



OPEN

Dynamic look-ahead feedrate scheduling method based on sliding mode velocity control

Liuquan Wang^{1,2}, Qiang Liu^{1,2}✉, Pengpeng Sun³, Shisheng Lv^{1,4}, Ruijie Yang^{1,5} & Zhiqi Yang^{1,5}

In the feedrate scheduling of complex curve direct interpolation, dynamic constraints such as axis acceleration and jerk are related to the actual state of the tool. Most existing methods convert dynamic constraints to velocity constraints at sampling points. However, it cannot guarantee the dynamic constraints are satisfied between sampling points. Addressing the issue, this paper proposes a dynamic look-ahead feedrate scheduling method based on sliding mode velocity control, which generates the motion command considering dynamic constraints in every interpolation cycle. To dynamically generate commands based on the current tool state, the acceleration and deceleration method based on sliding mode velocity control has been proposed, which can control tool state to transition to the command state with any initial state. To ensure sufficient distance for acceleration and deceleration, this paper uses braking distance to dynamically estimate the look-ahead distance. Then the minimum value within the look-ahead interval is selected as the command velocity for this scheduling cycle and the actual motion command is determined based on the dynamic constraints of each axis. Simulation and experiment results prove that compared with the existing method, this method effectively reduces the overshoot of dynamic constraints without significantly increasing the machining time. The analysis of real-time computation time has demonstrated the potential of the method proposed in this paper for real-time applications.

Keywords Feedrate scheduling, Dynamic look-ahead, Sliding mode control, Tracking error

As the standard form for curves in the STEP standard, Non-Uniform Rational B-Splines (NURBS) curves are widely used for the geometric expression of complex parts. Direct interpolation for NURBS curves could avoid discretizing the continuous NURBS curve to straight lines and arcs and improve machining efficiency and precision. In machining process, high-accuracy surface quality requires that tool motion satisfy constraints such as chord error, tangential acceleration, and tangential jerk¹. Additionally, surface quality is also affected by servo axis motion performance such as servo tracking bandwidth, maximum acceleration and jerk². Feedrate scheduling is an important component of NURBS direct interpolation, which can fulfill the requirement of machining accuracy and quality by adjusting the feedrate to satisfy constraints in geometry, kinematics and dynamics³. The common feedrate scheduling methods for NURBS direct interpolation can be divided into acceleration and deceleration(A/D) method and time-optimal optimization (TOO) method⁴.

The TOO method converts the feedrate scheduling to an optimal control problem and usually optimizes the velocity at sampling points to achieve maximum machining efficiency. After obtaining the time-optimized velocity at discrete sampling points, a continuous u - v curve can be fitted by spline to obtain the velocity at any position. Liu et al.⁵ used the square of the velocity as optimization variable, and considered the kinematics and dynamics constraints of the machine tool. Lu et al.⁶ employed a predictive deceleration method to predict switching points of jerk to ensure jerk constraints. Zhang et al.⁴ and Xiao et al.⁷ converted tracking error constraints into linear combinations of tangential velocity, acceleration and jerk, and have taken them into consideration in optimal control. Chen et al.⁸ linearized the contour error as a function of velocity and used the Frenet framework to determine the accurate upper limit of velocity under the contour error constraint. To avoid velocity fluctuation caused by spline fitting, Zhao et al.⁹ adopted A/D method to scheduling the feedrate between sampling points after

¹School of Mechanical Engineering and Automation, Beihang University, Beijing 100191, China. ²Jiangxi Research Institute of Beihang University, Nanchang 330096, China. ³Mechanics and Acoustics Division, National Institute of Metrology, Beijing 100029, China. ⁴Beijing Engineering Technological Research Center of High-Efficient and Green, CNC Machining Process and Equipment, Beijing 100191, China. ⁵Research and Application Center of Advanced, CNC Machining Technology, State Administration of Science, Technology and Industry for National Defense, Beijing 100191, China. ✉email: qliusmea@buaa.edu.cn

obtaining time-optimal velocity using optimal control. Yang et al.¹⁰ used particle swarm optimization to optimize the control nodes of the fitted u - v curve. Although TOO can achieve maximum machining efficiency, the computational load limits the use of TOO in real-time environments, for it hard to respond to override changes.

The A/D method generally divides the whole curve into sub-curve segments according to the velocity at sampling points. And for each sub-curve, a fixed A/D model is employed to generate the velocity profile in this sub-curve. The most common A/D model is S-shaped model¹¹ and here are 17 S-shaped velocity curve profiles according to different conditions¹². Traditional S-shaped A/D method causes jerk impact. To reduce the jerk impact during the manufacturing process, some jerk planning methods based on trigonometric function^{13,14}, sigmoid function¹⁵ and quartic function¹⁶ were proposed. However, these approaches increase the complexity of curve classification discussions. Wu et al.¹⁷ proposed a scheduling method combining quartic S-shaped and cubic S-shaped where quartic S-shaped is used to avoid jerk impact and cubic S-shaped is used to ensure machining efficiency. Ren et al.¹⁶ proposed a method of segment merging to reduce A/D stages. Sang et al.¹⁸ used morphological filters to optimize the feedrate limitation profile. Jia et al.¹⁹ proposed the concept of the velocity sensitive region, in which a uniform rate scheduling was applied to avoid exceeding constraints. Sun et al.²⁰ schedule the period of velocity stage to be an integer multiple of servo control cycle to eliminate velocity fluctuations. Currently, intelligent control algorithms in CNC systems, such as adaptive cutting force control²¹, often require changing override real-time. To meet the requirements, many scholars have developed dynamic look-ahead feedrate scheduling methods based on A/D methods. Sun et al.²² proposed an acceleration look-ahead method with \sin^2 acceleration curve to avoid frequent acceleration fluctuations during short segments. Zhang et al.²³ proposed an iterative method that detects exceeding constraint points by pre-interpolation and re-divides sub-segments at exceeding constraints points. However, the iteration time is uncontrollable. Song et al.²⁴ proposed a dynamic moving look-ahead window method which uses braking distance to estimate look-ahead window length. Sun²⁵ proposed a dynamic look-ahead method, which constructs a hyperbolic tangent function relationship between velocity and acceleration to generate commands based on tool state. However, this method cannot guarantee stable arrival at end point of curve and its stability has not been proven.

As the complexity of parts increases, in the finishing stage, machining stability is more important than efficiency, and it is required to satisfy constraints such as chord error, axis velocity, axis acceleration and axis jerk throughout the entire machining process. Only related to the geometric properties of the curve, static constraints such as chord error and axis velocity can be converted to tangential velocity constraints. However, dynamic constraints such as axis acceleration and jerk are related to the actual state of the tool. Existing methods often assume that the tool's acceleration and jerk are zero, thereby converting dynamic constraints to velocity constraints at sampling points. However, it cannot guarantee satisfying constraints between sampling points.

To address the needs for real-time override adjustment and dynamic constraints, this paper proposes a dynamic lookahead method based on sliding mode velocity control (SMVC). Compared to S-shaped A/D which constructs a relationship between jerk and time, SMVC generates commands using current tool state, which can adjust jerk more flexibly. Based on SMVC, the dynamic look-ahead feedrate scheduling considers dynamic constraints in every interpolation cycle, achieving consideration of dynamic constraints in the entire machining process. The remainder of this manuscript is organized as follows. “Principle of proposed method” details the principles of proposed scheduling method. “Sliding mode velocity control acceleration and deceleration method” introduces the sliding mode velocity control method. “Feedrate scheduling method” introduces the specific implementation of proposed scheduling method. And “Simulation” and “Experiment” present the simulation and experimental results.

Principle of proposed method

The dynamic look-ahead feedrate scheduling method is shown in Fig. 1. The feedrate scheduling method is divided into two stages: non-real-time pre-processing and real-time tasks. The non-real-time pre-processing stage realizes the sampling of NURBS to obtain the differential properties of NURBS at sampling points. Combined with the feedrate constraints model such as chord error and axial velocity, the feedrate limitation profile satisfying the constraints is obtained. Meanwhile, the relationship between feedrate and look-ahead distance is established. The above information will be saved in the buffer to support the real-time tasks.

In the real-time task stage, the look-ahead distance of current period is calculated firstly according to the actual feedrate and the look-ahead distance function. After seeking the look-ahead interval, the minimum value of feedrate limitation profile in this interval is to serve as the command feedrate $v_{cmd}(k)$ of this period. Then SMVC method calculates the desired tangential acceleration and jerk command $a_{cmd}(k)$, $J_{cmd}(k)$ using the current tool status. Subsequently, dynamic constraints are used to adjust the $a_{cmd}(k)$ and $J_{cmd}(k)$ and the actual jerk command $J_{act}(k)$ satisfying dynamic constraints is derived. If there is no solution that meets the dynamic constraints, then $J_{act}(k)$ only needs to meet the maximum tangential jerk constraint. Once $J_{act}(k)$ is obtained, the arc length increment $\Delta s(k)$ and parameter increment $\Delta u(k)$ can be calculated. According to the $\Delta u(k)$, the command position and NURBS differential properties of next period can be obtained.

Sliding mode velocity control acceleration and deceleration method

Define tool tip state $\mathbf{x} = [v_{act} \ a_{act}]^T$, where v_{act} represents the tool tip's velocity and a_{act} represents the tool tip's acceleration. The command state can be represented as $\mathbf{x}_{cmd} = [v_{cmd} \ 0]^T$, where the v_{cmd} represents the command tool tip's velocity. The A/D process is the process that tool tip's state transitions to command state \mathbf{x}_{cmd} from current state \mathbf{x} . And in this process, the velocity and acceleration should be continuous and $a_{act} \in [-A_{max}, A_{max}]$. Consider the following relationship between a_{cmd} and v_{act}

