S & M 4107

Six-degrees-of-freedom Redundant Task Trajectory Planning for Green Converter Station Robotic Arm Based on Multisensor Detection

Yang Li,[1] Pengwang Zhang,[1] Jinyun Yu,[1] Keying Zou,[2]
Xianguang Jia,[2*] Junwei Yang,[3] and Jing Bao[4]

[1]Ultra High Voltage Transmission Company Dali Bureau, China Southern Power Grid Co., Ltd., 671000, China
[2]Faculty of Transportation, Kunming University of Science & Technology, Kunming 650500, China
[3]Longshine Technology Group Co., Ltd., Wuxi 214000, China
[4]Faculty of Civil Aviation and Aeronautics, Kunming University of Science & Technology,
Kunming 650500, China

(Received October 7, 2024; accepted June 24, 2025)

Keywords: trajectory planning, polynomial interpolation, arithmetic optimization algorithm, Latin hypercube sampling

In recent years, robotic arms have become increasingly common in green converter stations. In this paper, we introduce a time-optimal trajectory planning method based on the arithmetic optimization algorithm (AOA). The method utilizes sensor data as constraints and generates trajectories using 3-5-3 polynomial interpolation. To enhance the performance of the AOA, we propose an improved version, the improved AOA, which incorporates the Latin hypercube sampling and Gaussian variation. Simulation results demonstrate that the proposed algorithm effectively plans trajectories for industrial robotic arms, offering improved efficiency and accuracy in motion planning.

1. Introduction

In recent years, the operation and maintenance of green converter stations have increasingly shifted towards automation and intelligent systems. A converter station is a key facility used for voltage conversion and DC transmission within power systems. During the daily operation and maintenance of these stations, operators must perform tasks such as equipment inspection, maintenance, and troubleshooting. Effective trajectory planning in these operations is crucial for improving efficiency, reducing operation time, and minimizing safety risks.

In green converter stations, the use of mobile robots equipped with robotic arms has become an essential component of automation.(1) These robots are designed to operate in environments that are hazardous to human workers, such as high-voltage areas and confined spaces. A typical application involves the autonomous navigation of the robot through the control rooms and other sections of the converter station, where it performs routine inspections of electrical equipment, such as circuit breakers, transformers, and switchgears. The mobile robot is equipped with a

*Corresponding author: e-mail: jxg@kust.edu.cn

https://doi.org/10.18494/SAM5629

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

robotic arm that can manipulate or adjust components, carry out preventive maintenance tasks such as cleaning or replacing parts, and troubleshoot issues by performing diagnostic tasks. The arm can be used to open and close circuit breakers, check the condition of electrical components, and even make minor repairs, such as tightening bolts or replacing fuses.

Trajectory planning is a constrained optimization problem that requires the simultaneous consideration of the robotic arm's kinematic constraints, collision avoidance, and singularity issues.(2,3) For the problem of time-optimal trajectory planning, scholars have proposed various intelligent optimization algorithms,(4,5) mathematical models,(6,7) and interdisciplinary solutions.(8) Among these, the arithmetic optimization algorithm (AOA) has attracted attention in recent years owing to its powerful global search capability and high convergence speed.(9) It has been widely applied to global optimization problems,(10) practical engineering,(11,12) parametric analysis,(13,14) and other fields. This algorithm has proven particularly valuable in the operation of converter stations.

In this paper, we propose an improved AOA (IAOA) combined with robotic arm trajectory optimization. By optimizing the operation path, the algorithm reduces operation time, lowers safety risks, and enhances operation quality. In conclusion, time-optimal trajectory planning is essential for improving the efficiency of industrial robots, particularly in converter station operations.

2. Mathematical Model

2.1 Establishment and analysis of Denavit–Hartenberg (D–H) models for industrial six-degrees-of-freedom (6-DOF) robotic arm

In this paper, we examine a 6-DOF robotic arm, capable of reaching any position and orientation in 3D space. It uses the D–H method to establish kinematic models for trajectory planning and motion control. The D–H convention is applied to define the relationship between the coordinate systems of each joint. The z-axis aligns with the joint axis and the x-axis points along the normal between neighboring joint axes. The transformation matrices are derived from four D–H parameters: link length, twist, angle, and offset. Figure 1 shows the D–H frames for the robotic arm, with {0} as the base and {6} as the end-effector. Table 1 lists the D–H parameters for each link. These parameters enable the calculation of the end-effector pose (forward kinematics) and the determination of joint motions for a given trajectory (inverse kinematics), which are essential for the robot's trajectory planning.

Unlike standard robotic arms used in industrial applications, which are typically designed for tasks such as assembly or material handling, the robotic arm used in green converter stations must meet specific operational and safety requirements because of the high-voltage environment in such stations. These specialized robotic arms have enhanced insulation to prevent electrical shocks, as well as features to reduce electromagnetic interference that can affect the arm's precision and operation in a high-voltage environment. Furthermore, these robotic arms are designed for high precision and reliability in performing critical tasks such as switching electrical components, inspecting transformers, and performing maintenance on circuit breakers, all of which require a higher degree of control and accuracy compared with more common industrial robotic arms.

