



Smooth trajectory generation based on contour error constraint and parameter correction b-spline

Kai Zhao^{1,2} · Shurong Li³ · Zhongjian Kang⁴ · Zhe Liu³

Received: 5 May 2021 / Accepted: 9 November 2021 / Published online: 11 January 2022
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2021

Abstract

In terms of CNC machining task, high feedrate and high accuracy are of great significance. In this paper, an integrated jerk-limited method of minimal time trajectory planning with confined contour error (IJLMTPC) is proposed. Firstly, an optimal control method is utilized to obtain the minimal time feedrate profile. To compensate the contour error, it is added into the optimal control framework as a constraint. Based on the high-order transfer functions of every axis servo model, the contour errors constraint is formulated as the function of tracking errors. However, the optimal control problem (OCP) is difficult to be solved. Hence, the OCP is transformed into two convex subproblems. Since the axial jerk elements are included in the contour error constraint, they are no longer considered as dependent constraints in the two subproblems specially. Afterwards, two subproblems are discretized with control vector parameterization. Consequently, two subproblems are more efficient to be solved with nonlinear programming, and the minimal time feedrate is obtained. Secondly, considering that the OCP is solved on the domain of curve parameter, it is inconvenient for real-time interpolation. The resulting feedrate from solving the OCP is transformed into the corresponding profile on time domain, while the limited tangent jerk is imposed on. Hence, the feedrate is smoothed. Finally, to alleviate the fluctuation of feedrate, the real-time interpolation is performed based on the parameter correction b-spline. Two benchmark tool paths are adopted to test the proposed scheme, and the effectiveness is verified.

Keywords Parameter correction b-spline · Computer numerical control · Control vector parameterization · Interpolation · Smooth trajectory planning · Contour error constraint

1 Introduction

In general, the high-quality machining of a complex curve consists of two aspects, i.e., high speed and high accuracy.

In terms of high speed machining, it means that a machine tool can processing a complex curve in the shortest cycle time. An appropriate feedrate scheduling method guarantees short cycle time. For example, in view of a

long tool path, an interval adaptive feedrate scheduling method based on a dynamic moving look-ahead window is proposed [1]. This method divides a long spline into several intervals, and generates the smooth s-shape feedrate profile for each interval successively based on limited jerks. The finite impulse response filter (FIR) is another efficient smooth feedrate scheduling method. Based on FIR-chain techniques, the smooth and accurate reference motion profile can be generated, as well as the acceleration and jerk continuous motion profiles [2]. For a blend path, it is of significance with an efficient feedrate scheduling method. A typical blend path consists of small line segments and small parametric curves. And the sharp corners formed by the intersection of two line segments are smoothed with parametric curves which have C^2 [3] or C^3 continuity [4]. The feedrate of this kind of blend curve can be planned with the bidirectional scan algorithm [5].

More recently, for pursuing short cycle time, the optimization method is introduced to feedrate scheduling. For example, a heuristic trajectory planning algorithm is

✉ Shurong Li
lishurong@bupt.edu.cn

¹ College of Control Science and Engineering, China University of Petroleum (East China), Qingdao, 266580, China

² College of Computer Science and Information Engineering, Anyang Institute of Technology, Anyang, 455000, China

³ School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, 100876, China

⁴ College of New Energy, China University of Petroleum (East China), Qingdao, 266580, China

proposed [6]. It can provide a near-optimal minimum time trajectory for problems with higher order dynamic states. To suppress the unwanted vibration in the processing of parts, motion commands are generated based on optimal frequency spectra of each axial acceleration [7]. In this method, kinematic limits are considered to constrain the frequency spectra, and the Quadratic programming is used to obtain the optimal solution. In addition, optimization methods also play an important role in the corner blending problems. Especially, mixing linear path with polynomial curves can be formulated as a convex optimization problem [8].

Since the optimal control has inherent advantages in path planning [9, 10] and trajectory generation [11, 12], it has attracted more and more attentions. A Bang-Bang control is adopted to plan the feedrate for a multi-axis system in Frenet-Serret frame [13]. Afterwards, linear programming is used to solve the optimal control problem restrained by kinematic constraints. And a feedback interpolation method is developed for real-time interpolation. Meanwhile, a parallel windowing (PWin) algorithm is also integrated into the process of feedrate optimization [14]. Considering dynamics constraints, the path tracking task of a manipulator is formulated as an optimal control problem [15]. In this literature, to reduce the solution difficulty, the optimal control problem is transformed into a convex optimization problem. Since the feedrate profile of a machine tool is optimized, the cycle time is significantly reduced with these methods. However, to improve optimization efficiency, appropriate solving techniques should be adopted during applying optimization-based methods.

In terms of the high-precision machining, there are many ways to realize this objective. For example, through restraining some indexes, high accuracy machining results can be gained. Tracking errors of a multi-axis machine tool are usually considered as constraints of an optimal control problem except kinematic ones, such as jerks, accelerations, and feedrate [16]. Then machining accuracy can be improved by restraining tracking errors below permissible value. However, the tracking error reduction is an indirect way to improve accuracy, whereas the contour error is a direct manner. Here is a simple explanation. In general, a machining task is implemented in cartesian space. Taking a 3-axis machine tool for example, a processed curve is the synthesis of displacement of the tool tip on three axes. Three independent servo systems work in coordination to drive the tool tip to the reference position. Even if tracking errors on all axes are small, the contour error is not necessarily small, and vice versa. Therefore,

an appropriate controller can be designed to control the contour error directly, such as cross coupling controller (CCC). For example, a single-input fuzzy-logic controller is taken as the CCC to improve the contouring accuracy [17]. Except CCC strategy, the model predictive control (MPC)-based scheme is also used for contour error suppression. In each receding optimization, both contour error components and tracking errors are taken into performance index, with assigning the former greater weight than the latter [18]. For a dual 5-dof manipulator, a MPC-based controller is adopted to reduce the equivalent contour error [19], and the integral sliding mode control [20] is integrated to improve robustness.

From above, contour error reduction and feedrate optimization are generally implemented alone. Actually, we can perform both at the same time. For instance, through introducing contour error constraints, servo error pre-compensation and time optimal feed optimization are performed synchronously [21]. Under the optimal control framework, this kind of synchronous optimization is more easily to be realized.

The real-time interpolation is another significant factor which affects the machining accuracy. For parameter tool paths, there are many mature interpolation methods, such as Taylor expansion-based interpolator [22], predictor-corrector-based interpolator [23], and Adams-Bashforth methods [24]. However, there are some drawbacks needed to be overcome for these methods. For instance, some methods may lead to the fluctuation of feedrate. Some other methods may take longer time than interpolation period to obtain a next interpolation point. Hence, some scholars proposed minimal fluctuation methods to suppress feedrate fluctuation. A polynomial equation-based interpolation method for Nurbs tool paths is proposed [25]. According to sample step size, this method formulates the polynomial equation with respect to the curve parameter. Afterwards, the Newton's method is used to obtain numerical solution of the next interpolation point. Nurbs curves can be also used to construct the relationship between curve parameter and arc length of a tool path [26]. It is a key point of the minimal fluctuation methods that the exact functional relationship should be determined in advance.

In this paper, we propose an integrated jerk-limited method of minimal time trajectory planning with confined contour error (IJLMTPC). Firstly, considering the OCP of feed planning, we adopt the CVP technique to solve the convex optimization problems. To improve solving efficiency, the jerk constraints are replaced with contour error ones, because they are included in the contour error

deduced based on high-order transfer function. Secondly, since the feedrate profile obtained from the first stage is the function of curve parameter, it is mapped into time domain to adapting with six types of forward planning profiles. The limited tangent jerk is also used to further smooth the feed. The distance constraints are introduced into the backward planning to ensure that the deceleration can be forecast. Finally, a minimal fluctuation methods is used in the real-time interpolation. In order to accurately describe the functional relationship between arc length and curve parameter, we propose parameter correction b-spline and corresponding refining algorithm. The outline of the proposed scheme is elaborated with following diagram in Fig. 1.

The remainder of this paper is arranged as follows. For convenience, Section 2 introduces the parameter correction b-spline, as well as the refining algorithm. The basic optimal control formulation known as nominal problem of smooth trajectory generation (NPSTG) is investigated in Section 3. And the simplification of NPSTG is also described. Our proposed scheme, i.e., the integrated jerk-limited method of minimal time trajectory planning with confined contour error, is addressed in Section 4. The scheme is also named IJLMTPC for short. Section 5 illustrates the proposed

method with two benchmark tool paths. Finally, conclusions are given in Section 6.

2 Parameter correction b-spline

Since most free curves are of non-arc-length parameterization, researchers proposed many available methods for obtaining the relationship between parameters and arc lengths [27]. Herein, a quintic b-spline is adopted to realize the mapping between the arc length and the parameter of a free curve.

