applied sciences

MDPI

Article

# A High-Precision Planar NURBS Interpolation System Based on Segmentation Method for Industrial Robot

**Xun Liu [1,*], Yan Xu [2], Jiabin Cao [1], Jinyu Liu [1] and Yanzheng Zhao [1]**

[1] School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China; caojiabin@sjtu.edu.cn (J.C.); yzh-zhao@sjtu.edu.cn (Y.Z.)
[2] College of Information and Computer Engineering, Northeast Forestry University, Harbin 150036, China; xuyan@nefu.edu.cn
* Correspondence: liux_robot@sjtu.edu.cn

**Abstract:** NURBS curve parameter interpolation is extensively employed in precision trajectory tasks for industrial robots due to its smoother performance compared to traditional linear or circular interpolation methods. The trajectory planning systems for industrial robots necessitate four essential functional modules: first, the spline curve discretization technique ensuring chord error compliance; second, the contour scanning technique for determining the maximum feasible feed rate for multi-constraint and multi-segment paths; third, the technique for achieving a smooth feed rate profile; and fourth, the continuous curve parameter interpolation technique. Therefore, this paper proposes a high-precision planar NURBS interpolation system for industrial robots. Firstly, a segmentation method for NURBS curves based on a closed-loop chord error constraint is proposed, which segments the original global NURBS curve into a collection of Bezier curves that strictly meet the chord error constraint. Secondly, a bidirectional scanning technique is presented to meet the joint space constraint, establishing an analytical mapping between the tool tip kinematic constraint and the joint kinematic constraint. Then, based on the traditional S-shaped feed rate profile, an adaptive algorithm with a displacement constraint is introduced, considering the real-time speed adjustment requirements of robots. Finally, a compensation interpolation strategy based on arc length parameterization is adopted to solve the accumulated error problem in parameter interpolation. The effectiveness of and potential for enhancing the quality of planar machining of the proposed planar NURBS interpolation system for industrial robots are validated through simulations and experiments. The results demonstrate the system's applicability and accuracy, and its ability to improve planar machining quality.

**Keywords:** NURBS; parameter interpolation; industrial robot; closed-loop chord error constraint; S-shaped feed rate profile; bidirectional scanning

check for updates

**Citation:** Liu, X.; Xu, Y.; Cao, J.; Liu, J.; Zhao, Y. A High-Precision Planar NURBS Interpolation System Based on Segmentation Method for Industrial Robot. *Appl. Sci.* **2023**, *13*, 13210. https://doi.org/10.3390/app132413210

Academic Editors: Guoqin Huang, Changcai Cui, Fangchen Yin and Jingqi Zhang

Received: 15 November 2023
Revised: 10 December 2023
Accepted: 11 December 2023
Published: 13 December 2023

## 1. Introduction

Industrial robot machining systems not only have weak stiffness characteristics compared to traditional numerical control (CNC) machining systems, but also have constraints on tool path chord error and nonlinear kinematic mapping issues. These are the main factors that affect machining quality and trajectory accuracy. The purpose of robot body control and compensation is to reduce the influence of joint weak stiffness on trajectories, while the purpose of industrial robot interpolation systems is to improve the accuracy and efficiency of trajectories. Therefore, the adjustment of the feed rate in the industrial robot curve parameter interpolation system plays a key role in the high-speed and high-precision machining of molds, aerospace parts, and other applications [1]. Smooth trajectories can help robots avoid sudden load changes, extend the life of joints, and improve position control accuracy. On the other hand, discontinuous feed rates can cause vibrations in the mechanical structure and servo control system, thereby reducing contour accuracy, affecting trajectory accuracy, and leading to task failures or even accidents [2]. Therefore, the research on high-precision trajectory planning and interpolation systems for industrial robots is of great significance. In order to achieve high-speed and high-precision machining, many scholars have devoted themselves to studying techniques such as parameter curve segmentation, feed speed contour adjustment, and parameter curve interpolation. Traditional flat paths typically use a large number of short line segments instead of the entire curve, which inevitably leads to discontinuous tangent directions and abrupt changes in curvature, resulting in a decrease in feed rate. Due to the lack of G1 and G2 continuity at the junctions of adjacent line segments, rate fluctuations and changes in acceleration often occur. Nonuniform rational B-spline (NURBS) parameter curves have effectively addressed the above problems [3] and have become the standard format for free-form curves and surfaces.

Based on the above issues, the current mainstream research content on planning and interpolation systems can be divided into three aspects. Firstly, there is the estimation method for NURBS path chord error. Currently, the majority of studies utilize curvature threshold segmentation techniques, which belong to open-loop chord error control and cannot ensure that real-time curve interpolation meets the chord error requirements throughout the entire machining process. Although there have been some studies [4–6] on accurately calculating chord error, the calculations are quite complex, so approximate estimation methods such as the second-order penetration circle method [7] are commonly used. Hua et al. [8] proposed a B-spline approximation scheme method with tool tip position adjustment. NURBS is divided into several subdivisions according to the discrete curvature. However, it requires iterative computation of new control points, which is inefficient and alters the original curve shape. Bi et al. [9] proposed a B-spline fitting scheme taking advantage of the B-spline's strong convex hull property. In this process, the knot vectors and control points of curve segments, exceeding the prescribed tolerance for chord errors, are refined analytically. The optimization technique [10] fully utilized the tolerance band to minimize the maximum curvature of a B-spline curve with seven control points by adjusting the positions of two control points. However, this approach was specifically employed for local transitions. Ward et al. [11] proposed a novel approach for adaptively discretizing lengthy linear toolpaths to confine nonlinear interpolation errors. Nevertheless, the existing error models are not applicable to industrial robots. Recently, Shi et al. [12] constrained the nonlinear errors in the path in joint space, achieving motion smoothness. However, the drawback is that the errors in this approach cannot be analytically controlled. By using curvature scanning to obtain critical points on the curve and dividing the curve into multiple subcurves [13], it cannot be guaranteed that the chord error in the segmented curves satisfies the given constraints. Secondly, there is a need for real-time trajectory generation technology that calculates robot trajectories satisfying kinematic constraints for each axis. Known constraint parameters are required for feed rate scheduling. Ref. [14] proposes a relatively comprehensive S-shaped feed rate scheduling algorithm (CSFA) with restrictions on jerk, acceleration, and feed rate. It uses a dynamic forward–backward scanning strategy with restrictions on chord error, normal/tangential acceleration, jerk, and feed rate. However, in the error compensation part, a segmented rounding method is used. Although it eliminates the error in segmented arc length interpolation, it introduces feed rate fluctuations at adjacent spline connection points and does not consider robot joint constraint issues. Since the bidirectional scanning cannot guarantee the success of dynamic forward scheduling, ref. [15] introduces the feed rate bisection method to search for feasible boundary velocities. Although it can solve the problem of failed dynamic forward scheduling, its execution efficiency is severely reduced. To compensate for the limitations of unidirectional S-curves in NURBS curve feed rate scheduling applications, Liu et al. [16] proposed a bidirectional scanning algorithm that obtains feed rate parameters for key points and considers the rounding error and its compensation strategy to improve interpolation accuracy. However, this method does not consider the constraint of a contour error in the middle part of the curve segment, resulting in the speed exceeding limits in the planned curve.

To consider the continuous existence of constraints, it is essential to use smooth velocities for trajectory planning. Although the trapezoidal feed rate scheduling method is simple to implement [17], the discontinuity in acceleration can cause vibrations during frequent acceleration and deceleration, leading to a decrease in the lifespan of the motion equipment. The G1 and G2 discontinuities occurring at the junctions of two linear segments inevitably result in frequent feed rate fluctuations and acceleration variations [18]. Therefore, toolpath smoothing operations are crucial in numerical control systems to improve path continuity [19]. To address this problem, many feed rate scheduling methods that restrict acceleration jumps have been proposed, such as the time-optimal method [20,21], the sine-shaped method [22], and the S-shaped method [23,24]. Nguyen et al. [25] designed a systematic approach to design time-optimal S-curve motion profiles. Y. Bai et al. [26] studied the asymmetric motion properties of feed rate scheduling, as traditional or most feed rate planning methods rely on symmetric formulas for the start and stop segments, but the time is not optimized in those cases. Thirdly, for the parametric solution of NURBS curve interpolation, Shipitalni et al. [27] performed a first-order Taylor expansion treating the interpolation parameters as a function of time, but the calculation precision is not high. To improve the calculation precision, Yeh et al. [28] proposed an interpolation point parameter solution algorithm based on second-order Taylor expansion, but it requires a large amount of computation, which impacts the real-time capability of the algorithm. M. Tikhon [29] proposed that within each interpolation period, the maximum feed rate is determined by the specified chord error and constant material removal rate. In existing research [30,31], it has been demonstrated that the Taylor expansion method can generate trajectories identical to the reference method, but feed rate fluctuations are inevitable in regions with high curvature. An optimization algorithm [32] is proposed to improve the precision of NURBS curves. This algorithm employs an iterative process to calculate block displacement, with the NURBS parameter u as the optimization target. However, the requirement to repeatedly compute the accurate value of u results in the inability to perform online interpolation. Most interpolation methods use Taylor expansion to approximate the next interpolation parameter.

In response to the above issues, this article proposes a tool speed scheduling and high-precision interpolation system for industrial robots for NURBS curves. Firstly, a fast and accurate calculation method for tool path curvature is proposed. The knot insertion technique is introduced to segment the original tool path into segments based on curvature, which creates a set of rational Bezier curves. The entire processing curve is ensured to meet the requirements of the chord error. Secondly, considering the impact of the nonlinear kinematics of robot joints, a dynamic optimization of the maximum feed rate and maximum acceleration is conducted during the bidirectional scanning process of the traditional feed rate extrema curve. This optimization achieves dynamic speed regulation and braking functions within an adaptive S-shaped feed rate profile. The preset values for the maximum feed rate and maximum acceleration constraints in bidirectional feed rate scanning are adjusted, ensuring smooth robot interpolation while adhering to joint kinematic constraint. Finally, during the curve interpolation stage, a compensation interpolation strategy using Taylor expansion and arc length parameterization is applied to the segmented NURBS curve set. The parameter remapping method is used to ensure parameter continuity at the junction points of adjacent curve segments, without transferring all the original NURBS control points to the underlying structure. This reduces storage capacity and eliminates nonlinear errors in interpolation.

