Computer Communications 181 (2022) 446–453

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Time-optimal trajectory planning of manipulator with simultaneously searching the optimal path

Xiuli Yu, Mingshuai Dong, Weimin Yin *

School of Modern Post, Beijing University of Posts and Telecommunications, Haidian, Beijing, China

A R T I C L E I N F O

Keywords:
Time-optimal trajectory planning
Cubic B-spline interpolation
Genetic algorithm

A B S T R A C T

Time-optimal trajectory planning can improve the efficiency of the manipulator, which has a paramount research significance. Nonetheless, almost all the current algorithms have the problem of path limitation — the algorithm constantly searches for the optimal motion time of the manipulator on a specific motion path. To solve this problem, we design a time-optimal trajectory planning method for the manipulator by searching the optimal path simultaneously. We propose using a cubic uniform B-spline interpolation algorithm to derive the motion curve expression of the manipulator joint with unknown path points. Then we use a genetic algorithm to optimize the path point and the motion time of the manipulator at the same time to get the time-optimal motion path of the manipulator. Moreover, we prove that the algorithm proposed in this paper is comparable with similar algorithms with known paths through experiments.

1. Introduction

The manipulator is widely used in industrial production and logistics transportation, which plays an important role in improving production capacity [1]. Time-optimal trajectory planning can effectively help the manipulator to shorten the movement time and improve work efficiency. However, it is still challenging to guarantee that the manipulator can search for the optimal movement time in the optimal path.

At present, most researchers use polynomial interpolation, spline interpolation, and B-spline interpolation to fit the motion curve of the manipulator.

The polynomial interpolation algorithm mainly faces the robot's point-to-point motion, and the polynomial coefficient is only determined by the beginning and end states of each joint and the motion time. Polynomial interpolation algorithm has been widely used in robot trajectory planning because of its advantages, such as simple calculation, smooth curve, and controllable beginning and end states. However, the motion curve fitted by the polynomial interpolation algorithm has a gentle trend and a slow average speed. When fitting the multi-segment continuous motion curve, the velocity or acceleration of the piecewise points cannot be differentiated.

The spline interpolation algorithm divides the motion curve into several segments and uses polynomial to fit the motion curve of each segment. The spline interpolation algorithm calculates the polynomial coefficient under the condition that the state before and after the segment point is the same, ensuring the continuity of each segment curve at the segment point so that the segment point can be differentiated. However, the motion curves on different segments are determined by their polynomial parameters. Therefore, the curves are also considerably different, and it is difficult to guarantee the smoothness of the whole motion curve.

B-spline interpolation algorithm is used to fit the robot motion curve through a series of control points, and the motion curve is also divided into several segments. Several control points jointly determine the B-spline interpolation curve, and the number of control points is determined according to their order. B-spline interpolation algorithm guarantees the continuity of each segment curve and improves the smoothness of the whole motion curve.

Heuristic algorithms, such as genetic algorithm, ant colony algorithm, particle swarm optimization, and so on, are generally used to solve the kinematic time optimization problem of the manipulator. The heuristic optimization algorithm calculates a better solution based on existing experience and selects the most effective solution as the optimization result rather than solving it through fixed steps. Therefore, it can effectively solve the optimization problem with discrete objective functions and complex constraints.

The Genetic algorithm is an optimization method inspired by the theory of biological evolution in nature. The algorithm realizes the optimization and update of the solution set by genetic characteristics and retains excellent individuals by the survival of the fittest. Since individual excellence is only determined by fitness function and does not involve the derivation of the objective function, genetic algorithm has excellent advantages in complex optimization problems. However,

* Corresponding author.
E-mail addresses: yxl@bupt.edu.cn (X. Yu), 18811379500@163.com (W. Yin).

https://doi.org/10.1016/j.comcom.2021.10.005
Received 22 April 2021; Received in revised form 8 September 2021; Accepted 2 October 2021
Available online 11 October 2021
0140-3664/

there is a certain randomness in the solving process of genetic algorithm, which cannot ensure that the optimization results are global optimal every time, and it is easy to appear premature phenomenon.

Ant colony algorithm is an optimization method inspired by the process of ants searching for food. Each ant will leave a specific concentration of pheromone in the search process. The better the search results are, the higher the pheromone concentration will be, and the ants will continue to search in the area; on the other hand, the lower the pheromone concentration in the search area, the ants will expand their search area. The ant colony algorithm approaches the optimal solution gradually by this approximate positive feedback mechanism. However, the ant colony algorithm has many parameters and is coupled with each other. At present, the method of repeated trial and error is adopted to optimize the algorithm. Therefore, the optimization efficiency of the algorithm is not satisfactory.

Researchers proposed a particle swarm optimization (PSO) algorithm based on the study of birds' predation behavior. Birds can share food information, and all individuals are close to the current optimal individual in the following population information update. The algorithm adapts to different environments through information sharing among individuals. However, particle swarm optimization is more suitable for continuous problems than discrete or complex combinatorial optimization problems.

In this paper, we propose a cubic B-spline interpolation algorithm to fit the motion curve of the manipulator. And then we use an improved genetic algorithm for optimizing the path point and motion time to generate a satisfactory running curve. In summary, the contributions of this paper are as follows:

1. In this paper, we propose a cubic uniform B-spline interpolation algorithm to derive the joint motion expressions in the case of unknown path points in the point-to-point motion of the manipulator.

2. We propose to optimize the path points and time of manipulator motion simultaneously by improving the genetic algorithm to break the path limitations of the general time-optimal trajectory planning method.

