DOI 10.2478/ama-2025-0088

A CATMULL-ROM SPLINE BASED ANALYTICAL C3 CONTINUOUS TOOL PATH SMOOTHING METHOD FOR ROBOTIC MACHINING

Xu-Lin CAI[*/**], **Wen-An YANG**[**], **You-Peng YOU**[**]

[*]Luoyang Institute of Electro-Optical Equipment AVIC, LuoYang 471000, People's Republic of China
[**]College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, People's Republic of China

caixulin@nuaa.edu.cn, dreamflow@nuaa.edu.cn, youypeng@163.com

received 13 August 2025, revised 26 November 2025, accepted 10 December 2025

Abstract: Tool path smoothness is critical for ensuring the dynamic performance of robotic machining systems, as it directly influences machining efficiency and quality. In recent years, spline-based methods (e.g., Bézier-spline, B-spline, NURB-spline, and PH-spline) have been widely employed to achieve tool path smoothing. However, most existing studies have focused on achieving only G2 or C2 continuity of tool paths, leading to discontinuous jerk behavior and resulting in high-order resonance frequencies within the machining system. Although some attention has been given to the need for C3 continuity in tool paths, synchronization between tool tip position and orientation remains suboptimal due to the complex, high-dimensional nonlinear kinematics of robotic machining systems. An analytical C3 continuous tool path smoothing method based on Catmull-Rom splines is developed in this study for robotic machining systems. The method smooths corners between adjacent discrete linear segments by inserting an adjustable Catmull-Rom (ACR) spline, with control points and adjustment parameters specifically designed to minimize deviation errors between the smoothed and original tool paths. Subsequently, the tool tip position and orientation are synchronized with the tool tip displacement, maintaining C3 continuity, by replacing the remaining linear segments with ACR splines. These splines' control points can be directly selected without requiring any iterative calculations, and synchronization error is guaranteed to be zero. The developed method involves a fully analytical calculation process, eliminating the need for iterative methods. Numerical simulations demonstrate that the tool paths generated by the developed method satisfy preset tolerances with smooth, continuous jerks in both workpiece and joint coordinate spaces, and that synchronization errors are indeed zero.

Key words: Catmull-Rom spline, tool path smoothing, robotic machining, C3 continuity

1. INTRODUCTION

With advancements in robot technology, both the stiffness and accuracy of industrial robots have significantly improved. As a result, industrial robots are increasingly applied in the field of metal cutting [1, 2], particularly in the flexible machining of large aerospace components, due to their advantages of low cost, high efficiency, flexibility, and expansive workspace. Traditionally, industrial robots have been used for tasks such as transporting objects from one location to another, where only the final positioning accuracy was of concern. However, for contour motion tasks, such as tool path planning, it is essential to ensure both deviation error constraints and the continuity of the smoothed tool path throughout the entire trajectory. Consequently, tool path planning in these applications is far more complex than traditional point-to-point tasks, leading to growing research interest in recent years [3-5].

Discrete linear motion commands generated by Computer-Aided Manufacturing (CAM) software are commonly employed in Computer Numerical Control (CNC) machining [6]. However, these commands lead to discontinuities in feedrate and theoretically infinite acceleration at the corners due to tangential discontinuities. To ensure smooth and continuous feed drive motions, it is necessary to smooth these discrete linear motion commands to achieve high-order geometric continuity before they are sent to the robot drives for machining. Tool path planning methods can be categorized based on the reference frame in which the tool path command positions are represented. These methods are generally classified into joint space and task space methods [7, 8]. In the joint space method, tool paths are defined within the joint space, and motion commands are smoothed directly for individual robot joints [9, 10]. This method has the advantage of directly driving the robot joints using the planned path without the need for kinematic transformations. However, it does not allow for intuitive collision checking and faces challenges in constraining the positional deviation errors of the tool's orientation relative to the workpiece. To address these limitations, task space methods have been developed, wherein tool paths are defined in the workpiece coordinate system (WCS) as tool tip positions and orientations [11, 12]. Moreover, tool path smoothing methods can be further divided into global and local smoothing methods, depending on the spline construction method used. In the global smoothing method, the discrete linear commands are represented by an entire spline [12, 13], which facilitates smoothness and continuity along the entire tool path. However, this method faces difficulties in evaluating and constraining the deviation errors between the smoothed and original tool paths. In contrast, the local smoothing method introduces micro splines at the corners between adjacent discrete segments [14, 15], achieving smoothness along the tool path while addressing the drawbacks associated with global smoothing methods.

Several local smoothing methods have been developed for tool path smoothing in three, four, and five-axis machine tools to achieve various orders of motion continuity [16-18]. Among these, five-axis machine tools have garnered significant research interest due to their complex structures and kinematics. Jin et al. [19] developed a G2 continuous local smoothing method for five-axis tool paths using double G2 continuous cubic Bézier splines. Yan et al. [20] introduced a G2 continuous local smoothing method for five-axis tool paths based on a pair of double cubic NURBS curves. Huang et al. [21] developed a real-time G2 continuous local smoothing method for five-axis tool paths by replacing corners of the tool position and tool orientation paths with cubic B-splines. Zhao et al. [22] developed an analytical C2 continuous local smoothing method for five-axis tool paths using an asymmetric cubic B-spline and a pair of symmetric quartic spherical Bézier splines. Huang et al. [23] developed a C2 continuous local smoothing method for five-axis tool paths using high-order continuous arithmetic with peak-constrained jerk. The transition splines developed in these studies [19-23] are either G2 or C2 continuous, ensuring continuous acceleration but discontinuous jerk. As demonstrated by Yuen et al. [24], discontinuous jerk can excite high-order resonance frequencies in the mechanical system, resulting in increased vibrations and tracking errors. In response to this, Tulsyan et al. [25] developed an analytical C3 continuous local smoothing method for five-axis tool paths by introducing quintic and septic micro-splines to smooth the tool tip and tool orientation paths, respectively. Yang et al. [26] also developed an analytical C3 continuous local smoothing method for five-axis tool paths using a specially designed quintic micro-spline. Hu et al. [27] presented a real-time C3 continuous local smoothing method for five-axis tool paths based on C3 continuous PH splines. Since the tool orientations of five-axis machines are influenced by only two rotary axes, the deviation error constraint and synchronization problem can be more easily solved analytically. In contrast, the tool orientation of robotic machining systems is influenced by all six rotary joints simultaneously, resulting in high-dimensional nonlinear orientation kinematics between the WCS and the robot's coordinate system. This significantly complicates tool orientation smoothing and synchronization compared to five-axis machine tools. More recently, Yang et al. [28] developed an analytical C2 continuous an analytical tool path smoothing algorithm for 6-DOF robots using 5th degree PH-spline. In this method, the tool tip position and orientation are smoothed within the WCS and synchronized by replacing the remaining linear segments with specially constructed PH-splines.

In this study, an analytical C3 continuous tool path smoothing method based on Catmull-Rom (CR) splines is developed for robotic machining. First, the tool orientation, originally represented by a rotation matrix, is transformed into three rotary angles to facilitate smooth interpolation. Second, the tool tip position and orientation are smoothed separately using the developed adjustable Catmull-Rom (ACR) spline-based method, wherein deviation errors introduced by local corner smoothing are constrained through control points. Additionally, the deviation errors caused by ACR smoothing are optimized using a simple yet efficient method. Third, the tool tip position and orientation are synchronized with the tool tip displacement while maintaining C3 continuity by replacing the remaining linear segments with ACR splines. Notably, the control points for this process do not require additional computations, significantly reducing computational overhead. The remainder of this study is structured as follows: Section 2 presents the mathematical representation of tool tip position and orientation. Section 3 introduces the ACR spline-based corner smoothing method for tool tip position and orientation, including a brief overview of ACR splines and the deviation error optimization method. Section 4 details the synchronization method for tool tip position and orientation. Numerical simulation results verifying the developed method are provided in Section 5, followed by conclusions in Section 6.

2. REPRESENTATION OF TOOL TIP POSITION AND ORIENTATION

For the robotic machining system as shown in Fig. 1, the transfer matrix of tool frame related to workpiece frame can be represented by a homogeneous transformation matrix:

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{p} \\ 0 & 1 \end{bmatrix} \tag{1}$$

where $\boldsymbol{R}$ and $\boldsymbol{p}$ denote the rotation matrix and translation offset vector, respectively.

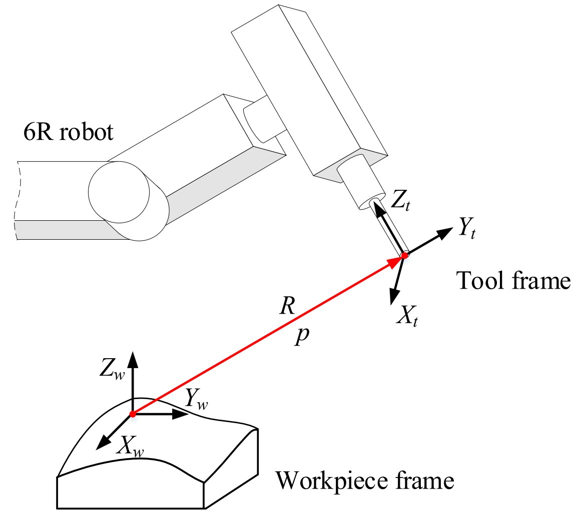
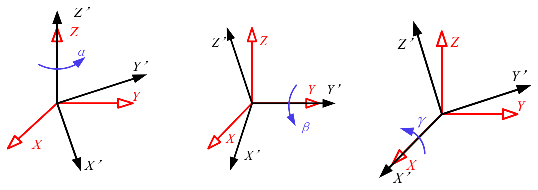


Fig. 1. Transformation of tool frame related to workpiece frame in robotic machining system

The tool tip position and orientation of industrial robot in the WCS can be determined by:

$$\boldsymbol{p} = [x, y, z]^T, \ \boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2}$$

The tool tip position of the industrial robot can be smoothed directly in the displacement space of $[x, y, z]^T$. However, to satisfy the mathematical constraint of the unit orthogonal transformation of tool orientation, the rotation matrix R should firstly be transformed as three rotary angles $[\alpha, \beta, \gamma]^T$ that rotated around the fixed global coordinate system as shown in Fig. 2, and then the tool orientation of the industrial robot can be smoothed in the rotation space of $[\alpha, \beta, \gamma]^T$.



a) Rotation around Z-axis

b) Rotation around Y-axis

c) Rotation around X-axis

Fig. 2. Rotation around fixed global coordinate system

$$\boldsymbol{R} = R_Z(\alpha)R_Y(\beta)R_X(\gamma) = \begin{bmatrix} C_\alpha C_\beta & -S_\alpha C_\gamma + C_\alpha S_\beta S_\gamma & S_\alpha S_\gamma + C_\alpha S_\beta C_\gamma \\ S_\alpha C_\beta & C_\alpha C_\gamma + S_\alpha S_\beta S_\gamma & -C_\alpha S_\gamma + S_\alpha S_\beta C_\gamma \\ -S_\beta & C_\beta S_\gamma & C_\beta C_\gamma \end{bmatrix} \tag{3}$$

where S_α , S_β , S_γ , C_α , C_β and C_γ denotes $\sin\alpha, \sin\beta, \sin\gamma, \cos\alpha, \cos\beta$ and $\cos\gamma$, respectively. It is not difficult to obtain the following relations by combining Eq. (2) and Eq. (3):

$$\begin{cases} \beta = \dfrac{\pi}{2} \\ \alpha = 0 \\ \gamma = \operatorname{atan2}(r_{12}, r_{22}) \end{cases} \quad if \quad \beta = \frac{\pi}{2}$$

$$\begin{cases} \beta = -\dfrac{\pi}{2} \\ \alpha = 0 \\ \gamma = \operatorname{atan2}(-r_{12}, r_{22}) \end{cases} \quad if \quad \beta = -\frac{\pi}{2}$$

$$\begin{cases} \beta = \operatorname{atan2}\left(-r_{31}, \sqrt{r_{11}^2 + r_{21}^2}\right) \\ \alpha = \operatorname{atan2}\left(\dfrac{r_{21}}{c_\beta}, \dfrac{r_{11}}{c_\beta}\right) \\ \gamma = \operatorname{atan2}\left(\dfrac{r_{32}}{c_\beta}, \dfrac{r_{33}}{c_\beta}\right) \end{cases} \quad if \quad \beta \neq \pm\frac{\pi}{2} \tag{4}$$

The flowchart of the developed ACR-spline-based tool path smoothing method for robotic machining system is shown in Fig. 3. In the tool path smoothing process, the maximum deviation errors of both tool tip position and orientation caused by local smoothing should be constrained. Besides, the tool tip position and orientation should be synchronized to the tool tip displacement with C3 continuity by replacing the remaining linear segments with ACR spline.