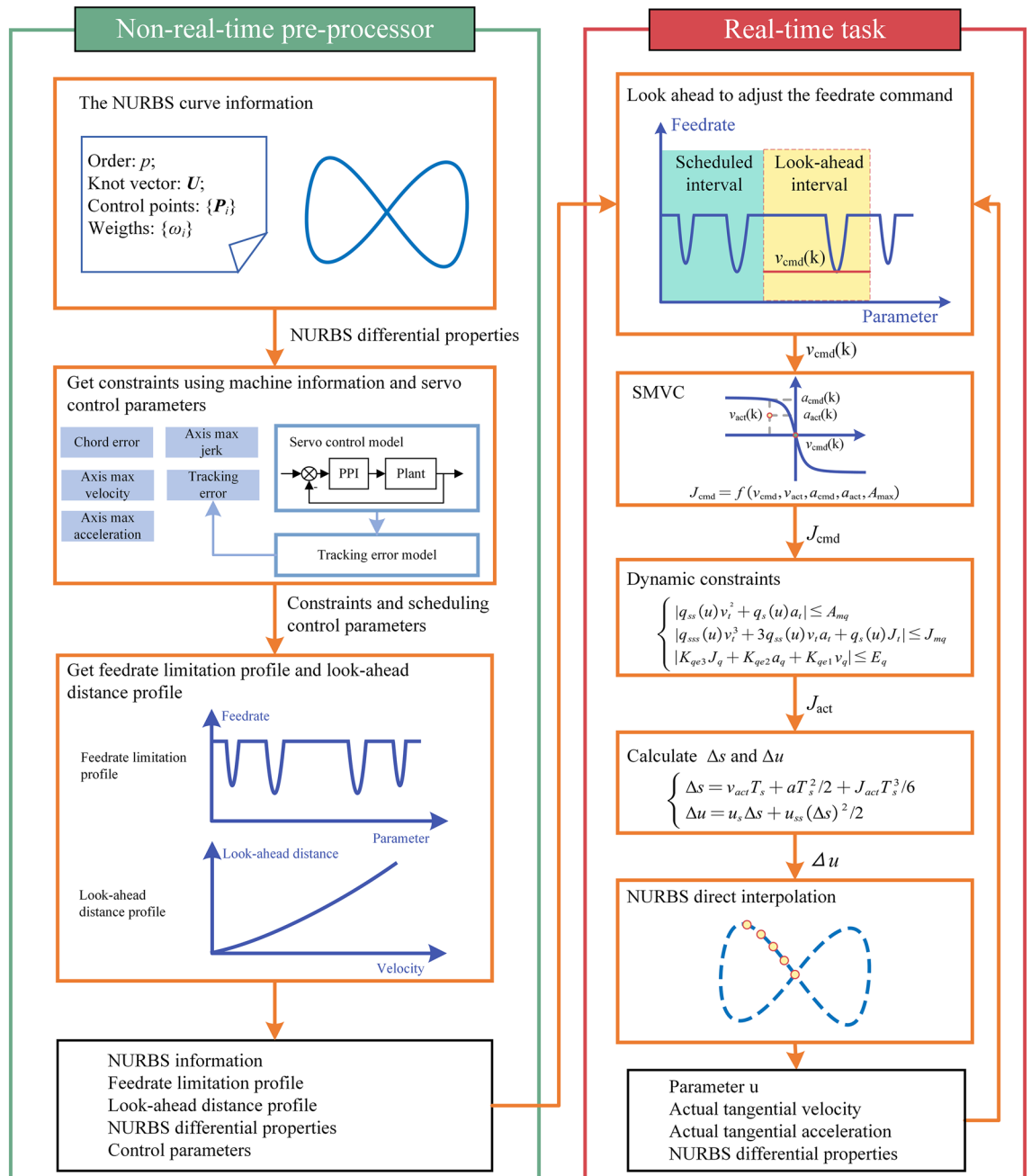


Figure 1. Schematic diagram of dynamic look-ahead feedrate scheduling method based on SMVC.

$$a_{cmd} = A_{max} \frac{K_1(v_{cmd} - v_{act})}{\sqrt{1 + (K_1(v_{cmd} - v_{act}))^2}} \tag{1}$$

where K_1 is a positive coefficient. The graph of Eq. (1) on phase plane is shown in Fig. 2.

If a_{cmd} satisfies Eq. (1), it will be continuous and in $[-A_{max}, A_{max}]$. To prove the stability of control method described by Eq. (1), let $v_e = v_{cmd} - v_{act}$ and a_e be the first-order derivative of the v_e with respect to time t , which can shift command status from $[v_{cmd} \ 0]^T$ to $[0 \ 0]^T$. Equation (1) becomes

$$a_e = \dot{v}_e = -A_{max} \frac{K_1 v_e}{\sqrt{1 + (K_1 v_e)^2}} \tag{2}$$

If $[v_e, a_e]^T$ can converge to $[0, 0]^T$, it implies that $[v_{act}, a_{act}]^T$ can converge to $[v_{cmd} \ 0]^T$. Define Lyapunov function

$$V_1 = \frac{1}{2} v_e^2 \tag{3}$$

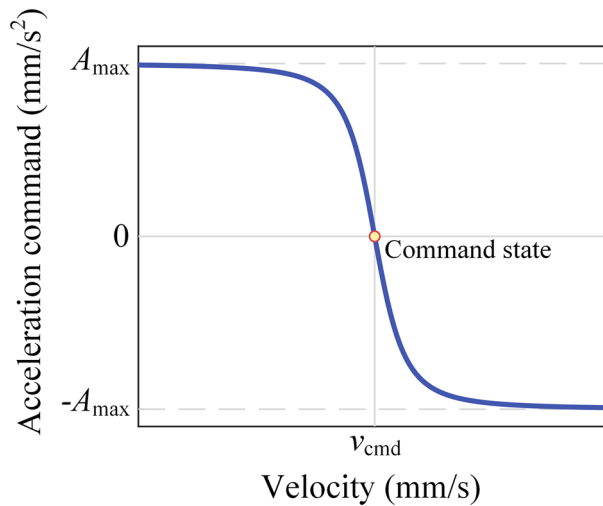


Figure 2. Graph of Eq. (1) on phase plane.

V_1 is obviously positive definite and

$$\dot{V}_1 = v_e \dot{v}_e = -A_{\max} \frac{K_1 v_e^2}{\sqrt{1+(K_1 v_e)^2}} \tag{4}$$

When $v_e \neq 0, \dot{V}_1 \neq 0$. It means that if tool tip system follows Eq. (1), it is asymptotically stable near $[v_{\text{cmd}} \ 0]^T$.

Due to the constraint on jerk, it is necessary to ensure the jerk of Eq. (1) satisfying the limit. There is

$$J_{\text{cmd}} = \frac{da_{\text{cmd}}}{dt} = \frac{da_{\text{cmd}}}{dv_{\text{act}}} \frac{dv_{\text{act}}}{dt} = \frac{da_{\text{cmd}}}{dv_{\text{act}}} a_{\text{cmd}} = -\frac{A_{\max}^2 K_1^2 (v_{\text{cmd}} - v_{\text{act}})}{(1+K_1^2 (v_{\text{cmd}} - v_{\text{act}})^2)^2} \tag{5}$$

Furthermore,

$$\frac{dJ_{\text{cmd}}}{dv_{\text{act}}} = -A_{\max}^2 K_1^2 \frac{1-3K_1^2 (v_{\text{cmd}} - v_{\text{act}})^2}{(1+K_1^2 (v_{\text{cmd}} - v_{\text{act}})^2)^3} \tag{6}$$

Letting $\frac{dJ_{\text{cmd}}}{dv_{\text{act}}} = 0$, there are $v_1 = v_{\text{cmd}} - \frac{\sqrt{3}}{3K_1}, v_2 = v_{\text{cmd}} + \frac{\sqrt{3}}{3K_1}$. In the interval $(-\infty, +\infty)$, it can be observed that

$$\begin{aligned} \max\{J_{\text{cmd}}\} &= \max\{J(+\infty), J(v_1)\} = \frac{3\sqrt{3}}{16} A_{\max}^2 K_1 \\ \min\{J_{\text{cmd}}\} &= \min\{J(-\infty), J(v_2)\} = -\frac{3\sqrt{3}}{16} A_{\max}^2 K_1 \end{aligned} \tag{7}$$

Given the maximum acceleration as J_{\max} , the K_1 should satisfy

$$K_1 \leq \frac{16}{3\sqrt{3}} \frac{J_{\max}}{A_{\max}^2} \tag{8}$$

When the tool tip state satisfies Eq. (1) and K_1 satisfies Eq. (8), the system will converge to the command state x_{cmd} . For states not satisfying Eq. (1), a sliding mode control method can be designed to make them approach the ideal state first.

Define sliding mode surface as

$$s_c = A_{\max} \frac{K_1 (v_{\text{cmd}} - v_{\text{act}})}{\sqrt{1+(K_1 (v_{\text{cmd}} - v_{\text{act}}))^2}} - a_{\text{act}} \tag{9}$$

When $s_c = 0$, the current tool state satisfies Eq. (1). When $s_c \neq 0$, it implies that the tool state is not on the sliding mode surface. By controlling s_c to tend towards 0, the tool state can be ensured to approach the sliding mode surface. Consider following equation:

$$\dot{s}_c = -\frac{A_{\max} K_1 a_{\text{act}}}{(1+K_1^2 (v_{\text{cmd}} - v_{\text{act}})^2)^{\frac{3}{2}}} - J_{\text{act}} = -K_2 s_c \tag{10}$$

where J_{act} is the jerk of tool tip and K_2 is a positive coefficient. For the system described by Eq. (10), the Lyapunov function $V_2 = s_c^2/2$ is chosen. It can easily be proven that V_2 is positive definite and \dot{V}_2 is negative definite. It implies s_c can stably converge to 0.

Increasing K_2 can improve the system's performance, but if K_2 is too large, it may cause chatter on the sliding mode surface. In this paper, K_2 is taken as $1/T_s$, where T_s is the interpolation period. From Eq. (10), it can be deduced that

$$J_{cmd} = K_2 \left(A_{max} \frac{K_1(v_{cmd} - v_{act})}{\sqrt{1 + (K_1(v_{cmd} - v_{act}))^2}} - a_{act} \right) - \frac{A_{max}K_1a_{act}}{(1 + K_1^2(v_{cmd} - v_{act})^2)^{\frac{3}{2}}} \tag{11}$$

In the actual process, the jerk of the tool tip needs to satisfy the condition $|J_{cmd}| \leq J_{max}$. It means Eq. (11) may not be satisfied. However, as long as J_{cmd} doesn't change sign, the system's convergence direction will not change. The limit of jerk only affects convergence speed and the system remains stable.

In the A/D method, it is important to achieve the command velocity without overshoot. Near $[v_{cmd} \ 0]^T$, as $v_{cmd} - v_{act}$ approaches 0, Eq. (1) and Eq. (11) can be approximated as

$$\begin{cases} a_{cmd} \approx A_{max}K_1(v_{cmd} - v_{act}) \\ J_{act} \approx (A_{max}K_1(v_{cmd} - v_{act}) - a_{act}) - A_{max}K_1a_{act} \end{cases}, \text{When } v_{act} \rightarrow v_{cmd} \tag{12}$$

Then the system described by Eq. (11) can be approximated as a linear system as shown in Fig. 3. The transfer function of the system shown in Fig. 3 is

$$G(s) = \frac{A_{max}K_1K_2}{s^2 + (A_{max}K_1 + K_2)s + A_{max}K_1K_2} \tag{13}$$

The system depicted in Eq. (13) is a typical second-order system. The natural frequency and damping ratio are

$$\begin{cases} \omega_n = \sqrt{A_{max}K_1K_2} \\ \xi = \frac{A_{max}K_1 + K_2}{2\sqrt{A_{max}K_1K_2}} \end{cases} \tag{14}$$

Using Cauchy's inequality, there is

$$\xi = \frac{A_{max}K_1 + K_2}{2\sqrt{A_{max}K_1K_2}} \geq \frac{2\sqrt{A_{max}K_1K_2}}{2\sqrt{A_{max}K_1K_2}} = 1 \tag{15}$$

equality is achieved when $A_{max}K_1 = K_2 = 0$.

