Artificial Life and Robotics (2022) 27:561–567
https://doi.org/10.1007/s10015-022-00779-2

ORIGINAL ARTICLE

Research on motion trajectory planning of the robotic arm of a robot

Xinghua Miao[1] · Huansen Fu[1] · Xiangqian Song[1]

Received: 16 March 2022 / Accepted: 1 July 2022 / Published online: 18 July 2022

Abstract
The emergence of robots has replaced repetitive manual labor, and good robotic arm route planning can effectively improve work efficiency. This paper briefly introduced the motion model and trajectory planning method of robotic arms. The motion trajectory of robot arms was optimized by the genetic algorithm-improved particle swarm optimization (PSO) algorithm, and simulation experiments were carried out. The results showed that the improved PSO algorithm converged faster and had the lowest fitness after stable convergence; the arm had continuous and smooth changes in angle, angular velocity and angular acceleration and consumed the shortest time while moving on the route planned by the improved particle swarm algorithm, and the improved PSO algorithm took the shortest time to compute the route.

Keywords Cross variation · Particle swarm optimization algorithm · Robotic arm · Trajectory planning

1 Introduction

With the development of technology, people's lives are becoming more and more convenient, especially in industrial production, where highly repetitive labor has been gradually replaced by industrial robots [1]. As an automation device, industrial robots combine technologies such as mechanics, control and programming and have the advantages of being programmable and working with high stability and efficiency. When the initial industrial robots used robotic arms to carry goods, their routes were set manually and the robotic arms worked in cycles according to the set routes [2], which were relatively simple but too rigid to cope with unexpected situations, and the manually planned routes might be not the best routes for robotic arm operation because of unstable joint changes during route changes. With the development of cybernetics and programming, the route planning of robotic arms has become more intelligent, and a suitable route connecting the starting and end points can be planned using intelligent algorithms [3]. Prianto et al. [4] proposed a soft actor-critic-based route planning algorithm and found through experiments that the performance of the algorithm was better than the current results. Wall et al. [5] proposed a new path planning algorithm to perform the high-speed computation to deal with fast-moving obstacles and found that the algorithm could calculate a safe path faster for the robotic arm. This paper briefly introduced the motion model and trajectory planning method of robotic arms. The motion trajectory of the robotic arm was optimized by the particle swarm algorithm improved by the genetic algorithm. The simulation experiments were conducted for the trajectory planning of the robot arm. The improved particle swarm algorithm was compared with the genetic and particle swarm algorithms.

2 Robotic arm motion model of robots

Figure 1 shows a schematic diagram of a robot arm with six-degree-of-freedom, and every node serves as an origin in the linkage coordinate system. There are six rotatable nodes in the robot arm [6]; then, there are seven origin points in the linkage coordinate system, where origin 0 is the part connecting the base to the first node. Table 1 shows the D–H parameters of the robotic arm in the linkage coordinate system [7]. α_{i-1} represents the angle required by Z_{i-1} to rotate to the Z_i axis around the X_{i-1} axis, which is used for determining the direction of the Z_i axis for node i, a_{i-1} represents the straight-line distance between the Z_{i-1} axis and the Z_i axis along the X_{i-1} axis,

✉ Xinghua Miao
xinglang3583846@163.com

[1] School of Shipping and Mechatronic Engineering, Taizhou University, No. 93, Jichuan East Road, Taizhou 225300, Jiangsu, China

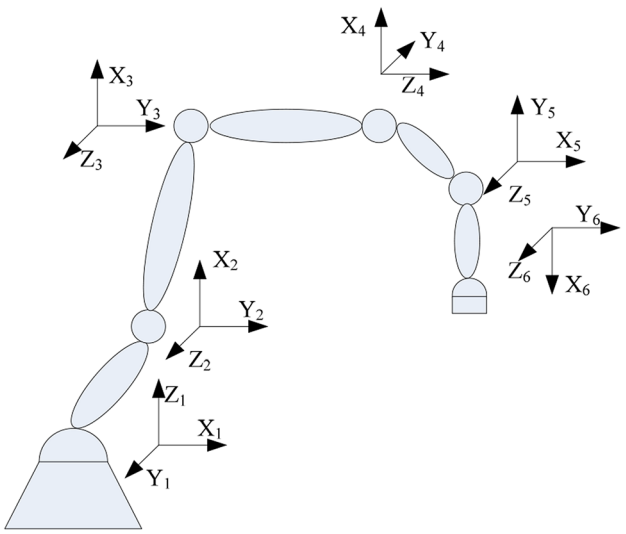

Fig. 1 Schematic diagram of a robotic arm with six-degree-of-freedom and its linkage coordinate system

