International Journal of Frontiers in Engineering Technology
ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

Trajectory planning algorithm of manipulator based on NURBS

Liangyun Nie*, Yuanchao Yang, Xiaohua Wang, Wenjie Wang

Xi'an Polytechnic University, Xi'an, Shaanxi, 710000, China
**Corresponding author: 200421120@stu.xpu.edu.cn*

Abstract: *This paper uses the 5Dof manipulator on the YouBot developed by KUKA as a model. Combining quintic polynomial interpolation with quintic non-uniform rational B splines, a novel segmented polynomial interpolation planning technique 5-B-5 is proposed in this study. CoppeliaSim constructs the model, and MATLAB controls the model to acquire the angular displacement, angular velocity, angular acceleration, final trajectory, and joint force for every joint angle. The new optimization algorithm presented in this work is utilized to optimize the manipulator's motion trajectory. Comparing it to the classic 3-5-3 algorithm and the B spline algorithm offers advantages over the higher-order B spline interpolation without increasing the manipulator control system's operational overhead. The experiments prove that the motion trajectory inaccuracy of the manipulator is minimized, the curve is continuous and smooth, and the smoothness is improved, thereby eliminating mechanical shock and other issues.*

Keywords: *Manipulator, Non-uniform Rational B-splines, Quintic Polynomial Interpolation, Trajectory Planning*

1. Introduction

The manipulator trajectory planning algorithm is studied to increase its precision and adaptability at the end-effector. In most cases, trajectory planning occurs in the joint space of the manipulator since the control action on the manipulator occurs at the joints. A series of control points are extracted from the planned end-effector trajectory, followed by kinematic computations to determine the manipulator joint values to construct the trajectory in joint space. Using interpolation functions and considering the kinematic restrictions placed on the manipulator's joints, trajectories are calculated in joint space. Whether planning in Cartesian space or joint space, the motion law formed by planning must not generate forces and moments at joints that are incompatible with provided restrictions. The likelihood of mechanical resonance can be substantially decreased. Therefore, the planning method must generate smooth trajectories[1].

B-splines can fit virtually all types of spatial curves, so many Chinese scholars utilize them for curve fitting. The NURBS curve based on B splines is the sole standard authorized by the International Standards Organization (ISO) for the geometric design of industrial goods [2].

A method for adding correction factors in the segment intervals of the third spline interpolation is presented in [3]. The correction factor in the first segment interval is a fifth correction function, making the starting acceleration zero. The remainder of the interval is a sixth correction function, making the acceleration of the third-order derivative of the motion trajectory continue and at the termination position of zero. It corrects the problem where the acceleration of the typical third spline interpolation varies suddenly at the initial and final locations.

The manipulator trajectory planning approach provided in the literature [4] employing the optimization strategy of the fifth NURBS curve in conjunction with the reparameterization of the motion time may effectively minimize energy consumption, shorten execution time, and increase motion smoothness. Each type-valued point of the constitutive curve is related to the trajectory planning curve created by the quintic polynomial. The local support property of the B-sample curve can efficiently optimize the curve after trajectory planning by modifying only a few neighboring type-valued points. Using 3-4-5 polynomials in operation space and quintic non-uniform rational B-spline motion laws in joint space[5] , the mapping relationship between joint space motion characteristics and operation space motion characteristics was established. The results of the comparative analysis indicate that the trajectory

ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

planning method based on the quintic non-uniform B spline motion law in the joint space appears to be more effective at reducing the vibration of the robot arm during motion and the amount of motor power required. Literature [4] investigated the establishment procedure of the five-segment cubic polynomial interpolation function, the 3-5-3 interpolation function, and the 4-3-4 interpolation function in further depth and calculated their equation expressions for manipulator trajectory planning in joint space.

Although these robot manipulator trajectory algorithms still suffer from smoothing problems and mechanical system shocks. To overcome these shortcomings, we decided to a new interpolation technique for trajectory planning called the "5-B-5 algorithm." The comparison simulation experiment of the 3-5-3 algorithm demonstrates that the algorithm improves the smoothness of the angular velocity and angular acceleration curves during the motion of the industrial manipulator, reduces the impact on the mechanical system caused by the sudden change in angular acceleration, and solves the problem that the velocity and acceleration of NURBS cannot be zero at the initial and final.

