



# An improved PSO algorithm for time-optimal trajectory planning of Delta robot in intelligent packaging

Cheng LIU<sup>1,2</sup> · Guo-Hua CAO<sup>1</sup> · Yong-Yin QU<sup>2</sup> · Yan-Ming CHENG<sup>2</sup>

Received: 13 August 2019 / Accepted: 13 September 2019 / Published online: 5 November 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

With the advancement of the times, robotics technology is also developing rapidly. Some large enterprises in China use robots to work in manufacturing and handling positions, and also make robots more and more widely used. This paper focuses on the trajectory planning strategy for three-degree-of-freedom high-speed parallel manipulator of Delta robot in Cartesian space under high-speed operation handling, the point-to-point “door” type handling operation trajectory under the condition of ensuring control accuracy and increasing productivity in intelligent packaging is established based on the inverse kinematics model of the manipulator. The 4–3–3–4 degree polynomial interpolation is presented to control height of obstacle avoidance and trajectory length, on this basis, the mapping relationship between the motion features in operation space and those in joint space is established. Taking into account the complexity of trajectory optimization due to multiple constraints, in order to reduce the difficulty of trajectory optimization, it is necessary to ensure smoothness and constrains of angular displacement, angular velocity and angular acceleration of each joint in space, an improved particle swarm optimization algorithm is proposed to optimize trajectory running time of the 4–3–3–4 degree polynomial interpolation. The simulation results by using Matlab indicate that the accurate and stable time-optimal trajectory planning of 4–3–3–4 degree polynomial interpolation can be achieved by means of the improved particle swarm optimization algorithm. Compared with other trajectory planning algorithms, the proposed algorithm is easier to implement, which not only improves the local convergence of particle swarm optimization algorithm, achieves the time optimal trajectory planning of Delta robot, but also realizes the controllability of obstacle avoidance height, therefore which realizes the fast, accurate and safe operation in intelligent packaging.

**Keywords** Intelligent manufacturing · 4–3–3–4 degree polynomial · Trajectory planning · Time-optimal · Particle swarm optimization · Delta robot · Intelligent packaging

## 1 Introduction

Compared with serial robots, parallel industrial robots have a series of advantages, such as compact structure, high stiffness, fast pickup and placing operation, high repetitive positioning accuracy, strong load-bearing capacity, etc. [1, 2]. Therefore, under the application of high-speed and high-precision, it must be widely concerned.

Now it is widely used in the assembly line of food packaging, food packaging, and electronic products grabbing, which implement mass quantities, repeatability, fast grabbing, sorting, assembly and other operations. The researches show that the trajectory and motion law of the robot end effector in motion process will affect the dynamic accuracy, running time, energy consumption and life of the mechanism, therefore effective trajectory planning is very important. Common motion laws include polynomial function [3, 4], B-spline curve [3–5], Bezier curve [6–8], etc. Polynomial interpolation function is often used in robot trajectory planning because of its simple calculation. The piecewise polynomials with successive angular displacement, angular velocity and angular acceleration and the minimum degree polynomial are combined into cubic spline interpolation. The trajectory

✉ Guo-Hua CAO  
bhdqxy@163.com

<sup>1</sup> College of Mechanical and Electric Engineering, Changchun University of Science and Technology, Changchun, China

<sup>2</sup> College of Electrical and Information Engineering, Beihua University, Jilin, China

planning of Delta robot mainly carries out point-to-point pickup and placing operation (PPO). The trajectory planning of Delta robot should have reasonable obstacle avoidance height. Generally, it is a “gate” trajectory. The main interpolation methods of “gate” trajectory are 3–3–3 interpolation and 3–5–3 polynomial. Formula interpolation and 4–3–4 polynomial interpolation are studied in [9, 10]. In consideration of the amount of interpolation calculation and the instability caused by high-speed polynomial interpolation, in this paper, the key points are added and the motion rules are interpolated by 4–3–3–4 degree polynomials to establish the mapping relationship between the motion features in operation space and those in joint space. On the premise of guaranteeing smoothness and constraints of angular displacement, angular velocity, angular acceleration smoothness and each joint and improving the motion speed of the robot, a particle swarm optimization algorithm for time optimal trajectory planning is proposed. Simulation results show that the algorithm is well applied in the time optimal trajectory planning of robots, and the search accuracy is significantly improved, which shows the effectiveness of the model and algorithm.

### 2 Kinematics analysis of parallel robot

The Delta parallel robot studied in this paper is composed of fixed platform, three active arms, three parallel quadrilateral driven arms, mobile platform and other basic parts. Three closed chains (O-Ai-Ci-Bi-P-O) are composed through rotary pair connections between the parts. The active arm receives the rotating input of three servo motors at the same time and generates end-mobile platform of robot along x axis, y axis and z axis. The base and the active arm are connected by rotating pairs, the active arm and the driven arm are connected by spherical hinges, and the driven arm and the mobile platform are also connected by spherical hinges [11].

The structure sketch of Delta parallel robot is depicted in Fig. 1. A1A2A3 is a fixed platform, B1B2B3 is a mobile platform, Ai is a rotating joint, Bi and Ci are spherical joints, AiCi is an active arm and CiBi is a driven arm, i = 1, 2, 3. O is the center of the fixed platform and P is the center of the mobile platform. R is the outer circle radius of the fixed platform, and r is the outer circle radius of the mobile platform. Both A1A2A3 and B1B2B3 are regular triangles.