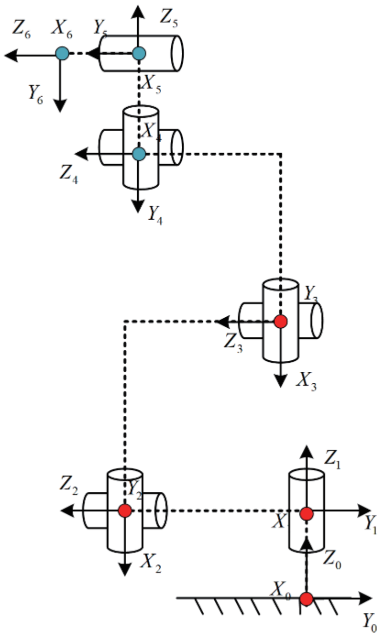


Fig. 1. (Color online) Robotic arm linkage coordinate system.

Table 1
D–H parameters of the industrial 6-DOF robot arm.

Joint i	Link twist α_i	Link length α_{i-1}	Joint offset d_i	Joint angle θ_i
1	0	0	d_1	θ_1
2	π/2	0	d_2	$\theta_2 - \pi/2$
3	0	$-\alpha_2$	$-d_3$	θ_3
4	0	$-\alpha_3$	d_4	$\theta_4 - \pi/2$
5	π/2	0	d_5	θ_5
6	−π/2	0	d_6	θ_6

2.2 Analysis of forward kinematics and inverse kinematics

The forward kinematics analysis of a robotic manipulator is aimed at determining the position and orientation of the end-effector frame relative to the base frame, given the values of the joint variables. For this 6-DOF robotic arm, the D–H convention is utilized to establish coordinated frames along the links and joints. On the basis of the D–H parameters, the relationship between two adjacent coordinate frames can be derived in the form of a homogeneous transform matrix. By multiplying these elementary transform matrices along the kinematic chain from the base to the end-effector, the forward kinematics can be formulated. This establishes the pose of the end-effector frame relative to the base frame. The complete forward kinematics mapping can be expressed as the multiplication of each transform matrix from the base to the end-effector. Therefore, given the joint angle values, the position and orientation of the end-effector can be solved. This establishes the relationship between the joint space and the Cartesian workspace, which is fundamental for the motion planning and control of the robotic arm. Its sub-transformation matrix between adjacent coordinate systems can be derived on the basis of the improved D–H parameters as follows:

$$
\begin{aligned}
{}^{i-1}_{\ \ i}T &= Rot_{X_{i-1}}(\alpha_{i-1})Trans_{X_{i-1}}(\alpha_{i-1})Rot_{Z_i}(\theta_i)Trans_{Z_i}(d_i) \\
&= \begin{bmatrix} c\theta_i & -s\theta_i & 0 & a_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\ s\theta_i c\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix},
\end{aligned}
\tag{1}
$$

$$
{}^0_6T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}
$$

The inverse kinematics for a 6-DOF robotic arm is about finding joint angles [θ_1, θ_2, ..., θ_6] to achieve a desired end-effector pose from Cartesian coordinates. It is more complex than forward kinematics, often having multiple solutions. With the D–H parameters, the problem can be formulated as nonlinear equations linking end-effector variables to joint angles. Various methods exist to solve these equations efficiently. Solving inverse kinematics is key for trajectory planning and control, creating a workspace-to-joint space mapping. The 6-DOF design's redundancy also aids in positioning the end-effector to navigate around obstacles.

$$
{}^1_6T = {}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T\,{}^5_6T = {}^0_1T^{-1}\,{}^0_6T = {}^0_1T^{-1} \times \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}
$$

2.3 Application of polynomials to trajectory planning

When planning a robotic arm's trajectory, choosing the right interpolation method is key. High-order polynomials can cause issues such as poor smoothness and high computation costs. Using them to generate joint angles might lead to unsmooth trajectories and instability during movement. Thus, lower-order polynomials such as the 3rd or 5th degree are often preferred for trajectory planning owing to their balance of continuity, smoothness, and computational efficiency. In this case, a 3-5-3 segmented polynomial approach is used: 3rd order at the endpoints for position, velocity, and acceleration continuity, and 5th order in the middle for trajectory optimization. This method offers good performance without excessive computational complexity.

The general formula for segmented polynomials is

$$
\begin{cases}
\theta_{i1}(t) = a_{i13}t^3 + a_{i12}t^2 + a_{i11}t + a_{i10}, \\
\theta_{i2}(t) = a_{i25}t^5 + a_{i24}t^4 + a_{i23}t^3 + a_{i22}t^2 + a_{i21}t + a_{i20}, \\
\theta_{i3}(t) = a_{i33}t^3 + a_{i32}t^2 + a_{i31}t + a_{i30}.
\end{cases}
\tag{4}
$$

In the equation, $\theta_{i1}(t)$ is the 3rd interpolation polynomial of the 1st segment of the *i*-th joint, $\theta_{i2}(t)$ is the 5th interpolation polynomial of the 2nd segment of the *i*-th joint, and $\theta_{i3}(t)$ is the 3rd interpolation polynomial of the 3rd segment of the *i*-th joint. The 1st, 2nd, and 3rd segments correspond to the periods of $t_0 \sim t_1$, $t_1 \sim t_2$, and $t_2 \sim t_3$, respectively. The coefficients a_{i1j}, a_{i2j}, and a_{i3j} are the first coefficients of the polynomials of the 1st, 2nd, and 3rd segments of the trajectory of the *i*-th joint, respectively. Among coefficients, $i = 1,2,3, ..., n$ denotes the *i*-th joint.

