

Article

# Analysis and Simulation of Polishing Robot Operation Trajectory Planning

Xinhong Zeng \* and Yongxiang Wang

School of Low-Altitude Equipment and Intelligent Control, Guangzhou Maritime University, Guangzhou 510725, China; gzmcyx@gzmtu.edu.cn

\* Correspondence: zxh5@gzmtu.edu.cn

**Abstract:** Trajectory planning is essential for robotic polishing tasks, as the effectiveness of this planning directly influences the quality of the work and the energy efficiency of the operation. This study introduces an innovative trajectory planning method for robotic polishing tasks, focusing on the development and application of quintic B-spline interpolation. Recognizing the critical impact of trajectory planning on the quality and energy efficiency of robotic operations, we analyze the structure and parameters of the ABB-IRB120 robot within a laboratory setting. Using the Denavit–Hartenberg parameter method, a kinematic model is established, and the robot’s motion equations are derived through matrix transformation. We then propose a novel approach by implementing both fifth-degree polynomial and quintic B-spline interpolation algorithms for planning the robot’s spatial spiral arc trajectory, which is a key contribution of this work. The effectiveness of these methodologies is validated through simulation in MATLAB’s robotics toolbox. Our findings demonstrate that the quintic B-spline interpolation not only significantly improves task precision but also optimizes energy consumption, making it a superior method for trajectory planning in robotic grinding applications. By integrating advanced interpolation techniques, this study provides substantial technological and environmental benefits, offering a groundbreaking reference for enhancing the precision and efficiency of robotic control systems.

**Keywords:** robot; grinding; trajectory planning



Academic Editor: Roberto Montemanni

Received: 20 November 2024

Revised: 4 January 2025

Accepted: 15 January 2025

Published: 18 January 2025

**Citation:** Zeng, X.; Wang, Y. Analysis and Simulation of Polishing Robot Operation Trajectory Planning. *Algorithms* **2025**, *18*, 53. <https://doi.org/10.3390/a18010053>

**Copyright:** © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the development of modern science and technology, robots are widely used in various production industries, which greatly improves the efficiency of industrial production. In the realm of manufacturing, the precision and efficiency of grinding robots are largely contingent upon the optimization of their trajectory planning. The core scientific problem addressed in this study is the optimization of trajectory planning to enhance both the precision and energy efficiency of robotic grinding operations. Trajectory planning is a pivotal aspect of robot motion control, encompassing the generation and optimization of motion paths for the robot’s end-effector (e.g., a grinding tool) to facilitate efficient and precise grinding operations. This involves addressing the scientific challenge of optimizing both the accuracy and the energy consumption of the robotic operations. The quality of trajectory planning affects the movement effect of the robot and then affects the quality of grinding. Trajectory planning is the prerequisite for achieving optimal performance in grinding robots. Linear and arc trajectory planning are two basic trajectory planning methods. Because the planned trajectory is relatively simple, the application surface is narrow.

Traditional trajectory planning techniques have many drawbacks, such as not finding a global solution, requiring gradients, not being used for discontinuous functions, etc. [1]. More and more scholars adopt intelligent algorithms in path planning, such as the particle swarm optimization (PSO) [2], genetic algorithm [3,4], ant colony algorithm [5], artificial bee colony algorithm [6,7], and so on. These intelligent algorithms aim to overcome the limitations of traditional methods and address the core scientific challenge of enhancing trajectory precision while reducing energy consumption.

Fang et al. [8] use an improved B-spline interpolation method to optimize the joint trajectory of the welding robot and verify the feasibility of this method through simulation experiments. It effectively enhances the smooth continuity of the angular values and angular velocity values of each joint's trajectory, making the robot's operation smoother and more stable [9]. Feng [10] and Zhou [11] perform kinematic analysis on the joints of the ABB IRB140 grinding robot, establish a robot model using an enhanced DH method, and deduce both forward and inverse kinematic equations through the analysis of the robot's geometric relationships and joint variables. Trajectory planning is carried out using the quintic polynomial function method, yielding improved joint trajectories and end-effector trajectory curves through simulation. In reference [12], Alaa Saadah et al. applied MATLAB to trajectory planning for the arc welding of a KUKA KR5 robot. The trajectory planning and simulation of the robot operation were realized utilizing the Jacobian matrix of linear paths and circular paths. The achievement of good results was tested through experimental setups. Wang Lei and others used a beetle swarm optimization algorithm for trajectory planning of robotic manipulators [3]. The beetle antennae search algorithm has distinct advantages in dealing with low-dimensional trajectory optimization problems. However, it may face efficiency issues in high-dimensional problems, such as trajectory planning for a six-axis robot, and its universality and robustness still need to be verified. Yunjeong and Byung Kook [13] propose a time-optimal trajectory planning algorithm for differential drive-wheeled mobile robots. By segmenting the trajectory and employing the bang-bang principle, it achieves rapid and safe mobility. The authors in [14–16] also propose a time-optimal trajectory planning algorithm for robot applications, achieving good results. Studies in the literature [6,11,17–20] made corresponding trajectory planning or optimization research for different robots, which achieved certain expected results.

Our study aims to further these efforts by addressing the dual challenges of optimizing trajectory precision and energy efficiency, particularly through the comparative study of fifth-degree polynomial and B-spline interpolation algorithms. This paper, with spiral arc trajectory as an example, through the simulation model in MATLAB R2021a/SIMULINK, aims to design the ABB-IRB120 six-axis robot model, build the robot space arc grinding trajectory motion model, take five polynomial interpolation algorithms and five B spline interpolation algorithms for comparative study, select the trajectory planning method to eliminate the grinding robot jitter in motion, and improve the quality of grinding.

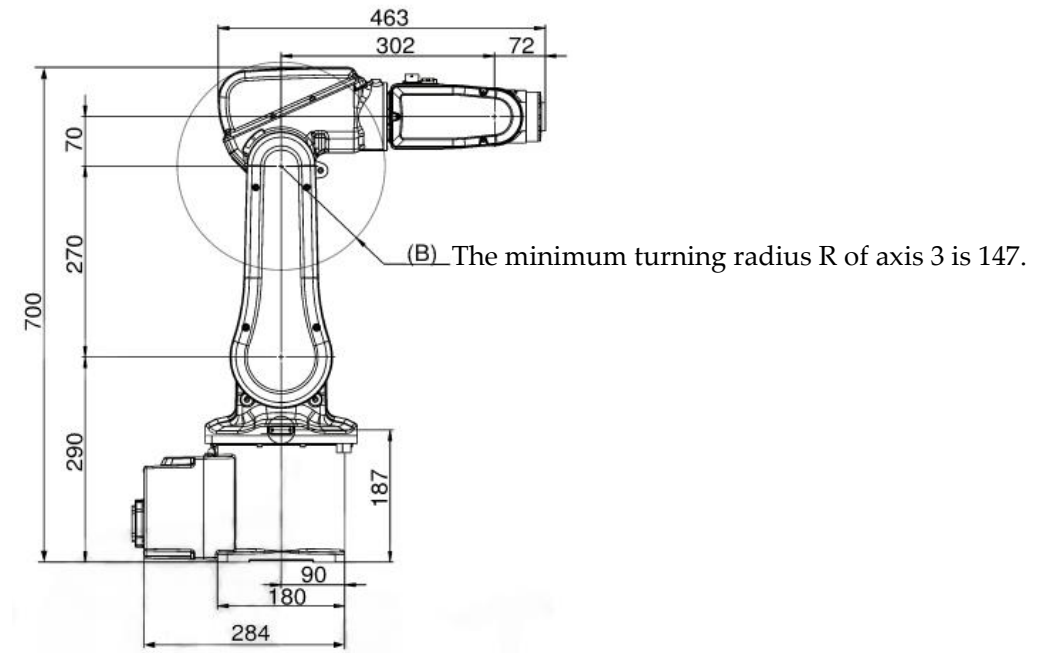
To provide a clearer outline of our work, this paper is organized as follows: In Section 2, we detail the establishment of the kinematic model for the ABB-IRB120 robot and describe our approach to modeling the spatial spiral curve trajectory. In Section 3, we discuss the application of quintic polynomial and B-spline interpolation algorithms, present the simulation results assessing their effectiveness and efficiency, and rigorously validate these methods through detailed comparative analyses. Finally, in Section 4 we summarize the findings and discuss the implications of our work, highlighting potential areas for future research.

## 2. Materials and Methods

### 2.1. The Establishment of the Robotic Arm Kinematics Model and the Spatial Spiral Curve Model

#### 2.1.1. Robot Kinematics and Modeling

The ABB-IRB120, as a compact six-axis industrial robot, has been widely adopted in the manufacturing industry due to its high precision, exceptional flexibility, and reliability. Figure 1 illustrates the planar schematic diagram of the ABB-IRB120 six-axis robot, which provides a clear visual representation of the robot’s structural composition and relationship between its various links. This schematic serves as a foundational basis for understanding the robot’s kinematic and dynamic characteristics.



