Int J Adv Manuf Technol (2012) 63:719–729
DOI 10.1007/s00170-012-3936-3

ORIGINAL ARTICLE

An adaptive off-line NURBS interpolator for CNC machining

Jiangang Li · Tinghua Zhang · Zexiang Li

Received: 10 January 2008 / Accepted: 16 January 2012 / Published online: 3 February 2012

Abstract As a key technique in CNC machining, non-uniform rational B-spline (NURBS) interpolator has been proposed to overcome the drawbacks caused by linear and circular interpolator, such as large data size, velocity discontinuity, shocks or variations in mechanical systems, and low machining efficiency. To improve machining quality and efficiency, an adaptive off-line interpolator was developed for NURBS curves in this paper. Both the chord accuracy and the acceleration/deceleration capability of machine tool are considered in the algorithm. There are four modules in the algorithm, adaptive feed rate adjustment, acceleration/deceleration disposal, feed rate modification, and S-type velocity generation. The acceleration/deceleration around the sharp corners is carefully calculated by those modules. A case study was provided to evaluate the feasibility of the interpolator.

Keywords NURBS interpolator · CNC · Adaptive · Modification · S-type velocity

1 Introduction

In modern CAD/CAM systems, machining profiles for parts, such as dies, aircraft models, car models, and turbine blades, are usually represented in parametric curves. High machining quality and efficiency are expected by such representation, since they are more smooth than lines and circles. Especially, nonuniform rational B-spline (NURBS) curve is one such type of parametric curve, and it has many numerical advantages. In recent years, many kinds of NURBS interpolation methods have been proposed in the literatures.

J. Li (✉) · T. Zhang · Z. Li
Harbin Institute of Technology,
Shenzhen, Guangdong, China
e-mail: jiangang_lee@163.com

The parametric curve interpolation algorithms, such as Taylor's expansions, speed-controlled Taylor's expansion, or Runge–Kutta methods, were proposed. All these algorithms have to calculate the first- or second-order derivative of the parametric curve directly, but the computation is very time consuming, especially for high-order NURBS curves. Another disadvantage is that the speed deviation of these algorithms cannot be evaluated by the end users. To overcome these, a predictor–corrector interpolator (PCI) was proposed by Tsai and Cheng [1], which uses a simple algorithm as a predictor, and the current feed rate command feedback compensation scheme as the corrector.

However, these algorithms have not considered chord error explicitly. Yeh and Hsu developed an adaptive parametric curve interpolation algorithm [2], which eliminates the drawbacks. The algorithm uses an approximation method to determine the relationship between chord error and curve speed. But the interpolator does not take the machine's acceleration/deceleration (acc/dec) capacities into account. Abrupt change of the velocity may occur during machining process, especially at the sharp corner of machining profiles. This will require infinite acc/dec, which would exceed the capability of the machine tool. As a result, the machining accuracy would be deteriorated. So an adaptive interpolator considering the acc/dec is needed. Although Yong and Narayanaswami proposed an off-line algorithm [3] to deal with the acc/dec at sharp corners, it cannot be applied to any kinds of NURBS curve. Zhang et al. presented a contour errors analysis and velocity adjustment NURBS curve interpolator [4]. He also implemented a CNC NURBS curve interpolator based on controlled velocity and precision [5]. A method to compute curvature minima and maxima of parametric curves is also presented in [6].

The real-time NURBS interpolation methods were proposed by some researchers as well, a real-time adaptive NURBS interpolator considering acc/dec capacity of each individual axis is proposed to confine both the chord error and the axis acceleration [7]. A real-time NURBS interpolator with a look-ahead function using a PC-based control architecture [8] was proposed by Wang et al. A new interpolation scheme for 2D NURBS curve [9] was proposed by Shen HY. Based on the ILF method and segmentation of NURBS curves, a new interpolation scheme is proposed in this paper.

Only trapezoidal feed rate profile can be realized in real-time NURBS interpolator in modern CNC system, because of its computation limitations. But this will result in discontinuity on the acc/dec profile. It may cause vibration, even break the mechanical system. Although it leads to complex computation, the S-type profile has continuous acc/dec profile.

In this paper, an adaptive off-line NURBS interpolator is proposed. The algorithm not only ensures the chord error in the desired tolerance but also keeps the acc/dec within the capacity of machine tools. The S-type profile for the feed rate is used, which is more smooth than trapezoidal profile. It can be applied to any kinds of NURBS curve, thus it is practical to today's industry.

