Research Article

A Robot Calibration Method Based on Joint Angle Division and an Artificial Neural Network

Zhirong Wang,1 Zhangwei Chen,1 Yuxiang Wang,1 Chentao Mao,1 and Qiang Hang2

1School of Mechanical Engineering, Zhejiang University, Hangzhou 310000, China
2Changzhou Institute of Inspection, Testing, Standardization and Certification, Changzhou 213000, China

Correspondence should be addressed to Zhangwei Chen; zhangweichen@econ-group.com

Received 13 December 2018; Revised 7 February 2019; Accepted 18 February 2019; Published 4 March 2019

Academic Editor: Marcello Pellicciari

Copyright © 2019 Zhirong Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Robot calibration is used to improve the accuracy of the kinematic model to achieve the higher positioning accuracy within the workspace. Due to some nongeometrical reasons such as joint and link flexibility, the errors are unevenly distributed in the workspace. In this case, it is difficult for the existing methods used to improve the absolute positioning accuracy to achieve good results in each region, especially for robots with large self-weights. In this paper, a novel calibration method is proposed, which deals with joint deflection dependent errors to enhance the robot positioning accuracy in the whole workspace. Firstly, the joint angle workspace is divided into several local regions according to the mass distribution of the robot. Then, its geometric parameters are modeled and identified using the Denavit–Hartenberg (DH) model in each region and in the whole workspace separately. Since the nongeometric error sources are difficult to model correctly, an artificial neural network (ANN) is applied to compensate for the nongeometric errors. Finally, the experiments using an 8 degree-of-freedom (DOF) engineering robot are conducted to demonstrate the validity of the proposed method. The combination of the joint angle division and ANN could be an effective solution for the robot calibration, especially for one with a large self-weight.

1. Introduction

With the popularization of robot applications, robot offline programming methods are widely used [1, 2], which greatly depend on the absolute positioning accuracy of the robots. The deviation between the nominal model and the physical robot, caused by various reasons such as the machining tolerance and assembly tolerance, significantly reduces the positioning accuracy of a robot. The phenomenon can be improved by calibration, in which a group of more accurate model parameters is identified, so as to build a better kinematic model for the robot.

Robot calibration has attracted great attention in academia and industry. Current researches mainly focused on the error source modeling of robots [3–5]. The errors were mainly divided into geometric errors such as the link length and nongeometric errors such as the gear backlash. Most of the previous work focused on improving the kinematic models and compensating for geometric errors while ignoring the nongeometric errors [6]. The effect caused by the nongeometric errors is simply assumed to be tiny. Stone et al. established the mapping of the kinematic errors of each single joint to the end position using the 6-parameter model and solved the model by using the least square method [7, 8]. Chen established the product-of-exponential (POE) model for the kinematic errors of a robot, which avoided the singular value problem of the error models [9]. Tan et al. combined the screw theory with the distance error model to simulate and verify the calibration of the robots [10]. These model-based methods mentioned above could model many error sources accurately and exhibited a fast and good computing power. However there are also many error sources that cannot be modeled accurately; the conventional model-based methods cannot compensate for the nongeometric errors.

In fact, the nongeometric errors still play a significant role in the robot positioning accuracy and the nongeometric error sources cannot be ignored [11]. The errors caused by robot compliance account for 8%–10% as well as other nongeometric errors such as gear transmission errors [12].
Some other researches focused on the compensation for nongeometric errors of robots through the model-free methods. Judd and Kaminski observed experimentally many error sources such as geometric errors, gear errors, servo error, and thermal change errors and they proposed a method which adds a homogeneous matrix at the robot end plate to improve the robot accuracy [13]. A comprehensive model incorporating the joint deflection model and the kinematic model was built to compensate for the geometric errors and nongeometric errors [14]. In addition, some methods which aimed at describing the relationship between the robot joint readings and its position errors were worked out. Some optimization methods, such as genetic programming (GP) and the artificial neural network (ANN), were used for robot calibration. A formula was built based on the ANN to describe the relationship between the robot joint angle and its corresponding joint errors [15]. GP was used to calibrate the robot by optimizing the fitness function [16]. However, the training data used in the ANN and GP obtained from the idealized inverse kinematics of the robot was inaccurate. For some robots with heavy self-weights, the nongeometric errors such as joint deflection cannot be ignored. And the positioning errors caused by the nongeometric errors were not uniform. The relationship between positioning errors in different spaces was discussed and a flexibility error model was proposed [17]. A dynamic parameter compensating method based on the spatial grid was proposed [18]. As the robot position came into a different grid, a different group of the DH parameters was adopted. However, these two methods were both performed in the Cartesian coordinate system. Since the different joint configuration can come to the same spatial location, these techniques cannot acquire the high positioning accuracy in the whole workspace.