**Figure 1.** Planar schematic of the ABB-IRB120 six-axis.

The link parameters of the ABB IRB120 robot are critical in defining its motion capabilities and performance. The link parameters for the ABB IRB120 polishing robot are referenced in the literature [8,21]. These parameters, such as link lengths, twist angles, and offset distances, are essential components of the robot’s kinematic model and are used to calculate its forward and inverse kinematics.

Table 1 presents the comprehensive DH (Denavit–Hartenberg) parameter table for the ABB-IRB120 six-axis robot. By using these parameters, it is possible to accurately determine the position and orientation of the robot’s end-effector relative to its base frame.

**Table 1.** The D-H parameters table for the ABB-IRB120 six-axis robot.

Link Rod $i$	Twist Angle $\alpha_{i-1}/\text{Rad}$	Link Length $a_{i-1}/\text{mm}$	Joint Angle $\theta_i/\text{rad}$	Link Offset $d_i/\text{mm}$	Other
1	0	0	$\theta_1$	$l_1$	$l_1 = 290 \text{ mm}$
2	$-\pi/2$	0	$\theta_2 - \pi/2$	0	—
3	0	$l_2$	$\theta_3$	0	$l_2 = 270 \text{ mm}$
4	$-\pi/2$	$l_3$	$\theta_4$	$l_4$	$l_3 = 70 \text{ mm}$ $l_4 = 302 \text{ mm}$
5	$\pi/2$	0	$\theta_5 + \pi$	0	—
6	$\pi/2$	0	$\theta_6$	$l_5$	$l_5 = 72 \text{ mm}$

In this setup, the coordinate system for the robotic arm’s links is established using the modified DH parameters method. When studying a generalized linkage system with  $n$  joints, the homogeneous transformation matrix between adjacent links is considered, considering the neighboring links and joint  $Axis_{i-1}$ ,  $Axis_i$ , and  $Axis_{i+1}$ , as shown in Figure 2.

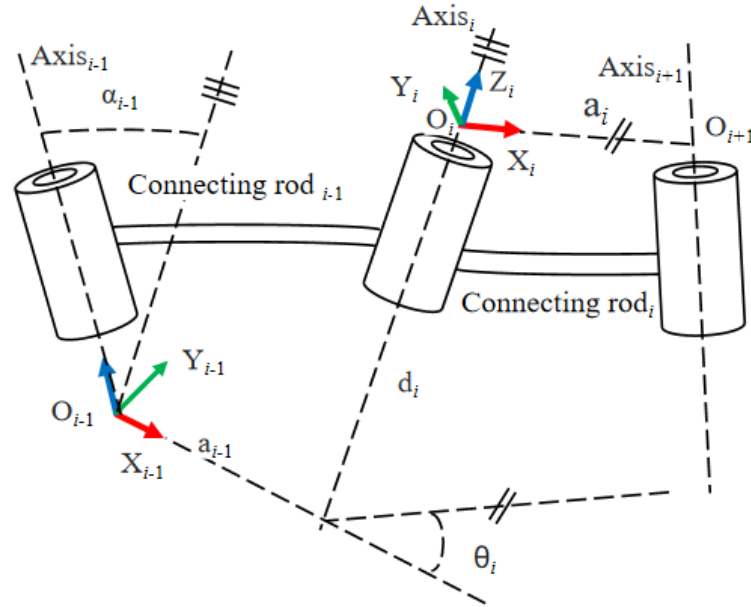


Figure 2. Schematic diagram of the parameters of the improved DH model.

The transformation matrix for the linkage of the improved DH coordinate system is as follows [10]:

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \cos\alpha_{i-1}\sin\theta_i & \cos\alpha_{i-1}\cos\theta_i & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\alpha_{i-1}\sin\theta_i & \sin\alpha_{i-1}\cos\theta_i & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where  $\alpha_i$ ,  $a_i$ , and  $\theta_i$  are the linkage parameters. Then, the positive kinematic equation of the robot is obtained from the pose transformation matrix of each joint of the robot [22]:

$${}^0T_6 = {}^0T_1(\theta_1) \cdot {}^1T_2(\theta_2) \cdot {}^2T_3(\theta_3) \cdot {}^3T_4(\theta_4) \cdot {}^4T_5(\theta_5) \cdot {}^5T_6(\theta_6) = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Here,

$$\begin{aligned} n_x &= s\theta_6[c\theta_1s\theta_4s(\theta_2 + \theta_3) - c\theta_4s\theta_1] - c\theta_6[c\theta_5s\theta_1s\theta_4 + c\theta_1s\theta_5c(\theta_2 + \theta_3) + c\theta_1c\theta_4c\theta_5s(\theta_2 + \theta_3)] \\ n_y &= s\theta_6[c\theta_1c\theta_4 + s\theta_1s\theta_4s(\theta_2 + \theta_3)] - c\theta_6[s\theta_1s\theta_5c(\theta_2 + \theta_3) - c\theta_1c\theta_5s\theta_4 + c\theta_4c\theta_5s\theta_1s(\theta_2 + \theta_3)] \\ n_z &= c\theta_6[s\theta_5s(\theta_2 + \theta_3) - c\theta_4c\theta_5c(\theta_2 + \theta_3)] + s\theta_4s\theta_6c(\theta_2 + \theta_3) \\ o_x &= c\theta_6[c\theta_1s\theta_4s(\theta_2 + \theta_3) - c\theta_4s\theta_1] + s\theta_6[c\theta_5s\theta_1s\theta_4 + c\theta_1s\theta_5c(\theta_2 + \theta_3) + c\theta_1c\theta_4c\theta_5s(\theta_2 + \theta_3)] \\ o_y &= c\theta_6[c\theta_1c\theta_4 + s\theta_1s\theta_4s(\theta_2 + \theta_3)] + c\theta_6[s\theta_1s\theta_5c(\theta_2 + \theta_3) - c\theta_1c\theta_5s\theta_4 + c\theta_4c\theta_5s\theta_1s(\theta_2 + \theta_3)] \\ o_z &= c\theta_6s\theta_4c(\theta_2 + \theta_3) - s\theta_6[s\theta_5s(\theta_2 + \theta_3) - c\theta_4c\theta_5c(\theta_2 + \theta_3)] \\ p_x &= c\theta_1[ls\theta_2 + c\theta_1c\theta_4s(\theta_2 + \theta_3)] \end{aligned}$$

$$\begin{aligned}
 p_y &= c\theta_1 c\theta_5 c(\theta_2 + \theta_3) - s\theta_5 [c\theta_4 s\theta_1 s(\theta_2 + \theta_3) - c\theta_1 c\theta_4] \\
 p_z &= -c\theta_5 s(\theta_2 + \theta_3) - c\theta_4 s\theta_5 c(\theta_2 + \theta_3) \\
 p_x &= c\theta_1 [l_2 s\theta_2 + l_3 s(\theta_2 + \theta_3) + l_4 c(\theta_2 + \theta_3) + l_5 c\theta_5 c(\theta_2 + \theta_3) - l_5 c\theta_4 s\theta_5 s(\theta_2 + \theta_3)] - l_5 s\theta_1 s\theta_4 s\theta_5 \\
 p_y &= s\theta_1 [l_2 s\theta_2 + l_3 s(\theta_2 + \theta_3) + l_4 c(\theta_2 + \theta_3) + l_5 c\theta_5 c(\theta_2 + \theta_3) - l_5 c\theta_4 s\theta_5 s(\theta_2 + \theta_3)] + l_5 c\theta_1 s\theta_4 s\theta_5 \\
 p_z &= l_1 + l_2 c\theta_2 + l_3 c(\theta_2 + \theta_3) - l_4 s(\theta_2 + \theta_3) - l_5 c\theta_5 s(\theta_2 + \theta_3) - l_5 c\theta_4 s\theta_5 s(\theta_2 + \theta_3)
 \end{aligned}$$

In the formulas,  $c\theta = \cos\theta$ ,  $s\theta = \sin\theta$ ,  $[n_x, n_y, n_z]^T$ ,  $[o_x, o_y, o_z]^T$ ,  $[a_x, a_y, a_z]^T$ ,  $[p_x, p_y, p_z]^T$  represent, respectively, the normal vector, orientation vector, approach vector of the end-effector in the base coordinate system, and the position vector of the end-effector in the base coordinate system [10]. After obtaining the above terminal pose matrix of the robot, we analyze the inverse kinematics of the robot and solve the motion variables of each joint of the robot, and the solution of  $\theta$  is obtained by the inverse kinematics as follows:

The first joint variable  $\theta_1$  can be obtained as follows:

$$\theta_1: \theta_1 = \text{atan2}(p_y - l_5 a_y, p_x - l_5 a_x) \tag{3}$$

The 2nd joint variable  $\theta_2$  can be obtained as follows:

$$\theta_2 = \text{atan2}(A_2, \pm \sqrt{1 - A_2^2}) - \phi_2 \tag{4}$$

In this formula,

$$A_2 = \frac{l_2^2 - l_3^2 - l_4^2 + [c\theta_1(p_x - l_5 a_x) + s\theta_1(p_y - l_5 a_y)]^2 + (p_z - l_1 - l_5 a_z)^2}{2l_2},$$

$$\phi_2 = \text{atan2}(p_z - l_1 - l_5 a_z, c\theta_1(p_x - l_5 a_x) + s\theta_1(p_y - l_5 a_y))$$

