Advances in Computer, Signals and Systems (2024)
Clausius Scientific Press, Canada
DOI: 10.23977/acss.2024.080617
ISSN 2371-8838 Vol. 8 Num. 6

Trajectory Planning of Rotor Welding Manipulator Based on an Improved Particle Swarm Optimization Algorithm

An Zhu[1,a], Zhongmin Wang[1,b]

[1]*School of Mechanical Engineering, Tianjin University of Technology and Education, Tianjin, China*

[a]*tjpuwuwei_zhuan@163.com*, [b]*wzmin86@126.com*

Keywords: Rotor welding robot; D-H method; Particle swarm algorithm; Polynomial interpolation function; Trajectory planning

Abstract: The rotor is the main core component of the powder separator, and the processing quality of the rotor directly affects the working efficiency of the separator and the operation safety of the separator, for the problems of unprotected welding quality of the rotor of the separator and high labor intensity of manual labor, for the problem of time-optimal trajectory planning of the welding robot, the welding robotic arm as the object of the study, using the D-H method of modeling and forward and inverse kinematics analysis. An improved particle swarm algorithm is proposed to optimize the trajectory of a spin-welding robot arm due to the inefficiency of traditional robot trajectory planning and unstable operation. The method effectively combines the 3-5-3 polynomial interpolation function with the improved algorithm using time as the fitness function. By comparing the traditional particle swarm algorithm, it is shown that the improved algorithm can be better applied to the time-optimal trajectory planning of the welding robot arm.

1. Introduction

Trajectory planning is one of the most important problems in robotics, which is related to whether to meet the time limitations and requirements of enterprise production and the operational efficiency of the robotic arm, time trajectory optimization has been a hot research issue in the development history of contemporary manipulators [1]. There are various methods for temporal optimization of the trajectory of robotic arm [2], such as Gray Wolf algorithm [3], genetic algorithm [4], particle swarm algorithm [5], ant colony algorithm [6], etc. Tian Xinghua et al [7] implemented a double selection strategy for inertia weight factors and added a perturbation operator in local optimization, which enhanced the ability of particle swarm algorithms to make sudden jumps, and improved the optimization ability compared to the traditional particle swarm [8]. Huang Chao et al[9] proposed a particle swarm optimization (NPSO) algorithm based on nonlinear dynamic change of inertia weights for the time optimization problem of joint space trajectory planning of a robotic arm, taking into account the motion constraints of the robotic arm. Li Xiaowei et al [10] used a particle swarm algorithm to optimize the constructed 3-5-3 polynomial trajectory with velocity as a constraint to obtain the optimal time; the

The standard particle swarm algorithm has the advantages of simple rules and easy implementation, but it is easy to fall into the dilemma of local optimization. For this reason, this paper uses the 3-5-3 polynomial interpolation method to construct the trajectory of the mechanical arm, takes the time optimization as the criterion, sets the speed and acceleration as the constraints, and puts forward an improved particle swarm algorithm, which changes the learning factor through the nonlinear function, and dynamically adjusts the size of the learning factor in the iterative process, to improves the convergence speed, and optimizes the time allocation problem in the 3-5-3 combinatorial segmented polynomials, so as to realize the time-optimal trajectory planning of the robotic arm of the welding robot.

2. Kinematic analysis of robotic armsOrganization of the Text

2.1 Kinematic Modeling

The experimental robotic arm used in the paper is the ARCMateM-10iD/8L tandem robotic arm manufactured by FANUC , and in order to reduce the difficulty of the kinematic inverse solution process, it is modeled by using Denavit Hartenberg (D-H) coordinates, which only requires four kinematic parameters, namely, θ_i(joint angle), d_i(link offset), a_{i-1}(link length), α_{i-1}(link twist angle), to characterize the robotic arm joints. The specific D-H parameters are shown in Table 1.