In this paper, a three-stage calibration method which can combine the advantages of the above models is proposed. Firstly, the joint angle workspace of the robot is divided into several local regions according to its mass distribution, whose innovation lies in avoiding the kinematic errors caused by the different mass distribution in the different angle configuration of the robot. Secondly, the DH model combined with the distance error model will be used to identify and compensate for the geometric errors in the local region and in the whole workspace separately. The introduction of the distance error model can avoid the transformation of the different coordinate systems and simplify the computing. At last, the residual errors will be compensated for through the ANN models instead of the complex modeling for the nongeometric errors. In order to demonstrate the validity of the method, a series of experiments is conducted on an 8 degree-of-freedom (DOF) engineering robot. Compared to conventional calibration methods, the calibration result results in a better positioning accuracy.

2. The Advanced Robot Calibration System

Industrial robots generally work through online teaching owing to lacking the high absolute positioning accuracy which is mainly caused by the serial structure of the robot, manufacturing and assembly process, and environment factors. The calibration method is based on the advanced robot calibration system, as shown in Figure 1. To increase the absolute positioning accuracy of the robot, a laser tracker is used to measure the real-time position of the tool point in the measuring coordinate system. The laser tracker can track the motion of the robot owing to its high sampling frequency. Taking the measuring position data and the configuration of the robot encoder as the input, the compensation model can correct the errors in the whole workspace. In the process of calibration, some important factors may influence the final calibration result. Firstly, modeling for the deflection of the link is a complex process and a large amount of calculation is consumed; secondly, the transformation between different coordinate systems may cause accumulated calculating errors. The method proposed in this paper provides available solutions to solve the problems mentioned above.

3. Methods Based on Joint Angle Division and an ANN

This section is organized as follows. Section 3.1 analyzes the division principle and divides the joint angle workspace of the robot into different regions. Section 3.2 builds the kinematic model of the robot and identifies the geometric parameters
using the DH model. Section 3.3 gives the nongeometric calibration method using the ANN.

3.1. Division of the Joint Angle Workspace. The parameter errors of the robot in the joint angle workspace are unevenly distributed due to factors such as the flexibility of the links and the motor control. The change in the compliance of the link is a continuous process that simplifies the flexible joint into a linear torsion spring model. The angle of joint deflection is proportional to the torque exerted on its axis.

\[ \Delta \theta = C_\theta \tau, \quad (1) \]

where \( \Delta \theta \) is a flexible deflection angle caused by either the load quality or the weight of the mechanical arm, \( C_\theta \) is the flexibility coefficient, \( \tau \) is the torque exerted on its axis.

Due to the influence of the robot weight and load, the errors of the joint angles will be relatively large with the change of joint configuration. Since there exist some deflections in the links of the robots, the kinematic parameters of the robot will also change to some extent. Therefore, the parameter errors can be expressed as a function of the location of the end effector given by

\[ \Delta p = (\Delta \theta, \Delta d, \Delta a, \Delta \alpha) = g(x, y, z), \quad (2) \]

where \( \Delta p \) is the error vector of the DH parameters, \((\Delta \theta, \Delta d, \Delta a, \Delta \alpha)\) denotes the error vector of the joint angle, the error vector of the link length, the error vector of the link offset, and the error vector of the link twist angle, respectively, and \((x, y, z)\) is the position vector at the end of the robot. According to (2), the parameter errors are related to the location of the end effector. For the same point in the Cartesian coordinate frame, there may exist several different configurations. It is difficult to establish error models in rectangular coordinates. But the joint angle configuration is unique in the whole workspace. Therefore, the problem can be solved in the joint angle workspace and the transformed formula is given by

\[ \Delta p = (\Delta \theta, \Delta \alpha, \Delta a, \Delta \alpha) = f(\theta_1, \theta_2, \theta_3, \ldots \theta_{n-1}, \theta_n), \quad (3) \]