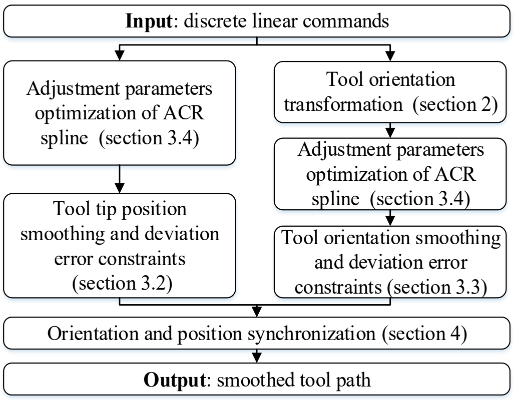


Fig. 3. Flowchart of the proposed tool path smoothing method

3. ANALYTICAL C3 CONTINUOUS CORNER SMOOTHING METHOD

In this section, a novel C3 continuous corner smoothing method based on ACR spline is developed. The ACR spline used in this method is derived from the quasi-CR spline function described in [30]. The developed ACR-spline-based tool path smoothing method not only achieves C3 continuity but also effectively controls deviation errors while preventing cusps and self-intersections through the optimization of two introduced adjustment parameters. Furthermore, the inserted splines and their control points are determined analytically, and the control points of the splines used to replace the remaining linear segments can be selected directly without requiring additional computations, thereby enhancing computational efficiency.

3.1. ACR spline

In this subsection, a mathematical operation of ACR spline is described to realize not only C3 continuity of spline, but also control the deviation error, avoid cusps and self-intersections.

For control points (x_i, y_i, z_i), $i = 0,1,\ldots,n$, the CR spline in $x \in [x_i, x_{i+1}]$ can be expressed as:

$$\begin{cases} R_{y,i}(x) = \sum_{j=0}^{3} b_j\left(\frac{x-x_i}{\Delta x}\right) y_{i+j-1} \\ R_{z,i}(x) = \sum_{j=0}^{3} b_j\left(\frac{x-x_i}{\Delta x}\right) z_{i+j-1} \end{cases} \quad i = 1,2,\ldots,n-2 \tag{5}$$

where $\Delta x = x_{i+1} - x_i$, $i = 0,1,\ldots,n-1$, $b_j(t)$ denote three basic functions related to the spline in this interval and will be detailed in following.

It should be noted that the use of the symbol *x* in Eq. (5) does not imply that the developed ACR spline construction relies on the Cartesian *x*-coordinate. Here, *x*, *y* and *z* simply denote the three components of a chosen orthogonal coordinate system, and the formulation in Eq. (5) uses *x* only as a representative example for clarity of exposition. In practice, the ACR spline is applied independently to each scalar component of any orthogonal coordinate representation, including Cartesian coordinates $[x, y, z]$ (Section 3.2) or rotational parameters such as Euler angles $[\alpha, \beta, \gamma]$ (Section 3.3). Therefore, the method is not restricted to the *x*-channel, and the case *x*=0 does not constitute a degenerate situation: as long as the remaining coordinates (e.g., *y* or *z*) are available, the spline can be constructed and evaluated in exactly the same manner. The same basis functions apply to every coordinate component, and the choice of which component is illustrated is merely for notational simplicity, not a structural dependency of the algorithm.

3.1.1. Basis functions of the ACR spline

To achieve C^3 continuity and adjustability of the constructed CR spline, the order of its basis functions is set to seven, with two parameters introduced to enhance flexibility. Additionally, these basis functions must satisfy key properties, including unitary, symmetry, and continuity. By formulating linear equations involving the coefficients of the basis functions, the required basis functions can be derived by solving these equations, yielding the following results:

$$\begin{cases} b_0(t)=\kappa l_0(t)+\lambda m_0(t) \\ b_1(t)=\kappa l_1(t)+\lambda m_1(t)+n_1(t) \\ b_2(t)=-\kappa l_0(t)+\lambda m_2(t)+\left(1-n_1(t)\right) \\ b_3(t)=-\kappa l_1(t)-\lambda\left(m_0(t)+m_1(t)+m_2(t)\right) \end{cases} \tag{6}$$

where $t\in[0,1]$, $\kappa,\lambda\in R$are introduced adjustment parameters and:

$$\begin{cases} l_0(t)=-t+20t^4-45t^5+36t^6-10t^7 \\ l_1(t)=15t^4-39t^5+34t^6-10t^7 \\ m_0(t)=t^2-10t^4+20t^5-15t^6+4t^7 \\ m_1(t)=-2t^2+25t^4-54t^5+43t^6-12t^7 \\ m_2(t)=t^2-20t^4+48t^5-41t^6+12t^7 \\ n_1(t)=1-35t^4+84t^5-70t^6+20t^7 \end{cases} \tag{7}$$

It is not difficult to obtain from Eq. (6): $\sum_{i=0}^{3} b_i(t)=1$and $b_i(1-t)=b_{3\text{-}i}(t)$, which verifies the unitary and symmetry of the constructed basis functions, respectively; besides, the following relationship can be obtained, which is of great importance to verify the interpolate property and continuity of the ACR spline:

$$\begin{cases} b_0(0)=0, b_1(0)=1, b_2(0)=0, b_3(0)=0 \\ b_0(1)=0, b_1(1)=0, b_2(1)=1, b_3(1)=0 \end{cases} \tag{8}$$

$$\begin{cases} b_0'(0)=-\kappa, b_1'(0)=1, b_2'(0)=\kappa, b_3'(0)=0 \\ b_0'(1)=0, b_1'(1)=-\kappa, b_2'(1)=0, b_3'(1)=\kappa \end{cases} \tag{9}$$

$$\begin{cases} b_0''(0)=2\lambda, b_1''(0)=-4\lambda, b_2''(0)=2\lambda, b_3''(0)=0 \\ b_0''(1)=0, b_1''(1)=2\lambda, b_2''(1)=-4\lambda, b_3''(1)=2\lambda \end{cases} \tag{10}$$

$$\begin{cases} b_0'''(0)=0, b_1'''(0)=0, b_2'''(0)=0, b_3'''(0)=0 \\ b_0'''(1)=0, b_1'''(1)=0, b_2'''(1)=0, b_3'''(1)=0 \end{cases} \tag{11}$$

3.1.2. Properties of ACR spline

Interpolate property: Let the original control-point sequence be (x_0,y_0,z_0) … (x_n,y_n,z_n) . If two auxiliary control points (x_{-1},y_{-1},z_{-1}) and $(x_{n+1},y_{n+1},z_{n+1})$ are appended to Eq. (5), then synthesizing Eq. (5) and Eq. (8) yields the interpolation property at the endpoints:

$$\begin{cases} R_{y,i}(x_i)=y_i \\ R_{y,i}(x_{i+1})=y_{i+1} \\ R_{z,i}(x_i)=z_i \\ R_{z,i}(x_{i+1})=z_{i+1} \qquad i=1,2,\ldots,n-1 \end{cases} \tag{12}$$

which means that ACR splines always interpolates these given control points.

Continuity: It can be derived by combining Eq. (5) with Eqs. (9)-(11):

$$\begin{cases} R_{y,i}'(x_{i+1})=\frac{\kappa}{\Delta x}(y_{i+3}-y_{i+1})=R_{y,i+1}'(x_{i+1}) \\ R_{z,i}'(x_{i+1})=\frac{\kappa}{\Delta x}(z_{i+3}-z_{i+1})=R_{z,i+1}'(x_{i+1}) \\ R_{y,i}''(x_{i+1})=\frac{2\lambda}{\Delta x^2}(y_{i+1}-2y_{i+2}+y_{i+3})=R_{y,i+1}''(x_{i+1}) \\ R_{z,i}''(x_{i+1})=\frac{2\lambda}{\Delta x^2}(z_{i+1}-2z_{i+2}+z_{i+3})=R_{z,i+1}''(x_{i+1}) \\ R_{y,i}'''(x_{i+1})=0=R_{y,i+1}'''(x_{i+1}) \\ R_{z,i}'''(x_{i+1})=0=R_{z,i+1}'''(x_{i+1}) \end{cases} \tag{13}$$

which means that ACR spline always satisfies C^3 continuity.

Adjustability: Since the basis function shown in Eq. (6) contains two parametersκandλ, the shape of the ACR spline can be adjusted by changing the values ofκandλwhile all related control points remain unchanged.

Local property: As can be seen from Eq. (5), each ACR spline function $R_i(x)$ is only affected by four control points $(x_{i+j},y_{i+j},z_{i+j})$, where$j=0,1,2,3$. Therefore, when the adjustment parameters$\kappa$and$\lambda$are selected, the change of the one control point will affect the shape of four segments ACR spline related to it.

Avoidance of cusps and self-intersections: The ACR construction employed in this work incorporates analytical design choices that intrinsically avoid cusps and self-intersections without requiring additional iterative procedures. Specifically, the control points used to anchor an inserted ACR spline at a corner are placed on the incident linear segments and arranged symmetrically with respect to the corner bisector; the explicit control-point construction is given in Eq. (14). This symmetric placement ensures geometric balance of the control polygon and prevents abrupt sign changes of the tangent direction at the corner, which is a principal cause of cusps.

In addition, the portion of each linear segment reserved for spline insertion is strictly bounded by the segment-span constraints stated in the paper: the inserted spline occupies no more than one third of a shared linear segment (see Eqs. (16)–(17)). These analytic length bounds limit excessive local bending on short segments and therefore reduce the geometric conditions that can lead to self-intersection.

Finally, the two adjustment parameters κ and λ that define the ACR basis functions control local convexity and curvature concentration. In the present method *κ* and *λ* are chosen by the optimization routine described in Section 3.4 so that the ACR shape minimizes the prescribed deviation metrics while remaining within prescribed parameter bounds; this constrained parameter selection further prevents curvature concentrations that could induce cusps or self-intersections. The combination of (i) analytic, symmetric control-point placement (Eq. (14)), (ii) explicit segment-span limits (Eqs. (16)–(17)), and (iii) constrained *κ*/*λ* adjustment via the optimization described in Section 3.4 is the basis upon which the manuscript guarantees the absence of cusps and self-intersections for the constructed ACR corner splines.