Table 1: D-H parameters

i	a_{i-1} /mm	α_{i-1} /(°)	d_i/mm	θ_i /(°)
1	0	0	450	θ1
2	75	-90	0	θ2
3	840	0	0	θ3
4	1100	-90	1100	θ4
5	0	90	0	θ5
6	0	-90	75	θ6

2.2 Forward and Inverse Kinematics Solving

The positive kinematic equations of the robotic arm are established through the D-H parameters and the linkage coordinate system, and the transformation formula between adjacent linkage coordinate systems is

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

To obtain the relationship between the base and the end-effector, the transformation matrix of each linkage is multiplied to obtain the end-position matrix of the robot arm as

$$ {}_6^0\mathbf{T} = {}_1^0\mathbf{T}\,{}_2^1\mathbf{T}\,{}_3^2\mathbf{T}\,{}_4^3\mathbf{T}\,{}_5^4\mathbf{T}\,{}_6^5\mathbf{T} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2} $$

The inverse kinematics solution is to find out the rotation angle of each joint of the robotic arm and the working position of the robotic arm according to the known parameters. Joint equation (1), equation (2), in this paper, the intermediate calculation process is omitted, and only the results of $\theta_1 \sim \theta_6$ are listed.

$$ \begin{cases} \theta_1 = \text{Atan2}\left(\pm p_y, \pm p_x\right) \\ \theta_2 = \text{Atan2}\left(k_3, \pm\sqrt{k_1^2 + k_2^2 - k_3^2}\right) - \text{Atan2}(k_1, k_2) \\ \theta_3 = \text{Atan2}\left(H, \pm\sqrt{a_3^2 + d_4^2 - H^2}\right) - \text{Atan2}(a_3, d_4) \\ \theta_4 = \text{Atan2}\left(h_2, h_1\right) \\ \theta_5 = \text{Atan2}\left(p_1, p_2\right) \\ \theta_6 = \text{Atan2}\left(n_x m_{31} + n_y m_{32} + n_z m_{33}, n_x m_{11} + n_y m_{12} + n_z m_{13}\right) \\ a_3 c_3 + d_4 s_3 = H \\ k_1 - a_2 c_2 = a_3 c_{23} + d_4 s_{23} \\ k_2 - a_2 s_2 = a_3 s_{23} - d_4 c_{23} \\ h_1 = a_x c_1 c_{23} + a_y s_1 c_{23} + a_z s_{23} \\ p_1 = a_x\left(s_1 s_4 + c_1 c_4 c_{23}\right) + a_y\left(s_1 c_4 c_{23} - c_1 s_4\right) + a_z\left(c_4 s_{23}\right) \\ p_2 = a_x\left(c_1 s_{23}\right) + a_y\left(s_1 s_{23}\right) + a_z\left(-c_{23}\right) \\ m_{11} = s_1 s_4 c_5 - c_1 s_5 s_{23} + c_1 c_4 c_5 c_{23} \\ m_{12} = -c_1 s_4 c_5 - s_1 s_5 s_{23} + s_1 c_4 c_5 c_{23} \\ m_{13} m_{31} = s_1 c_4 - c_1 s_4 c_{23} = s_5 c_{23} + c_4 c_5 s_{23} \\ m_{32} = -c_1 c_4 - s_1 s_4 c_{23} \\ m_{33} = -S_4 S_{23} \end{cases} \tag{3} $$

3. Time-optimal robot arm trajectory planning

3.1 3-5-3 Polynomial Function Construction

In order to study the trajectory planning of the type welding robot over the path points under the joint space, a “3-5-3” hybrid polynomial interpolation algorithm is used for planning.

$$f(x)=\begin{cases} l_{j1}(t)=a_{j13}t^3+a_{j12}t^2+a_{j11}t+a_{j10} \\ l_{j2}(t)=a_{j25}t^5+a_{j24}t^4+a_{j23}t^3+a_{j22}t^2+a_{j21}t+a_{j20} \\ l_{j3}(t)=a_{j33}t^3+a_{j32}t^2+a_{j31}t+a_{j30} \end{cases} \tag{4}$$