2. 5-B-5 trajectory planning algorithm

The velocity curve of NURBS has non-zero velocity values at the initial and final points, which indicates that the manipulator's velocity will undergo sudden fluctuations; guttering is undesirable and has a significant influence on the manipulator. To make the NURBS curve more applicable to the manipulator's trajectory planning, we propose a quintic polynomial that satisfies the conditions that the velocity and acceleration of both the initial and final points are zero. In this research, we present a hybrid method 5-B-5 using a quintic polynomial for the initial and final segments and NURBS for the intermediate segments, combining the benefits of both.

Since the non-uniform B spline curve cannot accurately represent the quadratic curve arcs excluding parabolas and the primary surfaces composed of quadratic surfaces and planes. Therefore, a set of weight factors ω_i is added to the general non-uniform B spline to regulate the action distribution of the control points. In this paper, we will consider the effect of modifying the weight factor ω_i to change the shape of the curve on the interval $u \in [u_i, u_{i+k+1})$. ω_i increases (decreases) and the point $\mathrm{C(u)}$ near (away) $\mathrm{P_i}$, the generated curve is pulled (pushed away from) $\mathrm{P_i}$. Similar to rational Bézier curves, NURBS curves are expressed in terms of chi-square coordinates. (A polynomial curve in $n+1$ dimensions represents a rational curve in n dimensions.) Let H be the perspective transformation defined by equation (1).

$$\mathrm{P} = \mathrm{H}\{\mathrm{P}^{\omega}\} = \mathrm{H}\{(\mathrm{X,Y,Z,W})\} = \begin{cases} \left(\frac{\mathrm{X}}{\mathrm{W}}, \frac{\mathrm{X}}{\mathrm{W}}, \frac{\mathrm{X}}{\mathrm{W}}\right), & \mathrm{W} \neq 0 \\ \mathrm{direction(X,Y,Z)}, & \mathrm{W} = 0 \end{cases} \tag{1}$$

Construct the control points with weights $\mathrm{P_i^{\omega}} = (\omega_i x_i, \omega_i y_i, \omega_i z_i, \omega_i)$ for a given set of control points $\mathrm{P_i}$ and weight factor ω_i, and then define the non-rational B-spline curve in four dimensions as

$$\mathrm{C}^{\omega}(\mathrm{u}) = \sum_{i=0}^{n} \mathrm{N_{i,k}(u) P_i^{\omega}} \tag{2}$$

The rational B-sample curve is obtained when the perspective transformation H is applied to $\mathrm{C}^{\omega}(\mathrm{u})$.

$$\mathrm{C(u)} = \mathrm{H}\{\mathrm{C}^{\omega}(\mathrm{u})\} = \left\{\sum_{i=0}^{n} \mathrm{N_{I,k}(u) P_i^{\omega}}\right\} = \mathrm{C(u)} = \frac{\sum_{i=0}^{n} \mathrm{N_{i,k}(u)} \omega_i \mathrm{P_i}}{\sum_{i=0}^{n} \mathrm{N_{i,k}(u)} \omega_i} \tag{3}$$

$$\begin{cases} \mathrm{N_{i,k}(u)} = \frac{u-u_i}{u_{i+k}-u_i} \mathrm{N_{i,k-1}(u)} + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} \mathrm{N_{i+1,k-1}(u)} \\ \mathrm{N_{i,0}(u)} = \begin{cases} 1 & u_i \leq u \leq u_{i+1} \\ 0 & \text{others} \end{cases} \end{cases} \tag{4}$$

We refer to both $\mathrm{C}^{\omega}(\mathrm{u})$ and $\mathrm{C(u)}$ without distinction as NURBS curves, where $\mathrm{P_i}, \mathrm{i} = 0,1,2,\ldots,\mathrm{n}$ are the control points; the kth B-sample basis function on the non-uniform node vector U is $\mathrm{N_{i,k}(u)}$, by equation (2), which specifies 0/0=0, where the last and first nodes have $k+1$ repetition. The nodal vector U varies for different B spline curves, with the non-uniform node vector being parameterized by the chord length parameterization to produce the control point parameterization procedure.

Let ω_i vary, $0 \leq \omega_i < +\infty$, recall that the curve cluster is $\mathrm{C(u;\omega_i)}$ below Fig1, and define the points $\mathrm{R} = \mathrm{C(u;\omega_i = 0)}$;$\mathrm{M} = \mathrm{C(u;\omega_i = 1)}$;$\mathrm{P} = \mathrm{C(u;\omega_i \neq 0,1)}$from the curve specified by equation (5).

ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

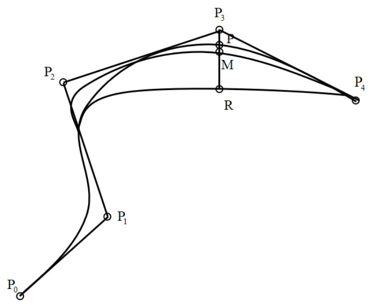


Figure 1: Geometric Meaning of Weight Factor

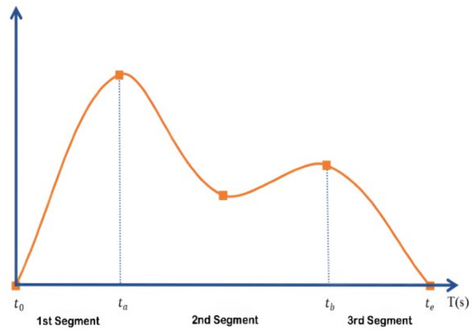


Figure 2: 5-b-5 algorithm segmentation diagram

M and P are written as:

$$M = (1-s)R + sP_i, P = (1-t)R + tP_i \tag{5}$$

Where $s = \frac{N_{j,k}(u)}{\sum_{i\neq j} N_{i,k}(u)\omega_i + N_{j,k}(u)}, t = \frac{N_{j,k}(u)\omega_j}{\sum_{i=0}^{n} N_{i,k}(u)\omega_i}$ from the above equation we get

$$\omega_j = \frac{\frac{\sum_{i\neq j} N_{i,k}(u)\omega_i}{N_{j,k}(u)}}{\frac{\sum_{i\neq j} N_{i,k}(u)\omega_i}{N_{j,k}(u)\omega_j}} = \frac{\frac{1-s}{s}}{\frac{1-t}{t}} = \frac{\frac{P_jM}{RM}}{\frac{P_jP}{RP}} \tag{6}$$

By modifying ω_j to reach a new position $\widehat{P}$, $d = |P - \widehat{P}|$. We cause the point P be pulled toward P_j, Then, the following equation may be used to get the new weight factor $\widehat{\omega}_j$

$$\widehat{\omega}_j = \omega_j \left[1 \pm \frac{d}{\frac{N_{j,k}(u)\omega_j}{\sum_{i=0}^{n} N_{i,k}(u)\omega_i}(P_jP \mp d)}\right] \tag{7}$$

Since it is pulling toward P, both ± and ∓ in the Eq are represented by the symbols above. During the actual operation of the manipulator, it is typically taught to obtain several key taught points, also known as date points, and use these date points to perform curve back-calculation and generate NURBS curves to obtain each joint trajectory through a given path point. The back-calculation procedure is as follows: based on the distribution of date points, the NURBS node vector U is generated, and then the NURBS basis function N is derived.

$$C_i(u) = \frac{\sum_{p=i}^{i+3} N_{I,3}(u)\omega_i P_i}{\sum_{p=i}^{i+3} N_{I,3}(u)\omega_i} = \frac{TM_i[w_iP_i \;\; w_{i+1}P_{i+1} \;\; w_{i+2}P_{i+2} \;\; w_{i+3}P_{i+3}\;]^T}{TM_i[w_i \;\; w_{i+1} \;\; w_{i+2} \;\; w_{i+3}]} \tag{8}$$

Where $T = [1 \;\; t \;\; t^2 \;\; t^3\;], t = \frac{u-u_{i+3}}{u_{i+4}-u_{i+3}}, TM_i = \left[\frac{(\Delta_{i+3})^2}{\Delta_{i+2}^2\Delta_{i+1}^3} \;\; 1 - \frac{(\Delta_{i+3})^2}{\Delta_{i+2}^2\Delta_{i+1}^3} - \frac{(\Delta_{i+2})^2}{\Delta_{i+2}^3\Delta_{i+2}^2} \;\; \frac{(\Delta_{i+2})^2}{\Delta_{i+2}^3\Delta_{i+2}^2} \;\; 0\right]$. Let $a_{i+1} = \frac{(\Delta_{i+3})^2}{\Delta_{i+2}^2\Delta_{i+1}^3}\omega_i$; $b_{i+1} = \left[1 - \frac{(\Delta_{i+3})^2}{\Delta_{i+2}^2\Delta_{i+1}^3} - \frac{(\Delta_{i+2})^2}{\Delta_{i+2}^3\Delta_{i+2}^2}\right]\omega_{i+1}$; $c_{i+1} = \frac{(\Delta_{i+2})^2}{\Delta_{i+2}^3\Delta_{i+2}^2}\omega_{i+2}$ to obtain $a_{i+1}P_i + b_{i+1}P_{i+1} + c_{i+1}P_{i+2} = (a_{i+1} + b_{i+1} + c_{i+1})P_{i+1} = Q_{i+1}, (i = 0,1,\dots,n-1)$