The rest of the paper is organized as follows. In Section 2, a brief introduction to the principle of parametric curve interpolation is given. In Section 3, the adaptive off-line NURBS interpolator and its modules are discussed in detail. In Section 4, simulation results are provided. Finally, some conclusions are drawn in Section 5.

2 Principle of parametric curve interpolation

The NURBS curves are represented by the following equations:

$$C(u) = \frac{\sum_{i=0}^{n} W_i N_{i,p}(u) P_i}{\sum_{i=0}^{n} W_i N_{i,p}(u)} = \sum_{i=0}^{n} R_{i,p}(u) P_i \tag{1}$$

$$R_{i,p}(u) = \frac{W_i N_{i,p}(u)}{\sum_{i=0}^{n} W_i N_{i,p}(u)} \tag{2}$$

where $\{P_i\}$ are the control points, $\{W_i\}$ are the corresponding weights of $\{P_i\}$, $(n+1)$ is the number of control points, and p is the degree of the NURBS curve. The recursive formulas for computing $N_{i,\ p}\ (u)$ is [10]

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

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

where $\{u_0,\ldots,u_m\}$ represents the knot vector and u is the parameter.

A NURBS curve is uniquely determined by its control points, the corresponding weights, and the knot vector.

The general parameter update using iteration method is

$$u_{i+1} = u_i + \Delta u_i$$

where u_i is the present parameter, u_{i+1} is the next parameter, and Δu_i is the incremental value. The resulting position on the curve is calculated by substituting u_i into the corresponding mathematical model.

The following equations prescribes how the valued of u_{i+1} can be calculated if u_i is given. By using Taylor's expansion, the first- and second-order approximation interpolation algorithms can be obtained by Eqs. 4 and 5, respectively.

$$u_{i+1} = u_i + \frac{V(u_i) \cdot T_s}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \tag{4}$$

$$u_{i+1} = u_i + \frac{V_i \cdot T_S}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} + \left(\frac{A_t}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} - \frac{V_i^2 \left(\frac{dC(u)}{du} \cdot \frac{d^2C(u)}{du^2} \right) \Big|_{u=u_i}}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \right) \cdot \frac{T_S^2}{2} \tag{5}$$

where $T_s = t_{i+1} - t_i$ is the interpolation period, V_i=$V(t_i)$ and $A_i = \frac{dV(t)}{dt}\Big|_{t=t_i}$ denote the values of feed rate and acc/dec at time t_i, respectively.

3 Design of adaptive off-line NURBS curve interpolator

3.1 Interpolator architecture

The architecture of the proposed interpolator is shown in Fig. 1, which mainly includes four function modules: adaptive feed rate adjustment module, acc/dec disposal module, feed rate modification module, and S-type velocity generation.

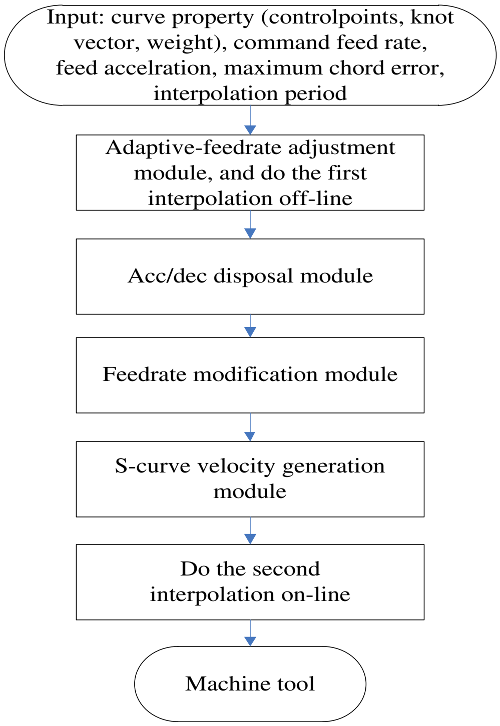


Fig. 1 Interpolator architecture

From Fig. 1, it can be seen that the curves are being interpolated twice. The inputs to the interpolator include the curve property, the command feed rate, the acc/dec, the interpolation period, the maximum allowable chord error. The first stage interpolation is implemented by adaptive feed rate algorithm according to the feed rate command and chord error tolerance, which guarantees the chord error. The first stage interpolation also results in a federate curve, it is the input to the acc/dec disposal module, where the feed rate sensitive corners are detected. To keep the acc/dec within the capacity of the machine tool, the feed rate modification module is applied to discard the undesired feed rate such that the feed rate curve is continuous. The S-type velocity generation module is to generate the velocity profile with continuous acceleration. Using the resulting velocity profile, both the requirement of chord error and the acc/dec capability are guaranteed. By applying the proposed interpolator, the feed rate varies with continuous acc/dec, so the machine tool can work more smoothly with high speed.

