Robotica (2024), **42**, pp. 2589–2618
doi:10.1017/S0263574724000973

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Study of a global calibration method for a planar parallel robot mechanism considering joint error

Qinghua Zhang[1,2], Huaming Yu[1,2], Lingbo Xie[1,2], Qinghua Lu[1,2] and Weilin Chen[1,2]

[1]School of Mechatronics Engineering and Automation, Foshan University, Foshan, China
[2]Guangdong Provincial Key Laboratory of Industrial Intelligent Inspection Technology, Foshan University, Foshan, China
Corresponding author: Qinghua Lu; Email: qhlu@fosu.edu.cn

Received: 23 November 2023; **Revised:** 14 April 2024; **Accepted:** 9 May 2024; **First published online:** 16 September 2024

Keywords: planar parallel robot; positioning accuracy; inverse distance weighting; radial basis function neural network; global calibration method

Abstract

In order to improve the positioning accuracy of industrial robots, this paper proposes a global calibration method for planar parallel robot considering joint errors, which solves the problem that the existing calibration methods only consider part of the error sources and the calibration accuracy is poor, and improves the calibration efficiency and robot positioning accuracy. Consequently, it improves calibration efficiency and the overall precision of robot positioning. Firstly, the error model of overdetermined equations combined with structural parameters is established, and the global sensitivity of each error source is analyzed. Based on the measurement data of laser tracker, the local error source is identified by the least square method, which improves the local error accuracy by 88.6%. Then, a global error spatial interpolation method based on inverse distance weighting method is proposed, and the global accuracy is improved by 59.16%. Finally, the radial basis function neural network error prediction model with strong nonlinear approximation function is designed for global calibration, and the accuracy is improved by 63.05%. Experimental results verify the effectiveness of the proposed method. This study not only provides technical support for the engineering application of this experimental platform but also provides theoretical guidance for the improvement of the accuracy of related robot platforms.

1. Introduction

With the rapid development of smart manufacturing and robotics technology, robots have been widely used in high-precision manufacturing fields such as drilling, milling, and assembly. Parallel robots consist of a moving platform connected to a fixed base through multiple kinematic chains. Due to the closed-loop structure formed by these kinematic chains, parallel robots inherently exhibit higher stiffness, lower motion inertia, and increased load-bearing capacity compared to serial robots. Spatial parallel robots, featuring multiple kinematic pairs, offer additional degrees of freedom but may compromise overall speed and positional accuracy [1]. In contrast, planar parallel robot systems usually achieve higher velocity and positioning accuracy, which is sufficient for a variety of manipulation tasks in the plane. However, due to the existence of uncertain factors, such as the operating environment of the robot, machining, assembly, and so on, the positioning accuracy of the robot is not satisfactory [2]. Therefore, improving the positioning accuracy of robots has received more and more attention from scholars.

Currently, there are two main methods to enhance the accuracy of robots. One approach is to employ error prevention techniques during the robot's manufacturing phase. Ridha Kelaiaia et al. [3] have proposed a novel multi-objective function-based Parallel Manipulators (PMs) dimension optimization design method to enhance the accuracy of parallel robots. The other approach involves employing precision compensation techniques after the robot assembly is completed. In practical engineering applications, the latter method is more widely utilized [4]. Kinematic calibration is typically an economical

© The Author(s), 2024. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

and effective method [5], which primarily involves the following four steps: error modeling, error measurement, parameter identification, and error compensation [6]. Establishing an error model that satisfies completeness, continuity, and minimization is a critical step in kinematic calibration. However, it's important to note that error sources encompass both geometric and non-geometric errors. The former includes not only static errors (such as manufacturing and assembly errors) but also pitch errors of the kinematic pair and straightness errors. The latter includes errors caused by backlash, friction, flexible deformation, and thermal deformation [7]. It is found that joint errors of the robot have a great impact on the positioning accuracy. It is found that the geometric parameter error accounts for about 80% of the robot positioning error, nearly 20% of the robot positioning error is attributed to the deformation of robot components, and the joint error of the robot has a great impact on the positioning accuracy [8].

In recent decades, researchers have done a lot of work on robot calibration. To enhance the accuracy of parallel kinematic manipulator, Allaoua Brahmia et al. [9] proposed a novel dimensionless sensitivity analysis method to identify crucial geometric errors in both parallel and serial manipulators. Ye et al. [10] proposed an error forward propagation method and an error identification algorithm to separately solve the nonlinear error propagation and ill-conditioned identification equations. Simulation and experimental results of a 5-degree-of-freedom redundantly constrained hybrid robot indicate that the proposed methods can predict the end-effector position and pose under geometric errors. Luo et al. [11] introduced a novel forward kinematics solution based on dual quaternions and a modified error model based on dimensionless error mapping matrix, and an iterative identification procedure is developed. Experimental results indicate that the residual position and orientation errors are reduced by at least 97.67% and 86.85%, respectively, compared to the original values. Although the approach has a significant improvement in accuracy, its drawback lies in the complexity of the modeling and identification processes, large amount of calculation, and low efficiency. The fundamental principle of the kinematic model method is to obtain the kinematic parameter errors of the robot through a measurement and identification procedures, and then modify the kinematic model of the robot. Allaoua Brahmia et al. [12] proposed a novel method for improving robot precision by optimizing geometric parameter tolerances using an interior-point algorithm. Ye et al. [13] established a general error model using the product of exponentials formula and derived the geometric error constraint conditions. Li et al. [14], through internal force analysis, established a geometric error model incorporating deformations induced by generalized forces. The method was applied to the kinematic calibration of redundantly actuated parallel robots. However, due to the randomness of the actual geometric error, the assumption of error constraint cannot be satisfied, which directly leads to the inaccuracy of the error model.

In this case, non-kinematic model calibration methods have been proposed. Chen et al. [15] utilized a rigid-flexible coupling error model and a comprehensive measurement approach to achieve non-kinematic calibration for industrial robots; however, it can't compensate for the non-geometric error, and the compensation effect is poor. Zhu et al. [16] proposed a bilinear interpolation method based on measuring boundary point errors to estimate target point errors. The most popular method for compensating robot positioning accuracy is machine learning, which has the advantage of predicting future behavior, which has the advantage of prediction and has been widely used in different fields to improve the positioning accuracy of robots [17]. Liu [18] employed a position and pose error decomposition strategy and established an error prediction model based on backpropagation neural network and Denavit-Hartenberg method. The position and pose error of the entire workspace can be predicted by measuring position and pose data. Nguyen et al. [19] proposed a kinematic parameter calibration method based on an improved Manta Ray Foraging Optimization algorithm by combining a robot model-based recognition method and an artificial neural network based on error compensation technique to reduce the absolute positioning error of the manipulator. However, the method heavily relies on the structure of the model and the data samples, usually requiring a substantial amount of data for training the prediction model, which can reduce the efficiency of calibration.

In summary, many researchers have carried out research on improving positioning accuracy of robots, but there are still two prominent shortcomings. Firstly, existing calibration methods are inefficient and calibration methods based on numerical analysis heavily rely on error models; however, the high computational complexity of the kinematic error model method makes the calibration technique of

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

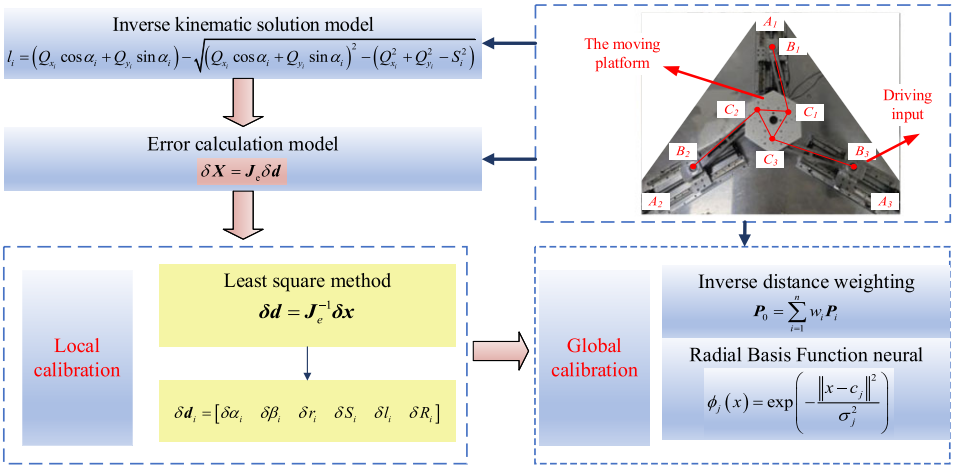


Figure 1. *Flow chart of the calibration method.*

this method more difficult. Moreover, many error models only consider partial geometric error sources, neglecting joint errors, which results in insufficient positioning accuracy in the overall workspace of robots for high-precision tasks. Secondly, some model-free calibration methods exhibit complex network structures and overreliance on extensive training data, making them challenging to apply in practical engineering scenarios. Therefore, in this paper, a data-driven global calibration method considering joint errors is proposed. This approach directly establishes a mapping relationship between robot pose errors and drive joints using spatial interpolation and radial basis function neural network (RBFNN), which can deal with the impact of both geometric and non-geometric errors. With a small amount of data, it can accurately and efficiently predict the position and pose error of the entire workspace of the robot. The contradiction between the measurement efficiency and the calibration accuracy of the traditional numerical method is solved, and the global error compensation of the robot is realized to meet the requirements of high-precision positioning.

This paper makes the following contributions: (1) The proposed global calibration method based on the distance-inverse law and radial basis function neural network makes up for the shortcomings of previous research methods, which suffered from low accuracy and high application difficulty. (2) The proposed calibration strategy has been successfully applied to a planar parallel robot, effectively improving the robot's global positioning accuracy.

The overall process of this paper is illustrated in Fig. 1. The first section is background introduction and literature review. In the second section, the kinematic forward and inverse models for a self-designed 3-PRR (P is Prismatic joint, R is Rotating joint) parallel robot were derived using the closed-loop vector method. In the third section, an error model was established, followed by a global sensitivity analysis for each error source was carried out. Local calibration was then performed using the least-squares method. The study revealed that calibration based on error models using least squares methods can only enhance local accuracy of the robot, failing to achieve comprehensive improvement across the entire workspace. Therefore, in order to enhance the overall precision of the robot, in the fourth section, the inverse distance weighting (IDW) method was employed for error spatial interpolation, and RBFNN was used for error prediction. The experiment was conducted to verify effectiveness of the proposed algorithm. Finally, the fifth section provides the conclusion and future work.

2. Kinematic modeling and analysis

2.1. Inverse kinematic solution

Kinematics modeling of parallel robot is divided into forward and inverse kinematics modeling. The forward kinematics refer to obtaining pose of the moving platform of the robot through the kinematic

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

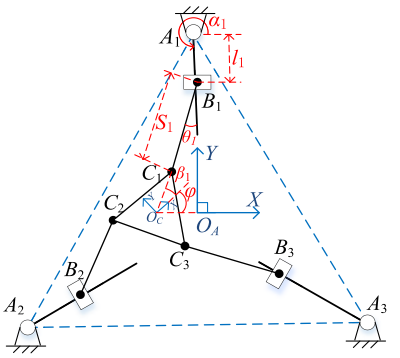


Figure 2. *Planar 3-PRR parallel robot mechanism sketch.*

variable of input joint (position, velocity, and acceleration). On the contrary, the inverse kinematics need to obtain the value of each drive joint under the known motion state of the moving platform [20]. By employing the kinematic model, we can perform error analysis and reveal the transmission relationship between the joint space and the workspace. For the parallel mechanism, the inverse kinematics is simpler than the forward kinematics, making it easier to obtain the analytical solution, whereas the forward kinematics is more complex. Additionally, the inverse kinematics serves as the foundation for trajectory planning and real-time control. Therefore, establishing the inverse kinematic solution model is the primary task.

As shown in Fig. 2, the sketch of the planar 3-PRR parallel robot mechanism was constructed by the regular triangle moving platform $C_1C_2C_3$, the fixed platform, and three symmetrical kinematic chains $A_1B_1C_1A_2B_2C_2A_3B_3C_3$. Each of the kinematic chains has one active prismatic pair (p) followed by two consecutive passive revolute (R) joint. A_1, A_2, A_3 are the regular triangle's three vertices. The vertices OA and O_C are center of the regular triangles $A_1A_2A_3$ and $C_1C_2C_3$, respectively. The $O_A - XY$ is the fixed frame and the $O_C - XY$ is the local moving frame. O_AX and O_CX are parallel to lines A_2A_3 and C_2C_1, respectively. Parameters $\alpha_i, \theta_i (i = 1, 2, 3)$ are the angles at $A_i, B_i (i = 1, 2, 3)$ between the $X -$ axis of the fixed frame and $A_iB_i, B_iC_i (i = 1, 2, 3)$, respectively. $l_i (i = 1, 2, 3)$ is the input displacement of the driving joints, S_i is the length of the passive linkage $B_iC_i (i = 1, 2, 3)$, r denotes the radius of the outer circle of moving platform, and R represents the radius of the outer circle of fixed platform. Additionally, β_i refers to the angle between the $x -$ axis of the local moving frame and $O_CC_i (i = 1, 2, 3)$. $\alpha_1 = 270^0, \alpha_2 = 30^0, \alpha_3 = 150^0, \beta_1 = 30^0, \beta_2 = 150^0, \beta_3 = 270^0, R = 733$ mm, $S_i = 430$ mm and $r = 100$ mm.

