IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received December 14, 2021, accepted December 20, 2021, date of publication December 23, 2021,
date of current version January 10, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3138049

Novel Feedrate Optimization Method for NURBS Tool Paths Under Various Constraints

TZYY-CHYANG LU[1], **(Member, IEEE), AND SHYH-LEH CHEN**[2,3], **(Member, IEEE)**
[1]Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan
[2]Department of Mechanical Engineering, National Chung Cheng University, Chiayi 621, Taiwan
[3]Advanced Institute of Manufacturing With High-Tech Innovations, National Chung Cheng University, Chiayi 621, Taiwan

Corresponding author: Tzyy-Chyang Lu (tclu@pu.edu.tw)

This work was supported in part by the Advanced Institute of Manufacturing with High-tech Innovations (AIM-HI) from The Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education (MOE) in Taiwan, and in part by the Ministry of Science and Technology, Taiwan, under Grant MOST 110-2218-E-194-009, Grant MOST110-2218-E-194-008, and Grant MOST 110-2634-F-194-003.

ABSTRACT A good feedrate profile for NURBS tool paths should be able to consider both machining efficiency and motion limits. In other words, the feedrate profile can lead to the machining time as small as possible and can satisfy various path and axis motion constraints. However, owing to the complicated relationship between trajectory motion and axes, it is difficult and time-consuming to check axis motion constraints (e.g., axis velocity, axis acceleration, axis jerk). In addition, the path length is not easy to calculate, so the machining time cannot be accurately estimated. This paper presents a feedrate profile planning method that can easily include, delete or organize the constraints and minimize the machining time for NURBS tool paths. The key idea is to use the same parameter u to represent both NURBS tool path and feedrate profile. When designing the feedrate profile with u, an analytical form of machining time can be obtained, and the constraints are converted into functions of u (including feedrate, tangential acceleration, tangential jerk, axis velocity, axis acceleration, axis jerk, chord error, and centripetal acceleration), which can be quickly checked. Particle swam optimization algorithm is employed to eliminate the solutions with long machining time or do not meet the constraints, and gradually optimize the feedrate profile. In the simulation, a star-shaped NURBS curve is selected to demonstrate the effectiveness of the proposed method. The results show that the proposed method can achieve not only smooth and high-speed machining under various constraints, but also high accuracy with the minimization of position error at the final interpolation point.

INDEX TERMS CNC machine tool, feedrate profile optimization, non-uniform rational B-spline (NURBS) interpolation, particle swarm optimization.

I. INTRODUCTION

Owing to its smooth properties and capability of generating free-form surfaces, the non-uniform rational B-splines (NURBS) has proven to be superior over linear and circular paths and become the standard format for computer-aided design (CAD) systems. To achieve a high-speed and high-precision computer numerical control (CNC) machining, the NURBS interpolator has been extensively studied, and many new interpolation algorithms have been proposed recently [1], [2]. There are two issues to realize a NURBS interpolator. One is to plan the feedrate profile, and the other is to generate accurate interpolation positions for each sampling time. So far, many efficient algorithms have been proposed for acquiring accurate interpolation positions according to a specified feedrate [3]–[5] or based on finite impulse response (FIR) filtering [6], [7]. Especially the FIR filtering-based algorithm can realize the real-time interpolation. The main concern of the present work is the planning of the feedrate profile, which is an important topic since it directly affects the machining efficiency and accuracy.

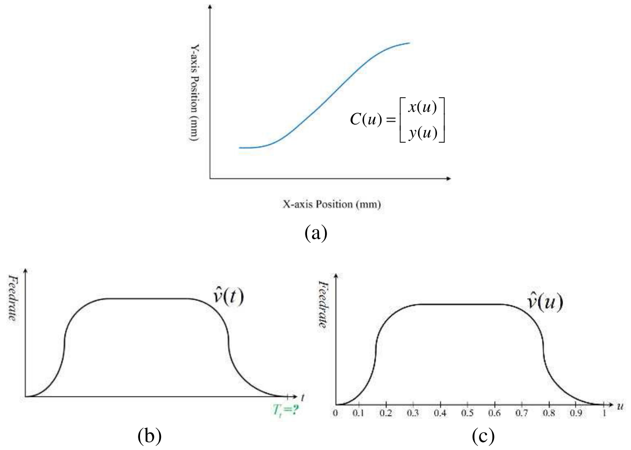
In general, it is preferred to have higher feedrate for machining efficiency. However, higher speed may easily lead to larger dynamic errors. In addition, there exist many motion limits on velocity, acceleration, and jerk due to physical constraints. A good feedrate profile should be able to consider both machining efficiency and motion limits. In other words, the feedrate profile can lead to the machining time as small as possible and can satisfy various path and axis motion constraints. However, owing to the complicated relationship between trajectory motion and axes, it is not easy to plan the speed under various constraints.

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenzhou Tang.

There have been many literatures on the feedrate scheduling of NURBS paths under various constraints. Gao *et al.* [8] applied a jerk-varied flexible feedrate scheduling method to slow down the acceleration near the points with local maximum curvature. Jia *et al.* [9] proposed a feedrate scheduling method with constant speed in the feedrate sensitive regions, which avoids frequent acceleration and deceleration in large curvature areas and guarantees geometric or drive constraints. Zhong *et al.* [10] developed a parametric curve interpolator that satisfies the chord error, feedrate limit, and axis velocity and acceleration limits. Ni *et al.* [11] considered the constraints of chord error, centripetal and tangential acceleration, and jerk. In addition, some feedrate profiles composed of S-curve [12] or bell-curve [13], [14] are used to meet acceleration and jerk limits along the path.

There are several problems in the above feedrate scheduling methods. One is the difficulty in changing the considered motion constraints. For example, constraints on feedrate, acceleration, and jerk along the path are considered in [12], but axis motion constraints will be difficult to be incorporated into the method proposed in [12]. Another problem is that to check the constraints (e.g., chord error, axis velocity, axis acceleration, axis jerk), all the interpolation positions along the tool path or on each axis need to be obtained. However, in the practical application, it is time-consuming and not feasible. Wang *et al.* [15] scanned the tool path to check the points with large curvature, and planned the speed of these points only. However, determining the exact position where the speed should be increased or decreased is still an open problem. Usually, integrals are used to estimate the distance for acceleration and deceleration (acc/dec). However, the numerical integration error may cause feedrate fluctuation [11], especially when the feedrate is high.

In addition to motion constraints, time optimization is another important issue for feedrate planning. For instance, Yuan *et al.* [16] developed a back and forward check algorithm under an acceleration constraint, which has higher feedrate or shorter operation time. Zhang *et al.* [17] proposed a greedy algorithm that enables at least one axis to reach its acceleration or jerk limit during machining. Afterwards, some other extended investigations with the linear programming algorithm were presented by Liu *et al.* [18] and Erkorkmaz *et al.* [19]. The key issue in time optimization is the difficult estimation of the machining time. There is a complicated relationship between the machining time, the feedrate profile, and path length. Hence, most existing studies did not plan feedrate from a time perspective. That is, the time is not optimized directly; instead, the feedrate is planned as large as possible by considering path geometry or motion constraints, such as curvature, acceleration limits, etc. However, the path length needs to be computed and also complexity increases dramatically when more motion constraints are considered.



(a)

(b)

(c)

FIGURE 1. Feedrate planning strategy. (a) NURBS tool path (represented in the parameter *u* domain), (b) conventional planning method in the time *t* domain, and (c) proposed method in the parameter *u* domain.

To design the feedrate from a time viewpoint, this study presents a novel planning method which designs the feedrate profile in the parameter domain. In general, the feedrate profile is represented in the time domain and specific type of acceleration/deceleration is adopted (Fig. 1(b)), e.g., trapezoidal, S-curve, etc. It is inconsistent to the NURBS path that is represented in the parameter domain (Fig. 1(a)). Such inconsistency makes the feedrate panning difficult and require complicated computation in relating the parameter and time. In this paper, the feedrate profile will be represented by the same parameter u used to define the NURBS tool path (Fig. 1(c)). As a result, the machining time can be easily obtained without the need of computing the path length and can be optimized directly. In the optimization, particle swarm optimization (PSO) [20]–[22] is employed to minimize the machining time, and fitness selection is used to deal with various constraints [22]. When a feedrate profile does not meet the constraints, a smaller fitness value is attached. Since both the NURBS path and feedrate profile are polynomial or rational polynomial functions in u, the check of motion constraints can be efficiently done by utilizing the property of polynomial function. The proposed method has the following advantages:

1) There exists a mapping from tool path to the designed feedrate and thus there is no need to calculate the path length which often causes feedrate fluctuation.
2) The machining time can be accurately estimated in advance.
3) Most constraints can be expressed as the functions of u and checked by evaluating the values from $u = 0$ to 1. It is more efficient than checking the fine interpolation points at every sampling period. Furthermore, using PSO algorithms, these constraints can be easily included, deleted, or organized.

The rest of this paper is organized as follows. Section 2 reviews some nature-inspired algorithms and two common methods to generate NURBS fine interpolation points. Section 3 describes the proposed feedrate planning strategy. Section 4 shows the simulation results of the proposed method applied to a star-shaped NURBS curve. Finally, section 5 summarizes this paper.

II. RELATED WORK

In this section, we briefly review some nature-inspired algorithms (Section II.A) and two common methods to generate NURBS fine interpolation points (Section II.B).

A. NATURE-INSPIRED ALGORITHMS

In recent years, various optimization problems were discussed [23], [24]. These optimization problems consist of different type of objective function and difficulty levels such as linear and/or nonlinear constraints, which can be equality and/or inequality, and variety of search space such as discrete and mixed type (continuous, discrete). To handle these problems, a current trend is to use nature-inspired algorithms, which work independently of whether the function involved in the problem is differentiable, continuous, or convex, etc. They just evaluate the objective function at given decision variables and consider the optimization problem as a black box. Such algorithms mainly include genetic algorithm (GA) [25], [26], PSO [27]–[32], fruit fly optimization algorithm (FOA) [33], queuing search algorithm (QSA) [34], atom search optimization (ASO) [35], equilibrium optimizer (EO) [36]–[38], evolutionary programming (EP) [39], differential evolution (DE) [40]–[42], etc. Many researchers extended these algorithms to solve the multi-objective optimization problems [43]–[45] or integrated them into multi-population methods to improve the optimization performance [46]–[48]. Among these algorithms, PSO is a considerably popular one which has some attractive features, such as simplicity, less parameters, low computational complexity, and ease to implement. Over the years, several PSO variants have been developed to solve complex industrial and engineering application problems. For example, Liu *et al.* [49] integrated PSO search strategy and Lévy flight to optimize the deployment of access points for wireless local area networks. Yiyang *et al.* [50] proposed an inverse kinematics calculation method based on improved PSO algorithm and applicable to general robots. Rahman *et al.* [51] presented a hybrid GA and PSO algorithm to solve real-time order acceptance and scheduling problems in a flow shop environment. To our best knowledge, no one has used nature-inspired algorithms to solve the NURBS tool path federate planning problems under constraints based on minimum machining time. Since PSO is relatively matured, commercially available, and can be easily accessed by industrial people, it is employed in this study.