3. We employ the AUBO I5 manipulator to verify the proposed algorithm through simulation. The experimental results show that the performance of our proposed algorithm is comparable with the state-of-the-art algorithm.

2. Related work

Currently, the mainstream time optimal trajectory planning methods mostly use polynomial interpolation and spline interpolation algorithms [2–4] to fit the manipulator motion path and then use genetic algorithm, ant colony algorithm, or particle swarm algorithm [5–9] to optimize the motion time. However, the path fitted by the above method is greatly affected by human factors, which often makes it not the optimal path in the trajectory space of the manipulator.

[10] uses super-concave convex surfaces to reconstruct the three-dimensional working space of the manipulator and proposes a method of time-optimal trajectory planning for the super-redundant manipulator, which optimizes the running time of the manipulator by a multi-population genetic algorithm.

Based on the five-order polynomial interpolation trajectory planning, [11] proposed an elite genetic algorithm to improve the optimization performance. Simultaneously, [11] combined the genetic algorithm with the singularity avoidance mechanism to avoid the occurrence of singularity points in the trajectory.

[12] apply an adaptive genetic algorithm to optimize the motion time of the trajectory fitted by cubic spline interpolation, which achieves a faster convergence speed by adaptively adjusting the probability of crossover and mutation.

[13] proposed a time-optimal trajectory planning algorithm based on two-population heredity and chaotic local search. The algorithm combines the advantages of genetic algorithm and chaotic algorithm, which have good global search performance and fast evolution speed, to plan the optimal trajectory of the manipulator.

Based on the trajectory planning algorithm of cubic spline interpolation, [14] uses a genetic algorithm improved by the clustering method to optimize the motion time. This method considers the diversity of the population while maintaining good individual performance and improving search efficiency.

All the above literature put forward good improvements in the time-optimal trajectory planning algorithm, which is of great help to the research of this paper. However, they all have the problem of path limitation. That is, the algorithms always search for the optimal motion time under a fixed path, but do not discuss whether this path is optimal. Imagine a situation where the end of the manipulator has to go from point A to point B without passing through any particular position in between. According to the current time optimal trajectory planning algorithm, the motion path of the manipulator should be determined according to the initial and end states of the manipulator or several path points should be determined manually to fit a fixed motion curve, and then the motion time should be optimized by the optimization algorithm. There is a high probability that the trajectory optimized this way will not be globally optimal.

Therefore, it is a valuable problem how to optimize the motion time on the optimal path as much as possible in the research of point-to-point trajectory planning of manipulator. To address this problem, we propose a time-optimal trajectory planning method for the manipulator, which simultaneously searches the optimal path. Firstly, the method uses a cubic uniform B-spline curve to fit the general expression of the manipulator joint motion curve with unknown path points. Then, we use the improved genetic algorithm to search for the optimal path points and the optimal motion time simultaneously. The fitness function adds penalty terms to restrict the angular velocity, angular acceleration, and angular jerk of the manipulator. Besides, the convergence speed of the algorithm is accelerated by greedy parent selection and adaptive crossover and mutation. In this paper, we simulate and verify our algorithm by using the Aubo-I5 manipulator model. The results show that our proposed algorithm in this paper is effective and has a shorter motion time than other similar algorithms.

3. Kinematics model of the serial manipulator

The method of time-optimal trajectory planning proposed in this paper is suitable for the serial manipulator structure, which is the most common manipulator structure at present. The serial manipulator is made of a group of rigid links connected in series by rotating joints. And establishing the kinematics model of the manipulator is the basis to control its motion trajectory.

The modified D–H method proposed by Khalil and Kleinfinger in 1986 is a standard modeling method for series manipulators at present [15]. This method establishes the origin of the connecting rod coordinate system at the end of the joint connecting rod, which has more straightforward steps and less ambiguity than the traditional D–H modeling method.

This paper establishes the kinematics model based on a six-axis series manipulator structure. Firstly, it constructs the three-dimensional coordinate system of every connecting rod shown in Fig. 1 according to the modified D–H method. The axis z_j coincides with the rotation center axis of the joint j. The origin o_j is the intersection of the axis z_j and the axis z_{j+1} when they are intersecting, or the junction of their public vertical line and the axis z_j. The axis x_j coincides with the public vertical line of the axis z_j and the axis z_{j+1}, whose direction is from z_j to z_{j+1}, and the axis y_j is determined by the right-hand rule according to the axis x_j and axis z_j. The calculation of the transformation matrix

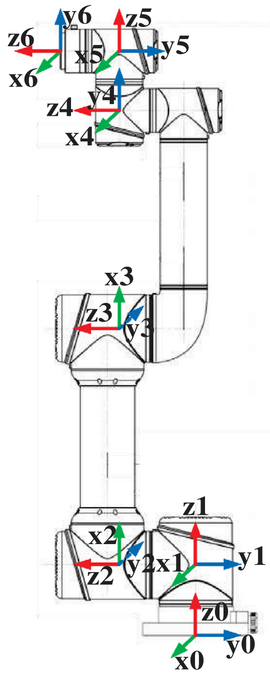

Fig. 1. Mechanical structure of six-axis manipulator.

between two adjacent joint coordinate systems is as follows.