The equations of the starting and ending trajectory functions of the 3rd polynomial in the segmented polynomial are as follows:

$$\begin{cases} \theta(0) = a_{i3}t_0^3 + a_{i2}t_0^2 + a_{i1}t_0 + a_{i0}, \\ \dot{\theta}(0) = 3a_{i3}t_0^2 + 2a_{i2}t_0 + a_{i1}, \\ \ddot{\theta}(0) = 6a_{i3}t_0 + 2a_{i2}, \\ \theta_e = a_{i5}t_e^5 + a_{i4}t_e^4 + a_{i3}t_e^3 + a_{i2}t_e^2 + a_{i1}t_e + a_{i0}, \\ \dot{\theta}_e = 3a_{i3}t_e^2 + 2a_{i2}t_e + a_{i1}, \\ \ddot{\theta}_e = 6a_{i3}t_e + 2a_{i2}. \end{cases} \tag{5}$$

In the equation, θ_f are the joint angles at the start and end points. $\dot{\theta}(0)$ and $\dot{\theta}_f$ are the joint velocities at the start and end points, respectively. $\ddot{\theta}_f$ are the joint accelerations at the start and end points, and t_0 and t_e are the start and end times, respectively.

Considering the equation of the trajectory function of Eq. (2) and the constraints of displacement, angle, angular velocity, and angular acceleration of the robot arm, the equation of the trajectory function of the starting and ending points of the 5th polynomial is solved as follows.

$$\begin{cases} \theta(0) = a_{i5}t_0^5 + a_{i4}t_0^4 + a_{i3}t_0^3 + a_{i2}t_0^2 + a_{i1}t_0 + a_{i0} \\ \dot{\theta}(0) = 5a_{i5}t_0^4 + 4a_{i4}t_0^3 + 3a_{i3}t_0^2 + 2a_{i2}t_0 + a_{i1} \\ \ddot{\theta}(0) = 20a_{i5}t_0^3 + 12a_{i4}t_0^2 + 6a_{i3}t_0 + 2a_{i2} \\ \theta_e = a_{i5}t_e^5 + a_{i4}t_e^4 + a_{i3}t_e^3 + a_{i2}t_e^2 + a_{i1}t_e + a_{i0} \\ \dot{\theta}_e = 5a_{i5}t_e^2 + 4a_{i4}t_e^3 + 3a_{i3}t_e^2 + 2a_{i2}t_e + a_{i1} \\ \ddot{\theta}_e = 20a_{i5}t_e^3 + 12a_{i4}t_e^2 + 6a_{i3}t_e + 2a_{i2} \end{cases} \tag{6}$$

The coefficients of the 3rd- and 5th-degree polynomials are solved as

$$\begin{cases} a_{i0} = \theta_0, \\ a_{i1} = \dot{\theta}_0, \\ a_{i2} = \dfrac{3}{t_f^2}(\theta_f - \theta_0) - \dfrac{2}{t_f^2}\dot{\theta}_0 - \dfrac{1}{t_f}\dot{\theta}_f, \\ a_{i3} = -\dfrac{2}{t_f^3}(\theta_f - \theta_0) + \dfrac{1}{t_f^2}(\dot{\theta}_f + \dot{\theta}_0), \end{cases} \tag{7}$$

$$\begin{cases} a_{i0} = \theta_0, \\ a_{i1} = \dot{\theta}_0, \\ a_{i2} = \dfrac{\ddot{\theta}_0}{2}, \\ a_{i3} = \dfrac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3}, \\ a_{i4} = \dfrac{30\theta_f - 30\theta_0 - (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4}, \\ a_{i5} = \dfrac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5}. \end{cases} \tag{8}$$

The optimization objective function and constraints for the *i*-th joint are

$$f(t) = \min \sum\nolimits_{i=1}^{3} t_i, \tag{9}$$

$$\max\left\{\left|\dot{\theta}_i\right|\right\} \le \dot{\theta}_{imax}. \tag{10}$$

In the equation, $\dot{\theta}_i$ is the velocity corresponding to the polynomial interpolation of each segment in the *i*-th joint, and $\dot{\theta}_{imax}$ is the maximum velocity allowed for the *i*-th joint.

3. Algorithm Description

The AOA is a novel population intelligence optimization algorithm based on four regular mixed operations proposed by Abualigah and Diabat in 2023.(9) The algorithm performs global exploration through arithmetic multiplication and division operations and local exploitation using addition and subtraction operations.

3.1 AOA

In AOA, the initial candidate solution set *X* is randomly generated; we obtained the initial population from the following equation.

$$X_{i,j} = rand \times (ub - lb) + lb \tag{11}$$

The upper and lower population individual limits are set to *ub* and *lb*, respectively.

3.2 Math optimizer accelerated (MOA)

MOA is an iteration parameter in the algorithm whose value changes continuously with iterations and is calculated as shown in Eq. (12). AOA selects the search phase using the value of *MOA*.

$$MOA = min + t \times \left(\frac{max - min}{T} \right) \tag{12}$$

Here, t denotes the current number of iterations and T denotes the maximum number of iterations; *max* and *min* denote the upper and lower bounds of *MOA*, which are usually taken as 1 and 0.2, respectively.