d_i represents the straight-line distance between the X_{i-1} axis and the X_i axis along the X_{i-1} axis, and θ_i is the angle required by the X_{i-1} axis to rotate to the Z_{i-1} axis along the X_{i-1} axis, i.e., the joint angle of node i rotating around the Z_i axis. The conversion formula of the fixed coordinates and the linkage coordinates is:

$$\begin{cases} {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1}\sin\theta_i & \cos\alpha_{i-1}\cos\theta_i & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\alpha_{i-1}\sin\theta_i & \sin\alpha_{i-1}\cos\theta_i & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \\ {}^{0}_{6}T = {}^{0}_{1}T^{1}_{2}T^{2}_{3}T^{3}_{4}T^{4}_{5}T^{5}_{6}T \end{cases} \tag{1}$$

Among the four *D–H* parameters used in the above conversion expression matrix, only joint angle θ_i is a variable. The remaining three *D–H* parameters are determined by the structure of the robotic arm, and the three parameters are constants for the same robotic arm [8].

3 Trajectory motion control algorithm

3.1 Joint space planning method

Planning the robotic arm trajectory is to move the end of the arm from the initial position to the set position by changing the joint angle. To keep the robotic arm stable along the planned path, interpolation calculation for intermediate transition points between the starting and end points is needed to fit the smooth route curve.

The Cartesian spatial planning method directly solves the spatial trajectory of the end of the robotic arm. The obtained trajectory is intuitive, but the computation required to calculate the joint angles corresponding to the spatial point location of the trajectory in real-time is huge, especially in the robotic arm whose joint has a high degree of freedom. The joint spatial planning method performs interpolation calculations on the joint angles corresponding to the starting and ending points. Although the trajectory obtained from the planning is not intuitive, the computation is much smaller. The goal of this paper is to optimize the running time of the robotic arm trajectory movement, so the joint space planning method is finally chosen to design the motion trajectory.

In this paper, the joint space planning method is chosen to plan the robotic arm trajectory. The general process of the joint space planning method [9] is: ① calculate the joint angles when the end of the robotic arm is at the starting point, transition point and end point; ② make interpolation calculations on the joint angles of two adjacent points and fit to get the function relationship between joint angle and time in the two adjacent points. In the first step, the joint angles are obtained through the inverse operation of Eq. (1); in the second step, the trajectory is fitted by polynomial interpolation [10].

To improve the stability of the path and reduce the computational effort, avoiding the Runge's phenomenon should be avoided. This paper combines the cubic and quintic polynomial interpolation methods, and the corresponding formulas are:

Table 1 *D–H* parameters of the robotic arm

Joint number i	1	2	3	4	5	6
α_{i-1}	0	−90°	0	−90°	90°	−90°
a_{i-1}	0	a_1	a_2	a_3	0	0
d_i	0	0	0	d_4	0	0
θ_i	θ_1	θ_2	θ_3	θ_4	θ_5	θ_6

$$\begin{cases} \theta_1(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3 \\ \theta_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3 + a_{24}t^4 + a_{25}t^5 \\ \theta_3(t) = a_{30} + a_{31}t + a_{32}t^2 + a_{33}t^3 \\ a = \left[a_{13}\ a_{12}\ a_{11}\ a_{10}\ a_{25}\ a_{24}\ a_{23}\ a_{22}\ a_{21}\ a_{20}\ a_{33}\ a_{32}\ a_{31}\ a_{30} \right] = D^{-1}m, \\ m = \left[0\ 0\ 0\ 0\ 0\ 0\ M_3\ 0\ 0\ M_0\ 0\ 0\ M_2\ M_1 \right] \\ D = \begin{bmatrix} D_{11} & D_{12} & D_{13} \\ 0 & 0 & D_{23} \\ D_{31} & 0 & D_{33} \end{bmatrix} \end{cases} \tag{2}$$