#### 2.1 Establishing coordinate system

Delta robot coordinate system is established as depicted in Fig. 2. the sketch of three-dimensional coordinate system is on the left side of Fig. 2, and the analysis and modeling of a single connecting rod on the XOZ plane is on the right side of Fig. 2. The mechanism parameters of the Delta robot system are as follows: the distance between the center point of the active arm and the center point of

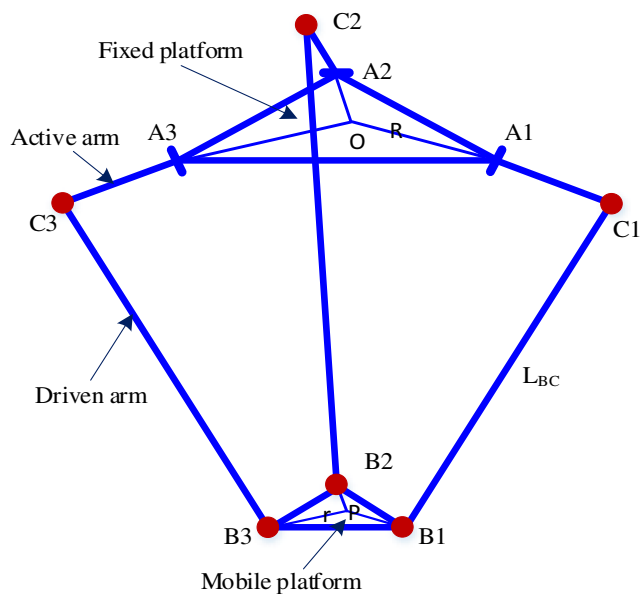


Fig. 1 Mechanism sketch of three-degree-of-freedom Delta robot

the coordinate system is R (205 mm), the length of the active arm is  $L_1$  (400 mm), the length of the driven arm is  $L_2$  (1000 mm), the distance between the center point of the mobile platform and the center point of the driven arm is r (50 mm), the angle between the active arm and the X axis is  $\theta$ , and  $L'_2$  is the projection of the driven arm on the cross section of the arm, and it is perpendicular to y of the mobile platform at the end.

#### 2.2 Inverse kinematics analysis

As can be seen from Fig. 2, the geometric modeling can be solved by using the end position of parallel mechanism. For driven arm  $L_2$  can be expressed as:

$$d_1^2 + d_2^2 = L_2^2 \tag{1}$$

At the same time,  $L_2$  is projected to  $L'_2$  on the cross section of the active arm to form a right triangle,  $L_2$  can be expressed on the Y axis of the coordinate system as follow:

$$L_2'^2 + y^2 = L_2^2 \tag{2}$$

The geometric relation can be expressed on the X-axis direction as follow:

$$|OA| + L_1 \cos(\theta_1) = -x + |DC| + d_1 \tag{3}$$

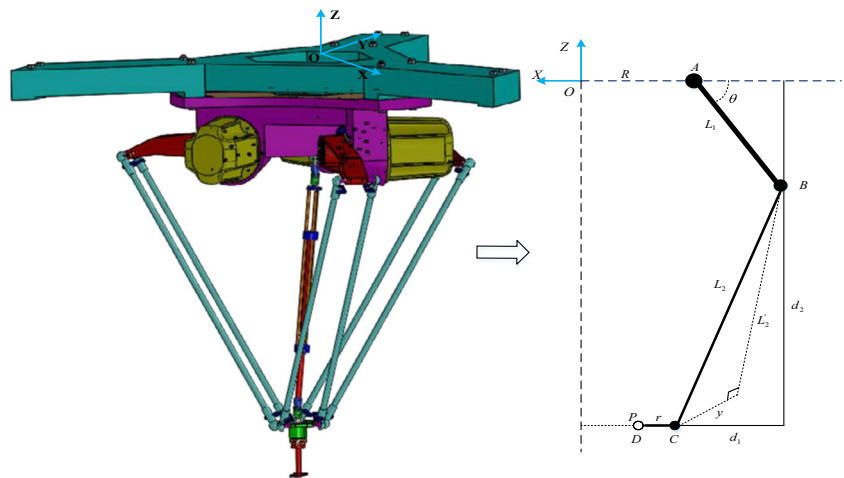
From formula (1) to (3),  $d_1$  can be calculated as follow:

$$d_1 = T + L_1 \cos(\theta_1) \tag{4}$$

Here,  $T = |OA| + x - |DC|$ .

On the Z-axis direction of the coordinate system, there is the geometric relation as follow:

**Fig. 2** Establishment of Delta Robot Coordinate System



$$d_2 = z + L_1 \sin(\theta_1) \tag{5}$$

Formulas (2), (4) and (5) are introduced into Formula (1) which is simplified as follow:

$$2TL_1 \cos(\theta_1) + 2zL_1 \sin(\theta_1) = K \tag{6}$$

Here,  $K = L_1^2 - L_2^2 - x_0^2 - z^2 - T^2$ .

By using the property formula of triangular identity expressed as Formula (7), the Formula (6) is solved and an independent variable t is constructed, the quadratic equation of t is expressed in Formula (8).

$$\begin{cases} \cos(\theta_1) = \frac{1-t^2}{1+t^2} \\ \sin(\theta_1) = \frac{2t}{1+t^2} \\ t = \tan\left(\frac{\theta_1}{2}\right) \end{cases} \tag{7}$$

$$e_1 t^2 + e_2 t + e_3 = 0 \tag{8}$$

Here,  $e_1 = 2T_1L + K, e_2 = -4zL_1, e_3 = -2T_1L + K$ .