B. FINE INTERPOLATION POINTS COMPUTATION

NURBS has been popularly used because it can represent both analytic and free-form surface. A NURBS curve can be expressed as follows:

$$\boldsymbol{C}(u) = \begin{bmatrix} x(u) \\ y(u) \end{bmatrix} = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{1}$$

where $\boldsymbol{P}_i$ is the control point, w_i is the weight of $\boldsymbol{P}_i$, $n+1$ is the number of the control points, p is the degree of NURBS, $N_{i,p}(u)$ is the B-spline basis function, and u is the interpolation parameter. The pth-degree B-spline basis function can be calculated using the recursive formulas given as follows:

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u \le u_{i+1} \\ 0 & otherwise \end{cases} \tag{2}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \tag{3}$$

where $\boldsymbol{U} = \{u_0, u_1, \ldots, u_{n+p+1}\}$ is the knot vector.

To generate the fine interpolation points, the interpolation parameter u for every sampling time needs to be determined. Taylor approximation is a commonly used method. By employing Taylor series expansion, u at $(k+1)th$ sampling time can be expressed as [52]

$$\begin{aligned} u(t_{k+1}) &= u(t_k) + \frac{v(t_k)}{\left|\boldsymbol{C}'(u(t_k))\right|} T_s \\ &+ \left(\frac{\sum_{i=0}^{n} N_{i,p}(u(t_k)) w_i \boldsymbol{P}_i}{\left|\boldsymbol{C}'(u(t_k))\right|} - \frac{v^2(t_k) \cdot \boldsymbol{C}'(u(t_k)) \cdot \boldsymbol{C}''(u(t_k))}{\left|\boldsymbol{C}'(u(t_k))\right|^4} \right) \\ &\times \frac{T_s^2}{2} + H(t_k) \end{aligned} \tag{4}$$

where T_s is the sampling time of the NURBS interpolation, $H(t_k)$ stands for the high order term in the Taylor expansion, $v(t_k)$ is determined according to the specified velocity profile, for example trapezoid or S-curve, and the mth derivatives of a NURBS curve and the B-spline basis function with respect to u are given respectively as [53]

$$\begin{aligned} \boldsymbol{C}^{(m)}(u) &= \frac{\sum_{i=0}^{n} N_{i,p}^{(m)}(u) w_i \boldsymbol{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \\ &- \frac{\sum_{i=1}^{m} \frac{m!}{(m-i)!i!} \left(\sum_{j=0}^{n} N_{j,p}^{(i)}(u) w_j \right) \boldsymbol{C}^{(m-i)}(u)}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \end{aligned} \tag{5}$$

$$N_{i,p}^{(m)}(u) = p \left(\frac{N_{i,p-1}^{(m-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(m-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) \tag{6}$$

To improve the accuracy, higher order Taylor expansion has been applied, but there is always a trade-off between the computing time and the desired accuracy.

Another popular method for determining the new interpolation parameter $u(t_{k+1})$ is to compute the incremental interpolation parameter $\Delta u(t_{k+1})$ based on the previous interpolation parameter $u(t_k)$ and the incremental path length $\Delta l(t_{k+1})$

$$\Delta l(t_{k+1}) = \int_{u(t_k)}^{u(t_k)+\Delta u(t_{k+1})} \sqrt{x'(u)^2 + y'(u)^2 du} \qquad (7)$$

where $\Delta l(t_{k+1}) = v(t_{k+1}) \times T_s$. The new interpolation parameter $u(t_{k+1})$ can then be obtained as follows:

$$u(t_{k+1}) = u(t_k) + \Delta u(t_{k+1}) \qquad (8)$$

After $u(t_{k+1})$ is obtained from (4) or (8), the next interpolation position $\boldsymbol{C}\,(u(t_{k+1}))$ can be determined as well.

III. QUINTIC PROPOSED FEEDRATE PLANNING

A NURBS curve is a curve represented by a vector function in a parameter u denoted as $\boldsymbol{C}\,(u) = [x\,(u)\,, y\,(u)]$. For the machining of a NURBS curve with a CNC machine tool, the command as a time function for each motion axis is required. To this aim, one needs to plan the feedrate profile along the path and then generate the fine interpolation points [3]–[5], [8], [9], [54]. This work presents a novel method to plan the feedrate profile for NURBS tool paths. The proposed method will optimize the machining time and generate the smooth commands that satisfy all of the prescribed motion constraints, such as axis acceleration and jerk limits, etc. The conventional approach is to perform the planning in the time domain, such as trapezoidal or S-curve feedrate profile [2], [9]–[12]. The problems in such approach include: (i) the total machining time cannot be accurately estimated; (ii) there exists feedrate fluctuation.

In this study, the feedrate planning will be performed in the parameter domain. In other words, both the path function and the feedrate profile will be represented in the same parameter u. The advantage of this approach is that u is always from 0 to 1 and the total machining time can be accurately estimated. To see this, let $\hat{\boldsymbol{v}}\,(u)$ be the feedrate profile and $s\,(u)$ be the arc length function, i.e.,

$$s'\,(u) = \sqrt{x'\,(u)^2 + y'\,(u)^2} \qquad (9)$$

where the differentiation convention of $\bullet' = \frac{d}{du}\,(\bullet)$, $\dot{\bullet} = \frac{d}{dt}\,(\bullet)$ is adopted in this paper. Thus, we can get

$$\hat{\boldsymbol{v}}\,(u) = \frac{ds\,(u)}{du}\frac{du}{dt} = s'\,(u)\,\dot{u} \Rightarrow dt = \frac{s'\,(u)}{\hat{\boldsymbol{v}}\,(u)}du \qquad (10)$$

Integrating on both sides of Eq. (10), the machining time $\hat{T}_t$ can be expressed as:

$$\hat{T}_t = \int_0^{\hat{T}_t} dt = \int_0^1 \frac{s'\,(u)}{\hat{\boldsymbol{v}}\,(u)}du \qquad (11)$$

Note that (11) can be accurately calculated within any specified error range.

Being able to obtain the accurate machining time is important since it allows us to implement the optimization algorithm for minimizing the machining time. Given a desired motion path described by NURBS, i.e., $C\,(u)$, it is to design a feedrate profile $\hat{\boldsymbol{v}}\,(u)$ to minimize the machining time $\hat{T}_t$ and at the same time satisfy all of the motion constraints. The PSO algorithm is employed. Fig. 2 is a flowchart of the proposed PSO-based feedrate optimization algorithm, which is detailed below.

A. FEEDRATE PROFILE GENERATION

Most conventional feedrate profiles adopt simple functions (e.g., trapezoidal or S-curve, etc.) in the time domain, which are restricted and difficult to be integrated with NURBS paths. In this study, a more flexible feedrate profile will be taken so that better optimization result can be expected. The proposed feedrate profile is formed by $M - 1$ pieces of function in u separated by M feed control points (FCPs) $(\tilde{u}_i,\ V_i), i = 1, 2, \ldots, M$, as shown in Fig. 3, where $\tilde{u}_i = \frac{i-1}{M-1}$ and V_i is the feedrate. Most pieces of the feedratre profile are defined by cubic spline of u, i.e., for $i = 2, 3, \ldots, M - 2$,

$$\hat{v}_i\,(u) = a_i\,(u - \tilde{u}_i)^3 + b_i\,(u - \tilde{u}_i)^2 + c_i\,(u - \tilde{u}_i) + d_i, \tilde{u}_i \leq u \leq \tilde{u}_{i+1} \qquad (12)$$

where $\{a_i, b_i, c_i, d_i\}$ are constant coefficients to be determined by the smoothness conditions of the profile at FCPs, as shown in Appendix A. For the first and last pieces ($i = 1$ and $M - 1$), the profile cannot be designed as a cubic spline (12) since ill condition will occur at the starting and ending points ($u = 0$ and $u = 1$), where the federate must be zero. Note that from (10), the tangential acceleration $\hat{\boldsymbol{a}}\,(u)$ can be obtained as

$$\hat{\boldsymbol{a}}\,(u) = \frac{d\hat{\boldsymbol{v}}\,(u)}{du}\frac{du}{dt} = \frac{\hat{\boldsymbol{v}}'\,(u)\,\hat{\boldsymbol{v}}\,(u)}{s'\,(u)} \qquad (13)$$

It is easy to see that $\hat{\boldsymbol{v}}\,(u)$ and $\hat{\boldsymbol{a}}\,(u)$ (and higher order derivatives like jerk $\hat{\boldsymbol{J}}\,(u)$, etc.) all vanish at $u = 0$ and $u = 1$, but $\hat{\boldsymbol{v}}'\,(u)$ is finite if (12) is assumed. This implies that the path can hardly start at $u = 0$ and moves slowly at almost zero speed near $u = 1$. To resolve such situation, a nonzero acceleration at $u = 0$ and $u = 1$ is required. From (13), this can be achieved by setting

$$\hat{\boldsymbol{v}}_1'\,(u)\,\hat{\boldsymbol{v}}_1\,(u) = 0.5p^2; \hat{\boldsymbol{v}}_{M-1}'\,(u)\,\hat{\boldsymbol{v}}_{M-1}\,(u) = -0.5q^2 \qquad (14)$$

with the condition that $\hat{\boldsymbol{v}}_1\,(0) = 0$; $\hat{\boldsymbol{v}}_{M-1}\,(1) = 0$. Here, p and q are coefficients and again they are to be determined by the smoothness conditions of the profile at FCPs, as discussed in Appendix A. Equation (14) leads to

$$\hat{v}_1\,(u) = p\sqrt{u}; \quad \hat{v}_{M-1}\,(u) = q\sqrt{1-u} \qquad (15)$$

