mathematics

MDPI

Article

A Chord Error-Priority Bilevel Interpolation Optimization Method for Complex Path Planning

Pengxuan Wei [1], Liping Wang [1,2], Dan Wang [3], Jun Qi [3] and Xiaolong Ye [1,*]

[1] School of Mechanical and Electrical Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2] State Key Laboratory of Tribology in Advanced Equipment, Department of Mechanical Engineering, Tsinghua University, Beijing 100084, China
[3] School of Electronic Information and Electrical Engineering, Chengdu University, Chengdu 610106, China
* Correspondence: yxl0608@uestc.edu.cn

Abstract

To address path deviation and efficiency reduction issues in traditional interpolation optimization algorithms for complex path machining, this paper proposes a chord error-priority bilevel interpolation optimization method (CPBI). First, arc length parametric modeling of the machining path is performed within the Frenet–Serret framework, yielding curvature and torsion information. After introducing geometric-based multi-machining constraints in the outer layer, the velocity upper limit is established by controlling chord error to dynamically adjust regions with curvature mutation. In the inner layer, combining the velocity limit with bidirectional scanning achieves adaptive optimization of interpolation step size and optimal velocity planning that balances precision and smoothness. Simulation results demonstrate that CPBI effectively reduces the number of interpolation points by 30–50% while ensuring the chord error. Compared with the reference method, the CPBI improved efficiency by 14.31% and 34.72% in machining experiments on S-shaped and wave-shaped paths, respectively. The results validated the CPBI's high precision and efficiency advantages in complex path machining, providing an effective solution for CNC path optimization in high-end manufacturing.

Keywords: chord error; interpolation optimization; path modeling; Frechet-Serret framework; machining efficiency

MSC: 68U07

check for updates

Academic Editor: Aleksandr Rakhmangulov

Received: 3 October 2025
Revised: 18 October 2025
Accepted: 20 October 2025
Published: 24 October 2025

Citation: Wei, P.; Wang, L.; Wang, D.; Qi, J.; Ye, X. A Chord Error-Priority Bilevel Interpolation Optimization Method for Complex Path Planning. *Mathematics* **2025**, *13*, 3385. https://doi.org/10.3390/math13213385

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

Path interpolation optimization has important applications in the machining of complex workpieces within high-end manufacturing sectors such as aerospace and precision molds. Since CNC machining achieves material removal by guiding cutting tools along predetermined paths, discontinuities in these paths and velocity irregularities often cause acceleration/deceleration impacts on machine tools, leading to machining errors and surface quality degradation [1–3]. To effectively control such issues, it is necessary to achieve path smoothing and rational feed rate allocation through interpolation optimization before machining. Existing research has proposed various interpolation optimization methods. Among them, methods based on polynomials or spline curves can reduce machining errors and optimize feed rates. However, in actual complex path machining, excessive velocity fluctuations or failure to approach the upper limit of permissible velocity often

occur, thereby affecting machining efficiency and quality [4,5]. Therefore, it is essential to investigate interpolation optimization methods for complex paths. The objective is to ensure smooth paths and machining accuracy while enhancing the machine tool's velocity utilization and overall machining efficiency.

Existing research on machining path interpolation optimization primarily focuses on curve parameterization and CNC constraint models, forming two main lines of inquiry centered on path interpolation and feed rate optimization [6–8].

The research on machining path interpolation has primarily focused on improving interpolation algorithms to enhance path accuracy, smoothness, and machining efficiency [8,9]. To overcome the issues of velocity discontinuity and acceleration mutation caused by traditional linear interpolation, parametric curve interpolation (such as NURBS and B-spline curves) has become the mainstream research approach [10]. Jia et al. proposed an adaptive NURBS curve interpolation algorithm that effectively reduces contour errors by dynamically adjusting the feed rate in real time to accommodate curvature variations [11]. Wang et al. deeply integrated feed rate scheduling with the interpolation process, considering machine tool dynamic constraints before generating interpolation points to achieve high-precision predictive interpolation [12]. Song et al. focused on the smooth connection of short-segment paths, employing B-spline curves to smooth discrete paths, significantly reducing machine tool vibration during machining [13]. However, most of the above studies focus on macro-path interpolation, while challenges remain in achieving smoothness and velocity planning at corners for high-precision machining paths. To address this issue, some scholars have proposed intelligent interpolation methods to enhance interpolation performance under complex paths. Sun et al. developed a neural network-based interpolation points prediction model that rapidly generates smooth interpolation sequences meeting accuracy requirements by learning optimal machining data [14]. Ward proposed a data-driven framework for adaptive adjustment of interpolation parameters to address uncertainties in the machining process [15]. Although these intelligent methods demonstrate significant potential in specific scenarios, they suffer from issues such as high dependence on training data and poor reliability when deployed in actual CNC systems.

The research on feed rate optimization focuses on generating continuous velocity curves that satisfy machine tool physical constraints, thereby reducing machining time while ensuring processing quality [16,17]. S-curve velocity planning technology serves as the foundational research in this field [18]. Sencer et al. effectively addressed the corner velocity issue by smoothly connecting multi-segment S-curves while considering trajectory contour error constraints [19]. Sun et al. effectively reduced velocity fluctuations in regions of curvature change by introducing adaptive NURBS interpolation with curvature and driving constraints [20]. Sang et al. optimized the feed rate by establishing a linear relationship between geometric error and feed rate, combined with kinematic constraints [21]. Xu et al. optimized machining paths by proposing a dual NURBS interpolation technique for primary and secondary surfaces, thereby enhancing machining stability and precision [2]. Liu et al. employed arc length compensation and iterative methods to reduce step error and feed rate fluctuations [22]. Although the above studies mostly employ curvature and kinematic parameters as constraint objectives, they provide insufficient support for step-size adaptation and global error consistency on paths with frequent curvature changes at multiple locations. In recent years, scholars have begun exploring feed rate optimization methods based on optimization theory to achieve optimal quality or efficiency. Zhao et al. modeled the feed rate optimization problem as a constrained optimization problem and solved it using the pseudo-spectral method, obtaining the optimal feed rate curve [23]. Yang et al. proposed a feed rate optimization method based on the PSO to balance machining time and accuracy, achieving multi-objective optimization [17]. Wang et al. introduced

model prediction methods for feed rate optimization, enhancing the robustness of feed rate scheduling by compensating for model errors through rolling optimization [24]. Although these optimization methods can achieve superior performance, their substantial computational demands limit their applicability, making them primarily used for interpolation optimization in simplified models. Balancing optimality and applicability remains a key challenge in this field.

To address these issues, this paper proposes a chord error-priority bilevel interpolation optimization method, offering a novel solution for achieving high-precision and high-efficiency interpolation in complex machining paths. The method's advantage lies in its geometry-driven path modeling and the introduction of multiple kinematic constraints. Utilizing the Frenet–Serret framework to extract curve information ensures path continuity and preservation of geometric properties. Simultaneously, the concept of bilevel interpolation optimization was introduced, which not only ensures minimal chord error but also achieves adaptive optimization of interpolation step size and feed rate through a bidirectional scanning method.

The structure of this paper is as follows: Section 2 covers path modeling, Section 3 details the proposed interpolation method, and Section 4 presents simulation verification.

2. Machining Path Modeling

2.1. Arc Length Parameterization of Path

In practical machining, complex paths are typically composed of multiple segments of curves with fast curvature changes. This paper uses non-uniform B-spline curves as an example for path modeling. First, the curve construction is defined according to Equation (1).

$$P(u) = \sum_{i=0}^{n} d_i N_{i,k}(u) \tag{1}$$

where d_i $(i = 1, 2, 3 \ldots n)$ denotes the control points of the curve, totaling $n + 1$ points. $N_{i,k}(u)$ $(i = 0, 1, 2 \ldots n)$ represents the curve's basis functions, with k being the degree of the basis function. It also indicates the repetition degree of the curve generated by the control points at the start and end positions. The basis functions of a B-spline curve are kth-degree piecewise polynomials determined by the knot vectors u, where the parameters u satisfy $u_0 \leq u_1 \leq u_2 \ldots \leq u_n$. The recursive equation for a B-spline curve is as follows.