By solving the quadratic equation of Formula (8), the expression of  $\theta_1$  can be obtained as follow:

$$\theta_1 = 2\arctan\left(\frac{-e_2 \pm \sqrt{e_2^2 - 4e_1e_3}}{2e_1}\right) \tag{9}$$

In the Formula (9), the joint angle of the first active arm can be obtained as  $\theta_1 = f(x, y, z)$ .

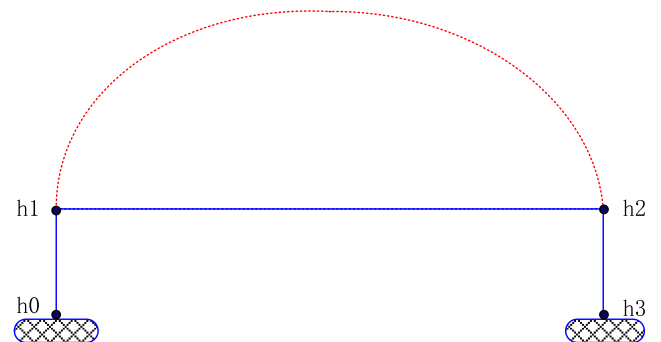
According to the above formulas, when the position coordinates of the moving platform are given, the angles of the three active arms can be obtained. Since each equation has two solutions, consequently, there are eight sets of inverse solutions. Delta robot will be limited by the motion angle in the process of running, so not all solutions meet the requirements. The value in '±' is chosen according to the actual situation, that is, the final inverse solution.

### 3 Trajectory planning

The trajectory planning of parallel robots should include the following contents: (1) describing the robot's motion path and joint space trajectory according to the task; (2) calculating the trajectory parameters of three joints in the process of rotation in real time; (3) acquiring the real-time values of the position, angular velocity and angular acceleration of the robot's joints.

#### 3.1 Description of the operating space path

In the fields of grasping and sorting, Delta Robot mainly performs point-to-point picking and placing (PPO). Gate-shaped trajectory is often used in pickup and placing operations, which includes pickup part, handling part and placing part. As shown in Fig. 3, a complete set of tasks of the mobile platform of the robot is carried out through two processes, namely, going (grab) and backing (place). In each grasping movement process, the robot manipulator act the ascending segment h0 ~ h1, the horizontal segment h1 ~ h2, and the descending segments h2 ~ h3 in turn; in each placing movement process, the robot manipulator act the ascending segment h3 ~ h2, the horizontal segment h2 ~ h1, and the descending segments h1 ~ h0 in turn. From Fig. 3, it can be seen



**Fig. 3** Trajectory of end of the robot manipulator

that on the one hand, a sudden change in the direction of motion at the corner can easily result in jerk of the robot manipulator and affect the accuracy; on the other hand, it also affects the working speed of the robot. Therefore, the arc transition should be set between point h1 and point h2, the corresponding trajectory is depicted in red dot line in Fig. 3.

**3.2 Piecewise polynomial interpolation in joint space**

In trajectory planning, the joint displacement curve can be expressed by polynomial spline function, the first derivative of polynomial spline function curve represents angular velocity curve, and the second derivative represents angular acceleration curve. Firstly, this paper uses 4–3–4 interpolation spline to solve the arc transition should be set between point h1 and point h2. According to the parameters of joint operation space of Delta robot, in Cartesian coordinate system, the starting point h0 (−400, −300, 750)mm and the terminal point h3 (200, 200, 750)mm of the space coordinates are set up, and then two interpolation points, h1 (−400, −300, 800) and h2 (200, 200, 800)mm are respectively obtained. Through resolving inverse kinematics equation Corresponding joint angel, corresponding joint angles are respectively obtained as: β<sub>j0</sub>(45.79°, 17.36°, −35.15°), β<sub>j1</sub>(49.48°, 23.06°, −26.16°), β<sub>j2</sub>(−17.48°, −4.79°, 27.27°) and β<sub>j3</sub>(−26.5°, −12.75°, 22.34°). The simulation is conducted to obtain space trajectory of 4–3–4 interpolation spline as shown in Fig. 4. It is obvious that only two key point h1 and h2 are added between picking point h0 and grabbing point h3 so that the trajectory of the handling section is uncontrollable especially the height of obstacle avoidance is up to 930 mm, as well as running time cannot be optimal. In line with the principle of reducing energy consumption and optimal running time, one

key point h4 (−100, −50, 850)mm is added as shown in Fig. 5, the entire trajectory is divided 4 segments. Setting t1, t2, t3 and t4 corresponds to the segment h0~h1, the segment h1~h4, the segment h4~h2 and the segment h2~h3, respectively. Therefore, this paper proposes 4–3–3–4 degree polynomial to carry out interpolation of these four segments, which can be expressed with Formula (10).

$$\begin{cases} h_{j1}(t) = a_{j14}t_1^4 + a_{j13}t_1^3 + a_{j12}t_1^2 + a_{j11}t_1 + a_{j10} \\ h_{j2}(t) = a_{j23}t_2^3 + a_{j22}t_2^2 + a_{j21}t_2 + a_{j20} \\ h_{j3}(t) = a_{j33}t_3^3 + a_{j32}t_3^2 + a_{j31}t_3 + a_{j30} \\ h_{j4}(t) = a_{j44}t_4^4 + a_{j43}t_4^3 + a_{j42}t_4^2 + a_{j41}t_4 + a_{j40} \end{cases} \tag{10}$$

Here, the coefficients, a<sub>j1i</sub>, a<sub>j2i</sub>, a<sub>j3i</sub>, a<sub>j4i</sub> are the ith coefficients of the spline interpolation functions of the first, second, third and fourth segments of the jth joint trajectory, HJ0 (t) is the starting point of the object grabbing, h<sub>j1</sub>(t), h<sub>j2</sub>(t) and h<sub>j3</sub>(t) is the key point of the path, and h<sub>j4</sub>(t) is the placement point of the object.

