, if we assume that the base coordinate  $x_a^2$  is known accurately before calibration, precise calibration results still can be obtained for the other 11 kinematic parameters by solving the problem:

$$\min J_3(l_{acr}^1, l_{asf}^1, l_{acr}^2, l_{asf}^2, l_{acr}^3, l_{asf}^3, l_b^1, l_b^2, l_b^3, x_a^3, y_a^3)$$

The same sampled configurations and the same initial estimations of the kinematic parameters are employed with that of the 12 parameter experiment, except that the base coordinates  $x_a^2$  is supposed to be known beforehand. Calibration results are shown in last column of the Table 4, from which one can see that all 11 kinematic parameters, including 3 sensor zero positions and 8 geometric parameters, can be calibrated accurately.

Calibrated variable	Actual value	Initial estimation	12 values calibrated with $J_3$	11values calibrated with $J_3$
$\Delta\theta_a^1$ (rad)	1.00000	1.01052	1.0072	1.00000
$\Delta\theta_a^2$ (rad)	1.00000	0.97612	1.0072	1.00000
$\Delta\theta_a^3$ (rad)	1.00000	1.02678	1.0072	1.00000
$l_a^1$ (mm)	244.1	244	244.0998	244.0999
$l_a^2$ (mm)	244.2	244	247.1998	244.2000
$l_a^3$ (mm)	243.5	244	247.3005	244.4999
$l_b^1$ (mm)	243.8	244	246.8959	243.7999
$l_b^2$ (mm)	244.2	244	244.3009	244.1999
$l_b^3$ (mm)	244.2	244	244.3005	244.2001
$x_a^3$ (mm)	433.05	433	436.7084	433.0502
$y_a^3$ (mm)	499.96	500	506.2960	499.9598
$x_a^2$ (mm)	433.04	433	440.3557	
$kmrmse$ of samples(mm)		3.3521		1.0211e-4
Number of iterations		34		70

Table 4. Calibration results of 12 parameters

4. Complete kinematic parameters auto-calibration using stochastic optimization algorithms

We have already selected and defined some error functions in the former section and optimized them using Matlab™ function. But, as we know, the Matlab™ function we used is a local optimization method. As a real-world optimization problem, the corresponding

error functions are very complex both on the number of variables and their multimodal features, thus it is very hard to converge to a global minimum using non-global optimization methods. According to No Free Lunch Theorems (NFLT, Wolper & Macready, 1997), no algorithm is perfect on all the problems. Although there have been a lot of tests on benchmark functions to evaluate different algorithms on different performances, for a specific optimization problem, an optimization algorithm has to be chosen appropriately to the structure of the problem itself.

There are mainly two classes of optimization algorithms. The first ones are the deterministic algorithms, including gradient-based algorithms *etc.* In this class, one must have some information about the objective function, such as gradients, and this information is used to determine the search direction in a deterministic manner at every step of the algorithm. If a problem is linear or nonlinear but convex, deterministic algorithms are readily applied to solve the problem and can perform very well. But, generally speaking, real-world problems are hardly such easy class. Most real-world problems are nonlinear, non-convex, multi-dimensional and have a lot of local minima. For these real-world problems, deterministic algorithms are inappropriate or bear very poor performances, because the objective function information is not available in many cases or the algorithms run very big risk to be trapped in local minimum and cannot escape. Due to these drawbacks, the use of deterministic algorithms in real-world applications is very limited. To address the problem of convergence to local optima, stochastic optimization algorithms are proposed and have been playing a rapidly growing role in the past few decades. Different from deterministic optimization algorithms, stochastic optimization algorithms deliberately introduce randomness into the search process and inherently accept weak candidate solutions, thus the search process could escape from local optima to local the global optimum. Moreover, the algorithms are less sensitive to noises and modeling errors. These algorithms mainly include genetic algorithms (GA, Holland, 1975; Goldberg, 1989), differential evolution (DE, Storn & Price, 1995) and particle swarm optimization (PSO, Kennedy & Eberhart, 1995) *etc.* Most of these algorithms are also inherently parallel, which makes the algorithms more efficient in searching for global solutions.

As for our calibration problem, it is not hard to find that the error functions in (37), (39) and (43) mentioned above are in continuous spaces, nonlinear, non-convex, multi-dimensional and have a lot of local minima. GA, PSO and DE are then the very natural choices. We designed auto-calibration based on GA, PSO and DE for simulation experiments and actual system calibration.

