

Research Article

The Inverse Solution Algorithm and Trajectory Error Analysis of Robotic Arm Based on MQACA-RBF Network

Xu Cheng ¹ and Ming Zhao ²

¹School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China

²School of Applied Technology, University of Science and Technology Liaoning, Anshan 114051, China

Correspondence should be addressed to Ming Zhao; as_zmyli@163.com

Received 12 September 2019; Revised 18 December 2019; Accepted 16 January 2020; Published 28 February 2020

Academic Editor: Shahram Payandeh

Copyright © 2020 Xu Cheng and Ming Zhao. 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.

In order to improve the position accuracy and trajectory accuracy of a 6R robotic arm, a robot arm inverse solution algorithm based on the MQACA- (improved quantum ant colony-) RBF network is proposed. This algorithm establishes the prediction model through the neural network and uses the quantum ant colony algorithm to optimize the output weight. In order to solve the problem that the quantum ant colony algorithm has low convergence precision and easy to fall into the local optimal solution in the inverse solution algorithm of the multifreedom robotic arm, improved measures such as 2-opt local optimization and maximum minimum pheromone limit and variation are adopted. By comparing the simulation results of the 6R robotic arm simulation results and the simulation results based on ACA, QACA, and RBF neural networks on the position and motion trajectory of the space point, the advantages in precision are obvious. This proves the feasibility and effectiveness of the scheme.

1. Introduction

In modern manufacturing, the robotic arm is a more important electromechanical integration device. It has a wide range of applications. The proportion of 6R robotic arms in these robotic arms is about 90% [1]. In order to adapt to different trajectory planning tasks, the robotic arm will be adjusted accordingly, which is closely related to the diversification of production. On the basis of dynamics and kinematics theory, the accuracy of the inverse solution [2] is a decisive factor for the accuracy and efficiency of the trajectory planning of the robot arm.

The inverse solution of the 6R robot arm is the basis and premise of the robot arm trajectory planning. Because the inverse solution equations have high dimensional and nonlinear characteristics, the solution is complex, and it is difficult to find accurate solutions. Traditional inverse solution can be roughly divided into algebraic methods [3–7] and geometric methods [8, 9]; due to poor results, it has been rarely used. In the more popular numerical iterative method at this stage [10–13], evolutionary, bionic and simulated annealing, neural networks, and other algorithms are more

common intelligent algorithms [14–16]. Although these methods have achieved remarkable results, they are still unsatisfactory and limited in terms of convergence effect and accuracy of solutions. Therefore, they need to be further explored and improved.

In the intelligent algorithm, the RBF neural network is a very important algorithm, which can approximate any nonlinear function within a finite set and arbitrary precision. Some scholars have realized the importance of the neural network and used it to solve the inverse solution of the 6R robotic arm to obtain a relatively ideal effect [17–19]. In this paper, an algorithm based on the MQACA- (improved quantum ant colony-) RBF network is proposed to optimize the output weight of the improved quantum ant colony, and good results are obtained.

2. 6R Mechanical Arm Kinematics Model

For the ZrARM arm, the number of degrees of freedom is six, and the joints are all rotating joints. The forward kinematics problem is solved and the ZrARM arm improved DH model coordinate system as shown in Figure 1 [20].

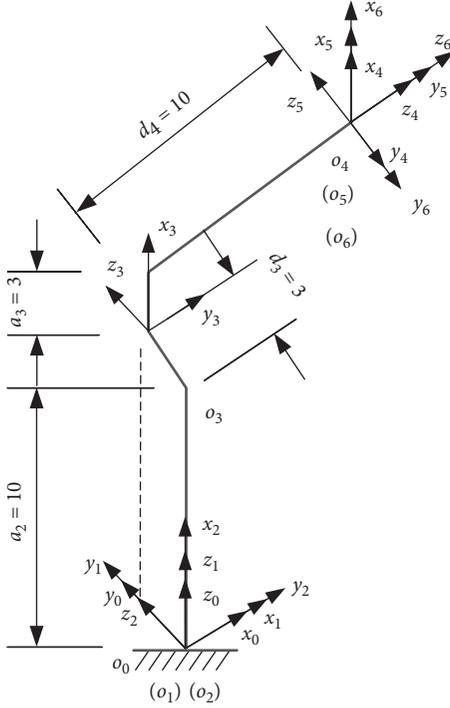


FIGURE 1: ZrARM arm improved DH model coordinate system.

From [21], it can be seen that the coordinate conversion matrix of the manipulator end joint coordinate system relative to the base coordinate system is

$${}^0_6T = {}^0_1T_1 {}^1_2T_2 {}^2_3T_3 {}^3_4T_4 {}^4_5T_5 {}^5_6T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

$[p_x, p_y, p_z]^T$ in this matrix represents the position of the end of the manipulator in the base coordinate system. The RPY Euler angle can be described by the following formula [21–23]:

$$\alpha = \arctan(a_y, a_x), \quad (2)$$

$$\beta = \arctan(a_x \cos a + a_y \sin a, a_z), \quad (3)$$

$$\gamma = \arctan(-n_x \sin a + n_y \cos a, o_y \cos a - o_x \sin a). \quad (4)$$

From the above formula, it can be seen that since the variables in (1) are all functions of the joint angle θ_i , the end position $p = [p_x, p_y, p_z, \alpha, \beta, \gamma]$ can be obtained in the case of θ_i determination.

3. Establishment of Prediction Model of Inverse Solution Based on MQACA-RBF Network

3.1. Establishment of the Prediction Model of Mechanical Arm Inverse Solution. In order to find the inverse solution of the robot arm, the inverse solution prediction model of the RBF neural network is established. In order to further improve

the inverse solution accuracy, the output weight value is optimized by improving the quantum ant colony algorithm, so that the error between the output and the actual output of the network is minimized.

As a kind of the neural network learning algorithm, the structure of the RBF neural network includes the input layer, hidden layer, and output layer. Figure 2 shows the structure diagram of the RBF neural network of the n -dof manipulator [20]. In the figure, it is assumed that the input of the RBF neural network is the end position variable x , $x = [p_x \ p_y \ p_z \ \alpha \ \beta \ \gamma]^T$, and the output variable is the inverse solution of the robot arm, that is, n joint angles y , $y = [\theta_1 \ \theta_2 \ \dots \ \theta_j \ \dots \ \theta_n]^T \in R^n$, where y is an n -dimensional vector, and the relationship between x and y is

$$y_n = \sum_{i=1}^k \omega_{ij} \exp\left(-\frac{1}{2\sigma^2} \|x_k - c_i\|^2\right), \quad 1 \leq n, \quad (5)$$

where

$$x_k = [p_x^k \ p_y^k \ p_z^k \ \alpha^k \ \beta^k \ \gamma^k]^T, \quad (6)$$

$$c_i = [c_{1i} \ c_{2i} \ \dots \ c_{hi}]^T,$$

where x_k —type k sample input parameters; c_i —the node center of the i th hidden layer function; h —number of nodes; ω_{ij} —weight from the implicit layer to output layer; y_n —the actual output of the n degree of freedom; and σ —variance of hidden layer functions.