In this paper, kinematic constraint equations are used to model the planar 3-PRR parallel robot mechanism [21]. The pose of the moving platform center O_C is denoted as (x, y, φ), and from this information, we can determine the input values $l_i (i = 1, 2, 3)$ for the driving joints. For the ith chain, the kinematic equations established by closed-loop vector method in fixed coordinate system O_A-XY are as follows:

$$\begin{cases} x_{A_i} + l_i \cos\alpha_i + S_i \cos\theta_i = x_{C_i} \\ y_{A_i} + l_i \sin\alpha_i + S_i \sin\theta_i = y_{C_i} \end{cases} \tag{1}$$

where $x_{A_i} = -R\cos\alpha_i, y_{A_i} = -R\sin\alpha_i, x_{C_i} = x - r\cos(\beta_i + \varphi), y_{C_i} = y - r\sin(\beta_i + \varphi)$

By rewriting Eq. (1), we can get:

$$\begin{cases} l_i \cos\alpha_i + S_i \cos\theta_i = Q_{x_i} \\ l_i \sin\alpha_i + S_i \sin\theta_i = Q_{y_i} \end{cases} \tag{2}$$

where $Q_{x_i} = x_{C_i} - x_{A_i} = x - r\cos(\beta_i + \varphi) + R\cos\alpha_i, Q_{y_i} = y_{C_i} - y_{A_i} = y - r\sin(\beta_i + \varphi) + R\sin\alpha_i$

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

According to trigonometric function theorem, the angle θ_i of the Eq. (2) can be eliminated, so, we can get a one-variable quadratic equation as follows:

$$l_i^2 - 2\left(Q_{x_i}\cos\alpha_i + Q_{y_i}\sin\alpha_i\right) l_i + \left(Q_{x_i}^2 + Q_{y_i}^2 - S_i^2\right) = 0 \tag{3}$$

By examining the discriminant $b^2 - 4ac > 0$, we determine that the equation has two solutions. However, considering the actual circumstances of the mechanism, only one of these solutions conforms to the motion equation constraints. The formula for the solvable driving input l_i is as follows:

$$l_i = \left(Q_{x_i}\cos\alpha_i + Q_{y_i}\sin\alpha_i\right) - \sqrt{\left(Q_{x_i}\cos\alpha_i + Q_{y_i}\sin\alpha_i\right)^2 - \left(Q_{x_i}^2 + Q_{y_i}^2 - S_i^2\right)} \tag{4}$$

The relationship between the moving platform pose and the driving input can be obtained by Eq. (4), and the driving input l_i can be solved from the target pose coordinate of the moving platform.

The kinematic Eq. (2) is differentiated with respect to time to obtain the velocity equation:

$$\begin{cases} \cos\alpha_i \dot{l}_i - S_i \sin\theta_i \dot{\theta}_i = \dot{x} - r\sin(\beta_i + \varphi)\,\dot{\varphi} \\ \sin\alpha_i \dot{l}_i + S_i \cos\theta_i \dot{\theta}_i = \dot{y} + r\cos(\beta_i + \varphi)\,\dot{\varphi} \end{cases} \tag{5}$$

For Eq. (5), the corresponding joint velocity can be solved in the form of a matrix:

$$\begin{bmatrix} \dot{l}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\alpha_i & -S_i\sin\theta_i \\ \sin\alpha_i & S_i\cos\theta_i \end{bmatrix}^{-1} \begin{bmatrix} \dot{x} - r\sin(\beta_i + \varphi)\,\dot{\varphi} \\ \dot{y} + r\cos(\beta_i + \varphi)\,\dot{\varphi} \end{bmatrix} \tag{6}$$

Differentiating the velocity Eq. (5) with respect to time yields the acceleration equation:

$$\begin{cases} \cos\alpha_i \ddot{l}_i - S_i \sin\theta_i \ddot{\theta}_i = \ddot{x} - r\cos(\beta_i + \varphi)\,\dot{\varphi}^2 - r\sin(\beta_i + \varphi)\,\ddot{\varphi} - S_i\cos\theta_i\dot{\theta}_i^2 \\ \sin\alpha_i \ddot{l}_i + S_i \cos\theta_i \ddot{\theta}_i = \ddot{y} - r\sin(\beta_i + \varphi)\,\dot{\varphi}^2 + r\cos(\beta_i + \varphi)\,\ddot{\varphi} - S_i\sin\theta_i\dot{\theta}_i^2 \end{cases} \tag{7}$$

Eq. (7) allows us to determine the joint acceleration corresponding to the movement of the moving platform in matrix form:

$$\begin{bmatrix} \ddot{l}_i \\ \ddot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\alpha_i & -S_i\sin\theta_i \\ \sin\alpha_i & S_i\cos\theta_i \end{bmatrix}^{-1} \begin{bmatrix} \ddot{x} - r\cos(\beta_i + \varphi)\,\dot{\varphi}^2 - r\sin(\beta_i + \varphi)\,\ddot{\varphi} - S_i\cos\theta_i\dot{\theta}_i^2 \\ \ddot{y} - r\sin(\beta_i + \varphi)\,\dot{\varphi}^2 + r\cos(\beta_i + \varphi)\,\ddot{\varphi} - S_i\sin\theta_i\dot{\theta}_i^2 \end{bmatrix} \tag{8}$$

2.2. Velocity Jacobian matrix

The velocity Jacobian matrix of the robot represents the mapping relationship between the input velocity vector and the output velocity vector. This matrix can be obtained using either the forward or the inverse kinematics model. The velocity Jacobian matrix plays a vital role in both mechanism singularity analysis and error analysis [22]. Consequently, to analyze the analysis of motion characteristics in parallel robots, the mechanism's inverse kinematics model is derived through the closed-loop vector method, and the velocity Jacobian matrix of the 3-PRR parallel robot mechanism is subsequently derived based on the inverse kinematics model.

When the velocity Eq. (5) is multiplied by $\cos\theta_i$ and $\sin\theta_i$, we obtain the following formula:

$$\begin{cases} \cos\alpha_i\cos\theta_i\dot{l}_i - S_i\sin\theta_i\cos\theta_i\dot{\theta}_i = \dot{x}\cos\theta_i - r\sin(\beta_i + \varphi)\cos\theta_i\dot{\varphi} \\ \sin\alpha_i\sin\theta_i\dot{l}_i + S_i\cos\theta_i\sin\theta_i\dot{\theta}_i = \dot{y}\sin\theta_i + r\cos(\beta_i + \varphi)\sin\theta_i\dot{\varphi} \end{cases} \tag{9}$$

$\dot{\theta}_i$ can be eliminated after summing, and the relationship between the mechanism input and output is as follows:

$$\dot{l}_i\cos(\alpha_i - \theta_i) = \dot{x}\cos\theta_i + \dot{y}\sin\theta_i - r\sin(\beta_i + \varphi - \theta_i)\,\dot{\varphi} \tag{10}$$

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

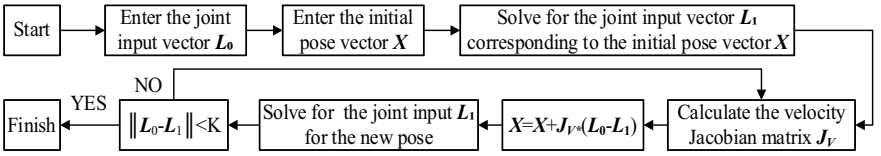


Figure 3. *Flow chart of velocity Jacobian matrix iteration.*

Rewriting for the matrix form is:

$$\begin{bmatrix} \cos(\alpha_1-\theta_1) & 0 & 0 \\ 0 & \cos(\alpha_2-\theta_2) & 0 \\ 0 & 0 & \cos(\alpha_3-\theta_3) \end{bmatrix} \begin{bmatrix} \dot{l}_1 \\ \dot{l}_2 \\ \dot{l}_3 \end{bmatrix} = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & -rsin(\beta_1+\varphi-\theta_1) \\ \cos\theta_2 & \sin\theta_2 & -rsin(\beta_2+\varphi-\theta_2) \\ \cos\theta_3 & \sin\theta_3 & -rsin(\beta_3+\varphi-\theta_3) \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} \tag{11}$$

The Eq. (11) can be expressed in brief description:

$$\boldsymbol{J}_{F1}\dot{\boldsymbol{L}} = \boldsymbol{J}_{F2}\dot{\boldsymbol{X}} \tag{12}$$

In which $\boldsymbol{J}_{F1}$ and $\boldsymbol{J}_{F2}$ are referred to as the input and output Jacobian matrices, respectively, we can get velocity $\dot{\boldsymbol{X}}$ of the moving platform by Eq. (12) as follows:

$$\dot{\boldsymbol{X}} = \boldsymbol{J}_{F2}^{-1}\boldsymbol{J}_{F1}\dot{\boldsymbol{L}} = \boldsymbol{J}_V\dot{\boldsymbol{L}} \tag{13}$$

where $\boldsymbol{J}_V$is velocity Jacobian matrix.

2.3. Forward kinematic solution

The forward kinematics, in contrast to the inverse kinematics, focus on determining the position and orientation of the moving platform based on given driving inputs. In the case of parallel robots, their kinematic equations are typically nonlinear due to the interplay and coupling effects between individual components. Given the position and orientation of an end effector, there can exist multiple sets of joint angles that lead to the same end position. This introduces complexity to the mathematical derivation and computational aspects of solving kinematic equations. Structural and kinematic equations often involve numerous three-dimensional geometric and vector operations, making the solving process laborious and resource-intensive [23]. There is no universally applicable closed-form solution for the forward kinematic solution of parallel robots. It must be tailored to the specific robot structure and tasks. Consequently, determining the positive solution of parallel mechanisms is challenging, and numerical methods are generally employed for this purpose. In this paper, for the planar 3-PRR parallel robot mechanism, the speed Jacobian iterative method is adopted to find the forward kinematic solution. This method gradually approaches the position and orientation of the robot's end effector to solve the positive kinematics of the robot. It offers advantages such as speed and high accuracy, making it a suitable real-time approach [24]. The core idea of the velocity Jacobian matrix iterative method is as follows: using the given input vector, $\boldsymbol{L}_0 = [l_{1_0}, l_{2_0}, l_{3_0}]$, and the known initial pose vector $\boldsymbol{X} = [x, y, \varphi]^{\mathrm{T}}$, along with the corresponding initial input vector, $\boldsymbol{L}_1 = [l_{1_1}, l_{2_1}, l_{3_1}]$, the velocity Jacobian matrix $\boldsymbol{J}_V$ is derived from input to output, given the calculation accuracy K [25]. The solving process of the speed Jacobian matrix iterative method is described in Fig. 3. Initially, the end-effector platform is moved to the initial position vector X through drive input. Using inverse kinematics, the drive input vector $\boldsymbol{L}_0$ is solved, and the velocity Jacobian matrix is determined. Utilizing the Jacobian matrix, the theoretically updated pose vector is calculated, along with the corresponding new drive input vector $\boldsymbol{L}_1$. The completion of the loop is determined by a discerning condition.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

3. Kinematic calibration

Kinematic calibration is an inverse problem in robotics used to determine the robot's kinematic parameters for precise control and motion planning purposes. By measuring the position and orientation of the robot's end-effector, joint angles and kinematic parameters can be deduced, thus improving the model to enhance the robot's motion accuracy, safety, and efficiency [26]. A primary task in improving the positioning accuracy of parallel robot mechanisms is parameter identification and calibration. For a planar parallel mechanism, the end-effector typically has three degrees of freedom, resulting in three positioning errors. These errors primarily stem from input errors and kinematic parameter errors. These errors propagate through the closed-loop motion chain to influence end-effector positioning accuracy. However, kinematic calibration faces challenges such as sensor noise, coupling effects between joints, and the complexity of the calibration process. This study simplifies the error model using small-perturbation approximations and decouples the mechanism through angle transformations to reduce complexity. This yields a linear error model for a planar 3-PRR parallel robot mechanism. The error model can be transformed into a parameter identification model through the generalized inverse transformation, resulting in an overdetermined system of equations. These equations can be solved using the least-squares method to obtain error parameters [27].