3.3 Global exploration phase

When the random number $r1 > MOA$ ($r1 \in [0,1]$), the algorithm enters the global search phase. AOA implements the global search through multiplication (M) and division (D), and the process is shown in Eq. (13).

$$X_{i,j}(t+1) = \begin{cases} X_{pbest} - MOP \times ((ub - lb) \times \mu + lb), & r3 > 0.5 \\ X_{pbest} + MOP \times ((ub - lb) \times \mu + lb). & r3 \leq 0.5 \end{cases} \tag{13}$$

Here, $r2$ is a random number between 0 and 1; X_{pbest} is the current global optimal individual position; μ is the control parameter, generally taken to be $\mu = 0.499$; ε is defined as a very small integer; and *MOP* is the math optimizer probability, which is calculated as

$$MOP = 1 - \left(\frac{t}{T} \right)^{\alpha}, \tag{14}$$

where α is the iterative sensitivity factor, which affects the development accuracy of the exploration phase and is generally taken to be $\alpha = 5$.

3.4 Partial development phase

The algorithm enters the local development phase when the random number $r1 < MOA$. AOA achieves local development through addition (A) and subtraction (S); the process is shown in Eq. (15).

$$X_{i,j}(t+1) = \begin{cases} X_{pbest} - MOP \times ((ub - lb) \times \mu + lb) & r3 > 0.5 \\ X_{pbest} + MOP \times ((ub - lb) \times \mu + lb) & r3 \leq 0.5 \end{cases} \tag{15}$$

Here, $r3$ is a random number between [0,1].

3.5 Latin hypercube sampling

The traditional AOA generates the initial population randomly, which makes it difficult to ensure the uniformity of the initial individual distribution and easily affects the convergence speed and solution accuracy of the algorithm. Therefore, in this paper, we introduce the Latin hypercube sampling to generate the initial candidate solutions of the algorithm, by which the initial population can be effectively ensured to be uniformly filled and non-overlapping in the solution space. The specific steps of the population initialization strategy based on the Latin hypercube are as follows.

1) Set the algorithm population size N, the individual dimension D, and the upper and lower bounds *ub* and *lb*, respectively.
2) Divide the solution space [*ub*, *lb*] of the algorithm into N equally spaced subspaces.
3) Randomly draw a point of dimension D in each subspace as an individual in the initial population.
4) Combine all the extracted points to form the initial population of AOA for subsequent iterations of the optimization search.

3.6 Nonlinear factor

AOA balances the algorithm's global optimization-seeking ability and local exploitation ability using the *MOA*. In the traditional AOA, *MOA* increases linearly from 0.2 to 1 with the number of iterations t. However, the optimization of the algorithm is nonlinear; thus, in this paper, we introduce a nonlinear factor to improve the calculation formula of *MOA*, as shown in Eq. (16), to make it more suitable for the actual optimization process and further improve the solving ability of the algorithm.

$$MOA = -(max - min) \times \cos(\pi / 2 \times t / T) + 0.9 \tag{16}$$

max and *min* are still taken as 1 and 0.2, respectively. The comparison graph of *MOA* before and after the improvement is shown in Fig. 2, which shows that the improved *MOA* grows slowly in the early iterations to ensure that the algorithm can fully explore the optimal solution globally, while in the late iterations, the *MOA* grows rapidly to a larger value, which increases the probability of the local exploitation of the algorithm and improves the optimization-seeking accuracy of the algorithm.

3.7 Gaussian variation mechanism

To address the shortcomings of AOA, such as early maturity in late iterations and the tendency to fall into a local optimum, the Gaussian variation operator is introduced to improve its update strategy. When the algorithm enters the late iteration, *i.e.*, when $t > T/2$ is considered, Gaussian variation is applied to each individual before the end of each iteration, and the individuals before and after the variation are retained on a merit basis in accordance with the fitness. The process is represented as

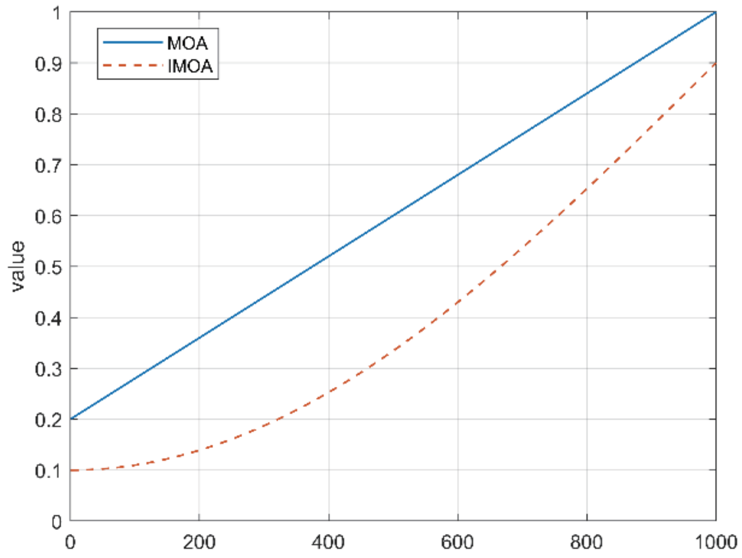


Fig. 2. (Color online) *MOA* before and after improvement.

$$X_{i,j}' = X_{i,j} \times [1 + Gauss(0,1)], \tag{17}$$

$$X_{i,j}(t+1) = \begin{cases} X_{i,j}', & fit(X_{i,j}') > fit(X_{i,j}(t+1)), \\ X_{i,j}(t+1), & \text{else.} \end{cases} \tag{18}$$

Equation (17) denotes an individual after Gaussian variation, *Gauss*(0,1) denotes a random variable satisfying a Gaussian distribution, and *fit*() denotes the fitness function.