$$
\begin{aligned}
&{}^{j-1}_{\ \ j}\boldsymbol{T} = \boldsymbol{Rot}(x,\alpha_{j-1})\cdot\boldsymbol{Trans}(x,a_{j-1})\cdot\boldsymbol{Rot}(z,\theta_j)\cdot\boldsymbol{Trans}(z,d_j)\\
&= \begin{bmatrix} \cos\left(\theta_j\right) & -\sin\left(\theta_j\right) & 0 & a_{i-1} \\ \cos\left(\alpha_{j-1}\right)\sin\left(\theta_j\right) & \cos\left(\alpha_{j-1}\right)\cos\left(\theta_j\right) & -\sin\left(\alpha_{j-1}\right) & -d_i\sin\left(\alpha_{j-1}\right) \\ \sin\left(\alpha_{j-1}\right)\sin\left(\theta_j\right) & \cos\left(\theta_j\right)\sin\left(\alpha_{j-1}\right) & \cos\left(\alpha_{j-1}\right) & d_i\cos\left(\alpha_{j-1}\right) \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{1}
$$

Among them, α_{j-1} is the positive rotation angle from z_{j-1} to z_j about x_{j-1}, a_{j-1} is the distance between z_{j-1} after rotation to z_j along the positive direction of x_{j-1}. θ_j is the positive rotation angle from x_{j-1} to x_j about z_j, which is also the joint angle that actually controls the motion of the robot arm, and d_j is the distance between x_{j-1} after rotation to x_j along the positive direction of z_j. The coordinate system (x_{j-1}, y_{j-1}, z_{j-1}) completely coincides with the coordinate system (x_j, y_j, z_j) after the above four transformations. The coordinate transformation matrix between the end coordinate system and the base coordinate system of the manipulator, namely, the forward solution of the kinematics model is as follows.

$$
\begin{aligned}
{}^0_6\boldsymbol{T} &= {}^0_1\boldsymbol{T}(\theta_1)\cdot{}^1_2\boldsymbol{T}(\theta_2)\cdot{}^2_3\boldsymbol{T}(\theta_3)\cdot{}^3_4\boldsymbol{T}(\theta_4)\cdot{}^4_5\boldsymbol{T}(\theta_5)\cdot{}^5_6\boldsymbol{T}(\theta_6)\\
&= \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}
\end{aligned}
\tag{2}
$$

Therefore, the position and posture of the manipulator end are computable by known joint angles, and the motion trajectory is also controllable throw the angular displacement curve Q (θ_j, t) of the manipulator's joints. The space trajectory planning of the manipulator is transformed into the angular displacement curve planning of the joints.

By multiplying both sides of Eq. (2) by ${}^0_1\boldsymbol{T}_{-1}$, we use the inverse transformation method to calculate the inverse solution of the kinematics model as follows.

$$
\begin{cases}
\theta_1 = \operatorname{atan2}(d_2, \pm\sqrt{r^2-d_2^2}) - \operatorname{atan2}(B, A)\\
\theta_5 = \operatorname{atan2}(\pm\sqrt{1-(a_y c_1 - a_x s_1)^2}, -a_y c_1 + a_x s_1)\\
\theta_6 = \operatorname{atan2}(\dfrac{o_y c_1 - o_x s_1}{-s_5}, \dfrac{n_y c_1 - n_x s_1}{s_5})\\
\theta_2+\theta_3+\theta_4 = \operatorname{atan2}(\dfrac{a_z}{s_5}, \dfrac{a_x c_1 + a_y s_1}{s_5})\\
\theta_2+\theta_3 = \operatorname{atan2}(\dfrac{B_2 - a_2 s_2}{a_3}, \dfrac{B_1 - a_2 c_2}{a_3})\\
\theta_2 = \operatorname{atan2}(C, \pm\sqrt{B_1^2+B_2^2-C^2}) - \operatorname{atan2}(B_1, B_2)\\
\theta_3 = (\theta_2+\theta_3) - \theta_2\\
\theta_4 = (\theta_2+\theta_3+\theta_4) - (\theta_2+\theta_3)
\end{cases}
\tag{3}
$$

Where $A = p_x - a_x d_6$, $B = a_y d_6 - p_y$, $r = \sqrt{A^2+B^2}$, $B_1 = p_x c_1 + p_y s_1 + d_5 s_{234} - d_6 c_{234} s_5$, $B_2 = p_z - d_1 - d_5 c_{234} - d_6 s_{234} s_5$, $C = \frac{B_1^2+B_2^2+a_2^2-a_3^2}{2a_2}$, $c_1 = \cos\theta_1$, $s_1 = \sin\theta_1$, $c_{234} = \cos(\theta_2+\theta_3+\theta_4)$, $s_{234} = \sin(\theta_2+\theta_3+\theta_4)$ and so on.

4. Trajectory interpolation algorithm of the manipulator with unknown path points

Since the time-optimal trajectory planning of the manipulator proposed in this paper includes the operation of searching the optimal path simultaneously, which needs a flexible and variable trajectory interpolation algorithm to fit the path of the manipulator.

The commonly used interpolation algorithms of manipulator trajectory include polynomial interpolation, spline interpolation, B-spline interpolation, etc. This paper uses the B-spline interpolation algorithms to fit the general expression of the manipulator joint motion curve with unknown path points because of its good continuity and smoothness.