The remaining sections of the paper are organized as follows. Section 2 proposes a method for segmenting NURBS curves into a set of rational Bezier convex curves based on a closed-loop chord error constraint. Section 3 introduces an improved adaptive S-shaped feed rate scheduling algorithm, which achieves an adaptive planning mode under a given displacement. It also presents a method for robot speed control and braking planning under arbitrary states, enabling smooth speed regulation and braking based on S-shaped feed rate scheduling. Section 4 proposes a bidirectional feed rate profile scanning algorithm that effectively combines robot joint constraints to meet maximum performance requirements. Section 5 addresses the problem of the continuous interpolation of convex curve sets and presents a parameter remapping method based on Taylor expansion to ensure parameter continuity at the junction points of adjacent curves. Section 6 conducts NURBS processing tests on an industrial robot and provides experiments and simulations, and the conclusion is given in Section 7.

## 2. Segment NURBS into Bezier Curves Based on Closed-Loop Chord Error Constraint

### *2.1. Definition of NURBS Curve and Knot Insertion*

This paper uses cubic NURBS curves to describe the planar machining path. The rational basis function representation of the NURBS curve equation is expressed as follows [3]:

$$\begin{cases} C(u) = \sum\limits_{i=0}^{n} P_i R_{i,k}(u) \\ R_{i,k}(u) = \frac{\omega_i N_{i,k}(u)}{\sum\limits_{j=0}^{n} \omega_i N_{i,k}(u)} \end{cases} \tag{1}$$

In the equation, the $k$-th rational basis function is denoted as $R_{i,k}(u)$ and has properties similar to those of the $k$-th normalized B-spline basis function $N_{i,k}(u)$, such as normality, local support, and differentiability. $\omega_i$ is the weight factor, and the control points are represented by $P_i$. Connecting them in order with line segments forms the control polygon.

$$\begin{cases} N_{i,0}(u) = \begin{cases} 1, & u_i \le u \le u_{i+1} \\ 0, & otherwise \end{cases} \\ N_{i,k}(u) = \frac{u-u_i}{u_{i+k}-u_i} N_{i,k-1}(u) + \frac{u_{i+k+1}-u}{u_{i+k+1}-u_{i+1}} N_{i+1,k-1}(u) \\ i = 0, 1, \ldots, n \end{cases} \tag{2}$$

The nonuniform knots vector $U$ is given as follows:

$$U = \left[ \underbrace{0, 0, \ldots 0,}_{k+1} u_{k+1}, \ldots, u_{n-k-1}, \underbrace{1, 1, \ldots, 1}_{k+1} \right] \tag{3}$$

Although the cubic NURBS curves described above can solve the issue of smoothness in the path, the currently adopted curvature-based speed limiting technology belongs to open-loop bow chord error control, which cannot guarantee real-time compliance with chord error requirements throughout the entire machining process. To address this issue, this paper proposes a closed-loop chord error control technology for planar NURBS curves to improve machining quality. Firstly, the knot insertion technique is used to segment the original NURBS curves based on curvature, dividing the curves into multiple segments of cubic rational Bezier curves. Then, the parameter points where the curvature is zero are solved for each curve segment, and a projection algorithm is used to project the curves onto the coordinate axes. The precise chord error is obtained by differentiating the analytic cubic rational Bezier formula and obtaining the extreme points.

In order to obtain an analytical set of segmented Bezier curves, it is necessary to introduce the technique of knot insertion, which is a crucial technique in the fundamental geometric properties of spline curves. When new knots are inserted into the knot vector interval of the original spline curve, the equality relationship between the number of control points and knot s results in an increase in the corresponding number of new control points. Therefore, the node insertion technique achieves the following. Modification of Local Properties: By inserting nodes, the local properties of the original spline can be altered, allowing for the control of local shapes and enhancing local characteristics. This, in turn, increases the flexibility of the spline; and Generation of Local Beziers: When inserted knots generate new control points according to a certain rule, the original spline can be constructed into new local Beziers without altering the shape of the original spline. This property is advantageous for locally segmenting the spline into Bezier segments, making the control of segmented Bezier curves simpler, more convenient, and more efficient.

Therefore, the introduction of node insertion technology not only enhances the local characteristics of the original spline but also facilitates the segmentation and control of the spline, leading to improved efficiency. This paper leverages the characteristics of node insertion to generate additional control points. When new knots are inserted into the node vector interval of the original spline curve, the equation relationship between the number of control points and the number of nodes leads to an increase in the corresponding number of new control points. This paper takes advantage of the knot insertion technique to achieve the segmented rational Bezier subdivision of a NURBS curve. Assuming the knot vector of the original spline of order $k$ with $n$ control points is $U = [u_0, u_1, \ldots u_{n+k+1}]$, and $u \in [u_i, u_{i+1}]$ is the knot to be inserted, then the new node vector is $U = [u_0, u_1, \ldots, u_i, u, u_{i+1}, \ldots u_{n+k+1}]$, which can be rewritten as $\overline{U} = [\overline{u}_0, \overline{u}_1, \ldots, \overline{u}_i, \overline{u}_{i+1}, \ldots \overline{u}_{n+k+2}]$. The new knot vector determines a set of new spline basis functions. In 1980, Boehm [33] introduced a method for calculating additional control points without changing the shape and parameters of the spline curve under the condition of knot insertion.

$$\overline{P}_j = P_j, j = 0, 1, \ldots, i-k \tag{4}$$

$$\overline{P}_j = \alpha_j P_j + \left(1 - \alpha_j\right) P_j, \quad j = i-k+1, \ldots, i-r \tag{5}$$

$$\overline{P}_j = P_{j-1}, \quad j = i-r+1, \ldots, n+1 \tag{6}$$

$$\alpha_j = \frac{t - t_j}{t_{j+k} - t_j}, \quad \left(\frac{0}{0} = 0\right) \tag{7}$$

By using the knot insertion technique, a NURBS curve is segmented into multiple segments of Bezier curves. The rational fraction of a cubic Bezier curve can be simplified to the form of a polynomial numerator and denominator:

$$B(u) = \left(\frac{1}{w(u)}\right)\left[\left(1 - u^3\right)\omega_0 P_0 + 3u(1-u)^2\omega_1 P_1 + 3u^2(1-u)\omega_2 P_2 + u^3\omega_3 P_3\right], 0 \le u \le 1 \tag{8}$$

$$w(u) = (1-u)^3\omega_0 + 3u(1-u)^2\omega_1 + 3u^2(1-u)\omega_2 + u^3\omega_3 \tag{9}$$

By using the knot insertion method, new control points and node vectors are created for NURBS by segmenting the NURBS curve into several Bezier curves. Table 1 includes the NURBS control points and knot vectors assigned in this study, and the results of the initial segmentation can be found in Figure 1.

**Table 1.** Bezier control points and node vectors obtained after the initial segmentation.

| Curve Types | Control Points $\left[\begin{smallmatrix} x \\ y \end{smallmatrix}\right]$ | Node Vectors |
|---|---|---|
| NURBS | $\begin{bmatrix} 5 & 6 & 11 & 8 & 12 & 11 \\ 4 & 12 & 10 & 4 & 3 & 9 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 1/3 & 2/3 & 1 & 1 & 1 & 1 \end{bmatrix}$ |
| Bezier 1 | $\begin{bmatrix} 5 & 6 & 8.5 & 9.25 \\ 4 & 12 & 12 & 9.5 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ |
| Bezier 2 | $\begin{bmatrix} 9.25 & 10 & 9 & 9.5 \\ 9.5 & 8 & 6 & 4.75 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ |
| Bezier 3 | $\begin{bmatrix} 9.5 & 10 & 12 & 11 \\ 4.75 & 3.5 & 3 & 9 \end{bmatrix}$ | $\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$ |

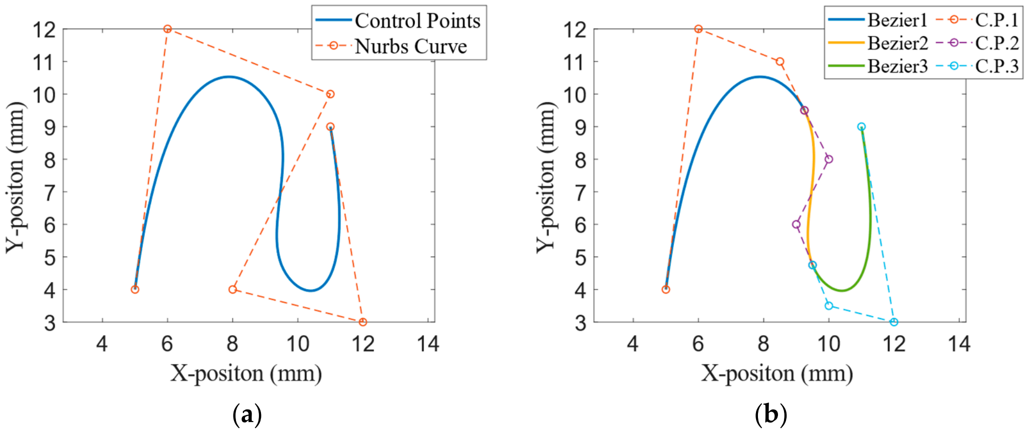


**Figure 1.** A sample NURBS: (**a**) NURBS curve with 6 control points; (**b**) using knot insertion method, segment NURBS curve into 3 Bezier curves and C.P.1-4 is control points.

*2.2. Curve Segment Based on Closed-Loop Chord Error*

Although the knot insertion method can transform a NURBS curve into multiple Bezier curves by inserting new control points without altering the geometric type of the curve, the subdivided cubic Bezier curves have two types: C and S. Figure 1b shows the insertion results, where Bezier1 and Bezier3 are in type-C, and Bezier2 is in type-S. For type-C, there is only one extremum point, while type-S has two extremum points. The curve can be split into two type-Cs at the point where the curvature is zero. One simple method to solve this is by projecting the curve parameter u onto the x-axis, in which case the y-axis represents the distance from the curve to the line, which is the chord error. The curvature is defined as:

$$\kappa = \frac{\|\dot{B}(u) \times \ddot{B}(u)\|}{\|\dot{B}(u)\|^3} \tag{10}$$

where $\dot{B}(u)$ and $\ddot{B}(u)$ are the first and second derivatives of the curve, respectively, and the knot with zero curvature is obtained through Equation (10). Figure 2 shows the segmentation results.

