L. Trajkovic et al. (Eds.)

doi:10.3233/ATDE240480

Research on Trajectory Planning of Industrial Robot Based on MATLAB

Chenxu DUAN[a,1], Yifan WANG[a] and Saru KUMARI[b]
[a] *School of Mechanical Engineering, Sichuan University Jinjiang College, Meishan 620680, China*
[b] *Department of Mathematics, Ch. Charan Singh University, Meerut, India*

Abstract. With the progress of industrial automation, industrial robots play an increasingly important role in the upgrading of the manufacturing industry. Trajectory planning is the key technology of precise motion control of industrial robots. At present, most industrial robots use the method of teaching programming for trajectory planning, which has the problems of low efficiency, long time-consuming and poor accuracy. Simulation programming overcomes many shortcomings of teaching programming. It uses information technology to program, debug and run in simulation scenarios, which reduces time costs and improves planning efficiency and accuracy. It has attracted more and more attention from manufacturing enterprises. In this paper, the trajectory planning process based on MATLAB platform is studied by taking GSK RB08A3 6-axis industrial robot as the research object. Firstly, the coordinate system of each link of RB08A3 robot is established according to the standard DH parameter method, and the mathematical model is built according to the parameters of the robot. The forward and inverse kinematics verification is completed by using the functions in the Robotics Toolbox of MATLAB, which further proves the correctness of the modeling. Then, the multi-trajectory planning problem of industrial robots in Cartesian space is studied. A continuous multi-path smooth transition algorithm is designed based on the improvement of S-shaped speed curve, and the simulation experiment is carried out on the MATLAB platform compared with the traditional S-shaped acceleration and deceleration algorithm with zero head and end speed. The results show that the execution efficiency of the algorithm is greatly improved compared with the traditional S-shaped speed curve trajectory planning, and the effect of smooth transition of adjacent trajectories is achieved, and the mechanical impact is also reduced. Finally, aiming at the problems of working efficiency, energy consumption and flexible impact during start-stop of industrial robots, a time-energy optimal 5-3-5 piecewise interpolation joint space trajectory planning method is proposed. The simulation results show that the optimized piecewise interpolation trajectory can effectively avoid the flexible impact during start-stop, greatly improve the working efficiency of industrial robots, greatly reduce energy consumption, and achieve the comprehensive optimization of time-energy consumption, which verifies the effectiveness of the trajectory planning method.

Keywords. Industrial robots, trajectory planning, DH parameter method, robot kinematics, MATLAB

[1] Corresponding Author: Chenxu DUAN, duanchenxu@scujj.edu.cn.

1. Introduction

Industrial robots are multi-degree-of-freedom manipulators or automated machinery and equipment widely used in industrial production and processing scenarios such as construction, automobile and electronics. They are the product of multi-domain and multi-direction cross-integration. According to the different use scenarios of industrial robots, the requirements for terminal displacement, speed and acceleration are also different, so it is necessary to plan the trajectory of the robot to meet the needs of different production activities [1]. In the case of considering the task and the performance of the robot, the trajectory planning gives the expected trajectory of the robot actuator, that is, on the basis of kinematics and dynamics, using the appropriate joint space and Cartesian space trajectory planning method, and solving the robot motion with time, and finally generating the trajectory [2]. The trajectory planning first gives the pose of each teaching point, then calculates the intermediate point position information according to different trajectory interpolation methods, and then obtains the joint angle of the robot from the inverse kinematics solution, and then passes it to the robot controller to realize the joint position control, and finally makes the end reach the target pose [3].

For Cartesian space trajectory planning methods, such as commonly used linear and circular trajectory interpolation, trapezoidal and s-shaped velocity curve planning, etc., are the object of continuous research by many scholars. Wu et al. designed a flexible acceleration and deceleration algorithm based on trigonometric function square sine curve, and finally obtained a sine curve with continuous displacement, velocity, acceleration and acceleration [4]. Laha et al. studied the Cartesian space trajectory planning method based on the homogeneous transformation matrix, and divided the trajectory into translation and rotation for pose interpolation, which overcomes the original shortcomings of linear and circular trajectory planning and has strong operability [5]. Chen et al. used Cartesian space trajectory planning for the spraying robot, integrating the idea of parabolic transition and local coordinate system. The simulation results show that the trajectory is smooth and the effect is very impressive [6]. Wang et al. proposed an asymmetric S-type velocity trajectory planning method, which introduced the arc length increment interpolation technology to achieve smooth transition of path transition. The experimental results show that this method can improve the working efficiency of the robot and reduce the sudden change of angular velocity [7].