$$D_{11} = \begin{bmatrix} t_1^3 & t_1^2 & t_1 & 1 & 0 \\ 3t_1^2 & 2t_1 & 1 & 0 & 0 \\ 6t_1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_2^5 \\ 0 & 0 & 0 & 0 & 5t_2^4 \\ 0 & 0 & 0 & 0 & 20t_2^3 \end{bmatrix} \quad D_{12} = \begin{bmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & -2 & 0 & 0 \\ t_2^4 & t_2^3 & t_2^2 & t_2 & 1 \\ 4t_2^3 & 3t_2^2 & 2t_2 & 1 & 0 \\ 12t_2^2 & 6t_2 & 2 & 0 & 0 \end{bmatrix} \quad D_{13} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & -2 & 0 & 0 \end{bmatrix},$$

$$D_{23} = \begin{bmatrix} 0 & t_3^3 & t_3^2 & t_3 & 1 \\ 0 & 3t_3^2 & 2t_3 & 1 & 0 \\ 0 & 6t_3 & 2 & 0 & 0 \end{bmatrix} \quad D_{31} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad D_{33} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3}$$

where $\theta_1(t)$, $\theta_2(t)$ and $\theta_3(t)$ are the joint angles in the first, second and third sections of route, t is the time, t_1 is the time required for the first section of route, t_2 is the time required for the second section of route, t_3 is the time required for the third section of route, a is the matrix of the unknown coefficients, D is the matrix of the equation set of the path constraints, and m is the four known insertion points of the three sections of route.

3.2 Particle swarm optimization-based trajectory optimization

In this paper, the joint space planning method divides the route into three sections, so there are four route points in a route. Using Eqs. (2) and (3), the corresponding planned route can be obtained as long as the four route points and the movement time are known. The route fitted by the polynomial interpolation method is smooth enough, so the optimization of the trajectory of the robotic arm in this paper mainly focuses on the running time, i.e., to make the robotic arm move from the starting point to the end point as fast as possible while maintaining stability.

There are various algorithms for optimizing motion trajectories in space, and this paper chooses the particle swarm optimization (PSO) algorithm [11]. The PSO algorithm imitates bird foraging in nature. In this algorithm, a population containing a plural number of particles is generated first. The position coordinates of every particle in the population in the search space is a candidate solution, i.e., a population is a set of candidate solutions. The fitness value of every particle in the population is calculated. Taking the best particle position as the target, the particle population moves in the search space. "The judgment of the current and historical optimal position of the particle—particle population movement" is repeated until the particle population converges at a certain position. The coordinates of this position is the optimal solution.

This paper uses the joint space planning method to plan the end movement trajectory of the robotic arm, which means that the movement of the end of the robotic arm is achieved by controlling the change of the joint angle. The starting and ending positions of the end of the robotic arm can obtain the angle of every joint angle through conversing Eq. (1). The movement of the end can be regarded as the change of the joint angle from the angle corresponding to the starting position to the angle corresponding to the end position. Since the end can have different paths from the starting point to the end, the changing process of the joint angle is also different.

To make the trajectory smoother and more stable, this paper chooses polynomial interpolation to fit the joint angle change curve with time and divides the process from

the starting point to the end point into three segments. The first and third segments are fitted with cubic polynomial interpolation, and the second segment is fitted with quintic polynomial interpolation. In this three-segment polynomial interpolation method, in addition to the starting point and the end point, two additional insertion points need to be inserted, and the corresponding joint angles can be obtained using Eq. (1).

Based on the starting point, the end point, and the joint angles corresponding to the two insertion points, polynomial interpolation can be performed in combination with Eqs. (3) and (4), with the final goal of obtaining the coefficient matrix a of every joint angle. In this paper, besides using the interpolation method to obtain a smooth moving trajectory, it is also necessary to make the moving time of the trajectory as small as possible so as to improve the efficiency of the robotic arm.

When using the PSO algorithm to optimize the trajectory of joint angle change, instead of gradually fitting coefficient a, the coordinates of the particles can be used as coefficient a and thus substituted into the polynomial, followed by the calculation of the consumed time. The time consumption is minimized during the population iteration. However, a robotic arm often has plural joint angles, and the trajectory polynomial of every joint angle requires a coefficient matrix a, which leads to a high dimensionality of particles within the PSO population and increases the search difficulty. The plural joint angle changes of the robotic arm are simultaneous, which means that the movement time of the robotic arm does not need to distinguish the joint angle; therefore, it searches by taking the movement time spent in the three segments of paths as the particle coordinates, and the time represented by the particles are substituted into Eqs. (3) and (4) to calculate the coefficient of every joint angle polynomial in the iteration process. The goal of the particle iteration process is still to minimize the total time, but to ensure the stability of the moving trajectory, after the polynomial function of the trajectory is obtained based on the consumed time, it is necessary to calculate the running speed of joint angles, and the particles exceeding the limit cannot be used as the optimal particle even if the total time is minimal.