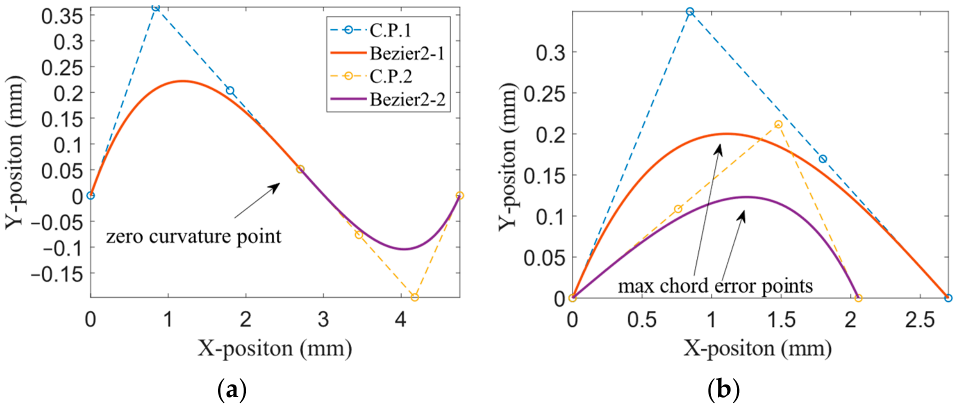


**Figure 2.** Segmentation results: (**a**) one type-S Bezier2 curve segmented into two type-C curves by zero curvature point; (**b**) the max value on y-axis of the curve is the chord error.

Finally, according to the set chord error $\delta$, the new Bezier curves are continuously segmented into halves until the chord error of all Bezier curves meet the requirement. Figure 3 shows the second segmentation and subsequent iterative segmentation results. Details can be found in Algorithm 1.

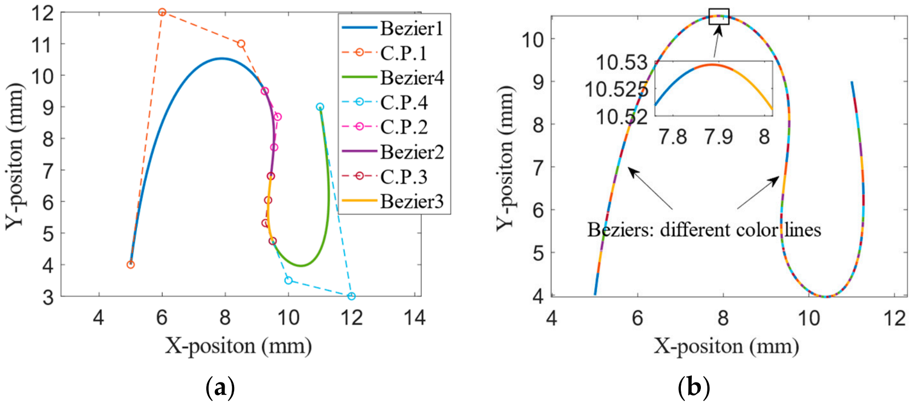


**Figure 3.** Simulation for NURBS curve. Each of the adjacent curves are distinguished by different colors: (**a**) NURBS curve segmented into type-C curve by zero curvature point; (**b**) NURBS curve segmented into 143 type-C curves that satisfy $\delta = 0.001$ mm.

The pseudo-code is proposed as follows to generate the chord error control method:

**Algorithm 1:** Segmentation of NURBS curve with chord error $\delta_c$.

```
Input:    Input NURBS curve parameters, Chord error $\delta$
Output:   Bezier curves that satisfy the $\delta_c$
          knot insertion $\rightarrow$ Bezier curve $Bi$;
          for $i$ = 1 to Bezier curve number do
            calculate the parameter $u$ with zero curvature in the type-S curve $Bi$.
            calculate the $\delta_{\max}$ in the type-C curve $Bi$.
          If $\delta_{\max} > \delta_c$ then
              Bisection method $\rightarrow$ segment curve $Bi \rightarrow 1Bi$, $2Bi$;
              do algorithm 1 for $1Bi$;
              do algorithm 1 for $2Bi$;
            else
              return $1Bi$, $2Bi$;
            end
          end
```

## 3. *S*-Shaped Feed Rate Profile Generation and Dynamic Speed Adjustment

### *3.1. S-Shaped Feed Rate Profile*

In this section, we introduce the double *S*-shaped feed rate profile briefly. The S-shaped ACC/DEC profile, which consists of seven phases ($T1$, $T2$, $T3$, $T4$, $T5$, $T6$, $T7$), is shown in Figure 4.

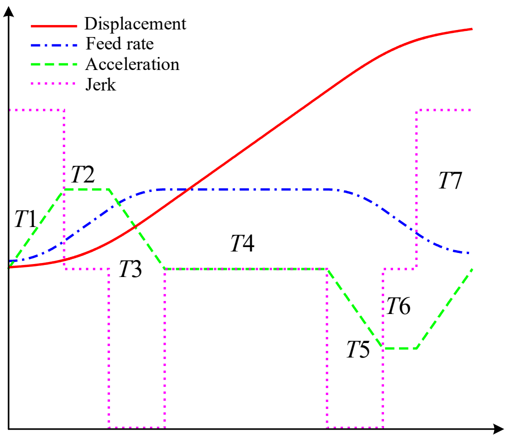


**Figure 4.** The *S*-shaped profile with seven sections.

The parameters include: $v_s$ representing the initial feed rate, $v_{spe}$ representing the operating feed rate, $v_e$ representing the final feed rate, $S$ representing the specified displacement, $V_{\max}$ representing the maximum feed rate (set to $v_{spe}$), $A_{\max}$ representing the maximum acceleration, and $J$ representing the maximum jerk. $t_s$ is defined as the variable time, $t_{ls}$ is the critical time for acceleration and deceleration, and $t_{sp}$ represents the duration of constant acceleration during the variable acceleration phase. The corresponding acceleration $a(t)$, feed rate $v(t)$, and displacement $S(t)$ can be calculated based on time and the three acceleration states using the following formulas:

$$a(t) = \begin{cases} \mathrm{sign}(v_e - v_s)Jt, 0 \le t \le \frac{t_s - t_{sp}}{2} \\ \frac{1}{2}\mathrm{sign}(v_e - v_s)J\left(t_s - t_{sp}\right), \frac{t_r - t_{sp}}{2} < t \le \frac{t_s + t_{sp}}{2} \\ \mathrm{sign}(v_e - v_s)J(t_s - t), \frac{t_s + t_{sp}}{2} < t \le t_s \end{cases} \tag{11}$$

$$v(t) = \begin{cases} v_s + \frac{1}{2}\mathrm{sign}(v_e - v_s)Jt^2, 0 \le t \le \frac{t_s - t_{sp}}{2} \\ v_s + \frac{1}{8}\mathrm{sign}(v_e - v_s)Jt_{ls}^2 \\ +\frac{1}{2}\mathrm{sign}(v_e - v_s)Jt_{ls}\left(t - \frac{t_{ls}}{2}\right), \frac{t_s - t_{sp}}{2} < t \le \frac{t_s + t_{sp}}{2} \\ v_s + \frac{1}{4}\mathrm{sign}(v_e - v_s)Jt_{ls}^2 \\ +\frac{1}{2}\mathrm{sign}(v_e - v_s)Jt_{ls}t_{sp} - \frac{1}{2}\mathrm{sign}(v_e - v_s)J(t_s - t)^2, \frac{t_s + t_{sp}}{2} < t \le t_s \end{cases} \tag{12}$$

$$S(t) = \begin{cases} v_s t + \frac{1}{6}\mathrm{sign}(v_e - v_s)Jt^3, 0 \le t \le \frac{t_s - t_{sp}}{2} \\ v_s t + \frac{1}{48}\mathrm{sign}(v_e - v_s)Jt_{ls}^3 + \frac{1}{8}\mathrm{sign}(v_e - v_s)Jt_{ls}^2\left(t - \frac{t_{ls}}{2}\right) \\ +\frac{1}{4}\mathrm{sign}(v_e - v_s)Jt_{ls}\left(t - \frac{t_{ls}}{2}\right)^2, \frac{t_s - t_{sp}}{2} < t \le \frac{t_s + t_{sp}}{2} \\ v_s t + \frac{1}{4}\mathrm{sign}(v_e - v_s)J\left(-\frac{1}{2}t_{ls}^3 - \frac{3}{2}t_{ls}^2 t_{sp} - t_{sp}^2 t_{ls} + t_{ls}^2 t + 2t_{ls}t_{sp}t\right) \\ +\frac{1}{6}\mathrm{sign}(v_e - v_s)J(t_s - t)^3, \frac{t_s + t_{sp}}{2} < t \le t_s \end{cases} \tag{13}$$

The implementation process is as follows:

Step 1: Based on the input motion parameters and the relationship between $v_s$, $v_e$, and $v_{spe}$, determine if there is constant linear interpolation. If $v_s$, $v_e$, and $v_{spe}$ are equal, it indicates that only constant linear interpolation exists in this planning, so proceed to interpolation with only the constant linear segment $d$. If the trajectory does not involve constant linear interpolation, proceed to step 2;

Step 2: According to the relevant formulas mentioned above, calculate the sum of displacements from $v_s$ to $v_{spe}$, and $v_{spe}$ to $v_e$, denoted as $S_1$. Calculate the variable displacement $S_2$ when directly accelerating from $v_s$ to $v_e$, without going through $v_{spe}$;

Step 3: Based on the parameter relationships described above, perform classification and judgment, as shown in the flowchart in Figure 5.

### 3.2. Dynamic Speed Control Planning

Since this paper is based on the simplification of the problem with zero initial and final accelerations, the *S*-shaped method cannot achieve real-time transitions from any motion state to another. Therefore, this paper proposes a dynamic speed control planning that ensures speed changes and an acceleration reset in the shortest possible time within the original S-shaped feed rate profile framework.

First, when the initial acceleration is not zero and acceleration replanning is required, the maximum jerk $J$ needs to be applied in order to make the acceleration become zero. By planning with the maximum jerk, we can calculate the time $t_{a0}$ and the corresponding velocity $v_{a0}$ needed to change the initial acceleration $a_s$ to zero.