The coefficients a in the polynomial can be derived according to the constraints. it is known that the start point X_{j0}, intermediate points X_{j1} and X_{j2}, and termination point X_{j3} of the joint j, the displacement, velocity, and acceleration are continuous between the intermediate points, and the velocities and accelerations at the start point and the termination point are both 0. Combining with the above constraints, the 14 unknown coefficients a_j of the 3-5-3 polynomial can be derived and the matrix A can be derived based on the constraints and the constraint boundaries can be found out the matrix A, and the constraints and constraint boundaries are only related to time t.

$$A=\begin{bmatrix} A_{11} & A_{12} & 0 \\ 0 & A_{22} & A_{23} \\ 0 & 0 & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \tag{5}$$

$$\theta=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & x_{j3} & 0 & 0 & x_{j0} & 0 & 0 & x_{j2} & x_{j1} \end{bmatrix}^T \tag{6}$$

$$a=A^{-1}\theta \tag{7}$$

3.2 Improved Particle Swarm Algorithm

KENNEDY[11] proposed an optimization algorithm, particle swarm optimization (PSO), based on the foraging behavior of birds, whose basic principle is to use all the individuals in the population to collaborate with each other to make the whole group find the optimal purpose in the process of movement, PSO algorithm adopts the velocity-position search model for iteration, and finally obtains the optimal solution of an algorithm, its expression is

$$V_{id}^{k+1}=\omega V_{id}^{k}+c_1r_1\left(P_{id}-X_{id}^{k}\right)+c_2r_2\left(P_{gd}-X_{id}^{k}\right) \tag{8}$$

$$X_{id}^{k+1}=X_{id}^{k}+V_{id}^{k+1} \tag{9}$$

The values of inertia weight ω and learning factor c in PSO algorithm are directly related to the convergence performance of the algorithm, the traditional particle swarm algorithm ω, c_1, c_2 take the value of a constant, and the particle swarm algorithm of Feng Bin et al [12] that uses a linearly decreasing change in the inertia weight ω can significantly reduce the time of the robot's joint space movement, and the optimization effect is more than 40%, but the learning factor of the traditional particle swarm algorithm, c_1, c_2 is still a fixed value, which still affects the algorithm's optimization speed and solution accuracy. 2 are still fixed values, which still affects the algorithm's optimization speed and solution accuracy. While in the late stage of searching iterations $c_1 < c_2$ can allow the particle to converge to the global optimum as soon as possible, thus improving the convergence speed [13]. In addition, although the linearly decreasing inertia weights make up for the shortcomings of the traditional particle swarm algorithm, they also have the defect that the initial solution only improves the algorithm's accuracy and convergence speed if it falls in the global optimal solution attachment.

In this paper, we adopt the inertia weights with trigonometric function variation, and construct the ω, c_1, c_2 value functions as follows.

$$\begin{cases} \omega = (\omega_{\max} - \omega_{\min}) \cdot \cos^2\left(\dfrac{t}{T_{\max}}\right) & t \leq 0.6T_{\max} \\ \omega = 0.2 + 0.1rand & t > 0.6T_{\max} \end{cases} \tag{10}$$

$$c_1 = 2\cos^2\left(\frac{\pi t}{3T_{max}}\right) \tag{11}$$

$$c_2 = 2\sin^2\left(\frac{\pi t}{1.5T_{max}}\right) \tag{12}$$

3.3 Trajectory Optimization Process

In order to ensure that the welding robotic arm improves its welding efficiency under the premise of satisfying the kinematic constraints, this paper takes the 3-segment interpolation time of the 3-5-3 hybrid polynomial as the optimization objective [14] with the following objective function.

$$f(t) = min(t_{i1} + t_{i2} + t_{i3}) \tag{13}$$

Where t_{i1}, t_{i2}, t_{i3} denote the running time of the joint in the three interpolated trajectories, respectively.