#### 4.1 Auto-calibration based on GA

GA is a population-based optimization algorithm, in which a candidate solution is called an individual and individuals constitute population. The quality of the individuals is termed as their fitness, the higher quality an individual has, the higher fitness it owns. The individuals evolve mainly through reproduction, crossover and mutation operations. In our auto-calibration work based on GA, individuals are represented as binary string. If the search range of a parameter is  $[X_{\min}, X_{\max}]$  and the precision requirement is  $p$ , then the length of the string to present this parameter is  $L = \lceil \log_2((X_{\max} - X_{\min})/p) \rceil$ . The individual length is the sum of the 12 parameters' string length. The population size is  $N$ . The initial population is



initialized randomly. Then by decoding, the real values of the parameters to be calibrated are obtained. Use the expression of the base coordinates (30) and the equations (32) and (33) to estimate the positions of the end-effector. When the positions of the end-effector are estimated, using error functions  $J_1$ ,  $J_2$  or  $J_3$  given in (37), (39) or (43) as the fitness functions, respectively, we can evaluate the fitness of the individuals. In the reproduction stage, we use tournament selection (Goldberg & Deb, 1991), which can make a lower selection pressure compared with roulette wheel selection. Copy the winner of the tournaments into the population. After reproduction, crossover operation is to be carried out with probability  $PCrossover$ . In our experiments, we apply single bit crossover, in which one bit is selected as the crossover point. With probability  $PMutation$ , we employ two bits mutation, in which two bits are mutated to produce a new individual. The process of reproduction, crossover and mutation iterates until the stop criteria that is defined later. In order to maintain the diversity of the population, half of the individuals are reinitialized when the population fitness does not improve for a number of generations  $NR$ . Both in the reproduction and reinitializing, the best individual is maintained.

#### 4.2 Auto-calibration based on PSO

PSO is a new population-based optimization algorithm. Different from GA, searching is carried out straightforward in the search space of PSO and no genetic operation is needed. Every solution in the PSO algorithm is called a particle. A particle has its location and flying velocity. Define a particle's location and velocity of the  $k$ th iteration as  $X^k$  and  $V^k$ , and the state of this particle in the next iteration can be calculated as follows:

$$\begin{aligned} V^{k+1} &= w^k V^k + C_1 R_1 (P_{best} - X^k) + C_2 R_2 (G_{best} - X^k) \\ X^{k+1} &= X^k + V^{k+1} \end{aligned} \quad (45)$$

in which  $w$  is inertia weight (Shi & Eberhart, 1998),  $C_1$  and  $C_2$  are predefined acceleration constants,  $R_1$  and  $R_2$  are random numbers generated in the range of  $[0,1]$ .  $P_{best}$  is the best location obtained ever by the particle itself, and  $G_{best}$  is the best location ever detected by the whole population.

In auto-calibration experiments based on PSO, suppose the population size is  $N$  and the  $N$  particles fly in a 12 dimensional search space. The location of the particle is represented as  $X=(x_1, x_2, \dots, x_{12})$ , corresponding to the solution of the 12 parameters. The velocity of the particles is represented as  $V=(v_1, v_2, \dots, v_{12})$ , corresponding to the flying over distance of the particle. The locations and velocities of the particles are initialized randomly. In the iterations, estimate the positions of the end-effector and evaluate the quality of the particles in the same way as it does in the auto-calibration based on GA. If a particle beats its  $P_{best}$ , update its  $P_{best}$ . And if it beats  $G_{best}$ , then update  $G_{best}$  correspondingly. Then according to (45), update every particle's location and velocity. In order to level up the search efficiency, search space  $[X_{min}, X_{max}]$  is set to constrain the particles' movement. If a particle outpaces the border, it is placed on the border of the space. Also, in order to maintain the diversity of the population, half of the particles are reinitialized except the best when the population does not improve for  $NR$  generations.