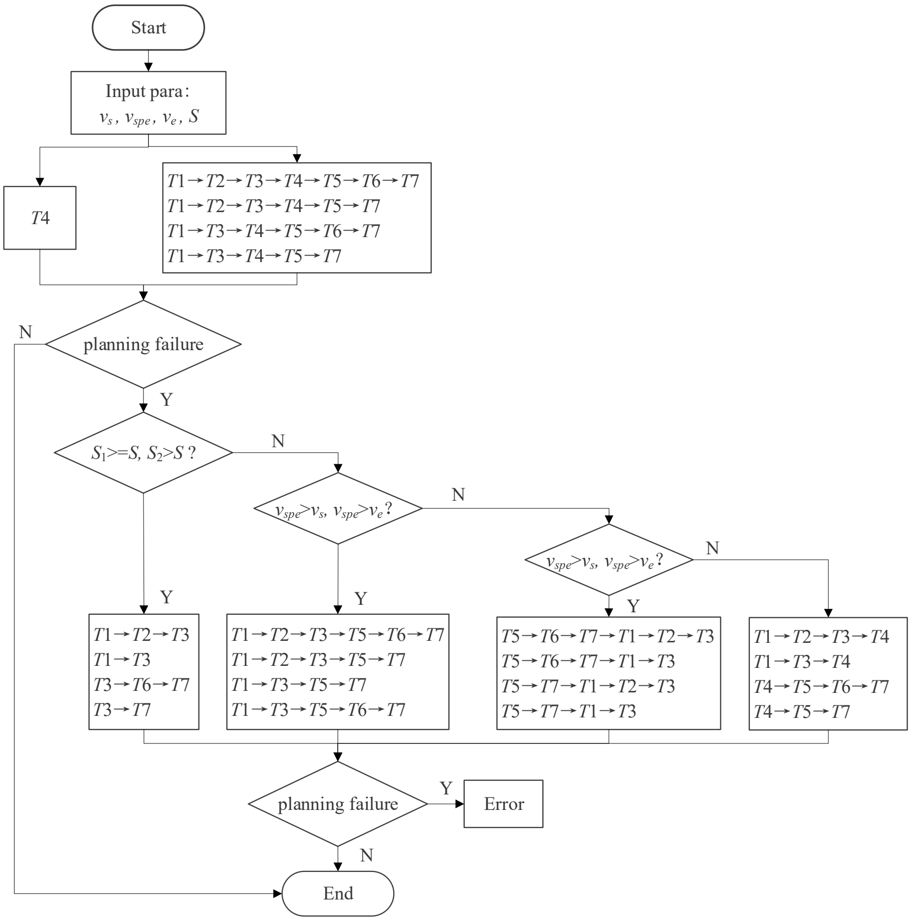


**Figure 5.** The flowchart of S-shaped feed rate profile generation.

$$t_{a0} = \left|\frac{a_s}{J}\right| \tag{14}$$

$$v_{a0} = v_s + a_s t_{a0} + \frac{J t_{a0}^2}{2} \tag{15}$$

The formulas for interpolating position, feed rate, and acceleration based on jerk are as follows:

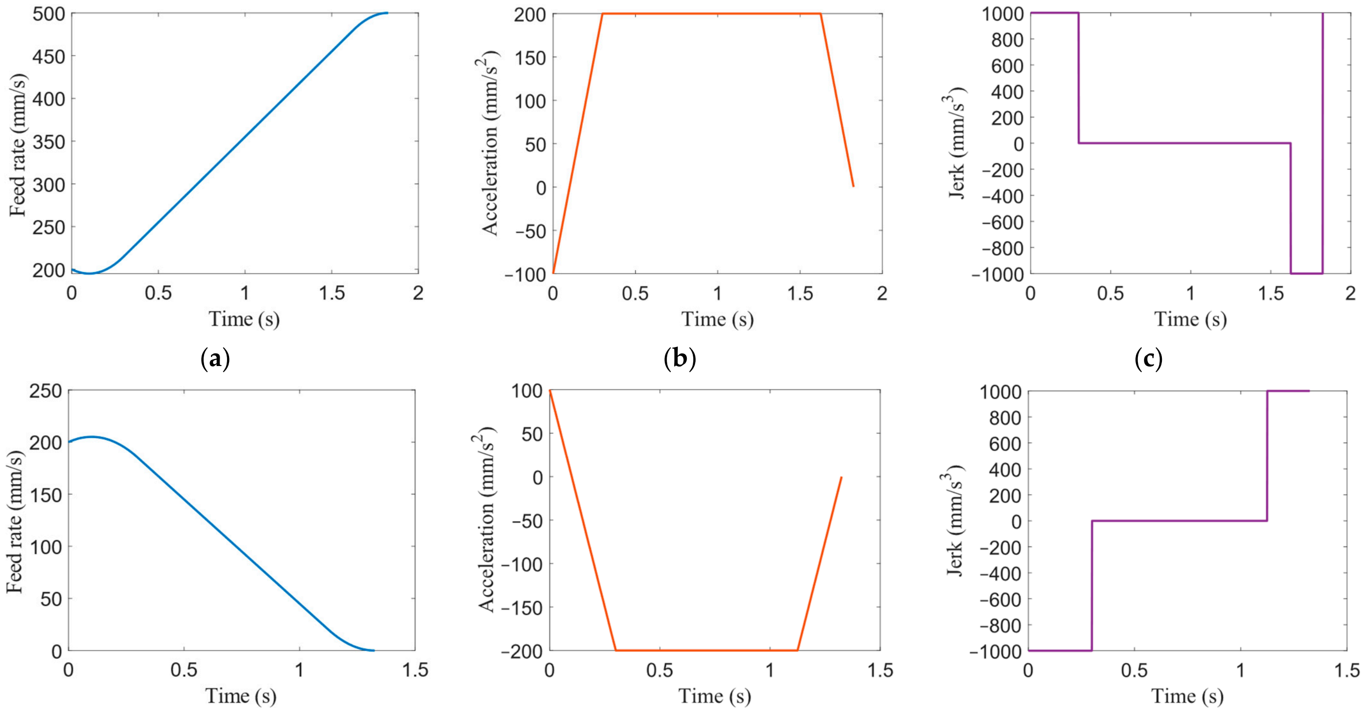
$$\begin{cases} S(t) = S(0) + (v_s + (\frac{a_s}{2} + \frac{Jt}{6})t)t \\ v(t) = v_s + (a_s + \frac{Jt}{2})t \\ a(t) = a_s + Jt \end{cases} \tag{16}$$

At this point, the acceleration is reduced to zero, but there is still a nonzero feed rate. In the second step, it is necessary to replan the acceleration to ensure that the feed rate changes to the desired state with acceleration also being zero.

$$S_a(v_s, v_e) = \begin{cases} (v_s + v_e)\sqrt{\frac{v_e - v_s}{J_{\max}}} & v_e - v_s \le \frac{{A_{\max}}^2}{J_{\max}} \\ \frac{1}{2}(v_s + v_e)\left[\frac{A_{\max}}{J_{\max}} + \frac{(v_e - v_s)}{A_{\max}}\right] & v_e - v_s > \frac{{A_{\max}}^2}{J_{\max}} \end{cases} \tag{17}$$

$$S_d(v_s, v_e) = \begin{cases} (v_s + v_e)\sqrt{\frac{v_s - v_e}{J_{\max}}} & v_s - v_e \le \frac{{A_{\max}}^2}{J_{\max}} \\ \frac{1}{2}(v_s + v_e)\left[\frac{A_{\max}}{J_{\max}} + \frac{(v_s - v_e)}{A_{\max}}\right] & v_s - v_e > \frac{{A_{\max}}^2}{J_{\max}} \end{cases} \tag{18}$$

In the formula, $A_{\max}$ represents the maximum acceleration, and $J_{\max}$ represents the maximum jerk. Based on the current $v_s$ and the desired $v_e$, a determination is made whether to accelerate or decelerate. If acceleration is required, Equation (17) is employed. Conversely, if deceleration is needed, Equation (18) is used. $S_a$ and $S_d$ denote the displacement for acceleration and deceleration, respectively. By ensuring an adequate length of displacement for feed rate lookahead, smooth braking can be achieved. Figure 6 illustrates these two scenarios.



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

(**d**) (**e**) (**f**)

**Figure 6.** Schematic diagram of dynamic speed control planning. Acceleration control case: (**a**) feed rate curve; (**b**) acceleration curve; (**c**) jerk curve; deceleration control case: (**d**) feed rate curve; (**e**) acceleration curve; (**f**) jerk curve.

## 4. Bidirectional Feed Rate Scanning of S-Shaped Profile

For the processing trajectory of the end effector of the manipulator, its feed rate vector is obtained by combining the tangential velocities of the tool tip and the tool axis direction, with its magnitude equal to the feed rate modulus and its direction determined by the tangential composition of the tool tip spline and the tool axis direction spline. In this study, the feed rate in the tool axis direction is set to zero. Similarly, the acceleration vector is obtained by combining the tangential acceleration and the normal acceleration of the tool tip and the tool axis motion. Therefore, the feed rate vector $\boldsymbol{V}$ and the acceleration vector $\boldsymbol{A}$ at any point on the processing trajectory of the manipulator‘s end effector can be expressed as follows:

$$\boldsymbol{V} = [v\boldsymbol{\tau}, 0] \tag{19}$$

$$\boldsymbol{A} = \left[a\boldsymbol{\tau} + \frac{v^2}{\rho}\boldsymbol{n}, 0\right] \tag{20}$$

where the unit tangential at any point on the tool tip spline is denoted as $\tau$, while the unit normal at any point on the tool tip spline is denoted as $n$. The curvature radius is represented by $\rho$.

The obtained representation pertains to the feed rate and acceleration of arbitrary points on the processing trajectory in the task space. However, during the actual machining process, the known variables are the velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$ constraints of each axis in the joint space. It becomes necessary to make real-time adjustments to the maximum feed rate and acceleration in the bidirectional velocity scanning, as well as the velocity extremum at the endpoints, based on the known kinematic constraints in the joint space. According to the differential kinematic relationship, the following mapping relationships exist between the velocity and acceleration vectors in the joint space and the task space.

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_{ac}{}^{-1}\boldsymbol{V} \tag{21}$$

$$\ddot{\boldsymbol{\theta}} = \boldsymbol{J}_{ac}{}^{-1}\boldsymbol{A} - \boldsymbol{J}_{ac}{}^{-1}\dot{\boldsymbol{J}}_{ac}\left(\boldsymbol{J}_{ac}{}^{-1}\boldsymbol{V}\right) \tag{22}$$

The feed rate constraints in the bidirectional scanning can be divided into geometric constraints and time constraints in the task space. Geometric constraints include the maximum feed rate $V_{\max}$, maximum normal acceleration $A_{n,\max}$, and maximum normal jerk $J_{n,\max}$. Time constraints consist of the maximum tangential acceleration $A_{t,\max}$ and maximum tangential jerk $J_{t,\max}$. The geometric parameters of the path include the maximum chord error $\delta$ and the curvature $\kappa_{i,\max}$ at key points along the curve segment. These parameters summarizing the tangential velocity constraints in the task space are discussed in Reference [9]. In this paper, we provide a summary in the bidirectional scanning flowchart as shown in Figure 7. In the figure, the velocity curve from right to left represents the backward scanning, and vice versa for the forward scanning.