3.1. Error modeling

The geometric error model serves as the foundation for error analysis and kinematic calibration of planar 3-PRR parallel robot mechanism, involving the mapping between the motion platform pose error and geometric source errors. Based on the closed-loop vector relationship, the equation for the moving platform pose (x, y, φ) is obtained as follows:

$$\begin{cases} x = l_i \cos\alpha_i + S_i \cos\theta_i - r\cos(\beta_i + \varphi) - R\cos\alpha_i \\ y = l_i \sin\alpha_i + S_i \sin\theta_i - r\sin(\beta_i + \varphi) - R\sin\alpha_i \end{cases} \tag{14}$$

In Eq. (14), except for θ_i being an indirect error source, all others are direct error sources. Introducing a minor error δ for each direct error source and setting $\gamma_i = \beta_i + \varphi$ yields Eq. (15).

$$\begin{cases} x + \delta x = (l_i + \delta l)\cos(\alpha_i + \delta\alpha_i) + (S + \delta S_i)\cos\theta_i - (r + \delta r_i)\cos(\gamma_i + \delta\gamma_i) - (R + \delta R_i)\cos(\alpha_i + \delta\alpha_i) \\ y + \delta y = (l_i + \delta l)\sin(\alpha_i + \delta\alpha_i) + (S + \delta S_i)\sin\theta_i - (r + \delta r_i)\sin(\gamma_i + \delta\gamma_i) - (R + \delta R_i)\sin(\alpha_i + \delta\alpha_i) \end{cases} \tag{15}$$

Eq. (15) is simplified by applying a small-perturbation equivalence formula:

$$\begin{cases} \sin\varepsilon \approx \varepsilon \\ \cos\varepsilon \approx 1 \end{cases} \quad (\varepsilon << 1) \tag{16}$$

The model of Eq. (15) reduces the higher-order terms through Eq. (16) and then finds the difference solution with Eq. (14) as follows:

$$\begin{cases} \delta x = (R - l_i)\sin(\alpha_i)\,\delta\alpha_i + r\sin(\gamma_i)\,\delta\gamma_i - \cos(\gamma_i)\,\delta\gamma_i + \delta S_i\cos\theta_i + (\delta l_i - \delta R_i)\cos(\alpha_i) \\ \delta y = (R - l_i)\cos(\alpha_i)\,\delta\alpha_i - r\cos(\gamma_i)\,\delta\gamma_i - \sin(\gamma_i)\,\delta\gamma_i + \delta S_i\sin\theta_i + (\delta l_i - \delta R_i)\sin(\alpha_i) \end{cases} \tag{17}$$

Due to the coupling between the position and orientation errors of parallel robot mechanisms, it is necessary to decouple position and orientation. The equations for δx and δy are multiplied by $\cos\theta_i$ and $\sin\theta_i$, respectively. After rearranging and calculations, the input and output forms are obtained as follows:

$$\cos\theta_i\delta x + \sin\theta_i\delta y - r\sin(\gamma_i - \theta_i)\,\delta\varphi = \begin{bmatrix} (R - l_i)\sin(\alpha_i - \theta_i)\,\delta\alpha_i + r\sin(\gamma_i - \theta_i)\,\delta\beta_i - \cos(\gamma_i - \theta_i)\,\delta r \\ +\delta S_i + \delta l_i\cos(\alpha_i - \theta_i) - \delta R_i\cos(\alpha_i - \theta_i) \end{bmatrix} \tag{18}$$

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table I. *Description of the error sources in the planar 3-PRR parallel robot mechanism.*

Error source (i = 1,2,3)	Meaning
$\delta\alpha_i$	Static platform angle error
$\delta\beta_i$	moving platform angle error
δr_i	Error of the outer radius of the moving platform $O_C C_i$
δs_i	Length error of the passive rod $B_i C_i$
δl_i	Joint input error $A_i B_i$
δR_i	Error of the outer radius of the static platform $O_A A_i$

In matrix form, this can be represented as follows:

$$\delta \boldsymbol{X} = \boldsymbol{J}_e \delta \boldsymbol{d} \tag{19}$$

where the formula includes:

$$\delta \boldsymbol{X} = \begin{bmatrix} \delta x & \delta y & \delta\varphi \end{bmatrix}^{\mathrm{T}}, \delta \boldsymbol{d}_i = \begin{bmatrix} \delta\alpha_i & \delta\beta_i & \delta r_i & \delta S_i & \delta l_i & \delta R_i \end{bmatrix}^{\mathrm{T}}, \delta \boldsymbol{d} = \begin{bmatrix} \delta \boldsymbol{d}_1 & \delta \boldsymbol{d}_2 & \delta \boldsymbol{d}_3 \end{bmatrix}^{\mathrm{T}}$$

$$\boldsymbol{J}_e = \boldsymbol{J}_1^{-1} \boldsymbol{J}_2, \boldsymbol{J}_1 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & -r\sin(\gamma_1 - \theta_1) \\ \cos\theta_2 & \sin\theta_2 & -r\sin(\gamma_2 - \theta_2) \\ \cos\theta_3 & \sin\theta_3 & -r\sin(\gamma_3 - \theta_3) \end{bmatrix}, \boldsymbol{J}_2 = \begin{bmatrix} \boldsymbol{A}_{11} & \boldsymbol{0}_{1*6} & \boldsymbol{0}_{1*6} \\ \boldsymbol{0}_{1*6} & \boldsymbol{A}_{22} & \boldsymbol{0}_{1*6} \\ \boldsymbol{0}_{1*6} & \boldsymbol{0}_{1*6} & \boldsymbol{A}_{33} \end{bmatrix}$$

$$\boldsymbol{A}_{ii} = \begin{bmatrix} (R - l_i)\sin(\alpha_i - \theta_i) & r\sin(\gamma_i - \theta_i) & -\cos(\gamma_i - \theta_i) & 1 & \cos(\alpha_i - \theta_i) & -\cos(\alpha_i - \theta_i) \end{bmatrix}$$

From Eq. (19), the kinematic error model of the planar 3-PRR parallel robot mechanism can be derived. Using the known error parameters, the corresponding end-effector pose error can be solved. This error model encompasses all geometric error sources in the mechanism, providing important guidance for subsequent calibration and optimization in the system.

3.2. Analysis of error source sensitivity

In order to evaluate the effects of geometric source errors on the pose accuracy of the moving platform, the sensitivity analysis is carried out generally after formulating the geometric error model. Error source sensitivity analysis is crucial in robotics and is used to assess the impact of various error sources on the calculation results of the robot's end-effector position and pose [28]. Through error source sensitivity analysis, we can determine which error sources have the most significant influence on the robot's kinematic performance. This, in turn, helps guide error correction and optimization measures to enhance the robot's motion accuracy and positioning precision. In the constructed error model of the planar 3-PRR parallel robot mechanism, there are 18 error sources that can affect the accuracy of end positioning. Some error sources have a considerable impact on the end positioning accuracy, while others have less sensitivity to the end positioning accuracy. Eighteen error sources are shown in Table I.

Consider the error transfer matrix $\boldsymbol{J}_e$. The elements in the first row correspond to the end error δx, those in the second row correspond to δy, and those in the third row correspond to $\delta\varphi$. For example, the first element of the error transfer matrix $\boldsymbol{J}_e$ determines the error source $\delta\alpha_1$ weight factor affecting the end error δx. In other words, the percentage of elements in the row of the error transfer matrix $\boldsymbol{J}_e$ reflects the sensitivity of error propagation [29]. The greater the weight of the error source is, the larger the introduced error, and the more it contributes to the terminal localization error. Consequently, the sensitivity of error propagation is higher. Since there are different error matrix values at different points on the platform, assume that all angle errors are 0.001° and all length errors are 0.05 mm. To calculate

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

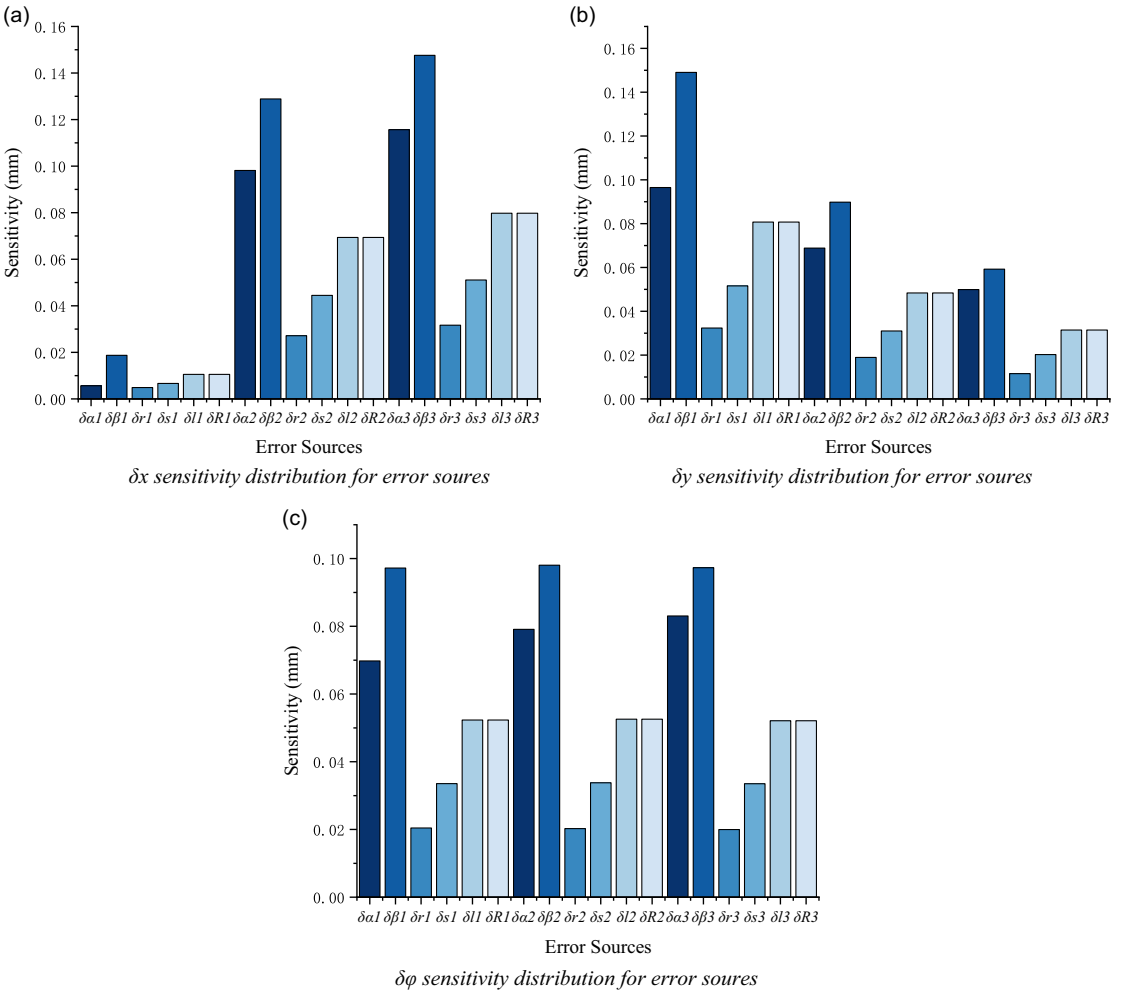


Figure 4. *Sensitivity distribution of the 18 error sources.*

the Global Error Sensitivity Index of error sources, select positioning error information from the four quadrants of the workspace, and utilize formula (20) as follows:

$$\mathrm{GESI} = \iint_s \left[\left(J_{eij} * \delta d_j\right) / \delta X_i\right] ds / \iint_s ds, (i = 1, 2, 3; j = 1 \sim 18) \tag{20}$$

where J_{eij} represents the weighting coefficient of the error source, δd_j is the error source, δX_i denotes the pose error vector, and 's' represents the workspace area.

Figure 4 is sensitivity distribution plot of the overall error sources. From Fig. 4, it is evident that in the planar 3-PRR parallel robot mechanism, the δx error is primarily influenced by joints 2 and 3, while δy is affected by all three joints simultaneously. The error $\delta\varphi$, related to rotational freedom, is caused by the symmetry of the three joints. This characteristic is inherent to the platform's structure, where the rotation of the platform axis must involve all three joints, resulting in $\delta\varphi$ being influenced by all three joints simultaneously. Table II is comparison of the effects of the three branched errors on the terminal errors.

Since the platform mounting angle error is multiplied by the radius, it magnifies the arc length error. Theoretically, angle bias leads to a significant bias in the end alignment. The installation angle error of the guide rail remains constant, while the driving distance varies. Therefore, the error is not a fixed

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table II. *Comparison of the effects of the three branched errors on the terminal errors.*

Main error source	δx	δy	$\delta \varphi$
Angle errors $\delta\alpha_i$ and $\delta\beta_i$	Branch Chains 2 and 3 take the lead.	Collective influence, with Branch Chain 1 being more prominent.	Impacted by all three branch chains.

