applied sciences

MDPI

Article

Inverse Kinematics of Large Hydraulic Manipulator Arm Based on ASWO Optimized BP Neural Network

Yansong Lin, Qiaoyu Xu *, Wenhao Ju and Tianle Zhang

College of Mechanical and Electrical Engineering, Henan University of Science and Technology, Luoyang 471000, China; lys@stu.haust.edu.cn (Y.L.); juwenhao@stu.haust.edu.cn (W.J.); ztl@stu.haust.edu.cn (T.Z.)

* Correspondence: xiaoyu0622@163.com; Tel.: +86-137-8311-5261

Abstract: In order to solve the problem of insufficient end positioning accuracy due to factors such as gravity and material strength during the inverse solution process of a large hydraulic robotic arm, this paper proposes an inverse solution algorithm based on an adaptive spider wasp optimization (ASWO) optimized back propagation (BP) neural network. Firstly, the adaptability of the SWO algorithm is enhanced by analyzing the phase change in population fitness and dynamically adjusting the trade-off rate, crossover rate, and population size in real time. Then, the ASWO algorithm is used to optimize the initial weights and biases of the BP neural network, effectively addressing the problem of the BP neural network falling into local optima. Finally, a neural network mapping relationship between the actual position of the robotic arm's end-effector and the corresponding joint values is established to reduce the influence of forward kinematic errors on the accuracy of the inverse solution. Experimental results show that the average positioning error of the robotic arm in the XYZ direction is reduced from (91.3, 87.38, 117.31) mm to (18.16, 24.67, 27.21) mm, significantly improving positioning accuracy by 80.11%, 71.78%, and 76.81%, meeting project requirements.

Keywords: spider wasp optimization; BP neural network; inverse kinematics; error compensation; adaptive regulation

Citation: Lin, Y.; Xu, Q.; Ju, W.; Zhang, T. Inverse Kinematics of Large Hydraulic Manipulator Arm Based on ASWO Optimized BP Neural Network. *Appl. Sci.* **2024**, *14*, 5551. https://doi.org/10.3390/app14135551

Academic Editor: Marco Troncossi

Received: 23 May 2024
Revised: 14 June 2024
Accepted: 23 June 2024
Published: 26 June 2024

1. Introduction

The rapid development of tunnel construction has dramatically increased the demand for large hydraulic robotic arms. Compared with the electronically controlled mechanical arm, the hydraulic mechanical arm has the advantages of strong power, high durability, and fast response speed. However, the deflection error caused by the huge size of the arm itself makes it difficult to ensure its end positioning accuracy. The end position error of the forward kinematics can be determined through deflection modeling, error compensation modeling, and other methods. Inverse kinematics directly affects the control speed and accuracy of the robotic arm [1]. How to avoid the influence of forward solution errors and improve the positioning accuracy of the inverse solution of large hydraulic robotic arms has become the focus of research.

Currently, inverse kinematics methods for robotic arms mainly include geometric, analytical, iterative, and intelligent algorithms. The geometric method [2,3] depends on the geometric characteristics of the robot, is less generalized, and is not applicable to robotic arms with complex structures or many joints. The analytical method [4] has high solution accuracy but is computationally complex and has the problem of multiple solutions. The iterative method [5,6] requires a large amount of computation and is difficult to meet real-time control requirements. Intelligent algorithms can be divided into two parts: neural networks and heuristic algorithms, which transform the complex inverse kinematics problem into a data model training and objective optimization problem and have become effective methods for the inverse kinematics solution of robotic arms.

Artificial neural networks are widely used to solve complex robotic arm inverse solutions because of their powerful mapping ability, avoiding complex computational processes, and meeting real-time control requirements [7–13]. Yang et al. [14] established an accurate kinematic model of a rigid-flexible variable-diameter robotic arm by quantitatively analyzing different pressure and load cases underwater and utilized a deep neural network to map the relationship between the position and attitude of the target terminal and the input pressure, achieving high inverse solution accuracy. Wagaa et al. [15] solved the kinematic inverse problem of a six-degree-of-freedom robot by five different deep learning networks. Gao et al. [16] proposed an excitation function-based improved BP neural network for robot inverse solution. Aravinddhakshan et al. [17] completed the inverse solution of an industrial grade 5 degree of freedom manipulator with path planning through a neural network. Aydogmus et al. [18] completed the inverse solution of a humanoid robotic arm by using Bayesian-optimized deep neural network structure. Almusawi et al. [19] proposed an artificial neural network with the current joint angle of the robotic arm and the desired position and attitude of the end in the input layer to improve the joint output accuracy. However, the above neural networks are prone to falling into local optimality and and find it difficult to find the global optimal solution when dealing with multi-peak high dimensional problems.

Meta-heuristic algorithms are more efficient in solving complex nonlinear problems [20–27], which can overcome the shortcomings of neural networks that are prone to fall into local optimums. Zong et al. [28] proposed an inverse kinematics solution method for a four-degree-of-freedom hydraulic manipulator by combining the global mapping of a BP neural network with the local search of a gravitational search algorithm, ensuring solution accuracy while avoiding the problem of multiple solutions. Bai et al. [29] proposed to optimize the initial weight bias of BP neural networks with a fruit fly optimization algorithm (FOA) to avoid neural networks from falling into local extremes. Rokbani et al. [30] proposed a new method for kinematic inverse solution of a three-jointed robotic arm based on a firefly algorithm. Jiang et al. [31] proposed particle swarm (PSO) optimized BP neural network algorithm to solve the robotic arm inverse solution problem and improve the convergence accuracy and speed of neural network. However, the meta-heuristic algorithm is often constrained by preset fixed algorithmic parameters when coping with multi-peak high-dimensional problems, which leads to the lack of adaptability of the algorithm and the inability to effectively adjust the exploration strategy to adapt to the different stages of optimization.

Aiming at the above problems, this paper proposes an inverse kinematics method for a large hydraulic robotic arm based on an adaptive adjustment SWO optimization BP neural network.

The main contributions of this paper are as follows:

1. By improving the SWO algorithm, an ASWO algorithm is proposed. During the iteration process, the algorithm parameters are dynamically adjusted in real time according to the phase change of population fitness to enhance the adaptability of the algorithm.
2. The ASWO algorithm is used to optimize the initial weights and biases of the neural network, which effectively solves the problem that the BP neural network is easy to fall into the local optimum.
3. Establish the neural network mapping relationship between the actual position information at the end of the robotic arm and the corresponding joint value, so as to reduce the influence of the positive kinematic error on the inverse solution accuracy.

This paper is organized as follows.

In Section 2, the robotic arm is first modeled with forward kinematics to analyze the positioning error relationship between forward and inverse kinematics, and then the forward solution error model is established to ensure the reliability of the data in the simulation stage. In Section 3, the SWO algorithm is improved through parameter adaptive tuning. In Section 4, the ASWO algorithm is pre-optimized to obtain the optimal initial

weights and biases of the neural network through continuous iteration of the population. In Section 5, the validity of the method proposed in this paper is verified by simulation and experiment.

2. Forward Modeling and Error Analysis of the Robotic Arm

In order to improve the accuracy of the inverse solution of the robotic arm, it is necessary to minimize the distance e between the actual position of the end-effector and the target position. By modeling the forward kinematics of the robotic arm and analyzing the error relationship between the theoretical and actual positions during the forward solution, the main source of the end-positioning error e in the inverse kinematics is identified.

2.1. Forward Modeling of the Robotic Arm

This paper takes the G3Zi rock drill dolly mechanical arm as the research object, which has five rotary joints and two translational joints. The length of the arm is about 7.84 m, and it reaches 11.5 m when the translational joints are fully extended. This is a kind of large hydraulic arm, as shown in Figure 1.

