Manipulator Trajectory Planning and Optimization Based on Improved Particle Swarm Optimization Algorithm

Honghua Zhao[1st, *]
School of Mechanical Engineering
University of Jinan
Ji Nan, China
me_zhaohh@ujn.edu.cn

Keyuan Zhang[2nd]
School of Mechanical Engineering
University of Jinan
Ji Nan, China
18736721478@163.com

Qianqian Tian[3rd]
School of Mechanical Engineering
University of Jinan
Ji Nan, China
tqquni@foxmail.com

Wenguang Ren[4th]
QINGDAO HAIER BIOMEDICAL CO., LTD.
Qing Dao, China
373806668@qq.com

Sai Zhang[5th]
Jinan Robot Phoenix Technology Co., Ltd.
Ji Nan, China
Sai.zhang@robotphoenix.com

Jin Yu[6th]
School of Mechanical Engineering
University of Jinan
Ji Nan, China
me_yuj@ujn.edu.cn

***Abstract*—In order to improve the motion performance and work efficiency of the grasping system,this paper takes the interpolation time of the manipulator trajectory as the target to optimise, and realizes the optimization analysis of the manipulator 3-5-3 mixed polynomial trajectory by improving the intelligent optimization algorithm. Targeting the defect that the standard particle swarm optimisation algorithm easily gets stuck in the local optimum when searching iteratively, the inertia weight function and learning factor in the algorithm are improved. Through simulation experiments, the trajectory running time and motion parameters optimized by the four algorithms are compared, and the improved particle swarm optimisation algorithm has the best optimization effect on the interpolation time of the manipulator 3-5-3 mixed polynomial trajectory.**

Keywords-Trajectory planninghy; brid polynomial; particle swarm optimization algorithm; simulation experiment

I. INTRODUCTION

Trajectory planning determines the manipulator's path to the target position, so the quality of trajectory planning directly affects the manipulator's work efficiency and accuracy. To reduce the time spent moving and the energy consumed in the manipulator's operation, Numerous national and international experts have carried out relevant research and made great progress. According to the different research objectives and methods, the trajectory optimization algorithm can be divided into three types: impact optimization [1], time optimization [2] and energy optimization. The method of using intelligent algorithms to optimize the trajectory of manipulators has been widely used, but the standard intelligent algorithms generally have problems such as iterative non-convergence and easy precocity in the optimization process. In this paper, the search capability and search efficiency of the standard particle swarm optimisation algorithm are improved, and the time-optimised planning of the trajectory of the manipulator is realised.

II. TRAJECTORY PLANNING

A. Planning Trajectories in Cartesian Space

Planning the pose of the end effector directly in Cartesian space is called trajectory planning, and the change function of the pose, end effector velocity and acceleration are examined. Position and attitude of manipulator end effector is known. After receiving the information about the target position and the position of end effector, the pose of the intermediate key position points can be solved by the trajectory interpolation algorithm, and then the inverse of the solution of each of the interpolation points is obtained. The relationship between the angle and time of each joint is calculated. Finally, the trajectory of the end-effector is obtained by controlling the rotation of the drive motor through the joint controller.

B. Joint Planning of the Space Trajectory

The joint planning of the space trajectory determines the motion state by the function of the angular displacement of each joint of the manipulator with time. After determining the interpolation points of the trajectory, the time function of each joint is constructed independently, and the polynomial is used to approximate the given path, thus, the trajectory of the manipulator from beginning to end is planned. The joint space trajectory planning has the advantages of small amount of calculation, fast planning speed and no singular position.

C. 3-5-3 Mixed Planning of Polynomial Trajectories

The 3-5-3 mixed planning of polynomial trajectories divides the trajectory into three different stages by setting four

979-8-3315-3508-7/25/$31.00 ©2025 IEEE

path points, that is, the first and last segments select the cubic polynomial for interpolation, and the middle section of the path selects the quintic polynomial for interpolation. Figure 1 shows the above interpolation algorithm curve. This method reduces the time required for path planning while ensuring the motion accuracy of the manipulator.