### 4.3 Auto-calibration based on DE

DE is a new population-based optimization algorithm. Its operations are the same as GA in name, but very different in nature. The concepts of individual and population are the same as those of GA. In an  $n$  variable problem, the individual in DE is represented by a vector  $X=(x_1, x_2, \dots, x_n)$ . And the population for each generation  $G$  can be represented as  $X_{i,G}$ ,  $i = 1, 2, \dots, N$ , in which  $N$  is the population size. In mutation,

$$V_{i,G+1} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}) \quad (46)$$

where there are random indexes  $r1, r2, r3 \in \{j \mid j \neq i, j \in [1, N]\}$  and  $F \in [0, 2]$ . And in crossover,

$$U_{i,G+1} = (U_{1i,G+1}, U_{2i,G+1}, \dots, U_{ni,G+1})$$

$$U_{ji,G+1} = \begin{cases} V_{ji,G+1} & \text{if } (randb(j) \leq CR) \text{ or } j = rnbr(i) \\ X_{ji,G} & \text{if } (randb(j) > CR) \text{ or } j \neq rnbr(i) \end{cases} \quad j = 1, 2, \dots, n \quad i = 1, 2, \dots, N \quad (47)$$

in which  $randb(j) \in [0, 1]$  is the  $j$ th evaluation of a norm random number.  $CR \in [0, 1]$  is the crossover constant set by the user.  $Rnbr(i)$  is a randomly chosen index from  $n$  dimensions to ensure that at least one dimension parameter from  $V_{i,G+1}$  can be attained by  $U_{i,G+1}$ . In selection stage,

$$X_{i,G+1} = \begin{cases} U_{i,G+1}, & \text{if } U_{i,G+1} \text{ is better than } X_{i,G} \\ X_{i,G}, & \text{otherwise} \end{cases} \quad (48)$$

In our simulation based on DE,  $n=12$ . The initial population is randomly generated. In every generation, according to (46), we employ mutation. And then according to (47), crossover operation is implemented. After mutation and crossover, we check the individuals whether they are still in the search range of  $[X_{min}, X_{max}]$ . If any parameter is out of this range, this parameter is randomly regenerated. When this checking is finished, in the same way as the evaluation of individuals in GA and PSO, we evaluate  $X_{i,G}$  and  $U_{i,G+1}$ . Based on the quality of  $X_{i,G}$  and  $U_{i,G+1}$ , selection operation is carried out according to (48) and the individuals that are selected constitute the new population of next generation.

### 4.4 Simulation experiments

In the simulation experiments, we supposed that the base coordinates  $x_a^1, y_a^1, x_a^2$  are equal to their nominal values. We sampled 50 configurations arbitrarily in the workspace of the manipulator, and recorded the encoder readings as  $\tilde{\theta}_{aj}^i, j = 1, 2, 3 \quad i = 1, 2, \dots, 50$ .

The values of the kinematic parameters, including their nominal values provided by the producer, 'actual values' we predefined, and their ranges in the search space are set as in Table.5.

The control variable settings for each algorithm are described as follows.

In GA, we define population  $N=100$ , precision requirement  $p=1.0e-4$ , crossover probability  $PCrossover=0.85$  and mutation probability  $PMutation=0.15$ , tournament scale is 4. Reinitialize half individuals when the error function value keeps still over  $NR=50$  iterations. For PSO, we define population  $N=100$ ,  $C_1=C_2=2$  and the inertia weight  $w$  decreased from 0.9 to 0.1

with the iterations. Reinitialize half particles when the error function value keeps still over  $NR=50$  iterations. In DE,  $N=150$  ,  $F=0.5$  and  $CR=0.8$  .

Our main task in simulation experiments is to test the convergence performances of each algorithm under error functions  $J_1, J_2$  and  $J_3$ . Since the population of DE is not equal to that of GA and PSO, and also there are differences in the nature of algorithms, it is meaningless to compare the convergence performances with the generations. Instead, we can compare their convergences with the evaluations of the error functions. Thus, in order to make this uniform comparing criterion, we define that search process of each algorithm stops when the error functions evaluation times reach  $5.0e7$ . The simulation experimental results are shown in Fig.3, Fig.4 and Fig.5, respectively. For clarity, the errors are presented as  $\log_{10}(J_i)$ ,  $i = 1,2,3$ , and this error is defined as the convergence performance. One can see from Fig.3 that under function  $J_1$ , GA performs worst. It converges to  $1.3477e3$ . DE performs best. It converges to  $2.5351e-3$  and its converging speed is also very fast. Between them is PSO, which converges to  $2.6869e2$ . From Fig.4 one can see that due to using error function  $J_2$ , the errors get lower compared with  $J_1$ . DE is still the best one. It converges to  $5.5626e-14$  at a fast speed. With the product items decoupled, the search process would become easy, which is also verified by simulation results shown in Fig. 5, from which one can see that under  $J_3$ , the search results improve to different extents for different algorithms. Among them, DE improves most. By means of DE,  $J_3$  reaches  $3.5946e-20$  in Fig. 4, while in Fig. 4,  $J_2$  only gets to  $5.5626e-14$ .