Quantum Ant Colony Optimization Algorithm (QACA) is an operation that enables all ants to implement quantum bit coding method and updates by quantum revolving door. Therefore, its population number is relatively small, the algorithm performance is relatively high, and the convergence rate is fast. In addition, the optimization ability of the quantum ant colony is significantly enhanced, and the number of iterations is significantly reduced. Therefore, its vitality and scientific research values are very strong [24, 25]. Although its advantages are very prominent, there are still some problems in optimization efficiency, robustness, and convergence speed, which are closely related to large coding space and complex calculation.

In this paper, QACA and RBF neural networks are combined to complement each other's advantages and make full use of the efficient optimization capability of the quantum ant colony, and improvement measures are proposed to form the inverse solution algorithm of the mechanical arm based on the MQACA-RBF network. MQACA is used to optimize the output weight of the RBF neural network. Improvements to QACA include: (1) using the 2-opt local optimization algorithm to increase the diversity of path selection of the ant colony in the grid; (2) using maximum and minimum pheromone constraints to increase the feasibility of searching for better solutions and ensure the inspiration of experience for the ant colony; (3) on the basis of the optimal path of each group of ant colony, the mutation operation of the genetic algorithm is used to improve the quantum ant colony algorithm to avoid local optimum and ensure the accuracy and convergence of the algorithm.

3.2. MQACA Algorithm. The training sample of the inverse solution algorithm of the robotic arm can be obtained by positive kinematics formulas, as long as any joint angles $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$, and θ_6 are known, through formulas (1)–(4). You can get the corresponding position variables $p_x, p_y, p_z, \alpha, \beta$, and γ . Then, using the idea of MQACA theory [26–28], the output weight of each working area is sought, and the sample selection is based on this.

3.2.1. Coding Scheme. In order to improve the randomness of population initialization coding, the following coding scheme is used to obtain a set of digital codes for each grid point [29]. Let m ants arrive at the end point from the starting point to get the solution of the output weight value ω_n . The established decoding formula is

$$\omega_n = \omega_{ij} = (-1)^{\omega(i,1)} \cdot \sum_{d=2}^d \omega(i,d) \cdot 10^{-(d-1)}, \quad (7)$$

where $i = 1, 2, \dots, 9$, $j = 1, 2, \dots, (d \times n + 2)$.

Combine the output weight ω_n obtained by decoding to obtain a set of solutions for $\omega = [\omega_1 \ \omega_2 \ \omega_3 \ \dots \ \omega_n]^T$. In this group solution, n represents the spatial dimension (6R robotic arm $n = 6$), when the size of the ant colony does not change, and if the space occupied by the ant increases, the search space can be doubled and the convergence speed can be greatly increased [30].

3.2.2. Status Transfer Rule. In the grid established at each output weight of the RBF network, qubits can represent the quantum pheromones of the nodes in the quantum ant colony algorithm, and their conversion can be completed by the quantum rotation gate. The adjustment form of the quantum rotation gate can be defined as [31]

$$\begin{bmatrix} \alpha_i^{t+1} \\ \beta_i^{t+1} \end{bmatrix} = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{bmatrix} \begin{bmatrix} \alpha_i^t \\ \beta_i^t \end{bmatrix}. \quad (8)$$

In the formula, $i = 1, 2, \dots, m$, $[\alpha_i^{t+1}, \beta_i^{t+1}]^T$ is the probability amplitude of the i th qubit of the t iteration, and θ_i is the rotate angle of the i th qubit.

The probability of ant k moving from node i to j [29] is defined as

$$P_{(i,j;r,c)}^k = \frac{[\tau_{(i,j;r,c)}]^\alpha [\mu_{r,c}]^\gamma}{\sum_{l=1}^{10} [\tau_{(i,j;l,c)}]^\alpha [\mu_{l,c}]^\gamma}, \quad (9)$$

where $i, r = 1, 2, \dots, 9$; $c = j + 1$; $j = 1, 2, \dots, (d \times n + 2)$; $\tau_{(i,j;r,c)}$ is the amount of pheromone that exists between the $\omega_{(i,j)}$ cell point and the $\omega_{(r,c)}$ cell point of the ant k ; The information heuristic factor is represented by α ($\alpha \geq 0$), which reflects the importance of the trajectory relative to the path. The more paths the ant goes, the greater the value of α ($\alpha \geq 0$) in general is. The network node is represented by $\mu_{(r,c)}$. The quantum information intensity can be expressed as $\theta_{(r,c)}$. The following is its expression:

$$\mu_{r,c} = \frac{1}{|\alpha_{r,c}|^2}. \quad (10)$$

In this expression, $|\alpha_{r,c}|^2$ represents the probability that the quantum state of the j qubit collapses to $|0\rangle$ [32].

3.2.3. Determine the Fitness Function. It corresponds to a given training sample set $(x_{k1}, x_{k2}, \dots, x_{kn}) \rightarrow (t_{k1}, t_{k2}, \dots, t_{kn})$, $k = 1, 2, \dots, n$, where the training sample number is represented by K . In order to determine the fitness function, it is necessary to calculate the mean square root error of the actual output and expected output of the neural network first and to determine the adaptability function, i.e.,

$$f(ki) = \sqrt{\frac{1}{n} \sum_{k=1}^K \sum_{i=1}^n (d_{ki} - y_{ki})^2}, \quad (11)$$

where K —total number of training samples; d_{ki}, y_{ki} —target output and network operation results for unit i of sample k .

3.2.4. Local Search 2-Opt Algorithm. The 2-opt local optimization algorithm is used to increase the diversity of ant colony and accelerate the convergence rate of the quantum ant colony algorithm.

The 2-opt algorithm is an abbreviation of 2-optimization. In simple terms, it is the optimization of two elements, also known as 2-exchange [33].

The main purpose of the 2-opt algorithm is to randomly select an interval segment for optimization. This optimization is only about the optimization of the current situation, not the optimization of the global. The steps of the algorithm are shown in Figure 3

The approach obtained by this algorithm is locally optimal, that is, it is based on the number of iterations. Different results may occur. Therefore, it is not an absolutely accurate result. It is only a relatively correct result after optimization.

3.2.5. Global Update of Pheromones Based on Maximum and Minimum Pheromones. In the process of convergence, the ant colony algorithm may show that the pheromones on one path are much larger than the rest of the paths, resulting in convergence to the local optimal solution. The maximum and minimum pheromone ant colony algorithm limits the amount of information on each path to $[\tau_{\min}, \tau_{\max}]$, which can effectively avoid local stagnation in the search process [34].