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

Then the derivative of the curve with respect to u can be inferred from the derivative of the basis function:

$$P_u(u) = \sum_i N'_{i,k}(u) d_i \tag{3}$$

$$P_{uu}(u) = \sum_i N''_{i,k}(u) d_i \tag{4}$$

$$P_{uuu}(u) = \sum_i N'''_{i,k}(u) d_i \tag{5}$$

The explicit expression for the derivative of a first-order basis function is as follows:

$$N'_{i,p}(u) = \frac{k}{u_{i+p} - u_i} N_{i,k-1}(u) - \frac{k}{u_{i+p+1} - u_{i+1}} N_{i+1,k-1}(u) \tag{6}$$

Terms with a denominator of zero are treated as zero. Higher-order derivatives of basis functions can be derived recursively as follows:

$$N''_{i,p}(u) = \frac{k}{u_{i+k} - u_i} N'_{i,k-1}(u) - \frac{k}{u_{i+k+1} - u_{i+1}} N'_{i+1,k-1}(u) \tag{7}$$

The advantage of using arc length as the parameter is that it is independent of the feed rate along the curve and is also unaffected by the choice of coordinate system. The arc length parameterization of the machining curve obtains the arc length function as shown in Equation (8). Subsequently, the tangential velocity is calculated as shown in Equation (9).

$$s(u) = \int_{u_0}^{u} \|P_u(\xi)\| d\xi \tag{8}$$

$$\|P_u(u)\| = \frac{ds}{du} \tag{9}$$

The value of $s(u)$ is obtained through Gaussian integration and combined with Newton's iteration method to invert for $u(s)$, as shown in Equation (10) [25]. Finally, the arc length parameter curve can be calculated as in Equation (11):

$$u_{i+1} = u_i - \frac{s(u_i) - s(u)}{\|P_u(u_i)\|} \tag{10}$$

$$P(s) = P(u(s)) \tag{11}$$

2.2. Path Curve Expansion Based on the Frenet–Serret Framework

The Frenet–Serret framework is a coordinate system method for automatic path planning. It describes the local geometry of a path curve using two scalar quantities (curvature and torsion) and provides an orthogonal, concise, and intuitionistic local coordinate system. This paper adopts this framework to further expand the machining path, which is structured as follows:

$$T = \frac{dP}{ds} \tag{12}$$

$$\kappa = \left\| \frac{dT}{ds} \right\| \tag{13}$$

$$N = \frac{1}{\kappa} \frac{dT}{ds} \tag{14}$$

$$B = T \times N \tag{15}$$

$$\tau = -\frac{dB}{ds} \cdot N \tag{16}$$

where T, N, and B represent the tangent, principal normal, and secondary normal vector, respectively. κ denotes curvature, and τ denotes torsion. Combining the above formulas, the curvature and torsion of the curve are expressed as follows:

$$\kappa(u) = \frac{\|P_u \times P_{uu}\|}{\|P_{uuu}\|^3} \tag{17}$$

$$\tau(u) = \frac{\det(P_u, P_{uu}, P_{uuu})}{\|P_u \times P_{uu}\|^2} \tag{18}$$

Then the arc length derivative is obtained as in Equation (19).

$$\kappa'(u) = \frac{d\kappa}{ds} = \frac{\kappa_u}{\|P_u\|} \tag{19}$$

The value of κ_u can be obtained by the difference between equal-arc-length meshes of the curve.

The machining path $P(u)$ and arc-length path $P(s)$ obtained from Section 2.1 have a fixed interpolation period of Δt. Let the arc length velocity be $v(s) = ds/dt$, and the single-step arc length be $\Delta s = v\Delta t$. The chord error represents the maximum deviation of the chord connecting the two machining interpolation points from the intended motion path, as shown in Figure 1. It can be seen that during actual machining, the tool's interpolation path moves in a straight line along the machining point, resulting in the actual path failing to coincide with the desired path. Moreover, this phenomenon becomes more pronounced in areas of greater curvature, leading to larger velocity fluctuations and potentially causing single-axis acceleration to exceed limits. Therefore, it is necessary to control the chord error to avoid this situation as much as possible, thereby ensuring machining smoothness and improving machining quality. The equation for chord error is as follows.

$$\varepsilon = dist(P(s_0 + \frac{\Delta s}{2}), \overline{P(s_0)P(s_0 + \Delta s)}) \tag{20}$$

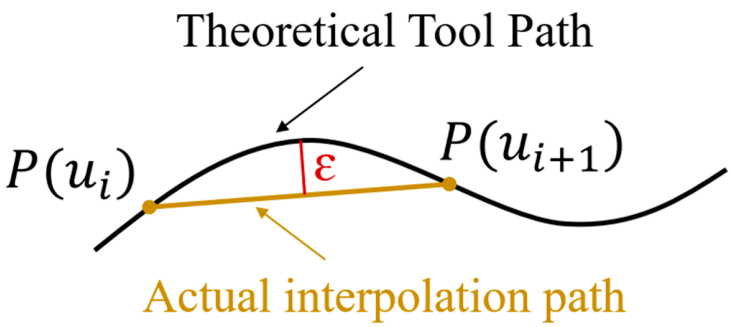


Figure 1. Chord difference and parameter curve points.

Perform a Taylor expansion at s_0 based on the Frenet–Serret framework.

$$P(s_0 + \Delta s) = P_0 + T_0\Delta s + \frac{1}{2}\kappa_0 N_0 \Delta s^2 + \frac{1}{6}(-\kappa_0^2 T_0 + \kappa_0' N_0 + \kappa_0 \tau_0 B_0)\Delta s^3 + O(\Delta s^4) \tag{21}$$

The expression for the chord error can be derived from Equations (20) and (21). For the specific derivation process, refer to Appendix A. The triangle inequality can be used to derive the chord error constraint.

$$\varepsilon \leq \frac{\kappa \Delta s^2}{8} + \frac{(|\kappa'| + \kappa|\tau|)\Delta s^3}{24} \leq \varepsilon_{\text{target}} \tag{22}$$

where ε_{target} represents the maximum target chord error. This ensures reasonable optimization of the machining step size under the constraint of maximum chord error. Let $a = \frac{(|\kappa'| + \kappa|\tau|)}{24}$ and $b = \frac{\kappa}{8}$. This yields the current step size constraint, which is dense in areas of high curvature and sparse in areas of low curvature.

$$a\Delta s^3 + b\Delta s^2 - \varepsilon_{\text{target}} \leq 0 \tag{23}$$

2.3. Geometry-Driven Machining Constraints

In actual machining operations, the feed rate of machine tools is often subject to an upper limit. Assuming the feed rate is v, common constraints include the following points:

Comprehensive velocity constraints for each axis:

$$v \leq v_{\max} \tag{24}$$

Tangential acceleration constraint:

$$a_t = \frac{dv}{dt} \leq a_{t\max} \tag{25}$$

Normal acceleration constraint:

$$a_n = v^2\kappa \leq a_{n\max} \tag{26}$$

Jerk Constraint:

$$\dot{a}_n = 2v\dot{v}\kappa + v^3\kappa' + v^3\kappa\tau \leq j_{\max} \tag{27}$$

where $\dot{v}$ denotes the first derivatives of velocity with respect to arc length. κ' denotes the first derivative of curvature with respect to arc length. For the specific derivation process, refer to Appendix B. From the above five constraints, the point-state velocity upper limit of the machining tools is derived as follows.

$$v_{p\max}(s) = \min\left\{v_{\max}, \sqrt{\frac{a_{n\max}}{\kappa(s)}}, \left(\frac{j_{\max}}{|\kappa'(s)|}\right)^{1/3}, \left(\frac{j_{\max}}{\kappa(s)|\tau(s)|}\right)^{1/3}\right\} \tag{28}$$

where v_{max} is the upper limit of velocity, a_{tmax} and a_{nmax} are the upper limits of tangential and normal acceleration, respectively, and j_{max} is the upper limit of jerk acceleration.

3. A Chord Error-Priority Bilevel Interpolation Optimization Method

3.1. Outer Layer—Global Velocity Optimization of Chord Error-Priority

From the single-step arc length, it follows that at each position s, the following inequality for feed rate v is present.

$$\left(\frac{|\kappa'| + \kappa|\tau|}{24}\Delta t^3\right)v^3 + \left(\frac{\kappa}{8}\Delta t^2\right)v^2 - \varepsilon_{\text{target}} \leq 0 \tag{29}$$

Taking the maximum positive real root of this cubic polynomial as the velocity upper limit $v_\varepsilon(s)$ dominated by the chord error, then Equation (30) can be obtained.

$$v_\varepsilon(s) = \max\left\{v > 0 \middle| a(s)v^3 + b(s)v^2 - \varepsilon_{\text{target}} \leq 0\right\} \tag{30}$$

If $a(s) = 0$, then $|\kappa'|+\kappa|\tau|$ is minimal, yielding the quadratic solution $v_\varepsilon(s) = \sqrt{\varepsilon_{max}/b(s)}$; if $b(s) = 0$, then κ is minimal, yielding the cubic solution $v_\varepsilon(s) = \sqrt[3]{\varepsilon_{max}/a(s)}$.