4.1. B-spline curve

The B-spline curve is a generalization of the Bessel curve, which is a smooth curve fitted by a series of control points $P(P_1, P_2, \ldots, P_n)$ [16], and the general expression of k-order B-spline function curve is as follows.

$$
p(u) = \sum_{i=1}^{n} P_i N_{i,k}(u) \tag{4}
$$

Where P_i is the coordinate of the control point, and $N_{i,k}(U)$ is the k-order basis function corresponding to the control point P_i, which is defined as:

$$
\begin{cases}
N_{i,0}(u) = \begin{cases} 1, & U_i \le u < U_{i+1} \\ 0, & else \end{cases} \quad ,k=0\\
N_{i,k}(u) = \dfrac{u-U_i}{U_{i+k}-U_i}N_{i,k-1}(u) + \dfrac{U_{i+k+1}-u}{U_{i+k+1}-U_{i+1}}N_{i+1,k-1}(u), k = 1,2,\ldots
\end{cases}
\tag{5}
$$

$U(\underline{U}_1, U_2, \ldots, U_m)$ is the node sequence and strictly abide by $m = n+k+1$. The B-spline curves of different orders fitted with the same control points are shown in Fig. 2. Obviously, the greater the order k is, the more control points each curve segment participates in fitting.

4.2. Trajectory interpolation algorithm based on B-spline

This paper uses a cubic uniform B-spline curve to fit the trajectory of the manipulator, which ensures that the angular jerk of the manipulator's joints is controllable and avoids the Runge phenomenon due to the higher-order function. The node sequence is an arithmetic sequence, and the practical region of the fitting curve is in $[U_4, U_{n+1}]$, whose

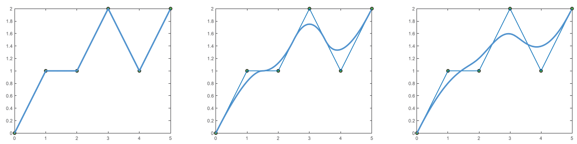

Fig. 2. 1-order B-spline curve, 2-order B-spline curve, and 3-order B-spline curve.

function expression can be calculated by the four control points P_i, P_{i+1}, P_{i+2}, P_{i+3} [17].

$$p_i(u) = P_i N_{i,3}(u) + P_{i+1} N_{i+1,3}(u) + P_{i+2} N_{i+2,3}(u) + P_{i+3} N_{i+3,3}(u) \tag{6}$$

Normalize the parameter u as:

$$u = (u - U_i)/(U_{i+1} - U_i), i = 4, 5, \dots, n \tag{7}$$

And the basic functions $N_{i,3}(u)$, $N_{i+1,3}(u)$, $N_{i+2,3}(u)$ can be simplified as follows.

$$\begin{cases} N_{i,3}(u) = \frac{1}{6}(-u^3 + 3u^2 - 3u + 1) \\ N_{i+1,3}(u) = \frac{1}{6}(3u^3 - 6u^2 + 4) \\ N_{i+2,3}(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1) \\ N_{i+3,3}(u) = \frac{1}{6}u^3 \end{cases} u \in [0, 1] \tag{8}$$

Paper [18] proves that the cubic spline curve with seven segments between arbitrary two points can fully describe the time-optimal trajectory of the manipulator and proposes that the cubic B-spline curve has a similar property, too. Therefore, in this paper, the point-to-point trajectory of the manipulator is also divided into seven segments, and there are ten control points needed. The control points $P(P_1, P_2, \dots, P_{10})$ can be calculated as follows when the path points $V(V_1, V_2, \dots, V_8)$ are known.

$$\begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ \vdots \\ P_9 \\ P_{10} \end{bmatrix} = 6 \begin{bmatrix} 6 & -6 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 4 & 1 \\ & & & & -6 & 6 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ V_1 \\ V_2 \\ \vdots \\ V_8 \\ 0 \end{bmatrix} \tag{9}$$

Both control points P and path points V contain joint angle information Q and motion time information T of the manipulator. So, the horizontal coordinate $t_i(u)$ and vertical coordinate $q_i(u)$ of the trajectory of the ith segment can be calculated by the formulas as follows.

$$t_i(u) = \frac{1}{6}\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} T_i \\ T_{i+1} \\ T_{i+2} \\ T_{i+3} \end{bmatrix} \tag{10}$$

$$q_i(u) = \frac{1}{6}\begin{bmatrix} u^3 & u^2 & u & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix} \begin{bmatrix} Q_i \\ Q_{i+1} \\ Q_{i+2} \\ Q_{i+3} \end{bmatrix} \tag{11}$$

4.3. Angular motion expression of manipulator's joint

Since the horizontal and vertical coordinates of each joint's angular motion curve are both parametric equations about the time parameter u, according to derivation rules of the functions with parameters, the joint's angular velocity at any time t is:

$$V = \frac{dq}{dt} = \frac{q'}{t'} \tag{12}$$

The expression of the joint's angular acceleration is:

$$A = \frac{dq}{dt^2} = \frac{q''t' - q't''}{t'^3} \tag{13}$$

And the expression of the joint's angular jerk is:

$$J = \frac{dq}{dt^3} = \frac{q'''t'^2 - q't't''' + 3q't''^2 - 3q''t't''}{t'^5} \tag{14}$$

Where $q'=dq/du$, $t'=dt/du$, $q'' = dq'/du$, $t'' = dt'/du$, $q''' = dq''/du$.

5. Path points and motion time optimization algorithm of the manipulator

Based on the research results of the manipulator's trajectory interpolation algorithm in the last chapter, the complete motion trajectory of the manipulator is easy to obtain when the model gives the condition of path points. Therefore, this paper uses an improved genetic algorithm to search the optimal path point containing the joint angle nodes and the motion time nodes. The genetic algorithm is improved to adapt to the application scenarios of this paper from many aspects, like gene coding, fitness function, crossover, and variation.