Figure 1. G3Zi rock drilling rig.

The standard DH method is used to establish the coordinate system of the robotic arm rod, as shown in Figure 2.

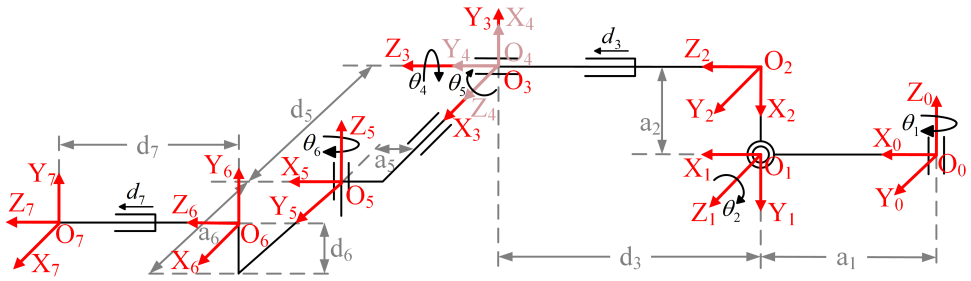

Figure 2. The DH model structure of the robotic arm.

The DH parameter values for each joint of the model, including the rotation angle θ_i, link offset d_i, twist angle α_i, and link length a_i, are presented in Table 1.

Table 1. DH parameters table.

i	θ_i/°	d_i/mm	α_i/°	a_i/mm	Joint Range
1	$\theta_1(0)$	0	−90	200	$\theta_1(-40,40)$
2	$\theta_2(90)$	0	90	−35	$\theta_2(-50,40)$
3	90	$d_3(4297)$	0	0	$d_3(0,2200)$
4	$\theta_4(90)$	0	90	0	$\theta_4(-180,180)$
5	$\theta_5(90)$	650	90	80	$\theta_5(-130,130)$
6	$\theta_6(90)$	679.5	90	241	$\theta_6(-12,58)$
7	0	$d_7(3058)$	0	0	$d_7(0,1600)$

θ_i denotes the angle at which the X_{i-1} axis is rotated about the Z_{i-1} axis until it is parallel to the X_i axis. d_i denotes the distance at which the X_{i-1} axis is translated along the Z_{i-1} axis until it is co-linear on the X_i axis. α_i denotes the angle at which the Z_{i-1} axis is rotated about the Z_{i-1} axis until it coincides with the Z_i axis. a_i represents the distance of translation along the X_i axis until point O_{i-1} moves to point O_i. The pose matrix between adjacent coordinate systems is shown in Equation (1), and the end-effector pose matrix is given in Equation (2), where $c\theta_i$ is an abbreviation for $cos\theta_i$, and $s\alpha_i$ is a shorthand for $sin\alpha_i$.

$$
{}^{i-1}_{i}T = \begin{bmatrix} c\theta_i & -s\theta_i c\alpha_i & s\theta_i s\alpha_i & a_i c\theta_i \\ s\theta_i & c\theta_i c\alpha_i & -c\theta_i s\alpha_i & a_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}
$$

$$
{}^{0}_{7}T = \prod_{i=1}^{7} {}^{i-1}_{i}T = \begin{bmatrix} {}^{0}_{7}R_{3\times3} & {}^{0}_{7}P_{3\times1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}
$$

In the configuration of the robotic manipulator, ${}^{i-1}_{i}T$ represents the pose transformation matrix from the i joint to the $i-1$ joint. As defined in Equation (2), ${}^{0}_{7}R_{3\times3}$ denotes the end-effector's orientation, and ${}^{0}_{7}P_{3\times1}$ signifies the end position.The end-effector's orientation ${}^{0}_{7}R_{3\times3}$ is the deflection angle of the end-effector coordinate system under the base coordinate system, and in this paper, we use the XYZ fixed angle method to describe the attitude, and the rotation matrix is shown in Equation (3). In inverse kinematics, the desired end-effector's orientation (typically specified in angles) can be swiftly converted into a rotation matrix.

$$
{}^{0}_{7}R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha c\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\alpha c\beta & s\alpha s\beta s\gamma + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \tag{3}
$$

2.2. Forward and Inverse Kinematic Error Analysis

The hydraulic manipulator arm is large in size, and its end position error is affected by the intertwining of multiple factors, including mechanical tolerances, software errors, and external load variations.

In order to accurately localize the robotic arm in real-time, the forward kinematics error compensation model is first constructed. The DH parameter method is used to calculate the theoretical position of the end-effector based on joint values. The actual position of the end-effector is recorded by a total station. The position error of the end-effector is obtained by calculating the difference between these two positions.

This project utilizes the Leica TS13 automatic total station, which offers an angular accuracy of $1''$ and a distance measurement accuracy of 1 mm + 1.5 ppm, meeting the accuracy requirements.

The procedure for data collection using the total station is as follows:

1. Two prisms are placed on the body of the rock drill truck, and the position of these prisms relative to the base coordinate system is measured.
2. Input the positions of the two prisms into the total station to complete the establishment of the base coordinate system.
3. A reflective sticker is placed on the actuator at the end of the robotic arm.
4. Use the total station to record the position of the reflective sticker, thus completing the collection of the actual position of the end effector.

The joint values and the end position errors are trained as the input and output of the neural network model, respectively, to obtain the mapping relationship between the joint values of the robotic arm and the position errors. The spatial error distance between the actual position and the theoretical position can be determined when the robotic arm is located at different joint angles. The model building process is shown in Figure 3.

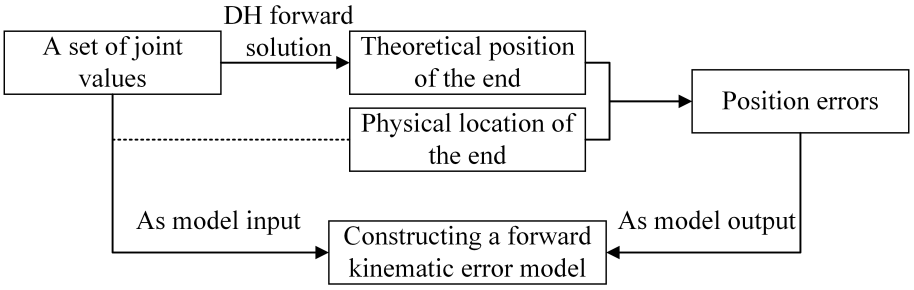

Figure 3. The process of establishing the forward kinematics error model.

During the operation of the robotic arm, although the forward kinematic error compensation model can accurately locate the actual position of the end of the robotic arm corresponding to a given combination of joint values, it does not completely solve the problem of positioning deviation between the end position and the expected target point. This forward kinematic positioning error, which is known in size but not compensated for, becomes a major factor affecting the accuracy of inverse kinematics. The robotic arm inverse kinematics error is shown in Figure 4.

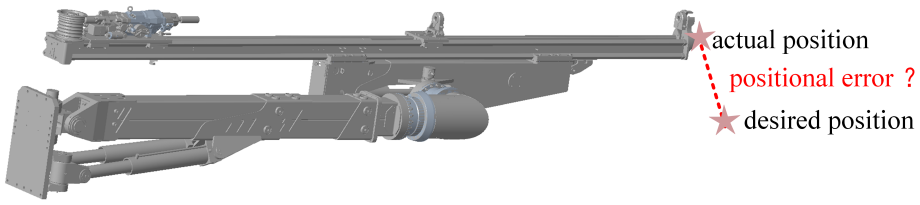

Figure 4. Robotic arm inverse kinematics error.

3. SWO Algorithm and Its Improvement

Mohamed Abdel-Basset proposed the SWO algorithm in 2023 [32]. In this chapter, the SWO algorithm and its optimized version, the ASWO algorithm, will be introduced, with a focus on the optimization process of the ASWO algorithm.