It must be ensured that the angular velocity and angular acceleration of each joint at any moment are less than the maximum value allowed by the robotic arm [15], so the following constraints exist:

$$max\{|v_{ij}|\} \leq v_{max} \tag{14}$$

$$max\{|a_{ij}|\} \leq a_{max} \tag{15}$$

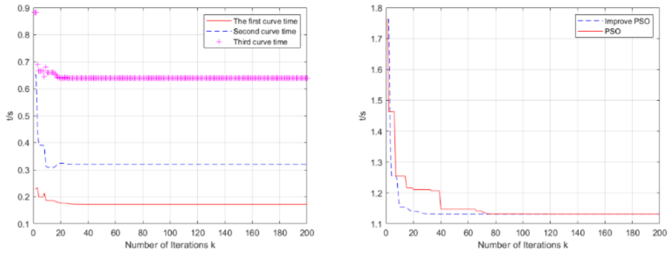
4. Simulation experiments and analysis of results

In order to verify the effectiveness and advantages and disadvantages of the improved particle swarm algorithm in the trajectory planning of the robotic arm, the paper takes the FANUC six-axis tandem robotic arm as the experimental object, and carries out the simulation experiments in MATLAB, and conducts the trajectory planning of the robotic arm in the Cartesian space, and determines four path nodes passed by the end of the robotic arm in the range of the welding work, and then converts it into the joints corresponding to the joints in the joint space through the inverse kinematics. The angle values are shown in Table 2.

Table 2: Joint space positions (rad)

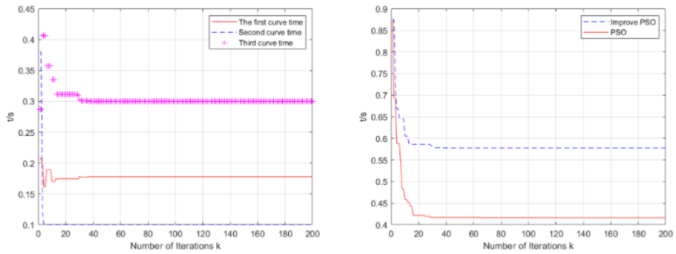
Joint	X_0	X_1	X_2	X_3
1	-23.496	-20.128	-7.361	-8.055
2	73.451	80.381	68.721	61.474
3	-62.949	-55.208	63.179	-68.520
4	77.747	78.321	-35.605	-29.593
5	41.879	35.783	27.940	32.293
6	-2.099	-12.636	53.129	46.917

In order to verify the effectiveness of the improved particle swarm algorithm, the standard particle swarm algorithm and the improved particle swarm algorithm are compared and simulated in MATLAB, taking the joints of the tandem robotic arm as an example, and the improved particle swarm algorithm is compared and simulated in MATLAB with the traditional particle swarm algorithm. ω_{max} is 0.9, ω_{min} is 0.4, the size of learning factor C_1, C_2 is between 0 and 2, the initial value of the three stage time is 2s, and the maximum iteration number T_{max} is 200. The interpolation time of each joint is optimized, and the iteration process is compared. Due to the limited space of the article, only the particle position evolution diagrams of the first 2 joints and the iterative diagrams of their convergence process are given here, as shown in Figures. 1 to 2.