Given a free parameter curve $C(q) = [x(q), y(q)]^T, q \in [0, 1]$, the approximation arc length $\Lambda(q_a, q_b)$ on the interval $[q_a, q_b]$ can be calculated with the Simpson rule, i.e.,

$$\Lambda(q_a, q_b) = \frac{\bar{h}}{3}(l(q_a) + 4l(q_c) + l(q_b)), \tag{1}$$

where $q_c = (q_a + q_b)$, $\bar{h} = (q_b - q_a)/2$, and $l = \|C'(q)\|_2$ denotes the 2-norm. Given predefined arc length tolerance error ε_1 , $\Lambda(q_a, q_b)$ is considered as the arc length, if it satisfies the following condition [28],

$$|\Lambda(q_a, q_c) + \Lambda(q_c, q_b) - \Lambda(q_a, q_b)| < 10 * \varepsilon_1. \tag{2}$$

If (2) is not satisfied, the interval $[q_a, q_b]$ is split into two equal parts, i.e., $[q_a, q_c]$ and $[q_c, q_b]$. Then, (1) and (2) are applying to the two intervals, respectively.

By applying above operations iteratively, the whole interval $[0, 1]$ is split into many subintervals, and (2) holds on each subinterval. All the endpoints of these subintervals form a set $\mathcal{E}_1 = \{q_0, q_1, \dots, q_{N-1}, q_N\}$, where $0 = q_0 < q_1 < \dots < q_{N-1} < q_N = 1$. Accordingly, at each $q_i (i = 0, 1, \dots, N)$, the corresponding arc length Λ_i is obtained. Note that Λ_i is arc length on the interval $[q_0, q_i]$, which can be easily computed through simple addition. If the interpolation technology was used to gain the relationship between q_i and Λ_i at this time, the interpolation accuracy would not be guaranteed.

To guarantee the accuracy, although (2) has been satisfied on each subinterval, we still construct another set $\mathcal{E}_2 = \{q_{(0,1)}, \dots, q_{(N-1,N)}\}$ as a test data set, where $q_{(i,i+1)} (i = 0, \dots, N - 1)$ is the midpoint of interval $[q_i, q_{i+1}]$. Define $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$, and arrange the elements in ascending order in set \mathcal{E} . Equation (1) is adopted to compute the arc length of $C(q)$ on each interval defined by any two adjacent elements of \mathcal{E} . As a result, a series of value pair (Λ_j, q_j) are obtained, where $q_j (j = 0, \dots, 2N) \in \mathcal{E}$, and Λ_j is the arc length on the interval $[q_0, q_j]$. To avoid ill-conditioning, Λ_j is normalized through

$$\tau_j^f = (\Lambda_j - \Lambda_0)/(\Lambda_{2N} - \Lambda_0). \tag{3}$$

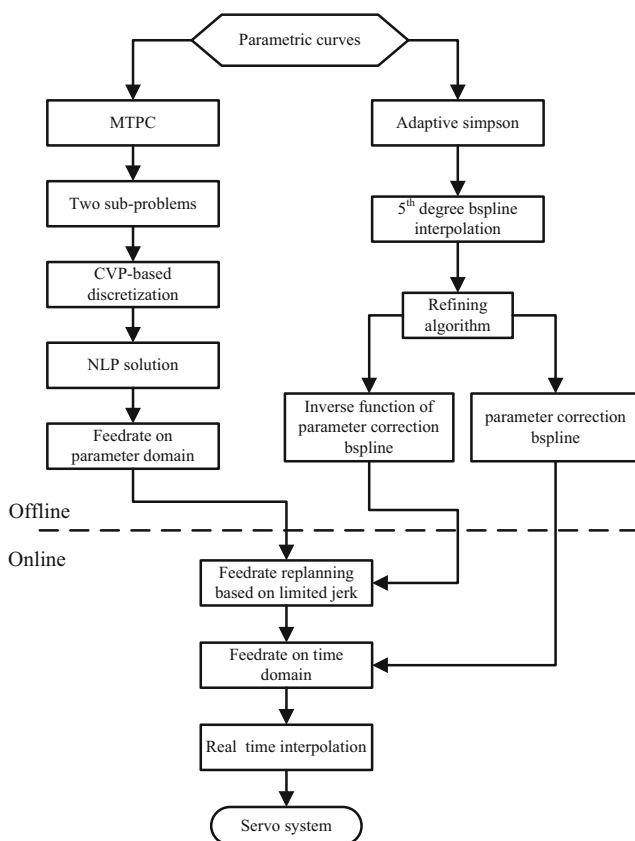


Fig. 1 The architecture of proposed IJLMTPC

A 5th-degree b-spline is interpolated to value pairs $(\tau_j^f, q_j), q_j \in \mathcal{E}_1$, which is expressed as

$$q_j = Q(\tau_j^f) = \sum_{l=0}^L F_{l,5}(\tau_j^f) P_l, \tag{4}$$

where P_l denotes the control point sequence, and $F_{l,5}(\tau_j^f)$ is the basis function.

The knots vector $\mathbf{SX} = [0, \dots, 0, \underbrace{sx_6, sx_7, \dots, sx_{M-p-1}, 1, \dots, 1}]$ is calculated through average method according to $q_j \in \mathcal{E}_1$, and $M = N + p + 1$. Define $\Phi = [q_0, \dots, q_j, \dots]^T, q_j \in \mathcal{E}_1$. From (4), the following linear equations are obtained,

$$\mathbf{TP} = \Phi, \tag{5}$$

where

$$\mathbf{T}_{(N+1) \times (N+1)} = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ F_{0,5}(q_1) & F_{1,5}(q_1) & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & F_{L-1,5}(q_{N-1}) & F_{L,5}(q_{N-1}) \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$\mathbf{P}_{(N+1) \times 1} = [P_0 \ P_1 \ \dots \ P_{N-1} \ P_N]^T,$$

and

$$\Phi_{(N+1) \times 1} = [\Phi_0 \ \Phi_1 \ \dots \ \Phi_{L-1} \ \Phi_L]^T.$$

Equation (5) can be solved efficiently with some methods, such as least square and LU decomposition. In this paper, the Bragman iteration algorithm is utilized to solve (5), in order to avoid possible ill-condition and to gain sparse solution. Once control points vector is obtained, the quintic b-spline is expressed. However, the initial interpolation curve $Q^0(\tau)$ is not necessarily satisfactory. We propose Algorithm 1 to refine the interpolation curve with test value pairs $[\tau_j^t, q_j], q_j \in \mathcal{E}_2$. The final output $q_j = Q^f(\tau_j^f)$ is the parameter correction b-spline, which is used to obtain precise parameter values during the interpolation stage of CNC machining.

Algorithm 1 Refine the interpolation curve $Q^{(0)}(\tau)$.

Require: $\varepsilon_2, Maxiter, Q^0(\tau), q_g^t, \tau_g^t, N^{(0)}$, where ε_2 is the allowable interpolation error, and $Maxiter$ denote the maximum number of iterations. Note that subscript g indicates the index of \mathcal{E}_2 , whereas the subscript j denotes the index of \mathcal{E} . q_g^t is the corresponding parameters of an element in \mathcal{E}_2 , and $N^{(0)}$ is the number of elements.

Ensure: Refined curve $Q^f(\tau)$.

- 1: **while** $iter \leq Maxiter$ **do**
- 2: **for** $g = 0 : N^{(0)}$ **do**
- 3: Substitute parameter q_g^t into $Q^0(\tau)$, and gain τ_g^0 .
- 4: **if** $|\tau_g^0 - \tau_g^t|/\tau_g^t > \varepsilon_2$ **then**
- 5: Find the nearest two elements to q_g^t in \mathcal{E}_1 , $q_g^t \in \mathcal{E}_2$. Then divide the interval formed by the two elements into three equal parts. Add three trisection points into \mathcal{E}_1 .
- 6: **end if**
- 7: **end for**
- 8: $iter = iter + 1$.
- 9: $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$. Arrange elements in \mathcal{E} in ascending order.
- 10: Perform Eqs. 1 and 3, and solve Eq. 5.
- 11: Interpolate curve $Q^f(\tau)$ to value pairs $[\tau_j^f, q_j], q_j \in \mathcal{E}_1$.
- 12: $Q^0(\tau) = Q^f(\tau)$.
- 13: **end while**

Likewise, the mapping function from parameter to normalized arc length can also be interpolated to value pairs

$$\tau_j^f = Q^{-1}(q_j) = \sum_{l=0}^L F_{l,5}(q_j) \bar{P}_l. \tag{6}$$

This quintic b-spline function, i.e., $\tau_j^f = Q^{-1}(q_j)$, is known as the inverse function of parameter correction b-spline, which can be used to calculate the arc lengths.