where \( q = (q_1, q_2, q_3, \ldots q_n) \) denotes the values of the joint variables and \( n \) is the DOF number. According to formula (3), if the joint variables of the robot are determined, the values of the parameter error will also be constant. It is assumed that the parameter errors are \( \Delta p_1 \) and \( \Delta p_2 \), respectively, when the joint variables are \( q_1 = (\theta_1^1, \theta_2^1, \theta_3^1, \ldots, \theta_{n-1}^1, \theta_n^1) \) and \( q_2 = (\theta_1^2, \theta_2^2, \theta_3^2, \ldots, \theta_{n-1}^2, \theta_n^2) \). The difference between the two positions can be given by

\[ E = \|\Delta p_1 - \Delta p_2\| < \varepsilon, \quad |q_1 - q_2| \rightarrow 0, \quad (4) \]

where \( E \) is the norm difference between \( \Delta p_1 \) and \( \Delta p_2 \) and \( \varepsilon \) is a selected threshold. When \( |q_1 - q_2| \) is tiny enough, a proper \( \varepsilon \) satisfying \( E < \varepsilon \) can be found.

Based on the above deduction, it is assumed that when the robot is in different positions, the parameter errors are also different. When the self-weight of a robot cannot be ignored, the joint deflections will be different due to different mass distribution in different positions. In this paper, the joint angle workspace will be divided into several local regions according to the difference of the mass distribution. Every local region has its own parameter errors. When the robot moves with a different joint configuration, it will compensate for the position error with different kinematic parameters. The smaller the regions are divided, the more accurate the parameter errors are and the greater the computation cost will be.

3.2. Geometric Parameters Identification

3.2.1. Kinematic Model of the Robot. The kinematic model of the robot in this paper is built according to the DH convention [19], as shown in Figure 2.

The coordinate frames are fixed on the joint axis of the robots. Two adjacent joint coordinate frames are connected by a homogeneous transformation matrix \([i^{-1}]_i T\) given by

\[ i^{-1} T = trotz(\theta_i) * transal(a_i, 0, 0) * transl(0, 0, d_i) \]

\[ * trotx(a_i), \quad (5) \]
where \((\theta_i, a_i, d_i, \alpha_i)\) denotes the joint variable, the link length, the link offset, and the link twist angle, respectively, and \(i\) denotes the sequence number of the joint. \(T_{\text{rot}}(*)\) is a \((4x4)\) matrix of the pure rotational transformations about an axis and \(T_{\text{trans}}(*)\) is a \((4x4)\) matrix of the translational ones along an axis \([19]\).

The position of the end effector \(\text{meas}\)T can be obtained by multiplying the corresponding transformation matrix given by

\[
\text{meas}\ T = \text{base}\ T \cdot \prod_{i=1}^{n} \text{trans}\ (a_0, b_0, c_0) \cdot \text{trotz}\ (\beta_0) \cdot \text{trotx}\ (\alpha_0) \cdot \text{trowy}\ (\gamma_0),
\]

where \(a_0, b_0, c_0\) are the translational parameters along the frame axis, \(\alpha_0, \beta_0, \gamma_0\) are the rotational parameters about the frame axis, \(n\)T represents the position of the tool related to the robot end effector given by

\[
s\text{tool}\ T = \text{trans}\ (a_{\text{tool}}, b_{\text{tool}}, c_{\text{tool}}),
\]

where \(a_{\text{tool}}, b_{\text{tool}}, c_{\text{tool}}\) are the translational parameters along the frame axis. According to the above formulas, the center point of the robot tool described in the robot base coordinate system is

\[
P = F(\theta, a, d, \alpha),
\]

where \(P\) represents the position of the TCP. \((\theta, a, d, \alpha)\) represents the kinematic parameters’ vector of the links.

3.2.2. Distance Error Model. The actual position of the TCP measured in the robot base coordinate system \{BASE\} is

\[
p_{\text{base}} = [x_{\text{base}}, y_{\text{base}}, z_{\text{base}}]T.
\]

And the position of the TCP represented in the measuring coordinate system \{MEAS\} is

\[
p_{\text{meas}} = [x_{\text{meas}}, y_{\text{meas}}, z_{\text{meas}}]T.
\]

The distance error of the robot means the deviation between the actual distance and the command distance \([20]\). The distance error between any two points \(\Delta d_k^{k+1}\) is given by

\[
\Delta d_k^{k+1} = ||p_{\text{base},k+1} - p_{\text{meas},k}|| - ||p_{\text{base},k} - p_{\text{base},k}||,
\]

where \(||*||\) is the norm of a vector. \(p_{\text{meas},j}\) is the position of the \(j\)th point in the measuring coordinate system, and \(p_{\text{base},i}\) is the position of the \(i\)th point in the robot base coordinate system, \(i \in Z_+\). The relationship between distance error and position error is given by

\[
\Delta d_k^{k+1} = \frac{(p_{\text{base},k+1} - p_{\text{base},k})^T}{||p_{\text{base},k+1} - p_{\text{base},k}||} (d_{p,k+1} - dp_k),
\]

where \(dp\) is the position error vector of some points \([21]\).