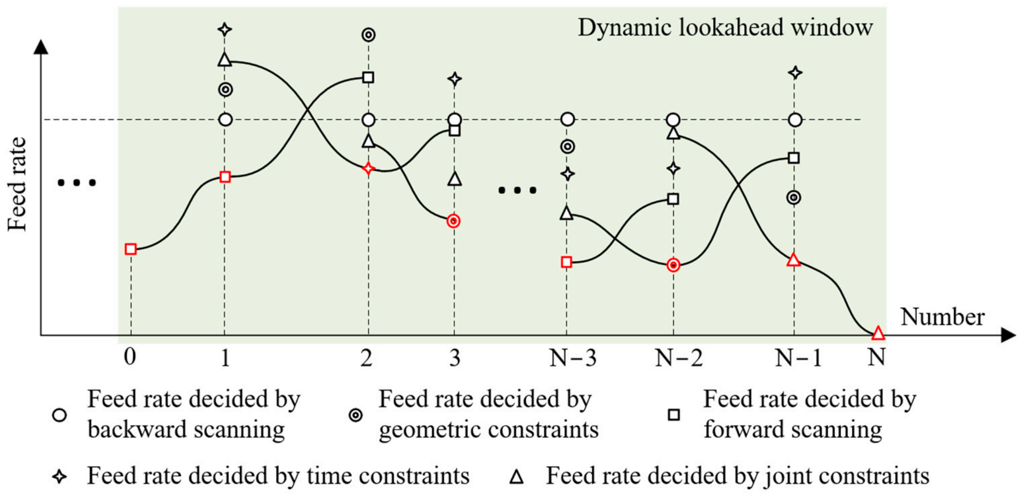

Dynamic lookahead window
Feed rate
Number
...
...
0
1
2
3
N−3
N−2
N−1
N
○ Feed rate decided by backward scanning
⊚ Feed rate decided by geometric constraints
□ Feed rate decided by forward scanning
✧ Feed rate decided by time constraints
△ Feed rate decided by joint constraints

**Figure 7.** The schematic diagram of bidirectional scanning.

In order to simplify calculations, it is common to approximate the feed rate at points along the path as the maximum or very close to the maximum velocity. Based on this approximation, the formulas for correcting the maximum tangential feed rate and acceleration are straightforward. By calculating the corresponding Jacobian matrix $J_{ac}$ and the curve unit tangent $\tau$, optimized values for the maximum tangential feed rate $V_{\max}^{\text{opt}}$ and maximum tangential acceleration $A_{\max}^{\text{opt}}$ can be obtained. The formulas are as follows:

$$V_{\max}^{opt} = \min\left(\boldsymbol{V}_{\max}, \frac{\dot{\boldsymbol{\theta}}_{\max}}{\boldsymbol{J}_{ac}{}^{-1}\boldsymbol{\tau}}\right) \tag{23}$$

$$A_{\max}^{opt} = \min\left(A_{t,\max}, \frac{\ddot{\theta}_{\max} + J^{-1}\dot{J}(J^{-1}V)}{J^{-1}\boldsymbol{\tau}}\right) \tag{24}$$

Assuming the current iteration is the *i*-th segment, with a lookahead window length of $N$, the end feed rate of the $(i-1)$th segment (the start feed rate of the *i*-th segment) is denoted as $V_{\text{end},i-1}$. The lookahead sublist of segment length is represented as $L^{(N)}_{i,\text{list}} = \{L_0, L_1, \cdots, L_i, \cdots, L_N\}$, and the lookahead sublist of feed rates is denoted as $F^{(N)}_{i,\text{list}} = \{F_0, F_1, \cdots, F_i, \cdots, F_N\}$. Additionally, the forward scanning feed rate is calculated as $V^{f}_{\text{end},i}$ and the backward scanning feed rate as $V^{b}_{\text{end},i}$. Considering all velocity constraints and optimization strategies, the dynamic lookahead bidirectional scanning algorithm follows the workflow outlined Algorithm 2 below:

**Algorithm 2:** Bidirectional Scanning.

```
Input:   $V_{end,i-1}$ is the end feed rate of the (i − 1)th segment (the start feed rate of the i-th
         segment); $L^{(N)}_{i,list}$ is the lookahead sublist of segment length; $F^{(N)}_{i,list}$ is the lookahead sublist
         of feed rates; $V_{\max}$ is the maximum feed rate; $A_{t,\max}/J_{t,\max}$ are the maximum tangential
         acceleration/jerk; $\delta$ is the maximum chord error; $\kappa_{i,max}$ is the curvature at key points
         along the curve segment; $A_{n,\max}/J_{n,\max}$ are the maximum normal acceleration/jerk;
         $T_s$ is the sampling period.
Output:  $V_{end,i}$
Step 1:  1.1. If $N = 1$, stop the lookahead calculation and let $V_{end,i} = 0$; Otherwise, go to 1.2.
         1.2. Let $k = N - 1$, $V^{b}_{end,i} = 0$;
Step 2:  2.1. $L^{(N)}_{i,list}[0]$, $V_{end,i-1}$, $F^{(N)}_{i,list}[0]$, $J_{t,\max}$, $A_{t,\max}$, $T_s$ are utilized in the backward scanning,
         recursively backtracking from the last segment of the lookahead window to the second
         segment, to calculate the end feed rate $V^{b}_{end,i}$ based on the current segment velocity and
         constraints;
         2.2. The minimum feed rate under the given geometric constraints, as well as the
         kinematic constraints in both the task space and joint space, is assigned as the end feed
         rate of the current segment. It is given that
         $V^{b}_{end,,i} = \min\left(V^{b}_{end,,i}, V^{opt}_{\max}, \frac{2}{T}\sqrt{\frac{2\delta}{\kappa_{i,\max}} - \delta^2}, \sqrt{\frac{A_{n,\max}}{\kappa_{i,\max}}}, \sqrt{\frac{J_{n,\max}}{\kappa^2_{i,\max}}}\right)$
Step 3:  Let $k = k - 1$, if $k = 0$, go to step 4; Otherwise, go to step 2;
Step 4:  Let $k = 0$, $V^{f}_{end,i} = 0$;
Step 5:  5.1. $L^{(N)}_{i,list}[k]$, $V_{end,i-1}$, $F^{(N)}_{i,list}[k]$, $J_{t,\max}$, $A_{t,\max}$, $T_s$ are utilized in the forward scanning,
         recursively from the first segment of the lookahead window to the (N − 1)th segment, to
         calculate the end feed rate $V^{f}_{end,i}$;
         5.2. Determine the feed rate at the end of the segment by
         $V^{b}_{end,i} = \min(V^{f}_{end,i} V^{b}_{end,i})$
Step 6:  Let $k = k + 1$, and if $k = N - 1$, stop and derive the result; Otherwise, go to step 4.
```

In order to ensure the success of bidirectional scanning, the aforementioned strategy employs a scanning approach with an initial feed rate of zero and the end feed rate of zero. However, when dynamic lookahead is performed, there may be cases where the initial feed rate is nonzero. Suppose the initial feed rate for the current segment $i$, which has achieved successful planning, is denoted as $V_i$. By performing a backward scanning to the feed rate of the $(i+1)$th segment, denoted as $V^{b}_{end,i+1}$, if the forward scanning based on $V_i$ yields a feed rate $V^{f}_{end,i+1}$ greater than $V^{b}_{end,i+1}$, there is a possibility of a planning failure as shown in Figure 8.

To address this issue, a feed rate bisection method was introduced in [10] to search for the feasible feed rate $V_{i+1}$. While this method resolves the problem of dynamic lookahead planning failure, it significantly reduces execution efficiency. Whenever the aforementioned condition occurs, the bisection method is invoked to search for the feasible feed rate $V_{i+1}$ between $V^{b}_{end,i+1}$ and $V^{f}_{end,i+1}$. However, in this paper, an adaptive *S*-shaped method is adopted, which allows for dynamic adjustment of the boundary feed rate based on the relationship between arc length and feed rate. By employing this adaptive *S*-shaped method, the optimal feed rate profile can be directly obtained. Consequently, by initiating the forward scanning anew based on the current situation, the execution efficiency of the actual program is improved.

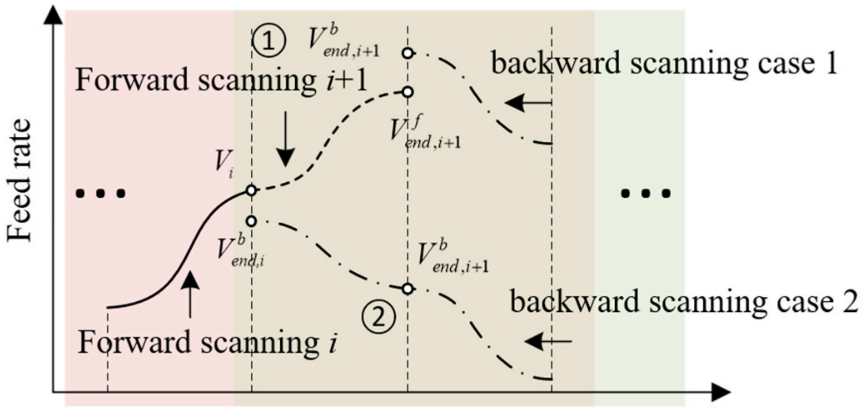


**Figure 8.** Bidirectional scanning case classification.

## 5. NURBS Interpolation

After segmenting the NURBS curve into Bezier curves, the derivative of the Bezier curve cannot be used directly for Taylor expansion since it does not satisfy the C2 continuity condition. Thus, each segment of the Bezier curve has to be mapped back to the original NURBS curve to obtain the parameter continuity defined by the knot vector. Assuming $u$ as the curve parameter of a discrete Bezier curve and $u$ as the corresponding parameter on the NURBS curve, $u$ can be expressed as $f_i(\mu)$, where $f_i(\cdot)$ is a mapping function that maps the NURBS curve parameter $u \in [m_i, n_i]$ to the Bezier curve parameter $u \in$ [0,1]. The knot vector interval of the $i$-th Bezier curve segment, obtained by decomposing the original NURBS curve through node insertion, is represented by $[m_i, n_i]$. Consequently, the following formula can be derived.