5.1. Gene coding

The genetic algorithm encodes the joint angle nodes and the motion time nodes of the manipulator trajectory simultaneously to search the optimal path while realizing the time-optimal trajectory planning of the manipulator. Because the state of the manipulator determines the first and the last joint angle nodes, there are six groups of joint angle nodes and seven motion time nodes needed to search by the algorithm.

The angle nodes of different joints are determined by the same ratio, inspired by the polynomial interpolation, to reduce the gene length and improve the smoothness and coordination of the motion trajectory. Therefore, in gene coding, six proportional coefficients and seven motion time nodes are uniformly coded by 10-bit binary, and each individual's gene is composed of a 130-bit binary code.

5.2. Fitness function with penalty mechanism

Trajectory planning needs to meet the inherent kinematic constraints of the manipulator. This paper adds a penalty mechanism in the calculation of individual fitness function to constrain the joint angular velocity, angular acceleration, and angular jerk of the manipulator [19]. The fitness function is expressed as follows.

$$Fit = \frac{K}{K_t \sum_{i=1}^{7} t_i + \sum_{j=1}^{6} E_j} \tag{15}$$

Where K, K_t are coefficients, t_i is the running time of each segment trajectory, and E_j is the penalty term designed for each joint according to the motion constraints of the manipulator, whose calculation formula is:

$$E_j = \mathrm{K}_V Max(\left|V_j\right|^{\max}, V_{js}) + \mathrm{K}_A Max(\left|A_j\right|^{\max}, A_{js}) + \mathrm{K}_J Max(\left|J_j\right|^{\max}, J_{js}) \tag{16}$$

K_V, K_A, K_J are the penalty coefficients of joint angular velocity, angular acceleration, and angular jerk respectively, and V_{js}, A_{js}, J_{js} are the critical values of them. The function *Max*$(|V_j|^{\max}, V_{js})$ takes $|V_j|^{\max}$ when $|V_j|^{\max} > V_{js}$, and takes 0 when $|V_j|^{\max} < V_{js}$. For the individual who does not meet the kinematic constraints of the manipulator, the fitness value and the survival probability are significantly reduced.

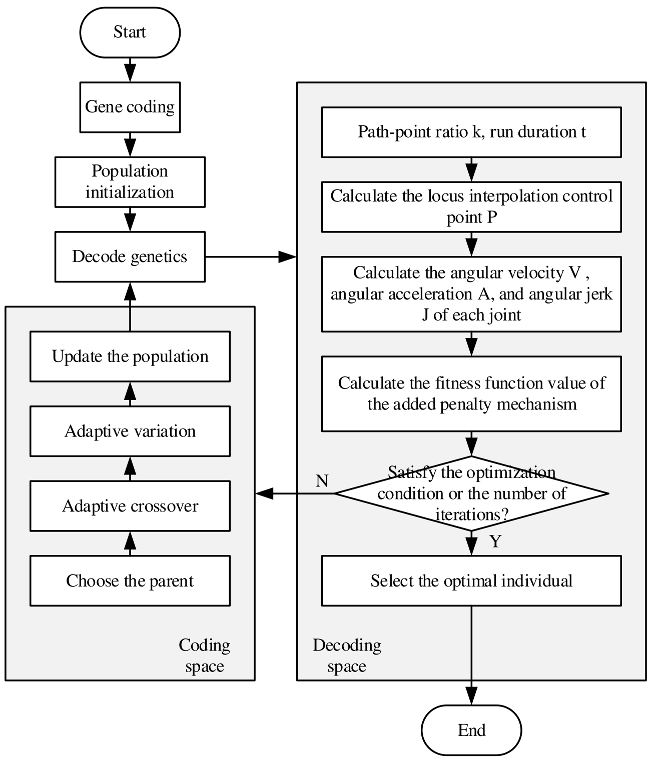

Fig. 3. Algorithm flow of time-optimal trajectory planning of manipulator with simultaneously searching optimal path.

5.3. Adaptive crossover and mutation

This paper sets that the crossover and mutation probabilities decrease adaptively with the iteration to achieve a faster convergence speed. In the early iteration, the larger crossover and mutation probabilities can ensure the population diversity, but in the later iteration, the smaller crossover and mutation probabilities can accelerate the convergence speed with retaining the optimal solution. The crossover and mutation probabilities in this paper are:

$$P_m = \frac{1000-d}{1000} PM, P_c = \frac{1000-d}{1000} PC \tag{17}$$

Where PM and PC are the basic probabilities of crossover and mutation, and d is the number of iterations of the current algorithm.

5.4. Overall algorithm flow

The flow chart of the time-optimal trajectory planning of the manipulator with simultaneously searching the optimal path at the same time is shown in Fig. 3. The algorithm fits the complete trajectory of joint angle by every individual gene in the decoding space and calculates their fitness function values. According to the fitness function value, the algorithm decides whether to continue to optimize. In the coding space, the population is updated adaptively by the genetic algorithm until the optimization conditions or the maximum number of iterations are satisfied.

6. Experiments and analyses

6.1. Simulation modeling of the manipulator

This paper takes the AUBO-I5 manipulator as the experimental platform, and the robotic toolbox in MATLAB is used to simulate it. According to the improved D–H method, the D–H parameters of the AUBO-I5 manipulator are shown in Table 1, and Fig. 4 shows the simulation model of the AUBO-I5 manipulator drawn with robotics toolbox.

Table 1
D–H parameters of AUBO-I5 manipulator.