3.2. Inner Layer—Interpolation Optimization Under Comprehensive Constraints

The comprehensive upper limit for feed rates can be derived from the machine tool point-state velocity limit defined by Equation (28).

$$\overline{v}(s;\varepsilon) = \min\left\{v_{p\max}(s), v_\varepsilon(s)\right\} \tag{31}$$

Tangential acceleration can be expressed as $|a_t| = |dv/dt| < a_{tmax}$, and in the arc length domain it can be represented as follows.

$$\left|\frac{dv}{ds}\right| = \frac{|dv/dt|}{v} \leq \frac{a_{t\max}}{v} \tag{32}$$

To avoid feed rate interpolation distortion caused by excessive step size, further constraints on the step size are required, as shown in Equation (33).

$$\Delta s_{dis} \leq \eta \frac{v(s)^2}{a_{t\max}} \tag{33}$$

where η represents the step velocity variation parameter for adjacent tool positions, with $\eta = 0.15$. Combining Equations (23) and (33) yields the final optimized step size as shown in Equation (34).

$$\Delta s = \min\{\Delta s_{\varepsilon}, \Delta s_{dis}\} \tag{34}$$

Based on the optimal step size, the forward and backward scanning equation for the interpolation point obtained from Equation (34) is given by Equations (35) and (36).

$$v_{i+1} \leq \min\left\{\overline{v}_{i+1}^2, x_i + 2a_{t\max}\Delta s\right\} \tag{35}$$

$$v_i \leq \min\left\{\overline{v}_i^2, x_{i+1} + 2a_{t\max}\Delta s\right\} \tag{36}$$

$$s_{i+1} = s_i + \Delta s = s_i + v_i \Delta t \tag{37}$$

where Equation (35) represents forward scanning, while Equation (36) represents backward scanning. s_i denotes the i-th discrete arc length parameter, $i = 0, \ldots, N-1$. N represents the number of interpolation points. Δt is the fixed interpolation period. The initial point location s_0 for bidirectional scanning is selected as the starting point of the machining path. The initial value v_0 of the scanning iteration is set as the comprehensive upper limit of the velocity. The iteration termination condition is based on a fixed scanning count k_{max}, where $k_{max} = 2$. The value of k_{max} is determined according to the complexity of the path. The bidirectional scanning proceeds until iteration ceases to update, ultimately yielding the maximum feasible interpolation point s_i that satisfies multiple constraints and the upper limit of chord error.

According to the scanning equations, a higher average velocity corresponds to a larger arc length per step. Consequently, a shorter number of interpolation points is required for a given machining path. During forward scanning only, the kinematic constraints of subsequent points remain unknown, resulting in conservative velocity estimates for interpolated points. However, backward scanning acquires information from the forward scanning pass. Since the constraints are now known, the velocity of interpolated points can be adjusted to approach the upper limit more closely. Therefore, employing the bidirectional scanning can effectively reduce the number of interpolated points required.

Let $v_i = \sqrt{x_i}$, then the actual machining time can be expressed by Equation (38).

$$T(\varepsilon) = \int_0^L \frac{1}{v_*(s;\varepsilon)} ds = \sum_{i=0}^{N-1} \frac{\Delta s}{\max(v_i, \underline{v})} \tag{38}$$

where L denotes the total arc length of the machining path; $v_*(s;\varepsilon)$ represents the optimal velocity solution; $\underline{v}$ is the smallest positive number to prevent the denominator from being 0. For a given ε, Equation (38) yields the shortest-time discrete solution under the constraints of Equations (33)–(37).

Based on the parametric method for tool paths and the analysis of interpolation optimization for inner and outer layers, the method flowchart for the chord error-priority bilevel interpolation optimization method is shown in Figure 2. Based on Figure 2, the specific implementation steps of this method are as follows:

- Step 1: Initialize kinematic upper limits based on machine tool parameters.
- Step 2: Parametric machining paths based on the Frenet–Serret framework.

- Step 3: Calculate and set multi-kinematic constraints driven by geometry.
- Step 4: By determining the constraints of chordal error, the maximum feed rate is then calculated.
- Step 5: Based on comprehensive constraints, the feed rate and optimal step length are calculated using forward and backward scanning methods to determine the interpolation points.

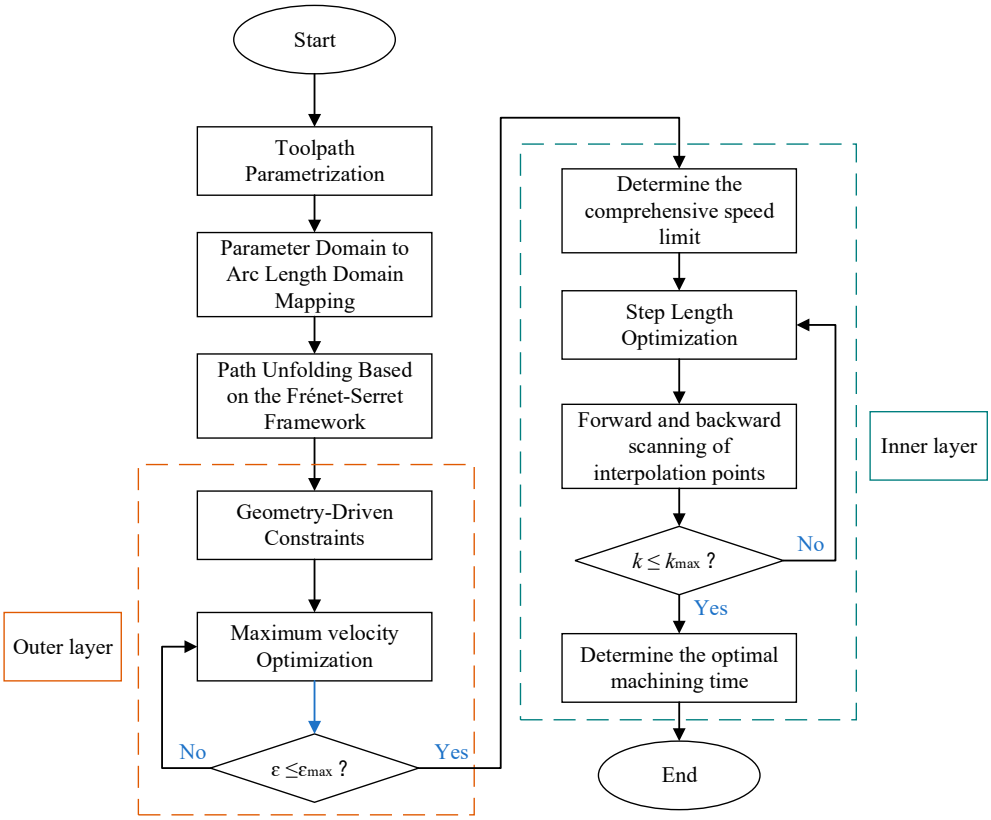


Figure 2. Flowchart of the chord error-priority bilevel interpolation optimization method.

4. Simulation Verification and Comparison

4.1. Machining Simulation Experiment for S-Shaped Path

To validate the performance of the proposed chord error-priority bilevel interpolation optimization method, this paper first selects an S-shaped machining path for simulation experiments. The S-shaped machining path is achieved by constructing a quasi-uniform rational B-spline curve of order 3, i.e., $k = 3$. The endpoints have order $k + 1$, while the intermediate nodes are approximated as equidistant. There are control points d_0, d_1, d_2, …, d_n. The basis functions for the i-th segment $(i = 0, 1, 2, \ldots, n - 3)$ are given by Equation (39).

$$\begin{cases} N_{i,3}(u) = \frac{1}{6}(-u^3 + 3u^2 - 3u + 1) \\ N_{i+1,3}(u) = \frac{1}{6}(3u^3 - 6u^2 + 4) \\ N_{i+2,3}(u) = \frac{1}{6}(-3u^3 + 3u^2 + 3u + 1) \\ N_{i+3,3}(u) = \frac{1}{6}u^3 \end{cases} \quad u \in [0, 1] \tag{39}$$

Based on the parameters in the above equation, the specific form of each segment can be expressed as Equation (39). The final processed path obtained is an S-shaped quasi-uniform rational curve, as shown in Figure 3. In this case, the number of control points is 16, and the node vector is $U = \{0\ 0\ 0\ 0\ 0.08\ 0.15\ 0.23\ 0.31\ 0.38\ 0.46\ 0.54\ 0.62\ 0.69\ 0.71\ 0.85\ 0.92\ 1\ 1\ 1\ 1\}$. The program for generating machining paths can be found in the Supplementary Materials documents.

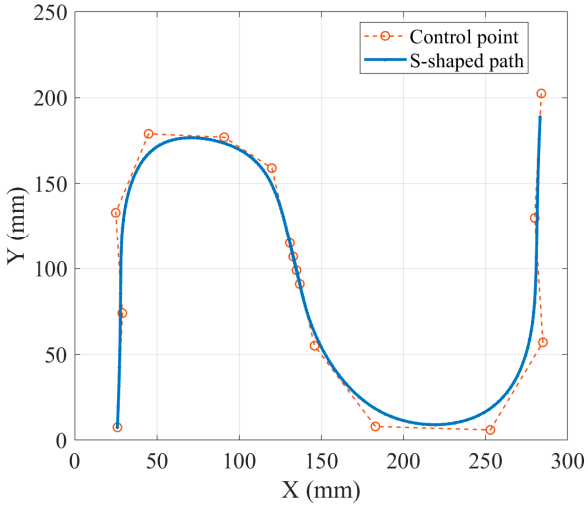


Figure 3. S-shaped machining path and control points.

Based on the characteristics of the S-shaped machining path, it is necessary to understand how the machining path varies with changes in the parameter domain, as shown in Figure 4. Simultaneously, the velocity of the path curve relative to parameter u is represented as shown in Figure 5. Further conversion of the parameter domain of this machining path into the arc length domain is shown in Figure 6. It can be observed that the arc length is unequal in steps in the parameter domain, which further highlights the necessity of velocity limiting in the arc length domain.