Here two additional conditions $P_0 = P_i, P_n = P_{n+1}$ are needed based on the known n date points. $n+2$ control points can be found by constructing the inverse coefficient matrix (9) based on $N_{i,3}(u)$

$$\begin{bmatrix} b_0 & c_0 & & & & \\ a_1 & b_1 & c_1 & & & \\ & a_2 & b_2 & c_2 & & \\ & & \ddots & \ddots & \ddots & \\ & & & a_n & b_n & c_n \\ & & & & a_{n+1} & b_{n+1} \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ \vdots \\ P_n \\ P_{n+1} \end{bmatrix} = \begin{bmatrix} Q_0 \\ Q_1 \\ Q_2 \\ \vdots \\ Q_n \\ Q_{n+1} \end{bmatrix} \tag{9}$$

NURBS can accommodate nearly all curves and surfaces by adding variable weights, allowing for increased flexibility in the shape of related curves. The introduction of weights complicates the calculation of curve generation and derivation, even though NURBS has the benefits of geometric continuity, convex wrapping, and local support. This is primarily because the curve inversion procedure becomes extremely unstable in the initial and final stages. In addition, the NURBS velocity curve has initial and final velocity values that are not zero, and the manipulator is prone to jittering. Quintic polynomial interpolation can circumvent these issues by setting initial and final velocity and acceleration to zero. A novel segmented polynomial interpolation algorithm is proposed to meet the requirements of displacement, angular velocity, and angular acceleration in manipulator trajectory planning, i.e., 5-B-5.

Using segmented polynomial interpolation for joint space trajectory planning. As shown in Figure 2, the method divides the trajectory planning of the manipulator into three segments: the trajectories of the 1st segment $t_0 \sim t_a$ and the 3rd segment $t_b \sim t_e$ have quintic polynomial interpolation; the 2nd segment $t_a \sim t_b$ uses NURBS. The positions of the joint points and the quintic polynomial interpolation determine the angular displacement, angular velocity, and angular acceleration of the initial and final points of NURBS.

The curvilinear expressions for the first and third segments' curves[6]

$$\Theta(t) = \theta_0 + \dot{\theta}_0 t + \frac{\dot{\theta}_0}{2} t^2 + \frac{20\theta_e - 20\theta_0 - (8\dot{\theta}_e + 12\dot{\theta}_0)t_e - (3\ddot{\theta}_0 - \ddot{\theta}_e){t_e}^2}{2t_e^3} t^3 + \frac{30\theta_0 - 30\theta_e - (14\dot{\theta}_e + 16\dot{\theta}_0)t_e + (3\ddot{\theta}_0 - 2\ddot{\theta}_e){t_e}^2}{2t_e^4} t^4 + \frac{12\theta_e - 12\theta_0 - (6\dot{\theta}_e + 6\dot{\theta}_0)t_e - (\ddot{\theta}_0 - \ddot{\theta}_e){t_e}^2}{2t_e^5} t^5 \quad (10)$$

Where θ_0 and θ_e are the angles at the initial and final moments; $\dot{\theta}_0$ and $\dot{\theta}_e$ are the angular velocities at the initial and final moments, respectively; and $\ddot{\theta}_0$ and $\ddot{\theta}_e$ are the angular accelerations at the initial and final moments, respectively. The quintic polynomial interpolation method can satisfy that the velocity and acceleration at the initial and final points are zero.