The third joint variable  $\theta_3$  can be obtained as follows:

$$\theta_3 = \text{atan2}(c\theta_1(p_x - l_5 a_x) + s\theta_1(p_y - l_5 a_y) - l_2 s\theta_2, p_z - l_1 - l_5 a_z - l_2 c\theta_2) - \theta_2 - \phi_3$$

in which

$$\phi_3 = \text{atan2}(l_4, l_3) \tag{5}$$

The fifth joint variable  $\theta_5$  can be obtained as follows:

$$\theta_5 = \text{atan2}(\pm \sqrt{1 - A_5^2}, A_5) \tag{6}$$

In this formula,

$$A_5 = (a_x c\theta_1 + a_y s\theta_1) c(\theta_2 + \theta_3) - a_z s(\theta_2 + \theta_3)$$

The fourth joint variable  $\theta_4$  can be obtained as follows:

$$\theta_4 = \text{atan2}(A_{42}, A_{41}) \tag{7}$$

In this formula,

$$A_{41} = \frac{-(a_x c\theta_1 + a_y s\theta_1) s(\theta_2 + \theta_3) - a_z c(\theta_2 + \theta_3)}{s\theta_5},$$

$$A_{42} = \frac{a_y c\theta_1 - a_x s\theta_1}{s\theta_5}$$

The sixth joint variable  $\theta_6$  can be obtained as follows:

$$\theta_6 = \text{atan2}(A_{62}, A_{61}) \tag{8}$$

In this formula,

$$A_{61} = \frac{-(n_x c\theta_1 + n_y s\theta_1)c(\theta_2 + \theta_3) + n_z s(\theta_2 + \theta_3)}{s\theta_5}$$

$$A_{62} = \frac{(o_x c\theta_1 + o_y s\theta_1)c(\theta_2 + \theta_3) - o_z s(\theta_2 + \theta_3)}{s\theta_5}$$

Taking the motion variables  $q = (\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{4}, \frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{4})$  as examples, the pose matrix of the end-effector can be obtained from Equation (2) as follows:

$${}^0T_6 = \begin{bmatrix} 0.5520 & 0.6310 & -0.5451 & 154.1186 \\ 0.7269 & -0.0439 & 0.6853 & 160.9804 \\ 0.4085 & -0.7745 & 0.4830 & 80.3997 \\ 0 & 0 & 0 & 1.0000 \end{bmatrix} \tag{9}$$

Using the end pose result of Equation (9), the  $q = \text{abb120\_ikine}(T)$  result is solved by the inverse kinematics equation as shown in Figure 3:

```

Command Window
>> Ikinе_test
    29.9998    59.9993    45.0002    89.9995    59.9974    45.0402
    
```

Figure 3. Inverse operation result graph.

By comparing the above selection of kinematic variables  $q = (\frac{\pi}{6}, \frac{\pi}{3}, \frac{\pi}{4}, \frac{\pi}{2}, \frac{\pi}{3}, \frac{\pi}{4})$  in forward kinematics, the results of inverse kinematics are not significantly different from the selection in forward kinematics, indicating the correctness of the established kinematics equations.

The robot multi-axis linkage angle value can be obtained through the above formula. The multi-axis linkage angle values are shown in Table 2.

Table 2. Multi-axis linkage angle values.

$\theta_1$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$
52.7652	21.8561	18.1634	-116.0455	62.3940	-10.7425
60.7978	29.3186	6.4654	-106.6156	65.5352	-18.9757
84.7266	6.5544	34.0268	-93.3423	85.9207	-22.0362
59.7385	-9.7148	49.1090	-112.4336	68.9605	-15.9549
52.7652	19.2235	15.5904	-113.4557	60.2128	-16.1320

Utilizing the robotics toolbox, a three-dimensional simulation model of the ABB-IRB120 robot is constructed in MATLAB, and the simulation robot workspace is created by the Monte Carlo method. The simulation model of the robot workspace and grinding curve is shown in Figure 4. This integrated visualization offers an intuitive understanding of the robot trajectory planning.

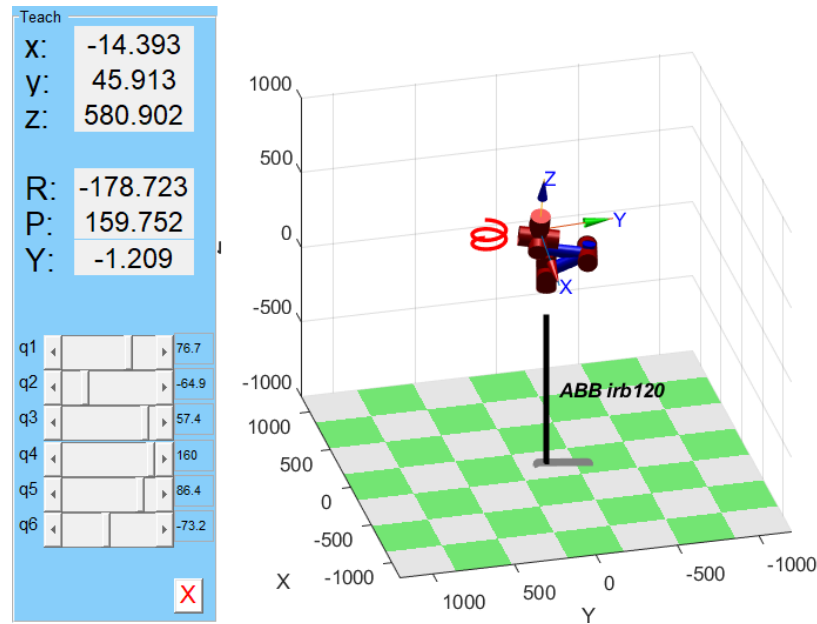


Figure 4. Simulation model of robot and grinding curve.

### 2.1.2. Space Spiral Arc Curve Model

Suppose the radius of the spatial spiral circular arc curve is  $r = 100$  mm, the pitch is  $p = 60$  mm, and the inclination angle is  $\varphi_0$ .

Assuming that the  $z$ -axis of the spatial coordinate system  $w$  for the space spiral line to be polished coincides with the axis of the helix [9], and the spatial coordinate origin is  $(x_t, y_t, z_t) = (0, 0, 0)$ , the parametric equations of the spiral in three-dimensional space are as follows:

$$\begin{cases} x = x_t + r * \cos(\frac{2\pi}{p} * \phi) \\ y = y_t + r * \sin(\frac{2\pi}{p} * \phi) \\ z = z_t + \phi \end{cases} \quad (10)$$

Spiral inclination angle  $\varphi_0 = \text{atan2}(\frac{p}{2\pi * r})$ , where  $\varphi$  is the variable parameter in Formula (10),  $\varphi$  increases from 0 equal step distance to  $2p$ , and the parameter Equation (10) can generate the spatial spiral arc curve of the corresponding pitch  $p$ .

Suppose that the spatial coordinate system  $w$  of the spatial helix to be polished is relative to the base coordinate frame  $o$ :

$${}^o_w T = \begin{bmatrix} 1 & 0 & 0 & 200 \\ 0 & 1 & 0 & 300 \\ 0 & 0 & 1 & 400 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

In the spiral parameter equation given in Equation (10), the parametric variable  $\varphi$  is assigned every  $p/n$ , starting from 0 and ending when the final value is  $2p$ . Then, obtain the junction position coordinates on the spatial spiral curves of the two pitches  $(x_{w(i)}, y_{w(i)}, z_{w(i)})$ . The total number is  $2n + 1$ , and we can obtain the following position matrix for the  $2n + 1$  nodes:

$$T_w(:, :, i) = \begin{bmatrix} 1 & 0 & 0 & x_{w(i)} \\ 0 & 1 & 0 & y_{w(i)} \\ 0 & 0 & 1 & z_{w(i)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (12)$$

Given  ${}^o_w T$  and  $T_w(:, :, i)$ , the position matrix of selected nodes on the spatial helix in the actual workpiece coordinate system can be obtained from the formula  ${}^o_w T$  and  $T_w(:, :, i)$ .

$$T_o(:, :, i) = \begin{bmatrix} 1 & 0 & 0 & x_{o(i)} \\ 0 & 1 & 0 & y_{o(i)} \\ 0 & 0 & 1 & z_{o(i)} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

Through Formula (13), obtain the position of each node, connect the end position, and form the model of the spatial helical curve.

Then, solve the joint trajectory  $q$  (position),  $\dot{q}$  (velocity), and  $\ddot{q}$  (acceleration) of the robot when passing through each node of the spatial solid curve.