Therefore, the actual system's damping ratio is greater than 1, indicating that the system is overdamped and will not overshoot during convergence. It should be noted that the system described by Eq. (13) is time-continuous. Actual computer numerical control is a discrete system, which may cause some overshoot due to the discrete step size.

Figure 4 illustrates the process of adjusting the system state to the command state using SMVC, providing three initial conditions: Case 1: Initial velocity = 0, initial acceleration = 0; Case 2: $0 < \text{initial velocity} < \text{command velocity}$, initial acceleration < 0 ; Case 3: Initial velocity $> \text{command velocity}$, initial acceleration > 0 . It can be observed that in all 3 cases, SMVC can effectively control system convergence to the command state.

Figure 5 shows the response of SMVC to override changes. It shows that SMVC could flexible response the changes of override.

When the system enters the sliding mode surface, the convergence rate is determined by Eq. (1) and related to the velocity error. With the velocity error approaching 0, the convergence rate becomes very small, which can reduce the system's efficiency. Although Eq. (8) limit the value of K_1 , for specific command velocity, the system does not pass through the jerk extrema on Eq. (1) during convergence. So, without causing overshoot, K_1 can be appropriately increased to improve the system's performance.

Due to the nonlinear characteristics of SMVC, it is difficult to obtain an analytical expression between K_1 and overshoot. Therefore, this paper uses an iterative method to solve for the optimal K_1 . Let $f(x)$ be the mapping from the parameter K_1 to the system's overshoot. The overshoot is calculated by $v_{max} - v_{cmd}$, where v_{max} is the maximum velocity of the system during the given time response process, and v_{cmd} is the command velocity. This mapping can be implemented through simulation. The process of using an iterative method to find the optimal K_1 is shown in Fig. 6.

For example, with $v_{cmd} = 50 \text{ mm/s}$, $A_{max} = 1000 \text{ mm/s}^2$, $J_{max} = 10,000 \text{ mm/s}^3$, according to Eq. (8), $K_1 = 0.0308 \text{ s/m}$ is obtained. Given tolerance of 0.01 mm/s , a more optimal parameter $K_1^* = 0.0385 \text{ s/mm}$ can be achieved. Figure 7 presents the response of SMVC under K_1 and K_1^* , while also comparing them to the response of the

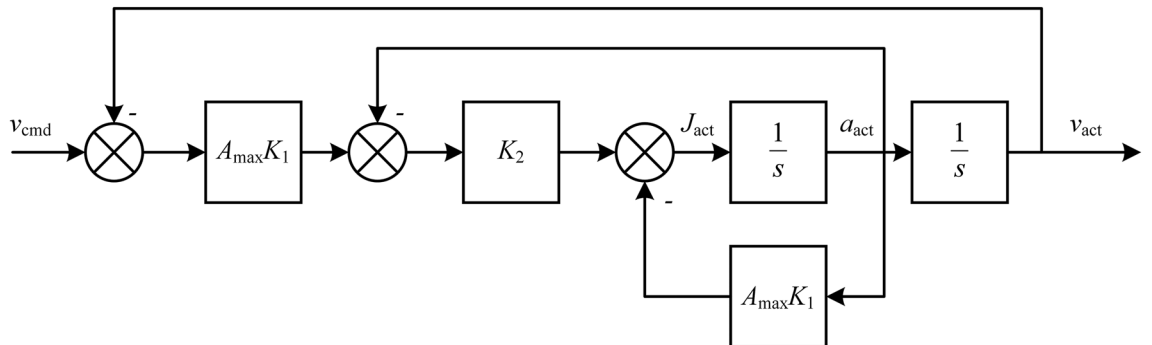


Figure 3. Approximated linear system.

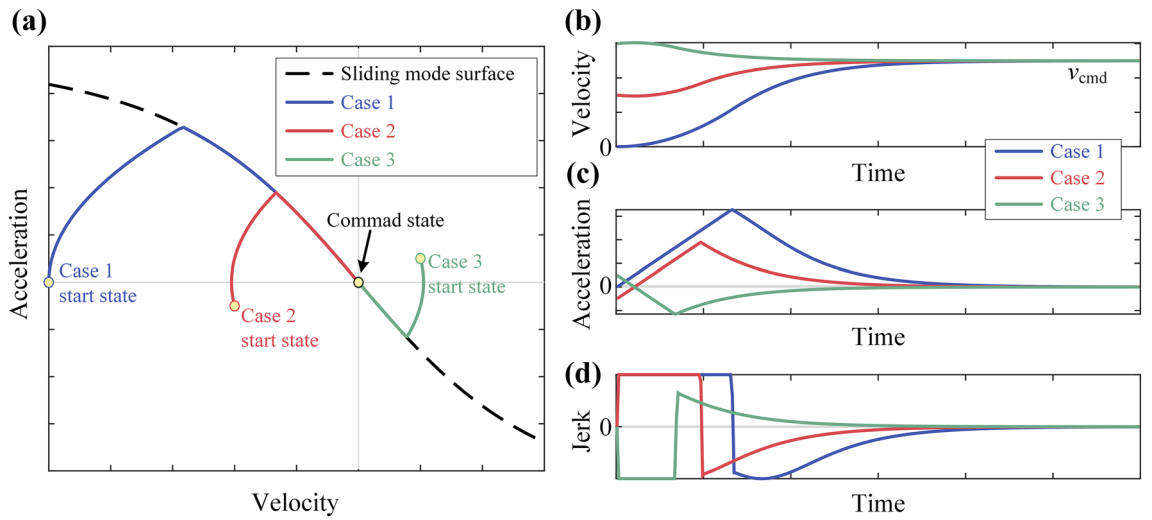


Figure 4. Sliding mode velocity control under 3 initial conditions: (a) results on phase plane;(b) velocity – time; (c) acceleration-time; (d) jerk-time.

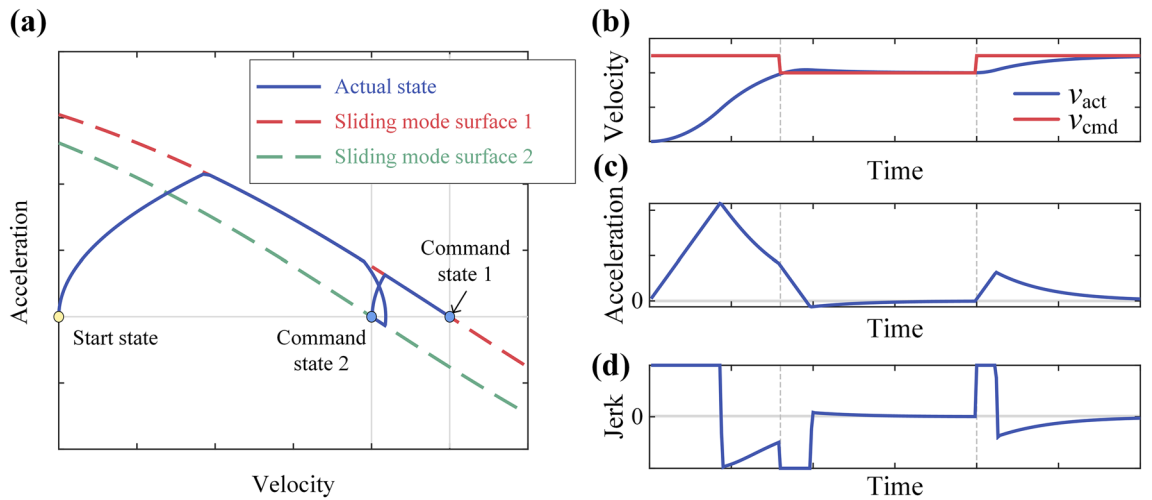


Figure 5. SMCVC response to override changes: (a) results on phase plane;(b) velocity – time; (c) acceleration-time; (d) jerk-time.

S-shaped A/D. It can be observed that the performance of the system after parameter optimization has become very close to S-shaped A/D method.

At the end of entire curve, the command state is $[0 \ 0]^T$. If tool system directly follows the $[0 \ 0]^T$, it may cause velocity to be 0 before reaching endpoint. To address this issue, this paper introduces a position-error-feedback velocity term in the endpoint command state, which changes the endpoint command state from $[0 \ 0]^T$ to $[K_3(p_{end} - p_{act}) \ 0]^T$. where p_{end} is the endpoint position, p_{act} is the current position, and K_3 is a positive coefficient. The added term ensures that the velocity of tool will not drop to 0 before reaching endpoint.

Feedrate scheduling method
NURBS description

The definition of an p -degree NURBS curve is as follows:

$$C(u) = \frac{\sum_{i=0}^n N_{i,p}(u)w_i P_i}{\sum_{i=0}^n N_{i,p}(u)w_i} \tag{16}$$

where $N_{i,p}(u)$, $i = 0, 1, \dots, n$ is the basis function of the p -degree B-spline defined on the node sequence $U = [u_0 \ u_1 \ \dots \ u_{n+p+1}]$, P_i , $i = 1, 2, \dots, n$ are the control points of the NURBS curve, w_i are the weight factors corresponding to the control points P_i . The B-spline basis function can be calculated by the de Boor-Cox recursive formulation:

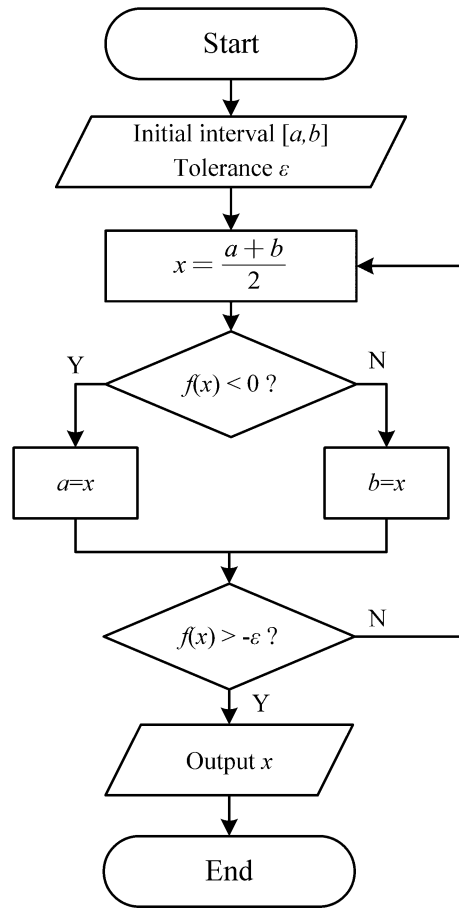


Figure 6. Iterative method for parameter optimization.

