International Journal of Machine Tools and Manufacture 125 (2018) 133–145

Contents lists available at ScienceDirect

International Journal of Machine Tools and Manufacture

journal homepage: www.elsevier.com/locate/ijmactool

A real-time interpolator for parametric curves

Wenbin Zhong, Xichun Luo *, Wenlong Chang, Fei Ding, Yukui Cai

Centre for Precision Manufacturing, DMEM, University of Strathclyde, Glasgow, G1 1XJ, UK

ARTICLE INFO

Keywords:
Real-time interpolator
Parametric curve
Lookahead
Machine dynamics
Contour error

ABSTRACT

Driven by the ever increasing need for the high-speed high-accuracy machining of freeform surfaces, the interpolators for parametric curves become highly desirable, as they can eliminate the feedrate and acceleration fluctuation due to the discontinuity in the first derivatives along the linear tool path. The interpolation for parametric curves is essentially an optimization problem, and it is extremely difficult to get the time-optimal solution. This paper presents a novel real-time interpolator for parametric curves (RTIPC), which provides a near time-optimal solution. It limits the machine dynamics (axial velocities, axial accelerations and jerk) and contour error through feedrate lookahead and acceleration lookahead operations, meanwhile, the feedrate is maintained as high as possible with minimum fluctuation. The lookahead length is dynamically adjusted to minimize the computation load. And the numerical integration error is considered during the lookahead calculation. Two typical parametric curves are selected for both numerical simulation and experimental validation, a cubic phase plate freeform surface is also machined. The numerical simulation is performed using the software (open access information is in the Acknowledgment section) that implements the proposed RTIPC, the results demonstrate the effectiveness of the RTIPC. The real-time performance of the RTIPC is tested on the in-house developed controller, which shows satisfactory efficiency. Finally, machining trials are carried out in comparison with the industrial standard linear interpolator and the state-of-the-art Position-Velocity-Time (PVT) interpolator, the results show the significant advantages of the RTIPC in coding, productivity and motion smoothness.

1. Introduction

Parametric curves are represented in parametric form, each coordinate of the curve is given by an explicit function of an independent parameter, in a form of $\boldsymbol{C}(u) = (x(u), y(u))\ u \in [a, b]$. Parametric representation has a number of advantages, as summarized by Piegl and Tiller [1], most importantly including easy to extend to higher dimensions, intuitive for geometric design, etc. Parametric curves have seen their wide applications in computer-aided design (CAD) of products like optics, molds/dies, biomedical implants, etc., where freeform surfaces are of great importance [2]. When machining these products, computer-aided manufacturing (CAM) software is usually used to generate tool path, which is generally composed of a large number of short linear segments. This approach results in the following undesirable problems:

(1) To achieve higher contour accuracy, more segments are needed to approximate the original surface, which pose huge burden for data transfer and memory of the computer numerical control (CNC) system;

(2) Increased feedrate and acceleration fluctuation, which is due to the discontinuity in the first derivatives along the tool path, reduces the average feedrate and causes vibration, i.e. lower productivity and poorer surface finish [3].

A potential solution to above problems is to present the tool path with parametric curves. However, mathematical challenges remain in the interpolation of those curves. Given the curve $\boldsymbol{C}(u)$, to determine u_k for the *kth* interpolation period, the re-parametrization of u with time t is required, i.e. solve the function $u = u(t)$. Moreover, the re-parametrization should be subject to the machine dynamics constraints and contour error tolerance:.

(1) Axial velocities and accelerations should be limited to avoid saturating axes, they are the first and second derivatives of the corresponding parametric function over time, respectively;
(2) Feedrate is the resultant value of axial velocities. The jerk, which is the second derivative of the feedrate over time, should be limited to guarantee smooth motion profiles;

* Corresponding author.
E-mail address: xichun.luo@strath.ac.uk (X. Luo).

https://doi.org/10.1016/j.ijmachtools.2017.11.010
Received 18 August 2017; Received in revised form 19 November 2017; Accepted 21 November 2017
Available online 22 November 2017

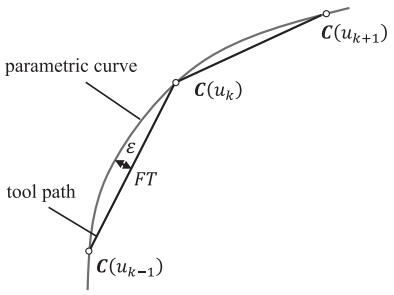


Fig. 1. Contour error estimation.

(3) The contour error ε increases with increasing feedrate, as shown in Fig. 1, the feedrate should be limited to achieve high contour accuracy.

Therefore, the interpolation of parametric curves becomes an optimization problem, the time-optimal solution is highly desirable, in which case the tool traverses the curve in minimum time while satisfying above constraints. Timar and co-workers [4] worked out the time-optimal solution for polynomial curves subject to only axial acceleration bound. It is generally impractical to find the time-optimal solutions subject to all the prescribed bounds for general parametric curves, because it involves solving many non-linear differential equations. Currently, two technical routes have been developed for the interpolation of parametric curves, i.e. arc length parametrization and recursive Taylor's expansion.

The feedrate is the differential of arc length s over time, which makes the interpolation straightforward. However, the arc length parametrization is extremely difficult, as it requires to solve the inverse function of the non-linear integral function for arc length evaluation

$$s(u) = \int_a^u \sqrt{\dot{x}(u)^2 + \dot{y}(u)^2}\,du \tag{1}$$

Erkorkmaz and co-workers [5–8] proposed an interpolator for non-uniform rational B-Spline (NURBS), it used a 7th order polynomial to approximate the arc length parametrization. Similarly, Liu and co-workers [9] adopted a cubic polynomial to generate piecewise approximation of arc length parametrization for NURBS. Although near arc length parametrization is possible with some numerical methods, the approximation error will accumulate along the curve, especially for the curve with large curvature variation or uneven parametrization [10]. Furthermore, the numerical method is not applicable for general parametric curves, and the process is computationally intensive. As a result, this paper will not adopt this technical route.

Alternatively, Shpitalni, Koren and Lo [11] developed the recursive Taylor's expansion method. Given the feedrate F and current interpolation point u_k, the next interpolation point can be calculated by

$$u_{k+1} = u_k + \frac{FT}{\sqrt{\dot{x}(u_k)^2 + \dot{y}(u_k)^2}} \tag{2}$$

where T is the interpolation period. Fanuc has applied this work to its NURBS interpolator [12]. Eq. (2) was deduced with the first-order approximation of the Taylor's expansion, Cheng, Tsai and Kuo [13] worked out the second-order approximation. Zhao, Zhu and Ding [14] made a step further by proposing a compensation scheme to reduce the truncation error. However, because the Taylor's expansion is performed recursively, the error will not accumulate. If T is very small (1 ms for the RTIPC) and the curve does not have extremely large curvature, the second-order approximation is adequate (The error is $\frac{T^3}{6}\dddot{u}(t)$, t is an indeterminate value between kT and $(k+1)T$). As a result, this paper will use the second-order approximation of the recursive Taylor's expansion for its efficiency and high accuracy. The problem now is how to schedule the feedrate within the Taylor's expansion while respecting the constraints.

Yeh and Hsu [15] provided the idea on how to confine the contour error based on the curvature. Yong and Narayanaswami [16] improved the idea by detecting feedrate sensitive corners offline. Nam and Yang [17] proposed a recursive trajectory generation method to limit the jerk, while Liu and co-workers [18] developed the time interval modulation method to achieve the same goal. However, none of them have dealt with all the important constraints. To respect as more constraints as possible, many researchers have developed various complicated algorithms. Sencer and co-workers [19] aimed to obtain the analytic solution to the feedrate which was expressed in cubic B-Spline form, but the process was computationally inefficient, the algorithm was run on the Matlab with long processing time. Sun and co-workers [20] proposed the curve evolution based feedrate scheduling method, it was an off-line process as well. Beudaert and co-workers [21] developed the velocity profile optimization (VPOp) method, which is an iterative algorithm that computes the intersection of the given constraints. However, the contour accuracy was not considered and the efficiency of this method was not reported.

The lookahead technique is commonly used to alleviate the computation load while respecting the constraints. Lin and co-workers [22] proposed a real-time lookahead technique for their NURBS interpolator, the curve was split into sub-curves according to the curvature. Annoni and co-workers [23] used similar lookahead technique to acquire segmented small curves and then chose different feedrate for each sub-curve. However, the curvature is usually changing continuously along the curve, the changing rate within each sub-curve is not uniform, so the above methods did not make the most of the continuity of the curve. Jin and co-workers [24] realized their lookahead method by calculating the length of deceleration repeatedly at each interpolation period. This kind of lookahead technique can determine deceleration position rapidly and tend to achieve near time-optimal feedrate profile. However, they did not consider the lookahead length and the numerical integration issues.