In this paper, we solve the problem of mapping the velocity of a robot in Cartesian space to the velocity in joint space using the vector product method. The displacements of the robot in Cartesian space  $x$  and joint space  $q$  can be represented by the following equations:

$$x = x(q) \tag{14}$$

After taking the derivative of both sides of Equation (14) concerning time, the relationship between the velocity of the robot in Cartesian space  $x$  and the velocity in joint space  $q$  is obtained as follows:

$$\dot{x} = J(q)\dot{q} \tag{15}$$

In the above formula,

$\dot{x}$  represents the velocity of the robot's end-effector in Cartesian space;

$\dot{q}$  is the joint velocity of the robot in the joint space;

$J(q)$  represents the Jacobian matrix for robots.

Multiplying the left-hand side of both sides of Equation (15) by  $J(q)^{-1}$ , we can obtain the formula for solving the joint speeds of the robot as follows:

$$\dot{q} = J(q)^{-1}\dot{x} \tag{16}$$

The velocity of the robot's end-effector in Cartesian space is a 6-dimensional column vector, which is  $\dot{x} = [v_x, v_y, v_z, w_x, w_y, w_z]^T$ . Assuming that the velocity of the robot's end-effector in the z-axis direction is uniform and the velocity value is  $v_z = 2 \text{ mm/s}$ , the remaining velocity components of the  $i$ -th node can be expressed as

$$\begin{cases} v_x(i) = w_z(i) * r * \sin((i - 1) * \alpha_c) \\ v_y(i) = w_z(i) * r * \cos((i - 1) * \alpha_c) \\ w_x(i) = 0 \\ w_y(i) = 0 \\ w_z(i) = (2 * \pi) / (\frac{p}{v_z}) \end{cases} \tag{17}$$

Here,  $\alpha_c$  represents the step angle.

Thus, the operational speed of the robot,  $x$ , has been determined. Furthermore, from Equation (16), we can obtain the joint speed.

By taking the derivative of both sides of Equation (15), we can obtain the robot joint acceleration equation:

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)\dot{q} \tag{18}$$



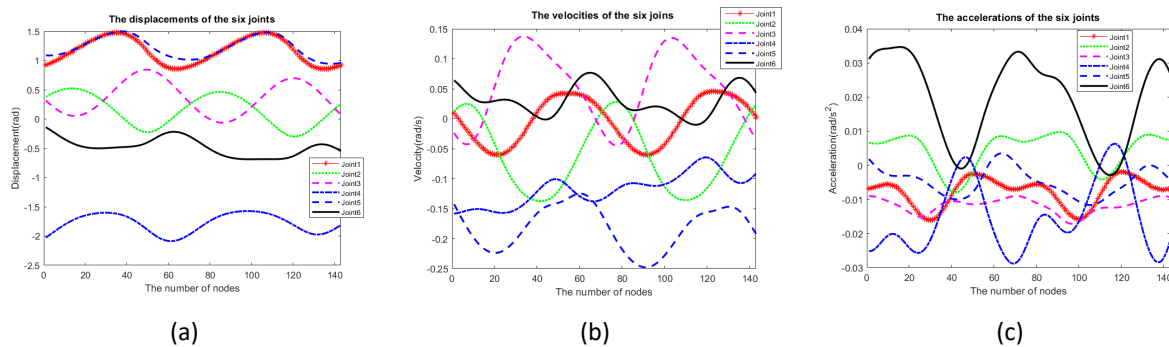
The following equation can be obtained from (18):

$$\ddot{\mathbf{q}} = J(q)^{-1} \left[ \ddot{\mathbf{x}} - \dot{J}(q)\dot{\mathbf{q}} \right] \tag{19}$$

Because the above assumes that the end speed of the robot is uniform in Cartesian space, it can be obtained as follows:

$$\ddot{\mathbf{x}} = [0, 0, 0, 0, 0, 0]^T \tag{20}$$

By combining Equations (18)–(20), we can solve for  $\ddot{\mathbf{q}}$ . Through the calculations above and by using MATLAB for programming, we can obtain the waveforms of displacement, velocity, and acceleration for each joint, as shown in Figure 5.



**Figure 5.** The waveforms of displacement, velocity, and acceleration for each joint. (a) Simulation curves of each joint displacement. (b) Simulation graphs for the velocity of each joint. (c) The acceleration simulation graphs for each joint.

### 2.2. Spiral Arc Trajectory Planning of Quintic Polynomial Interpolation Algorithm

Polynomial interpolation algorithms are frequently applied in trajectory planning, especially in robotics and other fields requiring smooth and precise path control. As the degree of the polynomial increases, the interpolation curve can approximate given data points more accurately, but this also introduces higher computational complexity and potential numerical instability. The quintic polynomial interpolation algorithm is a commonly used method in robot trajectory planning, as it can maintain trajectory smoothness and accuracy while relatively balancing computational complexity. It is particularly suited for planning robotic joint trajectories with continuous position, velocity, acceleration, and jerk (rate of change in acceleration), ensuring a smoother and more controlled movement. This complex algorithm utilizes the ability of polynomial functions to effectively model the complex relationship between joint displacement and time. By uniquely determining the coefficients of the polynomial based on specified boundary conditions, including the initial and final positions, velocities, and accelerations of the joint, corresponding trajectory plans can be tailored to the specific requirements of the robot and its tasks.

In the context of the quintic polynomial approach, the relationship between the joint displacement (which represents the change in position or angle of a particular joint over time) and time is expressed mathematically through a polynomial equation of the fifth degree. This equation encapsulates not just the instantaneous position of the joint at any given time but also its rate of change (i.e., velocity) and acceleration, which is beneficial to understanding the movement of the joint in its entire trajectory.

In the five-degree polynomial algorithm, the relationship between joint displacement and time is as follows:

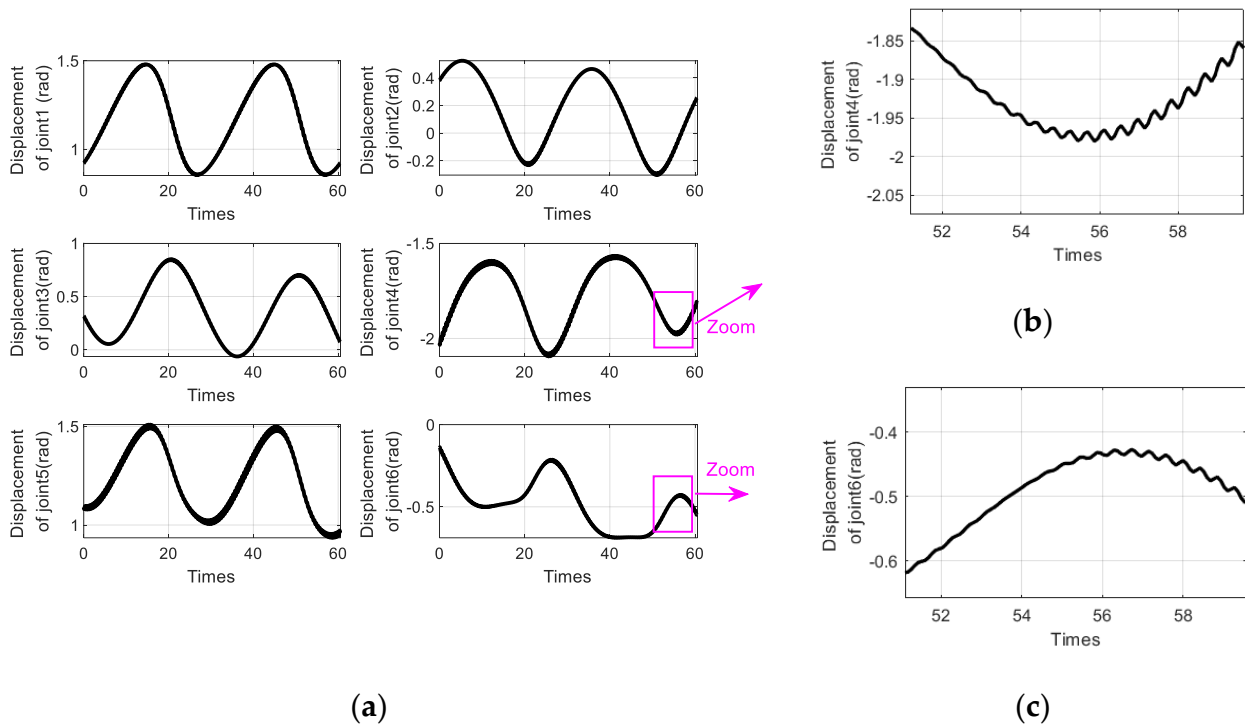
$$\theta(t) = a_0 + a_1t + a_2t^2 + a_3t^3 + a_4t^4 + a_5t^5 \tag{21}$$