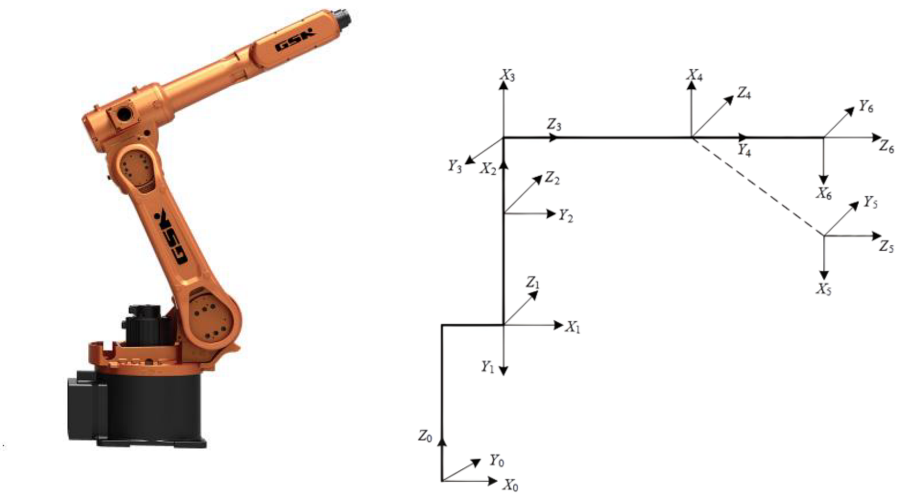
The research on trajectory planning in joint space is relatively more and more comprehensive. The common methods are simple polynomial interpolation, B-spline curve with good envelope and more complex NURBS curve. However, in the process of a large number of practical applications, it can be found that the trajectory planning scheme can not only focus on the motion effect, but also on the planning efficiency, energy consumption and stability. It is more practical for the project to find the optimal trajectory planning scheme that meets the actual production requirements under each working condition. Many scholars have carried out research in this area. Chen et al. [8] used the quantum-behaved particle swarm optimization algorithm to find the optimal trajectory of the joint motion trajectory of the quintic non-uniform B-spline curve. The results show that the efficiency is higher than the differential evolution algorithm and the standard particle swarm optimization algorithm. Nazarahari et al. [9] used the improved genetic algorithm to study the energy optimal trajectory planning problem, and used the kinetic energy of each joint as the energy function to search the optimal energy trajectory. Luan et al. [10] used genetic algorithm to study the hybrid optimal trajectory planning. First, the quintic uniform B-spline was used to interpolate the time nodes, and the hybrid genetic algorithm was used to carry out the time optimal trajectory planning, and the minimum task time was obtained. Then, based on the adaptive genetic algorithm, the optimal trajectory of the impact was searched, and the service life and work efficiency of the robot were ensured. However, it can be found that most scholars have carried out a lot of research on the three aspects of time optimization, energy optimization and impact optimization, and the length of the path, the size of the torque and other factors are equally important, and the actual working mode of the robot in the optimal trajectory planning cannot be ignored.

In summary, Cartesian space planning and joint space trajectory planning methods have their own advantages and disadvantages, and the trajectory interpolation methods contained in the two are also numerous. In the process of industrial production, the use of Cartesian space or joint space trajectory planning depends on the specific situation, and in practical application, it is also necessary to take into account factors such as time, energy or stability to find the appropriate optimal trajectory planning scheme.

2. Industrial Robot Modeling and Kinematics Simulation

2.1. Robot Model

The research object of this paper is GSK RB08A3 6-axis industrial robot. The robot is suitable for many industrial scenarios such as polishing, spraying, handling and loading and unloading of machine tools, and meets the Pieper criterion. To establish the model, the connecting rod coordinate system needs to be determined first. The three-dimensional model of the robot and the schematic diagram of the connecting rod structure are shown in Figure 1.



a) Three- dimensional model of robot b) Connecting rod coordinate system of robot

Figure 1. RB08A3 industrial robot.

The robot has six linkage coordinate systems at each position of the connecting rod. The DH parameters of the connecting rod are shown in Table 1. The spatial structure of the industrial robot can be determined by this parameter, and a mathematical model is established to describe the relationship between the connecting rods.

Table 1. DH parameters of RB08A3 industrial robot

i	$\boldsymbol{a_i(mm)}$	$\boldsymbol{\alpha_i(°)}$	$\boldsymbol{d_i(mm)}$	$\boldsymbol{\theta_i(°)}$
1	100	-90	615	θ_1
2	705	0	0	θ_2
3	135	-90	0	θ_3
4	0	90	755	θ_4
5	0	90	0	θ_5
6	0	0	85	θ_6

2.2 Forward Kinematics Analysis

The process of solving forward kinematics is essentially a homogeneous matrix multiplication operation. After knowing the value of each joint angle of the robot, the value of the transformation matrix T_i^{i-1} of the end relative to the robot base can be obtained according to the homogeneous transformation matrix formula and the DH parameters of the industrial robot [11]. According to the coordinate transformation method mentioned above, the transformation of the end of the industrial robot relative to the base is as follows: if T_6^0 is required, only the six homogeneous transformation matrices are multiplied right in turn, the transformation of the end of the industrial robot relative to the base is as follows:

$$T_6^0 = T_1^0 T_2^1 T_3^2 T_4^3 T_5^4 T_6^5 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

In the formula:

$$n_x = c_1[c_{23}(c_4c_5c_6 - s_4s_5) - s_{23}s_5c_5] + s_1(c_4c_5c_6 - c_4s_6) \tag{2}$$

$$n_y = s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6) \tag{3}$$

$$n_z = -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5c_6 \tag{4}$$

$$o_x = c_1[c_{23}(-c_4c_5c_6 - s_4c_6) + s_{23}s_5s_6] - c_1(-s_4c_5s_6 + c_4s_6) \tag{5}$$

$$o_y = s_1c_{23}(s_4c_6 - c_4c_5s_6) - c_1(s_4c_5s_6 + c_4c_6) + c_1s_{23}s_5s_6 \tag{6}$$

$$o_z = -s_{23}(-c_4c_5s_6 - s_4c_6) + c_{23}s_5s_6 \tag{7}$$

$$a_x = c_1(c_{23}c_4s_5 + s_{23}c_5) + s_1s_4s_5 \tag{8}$$

$$a_y = s_1(c_{23}c_4s_5 + s_{23}c_5) - c_1s_4s_5 \tag{9}$$

$$a_z = s_{23}c_4s_5 - c_{23}c_5 \tag{10}$$

$$p_x = c_1[a_2c_2 + a_3c_{23} - s_{23}d_4] - d_3s_1 \tag{11}$$

$$p_y = s_1[a_2c_2 + a_3c_{23} - s_{23}d_4] + d_3c_1 \tag{12}$$

n is called normal vector, o is called direction vector, a is called approach vector. In order to express concisely, the above trigonometric functions are simply abbreviated as:

$$\mathrm{c_i} = \cos\theta_\mathrm{i}\,, \mathrm{s_i} = \sin\theta_\mathrm{i}\,, \mathrm{c_{ij}} = \cos(\theta_\mathrm{i} + \theta_\mathrm{j})\,, \mathrm{s_{ij}} = \sin(\theta_\mathrm{i} + \theta_\mathrm{j}). \tag{13}$$

After substituting the parameters in the DH parameter table, T_6^0 is the total transformation matrix of the RB08A3 industrial robot connecting rod, that is, the pose description of the end effector relative to the base.