3.1. SWO Algorithm

The SWO algorithm is an optimization technique inspired by the behavior of spider wasps in nature. It effectively balances the exploration and exploitation processes by simulating the hunting, nesting, and mating behaviors of female spider wasps, as shown in Figure 5.

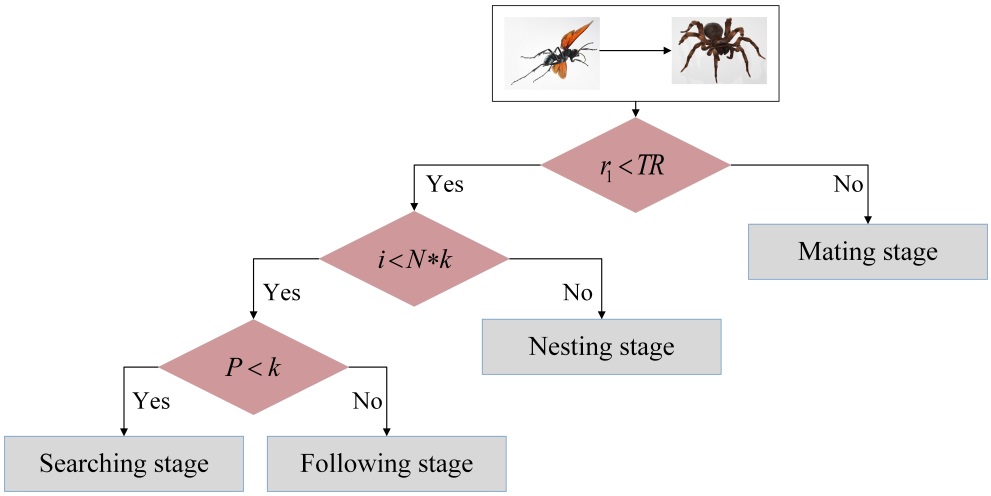

Figure 5. SWO algorithm judgement process.

Where r_1 is a 0~1 random number, TR is the trade-off rate, comparing the two to determine the current execution of hunting and nesting behaviors or mating behaviors. i is the current index value of the agent to be optimized, N is the population size, and k gradually decays from 1 to 0 with the growth in the number of iterations as shown in Equation (4), which determines the execution of hunting or nesting behaviors. p is a 0~1 random number, and the result of comparing it with the value of k determines whether the search agent executes the search or follows behavior. In addition, in the mating behavior, the crossover rate CR plays an important role in the decision.

$$k = 1 - (t/T_{\max}) \tag{4}$$

Among the behavioral strategies of the four algorithms mentioned above, different judgment conditions can be used to select suitable individual position updating methods in order to explore the solution space more efficiently. In the early stage of the search, the algorithms focus on extensive exploration to prevent prematurely stagnating at local optimum solutions. In the later stages of the search, they focus on utilizing known information to improve search efficiency, targeting the global optimum solution more quickly.

In addition, the algorithm introduces Levy flights. By adjusting the scaling factor or stability index of the step size, it can balance global exploration and local exploitation of the search process. With its long-range step size, the algorithm can escape local optima and explore search spaces far from the current position, helping to avoid getting stuck in local optima for extended periods in multi-peak problems. The continuous reduction of population size with population iteration can effectively reduce the computational burden brought by high-dimensional problems.

These features make the SWO algorithm show unique advantages in dealing with multi-peak and high-dimensional problems, which can adapt to the variable search space and locate the global optimal solution effectively. However, there are problems, such as the algorithm parameters are fixed and the local optimum cannot be jumped out in time in some iteration stages.

3.2. ASWO Algorithm

This study enhances the original SWO algorithm by incorporating a feedback adaptive adjustment that dynamically optimizes the population size N, trade-off rate TR, and crossover rate CR during the algorithm's execution. The average fitness variation rate $\Delta F\_avg(t)$ of the population is calculated for each generation as depicted in Equation (5). Where $F\_avg(t)$ represents the average fitness of generation 't' and $F\_avg(t-1)$ is the average fitness in generation $t-1$. If ΔF_avg does not improve significantly over a considerable number of generations, the parameters are adjusted to increase exploratory

capabilities. Conversely, when the algorithm demonstrates rapid improvement, parameters are adjusted to reduce exploration and increase the likelihood of exploiting potentially promising regions.

$$\Delta F\_avg(t) = -(F\_avg(t) - F\_avg(t-1))/F\_avg(t-1) \tag{5}$$

For optimization problems with vast solution spaces, it is necessary to adjust the strategy for optimizing population size. This strategy can be transitioned from a fixed iterative reduction to a dynamic adjustment based on the rate of change in fitness. Specifically, when there is a significant increase in population fitness, meaning that ΔF_avg remains above threshold th_u for consecutive 'm' generations, the population size is rapidly decreased to allow the algorithm to focus on high-potential areas. Conversely, when ΔF_avg remains below threshold th_l for consecutive 'm' generations, the reduction in population size is slowed to promote a broad exploration of the solution space and prevent the premature discarding of potential optimal solutions. This strategy is illustrated in Equation (6), where α_{fast}, α_{slow}, α_{normal} represent the three rate parameters that modulate the reduction in population size at different rates. This strategy maintains population diversity, assisting the algorithm in escaping local optima and potentially accelerating the global optimization process.

$$N(t+1) = \begin{cases} \max(N(t) - \alpha_{fast} * \frac{T_{\max}-t}{T_{\max}}, N_{\min}), count_{high} = m \\ \max(N(t) - \alpha_{slow} * \frac{T_{\max}-t}{T_{\max}}, N_{\min}), count_{low} = m \\ \max(N(t) - \alpha_{normal} * \frac{T_{\max}-t}{T_{\max}}, N_{\min}), otherwise \end{cases} \tag{6}$$

The trade-off rate TR and crossover rate CR are important parameters for controlling the positional variation of individual spider bees, and dynamic adjustment of TR and CR can balance global exploration and local exploitation to enhance the efficiency of the algorithm. When the average fitness change rate ΔF_avg is lower than the threshold th_l for m consecutive generations, TR and CR are boosted to promote exploration and increase population diversity, and local optimization is avoided global search. Conversely, if a significant improvement in algorithm performance is observed, i.e., when ΔF_avg is higher than the threshold th_u for m consecutive generations, decreasing TR and CR can focus on high-potential regions and retain high-quality solutions to enhance local search. This process is illustrated in Equations (7) and (8), where α_{TR} and α_{CR} represent the balanced increasing weights, and β_{TR} and β_{CR} represent the balanced decreasing weights.

$$TR(t+1) = \begin{cases} TR(t) + \alpha_{TR} * (th_l - \Delta F\_avg(t)), count_{low} = m \\ TR(t) - \beta_{TR} * (\Delta F\_avg(t) - th_u), count_{high} = m \end{cases} \tag{7}$$

$$CR(t+1) = \begin{cases} CR(t) + \alpha_{CR} * (th_l - \Delta F\_avg(t)), count_{low} = m \\ CR(t) - \beta_{CR} * (\Delta F\_avg(t) - th_u), count_{high} = m \end{cases} \tag{8}$$

In order to prevent the variables TR and CR from exceeding the reasonable range, the maximum value is set to 1 and the minimum value to 0. Boundary checking and correction are carried out using Equations (9) and (10) to ensure the reasonableness of the algorithm parameters and the stability of the algorithm's performance.

$$TR(t+1) = \min(\max(TR(t+1), 0), 1) \tag{9}$$

$$CR(t+1) = \min(\max(CR(t+1), 0), 1) \tag{10}$$

Adjusting the parameters based on the average fitness change rate over multiple consecutive generations can provide a smoother parameter adjustment mechanism and reduce overly drastic adjustments due to occasional fitness fluctuations. With this approach, the algorithm can make more robust and efficient parameter updates according to the current search stage, which helps the algorithm to move towards the optimal solution more efficiently.