3 Problem of smooth time optimal feedrate scheduling

The problem of smooth time optimal feedrate scheduling (PSTOF) is usually formulated as an optimization problem. To realize high speed machining, the time optimal objective function is given as

$$OBJ = \min_V \int_0^{t_f} 1 dt, \tag{7}$$

where V is the optimal feedrate profile. When machining a planar curve $\mathbf{C}(q)$, the motion of a machine tool is restrained

by its kinematic abilities, such as command feedrate, axial accelerations, and axial jerks, which are written as

$$V^2(t) \leq F_B^2, \quad |A_h(t)| \leq A_{Bh}, \quad |J_h(t)| \leq J_{Bh}, \quad (8)$$

where F_B is the command feedrate bound, A_{Bh} are axial acceleration bounds, J_{Bh} are axial jerk bounds, and $h \in \{x, y\}$. Nevertheless, the final time is free, which makes the PSTOF difficult to be solved.

The tool path $\mathbf{C}(q) = [x(q), y(q)]^T$ is a free parametric curve. Assume that $\mathbf{C}(q)$ has a third-order continuous derivative at least. Let the symbol “ $\dot{\cdot}$ ” be a derivative with respect to time. And Define that “ $\ddot{\cdot}$ ” denotes a derivative with respect to parameter q . After introducing parameter q into velocity, acceleration, and jerk, they can be written as [29]

$$\begin{cases} \mathbf{v} = \dot{\mathbf{C}} = \mathbf{C}'\dot{q}, \\ \mathbf{A} = \dot{\mathbf{v}} = \mathbf{C}''\ddot{q} + \mathbf{C}'\dot{q}^2, \\ \mathbf{J} = \dot{\mathbf{A}} = \mathbf{C}'''\ddot{q} + 3\mathbf{C}''\dot{q}\ddot{q} + \mathbf{C}'\dot{q}^3, \end{cases} \quad (9)$$

where

$$\mathbf{C}' = \frac{d\mathbf{C}}{dq}, \mathbf{C}'' = \frac{d^2\mathbf{C}}{dq^2}, \mathbf{C}''' = \frac{d^3\mathbf{C}}{dq^3},$$

with parameter being neglected for simplification.

Define state variables as $b = \dot{q}^2$ and $\alpha = \ddot{q}$. Let the control variable be $\beta = \frac{\ddot{q}}{\dot{q}}$ [30]. The PSTOF is transformed into an optimal control problem. Obviously, the system dynamics are expressed as

$$\begin{cases} b'(q) = 2\alpha(q), b(0) = \dot{q}_0^2, \\ \alpha'(q) = \beta(q), \alpha(0) = \ddot{q}_0, \end{cases} \quad (10)$$

with a path constraint and final time constraints,

$$\begin{cases} b(q) > 0, q \in [0, 1], \\ b(1) = \dot{q}_f^2, \alpha(1) = \ddot{q}_f. \end{cases} \quad (11)$$

Since $\dot{q} = \frac{dq}{dt}$, we have $dt = \frac{dq}{\dot{q}}$ [31]. Through replacing variable t with q , (7) can be transformed into following cost function

$$OBJ = \min_{b,\alpha,\beta} \int_0^1 \frac{1}{\dot{q}} dq = \min_{b,\alpha,\beta} \int_0^1 \frac{1}{\sqrt{b}} dq, \quad (12)$$

where \dot{q} is known as the parameter velocity. Likewise, after introducing state variables and control variable, the constraints (8) are rewritten as

$$V^2 = \|\mathbf{v}\|^2 = \|\mathbf{C}'\|^2 b \leq F_B^2, \quad (13a)$$

$$|A_h| = |\mathbf{C}''_h b + \mathbf{C}'_h \alpha| \leq A_{Bh}, \quad (13b)$$

$$\begin{cases} |J_h| = |\sqrt{b}\Gamma| \leq J_{Bh}, \\ \Gamma = \mathbf{C}'''_h b + 3\mathbf{C}''_h \alpha + \mathbf{C}'_h \beta, \end{cases} \quad (13c)$$

where $h \in \{x, y\}$. Consequently, the optimal problem consisting of (12), (10), (11), and (13) is known as the nominal problem of smooth trajectory generation (NPSTG) [32].

To improve solution efficiency, the NPSTG is further reformulated into two convex subproblems. One of them is the following problem of minimal time trajectory planning (PMTTP),

$$\begin{aligned} OBJ &= \min_{b,\alpha} \int_0^1 \frac{1}{\sqrt{b}} dq \\ s.t. &\begin{cases} b'(q) = 2\alpha(q), b(0) = \dot{q}_0^2, \\ \|\mathbf{C}'(q)\|b(q) \leq F_B^2, \\ |\mathbf{C}''_h(q)hb(q) + \mathbf{C}'_h(q)hu(q)| \leq A_{Bh}, \\ b(q) > 0, q \in [0, 1], \\ b(1) = \dot{q}_f^2, \alpha(1) = \ddot{q}_f, \\ h \in \{x, y\}. \end{cases} \end{aligned} \quad (14)$$

Consequently, the optimal solution $b^*(q)$ of (14) is adopted to relax (13c) as [29]

$$|J(t)| \leq |\sqrt{b^*(q)}\Gamma(q)| \leq J_{Bh}. \quad (15)$$

Replace (13c) with (15) in the NPSTG, and pseudo problem of smooth trajectory generation (PPSTG) is obtained. As a result, the nonlinearity of the NPSTG is significantly alleviated, and a smooth feedrate profile is generated. Furthermore, because of convexity of two subproblems, the unique solution is obtained, which guarantees that the same part is gained in each machining. However, the feedrate profile is a function of parameter q , other than that of time. This kind of feedrate profile is inconvenient for time-based interpolation method. In addition, the contour error constraint is not considered in problems above. To settle these drawbacks, we propose an integrated jerk-limited method of minimal time trajectory planning with confined contour error (IJLMTPC).

4 The proposed method

The contour error is defined as the shortest distance from the current actual position to the desired machined tool path. It is of significance for machining accuracy. In our opinion, limited jerks should be taken into consideration at last. In that case, the coarse feedrate profile gained with optimal control can be smoothed further. Hence, the contour error constraint is added into the NPSTG, instead of axial jerk constraints.

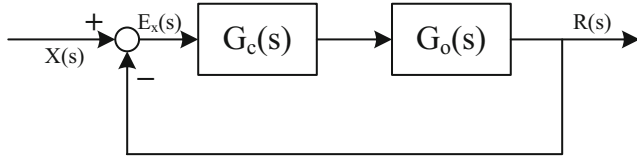


Fig. 2 A typical diagram of closed-loop servo model

4.1 Tracking error development

For a typical servo system of a single axis, its dynamic model is shown in Fig. 2 [33]. Taking x -axis for example, the closed-loop transfer function of the model is derived as

$$G_{tx}(s) = \frac{G_c(s)G_o(s)}{1 + G_c(s)G_o(s)} = \frac{R(s)}{X(s)}, \tag{16}$$

where $R(s)$ and $X(s)$ denote the output signal and reference signal, respectively. Accordingly, the tracking error transfer function is described as

$$G_{ex}(s) = \frac{X(s) - R(s)}{X(s)} = \frac{1}{1 + G_c(s)G_o(s)} = \frac{E_x(s)}{X(s)}. \tag{17}$$

Through Taylor series expansion of G_{ex} at $s = 0$, we have

$$E_x(s) = G_{ex}(0)X(s) + G_{ex}^{(1)}(0)sX(s) + \frac{1}{2!}G_{ex}^{(2)}(0)s^2X(s) + \frac{1}{3!}G_{ex}^{(3)}(0)s^3X(s) + \dots \tag{18}$$

For the servo system of a CNC machine tool, the instantaneous part of tracking error is very small. In terms of final theorem of Laplace transform, the tracking error is approximated with its steady part, i.e.,

$$e_x \approx \lim_{s \rightarrow 0} (sG_{ex}(s)X(s)). \tag{19}$$

The inverse Laplace transform of (18) is written as

$$e_x \approx \sum_{i=0}^{\infty} W_i^x x^{(i)}(t). \tag{20}$$

where $W_i^x = \frac{1}{i!}G_{ex}^{(i)}(0)$, indicates the dynamic error coefficient. Since the open loop model of a position servo system is I-type at least, the coefficient W_0^x is always zero [33]. Ignoring the high-order term, e_x is further approximated as

$$e_x \approx W_1^x x^{(1)}(t) + W_2^x x^{(2)}(t) + W_3^x x^{(3)}(t), \tag{21}$$

where $x^{(i)}(t)$ is the i th-order derivative with respect to time. According to the derivative relationship, we have

$$\begin{aligned} x^{(1)}(t) &= C'_x(q)\sqrt{b(q)}, \\ x^{(2)}(t) &= C''_x(q)b(q) + C'_x(q)\alpha(q), \\ x^{(3)}(t) &= \sqrt{b(q)}(C'''_x(q)b(q) + C''_x(q)\alpha(q) + C'_x(q)\beta(q)). \end{aligned} \tag{22}$$