3. Simulation comparison experiment

3.1 Manipulator model

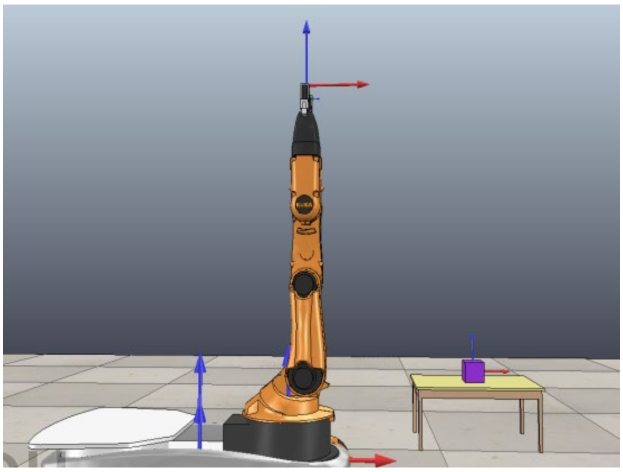


Figure 3: Coppeliasim Simulation Model

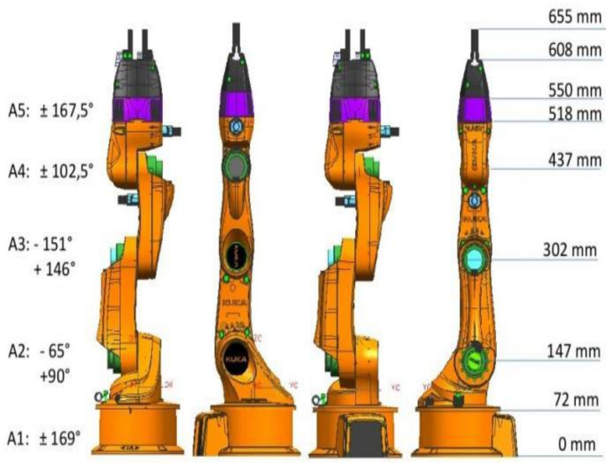


Figure 4: The Size of The KUKA YouBot's Arm

CoppeliaSim (formerly known as V-REP) is a general-purpose simulation adapter with an integrated

ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

development environment that offers a comprehensive library of models and scripts for existing mobile and stationary robots. It is used for algorithm development, factory simulation, remote monitoring, education, and other purposes. Using embedded scripts connected to specific scene objects, such as joints or sensors, simplifies the robot's behavior when simulated. Communication with external applications (Python, Matlab, ROS, or IoT) enables seamless portability to real devices[7].

The addition of a YouBot model to CoppeliaSim is depicted in Figure 3. Using CoppeliaSim's native Lua script, MATLAB's Robotic Toolbox toolkit, and the VrepLib library, regulate the motion of the robot manipulator. Combining MATLAB and CoppeliaSim's strengths enables the simulation of robot motion. Using the joint parameters (figure 4 and table 1) provided by MATLAB, CoppeliaSim is then used to construct the robot's 3D scene and control the manipulator trajectory planning.

The API trajectory file is read by MATLAB, which then interpolates and samples the trajectory, calculates the T-matrix between different coordinate systems, and identifies the robot's positive and negative kinematics. Communication links the MATLAB control program to CoppeliaSim, allowing the CoppeliaSim model to respond to the MATLAB control program for motion simulation. The CoppeliaSim help file details the specific data communication process between the two.

Table 1: D-H Parameter Table of YouBot Manipulator

Joint	θ_{n+1}(rad)	d_{n+1}(mm)	a_{n+1}(mm)	α_{n+1} (rad)
1	θ_1	75	33	90
2	θ_1	0	155	0
3	θ_1	0	135	0
4	θ_1	0	0	90
5	θ_1	218	0	0

3.2 Experimental Results

Step 1: Initializes (θ_i, t), $i = 1,2,\dots,n$;

Step 2: Calculates the NURBS velocity profile to obtain the initial and final velocity and acceleration of the second segment;

Step 3: Applies quintic polynomial interpolation algorithm to fit the first segment trajectory based on θ_1, θ_2, and the initial velocity and acceleration of NURBS;

Step 4: Applies the NURBS algorithm to fit the second segment trajectory according to θ_1.

Step 5: Applies quintic polynomial interpolation algorithm to fit the third segment trajectory based on θ_1, θ_2, and the final velocity and acceleration of NURBS;

Step 5: The results are compared with 3-5-3 algorithm and B-spline algorithm.

The spline curves of each joint are determined through simulation experiments. The points dotted line represents the interpolation of the 3-5-3 spline curve, while the dotted line and the solid line are B-spline curve and 5-B-5 spline curve. For a more straightforward presentation, Figures 5~7 only depict single joint curve figures. However, the results are consistent with the planning objective and do not affect the outcomes[8][9].