This paper presents a novel real-time interpolator which is applicable to not only NURBS but also general parametric curves. The interpolator will fully take into consideration of constraints from machine dynamics (axial velocities, axial accelerations and jerk) and contour error while maintaining the feedrate as high as possible. The dynamic lookahead length technique, the numerical integration error consideration, the multi-cases design for feedrate lookahead and intelligent activation of the acceleration lookahead are introduced for the first time, which greatly enhance the interpolation efficiency and accuracy.

Recently, PVT interpolation has been promoted by industrial CNC manufacturers, such as the Delta Tau Power PMAC [25] and Aerotech A3200 [26], etc. It enables the user to specify arbitrary tool path, thus enhances the CNC's capability in dealing with freeform surfaces. It will be used as the benchmark in this paper.

The rest of the paper is organized as follows: the detailed description of the proposed interpolator is given in Section 2; the numerical simulation is performed in Section 3; the real-time performance test, machining trials and discussions are presented in Section 4; Section 5 concludes the paper.

2. Algorithms for the RTIPC

2.1. Overall design of the RTIPC

Fig. 2 shows the overall design of the RTIPC. The parametric curve $\boldsymbol{C}(u)$ is transferred to the RTIPC with proprietary G-code. Code examples are given in Section 4. The lookahead length is calculated as the time of deceleration to a full stop. The feedrate lookahead module checks feedrate limits along the curve. Intermediate results are saved in a buffer to avoid repeated calculation. New data is added when lookahead length exceeds the buffered length, while outdated data is removed. The initial feedrate modification decisions are made by comparing the current and lookahead motion states. Decisions can be categorized into multi-cases,

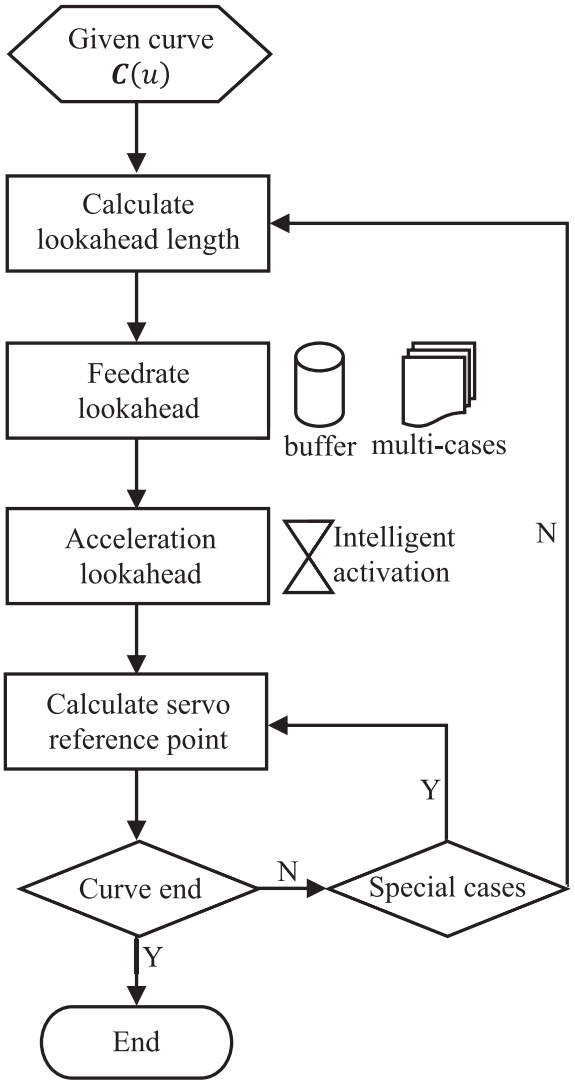


Fig. 2. Overall design of the RTIPC.

among which some special cases will bypass future lookahead operations for certain periods of time. Then, the acceleration lookahead module validates the initial decision by checking it against acceleration and jerk limits. The decision will be changed if the limits are to be violated. The acceleration lookahead will be intelligently activated when it is necessary. The final decision is fed to the module that generates servo reference points. A succession of reference points are generated without lookahead operations for special cases.

2.2. Dynamic lookahead length

The lookahead length should be long enough to scan possible feedrate abrupt changing points ahead, so that the interpolator has sufficient distance to schedule a deceleration profile without violating any constraints. However, excessive length results in unnecessary computation load. In the worst case, a full stop is required, e.g. curve end. So the dynamic lookahead length is calculated as the time required for a full stop. It guarantees reliable response of the RTIPC to any circumstances of the curve, as well as confidence of maintaining high feedrate. Meanwhile, it decreases computation load dramatically compared with constant lookahead length.

Given the current motion state (feedrate F_0 and tangent acceleration $a_{t,0}$), the deceleration profile for a full stop is determined. The conservative tangent deceleration A_t^{csv} is set to the minimum value among axial acceleration limits, it is used as the maximum deceleration value and results in the conservative lookahead length. It can be deduced that if the inequality (3) is satisfied, the deceleration profile is a trapezoidal one. Otherwise, the triangular deceleration profile is resulted.

$$\frac{2A_t^{csv^2} - a_{t,0}^2}{2J} < F_0 \tag{3}$$

Where J is the maximum jerk. In case of the trapezoidal profile, the lookahead length is calculated by

$$t_{look} = \frac{F_0}{A_t^{csv}} + \frac{a_{t,0} + A_t^{csv}}{J} + \frac{a_{t,0}^2}{2A_t^{csv}J} \tag{4}$$

In case of the triangular profile, the lookahead length is given by

$$t_{look} = \frac{a_{t,0} + \sqrt{4F_0J + 2a_{t,0}^2}}{J} \tag{5}$$

The number of lookahead points is the quotient of t_{look} divided by T.

2.3. Feedrate limit

The feedrate limit module has considered the command feedrate, the maximum axial velocity limits, contour error tolerance, and centripetal acceleration limit. Moreover, it has the flexibility to add other constraints.

Firstly, the feedrate should not exceed the command feedrate.

$$F_{lmt,1} = F_{cmd} \tag{6}$$

where $F_{lmt,i}$ denotes the *ith* feedrate limit, and F_{cmd} is the command feedrate specified by the user.

The second constraint on feedrate is the maximum axial velocity.

$$F_{lmt,2} = \min\left[\frac{V_x}{|\alpha|} \quad \frac{V_y}{|\beta|}\right] \tag{7}$$

where min[ ] denotes the minimum value in the square bracket, V_i is the maximum velocity of axis i, α and β are the first derivative factors given by

$$\begin{cases} \alpha = \dfrac{\dot{x}(u)}{|\dot{\boldsymbol{C}}(u)|} \\ \beta = \dfrac{\dot{y}(u)}{|\dot{\boldsymbol{C}}(u)|} \end{cases} \tag{8}$$

Fig. 1 gives the third constraint which is the contour accuracy.

$$F_{lmt,3} = \frac{2}{T}\sqrt{2\rho\varepsilon - \varepsilon^2} \tag{9}$$

where ρ is the radius of curvature. Since centripetal acceleration is proportional to the square of feedrate, limiting feedrate helps to reduce the risk of acceleration saturation. Representing axial accelerations with tangent acceleration a_t and centripetal acceleration a_n yields

$$\begin{cases} a_x = \alpha a_t - \beta a_n \\ a_y = \beta a_t + \alpha a_n \end{cases} \tag{10}$$

The acceleration values in Eq. (10) are signed scalars. Positive a_x and a_y indicate accelerations toward positive axis directions, while a positive a_t means feedrate is increasing. Particularly, a_n has the same sign as the signed curvature. Apply triangle inequality rule to Eq. (10) yields

$$\begin{cases} |a_x| \le |\alpha a_t| + |\beta a_n| \le A_x \dfrac{\delta y}{\delta x} \\ |a_y| \le |\beta a_t| + |\alpha a_n| \le A_y \end{cases} \tag{11}$$

where A_i is the maximum acceleration of axis i. From Eq. (11), the limit of a_n can be given by

$$\begin{cases} |\beta a_n| \le \lambda A_x \\ |\alpha a_n| \le \lambda A_y \end{cases} \quad \lambda \in (0,\ 1) \tag{12}$$

where λ is a safety factor, the lower λ is, the lower centripetal acceleration will be. Considering the relationship between a_n and F, the fourth feedrate limitation is given by

$$F_{lmt,4} = min\left[\sqrt{\frac{\lambda\rho A_x}{|\beta|}} \quad \sqrt{\frac{\lambda\rho A_y}{|\alpha|}}\right] \tag{13}$$

Combining all the limits together, the final limit of feedrate for a point on the curve is given by

$$F_{lmt} = min[F_{lmt,1} \quad F_{lmt,2} \quad F_{lmt,3} \quad F_{lmt,4}] \tag{14}$$

F_{lmt} will be used in the feedrate lookahead module. It is simple to achieve other features, such as cutting force limit, by adding the extra constraints in Eq. (14).