3.1.3. Advantages of Using the Catmull–Rom Spline for Corner Smoothing

The CR spline is adopted in this study mainly because it provides several properties that are particularly suitable for local corner smoothing in tool-path planning. In addition, the ACR formulation used here—obtained by introducing two shape parametersκandλinto the basis functions—retains the classic CR advantages while adding useful degrees of freedom for practical tool-path control.

- Interpolation of data points. The CR splines are interpolatory and the curve passes through its control points. This property ensures that discrete tool tip positions and orientations at commanded tool tip points and tool orientations are preserved and that smoothing does not shift the programmed trajectory. Compared with B-splines/NURBS, this avoids solving global interpolation systems and simplifies local corner treatment.
- Locality and analytically assignable control points. The CR splines have strictly local support: manipulation of a single control point affects only neighboring spline pieces. This locality enables analytical assignment and local adjustment of

control points at each corner without affecting distant portions of the path. The ACR retains this local property, so κ and λ can be tuned locally to alter curvature and convexity only in the intended corner region.

- Adjustable shape for cusp/self-intersection avoidance. The two parameters in ACR directly control higher-order basis coefficients and therefore the local curvature and inflection behavior of the inserted micro-spline. This extra tunability permits analytic enforcement of convexity constraints and reduction of local curvature peaks, which helps to avoid cusps and self-intersections without resorting to global re-fitting. In our algorithm these parameters are chosen (analytically or via the local optimization in Section 3.4) to minimize deviation error while satisfying curvature/clearance constraints.
- High-order smoothness with analytic control. Although a standard CR produces C^1 continuity by construction, the ACR basis (7th-order with parameterization) enables analytic control of derivatives up to third order at junctions when control points are assigned according to the scheme in Section 3.2–3.4. Achieving equivalent C^3 behavior with Bézier, Hermite or PH splines typically requires higher-degree polynomial pieces or constrained optimizations; ACR provides a compact analytic alternative that is straightforward to enforce locally.
- Parameterization flexibility and numerical stability. The ACR supports different parameterizations (e.g. chordal/ centripetal) which reduce oscillations and the risk of unwanted loops that may appear with uniform parameterization. The combination of centripetal parameterization and local (κ,λ) tuning enhances numerical robustness in corner regions.
- Computational efficiency and practical deployment. The ACR admits explicit analytic expressions for positions and derivatives and—importantly—permits closed-form selection rules for many control points. This reduces computational burden compared to optimization-heavy methods (e.g. global NURBS fitting or constrained quintic blends), making the method more suitable for near-real-time tool-path preprocessing on CNC/robotic controllers. For segments where computation cost must be minimized, the adjustment parameters can be set to nominal values and only problematic corners are tuned.
- Compatibility with arc-length reparameterization and feedrate/jerk control. Because analytic derivatives are available, ACR integrates readily with arc-length reparameterization and feedrate scheduling procedures to enforce curvature and jerk limits—facilitating construction of C^3 continuous trajectories that are synchronized in both position and orientation.

In summary, the ACR spline combines the interpolatory accuracy and local control of classical CR splines with additional, analytically tunable shape parameters that improve robustness (anti-self-intersection), enable local curvature shaping, and retain computational simplicity—features that together make ACR particularly well suited for corner smoothing in robotic machining.

3.2. Tool tip position corner smoothing method

In this subsection, the tool-tip position corner $\angle P_{i-1}P_iP_{i+1}$ shown in Fig. 4 is chosen as an example to illustrate the developed tool tip smoothing method, where P_{i-1}, P_i and P_{i+1} are three end points of the adjacent linear segments. The ACR spline described in Section 3.1 is utilized to smooth the tool-tip position at this transition corner under the positional error tolerance ε_p. Here ε_p is defined in the Cartesian coordinate system and has units of length.

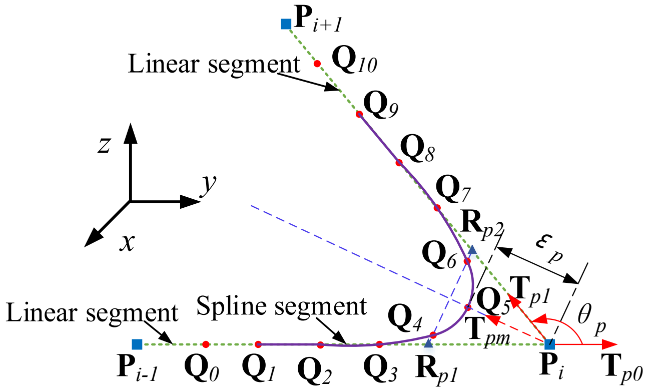


Fig. 4. Corner smoothing of tool tip position

In order to ensure the tangential continuity of the constructed ACR spline and the original line segment, considering the continuity and local property of the constructed CR spline, the control points $\boldsymbol{Q}_0-\boldsymbol{Q}_3$ and $\boldsymbol{Q}_7-\boldsymbol{Q}_{10}$ are located on the linear segments $\overrightarrow{\boldsymbol{P}_{i-1}\boldsymbol{P}_i}$ and $\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}$, respectively.

Besides, to realize the geometric symmetry of the inserted ACR splines with respect to the bisector of the angle formed by adjacent position segments, and to avoid cusps and self-intersections at the same time, the following control points are constructed:

$$\begin{cases} \boldsymbol{Q}_5=\boldsymbol{P}_i+\boldsymbol{T}_{pm}L_p \\ \boldsymbol{Q}_3=\boldsymbol{P}_i-\dfrac{2\boldsymbol{T}_{p0}L_p}{sin\left(\frac{\theta_p}{2}\right)} \\ \boldsymbol{Q}_7=\boldsymbol{P}_i+\dfrac{2\boldsymbol{T}_{p1}L_p}{sin\left(\frac{\theta_p}{2}\right)} \\ \boldsymbol{Q}_2=1.5\boldsymbol{Q}_3-0.5\mathbf{P}_i \\ \boldsymbol{Q}_8=1.5\boldsymbol{Q}_7-0.5\mathbf{P}_i \\ \boldsymbol{Q}_1=2\boldsymbol{Q}_3-\boldsymbol{P}_i \\ \boldsymbol{Q}_9=2\boldsymbol{Q}_7-\boldsymbol{P}_i \\ \boldsymbol{Q}_0=2.5\boldsymbol{Q}_3-1.5\boldsymbol{P}_i \\ \boldsymbol{Q}_{10}=2.5\boldsymbol{Q}_7-1.5\boldsymbol{P}_i \\ \boldsymbol{Q}_4=0.9\mathbf{R}_{p1}+0.1\mathbf{R}_{p2} \\ \boldsymbol{Q}_6=0.1\mathbf{R}_{p1}+0.9\mathbf{R}_{p2} \end{cases} \tag{14}$$

where

$$\begin{cases} \boldsymbol{T}_{p0} = \frac{\overrightarrow{\boldsymbol{P}_{i-1}\boldsymbol{P}_i}}{\|\overrightarrow{\boldsymbol{P}_{i-1}\boldsymbol{P}_i}\|} \\ \boldsymbol{T}_{p1} = \frac{\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}}{\|\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}\|} \\ \boldsymbol{T}_{pm} = \frac{(\boldsymbol{T}_{p1}-\boldsymbol{T}_{p0})}{\|\boldsymbol{T}_{p1}-\boldsymbol{T}_{p0}\|} \\ \boldsymbol{R}_{p1} = \frac{\boldsymbol{Q}_3}{3} + 2\frac{\left(\boldsymbol{P}_i - \frac{\boldsymbol{T}_{p0}L_p}{sin\left(\frac{\theta_p}{2}\right)}\right)}{3} \\ \boldsymbol{R}_{p2} = \frac{\boldsymbol{Q}_7}{3} + 2\frac{\left(\boldsymbol{P}_i + \frac{\boldsymbol{T}_{p1}L_p}{sin\left(\frac{\theta_p}{2}\right)}\right)}{3} \\ L_p \le \varepsilon_p \end{cases} \tag{15}$$

where $\theta_p = \arccos(\boldsymbol{T}_{p0}\boldsymbol{T}_{p1})$.

Since each linear segment is shared by two corners (except of the first and the last linear segments), therefore, one third of the segment is retained to adjust the synchronization of the tool tip position and tool orientation, that is, the linear segment used to construct ACR spline should be less than one third of the total length of the segment:

$$\begin{cases} \|\overrightarrow{\boldsymbol{Q}_0\boldsymbol{P}_i}\| \le \frac{\|\overrightarrow{\boldsymbol{P}_{i-1}\boldsymbol{P}_i}\|}{3} \\ \|\overrightarrow{\boldsymbol{P}_i\boldsymbol{Q}_{10}}\| \le \frac{\|\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}\|}{3} \end{cases} \tag{16}$$

Substituting Eq. (14) into Eq. (16) and combining with the error tolerance constraint of $L_p \le \varepsilon_p$, the final constraints of L_p can be obtained as:

$$L_p = min\left(\varepsilon_p, \frac{\|\overrightarrow{\boldsymbol{P}_{i-1}\boldsymbol{P}_i}\| \, sin\left(\frac{\theta_p}{2}\right)}{12\|\boldsymbol{T}_{p0}\|}, \frac{\|\overrightarrow{\boldsymbol{P}_i\boldsymbol{P}_{i+1}}\| \, sin\left(\frac{\theta_p}{2}\right)}{12\|\boldsymbol{T}_{p1}\|}\right) \tag{17}$$

3.3. Tool orientation corner smoothing method

In this subsection, the tool orientation corner $\angle\Psi_{i-1}\Psi_i\Psi_{i+1}$ shown in Fig. 5 is chosen as an example to illustrate the developed tool orientation smoothing method, where Ψ_{i-1}, Ψ_i and Ψ_{i+1} are three end point orientations of the adjacent linear segments (expressed in the chosen Euler-angle convention). The ACR spline described in Section 3.1 is utilized to smooth the tool orientation at this transition corner under the orientation error tolerance ε_o. Here ε_o is an angular tolerance defined on the Euler-angle coordinates and has units of radians.