Actually, it is clear that each axial jerk is included in the tracking error. In the next subsection, the tracking errors

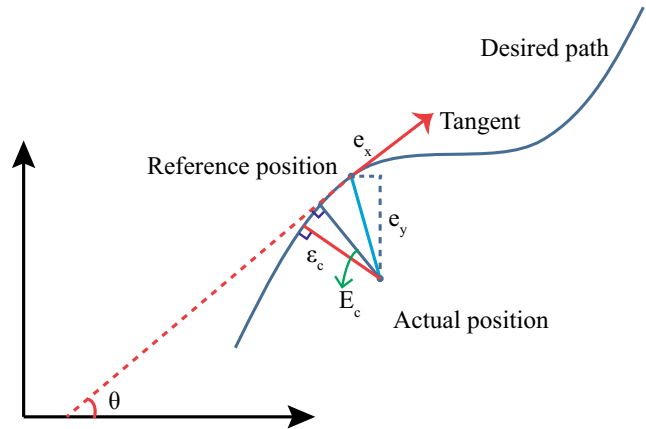


Fig. 3 Contour error for a 2D curve

of two axes are used to estimate the contour error. That is another reason why we replace (13c) with the contour error constraint.

4.2 Contour error estimation

Generally, to improve real-time performance, the contour error is estimated according to some information of real-time machining, such as desired positions, actual positions, and the tracking errors.

The contour error can be estimated by many methods. For a planer tool path shown in Fig. 3, the contour error is estimated as [34]

$$E_c = -\sin\theta e_x + \cos\theta e_y, \tag{23}$$

where e_x and e_y denote the tracking errors of x -axis and y -axis, respectively. θ is the angle between tangent of tool path and X -axis of the machine tool coordinate system, which can be calculated with

$$\theta = \cos^{-1} \frac{C'_x(q)}{\sqrt{C_x'^2(q) + C_y'^2(q)}}. \tag{24}$$

From (21), (22), and (23), the contour error is rewritten as

$$E_c = w_1b(q) + w_2\alpha(q) + w_3\sqrt{b(q)} + (w_4b(q) + w_5\alpha(q) + w_6\beta(q))\sqrt{b(q)}. \tag{25}$$

where $w_i (i = 1, \dots, 6)$ are real value functions of q .

4.3 trajectory planning with confined contour error

To clarify the problem of minimal time trajectory planning with confined error (MTPC), it is formulated as following

form,

$$OBJ = \min_{b, \alpha, \beta} \int_0^1 \frac{1}{\sqrt{b(q)}} dq$$

$$s.t. \begin{cases} b'(q) = 2\alpha(q), b(0) = \dot{q}_0^2, \\ \alpha'(q) = \beta(q), \alpha(0) = \ddot{q}_0, \\ \|C'(q)\|b(q) \leq F_B^2, \\ |C_h''(q)b(q) + C_h'(q)\alpha(q)| \leq A_{Bh}, \\ |E_c| \leq E_B, \\ b(q) > 0, q \in [0, 1], \\ b(1) = \dot{q}_f^2, \alpha(1) = \ddot{q}_f, \\ h \in \{x, y\}. \end{cases} \quad (26)$$

To efficiently solve problem (26), it is also divided into two convex subproblems. One is exactly the (14). Referring to the (15), the contour error constraint is relaxed as

$$|E_c| \leq \hat{E}_c$$

$$= |w_1b(q) + w_2\alpha(q) + (w_3 + w_4b(q) + w_5\alpha(q) + w_6\beta(q))\sqrt{b^*(q)}|. \quad (27)$$

After replacing $|E_c|$ with \hat{E}_c in (26), we have the other convex subproblem, which is

$$OBJ = \min_{b, \alpha, \beta} \int_0^1 \frac{1}{\sqrt{b(q)}} dq$$

$$s.t. \begin{cases} \text{other constraints,} \\ |\hat{E}_c| \leq E_B. \end{cases} \quad (28)$$

In order to obtain the optimal solution of problem (14) and (28), the control vector parameterization (CVP) is adopted to discretize them. Taking (28) for example, divide parameter domain into $K + 1$ grids, i.e., $0 = p_0 < p_1 < \dots < p_{K-1} < p_K = 1$. The corresponding control action is defined as

$$\beta = \sum_{k=1}^K \bar{\beta}_k \varpi_k(p), \quad (29)$$

where $\bar{\beta} = [\bar{\beta}_1, \dots, \bar{\beta}_K]^T$ denotes the decision vector, and $\varpi_k(p)$ is the k th basis function. For piecewise constant approximation, $\varpi_k(p)$ is defined as

$$\varpi_k(p) = \begin{cases} 1 & p \in [p_{k-1}, p_k), \\ 0 & \text{otherwise.} \end{cases} \quad (30)$$

Hence, in this paper, the control action and the decision vector is the same.

Substitute the control vector into system dynamics, and through solving derivative equations, corresponding state vectors are obtained, i.e.,

$$\begin{aligned} \bar{\alpha}(\bar{\beta}) &= [\bar{\alpha}_1, \dots, \bar{\alpha}_K]^T, \\ \bar{\mathbf{b}}(\bar{\beta}) &= [\bar{b}_1, \dots, \bar{b}_K]^T. \end{aligned} \quad (31)$$

Define $\Delta p = p_k - p_{k-1}$, the objective function is discretized as

$$OBJ = \sum_{k=1}^K \frac{1}{\bar{b}_k} \Delta p. \quad (32)$$

Consequently, problem (28) is transformed into a nonlinear programming problem as following,

$$OBJ = \min_{\bar{\beta}_k} \sum_{k=1}^K \frac{1}{\bar{b}_k} \Delta p$$

$$s.t. \begin{cases} \|C'(q_k)\|\bar{b}_k \leq F_B^2, \\ |C_h''(q_k)\bar{b}_k + C_h'(q_k)\bar{\alpha}_k| \leq A_{Bh}, \\ |\hat{E}_c(\bar{\beta}_k)| \leq E_B, \\ \bar{b}_k > 0, k = 1, \dots, K, \\ \bar{b}_K = \dot{q}_f^2, \bar{\alpha}_K = \ddot{q}_f. \end{cases} \quad (33)$$

Problem (33) can be solved with mature method, such as SQP. Likewise, problem (14) is also discretized and solved in the same way. Please note that problem (14) must be solved firstly.

4.4 Feedrate re-planning

Although the feedrate profile has been planned in the last subsection, it is a function of parameter p . In this subsection, the feedrate profile is re-planned, considering the limited jerk. Consequently, the feedrate profile becomes a function of time, which is convenient for interpolation.

Before re-scheduling feed, sharp corner points are searched firstly. Sharp corner points consist of minimum speed points, maximum speed points, the starting point, and the terminal point on the initial feedrate profile. The arc length between each two adjacent sharp corner points, i.e., $L_{ble}^i (i = 0, \dots, D)$, is easily computed by (6). Note that inverse normalization needs to be performed. For any two adjacent points, the feed at each point and the arc length between them are known. Given tangent jerk bound J_{tb} , the jerk-limited feedrate can be re-planned. However, these sharp corner points may be involved in “ripple effect” [24, 35]. It means that the arc length between two adjacent sharp points can not match the speeds at the two points based on bell shape feedrate profile. We deal with the “ripple effect” with following backward planning algorithm.

Algorithm 2 Backward planning method.

Require: Let v_s and v_e be the speeds at two end-points of the tool path respectively. $\mathbf{V} = [v_0, v_1, \dots, v_D, v_{D+1}]^T$ denotes the speed vector at sharp corner points, where $v_0 = v_s$ and $v_e = v_{D+1}$. $\mathbf{SC} = [0, p_1, \dots, p_D, 1]^T$ is the corresponding parameter vector.

Ensure: The attainable speed vector within arc length.