2.4. Acceleration limit

Tangent acceleration is usually used to schedule the feedrate and confine axial accelerations. Transform Eq. (10) yields the representation of a_t

$$\begin{cases} a_t = \dfrac{a_x}{\alpha} + \dfrac{\beta}{\alpha}a_n \\ a_t = \dfrac{a_y}{\beta} - \dfrac{\alpha}{\beta}a_n \end{cases} \tag{15}$$

In Eq. (15), a_n is determined by the expected feedrate and the radius of curvature at one point on the curve. a_t reaches its limit when one of the axes saturates in acceleration, the maximum tangent acceleration is given by

$$A_t^{max} = min\left[\frac{A_x}{|\alpha|} + \frac{\beta}{\alpha}a_n \quad \frac{A_y}{|\beta|} - \frac{\alpha}{\beta}a_n\right] \tag{16}$$

and the minimum tangent deceleration is given by

$$A_t^{min} = max\left[-\frac{A_x}{|\alpha|} + \frac{\beta}{\alpha}a_n \quad -\frac{A_y}{|\beta|} - \frac{\alpha}{\beta}a_n\right] \tag{17}$$

where $max[\,]$ denotes the maximum value in the square bracket. Particularly, A_t^{min} is set to $-A_t^{csv}$ when it is higher than the later, this is to guarantee that feedrate can be decreased fast enough to avoid violating feedrate limits in rare cases. A_t^{max} and A_t^{min} will be used in the acceleration lookahead module.

2.5. Numerical integration error

Integration plays an important role in lookahead operations, as feedrate is the integral of tangent acceleration over time, while distance is the integral of feedrate over time. However, the numerical integration error used in the digital interpolator has always been ignored by existing works. According to the integral rules

$$\begin{cases} a_t = a_{t,0} \pm Jt \\ F = F_0 + \int_0^t a_t dt = F_0 + a_{t,0}t \pm \dfrac{1}{2}Jt^2 \\ S = \int_0^t F dt = F_0 t + \dfrac{1}{2}a_{t,0}t^2 \pm \dfrac{1}{6}Jt^3 \end{cases} \tag{18}$$

where plus is used if a_t is to be increased, otherwise minus is used. However, for digital interpolators, those values are accumulated at each interpolation period

$$\begin{cases} a_{t,k} = a_{t,k-1} \pm JT \\ F_k = F_{k-1} + a_{t,k}T \\ \Delta S_k = F_k T \end{cases} \tag{19}$$

where $a_{t,k}$, F_k and ΔS_k denote the tangent acceleration, feedrate and distance increment in the *kth* interpolation period, respectively. If Eq. (18) were adopted to estimate feedrate and distance-to-go in lookahead operations, the results would deviate from actual output values calculated by the interpolator. The error accumulates with time, and will cause undesirable motion profiles due to incorrect lookahead outcome. One possible solution is to decrease the interpolation period further, so that the numerical integration error will accumulate more slowly. However, this solution imposes much more computation load on the interpolator. For the RTIPC, a new approach is developed, iterating Eq. (19) yields

$$\begin{cases} a_{t,N} = a_{t,0} \pm NJT \\ F_N = F_0 + Na_{t,0}T \pm \dfrac{1}{2}N(N+1)JT^2 \\ S_N = NF_0T + \dfrac{1}{2}N(N+1)a_{t,0}T^2 \pm \dfrac{1}{6}N(N+1)(N+2)JT^3 \end{cases} \tag{20}$$

where S_N is the total distance that will be travelled in N interpolation periods. The result of Eq. (20) will be identical to the output of the interpolator, which guarantees accurate and reliable lookahead operations.

2.6. Feedrate lookahead

The feedrate lookahead module with multi-cases design is introduced for the first time. The lookahead operation takes place at every interpolation period, but when special cases occur, it can be skipped for certain periods, which will increase the efficiency significantly. Fig. 3 shows the algorithm of the feedrate lookahead module, three cases are resulted by comparing F_{lmt} at the lookahead point and F_0 at current point. Case A implies that the feedrate can be increased, Case B represents that feedrate is supposed to be decreased, and Case C suggests that feedrate keep constant. The feedrate lookahead loop ends when either one of the cases concludes an exit situation or the lookahead length reaches its end.

2.6.1. Case A: feedrate increases

Fig. 4 illustrates the algorithm for Case A. If the current feedrate is increasing, it is vital to check whether the acceleration should be decreased immediately to avoid feedrate overshoot. Suppose that the feedrate reaches F_{lmt} after N interpolation periods, N becomes the independent variable in the second function of Eq. (20), which yields

$$\frac{JT^2}{2}N^2 + \left(\frac{JT^2}{2} - a_{t,0}T\right)N + F_{lmt} - F_0 = 0 \tag{21}$$

If the discriminant Δ is less than zero, it implies that the target feedrate cannot be reached, which results in Case A.1, where some distance is still available to increase the feedrate further. Otherwise, the target feedrate can be reached twice, the roots are N_1 and N_2 respectively. If the lookahead distance S_{look} falls between distance-to-go S_{N1} and S_{N2}, the actual feedrate at the lookahead point will exceed F_{lmt}, which leads to Case A.2, where feedrate increasing rate must be lowered, the feedrate lookahead loop can exit in advance. Note that Case A.2 is a special case, the lookahead operations will be bypassed for $N1$ interpolation periods. If S_{look} falls outside the span, the actual feedrate at the lookahead point will be lower than F_{lmt}. Case A.3 deals with the situation that current feedrate is decreasing, the deceleration rate can be decreased to achieve a U-turn of tangent acceleration. Case A.4 deals with the situation that

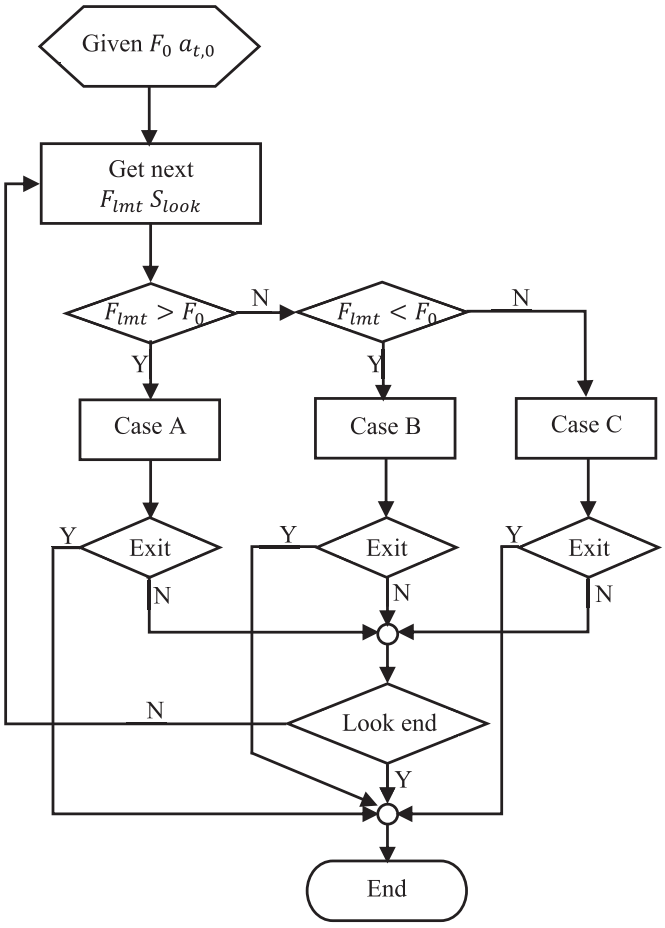


Fig. 3. Algorithm for the feedrate lookahead module.

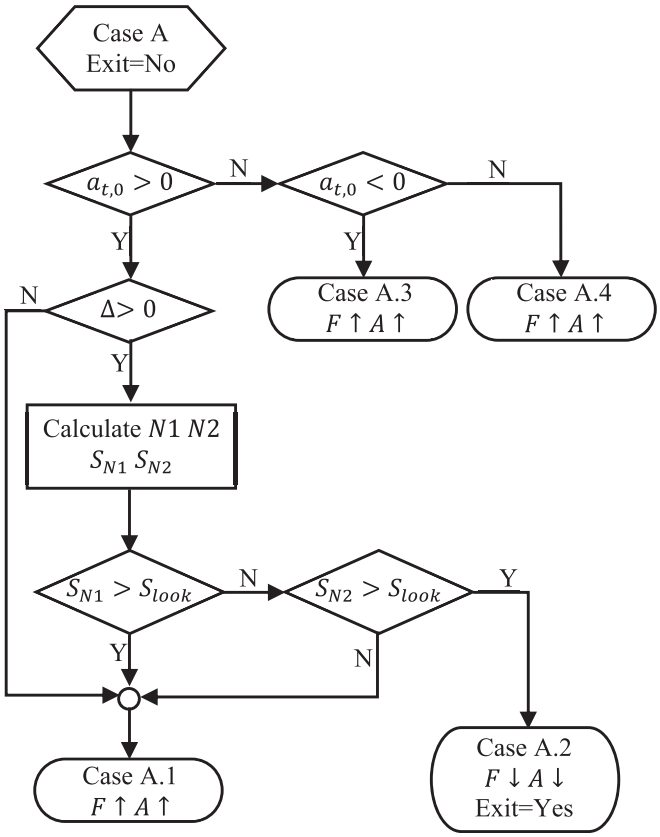


Fig. 4. Algorithm for case A.

current feedrate is constant, it suggests initiating acceleration again.