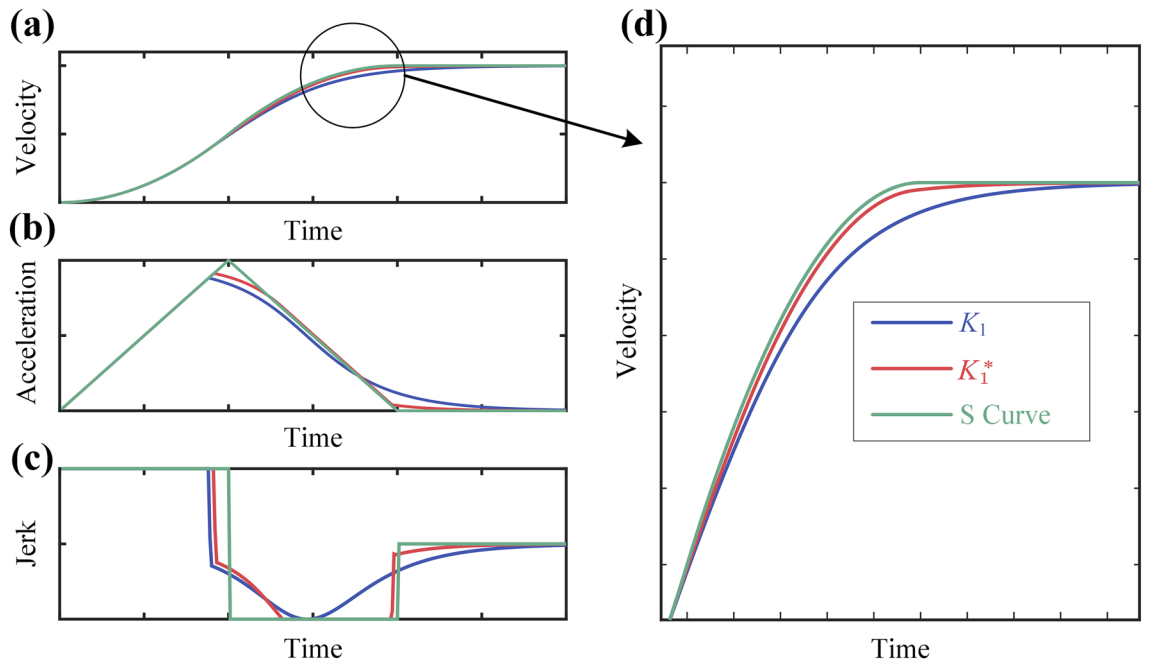


Figure 7. S-shaped response and SMVC response under K_1 , K_1^* : (a) velocity-time; (b) acceleration-time; (c) jerk-time; (d) partial enlargement of velocity-time.

$$\begin{cases} N_{i,0}(u) = \begin{cases} 0, & u_i \leq u \leq u_{i+1} \\ 1, & \text{other} \end{cases} \\ N_{i,p}(u) = \frac{u-u_i}{u_{i+p}-u_i}N_{i,p-1}(u) + \frac{u_{i+p+1}-u}{u_{i+p+1}-u_{i+1}}N_{i+1,p-1}(u) \end{cases} \quad (17)$$

For five-axis machine tools, machining curve is generally described by dual NURBS curve, which is composed of 2 NURBS curve, $C(u)$ and (u) , as defined by Eq. (16). And $C(u)$ represents the tool tip position and $O(u) = \frac{H(u)-C(u)}{\|H(u)-C(u)\|}$ represents the tool orientation vector.

The motion relationship between five physical axes and the tool tip can be expressed as

$$\begin{cases} q_s(u) = q_u(u)u_s(u) \\ q_{ss}(u) = q_{uu}u_s^2(u) + q_u(u)u_{ss} \\ q_{sss}(u) = q_{uuu}(u)u_s^3(u) + 3q_{uu}(u)u_s(u)u_{ss}(u) + q_u(u)u_{sss}(u) \end{cases}, q = X, Y, Z \dots \quad (18)$$

where $q_s(u)$, $q_{ss}(u)$, $q_{sss}(u)$ are the first order, second order and third-order derivative of the axis position with respect to the arc length s . $u_s(u)$, $u_{ss}(u)$, $u_{sss}(u)$ are the first order, second order and third-order derivative of the parameter u with respect to the arc length s .

Chord error constraint

To ensure the discrete points can reflect the characteristics of the original curve, it is necessary to consider the constraint of the chord error.

The chord error can be expressed as:

$$\delta(u) = \rho(u) - \sqrt{\rho^2(u) - \left(\frac{v_t T_s}{2}\right)^2} \leq \delta_m \quad (19)$$

where $\delta(u)$ is the chord error at the parameter u , $\rho(u)$ is the radius of curvature of the curve at u , v_t is the feedrate, T_s is the interpolation period, and δ_m is the maximum allowable chord error. The radius of curvature $\rho(u)$ can be calculated by

$$\rho(u) = \frac{1}{\|C_{ss}(u)\|} \quad (20)$$

Servo feed axis constraint

From Eq. (18), the velocity, acceleration, and jerk of the servo feed axis q can be expressed as:

$$\begin{cases} v_q = q_s(u)v_t \\ a_q = q_{ss}(u)v_t^2 + q_s(u)a_t \\ J_q = q_{sss}(u)v_t^3 + 3q_{ss}(u)v_t a_t + q_s(u)J_t \end{cases}, q = X, Y, Z, \dots \quad (21)$$

where v_t , a_t , J_t are the tangential velocity, tangential acceleration and tangential jerk. Each servo feed axis has a maximum allowable velocity, acceleration and jerk, so it needs to satisfy:

$$\begin{cases} |q_s(u)v_t| \leq V_{mq} \\ |q_{ss}(u)v_t^2 + q_s(u)a_t| \leq A_{mq} \\ |q_{sss}(u)v_t^3 + 3q_{ss}(u)v_t a_t + q_s(u)J_t| \leq J_{mq} \end{cases}, q = X, Y, Z, \dots \quad (22)$$

Tracking error constraint

The tracking error of tool affect the machining quality and precision¹¹, and limiting the tracking error of each axis could reduce the tracking error of tool. Currently, the servo feed axis usually adopts proportional-proportional-integral control. And the control structure is shown in Fig. 8.

where K_p is the proportional gain coefficient of the position loop. K_v is the proportional gain coefficient of the velocity loop. K_{vi} is the integral gain coefficient of the velocity loop. M is the equivalent mass. B_m is the equivalent damping coefficient and r_g is the transmission coefficient.

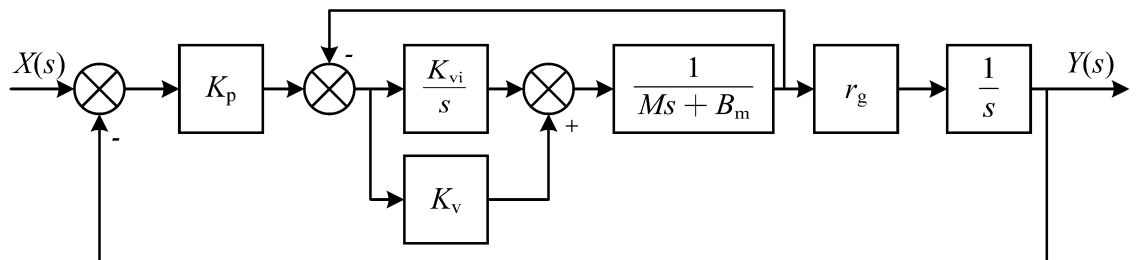


Figure 8. Feed axis control model.

The transfer function of the system is:

$$\Phi(s) = \frac{Y(s)}{X(s)} = \frac{K_p K_v r_g s + K_p K_{vi} r_g}{Ms^3 + (B_m + K_v r_g)s^2 + (K_{vi} + K_p K_v r_g)s + K_p K_{vi} r_g} \tag{23}$$

Define the tracking error $E(s) = X(s) - Y(s)$, then:

$$\Phi_e(s) = \frac{E(s)}{X(s)} = \frac{Ms^3 + (B_m + K_v r_g)s^2 + K_{vi} s}{Ms^3 + (B_m + K_v r_g)s^2 + (K_{vi} + K_p K_v r_g)s + K_p K_{vi} r_g} \tag{24}$$

Equation (24) can be written in differential form:

$$\frac{M}{K_p K_{vi} r_g} \ddot{e}(t) + \frac{B_m + K_v r_g}{K_p K_{vi} r_g} \dot{e}(t) + \frac{K_{vi} + K_p K_v}{K_p K_{vi}} e(t) + e(t) = \frac{M}{K_p K_{vi} r_g} J + \frac{B_m + K_v r_g}{K_p K_{vi} r_g} a + \frac{1}{K_p K_{vi} r_g} v \tag{25}$$

Where $e(t)$, $\dot{e}(t)$, $\ddot{e}(t)$ is the axis tracking error and its first-order, second-order and third-order derivative, J is the axis command jerk, a is the axis command acceleration, and v is the axis command velocity. According to the derivation of literature^{26,27}, if

$$\left| \frac{M}{K_p K_{vi} r_g} J + \frac{B_m + K_v r_g}{K_p K_{vi} r_g} a + \frac{1}{K_p r_g} v \right| \leq E \tag{26}$$

then mean $|e(t)| \leq E$.

Therefore, the tracking error of the axis is converted into a linear combination of velocity, acceleration, and jerk of the axis. The constraint of tracking error is:

$$|K_{qe3} J_q + K_{qe2} a_q + K_{qe1} v_q| \leq E_q, q = X, Y, Z, \dots \tag{27}$$

where

$$K_{qe3} = \frac{M_q}{K_{pq} K_{viq} r_{gq}}, K_{qe2} = \frac{B_{mq} + K_{vq} r_{gq}}{K_{pq} K_{viq} r_{gq}}, K_{qe1} = \frac{1}{K_{pq} r_{gq}}, q = X, Y, Z \tag{28}$$

Feedrate limitation profile

The axis acceleration, jerk and tracking error are related to the actual system state. In this paper, the critical velocity satisfying above constraints can be estimated according to the state of $a_t = 0, J_t = 0$, and the coefficient K_{dyn} can be used to achieve a more conservative estimate. In this paper, $K_{dyn} = 0.9$. For satisfying the above constraints, there is:

$$\begin{cases} v_t \leq K_{dyn} \sqrt{\frac{A_{mq}}{|q_{ss}(u)|}}, q = X, Y, Z, \dots \\ v_t \leq K_{dyn} \sqrt[3]{\frac{J_{mq}}{|q_{sss}(u)|}}, q = X, Y, Z, \dots \\ v_t \leq K_{dyn} \frac{E_q}{K_{qe1} |q_s(u)|}, q = X, Y, Z, \dots \end{cases} \tag{28}$$

The feedrate limitation can be calculated as follows:

$$v_t(u) \leq \min \left\{ \frac{2\sqrt{2\delta_m(\rho(u) - \delta_m)}}{T_s}, \frac{V_{mq}}{|q_s(u)|}, K_{dyn} \sqrt{\frac{A_{mq}}{|q_{ss}(u)|}}, K_{dyn} \sqrt[3]{\frac{J_{mq}}{|q_{sss}(u)|}}, K_{dyn} \frac{E_q}{K_{qe1} |q_s(u)|} \right\} \tag{29}$$

Dynamic lookahead distance

In the dynamic forward-looking feedrate scheduling based on SMVC, the look-ahead distance determines whether the system can satisfy constraints and smoothly follow command. To balance efficiency and smoothness, the look-ahead distance in this paper is calculated using the braking distance, which is the shortest distance required to decelerate to 0 at the current velocity and can be obtained by numerical simulation of SMVC. Due to the constraints of acceleration and jerk, the actual braking distance may be longer than in the ideal case. Then a positive coefficient K_4 can be multiplied to ensure sufficient look-ahead distance. Since the look-ahead distance is velocity-dependent, command velocity oscillation may occur when approaching the velocity minimum, and jitter can be suppressed using a mean filter.