Compared with the genetic algorithm, the PSO algorithm is simpler in principle and implementation, but in the process of searching for the optimal solution, the particles may fall into the locally optimal solution. To make the particle population jump out of the locally optimal solution, this paper introduces the concept of crossover and mutation from the genetic algorithm, so that the particles in the PSO population perform crossover and mutation operations after the iteration of position and velocity. The randomness of the two operations is used to avoid the particles falling into the locally optimal solution as much as possible.

Figure 2 shows the basic process of optimizing the joint angle trajectory of the robotic arm using an improved PSO algorithm, and the specific steps are as follows.

① Every particle in the population represents a planning scheme, and the coordinates of the particle in the search space are the scheme content. The time required for the first, second and third sections of the route is taken as the coordinate axes of the three-dimensional search space. The initialization of the population will randomly generate particles at different positions under the number of population sizes.

② Whether particles corresponding to the time required for the three sections of the route in the population can make matrix D singular is determined. Positions of particles that will make matrix D singular are adjusted, i.e., the time required for the three sections of route represented by the particles is adjusted.

③ The time required for the three sections of route represented by the particles are substituted into Eqs. (2) and (3) to calculate the coefficient of the three-segment polynomial function for every joint angle trajectory.

④ The maximum running speed of the joint angle in the path is calculated according to the three-segment polynomial function of every joint angle trajectory to deter-

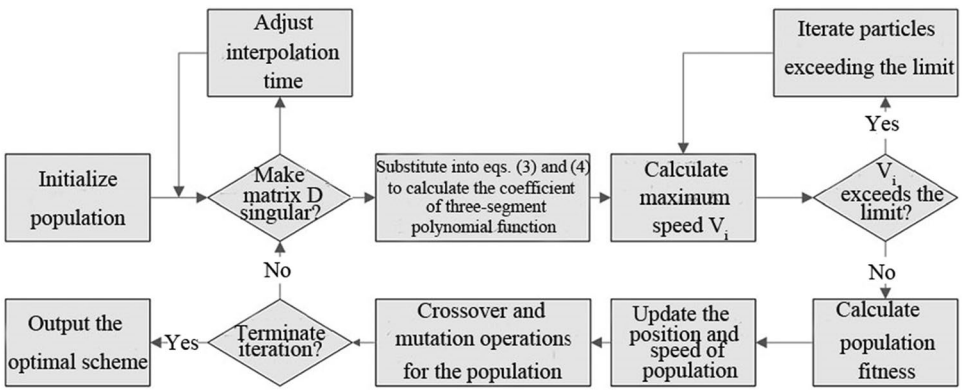

Fig. 2 Basic flow of joint space trajectory planning under an improved particle swarm optimization algorithm

mine whether the running speed exceeds the maximum running speed allowed by the joint angle. If not, the next step is performed; if it does, the particles exceeding the limit are processed by regular PSO iterations to make the joint angle speed decrease.

⑤ The fitness of the particles within the population is calculated, i.e., the total time required for the three sections of the route, and the particles are sorted in descending order of fitness.

⑥ The position and velocity of the particles within the population are updated [12] using the following equation:

$$\begin{cases} v_i(t+1) = \varpi v_i(t) + c_1 r_1 (P_i(t) - x_i(t)) + c_2 r_2 (G_g(t) - x_i(t)) \\ x_i(t+1) = x_i(t) + v_i(t+1) \end{cases}, \tag{4}$$

where $v_i(t+1)$ and $x_i(t+1)$ are the velocity and position of particle i after one iteration, $v_i(t)$ and $x_i(t)$ are the velocity and position of particle i before the iteration, ϖ is the inertia weight of the particle, c_1 and c_2 are learning factors, r_1 and r_2 are random numbers between 0 and 1, $P_i(t)$ is the individual historical optimal point of particle i, and $G_g(t)$ is the historical optimal point of particle swarm.

⑦ The three coordinate values of particles in the search space are regarded as three gene fragments. The single-point crossover [13] is used in this paper, which means exchanging the fragment of the same gene locus in two randomly selected particles according to the crossover probability. The mutation operation is also used, which means changing one gene fragment in a randomly selected particle according to the mutation probability.