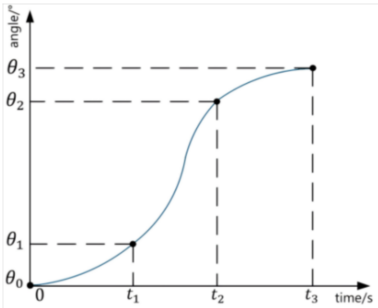


Figure 1. Interpolation Algorithm Curve Diagram

The four path points set by the 3-5-3 polynomial interpolation are the initial point, the two intermediate points and the termination point. The expressions of the t function in different stages are (1) ~ (3), which represent the angle and time relationship of the interpolation points in different stages, and represent the first polynomial coefficient in the segment function.

$0 \sim t_1$:

$$\theta_1(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \tag{1}$$

$t_1 \sim t_2$:

$$\theta_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 + a_{24}t^4 + a_{25}t^5 \tag{2}$$

$t_1 \sim t_2$:

$$\theta_3(t) = a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3 \tag{3}$$

III. OPTIMAL TRAJECTORY PLANNING

A. Particle Swarm Algorithm

Inspired by the foraging behaviour of birds in nature, the Particle Swarm Optimisation (PSO) algorithm [3]. Because the Particle Swarm Optimisation algorithm is simple and easy to understand, simple structure and convenient parameter adjustment, it has been the subject of rapid development in a short space of time and is now in use in many areas.

Particle Swarm Optimisation's optimisation process is similar to birds foraging in nature. Firstly, it is found that the birds of food are closest to the food, which is responsible for transmitting the location information of the food to other individuals in the birds, and then transporting the food back together through cooperation. Particle swarm algorithm uses this principle to regard the individuals in the bird swarm as particles with mass and volume of 0, and each of the particles represents a possible solution to the problem. As shown in Figure 2, the principle of updating the velocity and position of each particle in the particle swarm optimisation algorithm during the optimisation process.

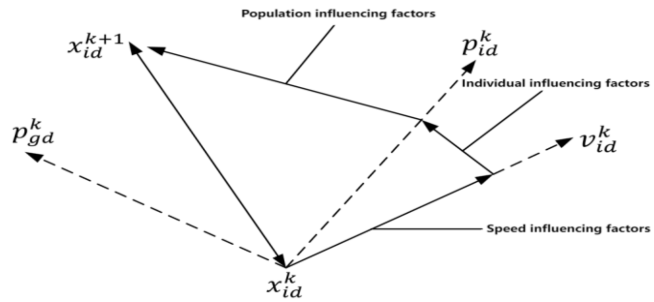


Figure 2. The update principle diagram of particle position and velocity

B. Improvement Analysis of Particle Swarm Algorithm

The particle swarm optimization algorithm has the characteristics of strong search ability and fast convergence speed in the process of finding the most solution. But when searching, it is easy to fall into the local optimum or the iteration does not converge. The results obtained in this way will be influenced by the control parameters such as the inertia weight and the learning factor. According to the manipulator trajectory optimization task, this paper proposes a corresponding improvement strategy.

1)Improvement of inertia weight function

A linearly adjusted inertia weight function is used in the standard particle swarm optimisation algorithm. The specific formula is shown in (4), where $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum inertia values, k represents the number of iterations of the algorithm at this time, and G represents the maximum number of optimization iterations.

$$\omega = \omega_{\max} + (\omega_{\min} - \omega_{\max}) \times \frac{k}{G} \tag{4}$$

As the number of iterations increases, the value of the linearly fitted inertia weighting function decreases, so the fine search can be realized under certain conditions, but this function cannot consider the change of particle fitness in the iterative process. At the same time, in the later stage of iterative optimization, the inertia weight decreases slowly, resulting in the weakening of local search ability. Therefore, the selection of the weight function needs to take into account the relationship between the number of iterations of the particle and the optimization target value. The initial iteration maintains a large value, which is conducive to improving the global search ability. When the particle searches to the optimal solution range, the value is reduced and the local search ability is improved. Therefore, we choose the inertial weight function in the form of exponential decline [4], the formula is as follows:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times e^{-(4k/G)^2} \tag{5}$$