Algorithm 1 lists the pseudocode for the ASWO.

Algorithm 1 ASWO algorithm

Input: Number of iterations $T_{\max}$, Dimension dim, Number of populations N, Minimum number of populations $N_{\min}$, TR, CR, Thresholds th_u th_l
Output: Optimal fitness value $fitness$, Optimal agent SW_*

```
Initialize the spider wasps, SW_i(i = 1, 2 · · · · · · N)
Calculating initial population fitness to select SW_*
t = 1 %Initialize iteration count
while t < t_max do
  if r < TR then
    for i = 1 : N do
      if i < N * k then
        Hunting phase;
      else
        Nesting phase;
      end if
      Calculating fitness to select SW_*;
    end for
  else
    for i = 1 : N do
      Mating phase;
      Calculating fitness to select SW_*;
    end for
    t = t + 1;
  end if
  Compute F_avg(t);
  Applying Equation (5);
  if ΔF_avg(t) < th_l then
    count_low+ = 1;
    count_high = 0;
  else if ΔF_avg(t) > th_u then
    count_low = 0;
    count_high+ = 1;
  else
    count_low = 0;
    count_high = 0;
  end if
  Applying Equation (6); %Population size N updates
  Applying Equations (7) and (8); %TR and CR updates
  Applying Equations (9) and (10); %Boundary checks and corrections
end while
```

4. ASWO-BP Inverse Solution Study

In order to address the shortcomings of the BP algorithm, such as its tendency to fall into local optima and its weak generalization ability, the ASWO algorithm is used to optimize the weights and biases of the BP neural network. This optimization improves the network's generalization ability and output accuracy. The flow of the algorithm is shown in Figure 6.

In the ASWO optimization BP neural network algorithm, the data set is first processed to determine the neural network topology, and random positions are generated to represent the search agents, i.e., individual spider bees, in the ASWO algorithm. The initial position of each agent represents a potential solution to the weights and biases in the neural network. Following the ASWO algorithm strategy, the search agents explore and update their positions in the solution space, and each population iteration performs fitness calculations and updates the algorithm parameters, which helps to guide the search agents to evolve towards a better solution and to find the optimal search agent through continuous

iterations. Finally, the optimal weights and biases obtained by the ASWO algorithm are used as the initial weights and biases of the BP neural network for training.

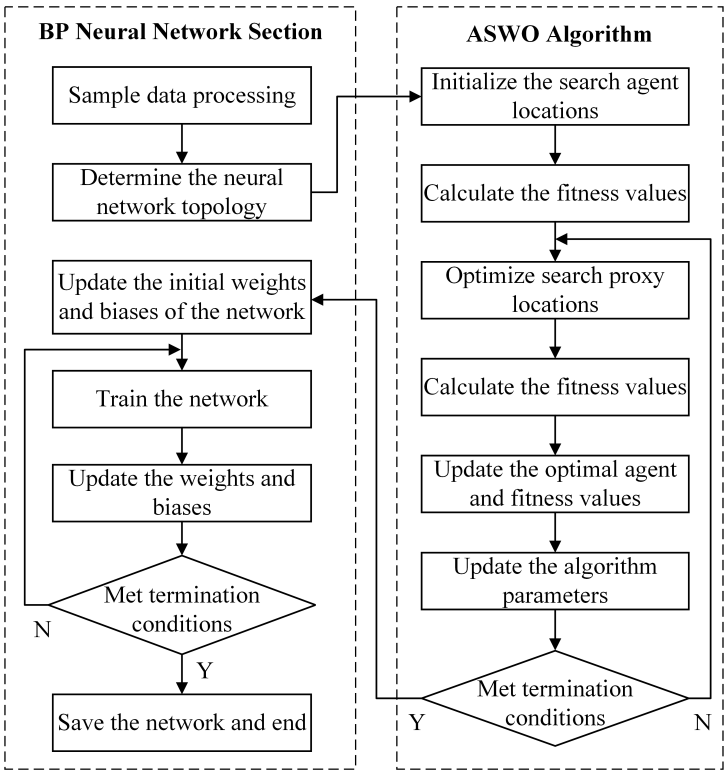

Figure 6. Flowchart of the BP neural network algorithm optimized by ASWO.

To construct the complete data set that covers the working range of the robotic arm, the complete spatial range reachable by the end of the drilling arm is first determined using the Monte Carlo pseudo-random number method within the range of each joint. During the data set establishment process, the straight-line distance between the rock drill dolly and the boring face is continuously adjusted, and the position points are planned within the coverage of the boring face so that they are evenly spread across it. The total station is used to collect and store the actual position data of the end of the robotic arm, while the control unit of the rock drill cart synchronizes and records the joint values of the robotic arm and the attitude of the end-effector.

The data set needs to be further processed by converting the attitude data into the 9 elements of the rotation matrix ${}^{0}_{7}R_{3\times3}$. The theoretical position of the end-effector is obtained by solving the forward kinematics of each set of joint values, and the spatial position error is obtained by subtracting the theoretical position from the actual position for use in the subsequent simulation process.

A total of 3095 sets of data were collected in this study, covering the entire working space of the robotic arm. Each group contains 7 joint values of the robotic arm, 3 actual position values of the end-effector, 3 theoretical positions, 3 spatial position errors, and 9 data units of the rotation matrix R. Eighty-five percent of the data set was used for training, and the remaining 15% (i.e., 464 data sets) was used for network prediction.

According to the analysis of multiple rounds of experiments, the optimal hidden layer structure is determined to be 2 layers, the number of nodes is 22 and 18, respectively, the hidden layer adopts the tansig function as the activation function, the number of training iterations is 1500, and the learning rate is set to 0.001, and the topology of the BP neural network is shown in Figure 7.

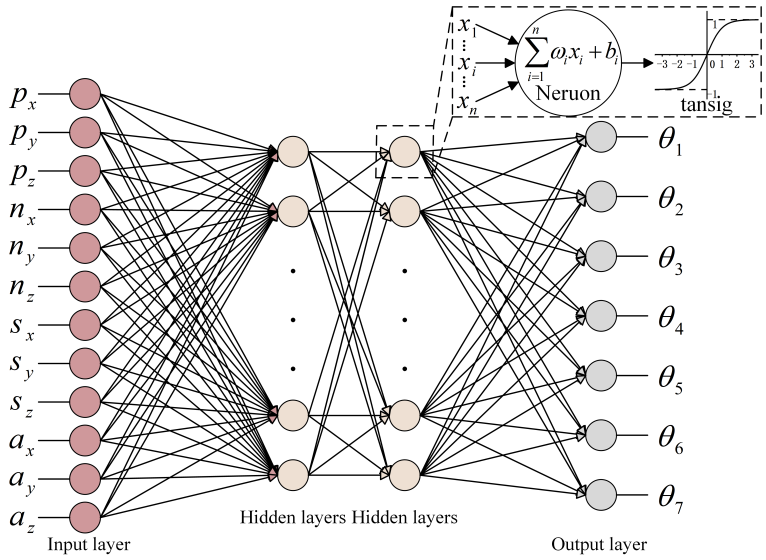

Figure 7. Topology of the BP neural network.

In order to find the global optimal solution, the mean square error (MSE) is used as the fitness value. MSE is the average of the squared errors between the predicted and actual values.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(\theta_i - \hat{\theta}_i\right)^2 \tag{11}$$

A smaller MSE value means a smaller gap between the predicted and true values, thus indicating a more accurate and better solution. In order to balance the influence of the telescopic and rotational joints on the overall fitness value when calculating the MSE, the two telescopic joints are unified in decimeters, ensuring equal importance of each joint parameter in the optimization process.