According to the zero velocity and acceleration of the starting and terminal points, and the continuity of velocity and acceleration of the intermediate points, the constraints can be obtained through Formula (11).

$$\begin{cases} h_{j1}(0) = \beta_{j0} \quad \dot{h}_{j1}(0) = 0 \quad h_{j1}(0) = 0 \\ h_{j2}(0) = h_{j1}(t_1) = \beta_{j1} \quad \dot{h}_{j2}(0) = \dot{h}_{j1}(t_1) \quad h_{j2}(0) = h_{j1}(t_1) \\ h_{j3}(0) = h_{j2}(t_2) = \beta_{j2} \quad \dot{h}_{j3}(0) = \dot{h}_{j2}(t_2) \quad h_{j3}(0) = h_{j2}(t_2) \\ h_{j3}(0) = h_{j2}(t_3) = \beta_{j3} \quad \dot{h}_{j3}(0) = \dot{h}_{j2}(t_3) \quad h_{j3}(0) = h_{j2}(t_3) \\ h_{j4}(t_4) = \beta_{j4} \quad \dot{h}_{j4}(t_4) = 0 \quad h_{j4}(t_4) = 0 \end{cases} \tag{11}$$

Matrix expression A and b of 4–3–3–4 polynomials can be constructed by deducing Formulas 10 and 11.

$$A = \begin{bmatrix} t_1^4 & t_1^3 & t_1^2 & t_1 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 12t_1^2 & 6t_1 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & t_2^3 & t_2^2 & t_2 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3t_2^2 & 2t_2 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6t_2 & 2 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_3^3 & t_3^2 & t_3 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3t_3^2 & 2t_3 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 6t_3 & 2 & 0 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t_4^4 & t_4^3 & t_4^2 & t_4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4t_4^3 & 3t_4^2 & 2t_4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12t_4^2 & 6t_4 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{12}$$

$$b = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ \beta_{j4} \ 0 \ 0 \ \beta_{j0} \ 0 \ 0 \ \beta_{j3} \ \beta_{j2} \ \beta_{j1}]^T \tag{13}$$

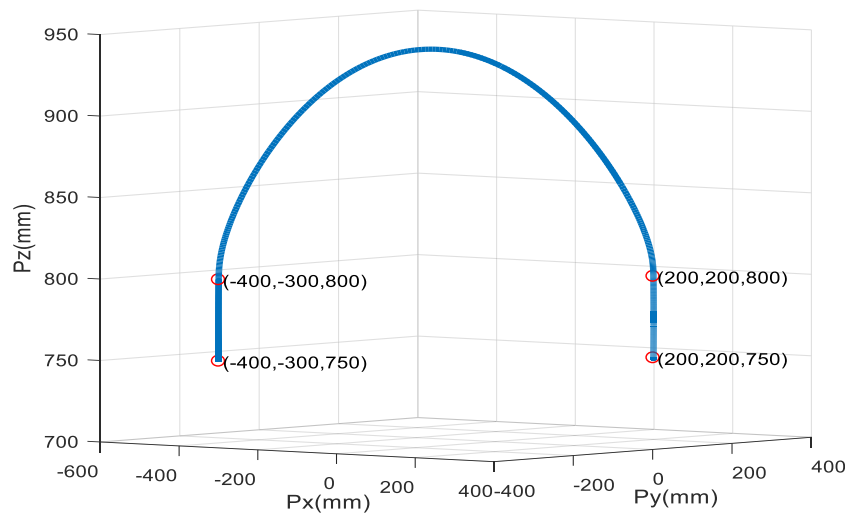


Fig. 4 Space trajectory of 4–3–4 interpolation spline

$\beta_{j0}, \beta_{j1}, \beta_{j2}, \beta_{j3}$  and  $\beta_{j4}$  respectively corresponding to the rotation angle of  $j$ th joint at starting point, two middle key points and placement point, which can be obtained by inverse kinematics equation, respectively.

Coefficients of Formula (10) can be obtained through Formulas (12) and (13).

$$a = A^{-1}b \tag{14}$$

Set  $t_1 = 0.5$  s,  $t_2 = 0.5$  s,  $t_3 = 0.5$  s,  $t_4 = 0.5$  s, the total time is 2 s. Modeling 4–3–3–4 degree polynomial interpolation in Matlab, Angular displacement curve, angular velocity curve and angular acceleration curve are shown in Figs. 6, Fig. 7 and Fig. 8, respectively. Comparison between space trajectory of 4–3–4 degree polynomial interpolation and space trajectory 4–3–3–4 degree polynomial interpolation is shown in Fig. 9. It is obvious that 4–3–3–4 degree polynomial interpolation can well control space trajectory and height of obstacle avoidance.