$$B_i(u) = B_i(f_i(\mu)) = \{x_i(f_i(\mu)), y_i(f_i(\mu)), z_i(f_i(\mu))\} \tag{25}$$

$$\dot{B}_i(\mu) = \dot{B}_i(u) \cdot \dot{f}_i(\mu) \tag{26}$$

$$\ddot{B}_i(\mu) = \dot{B}_i(u) \cdot \ddot{f}_i(\mu) + \ddot{B}_i(u) \cdot \left(\dot{f}_i(u)\right)^2 \tag{27}$$

$$u = f_i(\mu) = \frac{\mu - m_i}{m_i - n_i} \tag{28}$$

$$\dot{f}_i(\mu) = \frac{1}{n_i - m_i}, \ddot{f}_i(\mu) = 0 \tag{29}$$

Therefore, the relationship between the derivatives of NURBS curves and the derivatives of Bezier curves is:

$$\dot{B}_i(\mu) = \dot{B}_i(u)\frac{1}{n_i - m_i} \tag{30}$$

$$\ddot{B}_i(\mu) = \ddot{B}_i(u)\frac{1}{(m_i - n_i)^2} \tag{31}$$

Utilizing the composite Simpson's rule [34] to calculate the arc length can be perceived as a continuous iterative process for determining the value of the parameter $u$. This is achieved by computing the parameter $u_{i+1}$, which corresponds to the next position coordinate of the interpolation cycle, based on the current position's corresponding parameter $u_i$. Derivatives of the parameter $u$ with respect to time t can be derived, and by expanding these first-order derivatives into a second-order Taylor series, a second-order Taylor direct interpolation formula [35] based on feed rate can be obtained:

$$\ddot{B}_i(\mu) = \ddot{B}_i(u)\frac{1}{(m_i - n_i)^2} \tag{32}$$

$$u_{i+1} \approx u_i + T\frac{du}{dt} + \frac{T^2}{2}\frac{d^2u}{dt^2} \tag{33}$$

$$u_{i+1} = u_i + \frac{TV_i}{\|B'(u_i)\|} + \frac{T^2}{2}\left(\frac{A_i}{\|B'(u_i)\|} - \frac{V_i^2\langle B'(u_i), B''(u_i)\rangle}{\|B'(u_i)\|^4}\right) \tag{34}$$

$$B'(u_i) = \left.\frac{dB(u)}{du}\right|_{u=u_i} \tag{35}$$

$$B''(u_i) = \left.\frac{d^2B(u)}{du^2}\right|_{u=u_i} \tag{36}$$

$$u(s_{i+1}) = u(s_i) + \left.\frac{du}{ds}\right|_{s=s_i}\Delta s_i + \left.\frac{1}{2}\frac{d^2u}{ds^2}\right|_{s=s_i}\Delta s_i^2 \tag{37}$$

where $u_k$ is the curve parameter obtained from the previous interpolation step, and $T$ represents the interpolation period. $\Delta s_i$ is the arc length increment in the current interval.

The interpolation is directly computed using the anticipated increment $u$ without any compensatory measures, categorizing it as an open-loop Taylor direct interpolation. To address the aforementioned issues, initially, the second-order Taylor expansion is employed to calculate the initial value of the parameter $u$. Subsequently, an optimization model, considering both the arc length and parameters, is established:

$$u(s_{i+1}) = u(s_i) + \left.\frac{du}{ds}\right|_{s=s_i}\Delta s_i + \left.\frac{1}{2}\frac{d^2u}{ds^2}\right|_{s=s_i}\Delta s_i^2 \tag{38}$$

$$u_s = \arg\min_{u\in[0,1]}(B(u) - s)^2 \tag{39}$$

The entire system flow is illustrated in Figure 9.

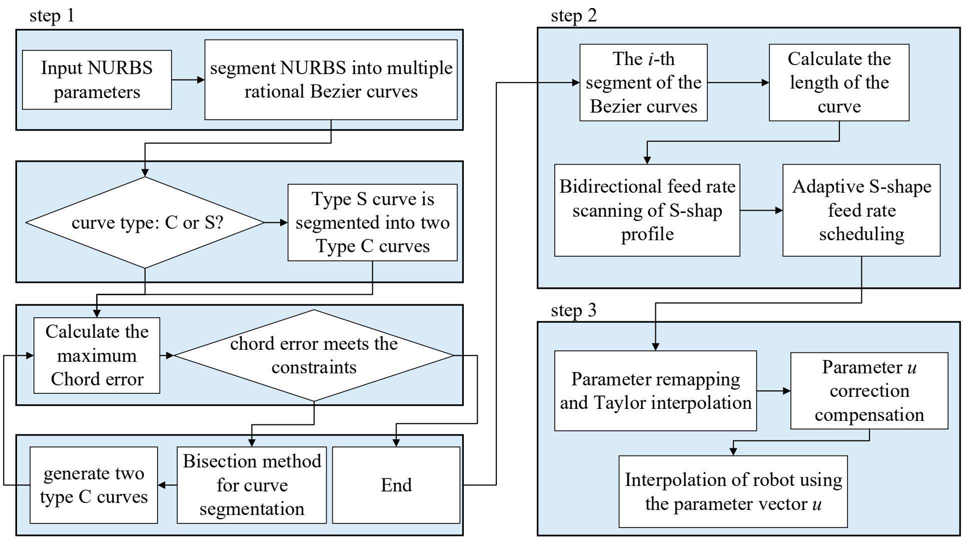

step 1
Input NURBS parameters
segment NURBS into multiple rational Bezier curves
curve type: C or S?
Type S curve is segmented into two Type C curves
Calculate the maximum Chord error
chord error meets the constraints
generate two type C curves
Bisection method for curve segmentation
End
step 2
The *i*-th segment of the Bezier curves
Calculate the length of the curve
Bidirectional feed rate scanning of S-shap profile
Adaptive S-shape feed rate scheduling
step 3
Parameter remapping and Taylor interpolation
Parameter *u* correction compensation
Interpolation of robot using the parameter vector *u*

**Figure 9.** Interpolation process of robot NURBS curve motion.

## 6. Experiment and Results

In order to verify the algorithm's robustness, two verification experiments are carried out in this section. The first one is used to verify the effectiveness of the high-precision NURBS interpolator proposed in this paper, and the second one is used to verify the practicality of the proposed robot tool tip and attitude synchronous interpolation.

### *6.1. NURBS Interpolator for Butterfly Curve*

In this section, simulations are conducted to show the effectiveness of the proposed method. The computer utilized has a CPU: I7-11800H-2.30 GHz (Intel Corporation, Santa Clara, CA, USA), 64 G-3200 MHZ memory, and the MATLAB (The MathWorks, Natick, MA, USA) version is 2022b. The plane butterfly NURBS curve (see Figure 10a) is employed as the test subject for algorithm evaluation. The chord error is set as $\delta = 0.001$ mm. Figure 10b shows the butterfly curve segmented into a type-C curve by the zero curvature point, and Figure 10c shows the butterfly curve segmented into 1579 type-C curves that satisfy $\delta = 0.001$ mm. Each Bezier curve is distinguished by color. In Figure 10d, with the chord error, the chord value is less than 0.001 mm, while the chord is a hundred times higher when without chord error control. In Figure 10e,f, respectively, the displacement and feed rate of the bidirectional feed rate scanning of the S-shaped profile are shown. Based on these figures, the plant area always has a longer Bezier curve, while the peak part has a shorter curve, which makes sense.

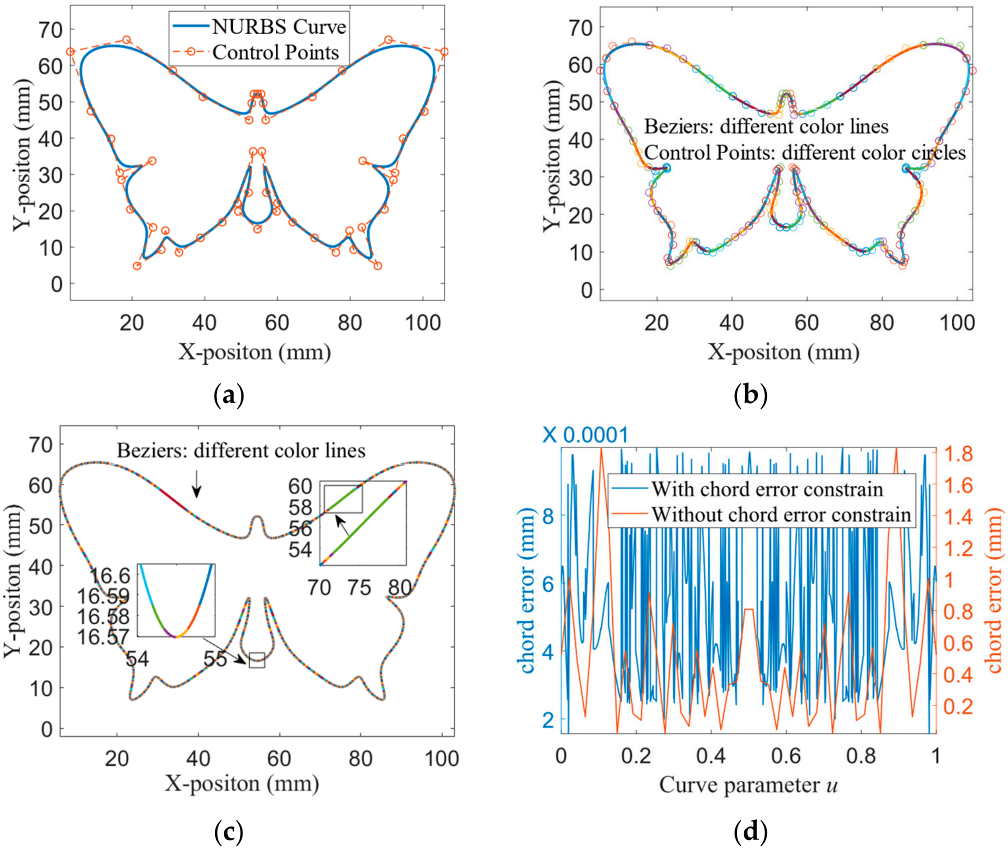
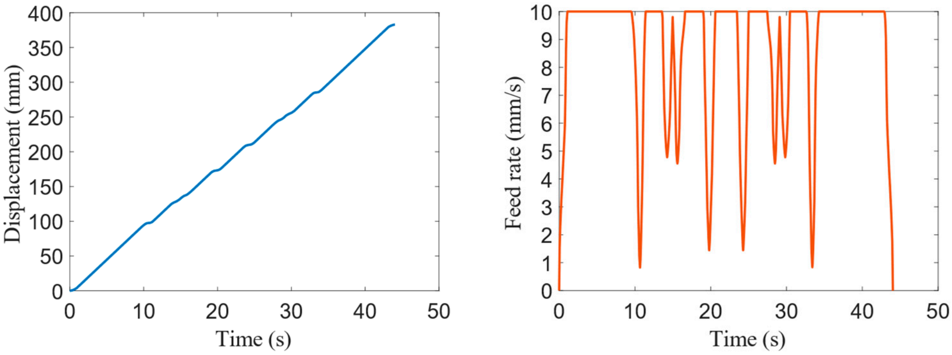