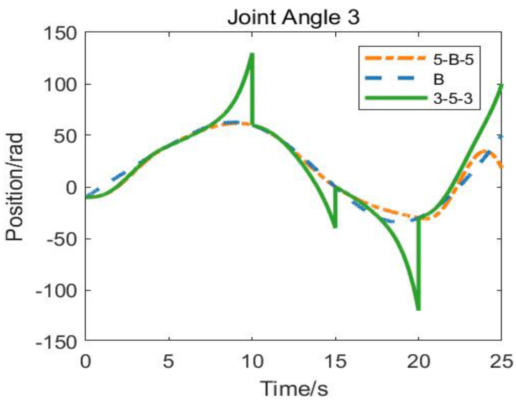


Figure 5: Comparison of Angular Curves for Joint 1

ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

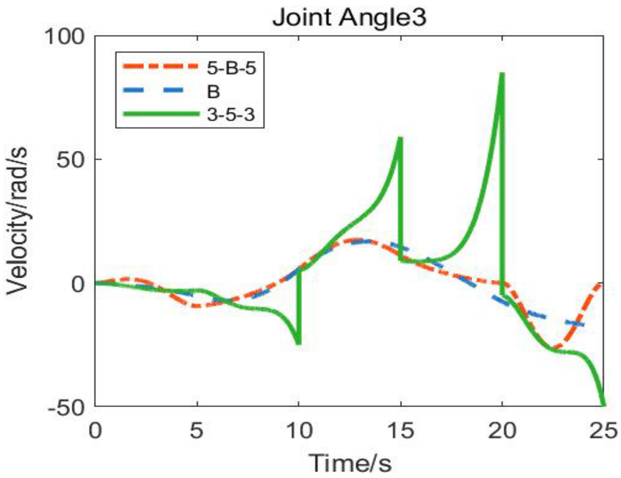


Figure 6: Comparison of Angular Velocity Curves for Joint 3

The comparison figure 5 of the angle curves of joint 1 shows that the trends of the angular displacement curves obtained by the three planning methods do not differ much. However, the angular displacement curves of the 5-B-5 algorithm are smoother and gentler than the joint angle curves of the other two trajectory planning, and the change rate is negligible. 3-5-3 spline curves show up-and-down fluctuations and significant change rate, and the shape is close to a sharp jagged angle, which is easy to make the manipulator vibrate. B spline curves have the start-stop characteristics problem. The 5-B-5 algorithm ensures that the kinematic parameters of the manipulator joints and end-effectors are continuous without sudden changes and improves the manipulator motion characteristics.

The figure 6 of angular velocity demonstrates that the 5-B-5 algorithm's curve initials and finals at 0, there is no abrupt change in the middle section, and the entire curve is smooth and continuous. However, the other two algorithms cannot make the initial and final velocities equal to zero, resulting in initial moments that are not equal to zero, which will cause tracking errors and significant vibrations. The 5-B-5 algorithm is not susceptible to the issue of mechanical shocks. Polynomial interpolation can also overcome the Runge phenomenon with an increase in order.

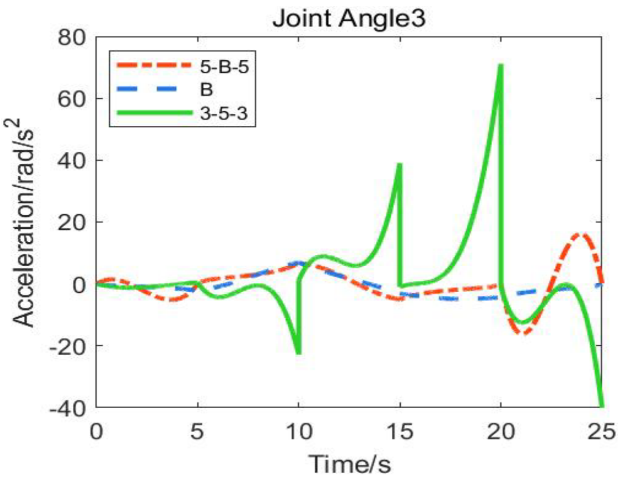


Figure 7: Comparison of Angular Acceleration Curves for Joint

The acceleration graph demonstrates that the entire curve of the 5-B-5 algorithm is continuous and smooth. However, the curve of the 3-5-3 algorithms and the corner of the middle section have sharp corners, and the acceleration varies rapidly, indicating the existence of mechanical shocks. The joint angle interpolation discrete points are kinematically solved positively to obtain the motion trajectory of the end-effector. The segmented interpolation trajectory is compared with the trajectory of the conventional interpolation algorithm, as depicted in Figure 7.