### 4 An improved particle swarm optimization trajectory

In order to carry out the time optimal trajectory planning of Delta robot, this paper proposes an improved particle swarm

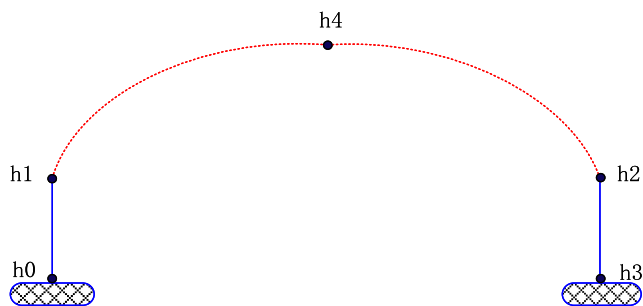


Fig. 5 Trajectory of end of the robot manipulator after inserting one key point

to optimize space trajectory 4–3–3–4 degree polynomial interpolation, particle parameters need be optimized to derive the optimal solution of each particle, so as to obtain the optimal solution of the optimization process. In the case of space trajectory 4–3–3–4 degree polynomial interpolation, the running time of Delta robot three joints passing through each interval is  $t_1, t_2, t_3$  and  $t_4$ , respectively. Formula (15) shows that  $\theta(t)$  is a function of time particle parameters:

$$\theta(t) = \min(t_1 + t_2 + t_3 + t_4) \tag{15}$$

$$\begin{cases} \max|\dot{\theta}_{j1}, \dot{\theta}_{j2}, \dot{\theta}_{j3}, \dot{\theta}_{j4}| \leq v_{\max} \\ \max|\ddot{\theta}_{j1}, \ddot{\theta}_{j2}, \ddot{\theta}_{j3}, \ddot{\theta}_{j4}| \leq a_{\max} \end{cases} \tag{16}$$

Velocity and acceleration constraints are provided by Formula (16),  $j = 1, 2, 3$  that respectively indicate three joints of Delta robot.

The fitness of each group of time particle parameters can be calculated by the time particle parameter function in Formula (15). The local and global optimal solutions of time particle parameters need to be calculated and compared by Formulas (7) ~ (21) and (15):

$$V_{id} = wV_{id} + c_1r_1(P_{id} - X_{id}) + c_2r_2(P_{gd} - X_{id}) \tag{17}$$

$$X_{id} = X_{id} + \alpha V_{id} \tag{18}$$

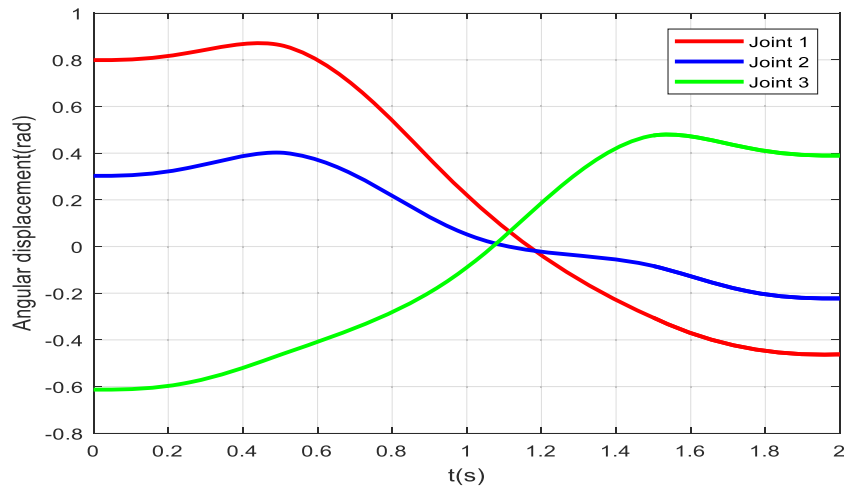
$$w = w_{\max} - (w_{\max} - w_{\min}) \left( \frac{n}{N} \right) \tag{19}$$

$$c_1 = 2\sin^2 \left[ \frac{\pi}{2} \left( 1 - \frac{1}{N} \right) \right] \tag{20}$$

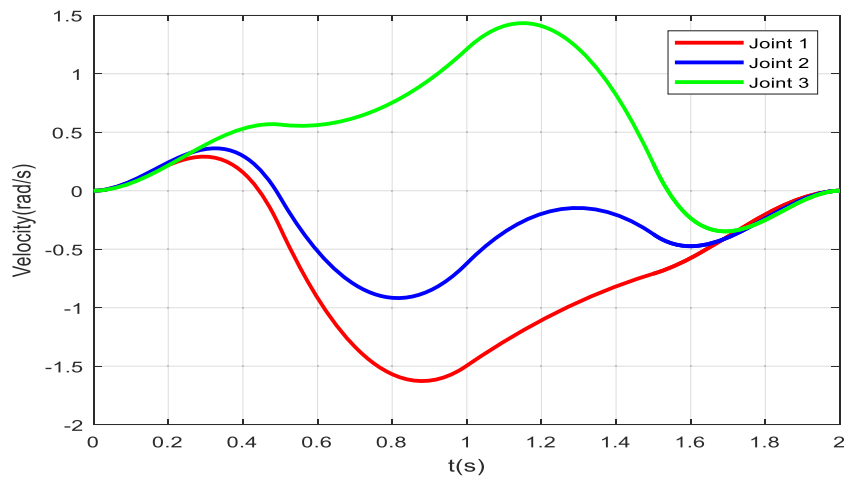
$$c_2 = 2\sin^2 \left[ \frac{\pi n}{2N} \right] \tag{21}$$

In Formulas (17) and (18),  $i = 1, 2, 3, \dots$  indicates the number of particles,  $d = 1, 2, 3$  indicates the number of dimensions;  $w$  indicates the inertia weight whose the range of values is non-negative which can be obtained in Formula (19),