For the motion trajectory of this design, for each node, only the first node, namely the time  $t_0$  of the starting point, is zero, and the time  $t_0$  of the other nodes is not zero. Therefore, the following equation can be obtained:

$$\begin{cases} \theta(0) = \theta_0 = a_0 + a_1t_0 + a_2t_0^2 + a_3t_0^3 + a_4t_0^4 + a_5t_0^5 \\ \theta(f) = \theta_f = a_0 + a_1t_f + a_2t_f^2 + a_3t_f^3 + a_4t_f^4 + a_5t_f^5 \\ \dot{\theta}(0) = \dot{\theta}_0 = a_1 + 2a_2t_0 + 3a_3t_0^2 + 4a_4t_0^3 + 5a_5t_0^4 \\ \dot{\theta}(f) = \dot{\theta}_f = a_1 + 2a_2t_f + 3a_3t_f^2 + 4a_4t_f^3 + 5a_5t_f^4 \\ \ddot{\theta}(0) = \ddot{\theta}_0 = 2a_2 + 6a_3t_0 + 12a_4t_0^2 + 20a_5t_0^3 \\ \ddot{\theta}(f) = \ddot{\theta}_f = 2a_2 + 6a_3t_f + 12a_4t_f^2 + 20a_5t_f^3 \end{cases} \tag{22}$$

In the equations,  $\theta(0), \dot{\theta}(0), \ddot{\theta}(0)$  and  $\theta(f), \dot{\theta}(f), \ddot{\theta}(f)$ , respectively, represent the joint angle, angular velocity, and angular acceleration at the starting moment and ending moment. By computational solving, the values of polynomial coefficients  $a_0, a_1, a_2, a_3, a_4,$  and  $a_5$  can be obtained.

Figure 6 shows the trajectory plots for each joint of the robot, achieved using the quintic polynomial interpolation algorithm.



**Figure 6.** The trajectory diagrams of each joint obtained using a quintic polynomial interpolation algorithm. (a) Displacement Curve diagram for six joint. (b) Enlarged view of displacement curve for Joint 4. (c) Enlarged view of displacement curve for Joint 6.

Figure 6 presents a visualization of the motion trajectories executed by the various joints of a robotic system, employing the quintic polynomial interpolation algorithm as its

methodology. This algorithm is renowned for its ability to generate smooth and continuous motion profiles, making it a popular choice in robotics applications.

Subfigure (a) of Figure 6 specifically showcases the displacement plots for all six joints of the robot over the course of its programmed movement. These plots offer a clear overview of how each joint contributes to the overall motion of the robot, revealing the interplay between them as they work in harmony to achieve the desired task.

To gain a deeper insight into the fine-grained behavior of specific joints, subfigures (b) and (c) zoom in on certain segments of the displacement curves for joint 4 and joint 6, respectively. While the quintic polynomial interpolation algorithm is designed to ensure a smooth and seamless trajectory, upon closer inspection, one can observe minute fluctuations in the form of a sawtooth pattern within these magnified segments.

These subtle fluctuations, though seemingly insignificant, can have a non-negligible impact on the robot's motion stability and precision. In precision-critical applications, such as assembly lines, surgical robots, or any scenario where even the slightest deviation from the intended path can lead to errors or damage, these fluctuations become a matter of concern.

Therefore, it is of great importance for us to study and formulate strategies to mitigate or eliminate these sawtooth patterns. Potential solutions may include refining the boundary conditions used in quintic polynomial interpolation, implementing additional smoothing algorithms, or exploring alternative trajectory planning methods that are inherently more stable and accurate. By solving these minor fluctuations, the overall performance and reliability of the robotic system can be significantly improved.

### *2.3. The Trajectory Planning of the Spiral Arc Curve Based on the Quintic B-Spline Interpolation Algorithm*

The quintic B-spline interpolation algorithm is an advanced and versatile technique for crafting precise polynomial curves in numerous applications, particularly in the realm of robotics and computer-aided design. This sophisticated method employs a quintic polynomial to meticulously interpolate through a predefined set of control points, allowing for the creation of smooth and continuous trajectories.

One of the most noteworthy characteristics of the quintic B-spline interpolation is its ability to ensure continuity up to the fifth-order derivatives of the generated curve. This feature is crucial in achieving ultra-smooth transitions between segments, eliminating abrupt changes or jerks that could otherwise compromise the precision and efficiency of the system. The continuity of these higher-order derivatives also ensures that the overall shape of the curve remains predictable and manageable, facilitating the precise control of the trajectory.

Furthermore, the quintic B-spline algorithm boasts a remarkable property known as local control. This means that adjustments made to individual control points directly affect only the local segment of the curve around that point, without disturbing the global shape significantly. This localized influence is invaluable in robot path planning, where even minor modifications to the trajectory can have significant implications on the robot's movement and performance. By enabling fine-tuned adjustments without compromising the integrity of the overall path, the quintic B-spline interpolation algorithm offers a high degree of flexibility and adaptability in designing efficient and optimal robot paths.

As we know, the quintic B-spline interpolation algorithm is a powerful tool that leverages the properties of a fifth-degree polynomial to construct smooth, continuous curves with controlled higher-order derivatives. Its local control feature further enhances its suitability for complex applications such as robot path planning, enabling precise and efficient trajectory planning in dynamic environments.

In addition to its local control and continuity properties, the quintic B-spline also falls into a broader category of B-spline curves characterized by their order. The  $k$ -th B-spline

has the characteristics of order  $k-1$ , and the quintic B-spline curve has the characteristics of fourth order. The  $k$ -th B-spline curve can be represented by segmented polynomials:

$$P(t) = \sum_{i=0}^n P_i F_{i,k}(t) \quad t \in [t_{k-1}, t_{n+1}] \tag{23}$$

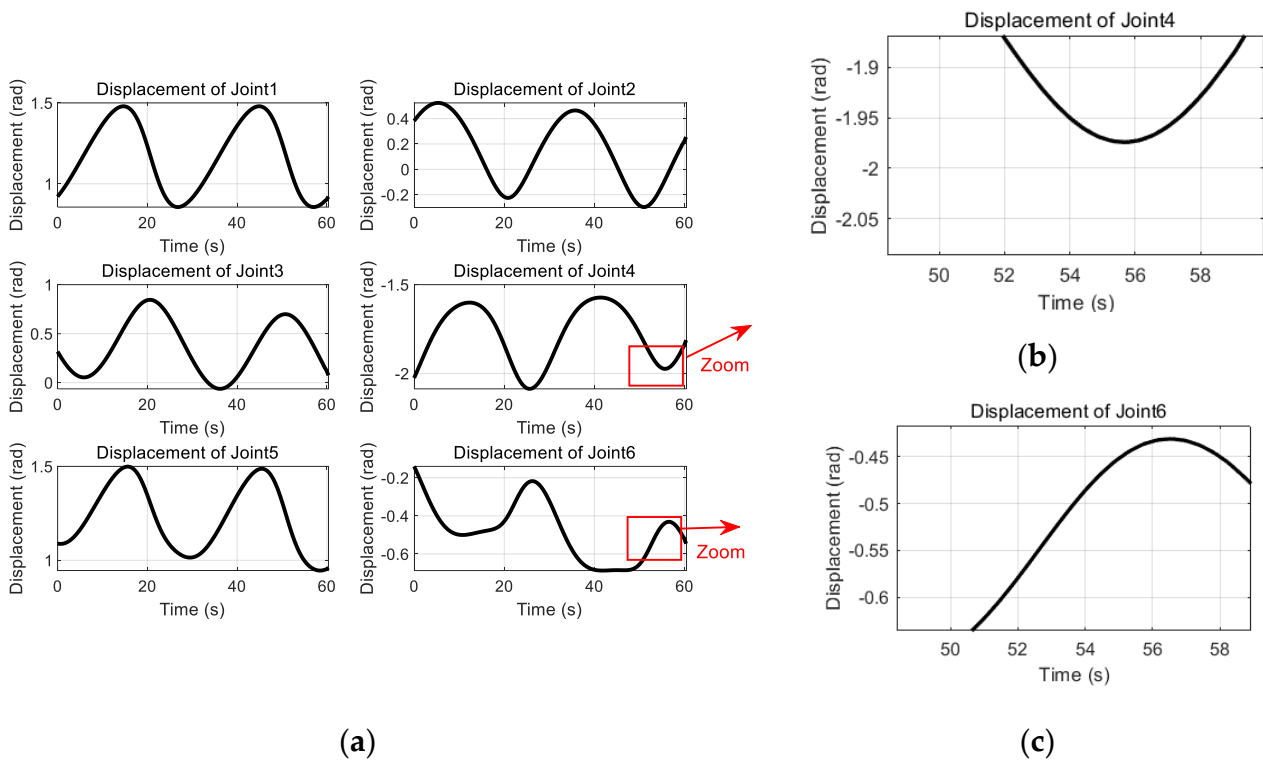
In the above equation,

$$\begin{cases} F_{i,0}(t) = \begin{cases} 1 & t_i \leq t \leq t_{i+1} \\ 0 & \text{others} \end{cases} \\ F_{i,k}(t) = \frac{t-t_i}{t_{i+k}-t_i} F_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} F_{i+1,k-1}(t) \\ \text{convention } \binom{0}{0} = 0 \end{cases}$$

Let the robot joint position point sequence be represented by  $(t_i, P_i)_{i=0, 1, \dots, n}$ , and the total movement time is  $t = t_f - t_0$ . To ensure the consistency between the starting end of the B spline and its actual data point, the repetition degree at both ends was taken as  $k + 1 = 6$ . Therefore, the corresponding knot value  $u_{k+I} = u_{5+I}$  ( $I = 0, 1, \dots, n$ ) requires  $n + 5$  control vertices  $Q_i$  ( $I = 0, 1, \dots, n + 4$ ) and a corresponding junction point vector  $u = [u_0, u_1, \dots, u_{n+10}]$ .