	$y_a^2 (mm)$	$x_a^3 (mm)$	$y_a^3 (mm)$	$l_a^1 (mm)$	$l_a^2 (mm)$	$l_a^3 (mm)$
Nominal Values	0	433	500	244	244	244
Actual Values	0.3	433.5	499.4	244.1	244.2	243.5
$X_{min}$	-5	428	495	239	239	239
$X_{max}$	5	438	505	249	249	249
	$l_b^1 (mm)$	$l_b^2 (mm)$	$l_b^3 (mm)$	$\Delta\theta_a^1 (rad)$	$\Delta\theta_a^2 (rad)$	$\Delta\theta_a^3 (rad)$
Nominal Values	244	244	244	0	0	0
Actual Values	243.8	244.2	244.6	0.01	-0.01	0.01
$X_{min}$	239	239	239	-0.2	-0.2	-0.2
$X_{max}$	249	249	249	0.2	0.2	0.2

Table 5. Value settings for the simulation experiments

All the results of the auto-calibration simulation experiments can also be seen in Table 6.

	GA	PSO	DE
$J_1$	1.3447e3	2.6869e2	2.5351e-3
$J_2$	8.9451e2	1.0079e2	5.5626e-14
$J_3$	1.0344e2	8.9561e0	3.5946e-20

Table 6. All the results of the simulation experiments

From the simulation experimental results given above, we can obtain the conclusions that: 1) Error function  $J_2$  is simpler than  $J_1$  by eliminating the items involving passive joint angles, and Error function  $J_3$  is simpler than  $J_2$  owing to decoupling the products items; 2) The DE has the best performances under all the three error functions both in convergence accuracy and speed. The algorithms on this calibration problem go from worst to best is: GA, PSO and DE, and 3) Since DE converges to 3.5946e-20 under  $J_3$ , very close to zero, the solution might be very close to the ‘actual values’.