Rod j	Joint angle θ_j(mm)	Offset d_j(mm)	Length a_{j-1}(mm)	Twist angle α_{j-1}(mm)
1	0	121.5	0	0
2	90	121.5	0	90
3	0	0	408	0
4	−90	0	376	0
5	0	102.5	0	−90
6	0	94	0	90

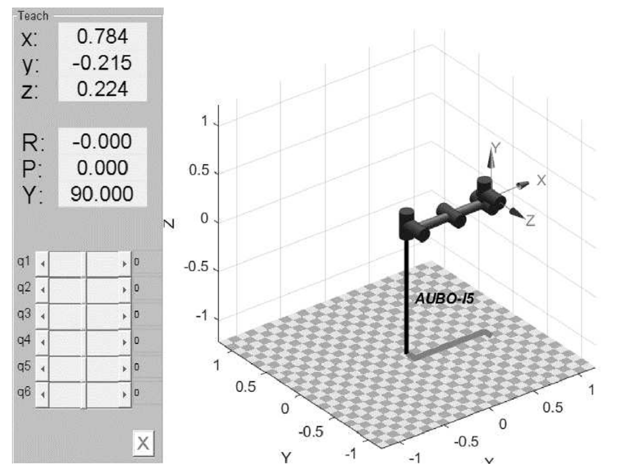

Fig. 4. Simulation model of AUBO-I5 manipulator drawn with robotics toolbox.

Table 2
Constraint conditions of manipulator joint.

Joint j	1	2	3	4	5	6
V_{js} (°/s)	150	150	150	180	180	180
A_{js} (°/s^2)	70	70	70	80	80	80
J_{js} (°/s^3)	100	100	100	120	120	120

6.2. Algorithm simulation experiment

For the trajectory interpolation of the manipulator with unknown path points, the path points V_1, V_8 are determined by the initial position and termination position of the manipulator. Based on the control variable method, the following simulation experiments are set to:

$$\begin{aligned} V_1 = q_0 &= \begin{bmatrix} 0.5236 & 0 & 2.0944 & 1.0472 & 1.5708 & 1.0472 \end{bmatrix} \text{(rad)} \\ V_8 = q_f &= \begin{bmatrix} 0 & 1.5708 & 0 & -1.5708 & 0 & 0 \end{bmatrix} \text{(rad)} \end{aligned} \tag{18}$$

The population size of genetic algorithm is set to 100, and the maximum iteration times is 100. In the design of fitness function, this paper sets the coefficients $K = 100$, $K_t = 100$ in formula (14), and the penalty coefficients of joint angular velocity, angular acceleration, and angular acceleration in E_j are set as $K_V = 10$, $K_A = 10$, $K_J = 1$, the probability of basic crossover variation is set as $PM = 0.6$, $PC = 0.01$. The kinematic constraints of the manipulator joints are shown in Table 2.

The optimal result of the simulation experiment with the parameter settings above can be decoded to the proportion coefficient $K_2 \sim K_7$ of the joints angle and the motion time nodes $T_1 \sim T_7$, and the control points is shown in Table 3.

As shown in the table above, the optimal trajectory planning time is 3.7341 s. The optimal trajectory curve, angular displacement curve, angular velocity curve, and angular acceleration curve, as shown in

Table 3
Time-optimal path point distribution and time interval.

V_i (rad)	Joint j						t_i (s)
	1	2	3	4	5	6	
1	0.52	0	2.09	1.05	1.57	1.05	0
2	0.37	0.47	1.47	0.27	1.10	0.74	0.64
3	0.18	1.03	0.73	−0.66	0.55	0.36	1.42
4	0.08	1.32	0.34	−1.15	0.25	0.17	2.03
5	0.04	1.45	0.17	−1.36	0.12	0.08	2.47
6	0.01	1.53	0.05	−1.50	0.04	0.03	2.89
7	0.00	1.56	0.01	−1.56	0.00	0.00	3.28
8	0	1.57	0	−1.57	0	0	3.73

Fig. 5, are obtained by calculating each joint's transverse and longitudinal coordinates under the optimal trajectory. The experimental results show that the trajectory fitted by the time-optimal manipulator trajectory planning algorithm designed in this paper is smooth, the motion curve has no apparent mutation, the maximum angular velocity, angular acceleration, and angular jerk are within the range of constraints.

6.3. Comparative verification experiment

Besides, this paper carries out some comparative experiments to verify that the time-optimal trajectory planning proposed in this paper has better performance than other similar algorithms. The comparative experiments are under the same task by different time-optimal trajectory planning, including the B-spline interpolation time-optimal trajectory planning algorithm with uniform path and random path, and based on the cubic polynomial interpolation and five interpolations polynomial time-optimal trajectory planning algorithm. The results of the comparative experiments are shown in Table 4.

Fig. 6 shows the angular displacement curves of the comparative experiments, Fig. 7 shows their angular velocity curves, and Fig. 8 shows the angular acceleration curves. Where the curves of the solid line represent the experimental result of the time-optimal trajectory planning proposed in this paper, the curves of the long-dotted line represent the experimental result of Experiment 2, the curves of the fine-dotted line represent the experimental result of Experiment 3, and the curves of the coarse-dotted line represent the experimental result of Experiment 4 and 5. The results of the above five groups of comparative experiments all meet the motion constraints of the AUBO-I5 manipulator.

In several comparative experiments, the proposed algorithm achieves satisfactory results. Among them, the angular acceleration of trajectory-based on cubic polynomial interpolation is uncontrollable. The response speed of trajectory-based on quintic polynomial interpolation is slow. The average velocity and average acceleration of trajectory-based on B-spline interpolation with uniform path points are low. And the trajectory based on B-spline interpolation with random path points has the worst performance.

