Trajectory Planning of Shipbuilding Welding Manipulator Based on Improved Whale Optimization Algorithm

Caiping Liang[1], Hao Yuan[2], Chen Wang[3], Wenxu Niu*[4], Yansong Zhang[5]
School of Intelligent Manufacturing and Control Engineering, Shanghai Polytechnic University, Shanghai 201209, China[1,2,3,4]
Thin Plate Structure Manufacturing Institute, Shanghai Jiao Tong University, Shanghai 200240, China[5]

***Abstract*—Time-optimal trajectory planning for shipboard welding robotic arms is a challenging problem due to strong kinematic constraints and the nonlinear coupling between trajectory parameters and execution time. Although various intelligent optimization algorithms have been combined with robotic arm trajectory planning in existing studies, most approaches primarily focus on algorithmic performance improvement and lack a clear formulation of time optimization within polynomial trajectory planning.To address this gap, this study proposes an Improved Whale Optimization Algorithm (IWOA) based on the traditional quintic polynomial trajectory planning method. In the proposed method, the trajectory execution time is explicitly formulated as the optimization objective under kinematic constraints, and the IWOA is designed to stably and efficiently search the time parameter space of the quintic polynomial trajectory. Specifically, chaotic sequence initialization is employed to enhance population distribution, an adaptive weight mechanism is introduced to balance global exploration and local exploitation, and a hybrid co-optimization strategy combining differential evolution and genetic operators is integrated to improve robustness and convergence stability. Simulation experiments are conducted to evaluate the effectiveness of the proposed algorithms. The results demonstrated that, while satisfying robotic arms kinematic constraints, the proposed method achieves an 18.3% reduction in operating time compared with the unoptimized trajectory. These results indicate that the proposed approach provides a systematic and effective solution for time-efficient trajectory planning of shipboard welding robotic arms.**

Keywords*—Shipboard welding robotic arm; quintic polynomial; Improved Whale Optimization Algorithm; time-optimal trajectory planning

I. Introduction

In recent years, with the rapid development of China shipbuilding industry and the increasing complexity of ship hull structures, higher requirements have been imposed on welding quality, efficiency, and consistency during ship construction [1]. Ship welding tasks are typically characterized by large-scale structures, complex curved surfaces, harsh working environments, and long continuous weld seams, which pose significant challenges to traditional manual and semi-automatic welding methods [2]. These conventional approaches are increasingly unable to satisfy the demands for high efficiency, precision, and repeatability required in modern shipbuilding. The application of robotic arm welding technology has emerged as an effective solution to these challenges. Robotic welding systems can significantly improve welding stability, reduce labor intensity, and ensure consistent weld quality, particularly in complex and hazardous environments [3]. With advancements in sensing, control, and intelligent planning technologies, robotic arms have been widely adopted in various industrial manufacturing fields due to their flexibility, efficiency, and high positioning accuracy [4], [5]. Consequently, shipboard welding robotic arms have attracted extensive attention from researchers and industrial practitioners worldwide, leading to notable progress in both system design and motion planning methods [6]. In terms of system development and application, Zhang et al. [7] proposed a magnetic tracked mobile robot for welding vertical steel ship hulls, enabling stable adhesion and motion on large steel surfaces. Liu et al. [8] developed a trajectory and velocity planning method for complex curved-surface welding tasks involving spherical–pipe intersection structures, achieving smooth path transitions and adaptive speed control to improve weld formation quality. Liang et al. [9] designed an autonomous mobile welding robot capable of identifying discontinuous weld seams caused by drainage holes in ship cabins and performing autonomous welding operations. These studies demonstrate the feasibility and effectiveness of robotic welding systems in shipbuilding environments. However, during ship construction, welding tasks often involve plates, pipes, and complex spatial weld seams, which require not only accurate path planning but also high operational efficiency [10]. To further enhance productivity, time-optimal trajectory planning for welding robotic arms has become a critical research topic [11].

Current studies on robotic arm trajectory planning primarily focus on optimizing performance indices such as operating time, energy consumption, smoothness, and dynamic impact [12]. Among these objectives, minimizing execution time while satisfying kinematic and dynamic constraints is of particular importance for improving industrial production efficiency.Traditional trajectory planning methods based on polynomial interpolation typically rely on predefined segment times and path nodes [13]. Although these methods can guarantee trajectory continuity and smoothness, they often lack flexibility and are difficult to achieve true time-optimal performance under complex constraints. In recent years, intelligent optimization algorithms have been increasingly introduced into robotic arm trajectory time optimization, forming an important research direction in motion planning [14]. Commonly used intelligent optimization methods include ant colony optimization [15], sparrow search algorithm [16], particle swarm opti-

mization [17], and whale optimization algorithm [18]. These algorithms have demonstrated certain advantages in improving robotic arm operating efficiency; however, they still suffer from inherent limitations such as premature convergence, susceptibility to local optima, and insufficient global search capability, which ultimately restrict their optimization performance [19].

Inspired by the foraging behavior of humpback whales in bubble webs, the Whale Optimization Algorithm (WOA) has been widely applied to various engineering optimization problems due to its advantages such as simple structure, few control parameters, and fast convergence speed [20]. However, similar to other swarm intelligence algorithms, the standard Whale Optimization Algorithm has inherent defects, including rapid loss of population diversity and easy premature convergence to local optima [21]. These defects are particularly prominent when dealing with high-dimensional, nonlinear, and constrained optimization problems [22]. Although existing research has attempted to alleviate the above problems through isolated strategies such as parameter tuning and fusion with other metaheuristic operators, such improvements are mostly designed for general test functions. For scenarios with strict kinematic constraints and industrial efficiency requirements, such as time-optimal trajectory planning for welding robotic arms, its direct applicability is still very limited [23].

To address these challenges, this study proposes an Improved Whale Optimization Algorithm (IWOA) specifically tailored for the time-optimal trajectory planning problem of welding robotic arms. Unlike existing studies that apply generic hybrid optimization strategies, the proposed IWOA introduces a problem-oriented improvement framework that integrates chaotic initialization, adaptive parameter adjustment, and a coordinated hybrid mechanism combining differential evolution and genetic operators. These strategies are not simply combined, but are systematically designed to address distinct deficiencies of the standard WOA at different stages of the optimization process: chaotic initialization enhances early-stage population diversity, adaptive parameter adjustment dynamically balances global exploration and local exploitation, and the hybrid evolutionary operators improve convergence robustness in the later optimization phase. Comparative simulations conducted in MATLAB demonstrate that, under identical constraints, the proposed IWOA achieves superior time-optimal performance compared to the standard WOA, effectively reducing the execution time of robotic arm trajectories and validating its applicability to industrial welding scenarios.

II. Joint Trajectory Planning

A. Trajectory Interpolation Based on 5th-order Polynomials

The trajectory planning of robotic arms is generally divided into two categories: joint space method and Cartesian space method [24].The joint space method directly interpolates the angles of each joint, which is simple to calculate and easily satisfies joint constraints. The Cartesian space method first plans the pose path of the end effector and then solves the inverse kinematics to obtain the joint motion, resulting in high path accuracy. Ship welding tasks require high precision in motion paths; therefore, this study employs the Cartesian space trajectory planning method. This method can control the motion trajectory of the robotic arm's end effector, achieving the operational objective along a specified path—a method that is difficult to replace using joint space planning [25].Furthermore, this study introduces a fifth-order polynomial interpolation function. By performing fifth-order polynomial fitting on the position, velocity, and acceleration of the starting and ending points, the continuity and smoothness of the joint motion process are ensured, effectively avoiding abrupt changes in velocity and acceleration. The constraints of the fifth-order polynomial interpolation include the starting and ending angles, angular velocities, and angular accelerations, requiring the determination of six coefficients. The joint variables are assumed to satisfy Eq. (1):

$$\begin{cases} \theta_i(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ \dot{\theta}_i(t) = a_1 + 2a_2 t + 3a_3 t^2 + 4a_4 t^3 + 5a_5 t^4 \\ \ddot{\theta}_i(t) = 2a_2 + 6a_3 t + 12a_4 t^2 + 20a_5 t^3 \end{cases} \tag{1}$$