**Figure 10.** *Cont.*



(**e**) (**f**)

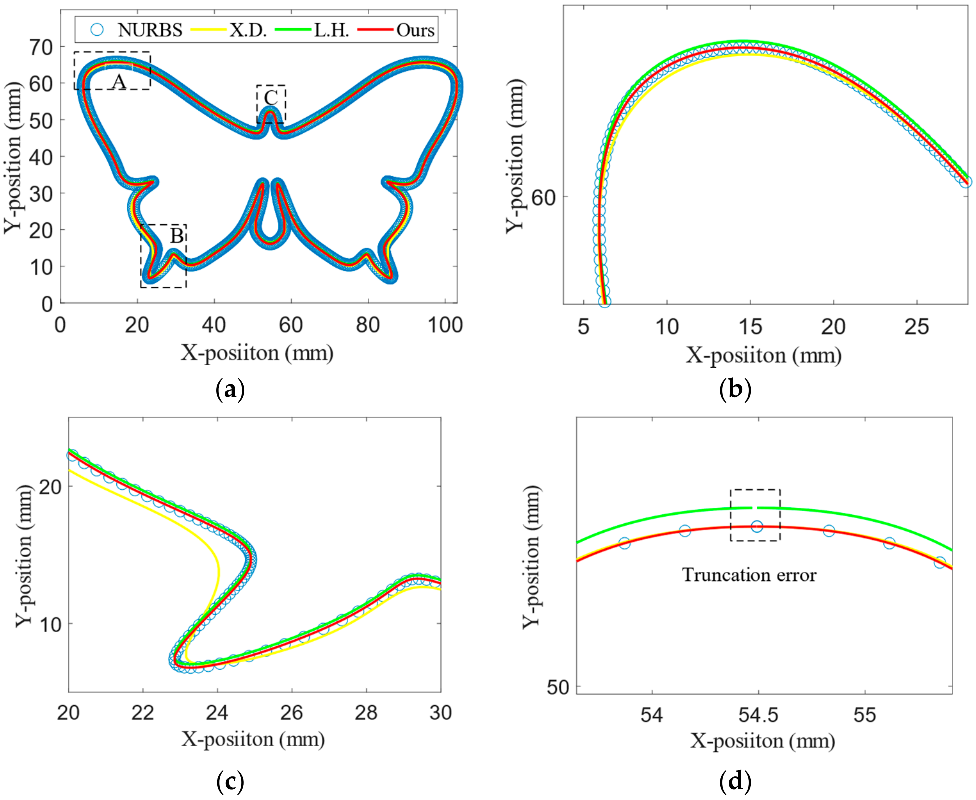
**Figure 10.** Simulation for butterfly NURBS curve: (**a**) butterfly NURBS curve; (**b**) curve segmented into type-C curve by zero curvature point; (**c**) type-C curves that satisfy $\delta = 0.001$ mm; (**d**) comparison with/without chord error constraint; (**e**) NURBS interpolation displacement; (**f**) NURBS interpolation feed rate.

Systematic comparisons are implemented between the proposed method and the other methods (L.H. [8], X.D. [14], L.F. [32]) mentioned in this paper. The different characteristics are compared and analyzed in Table 2. Due to the randomness in each run time, the recorded times in the table represent approximate values within the specified intervals. The methods employed by X.D. and L.F. fall under the category of open-loop chord error constraint curve segmentation, thus eliminating the need for iteration and avoiding an increase in time cost. On the other hand, L.H. and the method proposed in this paper primarily consume time in curve iteration and segmentation. Despite requiring more time compared to open-loop methods, the additional time cost is relatively small due to the chord error meeting the machining requirements, and its significance far outweighs the time cost. Regarding cumulative interpolation errors, both L.F. and the method proposed in this paper can achieve unbiased interpolation. However, L.F. incurs a longer processing time, primarily attributed to the continuous updating of the parameter $u$ in the binary search method, rendering it challenging for real-time interpolation processes. The total duration time increases compared to X.D., but this is acceptable because it meets the machining quality requirements, providing a certain level of efficiency and precision improvement compared to the other two methods.

**Table 2.** Performance parameters for different methods.

| Parameters | X.D. [14] | L.H. [8] | L.F. [32] | Ours |
|---|---|---|---|---|
| Number of iterations | 0 | 9 | 0 | 5 |
| Iterations' computational time (s) | 0 | 8 | 0 | 0.7 |
| Max chord error (mm) | 1.8 | 0.03 | 2 | 0.001 |
| End value of $u$ | 0.997 | 0.997 | 0.999 | 1 |
| Truncation error (mm) | 0.001 | 0.001 | 0 | 0 |
| Bidirectional scanning (s) | 0.01 | 0.1 | 0.01 | 0.3 |
| Once s-shaped feed rate profile (s) | 0.00006 | 0.00006 | 0.00006 | 0.00007 |
| Interpolation time (s) | 9 | 10 | 13 | 10 |
| Total duration time (s) | 10 | 19 | 14 | 12 |

Figure 11 illustrates the results of the segmentation and interpolation of NURBS using different methods. From Figure 11b,c, it can be observed that X.D. is subject to an open-loop chord error constraint, leading to significant trajectory deviation. L.H. involves altering control points to ensure chord error constraint, but such modifications result in trajectory displacement. Figure 11d presents truncation error, indicating that without closed-loop Taylor interpolation, the curve is influenced by cumulative interpolation and Taylor expansion truncation error, preventing closure. In contrast, the proposed method in this paper ensures trajectory accuracy while eliminating cumulative interpolation errors, resulting in an unbiased trajectory.



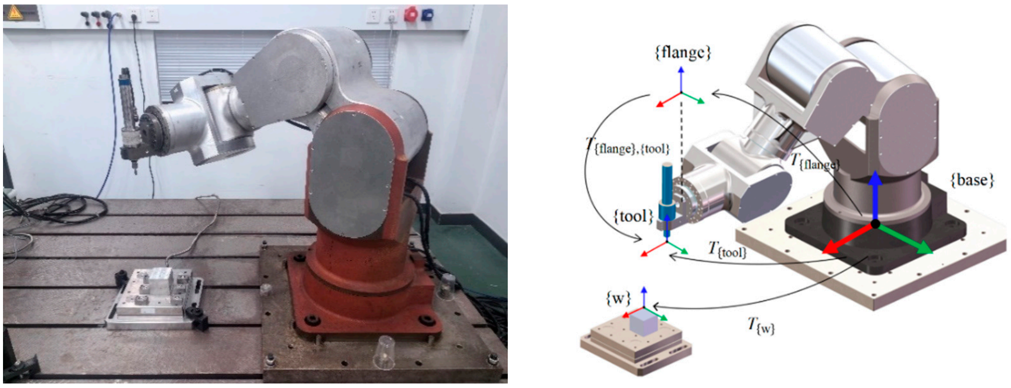
(**a**) (**b**)

(**c**) (**d**)

**Figure 11.** Curve segmentation and interpolation results: (**a**) interpolation of NURBS curves and the blue circles represent discrete points on the theoretical NURBS path; (**b**) detailed view of area A; (**c**) detailed view of area B; (**d**) detailed view of area C [8,14,32].

## 6.2. Experiment of Robot Path for a Fan NURBS Curve

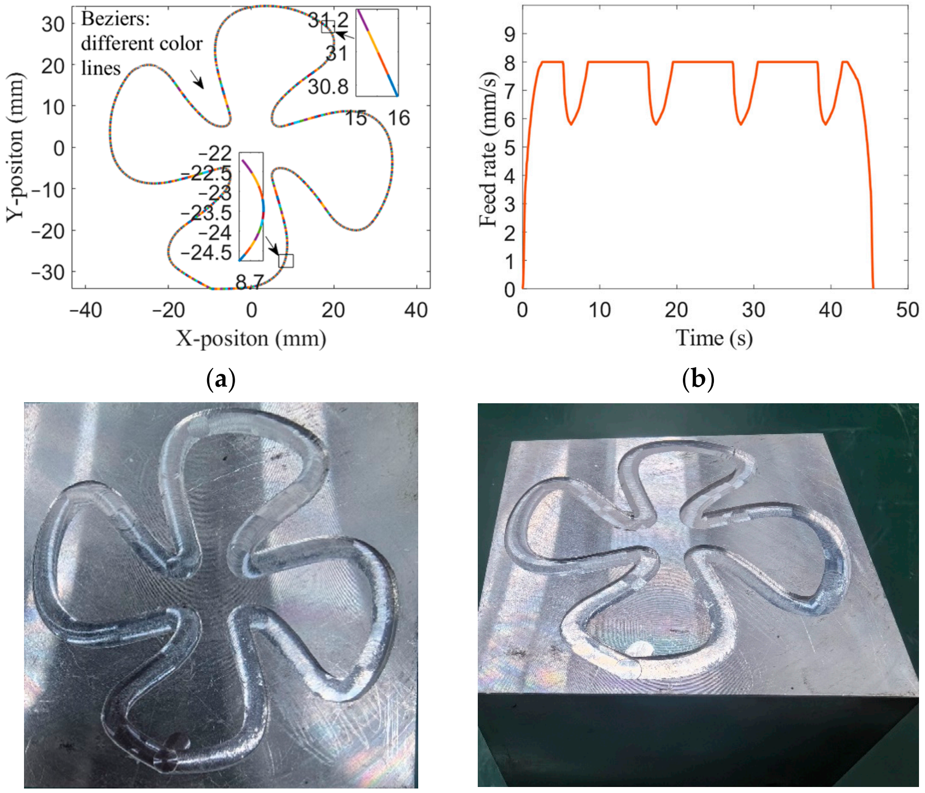
To demonstrate the proposed method, an experiment was conducted on a 6-DoF industrial robot. The robot machining system is shown in Figure 12. A complete robot system typically comprises three frames: the base frame, the tool frame, and the workpiece frame. The workpiece frame {w} is exclusively associated with the machining process and is unrelated to calibration. The base frame {base} is typically fixed beneath the first joint of the robot, serving as the primary reference system. The tool frame {tool} is located at the end of the robot.