**Fig. 6** Angular displacement of 4–3–3–4 degree polynomial interpolation



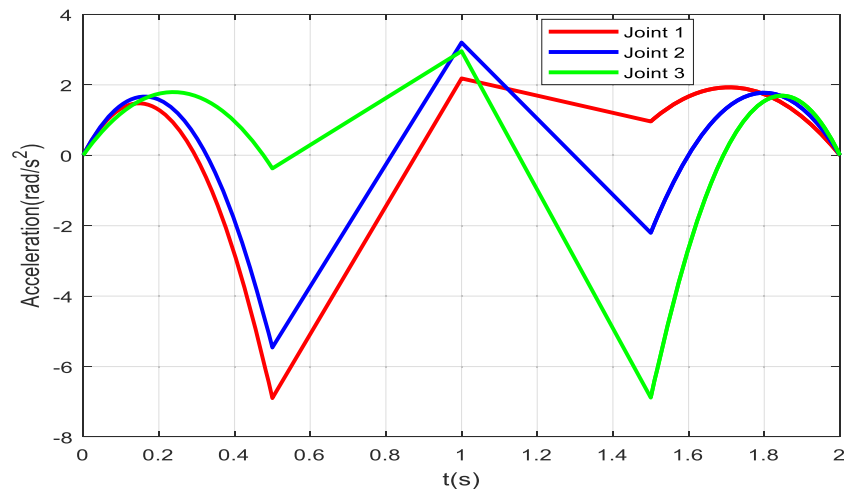
**Fig. 7** Velocity of 4–3–3–4 degree polynomial interpolation



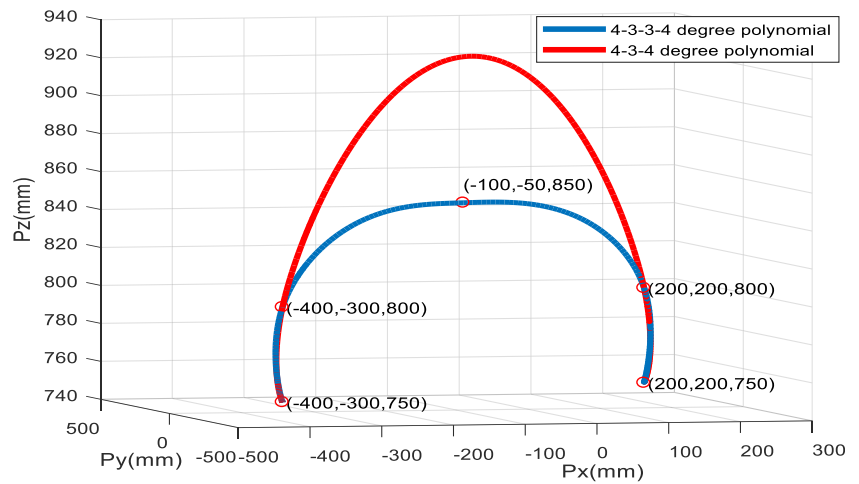
$w_{\max}$  is maximum inertia weight,  $w_{\min}$  is minimum inertia weight,  $n$  is current iteration times of run,  $N$  is total iteration times. The larger  $w$  has better global convergence ability, while the smaller  $w$  has stronger local convergence ability.

Therefore, as the number of iterations increases, the inertia weight  $w$  should be reduced continuously, so that the PSO algorithm has a strong global convergence ability in the initial stage, and a strong local convergence ability in the late stage.

**Fig. 8** Acceleration of 4–3–3–4 degree polynomial interpolation



**Fig. 9** Space trajectory comparison



$c_1$  and  $c_2$  are the acceleration constant whose the range of values is non-negative, they can be calculated by using Formulas (20) and (21);  $r_1$  and  $r_2$  are random numbers in the range of 0 ~ 1; And  $\alpha$  is the constraint factor that is used to control the weight of speed.

In this algorithm, dynamic change learning factor is used, and the learning factor is determined by using formula (20) and formula (21). The global learning factor decreases with the increase of search times, while the local learning factor increases with the increase of search times. The dynamic change learning factor makes the search reach global optimum quickly in the early stage and local optimum quickly in the later stage, which improves the search efficiency and the result shows that the local optimum solution is insufficient.

Detailed steps for the time trajectory planning of the  $j$ th joint of Delta robot optimized by PSO are as follows:

Step 1. Set ‘popsize’ as the size of the population that usually is 20. In the 3-dimensional search space of interpolation time, ‘popsize’ particles are randomly generated to form the initial population, and the position and velocity of the particles are initialized.

Step 2. The unknown coefficients of 4–3–3–4 degree polynomials are obtained by ‘popsize’ dimensional  $t_1, t_2, t_3$  and  $t_4$  substituted into Formulas (11) ~ (14).

Step 3. By introducing coefficients of 4–3–3–4 degree polynomials into Formula (10) and deriving the time, the joint angular velocity can be obtained and the real-time velocity and acceleration can be judged to satisfy Formula (16).

**Table 1** Range of rotation angle of joint

Rotation Angle of Joint $\theta_i$	Range(degree)
$\theta_1$	-40~80
$\theta_2$	-40~80
$\theta_3$	-40~80

Step 4. Calculate the fitness of each particle. The results of step 3 are screened. If the real-time velocity and acceleration of any segment in the four segments does not meet the Formula (16), the fitness value of the particle is set to a great constant. When searching for the optimal particle, the particle with larger fitness value will be excluded and not screened as the optimal particle by comparing the fitness values. And the particle itself will gradually move closer to the optimal value until it meets the velocity constraints of Formula (16). If the real-time maximum velocity and acceleration of the four segments is in accordance with the Formula (16), then the fitness function is formulated as Formula (15), and the objective of particle swarm optimization iteration is to obtain the minimum interpolation time.