2.3. Inverse Kinematics Analysis

The inverse kinematics and forward kinematics are solved in the opposite way. It is necessary to reverse the variable values of each joint according to the end pose, and convert the pose information from Cartesian space to joint space. The inverse kinematics solution is also an important basis for studying the trajectory planning algorithm [12].

The inverse kinematics analysis has a high degree of complexity, and the inverse solution of the multi-degree-of-freedom robot is often not unique. There are multiple solutions or even infinite solutions. Sometimes it is necessary to select the optimal solution according to the actual situation, and it is difficult to directly solve. The method of solving the problem is also varied. Which kind of inverse kinematics algorithm is selected will affect the overall performance of the industrial robot to a certain extent. The closed solution and the numerical solution are the main methods for solving the inverse solution, and the closed solution is more reliable. Among various closed solutions, the geometric method is to find the angle value of each joint by observing the shape of the robot and using geometric calculation. Although it is simple and intuitive, there is no doubt that the human error is large. The algebraic method is relatively fast in calculation speed, high in efficiency and small in error, and has become the mainstream method for many robot scholars to find the inverse solution [13]. Therefore, this paper will use a systematic algebraic solution to carry out inverse kinematics analysis.

When the total transformation matrix T_6^0 is known, θ_i is obtained by solving the following equation.

$$\theta_1 = A\tan 2(m_1, n_1) - A\tan 2\left(d_4, \pm\sqrt{{m_1}^2 + {n_1}^2 - {d_4}^2}\right) \tag{14}$$

$$\theta_2 = A\tan 2(s_2, c_2) \tag{15}$$

$$\theta_3 = \pm\arccos\left(\frac{{m_3}^2 + {n_3}^2 - {a_2}^2 - {a_3}^2}{2a_2a_3}\right), {m_3}^2 + {n_3}^2 \le (a_2 + a_3)^2 \tag{16}$$

$$\theta_4 = A\tan 2\big(-s_6(n_xc_1 + n_ys_1) - c_6(o_xc_1 + o_ys_1), o_zc_6 + n_zs_6\big) - \theta_2 - \theta_3 \tag{17}$$

$$\theta_5 = \pm \arccos(a_x s_1 - a_y c_1) \tag{18}$$

$$\theta_6 = A\tan 2(m_2, n_2) - A\tan 2\left(s_5, \pm\sqrt{{m_2}^2 + {n_2}^2 - s_5^2}\right) \tag{19}$$

From the inverse kinematics solution, it can be seen that some of the joint angle results are not unique. Some solutions may have exceeded the working range of the robot. Some solutions are within the working space, but do not meet the task requirements. Therefore, in the production requirements, the appropriate inverse solution should be selected according to the actual needs.

3. Industrial Robot Cartesian Space Trajectory Planning

3.1. Position and Attitude Interpolation

3.1.1. Spatial linear position interpolation

The general process of linear interpolation in space is to give the coordinates of some points in the task space, and calculate the linear trajectory by the position of these points for interpolation. The two-point coordinates of the known space straight line are:

The starting point $S(x_1, y_1, z_1)$, the end point $D(x_2, y_2, z_2)$.

Then the straight line distance S_e is:

$$S_e = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \tag{20}$$

If the interpolation period is T, the interpolation step can be expressed as:

$$\Delta S_e = v \cdot T \tag{21}$$

3.1.2. Space circular arc position interpolation

Different from space linear interpolation, space circular arc trajectory interpolation is to interpolate a circular arc curve in the task space. As shown in Figure 2, there are three non-collinear points in the known space:

$$P_1(x_1, y_1, z_1), P_2(x_2, y_2, z_2), P_3(x_3, y_3, z_3)$$

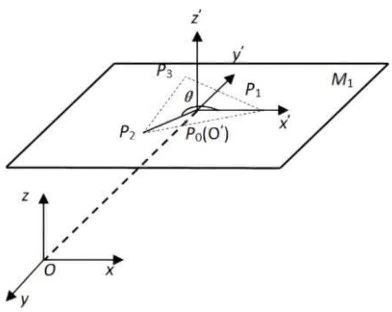


Figure 2. Space circular interpolation coordinate system transformation.

The trajectory planning of the arc determined by these three points is carried out, and the specific steps of position interpolation are as follows:

(1)The coordinate P_0 of the center of the space arc, the radius R of the arc and the plane equation M_1 where the circumcircle is uniquely determined by P_1, P_2 and P_3 are obtained.

(2) Create a new coordinate system $o' - x'y'z'$ for the plane of the arc under the original coordinate system $o - xyz$.

(3) Calculate the angle of the arc. According to the coordinates of P_1, P_2 and P_3 in the new coordinate system, the angle θ in the process of circular trajectory interpolation is calculated:

$$\theta = \begin{cases} \arctan 2(y_3', x_3') + 2\pi, & y_3' < 0 \\ \arctan 2(y_3', x_3'), & y_3' \geq 0 \end{cases} \tag{22}$$

(4) The trajectory planning is carried out on the plane M_1 where the arc is located. According to the homogeneous transformation principle, the following is obtained:

$$\begin{cases} \theta_k = \lambda_k \cdot \theta, \\ x_i' = r \cdot \cos \theta_k, \\ y_i' = r \cdot \sin \theta_k, \\ z_i' = 0, \end{cases} \tag{23}$$

Among them,λ_k is a normalization factor, $\lambda_k \in [0,1]$.