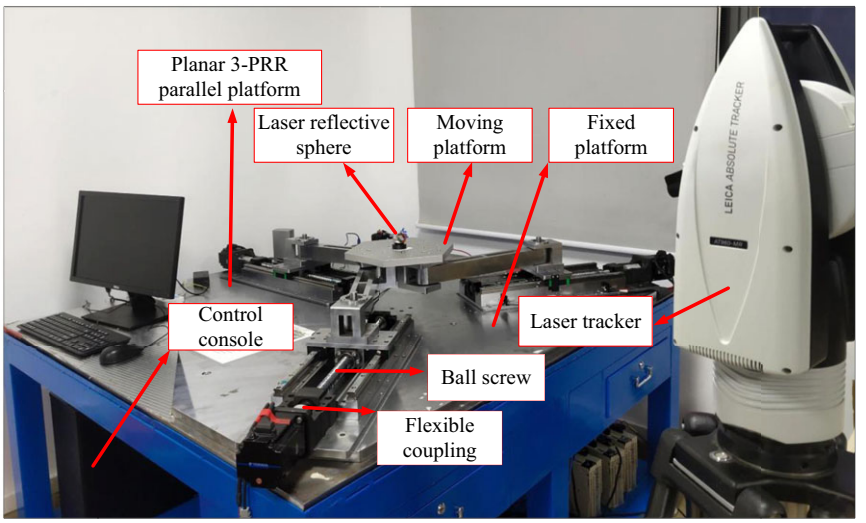


Figure 5. *Experimental equipment and environment.*

value but is related to the input distance. The larger the input distance is, the greater the end stroke, and consequently, the greater the error.

3.3. Error measurement

Error measurement is a critical step in kinematic calibration, as experimental measurement data directly influence parameter identification and compensation outcomes. The experimental system is built as seen in Fig. 5 The experimental platform is driven by a Yaskawa servo drive motor and controlled using a DMC-4183 PCI bus motion control card developed by the GALIL Company. The measurement instrument employed is a Leica AT960-MR laser tracker manufactured by Leica in Germany. It offers a resolution of 0.1 μm and a measurement accuracy of 10 μm.

The specific process of the experiment is outlined as follows:

(1) Point selection: Initially, 40 positioning points are chosen within the working space of the moving platform, following a $10 * 10$ mm distribution pattern, in each of the four quadrants. Simultaneously, 40 positioning points are selected at 10 mm intervals along the coordinate axes, resulting in a total of 80 theoretical positioning points for subsequent measurement experiments as shown in Fig. 6.

(2) Zeroing: The laser retroreflector is placed at the center point O_C of the moving platform. The fixed coordinate system $O_A - XY$ is used to a world coordinate system of the laser tracker. The center O_C of the moving platform is adjusted to coincide with the fixed coordinate original point O_A, thereby completing the zeroing process.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

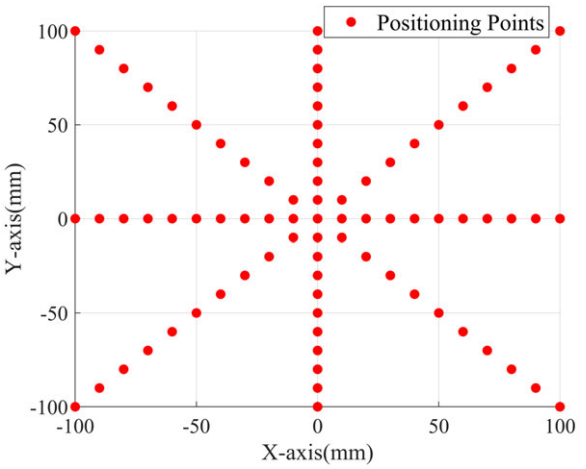


Figure 6. *The theoretical positioning point distribution.*

(3) Execution: The control interface is developed in Visual Studio manages planar 3-PRR the parallel robot mechanism. After connecting the control card, it receives input for the speed and acceleration of the drive joint. Using the inverse kinematic model, it calculates the drive input l_i, corresponding to the target positioning point (x_0, y_0, φ_0). It then initiates the operation of the servo drive motor based on the provided drive input displacement.

(4) Recording: After the moving platform has completely stopped, the real position $(\overline{x}_0, \overline{y}_0, \overline{\varphi}_0)$ is measured by the laser tracker. This process is repeated for 80 positioning points, and the recorded data samples are compiled. The center poses error of the moving platform $\delta \boldsymbol{X}$ is then solved for parameter identification.

3.4. Parameter identification

Matrix $\boldsymbol{J}_e$ represents the error transfer matrix of the planar 3-PRR parallel robot mechanism. By measuring the positioning error of the moving platform $\delta \boldsymbol{X}$ and performing calculations, $\delta \boldsymbol{d}$ is identified. The error model is then reformulated as the parameter identification formula as follows:

$$\delta \boldsymbol{d} = \boldsymbol{J}_e^{-1} \delta \boldsymbol{X} \tag{21}$$

where $\boldsymbol{J}_e^{-1}$ is generalized inverse matrix that can be obtained from the default square array mentioned earlier, where the number of equations is less than the number of unknowns ($3 < 18$). There are three moving platform errors and 18 error sources, making Eq. (21) an indefinite system [30]. To identify the parameters $(\delta\alpha_i, \delta\beta_i, \delta l_i, \delta r_i, \delta S_i, \delta R_i)(i = 1, 2, 3)$, redundant pose data are needed. This involves measuring the error at multiple points (more than 6 points) within the working space of the end platform experimentally, which allows for the calculation of $\delta \boldsymbol{d}$.

The calibration method based on the error model can be transformed into the problem of solving overdetermined equations. By calculating the least-squares solution of $\delta \boldsymbol{d}$, it becomes possible to identify the error source for each clade. Based on the data measured in section 3.1 and parameter identification according to Eq. (21), the solution results are presented in the following Table III.

3.5. Forward kinematics simulation

The values of the error sources are either close to 0 or equal to 0. Upon analyzing the results, it becomes evident that the least-squares solution of the overdetermined equation is not the exact solution. The accuracy of the identified parameters decreases as the model becomes more complex. It is noted that the

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table III. *Parameter identification results for three branches of error.*

	Branch chain 1	**Branch chain 2**	**Branch chain 3**
$\delta\alpha_i$ (°)	0.0094	0.0050	0.0030
$\delta\beta_i$ (°)	0.0073	−0.0018	0
δl_i (mm)	−0.2635	0.3707	−0.2933
δr_i (mm)	0	0	0.0054
δs_i (mm)	0.0041	0	0.0187
δR_i (mm)	0	0	0

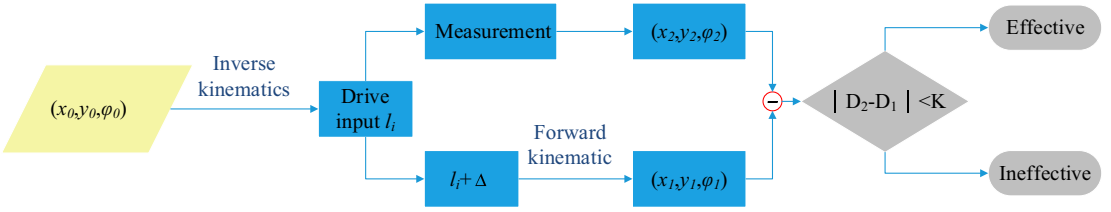


Figure 7. *Simulation of the forward kinematic solution.*

accuracy of the platform has not been effectively improved. Based on the measured localization error, it is observed that the further the localization error is from the origin, the larger the error becomes. When considering the mechanism and sensitivity analysis, it is apparent that the identified constant error parameter is relatively small compared to the variable error parameter. Moreover, the influence of coupling and the ball wire rod on the drive input is taken into account. The flexible deformation of the coupling and the wire rod drive leads to an increase in drive input error with greater drive distance.

Based on the aforementioned overdetermined equations, repeat parameter identification after removing error sources with smaller error values. The analysis indicates that the drive input error is significantly larger than the angle error; the identified error parameters are applied to compensate the theoretical kinematic model. To verify the effectiveness of error parameters, a simulation process was employed to emulate the actual operation of the parallel platform by solving the kinematic forward solution. As illustrated in Fig. 7, the process begins by using kinematic inverse solutions to calculate the drive inputs. Subsequently, the identified error parameters are incorporated into the kinematic forward solution model. Finally, a numerical iterative method is applied to determine the end-effector pose. Comparing the computed results with the actual moving platform pose error allows for an assessment of the effectiveness of the error parameters [31].

As observed in Fig. 8, most of the simulated motion errors align closely with the measured motion errors, confirming the validity of the results from the repeated parameter identification.

4. Experimental results and analysis

4.1. Identification parameter compensation of overdetermined equations

After compensating for the identified error parameters in the kinematic inverse model for the 80 points, the positioning error is remeasured using the aforementioned error measurement method. The measurement results demonstrate a significant improvement in positioning accuracy. Figure 9(a) depicts a 3D distribution map of the positioning error measured using the kinematic theoretical parameters. As depicted in Fig. 9(a), the positioning error tends to increase with the distance from the central origin, with smaller errors closer to the center. Figure 9(b) illustrates a 3D distribution map of positioning error measured after compensating for the identified errors in the forward kinematic model. Figure 9(c) provides a comparison of the positioning error distribution before and after calibration. In the uncompensated

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

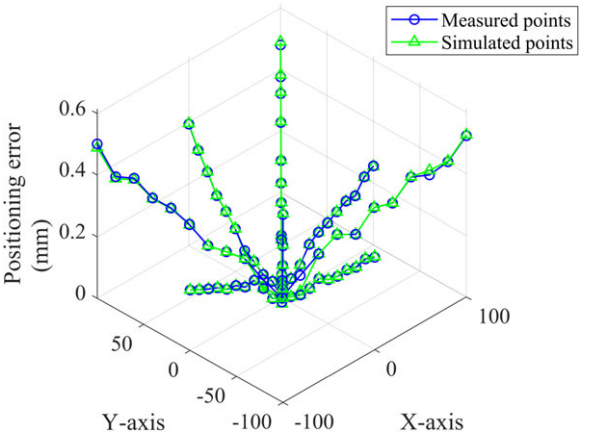


Figure 8. *Simulation positioning error comparison chart.*

state, the error tends to spread and expand around the origin. After calibration, the positioning error distribution exhibits more flatter. Considering nongeometric error sources and artificial measurement factors, the positioning errors do not disappear completely.

The data in Table IV show that the average positioning error is 0.2115 mm, with a maximum error of 0.6103 mm and a minimum error of 0.0141 mm. Moreover, the average positioning error is reduced to 0.0241 mm, with the maximum error being only 0.0583 mm and the minimum being 0 mm. The average positioning accuracy has effectively improved by 88.60%.

Figure 10 is a 3D distribution map of the attitude angle error measured by the kinematic theoretical parameters. As shown in Fig. 10(a), the attitude angle error is closer to the central origin and smaller, and the attitude angle error increases with the distance from the origin. Figure 10(b) is a 3D distribution map of the attitude angle error measured after the identification error is compensated in the kinematic model. As shown in Fig. 10(b), the attitude angle error decreases. Figure 10(c) shows a comparison of the error distribution of the attitude angle before and after calibration. The trend of the attitude angle before and after calibration changes less, but the overall error of the attitude angle is significantly reduced after calibration.

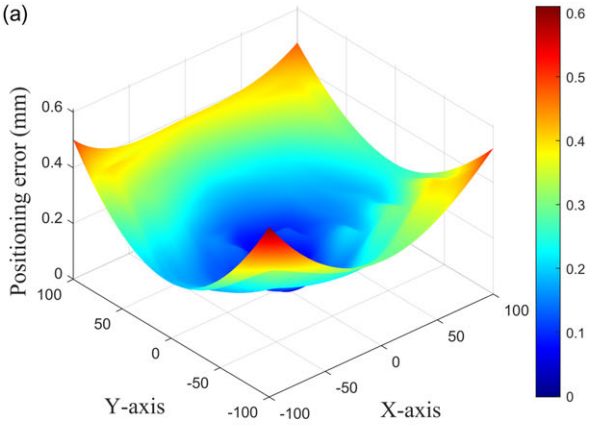
Table V provides statistical data for the experimental results of orientation angles, comparing errors before and after calibration. From Table 5, it can be observed that the average error of orientation angles decreased from 0.0859° to 0.0501°. The maximum orientation angle error decreased from 0.1892° to 0.0877°, and the minimum orientation angle error decreased from 0.0145° to 0.0002°. The local average accuracy of orientation angles improved by 41.64%.