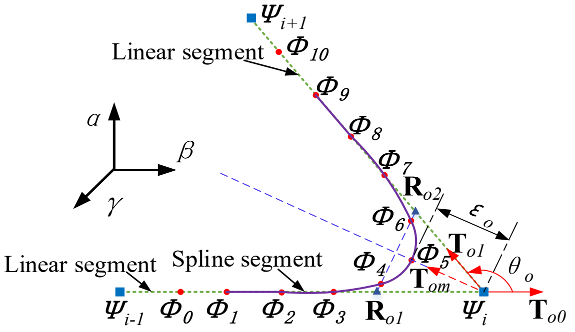


Fig. 5. Corner smoothing of tool orientation

Similar to the method of determining control points of tool tip spline described in last subsection, the control points of the tool orientation ACR spline can be determined by following equations:

$$\begin{cases} \boldsymbol{\Phi}_5 = \boldsymbol{\Psi}_i + \boldsymbol{T}_{om}L_o \\ \boldsymbol{\Phi}_3 = \boldsymbol{\Psi}_i - \frac{2T_{o0}L_o}{sin\left(\frac{\theta_o}{2}\right)} \\ \boldsymbol{\Phi}_7 = \boldsymbol{\Psi}_i + \frac{2T_{o1}L_o}{sin\left(\frac{\theta_o}{2}\right)} \\ \boldsymbol{\Phi}_2 = 1.5\boldsymbol{\Phi}_3 - 0.5\boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_8 = 1.5\boldsymbol{\Phi}_7 - 0.5\boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_1 = 2\boldsymbol{\Phi}_3 - \boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_9 = 2\boldsymbol{\Phi}_7 - \boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_0 = 2.5\boldsymbol{\Phi}_3 - 1.5\boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_{10} = 2.5\boldsymbol{\Phi}_7 - 1.5\boldsymbol{\Psi}_i \\ \boldsymbol{\Phi}_4 = 0.9\mathbf{R}_{o1} + 0.1\mathbf{R}_{o2} \\ \boldsymbol{\Phi}_6 = 0.1\mathbf{R}_{o1} + 0.9\mathbf{R}_{o2} \end{cases} \tag{18}$$

where

$$\begin{cases} \boldsymbol{T}_{o0} = \frac{\overrightarrow{\boldsymbol{\Psi}_{i-1}\boldsymbol{\Psi}_i}}{\|\overrightarrow{\boldsymbol{\Psi}_{i-1}\boldsymbol{\Psi}_i}\|} \\ \boldsymbol{T}_{o1} = \frac{\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Psi}_{i+1}}}{\|\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Psi}_{i+1}}\|} \\ \boldsymbol{T}_{om} = \frac{(\boldsymbol{T}_{o1}-\boldsymbol{T}_{o0})}{\|\boldsymbol{T}_{o1}-\boldsymbol{T}_{o0}\|} \\ \boldsymbol{R}_{o1} = \frac{\boldsymbol{\Phi}_3}{3} + 2\frac{\left(\boldsymbol{\Psi}_i - \frac{\boldsymbol{T}_{o0}L_o}{sin\left(\frac{\theta_o}{2}\right)}\right)}{3} \\ \boldsymbol{R}_{o2} = \frac{\boldsymbol{\Phi}_7}{3} + 2\frac{\left(\boldsymbol{\Psi}_i + \frac{\boldsymbol{T}_{o1}L_o}{sin\left(\frac{\theta_o}{2}\right)}\right)}{3} \\ L_o \le \varepsilon_o \end{cases} \tag{19}$$

where $\theta_o = \arccos(T_{o0}T_{o1})$.

Besides, one third of the segment should be retained to adjust the synchronization of the tool tip position and tool orientation:

$$\begin{cases} \|\overrightarrow{\boldsymbol{\Phi}_0\boldsymbol{\Psi}_i}\| \le \frac{\|\overrightarrow{\boldsymbol{\Psi}_{i-1}\boldsymbol{\Psi}_i}\|}{3} \\ \|\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Phi}_{10}}\| \le \frac{\|\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Psi}_{i+1}}\|}{3} \end{cases} \tag{20}$$

Substituting Eq. (18) into Eq. (20) and combining with the error tolerance constraint of, the constraints of L_o can be obtained as:

$$L_o = min\left(\varepsilon_o, \frac{\|\overrightarrow{\boldsymbol{\Psi}_{i-1}\boldsymbol{\Psi}_i}\| \, sin\left(\frac{\theta_o}{2}\right)}{12\|\boldsymbol{T}_{o0}\|}, \frac{\|\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Psi}_{i+1}}\| \, sin\left(\frac{\theta_o}{2}\right)}{12\|\boldsymbol{T}_{o1}\|}\right) \tag{21}$$

It must be noted that ε_o in Eq. (21) denotes an angular tolerance in the Euler-angle parameterization (units: radians), whereas positional tolerance in ε_p Section 3.2 is defined in the Cartesian coordinate (units: length) system. In practical application, the tool orientation error should be constrained in WCS, as shown in Fig. 6, while the error tolerance parameter ε_o utilized in Eq. (21) is constrained in the Cartesian coordinate system. Therefore, the relationship between ε_o and the tool orientation error ε_{ow} in the WCS should be established.

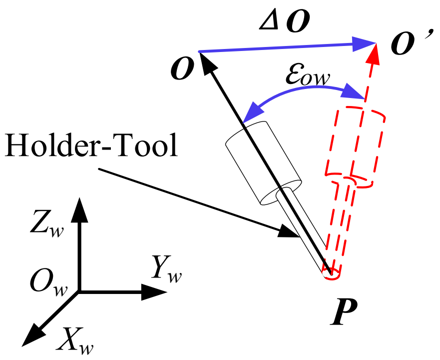


Fig. 6. Tool orientation error in the WCS

The tool orientation $\boldsymbol{O}_w = [O_i, O_j, O_k]$in the WCS can be expressed as:

$$\boldsymbol{O}_w = \boldsymbol{R} \cdot \boldsymbol{r}_{ot}$$
(22)

where the tool orientation matrix R has been obtained in Eq. (3), r_{ot}denotes the tool orientation vector related to the tool coordinate system and assumed that the tool center line is coincided with the z direction of the tool coordinate system, i.e. $r_{ot} = [0,0,1]^T$. Hence, the following relationship can be obtained:

$$\begin{cases} O_i = S_\alpha S_\gamma + C_\alpha S_\beta C_\gamma \\ O_j = -C_\alpha S_\gamma + S_\alpha S_\beta C_\gamma \\ O_k = C_\beta C_\gamma \end{cases} \tag{23}$$

The deviation of the tool orientation vector induced by the local smoothing in the WCS, as shown in Fig. 6, should satisfy the following relationship:

$$\Delta\boldsymbol{O} \le 2\,sin\left(\frac{\varepsilon_{ow}}{2}\right) \tag{24}$$

The approximate relationship between the deviation of tool orientation vector and rotary angles can be expressed as:

$$\Delta\boldsymbol{O} \approx \boldsymbol{J}_o \boldsymbol{\varepsilon}_o \tag{25}$$

which represents the first-order Taylor linearization of the orientation mapping, where higher-order terms have been omitted. More precisely, one may write $\Delta O = J_o \varepsilon_o + r(\boldsymbol{\varepsilon}_o)$ with $\|r(\boldsymbol{\varepsilon}_o)\| = \mathcal{O}(\|\boldsymbol{\varepsilon}_o\|^2)$. The relations that follow should therefore be interpreted as approximate bounds under the small-angle/small-deviation assumption. $\boldsymbol{J}_o$is the Jacobian matrix of the tool orientation and can be expressed as the following form by combining Eq. (23) and Eq. (25):

$$\boldsymbol{J}_o = \begin{bmatrix} \frac{\partial O_i}{\partial \alpha} & \frac{\partial O_i}{\partial \beta} & \frac{\partial O_i}{\partial \gamma} \\ \frac{\partial O_j}{\partial \alpha} & \frac{\partial O_j}{\partial \beta} & \frac{\partial O_j}{\partial \gamma} \\ \frac{\partial O_k}{\partial \alpha} & \frac{\partial O_k}{\partial \beta} & \frac{\partial O_k}{\partial \gamma} \end{bmatrix} =$$

$$\begin{bmatrix} C_\alpha S_\gamma - S_\alpha S_\beta C_\gamma & C_\alpha C_\beta C_\gamma & S_\alpha C_\gamma - C_\alpha S_\beta S_\gamma \\ S_\alpha S_\gamma + C_\alpha S_\beta C_\gamma & S_\alpha C_\beta C_\gamma & -C_\alpha C_\gamma - S_\alpha S_\beta S_\gamma \\ 0 & -S_\beta C_\gamma & -C_\beta S_\gamma \end{bmatrix} \tag{26}$$

Since the tool orientation error $\Delta\boldsymbol{O}$ is defined as the minimum deviation between the original tool orientation vector and the constructed spline, it can be obtained that:

$$\Delta\boldsymbol{O} \le \|\boldsymbol{J}_o(\boldsymbol{\Psi}_i - \boldsymbol{\Phi}_5)\| + \mathcal{O}(\|\boldsymbol{\varepsilon}_o\|^2) \tag{27}$$

Substitute Eq. (18) and Eq. (24) into the above equation:

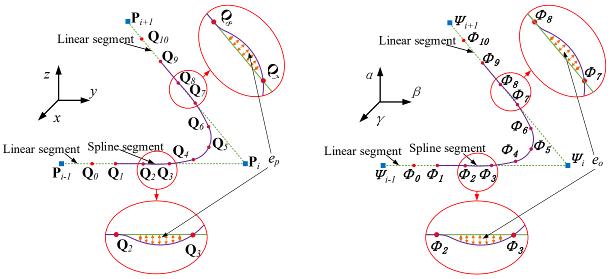
$$\varepsilon_o \le \frac{2\,sin\left(\frac{\varepsilon_{ow}}{2}\right)}{\|\boldsymbol{J}_o \boldsymbol{T}_{om}\|} + \mathcal{O}(\|\boldsymbol{\varepsilon}_o\|^2) \tag{28}$$

Under small-angle assumption, higher-order terms are negligible, and substitute Eq. (28) into Eq. (21), the final constraints of$L_o$can be obtained as:

$$L_o = min\left(\frac{2\,sin\left(\frac{\varepsilon_{ow}}{2}\right)}{\|\boldsymbol{J}_o \boldsymbol{T}_{om}\|}, \frac{\|\overrightarrow{\boldsymbol{\Psi}_{i-1}\boldsymbol{\Psi}_i}\|\,sin\left(\frac{\theta_o}{2}\right)}{12\|\boldsymbol{T}_{o0}\|}, \frac{\|\overrightarrow{\boldsymbol{\Psi}_i\boldsymbol{\Psi}_{i+1}}\|\,sin\left(\frac{\theta_o}{2}\right)}{12\|\boldsymbol{T}_{o1}\|}\right)$$
(29)

3.4. Constraint of the tool tip and tool orientation error

Smoothing the tool tip position and orientation at transition corner by using the developed ACR spline will not only result in smoothing error ε_p and ε_o, but also the deviation between the spline segment and the linear segment e_p and $e_o$due to the local property of the ACR spline (e.g. the spline between control points Q_2 and Q_3 is affected by Q_4), as shown in Fig. 7. Therefore, it is necessary to control the deviation error e_p and e_o by adjusting the shape of spline, and an analytical optimization method is developed to minimize$e_p$and$e_o$by optimizing the adjustment parameters κ and λ in this subsection.



a) corner smoothing of tool tip position

b) corner smoothing of tool orientation

Fig. 7. The deviation between spline segment and linear segment

Taking the spline segment and linear segment between control points $Q_2 = (x_2, y_2, z_2)$ and $Q_3 = (x_3, y_3, z_3)$as an example, the error between the spline segment and the original linear segment can be defined as:

$$e_p(\kappa, \lambda) = \int_{x_2}^{x_3} \left(R_{y,2}(x) - y(x)\right)^2 + \left(R_{z,2}(x) - z(x)\right)^2 dx \tag{30}$$

where $R_{y,2}(x)$and$R_{z,2}(x)$are ACR spline functions in the interval $x \in [x_2, x_3]$ with control points Q_1, Q_2, Q_3 and Q_4, $y(x)$ and $z(x)$ are the functions of linear segment between Q_2 and Q_3, can be expressed as:

$$\begin{cases} y(x) = y_2 + \frac{(y_3 - y_2)(x - x_2)}{x_3 - x_2} \\ z(x) = z_2 + \frac{(z_3 - z_2)(x - x_2)}{x_3 - x_2} \end{cases} \tag{31}$$