3.1.3. Attitude interpolation based on quaternion

Rotating matrix method, RPY angle method, Euler angle method and quaternion method are several common forms of robot attitude representation. Among them, the rotation matrix method is more complicated, and the Euler angle method and the RPY angle method have singularity problems in the solution process [14]. The unit quaternion has the advantages of small computation, more natural representation of rigid body rotation, and easy interpolation when describing the attitude. It is very effective for the smooth transition between multiple attitudes, and can solve the singularity problem. Therefore, this paper will use the quaternion method to interpolate the attitude.

If the homogeneous transformation matrices T_s and T_d of the starting point S and the end point D are known, the attitude interpolation steps based on quaternion are as follows:

(1) The rotation matrices R_s and R_d are extracted from the homogeneous transformation matrix and converted into quaternions Q_s and Q_d.

(2) The attitude quaternion interpolation formula is:

$$Q(t) = \frac{\sin[(1-t)\theta]}{\sin \theta} Q_s + \frac{\sin(t\theta)}{\sin \theta} Q_d \tag{24}$$

Among them,$\theta = \arccos(Q_s, Q_d)$.

(3) The quaternion $Q(\mathrm{t})$ is transformed into the rotation matrix R_k, R_k and the P_k obtained by position interpolation to obtain the homogeneous transformation matrix T_k corresponding to each interpolation point. After obtaining the position and attitude of

each interpolation point, the angle variables of each joint are obtained through the inverse solution of motion, and the interpolation motion is carried out.

3.2. Design of Smooth Transition Algorithm for Continuous Multi-segment Path

Firstly, the contour trajectory should be preprocessed, that is, how to segment the trajectory and how to connect after segmentation. The motion trajectory of the general industrial robot task space is composed of arc segments and straight segments. For the multi-segment continuous straight line path, it is necessary to smooth the connection of all adjacent straight segments, and establish an arc model at each corner for corner transition.

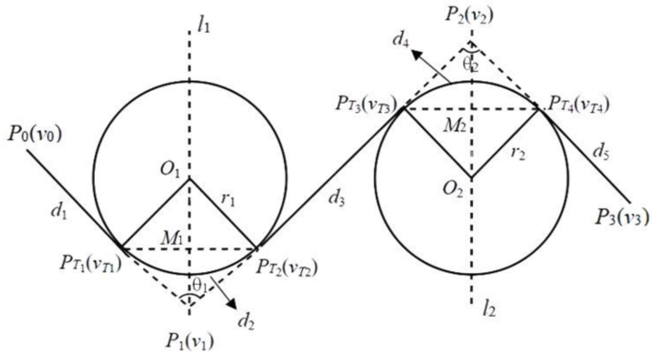


Figure 3. Adjacent trajectory segment connection model.

The connection model of adjacent trajectory segments composed of P_0, P_1, P_2 and P_3 is shown in Figure 3. O_i and r_i are the center and radius of the transition arc, respectively. P_i and v_i are the target point and speed of the pick-and-place path set by the user, respectively. P_{Ti} and v_{Ti} are the arc transition point and the speed of the transition point ; M_i is the midpoint of the two arc transition points ; d_i, θ_i and l_i are the segmented path length, path corner and corner bisector respectively. The design solution steps are as follows:

By solving the corner θ_1 and the transition transition transition points P_{T1} and P_{T2}, the geometric relationship is obtained:

$$\theta_1 = \arccos\frac{\overrightarrow{P_1P_0}\cdot\overrightarrow{P_1P_2}}{|\overrightarrow{P_1P_0}||\overrightarrow{P_1P_2}|} \tag{25}$$

$$\begin{cases} \overrightarrow{P_1P_{T1}} = \frac{r_1/\tan\frac{\theta_1}{2}}{|\overrightarrow{P_1P_0}|}\overrightarrow{P_1P_0} \\ \overrightarrow{P_1P_{T2}} = \frac{r_1/\tan\frac{\theta_1}{2}}{|\overrightarrow{P_1P_2}|}\overrightarrow{P_1P_2} \end{cases} \tag{26}$$

Solving the center O_1 and the path length d_i, M_1 is the midpoint of P_{T1} and P_{T2}.On the angular bisector l_1, it can be seen that:

$$\begin{cases} \overrightarrow{P_{T1}M_1} = \frac{\overrightarrow{P_{T1}P_{T2}}}{2} \\ \overrightarrow{P_1O_1} = \frac{|\overrightarrow{P_1O_1}|}{|\overrightarrow{P_1M_1}|}\overrightarrow{P_1M_1} \end{cases} \tag{27}$$

In $\triangle P_1P_{T1}O_1$, according to the trigonometric function relationship:

$$|\overrightarrow{P_1O_1}| = \frac{r_1}{\sin(\theta_1/2)} \tag{28}$$

According to the formulas (27) and (28), the coordinates of the center O_1 and M_1 can be obtained.

Through the above steps, the critical path points and related parameters can be obtained:

$$\begin{cases} d_1 = |\overrightarrow{P_0P_{T1}}| \\ d_2 = (\pi - \theta_1)r_1 \end{cases} \tag{29}$$

The remaining path points can be extrapolated.

3.3. Trajectory Planning Simulation

3.3.1. Linear interpolation simulation

In order to verify the effect of the trajectory planning algorithm, the Robotic Toolbox is modeled in MATLAB, and the coordinates of the initial point L_0 of the straight line are set as: $(370,127,402)$, and the coordinates of the end point L_1 are set as: $(354,140,410)$.The speed of the initial point and the end point of the system is 0, the interpolation speed is 30 mm / s, and the interpolation period is 0.002 s. The end displacement trajectory is shown in Figure 4.