4.2. Distance inversion method of space interpolation compensation

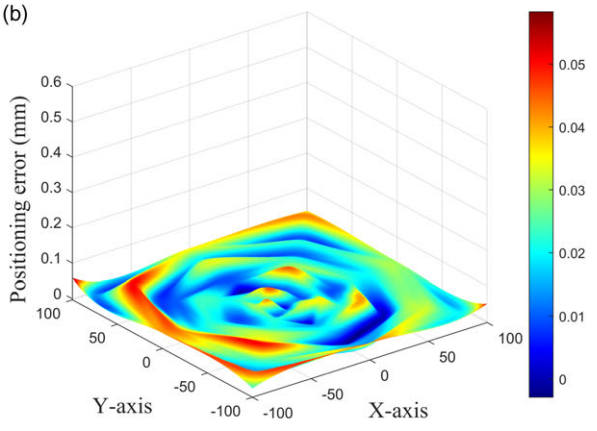
The distance inversion method (inverse distance weighting, IDW) is a spatial interpolation technique used to estimate the values of unknown points based on the values of known points [32]. This method relies on distance weighting, where known points closer to the unknown point have a greater influence on the estimation. In the context of error identification in the workspace, the distance inversion method is employed to interpolate unmeasured points within the workspace, predicting their error distribution across the entire workspace [33]. It assumes that the error value at an unknown position is inversely related to the error values at surrounding known positions, with closer known points having a more significant impact on the estimated value. The error vector at the unknown position can be calculated using the Eq. (22). This compensation approach can enhance the global positioning accuracy of the moving platform [34].

$$P_0 = \sum_{i=1}^{n} w_i P_i \tag{22}$$

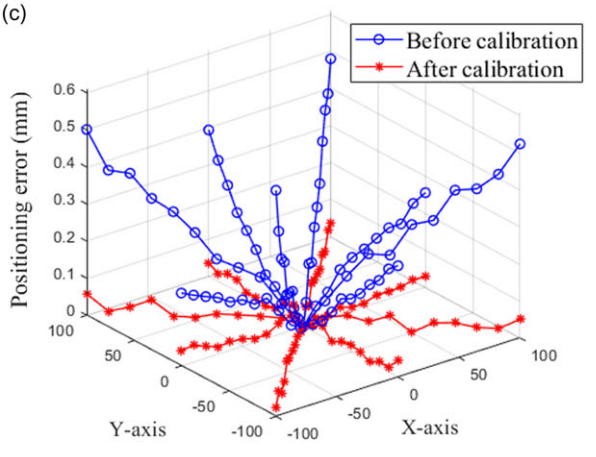
https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press



Positioning error of the moving platform before calibration



Positioning error of the moving platform after calibration



Comparison of the positioning error of the moving platform before and after calibrati

Figure 9. *Positioning error distribution of the moving platform.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

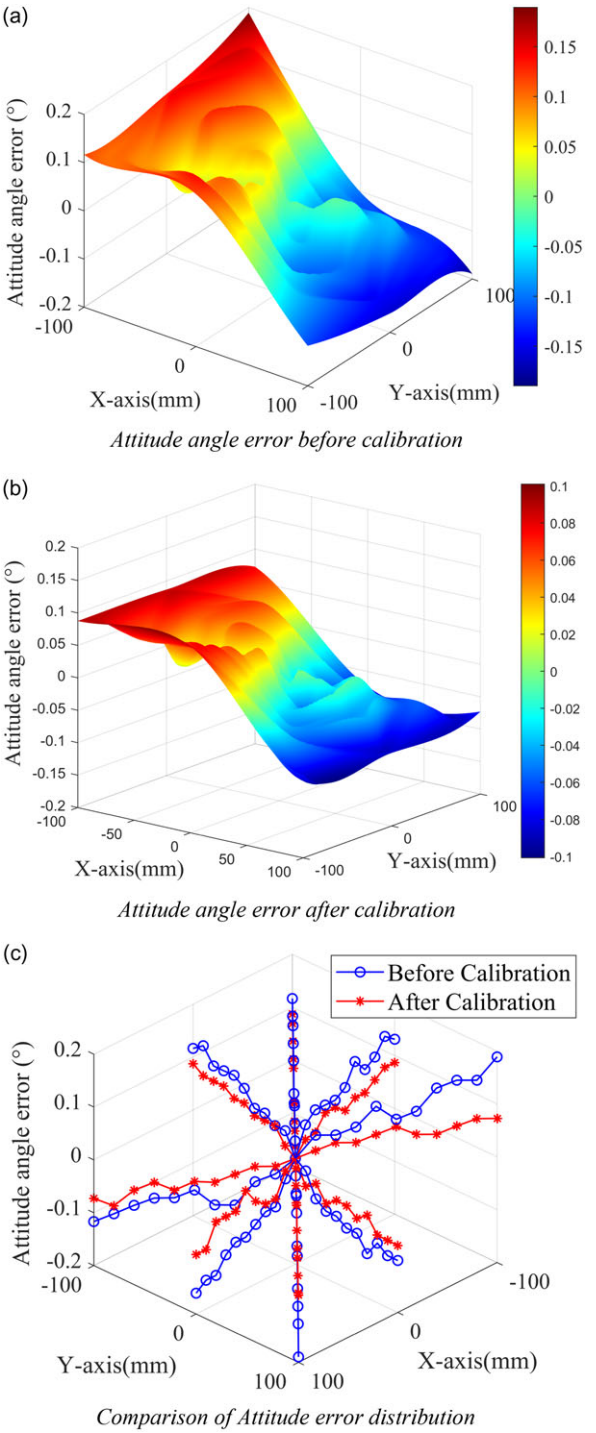


Figure 10. *Attitude angle error distribution of the moving platform.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table IV. *Results of positioning error calibration.*

For the 80 positioning points	Before calibration	After calibration
Average error/mm	0.2115	0.0241
Maximum error/mm	0.6103	0.0583
Minimum error/mm	0.0141	0.0000

Table V. *Attitude angle calibration results.*

Attitude angle	Before calibration	After calibration
Average error/°	0.0859	0.0501
Maximum error /°	0.1892	0.0877
Minimum error/°	0.0145	0.0002

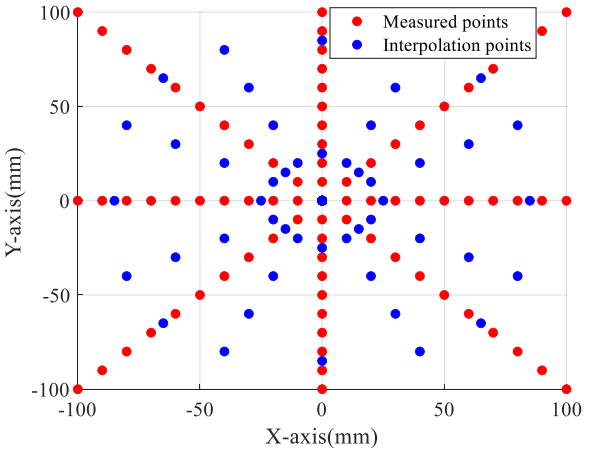


Figure 11. *Positioning points interpolation.*

where $w_i = \frac{f(d_i)}{\sum_{i=1}^{n} f(d_i)}, f(d_i) = \frac{1}{d_i^2}$, P_0 represents the error vector at the interpolation point, P_i represents the error vector at an unknown sample point (x_i, y_i), w_i represents the weight associated with the sample point, and d_i represents the distance from the sample point (x_i, y_i), to the interpolation point (x_0, y_0).

Figure 11 depicts the distribution of sample points, comprising a total of 128 localization points. As shown in Fig. 11, the real measurement of 80 positioning points (red points) is selected as sample in the above and used Eq. (22) to calculate the positioning error vectors for 48 interpolated points (blue points).

Figure 12 illustrates a partial display of the distance interpolation in the workspace. The red points represent sample points, and distance calculations are performed on the blue interpolation points to obtain corresponding weights. The process of interpolating the distance-inverse space is as follows:

(1) As shown in Fig. 12, selecting 48 unknown positioning points (blue points) based on their distances from the known positioning points (red points), starting with the farthest and moving toward the nearest.

(2) Assign the values of the known anchor driver input errors to the interpolation point.

(3) Calculate the distances between the unknown sites and the known sites.

(4) Calculate the weights of the unknown sites based on their distances.

(5) Solve the attribute values of the unknown locations using the calculated weights.

(6) Compensate the platform using the error parameters of the unknown sites.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table VI. *The results of positioning error after interpolation.*

48 points interpolated for localization	Before calibration	After calibration
average error/mm	0.3607	0.1473
maximum change/mm	0.8022	0.4373
least error/mm	0.0922	0.0000

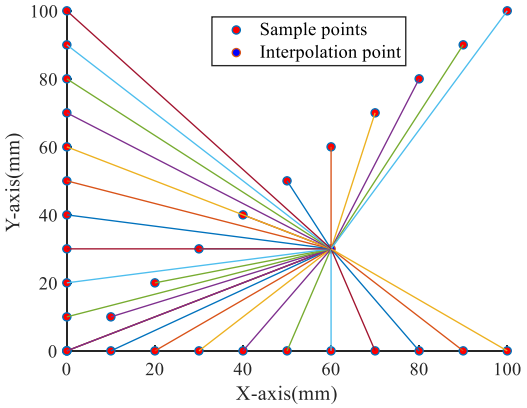


Figure 12. *Schematic diagram of inverse distance method.*

Table VI illustrates that by employing the distance-inverse method for spatially interpolating error parameters, the average positioning error decreased from 0.3607 mm before compensation to 0.1473 mm. Furthermore, the maximum error decreased from 0.80 mm to 0.44 mm, and the minimum error decreased from 0.92 mm to 0 mm. This resulted in an overall average positioning accuracy improvement of 59.16%.

Figure 13(a) presents a 3D distribution of the positioning error before interpolation, with errors generally increasing outward. This aligns with the observation that the primary error in the planar 3-PRR parallel robot mechanism tends to increase with greater distances. Figure 13(b) displays a 3D distribution map of positioning error after interpolation compensation. Although one end of the graph is more extreme, the overall error distribution appears to become more uniform. By examining the comparison map of the positioning error before and after calibration in Fig. 13*(c)*, it becomes evident that the overall error is significantly reduced after calibration.

Figure 14 depicts the three-dimensional distribution of attitude angle errors resulting from the IDW calibration experiment. As shown in Fig. 14(a), before calibration, the attitude angle errors in the workspace are smaller near the center origin and increase as the distance from the origin increases. Figure 14(b) presents the three-dimensional distribution of attitude angle errors measured after calibration, where the overall error decreases. Figure 14(c) compares the distribution of attitude angle errors before and after calibration. The trend in attitude angle errors before and after calibration shows a minor change, with an overall significant reduction in absolute error values.

Table VII provides statistical data for the experimental results of attitude angles using IDW calibration, comparing errors before and after calibration. From Table VII, it can be observed that the average error of attitude angles decreased from 0.0727° before calibration to 0.0429° after calibration. The maximum attitude angle error decreased from 0.1607° to 0.1435°, and the minimum attitude angle error decreased from 0.0114° to 0°. The global accuracy of attitude angles improved by 40.99%.

4.3. RBF neural network prediction error compensation experiment

The topology structure of the radial basis function (RBF) network is straightforward, providing the capability to store system information within neurons and their connection weights. This network exhibits

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

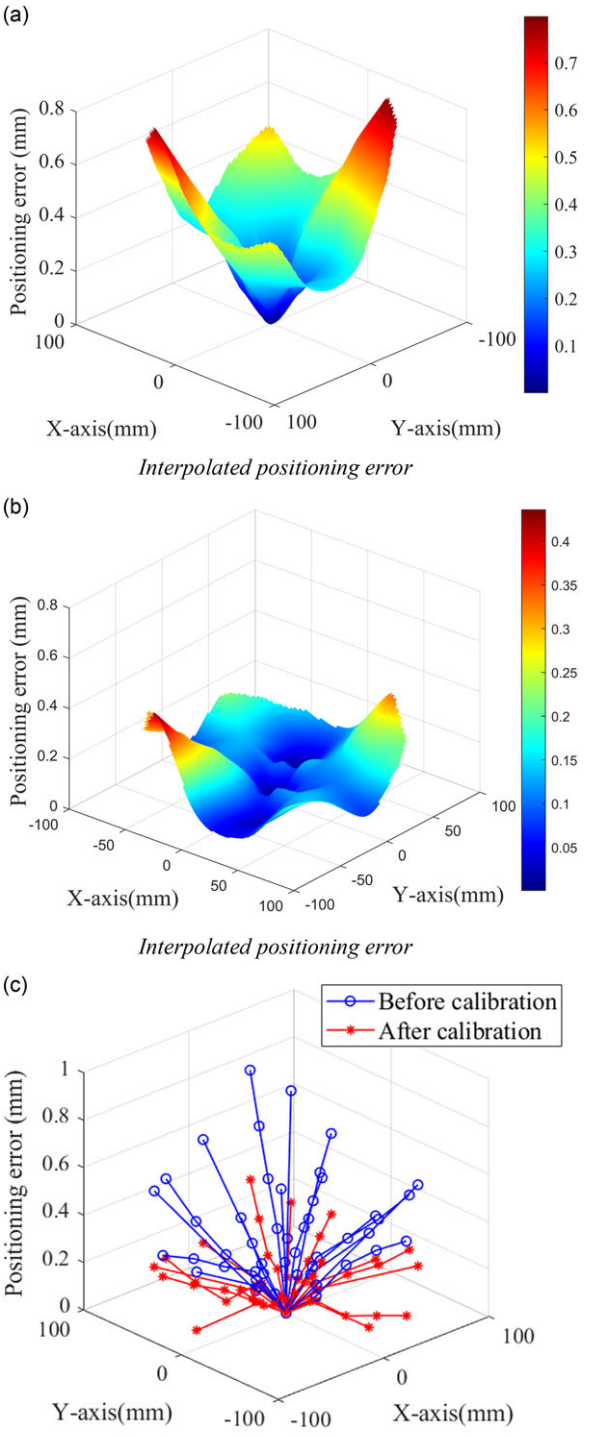


Figure 13. *Positioning error distribution.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