Scientific setting τ_{\max} and τ_{\min} is crucial to the maximum and minimum pheromone ant colony algorithm. In the search process, the maximum pheromone growing on each iteration path is $1/f(ki)$, where $f(ki)$ can represent the current global optimal moderate function, in other words, the mean square root error of the real output and the desired output. When the optimal solution changes in the iterative process, τ_{\max} and τ_{\min} will change at the same time. The

inverse proportional relationship between τ_{\max} and f functions and ρ , in contrast, is a positive proportional relationship with “elite ants”. It can be seen that dynamic adjustments can be made to τ_{\max} and τ_{\min} using the following method. The adjustment strategy is divided into the following steps:

$$\tau_{\min}(t) = \frac{\tau_{\max}(t)}{20}. \quad (12)$$

- (a) Before the initial moment pheromones are not updated, the adjustment strategy is as follows:

$$\tau_{\max}(t) = \frac{1}{2(1-\rho)} \cdot \frac{1}{f(ki)}. \quad (13)$$

- (b) After the pheromones are updated, the adjustment strategy is as follows:

$$\tau_{\max}(t) = \frac{1}{2(1-\rho)} \cdot \frac{1}{f(ki)} + \frac{\sigma}{f(ki)}. \quad (14)$$

In the formula, σ represents the number of “elite ants.”

3.2.6. Variation Operation. On the basis of the maximum minimum pheromone limit and the optimal path of each ant colony, the genetic algorithm (mutation operation) is used to reduce the local optimal solution problem in the search process of the quantum ant colony algorithm.

The ant colony algorithm is improved by using genetic algorithm variation [35]. In the process of variation, a certain q node of the ant path is randomly selected to mutate, and q is gradually reduced as the number of iterations increases to ensure the convergence of the ant colony algorithm. Let the number of path nodes of the ant be n , then there is

$$q = \text{round}\left(n \cdot \left(1 - \frac{t}{t_{\max}}\right)\right). \quad (15)$$

In this formula, the current number of iterations is represented by t , and the maximum number of iterations is represented by t_{\max} . The variation method is to randomly take a 0–9 integer from the selected series of nodes. If the result of the mutation is better than the original, the mutated ant path is replaced by the current path.

3.3. MQACA-RBF Network Algorithm Steps. In summary, induction of the MQACA-RBF network algorithm to determine ω and the corresponding steps are as follows:

- Step 1: QACA algorithm population initialization.
 Step 2: the initial value of the number of evolutions is $t = 1$.
 Step 3: by coding, the RBF neural network is modeled according to ω , c , and σ , and the joint angle prediction accuracy of each group of parameters is obtained.

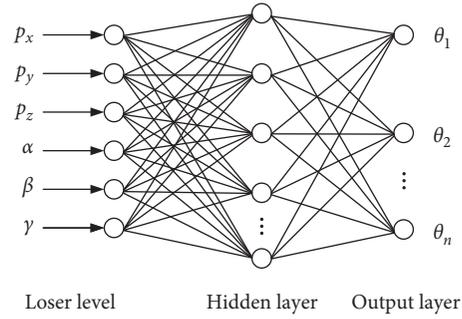


FIGURE 2: RBF neural network.

Step 4: update the rotation angle of the quantum rotation door, update the quantum pheromones at different nodes, achieve quantum initialization, and adjust the pheromone content on different paths.

Step 5: probability transition formula (9) needs to be chosen to proceed to the next lattice point. When ant k comes to the termination point, it can calculate the f value and then return to step (4) by decoding the formula.

Step 6: if $k = m$, then enter step 7.

Step 7: the optimal adaptation function f and optimal solution ω^* in the current ant population are selected, and the global pheromones are updated according to the maximum and minimum pheromones.

Step 8: local search 2-opt optimization, mutation, and other measures were carried out for ω^* .

Step 9: assuming that the optimal adaptive function value and the maximum number of iterations of the ant population have not changed too much, the optimal model parameters can be obtained, and then enter step (11).

Step 10: the number of iterations is added by 1, that is, $\text{Iter} = \text{Iter} + 1$ and then return to step (4).

Step 11: ω , c , and σ were calculated as the optimal parameter (the optimal parameter of each joint angle prediction model).

Step 12: The MQACA-RBF neural network is modeled on ω , c , and σ and predicts the test set. For this reason, the joint angle prediction value can be obtained through the output model.

According to the steps above, the flow chart of the MQACA-RBF inverse solution algorithm is obtained as shown in Figure 4.

3.4. Normalization of Sample Parameters. In the process of calculating the data through the model, due to the influence of many factors, the problem of data overflow may be caused. Therefore, it is necessary to standardize the sample parameters to ensure the computational efficiency of the prediction model. The sample parameters are normalized to range from 0 to 1. The normalization formula is

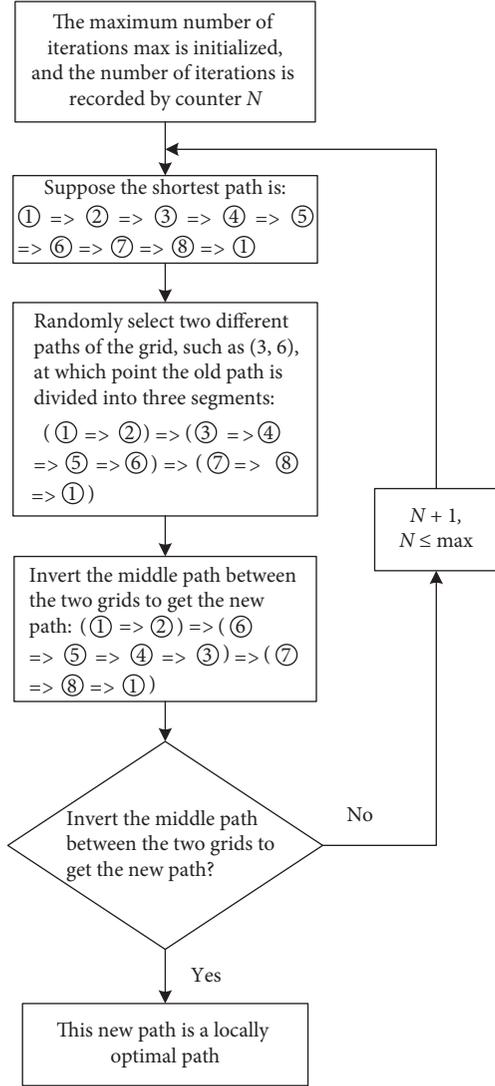


FIGURE 3: Local optimization process of the 2-opt algorithm.

$$z' = \frac{z - z_{\min}}{z_{\max} - z_{\min}}, \quad (16)$$

where z —parameters before normalization; z_{\min} —minimum parameter; z_{\max} —maximum parameter; and z' —normalized parameters.