In Eq. (1), $i = 1, 2, 3, \ldots, n$ represents the n joints of the robotic arm; t is the running time of each joint; a_j are the interpolation coefficients of each joint; $\theta_i(t)$, $\dot{\theta}_i(t)$, and $\ddot{\theta}_i(t)$ denote the angular position, angular velocity, and angular acceleration of the robotic arm joint, respectively. The constraints for the fifth-degree polynomial are:

$$\begin{cases} \begin{cases} \theta(t_0) = \theta_0 \\ \theta(t_f) = \theta_f \end{cases} \\ \begin{cases} \dot{\theta}(t_0) = 0 \\ \dot{\theta}(t_f) = 0 \end{cases} \\ \begin{cases} \ddot{\theta}(t_0) = 0 \\ \ddot{\theta}(t_f) = 0 \end{cases} \end{cases} \tag{2}$$

In Eq. (2), t_0 and t_f represent the start time and end time of the robotic arm operation, respectively, and θ_0 and θ_f represent the start angle and end angle, respectively. From Eq. (1) and Eq. (2), the corresponding kinematic equations for each joint can be derived:

$$\begin{cases} \theta_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 + a_4 t_0^4 + a_5 t_0^5 \\ v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 + 4a_4 t_0^3 + 5a_5 t_0^4 \\ \alpha_0 = 2a_2 + 6a_3 t_0 + 12a_4 t_0^2 + 20a_5 t_0^3 \\ \theta_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 + a_4 t_f^4 + a_5 t_f^5 \\ v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 + 4a_4 t_f^3 + 5a_5 t_f^4 \\ \alpha_f = 2a_2 + 6a_3 t_f + 12a_4 t_f^2 + 20a_5 t_f^3 \end{cases} \tag{3}$$

Then, substitute the known θ_0, θ_f, v_0, v_f, α_0, and α_f into Eq. (3) to obtain the parameter matrix of the kinematic equation for each joint:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix} = \begin{bmatrix} 1 & t_0 & t_0^2 & t_0^3 & t_0^4 & t_0^5 \\ 0 & 1 & 2t_0 & 3t_0^2 & 4t_0^3 & 5t_0^4 \\ 0 & 0 & 2 & 6t_0 & 12t_0^2 & 20t_0^3 \\ 1 & t_f & t_f^2 & t_f^3 & t_f^4 & t_f^5 \\ 0 & 1 & 2t_f & 3t_f^2 & 4t_f^3 & 5t_f^4 \\ 0 & 0 & 2 & 6t_f & 12t_f^2 & 20t_f^3 \end{bmatrix}^{-1} \begin{bmatrix} \theta_0 \\ v_0 \\ \alpha_0 \\ \theta_f \\ v_f \\ \alpha_f \end{bmatrix} \tag{4}$$

By substituting the solved coefficient matrix from Eq. (4) into Eq. (1), the spatial trajectory function of the robotic arm joint is obtained.

B. Objective Function and Constraint Conditions

To meet the requirement of improving the operation efficiency of the welding robotic arm in the ship welding process, this study takes the motion cycle of the robotic arm reaching each target point as the optimization objective. Then, the objective function can be expressed as Eq. (5):

$$T = \sum_{i=1}^{n-1} T_i = \sum_{i=1}^{n-1} (t_{i+1} - t_i) \tag{5}$$

In Eq. (5), $i = 1, 2, \ldots, n$ represents n path points, and T represents the running time of the robotic arm end on each segment of the path.

Since the motion constraints in the joint space of the robotic arm are strictly satisfied during its movement, the problem of optimizing the running time of the robotic arm can be formulated as Eq. (6):

$$\min T = T_1 + T_2 + \cdots + T_{n-1} \tag{6}$$

Since the position, velocity, and acceleration of each joint of the robotic arm cannot exceed the rated limits of the motor and reducer during movement, it is necessary to impose physical limit constraints on the robotic arm. The constraint conditions are shown in Eq. (7):

$$\begin{cases} |\theta_i| \le q_{i\max} \\ |\dot{\theta}_i| \le v_{i\max} \\ |\ddot{\theta}_i| \le a_{i\max} \end{cases} \quad (i = 1, 2, \ldots, n) \tag{7}$$

θ_i, $\dot{\theta}_i$, and $\ddot{\theta}_i$ represent the angular position, angular velocity, and angular acceleration of the i-th joint, respectively; $q_{i\max}$, $v_{i\max}$, and $a_{i\max}$ represent the rated upper limits of the angular position, angular velocity, and angular acceleration of the corresponding joint, respectively; n is the total number of joints of the robotic arm.

III. Algorithm Principle

A. Original Whale Optimization Algorithm

The core of the Whale Optimization Algorithm is to imitate three behaviors of whales during predation.

Encircling prey:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}^*(t) - \vec{X}(t) \right| \tag{8}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D} \tag{9}$$

In Eq. (8) and Eq. (9), $\vec{X}^*(t)$ represents the current optimal position, $\vec{X}(t)$ is the current position of the whale, $\vec{D}$ is the distance between the current position and the optimal position, and $\vec{A}$ and $\vec{C}$ are system vectors that control the convergence and search range.

Bubble-net attack:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) \tag{10}$$

In Eq. (10), $\vec{X}(t+1)$ represents the position vector of the individual at generation $t+1$; $\vec{X}^*(t)$ represents the position vector of the optimal individual at generation t; $\vec{D}' = |\vec{X}^*(t) - \vec{X}(t)|$ represents the distance between the current individual and the optimal individual; b is a constant that controls the shape of the logarithmic spiral; l is a random number within the interval $[-1, 1]$.

Searching for prey:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X}(t) \right| \tag{11}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D} \tag{12}$$

As shown in Eq. (11) and Eq. (12), $\vec{X}_{rand}$ is the position of a randomly selected individual in the population. $\vec{A} = 2a \cdot r - a$, where a decreases linearly from 2 to 0 during the iteration process. $\vec{C} = 2 \cdot r$, where r is a random number in the interval $[0, 1]$. When $|\vec{A}| < 1$: the whale tends to the current optimal solution; when $|\vec{A}| \ge 1$: the whale randomly searches for other prey.

B. Improvement Mechanism

Due to the shortcomings of the WOA, such as being prone to premature convergence and falling into local optima, as well as strong randomness leading to unstable results, this study proposes the following improvement strategies to enhance the performance of the original Whale Optimization Algorithm in robotic arm trajectory planning:

1) Chaotic initialization: First, the Tent chaotic map is used to replace random initialization to avoid the problem of uneven initial distribution of the population.

$$x_{k+1} = \begin{cases} \dfrac{x_k}{0.5}, & 0 \le x_k < 0.5 \\ \dfrac{1 - x_k}{0.5}, & 0.5 \le x_k \le 1 \end{cases} \tag{13}$$

As shown in Eq. (13), the Tent chaotic map has ergodicity and pseudo-randomness, which can ensure that individuals are evenly distributed in the search space, thereby improving the diversity of the initial population. In addition, to enhance the local exploitation ability, a small chaotic disturbance is applied to the current global optimal solution during the iteration, and the disturbance amplitude is gradually reduced to prevent the algorithm from stagnating at a local optimum.

2) Adaptive parameter adjustment: To address the limitation of linear decrement of parameters in the Whale Optimization Algorithm, this study designs an adaptive control mechanism based on nonlinear functions and cosine functions:

$$a_t = 2\left(1 - \left(\frac{t}{T}\right)^2\right) \tag{14}$$

a_t is the control parameter at the t-th iteration, where t is the current number of iterations and T is the maximum number of iterations. Eq. (14) allows the algorithm to maintain stronger exploration ability in the early stage and gradually enhance exploitation in the later stage.

The spiral control factor b_t is defined as Eq. (15):

$$b_t = 1 + 1.5\left(1 - \frac{t}{T}\right) \tag{15}$$

which dynamically adjusts the spiral search behavior during the optimization process. In addition, the concept probability parameter p_t, which controls the selection between encircling search and spiral search, is defined as Eq. (16):

$$p_t = 0.5 + 0.5\cos\left(\pi\frac{t}{T}\right) \tag{16}$$

Eq. (16) controls the selection probability of encircling search and spiral search, achieving a dynamic balance between exploration and exploitation.