Thus, the proposed feedrate profile is composed of $M - 1$ piecewise functions in the parameter u as:

$$\hat{\boldsymbol{v}}\,(u) = \hat{v}_i\,(u)\,, \quad \tilde{u}_i \leq u \leq \tilde{u}_{i+1},\ i = 1, 2, \ldots, M - 1 \qquad (16)$$

B. CONSTRAINT CHECK

For the PSO-based optimization, a population of sets of V_i, $i = 2, 3, \ldots, M - 1$, in FCPs, will be first randomly chosen over the feasible domain $[0, \min\,(\|\boldsymbol{v}_{\max}\|\,, F_{\max})]$, where $\boldsymbol{v}_{\max} = (v_{x\max}, v_{y\max})$ are the velocity limits for each

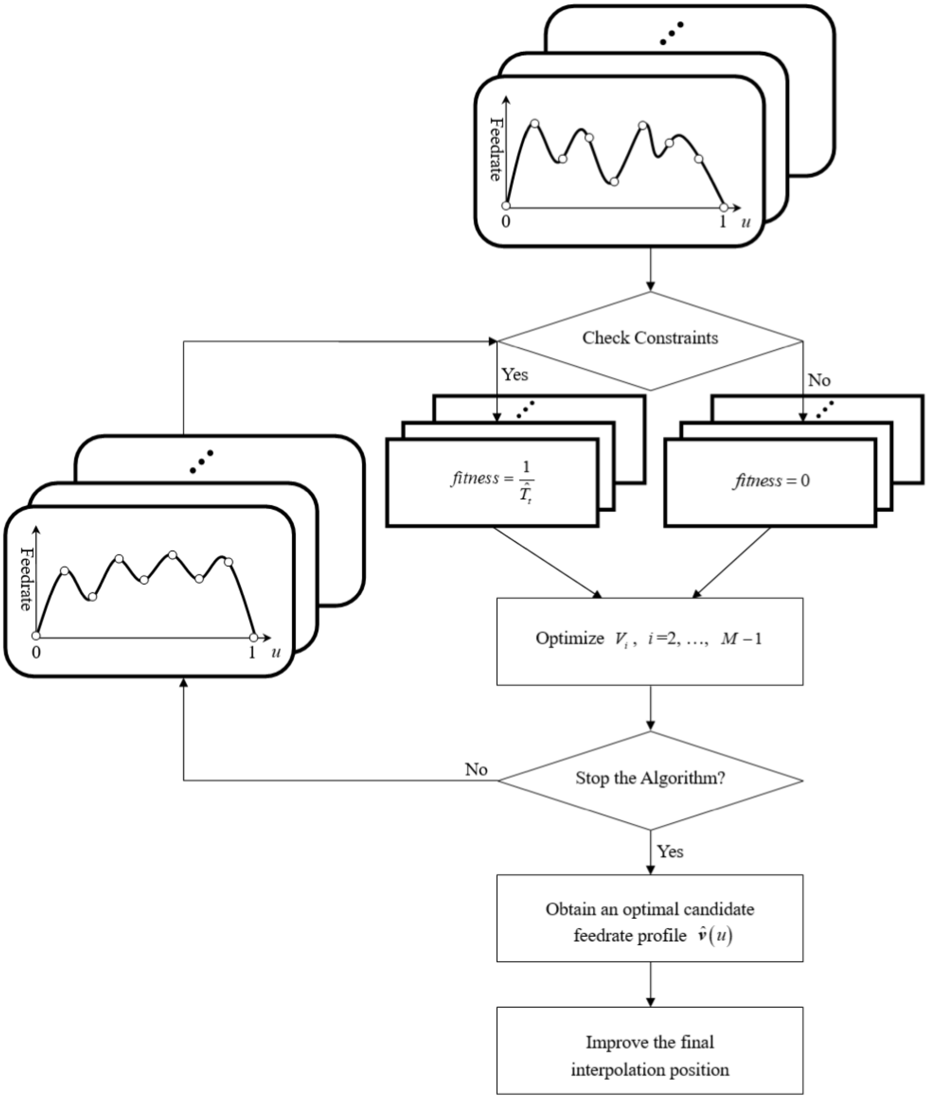

Feedrate
0
1
u
Check Constraints
Yes
No
fitness = $\frac{1}{\hat{T}_t}$
fitness = 0
Feedrate
0
1
u
Optimize V_i, i=2, …, $M-1$
Stop the Algorithm?
No
Yes
Obtain an optimal candidate feedrate profile $\hat{\boldsymbol{v}}(u)$
Improve the final interpolation position

FIGURE 2. Flow-chart of feedrate planning, where $\hat{T}_t$ is machining time.

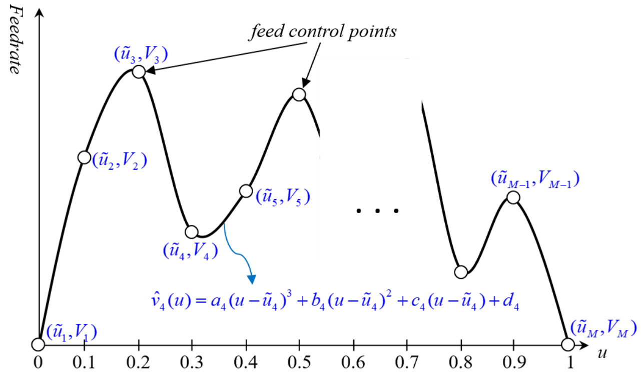


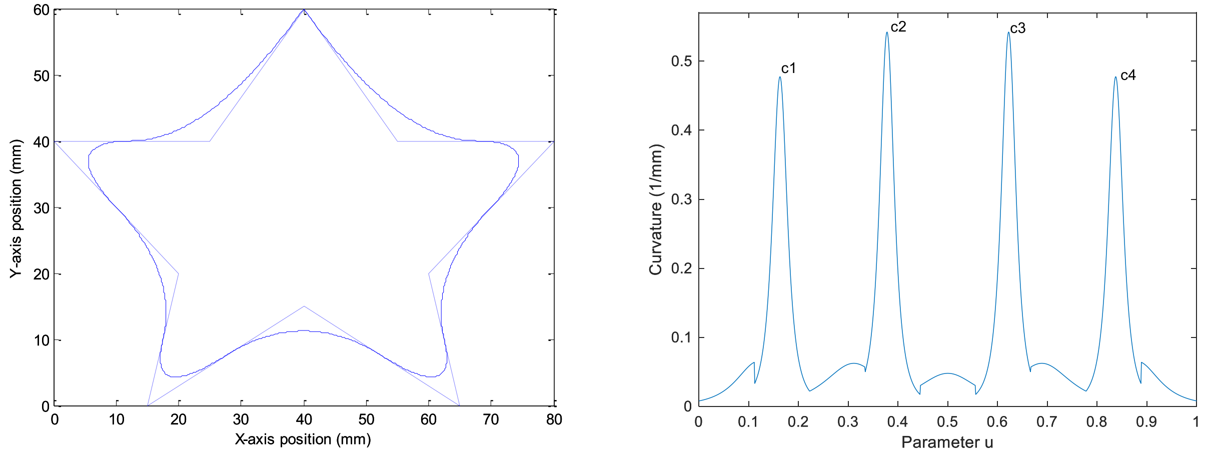
FIGURE 3. Proposed feedrate profile.

axis and $F_{\max}$ is the feedrate limit. Note that $V_1 = V_M = 0$. With V_i, the feedrate profile given by (16) can be generated. Next, one needs to check if the generated feedrate profile satisfies all of the motion constraints, such as acceleration limit, jerk limit, etc. To this aim, most conventional methods require to compute all of the interpolation positions along the tool path or on each axis [55]–[57].

In this study, a more efficient way is proposed by representing the motion functions in terms of u. The motion functions considered include tangential velocity $\hat{\boldsymbol{v}}(u)$, acceleration $\hat{\boldsymbol{a}}(u)$, and jerk $\hat{\boldsymbol{J}}(u)$, and axis velocity $\boldsymbol{v}(u)$, acceleration $\boldsymbol{a}(u)$, and jerk $\boldsymbol{J}(u)$, as shown in Table 1, where $\hat{\boldsymbol{a}}(u)$, $\hat{\boldsymbol{J}}(u)$, $\boldsymbol{v}(u)$, $\boldsymbol{a}(u)$, and $\boldsymbol{J}(u)$ are derived in Appendix B. Note that all of the motion functions are rational polynomial or square root of rational polynomial functions of u. This is because that both the NURBS path function $\boldsymbol{C}(u)$ and the feerate profile $\hat{\boldsymbol{v}}(u)$ are these types of functions. As a result, all of the motion constraints can be represented as polynomial inequalities. For example, let us consider the limit on the

TABLE 1. Some common constraints (According to Eqs. (A4), (A14), and (A15), $\dot{u}(t)$, $\ddot{u}(t)$, and $u(t)$ can be represented in mathematical form as functions of u).

Constraint	Expression
Feedrate	$\hat{\boldsymbol{v}}(u)$
Tangential acceleration	$\hat{\boldsymbol{a}}(u)=\hat{v}'(u)\dot{u}$
Tangential jerk	$\hat{\boldsymbol{J}}(u)=\hat{\boldsymbol{v}}''(u)\dot{u}^2+\dfrac{\hat{\boldsymbol{v}}'(u)^2 s'(u)-\hat{\boldsymbol{v}}(u)\hat{\boldsymbol{v}}'(u)s''(u)}{s'(u)^2}\dot{u}$
Axis velocity	$\boldsymbol{v}(u)=\boldsymbol{C}'(u)\dot{u}$
Axis acceleration	$\boldsymbol{a}(u)=\boldsymbol{C}''(u)\dot{u}^2+\boldsymbol{C}'(u)\ddot{u}$
Axis jerk	$\boldsymbol{J}(u)=\boldsymbol{C}'''(u)\dot{u}^3+3\boldsymbol{C}''(u)\dot{u}\ddot{u}+\boldsymbol{C}'(u)\dddot{u}$
Chord error	$\varepsilon(u)=\rho(u)-\sqrt{\rho^2(u)-\left(\dfrac{\hat{v}(u)T_s}{2}\right)^2}$, where $\rho(u)=\dfrac{\lVert\boldsymbol{C}'(u)\rVert^3}{\lVert\boldsymbol{C}'(u)\times\boldsymbol{C}''(u)\rVert}$ is the radius of curvature.
Centripetal acceleration	$a_c(u)=\dfrac{\hat{v}(u)^2}{\rho(u)}$



(a)

(b)

FIGURE 4. NURBS curve. (a) Star-shaped coutour and (b) curvature profile.

tangential acceleration:

$$|\hat{\boldsymbol{a}}(u)|\leq\alpha\Rightarrow\hat{\boldsymbol{a}}^2(u)\leq\alpha^2 \tag{17}$$

From Eq. (13), it is obvious that $\hat{\boldsymbol{a}}^2(u)$ is rational polynomial in u. Suppose that

$$\hat{\boldsymbol{a}}^2(u)=\frac{N_a(u)}{D_a(u)} \tag{18}$$

where $N_a(u)$ and $D_a(u)$ are polynomial functions. Thus, inequality (17) can be rewritten as

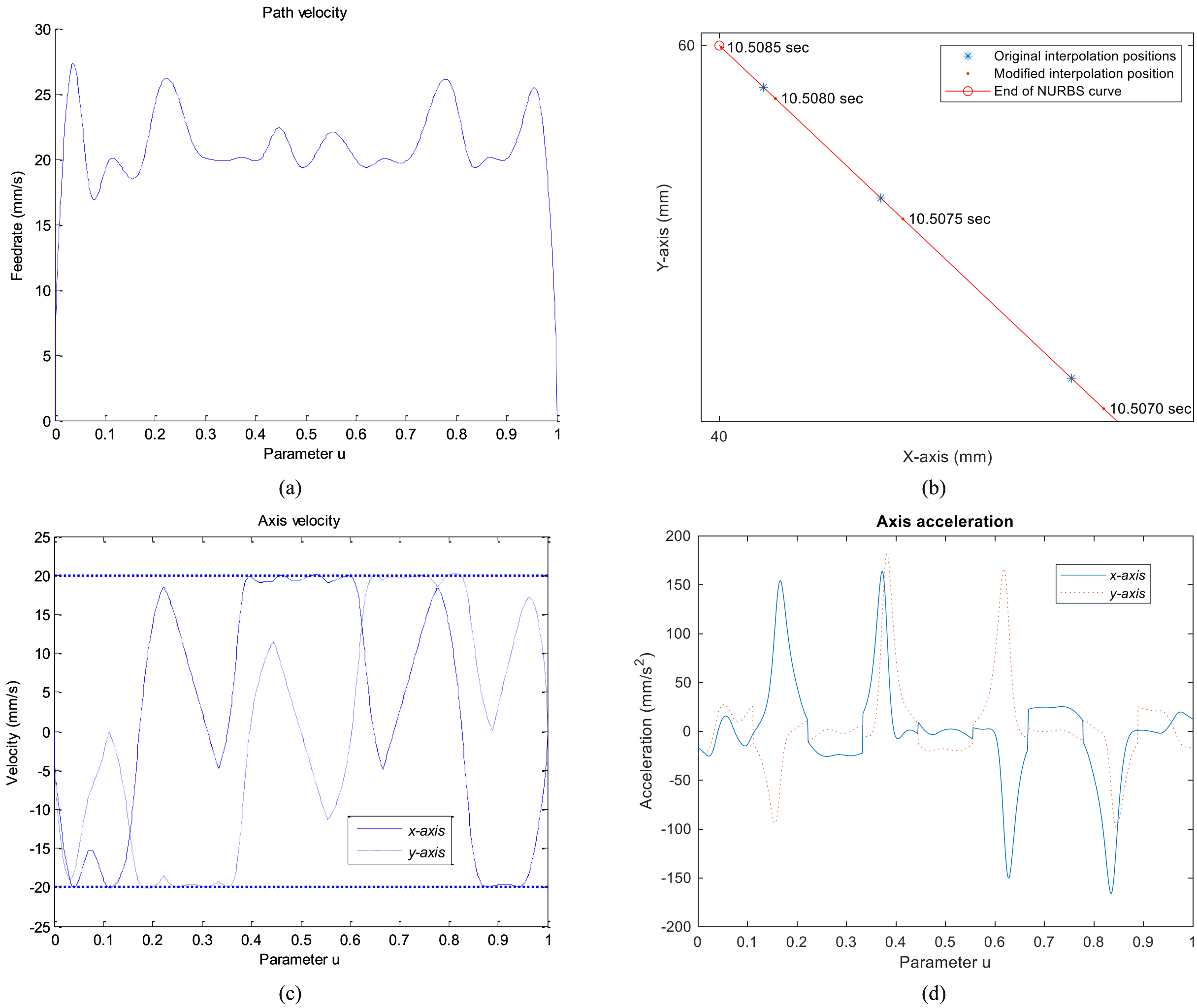
$$N_a(u)-\alpha^2 D_a(u)\leq 0 \tag{19}$$

The other motion constraints can be treated similarly.

Therefore, there is no need to generate all of the interpolation positions for checking the constraints. All of the motion constraints can easily be checked by solving the polynomial inequality or calculating their values from $u=0$ to 1.

C. OPTIMIZATION PROCESS

When one or more constraints are not satisfied, the fitness value is set to zero; otherwise the fitness is set to the inverse of the machining time, i.e., $1/\hat{T}_t$, which can be accurately computed by Eq. (11). Thus, an optimal set of V_i, $i=2,3,\ldots,M-1$, in FCPs can be obtained for the current iteration. Equivalently, an optimal feedrate profile $\hat{\boldsymbol{v}}(u)$ can be obtained. Next, the stop criterion needs to be checked (in this study, a maximum number of iterations is used). If the optimal feedrate profile does not meet the stop criterion, a new population sets of V_i will be generated and the PSO optimization process continues. The PSO optimization process is stopped if the stop criterion is met. The PSO optimization algorithm has been employed here. In fact, any other fitness-based optimization algorithm (e.g., GA, QSA, ASO, EO, etc.) can be applied to optimize $\hat{\boldsymbol{v}}(u)$ (more specifically, V_i in FCPs).



(a)

(b)

(c)

(d)

FIGURE 5. Planning results for axis velocity limit. (a) Feedrate profile (without modification), (b) last several original and modified interpolation positions of NURBS curve, (c) axis velocity, and (d) axis acceleration.

D. FINAL POSITION ERROR IMPROVEMENT

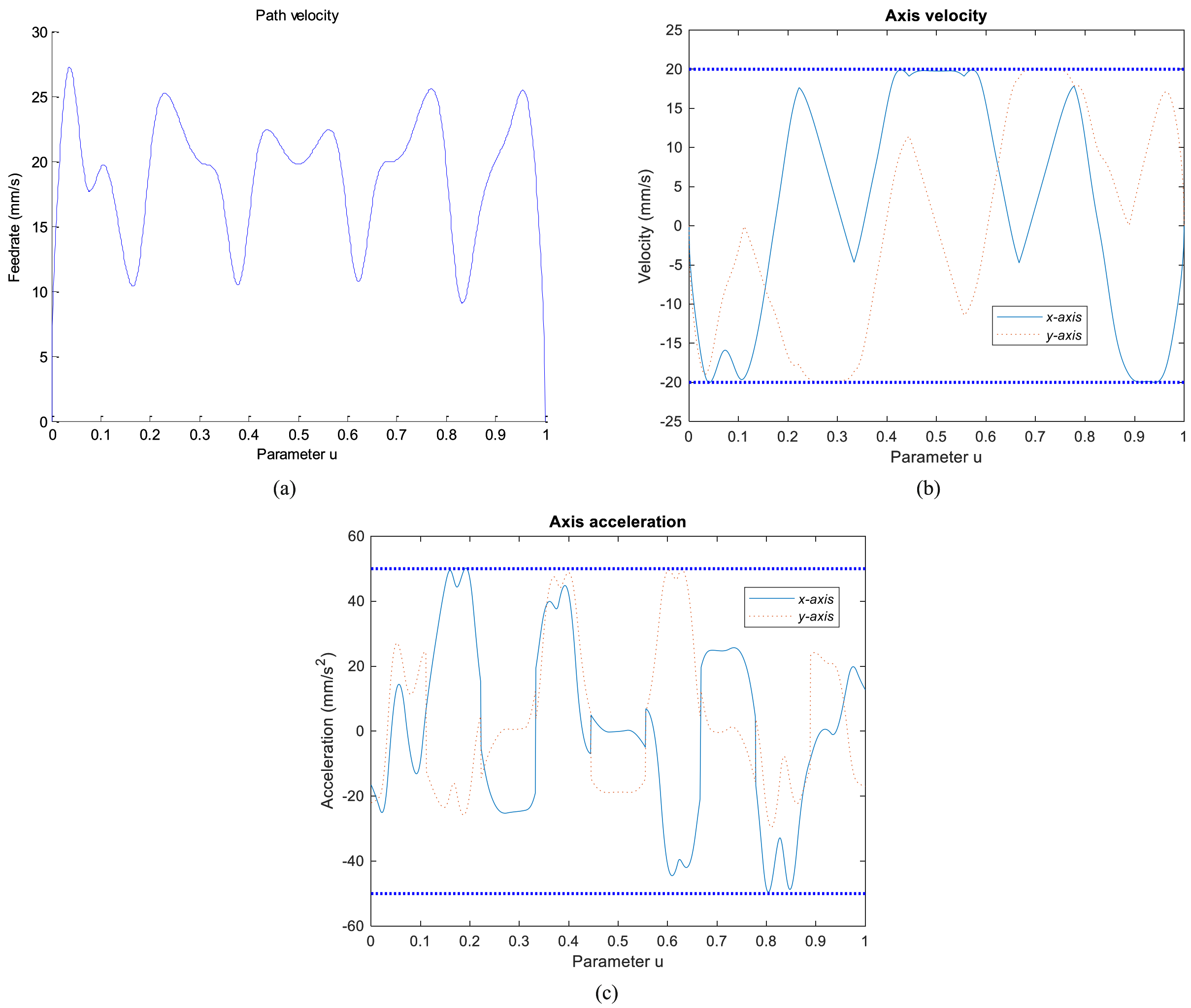
Since the machining time $\hat{T}_t$ obtained by (11) may not be an integer multiple of the interpolation period (sampling time) T_s, there will be a residual length for the final interpolation period [53]. The problem of residual length can be eliminated by slightly reducing the optimal velocity profile, which is detailed below. Let $N = \left\lceil \frac{\hat{T}_t}{T_s} \right\rceil$, where $\lceil x \rceil$ is the ceiling function denoting the minimum integer greater than x. Then, the optimal $\hat{\mathbf{v}}(u)$ can be modified as:

$$\hat{\mathbf{v}}_m(u) = \frac{\hat{T}_t}{NT_s}\hat{\mathbf{v}}(u) \tag{20}$$

By Eq. (11), the machining time of the feedrate profile $\hat{\mathbf{v}}_m(u)$ will be NT_s, an integer multiple of T_s. Since $\hat{\mathbf{v}}_m(u)$ is less than (or equal to) $\hat{\mathbf{v}}(u)$ for every parameter u, it is guaranteed that all of the motion constraints satisfied by $\hat{\mathbf{v}}(u)$ will also be satisfied by $\hat{\mathbf{v}}_m(u)$. The modification given by (20) is also one of the advantages of planning the feedrate in the parameter domain. If the feedrate had been planned as a time function, it is difficult to adjust the machining time and keep the path length (the integration of the feedrate up to the machining time) unchanged.

IV. SIMULATIONS

To verify the proposed method, a star-shaped NURBS curve as shown in Fig. 4(a) is used as an example. This curve has four high-curvature regions, as marked with c1, c2, c3, and c4, respectively, in Fig. 4(b). The high-curvature regions are critical for machining and thus the feedrate should be slowed down in these zones. So, this curve is helpful to validate the effectiveness of feedrate planning. In this example, 30 FCPs (i.e., $V_1, \ldots, V_{30}$) are created uniformly on $u \in [0, 1]$ to construct a feedrate profile, where V_1 and V_{30} are set to zero. PSO is employed to optimize $V_2, \ldots, V_{29}$. The population size is 15, and the maximum number of iterations is 500. Other relevant parameters are listed in Table 2.



(a)

(b)

(c)

FIGURE 6. Planning results for axis velocity and axis acceleration limits. (a) Feedrate profile, (b) axis velocity, and (c) axis acceleration.

TABLE 2. Curve parameters of NURBS.

Interpolator period T_s =0.5 ms	
Degree p =2	
Control points (mm)	{(40, 60), (25, 40), (0, 40), (20, 20), (15, 0), (40, 15), (65, 0), (60, 20), (80, 40), (55, 40), (40, 60)}
Weights	{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1}
Knots	{0, 0, 0, 1/9, 2/9, 1/3, 4/9, 5/9, 2/3, 7/9, 8/9, 1, 1, 1}

In the simulation, various constraints are analyzed. First, only the axis velocity limits are considered. In this case, we are going to design a high feedrate, but each axis speed does not exceed 20 mm/s. Figure 5(a) shows the result of the optimization of the feedrate profile, which produces a machining time of 10.50843 seconds. Clearly, it is not an integer multiple of interpolation period (i.e., 0.5 ms). This means that there will be a little time left when executing the last interpolation. To improve this, the feedrate in Fig. 5(a) is multiplied by a scale factor "10.50843/10.5085", where 10.5085 is a modified machining time that is slightly larger than 10.50843 but integer multiple of 0.5. Figure 5(b) shows the last several original and modified interpolation positions of the star-shaped NURBS curve. It can be seen that the modified feedrate can produce a more accurate final interpolation position at 10.5085 seconds. This result is mainly due to the fact that the machining time can be calculated in advance. Figures 5(c) and 5(d) show the velocities and accelerations, respectively, of the x- and y-xes. Figure 5(c) shows that the axis velocities are constrained within the given limit of [−20 mm/s, 20 mm/s].

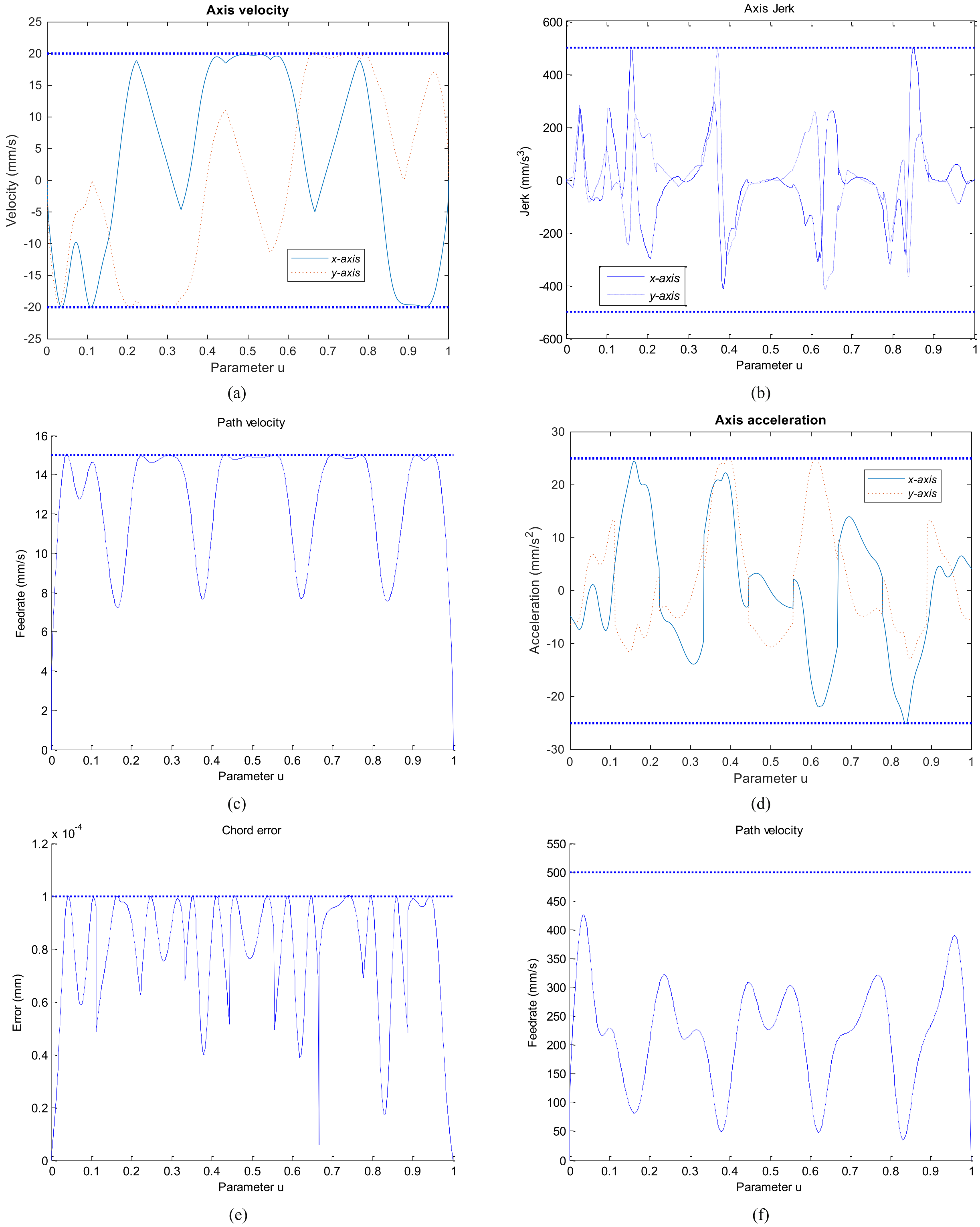
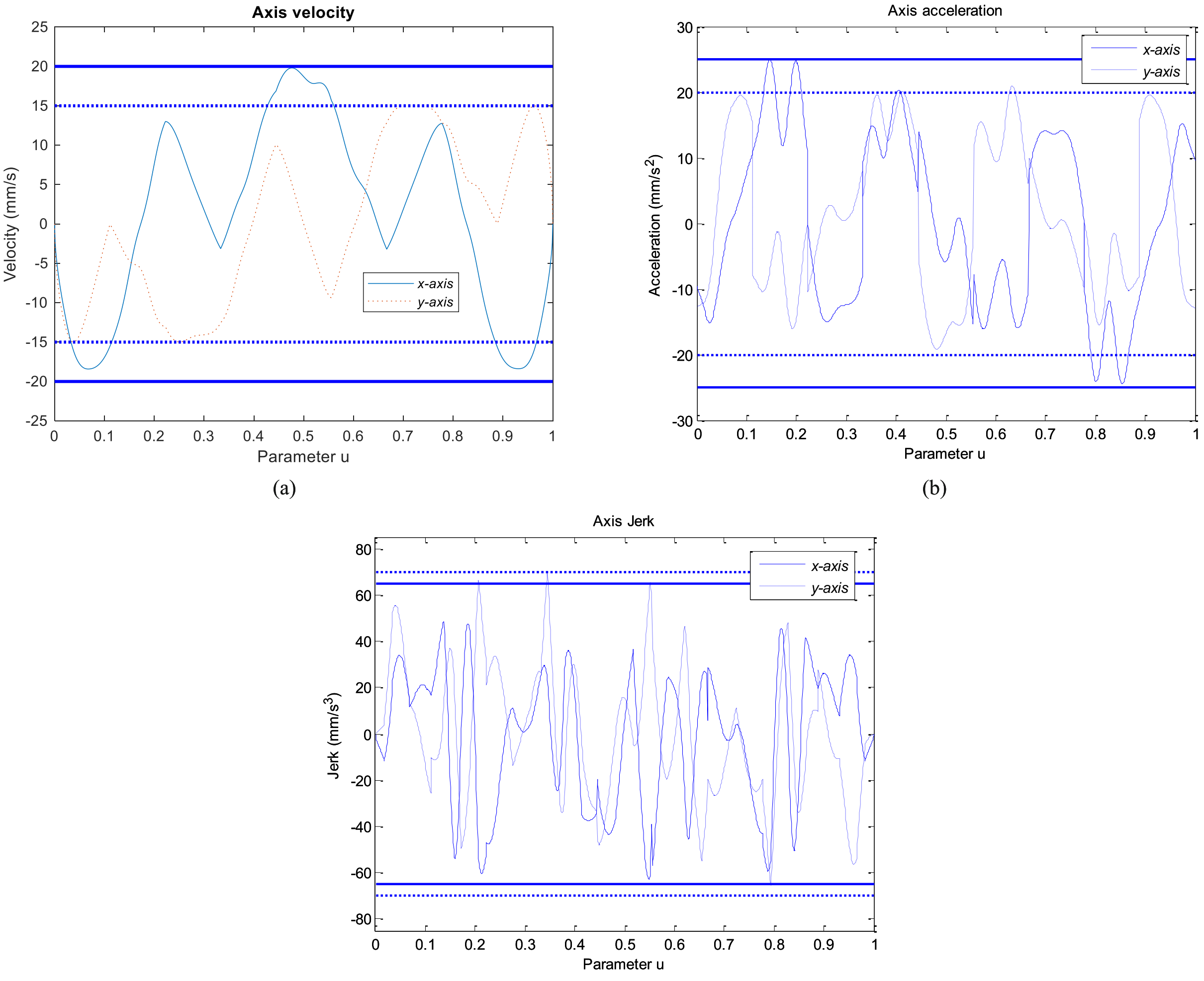


FIGURE 7. Planning results. (a) Axis velocity for Case A, (b) axis jerk for Case A, (c) feedrate profile for Case B, (d) axis acceleration for Case B, (e) chord error for Case C, and (f) feedrate profile for Case C.