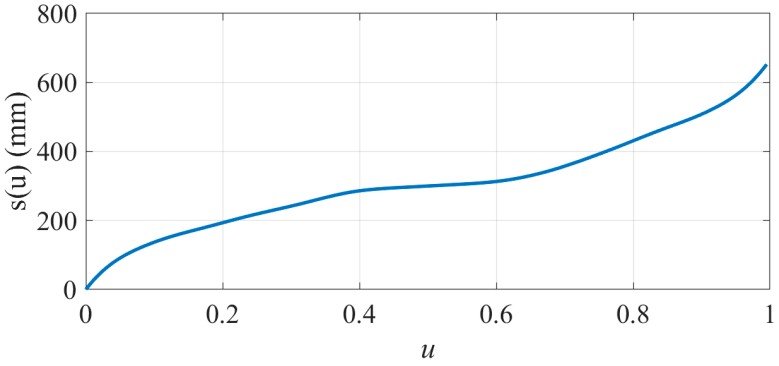


Figure 4. Parameter domain function of the path.

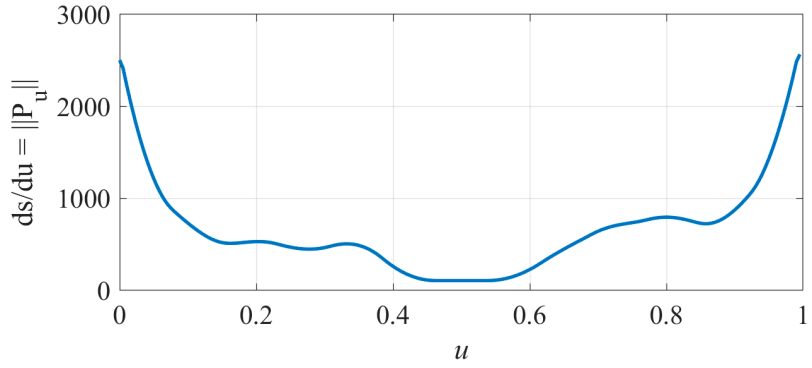


Figure 5. The rate of change in arc length with respect to the parameter u.

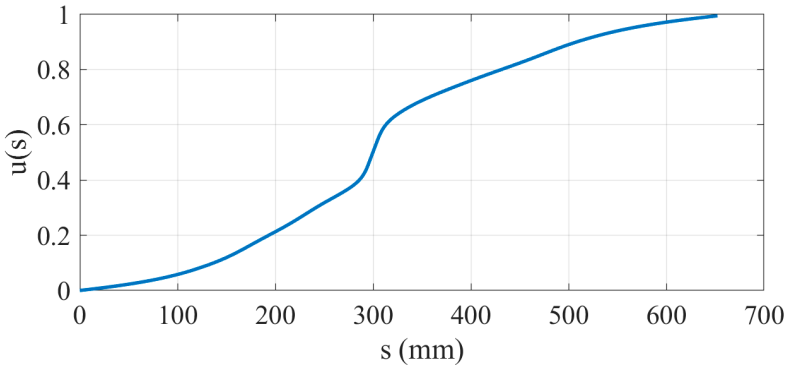
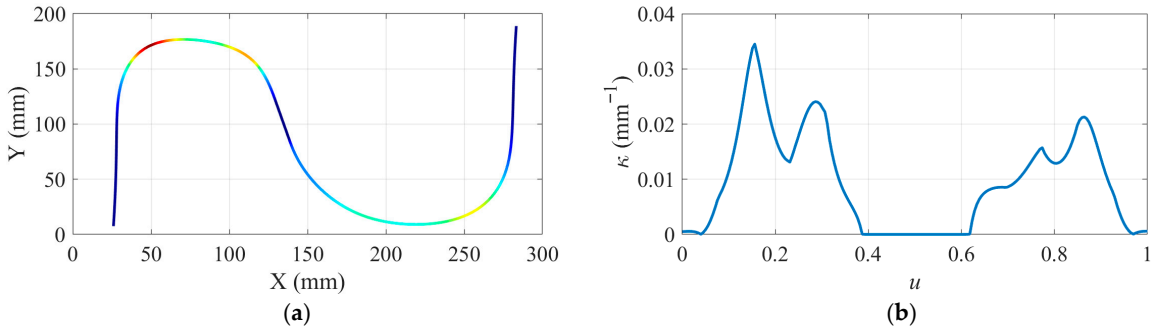


Figure 6. Mapping from the parameter domain to the arc length domain.

Since this paper expands the machining path based on the Frenet–Serret framework, curvature information is indispensable. The curvature visualization of the S-shaped machining path is shown in Figure 7. It can be observed that the curve exhibits abrupt curvature changes at certain positions, which easily restrict the feed rate and impact machining efficiency. Therefore, performing interpolation optimization at these positions can effectively achieve smooth transitions in the machining path.



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

Figure 7. Curvature analysis of machining paths: (**a**) attachment of path curvature (color indicates the magnitude of curvature); (**b**) curve of path curvature.

Based on path characteristics and machining requirements, it is necessary to set up complete simulation constraints. These include geometric parameters of the machining path, interpolation cycles, and multiple kinematic constraints. Specific experimental parameters are shown in Table 1.

Table 1. Experimental parameters for S-shaped machining path.

Experimental Parameters	Value
T	0.002 s
ε_{max}	0.001 mm
$\overline{v}$	70 mm/s
a_{nmax}	800 mm/s^2
a_{tmax}	750 mm/s^2
j_{max}	10,000 mm/s^3

Combining the experimental parameters from Table 1, the proposed method CPBI was applied to optimize the S-shaped path through simulation. The resulting path interpolation is shown in Figure 8. As shown in Figure 8, the initial number of interpolation points for the machining path is 200. After interpolation optimization, the number of interpolation points is reduced to 128, indicating that CPBI can effectively improve the computational efficiency of

toolpath points. If only forward or backward scanning methods are used for interpolation optimization, the number of interpolation points will increase to 177. Meanwhile, the chord error after interpolation optimization is shown in Figure 9. Meanwhile, the chord error after interpolation optimization is shown in Figure 9. It can be seen that the chord error is controlled within 0.0002 mm, far below the specified upper limit for chord error ε_{max}. This indicates that the error when machining along this interpolation-optimized path is extremely minimal. Note that the velocity variation parameter η is introduced in Section 3.2. The selection of this value affects both the number of interpolation points and the chord error, as detailed in Table 2. As shown in Table 2, when $\eta = 0.1$, the number of interpolation points is 184, with a maximum chord error of 0.0001 mm; When $\eta = 0.2$, the number of interpolation points is 97, with a maximum chord error of 0.0006 mm; when $\eta = 0.15$, a balance between the number of interpolation points and the maximum chord error is achieved. At the same time, an excessively large η is more likely to cause abrupt velocity changes. To ensure the algorithm's optimality, η = 0.15 is selected for the simulations in this paper.

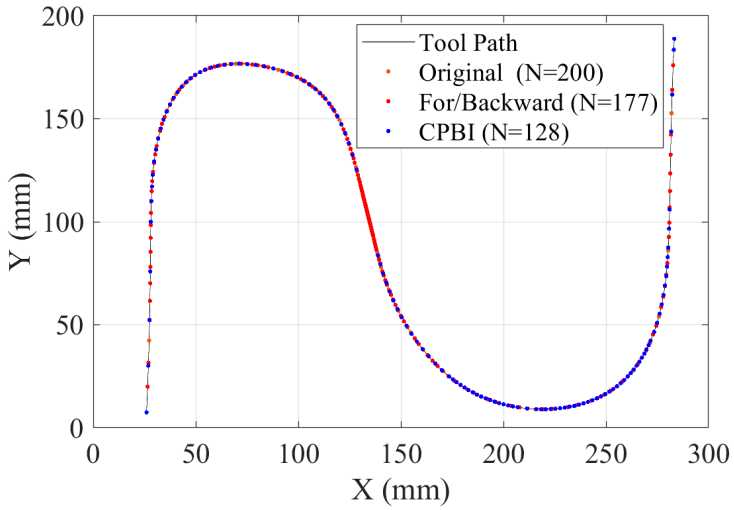


Figure 8. Position of machining points before and after interpolation optimization for the S-shaped path.

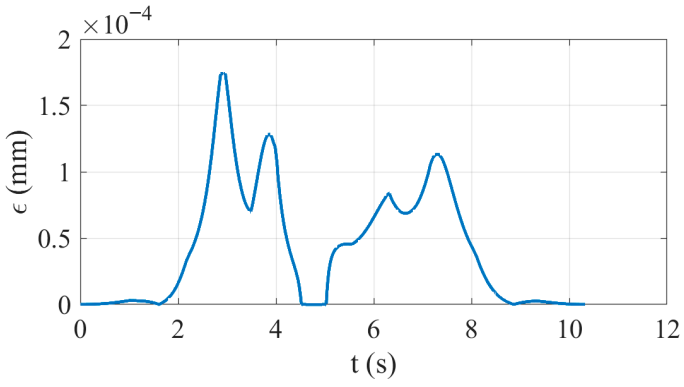


Figure 9. Chord error curve of S-shaped path.