⑧ If the termination condition is satisfied, then the optimal scheme in the population is output; if not, step ② is repeated until the termination condition is satisfied.

4 Simulation experiments

4.1 Experimental setup

The robotic arm used in the simulation experiment was a six-degree-of-freedom robotic arm, whose basic structure is shown in Fig. 1. The *D–H* parameters of the robotic arm are shown in Table 2. In Table 2, θ_i represents the rotation angle of joint angle i, whose value depends on the end position of the arm. The planned joint space trajectory was fitted by the three-segment polynomial interpolation method, so four interpolation points were needed [14], which were starting point M_0, transition points M_1 and M_2, and end point M_3. The joint angles corresponding to the four points are shown in Table 2.

To further demonstrate the performance of the improved PSO algorithm, this paper compared it with the traditional PSO algorithm and the genetic algorithm. The basic process of the traditional PSO algorithm was the process after removing the crossover and mutation steps in Fig. 2, and its relevant parameters were the same as those of the improved PSO algorithm except the crossover and mutation probabilities.

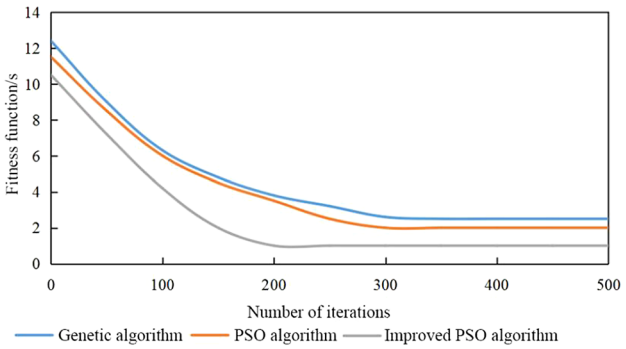

Fig. 3 Convergence curves of the three algorithms when optimizing the trajectory

4.2 Experimental results

Figure 3 shows the convergence curves when optimizing the joint angle space trajectory of the robotic arm using the genetic algorithm, the traditional PSO algorithm and the improved PSO algorithm, and the fitness function is the total time required for the three sections of the route. It was seen from Fig. 3 that after optimization by the three algorithms, the total movement time of the robotic arm was subsequently

Table 2 *D–H* parameters of the robotic arm

Node number i	1	2	3	4	5	6
α_{i-1}	0	−90°	0	−90°	90°	−90°
a_{i-1}	0	180 mm	640 mm	195 mm	0	0
d_i	0	0	0	740 mm	0	0
θ_i of M_0	0.365 rad	0.020 rad	0 rad	0.716 rad	0 rad	0 rad
θ_i of M_1	0.415 rad	0.050 rad	0.075 rad	0.082 rad	0.085 rad	0.065 rad
θ_i of M_2	0.581 rad	0.174 rad	0.235 rad	0.962 rad	0.175 rad	0.115 rad
θ_i of M_3	0.635 rad	0.248 rad	0.265 rad	1.040 rad	0.215 rad	0.195 rad