It can be seen from Fig. 5(d) that there are excessive centripetal accelerations owing to the high curvature at the four sharp corners. To test the ability of the proposed method to suppress these excessive accelerations, an axis acceleration constraint of 50 mm/s^2 is considered. For this case, the speed should decrease gradually to an adequate



(a)

(b)

(c)

FIGURE 8. Planning results for Case D. (a) Axis velocity, (b) axis acceleration, and (c) axis jerk.

value when approaching these high-curvature zones and then increase gradually to a permissible value. Figure 6(a) shows a new profile of the feedrate. It can be found that there exist four slow-down zones in the feedrate profile corresponding to the four high-curvature zones on the star-shaped path curve. The total machining time is obtained as 11.3550, which is slightly higher than that for the case that only considered the axis velocity constraint. Nevertheless, Figs. 6(b) and 6(c) show that both axis velocities and axis accelerations are constrained within the given limits of [−20 mm/s, 20 mm/s] and [−50 mm/s, 50 mm/s], respectively.

Next, consider the different constraints listed in Table 3. Figures 7(a) and 7(b) show the simulation results of the axis velocities and the axis jerks for Case A. Figures 7(c) and 7(d) show the feedrate and the axis accelerations for Case B. Case C tests the NURBS interpolation with a confined chord error. The simulation results of the feedrate and the chord error for this case are shown in Figs. 7(e) and 7(f). In Case D, different velocity, acceleration, and jerk limits on each axis are considered as constraints for smoother motion. The results are shown in Fig. 8. As can be seen, the proposed method

TABLE 3. Simulation results for various constraints.

Case	Constraints	Machining time (s)
A	Maximum axis velocity $\boldsymbol{v}_{\max}$ =20 mm/s Maximum axis jerk $\boldsymbol{J}_{\max}$ =500 mm/s^3	11.1070
B	Maximum feedrate $\hat{v}$ =15 mm/s Maximum axis acceleration $\boldsymbol{a}_{\max}$ =25 mm/s^2	17.3385
C	Maximum feedrate $\hat{v}$ =500 mm/s Maximum chord error ε =0.1 μm	1.0370
D	Maximum x-axis velocity $v_{x\max}$ =20 mm/s Maximum y-axis velocity $v_{y\max}$ =15 mm/s Maximum x-axis acceleration $a_{x\max}$ =25 mm/s^2 Maximum y-axis acceleration $a_{y\max}$ =20 mm/s^2 Maximum x-axis jerk $J_{x\max}$ =65 mm/s^3 Maximum y-axis jerk $J_{y\max}$ =70 mm/s^3	15.1085

TABLE 4. Optimization results and calculation times under three different population sizes in Case D (results from 10 runs averaged).

Population size	Machining time (s) / SD	Computational time (s)
20	15.10837 / 1.78×10^{-8}	3.507
30	15.10837 / 1.53×10^{-8}	4.898
50	15.10837 / 1.71×10^{-8}	6.450

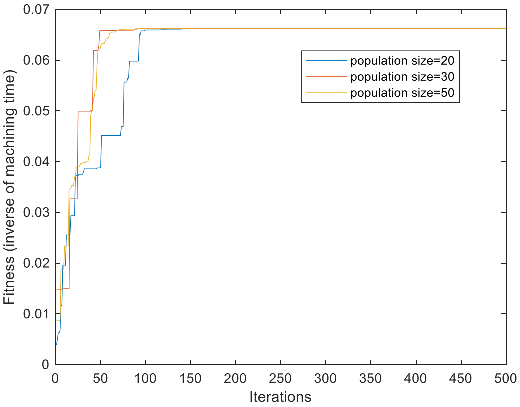


FIGURE 9. Response curves under population sizes of 20, 30, and 50.

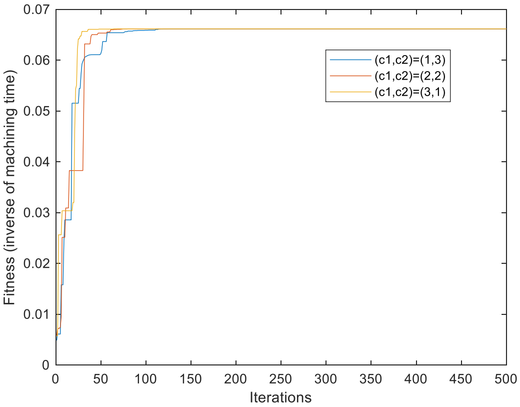


FIGURE 10. Response curves for different choices of (c_1, c_2).

achieves a maximum velocity along the x-axis of 20 mm/s and/or the y-axis of 15 mm/s and a maximum acceleration along the x-axis of 25 mm/s^2 and/or the y-axis of 20 mm/s^2 without exceeding the jerk constraints (x-axis: 65 mm/s^3; y-axis: 70 mm/s^2). The corresponding total machining times for these cases are shown in Table 3.

For PSO, increasing the population size will increase the diversity of the solution, but also increase the calculation time under the same number of iterations. Table 4 shows the optimization results and computational time under population sizes of 20, 30, and 50. As can be seen, PSO can always find fairly consistent solutions (machining time of 15.10837 seconds). It can also be seen from Fig. 9 that convergence to a stable solution is achieved after 200 iterations for all three choices. In addition, we also observed the convergence behavior with respect to the choice of (c_1, c_2) and w [58]. We found that these parameters have little effect on final results and have quick convergence (Figs. 10 and 11), thus saving a lot of parameter design work.

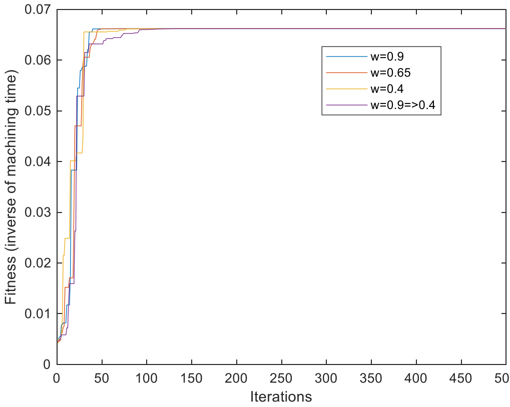


FIGURE 11. Response curves for different choices of w.

For the star-shaped NURBS curve path, the number of FCPs is determined by trial and error through simulations. Although in theory, using more FCPs helps to optimize the feedrate profile, it will inevitably take more optimization time. In future work, we will propose a systematic design strategy for the number and location of FCPs.

V. CONCLUSION

This study proposed a feederate optimization method for NURBS tool paths under various constraint combinations. The conclusions include the following:

1) The feedrate profile is designed with parameter u instead of commonly used time t, there is no need to calculate the path length which often causes feedrate fluctuation.
2) The machining time can be expressed as an analytical form. This facilitates optimizing the machining time and alleviate the residual length after executing the last interpolation.
3) Most constraints can be expressed as the functions of u (including feedrate, tangential acceleration, tangential jerk, axis velocity, axis acceleration, axis jerk, chord error, and centripetal acceleration) and easily checked. In addition, using the fitness-based optimization algorithms, these constraints can be easily included, deleted, or organized.

The proposed optimization method is also possible to be applied to a five-axis machine tools with two additional rotary axes. However, this requires further research.

APPENDIX A

In the proposed feedrate profile, M FCPs will produce $M - 1$ cubic splines (as shown in Fig. 2). Except for the first and last splines, which have only one polynomial coefficient (i.e., p and q, respectively), all other splines have

$$s'''(u) = \frac{\begin{array}{c}\left[x''(u)^2 + x'(u)\,x'''(u) + y''(u)^2 + y'(u)\,y'''(u) + z''(u)^2 + z'(u)\,z'''(u)\right] \\ \times\sqrt{x'(u)^2 + y'(u)^2 + z'(u)^2} - \left[x'(u)\,x''(u) + y'(u)\,y''(u) + z'(u)\,z''(u)\right]^2 \frac{1}{\sqrt{x'(u)^2+y'(u)^2+z'(u)^2}}\end{array}}{x'(u)^2 + y'(u)^2 + z'(u)^2}$$

four polynomial coefficients (i.e., a_j, b_j, c_j, and d_j). So, there are a total of $1 + 4(M-3) + 1 = 4M - 10$ unknown coefficients. Accordingly, $4M - 10$ equations are needed to determine these coefficients. First, we require that each spline connects adjacent FCPs, that is:

$$\hat{v}_i(\tilde{u}_i) = V_i \text{ and } \hat{v}_i(\tilde{u}_{i+1}) = V_{i+1} \tag{A.1}$$

where $i = 2, \ldots, M-2$. Hence, there are $2(M-3)$ of these conditions. Then, to make $\hat{\boldsymbol{v}}(u)$ as smooth as possible (first and second derivatives are continuous) at every $\tilde{u}_i$, we require:

$$\hat{v}'_i(\tilde{u}_{i+1}) = \hat{v}'_{i+1}(\tilde{u}_{i+1}) \text{ and } \hat{v}''_{i+1}(\tilde{u}_{i+1}) = \hat{v}''_{i+1}(\tilde{u}_{i+1}) \tag{A.2}$$

where $i = 1, 2, \ldots, M-2$. As a result there are $2(M-2)$ of these conditions. So, finally we have totally $2(M-3)+2(M-2) = 4M - 10$ equations and thus p, q, and $\{a_j, b_j, c_j, d_j\}$, $j = 2, \ldots, M-2$ can be determined.