Table 2. Sensitivity analysis of velocity variation parameters η..

η	Number of Interpolation Points	Maximum Chord Error
0.1	184	0.0001 mm
0.15	128	0.00015 mm
0.2	97	0.0006 mm

To validate the superiority of the proposed CPBI method, the iteration control vector parametrization method (ICVP) [26] was selected as a comparison. Simulations comparing the feed rates of both methods are shown in Figure 10. As shown in Figure 10, the ICVP maintains feed rates within the permissible range throughout the process. However, at locations with fast curvature changes, it adopts a constant-velocity machining strategy approaching the minimum feed rate. Therefore, this method can accomplish high-precision machining for S-shaped path but it inevitably increases machining time. In comparison, the feed rate curve obtained by CPBI not only avoids exceeding the limit but also better fits the permissible feed rate curve in regions with fast curvature changes. Furthermore, in the S-shaped machining path experiment, the CPBI algorithm achieved a 14.31% improvement in machining efficiency compared to the ICVP. Based on a comprehensive analysis of the results in Figures 9 and 10, it can be concluded that the proposed CPBI method can achieve high-precision and high-efficiency machining for an S-shaped path while satisfying multiple machining constraints.

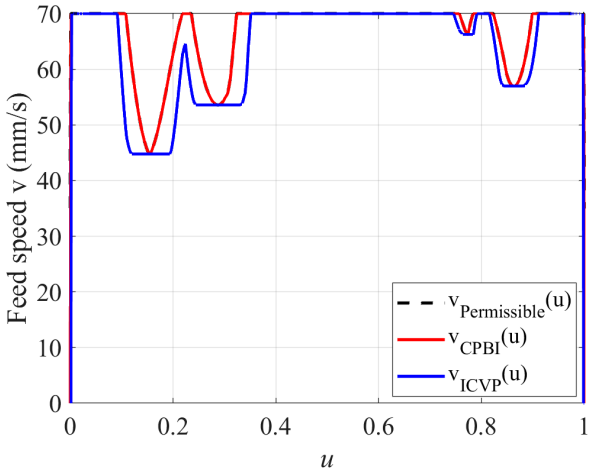


Figure 10. Comparison of feed rates before and after interpolation optimization for the S-shaped path.

Due to the constraints imposed by acceleration and jerk limits during actual machining processes, it is necessary to conduct further analysis of tangential acceleration, normal acceleration, and jerk accelerated acceleration. The results are shown in Figure 11. As evident from Figure 11, all three parameters remain within the maximum permissible ranges specified in Table 1. Then it can be further demonstrated that the CBPI method is effective for interpolation optimization of the S-shaped machining path.

4.2. Machining Simulation Experiment for Wave-Shaped Path

To further validate the effectiveness of CBPI for machining paths featuring multiple high-curvature segments and fast curvature changes, this paper additionally designed a wavy-shaped machining path as the experimental simulation object [5]. This machining path is achieved by constructing a non-uniform rational B-spline curve, where $k = 3$. There are 20 control points: d_0, d_1, d_2, …, d_{20}. The general weight is $w_i = 1$. High-curvature regions are constructed with larger weights: $w_4 = w_{11} = w_{15} = 2.0$ and $w_5 = w_{10} = w_{14} = 2.2$. The uniformly distributed node vectors are $U = \{0\,0\,0\,0\,u_4\,\ldots u_{19}\,1\,1\,1\,1\}$. Combining with Equation (2), the wave-shaped path can be expressed as follows:

$$P(u) = \frac{\sum_{i=0}^{19} d_i N_{i,3}(u)\omega_i}{\sum_{i=0}^{19} N_{i,3}(u)\omega_i} \tag{40}$$

Based on Equation (40), the designed path curve is shown in Figure 12. As seen in Figure 12, this machining path exhibits both high curvature and fast curvature variations. Upon further calculation, the curvature curve of the wave-shaped machining path is shown in Figure 13. It can be seen that there are over 10 locations with abrupt changes in curve curvature, which makes feed rate optimization more challenging. The program for generating machining paths can be found in the Supplementary Materials documents. Therefore, the proposed CBPI method is applied to perform interpolation optimization for this machining path, thereby demonstrating the high adaptability of the method.

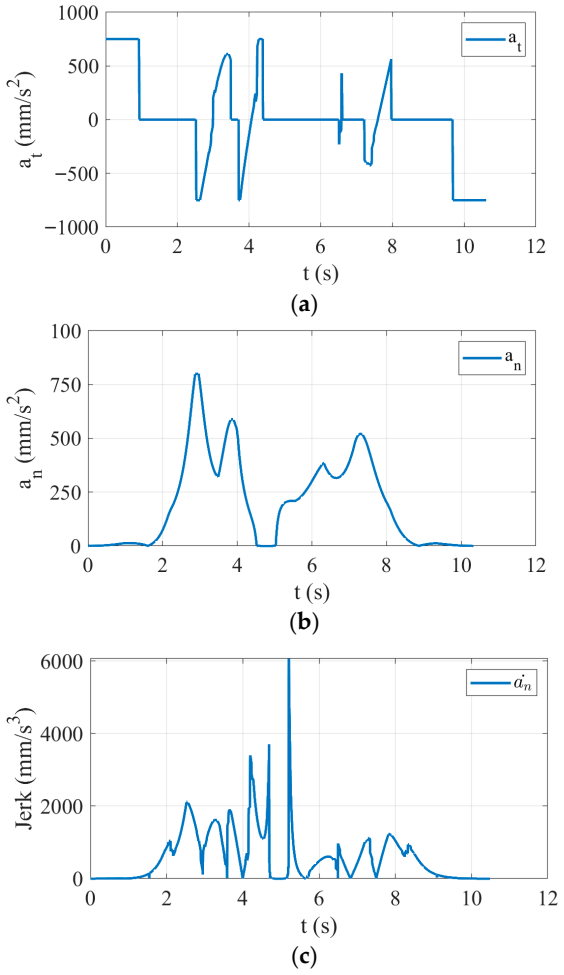


Figure 11. Relevant machining parameters for S-shaped path velocity planning (**a**) Tangential acceleration (**b**) Normal acceleration (**c**) Jerk accelerated acceleration.

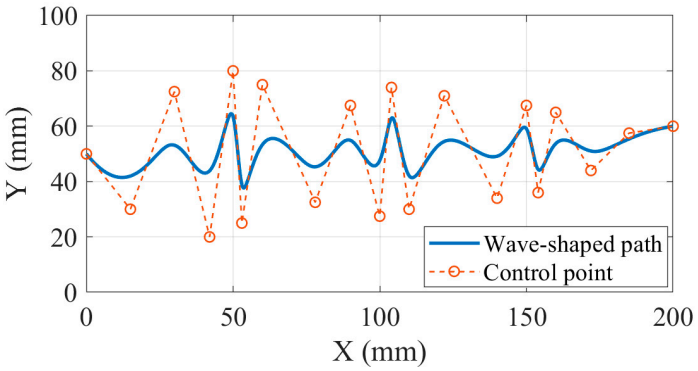


Figure 12. Wave-shaped machining path and control points.

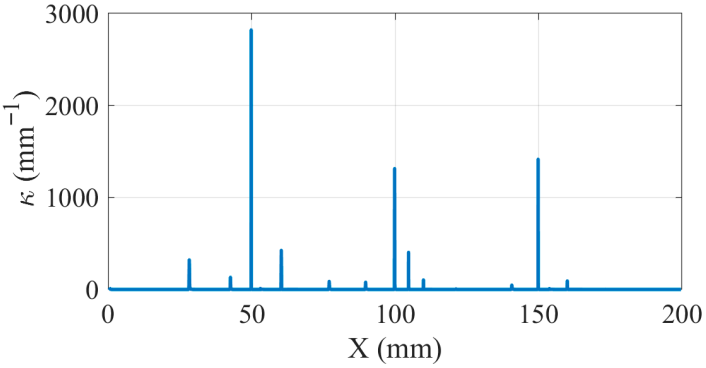


Figure 13. Curvature curve of the wave-shaped path.

Based on path characteristics and machining requirements, the simulation parameters for the wavy path are set, including the geometric parameters of the path, the interpolation cycle, and multiple kinematic constraints. Specific parameters are shown in Table 3.

Table 3. Experimental parameters for wave-shaped machining path.

Experimental Parameters	Value
T	0.002 s
ε_{max}	0.001 mm
$\overline{v}$	50 mm/s
a_{nmax}	500 mm/s^2
a_{tmax}	500 mm/s^2
j_{max}	10,000 mm/s^3

According to Table 3, the CPBI algorithm was employed to optimize the wave-shaped machining path. The interpolation optimization results are shown in Figure 14. The results indicate that the initial number of interpolation points for the wave-shaped machining path was 300, which was reduced to 151 after interpolation optimization. The reduction ratio approached 50%, significantly decreasing the number of machining instructions and enhancing machining efficiency. Similarly, if only forward or backward scanning methods are used for interpolation optimization, the number of interpolation points would increase to 238. Meanwhile, the chord error after interpolation optimization is shown in Figure 15. The optimization method effectively controls it within 0.00035 mm, far below the upper limit of chord error ε_{max}. Further comparisons were conducted using the CPBI method and ICVP method to evaluate the computational time, number of interpolation points, and maximum chord error for wave-shaped path optimization, as detailed in Table 4. As shown in Table 4, the CPBI method demonstrates improvements over the ICVP method in both computational time and the number of interpolation points, while exhibiting essentially identical maximum chord errors. This demonstrates that the CPBI method can reliably ensure the machining accuracy requirements for wave-shaped paths.