4. Simulation Experiment and Error Analysis of 6R Manipulator

4.1. Establishment of a Prediction Model for the Inverse Solution of 6R Robotic Arm. In order to follow the algorithm steps in Section 3.3 for simulation experiments, according to the MQACA algorithm to optimize the output weight of Figure 2, the ant gets each output weight in the process of passing through each $n * d$ grid [29]. In the work space of the 6R robotic arm, 8000 points are randomly generated, and 1000 points are selected as training samples. After normalization, the input parameters of the RBF network are formed, i.e., $x = [P'_x \ P'_y \ P'_z \ \alpha' \ \beta' \ \gamma']^T$. In order to obtain

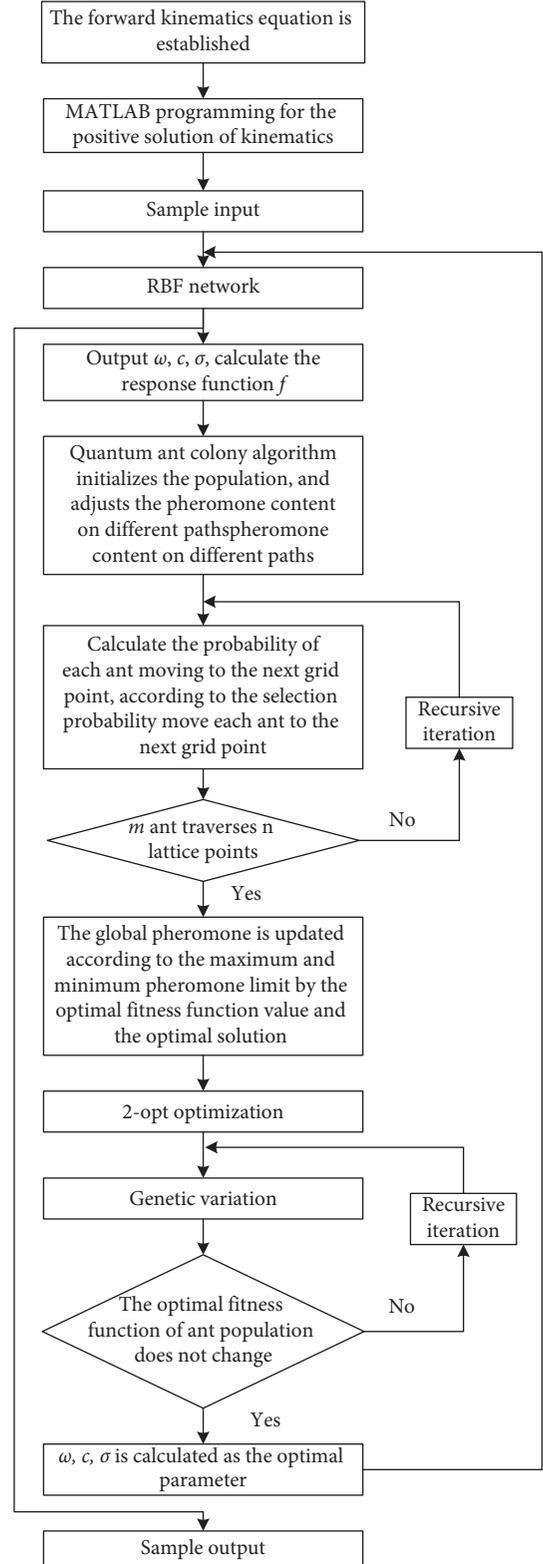


FIGURE 4: MQACA-RBF algorithm for inverse kinematics steps flow chart.

the output parameters of the RBF network, the joint angle (17) needs to be normalized, i.e., $y = [\theta'_1 \ \theta'_2 \ \theta'_3 \ \theta'_4 \ \theta'_5 \ \theta'_6]$. Based on this, a prediction model of the 6-input 6-output

inverse solution algorithm based on the MQACA-RBF network is established, training in predictive models based on a sample of 1000 points.

After determining the position parameters, the number of joint angles between 0 and 1 can be obtained by using the prediction model, and then the reverse normalization is performed, i.e.,

$$\theta_j = \theta_{j_{\min}} + \theta'_j (\theta_{j_{\max}} - \theta_{j_{\min}}), \quad j = 1, 2, \dots, 6, \quad (17)$$

where θ'_j —the output joint angle predicted after normalization.

4.2. Analysis of Spatial Random Point Errors. The space point simulation experiment of the ZrARM 6R manipulator is carried out. The DH parameters of the ZrARM arm are shown in Table 1 according to the structural parameters and the external dimensions of the arm.

In the MATLAB software environment, using the modeling in Section 3.1 and the 1000 points in the ZrARM model space as samples, the MQACA-RBF network method is used to simulate, and 100 points are randomly selected. Compared with the inverse solution algorithm simulation results of ACA, QACA, and RBF neural networks in the same operating environment, the position error of the four algorithms is shown in Figure 5 according to the distance between the actual coordinate point position and the target coordinate point position.

It is clearly seen from Figure 5 that the location accuracy of the inverse solution of the MQACA-RBF network is the highest in the four algorithms, and the error is stable and the change is smooth. In order to improve the real-time and convergence speed of the system, the method of online training is used in practice.

4.3. Error Analysis of Motion Trajectory

4.3.1. Selection of Locus Points. Figure 6 shows the end of the ZrARM robotic arm, a known elliptical trajectory that is simulated in the established model space.

In order to analyze the trajectory error, on this known trajectory curve, according to the trajectory length, 30 feature points are evenly selected as a description of the trajectory. The position of the end of the robot arm corresponding to the selected 30 points is calculated by equations (1)–(4) to obtain 30 sets of known $p_x, p_y, p_z, \alpha, \beta,$ and γ position parameter values and the corresponding 30 sets of known inverse $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,$ and θ_6 values are obtained, which are used as reference point for trajectory error analysis.

4.3.2. Error Analysis of Inverse Solution of Motion Trajectory.

The 30 selected feature points on the trajectory in Section 4.3.1 are used as inputs. After the inverse solution algorithm based on the MQACA-RBF network is calculated and normalized, 30 sets of predicted output values are obtained. In order to evaluate the prediction effect of the model objectively, absolute error index is used to evaluate it. As shown

TABLE 1: DH parameters of ZrARM mechanical arm.

Joint i	a_{i-1} (mm)	d_i (mm)	α_{i-1} ($^\circ$)	Θ_i ($^\circ$)
1	0	0	0	$-160 \sim 160$
2	0	0	-90	$-225 \sim 45$
3	10	3	0	$-45 \sim 225$
4	3	10	-90	$-110 \sim 170$
5	0	0	90	$-100 \sim 100$
6	0	0	-90	$-266 \sim 266$

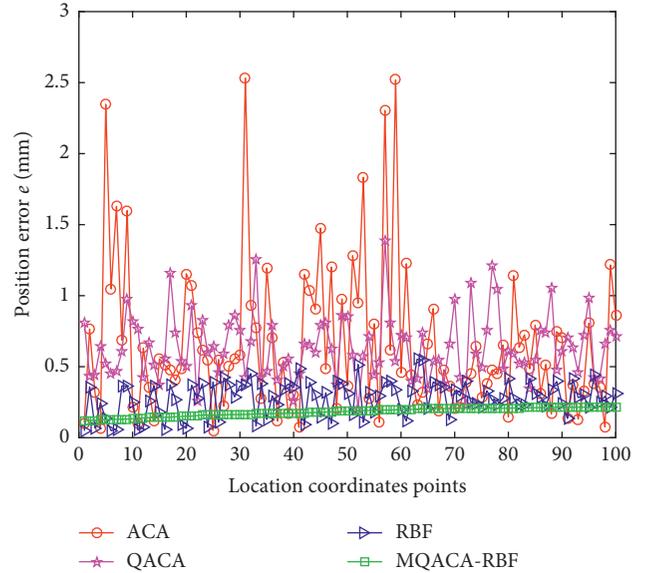


FIGURE 5: ACA, QACA, RBF, and MQACA-RBF position error comparison.