5. Simulation and Experimental Validation

In order to verify the effectiveness of the ASWO-BP algorithm in solving the inverse kinematics problem of the robotic arm, simulation and real experiments are carried out, respectively. MATLAB and CoppeliaSim joint simulation are used to simulate the real operation of the robotic arm based on the error model in Section 2.2. The three inverse solution methods, ASWO-BP (use actual positions as data set inputs), Jacobi iteration method, and ASWO-BP (use theoretical positions as data set inputs), are applied to the real operation of the G3Zi robotic arm, and the inverse solution error is calculated.

5.1. Simulation Verification

The simulation verification process is shown in Figure 8. The target position $X_{tgt}Y_{tgt}Z_{tgt}$ and the target attitude R_{tgt} are input into the ASWO-BP neural network inverse solution model, which results in a set of joint angles. Firstly, the DH forward solution is performed to obtain the theoretical position $X_{theo}Y_{theo}Z_{theo}$, and secondly, the set of joint angles is input into the forward solution error compensation model to obtain the position compensation value $X_{comp}Y_{comp}Z_{comp}$, and then the $X_{theo}Y_{theo}Z_{theo}$ and $X_{comp}Y_{comp}Z_{comp}$ are added together to obtain the actual position $X_{act}Y_{act}Z_{act}$ of the end. Finally, the difference between $X_{act}Y_{act}Z_{act}$ and $X_{tgt}Y_{tgt}Z_{tgt}$ is calculated to obtain the simulation inverse solution error $X_{error}Y_{error}Z_{error}$.

In order to verify the performance of the ASWO algorithm, three algorithms (GA, PSO, and SWO) are used for comparison experiments. To ensure a fair comparison of the four algorithms, the individual dimension dim, the population number N, the maximum number of iterations $T_{\max}$, and the fitness value function are unified. According to the BP neural network topology, the total number of weights and biases is 833, so the dimension

dim is set to 833, the population number N is set to 100, and the maximum number of iterations $T_{\max}$ is set to 50,000.

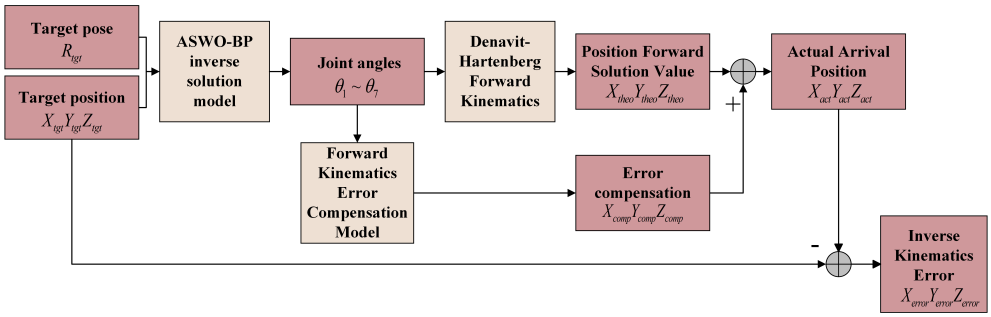

Figure 8. Simulation verification process.

In addition, to ensure a fair comparison among the four algorithms, other parameters were adjusted for each algorithm, and multiple sets of experiments were conducted to analyze the data and obtain optimal solutions.

The following is the optimal parameter configuration for each algorithm. In the ASWO and SWO algorithms, the initial minimum number of populations $N_{\min}$ is set to 80, the initial trade-off rate TR is set to 0.4, and the initial crossover probability CR is set to 0.3. The threshold th_l is set to 0.01 for the ASWO algorithm, th_u is set to 0.2, and m is set to 50. In the GA algorithm, the crossover rate CR is set to 0.3, and the variance rate VR is set to 0.1. In the PSO algorithm, the learning factors C_1 and C_2 are set to 2, and the inertia weight W is set to 0.6.

From the best fitness curve in Figure 9, it can be seen that the GA algorithm has a stronger global search ability and can obtain a better solution in fewer iterations, but it has weaker local search ability, with a best fitness value of 1.946. The PSO algorithm, due to its stronger local search ability, is prone to falling into local optima, with a best fitness value of 0.826. The SWO algorithm, due to its fixed parameters, can sometimes remain in the local optimum for a long time and is difficult to escape from, with a best fitness value of 0.641. The ASWO algorithm has a faster decreasing fitness value compared to the other three algorithms, is less likely to be troubled by local optima, and obtains a better solution with fewer iterations, with a best fitness value of 0.343.

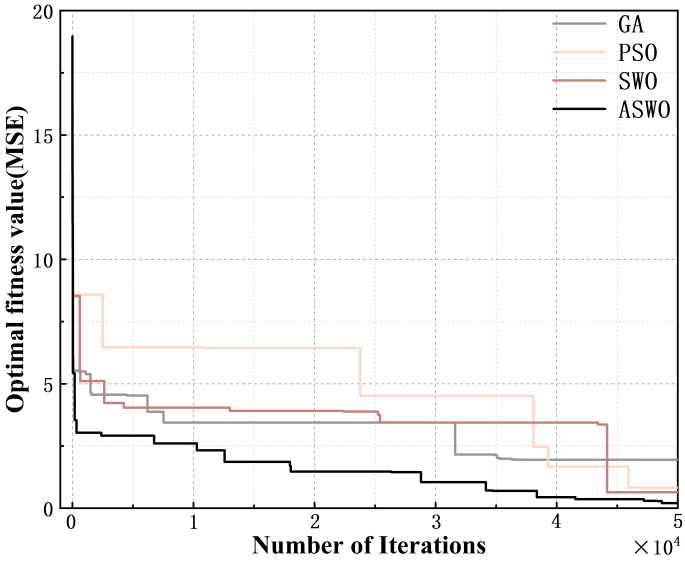

Figure 9. Optimal fitness curves of four algorithms.

The optimal solutions obtained by the four algorithms are used as the initial weights and biases of the BP neural network, respectively. After the training is completed, 464 sets

of test data are substituted into the four models to obtain the joint error range, MSE, and three-direction position error range, as shown in Table 2.

Table 2. Simulation results of the four algorithmic models.

Metrics	GA-BP	PSO-BP	SWO-BP	ASWO-BP
$\Delta\theta_1(^\circ)$	−0.941~0.86	−0.634~0.754	−0.536~0.417	−0.305~0.368
$\Delta\theta_2(^\circ)$	−1.121~0.953	−0.802~0.741	−0.382~0.459	−0.412~0.374
$\Delta d_3(\mathrm{mm})$	−113.6~102.7	−90.3~92.5	−85.2~79.6	−44.3~56.5
MSE	1.562	0.686	0.394	0.173
$\Delta X(\mathrm{mm})$	−86~110	−94~86	−61~49	−33~37
$\Delta Y(\mathrm{mm})$	−139~113	−89~100	−47~65	−42~40
$\Delta Z(\mathrm{mm})$	−155~124	−107~92	−66~73	−60~54

As can be seen from Table 2, compared with the other three algorithmic models, the first three joints of the ASWO-BP algorithm have smaller error ranges of (−0.305~0.368)°, (−0.412~0.374)°, and (−44.3~56.5) mm, respectively.The mean square error of the ASWO-BP algorithm is 0.173, which is lower than that of the optimal SWO-BP among the other three models, which is 0.394. The ASWO-BP algorithm's end position error ranges are (−33~37) mm, (−42~40) mm, and (−60~54) mm, which is lower than the SWO-BP, which is the best among the other three algorithm models.

The same target position (7950, 1800, 1540) mm is inversely solved by four methods respectively, and the four sets of joint values are substituted into the simulation model.