2.6.2. Case B: feedrate decreases

Fig. 5 shows the algorithm for Case B. When the feedrate is expected to be lowered, it is of great importance to check whether the acceleration should keep decreasing to avoid possible violation. There is only one valid root for Eq. (21) in this case. If the planned deceleration distance is longer than S_{look}, the actual feedrate at the lookahead point will exceed F_{lmt}, which are Case B.1 and B.3, the acceleration must keep decreasing and the lookahead operation can exit now. Otherwise, there are some distance to boost the feedrate. Note that the allowable deceleration value varies with curvature, when the planned deceleration exceeds A_t^{csv}, there is a risk of saturating axes in acceleration. In this case, a simulation of the deceleration process is necessary to determine the distance-to-go accurately, i.e. calculate N' and $S_{N'}$. It usually happens when F_{lmt} is much lower than the current feedrate, e.g. curve end. Both Case B.1 and B.3 are special cases. Case B.1 can bypass N lookahead operations, while Case B.3 can bypass lookahead operations until the actual deceleration reaches A_t^{csv}.

Particularly, a three-stage stop strategy is used when approaching the curve end. The interpolator will increase the jerk slightly in the first stage to decelerate to a low feedrate, then keep the low feedrate in the second stage, and at last applying a triangular deceleration profile to a full stop. This strategy can avoid the situation that feedrate drops to an extremely low value before the end and takes long time to the end, or motion stops abruptly at the end with relatively high feedrate.

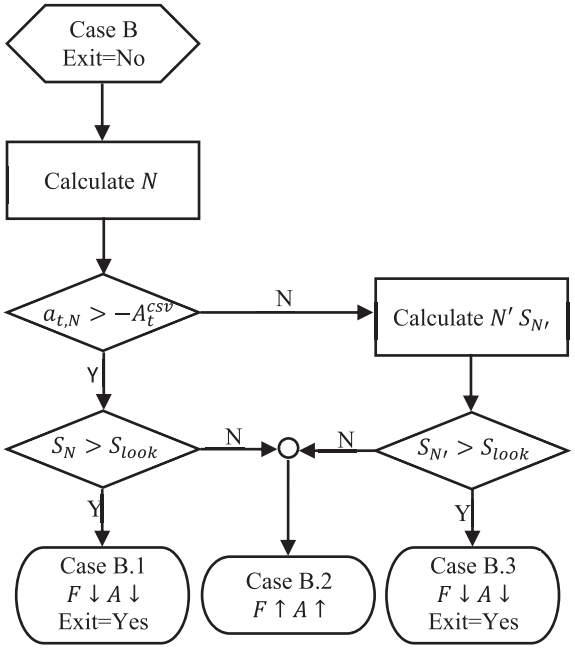


Fig. 5. Algorithm for case B.

2.6.3. Case C: feedrate is constant

When the feedrate is suggested to keep constant, the tangent acceleration should be scheduled toward zero. The concluded cases here are not mandatory. Particularly, if the lookahead operation reaches the end but no special case occurs, the predominant case will be issued to the next module.

2.7. Acceleration lookahead

Like the feedrate lookahead operation, it is not necessary to perform the acceleration lookahead at each interpolation period when there is no risk of violation of the acceleration or jerk limits. This measure can significantly reduce computational load. The acceleration lookahead module is only activated when one of the following feedrate modification decisions is issued:

(1) Increase feedrate with increasing acceleration, while the tangent acceleration is close to A_t^{max};
(2) Decrease feedrate with decreasing acceleration, while the deceleration value exceeds A_t^{csv}.

For other situations, this module is deactivated to decrease the computation burden. The curvature varies along the curve, so does the acceleration limits. Suppose that the acceleration is increasing, it is vital to check whether it is the time to decrease it to avoid possible saturation.

$$a_{t,0} - iJT > A_{t,i}^{max} \tag{22}$$

where $A_{t,i}^{max}$ is the maximum acceleration at the *ith* lookahead point, which is given by Eq. (16). If this inequality is true, it means that even the acceleration is decreased from now, the limit at the lookahead point will be violated. The tangent acceleration should be reduced immediately in this case, otherwise the original decision can be kept. Similarly, suppose that the acceleration keeps decreasing, the below criterion is used to check possible violation.

$$a_{t,0} + iJT < A_{t,i}^{min} \tag{23}$$

The final lookahead result will be used to calculate the feedrate for the next interpolation period, and the servo reference point can be calculated with the feedrate value using the second-order approximation of the Taylor's expansion.

3. Numerical simulation

The demonstration software (open access information is in the Acknowledgment section) has been developed to implement the proposed RTIPC algorithm. Specifying the curve definition as input, the software will generate the actual servo reference point set, as well as internal feedrate lookahead and acceleration lookahead data. The numerical simulation is performed to study the characteristics of the scheduled motion profiles, and to show the effectiveness of the lookahead operations. The interpolator parameters are set as shown in Table 1, and the command feedrate is 20 mm/s.

Two typical parametric curves are selected for both numerical simulation and experimental validation. The teardrop curve is an example of polynomial curves, which are usually used in curve fitting. The ribbon curve is an example of B-Splines, which are used in geometric data exchange. The teardrop curve is given by

$$\begin{cases} x(u) = -150u + 450u^2 - 300u^3 \\ y(u) = -150u + 150u^2 \end{cases} \quad u \in \left[0, 1\right] \tag{24}$$

Fig. 6 shows the interpolation results of the teardrop curve. The curvature reaches the maximum value for the first time at point A, as shown in Fig. 6(a). The feedrate is limited to the lowest value at point A accordingly, as shown in Fig. 6(c). The interpolator plans s-shaped feedrate profiles at the beginning and end of the curve, as shown in Fig. 6(b). In contrast to the linear or circular interpolation, the tangent acceleration limit varies during the s-shaped profile planning, as shown in zone 1 and 7 of Fig. 6(d), which makes it extremely difficult to decide when the tangent acceleration should be increased or decreased to avoid feedrate overshoot or undershoot. This problem is solved by the lookahead operations and the numerical integration technique in this paper. The three-stage stop strategy helps to achieve the accurate and smooth stop, as shown in zone 2 of Fig. 6(b).

The feedrate is maintained as high as possible, and the fluctuation is minimized, as shown in Fig. 6(c). The smoothness of the feedrate comes from the confined jerk, as shown in Fig. 6(f). The tangent acceleration reaches limit when one of the axes saturates in acceleration, as shown in Fig. 6(d) and (e). The acceleration of X axis exceeds the limit slightly in zone 6 of Fig. 6(e), because the interpolator has a delay when reacting to the error, just like any other digital controller.

The ribbon curve is constructed by a 3rd degree B-Spline with control points $\{\boldsymbol{P_i}\} = \{(-15, 0), (20, 30), (0, 50), (-20, 30), (15, 0)\}$ and knot vector $U = \{0, 0, 0, 0, 0.5, 1, 1, 1, 1\}$. Fig. 7 shows the interpolation results of the ribbon curve. Both examples achieve smooth velocity profiles and confined axial accelerations and jerk profiles. Actually, the time-optimal solution of the interpolation for parametric curves is that at least one of the values reaches its limit at any time in the velocity, acceleration and jerk profiles, and the RTIPC achieves the near time-optimal solution, as shown in Fig. 7(b), (e) and (f).

Fig. 8 shows the contour error of the interpolation for both curves. The error is small at the beginning and end as the feedrate and curvature are small. The error increases with increasing feedrate and reaches the maximum value at the points with largest curvature. The maximum contour error for both curves is below 4 nm, which is within the tolerance (10 nm).

Table 1
Interpolator parameters.