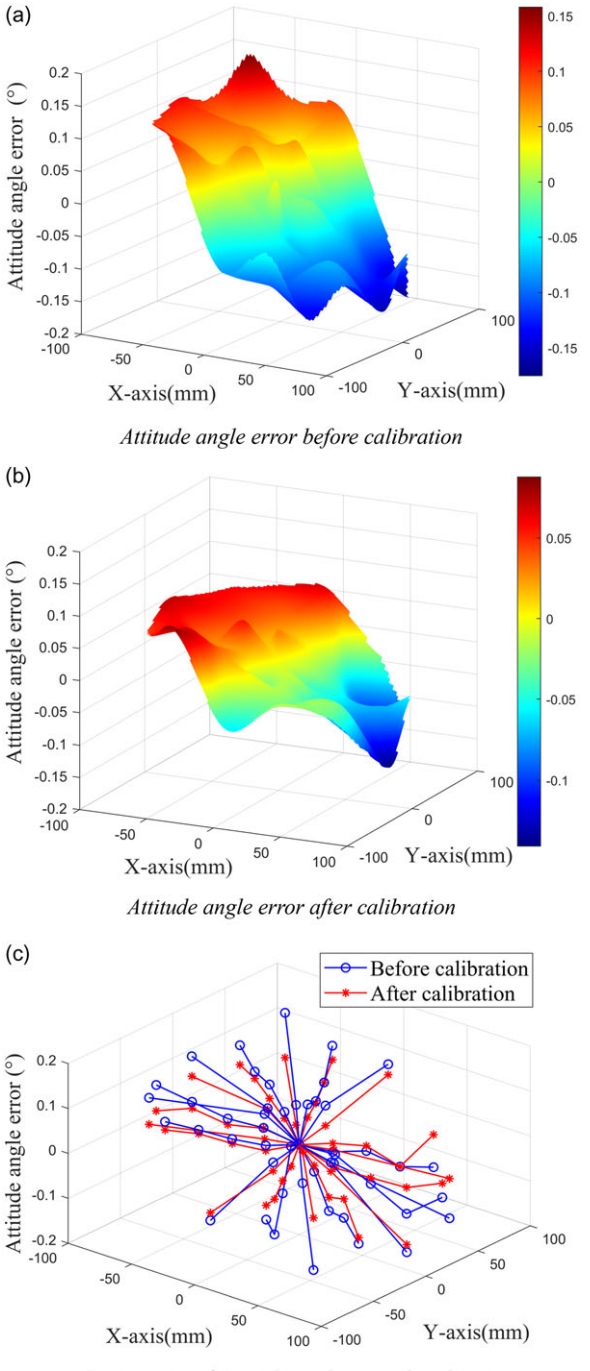


Figure 14. *Attitude angle error distribution of the moving platform.*

global approximation abilities, adheres to continuous incentive conditions, and boasts strong fault tolerance and robustness. Consequently, RBF networks have found extensive application in areas such as signal processing, pattern recognition, and modeling and control of nonlinear systems. Its topology is illustrated in Fig. 15. The RBF network is typically regarded as a three-layer network comprising an input layer, a hidden layer, and an output layer [35].

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table VII. *Attitude angle calibration results.*

Attitude angle	Before calibration	After calibration
Average error/°	0.0727	0.0429
Maximum error/°	0.1607	0.1435
Minimum error/°	0.0114	0

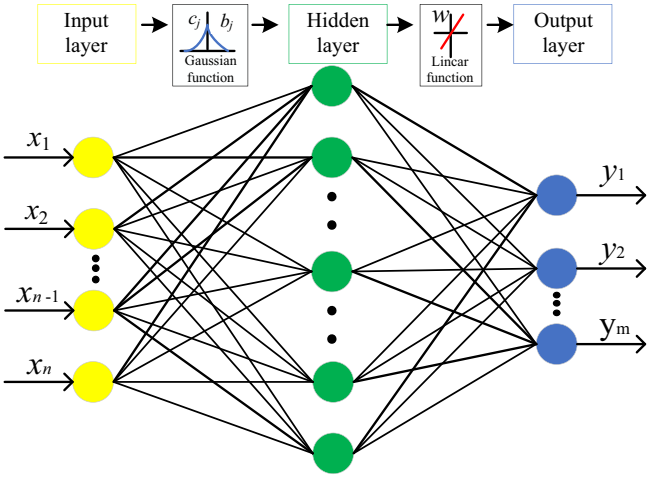


Figure 15. *Topology diagram of the radial basis function network.*

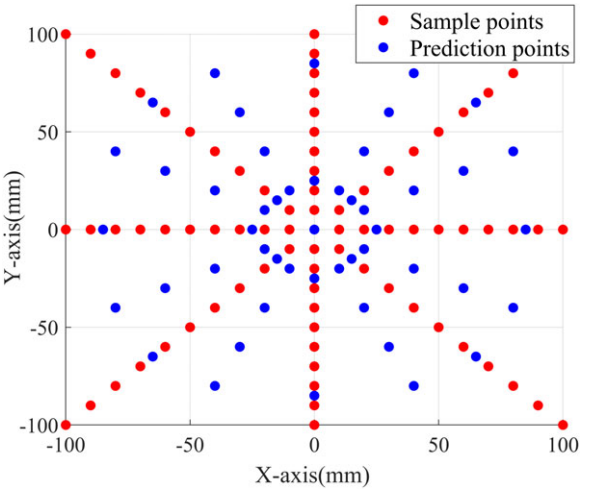


Figure 16. *Sample point distribution of the radial basis function network.*

The RBF neural network can calculate the output y_i for any input vector $\boldsymbol{X}$ through Eq. (23). In this paper, 80 positioning point (red points) errors serve as input vector, while 48 unknown positioning points (blue points) are the output vector as shown in Fig. 16 depicts the distribution of sample points for the parallel robot, where the red points represent sample points obtained through experimental measurements, and the blue points indicate unknown points used for prediction.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

The output of the RBF neural network [36] is computed for an arbitrary input vector $\boldsymbol{X}$

$$y_i = \sum_{j=1}^{L} w_{ij}\phi_j\left(\boldsymbol{X} - c_j\right) \tag{23}$$

where w_{ij} represents the weight connected between the hidden layer and the output, L is the number of neurons in the hidden layer, c_j is the prototype center of the hidden-layer neurons, and ϕ_j is a Gaussian function.

The hidden layer is a critical component of the RBF neural network, often referred to as the radial basis function layer [37]. The responses of the hidden neurons are determined using a similarity function, which relies on a distance-based measure for calculating the distance between two points. The similarity function employed in this paper is a Gaussian function [38]. The calculation formula is shown in Eq. (24).

$$\phi_j\left(\boldsymbol{X} - c_j\right) = \exp\left(-\frac{\left\|\boldsymbol{X} - c_j\right\|^2}{\sigma_j^2}\right) \tag{24}$$

Where σ_j is the spread parameter. At the beginning of the RBF training, c_j centers are randomly selected and their optimal values are found during the training phase using one of the clustering methods.

As depicted in Fig. 16, the study utilized positioning data from 128 specified points as training samples through the experimental platform. Among these, 80 sets of data were allocated to the training set, while 48 sets were designated for the validation set. The coordinates of the positioning points serve as inputs to the neural network model, with the error vectors being the outputs of the network model.

Once the neural network is trained, it can be used to predict the error vector at target points. After training the RBF neural network in MATLAB, various performance metrics are employed to assess the prediction model's effectiveness. These metrics include the mean absolute error (*MAE*), mean squared error (*MSE*), and determination coefficient (R^2) [39]. The R^2 is commonly used in regression analysis to measure the fit of the neural network model to the predicted data. The R^2 value ranges from 0 to 1, with a higher value indicating a better fit of the model to the data.

The formula MAE is calculated as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_e - y_p\right| \tag{25}$$

The formula MSE is calculated as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_e - y_p\right)^2 \tag{26}$$

where y_e represents the expected value and y_p represents the predicted value.

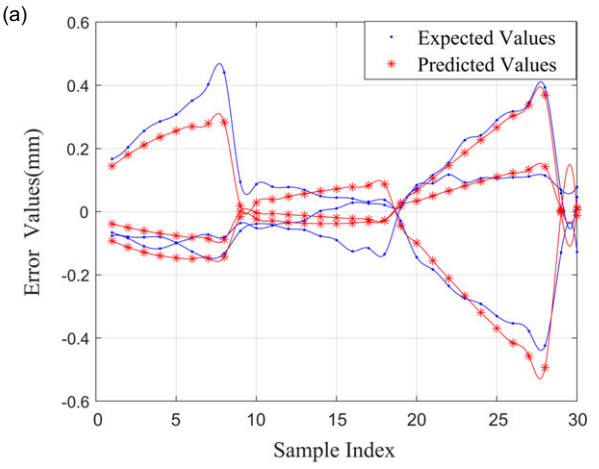
The R^2 formula is calculated as follows:

$$R^2 = \frac{\left(n\sum_{i=1}^{n} y_{ei}y_{pi} - \sum_{i=1}^{n} y_{ei}\sum_{i=1}^{n} y_{pi}\right)^2}{\left[n\sum_{i=1}^{n} {y_{ei}}^2 - \left(\sum_{i=1}^{n} y_{ei}\right)^2\right]\left[n\sum_{i=1}^{n} {y_{pi}}^2 - \left(\sum_{i=1}^{n} y_{pi}\right)^2\right]} \tag{27}$$

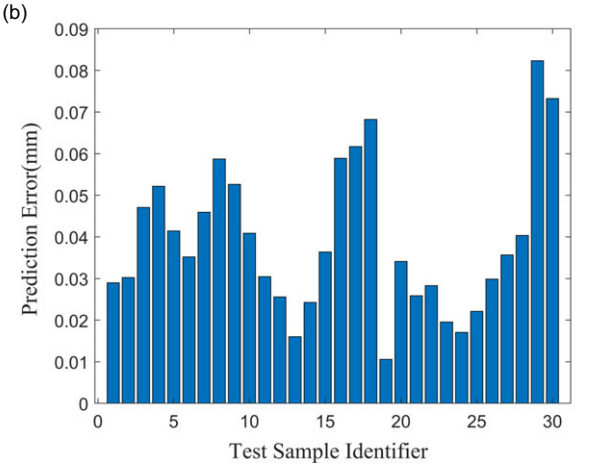
As shown in Fig. 17(a), the comparison between the neural network's post-training predictions and the expected values reveals a close alignment, with the curves demonstrating concordance. Additionally, as depicted in Fig. 17(b), the overall prediction errors are less than 0.1 mm. Simultaneously, Fig. 17(c) illustrates the fitting performance of the three branch predictions. It can be concluded that the fitting effect meets the platform's accuracy requirements.

After training, the neural network has 50 hidden-layer neurons. Table VIII displays the trained RBF neural network parameters, where the MAE is 0.0391, the MSE is 0.0018, and the coefficients of determination R^2 for different error vectors are 0.8854, 0.9656, and 0.9655. Additionally, when considering the error vector comparison graphs and distribution maps, it is evident that the trained RBF network exhibits a high degree of fitting and excellent predictive data performance.