Interpolation algorithms

After obtaining the command jerk for this period, the state at the end of this interpolation period can be written:

$$\begin{cases} a_{act}(k+1) = a_{act}(k) + J_{act}(k)T_s \\ v_{act}(k+1) = v_{act}(k) + a_{act}(k)T_s + \frac{1}{2}J_{act}(k)T_s^2 \\ \Delta s(k) = s(k+1) - s(k) = v_{act}(k)T_s + \frac{1}{2}a_{act}(k)T_s^2 + \frac{1}{6}J_{act}(k)T_s^3 \end{cases} \tag{30}$$

According to paper²⁸, using the second-order Taylor development, there is:

$$u(k+1) = u(k) + u_s(u(k))\Delta s(k) + \frac{1}{2}u_{ss}(u(k))(\Delta s(k))^2 \quad (31)$$

Simulation

To prove the effectiveness of proposed method, simulations are carried out. The simulations were carried out on a PC with Windows10 operation system. The simulation environment was Matlab 2020a and the CPU was Intel® Xeon® Silver 4110 CPU @ 2.10GHz.

The constraints and parameters for simulation of each axis are shown in Table 1:

Simulation 1: butterfly-shaped curve

A butterfly-shaped curve is used for simulation as shown in Fig. 9. And the curve information is shown in Appendix A Table A1.

The tangential feedrate command is set to 50 mm/s, the maximum tangential acceleration to 1000 mm/s², maximum tangential jerk to 10,000 mm/s³, chord error limit to 0.1 μm, $K_1 = 0.0385$, $K_3 = 5$, $K_4 = 2$. The feedrate scheduling results in parameter domain are shown in Fig. 10a and results in time domain is shown in Fig. 10b–d, the axes tracking error is shown in Fig. 10e. The results show that the method proposed could schedule feedrate satisfying the tangential velocity, acceleration and jerk constraints. And the tracking error of each axis is also within the allowable range.

Axis acceleration and jerk are the typical dynamic constraints. Comparing the considering dynamic constraints time-optimal method⁵ (M1) with the method proposed in this paper (M2), the results shows in Fig. 11 and Table 2. The results show that the method proposed in this paper, compared to M1, effectively reduces the exceeding of dynamic constraints without significantly increasing the machining time.

Simulation 2: open-pocket curve

The open-pocket curve is shown in Fig. 12. And the curve information is shown in Appendix A Table A2.

The simulation settings are same as Simulation1. And the feedrate scheduling results in parameter domain is shown in Fig. 13a and results in time domain is shown in Fig. 13b–d, the axis tracking error is shown in Fig. 13e,f. The tangential constraints and axes tracking error are all be satisfied.

Comparing M1 with M2, the results are shown in Fig. 14 and Table 3. Although M2 method has a greater number of exceeding constraints cycles compared to M1, the exceeding constraint ratio is very minor compared to M1. Therefore, it can still be considered that the method proposed in this paper has a better performance on satisfying dynamic constraints.

	Max velocity	Max acceleration	Max jerk	Max tracking error	K_{qe1}	K_{qe2}	K_{qe3}
X	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01667 s/mm	6.1642E-5 s ² /mm	4.5062E-8 s ³ /mm
Y	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01667 s/mm	6.1642E-5 s ² /mm	4.5062E-8 s ³ /mm
Z	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01667 s/mm	6.1642E-5 s ² /mm	4.5062E-8 s ³ /mm
A	1 rad/s	10 rad/s ²	100 rad/s ³	0.1 rad	1.25E-4 s/mm	1.0836E-6 s ² /mm	3.3796E-10 s ³ /rad
C	0.5 rad/s	5 rad/s ²	50 rad/s ³	0.1 rad	1.25E-4 s/rad	1.0836E-6 s ² /rad	3.3796E-10 s ³ /rad

Table 1. Axis constraints and parameters for simulation.

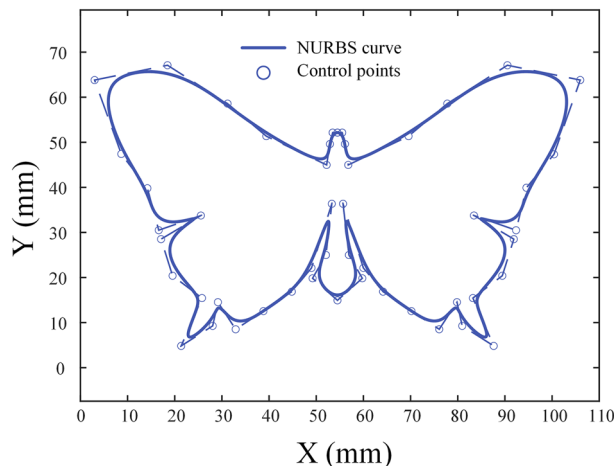


Figure 9. Butterfly-shaped curve.

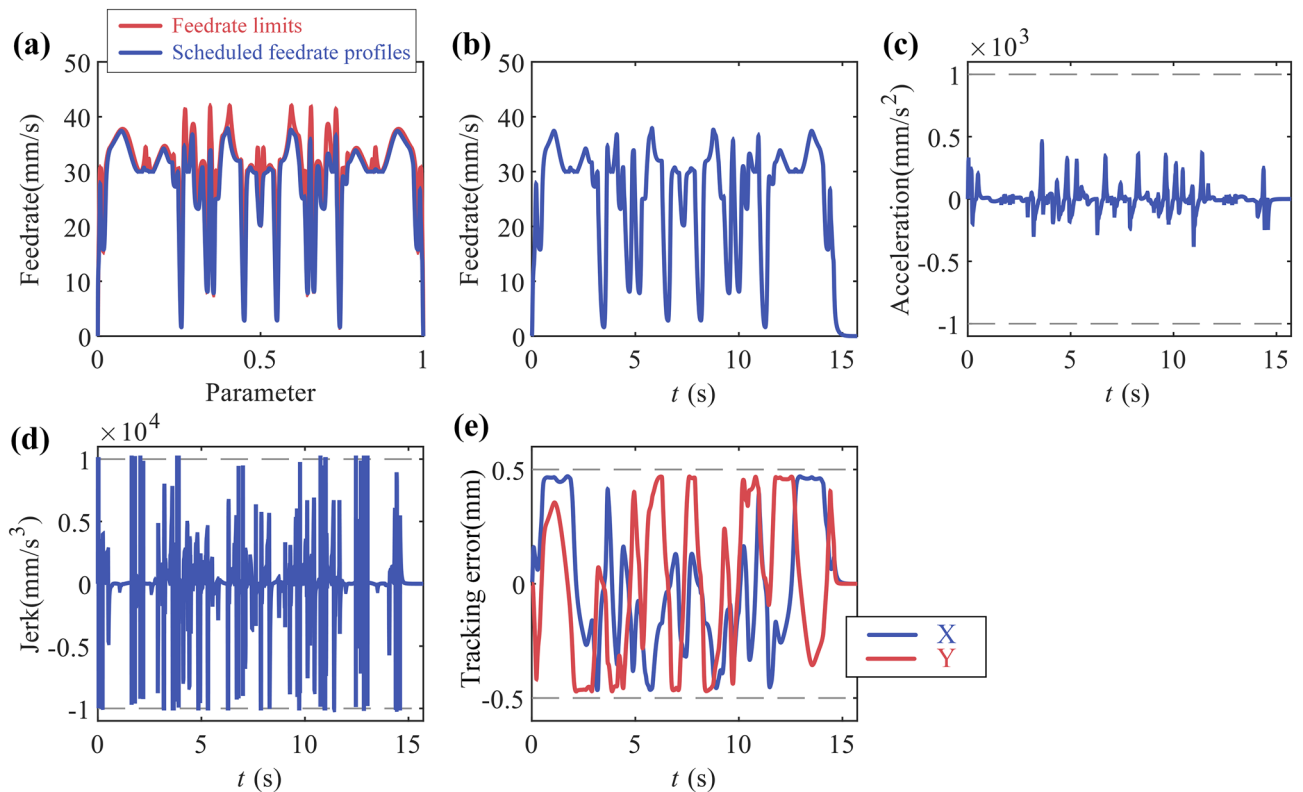


Figure 10. Butterfly-shaped curve simulation results: (a) feedrate in parameter domain; (b) feedrate in time domain; (c) tangential acceleration in time domain; (d) tangential jerk in time domain; (e) tracking error.

Experiment

To verify the real-time performance and tracking error constraints of the method proposed in this paper, experiments were carried out on a five-axis motion platform as shown in Fig. 15.

The control system of this platform is developed by TwinCAT3 and runs on a Windows 7 industrial computer an Inter® Core™ i7-6700 CPU @ 3.40 GHz. The interpolation and feedrate scheduling cycle is 2 ms. The tracking error coefficient can be identified through actual state and tracking error⁴. The identified result and constraints for each axis are shown in Table 4.

Experiment 1: butterfly-shaped curve with override change

The tangential feedrate command is set to 50 mm/s, maximum tangential acceleration to 1000 mm/s², maximum tangential jerk to 10,000 mm/s³, chord error limit to 0.1 μm, $K_1 = 0.0385$, $K_3 = 5$, $K_4 = 2$. Before $u \geq 0.5$, the override is set to 100%. After $u \geq 0.5$, the override is adjusted to 40%, which means the tangential feedrate command is reduced to 20 mm/s. After $u \geq 0.7$, the override is set back to 100%.

The results are shown in Fig. 16. Figure 16a demonstrates the capability of the method proposed in this paper to adjust the override in real time. This capability provides the possibility for online control of cutting forces, which is an effective method for improving machining quality. Figure 16b shows that the maximum tracking errors for the X and Y axes are 0.4832 mm and 0.4347 mm, both less than the constraint of 0.5 mm. This demonstrates the effectiveness of the tracking error constraints in this paper, and smaller tracking errors can indirectly reduce the machining contour errors, resulting in a surface with higher quality.

Experiment 2: open-pocket curve

Due to the dynamic performance of C axis, the experiment settings are same to experiment except $K_4 = 3.2$. The results are shown in Fig. 17. Figure 17a demonstrates the capability of the method proposed in this paper for 5-axis machining. Figure 17b,c indicate that the maximum tracking errors for linear axes X, Y and Z are 0.0715 mm, 0.4272 mm and 0.4636 mm, all less than the constraint 0.5 mm. The maximum tracking errors of rotary axes A and C are 0.0043 rad and 0.0114 rad, both less than the constrained value of 0.1 rad.