X/Y velocity limit (mm/s)	X/Y acceleration limit (mm/s^2)	Jerk limit (mm/s^3)	Contour error tolerance (μm)
30	30	200	0.01

4. Experiment

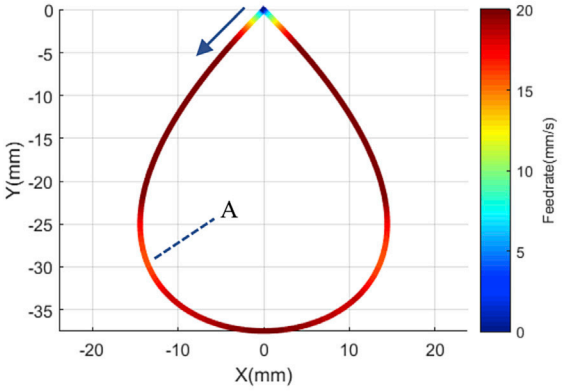
4.1. Real-time performance test

The real-time performance test is performed to show how efficient the RTIPC is. The RTIPC has been implemented in the in-house developed controller. The controller uses an ARM® Cortex®-M7 microcontroller as the main processor, which runs at 216 MHz. An external 16 MB memory is connected to the microcontroller to extend its ability to handle large data buffers such as the lookahead buffers. A Field Programmable Gate Array (FPGA) is integrated to generate pulse and direction signals to the motor drives. By contrast, the Delta Tau Power PMAC controller (model: Power Clipper) is used in the machining comparison experiment. It features a 1.0 GHz dual-core Power PC CPU and 1 GB memory.

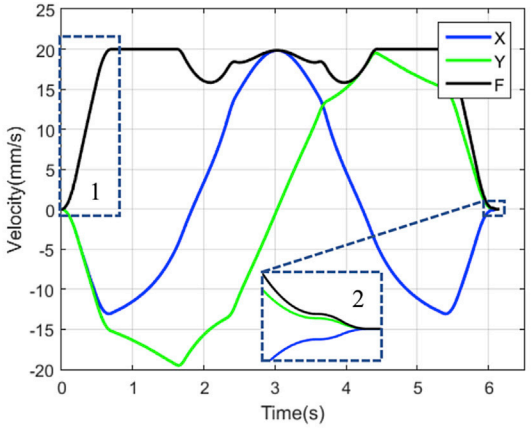
The same interpolator parameters shown in Table 1 are used. The command feedrate is set as 2 mm/s. Table 2 shows the interpolation time for both curves using the controller. The average cycle time for both curves is around 1 ms. Due to the complexity of B-Spline evaluation, the ribbon curve consumes more cycle time. Note that the algorithm is implemented using double precision floating-point (DPFP) arithmetic to achieve the best accuracy. However, the currently used microcontroller does not have a built-in DPFP unit, which can boost the algorithm execution by 7.2 times [27]. The real-time performance test on the current controller has approved that the developed interpolator is efficient enough to achieve the real-time characteristic on a more advanced controller, such as the multi-core digital signal processor (DSP) with built-in DPFP unit.

4.2. Machining trial

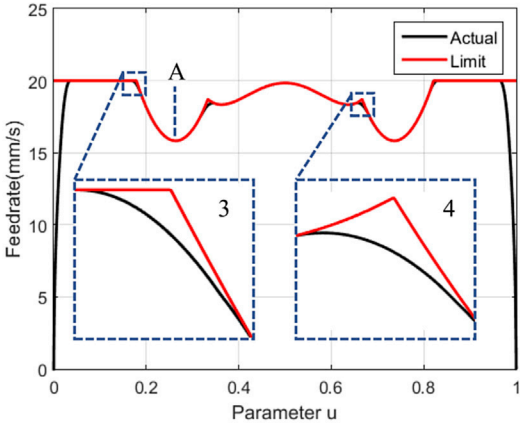
To further assess the performance of the RTIPC, a series of machining experiments are carried out on a desktop milling machine. In addition to the above two parametric curves, a cubic phase plate freeform surface, as shown in Fig. 9(a), is also machined. The profile of the surface in XZ



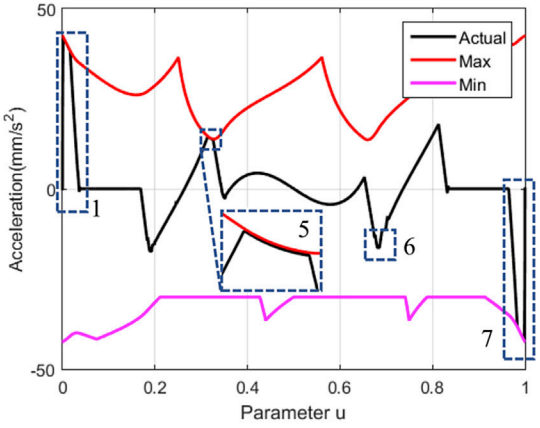
(a) Feedrate variation along the curve



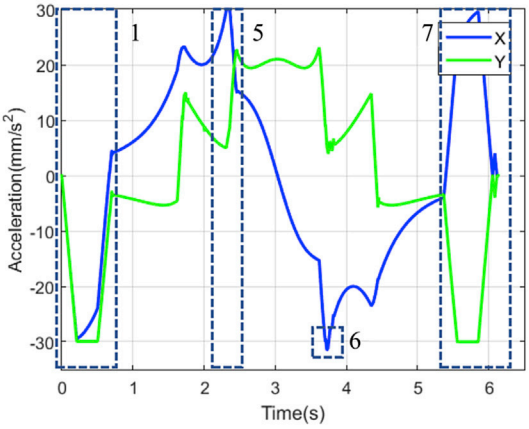
(b) Velocity profile



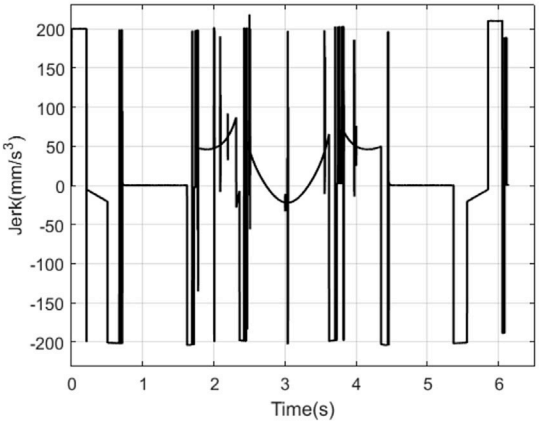
(c) Feedrate lookahead



(d) Acceleration lookahead



(e) Acceleration profile



(f) Jerk profile

Fig. 6. Interpolation results of the teardrop curve.

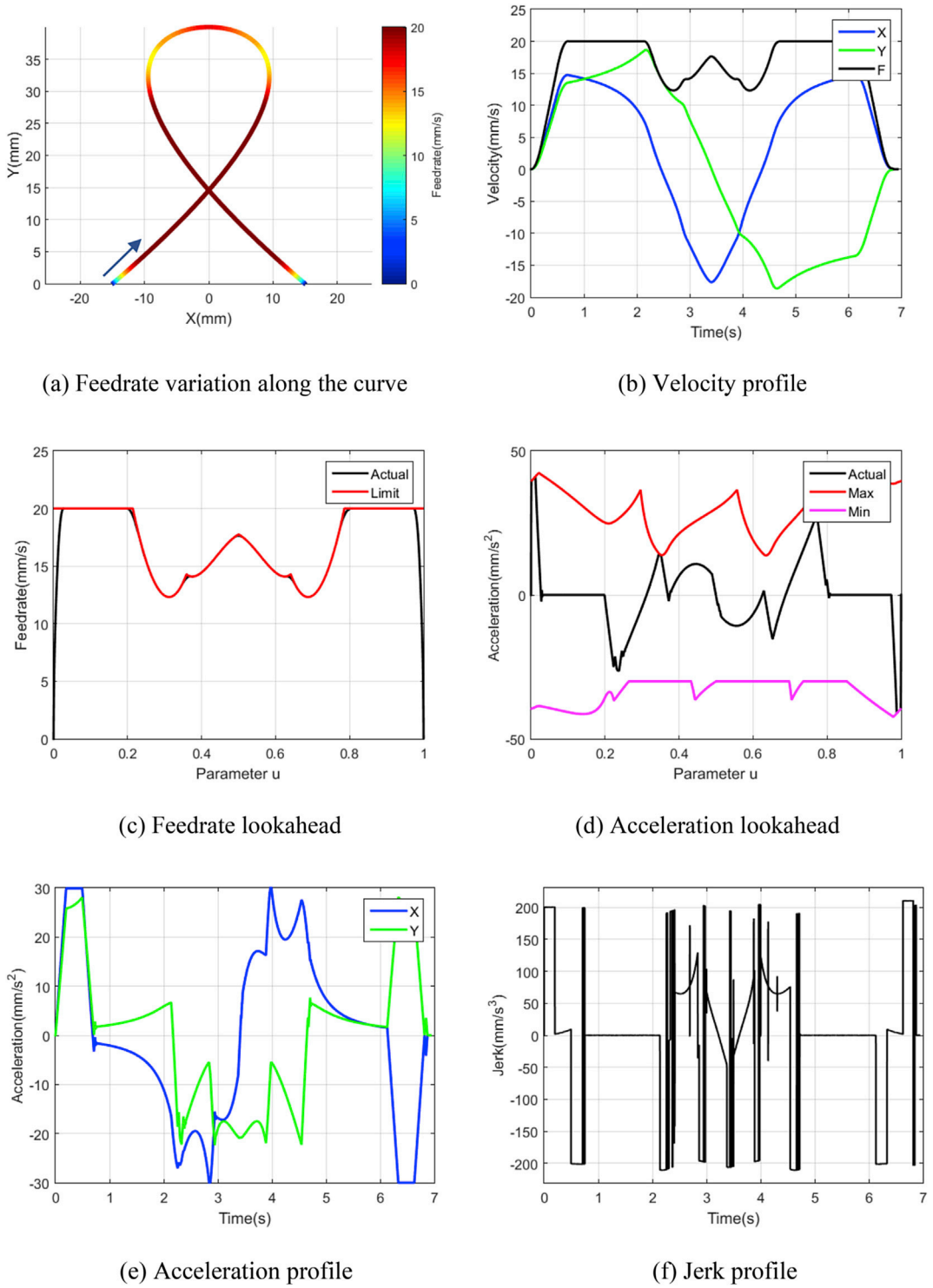


Fig. 7. Interpolation results of the ribbon curve.