- 1: Compute backward planning distance L_{ble}^i ($i = 0, \dots, D$), which is the arc length between two adjacent sharp points.
- 2: **for** $l = D + 1; l > 0; l --$ **do**
- 3: $v_L \leftarrow \min(v_l, v_{l-1}), v_H \leftarrow \max(v_l, v_{l-1});$
- 4: $i = l - 1;$
- 5: $L_i^{bf} = (v_L + v_H) \sqrt{\frac{v_H - v_L}{J_{tb}}};$
- 6: **if** $L_i^{bf} > L_i^{ble}$ **then**
- 7: Define $v_L \leq v^{temp} \leq v_H$.
- 8: **if** $v_{l-1} > v_l$ **then**
- 9: Search the appropriate v^{temp} satisfying $(v_L + v^{temp}) \sqrt{\frac{v^{temp} - v_L}{J_{tb}}} \leq L_i^{ble}$ with bisection method;
- 10: **end if**
- 11: **if** $v_{l-1} < v_l$ **then**
- 12: Search the appropriate v^{temp} satisfying $(v_H + v^{temp}) \sqrt{\frac{v_H - v^{temp}}{J_{tb}}} \leq L_i^{ble}$ with bisection method;
- 13: **end if**
- 14: $v_{l-1} \leftarrow v^{temp};$
- 15: **end if**
- 16: **end for**

After finishing backward algorithm, the “ripple effect” is overcome. Then the forward planning can be implemented. In terms of different distance between sharp corner points, there are six types of feedrate profiles. For any two adjacent sharp corner points, the distance between them has been obtained, i.e., L_{ble}^i . Given planned feedrates v_i and v_{i+1} , tangent acceleration A_{tb} , and command feedrate F_B , the forward planning types are determined according to Fig. 4.

Each arc whose length equals to L_{ble}^i is considered as a single acceleration or deceleration (Acc/Dec) unit. However, there are $D + 1$ Acc/Dec units. Which type is chosen for each Acc/Dec unit depends on three critical distances. L_{c1} is used to determine whether the arc length is long enough to allow the cutter to reach the command feedrate. It is defined as

$$L_{c1} = (v_i + F_B) \left(T_{c1} + \frac{T_{c2}}{2} \right) + (v_{i+1} + F_B) \left(T_{c5} + \frac{T_{c6}}{2} \right), \tag{34}$$

where T_{c1} and T_{c2} are calculated with

$$\begin{cases} \text{if } F_B - v_i > J_{tb}T^2, \\ T_{c1} = T, T_{c2} = \frac{F_B - v_i}{J_{tb}T} - T; \\ \text{otherwise,} \\ T_{c1} = \sqrt{\frac{F_B - v_i}{J_{tb}}}, T_{c2} = 0. \end{cases} \tag{35}$$

Likewise, T_{c5} and T_{c6} are calculated with

$$\begin{cases} \text{if } F_B - v_{i+1} > J_{tb}T^2, \\ T_{c5} = T, T_{c6} = \frac{F_B - v_{i+1}}{J_{tb}T} - T; \\ \text{otherwise,} \\ T_{c5} = \sqrt{\frac{F_B - v_{i+1}}{J_{tb}}}, T_{c6} = 0; \end{cases} \tag{36}$$

where $T = \frac{A_{tb}}{J_{tb}}$. L_{c2} is used to determine whether a peak speed is needed to be searched or not. Define $v_{high} = \max(v_i, v_{i+1})$, $v_{low} = \min(v_i, v_{i+1})$, and $\Delta v = v_{high} - v_{low}$. Then, L_{c2} is written as

$$L_{c2} = \begin{cases} (v_i + v_{i+1})(T + \Delta v/A_{tb})/2, & \Delta v > J_{tb}T^2; \\ (v_i + v_{i+1})\sqrt{\Delta v/A_{tb}}, & \Delta v \leq J_{tb}T^2. \end{cases} \tag{37}$$

L_{c3} can distinguish whether the acceleration profile is trapezoidal or triangular, and it is written as

$$L_{c3} = 2v_{low}T + J_{tb}T^3. \tag{38}$$

When one type of feedrate profile is selected, it will be used for real-time interpolation. Hence, the time parameters of jerk-limited feed profile should be computed. This kind of feed is also known as bell shape profile. A complete profile includes seven phases, and its time parameters are shown in Fig. 5. Some other profile may contain fewer than seven phases. For example, type 3 only has four phases whose parameters satisfy $T_5 = T_6 = T_7 = 0$. The detail solutions of these parameters of each type are listed in Table 1. It is worth mentioning that the peak speed v_p in type 2 is obtained with following equation,

$$v_p = \max(v_{pm}^j). \tag{39}$$

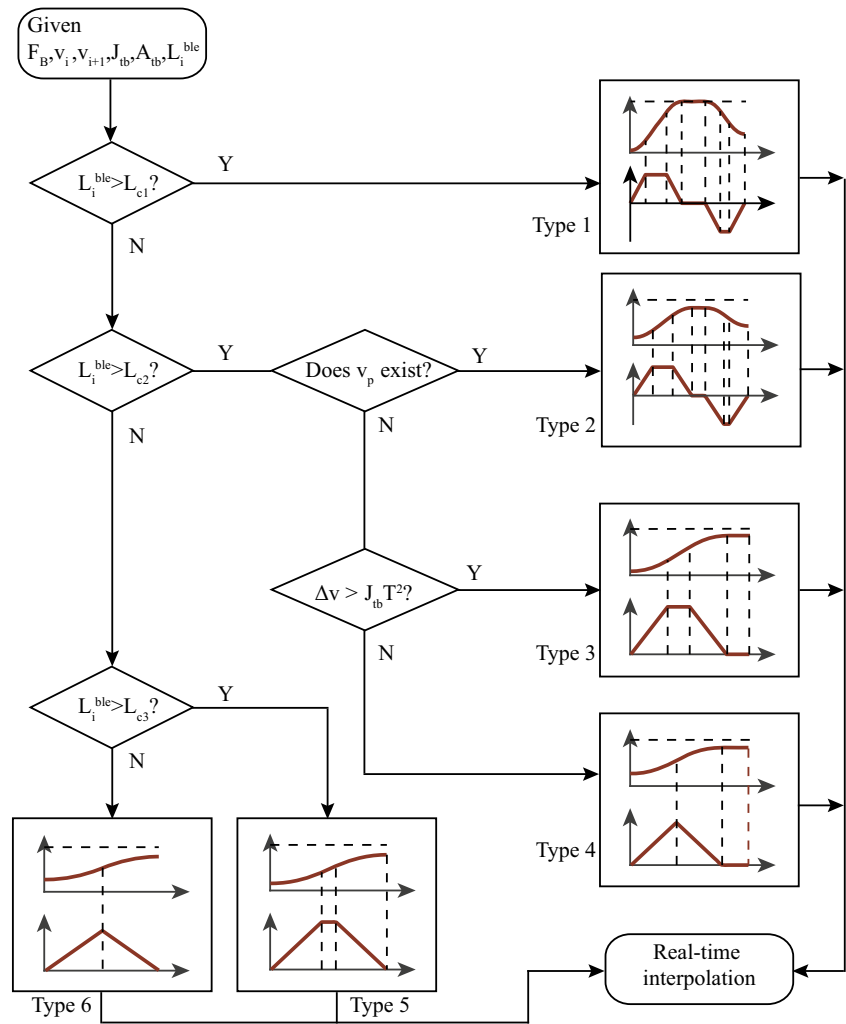
where $v_{pm}^j = v_{high} + j \cdot \frac{v_b - v_{high}}{\xi}$, $j = 1, 2, \dots, \xi$, $\xi \in N^+$, satisfying $L_{c2} < L_{acc} + L_{dec} < L_i^{ble}$. The maximal value of v_{pm}^j is selected with bisection search method. To avoid v_p being too close to v_{high} , ξ can be chosen as a large integer. That is why v_p may not exist.

4.5 Interpolation based on parameter correction B-spline

The first-order approximation interpolation based on Taylor expansion is expressed as

$$q_{i+1} = q_i + \frac{v_i T_s}{\|C'(q)\|} + H.O.T, \tag{40}$$

Fig. 4 The flowchart of types of forward planning determination



where T_s is the sample period, and $H.O.T$ is the truncation error which is neglected in practice. Since $v_i T_s$ is very small, it is considered as the arc length increment in i th sample period. Then (40) is rewritten as

$$\Delta q = q_{i+1} - q_i \approx \frac{v_i T_s}{\|C'(q)\|} = f(v_i T_s), \tag{41}$$

From (41), the more accurate the relationship between arc length and parameter, and the smaller the feedrate fluctuation and contour error are [36].

Theoretically, parameter correction b-spline can guarantee high accuracy relationship between normalized arc length and parameter. Define L_i be the whole displacement of tool tip before i th sample period. Then, $L_{i+1} = L_i + v_i T_s$. After normalization of L_{i+1} , substitute it into (4), and the corresponding parameter is obtained.