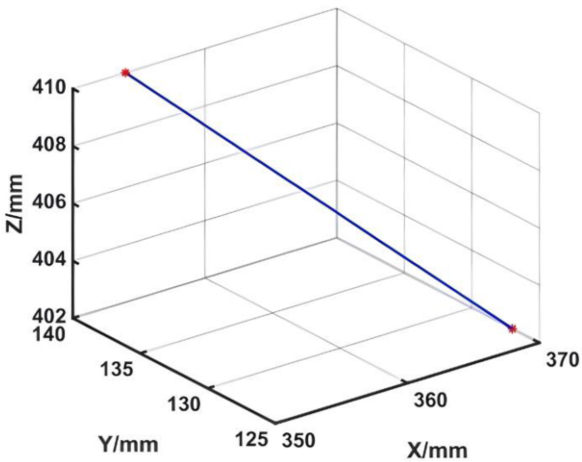


Figure 4. The trajectory of linear interpolation motion.

3.3.2. Arc interpolation simulation

In order to verify the effect of the trajectory planning algorithm, the Robotic Toolbox is used to model in MATLAB, and three points of non-collinear space are selected. The three-point coordinates are set as: starting pointP_0= (325,410,650), passing point P_1 = (250,320,400), and end point P_2 = (415,320,550). The motion parameters of the system are the same as those of linear interpolation. The end displacement trajectory obtained by simulation is shown in Figure 5.

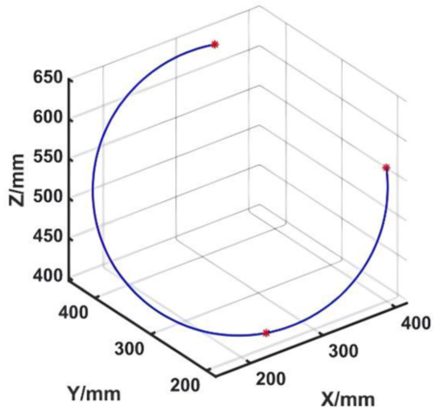


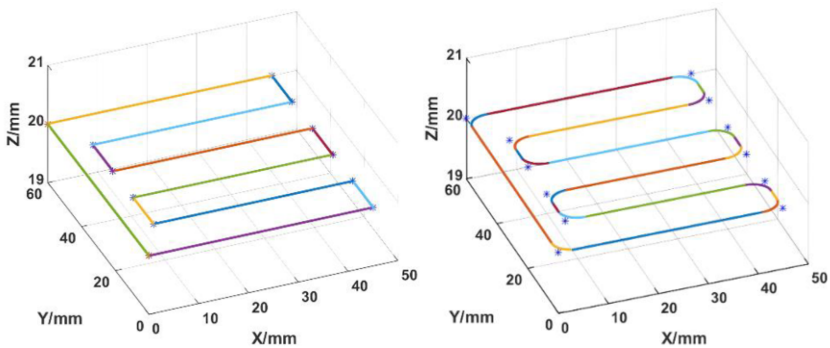
Figure 5. The trajectory of arc interpolation motion.

3.3.3. Multi-segment trajectory transition simulation

In the s-shaped or trapezoidal curve, when there is a continuous multi-segment trajectory, the most conventional method is to subdivide the path into multiple segments, and use the acceleration and deceleration algorithm with the initial and final speed of 0 to plan. Although this method is simple, it has many disadvantages. Next, two algorithms are used for experiments, and the interpolation algorithm designed in this chapter is compared with the trajectory map of the traditional acceleration and deceleration algorithm.

The target path selected by the experiment is composed of 12 trajectory points to simulate the trajectory of the industrial robot. The interpolation period is set to 0.002s, $a_{\max}$ = $500\mathrm{mm}/s^2$, $J_{\max}$ = $2000\mathrm{mm}/s^3$during the operation of the system. The simulated end displacement trajectory is shown in Figure 6.

From the trajectory comparison of the above figure, it can be seen that the traditional acceleration and deceleration algorithm has no transition of the segmented rotation angle and the contour curve is not smooth, while the interpolation algorithm establishes the arc curve transition at the 12 corners of the path, and the planned trajectory is smoother and more beautiful. And can modify the radius adjustment parameters according to the actual needs to achieve the desired effect.



a) The trajectory of traditional acceleration and deceleration algorithm

b) Trajectory of adjacent trajectory point interpolation algorithm

Figure 6. The trajectory of continuous motion

4. Industrial Robot Joint Space Trajectory Planning

Joint space trajectory planning is a time function for planning joint variables such as angular velocity and angular acceleration. In order to avoid mechanical impact, it is necessary to ensure the second-order continuity of the planned joint trajectory and time derivative in the production process. The trajectory planning algorithms used in the joint space are polynomial curves, spline curves and B-spline curves. The use of high-order polynomial or high-order spline interpolation can meet the requirements of smooth and continuous joint acceleration, but the process is cumbersome and the amount of calculation is large. And when the number is high, it is easy to appear Runge phenomenon, increasing the burden on the system. B-spline is widely used because of its excellent characteristics such as local support and derivative continuity. Reasonable use of B-spline interpolation can obtain similar or even better results than high-order polynomial and spline interpolation [15]. In this paper, a 5-3-5 piecewise mixed B-spline interpolation method is adopted. The middle section adopts cubic non-uniform B-spline interpolation, and the first and end ends adopt quintic polynomial interpolation. This not only retains the excellent characteristics of B-spline, but also solves the problem of flexible impact when starting and stopping.