3) Integration of differential evolution and genetic operators: To enhance the ability to jump out of local optima, this study introduces the crossover operators of differential evolution and genetic algorithm into the Whale Optimization Algorithm after its iterative update.

Differential evolution and genetic operator fusion:

The rand/1/bin strategy is adopted to generate candidate individuals, improving the directionality of the search. The mathematical model is as follows:

$$V_i = X_{r1} + F \cdot (X_{r2} - X_{r3}) \tag{17}$$

$$U_{i,j} = \begin{cases} V_{i,j}, & \text{rand} \le CR \\ X_{i,j}, & \text{otherwise} \end{cases} \tag{18}$$

Eq. (17) and Eq. (18) enhance the directional search ability of the population. Among them, V_i represents the mutation vector of the i-th candidate individual, i.e., the newly generated candidate solution. X_{r1}, X_{r2}, X_{r3}: three different individuals randomly selected from the population, and different from the current individual X_i. F is the scaling factor, usually ranging from 0.4 to 0.90, and CR is the crossover probability, usually ranging from 0.6 to 0.95.

In addition to DE operators, arithmetic crossover and Gaussian mutation from genetic algorithms are introduced to improve local search capability. The arithmetic crossover is defined as Eq. (19):

$$X' = \alpha X_i + (1-\alpha)X_j, \alpha \in [0,1] \tag{19}$$

X' is a new candidate solution generated by arithmetic crossover, and α, as a crossover coefficient, is used to control the contribution proportion of the two parent individuals X_i and X_j to the new individual X'. X_i and X_j are the two original candidate solutions involved in the crossover. Eq. (19) enables the fusion of two parent individuals to generate a new candidate solution through linear interpolation, thereby increasing the diversity of the population and providing a foundation for subsequent improvement of local search ability.

C. Steps of the Improved Whale Optimization Algorithm

1) Initialize the number of whales and define relevant parameters.

2) Use the Tent chaotic map to generate the initial population, replacing the traditional random initialization to improve population diversity and global coverage capability.

3) During the iteration process, adaptively and nonlinearly adjust parameters a, b, and p to ensure exploration capability in the early stage and enhance convergence in the later stage.

4) Update the position: update the whale position according to Eq. (9) and Eq. (10) to approach the optimal solution. Perform spiral update according to Eq. (11), and then conduct global random search according to Eq. (12) and Eq. (13).

5) After the update, apply evolutionary operators to some individuals: use the rand/1/bin strategy for mutation and crossover to generate candidate solutions.

6) Perform arithmetic crossover and Gaussian perturbation on some individuals to enhance local exploitation capability.

7) Update the global optimal solution and fitness. If the maximum number of iterations T is reached, terminate the loop and output the optimal solution; otherwise, return to step 3 to continue the search.

The algorithm flowchart is shown in Fig. 1.

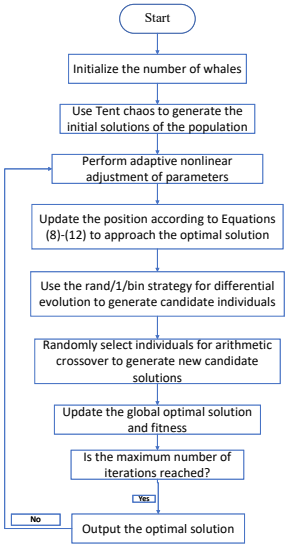
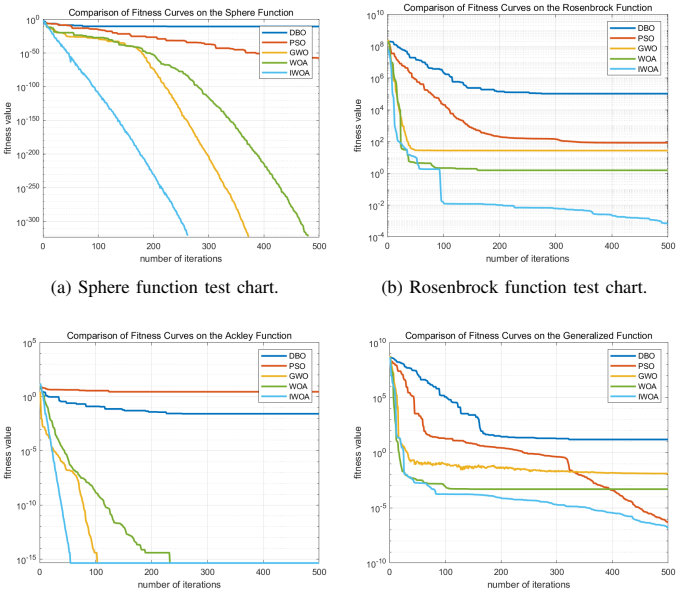


Fig. 1. Algorithm flowchart.

D. Validation of Algorithm Effectiveness

In this study, four classic CEC test functions are selected to verify the effectiveness of the Improved Whale Optimization Algorithm [26]: Sphere (F1), Rosenbrock (F5), Ackley (F10), and Generalized (F12) functions are used for test evaluation. These functions cover both unimodal and multimodal cases, and can evaluate the basic convergence performance, local



(a) Sphere function test chart.

(b) Rosenbrock function test chart.

(c) Ackley function test chart.

(d) Generalized function test chart.

Fig. 2. Comparison chart of different benchmark function algorithm tests.

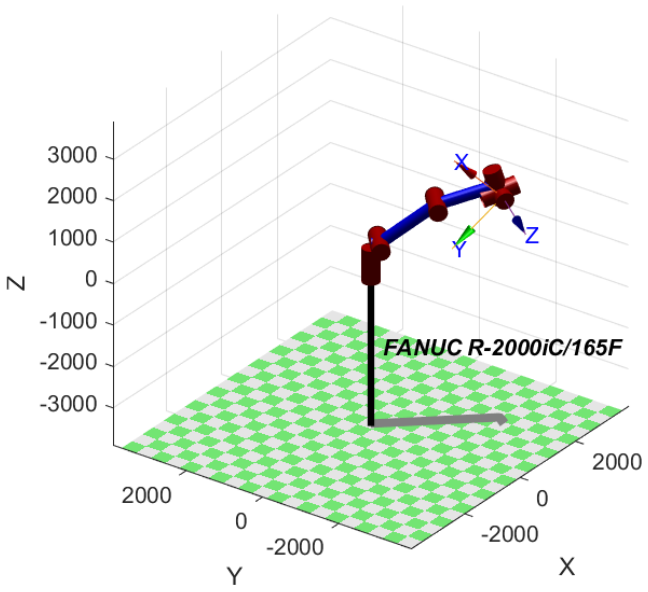


Fig. 3. Robotic arm model.

search ability and global search ability of the algorithm. The formulas of the selected test functions are shown in Table I.

In this algorithm validation, the IWOA is compared with the WOA, Dung Beetle Optimization algorithm (DBO), Particle Swarm Optimization algorithm (PSO), and Grey Wolf Optimization algorithm (GWO). The experimental parameters are set as follows: the population size is 50, the maximum number of iterations is 500, and the optimization dimension is 40. Each algorithm is run 30 times and the average results are taken to reduce the impact of randomness on the experimental results. The test results of each test function are shown in Table II.

It can be seen from Table II that IWOA exhibits excellent global optimization and convergence performance on all test functions. The best value is close to or reaches 0, the mean value is stable and the lowest, the standard deviation is the smallest, the result fluctuation is the least, and the convergence stability is the best. Fig. 2 shows the fitness iteration convergence diagram of each function.

The Sphere function is used to test the basic convergence performance of the algorithm. As can be seen from Fig. 2a, IWOA converges faster in the early stage and finally reaches the optimal value of 0, indicating that it has good global convergence ability. The Rosenbrock function is used to evaluate the local search performance. Fig. 2b shows that the algorithm decreases rapidly in the early stage, which benefits from the combination of Kent chaotic mapping and adaptive parameters. In contrast, the remaining algorithms converge slowly and are prone to falling into local optima. The Ackley function is used to verify the ability to escape from local optima. Fig. 2c shows that IWOA can quickly get rid of local traps and converge stably, while other algorithms are prone to stagnation. The Generalized function is used to measure the balance between global and local search. Fig. 2d shows that IWOA converges smoothly without premature convergence, showing stronger stability and global optimization performance.