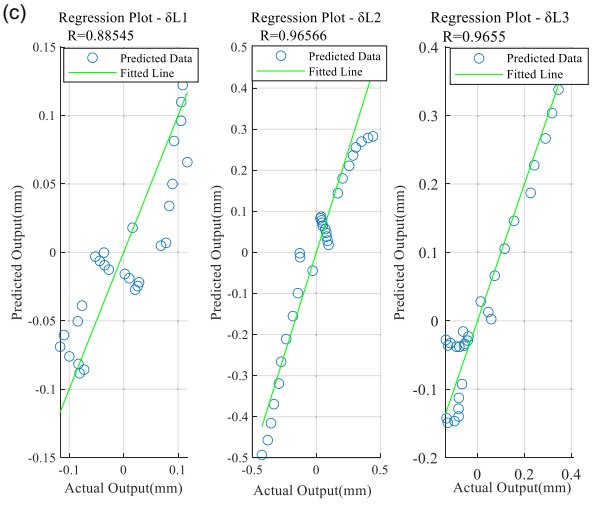
https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press



RBF prediction error vector comparison



RBF error vector distribution map



The regression performance plot of the RBF network

Figure 17. *Performance evaluation plots of the radial basis function neural network.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table VIII. *RBF training performance parameter.*

	MAE/mm	MSE/mm	R_1^2	R_2^2	R_3^2
RBF parameter	0.0391	0.0018	0.8854	0.9656	0.9655

Table IX. *Results of radial basis function prediction error compensation.*

48 predicted compensation points	Before compensation	After compensation
Average error/mm	0.1677	0.0619
Maximum error/mm	0.3701	0.13
Minimum error/mm	0.0583	0.02

Table X. *Attitude angle calibration results.*

Attitude angle	Before calibration	After calibration
Average error/°	0.0727	0.0389
Maximum error/°	0.1607	0.0850
Minimum error/°	0.0114	0

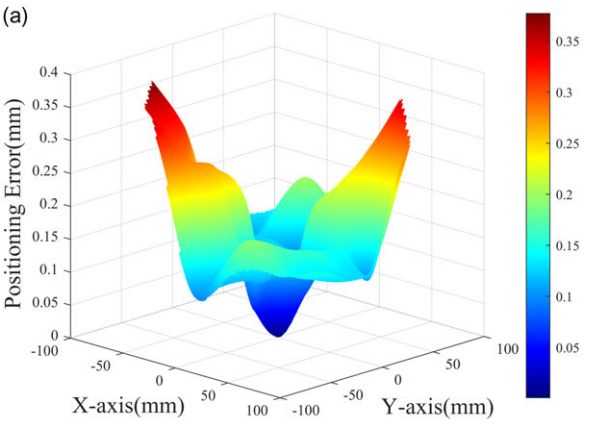
As depicted in Fig. 18(a), it is evident that the compensation effect extends from the center to the surrounding areas. In particular, there is a noticeable improvement in the first and third quadrants within the workspace. It is worth considering that differences in measurement equipment calibration or setup may contribute to variations in the positioning error distribution between the first two measurements. Despite these differences, as depicted in Fig. 18(b), the error distribution pattern after compensation generally follows the same trend as before compensation, but with a significant reduction in overall error values. Taking all factors into account, as Fig. 18(c) shows, it can be concluded that the error vector prediction compensation carried out by the RBF neural network has a substantial positive impact. This approach greatly enhances the compensation efficiency, positioning accuracy, and overall motion performance of the planar 3-PRR parallel robot mechanism.

Based on the data presented in Table IX, it can be observed that after implementing RBF prediction error compensation for 48 prediction points in the planar 3-PRR parallel robot mechanism, the average positioning error decreased from 0.1677 mm before compensation to 0.0619 mm. Additionally, the maximum positioning error decreased from 0.3701 mm to 0.1300 mm, and the minimum positioning error decreased from 0.0583 mm to 0.0200 mm. This compensation process resulted in an increase in accuracy of approximately 63.05%.

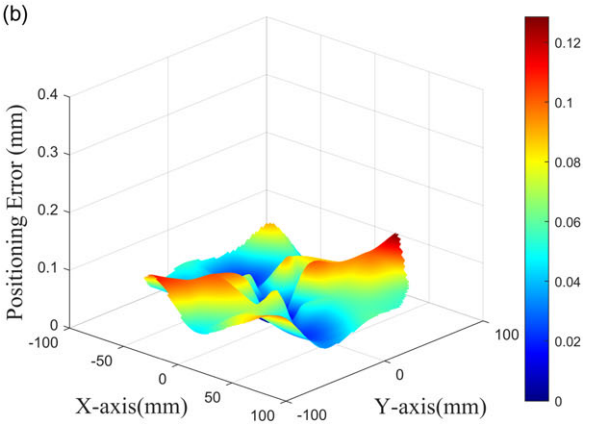
Figure 19 illustrates the three-dimensional distribution of attitude angle errors resulting from the RBF calibration experiment. As shown in Fig. 19(a), before calibration, attitude angle errors in the workspace are smaller near the center origin and increase as the distance from the origin increases. Figure 19(b) depicts the three-dimensional distribution of attitude angle errors measured after calibration, with an overall reduction in error and minimal trend change, indicating the effectiveness of the proposed calibration method. Figure 19(c) compares the distribution of attitude angle errors before and after calibration, demonstrating a significant overall reduction in absolute attitude angle errors.

Table X provides statistical data for the experimental results of attitude angles using the RBF model calibration, comparing errors before and after calibration. From Table 10, it can be observed that the average error of attitude angles decreased from 0.0727° to 0.0389°. The maximum attitude angle error decreased from 0.1607° to 0.085°, and the minimum attitude angle error decreased from 0.0114° to 0°. The global average accuracy of attitude angles improved by 46.46%.

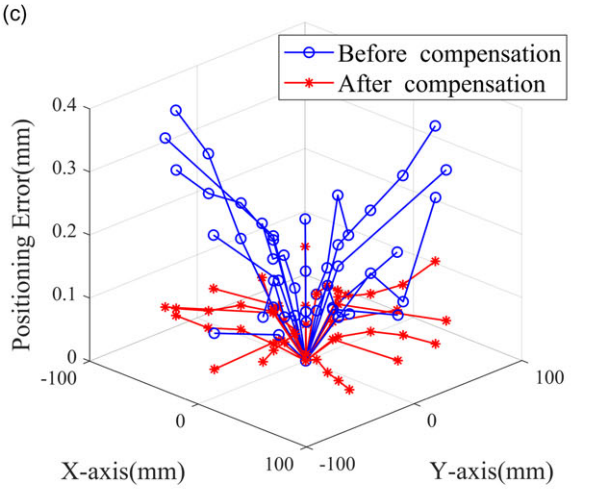
https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press



Error distribution chart before compensation



Error distribution chart after compensation



Comparison of error before and after RBF prediction point compensation

Figure 18. *Positioning error distribution.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

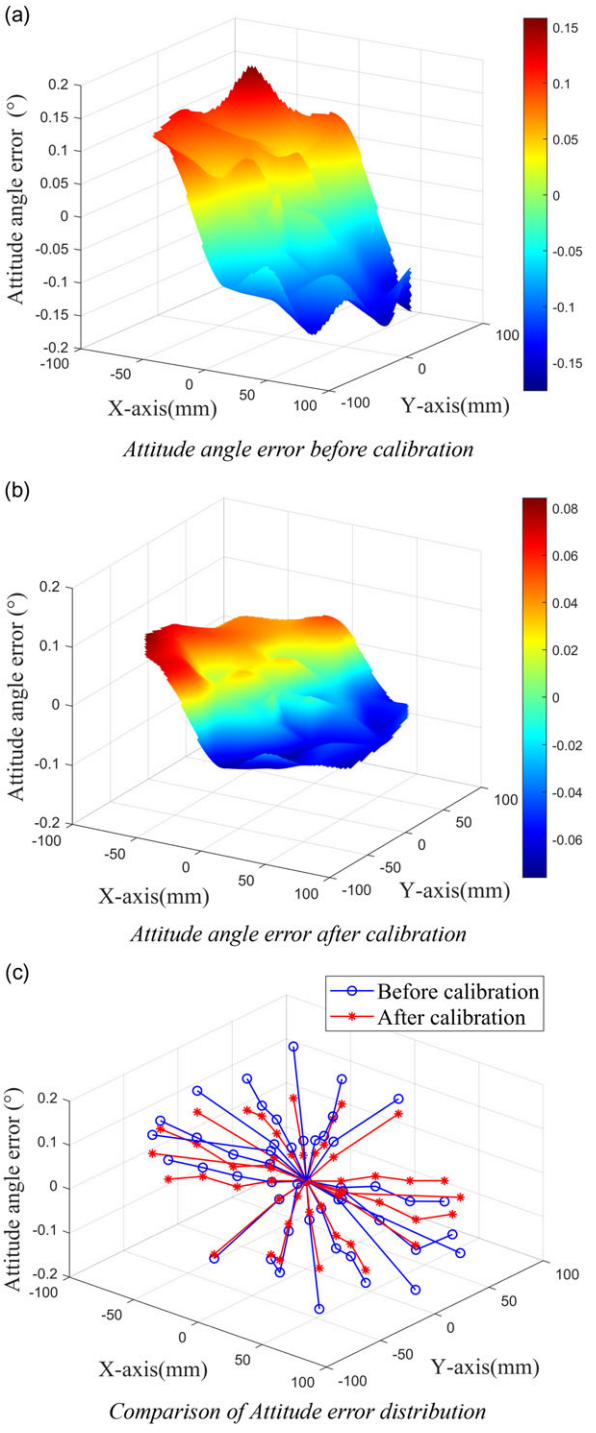


Figure 19. *Attitude angle error distribution of the moving platform.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table XI. *Circular tracking results.*

Order number	Target center/ mm		Actual center of the circle/ mm		Center deviation/ mm		Target radius/ mm	Real radius/ mm	Radius deviation/ mm
Coordinate	X	Y	X	Y	X	Y			
Round 1	50	0	49.9927	0.037	0.0073	−0.037	50	49.9836	0.0164
Round 2	0	50	−0.0693	50.0415	0.0693	−0.0415	50	50.128	−0.128
Round 3	−50	0	−50.0479	−0.058	0.0479	0.058	50	50.1541	−0.1541
Round 4	0	−50	0.0223	−50.0364	−0.0223	0.0364	50	50.0987	−0.0987
average					0.0367	0.0432			0.0993

4.4. Circular tracking experiment

To evaluate the performance of the calibrated platform within its working space, a trajectory tracking experiment was conducted on the planar 3-PRR parallel robot platform. As depicted in the Fig. 20(a), four circular trajectories with a radius of 50 mm were chosen in the working space of the moving platform. The experiment utilized consistent velocity and acceleration settings, which were subsequently measured and analyzed.

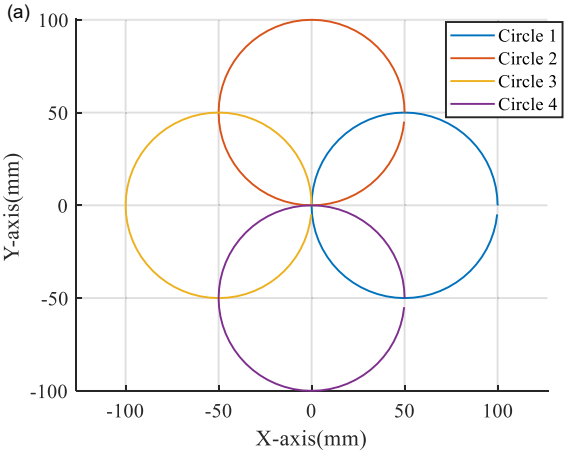
As depicted in the Fig. 20(b) and Fig. 20 (c), the circular trajectory tracking error graph clearly indicates that the tracking performance after calibration is significantly improved compared to that before calibration. The errors near the center of the working space are minimal, while the errors near the boundaries of the working space are more pronounced, consistent with the previously observed error distribution pattern of the mechanism. As per the statistics presented in the following Table XI, the average coordinate deviation for the circular track is 0.0566 mm, and the average radius deviation is 0.0993 mm.

5. Summary and future work

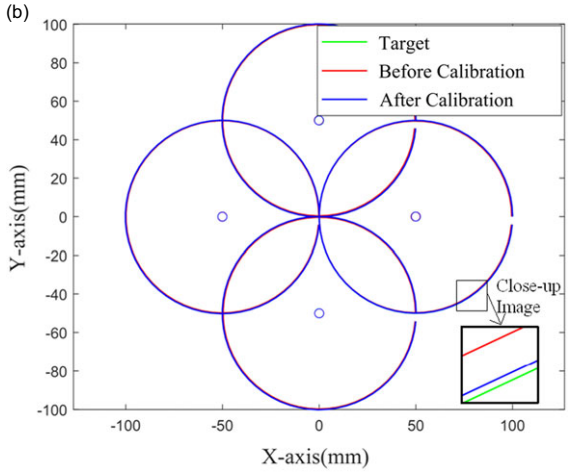
In this study, a global calibration method for a planar parallel platform considering joint errors is proposed. Initially, the experimental subject is a self-designed planar 3-PRR parallel robot mechanism. Calibration is conducted by employing the least squares method to solve for error parameters. The experimental results indicate that the local average positioning accuracy of the end moving platform has improved by 88.6% after calibration, and the attitude angle average accuracy has increased by 41.64%. However, research analysis reveals that methods based on mathematical error models cannot account for all joint errors. Additionally, the lack of high discernibility for some error sources limits this method to improving only the local accuracy of the robot's workspace. To achieve global calibration, two methods based on the distance-inverse law and RBFNN models are introduced, predicting global errors of the robot using a data-driven approach. Experimental results demonstrate enhancements in the average positioning accuracy of the parallel robot mechanism by 59.16% and 63.05%, and the attitude angle average accuracy by 40.99% and 46.46%, respectively. Finally, circular trajectory tracking experiments further validate the effectiveness of the calibration.

In this paper, the proposed algorithms effectively improve the positioning accuracy of the 3-PRR parallel robot from the perspective of kinematics, but don't consider the influence of factors such as the load of the moving platform, the collision, and friction between the joints. In the future, we will improve the positioning accuracy of the parallel robot from the perspective of dynamics.