Step 5. For each particle, its fitness value is compared with the fitness value of the best position  $P_{id}$  it has experienced. If it is better, it is replaced by the current best position  $P_{id}$ .

Step 6. Comparing the fitness value of the best position of each particle at present, the best overall particle at present is selected, and then compared with the best global position  $P_{gd}$  experienced by the population. If  $P_{id}$  is better, it replaces  $P_{gd}$ .

Step 7. According to Formulas (17) and (18), the velocity and position of particles are changed, and a new population is

**Table 3** Optimization results

Optimization	Time (s)				Total amount of $t_i$ (s)
	$t_1$	$t_2$	$t_3$	$t_4$	
1	0.0500	0.0873	0.0758	0.0500	0.2631
2	0.0500	0.0873	0.0759	0.0500	0.2632
3	0.0500	0.0873	0.0760	0.0500	0.2633
4	0.0500	0.0871	0.0772	0.0500	0.2643
5	0.0500	0.0887	0.0758	0.0500	0.2645

**Table 2** Optimization results

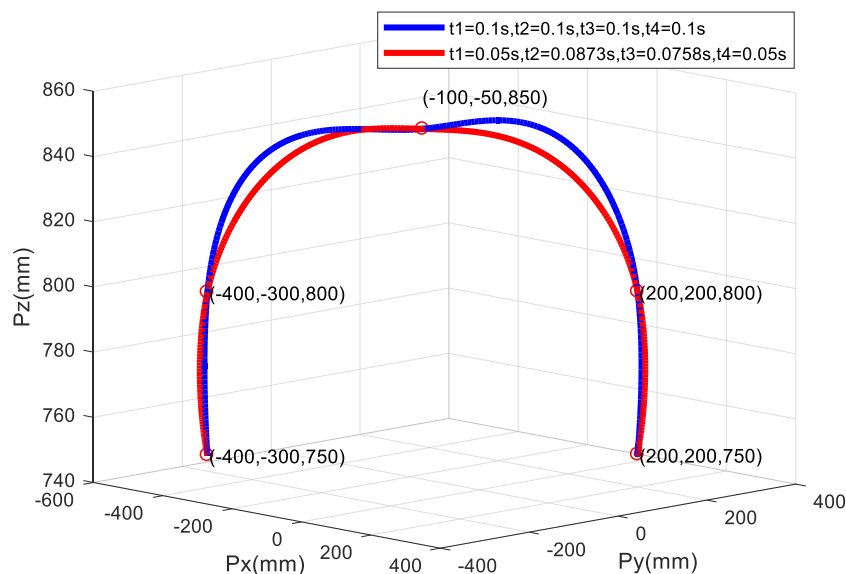
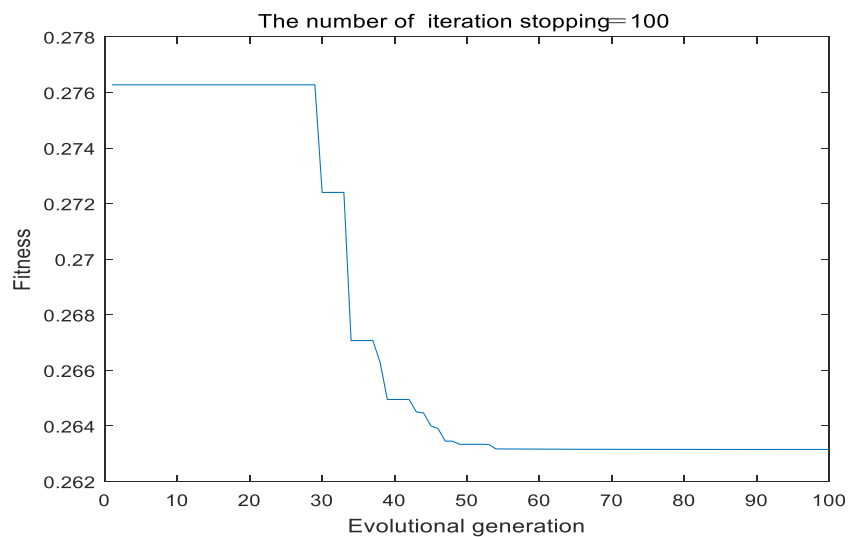
Trajectory points	Position coordinates (mm)	Corresponding rotation angle of three joints (degree)
h0	(-400, -300,750)	(45.79, 17.36, -35.15))
h1	(-400,-300,800)	(49.48, 23.06, -26.6)
h4	(-100,-50,850)	(12.66, 2.99, -5.12)
h2	(200,200,800)	(-17.48, -4.79,27.27)
h3	(200,200,750)	(-26.5, -12.75, 22.34)

reconstituted by the number of ‘popsize’ particles, and Steps 2,3 and 4 are repeated.

Step 8. If the termination condition is satisfied (usually a sufficiently good fitness value or a preset maximum number of iterations), the algorithm ends, otherwise step 2 is returned.

## 5 Modeling and simulation for Delta robot

The model of three-degree-of-freedom high-speed parallel manipulator of Delta robot is established by programming, the range of each arm joint variable is determined according to the manipulator structure, as summarized in Table 1.