As can be seen from Figure 10, there are three position points in each figure: the theoretical position obtained by the DH forward solution, the actual position after compensation for the forward solution error model, and the preset target position. The ASWO-BP inverse solution position error is (−12, 20, 4) mm, the SWO-BP inverse solution position error is (−18, −30, −29) mm, the PSO-BP inverse solution position error is (−33, −37, 35) mm, and the GA-BP inverse solution position error is (−51, −24, −52) mm. The simulation results show that the ASWO algorithm is able to optimize the initial weights and biases more efficiently, thereby assisting the BP neural network in the learning and training process and helping the network avoid local optima and find the global optimum or a better local optimum.

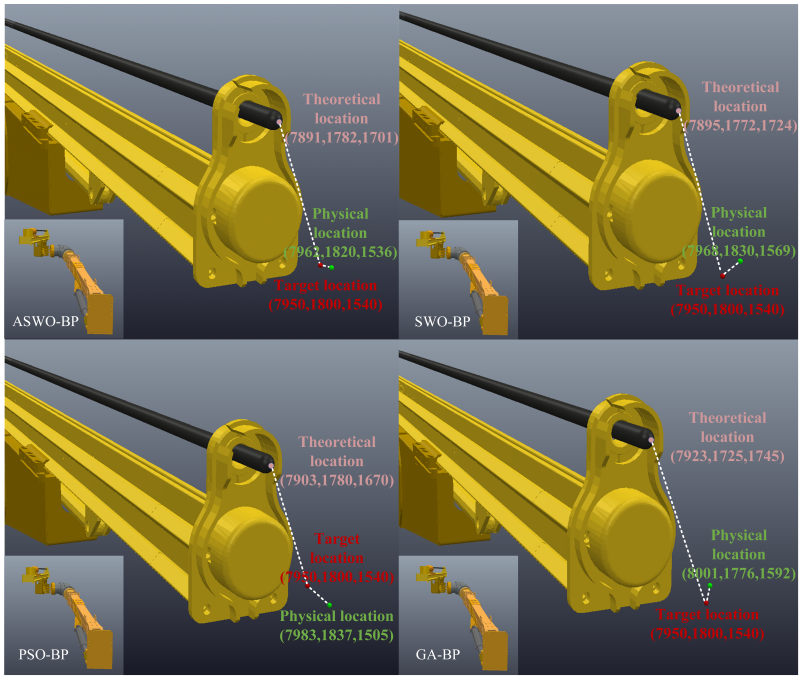

Figure 10. Comparison of end positioning accuracy of four inverse solution methods for robotic arms.

5.2. Experimental Validation of Robotic Arm Inverse Solution

In order to verify the validity of using actual positions as inputs to the BP neural network data set in this study, a comparison is made in Experiment 1, where theoretical positions are used as inputs to the data set, and the same algorithmic model as in this study is used for training. Fifty-five sets of data are randomly tested during the actual use of the robotic arm (the target positions are all within the operational space of the robotic arm), and the joint values and the actual positions of the robotic arm's end are recorded. The position errors of the two methods in the three XYZ directions under the same target positions are compared, and the experimental results are shown in Figure 11.

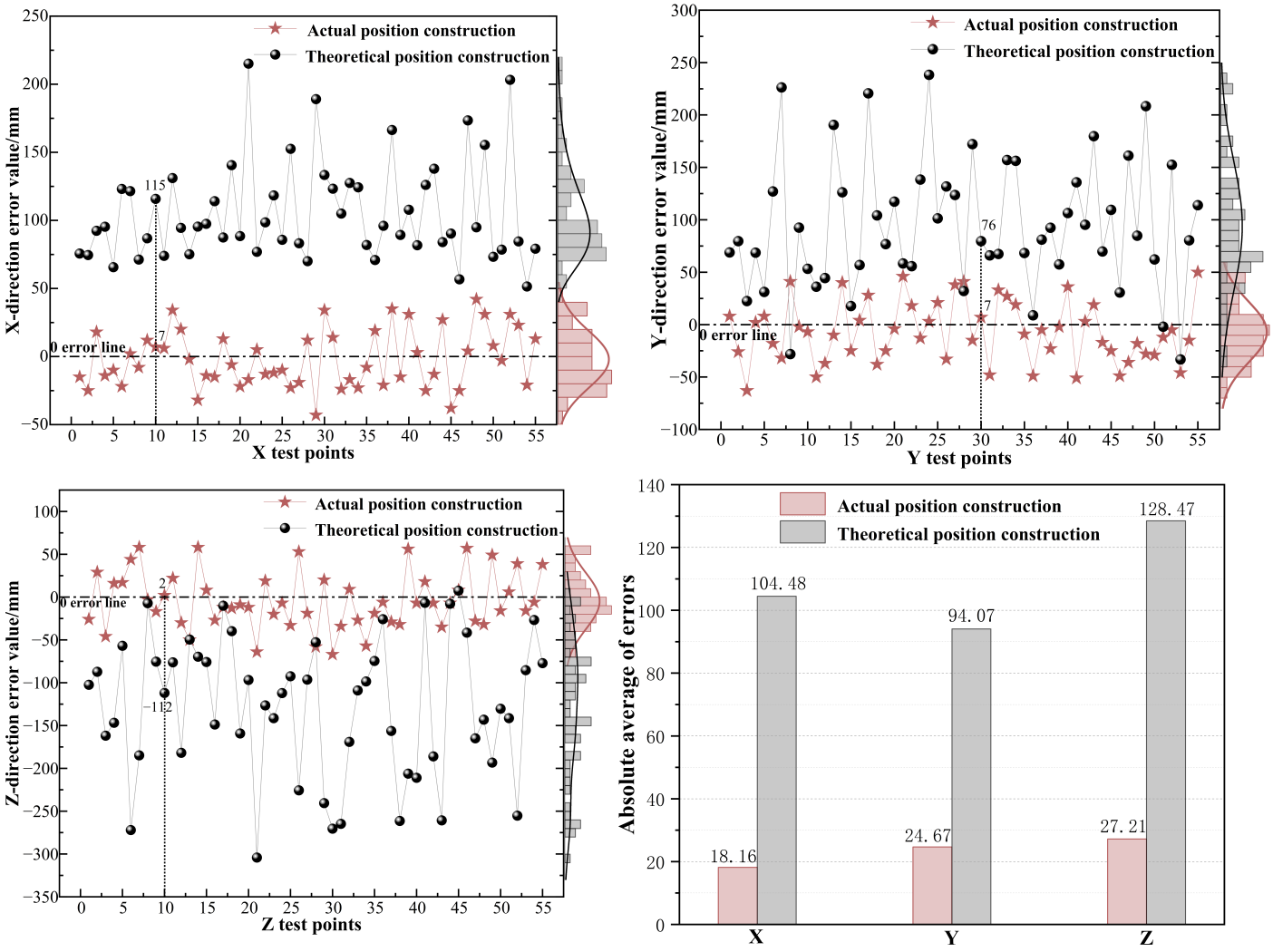

Figure 11. Comparison of errors for the two positions as inputs to the data set.

Each vertical axis in Figure 11 represents the positioning error in the XYZ directions, while the horizontal axis represents the test points. Using the theoretical position as the data set input does not effectively compensate for the forward solution error. Additionally, the inaccuracy of the inverse solution model itself further increases the localization error, resulting in average localization errors of (104.48, 94.07, 128.47) mm in the XYZ directions (indicated by black dots). In this study, using the actual position as the data set input can compensate for the aforementioned issues and significantly reduce the positional error. This approach reduces the average positioning error in the XYZ directions to (18.16, 24.67, 27.21) mm (indicated by red pentagrams), respectively. The experimental results confirm the validity and effectiveness of the data set settings in this paper.