Similarly, comparing the feed rate optimized by the CPBI method with that of the ICVP, the results are shown in Figure 16. As shown, the optimized feed rates from both methods remain within the permissible feed rate. However, the feed rate curve optimized by the CPBI method is significantly closer to the permissible feed rate compared to the ICVP. In contrast, in the wave-shaped machining path experiment, the CBPI algorithm improves machining efficiency by 34.72% compared to the ICVP while satisfying multiple machining constraints. Combined analysis of Figures 15 and 16 further demonstrates that the CBPI algorithm is equally applicable to interpolation optimization for machining paths

characterized by high curvature and fast curvature changes. It can effectively achieve high-precision and high-efficiency machining of wave-shaped paths.

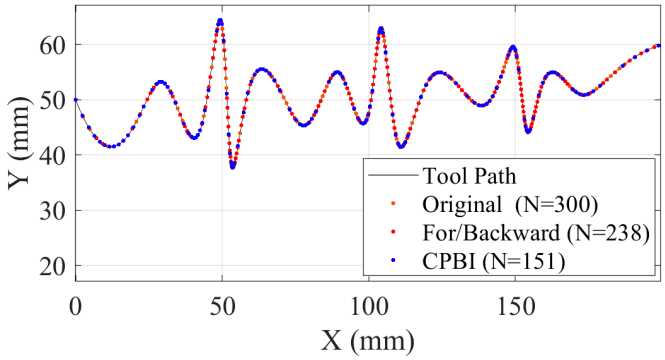


Figure 14. Position of machining points before and after interpolation optimization for wave-shaped path.

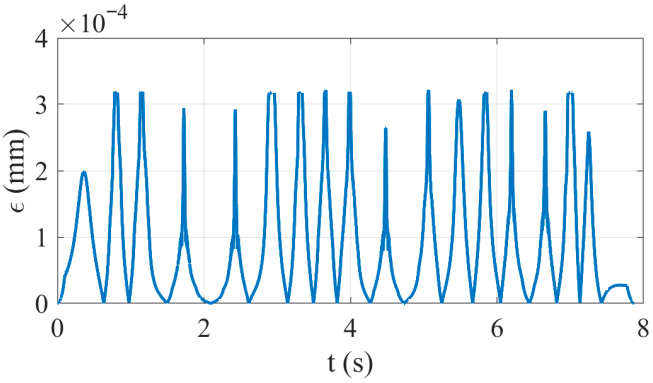


Figure 15. Chord error curve of wave-shaped path.

Table 4. Optimization comparison of CPBI and ICVP methods for wave-shaped paths.

	CPBI	ICVP
Computational time	3.47 s	4.28 s
Number of interpolation points	151	189
Maximum chord error	0.00035 mm	0.0003 mm

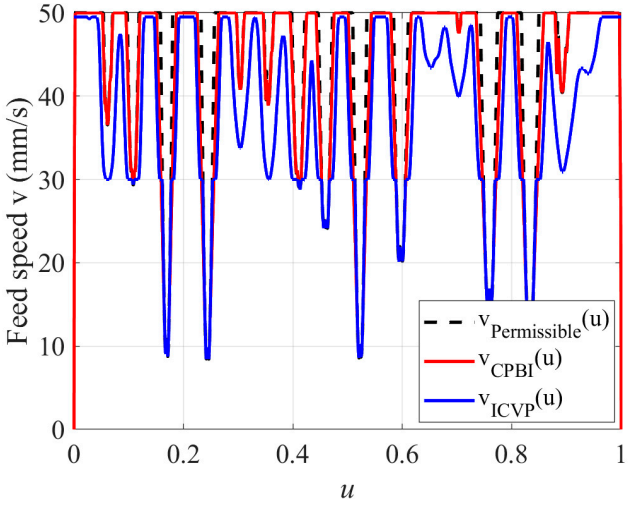


Figure 16. Comparison of feed rates before and after interpolation optimization for wave-shaped path.

After interpolation optimization of the wave-shaped path using the CBPI method, the resulting tangential acceleration, normal acceleration, and jerk accelerated acceleration are shown in Figure 17. As shown in the figure, all three parameters fall within the maximum permissible ranges specified in Table 3. In summary, the CBPI method proposed in this paper can generate optimized toolpaths with high precision under complex path machining conditions while significantly improving machining efficiency.

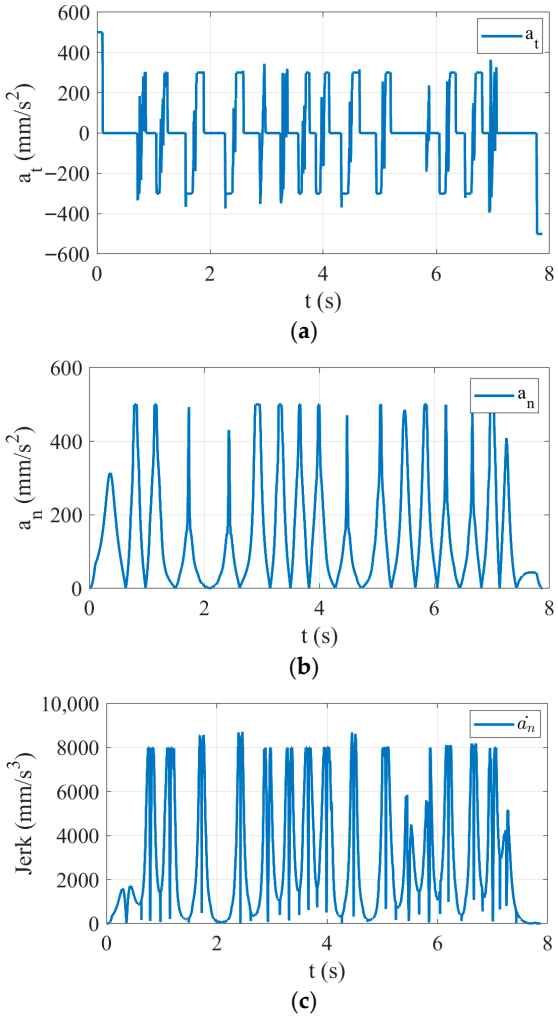


Figure 17. Relevant machining parameters for wave-shaped path velocity planning: (**a**) tangential acceleration; (**b**) normal acceleration; (**c**) jerk-accelerated acceleration.

5. Conclusions

This paper proposes a chord error-priority bilevel interpolation optimization method for complex path planning. Through global velocity optimization of chord error priority in the outer layer and interpolation optimization under comprehensive constraints in the inner layer, high-precision and high-efficiency machining is achieved for complex curvature paths. Experimental results demonstrate that the CPBI method effectively controls chord error while maintaining machining accuracy. It also significantly reduces the number of interpolation points and increases feed rate while ensuring tangential acceleration, normal acceleration, and jerk accelerated acceleration remain within permissible ranges. Compared to the ICVP, CPBI can improve machining efficiency by 14.31% and 34.72% on S-shaped and wave-shaped paths, respectively. This demonstrates its excellent adaptability and broad application prospects in the interpolation optimization of complex machining paths.

Future work will first demonstrate the effectiveness of the CPBI method in practical machining applications. By addressing potential issues encountered in machining experiments and multi-axis environments (e.g., the impact of tool wear or machine tool rigidity), further expand and improve the CPBI method. Concurrently, deep learning and AI technologies represent key directions for future development. It is essential to explore methods integrating interpolation optimization with these technologies. This will significantly promote its engineering applications in high-precision machining of complex parts.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/math13213385/s1, Document S1: the basic information for S-shaped machining path; Document S2: the basic information for wave-shaped machining path.

Author Contributions: Conceptualization, P.W., X.Y. and J.Q.; methodology, software, investigation, resources, visualization, writing—original draft preparation, P.W.; validation, L.W., X.Y. and J.Q.; data curation, X.Y. and D.W.; writing—review and editing, P.W. and J.Q.; project administration, L.W., X.Y. and D.W.; funding acquisition, L.W. and J.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Nos. 52375448, 52305480).