Computation time analysis

The computation time of Experiment 1 and 2 is assessed by TwinCAT3 software and shown in Fig. 18. For the butterfly-shaped curve, the computation time can be kept below 50 μs in most cases, with a maximum not exceeding 80 μs. For the open-pocket curve, the computation time can be kept below 20 μs in most cases, with a maximum not exceeding 50 μs. Although it requires five-axis coordinate transformation and solution of properties of two NURBS curves, the computation time for the open-pocket curve is still significantly less than butterfly-shaped

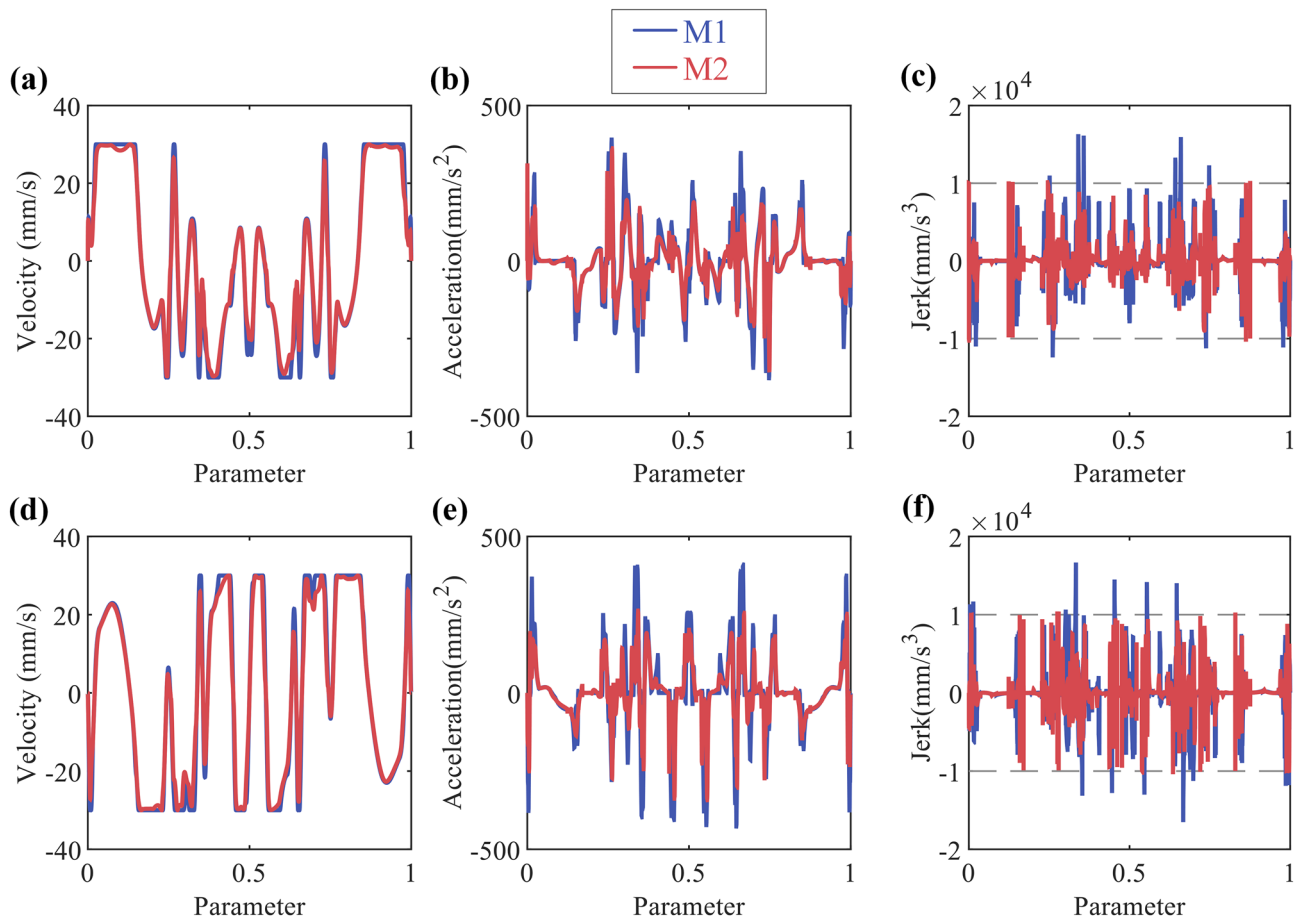


Figure 11. Butterfly-shaped curve simulation results: (a) X-axis velocity; (b) X-axis acceleration; (c) X-axis jerk; (d) Y-axis velocity; (e) Y-axis acceleration; (f) Y-axis jerk.

	Machine time ($u \geq 0.99999$)	Acceleration exceeding constraint cycle number		Jerk exceeding constraint cycle number	Maximum exceeding constraint ratio for jerk
M1	12.908 s	0	112		X:58.5%/Y:62.07%
M2	15.362 s	0	2		X:0.49%

Table 2. Performance on dynamic constraints of butterfly-shaped.

curve. This difference in computation time may be due to the differing complexities of the curves. The butterfly-shaped curve has 52 control points, while the open-pocket curve has 8 (Supplementary information).

Figure 18a illustrates the variation in computation time under different override the override was reduced when u was within $[0.5, 0.7]$. Figure 18a shows that after the reduction of override, the average computation time decreased. A possible reason is that after feedrate has decreased, the look-ahead distance becomes shorter, which means a shorter parameter look-ahead interval and fewer iterations required to find the minimum velocity.

Figure 18 demonstrates the computational efficiency of the method proposed in this paper for both complex planar tool paths and spatial five-axis tool paths, proving the great potential of the method for real-time applications.

Conclusion

High-precision direct interpolation of NURBS curves requires feedrate scheduling results satisfy the geometric, kinematic and dynamic constraints. Dynamic constraints are related to the current state of tool and cannot be directly constrained by velocity at sampling points. To achieve direct control of dynamic constraints, this paper proposes a dynamic look-ahead feedrate scheduling method based on sliding mode velocity control. The SMVC acceleration and deceleration method can control velocity to command velocity with any initial state and its stability has been proven. This paper also introduces the performance improvement method and end-point-reachable method. This paper analyzes the common constraints and linearizes the tracking error into a linear combination of velocity, acceleration, and jerk. The braking distance is used to estimate the look-ahead distance.

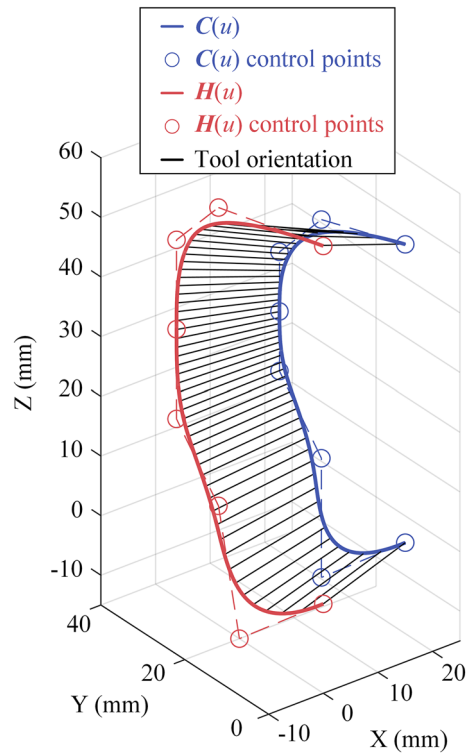


Figure 12. open-pocket curve.

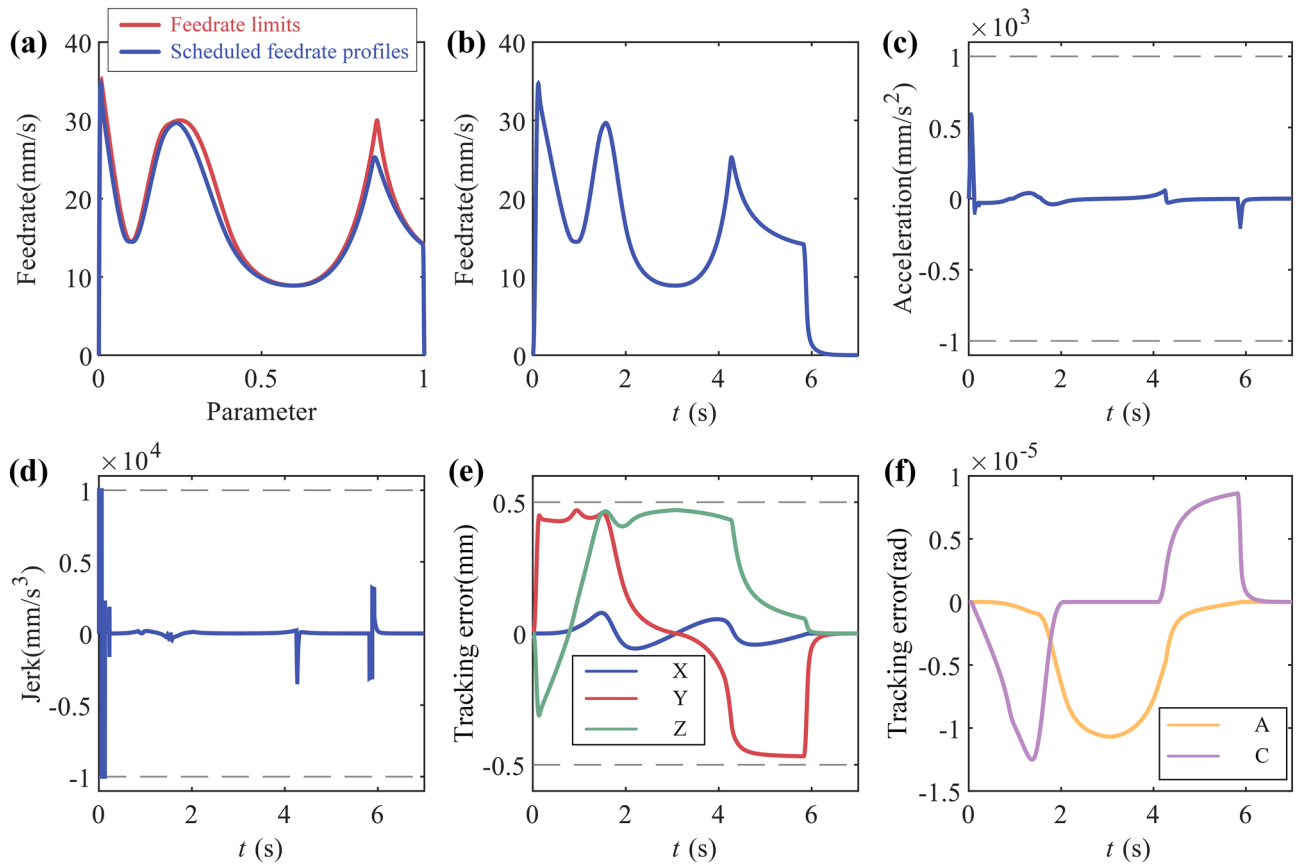


Figure 13. Open-pocket curve simulation results: (a) feedrate in parameter domain; (b) feedrate in time domain; (c) tangential acceleration in time domain; (d) tangential jerk in time domain; (e) tracking error of X,Y and Z; (f) tracking error of A and C.