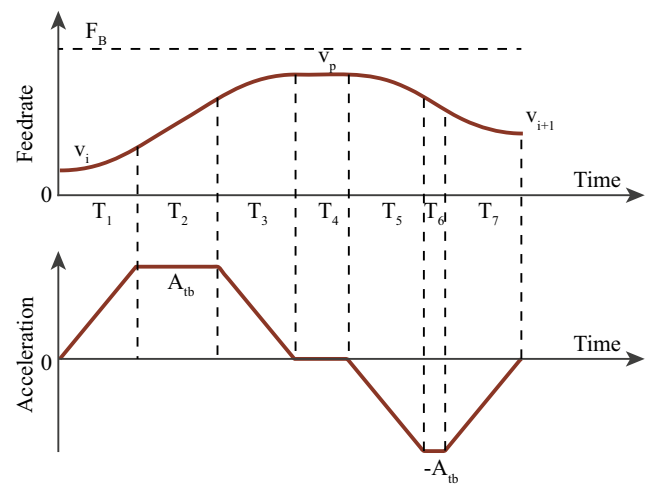


Fig. 5 Time parameters of type 2

Table 1 Time parameters for types 1 – 6

Types 1, 5, 6	See literature [37].
Type 2	If $v_p - v_i > J_{tb}T^2$, $T_1 = T_3 = T$, $T_2 = \frac{v_p - v_i}{J_{tb}T} - T$, and $L_{acc} = 0.5J_{tb}T_1T_2^2 + (v_i + 1.5J_{tb}T_1^2)T_2 + 2v_iT_1 + J_{tb}T_1^3$; otherwise $T_1 = T_3 = \sqrt{\frac{v_p - v_i}{J_{tb}}}$, $T_2=0$, and $L_{acc} = J_{tb}T^3 + 2v_iT$. If $v_p - v_{i+1} > J_{tb}T^2$, $T_5 = T_7 = T$, $T_6 = \frac{v_p - v_{i+1}}{J_{tb}T} - T$, and $L_{dec} = 0.5J_{tb}T_5T_6^2 + (v_{i+1} + 1.5J_{tb}T_5^2)T_6 + 2v_{i+1}T_5 + J_{tb}T_5^3$; otherwise $T_5 = T_7 = \sqrt{\frac{v_p - v_{i+1}}{J_{tb}}}$, $T_6=0$, and $L_{dec} = J_{tb}T^3 + 2v_{i+1}T$. $A_{m1} = J_{tb}T_1$, $A_{m2} = J_{tb}T_5$, and $T_4 = \frac{L_i^{ble} - L_{acc} - L_{dec}}{v_p}$.
Type 3	If $\Delta v \Rightarrow J_{tb}T^2$, let $E_1 = 0.5J_{tb}T$, $E_2 = v_{low} + 1.5J_{tb}T^2$, $E_3 = 2v_{low}T + J_{tb}T^3 - L_{c2}$. Then $T_1 = T_3 = T$, and $T_2 = \frac{(-E_2 + \sqrt{E_2^2 - 4E_1E_3})}{2E_1}$. So $A_{m1} = A_b$, $A_{m2} = 0$, $T_5 = 0$, $T_6 = 0$, and $T_7 = 0$. Finally, $T_4 = \frac{L_i^{ble} - L_{c2}}{v_{high}}$.
Type 4	If $\Delta v \leq J_{tb}T^2$, solve $J_{tb}t^3 + 2v_{low}t - L_{c2} = 0$. Then $T_1 = T_3 = t$, $T_2 = 0$, $A_{m1} = J_{tb}T_1$, $A_{m2} = 0$, $T_5 = T_6 = T_7 = 0$, and $T_4 = \frac{L_i^{ble} - L_{c2}}{v_{high}}$.

5 Numerical examples

To verify effectiveness of the proposed method, two benchmark tool paths [30, 33] are adopted to validate it. In the two cases, the constraint bounds and some parameters adopted are listed in Table 2. In addition, the servo models of x -axis and y -axis come from literature [38].

5.1 Example 1: A fan tool path

The test case in example 1 is a fan curve shown in Fig. 6a. The blue circles on the curve indicate all the sharp corner points (SCP) including the start point.

Apparently, some of sharp corner points are so close to each other. Therefore, the arc lengths between them are not long enough for the jerk-limited Acc/Dec process. By performing the backward planning (BPL), this kind of problems is alleviated. To clearly illustrate the effort of the backward planning, feedrate profiles are shown in Fig. 6c. The upper half of Fig. 6c is the feedrate before backward planning, i.e., the planning result of Section 4.3. The circled positions represent the speeds at sharp corner points before BPL. The bottom half demonstrates the feedrate profile after backward planning and forward planning. Note that the tangent jerk constraint has been imposed on the re-planned feedrate, and the profile has been smoothed. The sharp corner points are marked by inverted triangles, which have different tangent feedrate from those in the upper half figure.

During forward planning, the arc lengths between sharp corner points are needed to be computed. Different with the method based on Simpson's rule [35], (6) is used to obtain the arc length corresponding to given parameter. It is the inverse function of (4). The top sub-graph of Fig. 6b exhibits the mapping function curve after implementing refining Algorithm 1. The bottom sub-graph shows that the relative error has been reduced below predefined value. The middle

sub-graph indicates the interpolation relative error before refining. As stated in Algorithm 1, the relative error can be calculated with following formula,

$$RSE = |\tau_g^0 - \tau_g^t| / \tau_g^t. \quad (42)$$

The upper half of Fig. 6d shows the profile of acceleration, whereas the jerk profile is drawn in the bottom half. With limited jerk, the feedrate become smoother, as shown in Fig. 6c. When motion between two adjacent sharp corner points, the displacement of a tool tip must fall within the arc length range. To satisfy this requirement, the backward planning makes the feedrate at sharp corner points not too fast or too slow, but close to a certain intermediate value. To some extent, the cycle time is reduced. On the other hand, the contour error may become large at some sharp corners. The locations marked with diamonds on the profile in Fig. 6e are just such cases. But the overall contour error profile is not affected, because the contour

Table 2 The values of constraint bounds, predefined errors, and constants

Command feedrate F_B	100mm/s
Axial acceleration bound A_{Bh} , $h \in \{x, y\}$	200mm ² /s
Axial jerk bound J_{Bh} , $h \in \{x, y\}$	5000mm ³ /s
Contour error bound E_B	0.05mm
Tangent acceleration bound A_{tb}	200mm ² /s
Tangent jerk bound J_{tb}	5000mm ³ /s
Predefined arc length tolerance error ε_1	1×10^{-10} mm
The allowable interpolation error ε_2	1×10^{-6} (Dimensionless quantity)
The positive integer ξ for choosing peak speed v_p	50 (Dimensionless quantity)
The maximum number of iterations $Maxiter$	5 iterations