**Fig. 10** Fitness curve**Fig. 11** Comparison of space trajectories before and after time optimization



According to assumed starting point  $h_0$ , terminal point  $h_3$  and three key points  $h_1$ ,  $h_2$ , and  $h_4$  in section .32, corresponding rotation angle of three joints can be obtained by solving inverse kinematics equations, they are summarized in Table 2.

In this paper, popsize of 20 is set up; the maximum velocity of a particle is in range of  $[-2, 2]$ . The iteration times  $N$  is set up 200. The range of  $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$  corresponding to each segment of space trajectory 4–3–3–4 degree polynomial interpolation are set up from  $[0.01 \text{ s}, 0.01 \text{ s}, 0.01 \text{ s}, 0.01 \text{ s}]$  to  $[0.1 \text{ s}, 0.1 \text{ s}, 0.1 \text{ s}, 0.1 \text{ s}]$ ,  $[0.1 \text{ s}, 0.1 \text{ s}, 0.1 \text{ s}, 0.1 \text{ s}]$  as upper limit of  $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$  is assumed smaller values, as well as is verified by manual testing, which is reasonable under the constraints of maximum velocity and maximum acceleration. And then one least set  $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$  is searched through the improved PSO. Because the coefficients of degree polynomial interpolation are determined by time (particles), the angular velocity changes with each particle renewal in the process of particle swarm optimization, which result in that the combination of time optimum is not unique. Consequently, 5 optimizations in Table 3 are selected out as optimal trajectory planning time after a certain number of optimization operations have been conducted. Figure 10 depict iteration curve of fitness of one optimum time set ( $t_1 = 0.05 \text{ s}$ ,  $t_2 = 0.0873 \text{ s}$ ,  $t_3 = 0.075 \text{ s}$ ,  $t_4 = 0.05 \text{ s}$ ).

Space trajectory corresponding to ( $t_1 = 0.05 \text{ s}$ ,  $t_2 = 0.0873 \text{ s}$ ,  $t_3 = 0.075 \text{ s}$ ,  $t_4 = 0.05 \text{ s}$ ) and Space trajectory corresponding to ( $t_1 = 0.1 \text{ s}$ ,  $t_2 = 0.1 \text{ s}$ ,  $t_3 = 0.1 \text{ s}$ ,  $t_4 = 0.1 \text{ s}$ ) are shown in Fig. 11. By calculating, the time has been reduced by  $((0.4 - 0.2631) / 0.4) * 100\% = 34.23\%$  compared with the before the optimized time.

## 6 4. Conclusion

This paper firstly presented the 4–3–3–4 degree polynomial interpolation to achieved trajectory planning of Delta Robot, simulations results indicate that compared with the 4–3–4 degree polynomial interpolation, the 4–3–3–4 degree polynomial interpolation have better performances in optimizing time path and adjusting obstacle avoidance height flexibly. And then, in order to further optimizing trajectory planning time the 4–3–3–4 degree polynomial interpolation under the constraints of maximum velocity and maximum acceleration, an improved particle swarm optimization algorithm for time-optimal trajectory planning of delta robot is put forward, simulation results

show that compared with before and after time optimization, space trajectory time of the 4–3–3–4 degree polynomial interpolation for Delta robot can be greatly shortened which will further increase production efficiency. The follow-up issues to be taken into account such as jerk phenomenon of 3–3 degree polynomial interpolation in 4–3–3–4 degree polynomial interpolation, trajectory planning of 4–4–4–4 or 4–5–5–4 degree polynomial for Delta robot interpolation should be taken into consideration in the future.

**Acknowledgements** This works was supported by the Jilin provincial development and reform commission (Grant: 2018C035-1), Jilin provincial science and technology innovation talents & team project (Grant: 20190101018JH), and Jilin provincial science and Technology Department (Grant: 20160101276JC).

## References

1. Brogardh T (2006) Present and future robot control development-an industrial perspective. *Annu Rev Control* 31(1):69–79
2. Chen XS, Chen ZL, Kong MX (2002) Recent De-velopment and current status of Stewart platform research. *Robot* 24(5):464–470
3. Jiang MP, Zang JW, Qiao ZY (2016) Trajectory planning of 3-DOF Delta parallel manipulator. *Journal of Mechanical Engineering* 52(19):9–17
4. Gasparetto A, Zanotto V (2010) Optimal trajectory planning for industrial robots. *Adv Eng Softw* 41(4):548–556
5. Huashan L, Xiao BL, Wu WX (2013) Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints. *Robot Comput Integr Manuf* 29(2):309–317
6. Zhou Y (2012) Bezier curve Cartesian space trajectory planning based on improved generic algorithm. *Journal of Shanghai Dianji University* 15(4):237–240
7. Sun L, Zhang LS, Zhou L et al (2018) A new method on trajectory planning for Mobile robots using Bezier curves. *Journal of System Simulation* 30(03):962–968
8. Wang MM, Luo JJ, Ulrich W (2015) Trajectory planning of free-floating space robot using Partical swarm optimization (PSO). *Acta Astronautica* 112:77–88
9. He P, Liu H, Jin MH (2003) Trajectory planning of robot manipulator based on spline function. *Robot* 25(z1):614–618
10. Ju HH, Fu R (2012) Time-optimal trajectory planning algorithm based on GA for manipulator. *Control Engineering of China* 19(3):472–477
11. Misyurin YUS et al (2016) Determination of the degree of mobility and solution of the direct kinematic problem for an analogue of a delta robot. *J Mach Manuf Reliab* 45(5):403–411

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.