2) Improvement of learning factor

The learning factors c_1 and c_2 represent the individual cognitive ability and social cognitive ability of the particle in the optimization process, respectively. The two are random numbers that obey uniform distribution on the interval, and the value range is generally [0,4]. When the value of c_1 is large, the particle relies too much on its own cognitive experience, which will cause the particle to stay at the start of the iteration in the local optimum region, reduce the algorithm's ability to search globally, making the true optimal solution impossible to find. When the value of c_2 is large, the particles are greatly affected by other particles in the population, which leads to the blind rapid convergence of the particles at a certain position, in this way, the position of the optimal solution is not taken into account and the local optimum is reached.

In order to ensure the balance between individual cognitive ability and social cognitive ability in the iterative optimization process of particles, the nonlinear dynamic function is used to control the value of the learning factor [5], so that the value of the learning factor is related to the current number of iterations of the particle. In the initial stage of iterative optimization, the individual cognitive ability of the particle is more important, and the value of the learning factor c_1 is increased to make the particle converge to the individual optimal solution faster. As iterations increase, the social cognitive ability of the particles is more important, and the value of the learning factor is c_2 increased to increase the probability of particles jumping out of the local optimal solution. Therefore, c_1 should be a decreasing function and c_2 should be an increasing function. The improved dynamic learning factor c_1 and c_2 expressions are as follows:

$$c_1 = c_{\max} - (c_{\max} - c_{\min}) \times \sin^2\left(\frac{\pi n}{2N}\right) \tag{6}$$

$$c_2 = c_{\max} + (c_{\max} - c_{\min}) \times \sin^2\left(\frac{\pi n}{2N}\right) \tag{7}$$

C. the Establishment and Steps of Objective Function

To ensure that the specimen gripping manipulator reduces moving time and improves working efficiency while meeting kinematic constraints [6]. The improved particle swarm optimization algorithm is used to optimize the three-segment interpolation time in the 3-5-3 mixed polynomial. The total motion time t_1, t_2 and t_3 of a joint of the manipulator in the three-segment trajectory form a three-dimensional particle swarm. The fitness function of any joint j is shown in Equation (8):

$$f(t) = \min \sum_{i=0}^{n} (t_{i1} + t_{i2} + t_{i3}) \tag{8}$$

In the formula, t_1, t_2 and t_3 represent the running time of the joint in the three interpolation trajectories respectively.

Considering the constraints of the kinematics and dynamics of the manipulator itself and the complex working environment, it is necessary to ensure that the angular velocity and angular acceleration of the six joints of the manipulator do not exceed the allowable range during the movement, so as to complete the corresponding grasping task with high quality. The specific constraints are as shown in (9) and (10):

$$\max\{|v_{ij}|\} \le v_{\max} \tag{9}$$

$$\max\{|a_{ij}|\} \le a_{\max} \tag{10}$$

Among them, v_{max} and a_{max} represent the maximum speed and maximum acceleration allowed by the manipulator joint, respectively. The subscript represents the i-th joint of the sample grabbing manipulator, and the subscript j represents the j-th interpolation polynomial.

The interpolation time of the six joints of the sample grasping manipulator is optimized respectively. Figure 3 shows the trajectory planning flowchart:

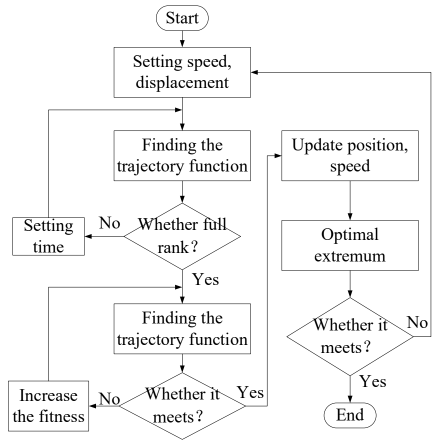


Figure 3. Time optimal trajectory planning flow chart of improved particle swarm optimization algorithm