ISSN 2706-655X Vol.5, Issue 3: 49-56, DOI: 10.25236/IJFET.2023.050308

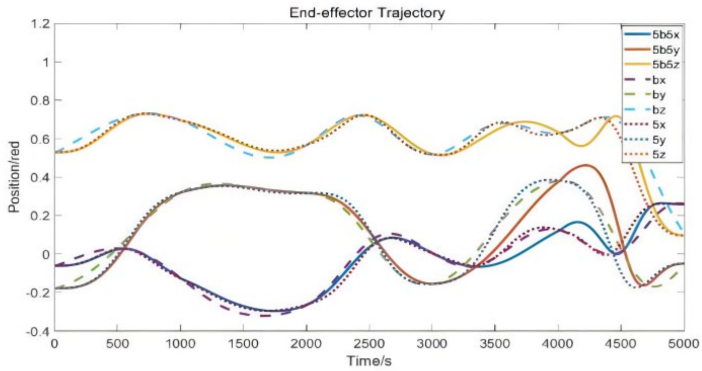


Figure 8: Comparison of Terminal Trajectories of Different Algorithms

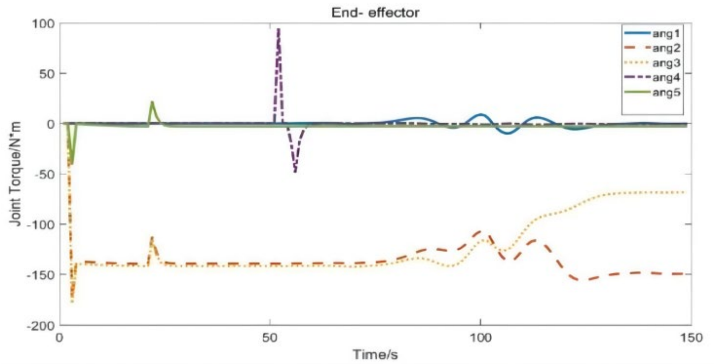
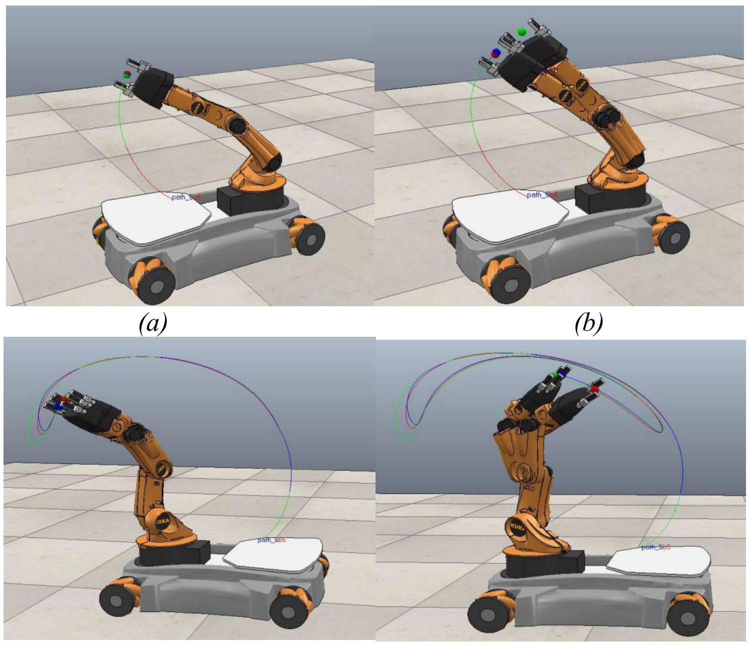


Figure 9: Joint Moment

The 5-B-5 trajectory can segment processing and shape rapidly, making it suitable for real-time control. As shown in figures 8 and 9 are the end trajectories of the manipulators and the moment diagrams of the joints, the initial point and final point of the curve of 5-B-5 are identical, and the starting force is not zero. The diagram reveals that the 5-B-5 algorithm cannot quickly generate momentary shocks. The curve has a slight undulation, making it more smooth and challenging to generate momentary shocks in motion planning[10].