IV. Simulation and Analysis

A. Robotic Arm Model

To verify the effectiveness of the algorithm, this study takes the R-2000iC/165F robotic arm of the industrial robot FANUC as the research object. First, the D-H modeling method is used to construct the robotic arm model, as shown in Fig. 3. The model includes a base and six robotic arm joints. The relevant parameters of the robotic arm model are shown in Table III, where i represents the link sequence number; a_{i-1} represents the link length, which is the length of the common normal between two rotational joint axes; the link twist α_{i-1} is the offset angle of the two joint axes; the link offset d_i is the vertical offset between two adjacent joints, that is, the offset distance along the z-axis; the joint angle θ_i is the rotation amount of the i-th joint relative to the initial position. The schematic diagram of the robotic arm joint coordinate system is shown in Fig. 4.

After establishing the coordinate system of the 6-degree-of-freedom robotic arm using D-H parameters, the obtained D-H transformation matrix needs to satisfy Eq. (20).

$$ {}^{i+1}_{i}A = \begin{bmatrix} \cos\theta_{i+1} & -\sin\theta_{i+1} & 0 & a_i \\ \sin\theta_{i+1}\cos\alpha_i & \cos\theta_{i+1}\cos\alpha_i & -\sin\alpha_i & -\sin\alpha_i d_{i+1} \\ \sin\theta_{i+1}\sin\alpha_i & \cos\theta_{i+1}\sin\alpha_i & \cos\alpha_i & \cos\alpha_i d_{i+1} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{20} $$

${}^{i+1}_{i}A$ represents the homogeneous transformation matrix from d_i to d_{i+1}. By substituting the corresponding parameters in the D-H parameters into Eq. (20), the corresponding transformation matrix is obtained, and then $\{{}^{1}_{0}A, {}^{2}_{1}A, {}^{3}_{2}A, {}^{4}_{3}A, {}^{5}_{4}A, {}^{6}_{5}A\}$ is calculated, as shown in Eq. 21.

TABLE I. TEST FUNCTIONS

Function no.	Function name	Function formula	Search space	Optimal value
F1 (Unimodal)	Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]$	0
F5 (Unimodal)	Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-30, 30]$	0
F10(Multimodal)	Ackley	$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]$	0
F12(Multimodal)	Generalized	$f_{12}(x) = \frac{\pi}{n}\left[10\sin^2\left(\pi\frac{x_1+1}{4}\right) + \sum_{i=1}^{n-1}\left(\frac{x_i+1}{4}\right)^2\right] \left(1 + 10\sin^2\left(\pi\frac{x_n+1}{4}\right)\right) + \left(\frac{x_n+1}{4}\right)^2 + \sum_{i=1}^{n} U(x_i, 10, 100, 4)$	$[-50, 50]$	0

TABLE II. COMPARISON OF TEST FUNCTION RESULTS OF DIFFERENT ALGORITHMS

No.	Index	IWOA	WOA	GWO	PSO	DBO
F1	Best	0	0	0	3.472×10^{-57}	2.98×10^{-14}
	Worst	7.294×10^{-287}	7.294×10^{-287}	0	9.676×10^{-26}	9.931×10^{3}
	Mean	0	1.495×10^{-289}	9.676×10^{-26}	2.460×10^{-28}	3.187×10^{3}
	Std	0	0	0	4.465×10^{-27}	2.854×10^{3}
F5	Best	1.755×10^{-4}	2.53	2.721×10^{1}	9.003×10^{4}	2.196×10^{8}
	Worst	5.596×10^{-4}	2.66	2.721×10^{1}	9.003×10^{4}	7.573×10^{8}
	Mean	2.459×10^{-4}	2.536	2.721×10^{1}	9.003×10^{4}	4.754×10^{8}
	Std	8.641×10^{-5}	2.172×10^{-2}	7.178×10^{-15}	3.325×10^{-1}	1.244×10^{8}
F10	Best	4.441×10^{-16}	4.441×10^{-16}	4.441×10^{-16}	3.856	1.474×10
	Worst	4.441×10^{-16}	4.441×10^{-16}	4.441×10^{-16}	2.050×10	2.213×10
	Mean	4.441×10^{-16}	4.441×10^{-16}	4.441×10^{-16}	1.445×10	2.088×10
	Std	0	0	0	3.242	8.693×10^{-1}
F12	Best	2.564×10^{-7}	1.003×10^{-3}	1.969×10^{-2}	8.325×10^{-1}	3.615×10^{8}
	Worst	2.834×10^{-7}	1.009×10^{-3}	1.969×10^{-2}	8.347×10^{-1}	2.389×10^{9}
	Mean	2.714×10^{-7}	1.003×10^{-3}	1.969×10^{-2}	8.326×10^{-1}	1.146×10^{9}
	Std	4.241×10^{-9}	1.135×10^{-6}	3.494×10^{-18}	2.626×10^{-4}	4.120×10^{8}

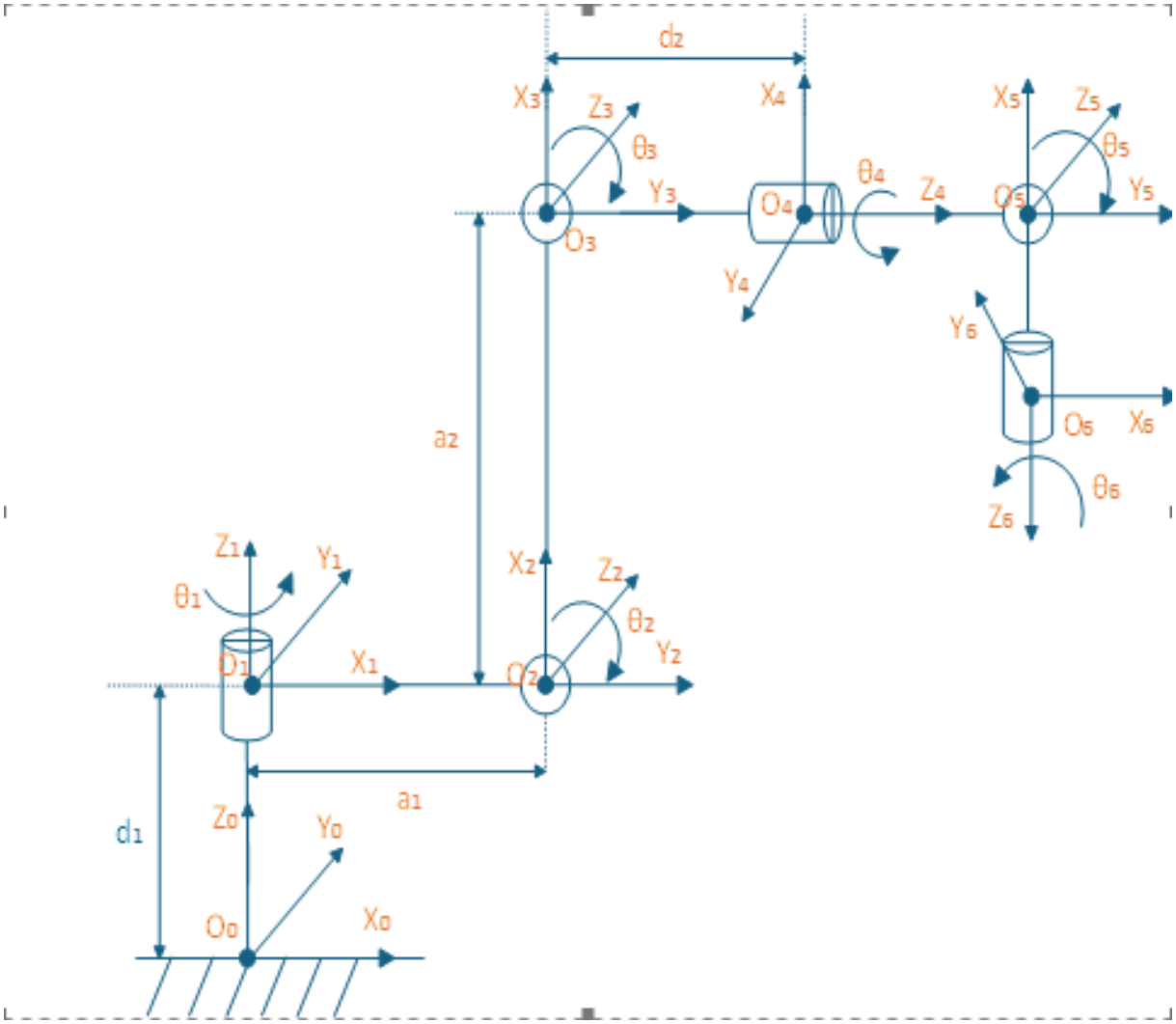


Fig. 4. Robotic arm joint coordinate system.