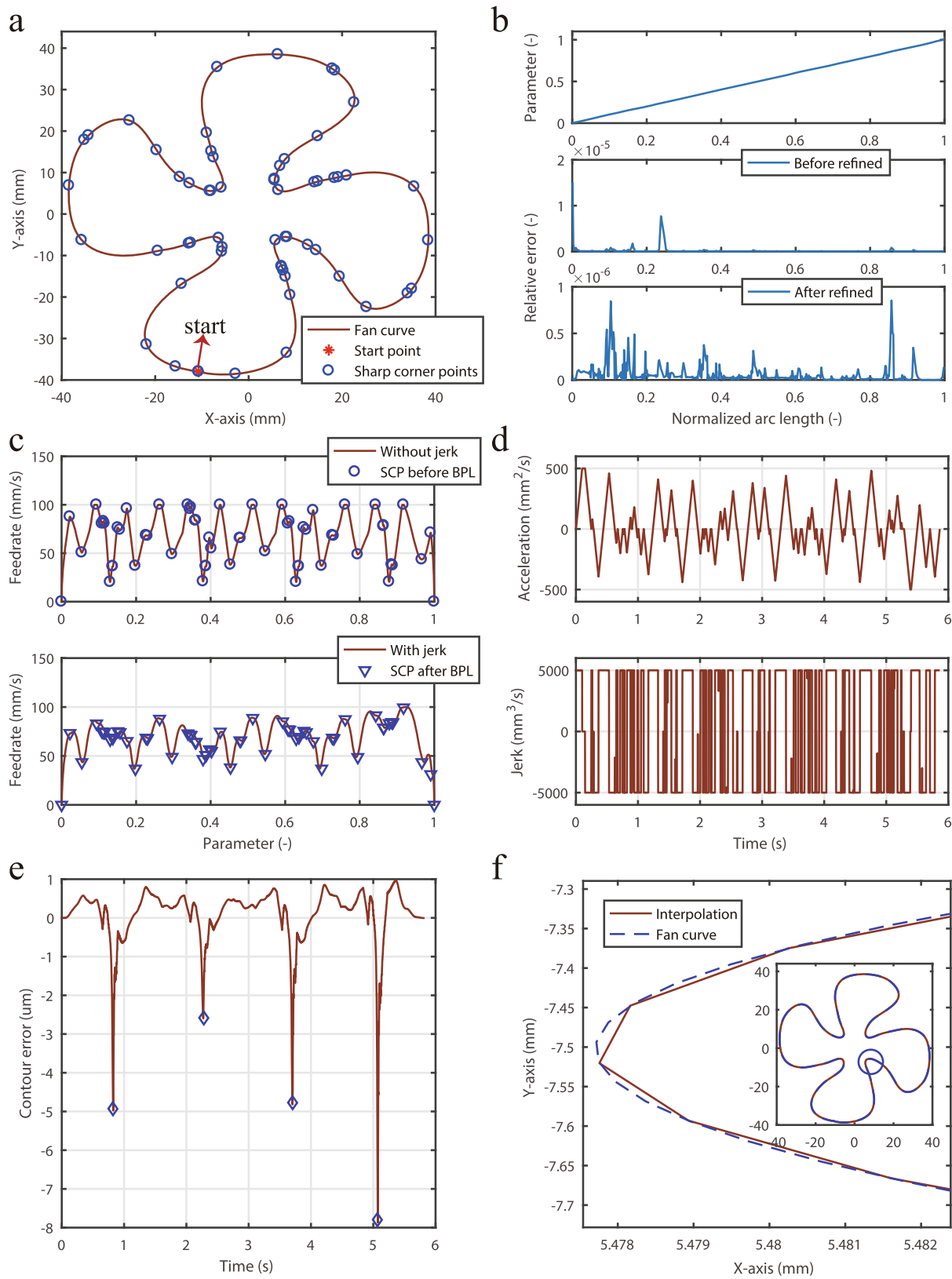
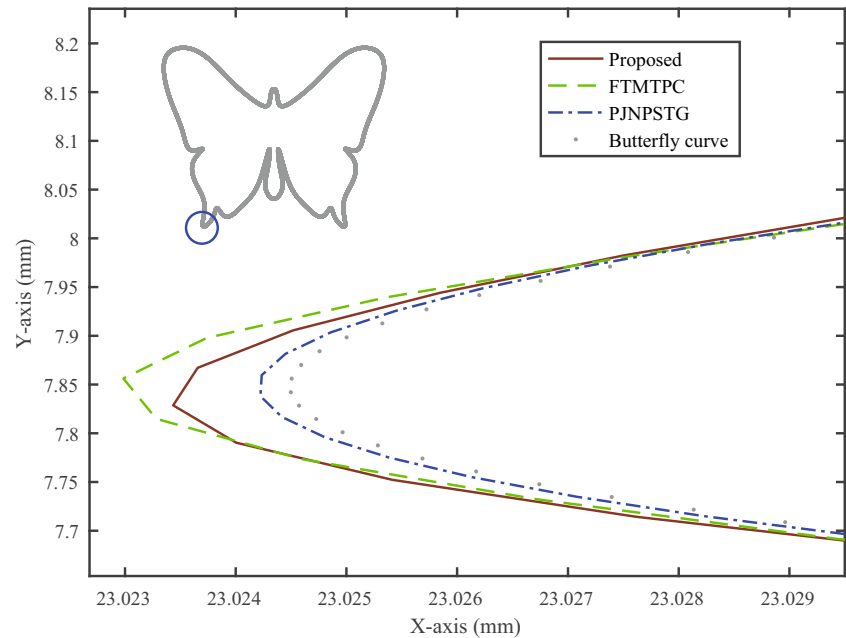


Fig. 6 The IJLMTPC results for a fan tool path: **a** the fan pattern; **b** the functional curve of the parameter correction b-spline; **c** the tangent feedrate profiles; **d** the profiles of tangent acceleration and tangent jerk; **e** the estimated contour error; **f** partial enlarged view of interpolation result of the fan path

Fig. 7 The butterfly pattern

errors at most interpolation points are very small. The partial enlarged view shown in Fig. 6f also proves this.

5.2 Example 2: A butterfly tool path

Another tested tool path is a butterfly pattern as shown in Fig. 7. The coordinates of reference points from interpolation methods are sent to servo systems in this example. And the actual coordinates generated by servo system are used to estimate the actual contour errors. The Newton Raphson iteration method [39] is utilized to calculate the actual contour errors offline.

For comparison, two other methods are adopted, except our proposed algorithm. One method plans the feedrate with NPSTG, and generates the reference coordinates with the first-order approximation interpolation based on Taylor series expansion. Since NPSTG is solved based on pseudo jerk, this method is named PJNPSTG for short. The other method also adopts the first order approximation interpolation, but plans the feedrate with the same way as the proposed, i.e., the IJLMTPC-based method. To distinguish it with the proposed scheme, this method is named FTMTPC. Detail explanations are listed in Table 3.

As shown in Fig. 8, the proposed method has the shortest cycle time, i.e., 6.426 s. The other two methods both finish machining the tool path with longer time, i.e., 7.123 s for FCMTPC and 7.912 s for PJNPSTG. In terms of cycle time, our proposed scheme has great advantages, comparing to the other two methods. Nevertheless, our proposed method leads to larger contour errors than the PJNPSTG, but smaller

than the FTMTPC (Fig. 9). In terms of contour errors, detail performances of three methods are listed in Table 4. Actually, the contour error based on our proposed method can be further reduced by imposing smaller E_B on (28). As a result, the cycle time will become longer. Therefore, one should make a trade-off between efficiency and accuracy, depending on actual situation.

6 Conclusion

To merge the contour error reduction and feedrate optimization into one procedure, we propose an integrated jerk-limited method of minimal time trajectory planning with confined contour error (IJLMTPC). Compared to the NPSTG problem, our method reduce the solution difficulty by excluding axial jerk constraints. Through adding

Table 3 Three methods for machining the butterfly pattern

Method	Planning	Interpolation
Proposed	IJLMTPC	Parameter correction b-spline
PJNPSTG	NPSTG (PMTPP+PPSTG)	First-order approximation interpolation
FTMTPC	IJLMTPC	First-order approximation interpolation

Fig. 8 The feed profiles of three interpolation methods for the butterfly tool path

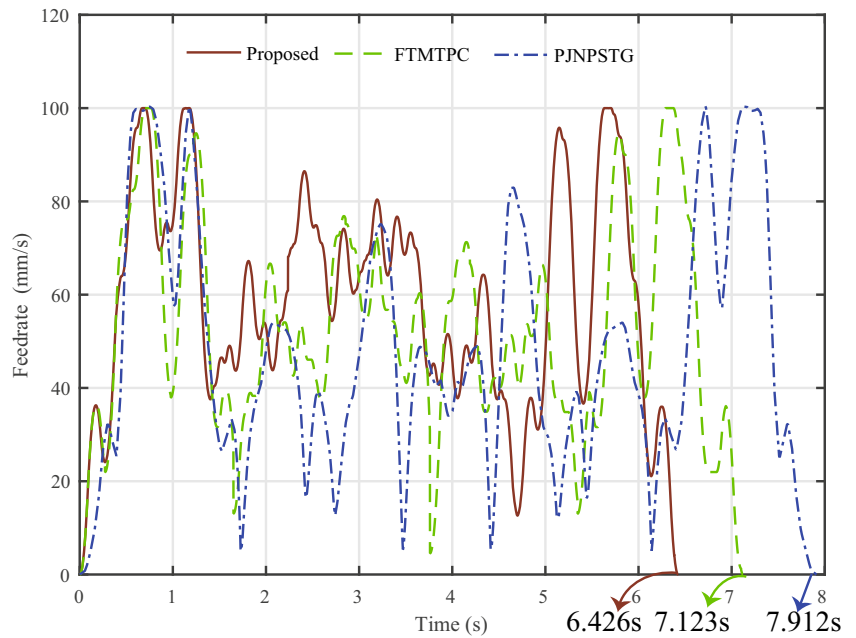
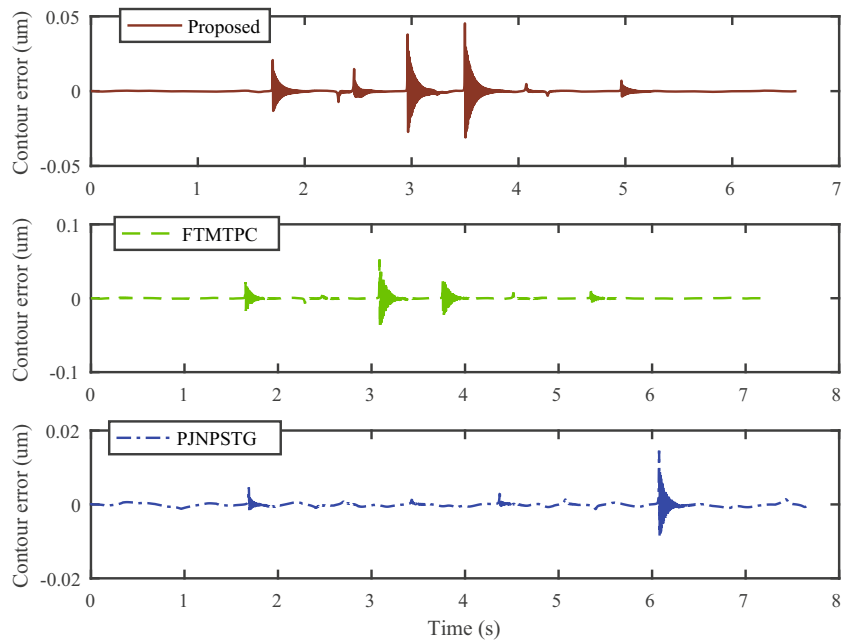