The ACR spline functions $R_{y,2}(x)$ and $R_{z,2}(x)$ in the interval$x \in [x_2, x_3]$can be rewritten as the following form by substituting Eq. (6) into Eq. (5):

$$\begin{cases} R_{y,2}(x) = A_{y,2}(x)\kappa + B_{y,2}(x)\lambda + C_{y,2}(x) \\ R_{z,2}(x) = A_{z,2}(x)\kappa + B_{z,2}(x)\lambda + C_{z,2}(x) \end{cases} \tag{32}$$

where

$$\begin{cases} A_{y,2}(x) = l_0(t)y_1 + l_1(t)y_2 - l_0(t)y_3 - l_1(t)y_4 \\ B_{y,2}(x) = m_0(t)y_1 + m_1(t)y_2 + m_2(t)y_3 - \big(m_0(t) + m_1(t) + m_2(t)\big)y_4 \\ C_{y,2}(x) = n_1(t)y_2 + \big(1 - n_1(t)\big)y_3 \\ A_{z,2}(x) = l_0(t)z_1 + l_1(t)z_2 - l_0(t)z_3 - l_1(t)z_4 \\ B_{z,2}(x) = m_0(t)z_1 + m_1(t)z_2 + m_2(t)z_3 - \big(m_0(t) + m_1(t) + m_2(t)\big)z_4 \\ C_{z,2}(x) = n_1(t)z_2 + \big(1 - n_1(t)\big)z_3 \end{cases} \tag{33}$$

where $t = \frac{(x-x_2)}{(x_3-x_2)}$, The following optimization model can be derived by substituting Eq. (32) into Eq. (30):

$$\begin{cases} min \qquad e_p(\kappa, \lambda) = D_1\kappa^2 + D_2\lambda^2 + D_3\kappa\lambda + D_4\kappa + D_5\lambda + D_6 \\ \qquad\qquad s.t. \qquad \kappa, \lambda \in R \end{cases} \tag{34}$$

where

$$\begin{cases} D_1 = \int_{x_2}^{x_3}\left(\left(A_{y,2}(x)\right)^2 + \left(A_{z,2}(x)\right)^2\right)dx \\ D_2 = \int_{x_2}^{x_3}\left(\left(B_{y,2}(x)\right)^2 + \left(B_{z,2}(x)\right)^2\right)dx \\ D_3 = \int_{x_2}^{x_3}\left(A_{y,2}B_{y,2} + A_{z,2}B_{z,2}\right)dx \\ D_4 = \int_{x_2}^{x_3}\left(A_{y,2}(x)\left(C_{y,2}(x) - y(x)\right) + A_{z,2}(x)\left(C_{z,2}(x) - z(x)\right)\right)dx \\ D_5 = \int_{x_2}^{x_3}\left(B_{y,2}(x)\left(C_{y,2}(x) - y(x)\right) + B_{z,2}(x)\left(C_{z,2}(x) - z(x)\right)\right)dx \\ D_6 = \int_{x_2}^{x_3}\left(\left(C_{y,2}(x) - y(x)\right)^2 + \left(C_{z,2}(x) - z(x)\right)^2\right)dx \end{cases} \tag{35}$$

It is not difficult to obtain that $e_p(\kappa, \lambda) \geq 0$, thus the minimum of $e_p(\kappa, \lambda)$can be solved by:

$$\begin{cases} \frac{\partial e_p(\kappa,\lambda)}{\partial \kappa} = 0 \\ \frac{\partial e_p(\kappa,\lambda)}{\partial \lambda} = 0 \end{cases} \tag{36}$$

By combining with Eq. (34):

$$\begin{cases} D_1\kappa + D_3\lambda = -D_4 \\ D_3\kappa + D_2\lambda = -D_5 \end{cases} \tag{37}$$

The adjustment parameters κ and λ corresponding to the minimum deviation error e_p can be obtained by solving Eq. (37).

4. SYNCHRONIZATION OF THE TOOL ORIENTATION AND TOOL TIP POSITION

In order to guarantee the C³ continuity of the smoothed tool path and joints of robot, the varying rate of the tool orientation should be synchronized with the speed of the tool tip displacement. The developed ACR spline is utilized to replace the remaining position and orientation linear segments after smoothing, as shown in Fig. 8, where $Q_{0,i}$, $Q_{1,i}$, $Q_{2,i}$ and $Q_{3,i}$ are control points of the smoothed position curves of the ith corner, $Q_{7,i-1}$, $Q_{8,i-1}$, $Q_{9,i-1}$ and $Q_{10,i-1}$ are control points of the smoothed position curves of the $(i-1)$thcorner, the linear segment between $Q_{10,i-1}$and $Q_{0,i}$ is the remaining position linear segment between two corners. Similarly, $\Phi_{0,i}$, $\Phi_{1,i}$, $\Phi_{2,i}$ and $\Phi_{3,i}$ are control points of the smoothed orientation curves of the$i$thcorner, $\Phi_{7,i-1}$,$\Phi_{8,i-1}$, $\Phi_{9,i-1}$and $\Phi_{10,i-1}$are control points of the smoothed orientation curves of the $(i-1)$thcorner, the linear segment between $\Phi_{10,i-1}$and $\Phi_{0,i}$is the remaining orientation line between two corners.

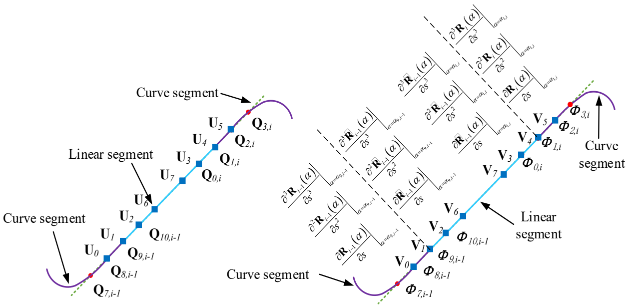


Fig. 8. Synchronization of the tool tip position and orientation

Due to local property of ACR spline, the curve between control points $Q_{1,i}$ and $Q_{2,i}$is just affected by $Q_{0,i}$,$Q_{0,i}$,$Q_{1,i}$,$Q_{2,i}$ and $Q_{3,i}$, which are all located on the original linear segment, thus, this curve is exactly coincident with the original linear segment. Similarly, the curves between control points $Q_{8,i-1}$and $Q_{9,i-1}$,$\Phi_{1,i}$and $\Phi_{2,i}$, $\Phi_{8,i-1}$and $\Phi_{9,i-1}$are all coincident with the original linear segment. Therefore, C³ continuity between the remaining position (orientation) linear segments and the smoothed curve segment can be realized if the control points of the remaining segment are located on the original linear segment.

In this study, the developed ACR splines are utilized to replace the remaining position and orientation linear segments. For the sake of simplicity, the control points $Q_{8,i-1}$,$Q_{9,i-1}$,$Q_{10,i-1}$, $\mathbf{Q}_{0,i}$ $Q_{0,i}$, $\mathbf{Q}_{1,i}$ $Q_{1,i}$and $Q_{2,i}$ of the two adjacent curve segments are considered as the control points of the remaining position linear segments, and rewritten as U_0,U_1, $\mathbf{U}_2$ U_3,U_4 and U_5, as shown in Figure 8.

Similarly, the control points $\Phi_{8,i-1}$, $\Phi_{9,i-1}$,$\Phi_{10,i-1}$, $\Phi_{0,i}$,$\Phi_{1,i}$ and $\Phi_{2,i}$ of the two adjacent curve segments are considered as the control points of the remaining orientation linear segments, and rewritten as V_0,V_1, $\mathbf{V}_2$ V_3,V_4 and V_5.

In addition to the abovementioned constraints, to satisfy the C³ continuous synchronization of the tool orientation with respect to the tool tip position, as shown in Figure 8, the first, second and third order differentials of the tool orientation $R(\alpha)$ relative to the tool tip displacements$s$should be equal at the intersection of the junction between the curve segment and the linear segment.

At the junction point V_1shown in Fig. 8, on the curve side, differentials of the tool orientation with respect to the tool tip displacement are:

$$\begin{cases} \left.\frac{\partial \boldsymbol{R}_{i-1}(\alpha)}{\partial s}\right|_{\alpha=\alpha_{9,i-1}} = \left(\frac{\partial \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha}\frac{\partial \alpha}{\partial s}\right)_{\alpha=\alpha_{9,i-1}} \\ \left.\frac{\partial^2 \boldsymbol{R}_{i-1}(\alpha)}{\partial s^2}\right|_{\alpha=\alpha_{9,i-1}} = \left(\frac{\partial^2 \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha^2}\left(\frac{\partial \alpha}{\partial s}\right)^2 + \frac{\partial \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha}\frac{\partial^2 \alpha}{\partial s^2}\right)_{\alpha=\alpha_{9,i-1}} \\ \left.\frac{\partial^3 \boldsymbol{R}_{i-1}(\alpha)}{\partial s^3}\right|_{\alpha=\alpha_{9,i-1}} = \left(\frac{\partial^3 \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha^3}\left(\frac{\partial \alpha}{\partial s}\right)^3 + 3\frac{\partial^2 \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha^2}\frac{\partial \alpha}{\partial s}\frac{\partial^2 \alpha}{\partial s^2} + \frac{\partial \boldsymbol{R}_{i-1}(\alpha)}{\partial \alpha}\frac{\partial^3 \alpha}{\partial s^3}\right)_{\alpha=\alpha_{9,i-1}} \end{cases} \tag{37}$$

On the linear side, differentials of the tool orientation with respect to the tool tip displacement are:

DOI 10.2478/ama-2025-0088

$$\begin{cases} \left.\dfrac{\partial \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s}\right|_{\alpha=\alpha_{9,i-1}} = \left(\dfrac{\partial \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha}\dfrac{\partial \alpha}{\partial s}\right)_{\alpha=\alpha_{9,i-1}} \\ \left.\dfrac{\partial^2 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s^2}\right|_{\alpha=\alpha_{9,i-1}} = \left(\dfrac{\partial^2 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha^2}\left(\dfrac{\partial \alpha}{\partial s}\right)^2 + \dfrac{\partial \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha}\dfrac{\partial^2 \alpha}{\partial s^2}\right)_{\alpha=\alpha_{9,i-1}} \\ \left.\dfrac{\partial^3 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s^3}\right|_{\alpha=\alpha_{9,i-1}} = \left(\dfrac{\partial^3 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha^3}\left(\dfrac{\partial \alpha}{\partial s}\right)^3 + 3\dfrac{\partial^2 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha^2}\dfrac{\partial \alpha}{\partial s}\dfrac{\partial^2 \alpha}{\partial s^2} + \dfrac{\partial \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial \alpha}\dfrac{\partial^3 \alpha}{\partial s^3}\right)_{\alpha=\alpha_{9,i-1}} \end{cases} \tag{38}$$

Since the control points related to the ACR spline between control points $\varPhi_{8,i-1}$and $\varPhi_{9,i-1}$,$V_1$and $V_2$are all located on the original linear segment. Therefore, in order to realize the synchronous control of the tool tip position and tool orientation related the tool tip displacement, it is necessary to insert four additional control points U_6,$U_7$and V_6,V_7, as shown in Fig. 8, and the following formulas must be satisfied:

$$\begin{aligned} &\left\|\overrightarrow{\boldsymbol{U}_0\boldsymbol{U}_1}\right\| = \left\|\overrightarrow{\boldsymbol{U}_1\boldsymbol{U}_2}\right\| = \left\|\overrightarrow{\boldsymbol{U}_2\boldsymbol{U}_6}\right\| \\ &\left\|\overrightarrow{\boldsymbol{U}_5\boldsymbol{U}_4}\right\| = \left\|\overrightarrow{\boldsymbol{U}_4\boldsymbol{U}_3}\right\| = \left\|\overrightarrow{\boldsymbol{U}_3\boldsymbol{U}_7}\right\| \\ &\left\|\overrightarrow{\boldsymbol{V}_0\boldsymbol{V}_1}\right\| = \left\|\overrightarrow{\boldsymbol{V}_1\boldsymbol{V}_2}\right\| = \left\|\overrightarrow{\boldsymbol{V}_2\boldsymbol{V}_6}\right\| \\ &\left\|\overrightarrow{\boldsymbol{V}_5\boldsymbol{V}_4}\right\| = \left\|\overrightarrow{\boldsymbol{V}_4\boldsymbol{V}_3}\right\| = \left\|\overrightarrow{\boldsymbol{V}_3\boldsymbol{V}_7}\right\| \end{aligned} \tag{39}$$