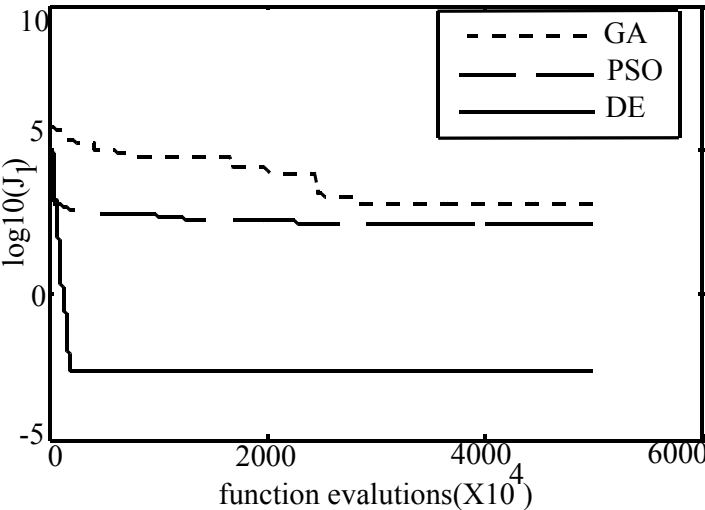


Fig. 3. The performances of the algorithms under function  $J_1$

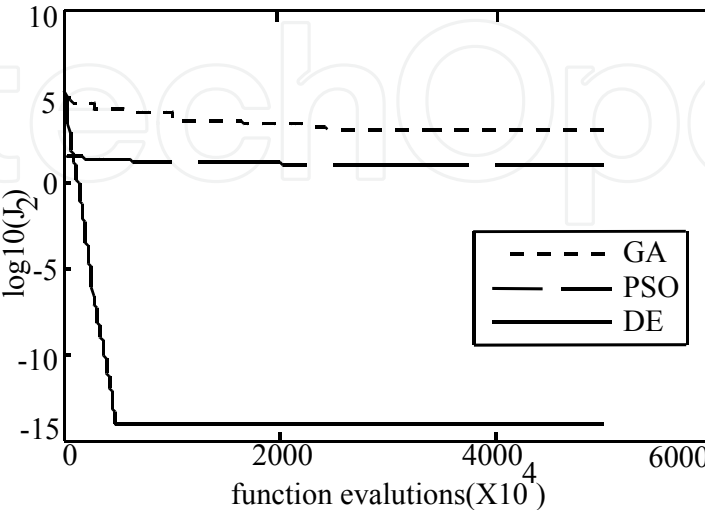


Fig. 4. The performances of the algorithms under function  $J_2$

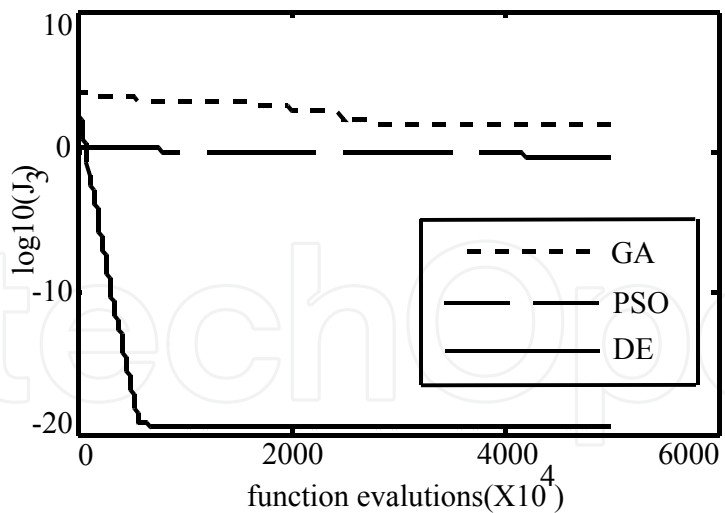


Fig.5 The performances of the algorithms under function  $J_3$

In order to find whether DE's calibration solution under  $J_3$  is really close to the 'actual values', we compare the solution's values to the 'actual values' and calculate the *kmrmse* of the solution, which is presented in Table 7.

parameters	actual values	DE calibration solution
$y_a^2 (mm)$	0.3000	0.3076
$x_a^3 (mm)$	433.5000	433.1470
$y_a^3 (mm)$	499.4000	499.6045
$l_a^1 (mm)$	244.1000	244.0004
$l_a^2 (mm)$	244.2000	244.1004
$l_a^3 (mm)$	243.5000	243.4006
$l_b^1 (mm)$	243.8000	243.7005
$l_b^2 (mm)$	244.2000	244.1004
$l_b^3 (mm)$	244.6000	244.5002
$\Delta\theta_a^1 (rad)$	0.01000	0.0107
$\Delta\theta_a^2 (rad)$	-0.01000	-0.0092
$\Delta\theta_a^3 (rad)$	0.01000	0.0107
<i>kmrmse</i> of samples(mm)		0.0371

Table 7. Calibration results of DE under  $J_3$

One can see from Table 7 that the largest error between DE calibration solution and 'actual values' is 0.3530mm, most of the errors are less than 0.1mm, and the errors of the three sensor zero positions are all less than 0.001rad. Also, the *kmrmse* of the solution is only 0.0371mm, demonstrating that the calibration solution is very close to the 'actual values' indeed.

4.5 Actual calibration

In the simulation experiments above, the results demonstrate that  $J_3$  is the best error function and DE is the most efficient algorithm for our calibration problem. Based on this conclusion, we applied DE calibration with error function  $J_3$  on actual parallel manipulator. The nominal parameter values of the manipulator have been mentioned before. The search ranges are the same as presented in Table 5.

First we sampled 50 configurations all in a line in the work space of the manipulator. But we came across great difficulties in calibrating a rational result. A rather long time had passed when we found that the problem was in the sampled configurations. If the sampled configurations are in a line, the movements of the links are not sufficient, some of the links move widely, but others move little. Thus not all the components of the manipulator are excited sufficiently. Besides, due to the inherent error in sensors, if the sampled configurations are in a line, the errors in the three sensors are not balanced. On considering these factors, we sampled 50 configurations in a circle around the geometrical center of the work space. By doing so, all the components of the manipulator can be excited sufficiently and the sensor errors can be balanced completely. The calibration results based on these sampled configurations proved that our analysis was right.