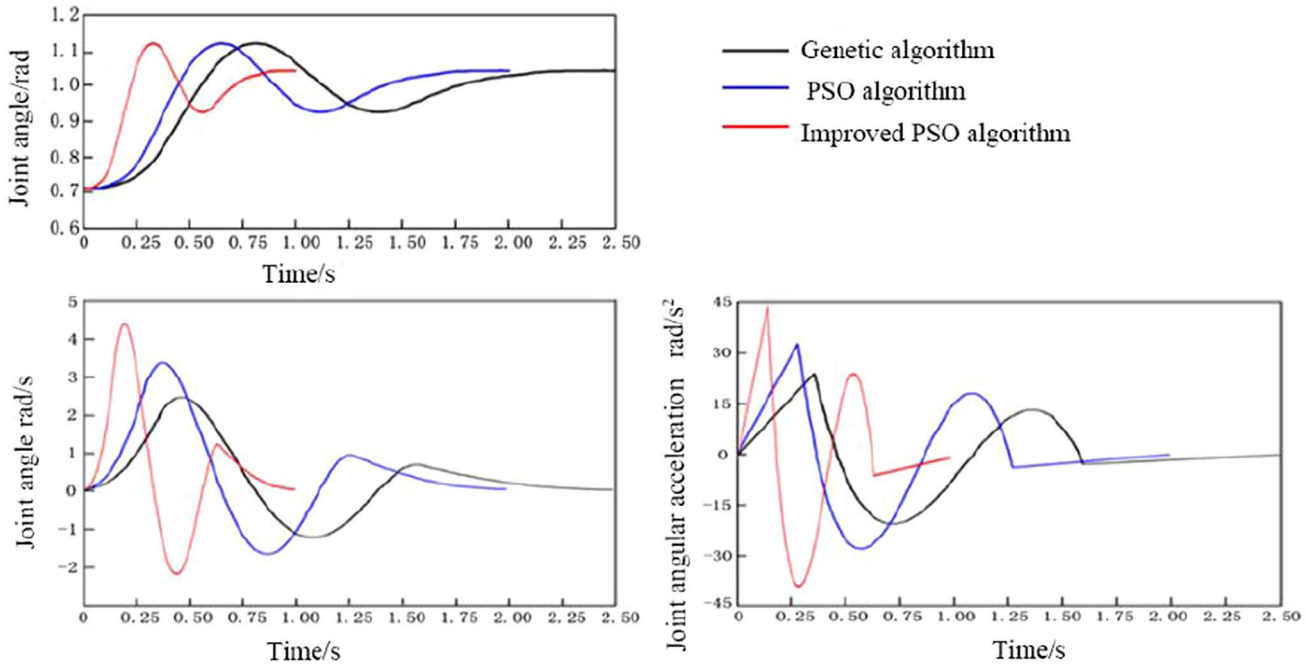

Fig. 4 Changes in angle, angular velocity and angular acceleration of joint 1 under three algorithms

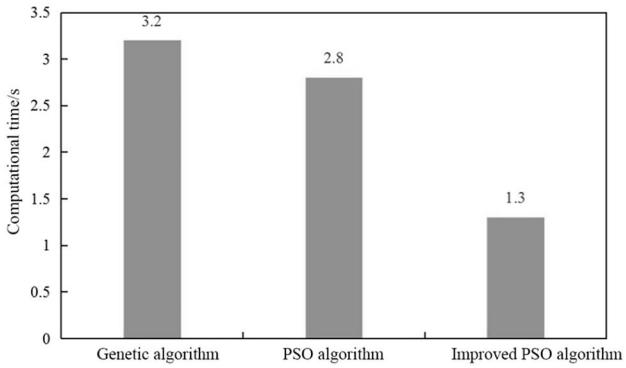

Fig. 5 Computational time of the three algorithms

reduced, and the improved PSO algorithm iterated the minimum number of times and had the lowest fitness before stability.

Due to space limitation, this paper only shows the changes of angle, angular velocity and angular acceleration of joint angle 1 within the planned time, as shown in Fig. 4. Figure 4 shows that the route obtained by the genetic algorithm took 2.50 s, the route obtained by the PSO algorithm took 2.00 s, and the route obtained by the improved PSO algorithm took 1.00 s. It was seen from Fig. 4 that the changes in the joint angle and angular velocity were continuous and smooth under the routes planned by the three algorithms, and the quintic polynomial interpolation method was used to plan the intermediate route, which avoided the problem of sudden changes in angular acceleration in the planning of the cubic polynomial interpolation method and improved the joint stability of the robotic arm.

The computation time of the three optimization algorithms is shown in Fig. 5. The genetic algorithm took 3.2 s, the PSO algorithm took 2.8 s, and the improved PSO algorithm took 1.3 s. It was seen from Fig. 5 that the genetic algorithm consumed the most time in computation, followed by the PSO algorithm and the improved PSO algorithm. The reason for the above result is as follows. Although the genetic algorithm used the fitness function as the guiding target when iterating on the population, individuals in the population were randomly adjusted according to the crossover and mutation probabilities, and the fitness function played more of a screening role. The PSO algorithm used the fitness function to select the optimal individuals when adjusting the individuals so that the other individuals were adjusted toward the optimal individuals, which had a more precise direction; therefore, it obtained the optimal solution faster than the genetic algorithm. As the PSO algorithm aimed at adjusting the particles to make them optimal, the optimal individual would affect the whole algorithm; thus, to avoid the optimal individual be a local optimum, the improved PSO algorithm used the crossover and mutation operations in the genetic algorithm to adjust the individuals, thus jumping out of the locally optimal solution, accelerating the convergence and improving the computational speed.