IV. EXPERIMENTS AND ANALYSIS

A. Manipulator Grab System module

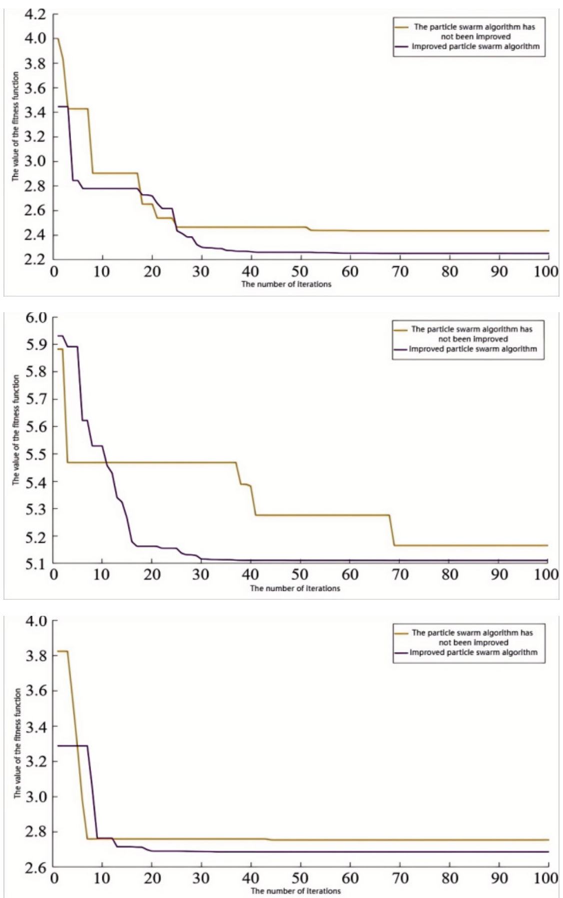
The main control system is analyzed from the perspective of system performance and economy through market investigation and product analysis. The FR5 manipulator and the Z-EFG-FS two-finger gripper launched by Huiling Technology Co., Ltd. are selected as the grasping module of the composite robot. The FR5 manipulator is composed of a control box, a manipulator body and an emergency stop control switch. The body has six joints, the maximum end load is 5kg, and the repetitive positioning accuracy can reach $\pm$0.03mm, which can be fixed to the mobile chassis of the robot through four 8.5mm holes set on the base. In addition, the manipulator can be controlled by MoveIt software module under Linux system, so it can cooperate with the mobile chassis. The main structure of FR5 is shown in Figure 4.

Figure 4. FR5 Manipulator and Hui ling Claw

B. Simulation Experiment

To verify the effect of the Particle Swarm Optimisation algorithm on optimising 3-5-3 mixed polynomial trajectories, this paper uses improved particle swarm optimization algorithm and ordinary particle swarm optimization algorithm to optimize the interpolation time of 3-5-3 mixed polynomial.

The values of the four path points in the trajectory are shown in Table 1. The parameters of the two algorithms have the following settings: the inertia weight ω of the improved particle swarm optimization algorithm is improved, and the learning factors c_1 and c_2 are dynamically adjusted according to Formulas (5), (6) and (7). Through the simulation test of the algorithm in References [7] and the improved particle swarm optimization algorithm in this paper, the ω_{max} and ω_{min} of the inertia weight ω are determined to be 0.9 and 0.4; the c_{max} and c_{min} of c_1 and c_2 are 2.3 and 0.5, respectively. The inertia weight w is set to 0.9 in the standard particle swarm optimisation algorithm, and the values of learning factors c_1 and c_2 are set to 2. The remaining parameters of the Particle Swarm Optimisation algorithm remain the same before and after tuning. The particle population m = 30, the maximum number of iterations N is 100, and the speed and acceleration of the starting point 1 and the ending point 4 are both 0. At the same time, the maximum acceleration of each joint is constrained to be 8m/s^2. Through MATLAB simulation experiments, the optimal particle fitness curve of each joint of the manipulator under different algorithm planning is obtained as shown in Figure 5.