Using the 50 sampled configurations in a circle around the geometrical center of the work space and through DE method, whose parameter settings are the same as those in the simulation experiments, we obtained the all the 12 parameters. Because of the inevitable inaccuracy of the sensors, the error  $J_3$  in actual experiment cannot get down to the level of the simulation. After about  $1e6$  evaluations of the error function of  $J_3$ , the error does not improve any more. We stopped the optimization procedure then, and regarded the best solution found ever as the calibration results. The results are reported in Table 8.

parameters	nominal values	values through calibration
$y_a^2 (mm)$	0.0000	2.7571
$x_a^3 (mm)$	433.0000	436.2436
$y_a^3 (mm)$	500.0000	501.9123
$l_a^1 (mm)$	244.0000	243.6527
$l_a^2 (mm)$	244.0000	242.5634
$l_a^3 (mm)$	244.0000	242.4579
$l_b^1 (mm)$	244.0000	243.8194
$l_b^2 (mm)$	244.0000	243.4168
$l_b^3 (mm)$	244.0000	246.8952
$\Delta\theta_a^1 (rad)$	0.0000	6.2385e-3
$\Delta\theta_a^2 (rad)$	0.0000	1.0623e-2
$\Delta\theta_a^3 (rad)$	0.0000	2.8643e-3

Table 8. The results of the actual calibration

For the purpose of comparison, we calculated the  $J_3$  error under the two sets of parameters. Under the nominal values, the  $J_3$  error is  $2.1517e5$ , while under the calibrated values,  $J_3$  gets down to  $3.1841e2$ , which demonstrates that the accuracy of calibrated parameters is much higher than that of the nominal values.

## 5. Conclusion

In this chapter, we implemented the kinematic auto-calibration of a redundant planar 2-dof parallel manipulator. In this process, we first calibrated the error of the sensor zero positions by optimizing an projected tracking error function, and also the robustness of this method has been proved. Furthermore, in order to calibrate the other parameters of this parallel manipulator, we gave another error function based on the closed-loop constraint equations. By decoupling the product items in the error function, we simplified the optimization and more precise result was obtained. But, at most 11 out 12 parameters could be calibrated using only local optimization method. In order to calibrate all of the parameters, global optimization methods including GA, PSO and DE were applied. In simulation experiments, differential evolution was proved to be the most appropriate algorithm for the calibration problem. Finally, all the parameters of an real-world redundant planar 2-dof parallel manipulator were calibrated successfully by applying differential evolution to optimize the decoupled error function.

## 6. Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 50375148.

## 7. References

- Baron, L. & Angeles, J. (1998). The on-line direct kinematics of parallel manipulators under joint-sensor redundancy. *Proceedings of international symposium on advances in robot kinematics*, pp 126-137, Strobl, Austria, 1998.
- Cheng, H.; Yiu, Y.K. & Li, Z.X. (2003) Dynamics and Control of Redundantly Actuated Parallel Manipulators. *IEEE/ASME Transactions on Mechatronics*. Vol.8, No.4, pp.483-491.
- Chiu, Y.J. & Perng, M.H. (2004). M.H. Self-calibration of a general hexapod manipulator with enhanced precision in 5-DOF motions. *Mechanism and Machine Theory* Vol.39, No.1, pp.1-23.
- Glodberg, D.E.(1989). *Genetic Algorithms in Search. Optimization and Machine Learning*. Addison-Wesley Press.
- Goldberg, D. E. & Deb, K. (1991). A comparative analysis of selection schemes used in genetic algorithms. *Foundations of Genetic Algorithms*. pp. 69-93, San Mateo, CA: Morgan Kaufmann.
- Holland,J. (1975). *Adaptation in Nature and Artificial System*, The University of Michigan Press.