plane is a polynomial curve, as shown in Fig. 9(b), which is given by Eq. (25). The tool path for the finish machining of the surface can be composed of a series of polynomial curves by spacing Y uniformly.

$$\begin{cases} x(u) = u \\ z(u) = 0.007y^3 + 0.007u^3 \end{cases} \quad u \in [-5,\ 5] \tag{25}$$

The machining results are compared with the results obtained from the Delta Tau Power PMAC controller, which has the industrial standard linear interpolator and the state-of-the-art PVT interpolator. The PVT interpolation is a special case of Hermite interpolation, it provides low level interface for users to specify the tool path and the motion profiles. Users should supply the exact position and velocity at the boundaries of each segment, as well as the travel time for the segment. Suppose that time is normalized to $[0,\ 1]$, a PVT move is given by

$$\boldsymbol{C}(t) = [1 \quad t \quad t^2 \quad t^3] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -3 & -2 & -1 & 3 \\ 2 & 1 & 1 & -2 \end{bmatrix} \begin{bmatrix} \boldsymbol{P}(0) \\ \dot{\boldsymbol{P}}(0) \\ \dot{\boldsymbol{P}}(1) \\ \boldsymbol{P}(1) \end{bmatrix} = \sum_{i=0}^{3} \boldsymbol{a}_i t^i \tag{26}$$

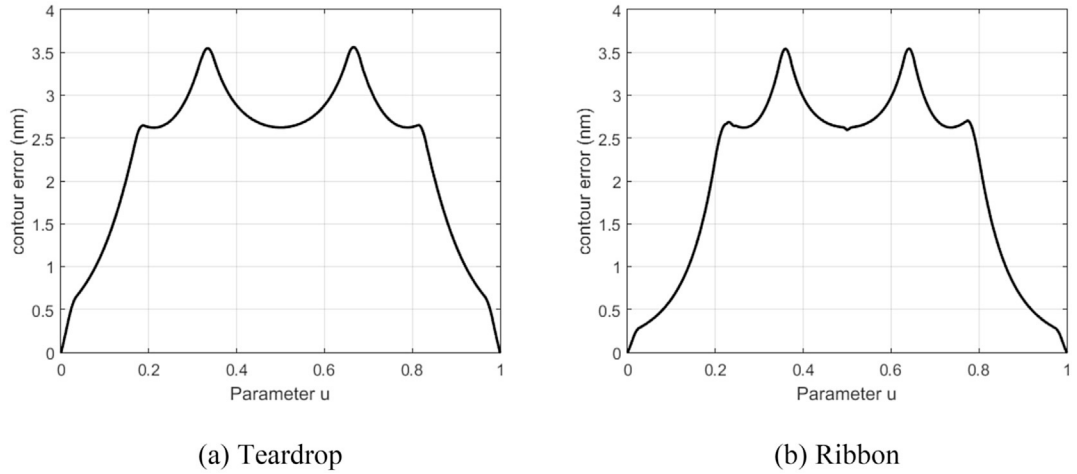


Fig. 8. Contour error of the interpolation.

Table 2
Real-time performance test.

Curve	Teardrop	Ribbon
Total points	51176	55342
Total time (ms)	37236	62846
Max. cycle time (ms)	4	11
Average cycle time (ms)	0.728	1.136

where $\boldsymbol{P}(0)$ and $\boldsymbol{P}(1)$ are the start and end points of the segment, $\dot{\boldsymbol{P}}(0)$ and $\dot{\boldsymbol{P}}(1)$ are the corresponding velocities, $\{\boldsymbol{a}_i\}$ are the coefficients related with the positions and velocities.

4.2.1. Machining trial setup

Fig. 10 shows the machining trial setup which is taking place on a three-axis desktop machine. The machine has a stacked X and Y axes to carry the workpiece. A high-speed air-bearing spindle is mounted on the vertical Z axis. The three linear axes are driven by stepper motors. Due to the adoption of open loop control, the influence of the servo algorithm on the performance of the interpolation algorithm is eliminated. A command feedrate of 2 mm/s, a spindle speed of 12,000 rpm, a depth of cut of 0.1 mm, and a milling cutter with a diameter of 2 mm are used in the curve machining experiment. A 1 mm diameter ball end mill is used in the finish machining of the freeform surface, while the machining allowance and the cutting width along Y axis are 20 μm and 10 μm, respectively. The workpiece material is aluminum. In this paper, proprietary G-code are developed for the RTIPC. The CAM software – Pro/ENIGNEER Wildfire 4.0 is used to generate the linear interpolation code. As the CAM does not support PVT interpolator, a customized program is developed to generate PVT code for those curves and the surface.

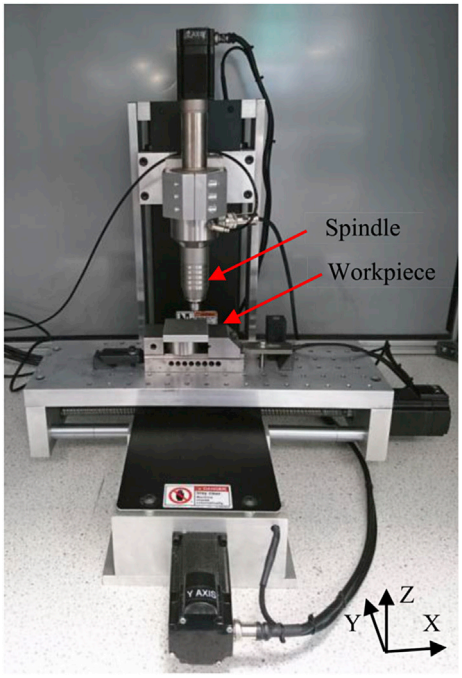


Fig. 10. Machining trial setup.

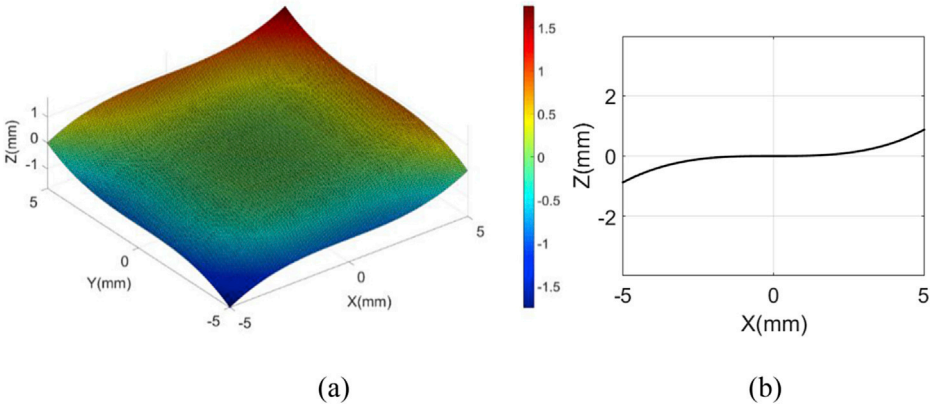


Fig. 9. (a) Cubic phase plate $z = 0.007(x^3 + y^3)$ $x \in [-5, 5]$ $y \in [-5, 5]$ (b) Surface profile at $y = 0$.

Table 3
Teardrop curve interpolation comparison.

	RTIPC	Linear	PVT
NC code	G06.1 X {-150*U+450*U2–300*U3} Y {-150*U+150*U2} U[0 1] F2	G01 X-0.066 Y-.066 F2 X-0.133 Y-.133 X-0.198 Y-.199 X-0.264 Y-.265 ⋮	PVT200 X-0.2823: 1.4088 Y-0.2834: 1.4196 X-0.5635: 1.4033 Y-0.5678: 1.4250 X-0.8436: 1.3977 Y-0.8534: 1.4305 ⋮
Line count	1	2324	256
Machining time (s)	51.2	489.4	51

Table 4
Ribbon curve interpolation comparison.