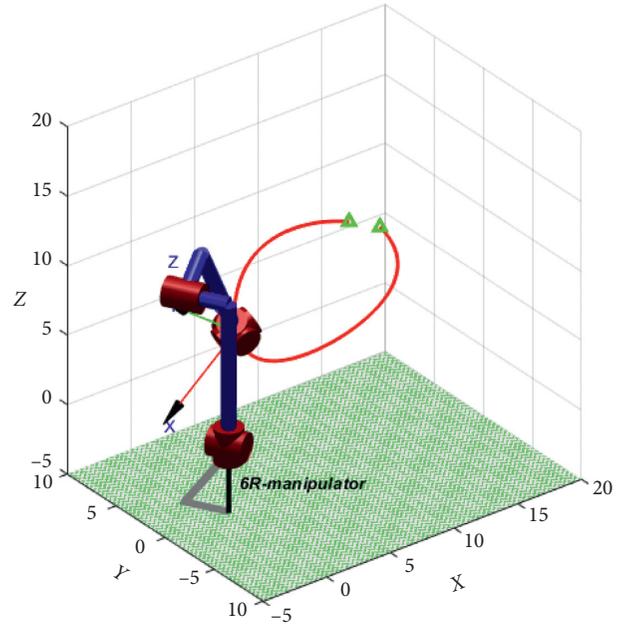
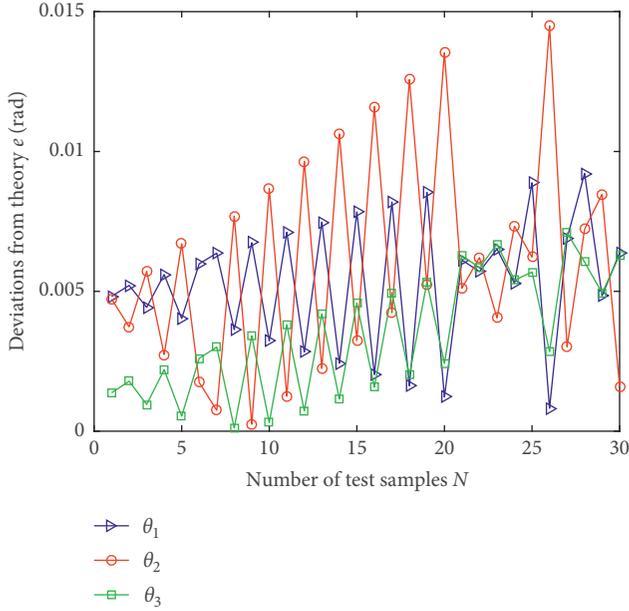
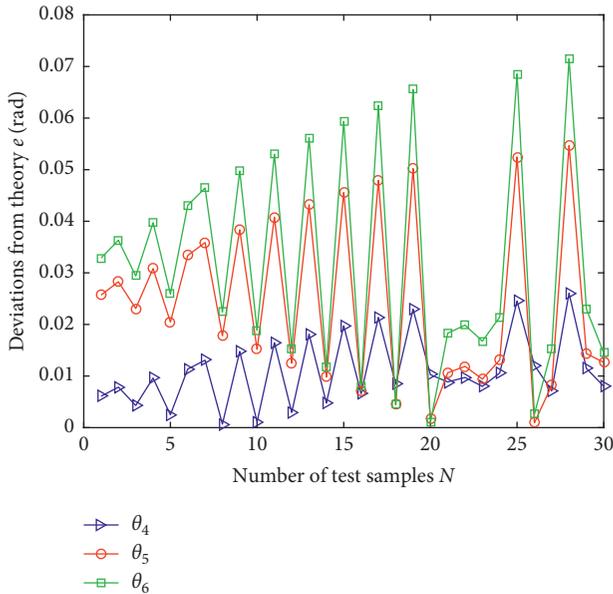


FIGURE 6: Simulation model of the ZrARM robotic arm trajectory.

in Figures 7 and 8, the inverse solution prediction values of 30 sets of joint angles θ_1 to θ_6 corresponding to the selected 30 points are compared with the absolute error of the real

FIGURE 7: Absolute error of joint angles θ_1 , θ_2 , and θ_3 .FIGURE 8: Absolute error of joint angles θ_4 , θ_5 , and θ_6 .

values in the same posture. Among them, the minimum absolute error differences of θ_1 , θ_2 , θ_3 , θ_4 , θ_5 , and θ_6 joint angles were $2.1679e-05$ rad, $8.5494e-05$ rad, $3.4619e-05$ rad, $3.1973e-05$ rad, $9.5498e-05$ rad, and $5.8276e-05$ rad, and the maximum absolute error differences were 0.10825 rad, 0.3027 rad, 0.10811 rad, 0.12094 rad, 0.18941 rad, and 0.25166 rad, respectively. According to formula (12), the mean square root errors of joint angle θ_1 , θ_2 , θ_3 , θ_4 , θ_5 , and θ_6 are 0.12147 rad, 0.37941 rad, 0.11798 rad, 0.19206 rad, 0.28645 rad, and 0.38298 rad, respectively.

Therefore, it can be seen that, in the prediction model of the 6R robotic arm inverse solution based on MQACA-RBF,

TABLE 2: Comparison of inverse solutions of the ZrARM robotic arm point A and point B (°): A-point inverse solution result comparison.

	True value	ACA	QACA	RBF	MQACA-RBF
θ_1	-0.47170	-0.47080	-0.38047	-0.41524	-0.41825
θ_2	-1.54250	-0.12161	-0.20561	-1.54780	-1.55540
θ_3	0.02829	3.59860	3.75210	0.08476	0.08104
θ_4	0.99840	1.59010	1.58180	0.98012	0.98327
θ_5	1.49760	1.45150	0.87234	1.47020	1.48250
θ_6	1.99680	-2.86380	2.28000	1.96020	1.97840

TABLE 3: Comparison of inverse solutions of ZrARM robotic arm point A and point B (°):B-point inverse solution result comparison.