4.1. Blending B-spline Curve Construction

(1) Non-uniform b-spline

The uniform B-spline is only suitable for the case where the node spacing is the same or approximately the same, and when the feature point spacing is not equal, the non-uniform B-spline shows its advantages. The formula of k degree non-uniform B-spline is:

$$p(\mathbf{u}) = \sum_{i=0}^{n} d_i N_{i,k}(u) \tag{30}$$

Among them, d_i is the control vertex, $i = 0,1, \ldots, n$, $N_{i,k}(u)$ is the basis function of k-degree normalized B-spline, and the basis function interval is $u \in [u_i, u_{i+k+1}]$, and has the following definition:

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, u_i \leq u \leq u_{i+1} \\ 0, \text{others} \end{cases} \\ N_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ \frac{0}{0} = 0 \end{cases} \tag{31}$$

The B-spline curve expression is further converted by the De Boole recursive algorithm. Let m be the order of the derivative function, and the m-order derivative function of each joint is calculated:

$$\begin{cases} p(\mathrm{u}) = \sum_{j=i-k+1}^{i} d_j^l N_{j,k-1}(u) = \cdots = d_i^{k-m}, \\ \qquad u \in [u_i, u_{i+1}] \\ d_j^l = \begin{cases} d_j, \\ (k+1-l)\frac{d_{j-1}^{l-1}-d_j^{l-1}}{u_{j+k+1-l}-u_j}, \begin{cases} l = 1,2,\cdots m \\ j = i-k+l,\cdots i \end{cases} \end{cases} \end{cases} \tag{32}$$

In the actual trajectory planning, in order to make the first and last position points and the piecewise continuous points consistent with the corresponding setting points, the B-spline control points need to be inversely calculated. Given the position point $(p_i, t_i)(i = 0,1,\cdots n)$, the non-uniform B-spline is composed of n piecewise functions with $\mathrm{n} + 2k + 1$ node vectors. In order to obtain $\mathrm{n} + k$ control points, the repeatability of the first and last nodes is $k + 1$, and the node vector is normalized according to the interpolation curve. The distance between adjacent nodes can be expressed as:

$$u_i - u_{i-1} = \frac{\sum_{j=i-k}^{i-1} |t_j - t_{j-1}|}{\sum_{i=k+1}^{n+k} \sum_{j=i-k}^{i-1} |t_j - t_{j-1}|}, i = k+1, k+2, \cdots, n+k \tag{33}$$

Then the expression of each node vector can be calculated:

$$\begin{cases} u_0 = u_1 = \cdots = u_k = 0 \\ u_i = \sum_{j=k+1}^{i} (u_j - u_{j-1}), i = k+1, \cdots, n+k-1 \\ u_{n+k} = u_{n+k+1} = \cdots = u_{n+2k} = 1 \end{cases} \tag{34}$$

Let T be the total interpolation time, the first derivative of the first and last points of the curve can be obtained:

$$\begin{cases} p_0' = p'(u_k) = d_1^1 = k\frac{d_1-d_0}{u_{k+1}-u_1} \\ p_{n+2k}' = p'(u_{n+k}) = \sum_{j=n}^{n+k-1} d_j^1 N_{j,k-1}(u_n) \\ \qquad = d_{n+k-1}^1 = k\frac{d_{n+k-1}-d_{n+k-2}}{u_{n+2k-1}-u_{n+k-1}} \end{cases} \tag{35}$$

For the cubic non-uniform B-spline, that is, when k = 3, the first-order derivative of the first and last nodes is sufficient, and the fifth or higher order needs to carry out multi-order derivation. Combined with the above first-order derivative formula, n + 3 equations can be obtained, and the control points of the cubic B-spline can be obtained. The final solution equation is:

$$S_N d = p \tag{36}$$

Among them:

$$S_N = \begin{bmatrix} N_{0,3}(u_3) & N_{1,3}(u_3) & & & & \\ & N_{1,3}(u_4) & N_{2,3}(u_4) & N_{3,3}(u_4) & & \\ & \ddots & \ddots & \ddots & & \\ & & N_{n-1,3}(u_{n+2}) & N_{n,3}(u_{n+2}) & N_{n+1,3}(u_{n+2}) & \\ & & & N_{n,3}(u_{n+3}) & N_{n+1,3}(u_{n+3}) & N_{n+2,3}(u_{n+3}) \\ f_{v1}^{s} & f_{v2}^{s} & & & & \\ & & & & f_{v1}^{z} & f_{v2}^{z} \end{bmatrix} \tag{37}$$

f_v^s and f_v^z are the position coefficients of the starting and ending position control points, respectively. Their values are:

$$f_{v1}^{s} = -k(u_{k+1} - u_1)^{-1} \tag{38}$$

$$f_{v2}^{s} = k(u_{k+1} - u_1)^{-1} \tag{39}$$

$$f_{v1}^{z} = -k(u_{u+2k+1} - u_{n+k-1})^{-1} \tag{40}$$

$$f_{v2}^{z} = k(u_{n+2k+1} - u_{n+k-1})^{-1} \tag{41}$$

$p = (p_0, p_1, \cdots, p_n, v_s, v_z)^T$, v_s and v_z are the velocities of the starting and ending points.

At this point, the cubic non-uniform B-spline curve planning ends. If the order is increased, the parameters of the starting and ending point acceleration or even the acceleration need to be calculated.

4.2. Polynomial Quadratic Interpolation

Because the traditional classical quintic B-spline curve cannot guarantee the requirement that the acceleration of the first and last positions of each joint of the robot is 0, it will produce flexible impact. Therefore, the quintic polynomial is introduced to optimize the first and last two intervals of the cubic B-spline, and the convex hull of the B-spline is retained. At the same time, the angular acceleration is continuous, which meets the requirement that the acceleration of the first and last positions is 0, so as to avoid impact and vibration.