	RTIPC	Linear	PVT
NC code	G06.2 K0 X-15 Y0 F2 K0 X20 Y30 K0 X0 Y50 K0 X-20 Y30 K0.5 X15 Y0 K1 K1 K1 K1	G01 X-14.902 Y.084 F2 X-14.815 Y.159 X-14.721 Y.239 X-14.634 Y.315 ⋮	PVT200 X-14.6965:1.5162 Y0.2606:1.3043 X-14.3935:1.5138 Y0.5217:1.3070 X-14.0910:1.5114 Y0.7834:1.3098 ⋮
Line count	9	2641	277
Machining time (s)	55.3	556.9	55.2

Table 5
Cubic phase plate freeform surface interpolation comparison.

	RTIPC	Linear	PVT
NC code	G06.1 X{U} Y {-0.875 + 0.007*U3} U[-5 5] F2 G01 Y-4.99 Z0.005 F2 G06.1 X{-U} Y{-0.87-0.007*U3} U[-5 5] F2 ⋮	G01 X4.7261 Z-0.0219 F2 X4.6657 Z-0.0524 X4.605 Z-0.0822 X4.5441 Z-0.1114 ⋮	PVT100 X-4.8216:1.7973 Z-1.6596:0.8774 X-4.6406:1.8223 Z-1.5746:0.8241 X-4.4572:1.8458 Z-1.4948:0.7701 ⋮
Line count	2001	122070	54068
Machining time (s)	5640	20820	5400

4.2.2. Machining trial results and discussions

Tables 3–5 list the code and machining time for the specified curves and surface. The RTIPC can reduce the code size in three orders of magnitude compared with the linear interpolator and two orders of magnitude compared with the PVT interpolator. In addition, the RTIPC code are much more concise and readable. The RTIPC is as productive as the PVT interpolator, both can achieve up to ten times productivity increase over the linear interpolator. The machined workpieces are shown in Figs. 13–15 in Appendix A.

The RTIPC is also superior to the linear interpolation and PVT interpolation in terms of motion smoothness. Fig. 11(a) shows the motion profiles of linear interpolation for the first segment of the teardrop curve. Triangular acceleration profile is observed as the segment is too short. The attainable feedrate is 0.933 mm/s which is much lower than the target value (2 mm/s). Moreover, such motion profiles occur in each segment, which decreases average feedrate and causes vibration.

The feedrate and acceleration profiles in each PVT segment are a quadratic polynomial and a linear function respectively, as can be deduced from Eq. (26). Although feedrate and acceleration are continuous within the segment, they are not necessarily continuous at the boundaries, as the coefficients in Eq. (26) vary for different segments. Fig. 11(b) shows the motion profiles of PVT interpolation for the first two segments of the teardrop curve, each segment takes 0.2s. The feedrate is maintained to be higher than the linear interpolation, but it is not smooth at the boundaries, and overshoot is observed. Moreover, the discontinuity of acceleration leads to infinite jerk, which causes shock to the machine.

Fig. 16 in Appendix A shows the surface topography at the center of the cubic phase plate measured by a white light interferometer (Zygo CP200 with 5× objective). The field of view is 1.45 mm × 1.08 mm. The surface profiles at $y = 0$ are extracted as shown in Fig. 12. The maximum absolute form errors of the profile machined by linear interpolator, PVT and RTIPC are 2.45 μm, 6.82 μm and 2.33 μm, respectively. The standard deviations of the form error are 0.927 μm, 3.20 μm and 0.764 μm, respectively. It shows that the RTIPC achieves the best form accuracy. The result is related with the motion smoothness which is determined by the interpolator. Smoother motions make the stepper motors less likely to lose step. This is especially important when there are no axis feedbacks to be used to correct the motion.

The advantage of the RTIPC is also found in the contour error limit. To achieve higher contour accuracy, the linear interpolator needs more segments to approximate to original surface, which pose huge burden for the CNC memory and decrease the average feedrate further. The PVT interpolator uses cubic polynomial curves to approximate the original surface. The end points of each segment are on the surface, but it is difficult to control the deviation between the polynomial curve and the surface. Moreover, the PVT interpolation is error-prone, as it requires to

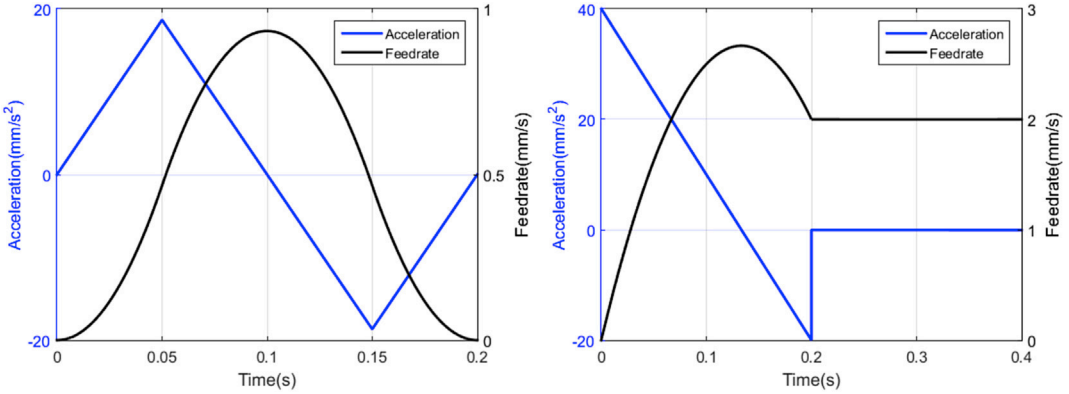


Fig. 11. Acceleration and feedrate fluctuation.

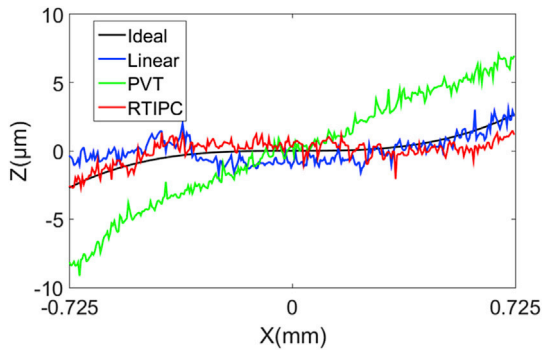


Fig. 12. Surface profiles at the center of the cubic phase plate.

map the parameter of the parametric curve to time. Unsuitable mapping will result in undesirable motion profiles. For the RTIPC, the contour error is limited internally by the lookahead operations, the assistance of CAM is eliminated.

5. Conclusions

The linear interpolator has its intrinsic drawback in acceleration and feedrate fluctuation. The driven demand for the high-speed high-accuracy machining of freeform surfaces makes the interpolator for parametric curves highly desirable. This paper points out that the interpolation of parametric curves is essentially an optimization problem. And the paper proposes a novel solution to the problem using feedrate lookahead and acceleration lookahead operations. The experiments have shown that it can achieve the real-time feature. Moreover, it can not only simplify the coding significantly, but also achieve ten times productivity increase compared with the linear interpolator. The proposed interpolator can also achieve much smoother motion profiles than the state-of-the-art PVT interpolator and has the advantage in limiting contour error. The main advantages of this interpolator comparing with existing works can be summaries as follows:

(1) Its application is not limited to NURBS but general parametric curves with real-time feature;
(2) It limits both machine dynamics (axial velocities, axial accelerations and jerk) and contour error while maintaining the feedrate as high as possible with minimum fluctuation.
(3) The dynamic lookahead length technique, the numerical integration error consideration, the multi-cases design for feedrate lookahead and intelligent activation of the acceleration lookahead are introduced for the first time, which greatly enhance the interpolation efficiency and accuracy.

Acknowledgment

This paper is supported by the Engineering and Physical Sciences Research Council (EPSRC) under the program EP/K018345/1. The demonstration software associated with this research is openly available from https://doi.org/10.15129/75c22cfe-2cee-4527-9c77-9611658876d1.

Appendix A

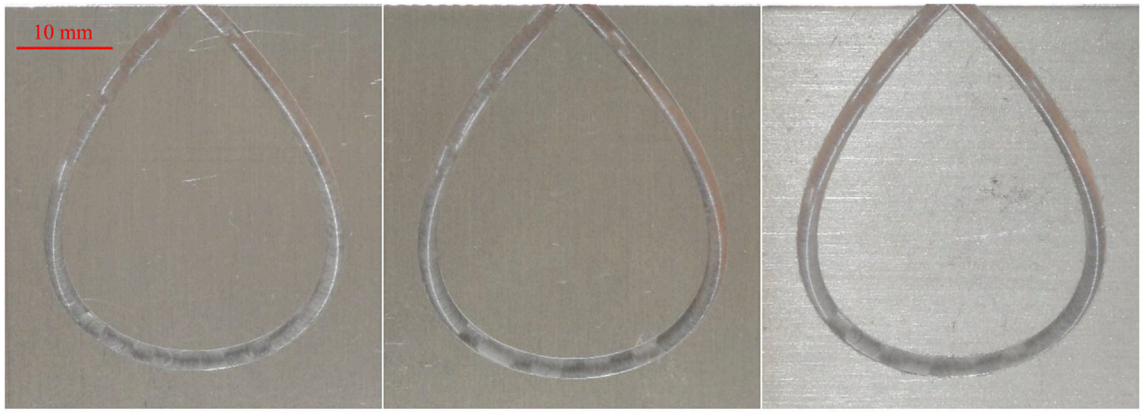


Fig. 13. Teardrop curve machined workpieces.