	True value	ACA	QACA	RBF	MQACA-RBF
θ_1	-0.40893	-0.41000	-0.39114	-0.45849	-0.42937
θ_2	-1.48010	-0.22064	0.03381	-1.48780	-1.50210
θ_3	0.09107	3.80100	3.50300	0.04151	0.06909
θ_4	0.98341	1.50450	1.23030	1.04950	1.00130
θ_5	1.47510	1.50030	1.32880	1.57430	1.51030
θ_6	1.96680	2.06570	3.03530	2.09910	2.01770

the accuracy error of angular prediction is small, and the prediction accuracy is high.

In order to compare with other inverse solutions, randomly select points A (spatial coordinates: 10.6601, -2.0700, and 12.4255) and B (spatial coordinates: 11.5458, -1.7339, and 11.1018) among the above 30 points, The inverse solutions of point A and point B are obtained by using four algorithms: MQACA-RBF network, ACA, QACA, and RBF network. Tables 2 and 3 show the inverse results of the four algorithms under the same posture conditions. From the specific data comparison, it can be seen that the RBF network algorithm has better results in the inverse solutions of ACA, QACA, and RBF networks. However, the prediction accuracy of the MQACA-RBF network proposed in this paper is the highest, and it is closer to the real value.

4.3.3. Coordinate Point Error Analysis. In order to analyze the error of the trajectory from the angle of the coordinate point position, for the 30 feature points selected in Figure 4, the joint angle data of the inverse solution of each of the four algorithms ACA, QACA, RBF neural network, and MQACA-RBF network are used. After entering the positive solution equation, the actual values corresponding to the x -axis, y -axis, and z -axis coordinate values can be obtained. The coordinate error of the coordinate point position is

$$e_{wm} = |p_{wpm} - p_{wtm}|, \quad m = 1, 2, \dots, 30, \quad (18)$$

where m — number of feature point; w — represents x , y , or z coordinates; e_{wm} —absolute error between the actual value x (y or z) of point m coordinates and the true value x (y or z) of the coordinates; p_{wpm} —the actual value of point x (y or z) coordinates; and p_{wtm} —the true value of point x (y or z) coordinates.

Figure 9 shows the error comparison of the x , y , and z coordinates of the above four algorithms for the 30 feature points. It can be seen that the MQACA-RBF network algorithm has the smallest error.

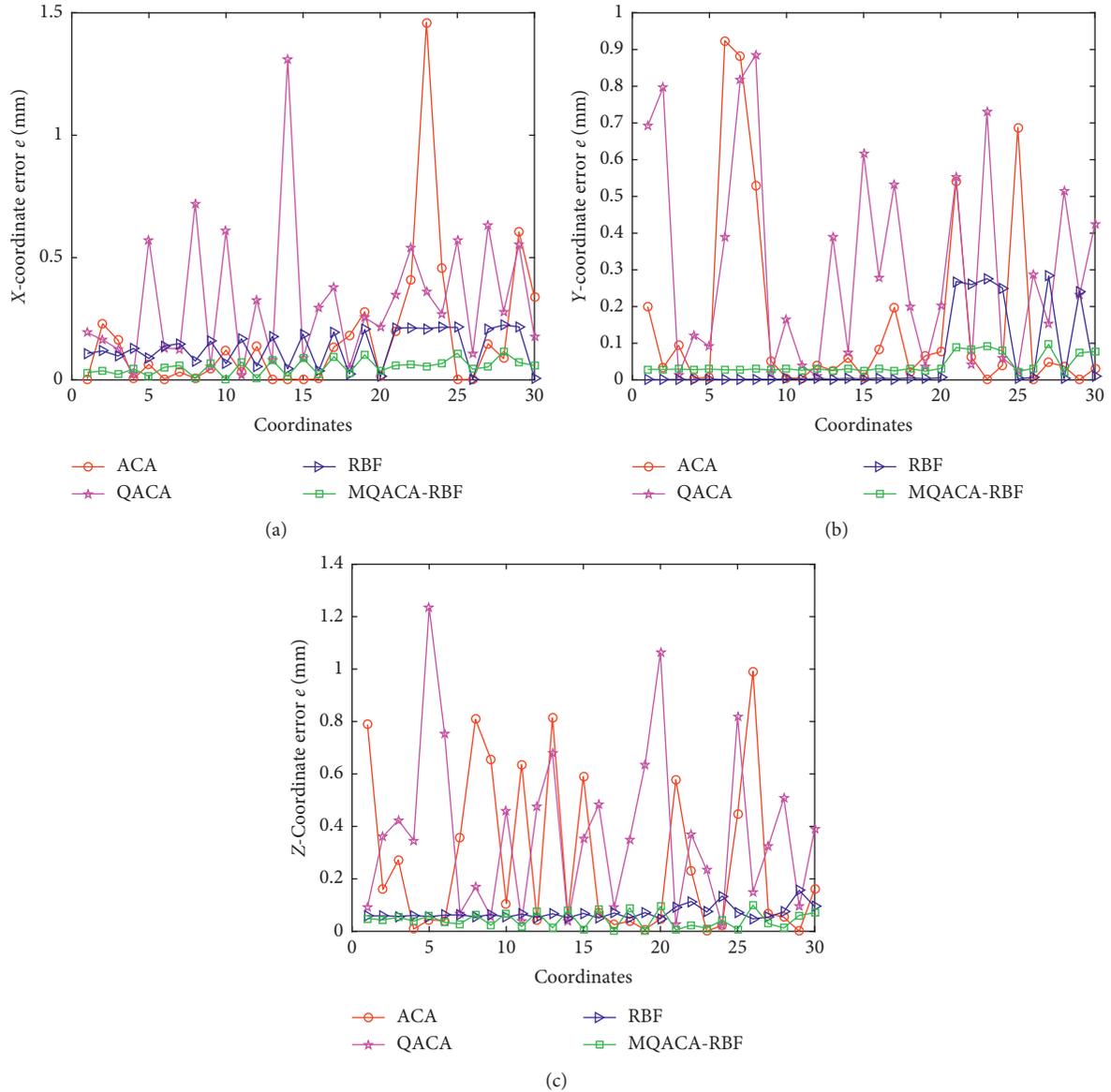


FIGURE 9: Comparison of coordinate errors of the selected points. (a) X-coordinate error comparison. (b) Y-coordinate error comparison. (c) Z-coordinate error comparison.

TABLE 4: Comparison of coordinate and position errors (mm).