- Hollerbach, J.M. & Wampler, C.W. (1996). The calibration index and taxonomy for robot kinematic calibration methods. *International Journal of Robotics Research* Vol.15, No.6, pp.573-591.
- Jeong, W.J.; Kim, H.S. & Kwak, K.Y. (1999). Kinematics and workspace analysis of a parallel wire mechanism for measuring a robot pose. *Mechanism and Machine Theory*, Vol. 34, No. 6, pp. 825-841.
- Kennedy, J. & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference of Neural Networks*, pp.1942-1948, Perth, Australia, 1995.
- Koseki, Y.; Arai, T.; Sugimoto, K.; Takatuji, T. & Goto, M. (1998). Design and accuracy evaluation of high-speed and high-precision parallel mechanism. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1340-1345, Leuven, May, 1998.
- Khalil, W. & Besnard, S. (1999). Self calibration of stewart-gough parallel robots without extra sensors. *IEEE Transactions on Robotics and Automation* Vol.15, No.6, pp.1116-1121.
- Kock, S. & Schumacher, W. (2000a). Mixed elastic and rigid-body dynamic model of an actuation redundant parallel robot with high-reduction gears. *Proceedings of the International Conference on Robotics and Automation*, pp. 1918-1923, 2000.
- Kock, S. & Schumacher, W. (2000b) Control of a fast parallel robot with a redundant chain and gearboxes: experimental results. *Proceedings of the International Conference on Robotics and Automation*, pp. 1924-1929, 2000.
- Liu, G.F.; Cheng, H.; Xiong, Z.H.; Wu, X.Z.; Wu, Y.L. & Li, Z.X. (2001a). Distribution of singularity and optimal control of redundant parallel manipulators. *Proceedings of the International Conference on Intelligent Robots and Systems*, pp. 177-182, Maui, 2001.
- Liu, G.F.; Wu, Y.L.; Wu, X.Z.; Yiu, Y.K. & Li, Z.X. (2001b). Analysis and control of redundant parallel manipulators. *Proceedings of the International Conference on Robotics and Automation*, pp. 3748-3754, 2001
- Liu, G.F. & Li, Z.X. (2002) A unified geometric approach to modeling and control of constrained mechanical systems. *IEEE Transactions on Robotics and Automation*. Vol.18, No.4, pp.574-587.
- Liu, G.F.; Lou, Y. & Li, Z. (2003). Singularities of parallel manipulators: A geometric treatment. *IEEE Transactions on Robotics and Automation*. Vol.19, No.4, pp.579-594.
- Masory, O.; Wang, J. & Zhuang, H. (1993). On the accuracy of a stewart platform-Part II: kinematic calibration and compensation. *Proceedings of the IEEE international conference on robotics and automation*, pp 725-731, Atlanta, USA, 1993.
- Patel, A.J. & Ehmann, K.F. (2000). Calibration of a hexapod machine tool using a redundant leg. *International Journal of Maching Tools and Manufacture* Vol.40, No.4, pp.489-512.
- Renaud, P.; Andreff, N.; Lavest, J.M. & Dhome, M. (2006). Simplifying the kinematic calibration of parallel mechanisms using vision-based metrology. *IEEE Transaction on Robotics*, Vol. 22, No. 1, pp. 12-22.