It can be found that the quintic polynomial interpolation function is similar to the cubic polynomial solution method, but there are only two unknowns, and the same place is no longer described in detail. The angle, angular velocity and angular acceleration of each joint at the starting and ending positions are selected as constraints, and the expression is:

$$\theta(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \tag{42}$$

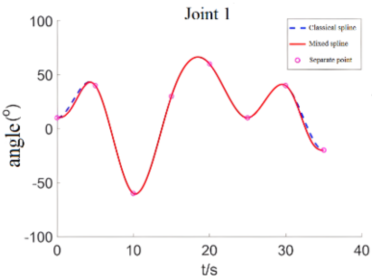
In the formula, $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ are the six joint angles of the robot, and $c_{0\sim5}$ is the six unknown coefficients. Substituting the six parameters of angle, angular velocity and angular acceleration of each joint at the beginning and end time into the above formula and deriving, we can get:

$$\begin{cases} \theta(t_0) = \theta(0) = c_0 \\ \theta'(t_0) = \theta'(0) = c_1 \\ \theta''(t_0) = \theta_0'' = 2c_2 \\ \theta(t_f) = \theta_f = c_0 + c_1 t_f + c_2 t_f^2 + c_3 t_f^3 + c_4 t_f^4 + c_5 t_f^5 \\ \theta'(t_f) = \theta_f' = c_1 + 2c_2 t_f + 3c_3 t_f^2 + 4c_4 t_f^3 + 5c_5 t_f^4 \\ \theta''(t_f) = \theta_f'' = 2c_2 + 6c_3 t_f + 12c_4 t_f^2 + 20c_5 t_f^3 \end{cases} \tag{43}$$

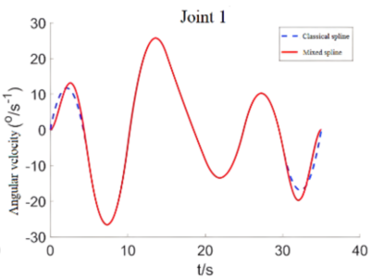
Based on the above six equations, six coefficients can be solved, so as to plan the fifth-order polynomial trajectory curve of the first and last intervals.

4.3. Trajectory Simulation Based on Mixed B-spline Interpolation

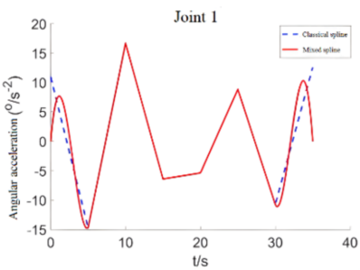
The GSK RB08A3 industrial robot joint one to joint three is used as the research object, and MATLAB is used as the simulation platform for simulation. The joint trajectory is constructed by cubic non-uniform B-spline and quintic polynomial piecewise hybrid interpolation, and compared with the traditional classical cubic B-spline interpolation. The velocity and acceleration of the three joints are set to 0, and the simulation results are shown in Figure 7-9.



a) Angle simulation results



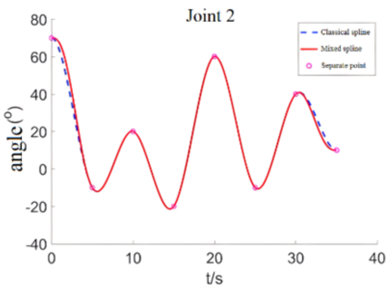
b) Angular velocity simulation results



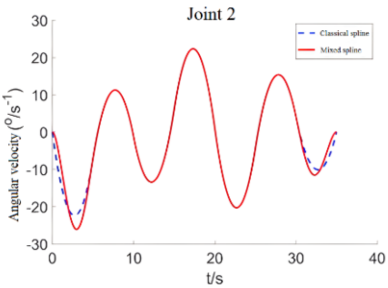
c) Angular acceleration simulation results

Figure 7. Joint 1 simulation results.

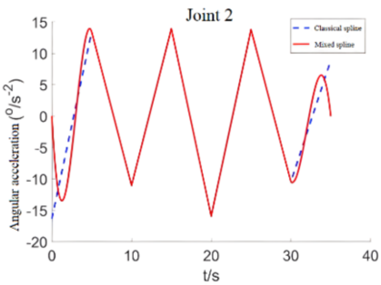
It can be seen from the above figure that there is no significant difference between the hybrid B-spline and the traditional classical B-spline in the angle and angular velocity curves. However, for the acceleration curve, because the hybrid B-spline adopts the quintic polynomial quadratic interpolation, the acceleration curve of the first and last segments is smooth and continuous without mutation, and it can meet the requirement that the angular acceleration at the first and last positions is 0, avoiding the flexible impact at the start and stop, and satisfying the constraint conditions.



a) Angle simulation results

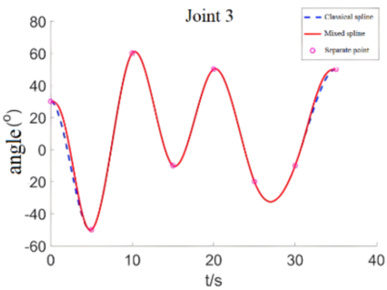


b) Angular velocity simulation results

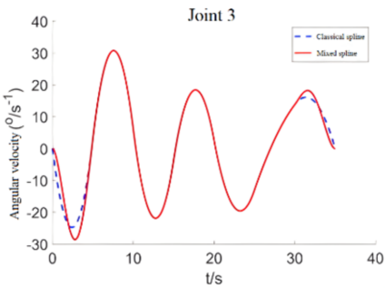


c) Angular acceleration simulation results

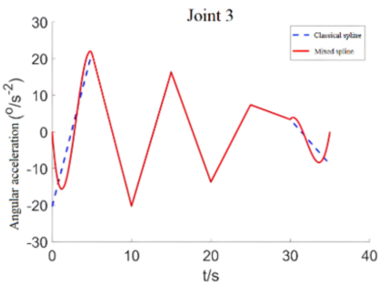
Figure 8. Joint 2 simulation results.



a) Angle simulation results



b) Angular velocity simulation results



c) Angular acceleration simulation results

Figure 9. Joint 3 simulation results.

By mixing the non-uniform B-spline with the polynomial, a smooth and continuous trajectory with specified velocity and acceleration at the beginning and end can be obtained, which can greatly shorten the planning time, greatly reduce the energy loss, and achieve the dual-objective comprehensive optimization of time-energy consumption.