The specific derivation steps of Eq (13) are detailed in Appendix A, and then the following equations can be obtained:

$$\begin{cases} \left.\dfrac{\partial \boldsymbol{R}_{i-1}(\alpha)}{\partial s}\right|_{\alpha=\alpha_{9,i-1}} = \left.\dfrac{\partial \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s}\right|_{\alpha=\alpha_{9,i-1}} \\ \left.\dfrac{\partial^2 \boldsymbol{R}_{i-1}(\alpha)}{\partial s^2}\right|_{\alpha=\alpha_{9,i-1}} = \left.\dfrac{\partial^2 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s^2}\right|_{\alpha=\alpha_{9,i-1}} = 0 \\ \left.\dfrac{\partial^3 \boldsymbol{R}_{i-1}(\alpha)}{\partial s^3}\right|_{\alpha=\alpha_{9,i-1}} = \left.\dfrac{\partial^3 \widetilde{\boldsymbol{R}}_{i-1}(\alpha)}{\partial s^3}\right|_{\alpha=\alpha_{9,i-1}} = 0 \end{cases} \tag{40}$$

For computational efficiency we adopt the nominal choice$\kappa = \lambda = 0$when constructing the ACR basis for the remaining linear segments; this nominal choice reduces the basis to the standard CR form while retaining analytic expressions for positions and derivatives. Importantly, choosing $\kappa = \lambda = 0$does not cause a vanishing first derivative or a parametric singularity under normal (non-degenerate) geometric conditions: because the remaining linear segments have non-zero span and the control points used for the remaining-segment construction are distinct points lying on the original linear segment (see Figure 8 and Eq. (39)–(40)), the chord lengths are positive and the resulting CR basis yields non-zero local derivatives. In other words, derivative values on these remaining segments are determined by the control-point geometry (not byκ,λalone), and therefore C^1–C^3 matching conditions remain enforceable when$\kappa = \lambda = 0$.

The implementation includes a simple detection-and-fallback step to handle the geometric degenerate cases that can produce near-zero parametric derivatives (for example, coincident control points or zero-length linear segments). Before finalizing the remaining-segment construction the algorithm computes the magnitude of the first derivative on the remaining segment. If $\|R'(s)\| < \tau$ (withτa small numerical threshold), one of two light-weight remedies is applied: (i) restore small nominal shape perturbation by setting $\kappa = \kappa_{eps}$ or $\lambda = \lambda_{eps}$ with $\kappa_{\mathrm{eps}}, \lambda_{\mathrm{eps}} \in [10^{-6}, 10^{-3}]$; or (ii) apply a tiny geometric perturbation to the remaining-segment control points (move them along the segment by a distance$\delta \ll \varepsilon_p$). Either remedy restores a non-zero derivative while preserving deviation bounds. This fallback is algorithmic (post-construction) and does not change the analytical derivations; it only activates in pathological geometric cases (coincident points or zero-length segments) which are easily detected at runtime.

5. NUMERICAL SIMULATION

In this section, the developed ACR-spline-based C^3 continuous tool path smoothing method for robotic machining is validated through numerical simulation. The test tool path used in the simulation is depicted in Fig. 9, consisting of four linear segments and three corner transitions, with the corresponding parameters listed in Tab. 1. The deviation error tolerances for the tool tip position and orientation are set as $\varepsilon_p = 0.8$ mm and $\varepsilon_o = 0.01$rad, respectively. The resulting smoothed tool tip position and tool orientation are illustrated in Fig. 10.

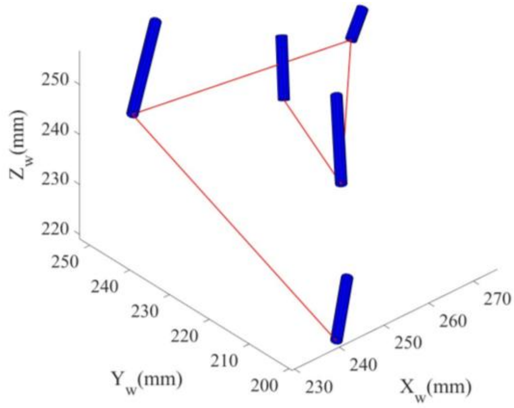
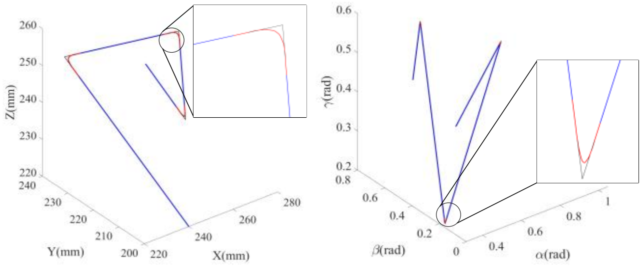


Fig. 9. The testing tool path utilized in numerical simulation

Tab. 1. Parameters of the testing tool path

<table>
<tr><th>No. of corners</th><th>No. of tool path</th><th>Tool tip position (x, y, z) (mm)</th><th>Tool orientation (α, β, γ) (rad)</th></tr>
<tr><td rowspan="4">1</td><td rowspan="3">1</td><td rowspan="2">(240, 200, 220)</td><td rowspan="2">(0.6283, 0.5236, 0.3142)</td></tr>
<tr></tr>
<tr><td rowspan="2">(230, 240, 250)</td><td rowspan="2">(1.0472, 0.7854, 0.3927)</td></tr>
<tr><td rowspan="2">2</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">(270, 230, 252.2)</td><td rowspan="2">(0.3142, 0.1571, 0.2094)</td></tr>
<tr><td rowspan="2">3</td></tr>
<tr><td rowspan="4">3</td><td rowspan="2">(250, 210, 242)</td><td rowspan="2">(0.6283, 0.7854, 0.5236)</td></tr>
<tr><td rowspan="3">4</td></tr>
<tr><td rowspan="2">(235, 230, 246.8)</td><td rowspan="2">(0.3142, 0.3927, 0.5236)</td></tr>
<tr></tr>
</table>



a) Tool tip position b) Tool orientation

Fig. 10. The smoothed tool tip position and tool orientation

The deviation errors of the tool tip position and orientation in the WCS under the preset deviation error tolerances are presented in Tab. 2. It can be observed that the maximum deviation errors for both the tool tip position ε_p and orientation ε_o in the WCS do not exceed the predefined tolerances. Furthermore, since the tool tip position e_p and orientation e_o are controlled by the adjustment parameters κ , λ , it can be concluded from Tab. 2 that e_p and e_o remain sufficiently small to ensure minimal deviation between the smoothed tool path and the original tool path. The corresponding κ,λ are provided in parentheses.

Tab. 2. Deviation errors of the tool tip and tool orientation with error tolerances $\varepsilon_p = 0.8$mm and $\varepsilon_o = 0.01$rad

No. of corners		1	2	3
Tool tip Position error	ε_p (mm)	0.8	0.8	0.8
	e_p (mm^3) (κ,λ)	5.4858e-06 (0.48, -5.4e-4)	2.3160e-06 (0.51, -1.4e-4)	3.9818e-05 (0.47, 5.3e-4)
Tool orientation error	ε_o (rad)	0.01	0.01	0.01
	e_o (rad^3) (κ,λ)	2.2647e-11 (0.46, -2.4e-3)	1.2184e-10 (0.49, 7.2e-3)	2.8408e-11 (0.47, -3.0e-3)

To further assess the efficacy of the developed ACR-spline-based tool path smoothing method under varying error tolerances, the deviation error thresholds for the tool tip position and orientation were reset to $\varepsilon_p = 1.2$mm and $\varepsilon_o = 0.005$rad, respectively.

As indicated in Tab. 3, the maximum deviation errors for both the tool tip position ε_p and orientation ε_o remain below the preset tolerances. Moreover, e_p and e_o controlled by the adjustment parameters κ, λ are still small enough to ensure that the deviation between the smoothed tool path and the original tool path.

Tab. 3. Deviation errors of the tool tip and tool orientation with error tolerances $\varepsilon_p = 1.2$mm and $\varepsilon_o = 0.005$rad

No. of corners		1	2	3
Tool tip Position error	ε_p (mm)	1.2	1.2	1.2
	e_p (mm^3) (κ,λ)	1.8557e-05 (0.48, -8.1e-4)	7.8179e-06 (0.51, -2.2e-4)	1.3406e-04 (0.47, 7.9e-4)
Tool orientation error	ε_o (rad)	0.005	0.005	0.005
	e_o (rad^3) (κ,λ)	2.8111e-12 (0.46, -1.2e-3)	1.5905e-11 (0.49, 3.6e-3)	3.5244e-12 (0.47, -1.5e-3)

In Section 4, it is deduced that the synchronization errors of the tool orientation relative to the tool tip displacement at the junctions between curve splines and linear splines are identically zero. To verify this conclusion, the 1st-, 2nd- and 3rd-order synchronization errors at these junctions are computed, as illustrated in Fig. 11 and Fig. 12. For comparison, the synchronization errors of the smoothed tool path obtained using the method described in [29] is also evaluated on the same test tool path. The results clearly indicate that the synchronization errors of the tool path smoothed by the developed ACR-spline-based method are zero at all examined orders, whereas the method in [29] yields non-zero errors. This outcome verifies the C^3 continuous synchronization of the tool orientation with respect to the tool tip displacement achieved by the developed method.

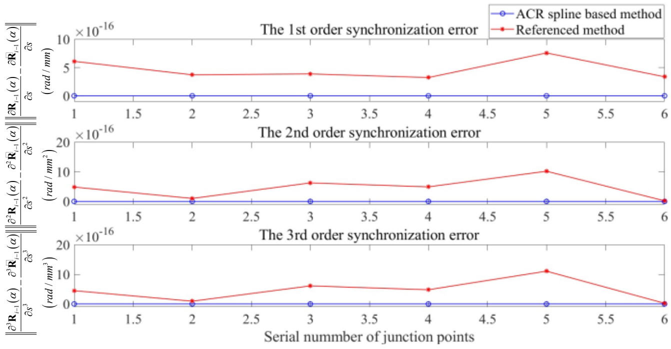


Fig. 11. Synchronization errors of the tool orientation with respect to the tool tip displacement with $\varepsilon_p = 0.8$mm and $\varepsilon_o = 0.01$rad

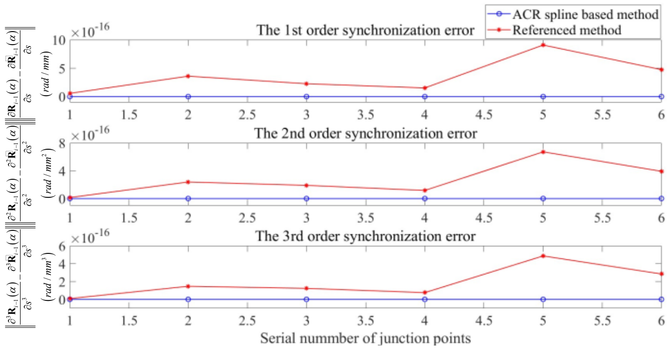


Fig. 12. Synchronization errors of the tool orientation with respect to the tool tip displacement with $\varepsilon_p = 1.2$mm and $\varepsilon_o = 0.005$ rad

Since the developed ACR-spline-based C^3 continuous corner smoothing method achieves 3rd-order continuity for both the tool tip position and tool orientation, it mathematically guarantees continuous jerk commands. To validate this performance, a C^3 continuous cubic acceleration profile [31] is employed to interpolate the smoothed tool path generated by the developed method under the preset error tolerances for tool tip position and orientation. In this interpolation, the nominal tangential feedrate, acceleration, and jerk were set to 30 mm/s, 3000 mm/s², and 30000 mm/s³, respectively. Fig. 13 presents the resulting kinematic profiles—namely, displacement, velocity, acceleration, and jerk-which are smooth and continuous up to the jerk level, as expected.