The introduction of the distance error model can avoid the transformation of different coordinate frames and simplify the measurement process.
output layer. Neurons in the input layer are used to receive input from the world outside. The hidden layer in the middle is used to transform and process information and eventually outputs information through the output layer. Every neuron in the ANN model is connected to some neurons in the preceding layer with dynamic and different weights. The structure of a neuron in the hidden layer is as shown in Figure 3.

The algorithm updates the neuron network weights according to the strategy given by

\[ W_{ik}(t+1) = W_{ik}(t) + \beta g(t), \]  

where \( W_{ik}(t) \) is the connection weight from neuron \( i \) to neuron \( k \) at time \( t \), \( \beta \) is the learning rate, and \( g(t) \) is the gradient at the time \( t \). The number of neurons in the hidden layer has a great influence on the training effect of the ANN [24]. Suppose \( m, n, \) and \( f \) are the numbers of the neurons in the input layer, the output layer, and the hidden layer, respectively. Their relationship is given by

\[ f = a \sqrt{m+n}, \]  

where \( a \) is a constant satisfying \( 1 < a < 10 \). Compared with the error compensation obtained by modeling for noneometric errors, the error compensation by BP neural network approximation is more accurate [25]. In the calibration process for the robot, the neural network compensation method works as shown in Figure 4.

The neural network model must be trained before being used to compensate for the errors caused by the noneometric reasons. A supervised learning method has been selected to train the model in this paper. The joint encoder readings of the robot \( q = (\theta_1, \theta_2, \theta_3, \ldots, \theta_{n-1}, \theta_n) \) and the corresponding error vector \( \Delta E = [\Delta x, \Delta y, \Delta z] \) are taken as the training data.

The well trained ANN model can predict the corresponding position error vector \( \Delta E = [\Delta x, \Delta y, \Delta z] \) after we input a set of robot joint encoder readings \( q = (\theta_1, \theta_2, \theta_3, \ldots, \theta_{n-1}, \theta_n) \). At last, we compensate for the residual errors with the position error vector to reach the final position.

An ANN model with three layers has been constructed as shown in Figure 5. The input layer consists of eight nodes, which corresponds to the DOF number. According to (18), the hidden layer consists of 4 neurons. A tan-sigmoid function has been used as the activating function. The input value of this function is arbitrary, and the range of output value for any input is always between -1 and 1. The function image is shown in Figure 6. The output layer has three nodes, which correspond to three elements of the position error vector \( \Delta E = [\Delta x, \Delta y, \Delta z] \). In this layer, we use a linear function as the activating function. The process of compensating for the noneometric errors can be described as shown in Figure 7.

4. Experiments and Analysis

To demonstrate the effectiveness of the proposed method, the conventional method and the proposed method were used to calibrate an 8-DOF engineering robot separately. The self-weight of the robot is 5500 kg which makes the weight of the end load negligible. A FARO laser tracker has been adopted as the measuring equipment. The equipment and the scene are shown in Figure 8.

4.1. The Kinematic Model of the 8-DOF Robot. According to the structure of the robot, the transformation relationship between adjacent links is shown in Figure 9. The values of the nominal DH parameters are given in Table 1. According
Figure 5: A three-layer BP neural network.

Figure 6: The image of the tan-sigmoid function.

Figure 7: The process of compensating for the nongeometric errors.
to the DH convention [19], the nominal kinematic model was built.

4.2. Parameters’ Setting of the Joint Angle Division. Through the analysis for the mass distribution of the robot, the conclusion can be made that the link deflections make a great difference on the position accuracy. The sixth joint $J_6$ is a rotating joint which relates to the mass distribution of the three links in the rear. When $J_6$ moves in the range $(-180^\circ, 180^\circ)$, there exists a most obvious change in the deflection of the robot system. So, in order to minimize the impact of the joint $J_6$ on the position accuracy, the joint angle workspace will be divided into four local regions in this study. The scope of each region is given in Table 2.