- Shen, H.; Wu, X.Z; Liu, G.F & Li, Z.X. (2003). Hybrid position/force adaptive control of redundantly actuated parallel manipulators. *Zidonghua Xuebao/Acta Automatica Sinica*, Vol.29, No.4, pp.567-572.
- Shi, Y. & Eberhart, R. (1998). A Modified Particle Swarm Optimize. *Evolutionary Computation Proceedings 1998. IEEE World Congress on Computational Intelligence, the 1998 IEEE International Conference on*, pp.69-73.
- Storn.R. & Price.K. (1995). Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *Technical Report TR-95-012*, ICSI.
- Vincze, M.; Prenninger, J.P. & Gander, H. (1994). A laser tracking system to measure position and orientation of robot end-effectors under motion. *International Journal of Robotics Research*, Vol. 13, No. 4, pp. 305-314.
- Wampler, C.W.; Hollerbach, J.M. & Arai, T. (1995). An implicit loop method for kinematic calibration and its application to closed-chain mechanisms. *IEEE Transactions on Robotics and Automation* Vol.11, No.5, pp.710-724.
- Wang, J. & Masory, O. (1993). On the accuracy of a stewart platform-Part I, the effect of manufacturing tolerances. *Proceedings of the IEEE international conference on robotics and automation*, Atlanta, USA, pp.114 – 120, 1993.
- Wolpert, D.H. & Macready, W.G.. (1997). No Free Lunch Theorems for Optimization. *IEEE Transaction on Evolutionary Computation*, Vol.1, No.1, pp.67-82.
- Yiu, Y.K. & Li, Z.X. (2001). Dynamics of a Planar 2-dof Redundant Parallel Robot. *International Conference on Mechatronics Technology*, pp. 339-244, 2001
- Yiu, Y.K. & Li, Z.X. (2003a). PID and adaptive robust control of a 2-dof over-actuated parallel manipulator for tracking different trajectory. *International Symposium on Computational Intelligence in Robotics and Automation*, pp. 1052-1057, 2003.
- Yiu,Y.K. & Li, Z.X. (2003b). Optimal forward kinematics map for a parallel manipulator with sensor redundancy. *Proceedings of International Symposium on Computational Intelligence in Robotics and Automation*, pp. 354-359 ,2003.
- Yiu, Y.K.; Meng, J. & Li, Z.X. (2003c). Auto-calibration for a parallel manipulator with sensor redundancy. *Proceedings of the IEEE international conference on robotics and automation*, pp 3660-3665, Taipei, Taiwan, 2003.
- Zhang, Y.X. & Cong, S. (2005). Optimal motion control and simulation of redundantly actuated 2-dof planar parallel manipulator. *Journal of System Simulation*. Vol.17, No.10, pp.2450-2454.
- Zhang, L.J. (2006). Performance analysis and dimension optimization of 2-dof parallel manipulators, *Dissertation for the Doctoral Degree in Engineering*, Yanshan university.
- Zhuang, H.; Masory, O. & Yan, J. (1995). Kinematic calibration of a stewart platform using pose measurements obtained by a single theodolite. *Proceedings of the IEEE international conference on intelligent robots and systems*, pp 329-334, Pittsburgh, USA, (1995)
- Zhuang, H. (1997). Self-calibration of parallel mechanisms with a case study on stewart platforms. *IEEE Transaction on Robotics and Automation* Vol.13, No.3, pp.387-397.

- Zhuang, H.; Yan, J. & Masory, O. (1998). Calibration of stewart platforms and other parallel manipulators by minimizing inverse kinematic residuals. *Journal of Robotics and System* Vol.15, No.7, pp.395-405.
- Zou, H. & Notash, L. (2001). Discussions on the camera-aided calibration of parallel manipulators. *Proceedings of CCToMM Symposium on Mechanisms, Machines, and Mechatronics*, Montreal, Canada, 2001.



## **Parallel Manipulators, New Developments**

Edited by Jee-Hwan Ryu

ISBN 978-3-902613-20-2

Hard cover, 498 pages

**Publisher** I-Tech Education and Publishing

**Published online** 01, April, 2008

**Published in print edition** April, 2008

Parallel manipulators are characterized as having closed-loop kinematic chains. Compared to serial manipulators, which have open-ended structure, parallel manipulators have many advantages in terms of accuracy, rigidity and ability to manipulate heavy loads. Therefore, they have been getting many attentions in astronomy to flight simulators and especially in machine-tool industries. The aim of this book is to provide an overview of the state-of-art, to present new ideas, original results and practical experiences in parallel manipulators. This book mainly introduces advanced kinematic and dynamic analysis methods and cutting edge control technologies for parallel manipulators. Even though this book only contains several samples of research activities on parallel manipulators, I believe this book can give an idea to the reader about what has been done in the field recently, and what kind of open problems are in this area.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Shuang Cong, Chunshi Feng, Yaoxin Zhang, Zexiang Li and Shilon Jiang (2008). Kinematic Parameters Auto-Calibration of Redundant Planar 2-DOF Parallel Manipulator, Parallel Manipulators, New Developments, Jee-Hwan Ryu (Ed.), ISBN: 978-3-902613-20-2, InTech, Available from:

[http://www.intechopen.com/books/parallel\\_manipulators\\_new\\_developments/kinematic\\_parameters\\_auto-calibration\\_of\\_redundant\\_planar\\_2-dof\\_parallel\\_manipulator](http://www.intechopen.com/books/parallel_manipulators_new_developments/kinematic_parameters_auto-calibration_of_redundant_planar_2-dof_parallel_manipulator)

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](https://creativecommons.org/licenses/by-nc-sa/3.0/), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen