



# Smooth minimum time trajectory planning with minimal feed fluctuation

Kai Zhao<sup>1,2</sup> · Shurong Li<sup>3</sup> · Zhongjian Kang<sup>1</sup>

Received: 15 April 2019 / Accepted: 14 August 2019 / Published online: 24 August 2019  
© Springer-Verlag London Ltd., part of Springer Nature 2019

## Abstract

Smooth and effective feedrate profile is very important for high-speed and high-precision machining. In this paper, a two-stage feedrate scheduling scheme (TFSS) is proposed to efficiently generate the smooth feedrate profile. The minimum time trajectory planning (MTTP) is firstly employed to generate the initial feedrate profile considering constraints of both the kinematics and the chord error. To improve the computational efficiency, the MTTP is transformed into a problem of linear programming by applying direct transcription method. Then, all the sharp corners along the initial feedrate profile are identified, especially those that are interacted by each other. The jerk-limited method of eliminating interaction (JMEI) at sharp corners is applied to smooth the final feedrate profile. Finally, the improved feed correction polynomials are applied to reduce the fluctuations owing to non-arc-length parameterization. To illustrate the validity and rationality of the proposed scheme, two curves were employed to verify the proposed method. The simulation and experimental results demonstrate the efficiency of the proposed scheme for the nonuniform rational basis spline (NURBS) tool paths.

**Keywords** Minimum time · Trajectory planning · Linear programming · NURBS interpolation · Feedrate planning

## 1 Introduction

In modern processing industry, the parametric curves are extensively applied for workpieces of various fields, such as aerospace, transportation, automobile, and so on. For this reason, many parameter interpolation algorithms have been proposed to generate reference positions as the input of motor controllers. Direct approximation method with Taylor's expansion is widely employed in the interpolation process. The first-order Taylor expansion is usually adequate. If it is not adequate, the second-order approximation could be adopted. In order to further improve the approximation accuracy, the Adams-Bashforth method (ABM) and fourth-order Runge-Kutta method are applied instead of Taylor's

expansion [1]. By introducing the concept of “feedback,” the predictor-corrector method is presented to avoid directly calculating the first- or second-order derivatives of the parametric curve [2]. Since some parametric curves are not parameterized by arc length, some unexpected feedrate fluctuations may appear during the interpolation. The interpolation method of feed correction polynomial is proposed [3]. This method is further improved by introducing the adaptive quadrature length method [4] and the multiple inverse length function method [5].

However, no matter which interpolation method is used, the appropriate feedrate is needed before interpolation. Conservative constant feed can not make full use of the performance of machine tools, which may lead to larger chord error in places with larger curvature. So, the algorithms which can adjust the feedrate adaptively have been extensively researched. By considering the chord error, curvature, and contour error, an integrated look-ahead dynamics-based algorithm is proposed [6]. This method can reduce the feedrate when the chord error exceeds the tolerance value or the curvature is too large. In addition, if the contour error estimated through high-order transfer function is greater than the error bound, the algorithm will re-plan the feedrate profile. Sun et al. present a method of adaptive feedrate for improving precision of five-axis computer

✉ Kai Zhao  
upczhaokai@163.com

<sup>1</sup> College of Information and Control Engineering,  
China University of Petroleum (East China),  
Qingdao, 266580, China

<sup>2</sup> College of Computer Science and Information Engineering,  
Anyang Institute of Technology, Anyang, 455000, China

<sup>3</sup> School of Automation, Beijing University of Posts  
and Telecommunications, Beijing, 100876, China

numerical control (CNC) machining [7]. This method takes consideration of both the confined nonlinear geometric error and the angular feedrate. Hence, tangential velocity at sensitive regions can be adjusted based on geometric error with bi-directional scan algorithm. Another online speed profile generator is developed for industrial machine tools based on neuro-fuzzy network [8]. In this literature, it is assumed that the machining mission must be divided into several parts due to the capacity of NC system. The feedrate is adjusted adaptively according to different feed limits which are caused owing to changing the spindle speed for processing each part.

Generally, a smooth feed profile is generated by restraining the jerk. However, most of jerk-limited methods with adaptive feedrate require an initial command feedrate which is empirical and conservative. The optimal feedrate profile can not be generated. Therefore, optimization methods are introduced to the trajectory planning by some researchers. For example, two optimal time-jerk trajectory planning methods are analyzed and validated for robotic manipulators [9]. In the two algorithms, both the time and jerk square are included in the objective function, so as to satisfy the need of fast execution and the need of a smooth trajectory simultaneously. Through modifying two weights of the jerk square and time, optimal time-jerk trajectory could satisfy kinds of requirements. Zhang et al. propose an algorithm which can generate the smooth time-optimal trajectory for cartesian computer numerical control (CNC) manufacturing systems [10]. In this algorithm, smooth minimum time feed profile is obtained by solving a nonlinear program (NLP) with constraints of axial feeds, axial accelerations, and axial jerks. Considering G01 paths, Lin et al. present a trajectory planning model combining fitting error constraints and dynamic constraints [11]. In this literature, the cubic B-spline fitting with confined error and speed planning method are combined together instead of traditional sequential implementation.

To further reduce the planning time, linear programming is applied to solve the optimization problem of feedrate. In order to obtain an optimal motion trajectory for a robotic manipulator, a robust planning algorithm based on linear programming is addressed, since the manipulator dynamics and torque parameters are uncertain [12]. Considering the feedrate bound, axis acceleration bounds, and axis tracking errors, Guo et al. transform feedrate planning into an optimal control problem [13]. In order to solve the optimal problem with linear programming, some constraints are simplified and jerk bounds are not considered. For integrating jerk constraints into an optimization problem, Liu et al. firstly obtain an upper bound of feedrate for spline curve tool paths [14]. Secondly, the jerk constraints are approximated as linear ones based on the upper bound. Finally, the linear programming method can be adopted to solve the

optimization problem. Ye et al. extend the feedrate planning method based on optimization to the Frenet-Serret frame by re-parameterizing parametric tool paths [15]. Afterwards, a time optimal method for parametric trajectory generation is proposed based on linear programming algorithm.

However, most of optimal trajectory planning methods are mainly concerned about feedrate planning. There is few mention about the interpolation method which is very important for generating servo controlling commands. Although a method of interpolation point generation for robotic manipulators is developed [16], it is time consuming since high-order derivative of parameter is required.

In this paper, a two-stage feedrate scheduling scheme (TFSS) is proposed to obtain smooth profile. The first stage constructs a problem of minimum time trajectory planning (MTTP) in order to fully utilize the performance of the machine tool. Then, the MTTP problem is transformed into a linear programming problem for further improving the computation efficiency. According to the solution of MTTP, the second stage re-plan the feedrate based on a bell-shape profile. The final feedrate is suitable for real-time interpolation. In the second stage, in order to deal with the overlapping phenomenon, known as “ripple effect” [1] at the sharp corners, the jerk-limited method of eliminating interaction (JMEI) is proposed. Since the feedrate is re-planned by the JMEI in the time domain, the results of JMEI can be directly used for interpolation of feed correction polynomial. Finally, the reference positions calculated by the interpolation method are sent to servo controllers. The flowchart of the TFSS and feed correction polynomial is shown as Fig. 1, where  $S_{i+1}$  and  $S_i$  denote displacements in the  $(i + 1)$ th and the  $i$ th interpolation period, respectively,  $V_i$  is a planned feedrate in the  $i$ th period,  $T$  is the interpolation period and  $m_{i+1}$  is curve parameter yielded by feed correction polynomial in the  $(i + 1)$ th period.

The left portions of this paper are organized as follows. In Section 2, MTTP for a cartesian 3-axis system is formulated, imposing the constraints of the resultant feedrate, the axial accelerations, and the chord error. Then, this MTTP problem is discretized with direct transcription method and reformulated as a linear programming problem. These are illustrated in Section 3. The method of smooth feedrate generation, namely JMEI, is proposed in Section 4. Machining results of simulation and experiment for two NURBS tool paths are shown in Section 5. Section 6 concludes the whole paper finally.

## 2 MTTP problem

Let  $\mathbf{C} = [x(m) \ y(m) \ z(m)]$ ,  $m \in [0, 1]$  be a curve path, where  $m$  denotes the path parameter. The MTTP problem aims to make the machine tool traverse the entire tool path

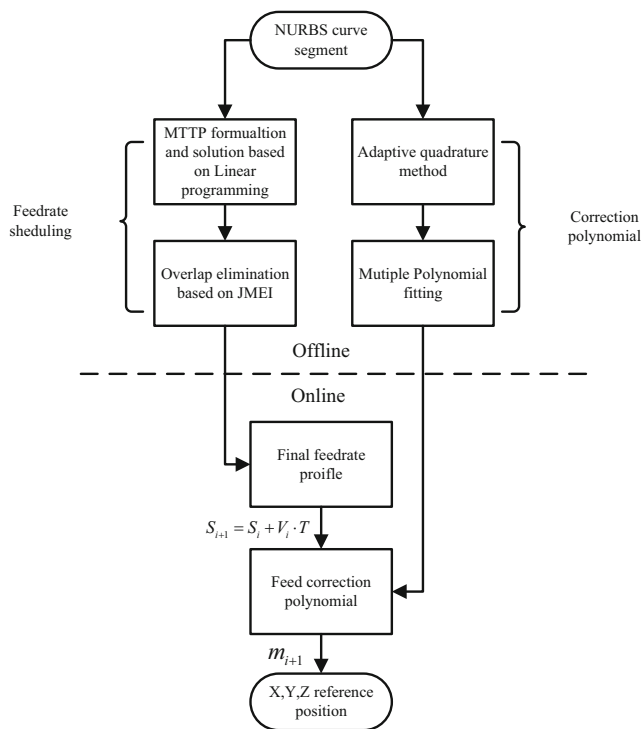


Fig. 1 Flowchart of TFSS and feed correction polynomial

in the shortest time, so the objective of the MTTP problem can be written as

$$\min_{\mathbf{v}} T = \int_{t_0}^{t_n} 1 dt \tag{1}$$

where  $\mathbf{v} = [v_x \ v_y \ v_z]$  denotes velocity vector of the cutter tip at any one point on the path. The velocity vector needs to be planned to minimize the objective function. While the velocity is planned, the acceleration of each axis is also planned and let  $\mathbf{a} = [a_x \ a_y \ a_z]$  be the axial acceleration vector.

Define the derivative of an arbitrary variable  $w$  with respect to parameter  $m$  as  $w'$  and the derivative with respect to time  $t$  as  $\dot{w}$ . For example,  $\dot{x} = \frac{dx}{dt}$  represents derivative of displacement along  $x$  axis w.r.t time  $t$ , and  $x' = \frac{dx}{dm}$  indicates derivative of displacement along the  $x$  axis w.r.t parameter  $m$ . They have the following relationship:

$$\dot{x} = \frac{dx}{dt} = \frac{dx}{dm} \frac{dm}{dt} = x' \dot{m} \tag{2}$$

Obviously, the MTTP problem must be subject to the pre-defined path and other constraints, such as the feedrate, axial accelerations, and the chord error. According to differential relations of displacement, velocity, and acceleration, the following equation can be derived as

$$\dot{\mathbf{C}}(t) = \mathbf{v}, \dot{\mathbf{v}}(t) = \mathbf{a} \tag{3}$$

From Eqs. 2 and 3, the feedrate can be obtained from the velocity vector

$$v(t) = \|\mathbf{v}\| = \|\dot{\mathbf{C}}\| = \|\mathbf{C}'\| \dot{m} \leq v_F \tag{4}$$

where  $v_F$  denotes the command feedrate, and  $\|\bullet\|$  is the 2-norm. In addition, the acceleration for each axis also is limited because of the capacity of a machine tool, which is described as

$$\begin{cases} \mathbf{a}_{\min} \leq \mathbf{a} \leq \mathbf{a}_{\max} \\ \mathbf{a} = \dot{\mathbf{v}} = \mathbf{C}'\ddot{m} + \mathbf{C}''\dot{m}^2 \end{cases} \tag{5}$$

Meanwhile, in actual machining, two adjacent interpolation points are not connected by an arc, but a short straight line. The chord error represents the maximum distance between the straight line and the arc, which is an important part of machining error in a NC system. According to the definition of the chord error [18], there is the following relationship shown as

$$v^2(t) \leq \frac{8\delta\rho(t) - 4\delta^2}{T^2} \approx \frac{8\delta\rho(t)}{T^2} = \frac{8\delta}{\kappa(t)T^2} \tag{6}$$

where  $\delta$  is the chord error bound,  $\kappa$  denotes the curvature, and  $\rho$  is the radius of curvature. Let  $ch(t)$  denotes the chord error, when the tangential velocity is  $v(t)$ , there is

$$ch(t) = \frac{\kappa(t)T^2v^2(t)}{8} \leq \delta \tag{7}$$

Since the centripetal acceleration of the cutter tip is defined as  $a_c(t) = \kappa(t)v^2(t)$ , from Eqs. 6 and 7, the chord error constraint can be transformed to the centripetal acceleration constraint. Hence, the centripetal acceleration is restrained as follows:

$$a_c(t) \leq \mu \quad \text{with} \quad \mu = \frac{8\delta}{T^2} \tag{8}$$

In view of a parametric path, the curvature at parameter value  $m$  can be described as

$$\kappa(m) = \frac{|\mathbf{C}'(m) \times \mathbf{C}''(m)|}{\|\mathbf{C}'(m)\|^3} \tag{9}$$

By substituting Eqs. 4 and 9 into Eq. 8, the centripetal acceleration constraint can be rewritten as the following function of parameter  $m$ :

$$a_c(m) = \frac{|\mathbf{C}'(m) \times \mathbf{C}''(m)|}{\|\mathbf{C}'(m)\|} \dot{m}^2 \leq \mu \tag{10}$$

Thus, the constraints of the feedrate, the axial acceleration, and the chord error are all transformed into the functions of parameter velocity and parameter acceleration. Therefore, MTTP with these three constraints is formulated as an optimal control problem about parameter, while multi-variable optimization problem is reduced to a problem where only one parameter variable needs to be optimized.

Imposing boundary conditions and system dynamic, the time optimal control problem can be formulated as

$$\begin{aligned} \min_{\dot{m}} \quad & T = \int_{t_0}^{t_n} 1 \, dt \\ \text{s.t.} \quad & \begin{cases} \frac{dm}{dt} = \dot{m}(t), \frac{d\dot{m}}{dt} = \ddot{m}(t) \\ m(t_0) = 0, m(t_n) = 1, \dot{m}(t_0) = \dot{m}_0, \dot{m}(t_n) = \dot{m}_n \\ \|\mathbf{C}'(m)\|^2 \dot{m}^2 \leq v_F^2 \\ \mathbf{a}_{\min} \leq \mathbf{C}'\ddot{m} + \mathbf{C}''\dot{m}^2 \leq \mathbf{a}_{\max} \\ \frac{|\mathbf{C}'(m) \times \mathbf{C}''(m)|}{\|\mathbf{C}'(m)\|} \dot{m}^2 \leq \mu \end{cases} \end{aligned} \tag{11}$$

where  $\dot{m}_0 = \frac{v_0}{\|\mathbf{C}'(0)\|}$ ,  $\dot{m}_n = \frac{v_n}{\|\mathbf{C}'(1)\|}$  are constraints of the start point and the end point, respectively. In Eq. 11, system dynamic is linear while process constraints are nonlinear. Therefore, the formulation is actually a time optimal control problem for a linear system with constraints of nonlinear equality and inequality.

### 3 Solution of MTPP problem

Equation 11 is an optimal control of second-order dynamic system with nonlinear constraints. In addition, the final time is free. Parameter  $\dot{m}$  obviously becomes the state variable, and constraints are nonlinear functions of parameter velocity  $\dot{m}$ . Since constraints have strong nonlinearity, Eq. 11 is difficult to solve efficiently. To solve such a problem, it is best to convert the free time into fixed time by introducing a time scale factor. Then the free time optimal control problem is transformed to the corresponding fixed time one. As the parametric formula of the tool path is known, Eq. 11 can be mapped from the time domain  $t \in [t_0, t_n]$  to the parameter domain  $m \in [0, 1]$ , which can effectively weaken nonlinearity.

#### 3.1 Dynamic system model reduction

Since  $\dot{m} = \frac{dm}{dt}$ , the objective function of Eq. 11 can be rewritten as

$$\min J = \int_0^1 \frac{1}{\dot{m}} \, dm \tag{12}$$

Define new variables [19, 20]

$$\alpha(m) = \dot{m}^2, \beta(m) = \ddot{m} \tag{13}$$

Then, constraints of the feed, the axial accelerations, and the centripetal acceleration can be rewritten as the linear functions of  $(\alpha, \beta)$

$$\begin{cases} \|\mathbf{C}'(m)\|^2 \alpha \leq v_F^2 \\ \mathbf{a}_{\min} \leq \mathbf{C}'\beta + \mathbf{C}''\alpha \leq \mathbf{a}_{\max} \\ \frac{|\mathbf{C}'(m) \times \mathbf{C}''(m)|}{\|\mathbf{C}'(m)\|} \alpha \leq \mu \end{cases} \tag{14}$$

The following differential relations can also be derived as

$$\alpha'(m) = \frac{d\dot{m}^2}{dm} = 2\dot{m} \frac{d\dot{m}}{dm} = 2\dot{m} \frac{d\dot{m}}{dt} \frac{dt}{dm} = 2\ddot{m} = 2\beta(m) \tag{15}$$

Taking parameter acceleration  $\beta$  as the optimized variable, Eq. 11 can be reformulated as Eq. 16:

$$\begin{aligned} \min_{\beta} J = \int_0^1 \frac{1}{\sqrt{\alpha}} \, dm \\ \text{s.t.} \quad \begin{cases} \alpha'(m) = 2\beta(m), \alpha(0) = \alpha_0, \alpha(1) = \alpha_1 \\ \|\mathbf{C}'(m)\|^2 \alpha \leq v_F^2 \\ \mathbf{a}_{\min} \leq \mathbf{C}'\beta + \mathbf{C}''\alpha \leq \mathbf{a}_{\max} \\ \frac{|\mathbf{C}'(m) \times \mathbf{C}''(m)|}{\|\mathbf{C}'(m)\|} \alpha \leq \mu \end{cases} \end{aligned} \tag{16}$$

Equation 16 is actually the convex optimal control problem of the first-order linear system [20], which has much weaker nonlinearity and lower system order compared with Eq. 11. Because Eq. 16 is convex, local optimal solution for Eq. 16 is just the global one. Furthermore, the optimal solution of Eq. 16 is unique and maximum among all feasible solutions at any time [13, 21].

Although Eq. 16 can be efficiently solved by approaches based on gradient, we still want to find a more efficient method like linear programming [13]. Considering minimum processing time may need maximum feed profile, the optimal control Eq. 16 can be replaced with

$$\max_{\beta} J = \int_0^1 \|\mathbf{C}'(m)\|^2 v^2 \, dm = \int_0^1 \|\mathbf{C}'(m)\|^4 \alpha(m) \, dm \tag{17}$$

Note that constraints of Eq. 17 is omitted, because they are the same as that of Eq. 16. Actually, solution of Eq. 17 is just that of Eq. 16, which can be proved by theorem 1.

**Theorem 1** Equation 17 has the same and unique optimal solution with Eq. 16.

Before proving the theorem, two assumptions must be stated.

Assumption 1 The predefined path  $\mathbf{C}(m)$  is bounded and has a two-order continuous derivative.

Assumption 2 The predefined path  $\mathbf{C}(m)$  doesn't present singularity.

*Proof* Consider that  $\gamma^*(m)$  is the optimal solution of Eq. 16. Since Eq. 17 has the same constraints with Eq. 16,  $\gamma^*(m)$  is also a feasible solution to Eq. 17. Let  $R^n$  be the space of all the feasible solutions. Since the optimal solution is unique and maximum among all feasible solutions, so as to an arbitrary  $\gamma_f(m) \in R^n$ , there is  $\gamma^*(m) \geq \gamma_f(m)$ ,  $m \in [0, 1]$ . Then, with regard to the objective function of Eq. 17, there is

$$\int_0^1 \|\mathbf{C}'(m)\|^4 \gamma^*(m) \, dm > \int_0^1 \|\mathbf{C}'(m)\|^4 \gamma_f(m) \, dm \tag{18}$$

As a consequence,  $\gamma^*(m)$  is also the optimal solution of Eq. 17, and the theorem is proved. Theorem 1 allows to conclude that minimum time feedrate trajectory can be generated by solving a simpler linear optimal control problem. In the next section, through discretizing the state and controlled variables, Eq. 17 can be rewritten as the standard linear programming formulation.  $\square$

### 3.2 Linear programming–based formulation

An effective method of numerical solution, namely direct transcription method, which discretizes both the state and controlled variables, may be proper for Eq. 17. Since the system dynamic of Eq. 17 is a first-order linear system, radial basis functions can be applied to parameterize the state variable [22]. The state variable can be approximated by

$$\alpha(m) \approx \hat{\alpha}(m) = \sum_{i=1}^L N_i(m)\xi_i \tag{19}$$

where  $N_i(m) = \exp(-(m - \bar{m}_i)^2/2\delta^2) > 0$  is the  $i$ th radial basis function, and  $\xi_i$  is the  $i$ th weight coefficient. The scope of each radial basis function can be restricted by the function center  $\bar{m}_i$  and the width parameter  $\delta$ . Substituting Eq. 19 into Eq. 15, the control variable can be derived as

$$\beta(m) = \hat{\beta}(m) = \frac{1}{2} \sum_{i=1}^L N'_i(m)\xi_i \tag{20}$$

where  $N'_i(m) = -\frac{(m-\bar{m}_i)}{\delta^2} \exp\left(-\frac{(m-\bar{m}_i)^2}{2\delta^2}\right)$ .

Function centers of radial basis functions may or may not coincide with grid points, and this paper assumes that they are the same. The grid sequences can be described as

$$\mathbf{m} = [\bar{m}_1, \dots, \bar{m}_L] \tag{21}$$

where  $0 < \bar{m}_1 < \bar{m}_2 < \dots < \bar{m}_L < 1$ . Because the grid points are uniformly distributed, each interval between two adjacent grids is equal to  $\Delta m = 1/(L + 1)$ . All the constraints including path constraints are estimated at grid points. This method is known as point-wise discretization which has been proved convergent [23]. Meanwhile, the system dynamic is eliminated.

Take the weight coefficient vector  $\xi = [\xi_1, \dots, \xi_L]^T$  as optimized variables, the maximum objective function of Eq. 17 can be deduced as

$$\begin{aligned} \max J &\approx \sum_{j=1}^L \left( \left\| \mathbf{C}'(\bar{m}_j) \right\|^4 \sum_{i=1}^L N_i(\bar{m}_j)\xi_i \Delta m \right) \\ &= \sum_{j=1}^L \left( \sum_{i=1}^L \left\| \mathbf{C}'(\bar{m}_j) \right\|^4 \Delta m N_i(\bar{m}_j)\xi_i \right) \\ &= \mathbf{F}\xi \end{aligned} \tag{22}$$

where

$$\begin{cases} \mathbf{F} = [F_1 \ F_2 \ \dots \ F_L] \\ F_i = \sum_{j=1}^L \left\| \mathbf{C}'(\bar{m}_j) \right\|^4 \Delta m N_i(\bar{m}_j) \end{cases}$$

The feedrate constraint can be approximated as  $\mathbf{B}\xi \leq v_F^2 \mathbf{I}$ , where

$$\begin{cases} \mathbf{B} = [\mathbf{B}_1 \ \dots \ \mathbf{B}_L]^T \\ \mathbf{B}_j = d(\bar{m}_j) [N_1(\bar{m}_j) \ \dots \ N_L(\bar{m}_j)] \\ d(\bar{m}_j) = \left\| \mathbf{C}'(\bar{m}_j) \right\|^2 \end{cases}$$

Similarly, the centripetal acceleration constraint can also be written as  $\mathbf{H}\xi \leq \mu \mathbf{I}$ . Since two axes have the same form of acceleration constraint, we take the  $x$  axis for example. The approximation formula of  $x$  axis acceleration constraint is given as

$$a_{x \min} \mathbf{I} \leq \mathbf{A}_x \xi \leq a_{x \max} \mathbf{I} \tag{23}$$

where

$$\begin{cases} \mathbf{A}_x = [\mathbf{A}_{x1} \ \mathbf{A}_{x2} \ \dots \ \mathbf{A}_{xL}]^T \\ \mathbf{A}_{xj} = \begin{bmatrix} x''(\bar{m}_j)N_1(\bar{m}_j) + \frac{1}{2}x'(\bar{m}_j)N'_1(\bar{m}_j) \\ \vdots \\ x''(\bar{m}_j)N_L(\bar{m}_j) + \frac{1}{2}x'(\bar{m}_j)N'_L(\bar{m}_j) \end{bmatrix}^T \end{cases}$$

In the same way, the constraint on  $y$  axis can be obtained as

$$a_{y \min} \mathbf{I} \leq \mathbf{A}_y \xi \leq a_{y \max} \mathbf{I} \tag{24}$$

Until now all constraints have been converted to linear functions of  $\xi$ , and Eq. 17 is easily reformulated to the standard form of linear programming

$$\begin{aligned} &\max_{\xi} \mathbf{F}\xi \\ &\text{s.t.} \begin{cases} a_{x \min} \mathbf{I} \leq \mathbf{A}_x \xi \leq a_{x \max} \mathbf{I} \\ a_{y \min} \mathbf{I} \leq \mathbf{A}_y \xi \leq a_{y \max} \mathbf{I} \\ \mathbf{B}\xi \leq v_F \mathbf{I} \\ \mathbf{H}\xi \leq \mu \mathbf{I} \end{cases} \end{aligned} \tag{25}$$

where  $\mathbf{I} \in \mathbb{R}^{L \times 1}$  is unit column vector,  $\mathbf{A}_x \in \mathbb{R}^{L \times L}$ ,  $\mathbf{A}_y \in \mathbb{R}^{L \times L}$ ,  $\mathbf{B} \in \mathbb{R}^{L \times L}$  and  $\mathbf{H} \in \mathbb{R}^{L \times L}$  denote Jacobian matrices with the sparse structure.

Solving Eq. 25 with linear programming, the optimal coefficient vector  $\xi_o$  is obtained. Substituting  $\xi_o$  into Eq. 19, the optimal parameter speed at any parameter can be estimated. Therefore, the optimal feedrate profile is easily gained according to Eq. 4. This feedrate profile needs to be interpolated to generate reference positions for servo controllers. But the feedrate profile is a function of parameter, not the time. The interpolation methods developed for this kind of feedrate either unstable or time consuming. Moreover, jerk constraints are not considered. Hence, the feedrate gained by solving MTTP is taken as an initial feedrate, since it is the maximum in the feasible region and has ‘‘bang-bang’’ structures [13, 17, 21]. we re-plan the feedrate on time domain considering jerk constraints in next section.

## 4 Feedrate re-planning and NURBS interpolation

In this section, the JMEI method is presented to re-plan the feedrate, which can simultaneously consider the influence of acceleration and jerk. Moreover, the “ripple effect” is eliminated.

### 4.1 Feedrate profile of bell shape

If the feedrate profile has discontinuity during the processing, dents and defects will appear on the surface of workpieces. So the feedrate profile needs to be smoothed, and bell-shape profile are proved to be efficient to limit

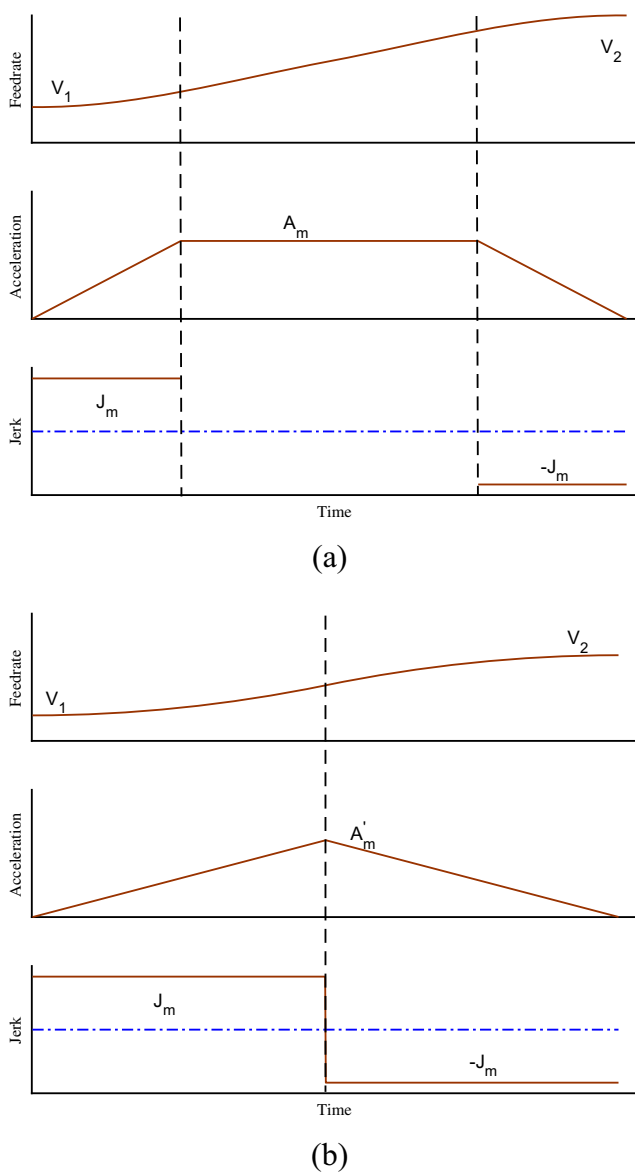


Fig. 2 The feedrate, acceleration, and jerk of acceleration segment: **a** three substages and **b** two substages

the acceleration and jerk within the abilities of the machine tool. A complete bell-shape profile covers the acceleration segment, the constant speed segment, and the deceleration segment. Three segments consist of seven substages at most. Here, the acceleration segment is taken as an example, which is divided into three substages or two substages. Three sub-stages are increasing acceleration stage, constant acceleration stage, and decreasing acceleration stage, as shown in Fig. 2a.  $V_1$  and  $V_2$  denote the starting speed and terminal speed, respectively.  $A_m$  is the maximum acceleration and  $J_m$  is the maximum jerk. If  $|V_1 - V_2| \leq \frac{A_m^2}{J_m}$ , there is no constant acceleration substage, and the acceleration segment becomes what is shown in Fig. 2b. And  $A_m' = \sqrt{|V_1 - V_2|J_m}$  denotes the maximum acceleration which can be achieved during the acceleration segments, satisfying  $A_m' \leq A_m$ .

### 4.2 Proposed feedrate scheduling algorithm

The proposed feedrate scheduling algorithm makes sure that the acceleration and the jerk are limited within allowable ranges, and meanwhile considers the maximum initial feedrate obtained from MTTP. Moreover, the “ripple effect” shown in Fig. 3 also is handled, which represents the overlap between adjacent sharp corners. A sharp corner represents a local minimum point of speed on the feedrate profile. The local minimum points of speed are the results of reducing the speed greatly to satisfy kinds of constraints when the MTTP is solved. Generally, the larger command feedrate is given, the “ripple effect” more easily appears. The reason can be stated as follows. On the one hand, the higher command feedrate leads to more sharp corners on the initial feedrate profile. On the other hand, the distance of two adjacent sharp corners along the tool path becomes smaller when higher command feedrate is required. And the distance is not long enough to complete the feedrate planning of the bell-shape profile. We divide the sharp corners into three types according to the severity of “ripple effect”, that

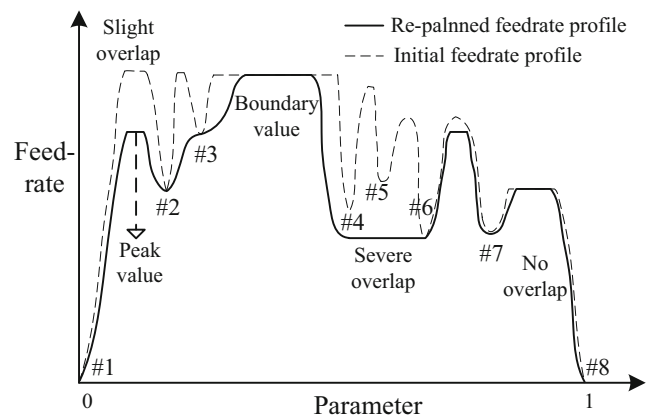


Fig. 3 The overlapping phenomenon

is, the severe overlap (such as corner 4, 5, and 6), the slight overlap (such as corner 1 and 2), and no overlap (such as corner 7 and 8).

Therefore, the JMEI searches the sharp corners, distinguishes the types of “ripple effect,” and re-plans the feedrate near them. To realize these functions, the JMEI method is composed of three main steps: the severe overlap detection, the slight overlap detection, and no overlap replanning.

When the cutter encounters the corners with severe overlap, it cannot even accelerate or decelerate from one sharp corner to another adjacent one according to Fig. 2. The sharp corners involved in severe overlap needs to be combined together through reducing the speed at each corner to the minimum value of all. Algorithm 1 is presented to detect the severe overlap corners and to merge them.

---

**Algorithm 1** Detect and merge sharp corners with severe overlap.

---

**Input:** The initial feedrate at grid points obtained by solving the MTTP with LP. **Output:** The feedrate at grid points without severe overlap.

1. Detect the sharp corners. The feedrate  $V_{l,i}$  at sharp corners satisfy  $V_{l,i-1} > V_{l,i}$  and  $V_{l,i+1} > V_{l,i}$ . Store the speed at sharp corners in an array  $\mathbf{V}_l$ . The corresponding parameters are stored in an array  $\mathbf{m}_l$ .
2. Suppose that there are  $K + 1$  sharp corners. Calculate the arc length between each two adjacent sharp corners with numerical integration method such as Simpson’s method or trapezoidal method. Store the arc length sequences into the array  $\mathbf{P} = [P(0, 1), \dots, P(j, j + 1), \dots, P(K - 1, K)]$ , where  $P(j, j + 1)$  denotes the arc length between the  $j$ th and  $(j + 1)$ th sharp corners.
3. Compute the minimum distance. If the feedrate is planned according to the method in the Section 4.1, a minimum distance is required. The minimum distance between  $j$ th and  $(j + 1)$ th sharp corners can be computed as follows,

$$D_{\min}(j, j + 1) = \begin{cases} \frac{V_{l,j+1} + V_{l,j}}{2} \left( \frac{A_m}{J_m} + \frac{|V_{l,j+1} - V_{l,j}|}{A_m} \right), & |V_{l,j+1} - V_{l,j}| > \frac{A_m^2}{J_m} \\ (V_{l,j+1} + V_{l,j}) \sqrt{\frac{|V_{l,j+1} - V_{l,j}|}{J_m}}, & \text{otherwise} \end{cases} \tag{26}$$

where  $V_{l,j}$  denotes the speed of  $j$ th sharp corner and  $j \in \{0, 1, \dots, K\}$ .

4. Detect and merge the severe overlap. Check if the condition  $D_{\min}(j, j + 1) < P(j, j + 1)$  is true. If it is true, let  $V_{l,j} = V_{l,j+1} = \min(V_{l,j}, V_{l,j+1})$ . Then re-evaluate points  $\{0, \dots, j - 1\}$  recursively. If it is not true, proceed with the next sharp corner until all corners are checked.

After applying algorithm 1 to the initial feedrate, the sharp corners involved in severe overlap are detected and merged. For example, corners 4, 5, and 6 are merged into one corner. Hence, the number of sharp corners drop to  $M + 1$ . The new arc length array  $\mathbf{P}'$  is easily obtained from  $\mathbf{P}$  by simple addition operation. Let  $\mathbf{V}'_l$  be the new feedrate array at  $M + 1$  sharp corners. And the corresponding array of parameters is stored in  $\mathbf{m}'_l$ . As a result, the slight overlap corners and no overlap corners are left. The slight overlap generally consists of two adjacent sharp corners. Take corners 1 and 2 for example, the cutter can accelerate to a certain peak feedrate from corner 1 and then decelerate to corner 2. But the peak feedrate of the cutter cannot reach the boundary value. The peak feedrate attainable needs to be searched. Algorithm 2 is proposed to find the peak feedrate.

---

**Algorithm 2** Detect the slight overlap corners and search the peak feedrate attainable between sharp corners involved in the slight overlap.

---

**Input:** The feedrate at grid points which have been processed with algorithm 1. **Output:** The feedrate at grid points without slight overlap.

1. Search the boundary values. Find the maximum speed as a boundary value from speeds of grid points between every two adjacent sharp corners. Store the maximum speed in an array  $\mathbf{F}_m$ . The corresponding parameters are stored in an array  $\mathbf{m}_B$ .
2. Compute the distance accelerating from the speed of  $j$ th sharp corner to the  $j$ th boundary value with following formula,

$$S_{a\min}(j) = \begin{cases} \frac{V'_{l,j} + F_{m,j}}{2} \left( \frac{A_m}{J_m} + \frac{|F_{m,j} - V'_{l,j}|}{A_m} \right), & |F_{m,j} - V'_{l,j}| > \frac{A_m^2}{J_m} \\ (F_{m,j} + V'_{l,j}) \sqrt{\frac{|F_{m,j} - V'_{l,j}|}{J_m}}, & \text{otherwise} \end{cases} \tag{27}$$

where  $F_{m,j}$  is the  $j$ th boundary value and  $j \in \{0, 1, \dots, M - 1\}$ . The distance decelerating from the  $j$ th boundary value to the speed of the  $(j + 1)$ th sharp corner can be computed by slightly modifying Eq. 27. Let  $S_{d\min}(j)$  be the decelerating distance.

3. Detect the slight overlap and determine the peak feedrate. Test if  $S_{a\min}(j) + S_{d\min}(j) > P(j, j + 1)'$  is satisfied. If it is true, the current boundary value cannot be achieved. Search a new peak feedrate to replace the boundary value  $F_{m,j}$  with the bisection searching method. If it is not true, keep the boundary value unchanged and proceed with the next boundary value. After all the boundary value are evaluated, the proper boundary value array  $\mathbf{F}'_m$  is gained.

After applying algorithm 2, slight overlap corners have been transformed into no overlap. No overlap means the cutter can reach the boundary value when motions between two adjacent sharp corners. The proper boundary value array  $\mathbf{F}_m'$  has been determined by algorithm 2. Until now, for the two adjacent  $j$ th and  $(j + 1)$ th sharp corners, we have known the acc/dec structures of motion, starting feedrate, terminal feedrate, arc length, and boundary value between them. When we re-plan the feedrate according to the method in Section 4.1, the mathematical expression of Fig. 2 can be obtained through solving equations [24, 25]. The re-planned feedrate profile is a function of time, which is convenient to adopt in real-time interpolation.

### 4.3 Interpolation with feed correction polynomial

Since some NURBS curves are not parameterized by arc length, the mapping relationship between NURBS parameter and the displacement of cutter tip along the curve is not accurate, and feedrate fluctuations appear [26].

In order to avoid the fluctuations of feedrate, feed correction polynomial is employed for interpolation [27]. In view of a non-arc-length parameterized NURBS curve, the adaptive quadrature method [28] are used to split the curve. As a result, a set of  $(m_i, S_i)$  pairs is obtained, where  $m_i$  is the parameter and  $S_i$  is the arc length between 0 and  $m_i, i = 1, 2, \dots, N$ . Different with previous methods, we go on calculating  $\tilde{S}_j$  which is the arc length from 0 to  $\tilde{m}_j$ , where

$$\tilde{m}_j = \frac{m_k + m_{k+1}}{2} \quad j, k \in 1, 2, \dots, N - 1 \quad (28)$$

Hence, we have another set of  $(\tilde{m}_j, \tilde{S}_j)$  pairs, which will be used to test the feed correction polynomial. Then, the arc length is normalized with the following formula:

$$\lambda_i = \frac{S_i - S_0}{S_N - S_0} \quad (29)$$

These pairs of  $(m_i, \lambda_i)$  are used to evaluate the coefficients of the 7-order polynomial. Suppose the polynomial  $f(\lambda)$  is approximated as

$$f(\lambda) = \tau_0\lambda^7 + \tau_1\lambda^6 + \tau_2\lambda^5 + \tau_3\lambda^4 + \tau_4\lambda^3 + \tau_5\lambda^2 + \tau_6\lambda + \tau_7 \quad (30)$$

The first and second derivatives with respect to  $S$  are deduced as

$$\begin{cases} \frac{df}{dS} = \frac{7\tau_0\lambda^6 + 6\tau_1\lambda^5 + 5\tau_2\lambda^4 + 4\tau_3\lambda^3 + 3\tau_4\lambda^2 + 2\tau_5\lambda + \tau_6}{S_N - S_0} \\ \frac{d^2f}{dS^2} = \frac{42\tau_0\lambda^5 + 30\tau_1\lambda^4 + 20\tau_2\lambda^3 + 12\tau_3\lambda^2 + 6\tau_4\lambda + 2\tau_5}{(S_N - S_0)^2} \end{cases} \quad (31)$$

Hence, the parameter value of a NURBS curve  $m$  could be predicted with Eq. 30, and the matrix form is written as

$$\hat{\mathbf{m}} = \begin{bmatrix} 0 \\ \widehat{m}_1 \\ \widehat{m}_2 \\ \vdots \\ \widehat{m}_N \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & \dots & 1 \\ \lambda_1^7 & \lambda_1^6 & \lambda_1^5 & \dots & 1 \\ \lambda_2^7 & \lambda_2^6 & \lambda_2^5 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}}_{\Theta} \cdot \underbrace{\begin{bmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_6 \\ \tau_7 \end{bmatrix}}_{\tau} \quad (32)$$

where  $\hat{\mathbf{m}}$  denotes the predicted parameter value,  $\Theta$  the regression matrix, and  $\tau$  the coefficient vector. Then, the vector of prediction errors is expressed as

$$\mathbf{e} = \mathbf{m} - \hat{\mathbf{m}} \quad (33)$$

In order to gain the optimal  $\tau$ , the quadratic cost function is defined as

$$J = \frac{1}{2} \mathbf{e}^T \mathbf{e} = \frac{1}{2} (\mathbf{m} - \Theta\tau)^T (\mathbf{m} - \Theta\tau) \quad (34)$$

As to the NURBS curve, the differentials of arc length with respect to parameter are deduced as

$$\begin{cases} \frac{dS}{dm} = \sqrt{x'(m)^2 + y'(m)^2 + z'(m)^2} \\ \frac{d^2S}{dm^2} = -\frac{x'(m)x''(m) + y'(m)y''(m) + z'(m)z''(m)}{x'(m)^2 + y'(m)^2 + z'(m)^2} \end{cases} \quad (35)$$

Through imposing the boundary conditions at start and end points, the continuity between connecting segments can be ensured. Therefore, the derivatives at the boundary obtained with Eq. 35 should be equal to what are evaluated with Eq. 31. Let  $m^b$  and  $m^e$  denote the parameters of

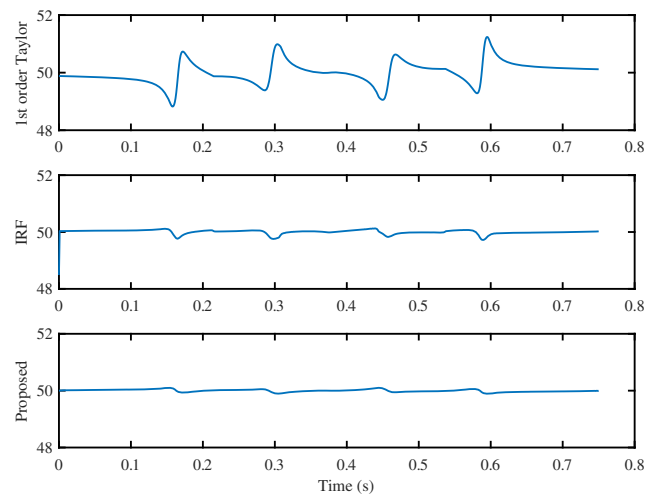
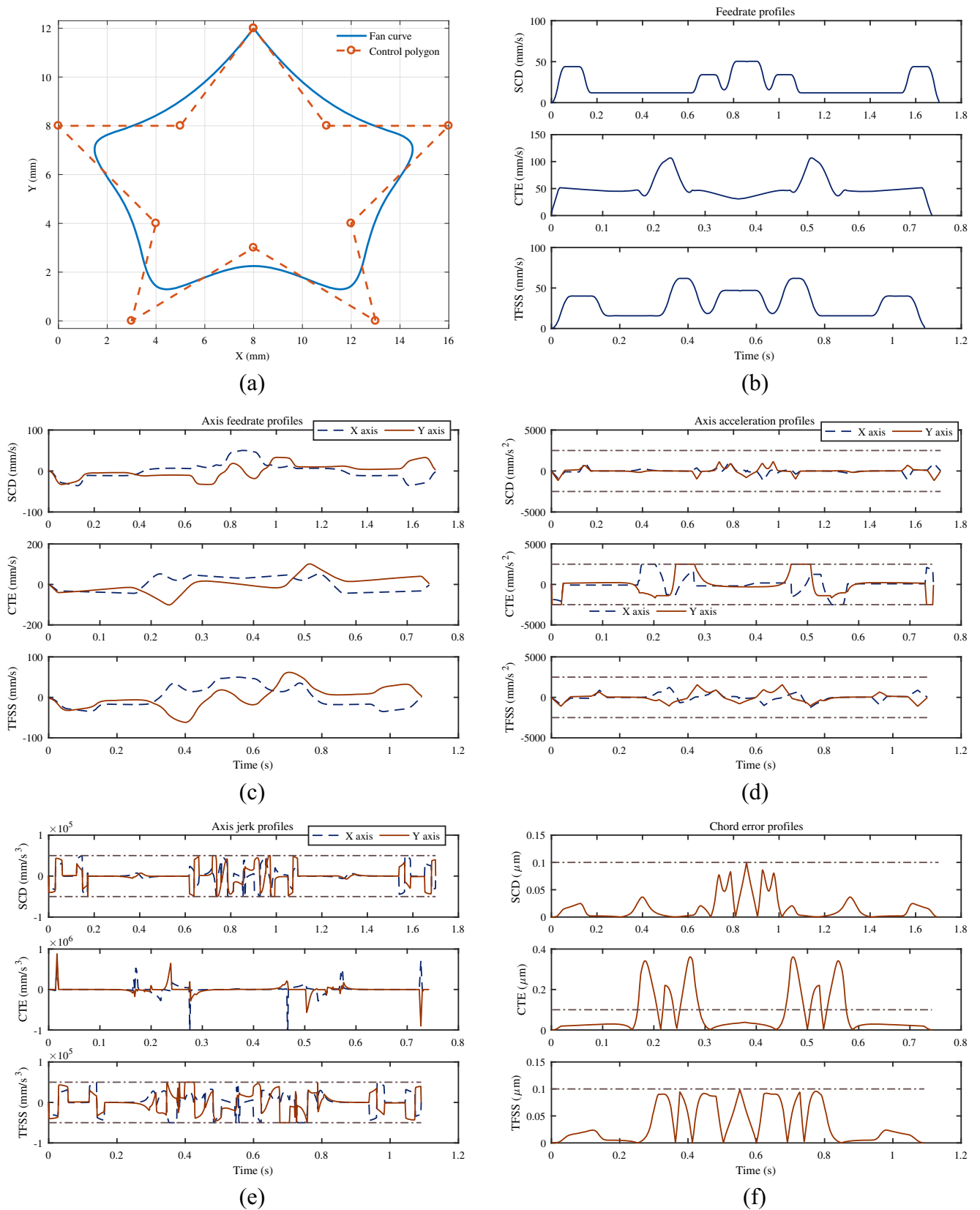


Fig. 4 Comparison of three interpolation techniques





**Fig. 5** Simulation results of star curve with SDC, CTE, and TFSS. **a** Star curve. **b** Resultant feedrate profiles. **c** Axial feedrate profiles. **d** Axial acceleration profiles. **e** Axial jerk profiles. **f** Chord error profiles

**Table 1** Performance comparison for SCD, CTE, and TFSS for the star curve

Method	SCD	CTE	TFSS
Planning time	0.2 s	0.5 s	0.6 s
Machining time	1.702 s	0.743 s	1.093 s
Mean contour error	20.1 μm	28.3 μm	22.9 μm
Maximum contour error	59.3 μm	74.2 μm	60.6 μm

boundary points. Then, the following expressions can be obtained:

$$\underbrace{\begin{bmatrix} m^b \\ (S_N - S_0)m_s^b \\ (S_N - S_0)^2 m_{ss}^b \\ m^e \\ (S_N - S_0)m_s^e \\ (S_N - S_0)^2 m_{ss}^e \end{bmatrix}}_{\zeta} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ 42 & 30 & 20 & 12 & 6 & 2 & 0 & 0 \end{bmatrix}}_{\Gamma} \cdot \underbrace{\begin{bmatrix} \tau_0 \\ \tau_1 \\ \vdots \\ \tau_6 \\ \tau_7 \end{bmatrix}}_{\tau} \quad (36)$$

where

$$\begin{cases} m_s^b = \frac{dS}{dm} |_{m=m^b}, m_{ss}^b = \frac{d^2S}{dm^2} |_{m=m^b} \\ m_s^e = \frac{dS}{dm} |_{m=m^e}, m_{ss}^e = \frac{d^2S}{dm^2} |_{m=m^e} \end{cases} \quad (37)$$

The constrained optimization problem is constructed by minimizing the cost function Eq. 34 and taking Eq. 36 as the constraint. The method of multipliers is applied to solve the problem, and the following linear equation system is gained:

$$\begin{bmatrix} \Theta^T \Theta & \Gamma^T \\ \Gamma & \mathbf{0} \end{bmatrix} \times \begin{bmatrix} \tau \\ \Lambda \end{bmatrix} = \begin{bmatrix} \Theta^T \mathbf{m} \\ \zeta \end{bmatrix} \quad (38)$$

where  $\Lambda$  is the multiplier vector. The linear equation system can be easily solved, and coefficients of the polynomial is included in vector  $\tau$ .

One feed correction polynomial may not reflect the relationship between the parameter and the arc length accurately. If one polynomial can not fit all  $(m_i, S_i)$  pairs, the multiple 7th-order polynomials are adopted to improve the accuracy [4]. In this section, whether multiple polynomials are adopted or not depends on the difference between the predictive value  $\hat{m}_j$  and the test value  $\tilde{m}_j$ . The difference can be computed with

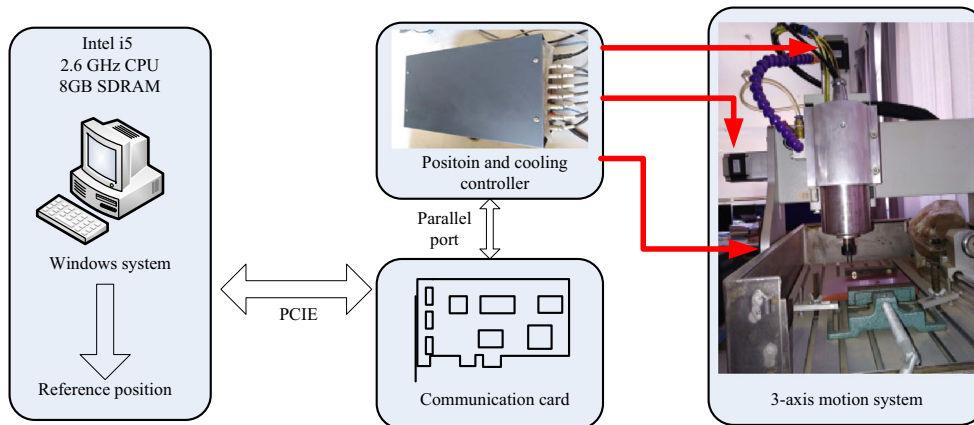
$$\sum_{j=1}^{N_{sub}} \frac{|\tilde{m}_j - \hat{m}_j|}{N_{sub}} \leq \varepsilon \quad (39)$$

where  $N_{sub}$  is the total numbers of  $(\tilde{m}_j, \tilde{S}_j)$  pairs and  $\varepsilon$  is the tolerance. If Eq. 39 is not satisfied, the  $(m_i, S_i)$  pairs are divided into two sets of the same size, so is the  $(\tilde{m}_j, \tilde{S}_j)$  pairs. Then, the fitting and testing steps above are implemented again. The subdivision terminates when Eq. 39 is satisfied or  $(m_i, S_i)$  pairs are equal to the order of polynomial.

Figure 4 shows the comparison of three interpolation method: the first-order Taylor approximation, the improved robust fast interpolator [5], and the proposed interpolation method. The maximum fluctuation ratio [29, 30] of the three methods are 2.48%, 0.25%, and 2%, respectively. From Fig. 4, the presented method get the minimal fluctuations.

### 5 Simulation and experimental examples

Two curves are used to evaluate the performance of the proposed method. One is a star pattern for simulation. The simulation is performed by MATLAB installed on a PC. The PC is equipped with 4G memory and AMD A8 processor. The other is a fan pattern for experiment. The experiment is implemented on a typical 3-axis CNC machine.



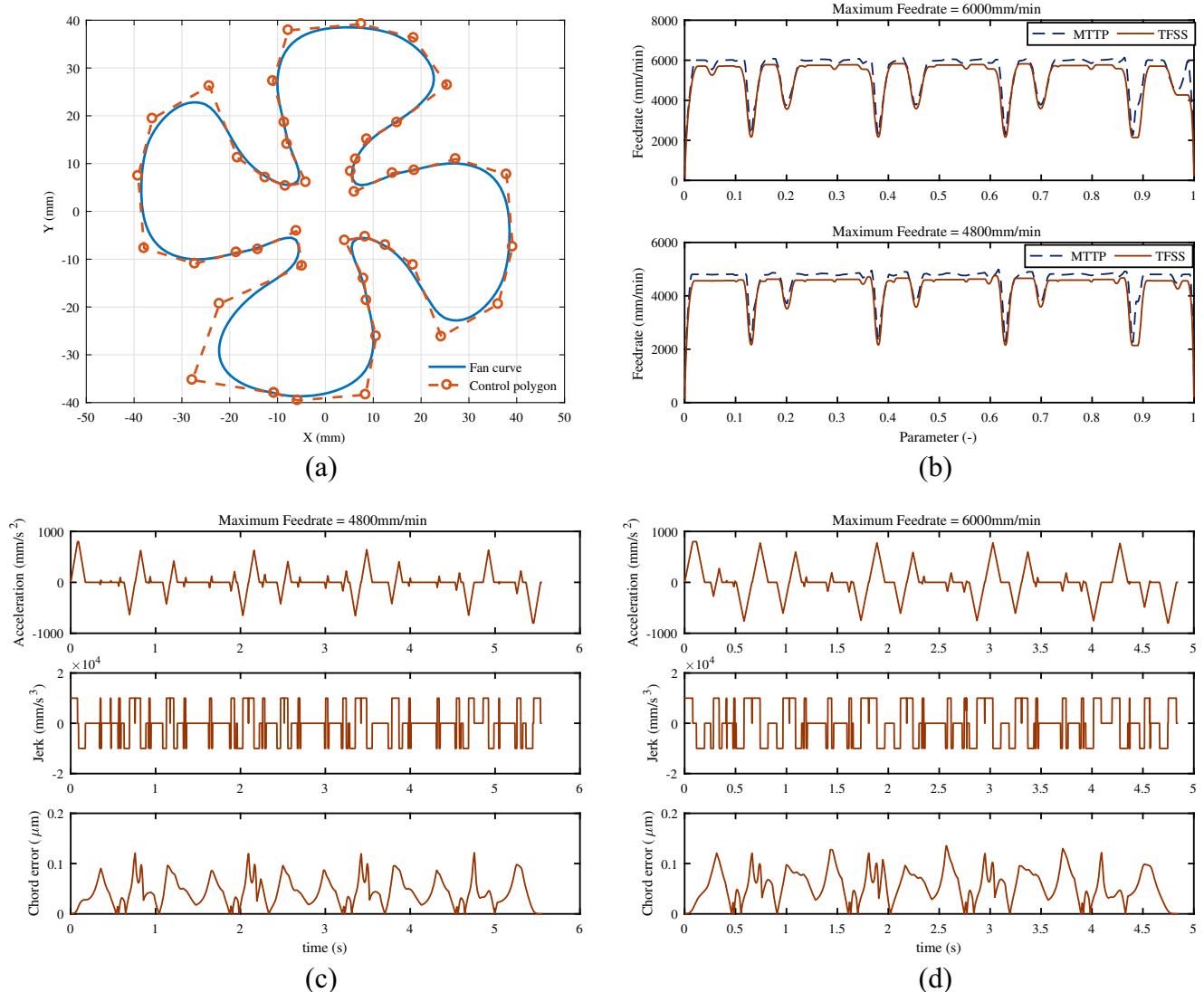
**Fig. 6** The experimental platform

### 5.1 Simulation results for star curve

The star curve is a single NURBS curve with five control points as shown in Fig. 5a. In this example, the bound of feedrate is set to 500 mm/s and the tolerance of chord error is set to 0.1  $\mu\text{m}$ . The maximum acceleration/deceleration of each axis is 2500  $\text{mm/s}^2$ , and the maximum jerk is 50000  $\text{mm/s}^3$ . The tracking error bound for each axis is set to 0.05 mm. The contour error bound is set to 0.05mm, too. For comparison, the feedrate is generated with three methods: sharp corner detection (SCD) [6], confined tracking error (CTE) [13], and the proposed scheme (TFSS).

Figure 5b shows the feedrate profiles planned by the three methods. Compared with the profile obtained by CTE,

SCD, and TFSS can both get smoother feedrate which is favorable to improve processing quality. Figure 5c shows the profiles of axial feedrates, and the axial acceleration profiles are plotted in Fig. 5d. The accelerations of CTE always reach bounds, while acceleration profiles of the other two methods don't. That is because CTE does not consider jerk limits but tracking errors which are easily transformed into linear constraints. Hence, the jerk profiles of CTE exceed the boundary value as shown in Fig. 5e. The jerk profiles of SCD and TFSS are bounded, since the jerk-limited module is introduced. The profiles of chord errors are evaluated according to Eq. 7, which are shown in Fig. 5f. The chord error profile of CTE is still beyond the boundary value because the chord error constraint is not considered. The chord error profile of SCD far below the boundary



**Fig. 7** Experimental results of fan curve based on TFSS when the maximum feedrate is set to 4800 mm/min and 6000 mm/min. **a** Fan curve. **b** Resultant feedrate profiles. **c** Profiles of resultant acceleration,

resultant jerk, and chord error when the maximum feedrate is 4800 mm/min. **d** Profiles of resultant acceleration, resultant jerk, and chord error when the maximum feedrate is 6000 mm/min

value, since the conservative feed, is adopted. From these figures, the TFSS scheme has the best performance.

Other comparison of performances are all listed in Table 1. Both the planning time and machining time are listed. The whole time spent on machining and planning by the three methods are 1.902 s, 1.243 s, and 1.693 s. The CTE method spends shortest time on machining. That is because it has highest feed owing to the unlimited jerks. The SCD method takes the least planning time owing to no optimization progress. Therefore, the feed of SCD is not optimized, which leads to the conservative feed and long machining time. Comparing with the other two methods, the TFSS takes both processing time and planning time into account. Certainly, we can further decrease the planning time by reducing the grid points. But too few grid points may lead to incorrect results. Of course, adaptive strategy can be used to obtain ideal grid number. In the simulation, 252 grids are adopted for both the CTE and TFSS. In addition, the contour errors of three methods are also estimated according to Zhang et al. [17]. The contour errors of SCD (without dynamics module) and TFSS are very similar. They can all get the smaller contour error than CTE.

## 5.2 Experimental results for fan curve

The experiment is implemented on a typical 3-axis CNC machine, which was updated to meet the needs of implementing new algorithms. The structure of the experimental platform is illustrated in Fig. 6. The reference positions for each axis is generated by a PC with a MATLAB environment. Then, these positions are sent to the motion controller equipped with PID algorithm through a parallel communication card installed in a PCIE slot. In this experiment, the maximum axial acceleration of the platform is set to  $800 \text{ mm/s}^2$  and the maximum axial jerk is set to  $1000 \text{ mm/s}^3$ . The chord error tolerance is set to  $1 \mu\text{m}$ . And the real-time machining is not considered, all the reference positions are generated by MATLAB off-line. The processed curve is a fan-shape pattern which has 45 control points as shown in Fig. 7a.

In order to fully evaluate the effectiveness of the proposed scheme, the command feedrate is set to 4800 mm/min and 6000 mm/min in succession. The resultant feedrate profiles of two different command speeds are represented with solid lines in Fig. 7b, where MTTP constraints are indicated with dashed lines. The machining time for two command feedrate is 5.554 s and 4.847 s, respectively. The lower command feedrate takes 14.5% more time than a higher one. Figure 7 c and d show resultant accelerations, resultant jerks, and chord errors of two different command feedrates, respectively. In the experiment, the dynamics of axes are not identified. Hence, the chord error estimated is applied for

investigating the accuracy. The chord errors for two command feedrates are far lower than the required tolerance. The maximum chord error for lower feedrate is reduced 10.3% than that for a higher one. They are  $0.122 \mu\text{m}$  and  $0.136 \mu\text{m}$ , respectively. So the feedrate should be chosen based on the trade off between machining time and precision.

## 6 Conclusion

In this paper, a complete scheme of smooth minimum time trajectory planning is proposed, and the numerical simulation and the experiment are employed to verify the effectiveness and practicability of the proposed scheme. Initially, the MTTP problem for curve machining is discretized with the direct transcription method, and then the discrete MTTP problem is transformed into linear programming by employing a proper objective function. The linear programming form of MTTP could greatly improve the efficiency of solving MTTP. Moreover, jerk constraints are not considered in MTTP, which further reduce the computation cost. To smooth the feedrate profile, the jerk-limited method of eliminating interaction (JMEI) is applied to re-plan the feedrate. Since JMEI is introduced, the planning problem in the parameter domain is transferred to that in the time domain. A bridge between optimal programming in the parameter domain and real-time interpolation is built by this transferring process. Therefore, the interpolation of a feed correction polynomial can be used to minimize the fluctuations. Finally, results of simulation and experiment demonstrate that the proposed scheme can balance time consumed and precision.

**Funding information** This work is supported by the National Natural Science Foundation of China (grant number 61573378). This work is also partially supported by the Key Technologies R&D Program of Henan province (grant number 182102210197).

## References

- 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 Manuf Technol* 77(9-12): 1889–1905
- Cheng MY, Tsai MC, Kuo JC (2002) Real-time NURBS command generators for CNC servo controllers. *Int J Mach Tools Manuf* 42(7):801–813
- Erkorkmaz K, Altintas Y (2005) Quintic spline interpolation with minimal feed fluctuations. *J Manuf Sci Eng* 127(2):339–349
- Heng M, Erkorkmaz K (2010) Design of a NURBS interpolator with minimal feed fluctuations and continuous feed modulation capability. *Int J Mach Tools Manuf* 50(3):281–293
- Lei WT, Wang SB (2009) Robust real-time NURBS path interpolators. *Int J Mach Tools Manuf* 49(7-8):625–633

6. 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
7. Sun Y, Bao Y, Kang K (2013) An adaptive feedrate scheduling method of dual nurbs curve interpolator for precision five-axis CNC machining. *Int J Adv Manuf Technol* 68(9-12):1977–1987
8. Rutkowski L, Przybył A, Cpałka K (2012) Novel online speed profile generation for industrial machine tool based on flexible neuro-fuzzy approximation. *IEEE Trans Ind Electron* 59(2):1238–1247
9. Gasparetto A, Lanzutti A, Vidoni R, Zanotto V (2012) Experimental validation and comparative analysis of optimal time-jerk algorithms for trajectory planning. *Robot Comput Integr Manuf* 28(2):164–181
10. Zhang Q, Li S, Guo J (2012) Smooth time-optimal tool trajectory generation for CNC manufacturing systems. *J Manuf Syst* 31(3):280–287
11. Lin F, Shen L, Yuan C, Mi Z (2018) Certified space curve fitting and trajectory planning for CNC machining with cubic B-splines. *Comput Aided Des* 106:13–29
12. Zhang Q, Li SR, Guo JX, Gao XS (2015) Tractable algorithm for robust time-optimal trajectory planning of robotic manipulators under confined torque. *International Journal of Computers, Communications and Control* 10(1):123–135
13. Guo J, Zhang Q, Gao XS, Li HB (2015) Time optimal feedrate generation with confined tracking error based on linear programming. *J Syst Sci Complex* 28(1):80–95
14. Liu H, Liu Q, Yuan S (2017) Adaptive feedrate planning on parametric tool path with geometric and kinematic constraints for cnc machining. *Int J Adv Manuf Technol* 90(5-8):1889–1896
15. Ye P, Zhang Y, Xiao J, Zhao M, Zhang H (2018) A novel feedrate planning and interpolating method for parametric toolpath in frenet-serret frame. *Int J Adv Manuf Technol* 101:1915–1925
16. Zhang Q, Li SR, Guo JX, Gao XS (2016) Time-optimal path tracking for robots under dynamics constraints based on convex optimization. *Robotica* 34(09):2116–2139
17. Zhang Q, Li SR (2013) Efficient computation of smooth minimum time trajectory for CNC machining. *Int J Adv Manuf Technol* 68(1-4):683–692
18. Chunming Y, Zhang K, Fan W (2013) Time-optimal interpolation for CNC machining along curved tool paths with confined chord error. *J Syst Sci Complex* 26(5):836–870
19. Verscheure D, Demeulenaere B, Swevers J, De Schutter J, Diehl M (2009) Time-optimal path tracking for robots: a convex optimization approach. *IEEE Trans Autom Control* 54(10):2318–2327
20. Shiller Z (1996) Time-energy optimal control of articulated systems with geometric path constraints. *IEEE International Conference on Robotics and Automation*, 1994. *Proceedings IEEE*, 1996:2680–2685, vol 4
21. Dong J, Stori JA (2006) A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization. *J Dyn Syst Meas Control* 128(2):725–739
22. Zhang Q, Li S, Guo J (2014) Mixed integer optimal control in minimum time multi-points traversal problem of robotic manipulators. *IFAC Proceedings Volumes* 47(3):7898–7903
23. Chen TWC, Vassiliadis VS (2005) Inequality path constraints in optimal control: a finite iteration  $\epsilon$ -convergent scheme based on pointwise discretization. *J Process Control* 15(3):353–362
24. 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
25. 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
26. Wu J, Zhou H, Tang X, Chen J (2012) A nurbs interpolation algorithm with continuous feedrate. *Int J Adv Manuf Technol* 59(5-8):623–632
27. Erkorkmaz K (2015) Efficient fitting of the feed correction polynomial for real-time spline interpolation. *J Manuf Sci Eng* 137(4)
28. Lei WT, Sung MP, Lin LY, Huang J (2007) Fast real-time NURBS path interpolation for CNC machine tools. *Int J Mach Tools Manuf* 47(10):1530–1541
29. Wu J, Zhou H, Tang X, Chen J (2012) A NURBS interpolation algorithm with continuous feedrate. *Int J Adv Manuf Technol* 59(5-8):623–632
30. Liu H, Liu Q, Zhou S, Li C, Yuan S (2015) A NURBS interpolation method with minimal feedrate fluctuation for CNC machine tools. *Int J Adv Manuf Technol* 78(5-8):1241–1250

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