Follow the steps of the quintic B-spline interpolation for solving the trajectory of a spatial spiral circular arc. First, calculate the knot vector  $u = [u_0, u_1, \dots, u_{n+10}]$ ; then, find  $n + 5$  control points  $Q_i$ . The  $Q$  value of the control vertices can be obtained according to He et al. [22].

Figure 7 shows the trajectory plots of each joint of the robot achieved by using the quintic B-spline interpolation algorithm.



**Figure 7.** The trajectory diagrams of each joint obtained using a quintic B-spline interpolation algorithm. (a) Displacement Curve diagram for six joint. (b) Enlarged view of displacement curve for Joint 4. (c) Enlarged view of displacement curve for Joint 6.

Figure 7a displays the displacement diagrams of the robot’s six joints, which were obtained using the quintic B-spline interpolation algorithm. In Figure 7b, by zooming in on a specific segment of joint 4’s displacement, we can observe the remarkable level of detail

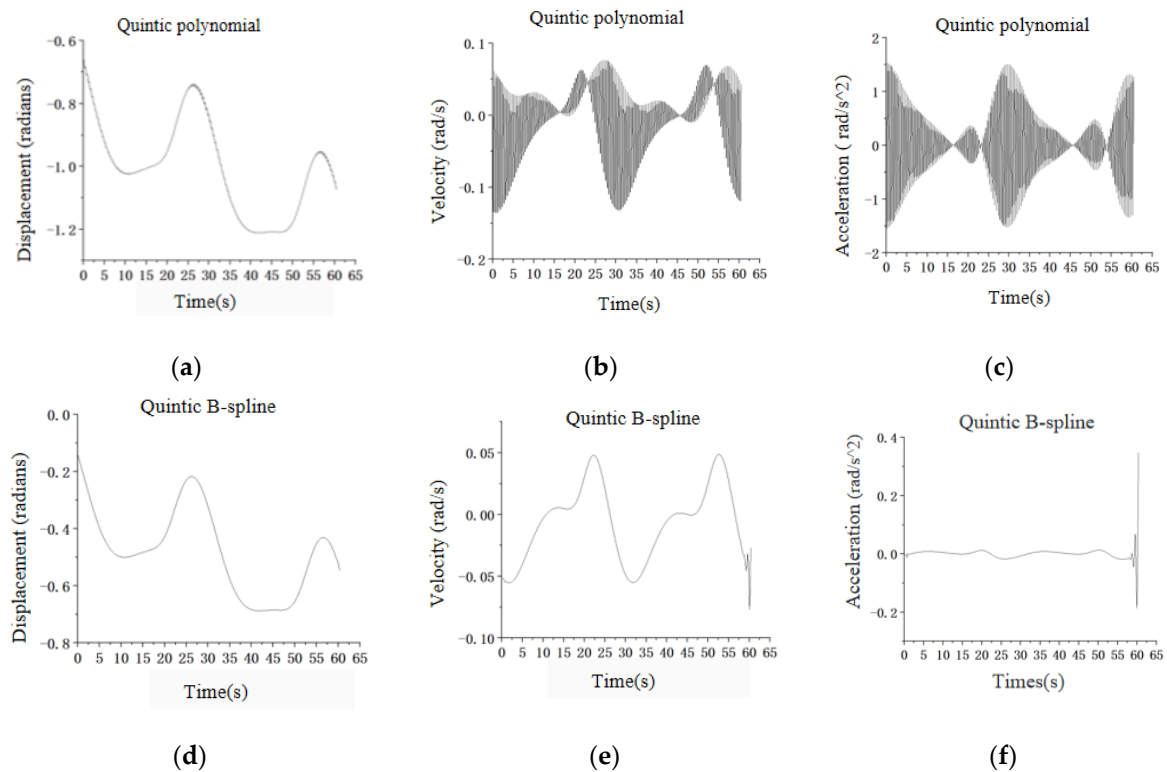
in the joint trajectory planning. The curve exhibits an exceptional degree of smoothness, devoid of any jagged edges or abrupt changes that could potentially compromise the robot's performance. This high level of precision ensures that the joint movements are not only accurate but also fluid, contributing significantly to the robot's overall effectiveness and responsiveness.

Similarly, Figure 7c, which focuses on a portion of joint 6's displacement, also demonstrates the smoothness of the curve, further reinforcing the advantages of the B-spline approach. The seamless transition between displacement points, without any abrupt changes or distortions, showcases the algorithm's ability to generate highly accurate and predictable motion trajectories. This is crucial for effective robot trajectory planning.

Furthermore, the B-spline interpolation algorithm's adaptability and flexibility make it an ideal choice for handling complex robot motions. Its ability to approximate curves with high accuracy and minimal computational overhead allows for real-time implementation, enabling robots to quickly respond to changes in their environment or adjust their movements on the fly [23]. This capability is essential for robots operating in dynamic and unpredictable environments, where swift decision-making and precise execution are of utmost importance.

In conclusion, the displacement diagrams presented in Figure 7, particularly the zoomed-in views of joints 4 and 6, serve as compelling evidence of the superiority of the B-spline interpolation algorithm in achieving superior interpolation effects and generating smooth, predictable motion profiles for robotic systems.

Using the B-spline interpolation algorithm, taking joint 6 as an example, the simulation curves of joint displacement, angular velocity, and angular acceleration are obtained through MATLAB programming, as shown in Figure 8d–f.



**Figure 8.** Comparison of joint 6 motion characteristics using quintic polynomial and quintic B-spline interpolation techniques. (a–c) depict the angular displacement, velocity, and acceleration profiles, respectively, using a quintic polynomial interpolation. (d–f) show the same motion characteristics using a quintic B-spline interpolation, demonstrating smoother and more continuous motion profiles.

### 3. Comparison of Trajectory Planning Between Quintic Polynomial Interpolation Algorithm and Quintic B-Spline Interpolation Algorithm

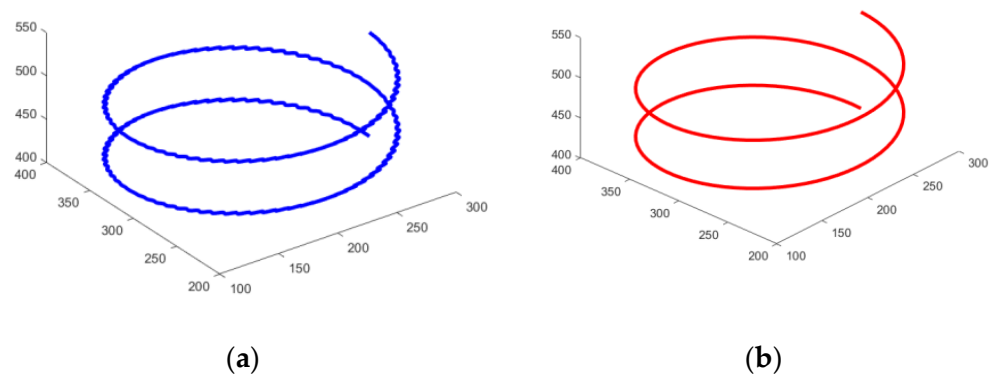
To accurately capture the dynamics of each robotic joint, MATLAB was used as the primary tool to compute the crucial kinematic parameters: angular displacement, angular velocity, and angular acceleration. These computations are essential for understanding the behavior of the robotic system and ensuring smooth, predictable movements. We utilized line charts to depict the relationships between these variables and time, providing a clear and intuitive representation of the data trends.

We focused on comparing the performance of two interpolation methods: the conventional fifth-degree polynomial interpolation and the more advanced fifth-degree B-spline interpolation algorithm. To illustrate the differences, we chose joint 6 as a representative example and plotted its angular displacement, angular velocity, and angular acceleration over time for both interpolation methods, as shown in Figure 8.

The simulated curve graphs in Figure 8a–c represent the joint displacement, angular velocity, and angular acceleration of joint 6, which were obtained using MATLAB programming with a quintic polynomial interpolation algorithm.

Figure 8 presents a detailed comparison of two interpolation techniques, highlighting notable differences in the motion trajectories' smoothness and continuity. The quintic B-spline interpolation algorithm notably provides a smoother and more natural motion profile for joint 6, affecting its angular displacement, velocity, and acceleration. This superior performance is attributed to the B-spline curves' inherent smoothness and local control capabilities, allowing precise motion adjustments and reducing oscillations or discontinuities.