Coordinates	Error	ACA	QACA	RBF	MQACA-RBF
X	Biggest	1.46050	1.30880	0.22370	0.11588
	Minimum	0.00067	0.01779	0.00329	0.00016
	Average	0.17214	0.31905	0.13219	0.05269
Y	Biggest	0.92384	0.88523	0.28325	0.09689
	Minimum	0.00041	0.00989	0.00017	0.02250
	Average	0.15881	0.31308	0.05509	0.04109
Z	Biggest	0.99137	1.23370	0.15607	0.09947
	Minimum	0.00001	0.02201	0.04758	0.00231
	Average	0.27080	0.37015	0.06997	0.04306
Location	Biggest	1.46050	1.36330	0.36006	0.12036
	Minimum	0.01075	0.05686	0.04820	0.06407
	Average	0.48815	0.69687	0.17773	0.09144

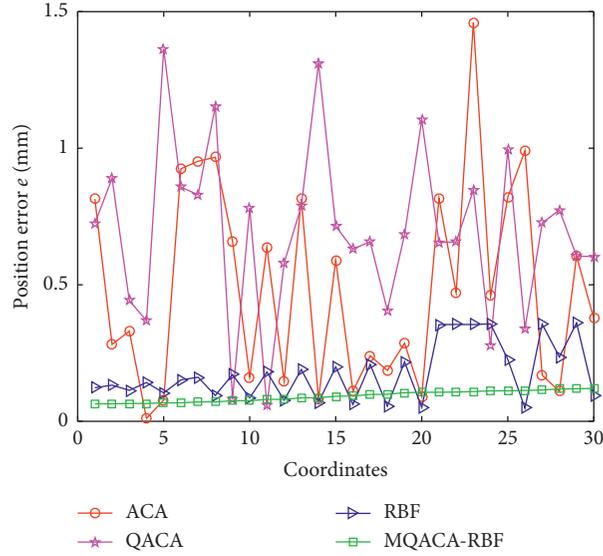


FIGURE 10: Location error comparison of selected points.

From the spatial position point of view, the location error comparison of the four algorithms ACA, QACA, RBF network, and MQACA-RBF network can be obtained by calculating the position error formula of equation (19), as shown in Figure 8. Table 4 shows the data comparison of the spatial positions, maximum, minimum, and average of the four algorithms:

$$e_{pm} = \sqrt{(p_{xpm} - p_{xtm})^2 + (p_{ypm} - p_{ytm})^2 + (p_{zpm} - p_{ztm})^2}. \quad (19)$$

where e_{pm} —location error of point m on trajectory.

As can be seen from Figures 9 and 10 and Table 4, the trajectory coordinate error and position error based on the inverse solution algorithm of the MQACA-RBF network are minimal; The average error of x , y , and z coordinates of the actual point position is 0.05269 mm, 0.04109 mm, and 0.04306 mm, respectively. The average error of the actual point position is 0.09144 mm, which is superior to the ACA, QACA, and RBF network algorithms. It can be seen that the coordinate points corresponding to the position error are combined with x , y , and z coordinate points, which best reflect the position error of the end of the manipulator. Based on the inverse solution algorithm of MQACA-RBF network, the fitting error between the end of the robot arm and the set trajectory is minimal, and it is very stable and robust, which proves that the scheme is feasible.

5. Conclusion

In this paper, a manipulator inverse solution algorithm based on the MQACA-RBF network is proposed. This method effectively synthesizes the characteristics of the RBF neural network and MQACA algorithm and gives play to their respective strengths. From the results of spatial random selection, trajectory simulation experiments, and position error simulation, it can be seen that

the inverse solution algorithm using the MQACA-RBF network is compared with the inverse solution algorithm based on ACA, QACA, and RBF networks, and the position error increased by 3.89%, 6.19%, and 0.76%, respectively. It can be seen that the robot arm inverse solution prediction model based on the MQACA-RBF network has higher accuracy and universality, and it has the reference value for other inverse solution methods and further robot arm spatial path planning.

Data Availability

The data used to support the findings of this study are included within the article. The source program used to support the findings of this study is included within the supplementary information file.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The research was funded by the Basic Research Project of the Liaoning University of Science and Technology of China.

Supplementary Materials

Running a file named “main3.m,” you can get the motion track error analysis in Section 4.3 of the paper required by the simulation map and data. The folder named “matlab program for continuous function optimization problem based on basic quantum ant colony algorithm” contains the quantum ant colony algorithm for the inverse solution of the manipulator. The file, called “Q_QACA.mat,” is the program running in the “matlab program for continuous function optimization problem based on the basic quantum ant colony algorithm” folder generated by the quantum ant

colony algorithm to solve the manipulator inverse solution of the file. The folder named “basic ant colony algorithm continuous function optimization problem matlab program” contains the ant colony algorithm to solve the inverse solution of the robot arm program. The folder named “Q_ACA.mat” is the program in the process of “basic ant colony algorithm continuous function optimization problem matlab program” folder which generated the ant colony algorithm to solve the inverse solution of the robot arm file. The folder named “MQACA_rbf.mat” uses the RBF toolbox to model, uses quantum ant colony algorithm to optimize the output weight of RBF neural network, and calls the “matlab program for continuous function optimization problem based on basic quantum ant colony algorithm.” The folder named “opt2.m” is for 2-opt local optimization. The folder named “update_all_tao.m” is for Max Min pheromone limit, and the document, called “ObjV.m,” uses genetic variation and other improvements to improve the system’s motion accuracy by further improving its heuristics, increasing the diversity of population and path selection. (*Supplementary Materials*)