5 Conclusion

This paper briefly introduced the motion model of the robotic arm and the trajectory planning method. The trajectory of the robotic arm was optimized using the genetic algorithm-improved PSO algorithm; simulation experiments were conducted on the trajectory planning of the robotic arm; the improved algorithm was compared with the genetic algorithm and the PSO algorithm. The genetic algorithm iterated about 330 times before stability and had the highest fitness; the PSO algorithm iterated about 300 times before stability and had the second-highest fitness; the improved PSO algorithm iterated about 200 times before stability and had the third-highest fitness. The paths planned by the genetic algorithm, the PSO algorithm and the improved PSO algorithm took 2.50 s, 2.00 s and 1.00 s, respectively, and the changes in the angle, angular and angular acceleration were continuous and smooth. The genetic algorithm consumed 3.2 s in computing, the PSO algorithm consumed 2.8 s, and the improved PSO algorithm consumed 1.3 s.

The limitation of this paper is that only the improved PSO algorithm was compared with the traditional PSO algorithm and GA, the effectiveness of the improved PSO algorithm was initially verified, and the improved PSO algorithm was more suitable for trajectory planning with fixed targets. Therefore, the future research direction is to compare the improved PSO algorithm with other optimization algorithms and try to realize trajectory planning with real-time changing targets.

Acknowledgements This study was supported by Scientific Research Foundation for Advanced Talents, Taizhou University: Research on key technologies of precision detection for intelligent manufacturing (TZXY2020QDJJ006).

References

1. Chen G, Su SH (2020) Driver-behavior-based robust steering control of unmanned driving robotic vehicle with modeling uncertainties and external disturbance. Proc Inst Mech Eng Part D J Automob Eng 234(6):095440701989515
2. Marchese AD, Tedrake R, Rus DL (2015) Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator. Proc IEEE Int Conf Robot Autom 8:2528–2535
3. Kim JJ, Lee JJ (2017) Trajectory optimization with particle swarm optimization for manipulator motion planning. IEEE Trans Industr Inf 11(3):620–631
4. Prianto E, Kim MS, Park JH, Bae JH, Kim JS (2020) Path planning for multi-arm manipulators using deep reinforcement learning: soft actor-critic with hindsight experience replay. Sensors 20(20):5911
5. Wall DG, Economou J, Knowles K (2018) Quasi-real-time confined environment path generation for mobile robotic manipulator arms. Proc Inst Mech Eng Part I J Syst Control Eng 232(3):095965181775131
6. Jia Q, Li T, Chen G, Sun H, Zhang J (2016) Trajectory optimization for velocity jumps reduction considering the unexpectedness characteristics of space manipulator joint-locked failure. Int J Aerosp Eng 2016:1–14
7. Gallant A, Gosselin C (2018) Extending the capabilities of robotic manipulators using trajectory optimization. Mech Mach Theory 121:502–514
8. Shareef Z, Trächtler A (2016) Simultaneous path planning and trajectory optimization for robotic manipulators using discrete mechanics and optimal control. Robotica 34(06):1322–1334
9. Chen G, Yuan B, Jia Q, Fu Y, Tan J (2019) Trajectory optimization for inhibiting the joint parameter jump of a space manipulator with a load-carrying task. Mech Mach Theory 140(11):59–82
10. Bagheri M, Naseradinmousavi P (2017) Novel analytical and experimental trajectory optimization of a 7-DOF baxter robot: global design sensitivity and step size analyses. Int J Adv Manuf Technol 93(9–12):1–15
11. Ren ZW, Li CG, Sun LN (2016) Minimum-acceleration trajectory optimization for humanoid manipulator based on differential evolution. Int J Adv Rob Syst 13(2):1
12. Ayten KK, Dumlu A (2017) Trajectory optimization for redundant/hyper redundant manipulators. Int J Adv Appl Sci 4(12):1–4
13. Wang QY, Cheng X, Wang CL, Guo WQ, Li ZY (2019) Inverse solution optimization and research on trajectory planning of cleaning manipulator for insulator. IOP Conf Ser Mater Sci Eng 493(1):12060–12060
14. Zhou HB, Zhou S, Yu J, Zhang ZD, Liu ZZ (2020) Trajectory optimization of pickup manipulator in obstacle environment based on improved artificial potential field method. Appl Sci 10(3):935