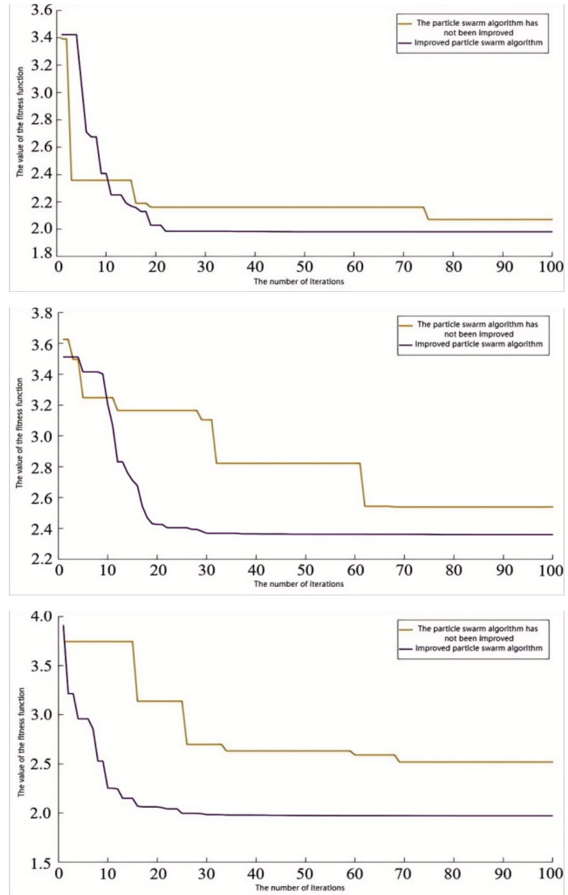


Figure 5. The fitness curve of each joint of manipulator

Compared to the particle swarm optimisation algorithm, it can be seen from the joint fitness curve 5, the improved particle swarm optimization algorithm greatly shortens the time of local convergence and global convergence in the trajectory optimization process of the six joints of the robot hand, and significantly reduces the number of iterations to global optimum.For time optimisation of the manipulator's gripping trajectory, the improved particle swarm optimisation algorithm is more appropriate.

The joint trajectory movement time of the 3-5-3 mixed polynomial optimized by the improved particle swarm optimization algorithm in three different stages is analyzed. The running trajectory is ' starting point 1, intermediate point 2, intermediate point 3 and termination point 4 '. In order to ensure that each joint of the manipulator can complete the trajectory action, it is necessary to take the maximum time of each joint movement in the three trajectories as the final movement time. The specific data are shown in Table 1.

TABLE 1. THE RUNNING TIME OF EACH STAGE

Joint	Total Time/s	Stage 1 Time/s	Stage 2 Time/s	Stage 3 Time/s
comparison value	9	3	3	3
1	2.8487	2.1383	0.1374	0.5730
2	5.2902	2.2898	1.8045	1.1959
3	2.9486	1.3896	0.7924	0.7666
4	2.1848	0.6373	0.5025	1.0450
5	2.3710	2.1205	0.0103	0.2402
6	2.2668	1.3621	0.7038	0.2009

By comparing the motion time of six joints in different trajectory stages in Table 1, the maximum values of the motion time after optimization of the three stages are 2.2898s, 1.8045s and 1.1959s, respectively. The total running time after optimization is 5.2902s, which is 3.7098s less than the running time of 9s before optimization. There is evidence that the improved particle swarm optimisation algorithm has a reduction in execution time.

The running time of each stage after optimization is brought into the expression corresponding to the 3-5-3 mixed polynomial. The angular displacement is obtained from the MATLAB program. angular velocity and angular acceleration of the six joints of the manipulator with time. The curve is shown in Figure 6.