APPENDIX B

Using the chain rule, the tangential acceleration $\hat{\boldsymbol{a}}(u)$ and jerk $\hat{\boldsymbol{J}}(u)$ can be expressed, respectively, as:

$$\hat{\boldsymbol{a}}(u) = \frac{d\hat{\boldsymbol{v}}(u)}{dt} = \hat{\boldsymbol{v}'}(u)\,\dot{u} \tag{A.3}$$

$$\begin{aligned}\hat{\boldsymbol{J}}(u) &= \frac{d\hat{\boldsymbol{a}}(u)}{dt} \\ &= \hat{\boldsymbol{v}}''(u)\,\dot{u}^2 + \frac{\hat{\boldsymbol{v}}'(u)^2\,s'(u) - \hat{\boldsymbol{v}}(u)\,\hat{\boldsymbol{v}}'(u)\,s''(u)}{s'(u)^2}\dot{u}\end{aligned} \tag{A.4}$$

where $s''(u) = \frac{x'(u)x''(u)+y'(u)y''(u)+z'(u)z''(u)}{\sqrt{x'(u)^2+y'(u)^2+z'(u)^2}}$. Transforming Eq. (2) yields:

$$\dot{u} = \frac{\hat{\boldsymbol{v}}(u)}{s'(u)} \tag{A.5}$$

By taking the derivative of Eq. (A.5) with respect to t, the second derivative of $u(t)$ is obtained as:

$$\ddot{u} = \frac{\hat{\boldsymbol{a}}(u)}{s'(u)} - \frac{\hat{\boldsymbol{v}}(u)\,s''(u)\,\dot{u}}{s'(u)^2} \tag{A.6}$$

Substituting Eqs. (A.3) and (A.5) into Eq. (A.6) yields:

$$\ddot{u} = \frac{\hat{\boldsymbol{v}'}(u)\,\hat{\boldsymbol{v}}(u)}{s'(u)^2} - \frac{s''(u)\,\hat{\boldsymbol{v}}(u)^2}{s'(u)^3} \tag{A.7}$$

Similarly, the third derivative of $u(t)$ is derived as:

$$\begin{aligned}u = {}&\frac{\hat{\boldsymbol{v}}''(u)\,\hat{\boldsymbol{v}}(u)^2}{s'(u)^3} + \frac{\hat{\boldsymbol{v}}'(u)^2\,\hat{\boldsymbol{v}}(u)}{s'(u)^3} - \frac{\hat{\boldsymbol{v}}'(u)\,\hat{\boldsymbol{v}}(u)^2\,s''(u)}{s'(u)^4} \\ &- \frac{3s''(u)\,\hat{\boldsymbol{v}}(u)\,\ddot{u}}{s'(u)^2} - \frac{s'''(u)\,\hat{\boldsymbol{v}}(u)\,\dot{u}^2}{s'(u)^2}\end{aligned} \tag{A.8}$$

where $s'''(u)$, as shown at the top of the page. According to $\dot{u}$, $\ddot{u}$, and u, the axis velocity $\boldsymbol{v}(u)$, axis acceleration $\boldsymbol{a}(u)$, and axis jerk $\boldsymbol{J}(u)$ can be expressed with u, respectively, as:

$$\boldsymbol{v}(u) = \frac{d\boldsymbol{C}(u)}{dt} = \frac{d\boldsymbol{C}(u)}{du}\frac{du}{dt} = \boldsymbol{C}'(u)\,\dot{u} \tag{A.9}$$

$$\begin{aligned}\boldsymbol{a}(u) &= \frac{d\boldsymbol{v}(u)}{dt} = \frac{d\boldsymbol{C}'(u)}{dt}\frac{du}{dt} + \boldsymbol{C}'(u)\frac{d^2u}{dt^2} \\ &= \frac{d\boldsymbol{C}'(u)}{du}\left(\frac{du}{dt}\right)^2 + \boldsymbol{C}'(u)\frac{d^2u}{dt^2} \\ &= \boldsymbol{C}''(u)\,\dot{u}^2 + \boldsymbol{C}'(u)\,\ddot{u}\end{aligned} \tag{A.10}$$

$$\begin{aligned}\boldsymbol{J}(u) &= \frac{d\boldsymbol{a}(u)}{dt} = \frac{d\boldsymbol{C}''(u)}{dt}\left(\frac{du}{dt}\right)^2 \\ &\quad +2\boldsymbol{C}''(u)\frac{du}{dt}\frac{d^2u}{dt^2} + \frac{d\boldsymbol{C}'(u)}{dt}\frac{d^2u}{dt^2} + \boldsymbol{C}'(u)\frac{d^3u}{dt^3} \\ &= \boldsymbol{C}'''(u)\,\dot{u}^3 + 3\boldsymbol{C}''(u)\,\dot{u}\ddot{u} + \boldsymbol{C}'(u)\,u\end{aligned} \tag{A.11}$$

REFERENCES

[1] H. Ni, J. Yuan, S. Ji, C. Zhang, and T. Hu, "Feedrate scheduling of NURBS interpolation based on a novel jerk-continuous ACC/DEC algorithm," *IEEE Access*, vol. 6, pp. 66403–66417, 2018.

[2] K. Erwinski, M. Paprocki, and G. Karasek, "Comparison of NURBS trajectory interpolation algorithms for high-speed motion control systems," in *Proc. IEEE 19th Int. Power Electron. Motion Control Conf. (PEMC)*, Apr. 2021, pp. 527–533.

[3] K. Zhao, Z. Kang, and X. Guo, "Smooth trajectory generation for predefined path with pseudo spectral method," *IEEE Access*, vol. 8, pp. 158735–158744, 2020.

[4] X.-C. Xi, H. Chen, H.-D. Liu, M. Chen, and W.-S. Zhao, "Extended unit arc length increment interpolation for generalized NURBS curves in multi-axis EDM," *Precis. Eng.*, vol. 59, pp. 81–89, Sep. 2019.

[5] S.-L. Chen and C.-Y. Chou, "Contouring control of multi-axis motion systems for NURBS paths," *IEEE Trans. Autom. Sci. Eng.*, vol. 13, no. 2, pp. 1062–1071, Apr. 2016.

[6] D.-N. Song, J.-W. Ma, Y.-G. Zhong, D. Xiao, J.-J. Yao, and C. Zhou, "A fully real-time spline interpolation algorithm with axial jerk constraint based on FIR filtering," *Int. J. Adv. Manuf. Technol.*, vol. 113, nos. 7–8, pp. 1873–1886, Apr. 2021.

[7] D.-N. Song, J.-W. Ma, Y.-G. Zhong, and J.-J. Yao, "Global smoothing of short line segment toolpaths by control-point-assigning-based geometric smoothing and FIR filtering-based motion smoothing," *Mech. Syst. Signal Process.*, vol. 160, Nov. 2021, Art. no. 107908.

[8] X. Gao, X. Liu, L. Qiu, and S. Zhang, "NURBS curve interpolation method with flexibility and high accuracy based on curvature constraint and displacement compensation," *J. Comput.-Aided Des. Comput. Graph.*, vol. 30, no. 12, p. 2213, 2018.

[9] Z.-Y. Jia, D.-N. Song, G.-Q. Hu, W.-W. Su, and J.-W. Ma, "A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints," *Int. J. Mach. Tools Manuf.*, vol. 116, pp. 1–17, May 2017.

[10] W. Zhong, X. Luo, W. Chang, F. Ding, and Y. Cai, "A real-time interpolator for parametric curves," *Int. J. Mach. Tools Manuf.*, vol. 125, pp. 133–145, Feb. 2018.

[11] H. Ni, C. Zhang, C. Chen, T. Hu, and Y. Liu, "A parametric interpolation method based on prediction and iterative compensation," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 1, pp. 1–10, Feb. 2019.

[12] H. Ni, C. Zhang, S. Ji, T. Hu, Q. Chen, Y. Liu, and G. Wang, "A bidirectional adaptive feedrate scheduling method of NURBS interpolation based on S-shaped ACC/DEC algorithm," *IEEE Access*, vol. 6, pp. 63794–63812, 2018.

[13] W. Chen, W. Yuan, Z. Zhu, and X. Zhang, "NURBS curve adaptive five-segment acceleration and deceleration control algorithm," in *Proc. IEEE 9th Joint Int. Inf. Technol. Artif. Intell. Conf. (ITAIC)*, Dec. 2020, pp. 611–619.

[14] Y. Wang and L. Zhang, "A high-speed machining algorithm for continuous corners," *J. Mech. Civil Eng.*, vol. 15, no. 5, pp. 64–69. 2018.

[15] Y. Wang, D. Yang, R. Gai, S. Wang, and S. Sun, "Design of trigonometric velocity scheduling algorithm based on pre-interpolation and look-ahead interpolation," *Int. J. Mach. Tools Manuf.*, vol. 96, pp. 94–105, Sep. 2015.

[16] M. Yuan, Z. Chen, B. Yao, and X. Zhu, "Time optimal contouring control of industrial biaxial gantry: A highly efficient analytical solution of trajectory planning," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 1, pp. 247–257, Feb. 2017.

[17] K. Zhang, X. S. Gao, H. B. Li, and C. M. Yuan, "A greedy algorithm for feedrate planning of CNC machines along curved tool paths with confined jerk," *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 4, pp. 472–483, Aug. 2012.

[18] H. Liu, Q. Liu, P. Sun, Q. Liu, and S. Yuan, "The optimal feedrate planning on five-axis parametric tool path with geometric and kinematic constraints for CNC machine tools," *Int. J. Prod. Res.*, vol. 55, no. 13, pp. 3715–3731, Jul. 2017.

[19] K. Erkorkmaz, Q.-G. Chen, M.-Y. Zhao, X. Beudaert, and X.-S. Gao, "Linear programming and windowing based feedrate optimization for spline toolpaths," *CIRP Ann.*, vol. 66, no. 1, pp. 393–396, 2017.

[20] H.-Y. Tseng, P.-H. Chu, H.-C. Lu, and M.-J. Tsai, "Easy particle swarm optimization for nonlinear constrained optimization problems," *IEEE Access*, vol. 9, pp. 124757–124767, 2021.

[21] Y. J. Gong, J. J. Li, Y. Zhou, Y. Li, H. S. H. Chung, Y. H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE Trans. Cybern.*, vol. 46, no. 10, pp. 2277–2290, Oct. 2016.

[22] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019.

[23] Z. Wu, G. Li, S. Shen, X. Lian, E. Chen, and G. Xu, "Constructing dummy query sequences to protect location privacy and query privacy in location-based services," *World Wide Web*, vol. 24, no. 1, pp. 25–49, Jan. 2021.

[24] W. Yan, G. Li, Z. Wu, S. Wang, and P. S. Yu, "Extracting diverse-shapelets for early classification on time series," *World Wide Web*, vol. 23, no. 6, pp. 3055–3081, Nov. 2020.