In order to avoid the influence of neural network model errors on the experiment, the Jacobi iterative method, which is commonly used in the industrial field, is employed in Experiment II to compare with the research method of this paper. The experimental steps are similar to those in Experiment I. The spatial position errors of the two methods in the XYZ directions under 55 sets of target positions are calculated, and the test results are shown in Figure 12.

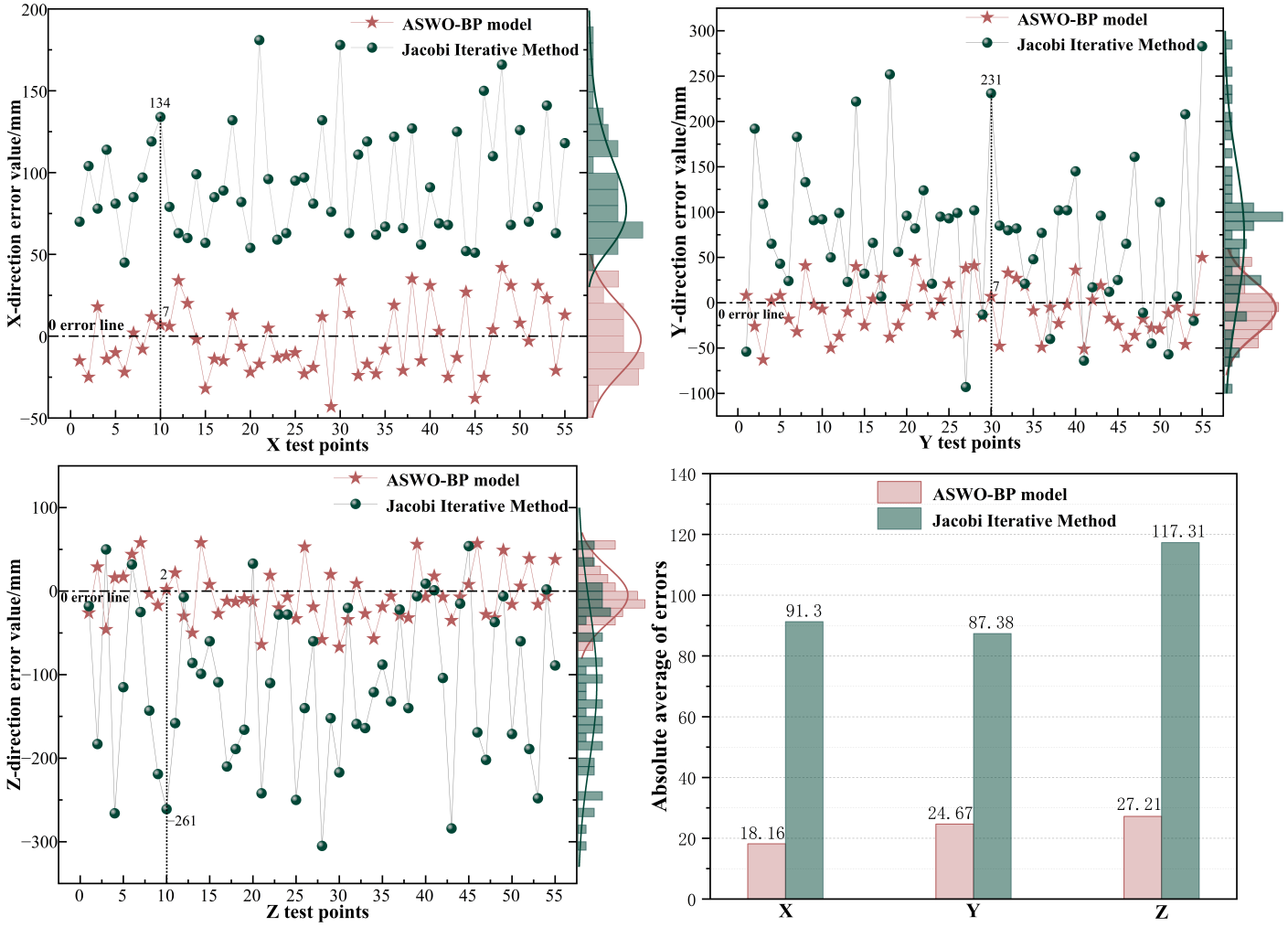

Figure 12. Comparison of end position error between ASWO-BP method and Jacobi iterative method.

The Jacobi iterative method is more accurate and can make the theoretical position of the end reach the target position accurately, but it cannot compensate for the existing forward solution error, leading to a larger error in the final position. As can be seen from the green dots in the error distribution in Figure 12, these points are relatively far from the zero-error line. In contrast, the inverse solution error (indicated by the red pentagrams) obtained using the ASWO-optimized BP neural network algorithm is significantly lower, and its distribution is close to the zero-error line. Compared to the Jacobi iterative method, the maximum positioning errors in the XYZ directions are reduced from (181.26, 284.74, 305.81) mm to (42.59, 63.71, 67.08) mm, and the average positioning errors from (91.3, 87.38, 117.31) mm to (18.16, 24.67, 27.21) mm, thus greatly improving the positioning accuracy of the robotic arm by 80.11%, 71.78%, and 76.81% in the respective directions.

As can be seen from Table 3, the inverse solution position errors reported in the literature [11,19,31] are significantly lower than those in this paper and in the literature [14,28], primarily due to the smaller size of the electro-mechanical arm and its more stable operation. The average spatial position error of the method in this paper is 32.21 mm. Compared to the

hydraulic robotic arm described in the literature [14,28], the position error of the proposed method in this paper is larger, mainly because of the higher degree of freedom (DOF) and larger size of the robotic arm studied in this paper. An increase in the size of the robotic arm exponentially increases the position error of the end-effector. In addition, the literature [14,28] provides only a single experimental result, which may exhibit a larger positional error. Therefore, the method proposed in this paper can effectively compute the inverse solutions for large hydraulic mechanical arms with high accuracy.

Table 3. System performance comparison between this study and other studies mentioned in the literature.

Study	Method	DOF	Robotic Arm Size/m	Driving Method	Position Error/mm
Proposed	ASWO-BP	7	7.84~11.5	Hydraulic drive	32.21
Zong et al. 2023 [28]	SAGSA+BP	4	3.89	Hydraulic drive	10
Yang et al. 2021 [14]	Improved WHAM+DNN	-	0.5	Water-hydraulic drive	5
Jiang et al. 2017 [31]	PSO-BP	6	0.88	Electric drive	0.176
Cheng et al. 2020 [11]	MQACA-RBF	6	0.25	Electric drive	0.09144
Almusawi et al. 2016 [19]	ANN	6	0.42	Electric drive	0.35

6. Conclusions

In this paper, a robotic arm inverse solution method based on an ASWO-optimized BP neural network is proposed. The ASWO algorithm performs real-time adaptive adjustment of parameters, enhancing the algorithm's adaptability. A mapping relationship between the actual position and the joint angle is established in the neural network model, further improving the positioning accuracy of the inverse solution. The ASWO-BP algorithm improves the convergence accuracy and generalization ability of the neural network in comparison with the GA-BP, PSO-BP, and SWO-BP algorithms. Experimental results show that the ASWO-BP algorithm effectively reduces the spatial distance between the target position and the actual position. Its inverse kinematics solving accuracy is significantly better than that of the Jacobi iteration method. The three-direction end positioning accuracy of the robotic arm is improved by 80.11%, 71.78%, and 76.81%, meeting the actual accuracy requirements of engineering.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.L.; formal analysis, Y.L.; investigation, Y.L. and Q.X.; resources, Y.L.; data curation, Y.L., W.J. and T.Z.; writing—original draft preparation, Y.L.; writing—review and editing, Q.X. and Y.L.; visualization, Y.L.; supervision, Q.X.; project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request. The data are not publicly available due to privacy.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, W.; Chen, D.; Steil, J.J. Analytical Inverse Kinematics Solver for Anthropomorphic 7-DOF Redundant Manipulators with Human-Like Configuration Constraints. *J. Intell. Robot. Syst.* **2017**, *86*, 63–79. [CrossRef]
2. Featherstone, R. Position and Velocity Transformations Between Robot End-Effector Coordinates and Joint Angles. *Int. J. Robot. Res.* **1983**, *2*, 35–45. [CrossRef]