TABLE III. D-H PARAMETERS OF THE ROBOTIC ARM

Joint No. (i)	a_{i-1}	α_{i-1}	d_i	θ_i
1	175	$-\pi/2$	490	θ_1
2	1500	0	0	θ_2
3	1350	$-\pi/2$	0	θ_3
4	0	$\pi/2$	162	θ_4
5	0	$-\pi/2$	0	θ_5
6	0	0	220	θ_6

$$
\begin{aligned}
{}^{0}_{1}A &= \begin{bmatrix} C1 & -S1 & 0 & 175 \\ 0 & 0 & 1 & 490 \\ -S1 & -C1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix};
{}^{2}_{1}A = \begin{bmatrix} C2 & -S2 & 0 & 1500 \\ S2 & C2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};
{}^{3}_{2}A = \begin{bmatrix} C3 & -S3 & 0 & 1350 \\ 0 & 0 & 1 & 0 \\ -S3 & -C3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \\
{}^{4}_{3}A &= \begin{bmatrix} C4 & -S4 & 0 & 0 \\ 0 & 0 & -1 & -162 \\ S4 & C4 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};
{}^{5}_{4}A = \begin{bmatrix} C5 & -S5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S5 & -C5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix};
{}^{6}_{5}A = \begin{bmatrix} C6 & -S6 & 0 & 0 \\ S6 & C6 & 0 & 0 \\ 0 & 0 & 1 & 220 \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{21}
$$

In Eq. 22, $C1$ represents $\cos\theta_1$, and $S1$ represents $\sin\theta_1$. ${}^{6}_{0}A$ denotes the total transformation matrix from the robot base to the end-effector, that is, the position of the end of the link in coordinate system O_0 relative to the base, which is obtained in matrix form through Eq. 22.

$$ {}^{6}_{0}A = {}^{1}_{0}A^{2}_{1}A^{3}_{2}A^{4}_{3}A^{5}_{4}A^{6}_{5}A \tag{22} $$

By establishing the link structure model of the robotic arm using the D-H parameter method, the position and posture of the robotic arm's end-effector in the coordinate system can be obtained through the forward kinematics equations under the premise that each joint angle is known.

B. Robotic arm Trajectory Planning

To verify the effectiveness of the Improved Whale Optimization Algorithm in robotic arm trajectory planning, three waypoints are specified within the kinematic constraint space of the FANUC R-2000iC/165F robotic arm. Fifth-order polynomial interpolation is adopted for trajectory planning to ensure continuous velocity and acceleration of the robotic arm. On this basis, the Improved Whale Optimization Algorithm is used to optimize the trajectory time of the robotic arm, and a comparison is made with the original whale optimization algorithm. Table IV shows the Cartesian space information of the end-effector of the FANUC R-2000iC/165F robotic arm passing through the three specified waypoints. Then, the joint position information of the waypoints in the robotic arm is obtained through inverse kinematics analysis, and Table V presents the corresponding joint position information of these three waypoints.

During the process of passing through two path segments, the operation time of each joint in each path segment is set to 5 seconds. Meanwhile, to meet the constraint conditions of the robotic arm, the maximum angular velocity of the robotic arm joints is set to $\dot{\theta}_{j\max} = 3\,\text{rad/s}$, and the maximum angular acceleration is set to $\ddot{\theta}_{j\max} = 3.5\,\text{rad/s}^2$. According to the requirements of the ship spot welding process, the velocity of the robotic arm must be 0 when passing through each welding waypoint. By substituting the above three interpolation points into the fifth-order polynomial interpolation function and using MATLAB R2024a for trajectory planning of the robotic arm, the trajectory of the end-effector of the robotic arm in the workspace is obtained, as shown in Fig. 5. The position, velocity, and acceleration curves of the six joints after fifth-order polynomial trajectory planning are shown in Fig. 6.

It can be observed from Fig. 6 that the interpolation time combination of the fifth-order polynomial is (5, 5), so the total time is 10 s. Among them, the maximum angular velocity is 1.53 rad/s, and the maximum angular acceleration is 0.94 rad/s², which meet the constraint conditions of the robotic arm. However, to further improve the operating efficiency of the system, it is still necessary to optimize the trajectory to shorten the operating time on the basis of ensuring trajectory smoothness.

C. Optimal Time Planning of Robotic Arm

This study adopts the IWOA for time optimization of the robotic arm trajectory. In the experiment, the population size is set to 30, and the maximum number of iterations is 300. Due to the randomness of trajectory planning results, each algorithm is run 50 times, and the average value is taken as the performance evaluation index. To verify the algorithm performance, the motion time curves of joint angle, angular velocity, and angular acceleration between IWOA and the original WOA are compared and analyzed, as shown in Fig. 7.

It can be seen from Fig. 7 that the displacement, velocity, and acceleration curves of the robotic arm optimized by the WOA and the IWOA all show smooth and continuous changes without obvious mutations, indicating that the robotic arm can pass through each waypoint stably. Meanwhile, by sorting out the optimized data of each joint in Fig. 6 and Fig. 7, the optimization times of the fifth-order polynomial, WOA, and IWOA when passing through each waypoint are obtained, as shown in Table VI.

It can be seen from Table VI that the time combination of the fifth-order polynomial is (5, 5) with a total time of 10 s. The optimal time combination of WOA is (3.26, 2.48) and the total time is 5.74 s. The optimal time combination of IWOA is (2.76, 1.93) with a total time of 4.69 s. Both the WOA and IWOA algorithms can achieve the effect of optimizing the operation time, but the IWOA is more effective in optimization performance, which greatly improves the working efficiency of the robotic arm.

V. Experiments

A. Experimental Purpose

To further verify the feasibility and effectiveness of the robotic arm trajectory planning based on the Improved Whale Optimization Algorithm (IWOA) in actual industrial environments, this chapter designs a comparative experiment based on the RoboGuide simulation platform and the FANUC physical robotic arm system. By executing the same path task in both virtual and real environments, the performance of the trajectory generated by the algorithm in terms of path accuracy and time optimization effect is evaluated, thereby verifying the algorithm's implementability and robustness.

The simulation completed in the MATLAB environment is mainly used to verify the performance of IWOA-DEGA at the mathematical model level, including the time optimality and path smoothness of trajectory planning. At this stage, the focus is on evaluating the algorithm's global optimization capability and optimization effect. However, simulations only in MATLAB cannot fully reflect the execution feasibility of the algorithm in real robotic arm systems.

For this reason, this study further conducts engineering-level simulation in the FANUC RoboGuide platform. RoboGuide can realistically simulate the geometric structure, joint motion constraints, and dynamic characteristics of the robotic arm, and supports the modeling of external workpieces, fixtures, sensors, and other elements. By importing the optimal trajectory generated by IWOA into RoboGuide and setting a cuboid workpiece model consistent with the laboratory environment, the executability, safety, and spatial rationality of the planned trajectory in the actual space can be effectively verified.