[25] H. Yu, G. Yu, Y. Xu, and J. Zou, "Torque performance improvement for slotted limited-angle torque motors by combined SMA application and GA optimization," *IEEE Trans. Magn.*, vol. 57, no. 2, pp. 1–5, Feb. 2021.

[26] S. Kanwal and S. Asghar, "Speech emotion recognition using clustering based GA-optimized feature set," *IEEE Access*, vol. 9, pp. 125830–125842, 2021.

[27] A. Iqbal and G. K. Singh, "PSO based controlled six-phase grid connected induction generator for wind energy generation," *CES Trans. Electr. Mach. Syst.*, vol. 5, no. 1, pp. 41–49, Mar. 2021.

[28] J. Wu, X. Sun, and J. Zhu, "Accurate torque modeling with PSO-based recursive robust LSSVR for a segmented-rotor switched reluctance motor," *CES Trans. Electr. Mach. Syst.*, vol. 4, no. 2, pp. 96–104, Jun. 2020.

[29] J. Wang and T. Kumbasar, "Parameter optimization of interval type-2 fuzzy neural networks based on PSO and BBBC methods," *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 1, pp. 247–257, Jan. 2019.

[30] Y. Wang, Y. Li, L. Jiang, Y. Huang, and Y. Cao, "PSO-based optimization for constant-current charging pattern for li-ion battery," *Chin. J. Electr. Eng.*, vol. 5, no. 2, pp. 72–78, Jun. 2019.

[31] Z. Long, Z. Jiang, C. Wang, Y. Jin, Z. Cao, and Y. Li, "A novel approach to control of piezo-transducer in microelectronics packaging: PSO-PID and editing trajectory optimization," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 10, no. 5, pp. 795–805, May 2020.

[32] M. Zhou, F. Lin, Q. Hu, Z. Tang, and C. Jin, "AI-enabled diagnosis of spontaneous rupture of ovarian endometriomas: A PSO enhanced random forest approach," *IEEE Access*, vol. 8, pp. 132253–132264, 2020.

[33] J. Fan, Q. Hu, and Z. Tang, "Predicting vacant parking space availability: An SVR method with fruit fly optimisation," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1414–1420, Dec. 2018.

[34] J. Zhang, M. Xiao, L. Gao, and Q. Pan, "Queuing search algorithm: A novel Metaheuristic algorithm for solving engineering optimization problems," *Appl. Math. Model.*, vol. 63, pp. 464–490, Nov. 2018.

[35] W. Zhao, L. Wang, and Z. Zhang, "A novel atom search optimization for dispersion coefficient estimation in groundwater," *Future Gener. Comput. Syst.*, vol. 91, pp. 601–610, Feb. 2019.

[36] A. Faramarzi, M. Heidarinejad, B. Stephens, and S. Mirjalili, "Equilibrium optimizer: A novel optimization algorithm," *Knowl.-Based Syst.*, vol. 191, Mar. 2020, Art. no. 105190.

[37] Y. Gao, Y. Zhou, and Q. Luo, "An efficient binary equilibrium optimizer algorithm for feature selection," *IEEE Access*, vol. 8, pp. 140936–140963, 2020.

[38] A. Mostafa, M. Ebeed, S. Kamel, and M. A. Abdel-Moamen, "Optimal power flow solution using levy spiral flight equilibrium optimizer with incorporating CUPFC," *IEEE Access*, vol. 9, pp. 69985–69998, 2021.

[39] A. Dolganov, "Application of the evolutionary programming for indirect evaluation of blood pressure based on heart rate variability parameters," in *Proc. Ural Symp. Biomed. Eng., Radioelectronics Inf. Technol. (USBEREIT)*, May 2020.

[40] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1430–1442, Mar. 2021.

[41] X. Lang and W. Yuan, "Localization method of multiple leaks based on time-frequency analysis and improved differential evolution," *IEEE Sensors J.*, vol. 20, no. 23, pp. 14383–14390, Dec. 2020.

[42] V. Parque and T. Miyashita, "Smooth curve fitting of mobile robot trajectories using differential evolution," *IEEE Access*, vol. 8, pp. 82855–82866, 2020.

[43] X. Wen, K. Wang, H. Li, H. Sun, H. Wang, and L. Jin, "A two-stage solution method based on NSGA-II for Green Multi-Objective integrated process planning and scheduling in a battery packaging machinery workshop," *Swarm and Evol. Comput.*, vol. 61, Dec. 2021, Art. no. 100820.

[44] X. Lai and Y. Zhou, "Analysis of multiobjective evolutionary algorithms on the biobjective traveling salesman problem (1,2)," *Multimedia Tools Appl.*, vol. 79, nos. 41–42, pp. 30839–30860, Nov. 2020.

[45] S. Zhang and T. N. Wong, "Integrated process planning and scheduling: An enhanced ant colony optimization heuristic with parameter tuning," *J. Intell. Manuf.*, vol. 29, no. 3, pp. 585–601, Mar. 2018.

[46] H. Ma, S. Shen, M. Yu, Z. Yang, M. Fei, and H. Zhou, "Multi-population techniques in nature inspired optimization algorithms: A comprehensive survey," *Swarm Evol. Comput.*, vol. 44, pp. 365–387, Feb. 2019.

[47] X. Li, S. Ma, and Y. Wang, "Multi-population based ensemble mutation method for single objective bilevel optimization problem," *IEEE Access*, vol. 4, pp. 7262–7274, 2016.

[48] X. Wang and L. Tang, "An adaptive multi-population differential evolution algorithm for continuous multi-objective optimization," *Inf. Sci.*, vol. 348, pp. 124–141, Jun. 2016.

[49] P. Liu, Q. Hu, K. Jin, G. Yu, and Z. Tang, "Toward the energy-saving optimization of WLAN deployment in real 3-D environment: A hybrid swarm intelligent method," *IEEE Syst. J.*, early access, Apr. 5, 2021, doi: 10.1109/JSYST.2021.3065434.

[50] L. Yiyang, J. Xi, B. Hongfei, W. Zhining, and S. Liangliang, "A general robot inverse kinematics solution method based on improved PSO algorithm," *IEEE Access*, vol. 9, pp. 32341–32350, 2021.

[51] H. F. Rahman, M. N. Janardhanan, and I. E. Nielsen, "Real-time order acceptance and scheduling problems in a flow shop environment using hybrid GA-PSO algorithm," *IEEE Access*, vol. 7, pp. 112742–112755, 2019.

[52] A.-C. Lee, M.-T. Lin, Y.-R. Pan, and W.-Y. Lin, "The feedrate scheduling of NURBS interpolator for CNC machine tools," *Comput.-Aided Design*, vol. 43, pp. 612–628, Jun. 2011.

[53] L. Piegl and W. Tiller, *The NURBS Books*, 2nd ed. Berlin, Germany: Springer, 1997.

[54] M.-S. Tsai, H.-W. Nien, and H.-T. Yau, "Development of integrated acceleration/deceleration look-ahead interpolation technique for multi-blocks NURBS curves," *Int. J. Adv. Manuf. Technol.*, vol. 56, nos. 5–8, pp. 601–618, Sep. 2011.

[55] S. H. Suh, S. K. Kang, D. H. Chung, and I. Stroud, *Theory and design of CNC Systems*. Cham, Switzerland: Springer, 2008.

[56] K. Erkorkmaz and Y. Altintas, ‘‘High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation,’’ *Int. J. Mach. Tools Manuf.*, vol. 41, no. 9, pp. 1323–1345, Jul. 2001.

[57] Q. Hu, Y. Chen, X. Jin, and J. Yang, ‘‘A real-time C3 continuous local corner smoothing and interpolation algorithm for CNC machine tools,’’ *J. Manuf. Sci. Eng.*, vol. 141, no. 4, Feb. 2019, Art. no. 041004.

[58] T.-C. Lu and S.-L. Chen, ‘‘Real-time local optimal Bézier corner smoothing for CNC machine tools,’’ *IEEE Access*, vol. 9, pp. 152718–152727, 2021.

TZYY-CHYANG LU (Member, IEEE) received the B.S. degree in electrical engineering from the National Cheng Kung University, Tainan, Taiwan, in 1998, the M.S. degree in electrical engineering from I-Shou University, Kaohsiung, Taiwan, in 2000, and the Ph.D. degree from the National Cheng Kung University, in 2011.

He is currently an Assistant Professor with the Department of Computer Science and Information Engineering, Providence University, Taichung, Taiwan. His research interests include optimization algorithm, machine learning, and intelligent control.

SHYH-LEH CHEN (Member, IEEE) was born in Keelung, Taiwan, in October 1964. He received the B.S. and M.S. degrees in power mechanical engineering from the National Tsing Hua University, Hsinchu, Taiwan, in 1987 and 1989, respectively, and the Ph.D. degree in mechanical engineering from Michigan State University, East Lansing, Michigan, USA, in 1996.

Since 1996, he has been with the National Chung Cheng University, Chiayi, Taiwan, where he is currently a Distinguished Professor with the Department of Mechanical Engineering. He served as the Deputy Director of the Advanced Institute of Manufacturing With High-Tech Innovations (AIM-HI), from 2011 to 2017. He also served as the Director of the Advanced Machine Tools Research Center, from 2010 to 2013. His research interests include nonlinear dynamics and control, wavelet analysis, with application to motion control of multiaxis systems, active magnetic bearings, and ship stabilization.

Prof. Chen has served as the program chairs/invited session chairs of several international conferences, including 9th World Congress on Intelligent Control and Automation (WCICA 2011), 2013 CACS International Automatic Control Conference (CACS 2013), the 5th International Conference on Advanced Manufacturing (ICAM 2014), the tenth annual IEEE International Conference on Automation Science and Engineering (IEEE CASE 2014), and the 35th Annual Conference of Chinese Society of Mechanical Engineers (CSME 2018). He received the Automatic Control Award for Young Scholars and Outstanding Automatic Control Award from the Chinese Automatic Control Society, in 2003 and 2015, respectively; the Outstanding Teaching Award from CCU, in 2006; the Delta Award from IEEE Tainan Section, in 2017; and the Distinguished Research Award from CCU, in 2018. He was a recipient of the Best Paper Award of The 2011 IAENG International Conference on Control and Automation, 2013 Conference on Precision Machinery and Manufacturing Technology (PMMT 2013), 2015 CACS International Automatic Control Conference (CACS 2015), and 2017 CACS International Automatic Control Conference (CACS 2017).

• • •