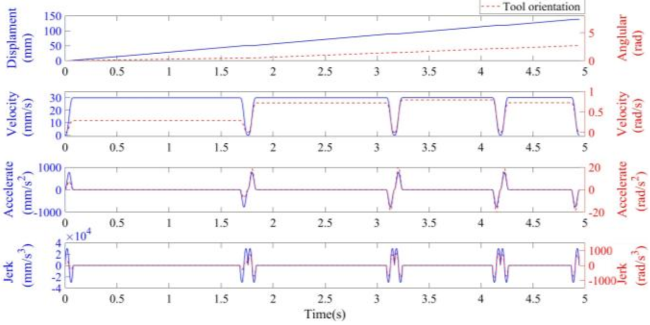
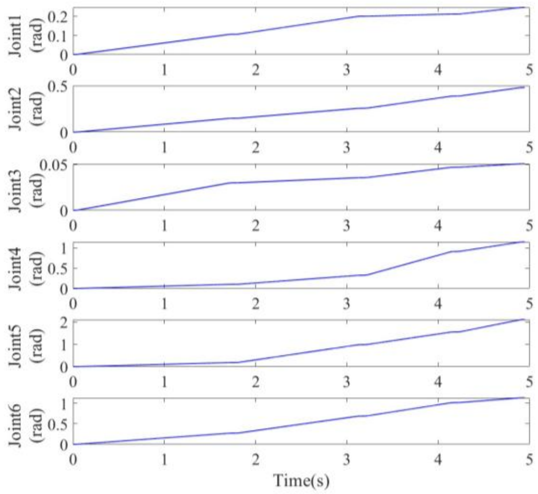
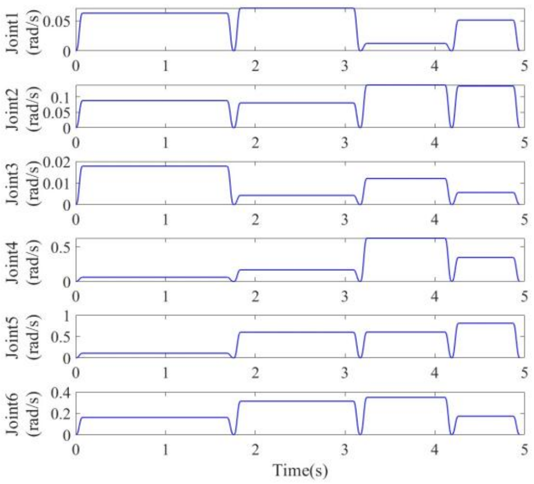


Fig. 13. Kinematic profile after interpolation

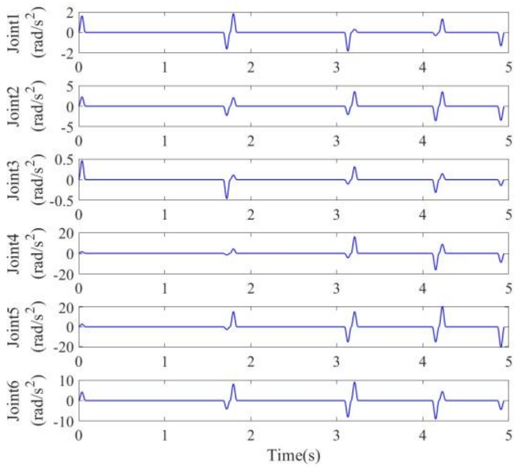
Furthermore, Fig. 14 illustrates the kinematic profiles for each joint, clearly demonstrating that the motion of every robot joint maintains third-order continuity without any abrupt changes in jerk. This observation further substantiates the effectiveness and practical applicability of the developed tool path smoothing method.



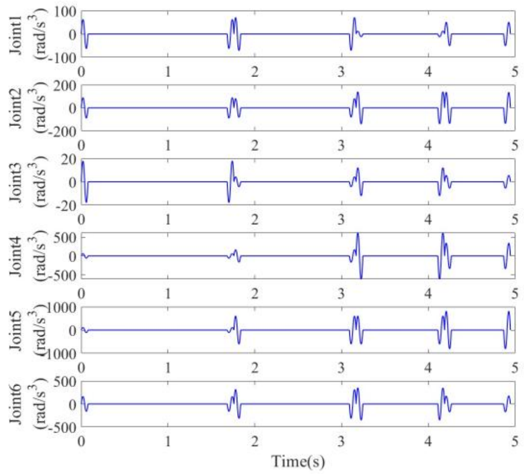
a) Angular of joints



b) Velocity of joints



c) Accelerate of joints



d) Jerk of joints

Fig. 14. Kinematic profiles of each joint of robot

6. CONCLUSIONS

Ensuring high-order smoothness and analytical continuity of tool paths is essential for robotic machining systems to achieve both computational efficiency and desirable dynamic behaviour. In this paper we presented an analytical C³-continuous smoothing method based on ACR splines. By inserting ACR segments at corners between discrete linear tool paths and computing their control points analytically, the proposed approach avoids iterative construction and delivers substantial computational savings. Local deviation introduced by corner smoothing is explicitly constrained through the control-point geometry, while the residual approximation error from the ACR basis is minimized using a simple yet effective optimization. The resulting procedure enforces C³ continuity by synchronizing tool-tip orientation with tool-tip displacement and by replacing the remaining linear segments with analytically determined ACR arcs. Numerical simulations demonstrate that the method meets prescribed maximum-deviation tolerances, preserves third-order synchronization between orientation and displacement, and produces smooth jerk profiles in both workpiece and joint spaces—properties that are important for high-precision robotic machining.

We note that orientation-parameterization issues commonly encountered in CAM/CNC workflows—most notably gimbal lock, the non-uniqueness of Euler-angle representations, and discontinuities introduced by angle wrapping—arise at the implementation level within post-processing and controller-interface stages. The present paper focuses on positional smoothing and feedrate planning within the CNC motion-planning layer; in our formulation, any required orientation synchronization is expressed only as a constraint, while the machine-specific representation and final conversion of orientations are assumed to be handled by the CAM post-processor and/or the controller interface (i.e., the software modules that consume the smoothed tool path and generate machine-ready commands). To make this assumption explicit, Appendix B provides recommended implementation-level strategies—including the use of singularity-free representations such as unit quaternions or rotation matrices, SLERP interpolation, $SO(3)$ exponential/log-map updates, and safe conversion procedures to machine-specific Euler conventions. These recommendations support robust integration of the proposed positional-smoothing method into practical CAM/CNC pipelines without altering the theoretical framework or the numerical results presented in this work.

Future work will pursue several complementary directions. We plan to incorporate real-time adaptive adjustments driven by process dynamics and force feedback to further improve robustness under varying cutting conditions. Experimental validation on an industrial robot platform will assess the method's performance in real machining tasks. We will also investigate automatic parameter tuning—potentially leveraging machine-learning techniques—to increase efficiency and generality across a wider range of workpieces and machine systems. As part of these extensions, the orientation-handling measures described in Appendix B will be integrated into a complete post-processing pipeline and their impact on controller tracking and overall process quality will be evaluated.

Appendix A. Derivation of the equal–chord–length synchronization conditions

This section derives the synchronization conditions for the tool position and orientation, ensuring both are smoothly synchronized at the junctions of adjacent spline segments. The derivation utilizes Eq. (13), which defines the C^3 continuity for the ACR spline, and is expressed in a simplified format consistent with the notation in the manuscript.

Step 1: Review of ACR Spline Properties

The ACR spline is used to smooth the tool path, and it has the following properties:

Interpolation: The spline passes through the given control points.

C^3 Continuity: The spline ensures that position, velocity, acceleration, and jerk (third derivative) are continuous at junctions.

Locality: Each spline segment is influenced by four control points.

Adjustability: The shape of the spline can be controlled via two parameters, which help in shaping the curvature to avoid cusps and ensure smooth transitions.

Step 2: Review of Eq. (13) — C^3 Continuity

Eq. (13) guarantees C^3 continuity for the ACR spline, ensuring that at the junction of two adjacent segments, the position, velocity, acceleration, and jerk are continuous. Specifically, it ensures that for any junction t_i, the following continuity conditions hold:

$$\frac{d^k[\boldsymbol{p}(t)]}{dt^k}|t = t_i = continuous, \frac{d^k[\boldsymbol{R}(t)]}{dt^k}|t = t_i = continuous$$

which ensure that both the tool position $p(t)$ and orientation $R(t)$ are smooth at the junctions, where $k=0,1,2,3$.

Step 3: Synchronization of Position and Orientation

To ensure synchronization between the tool position $p(t)$ and tool orientation $R(t)$, the derivatives of both the position and orientation must match at the junctions. Specifically, we need to ensure that:

$$\frac{d^k[\boldsymbol{p}(t)]}{dt^k} = \frac{d^k[\boldsymbol{R}(t)]}{dt^k}$$

which ensures that not only the position but also the velocity, acceleration, and jerk of the tool's position and orientation are synchronized at each junction.

Step 4: Equal-Chord-Length Condition for Synchronization

To achieve synchronization, we need to ensure that the parameterization of the tool path is consistent across segments. The equal-chord-length condition ensures that the parameter intervals between adjacent control points are equal, which helps maintain the synchronization of the derivatives.

The equal-chord-length condition can be written as:

$$\Delta t_1 = \Delta t_2 = \cdots = \Delta t_n$$

where Δt_i is the parameter interval for each spline segment. This condition guarantees uniformity in the parameterization, aiding in the synchronization of both position and orientation.

Step 5: Control Points and Synchronization

Since the control points of the ACR spline between the adjacent segments are located on the original linear segment, we can derive that to synchronize the tool tip position and tool orientation, four additional control points must be inserted. These control points, denoted U_6, U_7, V_6, and V_7, ensure that both position and orientation are synchronized at the junction, as shown in the figure.