Data Availability Statement: The original contributions presented in this study are included in the Supplementary Material. Further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors confirm that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

CPBI	Chord error- priority bilevel interpolation optimization method
CNC	Computer numerical control
NURBS	Non-uniform rational B-spline
ICVP	Iteration control vector parametrization

Appendix A

Derivation Process of Chord Error Inequality (Equation (22))

Combining Equations (12)–(18) in the main paper, the following equations can be further derived within the Frenet–Serret framework:

$$T' = \kappa N \tag{A1}$$

$$N' = -\kappa T + \tau B \tag{A2}$$

$$B' = -\tau N \tag{A3}$$

The first, second, and third derivatives are as follows:

$$P' = T \tag{A4}$$

$$P'' = \kappa N \tag{A5}$$

$$P''' = \kappa' N + \kappa\tau B - \kappa^2 T \tag{A6}$$

Performing a third-order Taylor expansion of curve P at s_0, Equation (21) in the main paper is obtained. Then the chord vector can be expressed as follows:

$$P(\Delta s) = T\Delta s + \frac{1}{2}\kappa N\Delta s^2 + \frac{1}{6}(-\kappa^2 T + \kappa' N + \kappa\tau B)\Delta s^3 + O(\Delta s^4) \tag{A7}$$

The components of the chord vector in $\{T,\ N,\ B\}$ can be expressed as follows:

$$P_T(\Delta s) = \Delta s - \frac{1}{6}\kappa^2\Delta s^3 \tag{A8}$$

$$P_N(\Delta s) = \frac{1}{2}\kappa\Delta s^2 + \frac{1}{6}\kappa'\Delta s^3 \tag{A9}$$

$$P_B(\Delta s) = \frac{1}{6}\kappa\tau\Delta s^3 \tag{A10}$$

The third-order approximation of the chord length can be expressed as follows:

$$\begin{aligned} \|P(\Delta s)\| &= \sqrt{P_T^2(\Delta s) + P_N^2(\Delta s) + P_B^2(\Delta s)} \\ &= \Delta s\sqrt{1 - \tfrac{\kappa^2\Delta s^2}{12}} + O(\Delta s^3) \\ &= \Delta s\left(1 - \tfrac{\kappa^2\Delta s^2}{24}\right) + O(\Delta s^3) \end{aligned} \tag{A11}$$

Further, the unit chord vector can be obtained.

$$\begin{aligned} P_u &= \tfrac{P(\Delta s)}{\|P(\Delta s)\|} = \tfrac{P_T(\Delta s)}{\|P(\Delta s)\|}T + \tfrac{P_N(\Delta s)}{\|P(\Delta s)\|}N + \tfrac{P_B(\Delta s)}{\|P(\Delta s)\|}B \\ &= \left(1 - \tfrac{\kappa^2\Delta s^2}{8}\right)T + \left(\tfrac{\kappa\Delta s}{2} + \tfrac{\kappa'\Delta s^2}{6}\right)N + \tfrac{\kappa\tau\Delta s^2}{6}B + O(\Delta s^3) \end{aligned} \tag{A12}$$

Correspondingly, the components of vector P_u in $\{\ N,\ B\}$ can be expressed as follows:

$$(P_u)_N = \frac{\kappa\Delta s}{2} + \frac{\kappa'\Delta s^2}{6} \tag{A13}$$

$$(P_u)_B = \frac{\kappa\tau\Delta s^2}{6} \tag{A14}$$

The vector from the start point to the midpoint of the chord is represented as follows:

$$m = \frac{\Delta s}{2}T + \frac{\Delta s^2}{8}N + \frac{\Delta s^3}{48}(-\kappa^2 T + \kappa' N + \kappa\tau B) + O(\Delta s^4) \tag{A15}$$

Correspondingly, the components of vector m in $\{T,\ N,\ B\}$ can be expressed as follows:

$$m_T = \frac{\Delta s}{2} - \frac{\kappa^2\Delta s^3}{48} \tag{A16}$$

$$m_N = \frac{\kappa\Delta s^2}{8} + \frac{\kappa'\Delta s^3}{48} \tag{A17}$$

$$m_B = \frac{\kappa\tau\Delta s^3}{48} \tag{A18}$$

The chord error is defined as the difference between two adjacent machining points and the theoretical circular arc path. Then its expression can be represented as the vector m and the orthogonal distance to the line of the adjacent point.

$$\varepsilon = \|m - (m \cdot P_u)P_u\| \tag{A19}$$

where

$$\begin{aligned} m \cdot P_u &= \left(\tfrac{\Delta s^2}{2} - \tfrac{\kappa^2 \Delta s^3}{48}\right)\left(1 - \tfrac{\kappa^2 \Delta s^2}{8}\right) + \left(\tfrac{\kappa \Delta s^2}{8} + \tfrac{\kappa' \Delta s^3}{48}\right)\left(\tfrac{\kappa \Delta s}{2}\right) + O(\Delta s^4) \\ &= \tfrac{\Delta s}{2} + O(\Delta s^4) \end{aligned} \tag{A20}$$

Then the chord error vector can be expressed as follows:

$$\begin{aligned} \vec{\varepsilon} &= m - (m \cdot P_u) P_u \\ &= (m_N - (m \cdot P_u)(P_u)_N) N + (m_B - (m \cdot P_u)(P_u)_B) B + O(\Delta s^4) \end{aligned} \tag{A21}$$

Substituting the Equations (A13), (A14) and (A20) into Equation (A21) yields the following:

$$\begin{aligned} m_N - (m \cdot P_u)(P_u)_N &= \left(\tfrac{\kappa \Delta s^2}{8} + \tfrac{\kappa' \Delta s^3}{48}\right) - \tfrac{\Delta s}{2}\left(\tfrac{\kappa \Delta s}{2} + \tfrac{\kappa' \Delta s^2}{6}\right) + O(\Delta s^4) \\ &= -\tfrac{\kappa \Delta s^2}{8} - \tfrac{\kappa' \Delta s^3}{24} + O(\Delta s^4) \end{aligned} \tag{A22}$$

$$\begin{aligned} m_B - (m \cdot P_u)(P_u)_B &= \tfrac{\kappa \tau \Delta s^3}{48} - \tfrac{\Delta s}{2}\left(\tfrac{\kappa \tau \Delta s^2}{6}\right) + O(\Delta s^4) \\ &= -\tfrac{\kappa \tau \Delta s^3}{16} + O(\Delta s^4) e \end{aligned} \tag{A23}$$

By combining the above various equations, the chord error vector for the third-order Taylor expansion can be obtained.

$$\vec{\varepsilon} = -\frac{\kappa \Delta s^2}{8} N - \frac{\kappa' \Delta s^3}{24} N - \frac{\kappa \tau \Delta s^3}{16} B + O(\Delta s^4) \tag{A24}$$

Then the modulus of the chord error is as follows:

$$\begin{aligned} \varepsilon &= \left\| \vec{\varepsilon} \right\| \\ &= \sqrt{\left(\tfrac{\kappa \Delta s^2}{8} N + \tfrac{\kappa' \Delta s^3}{24}\right)^2 + \left(\tfrac{\kappa \tau \Delta s^3}{16}\right)^2} + O(\Delta s^4) \\ &= \tfrac{\kappa \Delta s^2}{8} + \tfrac{(\kappa' + \kappa \tau) \Delta s^3}{24} + O(\Delta s^4) \end{aligned} \tag{A25}$$

From this, Equation (22) in the main paper can be derived.

Appendix B

Jerk is the derivative of normal acceleration with respect to time. It is expressed as follows:

$$\begin{aligned} \dot{a}_n &= \tfrac{d}{dt}\left(v^2 \kappa \vec{n}\right) \\ &= \tfrac{d}{t}\left(v^2 \kappa\right) + \left(v^2 \kappa\right) \tfrac{dn}{dt} \end{aligned} \tag{A26}$$

where $\vec{n}$, $\vec{b}$ and $\vec{t}$ are the normal vector, binormal vector, and tangent vector, respectively.

The two sub-terms in Equation (A26) are calculated as follows:

$$\begin{aligned} \tfrac{d}{t}\left(v^2 \kappa\right) &= 2 v \dot{v} \kappa + v^2 \left(\tfrac{d\kappa}{ds} \tfrac{ds}{dt}\right) \\ &= 2 v \dot{v} \kappa + v^3 \kappa' \end{aligned} \tag{A27}$$

$$\begin{aligned} \tfrac{dn}{dt} &= \tfrac{dn}{ds} \tfrac{ds}{dt} \\ &= \left(-\kappa \vec{t} + \tau \vec{b}\right) v \\ &= -v \kappa \vec{t} + v \tau \vec{b} \end{aligned} \tag{A28}$$

where $\dot{v}$ denotes the first derivatives of feed rate with respect to arc length. κ' denotes the first derivative of curvature with respect to arc length.

Substituting Equations (A27) and (A28) into Equation (A26) yields the complete jerk equation.

$$\begin{aligned} \dot{a}_n &= \tfrac{d}{dt}\left(v^2\kappa\vec{n}\right) \\ &= \left(2v\dot{v}\kappa + v^3\kappa'\right)\vec{n} + v^2\kappa\left(-v\kappa\vec{t} + v\tau\vec{b}\right) \\ &= \left(2v\dot{v}\kappa + v^3\kappa'\right)\vec{n} + v^3\kappa\tau\vec{b} - v^3\kappa^2\vec{t} \end{aligned} \tag{A29}$$

As seen in Equation (27) in the main paper, the tangential component has been omitted. This is because, for ensuring surface quality in actual feed operations, the focus lies on variations in normal acceleration. Since the tangential component does not contribute to normal jerk, it has been omitted. Consequently, Equation (27) in the main paper is derived.