Consequently, we chose the quintic B-spline interpolation algorithm for planning the spiral arc grinding trajectory, enhancing the robotic system's precision, reducing vibrations, and boosting overall performance. With these benefits, B-spline interpolation is ideal for complex, high-precision tasks, improving the robotic grinding process's reliability and efficiency. Regarding the robot's movement, the end-effector maintains a constant speed, ensuring consistent time intervals between consecutive nodes. With a radius of 100 mm and a pitch of 60 mm, we calculated an initial inclination angle of  $\varphi_0 = 0.0952$  and a maximum step angle of  $\alpha_c \leq 0.089$ . The spatial segmentation includes 71 nodes, with a set velocity of 2 mm/s along the z-axis, resulting in a time of 0.4258 s between nodes. Figure 9 shows the corresponding end position values of the robot.



**Figure 9.** Terminal trajectory of the robot. (a) Quintic polynomial interpolation for end-effector trajectory of the robot. (b) Quintic B-spline interpolation for the end-effector trajectory of the robot.

As shown in Figures 8 and 9, it becomes evident that the adoption of the quintic B-spline interpolation algorithm for trajectory planning yields significant advantages in terms of smoothness and stability. Specifically, Figure 8 not only showcases the joint trajectory but also underscores the remarkable smoothness in the profiles of joint velocity and acceleration along the trajectory. This absence of any sharp or abrupt changes is crucial as it ensures



that the robot's movements are not only precise but also free from any jarring motions that could potentially damage the mechanical components or compromise the accuracy of the task being performed.

Furthermore, the visualization in Figure 9 underscores the effectiveness of utilizing B-spline interpolation for trajectory planning by demonstrating the smooth and continuous motion path traced by the robot's end-effector. This seamless progression, devoid of any jerky movements, underscores the robustness and reliability of the planning method. The end position values presented in this figure serve as concrete evidence that the trajectory is not only well defined but also accurately executed, underscoring the high degree of control and predictability achieved through this approach.

Additionally, the smooth motion trajectories generated by the B-spline interpolation algorithm are particularly advantageous in applications requiring precise and delicate manipulation, such as in surgical robotics, where even the slightest deviation or instability can have significant consequences. By ensuring that the robot moves smoothly and without any large vibrations, the use of this interpolation technique significantly enhances the safety and effectiveness of such operations. Thus, we believe that integrating the quintic B-spline interpolation algorithm into trajectory planning strategies serves as a valuable tool for achieving smooth and stable robot movements. As demonstrated in Figures 8 and 9, this approach not only ensures precision in the execution of complex trajectories but also minimizes the risk of mechanical stress and wear, thereby enhancing the overall performance and longevity of the robotic system.

Moreover, the calculated MATLAB data were used to simulate the planning of a spatial spiral curve path for a space station polishing operation in RobotStudio 6.08, comparing three methods: no interpolation, quintic polynomial interpolation, and quintic B-spline interpolation. Enabling the simulation signal monitor in RobotStudio software, set up the signal monitor to record the total motor power and cumulative motor energy. Export the total motor power and cumulative motor energy for each of the three scenarios. Using Origin as an analytical tool, these data are processed and plotted on line charts as shown in Figures 10 and 11. Figure 10 compares the energy consumption of three interpolation methods: no interpolation, quintic polynomial, and quintic B-spline interpolation. It is observed that no interpolation consumes the least power, highlighting its efficiency despite the potential compromise in motion smoothness. Quintic polynomial interpolation, while providing smoother trajectories, requires significantly more power due to frequent dynamic adjustments in motor control. In contrast, quintic B-spline interpolation offers a balance, showing moderated energy consumption with improved trajectory smoothness. The data from Figure 11 shows that the energy consumption of the no-interpolation and quintic B-spline interpolation methods are comparable, highlighting that the advanced quintic B-spline interpolation achieves enhanced motion smoothness and accuracy without significantly increasing energy use compared to simpler or no-interpolation methods. This underscores the quintic B-spline's advantage of optimizing motion control while maintaining energy efficiency, making it an ideal choice for precision-demanding applications where both factors are crucial. In addition, we introduce a concise quantitative comparison for Figures 10 and 11 for clear evaluation: (1) No Interpolation: The total energy consumption for the no-interpolation method is 646.25275 watt-hours, which reflects the most energy-efficient approach among the three methods evaluated; (2) Quintic polynomial interpolation: This method shows the highest energy consumption at 805.16337 watt-hours, indicating that while it may offer smoother trajectories, it does so at the cost of higher energy use. (3) Quintic B-spline interpolation: The energy consumption for the quintic B-spline method is 630.00482 watt-hours, positioning it as slightly more energy-efficient

than the no-interpolation method, while potentially offering better control and smoothness in trajectory planning.

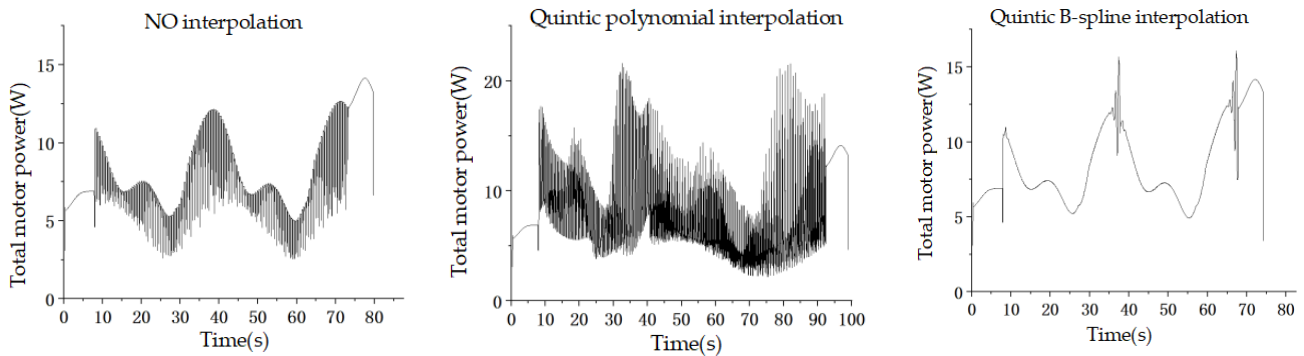


Figure 10. Total motor power for three results.

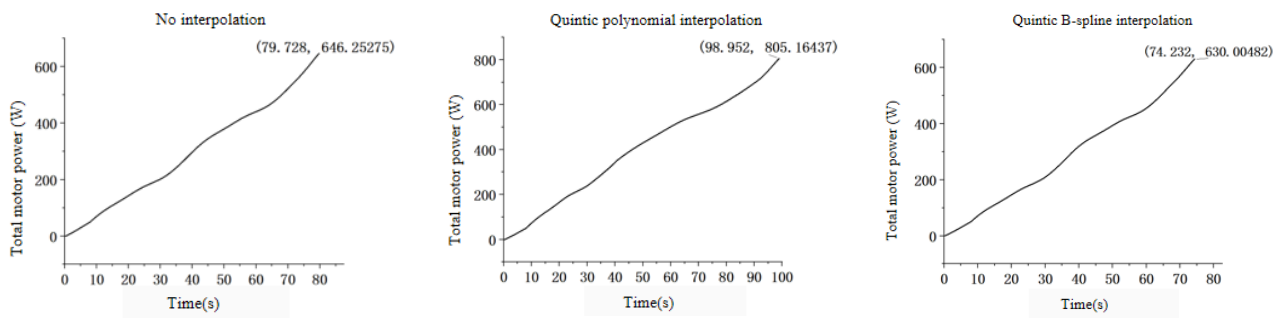


Figure 11. Total energy output of the motor for three different results.

From the comprehensive analysis of the graphical results, it becomes evident that when tasked with grinding a consistent spatial spiral curve, the quintic polynomial interpolation algorithm, while effective, exhibits certain limitations. Specifically, it necessitates the longest execution time, indicating a less efficient trajectory planning process. Furthermore, the total motor power curve exhibits the most pronounced fluctuations, suggesting that the robot’s motor system is subjected to increased stress and potentially higher wear rates. Consequently, this algorithm consumes the greatest amount of total motor energy, impacting operational costs and potentially reducing the robot’s overall energy efficiency.

In contrast, the quintic B-spline interpolation algorithm emerges as the optimal choice for this specific application. By consuming the least amount of time for trajectory planning, it significantly enhances the robot’s productivity and response speed. Moreover, the smooth variation in the total motor power curve demonstrates the algorithm’s capability to minimize stress on the motor system, promoting longer component lifespans and reduced maintenance requirements. This smoothness also translates into reduced energy consumption, making the robot’s operation more environmentally sustainable and cost-effective.

The planning effect of not using interpolation falls between the two algorithms. While this approach may suffice for certain applications with less stringent requirements, it fails to match the efficiency and precision offered by the quintic B-spline interpolation algorithm.