In summary, the flow chart of the IAOA based on the Latin hypercube, nonlinear factor, and Gaussian variance is shown in Fig. 3.

4. Experimental Results and Analysis

In this study, the algorithm programming tool is MATLAB R2021a, the operating system is Windows 10, the computer memory is 16 GB, and the CPU is Intel i5-1135G7.

In this study, the proposed robotic arm system for green converter stations incorporates multiple sensors that enhance the precision and safety of operation during tasks such as equipment inspection, maintenance, and troubleshooting. These sensors not only enable the robotic arm to perform with high accuracy but also allow for real-time adjustments during trajectory planning, which is crucial in environments with high risks such as green converter stations.

To optimize trajectory planning and ensure precise motion control, angle sensors are installed at critical joints, such as the shoulder, elbow, and wrist, of the robotic arm. These sensors measure the rotational angles of each joint, allowing the control system to adjust the arm's

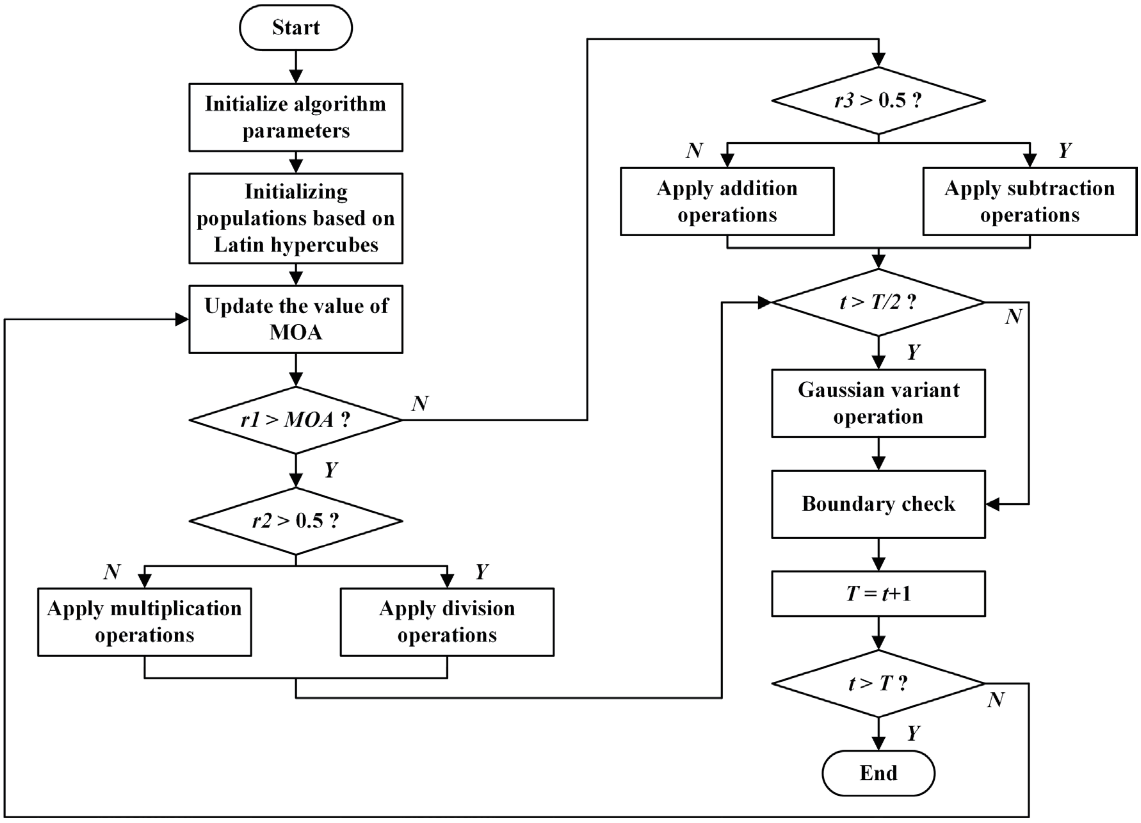

Start
Initialize algorithm parameters
Initializing populations based on Latin hypercubes
Update the value of MOA
r1 > MOA ?
N
Y
r2 > 0.5 ?
N
Y
Apply multiplication operations
Apply division operations
r3 > 0.5 ?
N
Y
Apply addition operations
Apply subtraction operations
t > T/2 ?
N
Y
Gaussian variant operation
Boundary check
T = t+1
t > T ?
N
Y
End

Fig. 3. Flow chart of the IAOA.

position and orientation precisely. The data from these sensors are essential for trajectory optimization, particularly in tasks that require high accuracy, such as inspecting electrical components or manipulating sensitive devices. By using the angle data, the robot's trajectory can be optimized in real time, ensuring the arm follows the most efficient and safe path to its target.

Another important sensor integrated into the robotic arm is the laser rangefinder, mounted at the end-effector. This sensor continuously measures the distance between the robotic arm and its target, which is critical for tasks such as cleaning or adjusting electrical components in a converter station. The distance data from the laser rangefinder allow the system to avoid potential collisions by adjusting the arm's position or stopping movement when an obstacle is detected. These measurements are fed directly into the trajectory optimization algorithm, which uses them to modify the planned path, ensuring that the arm operates with high accuracy and safety.