The comparative experimental results show that the time-optimal trajectory planning algorithm designed in this paper has the ability to search for the global optimal trajectory. Compared with other time-optimal trajectory planning algorithms, the response of angular displacement and angular velocity of the manipulator is faster, and the change of angular acceleration is smaller, which conforms to the optimization standard of trajectory planning.

7. Conclusions

This paper proposes a time-optimal trajectory planning of manipulator with simultaneously searching optimal path to address the problem that the mainstream methods always search for the optimal motion time under a fixed path, but do not discuss whether this path is optimal. Firstly, the technique uses a cubic uniform B-spline curve to fit the

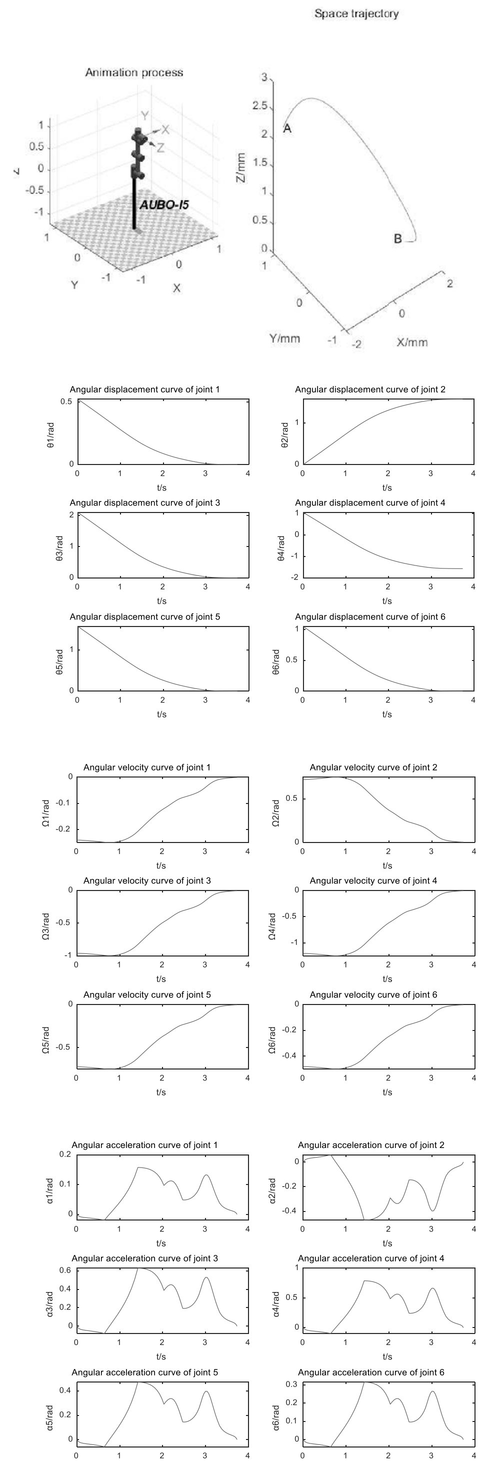

Fig. 5. Trajectory curve, angular displacement curve, angular velocity curve, and angular acceleration curve of each joint.

general expression of the manipulator joint motion curve with unknown path points. Then, an improved genetic algorithm is used to search the optimal path points and the optimal movement time simultaneously, where the fitness function adds penalty term to constrain the angular velocity, angular acceleration, and angular acceleration of the manipulator. Finally, this paper carries some simulation experiments on the AUBO-I5 manipulator model and proves that the proposed algorithm is effective and has a shorter motion time than other similar algorithms.

Table 4
Results of multi *group* comparative experiments.

Experiment serial number	Trajectory interpolation algorithm	Path points selection	Motion time (s)
1	B-spline	Genetic algorithm	3.7341
2	B-spline	Uniform selection	3.9238
3	B-spline	Random selection	5.3460
4	Cubic polynomial	–	3.9485
5	Quintic polynomial	–	3.7954

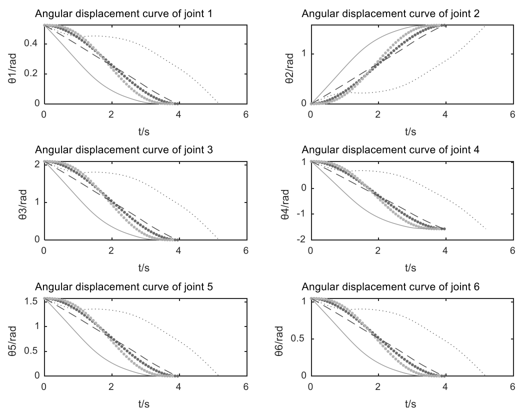

Fig. 6. Angular displacement curve of each joint in comparative test.

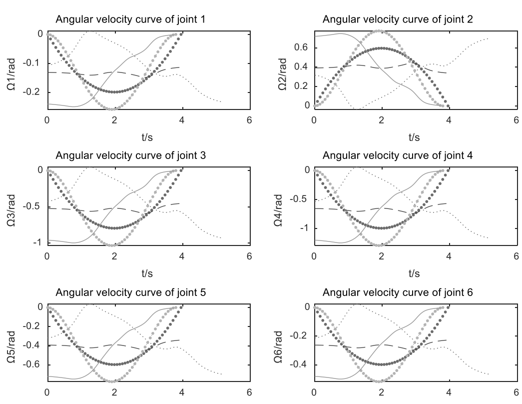

Fig. 7. Angular velocity curve of each joint in comparative test.