Appendix C

Table A1. Unified list of symbols.

Symbols	Meaning of the Symbol
$P(u)$	Parametric curve equation
d_i	Control point
$N_{i,k}(u)$	Basis function
$P_u(u)$	First derivative of the parameter curve with respect to u
$P_{uu}(u)$	Second derivative of the parameter curve with respect to u
$P_{uuu}(u)$	Third derivative of the parameter curve with respect to u
$s(u)$	Arc length curve equation
T	Tangent vector
N	Principal normal vector
B	Secondary normal vector
κ	Curvature
τ	Torsion
κ'	The first derivative of curvature with respect to arc length
s_i	The *i*-th discrete arc length parameter
v_i	The *i*-th step velocity
Δs	Single-step arc length
k_{max}	Maximum Iterations for Bidirectional Scanning
ε	Chord error
ε_{target}	The maximum target chord error
v	velocity
v_{max}	The upper limit of velocity
a_t	Tangential acceleration
a_{tmax}	The upper limits of tangential acceleration
a_n	Normal acceleration
a_{nmax}	The upper limits of normal acceleration
$\dot{a}_n$	Jerk acceleration
$\dot{v}$	The first derivatives of velocity with respect to arc length
v_{pmax}	the point-state velocity upper limit of the machining tools
j_{max}	The upper limit of jerk acceleration
v_ε	The upper limit of velocity dominated by the chord error
$\overline{v}$	The comprehensive upper limit for velocity
η	The step velocity variation parameter for adjacent tool positions
Δs_{dis}	Single-step arc length under distortion constraints
Δs_ε	Single-step arc length under chord error constraints
Δt	Interpolation period
$T(\varepsilon)$	Actual machining time
v_*	The optimal velocity solution
$\underline{v}$	The smallest positive number to prevent the denominator from being 0
L	The total arc length of machining path
P_T	Tangent component of the parameter curve
P_N	Principal normal component of the parameter curve
P_B	Secondary normal component of the parameter curve
m	The vector from the start point to the midpoint of the chord
m_T	Tangent component of m
m_N	Principal normal component of m
m_B	Secondary normal component of m
$\vec{\varepsilon}$	chord error vector

References

1. Hu, Y.; Jin, X.; Jiang, X.; Zheng, Z. Inverse kinematics model and trajectory generation of a dual-stage micro milling machine. *J. Manuf. Process.* **2024**, *132*, 425–450. [CrossRef]
2. Xu, Y.; Zaman, M.; Zhou, F. Optimized dual NURBS curve interpolation for high-accuracy five-axis CNC path planning. *Sci. Rep.* **2025**, *15*, 25409. [CrossRef] [PubMed]
3. Lu, Z.; Yang, X.; Zhao, J. Tool-path planning method for kinematics optimization of blade machining on five-axis machine tool. *Int. J. Adv. Manuf. Technol.* **2022**, *121*, 1253–1267. [CrossRef]
4. Liao, Z.; Li, J.; Xie, H.; Wang, Q.; Zhou, X. Region-based toolpath generation for robotic milling of freeform surfaces with stiffness optimization. *Robot. Comput. Integr. Manuf.* **2020**, *64*, 101953. [CrossRef]
5. Wang, L.; Zhang, J.; Li, W.; Wang, Y.; Zhou, Y. Research on feedrate scheduling method for NURBS toolpath interpolation in high-speed CNC machining. *Int. J. Adv. Manuf. Technol.* **2025**, *138*, 3293–3313. [CrossRef]
6. Zhong, W.; Luo, X.; Chang, W.; Cai, Y.; Ding, F.; Liu, H.; Sun, Y. Toolpath Interpolation and Smoothing for Computer Numerical Control Machining of Freeform Surfaces: A Review. *Int. J. Autom. Comput.* **2019**, *17*, 1–16. [CrossRef]
7. Ma, H.; Shen, L.; Jiang, X.; Zou, Q.; Yuan, C. A survey of path planning and feedrate interpolation in computer numerical control. *J. Graph.* **2022**, *43*, 967–986. [CrossRef]
8. Zhou, Y.; Jiang, Y.; Lu, C.; Huang, J.; Pei, J. A review of 5-axis milling techniques for centrifugal impellers: Tool-path generation and deformation control. *J. Manuf. Process.* **2024**, *132*, 160–186. [CrossRef]
9. Zhou, X. An Adapted NURBS Interpolator with a Switched Optimized Method of Feed-Rate Scheduling. *Machines* **2024**, *12*, 186. [CrossRef]
10. Zhang, G.; Gao, J.; Zhang, L.; Wang, X.; Luo, Y. Generalised NURBS interpolator with nonlinear feedrate scheduling and interpolation error compensation. *Int. J. Mach. Tools Manuf.* **2022**, *183*, 103956. [CrossRef]
11. Jia, Z.; Song, D.; Ma, J.; Hu, G.; Su, W. A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. *Int. J. Mach. Tools Manuf.* **2017**, *116*, 1–17. [CrossRef]
12. Wang, L.; Liu, Q.; Sun, P.; Lv, S.; Yang, R.; Yang, Z. Dynamic look-ahead feedrate scheduling method based on sliding mode velocity control. *Sci. Rep.* **2024**, *14*, 15424. [CrossRef]
13. Song, D.; Ma, J.; Zhong, Y.; Yao, J. Global smoothing of short line segment toolpaths by control-point-assigning-based geometric smoothing and FIR filtering-based motion smoothing. *Mech. Syst. Signal Process.* **2021**, *160*, 107908. [CrossRef]
14. Sun, C.; Dominguez-Caballero, J.; Ward, R.; Ayvar-Soberanis, S.; Curtis, D. Machining cycle time prediction: Data-driven modelling of machine tool feedrate behavior with neural networks. *Robot. Comput. Integr. Manuf.* **2022**, *75*, 102293. [CrossRef]
15. Ward, R.; Sun, C.; Dominguez-Caballero, J.; Ojo, S.; Ayvar-Soberanis, S.; Curtis, D.; Ozturk, E. Machining Digital Twin using real-time model-based simulations and lookahead function for closed loop machining control. *Int. J. Adv. Manuf. Technol.* **2021**, *117*, 3615–3629. [CrossRef]
16. Sun, S.; Zhao, P.; Zhang, T.; Li, B.; Yu, D. Smoothing interpolation of five-axis tool path with less feedrate fluctuation and higher computation efficiency. *J. Manuf. Process.* **2024**, *109*, 669–693. [CrossRef]
17. Yang, J.; Yin, X.; Sun, Y. PSO-Based Feedrate Optimization Algorithm for Five-Axis Machining with Constraint of Contour Error. *Machines* **2023**, *11*, 501. [CrossRef]
18. Wang, Y.; Xie, Z.; Xie, F.; Liu, X. A symmetrical arrangement strategy based real-time toolpath planning method for a 5-DoF fully parallel machining robot. *Int. J. Adv. Manuf. Technol.* **2024**, *134*, 2299–2317. [CrossRef]
19. Sencer, B.; Ishizaki, K.; Shamoto, E. High speed cornering strategy with confined contour error and vibration suppression for CNC machine tools. *CIRP Ann. Manuf. Technol.* **2015**, *64*, 369–372. [CrossRef]
20. Sun, Y.; Zhao, Y.; Bao, Y.; Guo, D. A novel adaptive feedrate interpolation method for NURBS tool path with drive constraints. *Int. J. Mach. Tools Manuf.* **2014**, *77*, 74–81. [CrossRef]
21. Sang, Y.; Yao, C.; Lv, Y.; He, G. An improved feedrate scheduling method for NURBS interpolation in five-axis machining. *Precis. Eng.* **2020**, *64*, 70–90. [CrossRef]
22. Liu, X.; Yu, P.; Chen, H.; Peng, B.; Wang, Z.; Liang, F. Feedrate fluctuation minimization for NURBS tool path interpolation based on arc length compensation and iteration. *Micromachines* **2025**, *16*, 402. [CrossRef]
23. Zhao, K.; Kang, Z.; Guo, X. Smooth Trajectory Generation for Predefined Path with Pseudo Spectral Method. *IEEE Access* **2020**, *8*, 158735–158744. [CrossRef]
24. Wang, L.; Yang, R.; Lv, S.; Yang, Z.; Xin, S.; Liu, Y.; Liu, Q. Combined contour error control method for five-axis machine tools based on digital twin. *Sci. Rep.* **2025**, *15*, 17809. [CrossRef] [PubMed]

25. Wang, J.; Song, C.; Zhang, D.; Li, D.; Qu, S. Fast and Stable Iterative Algorithm for Searching Wheel-Rail Contact Point Based on Geometry Constraint Equations. *J. Comput. Nonlinear Dyn.* **2023**, *18*, 011004. [CrossRef]
26. Ma, H.; Yuan, C.; Shen, L.; Gao, X. Optimal feedrate planning on a five-axis parametric tool path with global geometric and kinematic constraints. *J. Comput. Des. Eng.* **2022**, *9*, 2355–2374. [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.