TABLE IV. CARTESIAN SPACE INFORMATION CORRESPONDING TO THREE SPECIFIED WAYPOINTS

Waypoint	x	y	z
X_1	0.1	0.1	0.5
X_2	-1.5	0.75	2.344
X_3	0.5	0.5	2.0

TABLE V. JOINT INFORMATION CORRESPONDING TO EACH WAYPOINT (RAD)

Joint	X_1	X_2	X_3
1	-1.0228	2.6619	0.5174
2	-2.8352	-1.5037	-2.1353
3	-3.0950	0.9949	1.7531
4	1.2294	-0.4553	1.3274
5	2.4167	-0.2435	1.0696
6	0	0	0

TABLE VI. OPTIMIZATION RESULTS OF JOINT MOTION TIME

Method	$t_1(s)$	$t_2(s)$	Total Time $t(s)$
Fifth-order polynomial	5	5	10
WOA	3.26	2.48	5.74
IWOA	2.76	1.93	4.69

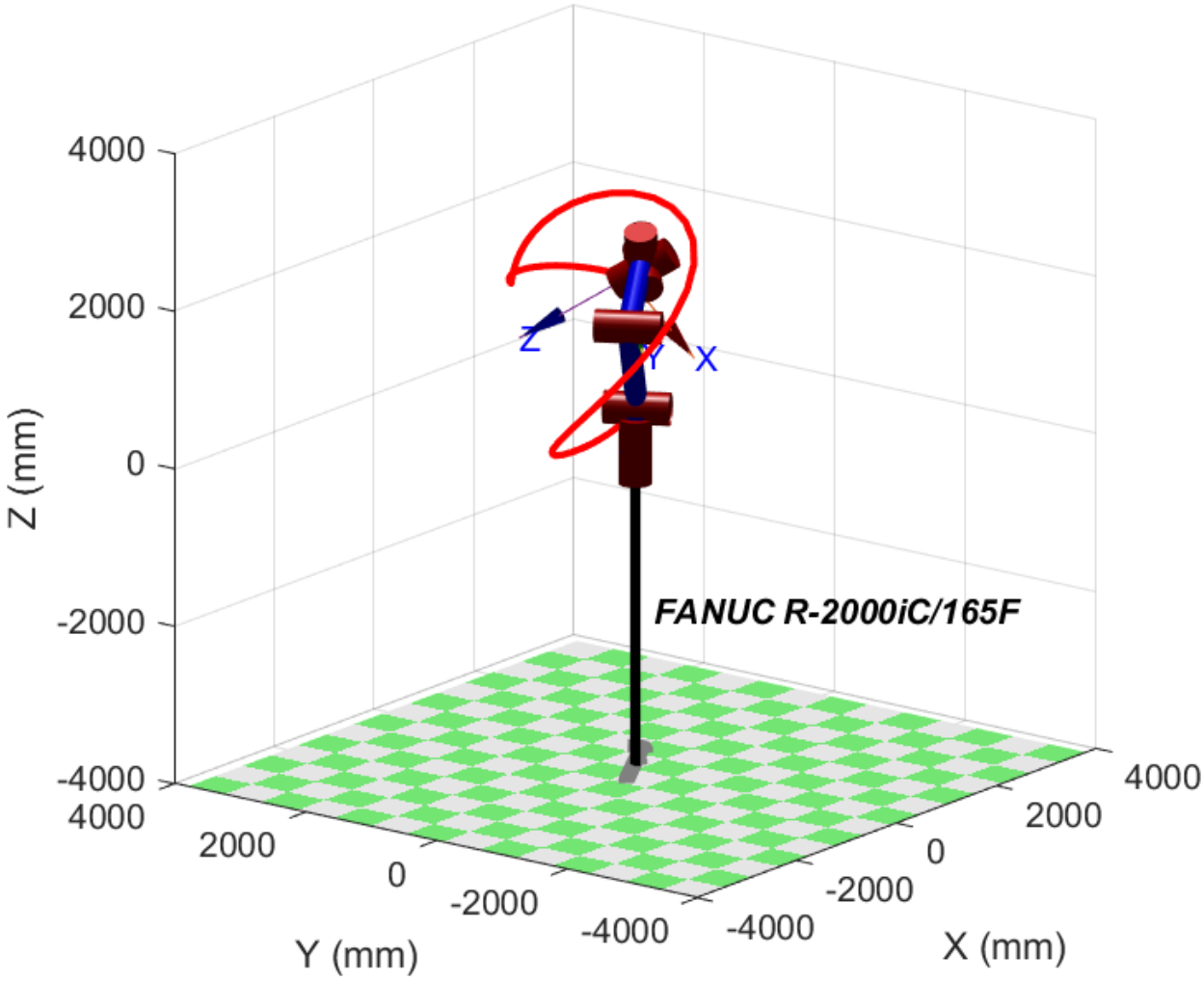


Fig. 5. Robot arm motion trajectory.

B. Experimental Platform and Equipment

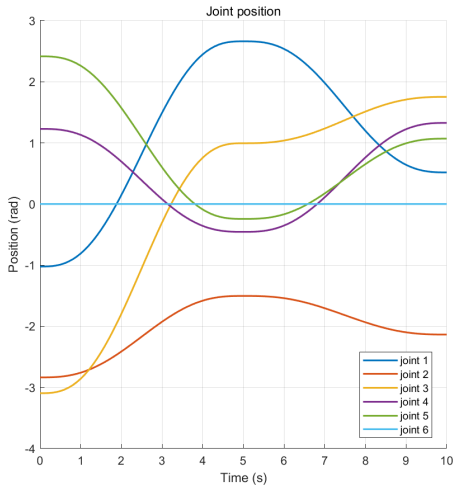
1) Simulation experiment platform: The simulation experiment is carried out based on the FANUC RoboGuide software. This platform can realize 3D modeling, path planning, collision detection, and offline programming of the robotic arm. The robotic arm model used in the experiment is FANUC R-2000iC/165F, and its virtual model is established in RoboGuide. A cuboid workpiece with dimensions of L×W×H = 300 mm × 200 mm × 150 mm is placed as the operation object. Four fixed waypoints (P1, P2, P3, P4) are selected on the surface of the workpiece, and the robotic arm needs to pass through these four waypoints in sequence to complete the trajectory movement.

2) Laboratory physical platform: The physical experiment constructs a test environment consistent with the simulation in the laboratory. The same FANUC R-2000iC/165F robotic arm is adopted, driven and controlled by the R-30iB control system. A cuboid workpiece with the same dimensions and position as in the simulation is placed on the test bench to ensure the consistency of experimental conditions. The robotic arm control program is generated through offline programming in RoboGuide and imported into the physical control system via the teach pendant for execution.

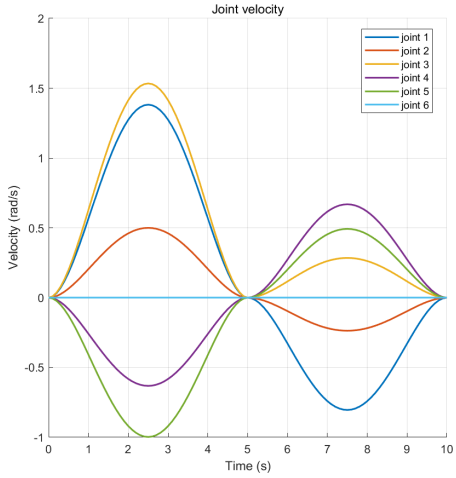
C. Experimental Results and Analysis

1) Simulation results: Fig. 8 shows the changes in the motion posture of the robotic arm when it passes through waypoints P1–P4 in sequence during the RoboGuide simulation. It can be seen from the figure that the end effector of the robotic arm moves smoothly and continuously without oscillation or sudden changes, indicating that the trajectory generated by IWOA has good dynamic smoothness and control stability. Consistent with the results obtained in the MATLAB simulation, the RoboGuide simulation further verifies the effectiveness of IWOA in terms of time optimization. According to the operation time statistics output by RoboGuide, the average operation time of the trajectory planned by IWOA is 18% less than that of the unoptimized trajectory, and the acceleration and deceleration transitions between each path segment are smoother, which effectively reduces the idle travel time of the robotic arm and unnecessary speed fluctuations.