5. Conclusions

This paper first discusses the research status of industrial robot trajectory planning at home and abroad. Then, it mainly focuses on the kinematics modeling and analysis, trajectory planning algorithm and motion planning simulation of six-degree-of-freedom robots, and uses formula derivation, model creation and algorithm programming methods and MATLAB software platform to carry out related research. The specific work of this paper is as follows:

(1) Kinematics analysis and modeling of six-degree-of-freedom industrial robot

This paper takes the GSK RB08A3 industrial robot as the research object. Firstly, the modeling method of industrial robot is expounded. Then the kinematics model is established by SDH method. Finally, the forward and inverse kinematics are deduced and calculated by homogeneous matrix method and algebraic method respectively.

(2) Research on Cartesian space trajectory planning of industrial robot

The trajectory planning scheme of industrial robot in Cartesian space is studied. Firstly, the position and attitude interpolation methods are discussed, and the spatial linear and circular position interpolation and the attitude interpolation of the unit quaternion are studied. Then, based on the S-shaped velocity curve, a multi-segment trajectory smooth transition algorithm is designed. An arc connection model is established between adjacent trajectory segments, and the speed of the trajectory connection is processed. The algorithm program is designed and written. Finally, taking GSK RB08A3 industrial robot as the research object, the linear and circular interpolation simulation and multi-segment trajectory transition experiment are carried out on the MATLAB platform, and compared with the traditional S-shaped velocity curve, the superiority of the multi-segment trajectory smooth transition algorithm is verified.

(3) Research on joint space trajectory planning of industrial robot

The trajectory planning scheme of industrial robot joint space is studied. In order to solve the problems of industrial robot efficiency, energy consumption and flexible impact during start-stop, the trajectory planning method based on hybrid B-spline interpolation is deeply studied. Firstly, the cubic B-spline interpolation and quintic polynomial interpolation methods are introduced, and the idea of B-spline and polynomial hybrid construction trajectory is proposed. The middle section uses cubic B-spline interpolation, and the first and last sections use quintic polynomial quadratic interpolation. Then, the GSK RB08A3 industrial robot is taken as the research object, and the 5-3-5 hybrid interpolation trajectory and the classical cubic B-spline interpolation trajectory are simulated and compared. The results show that the start-stop impact problem can be avoided, the planning time can be shortened, and the energy consumption can be reduced, which verifies the superiority of the hybrid interpolation algorithm.

The Cartesian space multi-trajectory planning scheme based on S-shaped velocity curve designed in this paper can take corner error and dynamic constraints into consideration to further optimize the control algorithm and improve the practicability of the algorithm. In the 5-3-5 joint space trajectory algorithm, the adverse effects of jerk discontinuity on trajectory pulsation are not considered. In the future research, the solution of pulsation discontinuity will be further analyzed based on this.

References

[1] Gasparetto A, Boscariol P, Lanzutti A, et al. Path planning and trajectory planning algorithms: A general overview. Motion and Operation Planning of Robotic Systems: Background and Practical Approaches, 2015: 3-27.

[2] Lan J, Xie Y, Liu G, et al. A multi-objective trajectory planning method for collaborative robot. Electronics, 2020, 9(5): 859.

[3] Ma W, Hu T, Zhang C, et al. A robot motion position and posture control method for freeform surface laser treatment based on NURBS interpolation. Robotics and Computer-Integrated Manufacturing, 2023, 83: 102547.

[4] Wu H, Huang J, Lin S, et al. Application of improved S-curve flexible acceleration and deceleration algorithm in smart live working robot. Journal of Physics: Conference Series. IOP Publishing, 2021, 2005(1): 012065.

[5] Laha R, Rao A, Figueredo L F C, et al. Point-to-point path planning based on user guidance and screw linear interpolation. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. American Society of Mechanical Engineers, 2021, 85451: V08BT08A010.

[6] Chen W, Li X, Ge H, et al. Trajectory planning for spray painting robot based on point cloud slicing technique. Electronics, 2020, 9(6): 908.

[7] Fang S, Cao J, Zhang Z, et al. Study on high-speed and smooth transfer of robot motion trajectory based on modified S-shaped acceleration/deceleration algorithm. IEEE Access, 2020, 8: 199747-199758.

[8] Chen W, Wang H, Liu Z, et al. Time-energy-jerk optimal trajectory planning for high-speed parallel manipulator based on quantum-behaved particle swarm optimization algorithm and quintic B-Spline. Engineering Applications of Artificial Intelligence, 2023, 126: 107223.

[9] Nazarahari M, Khanmirza E, Doostie S. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Systems with Applications, 2019, 115: 106-120.

[10] Luan P G, Thinh N T. Hybrid genetic algorithm based smooth global-path planning for a mobile robot. Mechanics Based Design of Structures and Machines, 2023, 51(3): 1758-1774.

[11] Chakraborty S, Aithal P S. Forward Kinematics Demonstration of 6DF Robot using CoppeliaSim and C. International Journal of Applied Engineering and Management Letters (IJAEML), 2021, 5(1): 29-37.

[12] Xiao F, Li G, Jiang D, et al. An effective and unified method to derive the inverse kinematics formulas of general six-DOF manipulator with simple geometry. Mechanism and Machine Theory, 2021, 159: 104265.

[13] Zaplana I, Hadfield H, Lasenby J. Closed-form solutions for the inverse kinematics of serial robots using conformal geometric algebra. Mechanism and Machine Theory, 2022, 173: 104835.

[14] Wei Y. Kinematics Analysis and Modelling Based on 4 DOF Industrial Robot. Journal of Physics: Conference Series. IOP Publishing, 2023, 2562(1): 012076.

[15] Nguyen N T, Schilling L, Angern M S, et al. B-spline path planner for safe navigation of mobile robots. 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021: 339-345.