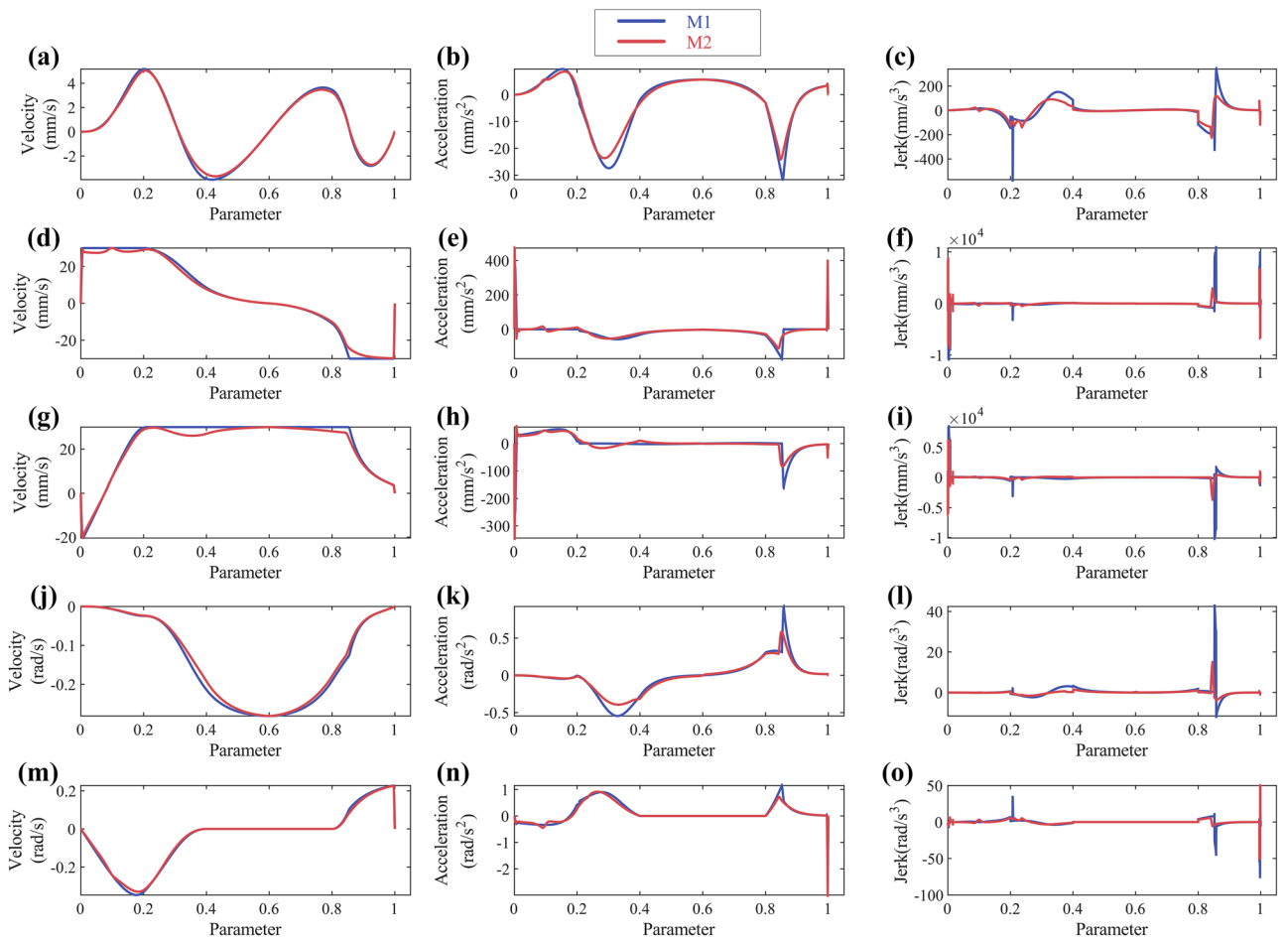


Figure 14. Open-pocket curve simulation results: (a)(b)(c) X-axis velocity/acceleration/jerk; (d)(e)(f) Y-axis velocity/acceleration/jerk; (g)(h)(i) Z-axis velocity/acceleration/jerk; (j)(k)(l) A-axis velocity/acceleration/jerk; (m)(n)(o) C-axis velocity/acceleration/jerk.

	Machine time ($u \geq 0.99999$)	Acceleration exceeding constraint cycle number	Jerk exceeding constraint cycle number	Maximum exceeding constraint ratio for jerk
M1	5.9540 s	0	30	Y:7.43%/Z:0.8%/C:48.5%
M2	6.6420 s	0	51	C:0.03%

Table 3. Performance on dynamic constraints of open pocket curve

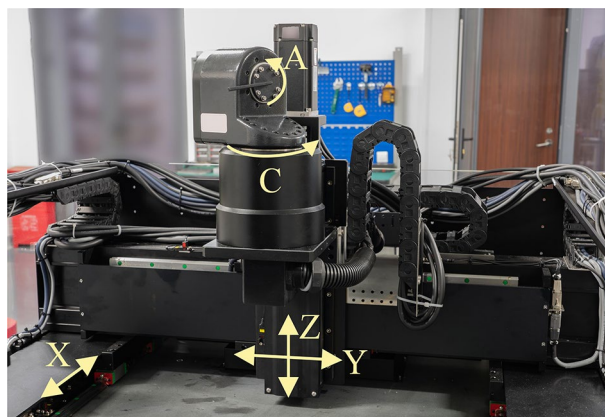


Figure 15. Five-axis motion platform.

	Max velocity	Max acceleration	Max jerk	Max tracking error	K_{qe1}	K_{qe2}	K_{qe3}
X	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01249 s/mm	1.1639E-6 s ² /mm	3.9411E-8 s ³ /mm
Y	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01381 s/mm	3.5954E-5 s ² /mm	6.0287E-7 s ³ /mm
Z	100 mm/s	1000 mm/s ²	10,000 mm/s ³	0.5 mm	0.01282 s/mm	4.3869E-5 s ² /mm	2.3732E-7 s ³ /mm
A	1 rad/s	10 rad/s ²	100 rad/s ³	0.1 rad	0.01200 s/rad	1.6516E-5 s ² /mm	4.4417E-8 s ³ /mm
C	0.5 rad/s	5 rad/s ²	50 rad/s ³	0.1 rad	0.02913 s/rad	3.1890E-4 s ² /mm	1.6285E-7 s ³ /mm

Table 4. Axis constraints and parameter of five-axis platform.

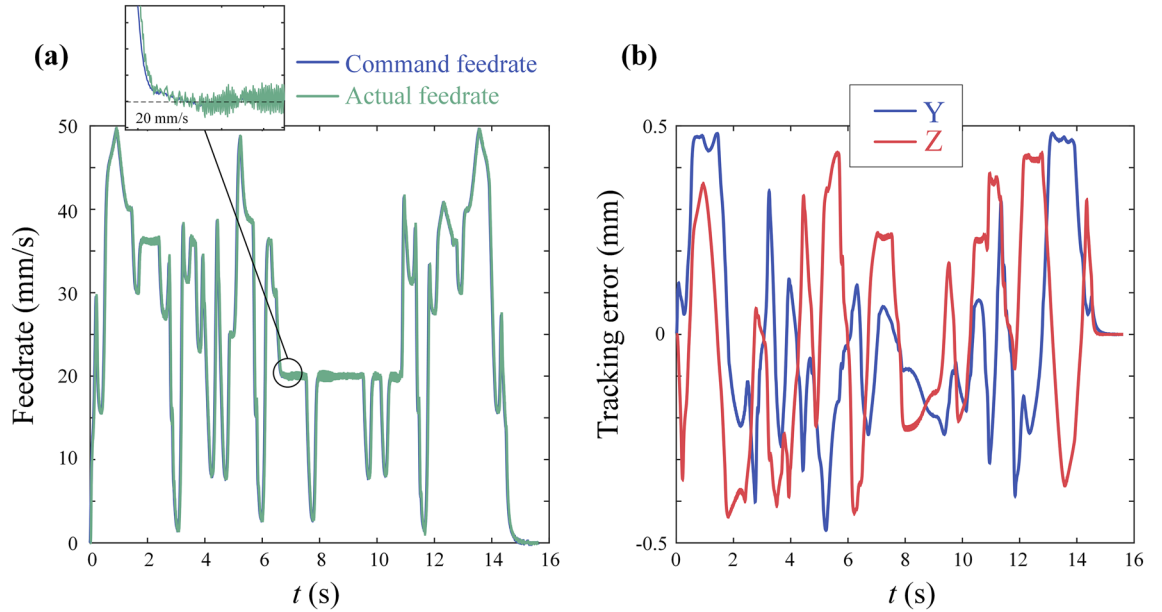


Figure 16. Experiment results of butter-shaped curve: (a) command and actual feedrate profile in time domain; (b) tracking error of axis.

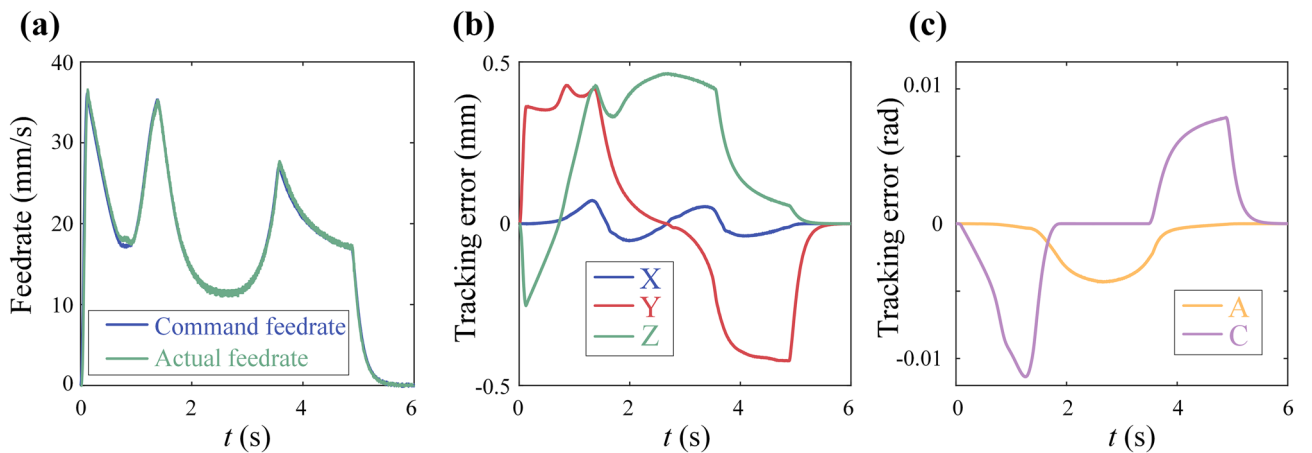


Figure 17. Experiment results of open-pocket curve: (a) command and actual feedrate profile in time domain; (b) tracking error of X, Y and Z; (c) tracking error of A and C.