Additionally, force sensors are installed at key joints, such as the elbow and wrist, to monitor the forces applied during operations. These sensors provide feedback on the torque and force at the joints, which is particularly important for tasks that require delicate handling, such as tightening components or applying specific amounts of pressure. The force sensor data are used to adjust the trajectory and speed in real time, ensuring that the robot's movements do not exceed the physical limits of the arm or the objects being manipulated. This feedback helps optimize the trajectory planning by ensuring that forces are applied within safe and efficient limits.

The integration of these sensors plays a critical role in the optimization of the robotic arm's trajectory. By continuously providing real-time data about the arm's position, distance to the target, and applied forces, the sensors enable the trajectory optimization algorithm to adjust the planned path dynamically. This approach not only enhances the efficiency of the robotic arm but also reduces the risk of errors and ensures precise and safe operation in the high-risk environment of a green converter station.

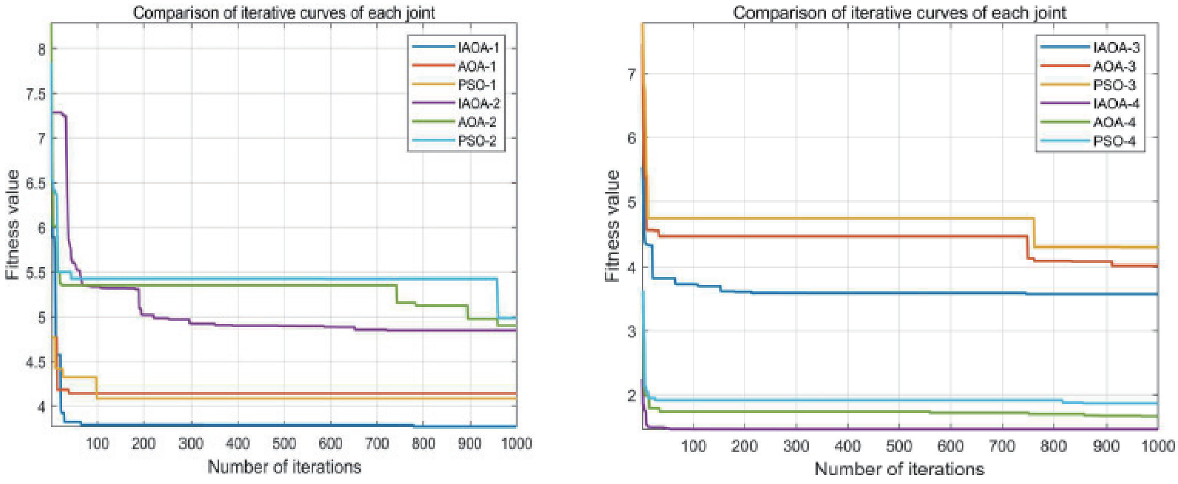
4.1 Parameter setting

In IAOA, the maximum value of *MOA* is *MOP_Max* = 1 and the minimum value is *MOP_Min* = 0.2. In particle swarm optimization (PSO), the particle learning rate is $c1 = c2 = 1.5$, the maximum particle velocity is $v_{max} = 1$, the minimum velocity is $v_{min} = -1$, the population size is $N = 100$, and the maximum number of iterations is $T = 1000$. The joint v_{max} is [127, 127, 104, 177, 155, 187] and the joint a_{max} is [127, 127, 104, 177, 155, 187].

4.2 Simulation results

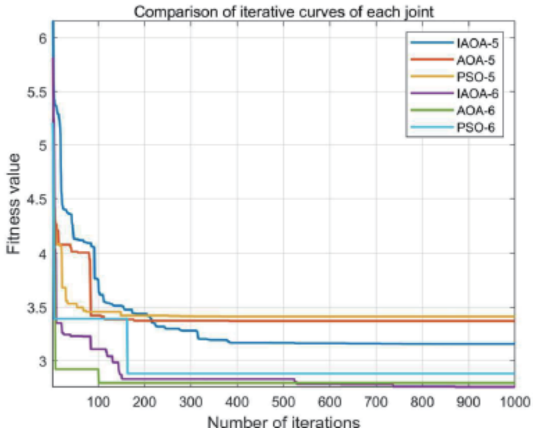
The IAOA was compared with the popular particle swarm algorithm, and the adaptive convergence plots for the six joints are shown in Fig. 4. The position, velocity, and acceleration curves of the six joints are shown in Fig. 5.

The objective function values for IAOA and PSO runs at each joint are shown in Table 2. The average objective function values for IAOA runs at each joint are better than those for AOA and PSO: 7.35% and 4.81%, respectively. These percentage improvements (7.35% over AOA and