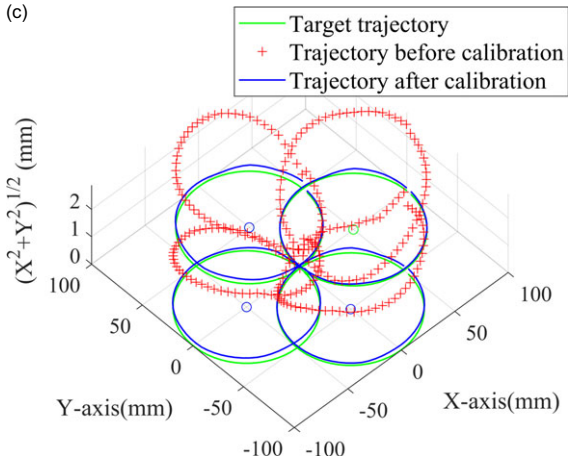
https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press



Ideal circle plot



2D map of circular trajectory tracking error



3D map of circular trajectory tracking error

Figure 20. *Circular trajectory tracking error.*

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Author contributions. Qinghua Zhang, Lingbo Xie, and Qinghua Lu (Corresponding author) conceived and designed the study. Qinghua Zhang, Huaming Yu, and Lingbo Xie collected data and wrote the article gathering. Weilin Chen performed statistical analyses.

Financial support. This work was supported by Guangdong Basic and Applied Research Foundation (No. 2021B1515120017, 2020B1515120070, 2019A1515111027), the National Natural Science Foundation of China (Grant No. 62273097, 52275010), and Research Projects of Universities in Guangdong Province (Grant Nos. 2021KTSCX118, 2020KCX TD015).

Competing interests. The authors declare no competing interests exist.

Ethical approval. None.

References

[1] C. Tsai, C. Yu, P. Yeh and C. Lan, "Parametric joint compliance analysis of a 3-UPU parallel robot," *Mech Mach Theory* **170**, 104721 (2022).
[2] T. Olsson, M. Haage, H. Kihlman, R. Johansson, K. Nilsson, A. Robertsson, M. Björkman, R. Isaksson, G. Ossbahr and T. Brogårdh, "Cost-efficient drilling using industrial robots with high-bandwidth force feedback," *Robot Comp-Int Manuf* **26**(1), 24–38 (2010).
[3] R. Kelaiaia, A. Chemori, A. Brahmia, A. Kerboua, A. Zaatri and O. Company, "Optimal dimensional design of parallel manipulators with an illustrative case study: A review, "*Mech Mach Theory* **188**, 105390 (2023).
[4] G. Chen, L. Kong, Q. Li and H. Wang, "A simple two-step geometric approach for the kinematic calibration of the 3-PRS parallel manipulator," *Robotica* **37**(5), 837–850 (2019).
[5] W. Gao, Q. Huang, R. Luo and Y. Zhang, "An improved minimal error model for the robotic kinematic calibration based on the POE formula," *Robotica* **40**(5), 1607–1626 (2022).
[6] Z. Li, S. Li and X. Luo, "Efficient industrial robot calibration via a novel unscented kalman filter-incorporated variable step-size Levenberg-Marquardt algorithm," *IEEE Trans Instru Meas* **72**, 1–12 (2023).
[7] H. Cao, H. Nguyen, T. Tran, H. Tran and J. Jeon, "A robot calibration method using a neural network based on a butterfly and flower pollination algorithm," *IEEE Trans Ind Electron* **69**(4), 3865–3875 (2022).
[8] T. Sun, B. Lian, J. Zhang and Y. Song, "Kinematic calibration of a 2-doF over-constrained parallel mechanism using real inverse kinematics," *IEEE Access* **6**, 67752–67761 (2018).
[9] A. Brahmia, R. Kelaiaia, O. Company and A. Chemori, "Kinematic sensitivity analysis of manipulators using a novel dimensionless index," *Robot Auto Syst* **150**, 104021 (2022).
[10] H. Ye, J. Wu and T. Huang, "Kinematic calibration of over-constrained robot with geometric error and internal deformation," *Mech Mach Theory* **185**, 105345 (2023).
[11] X. Luo, F. Xie, X. Liu and Z. Xie, "Kinematic calibration of a 5-axis parallel machining robot based on dimensionless error mapping matrix," *Robot Comp-Int Manuf* **70**, 102115 (2021).
[12] A. Brahmia, A. Kerboua, R. Kelaiaia and A. Latreche, "Tolerance synthesis of delta-like parallel robots using a nonlinear optimisation method," *Applied Sciences* **13**(19), 10703 (2023).
[13] H. Ye, J. Wu and D. Wang, "A general approach for geometric error modeling of over-constrained hybrid robot," *Mech Mach Theory* **176**, 104998 (2022).
[14] F. Li, Q. Zeng, K. F. Ehmann, J. Cao and T. Li, "A calibration method for overconstrained spatial translational parallel manipulators," *Robot Comp-Int Manuf* **57**, 241–254 (2019).
[15] X. Chen, Q. Zhang and Y. Sun, "Non-kinematic calibration of industrial robots using a rigid-flexible coupling error mo-del and a full pose measurement method," *Robot Comp-Int Manuf* **57**, 46–58 (2019).
[16] W. Zhu, W. Qu, L. Cao, D. Yang and Y. Ke, "An off-line programming system for robotic drilling in aerospace manufacturing," *Int J Adv Manuf Tech* **68**(9-12), 2535–2545 (2013).
[17] B. Li, W. Tian, C. Zhang, F. Hua, G. Cui and Y. Li, "Positioning error compensation of an industrial robot using neural networks and experimental study," *Chinese J Aeron* **35**(2), 346–360 (2022).
[18] H. Liu, Z. Yan and J. Xiao, "Pose error prediction and real-time compensation of a 5-DOF hybrid robot," *Mech Mach Theory* **170**, 104737 (2022).
[19] H. Nguyen, P. Le and H. Kang, "A new calibration method for enhancing robot position accuracy by combining a robot model–based identification approach and an artificial neural network–based error compensation technique," *Adv Mech Eng* **11**(1), 2072051381 (2019).
[20] L. Bo, T. Wei, Z. Chufan, H. Fangfang, C. Guangyu and L. Yufei, "Positioning error compensation of an industrial robot using neural networks and experim-ental study," *Chinese J Aeronaut* **35**(2), 346–360 (2022).
[21] Y. Miao, D. Zhijiang, S. Lining and D. Wei, "Optimal design, modeling and control of a long stroke 3-PRR comcspliant parallel manipulator with variable thickness flexure pivots," *Robot Comp Int Manuf* **60**, 23–33 (2019).
[22] X. Huo, B. Lian, P. Wang, Y. Song and T. Sun, "Dynamic identification of a tracking parallel mechanism," *Mech Mach Theory* **155**, 104091 (2020).

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

[23] D. Naderi, M. Tale-Masouleh and P. Varshovi-Jaghargh, "Gröbner basis and resultant method for the forward displace-ment of 3-doF planar parallel manipulators in seven-dimensional kinematic space," *Robotica* **34**(11), 2610–2628 (2016).
[24] J. si Mo, *Research On Plane 3-PRR Parallel Positioning System under Micro-Nano Operating Environment* (South China University of Technology, 2016).
[25] Y. Zhao, "Singularity, isotropy, and velocity transmission evaluation of a three translational degrees-of-freedom parallel robot, "*Robotica* **31**(2), 193–202 (2013).
[26] Y. Ning, T. Li, W. Du, C. Yao, Y. Zhang and J. Shao, "Inverse kinematics and planning/control co-design method of redundant manipulator for precision operation: Design and experiments," *Robot Comp-Int Manuf* **80**, 102457 (2023).
[27] L.-B. Xie, Z.-C. Qiu and X.-M. Zhang, "Development of a 3-PRR precision tracking system with full closed-loop measureme-nt and control," *Sensors* **19**(8), 1756 (2019).
[28] Y. Chen, K. He, M. Jiang, X. Wang and L. Xie, "Kinematic calibration and feedforward control of a heavy-load manipulator using parameters optimization by an ant colony algorithm," *Robotica* 1–29 (2024).
[29] X. Zhao and Q. Luan, "Kinematic modeling and error analysis of the 3-RRRU parallel robot," *Mech Design Manuf* (1), 274–276 (2020).
[30] R. Miranda-Colorado and J. Moreno-Valenzuela, "Experimental parameter identification of flexible joint robot manipulators," *Robotica* **36**(3), 313–332 (2018).
[31] K. Sourabh and T. J.C., "Forward kinematics solution for a general Stewart platform through iteration based simulation," *Int J Adv Manuf* **126**(1-2), 813–825 (2023).
[32] Q. Rui, H. Huijie, X. Keke, L. Bingchuan, L. Sha, H. Jingping, B. Shijie and Y. Jiakuan, "Prediction on the combined toxicities of stimulation-only and inhibition-only contaminants using improved inverse distance weighted interpolation," *Chemosphere* **287**(Pt 3), 132045 (2021).
[33] C. B. Eduardo, G. L. Luis and B. G. Marcela, "Geometric techniques for robotics and HMI: Interpolation and haptics in conformal geometric algebra and control using quaternion spike neural networks[J]," *Robot Auton Syst* **104**, 72–84 (2017).
[34] X. Zhang and L. Zeng, "Considered the kinematic calibration of the 3-RRR parallel mechanism with the back gap of the reducer [J]," *J South China Univ Tech* **44**(7), 47–54 (2016).
[35] L. Tong, C. Sheng, Y. Po, Z. Yunpeng and C. J. Harris, "Efficient adaptive deep gradient RBF network for multi-output nonlinear and nonstationa-ry industrial processes," *J Process Contr* **126**, 1–11 (2023).
[36] C. Eva, G. Harris, P. Nikos and T. Theodoridis, "Particle swarm optimization and RBF neural networks for public transport arrival time prediction using GTFS data," *Int J Inform Manage Data Insigh* **2**(2), 10086 (2022).
[37] M. Devesh and C. Amrita, *Reuse Estimate and Interval Prediction Using MOGA-NN and RBF-NN in the Functional Paradigm* (Science of Computer Programming, 2021).
[38] L. Guosheng, M. Shuaichao, Z. Dequan, Y. Leping, Z. Weihua and W. Zeping, "An efficient sequential anisotropic RBF reliability analysis method with fast cross-validation and parallelizability," *Reliab Eng Syst Safe* **241**, 1–14 (2024).
[39] H. Dongyang, C. Jinshi, Z. Han, S. Yiran and W. Tongyang, "Intelligent prediction for digging load of hydraulic excavators based on RBF neural network," *Measurement* **206**, 112210 (2023).

Appendix

Table XII. *The meaning of the given symbols.*

Symbol	Meaning
x_{A_i}, y_{A_i}	The coordinates of Point A on Link i in a fixed coordinate system.
x_{C_i}, y_{C_i}	The coordinates of Point C on Link i in a fixed coordinate system.
x, y, φ	The pose coordinates of the robot's moving platform.
$\delta x, \delta y, \delta\varphi$	The pose coordinate error of the moving platform.
$\alpha_1, \alpha_2, \alpha_3$	The installation angle of the fixed platform's guide rails.
$\delta\alpha_1, \delta\alpha_2, \delta\alpha_3$	The installation angle error of the fixed platform's guide rails.
$\beta_1, \beta_2, \beta_3$	The angle between $O_C C_i$ and the positive direction of the X-axis of the moving coordinate system.
$\delta\beta_1, \delta\beta_2, \delta\beta_3$	The installation angle error of the moving platform.
r_1, r_2, r_3	The distance of the coordinate system $O_c C_i$ of the moving platform.
$\delta r_1, \delta r_2, \delta r_3$	The distance error of the coordinate system $O_c C_i$ of the moving platform.
S_1, S_2, S_3	The length of the driven rods.

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press

Table XII. *Continued.*

Symbol	Meaning
$\delta S_1, \delta S_2, \delta S_3$	The length error of the driven rod.
l_1, l_2, l_3	The driving inputs.
$\delta l_1, \delta l_2, \delta l_3$	The driving input error.
R_1, R_2, R_3	The distance of the fixed coordinate system $O_A A_i$.
$\delta R_1, \delta R_2, \delta R_3$	The distance error of the fixed coordinate system $O_A A_i$.
$\theta_1, \theta_2, \theta_3$	The angle between the driven rod $B_i C_i$ and the line segment $A_i B_i$.
$\delta \boldsymbol{X}$	Pose error vector of the moving platform.
$\boldsymbol{J}v$	The velocity Jacobian matrix.
$\boldsymbol{J}e$	Error propagation matrix
$\boldsymbol{P_0}$	Error vector at the interpolation point.

Table XIII. *The meaning of the given symbols(continued).*

Symbol	Meaning
$\boldsymbol{P_i}$	The error vector at an unknown point.
c_j	Prototype centers of the neurons in the hidden layer.
ϕ_j	Gaussian function
δ_j	Spread parameter
R^2	Determination coefficients
y_e	Expected values
y_p	Predicted values

Cite this article: Q. Zhang, H. Yu, L. Xie, Q. Lu and W. Chen (2024). "Study of a global calibration method for a planar parallel robot mechanism considering joint error", Robotica **42**, 2589–2618. https://doi.org/10.1017/S0263574724000973

https://doi.org/10.1017/S0263574724000973 Published online by Cambridge University Press