(a) Time distribution of the curves by segment (b) Convergence process

Figure 1: Evolution of optimal particle positions for joint 1



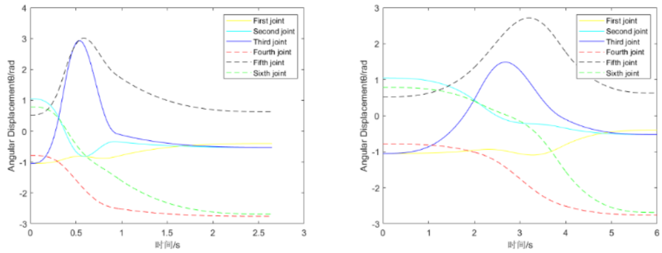
(a) Time distribution of the curves by segment (b) Convergence process

Figure 2: Evolution of optimal particle positions for joint 2

Comparing the three curves from Figure. 1 to Figure.2, it can be learned that after optimization using the improved particle swarm algorithm, the number of iterations required for convergence of the three joints is significantly lower than that of the basic PSO algorithm, and most of the joints are optimized to reach global convergence with the number of iterations within 30. Simulation results show that compared with the traditional particle swarm algorithm, the convergence speed and solution accuracy of the improved algorithm are improved.

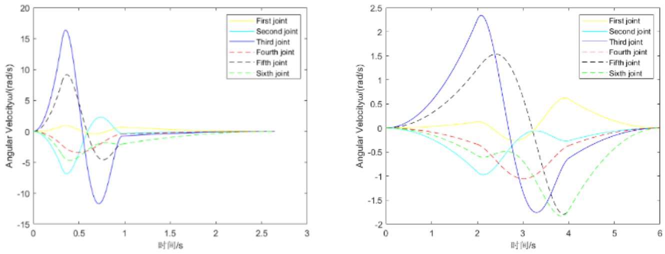
In order to ensure the smooth operation of each joint of the robot, the joints should be synchronized in the movement time, which ensures that the joints can reach the path point on time, and the maximum running time of each phase is 0.311, 0.660 and 1.676s, respectively, and the total running time of the three phases is 2.647s, which is a shortening of the running time by 56%, and the results show that the improved algorithm meets the kinematic constraints to greatly shorten the working time of the robot and improve the efficiency of the welding robot. The results show that the improved algorithm greatly reduces the robot working time and thus improves the efficiency of the welding robot under the condition of satisfying the kinematic constraints.

The 3-5-3 polynomial interpolation trajectory planning simulation of the robot arm is carried out using MATLAB, and the optimal time is substituted into the program to obtain the running trajectory of the end of the robot arm through the preset points, as well as the optimized curves of angular displacement, angular velocity, and angular acceleration of the joints with respect to the time, as shown in Figure. 3 to Figure. 5.



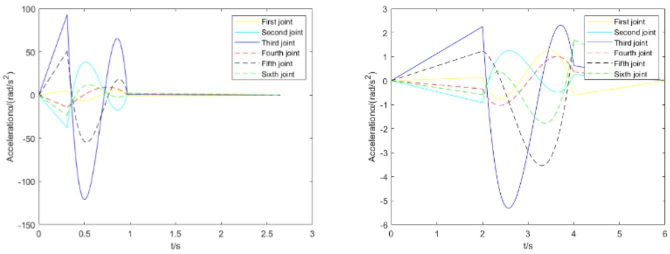
(a) Before optimization (b) Optimized

Figure 3: Joint position curve



(a) Before optimization (b) Optimized

Figure 4: Joint velocity profile



(a) Before optimization (b) Optimized

Figure 5: Joint acceleration curves

From Figure.3 to Figure.5, it can be found that the angular displacement, angular velocity and angular acceleration of each joint of the robotic arm have smooth curves without sudden changes, and the whole process of change is kept within an ideal continuity interval, which effectively proves that the trajectory planning algorithm ensures the normal and smooth operation of the robotic arm and shortens the running time of the end of the robotic arm, and further verifies the reliability of its time-optimal trajectory planning for the robotic arm. It further verifies the reliability of its time-optimal trajectory planning for the robotic arm. The results show that the improved trajectory planning method can ensure the completion of the target movement on the basis of the end trajectory of shorter length, greatly reducing the movement time of the robotic arm and effectively improving the working efficiency of the robotic arm.