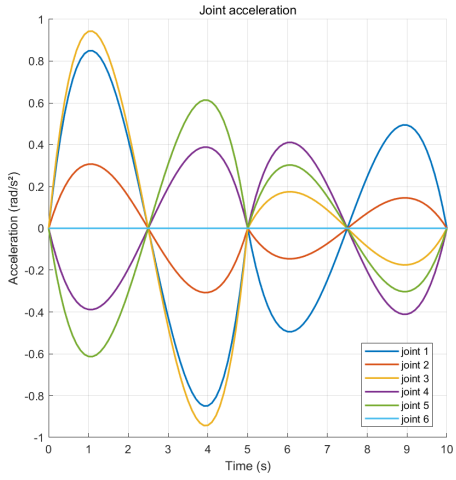
2) Physical experiment results: Fig. 9 shows the motion states of the FANUC R-2000iC/165F robotic arm in the laboratory as it passes through waypoints P1–P4 in sequence.



(a) Angular displacement variation curves of each joint.

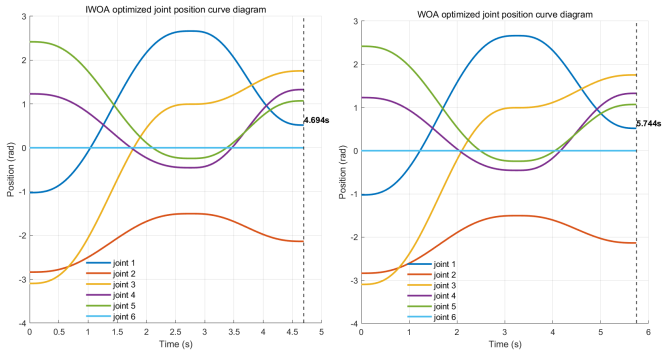


(b) Velocity variation curves of each joint.

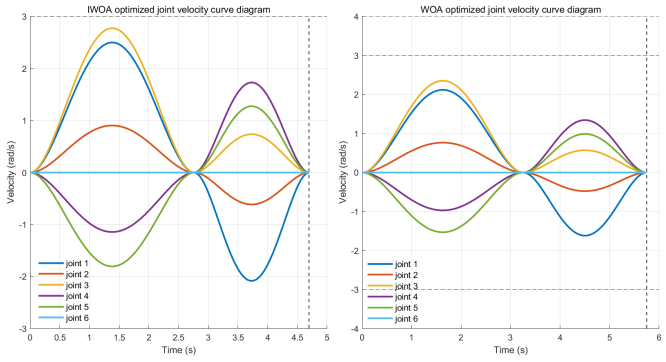


(c) Acceleration variation curves of each joint.

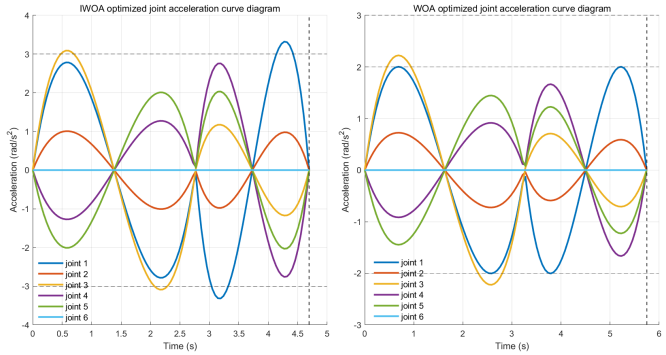
Fig. 6. Motion curve of six-degree-of-freedom robotic arm trajectory planning based on quintic polynomial interpolation.



(a) Comparison diagram of angular displacement curves of each joint.



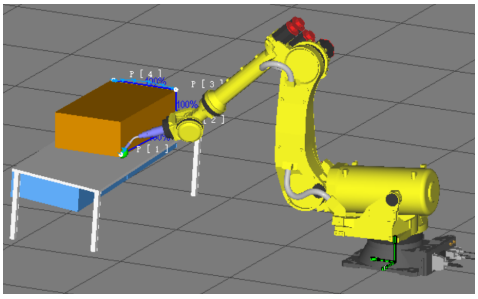
(b) Comparison diagram of angular velocity curves of each joint.



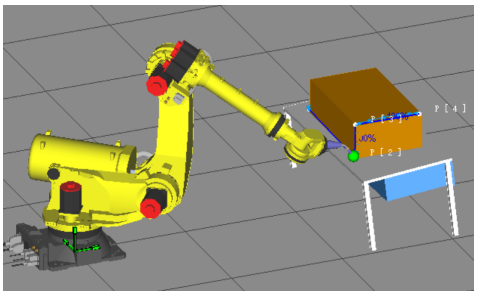
(c) Comparison diagram of angular acceleration curves of each joint.

Fig. 7. Comparison diagram of joint motion curves before and after the improved algorithm.

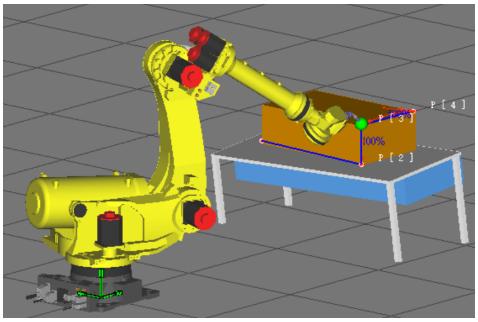
In the experiment, the robotic arm control program is generated by offline programming in RoboGuide, ensuring that the experimental path is completely consistent with the simulation path. It can be seen from the experimental results that the robotic arm moves smoothly and maintains consistent postures between each waypoint, with the end position error controlled within ±2 mm. More importantly, the operation time trend of the physical test is consistent with the simulation results. The trajectory optimized by IWOA reduces the average operation time by 16.4% compared with the original trajectory for the same task, verifying the time optimization effect of the algorithm in actual industrial control systems.



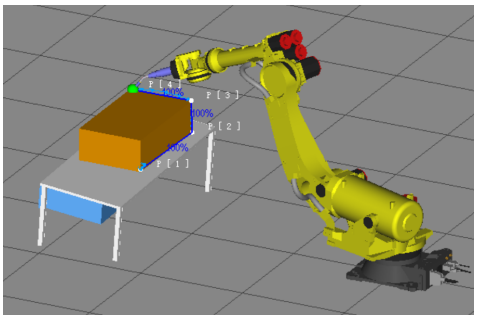
(a) The robotic arm reaches waypoint P1 in RoboGuide.



(b) The robotic arm reaches waypoint P2 in RoboGuide.



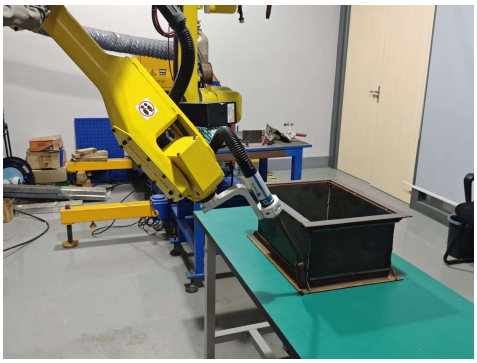
(c) The robotic arm reaches waypoint P3 in RoboGuide.



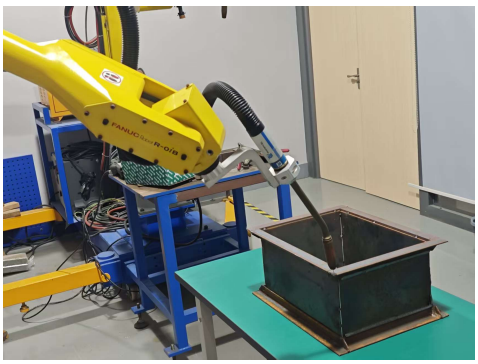
(d) The robotic arm reaches waypoint P4 in RoboGuide.

Fig. 8. Trajectory diagram of the robotic arm in RoboGuide simulation.

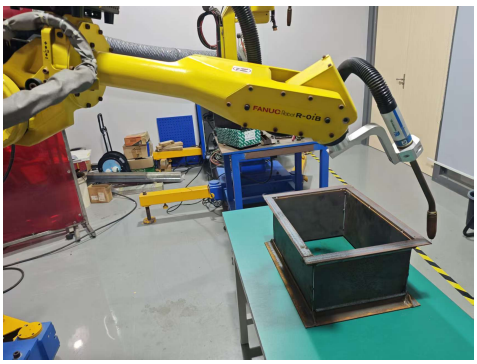
(a) The laboratory robotic arm reaches waypoint P1.