References

- [1] X. Li, H. Jiang, G. Yin et al., “Decoupling analysis of trajectory error compensation for multi-joint welding robot,” *Journal of Welding*, vol. 3410, pp. 63–66, 2013.
- [2] C. Ding, S. Yang, and C. Gan, “Robot dynamic behavior induced by random uncertainty disturbance and its optimization,” *Journal of Agricultural Machinery*, vol. 478, pp. 356–363, 2016.
- [3] S. Venanzi and V. Parenti-Castelli, “A new technique for clearance influence analysis in spatial mechanisms,” *Journal of Mechanical Design*, vol. 127, no. 3, pp. 446–455, 2005.
- [4] S. Liu, S. Zhu, and X. Wang, “Real-time and high-accurate inverse kinematics algorithm for general 6R robots based on matrix decomposition,” *Chinese Journal of Mechanical Engineering*, vol. 44, no. 11, pp. 304–309, 2008.
- [5] S. Lu, D. Zhang, and H. Liu, “Solving spinor equation of inverse kinematics of 6R robot based on Wu’s method,” *Journal of Mechanical Engineering*, vol. 4617, pp. 35–41, 2010.
- [6] H. Liu, S. Zhu, and J. Wu, “Real-time inverse solution algorithm of robot based on vector inner product,” *Journal of Agricultural Machinery*, vol. 406, pp. 212–216, 2009.
- [7] M. L. Husty, M. Pfulner, and H.-P. Schröcker, “A new and efficient algorithm for the inverse kinematics of a general serial 6R manipulator,” *Mechanism and Machine Theory*, vol. 42, no. 1, pp. 66–81, 2007.
- [8] N. Qiu, Z. Sui, M. Li et al., “Space marking trajectory planning algorithm for six degrees of freedom robots,” *Journal of Jilin University Engineering Edition*, vol. 435, pp. 1307–1313, 2013.
- [9] X. Li, Y. Guo, J. Zhang et al., “The inverse kinematics solution and verification of modular six-dof manipulator,” *Journal of Agricultural Machinery*, vol. 444, pp. 246–251, 2013.
- [10] G. Antonelli, S. Chiaverini, and G. Fusco, “A new on-line algorithm for inverse kinematics of robot manipulators ensuring path tracking capability under joint limits,” *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 162–167, 2003.
- [11] O. Kemal and M. Ozgoren, “Topological analysis of 6-joint serial manipulators and their inverse kinematic solutions,” *Mechanism and Machine Theory*, vol. 375, pp. 511–547, 2002.
- [12] E. Bayro-Corrochano and J. Zamora-Esquivel, “Differential and inverse kinematics of robot devices using conformal geometric algebra,” *Robotica*, vol. 25, no. 1, pp. 43–61, 2007.
- [13] Y. Zhou, J. Zhang, and Y. Dong, “Combination optimization of complex trajectory inverse kinematics of spot welding robot,” *Journal of Welding*, vol. 319, pp. 21–24, 2010.
- [14] Y. Ran, *Research on Control Algorithm of Manipulator Based on Chaos Ant Colony Theory*, University of Science and Technology Liaoning, Anshan, China, 2017, <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CMFD&dbname=CMFD201801&filename=1017262507.nh&v=MDAwNDFyQ1VS N3FmWnVkdEZpSGtWYnJCVkYyNkdRytITIRncUpFYIBJ UjhlWDFMdxhZUzdEaDFUM3FUclDNMUY=>.
- [15] D. Li, *Research and Application of Path Planning for Obstacle Avoidance of Manipulator*, University of Science and Technology Liaoning, Anshan, China, 2018, <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CMFD&dbname=CMFD201901&filename=1018288232.nh&v=MTk1MjBlWDFMdxhZUzdEaDFUM3FUclDNMUYzQ1VSN3FmWnVkdEZpSGx Vny9BVkYyNkZyR3dGdFBQclpFYIBJ Ujg=>.
- [16] X. Tian, H. Sun, J. Zhen et al., *Inverse Kinematics Solution of ABB Robot Based on GA-RBF Network*, vol. 31, Mechanical Design and Manufacturing, Shenyang, China, 2010, <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFD2010&filename=JSYZ201001070&v=MTg5MzJU M3FUclDNMUYzQ1VSN3FmWnVkbUZpam5VcnJKTHo3 U2RMRzRIOUhNcm85Q1pJUjhlWDFMdxhZUzdEaDE=>.
- [17] R. Köker, C. Öz, T. Çakar, and H. Ekiz, “A study of neural network based inverse kinematics solution for a three joint robot,” *Robotics and Autonomous Systems*, vol. 49, no. 3-4, pp. 227–234, 2004.
- [18] B. Karlik and S. Aydin, “An improved approach to the solution of inverse kinematics problems for robot manipulators,” *Engineering Applications of Artificial Intelligence*, vol. 13, no. 2, pp. 159–164, 2000.
- [19] R. Köker, “A genetic algorithm approach to a neural network based inverse kinematics solution of robotic manipulators based on error minimization,” *Information Sciences*, vol. 222, pp. 528–543, 2013.
- [20] R. Da, C. Zheng, and M. Sun, *Improved DH Method for Establishing Robot Kinematics Model and Calculation of Forward and Inverse Solutions*, vol. 3710, Mechanical and Electronic, Shanghai, China, 2019, <https://kns.cnki.net/KCMS/detail/detail.aspx?dbcode=CJFQ&dbname=CJFDLAS T2019&filename=JXYD201910017&v=MTE2MDg3RGgxVD NxVHJXTTFGckNVUjdxZlp1UnJGeW5uVmIzT0x6WFnH ckc0SDlqTnI0OUVZNFI4ZVgxTHV4WVVM=>.
- [21] Z. Cai, “Chinese robotics for 40 years,” *Science & Technology Review*, vol. 33, no. 21, pp. 23–31, 2015.
- [22] J. Luo and H. Chen, “Kinematics analysis of manipulator based on artificial neural network,” *Information and Computers*, vol. 23, pp. 55–57, 2018.
- [23] H. Ge and R. Guo, “The Unity3D based Euler angles implements the application of object rotation,” *Agricultural Mechanization*, vol. 13, pp. 119–121, 2019.
- [24] L. Zhang, M. Cheng, L. Wang et al., “Short-term prediction analysis of chaotic vibration of pumping station pipeline,” *Hydroelectric Energy Science*, vol. 3710, pp. 80–84, 2019.
- [25] L. Wang, N. Qun, and R. Min, “A novel quantum ant colony optimization algorithm and its application to fault diagnosis,” *Transactions of the Institute of Measurement and Control*, vol. 303, pp. 313–330, 2008.
- [26] L. Wang, Q. Niu, and R. Min, “The third section ant colonies and particle swarm optimization and application a novel

- quantum ant colony optimization algorithm,” *Lecture Notes in Computer Science*, vol. 4688, pp. 277–286, Springer, Berlin, Germany, 2007.
- [27] J. Wang, P. Li, and C. Yang, “Theory and application of ant colony algorithm,” *Mechanical and Electrical Engineering*, vol. 205, pp. 126–129, 2003.
- [28] M. Zhou and X. Ao, “Research on fault diagnosis of electronic control engine based on ant colony neural network,” *Journal of Natural Sciences of Heilong Jiang University*, vol. 283, pp. 415–420, 2011.
- [29] M. Zhao and Y. Dai, “Application of fuzzy ant colony algorithm to robotics arm inverse kinematics problem,” *ICIC Express Letter*, vol. 101, pp. 5976–5983, 2016.
- [30] V. K. Jayaraman, B. D. Kulkarni, S. Karale, and P. Shelokar, “Ant colony framework for optimal design and scheduling of batch plants,” *Computers & Chemical Engineering*, vol. 24, no. 8, pp. 1901–1912, 2000.
- [31] S. Gao, “Improved quantum ant colony algorithm for solving traveling salesman problem,” *System Engineering Theory and Practice*, vol. 93, pp. 100–104, 2015.
- [32] Y. Li and L. Ma, “Quantum ant colony algorithm for solving large scale traveling sales man problems,” *Journal of Shang Hai University of Technology*, vol. 344, pp. 355–358, 2012.
- [33] R. Jia and W. Ma, “Improved maximum and minimum ant colony algorithm based on neighborhood search,” *Computer Simulation*, vol. 3112, pp. 261–264, 2014.
- [34] J. Watson, C. Ross, V. Eisele et al., “The traveling salesrep problem, edge assembly crossover, and 2-opt parallel problem solving from nature PPSNV,” *Lecture Notes in Computer Science*, Springer, vol. 259, pp. 823–832, Berlin, Germany, 1998.
- [35] Y. He, X. Wang, W. Li et al., “Study on solving discount 0-1 knapsack problem based on genetic algorithm,” *Computer Science*, vol. 3912, pp. 2614–2630, 2016.