3. Kumar, R.R.; Chand, P. Inverse kinematics solution for trajectory tracking using artificial neural networks for SCORBOT ER-4u. In Proceedings of the 2015 IEEE 6th International Conference on Automation, Robotics and Applications (ICARA), Queenstown, New Zealand, 17–19 February 2015; pp. 364–369.
4. Manocha, D.; Canny, J.F. Efficient inverse kinematics for general 6R manipulators. *IEEE Trans. Robot. Autom.* **1994**, *10*, 648–657. [CrossRef]
5. Manocha, D.; Zhu, Y. A fast algorithm and system for the inverse kinematics of general serial manipulators. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; Volume 4, pp. 3348–3353.
6. Reiter, A.; Müller, A.; Gattringer, H. On Higher Order Inverse Kinematics Methods in Time-Optimal Trajectory Planning for Kinematically Redundant Manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1681–1690. [CrossRef]
7. Xu, Z.; Li, S.; Zhou, X.; Cheng, T. Dynamic neural networks based adaptive admittance control for redundant manipulators with model uncertainties. *Neurocomputing* **2019**, *357*, 271–281. [CrossRef]
8. Li, S.; Zhang, Y. Neural Networks for Robot Arm Cooperation with a Hierarchical Control Topology. In *Proceedings of the SpringerBriefs in Applied Sciences and Technology*; Springer: Singapore, 2018.
9. Li, X.; Xu, Z.; Li, S.; Wu, H.; Zhou, X. Cooperative Kinematic Control for Multiple Redundant Manipulators Under Partially Known Information Using Recurrent Neural Network. *IEEE Access* **2020**, *8*, 40029–40038. [CrossRef]
10. Cagigas-Muñiz, D. Artificial Neural Networks for inverse kinematics problem in articulated robots. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107175. [CrossRef]
11. Cheng, X.; Zhao, M. The Inverse Solution Algorithm and Trajectory Error Analysis of Robotic Arm Based on MQACA-RBF Network. *J. Robot.* **2020**, *2020*, 7807952. [CrossRef]
12. Toquica, J.S.; Oliveira, P.S.; Souza, W.S.; Motta, J.M.S.; Borges, D.L. An analytical and a Deep Learning model for solving the inverse kinematic problem of an industrial parallel robot. *Comput. Ind. Eng.* **2021**, *151*, 106682. [CrossRef]
13. Gholami, A.; Homayouni, T.; Ehsani, R.; Sun, J.Q. Inverse kinematic control of a delta robot using neural networks in real-time. *Robotics* **2021**, *10*, 115. [CrossRef]
14. Yang, C.; Xu, H.; Li, X.; Yu, F. Kinematic modeling and solution of rigid-flexible and variable-diameter underwater continuous manipulator with load. *Robotica* **2021**, *40*, 1020–1035. [CrossRef]
15. Wagaa, N.; Kallel, H.; Mellouli, N. Analytical and deep learning approaches for solving the inverse kinematic problem of a high degrees of freedom robotic arm. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106301. [CrossRef]
16. Gao, R. Inverse kinematics solution of Robotics based on neural network algorithms. *J. Ambient Intell. Hum. Comput.* **2020**, *11*, 6199–6209. [CrossRef]
17. Aravinddhakshan, S.; Apte, S.; Akash, S. Neural network based inverse kinematic solution of a 5 DOF Manipulator for industrial application. *J. Phys. Conf. Ser.* **2021**, *1969*, 012010. [CrossRef]
18. Aydogmus, O.; Boztas, G. Implementation of singularity-free inverse kinematics for humanoid robotic arm using Bayesian optimized deep neural network. *Measurement* **2024**, *229*, 114471. [CrossRef]
19. Almusawi, A.R.; Dülger, L.C.; Kapucu, S. A new artificial neural network approach in solving inverse kinematics of robotic arm (denso vp6242). *Comput. Intell. Neurosci.* **2016**, *2016*, 5720163. [CrossRef] [PubMed]
20. Dereli, S.; Köker, R. IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator. *Sigma J. Eng. Nat. Sci.* **2018**, *36*, 77–85.
21. Ayyıldız, M.; Cetinkaya, K. Comparison of four different heuristic optimization algorithms for the inverse kinematics solution of a real 4-DOF serial robot manipulator. *Neural Comput. Appl.* **2016**, *27*, 825–836. [CrossRef]
22. Khan, A.H.; Li, S.; Chen, D.; Liao, L. Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach. *Neurocomputing* **2020**, *400*, 272–284. [CrossRef]
23. Khan, A.H.; Cao, X.; Li, S.; Katsikis, V.N.; Liao, L. BAS-ADAM: An ADAM based approach to improve the performance of beetle antennae search optimizer. *IEEE/CAA J. Autom.* **2020**, *7*, 461–471. [CrossRef]
24. Dereli, S.; Köker, R. A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: Quantum behaved particle swarm algorithm. *Artif. Intell. Rev.* **2019**, *53*, 949–964. [CrossRef]
25. Pham, D.T.; Castellani, M.; Thi, H.A.L. Nature-Inspired Intelligent Optimisation Using the Bees Algorithm. *Trans. Comput. Collect. Intell.* **2013**, *13*, 38–69.
26. Shi, J.; Mao, Y.; Li, P.; Liu, G.; Liu, P.; Yang, X.; Wang, D. Hybrid Mutation Fruit Fly Optimization Algorithm for Solving the Inverse Kinematics of a Redundant Robot Manipulator. *Math. Probl. Eng.* **2020**, *2020*, 1–13. [CrossRef]
27. Wu, H.; Zhang, X.; Song, L.; Zhang, Y.; Gu, L.; Zhao, X. Wild Geese Migration Optimization Algorithm: A New Meta-Heuristic Algorithm for Solving Inverse Kinematics of Robot. *Comput. Intell. Neurosci.* **2022**, *2022*, 1–38.
28. Zong, Y.; Huang, J.; Bao, J.; Cen, Y.; Sun, D. Inverse kinematics solution of demolition manipulator based on global mapping. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2024**, *238*, 1561–1572. [CrossRef]
29. Bai, Y.; Luo, M.; Pang, F. An Algorithm for Solving Robot Inverse Kinematics Based on FOA Optimized BP Neural Network. *Appl. Sci.* **2021**, *11*, 7129. [CrossRef]
30. Rokbani, N.; Casals, A.; Alimi, A.M. IK-FA, a New Heuristic Inverse Kinematics Solver Using Firefly Algorithm. In *Proceedings of the Computational Intelligence Applications in Modeling and Control*; Springer: Cham, Switzerland, 2015.

31. Jiang, G.; Luo, M.; Bai, K.; Chen, S. A Precise Positioning Method for a Puncture Robot Based on a PSO-Optimized BP Neural Network Algorithm. *Appl. Sci.* **2017**, *7*, 969. [CrossRef]
32. Abdel-Basset, M.; Mohamed, R.; Jameel, M.; Abouhawwash, M. Spider wasp optimizer: A novel meta-heuristic optimization algorithm. *Artif. Intell. Rev.* **2023**, *56*, 11675–11738. [CrossRef]