Fig. 9 The estimated contour error profiles of three interpolation methods for the butterfly pattern



contour error constraint, servo errors are pre-compensated. Furthermore, the feedrate profile on the parameter domain solved by the optimal control, is transformed to that on time domain. Our method not only utilizes the advantage of the optimal control, but also makes the resulting time optimal feedrate convenient for real-time interpolation. This will make optimal control-based trajectory planning method

Table 4 Performance comparison of three methods in terms of contour errors

Method	Max absolute contour error (μm)	Mean contour error (μm)
Proposed	45	0.83
PJNPSTG	15	0.38
FTMTPC	52	0.96

for CNC machining more practical than ever. Meanwhile, for realizing real-time interpolation, parameter correction b-spline is introduced. It reduces the fluctuation of feedrate and facilitates re-planning.

Funding This work is supported by the National Natural Science Foundation of China under Grant 61573378.

Availability of data and material The manuscript has no associated data and materials.

Declarations

Conflict of interest The authors declare no competing interests.

References

- Song D, Zhong Y, Ma JW (2020) Look-ahead-window-based interval adaptive feedrate scheduling for long five-axis spline toolpaths under axial drive constraints. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 234(13):1656–1670
- Tajima S, Sencer B, Shamoto E (2018) Accurate interpolation of machining tool-paths based on FIR filtering. *Precis Eng* 52:332–344
- Lin M, Lin W, Lin Y, Lee C, Lee J (2019) Smooth Feedrate Planning and Trajectory Generation for Five-axis Tool Path. In: 2019 IEEE 15th International Conference on Control and Automation (ICCA), Edinburgh, UK, vol 2019, pp 405–410
- Yang J, Yuen A (2017) An analytical local corner smoothing algorithm for five-axis CNC machining. *Int J Mach Tools Manuf* 123:22–35
- Dong J, Stori JA (2006) A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization. *J Dynam Syst Measure Cont* 128(2):379–390
- Bharathi A, Dong J (2016) Feedrate optimization for smooth minimum-time trajectory generation with higher order constraints. *Int J Adv Manufact Technol* 82(5):1029–1040
- Sencer B, Dumanli A, Yamada Y (2018) Spline interpolation with optimal frequency spectrum for vibration avoidance. *CIRP Ann* 67(1):377–380
- Farouki RT, Pelosi F, Sampoli ML (2019) Optimization of corner blending curves. *Comput Aided Des* 117:102739
- Yang Y, Liao L, Yang H, Li S (2020) An optimal control strategy for multi-UAVs target tracking and cooperative competition. *IEEE/CAA J Automatica Sinica* 99:1–15
- Elbanhawi M, Simic M (2014) Sampling-Based Robot motion planning: a review. *IEEE Access* 2(1):56–77
- Tang X, Chen J (2016) Direct trajectory optimization and costate estimation of infinite-horizon optimal control problems using collocation at the flipped legendre-gauss-radau points. *IEEE/CAA J Automatica Sinica* 3(2):174–183
- Ji C, Kong M, Li R (2019) Time-Energy Optimal trajectory planning for variable stiffness actuated robot. *IEEE Access* 7:14366–14377
- Ye P, Zhang Y, Xiao J, Zhao M, Zhang H (2019) A novel feedrate planning and interpolating method for parametric toolpath in Frenet-Serret frame. *Int J Adv Manufact Technol* 101(5-8):1915–1925
- Erkorkmaz K, Chen QG, Zhao MY, Beudaert X, Gao XS (2017) Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Ann* 66(1):393–396
- Zhang Q, Li SR, Guo JX, Gao XS (2016) Time-optimal path tracking for robots under dynamics constraints based on convex optimization. *Robotica* 34(9):2116–2139
- Zhang Q, Li SR, Guo JX (2014) Minimum time trajectory optimization of CNC machining with tracking error constraints. *Abstract Appl Anal* 2014(5):835098
- Saleem O, Rizwan M (2020) ILC-Adapted parameter optimization of cross-coupled single-input fuzzy tracking controllers for an X-Y positioning table. *J Chin Inst Eng* 43(6):519–531
- El Khalick MA, Uchiyama N (2011) Discrete-time model predictive contouring control for biaxial feed drive systems and experimental verification. *Mechatronics* 21(6):918–926
- Kornmaneesang W, Chen SL (2020) MPC-Based robust contouring control for a robotic machining system. *Asian J Control* 2020:1–13
- El Khalick MA, Uchiyama N, Sano S (2013) Sliding mode contouring control design using nonlinear sliding surface for three-dimensional machining. *Int J Mach Tools Manuf* 65:8–14
- Kim H, Okwudire CE (2020) Simultaneous servo error pre-compensation and feedrate optimization with tolerance constraints using linear programming. *Int J Adv Manufact Technol* 109(2):809–821
- Du D, Liu Y, Guo X, Yamazaki K, Fujishima M (2010) An accurate adaptive NURBS curve interpolator with Real-Time flexible Acceleration/Deceleration control. *Robot Comput Integr Manuf* 26(4):273–281
- Tsai MC, Cheng CW (2003) A Real-Time predictor corrector interpolator for CNC machining. *J Manufact Sci Eng-Trans ASME* 125(3):449–460
- Jahanpour J, Alizadeh MR (2015) A novel acc-jerk-limited nurbs interpolation enhanced with an optimized s-shaped quintic feedrate scheduling scheme. *Int J Adv Manufact Technol* 77(9-12):1889–1905
- Liu H, Liu Q, Sun P, Liu Q, Yuan S (2017) A polynomial equation-based interpolation method of NURBS tool path with minimal feed fluctuation for high-quality machining. *Int J Adv Manufact Technol* 90(9):2751–2759
- Lu L, Zhang L, Gu Y, Zhao J (2017) A parametric interpolation method with minimal feedrate fluctuation by nonuniform rational basis spline. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 231(18):3301–3317
- Heng M, Erkorkmaz K (2010) Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability. *Int J Mach Tools Manuf* 50(3):281–293
- Lei WT, Sung MP, Lin LY, Huang JJ (2007) Fast real-time NURBS path interpolation for CNC machine tools. *Int J Mach Tools Manuf* 47(10):1530–1541
- Zhang Q, Li SR, Guo JX (2012) Smooth time-optimal tool trajectory generation for CNC manufacturing systems. *J Manuf Syst* 31(3):280–287
- Zhang Q, Li SR (2013) Efficient computation of smooth minimum time trajectory for CNC machining. *Int J Adv Manufact Technol* 68(1-4):683–692
- Zhang Q, Li S (2011) Minimum time feed-rate optimization along predefined toolpath with acceleration constraints on each axis. *Proceedings of the 30th Chinese Control Conference*, (2011):5462–5467
- Zhao K, Kang ZJ, Guo XB (2020) Smooth trajectory generation for predefined path with pseudo spectral method. *IEEE Access* 8:158735–158744

33. Zhang Q, Gao XS (2016) Practical feedrate optimization for planar high precision contouring. In: 2016 IEEE International Conference on Information and Automation (ICIA), Ningbo, China, vol 2016, pp 1872–1877
34. Koren Y, Lo Ch-Ch (1991) Variable-Gain Cross-Coupling Controller for Contouring. *CIRP Annals* 40(1):371–374
35. Zhao K, Li SR, Kang ZJ (2019) Smooth minimum time trajectory planning with minimal feed fluctuation. *Int J Adv Manufact Technol* 105(9-12):1099–1111
36. Wu JC, Zhou HC, Tang XQ, Chen JH (2012) A NURBS interpolation algorithm with continuous feedrate. *Int J Adv Manufact Technol* 59(5-8):623–632
37. Dong J, Wang T, Li B, Ding Y (2014) Smooth feedrate planning for continuous short line tool path with contour error constraint. *Int J Mach Tools Manuf* 76:1–12
38. Tsai MS, Nien HW, Yau HT (2008) Development of an integrated look-ahead dynamics-based nurbs interpolator for high precision machinery. *Comput Aided Des* 40(5):554–566
39. Yang M, Yang JX, Ding H (2018) A high accuracy on-line estimation algorithm of five-axis contouring errors based on three-point arc approximation. *Int J Mach Tools Manuf* 130-131:73–84

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.