The synchronization condition between position and orientation is then expressed as:

$$\begin{aligned}
\|\overrightarrow{\boldsymbol{U}_0\boldsymbol{U}_1}\| &= \|\overrightarrow{\boldsymbol{U}_1\boldsymbol{U}_2}\| = \|\overrightarrow{\boldsymbol{U}_2\boldsymbol{U}_6}\| \\
\|\overrightarrow{\boldsymbol{U}_5\boldsymbol{U}_4}\| &= \|\overrightarrow{\boldsymbol{U}_4\boldsymbol{U}_3}\| = \|\overrightarrow{\boldsymbol{U}_3\boldsymbol{U}_7}\| \\
\|\overrightarrow{\boldsymbol{V}_0\boldsymbol{V}_1}\| &= \|\overrightarrow{\boldsymbol{V}_1\boldsymbol{V}_2}\| = \|\overrightarrow{\boldsymbol{V}_2\boldsymbol{V}_6}\| \\
\|\overrightarrow{\boldsymbol{V}_5\boldsymbol{V}_4}\| &= \|\overrightarrow{\boldsymbol{V}_4\boldsymbol{V}_3}\| = \|\overrightarrow{\boldsymbol{V}_3\boldsymbol{V}_7}\|
\end{aligned}$$

This ensures that the control points for both the position and orientation are distributed evenly across the spline segments, providing synchronization between the two.

Thus, Eq. (39) ensures synchronization of both the tool position and orientation, guaranteeing that the motion is continuous up to the third derivative. By introducing the equal-chord-length condition and the additional control points, we can achieve the necessary synchronization for high-precision robotic machining.

Appendix B. Notes on orientation handling and recommended CNC post-processing

Since the core contribution of this paper is a positional C^3 smoothing algorithm and its deviation guarantees, the main text focuses on tool-centerline smoothing and feedrate planning. For practitioners who integrate the proposed positional smoothing into a CAM/CNC workflow, the following practical measures are recommended to avoid orientation singularities in the post-processing stage:

Use a singularity-free internal representation. Represent orientations internally as unit quaternions ($q \in S^3$) or rotation matrices ($R \in SO(3)$). Do not perform interpolation or gradient-based optimization directly on Euler angles.

Interpolation and composition. For interpolation between key orientations use spherical linear interpolation (SLERP). For small optimization increments, use the $SO(3)$ exponential and logarithm maps: represent local updates as 3-vector elements of $SO(3)$ and apply them by exponentiation to the current rotation.

Quaternion sign consistency. Because q and $-q$ represent the same physical rotation, enforce sign consistency before interpolation or comparison: if $\mathrm{dot}(q_1, q_2) < 0$ then replace q_2 by $-q_2$. This guarantees interpolation along the shortest arc on S^3 and avoids spurious 180° flips.

Optimization parameterization. When orientation increments are required in optimization loops, use a minimal local parameterization $SO(3)$ rotation vector via the log map) around the current orientation rather than global Euler parameters. This avoids singular linearizations and improves convergence properties.

Final conversion and continuity enforcement. Convert quaternion/matrix results to the machine's required Euler convention only at the final output stage. During conversion, perform angle unwrapping and branch selection to maintain temporal continuity and to minimize angular jumps. If a resulting Euler angle is close to a known gimbal configuration (e.g., pitch ≈ ±90°), consider switching to an alternate representation for that interval or slightly perturbing the sampling to avoid controller singularity crossing.

Controller interface and verification. If the target NC controller supports quaternion or matrix inputs, prefer those interfaces. Otherwise, include a verification pass in the post-processor that checks for excessive angular velocity or acceleration after conversion to controller angles and adjusts the timeline or re-parameterizes orientations if machine limits would be violated.

These measures are standard practice in robotics and advanced CAM implementations and can be implemented without changing the ACR-based positional smoothing algorithm presented in this study.

REFERENCES

1. Verl A, Valente A, Melkote S, Brecher C, Ozturk E, Tunc LT. Robots in machining. CIRP Ann. 2019;68(2):799–822. Available from: https://doi.org/10.1016/j.cirp.2019.05.009
2. Kim SH, Nam E, Ha TI, Hwang SH, Lee JH, Park SH, Min BK. [Title missing in your source — keep as original]. Int J Precis Eng Manuf. 2019;20(9-12):1629–1642. Available from: https://doi.org/10.1007/s12541-019-00187-w
3. Lu L, Zhang J, Fuh JYH, Han J, Wang H. Time-optimal tool motion planning with tool-tip kinematic constraints for sculptured surfaces. Robot Comput-Integr Manuf. 2020;65:101969. Available from: https://doi.org/10.1016/j.rcim.2020.101969
4. Shrivastava A, Dalla VK. Strategy of smooth motion planning of multi-axes space manipulator avoiding dynamic singularity in Cartesian space. J Braz Soc Mech Sci Eng. 2022;44(7):278. Available from: https://doi.org/10.1007/s40430-022-03578-9
5. Guachetá-Alba JC, Nunez DA, Dutra MS, Mauledoux M, Aviles OF. Multi-objective optimization of 6-DOF deposition trajectories using NSGA-II. J Braz Soc Mech Sci Eng. 2023;45(11):610. Available from: https://doi.org/10.1007/s40430-023-04495-1
6. Han J, Liu X, Jiang Y, Xia L, Lu L. An irredundant G01 tool path generation method for five-axis machining considering tool tip and orientation errors. Int J Adv Manuf Technol. 2019;103(1-4):1033–1044. Available from: https://doi.org/10.1007/s00170-019-03583-8
7. Veysi M, Soltanpour MR, Khooban MH. A novel self-adaptive modified bat fuzzy sliding mode control of robot manipulator in presence of uncertainties in task space. Robotica. 2015;33(10):2045–2064. Available from: https://doi.org/10.1017/S0263574714001258
8. Izadbakhsh A, Khorashadizadeh S. Robust task-space control of robot manipulators using differential equations for uncertainty estimation. Robotica. 2017;35(9):1923–1938. Available from: https://doi.org/10.1017/S0263574716000588
9. Wang P, Yang H, Xue K. Jerk-optimal trajectory planning for Stewart platform in joint space. In: 2015 IEEE International Conference on Mechatronics and Automation (ICMA); 2015. p.1932–1937. Available from: https://doi.org/10.1109/ICMA.2015.7237781
10. Chettibi T. Smooth point-to-point trajectory planning for robot manipulators by using radial basis functions. Robotica. 2019;37(3):539–559. Available from: https://doi.org/10.1017/S0263574718001169
11. Kucuk S. Maximal dexterous trajectory generation and cubic spline optimization for fully planar parallel manipulators. Comput Electr Eng. 2016;56:634–647. Available from: https://doi.org/10.1016/j.compeleceng.2016.07.012
12. Liu Q, Huang T. Inverse kinematics of a 5-axis hybrid robot with non-singular tool path generation. Robot Comput-Integr Manuf. 2019;56:140–148. Available from: https://doi.org/10.1016/j.rcim.2018.06.003
13. Lu Y, Zhang J, Fuh JYH, Han J, Wang H. Smooth tool path optimization for flank milling based on the gradient-based differential evolution method. J Manuf Sci Eng. 2016;138(8):081009. Available from: https://doi.org/10.1115/1.4032969
14. Li D, Zhang W, Zhou W, Shang T, Fleischer J. Dual NURBS path smoothing for 5-axis linear path of flank milling. Int J Precis Eng Manuf. 2018;19(12):1811–1820. Available from: https://doi.org/10.1007/s12541-018-0209-6
15. Hu Q, Chen Y, Jin X, Yang J. A real-time C3 continuous local corner smoothing and interpolation algorithm for CNC machine tools. J Manuf Sci Eng. 2019;141(4):041004. Available from: https://doi.org/10.1115/1.4042606
16. Zhang Y, Ye P, Zhang H, Zhao M. A local and analytical curvature-smooth method with jerk-continuous feedrate scheduling along linear toolpath. Int J Precis Eng Manuf. 2018;19(10):1529–1538. Available from: https://doi.org/10.1007/s12541-018-0180-2
17. Xiao QB, Wan M, Liu Y, Qin XB, Zhang WH. Space corner smoothing of CNC machine tools through developing 3D general clothoid. Robot Comput-Integr Manuf. 2020;64:101949. Available from: https://doi.org/10.1016/j.rcim.2020.101949
18. Yang X, You Y, Yang WA. Simultaneous optimization of curvature and curvature variation for tool path generation in high-speed milling of corners. J Braz Soc Mech Sci Eng. 2022;44:68. Available from: https://doi.org/10.1007/s40430-022-03360-x
19. Jin Y, Bi Q, Wang Y. Dual-Bezier path smoothing and interpolation for five-axis linear tool path in workpiece coordinate system. Adv Mech Eng. 2015;7(7):1687814015595211. Available from: https://doi.org/10.1177/1687814015595211
20. Yan Y, Zhang L, Zhang K. Corner smoothing transition algorithm for five-axis linear tool path. Procedia CIRP. 2016;56:604–609. Available from: https://doi.org/10.1016/j.procir.2016.10.119
21. Huang J, Du X, Zhu LM. Real-time local smoothing for five-axis linear toolpath considering smoothing error constraints. Int J Mach Tools Manuf. 2018;124:67–79. Available from: https://doi.org/10.1016/j.ijmachtools.2017.10.001
22. Zhao X, Zhao H, Wan S, Li X, Ding H. An analytical decoupled corner smoothing method for five-axis linear tool paths. IEEE Access. 2019;7:22763–22772. Available from: https://doi.org/10.1109/ACCESS.2019.2898703
23. Huang X, Zhao F, Tao T, Mei X. A novel local smoothing method for five-axis machining with time-synchronization feedrate scheduling. IEEE Access. 2020;8:89185–89204. Available from: https://doi.org/10.1109/ACCESS.2020.2992022
24. Yuen A, Zhang K, Altintas Y. Smooth trajectory generation for five-axis machine tools. Int J Mach Tools Manuf. 2013;71:11–19. Available from: https://doi.org/10.1016/j.ijmachtools.2013.04.002
25. Tulsyan S, Altintas Y. Local toolpath smoothing for five-axis machine tools. Int J Mach Tools Manuf. 2015;96:15–26. Available from: https://doi.org/10.1016/j.ijmachtools.2015.04.014
26. Yang J, Yuen A. An analytical local corner smoothing algorithm for five-axis CNC machining. Int J Mach Tools Manuf. 2017;123:22–35. Available from: https://doi.org/10.1016/j.ijmachtools.2017.07.007
27. Hu Q, Chen Y, Jin X, Yang J. A real-time C3 continuous tool path smoothing and interpolation algorithm for five-axis machine tools. J Manuf Sci Eng. 2020;142(4):041002. Available from: https://doi.org/10.1115/1.4046091
28. Yang J, Qi Q, Adili A, Ding H. An analytical tool path smoothing algorithm for robotic machining with the consideration of redundant kinematics. Robot Comput-Integr Manuf. 2024;89:102768. Available from: https://doi.org/10.1016/j.rcim.2024.102768
29. Yang J, Li D, Ye C, Ding H. An analytical C3 continuous tool path corner smoothing algorithm for 6R robot manipulator. Robot Comput-Integr Manuf. 2020;64:101947. Available from: https://doi.org/10.1016/j.rcim.2019.101947
30. Li J, Liu C. The C3 Quasi Catmull-Rom spline function with parameters. Mathematica Numerica Sinica. 2018;40(1):96–106. Available

from: https://doi.org/10.12286/jssx.2018.1.96
31. Erkorkmaz K. Optimal trajectory generation and precision tracking control for multi-axis machines [PhD thesis]. Vancouver (BC): The University of British Columbia; 2004.
Available from: https://open.library.ubc.ca/collections/ubctheses/831/items/1.0080705

This study is funded by the Fundamental Research Funds for the Central Universities (NT2021019), National Natural Science Foundation of China (51775279).

Xu-Lin Cai: https://orcid.org/0009-0001-3471-3012

Wen-An Yang: https://orcid.org/0000-0003-2242-5678