4.3. Acquisition of the Position Data. The nominal parameters of the kinematic model have been described in Table 1. In order to identify the geometric parameters and train the BP neural network model, the robot is commanded to move its end effector to 400 different points. And the 400 points are made to cover the whole workspace that the end effector can reach. In order to validate the performance of the joint angle division, the 400 points are distributed evenly into 4 regions.

Owing to the shaking of the end effector, the laser tracker would not record the coordinate value of the point until it was stable. Once the robot reaches a point, the three-dimensional position values $P_i(x, y, z)$ ($i = 1, 2, \ldots 400$) of the point in the measuring system will be measured by the laser tracker. The corresponding joint values $R_i(\theta_1, \theta_2, d_3, \theta_4, \theta_5, \theta_7, d_8)$ ($i = 1, 2, \ldots 400$) can also be saved in the file.

4.4. Training of the Neural Network Model. After the calibrated DH parameters and the test data were obtained, the BP neural network model was established. Through our repeated experiments, the parameters’ setting of the model training was determined in detail. The concrete setting is given in Table 3.

4.5. Analysis of Experiments’ Results. In the practical calibration for the 8-DOF engineering robot, the distance error model was used to calibrate the parameters of the DH model in the whole workspace. The calibrated DH parameters were written back to the motion controller of the robot. After compensation for the geometric errors of the robot, the average position error has been reduced from 17 cm (before compensation) to 9 cm. The calibrated results obtained from the traditional method are shown in Table 4 and Figure 10.

As shown in Figure 10, the errors of some points still reach about 14 cm owing to the nongeometric errors. To demonstrate the validity of the joint angle division, two other different experiments were conducted. In the first experiment,
Table 1: The nominal DH parameters of the 8-DOF robot.

<table>
<thead>
<tr>
<th>No.</th>
<th>Joint type</th>
<th>( \theta_i )</th>
<th>( a_i )</th>
<th>( d_i )</th>
<th>( \alpha_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R</td>
<td>( \theta_1 )</td>
<td>300</td>
<td>0</td>
<td>-90</td>
</tr>
<tr>
<td>2</td>
<td>R</td>
<td>-90+( \theta_2 )</td>
<td>20</td>
<td>0</td>
<td>-90</td>
</tr>
<tr>
<td>3</td>
<td>P</td>
<td>180</td>
<td>80</td>
<td>5625+( d_3 )</td>
<td>-90</td>
</tr>
<tr>
<td>4</td>
<td>R</td>
<td>-90+( \theta_4 )</td>
<td>340</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>5</td>
<td>R</td>
<td>-90+( \theta_5 )</td>
<td>0</td>
<td>0</td>
<td>-90</td>
</tr>
<tr>
<td>6</td>
<td>R</td>
<td>-90+( \theta_6 )</td>
<td>191</td>
<td>411</td>
<td>90</td>
</tr>
<tr>
<td>7</td>
<td>R</td>
<td>( \theta_7 )</td>
<td>585</td>
<td>-604</td>
<td>-90</td>
</tr>
<tr>
<td>8</td>
<td>P</td>
<td>0</td>
<td>0</td>
<td>3850+( d_8 )</td>
<td>0</td>
</tr>
</tbody>
</table>

8-DOF robot positioning accuracy improvements after calibration

![8-DOF robot positioning accuracy improvements after calibration](image)

- ☒ - Before calibration
- ◊ - After calibration

Figure 10: The calibration results after compensation for geometric errors.

Table 2: The scope of each region.

<table>
<thead>
<tr>
<th>No.</th>
<th>Joint scope</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(0,90)</td>
<td>G1</td>
</tr>
<tr>
<td>2</td>
<td>(90,180)</td>
<td>G2</td>
</tr>
<tr>
<td>3</td>
<td>(-90,0)</td>
<td>G3</td>
</tr>
<tr>
<td>4</td>
<td>(-180,-90)</td>
<td>G4</td>
</tr>
</tbody>
</table>

In the second experiment, with the 100 points in each region taken as the input, the geometric errors of the robot were calibrated in each region. There were four groups of DH parameters which correspond to the four regions (G1, G2, G3, and G4). These four groups of DH parameters were integrated into the motion control algorithm. When the robot moved its end effector to a point, a group of DH parameters could be used to fix the position error.