5. Summary

Aiming at the actual operational requirements of rotor welding robotic arm, this study proposes a 3-5-3 polynomial interpolation trajectory planning strategy combined with the improved particle swarm algorithm, which pursues the time optimization with the constraints of robotic arm kinematics limitations. Through MATLAB comparative analysis, the improved particle swarm algorithm significantly shortens the running time of each joint of the robot compared with other algorithms. The angular displacement, velocity and acceleration curves of the joint motion show that the robot is smooth, reduces vibration and shock, meets the operational requirements and improves work efficiency, which verifies the reliability of the improved algorithm. Subsequent research needs to incorporate energy consumption, vibration and shock constraints for comprehensive validation to provide a comprehensive theoretical basis for robot trajectory planning.

References

[1] YUAN Xuhua, LIU Yu, LIN Xihui. Spline curve fitting and intelligent planning for time-optimal trajectories of robotic arms[J]. Mechanical Design and Manufacturing,2022,(09):162-167.

[2] LI Li, SHANG Junyun, FENG Yanli, et al. A review of research on trajectory planning for articulated industrial robots[J]. Computer Engineering and Application,2018,54(05):36-50.

[3] WANG Yongqi,JIANG Xiaoxia. Robot path planning based on hybrid gray wolf algorithm[J]. Computer Engineering and Science,2020,42(07):1294-1301.

[4] LAMINI C, BENHLIMA S, ELBEKRI A. Genetic algorithm based approach for autonomous mobile robot path planning[J]. Procedia Computer Science, 2018,127: 180-189.

[5] ZHEN ZHANG, QUN FANG, JINFENG SONG, et al. Research on dynamic path planning algorithm for spacecraft cluster based on cooperative particle swarm algorithm[J]. Journal of Northwestern Polytechnical University, 2021, 39(06): 1222-1232.

[6] LUO Q, WANG H B, ZHENG Y, et al. Research on path planning of mobile robot based on improved ant colony algorithm[J]. Neural Computing and Applications,2020, 32(6): 1555-1566.

[7] X. H. Tian,J. H. Zhang,Y. Li. Adaptive annealing particle swarm algorithm based on chaotic mapping[J]. Complex Systems and Complexity Science,2020,17(01):45-54.

[8] Chen Bo,Liu Youyu. Research on time-optimal trajectory planning by two-population particle swarm algorithm[J]. Mechanical Science and Technology,2022,41(10):1530-1535.

[9] HUANG Chao, MAO Jian, MA Li, et al. Time-optimal robotic arm trajectory planning based on improved particle swarm algorithm[J]. Journal of Shanghai University of Engineering and Technology,2020,34(03):238-246.

[10] LI Xiaowei,HU Likun,WANG Kun. Time-optimal trajectory planning of six-degree-of-freedom robotic arm with PSO under velocity constraint[J]. Journal of Intelligent Systems,2015,10(03):393-398.

[11] KENNEDYJ, EBERHARTR. particle swarmoptimization [C] //Proceedings of ICNN'95-International Conference on NeuralNetworks, IEEE, Perth, WA. Australia, 1995.

[12] Feng B, Liu F, Zheng Yimo. Optimal motion trajectory planning in robot joint space based on particle swarm algorithm[J]. Combined Machine Tools and Automated Machining Technology,2018,(05):1-4.

[13] MA Bin,WU Zezhong. Solving supply chain network equilibrium problem based on improved particle swarm algorithm[J]. Operations Research and Management,2020,29(02):122-128.

[14] LONG Zhang,LI Xiantao,SHUAITO,et al. A review of the current status of industrial robot trajectory planning research[J]. Mechanical Science and Technology,2021,40(06):853-862.

[15] GUO Xinxin, BO Ruifeng, JIA Junchen, et al. Time-optimal trajectory planning of robotic arm based on improved firefly algorithm[J]. Mechanical Design and Research,2021,37(03):55-59.