When the trajectory planning program derived from the B-spline interpolation algorithm is integrated into the robot experimental platform, the results are truly remarkable. The robot seamlessly follows the planned trajectory, executing the grinding task with stability and precision. Figure 12 is a diagram demonstrating the robot’s operation, which vividly illustrates the robot’s smooth and vibration-free operation and serves as a testament to the algorithm’s effectiveness.





**Figure 12.** Robot work demonstration diagram.

In conclusion, the adoption of the quintic B-spline interpolation algorithm for trajectory planning of robots engaged in spatial spiral curve grinding offers a trifecta of benefits: a shortened running time, reduced energy consumption, and the elimination of vibrations. These advantages not only elevate the robot's performance but also contribute to a more sustainable and cost-effective manufacturing process.

While our study focused exclusively on a spiral trajectory, we chose this due to its complexity and relevance in precision-demanding applications such as 3D printing and robotic assembly. To substantiate the choice and effectiveness of the interpolation methods used—quintic polynomial and B-spline—across various trajectory types, we refer to several studies that have successfully applied these techniques in broader contexts. For instance, Ref. [24] demonstrates the efficacy of B-spline interpolation in creating smooth and precise trajectories for industrial robots. Similarly, quintic polynomial interpolation's applicability and benefits across different trajectory shapes are well documented [25]. These references underscore the robustness and versatility of the interpolation methods utilized in our study, supporting their effectiveness beyond the specific case of spiral trajectories examined. Although our paper currently presents results for a single trajectory type, the foundational principles and previous successful applications of these interpolation techniques reinforce the validity of our approach and findings within a broader context.

#### 4. Conclusions

In this study, the application of quintic B-spline interpolation for spiral arc curve trajectory planning in robotic systems was demonstrated to effectively enhance motion smoothness and stability across various robot joints, leading to significantly improved performance outcomes. Our results validate that this method not only minimizes energy consumption but also effectively eliminates robot jitter, thereby enhancing the efficiency of robot trajectory planning. The quintic B-spline approach has proven to be a robust solution for maintaining high precision in complex tasks without compromising energy efficiency. Future work will continue to optimize this approach by focusing on reducing execution time, minimizing wear on robot joints to extend their lifespan, and enhancing obstacle avoidance capabilities. Additionally, integrating advanced sensing technologies such as vision systems and force sensors will enable dynamic adjustments to the trajectory in response to real-time environmental feedback, further ensuring task success under variable conditions. This comprehensive approach promises to significantly advance the field of robotic trajectory planning, particularly for precision-demanding tasks like robot grinding.

**Author Contributions:** Conceptualization, X.Z.; methodology, Y.W.; software development, X.Z.; validation, X.Z.; formal analysis, Y.W.; investigation and data curation, X.Z.; writing—original draft preparation, X.Z.; writing—review and editing, X.Z.; algorithm design and planning X.Z.; supervision, Y.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Acknowledgments:** We would like to express our gratitude to all the colleagues who provided valuable assistance and encouragement throughout the course of this research. Special thanks go to the technical support team for their indispensable contributions.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Wang, L.; Wu, Q.; Lin, F.; Li, S.; Chen, D. A New trajectory-planning beetle swarm optimization algorithm for trajectory planning of robot manipulators. *IEEE Access* **2019**, *7*, 154331–154345. [[CrossRef](#)]
2. Zhou, W. Research on Trajectory Planning of Robotic Arm Based on Particle Swarm Optimization. Master's Thesis, Nanjing University of Posts and Telecommunications, Nanjing, China, May 2022.
3. Chen, Y. Research on Trajectory Planning and Optimization of 6-DOF Manipulator. Master's Thesis, Changchun University of Technology, Changchun, China, June 2022.
4. Seddaoui, A.; Saaj, C.M. Collision-free optimal trajectory generation for a space robot using genetic algorithm. *Acta Astronaut.* **2021**, *179*, 311–321. [[CrossRef](#)]
5. Luo, Q.; Wang, H.; Zheng, Y.; He, J. Research on path planning of mobile robots based on improved ant colony algorithm. *Neural Comput. Appl.* **2020**, *32*, 1555–1566. [[CrossRef](#)]
6. Guo, B.; Tu, X. Robotic trajectory planning based on improved butterfly optimization algorithm. *Mech. Eng.* **2024**, *5*, 5–8.
7. Feng, Y.; Deb, S.; Wang, G.G.; Alavi, A.H. Monarch butterfly optimization: A comprehensive review. *Expert Syst. Appl.* **2021**, *168*, 114418–114444. [[CrossRef](#)]
8. Fang, J.-W.; Chao, Y.-S.; Yuan, Y.-P. Research on the Trajectory Optimization of Welding Robot Based on Improved B-Spline Interpolation Method. *Mach. Des. Manuf.* **2021**, *10*, 1–5. (In Chinese) [[CrossRef](#)]
9. Pang, C.; Yan, S.; Li, Z.; He, Y. Kinematics Analysis and Simulation Research of Fruit and Vegetable Picking Arm Based on MATLAB. *J. Phys. Conf. Ser.* **2021**, *1748*, 062050. [[CrossRef](#)]
10. Feng, S.X.; Liu, Y.J.; Xia, H.Q. Trajectory planning and motion simulation of welding robot. *Manuf. Technol. Mach. Tool* **2020**, *11*, 61–65.
11. Jia, L.; Liu, S.; Cao, C.; Kang, Y.; Zhu, Y.; Wang, L.; Xu, D.; Cheng, R. Kinematics analysis and simulation of six-axis robot based on DH parameters. *Sci. Rep.* **2024**, *14*, 13649.
12. Saadah, A.; Al Kadi, M.; Géza, H. KUKA arc KR5 industrial manipulator's trajectory planning modeling using MATLAB based on kinematics study. In Proceedings of the 2023 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, 20–23 February 2023; ISBN 978-1-6654-5474-2/23.
13. Kim, Y.; Kim, B.K. Time-optimal trajectory planning based on dynamics for differential-wheeled mobile robots with a geometric corridor. *IEEE Trans. Ind. Electron.* **2017**, *64*, 5502–5512. [[CrossRef](#)]
14. Schafer, L.; Manzinger, S.; Althoff, M. Computation of solution spaces for optimization-based trajectory planning. *IEEE Trans. Intell. Veh.* **2021**, *8*, 216–231. [[CrossRef](#)]
15. Chi, W.; Ding, Z.; Wang, J.; Chen, G.; Sun, L. A generalized Voronoi diagram-based efficient heuristic path planning method for RRTs in mobile robots. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4926–4937. [[CrossRef](#)]
16. He, S.; Hu, C.; Lin, S.; Zhu, Y. An online time-optimal trajectory planning method for constrained multi-axis trajectory with guaranteed feasibility. *IEEE Robot. Autom. Lett.* **2022**, *7*, 7375–7382. [[CrossRef](#)]
17. Ning, Q.; Shi, L.; Tang, B.; Zhang, B.N. Kinematics analysis and simulation of 6DOF welding robot. *Mach. Des. Manuf.* **2022**, *4*, 246–252.
18. Li, X.-X.; Wang, M.-L.; Liu, K.; Jiang, R. Smooth trajectory planning based on five degrees B-spline for manipulators in joint space. *Modul. Mach. Tool Autom. Manuf. Tech.* **2012**, *8*, 39–42.
19. Zhu, Q. Research on the Motion Planning of Six-Axis Industrial Robots. Master's Thesis, Nanjing University of Information Science and Technology, Nanjing, China, 2021.
20. Zhang, Y.; Di, H.; Chen, Z. Trajectory planning of manipulator based on NSGA-II under multi-objectives. *Modul. Mach. Tool Autom. Manuf. Tech.* **2024**, *5*, 65–70.

21. Guo, Y.; Niu, W.; Zhou, J.; Liu, H. Near-time optimal federate planning for the NURBS curve considering interpolation error constraints. *Robot. Comput.-Integr. Manuf.* **2024**, *86*, 102679. [[CrossRef](#)]
22. He, Z.; Cui, L.; Zhao, S. A novel inverse P-M diffusion enhanced code spraying robot for express security inspection. *IEEE Access* **2022**, *10*, 32350–32360. [[CrossRef](#)]
23. Zhang, W.; Wang, H.L.; Hui, Z.G.; Luo, X.; Zhang, Z. Improved Pure Pursuit Agricultural Machinery Navigation Curve Path Tracking Method Based on B-spline Optimization. *Trans. Chin. Soc. Agric. Mach.* **2024**, *55*, 7.
24. Mohamed, E.; Simic, M.; Jazar, R.N. Continuous path smoothing for car-like robots using B-spline curves. *J. Intell. Robot. Syst.* **2015**, *80*, 23–56.
25. Fang, S.; Ma, X.; Zhao, Y.; Zhang, Q.; Li, Y. Trajectory planning for seven-DOF robotic arm based on quintic polynomial. In Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 24–25 August 2019; Volume 2.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.