(a) *(b)* *(c)* *(d)*

Figure 10: Terminal Planning Trajectory

The robot was simulated in CoppeliaSim to plan the end-effector along the specified path to obtain the end-effector trajectory of the YouBot. The obtained trajectory data was used to generate a comparison figure 10. The simulation results depicted in Figure 8(b) indicate that the end-effector trajectory motion of the 5-B-5 algorithm follows the specified path and is rapid and smooth. According to Fig. 8(c), it is evident that the other trajectory planning algorithms deviate from the desired trajectory for the second trajectory planning. The absence of joint vibration throughout the 5-B-5 process indicates that this trajectory planning algorithm is accurate and practicable.

4. Conclusions

In the direction of intelligent manipulator manufacturing, it would be beneficial to conduct additional research on balancing time, energy consumption, and smooth trajectory planning. This paper investigates the manipulator trajectory planning problem under constraints based on kinematics and interpolation. The study is based on a thorough analysis of domestic and international research.

The 5-B-5 algorithm for trajectory planning is based on NURBS and a quintic polynomial interpolation curve. MATLAB and CoppeliaSim are compared in a joint simulation experiment to determine the algorithm's efficiency. During the motion of the manipulator, the end-effector position, angular velocity, angular acceleration, and moment are smooth and continuous, with no abrupt acceleration or deceleration; this prevents mechanical shock and effectively reduces the impact of the mechanical system. The initial and final positions can guarantee that the angular velocity and acceleration are both zero. This resolves the problem that the 3–5–3 algorithm has abrupt changes in angular acceleration at the initial and final points. The comparison shows that the intermediate transition curve of the 5-B-5 algorithm is demonstrably smoother. It provides theoretical support and references for enhancing the manipulator control system's application of the segmented spline interpolation algorithm. In this paper, we use YouBot's 5dof manipulator to get ready for a future study on planning the path of the mobile manipulator.

Acknowledgements

Thanks to Professor Yang Yuanchao's National Natural Science Foundation of China Youth Science Foundation (62003257) support.

References

[1] Chettibi T. Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions [J]. Robotica, 2019, 37(3):539-559.
[2] Piegl L A, Tiller W. The NURBS book [M]. Springer Berlin Heidelberg,1997.
[3] Liu Bao, Di Xin, Han Lihua. Application of improved cubic spline interpolation in manipulator trajectory planning [J]. Mechanical Science and technology, 2021, 40(8) : 1158-1163.
[4] Wu G, Zhao W, X Zhang. Optimum Time-Energy-Jerk Trajectory Planning for Serial Robotic Manipulators by Reparameterized Quintic NURBS Curves [J]. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, 2020.
[5] Saravanan R, Ramabalan S, Sriram P, et al. Non-uniform rational B-spline-based minimum cost trajectory planning for autonomous robots[J]. International Journal of Intelligent Systems Technologies & Applications, 2010, 9 (2):121-14.
[6] LIN Menghao, ZHANG Lei, LI Pengfei, WANG Xiaohua. Trajectory planning of segmented manipulator based on complex trajectory[J].Journal of Xi'an Polytechnic University,2020,34(4):43-50.doi:10.13338/j.issn.1674-649x.2020.04.008
[7] K. Miao, Z. Zhou and Y. Zhang, "Trajectory Planning And Simulation Of Educational Robot Based On ROS," 2019 2nd International Conference of Intelligent Robotic and Control Engineering (IRCE), 2019, pp. 18-22, doi: 10.1109/IRCE.2019.00011.
[8] L. Wang, Q. Wu, F. Lin, S. Li and D. Chen, "A New Trajectory-Planning Beetle Swarm Optimization Algorithm for Trajectory Planning of Robot Manipulators," in IEEE Access, vol. 7, pp. 154331-154345, 2019, doi: 10.1109/ACCESS.2019.2949271.
[9] A. T. Khan, S. Li, S. Kadry and Y. Nam, "Control Framework for Trajectory Planning of Soft Manipulator Using Optimized RRT Algorithm," in IEEE Access, vol. 8, pp. 171730-171743, 2020, doi: 10.1109/ACCESS.2020.3024630.
[10] Mingming Wang, Jianjun Luo, Jing Fang, Jianping Yuan, Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm, Advances in Space Research, Volume 61, Issue 6, 2018, Pages 1525-1536, ISSN 0273-1177.