(b) The laboratory robotic arm reaches waypoint P2.



(c) The laboratory robotic arm reaches waypoint P3.



(d) The laboratory robotic arm reaches waypoint P4.

Fig. 9. Trajectory diagram of the laboratory robotic arm.

Through comparison, it is found that the physical experiment results are highly consistent with the RoboGuide simulation results in terms of path shape, attitude change, and operation time. This indicates that the IWOA can effectively generate time-optimal and executable robotic arm trajectories in real-world environments, demonstrating excellent engineering feasibility and application potential.

VI. Conclusion

This study proposes a trajectory planning for the FANUC R-2000iC/165F welding robotic arm based on an Improved Whale Optimization Algorithm (IWOA). The main theoretical contribution of this work lies in integrating quintic polynomial trajectory planning with the IWOA for time-optimal trajectory generation, thereby enhancing both the convergence speed and global optimization capability of traditional metaheuristic algorithms in robotic arm applications. Simulation and experimental results demonstrate that the IWOA significantly outperforms the original WOA in terms of trajectory optimization. Under identical kinematic constraints, the IWOA reduces the total operation time from 10 s to 4.69 s, compared with 5.74 s achieved by WOA. In RoboGuide simulation, the IWOA-optimized trajectory reduces average operation time by 18%, while physical experiments confirm a 16.4% reduction, with end-effector positioning errors controlled within ±2 mm. The results highlight the algorithm's effectiveness in generating smooth, continuous, and executable trajectories while maintaining operational safety. This also demonstrates that the metaheuristic algorithm employing chaotic mapping and adaptive parameter enhancement can effectively address the trade-off between global search and local optimization in high-dimensional trajectory optimization problems.

Despite these contributions, several limitations exist. The current study considers only a single robotic arm model and a limited number of waypoints, and does not account for dynamic external disturbances or varying payloads, which may affect real-world performance. Future research will extend the method to multi-arm coordinated welding tasks, incorporate dynamic obstacle avoidance, and explore real-time adaptive trajectory optimization under uncertain industrial environments.

References

[1] H. Zhang, Q. Zhuang, and G. Li, "Robot path planning method based on indoor spacetime grid model," vol. 16, 2024.

[2] S. Dian, J. Zhong, B. Guo, J. Liu, and R. Guo, "A smooth path planning method for mobile robot using a bes-incorporated modified qpso algorithm," *Expert Systems with Applications*, vol. 208, p. 118256, 2022.

[3] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, p. 120254, 2023.

[4] S. Wu, Q. Ze, J. Dai, N. Udipi, G. H. Paulino, and R. Zhao, "Stretchable origami robotic arm with omnidirectional bending and twisting," *Proceedings of the National Academy of Sciences*, vol. 118, no. 36, p. e2110023118, 2021.

[5] J. Meng, S. Zhang, A. Bekyo, J. Olsoe, B. Baxter, and B. He, "Noninvasive electroencephalogram based control of a robotic arm for reach and grasp tasks," *Scientific reports*, vol. 6, no. 1, p. 38565, 2016.

[6] Y. Liu, Q. Tang, X. Tian, and S. Yang, "A novel offline programming approach of robot welding for multi-pipe intersection structures based on nsga-ii and measured 3d point-clouds," *Robotics and computer-integrated manufacturing*, vol. 83, p. 102549, 2023.

[7] X. Zhang, M. Zhang, S. Jiao, L. Sun, and M. Li, "Design and optimization of the wall climbing robot for magnetic particle detection of ship welds," *Journal of Marine Science and Engineering*, vol. 12, no. 4, p. 610, 2024.

[8] Y. Liu, Q. Tang, and X.-C. Tian, "Model-driven path planning for robotic plasma cutting of branch pipe with single y-groove based on pipe-rotating scheme," *Advances in Manufacturing*, vol. 12, no. 1, pp. 94–107, 2024.

[9] L. Guo and H. Zhang, "Autonomous mobile welding robot for discontinuous weld seam recognition and tracking," *The International Journal of Advanced Manufacturing Technology*, vol. 119, no. 7, pp. 5497–5509, 2022.

[10] J. Liu, Y. Cheng, X. Jing, X. Liu, and Y. Chen, "Prediction and optimization method for welding quality of components in ship construction," *Scientific Reports*, vol. 14, no. 1, p. 9353, 2024.

[11] Y. Du and Y. Chen, "Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization," *Chinese Journal of Electronics*, vol. 31, no. 5, pp. 906–914, 2022.

[12] H. Jiachen and F. Li-hui, "Robot path planning based on improved dung beetle optimizer algorithm," *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, vol. 46, no. 4, p. 235, 2024.

[13] F. Ding, R. Wang, T. Zhang, G. Zheng, Z. Wu, and S. Wang, "Real-time trajectory planning and tracking control of bionic underwater robot in dynamic environment," *Cyborg and Bionic Systems*, vol. 5, p. 0112, 2024.

[14] N. Wu, D. Jia, Z. Li, and Z. He, "Trajectory planning of robotic arm based on particle swarm optimization algorithm," *Applied Sciences*, vol. 14, no. 18, p. 8234, 2024.

[15] Q. Luo, H. Wang, Y. Zheng, and J. He, "Research on path planning of mobile robot based on improved ant colony algorithm," *Neural Computing and Applications*, vol. 32, no. 6, pp. 1555–1566, 2020.

[16] Z. Zhang, R. He, and K. Yang, "A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm," *Advances in Manufacturing*, vol. 10, no. 1, pp. 114–130, 2022.

[17] L. Yiyang, J. Xi, B. Hongfei, W. Zhining, and S. Liangliang, "A general robot inverse kinematics solution method based on improved pso algorithm," *Ieee Access*, vol. 9, pp. 32 341–32 350, 2021.

[18] F. Gul, I. Mir, W. Rahiman, and T. U. Islam, "Novel implementation of multi-robot space exploration utilizing coordinated multi-robot exploration and frequency modified whale optimization algorithm," *IEEE access*, vol. 9, pp. 22 774–22 787, 2021.

[19] Y. Du and Y. Chen, "Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization," *Chinese Journal of Electronics*, vol. 31, no. 5, pp. 906–914, 2022.

[20] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing and Applications*, vol. 32, no. 20, pp. 16 245–16 277, 2020.

[21] Z. A. A. Alyasseri, N. S. Ali, M. A. Al-Betar, S. N. Makhadmeh, N. Jamil, M. A. Awadallah, M. Braik, and S. Mirjalili, "Recent advances of whale optimization algorithm, its versions and applications," *Handbook of whale optimization algorithm*, pp. 9–31, 2024.

[22] M. S. Uzer and O. Inan, "Application of improved hybrid whale optimization algorithm to optimization problems," *Neural Computing and Applications*, vol. 35, no. 17, pp. 12 433–12 451, 2023.

[23] A. T. Elgohr, M. Elazab, A. Z. Emam, M. G. Mostafa, M. Al-Razgan, H. M. Kasem, and M. S. Elhadidy, "Particle swarm optimization vs. whale optimization algorithm for robotic arm path planning: A controlled study on the kuka kr4 r600," *Results in Engineering*, p. 108299, 2025.

[24] Y. Tan, J. Ouyang, Z. Zhang, Y. Lao, and P. Wen, "Path planning for spot welding robots based on improved ant colony algorithm," *Robotica*, vol. 41, no. 3, pp. 926–938, 2023.

[25] L. Zhang, Y. Wang, X. Zhao, P. Zhao, and L. He, "Time-optimal trajectory planning of serial manipulator based on adaptive cuckoo search algorithm," *Journal of Mechanical Science and Technology*, vol. 35, no. 7, pp. 3171–3181, 2021.

[26] Y. Du and Y. Chen, "Time optimal trajectory planning algorithm for robotic manipulator based on locally chaotic particle swarm optimization," *Chinese Journal of Electronics*, vol. 31, no. 5, pp. 906–914, 2022.