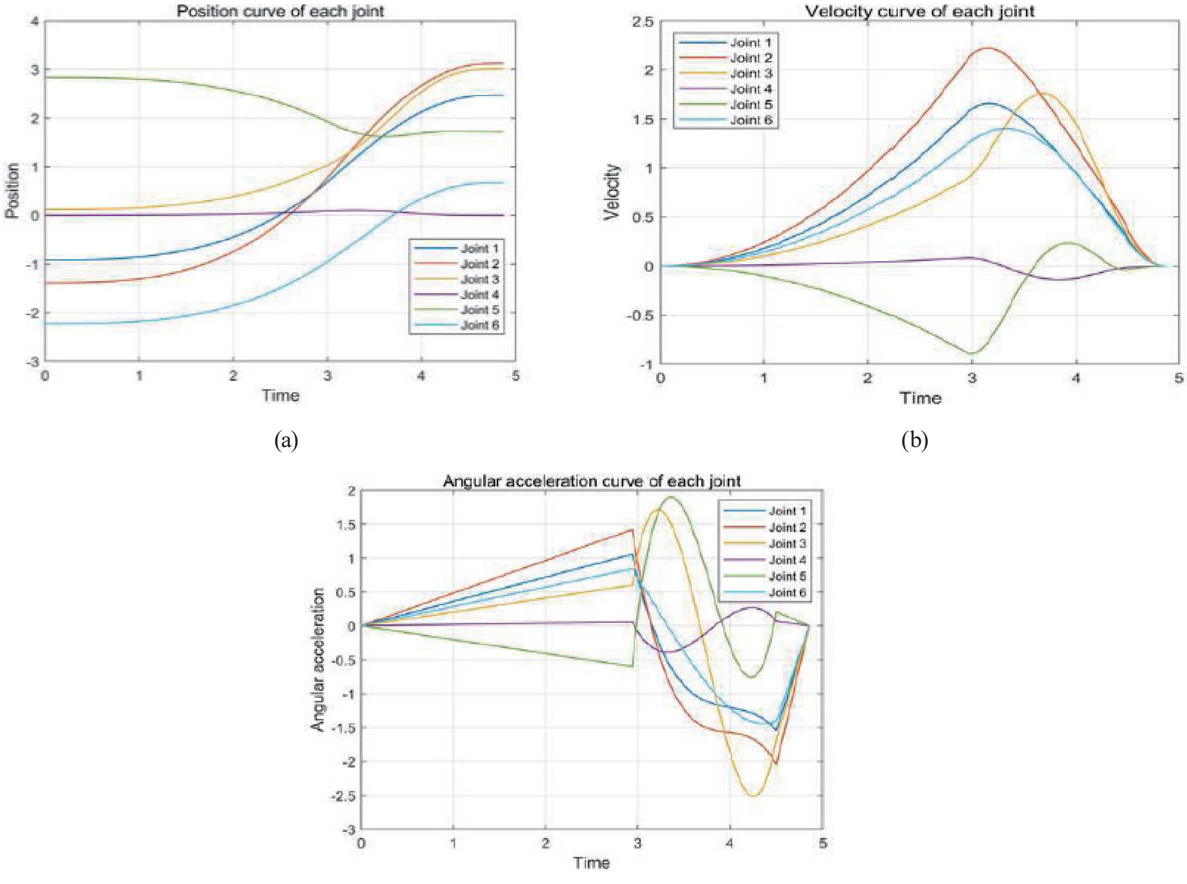
(a) (b)

Fig. 4. (Color online) Adaptive convergence diagram for (a) joints 1 and 2, (b) 3 and 4, and (c) 5 and 6.



(c)

Fig. 4. (Color online) (Continued) Adaptive convergence diagram for (a) joints 1 and 2, (b) 3 and 4, and (c) 5 and 6.



(a)

(b)

(c)

Fig. 5. (Color online) (a) Position, (b) velocity, and (c) acceleration curves of the six joints.

Table 2
Comparison of the objective function values run at each joint.

Objective function value	IAOA	AOA	PSO
Joint 1	3.72947	4.14534	4.08042
Joint 2	4.84050	4.90173	4.84067
Joint 3	3.54643	4.02099	3.81075
Joint 4	1.46194	1.66809	1.46455
Joint 5	3.13866	3.37097	3.41597
Joint 6	2.74323	2.79253	2.88849

4.81% over PSO) were derived by first calculating the average objective function value across all six joints for each algorithm. The IAOA's joint-wide average was then compared with those of AOA and PSO, with the percentage reductions computed on the basis of the relative difference between IAOA's lower average value and the higher averages of the two benchmark algorithms.

5. Conclusions

In this paper, we introduced a time-optimal trajectory planning model for a 6-DOF robotic arm, utilizing the 3-5-3 polynomial interpolation method. An IAOA was developed to address the nonlinear optimization challenges, incorporating methods such as Latin hypercube sampling and Gaussian variation to enhance global search capabilities. The results from the simulations demonstrate that the IAOA achieves faster convergence and superior solution quality compared with traditional algorithms, confirming its potential for optimizing the robotic arm's motion in real time.

Although the presented results are based on simulations, they offer valuable insights into how trajectory optimization can significantly improve the operation and efficiency of robotic arms used in green converter stations. By applying the optimized trajectory planning model, green converter stations can achieve precise and efficient robotic arm movements, reducing operation times and enhancing the safety of maintenance tasks. Specifically, the optimized paths derived from the simulation results ensure that the robotic arm performs tasks such as equipment inspection, maintenance, and troubleshooting with minimal risk of error or damage to sensitive components.

In practice, the improvements in trajectory planning and real-time optimization derived from this research will help enhance the automation of green converter stations. The results can be used to program robotic arms that can autonomously perform critical operations, thereby reducing human intervention in hazardous environments, improving operational efficiency, and ensuring higher safety standards. Furthermore, the simulation results serve as a foundation for future real-world implementation, providing a roadmap for the further development and optimization of robotic arm control systems in industrial applications.