(**a**) (**b**)

**Figure 12.** Robot machining system: (**a**) industrial robot; (**b**) frames.

The controller's hardware platform is equipped with the SpeedGoat RCP (The SpeedGoat, Bern, Switzerland) real-time simulation platform, featuring a computation cycle of 1 ms. The control system is developed using the MDH model and leverages the real-time functionality of MATLAB Simulink (The MathWorks, Natick, MA, USA). The computer utilized has a CPU: I7-11800H-2.30 GHz (Intel Corporation, Santa Clara, CA, USA), 64 G-3200 MHZ memory, and the MATLAB version is 2022b. In Figure 13a, the fan curve segmentation result is presented, employing the chord error constraint method with 1454 segments, and the maximum chord error is 0.001 mm. Figure 13b displays the bidirectional feed rate scanning of the S-shaped profile. The surface of the fan NURBS curve is shown in Figure 13c,d. It can be observed from the figures that the machining surface obtained by the proposed method exhibits no obvious overcutting, indicating a relatively smooth surface quality.



(**a**) (**b**)

(**c**) (**d**)

**Figure 13.** Fan NURBS curve: (**a**) fan curve segmentation result by using chord error segmentation method; (**b**) the feed rate of bidirectional feed rate scanning of S-shaped profile; (**c**) overhead view; (**d**) side view.

The proposed method was validated through a milling experiment using a planar fan NURBS curve path on a robot. The results confirm the correctness and effectiveness of the proposed approach. It is demonstrated that the method can generate smoother machining paths, leading to a smoother tool tip feed rate. Consequently, this significantly enhances both the efficiency and quality of robot machining.

## 7. Conclusions

This paper introduces a high-precision planar NURBS interpolation system for industrial robots. Leveraging knot insertion technology, the global NURBS curve is segmented into multiple rational Bezier subcurves that satisfy the chord error constraint. Subsequently, to address limitations in the current S-shaped profile, a real-time speed adjustment scheduling method is proposed to ensure feed rate adjustments within the lookahead range during bidirectional scanning. Additionally, by integrating segmented Bezier parameter remapping theory, the continuity of parameters at the junctions of two segments is preserved. The robustness of the proposed scheme is validated through butterfly NURBS curve simulations, and the effectiveness of the proposed approach is verified by an experiment conducted on an industrial robot using a fan NURBS curve.

**Author Contributions:** X.L. contributed the central idea and wrote the initial draft of the paper. Y.X., J.C., J.L. and Y.Z. contributed to the refining of the ideas and revised this paper. Software and Writing—original draft, X.L.; resources and writing—original draft, Y.X.; writing—review and editing, J.C., J.L. and Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Key Research and Development Program for Robotics Serialized Harmonic Reducer Fatigue Performance Analysis and Prediction and Life Enhancement Technology Research, Grant No. 2017YFB1300603.

**Data Availability Statement:** All data generated or analyzed during this study are included in this paper or are available from the corresponding authors on reasonable request. The data are not publicly available due to intellectual property protection.

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

## References

1. Abbasnejad, G.; Yoon, J.; Lee, H. Optimum kinematic design of a planar cable-driven parallel robot with wrench-closure gait trajectory. *Mech. Mach. Theory* **2016**, *99*, 1–18. [CrossRef]
2. De Souza, A.F.; Coelho, R.T. Experimental investigation of feed rate limitations on highspeed milling aimed at industrial applications. *Int. J. Adv. Manuf. Technol.* **2007**, *32*, 1104–1114. [CrossRef]
3. Piegl, L.; Tiller, W. *The NURBS Books*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 1996.
4. Elber, G.; Cohen, E. Tool path generation for freeform surface models. In Proceedings of the Second ACM Symposium on Solid Modeling and Applications, Montreal, QC, Canada, 19–21 May 1993; pp. 419–428.
5. Lasemi, A.; Xue, D.; Gu, P. Recent development in CNC machining of freeform surfaces: A state-of-the-art review. *Comput. Aided Des.* **2010**, *42*, 641–654. [CrossRef]
6. Sarkar, S.; Dey, P.P. A new iso-parametric machining algorithm for free-form surface. *Proc. IMechE Part E J. Eng. Manuf.* **2014**, *228*, 197–209. [CrossRef]
7. Lee, R.S.; She, C.H. Tool path generation and error control method for multi-axis NC machining of spatial cam. *Int. J. Mach. Tools Manuf.* **1998**, *38*, 277–290. [CrossRef]
8. Hua, L.; Huang, N.; Yi, B.; Zhao, Y.; Zhu, L. Global toolpath smoothing for CNC machining based on B-spline approximation with tool tip position adjustment. *Int. J. Adv. Manuf. Technol.* **2023**, *125*, 3651–3665. [CrossRef]
9. Bi, Q.; Huang, J.; Lu, Y.; Zhu, L.; Ding, H. A general, fast and robust B-spline fitting scheme for micro-line tool path under chord error constraint. *Sci. China Technol. Sci.* **2019**, *62*, 321–332. [CrossRef]
10. Huang, N.; Hua, L.; Huang, X.; Zhang, Y.; Zhu, L.; Biermann, D. B-spline-based corner smoothing method to decrease the maximum curvature of the transition curve. *J. Manuf. Sci. Eng.* **2022**, *144*, 054503. [CrossRef]
11. Ward, R.A.; Sencer, B.; Jones, B.; Ozturk, E. Five-axis trajectory generation considering synchronization and nonlinear interpolation errors. *J. Manuf. Sci. Eng.* **2022**, *144*, 081002. [CrossRef]
12. Shi, Z.; Zhang, W.; Ding, Y. A local toolpath smoothing method for a five-axis hybrid machining robot. *Sci. China Technol. Sci.* **2023**, *66*, 721–742. [CrossRef]
13. Lin, M.T.; Tsai, M.S.; Yau, H.T. Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. *Int. J. Mach. Tools Manuf.* **2007**, *47*, 2246–2262. [CrossRef]
14. Du, X.; Huang, J.; Zhu, L.M. A complete S-shape feed rate scheduling approach for NURBS interpolator. *J. Comput. Des. Eng.* **2015**, *2*, 206–217. [CrossRef]
15. Huang, J.; Du, X.; Zhu, L.M. Parallel acceleration/deceleration feedrate scheduling for computer numerical control machine tools based on bi-directional scanning technique. *Proc. IMechE Part B J. Eng. Manuf.* **2019**, *233*, 937–947. [CrossRef]
16. Xinhua, L.; Junquan, P.; Lei, S.; Zhongbin, W. A novel approach for NURBS interpolation through the integration of acc-jerk-continuous-based control method and look-ahead algorithm. *Int. J. Adv. Manuf. Technol.* **2017**, *88*, 961–969. [CrossRef]
17. Bollinger, J.G.; Duffie, N.A. *Computer Control of Machines and Processes*; Addison-Wesley Longman Publishing Co. Inc.: Reading, MA, USA, 1988.
18. Ishizaki, K.; Shamoto, E. A new real-time trajectory generation method modifying trajectory based on trajectory error and angular speed for high accuracy and short machining time. *Precis. Eng.* **2022**, *76*, 173–189. [CrossRef]
19. He, S.; Yan, C.; Deng, Y.; Lee, C.H.; Zhou, X. A tolerance constrained G2 continuous path smoothing and interpolation method for industrial SCARA robots. *Robot Comput. Integr. Manuf.* **2020**, *63*, 101907. [CrossRef]

20. Barnett, E.; Gosselin, C. A Bisection Algorithm for Time-Optimal Trajectory Planning along Fully Specified Paths. *IEEE Trans. Robot.* **2020**, *37*, 131–145. [CrossRef]
21. Mora, P.R. *On the Time-Optimal Trajectory Planning along Predetermined Geometric Paths and Optimal Control Synthesis for Trajectory Tracking of Robot Manipulators*; University of California: Berkeley, CA, USA, 2013.
22. Lee, A.C.; Lin, M.T.; Pan, Y.R.; Lin, W.Y. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Comput. Aided Des.* **2011**, *43*, 612–628. [CrossRef]
23. Lai, J.Y.; Lin, K.Y.; Tseng, S.J.; Ueng, W.D. On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk. *Int. J. Adv. Manuf. Technol.* **2008**, *37*, 104–121. [CrossRef]
24. Nam, S.H.; Yang, M.Y. study on a generalized parametric interpolator with real-time jerk-limited acceleration. *Comput. Aided Des.* **2004**, *36*, 27–36. [CrossRef]
25. Nguyen, K.D.; Ng, T.C.; Chen, I.M. On algorithms for planning s-curve motion profiles. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 11. [CrossRef]
26. Bai, Y.; Chen, X.; Yang, Z. A generic method to generate AS-curve profile in commercial motion controller. In Proceedings of the ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Cleveland, OH, USA, 6–9 August 2017.
27. Shpitalni, M.; Koren, Y.; Lo, C.C. Realtime curve interpolators. *Comput. Aided Des.* **1994**, *26*, 832–838. [CrossRef]
28. Yeh, S.S.; Hsu, P.L. Adaptive-feedrate interpolation for parametric curves with a confined chord error. *Comput. Aided Des.* **2002**, *34*, 229–237. [CrossRef]
29. Tikhon, M.; Ko, T.J.; Lee, S.H.; Kim, H.S. NURBS interpolator for constant material removal rate in open NC machine tools. *Int. J. Mach. Tools Manuf.* **2004**, *44*, 237–245. [CrossRef]
30. Erwinski, K.; Paprocki, M.; Karasek, G. Comparison of NURBS trajectory interpolation algorithms for high-speed motion control systems. In Proceedings of the 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC), Gliwice, Poland, 25–29 April 2021; pp. 527–533.
31. Huang, J.; Zhu, L.M. Feedrate scheduling for interpolation of parametric tool path using the sine series representation of jerk profile. *Proc. IMechE Part B J. Eng. Manuf.* **2017**, *231*, 2359–2371. [CrossRef]
32. Fang, L.; Liu, G.; Li, Q.; Zhang, H. A high-precision non-uniform rational B-spline interpolator based on S-shaped feedrate scheduling. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 2585–2595. [CrossRef]
33. Boehm, W. Inserting new knots into B-spline curves. *Comput. Aided Des.* **1980**, *12*, 199–201. [CrossRef]
34. Lei, W.T.; Sung, M.P.; Lin, L.Y.; Huang, J.J. Fast real-time NURBS path interpolation for CNC machine tools. *Int. J. Mach. Tools Manuf.* **2007**, *47*, 1530–1541. [CrossRef]
35. Zhao, H.; Zhu, L.M.; Ding, H. A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segment. *Int. J. Mach. Tools Manuf.* **2013**, *65*, 88–98. [CrossRef]

**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.