As shown in Table 6 and Figure 12, the average position error has been reduced from 9 cm (after the geometric error compensation) to 7 cm (compensation through joint angle division) after the compensation through the joint angle division. At last, the same BP neural network model was used to estimate the nongeometric errors in the whole workspace. After the compensation for the nongeometric errors through the BP network model, the average position error has been reduced to 4.5 cm.

Obviously, the results obtained from the second experiment are better than those obtained from the first one.

the BP neural network model was used to estimate the non-geometric errors. For each point \( R(\theta_1, \theta_2, d_3, \theta_4, \theta_5, \theta_6, \theta_7, d_8) \) in the joint configuration, there existed the corresponding final nongeometric error \( \Delta E = [\Delta x, \Delta y, \Delta z] \). When the end effector of the robot moved to a point, the corresponding values of the nongeometric errors were added. As shown in Table 5 and Figure 11, the average position error has been reduced from 9 cm (after geometric error compensation) to 6.5 cm. Although the ANN compensates for nongeometric errors to some extent, the maximum value of the errors still reaches 10.22 cm.
Table 3: The parameter setting of model training.

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Number of samples</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>Number of validation samples</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>Network type</td>
<td>BP</td>
</tr>
<tr>
<td>4</td>
<td>Input layer</td>
<td>8</td>
</tr>
<tr>
<td>5</td>
<td>Hidden layer</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>Output layer</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Learning rate</td>
<td>0.1</td>
</tr>
<tr>
<td>8</td>
<td>Training times</td>
<td>1000</td>
</tr>
<tr>
<td>9</td>
<td>Training function</td>
<td>Tan-sigmoid</td>
</tr>
</tbody>
</table>

Table 4: The results calibrated by the traditional method (cm).

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Before compensation</td>
<td>17</td>
<td>12.52</td>
<td>21.37</td>
</tr>
<tr>
<td>After geometric error compensation</td>
<td>9</td>
<td>7.86</td>
<td>14.01</td>
</tr>
</tbody>
</table>

Table 5: The results calibrated by the ANN model directly (cm).

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>After geometric error compensation</td>
<td>9</td>
<td>7.86</td>
<td>14.01</td>
</tr>
<tr>
<td>Compensation through the BP network</td>
<td>6.5</td>
<td>5.12</td>
<td>10.22</td>
</tr>
</tbody>
</table>

Table 6: The results calibrated by the new method (cm).

<table>
<thead>
<tr>
<th></th>
<th>Average</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>After geometric error compensation</td>
<td>9</td>
<td>7.86</td>
<td>14.01</td>
</tr>
<tr>
<td>Compensation through joint angle division</td>
<td>7</td>
<td>5.01</td>
<td>9.22</td>
</tr>
<tr>
<td>Compensation through the BP network</td>
<td>4.5</td>
<td>3.79</td>
<td>6.12</td>
</tr>
</tbody>
</table>

8-DOF robot positioning accuracy improvements after calibration

**Figure II:** The calibration results after compensation through the ANN directly.
This demonstrates that the joint angle division is useful for improving the positioning accuracy.

As shown in Figure 13, the positioning accuracy of the robot is normally improved. The validation results show that the average position errors have been reduced from 17 cm (before compensation) to 4.5 cm (after compensation). And the calibration method proposed in this paper is better than the other two methods.

5. Conclusions

In this paper, an improved robot calibration method based on joint angle division and an ANN is proposed. To reduce the
influence of the link deflection on the positioning accuracy, the joint angle workspace is divided into several local regions and each region has its own group of DH parameters. The introduction of the distance error model can avoid the transformation between different coordinate systems and improve the calibration accuracy of the geometric errors. The BP neural network model is used to build the relationship between the joint angle configurations and the residual errors. The proposed method can avoid the overfitting problem in the pure BP network model and the complex modeling for the deflection.

In order to validate the performance of the proposed method, the calibration experiments on an 8-DOF engineering robot with a very large self-weight have been conducted. The practical calibration results demonstrate the effectiveness and correctness of the proposed method. Further researches focusing on the online real-time compensation for robots will be conducted in the future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors’ Contributions

Zhirong Wang and Yuxiang Wang conceptualized the study; Zhirong Wang created the methodology; Chentao Mao ran the software; Zhirong Wang, Yuxiang Wang, and Qiang Hang carried out the validation; Zhirong Wang prepared the original draft; Zhirong Wang and Zhangwei Chen reviewed and edited the manuscript; Zhangwei Chen acquired the funding.

Acknowledgments

This research was supported by the 2017 National Key R&D Program of China (No. 2017YFB1301400).

References