(1) A time-optimal trajectory planning model for a 6-DOF robot arm was developed on the basis of the 3-5-3 polynomial interpolation method.

(2) An IAOA was proposed for solving highly nonlinear-constrained optimization problems.

(3) The IAOA improved the global search capability by Latin hypercube sampling, nonlinear modulation, and Gaussian variation.

(4) The model and algorithm were applied to the simulation of a 6-DOF robot arm, and the effectiveness of IAOA was verified.

In the trajectory planning of a robotic arm, in addition to avoiding collisions and reaching the target position, other constraints can be considered, such as avoiding singularities, maintaining specific forces and torques, and following specific speed and acceleration constraints. These constraints can be incorporated into the algorithm by modifying the optimization objective function or adding additional constraints.

Acknowledgments

This work was supported by the Digital Operation (Personalization) Construction Project of China Southern Power Grid Co., Ltd. under Grant no. 011000HK23110002.

References

1 D. Li, D. Yang, and J. S. Yang: Control Eng. China **29** (2022) 2365. https://doi.org/10.14107/j.cnki.kzgc.20210654

2 X. H. Qie, C. F. Kang, G. C. Zong, and S. J. Chen: Sensors **22** (2022) 4071. https://doi.org/10.3390/s22114071

3 C. X. Su, B. Li, W. Zhang, W. Tian, and W.-H. Liao: Reliab. Eng. Syst. Saf. **254** (2025) 110626. https://doi.org/10.1016/j.ress.2024.110626

4 O. Ekrem and B. Aksoy: Eng. Appl. Artif. Intell. **122** (2023) 106099. https://doi.org/10.1016/j.engappai.2023.106099

5 Y. X. Du and Y. H. Chen: Chinese J. Electronics **31** (2022) 406. https://doi.org/10.1049/CJ2021.00.373

6 H. Yervilla-Herrera, I. Becerra, R. Murrieta-Cid, L. E. Sucar, and E. F. Morales: J. Intell. Rob. Syst. **105** (2022) 82. https://doi.org/10.1007/s10846-022-01696-z

7 G. L. Wang and S. D. Zhang: Proc. Inst. Mech. Eng., Part C: J. Mech. Eng. Sci. **236** (2022) 10852. https://doi.org/10.1177/09544062221106632

8 B. Y. Song, J. Q. Li, X. Y. Liu, and G. L. Wang: ASME J. Comput. Inf. Sci. Eng. **24** (2024) 091003. https://doi.org/10.1115/1.4065814

9 L. Abualigah and A. Diabat: J. Intell. Manuf. **34** (2023) 1833. https://doi.org/10.1007/s10845-021-01877-x

10 T. Siron Anita Susan and N. Balasubramanian: Int. J. Eng. Technol. Innov. **13** (2023) 296. https://doi.org/10.46604/ijeti.2023.11552

11 P. Niranjan, R. K. Singh, and N. K. Choudhary: Int. J. Eng. Technol. Innov. **12** (2022) 260. https://dou.org/10.46604/ijeti.2022.8462

12 S. Q. Yan, W. D. Liu, P. Yang, F. X. Wu, D. L. Zhu, and G. Chen: J. Funct. Spaces **2023** (2023) 9841681. https://doi.org/10.1155/2023/9841681

13 A. Karolczuk and M. Kurek: Int. J. Fatigue **160** (2022) 106867. https://doi.org/10.1016/j.ijfatigue.2022.106867

14 Z. X. Feng, X. He, W. Cui, J. T. Zhao, M. Q. Zhang, and Y. Y. Yang: Comput. Integr. Manuf. Syst. **29** (2023) 604. https://doi.org/10.13196/j.cims.2023.02.021

About the Authors

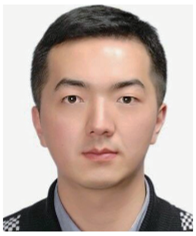
Yang Li is an engineer in the Dali Bureau of China Southern Power Grid EHV Transmission Company. He is mainly engaged in the operation and management of high-voltage DC transmission automation technology. (1053125623@qq.com)

Pengwang Zhang is a senior engineer in the Dali Bureau of China Southern Power Grid EHV Transmission Company. He is mainly engaged in the operation and management of high-voltage DC transmission automation technology. (zhangpengwang@im.ehv.csg)

Jinyun Yu is a senior engineer in the Dali Bureau of China Southern Power Grid EHV Transmission Company. He is mainly engaged in the operation and maintenance of HVDC transmission automation technology. (13708798055@139.com)

Keying Zou is currently a student at Kunming University of Science and Technology in Yunnan Province. Her main research direction is artificial intelligence and intelligent systems. (3332659904@qq.com)

Xianguang Jia is currently an associate professor at the Faculty of Transportation Engineering at Kunming University of Science and Technology, Kunming, China. His main research directions include modeling of traffic control, big data mining and analysis, and artificial intelligence. (jxg@kust.edu.cn)



Junwei Yang is a senior engineer in the Longxin Science and Technology Group Co., Ltd. His main research directions include electricity metering, load management and demand response, energy big data center, and vehicle-pile network cluster interaction. (yangjunwei@longshine.com)

Jing Bao is an undergraduate student in the Faculty of Civil Aviation and Aeronautics, Kunming University of Science and Technology. His main research direction is artificial intelligence and intelligent systems. (baojing@stu.kust.edu.cn)