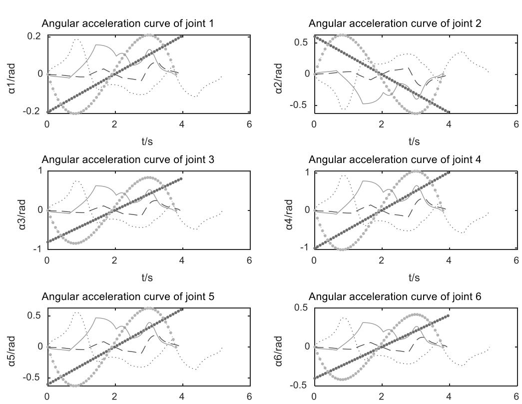

Fig. 8. Angular acceleration curve of each joint in comparative test.

The main contribution of this paper is to propose a path variable time-optimal trajectory planning algorithm, which is different from the mainstream algorithms which search for the optimal motion time under the fixed path. While searching for the optimal motion time, the proposed method also continuously optimizes the motion path of the manipulator.

However, the proposed algorithm still has some limitations. Firstly, it is difficult to verify whether the optimal motion trajectory obtained by the algorithm is the global optimal solution. Secondly, the fitness function of the algorithm has a lot of room to improve. Therefore, in future work, it is necessary to design a more comprehensive and scientific evaluation index to evaluate the motion path of the manipulator. In addition, aiming at the complex problem of hybrid optimization of the manipulator's motion time and motion path, we will try to design a more useful fitness function combined with a multi-objective genetic algorithm to select the optimal motion trajectory of the manipulator from the Pareto solution set.

CRediT authorship contribution statement

Xiuli Yu: Conception or design of this paper. **Mingshuai Dong:** Conception or design of this paper. **Weimin Yin:** Conception or design of this paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The funding source is: Fundamental Research Funds for the Central Universities (Grant No. 2020XD-A04-2).

References

[1] A. Nagy, I. Vajk, Nonconvex time-optimal trajectory planning for robot manipulators, J. Dyn. Sys. Meas. Control 141 (11) (2019).
[2] C. Curtis, T. Lianfang, An effective robot trajectory planning method using a genetic algorithm, Mechatronics: Sci. Intell. Mach. 14 (5) (2004) 455–470.
[3] Y. Tingdong, C. Taobo, Z. Songbing, Trajectory planning method of robot teaching based on multi-knots spline interpolation, Adv. Commun. Technol. Syst. (2015) 399–406.
[4] W. Wenjie, T. Qing, C. Yuting, et al., Robot time-optimal trajectory planning based on improved cuckoo search algorithm, IEEE Access 8 (2020) 86923–86933.
[5] R. Roy, M. Mahadevappa, C.S. Kumar, Trajectory path planning of EEG controlled robotic arm using GA, Procedia Comput. Sci. 84 (2016) 147–151.
[6] A. Hj, B. Abi, D. Akc, et al., Quantum and classical genetic algorithms for multilevel segmentation of medical images: A comparative study, Comput. Commun. 162 (2020) 83–93.
[7] F.Z. Baghli, L.E. bakkali, Y. Lakhal, Optimization of arm manipulator trajectory planning in the presence of obstacles by ant colony algorithm, Procedia Eng. 181 (2017) 560–567.
[8] J.J. Kim, J.J. Lee, Trajectory optimization with particle swarm optimization for manipulator motion planning, IEEE Trans. Ind. Inf. 11 (3) (2015) 620–631.
[9] N. Ai, B. Wu, B. Li, et al., 5G heterogeneous network selection and resource allocation optimization based on cuckoo search algorithm, Comput. Commun. 168 (2021) P2.
[10] Elias K. Xidias, Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces, Robot. Comput.-Integr. Manuf. 50 (2018) 286–298.
[11] Y. Liu, C. Guo, Y. Weng, Online time-optimal trajectory planning for robotic manipulators using adaptive elite genetic algorithm with singularity avoidance, IEEE Access 7 (2019) 146301–146308.
[12] Y. Fengjian, L. Qinghua, C. Xu, et al., Research on mechanical arm joint space trajectory planning algorithm based on optimal time, Mach. Des. Res. 33 (05) (2017) 12–15.

[13] D. Wei, Z. Qiwan, L. Ping, et al., Optimal time trajectory planning based on dual population genetic and chaotic optimization algorithm, Comput. Integr. Manuf. Syst. 24 (1) (2018) 101–106.

[14] Q. Lv, W. Mulan, Z. Xiaochun, et al., The time-optimal trajectory planning of robot based on improved genetic algorithm, Mach. Des. Manuf. Eng. 47 (11) (2018) 63–67.

[15] N.A. Aspragathos, J.K. Dimitros, A comparative study of three methods for robot kinematics, IEEE Trans. Syst. Man. Cybern. B. Cybern. 28 (2) (1998) 135–145.

[16] H. Junsen, H. Pengfei, W. Kaiyuan, et al., Optimal time-jerk trajectory planning for industrial robots, Mech. Mach. Theory 121 (2018) 530–544.

[17] R. Kumar, A. Choudhary, S. Baskar, Modified cubic B-spline quasi-interpolation numerical scheme for hyperbolic conservation laws, Appl. Anal. 99 (1) (2020) 158–179.

[18] L. Ruifeng, Research on Minimum—Time Traj Ectory Planning for Robot Point to Point Task (MA thesis), Dalian University of Technolog, 2016.

[19] L. Huashan, L. Xiaobo, W. Wenxiang, Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints, Robot. Comput.-Integr. Manuf. 29 (2) (2013) 309–317.