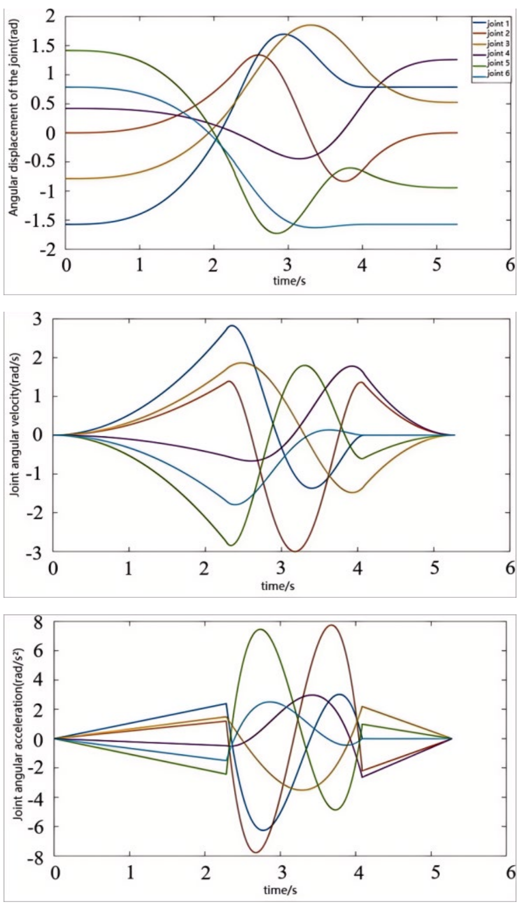


Figure 6. The motion curve of each joint optimized by improved particle swarm optimization algorithm

Figure 6 shows the angular displacement, angular velocity and angular acceleration curves of the six joints of the sample grasping manipulator are smooth and continuous, and there is no large mutation. Therefore, this optimization algorithm can be applied to the operation of the manipulator and can maintain the stability of the manipulator during the movement. The feasibility of the improved particle swarm optimization algorithm to optimize the trajectory is proved.

V. CONCLUSION

In this paper, through the comparative analysis of several common trajectory planning methods, the 3-5-3 mixed polynomial interpolation method is selected for manipulator trajectory planning, and the improved particle swarm optimization algorithm is used to optimize the three-stage trajectory of the 3-5-3 mixed polynomial trajectory planning method. Aimed at the defect that the standard particle swarm optimisation algorithm easily falls into a local optimum or iteration does not converge, this paper optimizes by improving the inertia weight function and learning factor, which improves the ability to search in the early stage of the iteration and the ability to search locally in the later stage of the iteration. Finally, the improved intelligent algorithm is subjected to a trajectory optimisation simulation experiment. The experimental results demonstrate the effect and feasibility of the improved particle swarm optimization algorithm to shorten the trajectory running time.

ACKNOWLEDGMENT

This study is funded by the key R & D project of Shandong Province (No.2022CXGC020901).

REFERENCES

[1] Krmer M, Muster F I, Rsmann C. An optimization-based approach for elasticity-aware trajectory plannin of link-elastic manipulators[J]. Mechatronics, 2021, 75(1):102523.

[2] Faroni A P N . A real-time trajectory planning method for enhanced path-tracking performance of serial manipulators[J]. Mechanism and Machine Theory: Dynamics of Machine Systems Gears and Power Trandmissions Robots and Manipulator Systems Computer-Aided Design Methods, 2021, 156(1).

[3] Wu P. Research on Integration Technology of Trajectory Tracking and Vibration Suppression for Delta High-Speed Parallel Robot[D]. Central North University, 2023.

[4] Jiang H H, Jin X X, Xing Y F. Five-degree-of-freedom Manipulator Trajectory Planning Based on PSOParticle Algorithm[J]. Machine Design and Research, 2020, 36(01): 107-110.

[5] Yang F W, Qian W Y. Overview of inertia weight improvement strategy in particle swarm optimization algorithm[J]. Journal of Bohai University (Natural Science Edition), 2019, 40(03): 274-288.

[6] Teng R M, Zhang W S, Wang X. Time optimal trajectory planning for demolition robot to demolish kiln lining[J]. Chinese Journal of Construction Machinery, 2023, 21(06): 562-567.

[7] Jiang H. Simulation Research on Trajectory Planning of a 6-DOF Robotic Arm [D]. Shen Yang: Shengyang Ligong University, 2023.