Fig. 14. Ribbon curve machined workpieces.

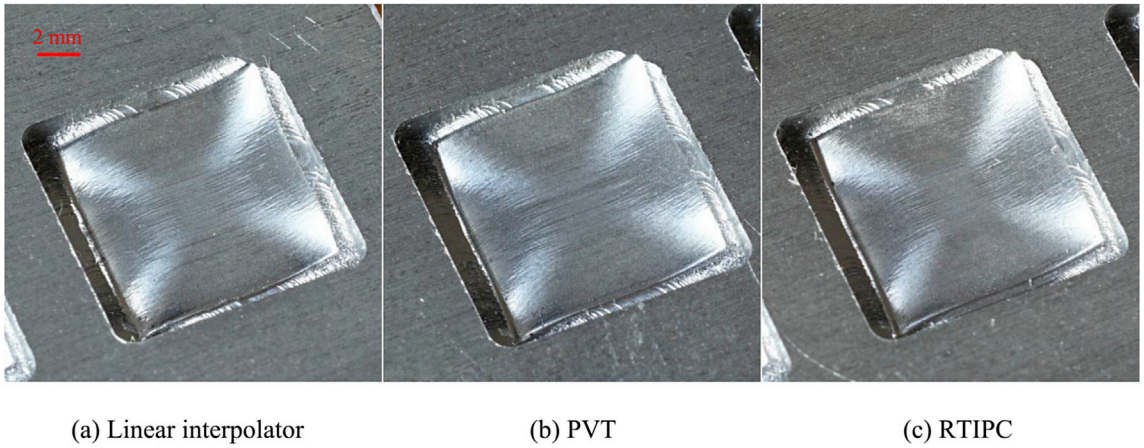


Fig. 15. Cubic phase plate machined workpieces.

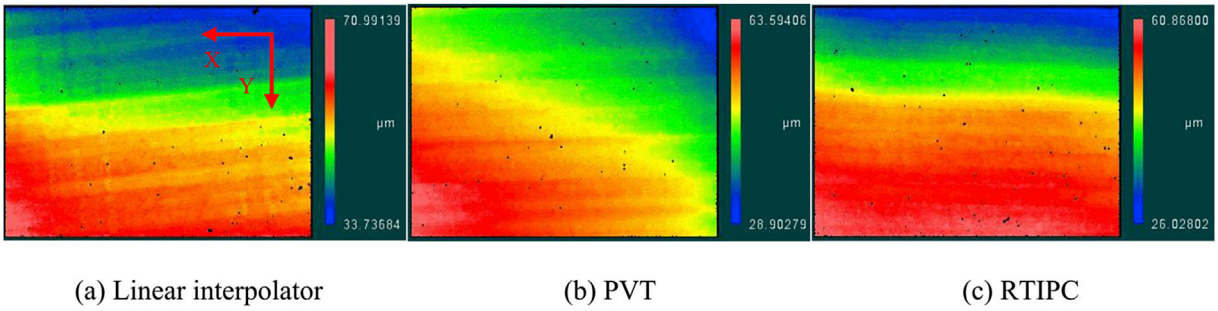


Fig. 16. Surface topography at the center of the cubic phase plate.

References

[1] L. Piegl, W. Tiller, The NURBS Book, second ed., Springer-Verlag New York, Inc., New York, NY, USA, 1997.
[2] C. Brecher, S. Lange, M. Merz, F. Niehaus, C. Wenzel, M. Winterschladen, NURBS based ultra-precision free-form machining, CIRP Ann. - Manuf. Technol. 55 (1) (2006) 547–550.
[3] S. J. Yutkowitz, Apparatus and Method for Smooth Cornering in a Motion Control System, United States patent 6,922,606 B1, 2005.
[4] S.D. Timar, R.T. Farouki, Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds, Robot. Comput. Integr. Manuf. 23 (5) (2007) 563–579.
[5] K. Erkorkmaz, Y. Altintas, Quintic spline interpolation with minimal feed fluctuation, J. Manuf. Sci. Eng. 127 (2) (2005) 339.
[6] K. Erkorkmaz, M. Heng, A heuristic feedrate optimization strategy for NURBS toolpaths, CIRP Ann. - Manuf. Technol. 57 (1) (2008) 407–410.
[7] M. Heng, K. Erkorkmaz, Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability, Int. J. Mach. Tools Manuf. 50 (3) (2010) 281–293.
[8] K. Erkorkmaz, S.E. Layegh, I. Lazoglu, H. Erdim, Feedrate optimization for freeform milling considering constraints from the feed drive system and process mechanics, CIRP Ann. - Manuf. Technol. 62 (1) (2013) 395–398.
[9] M. Liu, Y. Huang, L. Yin, J. Guo, X. Shao, G. Zhang, Development and implementation of a NURBS interpolator with smooth feedrate scheduling for CNC machine tools, Int. J. Mach. Tools Manuf. 87 (2014) 1–15.

[10] R.T. Farouki, Y.F. Tsai, Exact Taylor series coefficients for variable-feedrate CNC curve interpolators, CAD Comput. Aided Des. 33 (2) (2001) 155–165.
[11] M. Shpitalni, Y. Koren, C.C. Lo, Realtime curve interpolators, Comput. Des. 26 (11) (1994) 832–838.
[12] T. Otsuki, H. Kozai, and Y. Wakinotani, Free-form curve interpolation method and apparatus, U. S. Patent 5,815,401, 1998.
[13] M.-Y. Cheng, M.-C. Tsai, J.-C. Kuo, Real-time NURBS command generators for CNC servo controllers, Int. J. Mach. Tools Manuf. 42 (7) (May 2002) 801–813.
[14] H. Zhao, L. Zhu, H. Ding, A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction, Int. J. Mach. Tools Manuf. 75 (2013) 1–8.
[15] S.S. Yeh, P.L. Hsu, Adaptive-feedrate interpolation for parametric curves with a confined chord error, CAD Comput. Aided Des. 34 (3) (2002) 229–237.
[16] T. Yong, R. Narayanaswami, A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining, CAD Comput. Aided Des. 35 (13) (2003) 1249–1259.
[17] S.H. Nam, M.Y. Yang, A study on a generalized parametric interpolator with real-time jerk-limited acceleration, CAD Comput. Aided Des. 36 (1) (2004) 27–36.
[18] X. Liu, F. Ahmad, K. Yamazaki, M. Mori, Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics, Int. J. Mach. Tools Manuf. 45 (4–5) (2005) 433–444.
[19] B. Sencer, Y. Altintas, E. Croft, Feed optimization for five-axis CNC machine tools with drive constraints, Int. J. Mach. Tools Manuf. 48 (7–8) (2008) 733–745.
[20] Y. Sun, Y. Zhao, J. Xu, D. Guo, The feedrate scheduling of parametric interpolator with geometry, process and drive constraints for multi-axis CNC machine tools, Int. J. Mach. Tools Manuf. 85 (2014) 49–57.
[21] X. Beudaert, S. Lavernhe, C. Tournier, Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path, Int. J. Mach. Tools Manuf. 57 (Jun. 2012) 73–82.
[22] M.T. Lin, M.S. Tsai, H.T. Yau, Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm, Int. J. Mach. Tools Manuf. 47 (15) (2007) 2246–2262.
[23] M. Annoni, A. Bardine, S. Campanelli, P. Foglia, C.A. Prete, A real-time configurable NURBS interpolator with bounded acceleration, jerk and chord error, CAD Comput. Aided Des. 44 (6) (2012) 509–521.
[24] Y.A. Jin, Y. He, J.Z. Fu, Z.W. Lin, W.F. Gan, A fine-interpolation-based parametric interpolation method with a novel real-time look-ahead algorithm, CAD Comput. Aided Des. 55 (2014) 37–48.
[25] Power PMAC User's Manual, Delta Tau Data Systems, Inc., California, 2016.
[26] Aerotech, Standard Controller Capabilities, 2017 [Online]. Available, https://www.aerotech.com/product-catalog/motion-controller/standard-controller-capabilities.aspx [Accessed: 20-Jul-2017].
[27] Application note: Floating Point Unit Demonstration on STM32 Microcontrollers, STMicroelectronics, Geneva, Switzerland, 2016.