The effectiveness of the proposed method in this paper is verified through simulation and experiment. In future research, selecting a smoother and more efficient sliding mode surface is a research direction. The convergence speed at the endpoint also needs to be optimized, and further research is required on the selection of various coefficients.

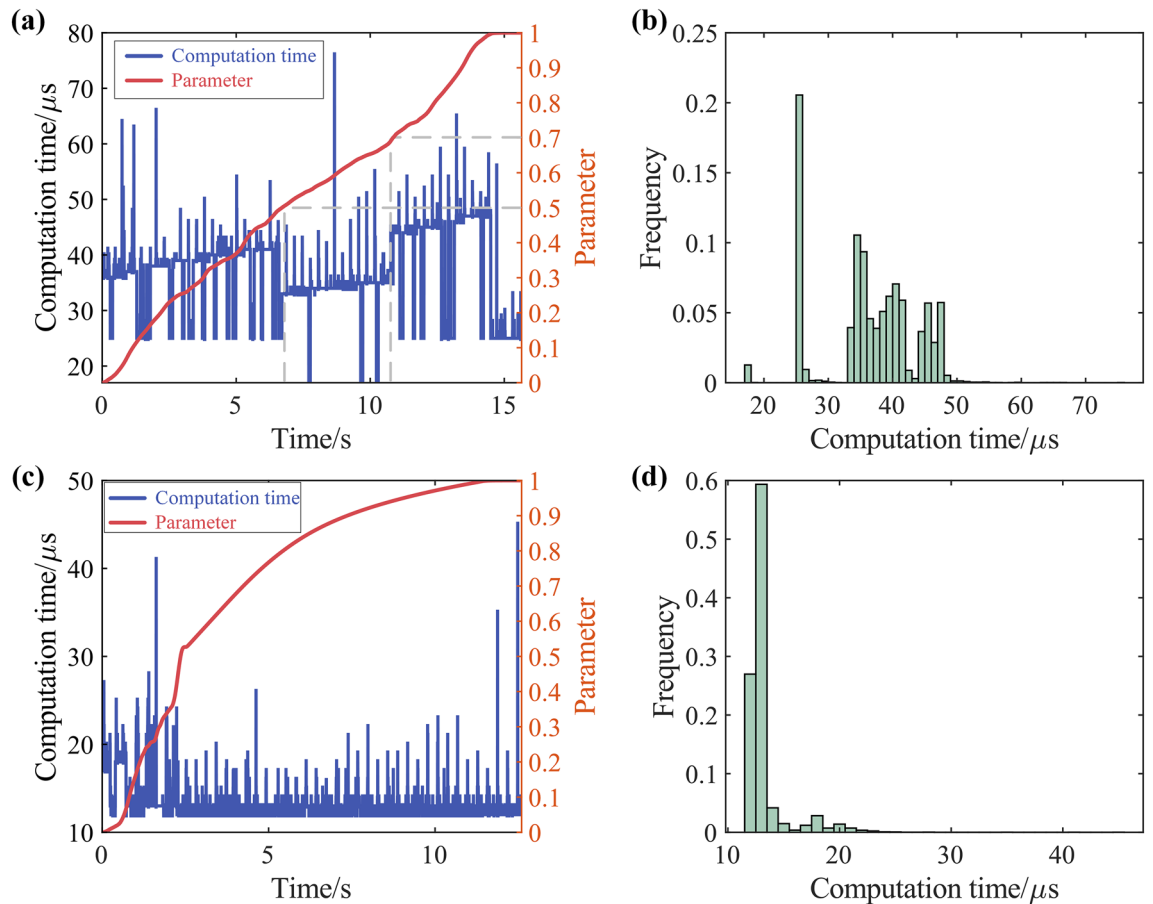


Figure 18. Computation time: (a) butterfly-shaped curve computation time and parameter timing diagram; (b) butterfly-shaped curve computation time histogram; (c) open-pocket curve computation time and parameter timing diagram; (d) open-pocket curve computation time histogram.

Data availability

The datasets used or analyzed during the current study are available from the corresponding author on reasonable request.

Received: 17 May 2024; Accepted: 28 June 2024

Published online: 04 July 2024

References

- Zhao, X., Zhao, H., Yang, J. & Ding, H. An Adaptive Feedrate Scheduling Method with Multi-constraints for Five-Axis Machine Tools. In *INTELLIGENT ROBOTICS AND APPLICATIONS (ICIRA 2015)* (ed. Liu, H.) (Springer-Verlag, 2015).
- Kim, H. & Okwudire, C. E. Intelligent feedrate optimization using a physics-based and data-driven digital twin. *CIRP Ann.* **72**, 325–328 (2023).
- Sun, Y., Jia, J., Xu, J., Chen, M. & Niu, J. Path, feedrate and trajectory planning for free-form surface machining: A state-of-the-art review. *Chin. J. Aeronaut.* **35**, 12–29 (2022).
- Zhang, Y., Ye, P., Zhao, M. & Zhang, H. Dynamic feedrate optimization for parametric toolpath with data-based tracking error prediction. *Mech. Syst. Signal Process.* **120**, 221–233 (2019).
- Liu, H., Liu, Q., Sun, P., Liu, Q. & Yuan, S. The optimal feedrate planning on five-axis parametric tool path with geometric and kinematic constraints for CNC machine tools. *Int. J. Prod. Res.* **55**, 3715–3731 (2017).
- Lu, L., Zhang, L., Ji, S., Han, Y. & Zhao, J. An offline predictive feedrate scheduling method for parametric interpolation considering the constraints in trajectory and drive systems. *Int. J. Adv. Manuf. Technol.* **83**, 2143–2157 (2016).
- Xiao, J., Fang, J., Li, B. & Zhang, H. Dynamic velocity planning method for parametric toolpath with mode-based tracking error control. *Int. J. Adv. Manuf. Technol.* **128**, 4805–4817 (2023).
- Chen, M. & Sun, Y. Contour error-bounded parametric interpolator with minimum feedrate fluctuation for five-axis CNC machine tools. *Int. J. Adv. Manuf. Technol.* **103**, 567–584 (2019).
- Zhao, K., Li, S., Kang, Z. & Liu, Z. Smooth trajectory generation based on contour error constraint and parameter correction b-spline. *Int. J. Adv. Manuf. Technol.* **119**, 4359–4373 (2022).
- Yang, J., Yin, X. & Sun, Y. PSO-Based Feedrate Optimization Algorithm for Five-Axis Machining with Constraint of Contour Error. *Machines* **11**, 501 (2023).
- Erkorkmaz, K. & Altintas, Y. High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation. *Int. J. Mach. Tools Manuf.* **41**, 1323–1345 (2001).
- Du, X., Huang, J. & Zhu, L.-M. A complete S-shape feed rate scheduling approach for NURBS interpolator. *J. Comput. Des. Eng.* **2**, 206–217 (2015).

13. Lee, A.-C., Lin, M.-T., Pan, Y.-R. & Lin, W.-Y. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Comput. Aided Des.* **43**, 612–628 (2011).
14. Ni, H., Yuan, J., Ji, S., Zhang, C. & Hu, T. Feedrate Scheduling of NURBS Interpolation Based on a Novel Jerk-Continuous ACC/DEC Algorithm. *IEEE Access* **6**, 66403–66417 (2018).
15. Li, H. *et al.* A novel feedrate scheduling method based on Sigmoid function with chord error and kinematic constraints. *Int. J. Adv. Manuf. Technol.* **119**, 1531–1552 (2022).
16. Ren, X., Fan, J. & Pan, R. A novel and efficient jerk-smooth feedrate scheduling algorithm for NURBS interpolation. *Int. J. Adv. Manuf. Technol.* **130**, 1221–1239 (2024).
17. Wu, B., Ma, J., Wei, L., Liao, X. & Lu, J. NURBS Interpolator with Scheduling Scheme Combining Cubic and Quartic S-shaped Feedrate Profiles Under Drive and Chord Error Constraints. *Comput. Aided Des.* **152**, 103380 (2022).
18. Sang, Y., Yao, C., Lv, Y. & He, G. An improved feedrate scheduling method for NURBS interpolation in five-axis machining. *Precis. Eng.* **64**, 70–90 (2020).
19. Jia, Z., Song, D., Ma, J., Hu, G. & Su, W. A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. *Int. J. Mach. Tools Manuf.* **116**, 1–17 (2017).
20. Sun, S., Zhao, P., Zhang, T., Li, B. & Yu, D. Smoothing interpolation of five-axis tool path with less feedrate fluctuation and higher computation efficiency. *J. Manuf. Process.* **109**, 669–693 (2024).
21. Tong, X., Liu, Q., Zhou, Y. & Sun, P. A digital twin-driven cutting force adaptive control approach for milling process. *J. Intell. Manuf.* <https://doi.org/10.1007/s10845-023-02193-2> (2023).
22. Sun, Z. *et al.* Enhanced feedrate scheduling algorithm for CNC system with acceleration look-ahead and sin2 acceleration profile. *Int. J. Adv. Manuf. Technol.* **119**, 217–231 (2022).
23. Zhang, Y., Zhang, H., Shi, J., Jiang, J. & Zhao, M. Development of an Adaptive-Feedrate Planning and Iterative Interpolator for Parametric Toolpath with Normal Jerk Constraint. *Int. J. Precis. Eng. Manuf.* **22**, 73–81 (2021).
24. Song, D.-N., Zhong, Y.-G. & Ma, J.-W. Look-ahead-window-based interval adaptive feedrate scheduling for long five-axis spline toolpaths under axial drive constraints. *Proc. Inst. Mech. Eng. PART B-J. Eng. Manuf.* **234**, 1656–1670 (2020).
25. Sun, P. *Research on Key Technologies of Opto-electromechanical Cooperative Control in Laser Engraving* (Beihang University, 2023).
26. Guo, J.-X., Zhang, K., Zhang, Q. & Gao, X.-S. Efficient time-optimal feedrate planning under dynamic constraints for a high-order CNC servo system. *Comput. Aided Des.* **45**, 1538–1546 (2013).
27. Guo, J., Zhang, Q., Gao, X.-S. & Li, H. Time optimal feedrate generation with confined tracking error based on linear programming. *J. Syst. Sci. Complex.* **28**, 80–95 (2015).
28. Zhao, H., Zhu, L. & Ding, H. A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction. *Int. J. Mach. Tools Manuf.* **75**, 1–8 (2013).

Author contributions

L.W.: conceptualization, methodology, formal analysis, software, validation, original draft, review and editing; Q.L.: conceptualization, supervision; P.S.: conceptualization, methodology; S.L.: methodology, review and editing, R.Y.: validation, review and editing; Z.Y.: validation, review and editing.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-024-66203-w>.

Correspondence and requests for materials should be addressed to Q.L.

Reprints and permissions information is available at www.nature.com/reprints.

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



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024