3.2 Adaptive feed rate adjustment

The feed rate is a quite important factor which influences the chord error; the higher the feed rate is, the larger the chord error would be. Therefore, to keep the chord error within a tolerance range, the curve speed has to be changed adaptively during the interpolation process. To determine the relation between the chord error and the curve speed, a circular approximation is applied, which is shown in Fig. 2.

According to [2], the curve speed $V(u_i)$ corresponding to the chord error can be derived as:

$$V(u_i) = \frac{2 \cdot \sqrt{\rho_i^2 - (\rho_i - Er)^2}}{T_s} \tag{6}$$

The feedrate can change adaptively along the curve subjecting to the following law [2]:

$$V(u_i) = \begin{cases} F, & if\ \frac{2\cdot\sqrt{\rho_i^2-(\rho_i-Er)^2}}{T_s} > F \\ \frac{2\cdot\sqrt{\rho_i^2-(\rho_i-Er)^2}}{Ts}, & if\ \frac{2\cdot\sqrt{\rho_i^2-(\rho_i-Er)^2}}{T_s} \le F \end{cases} \tag{7}$$

where F is the designed feed rate command.

By applying the adaptive feed rate algorithm, the feed rate curve satisfying the requirement of the chord error can be obtained, then the first interpolation can be realized now.

Since the arc length is an important element in the following algorithms, we save the arc length between the start point of the NURBS curve and the interpolation point results from adaptive feed rate algorithm, so as to simplify the following algorithms. As a result, when given the arc length $S(u)$, the corresponding curve parameter u can be figured out.

3.3 Predetermination of feed rate sensitive corners

After the first interpolation, the next procedure is to take the feed rate obtained from the adaptive feed rate algorithm as the input to the acceleration/deceleration disposal module and the feed rate modification module. In this section, the algorithm for determining the feed rate sensitive corners is introduced. The first step of the algorithm is to detect the feed rate sensitive corner, and identify the acceleration block, deceleration block, and block with uniform feed rate

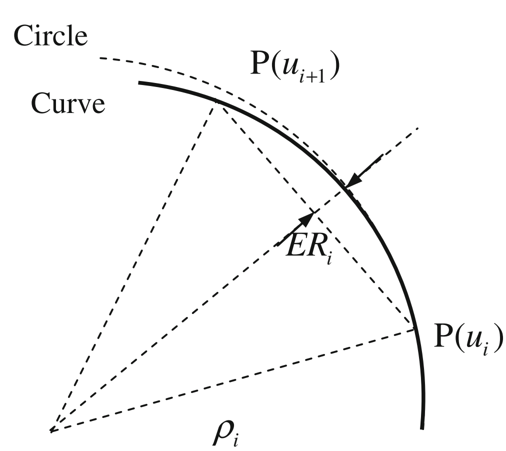


Fig. 2 Circular approximation

from the feed rate curve obtained by the adaptive feed rate algorithm. Then store the parameter value and feed rate of the start point and end point of each acceleration/deceleration/uniform blocks.

After locating acceleration and deceleration blocks, it is necessary to determine the new start and end points of acceleration/deceleration blocks so that the change of feed rate can be kept within the capability of machine tool. The algorithm [3] for determining the new start and end points of acceleration is introduced as follows, and the algorithm for deceleration is similar.

Suppose the original start point of the acceleration block is u_{mm} and its corresponding feed rate is V_{mm}, while the original end point is u_k and its corresponding feedrate is V_k, shown as Fig. 3:

1. Start with the first point of the acceleration block, check if the change of feed rate between consecutive points over the sampling interval Δt for acceleration is larger than the specified acceleration rate A, which falls within the machine's capability.

$$a = \frac{V_{i+1} - V_i}{T} \geq A \tag{8}$$

2. If the test is true, jump to step 4. If not, check if the distance S_p^{i+1} required to increase feed rate from V_{mm} to V_{i+1} with A is less than the curve length S_a^{i+1} between u_{mm} and u_{i+1}.
3. If the test is true, then the start point of acceleration should be postponed, i.e., u_{mm} should be increased to u'_{mm} such that S_a^{i+1}can be decreased to satisfy $S_p^{i+1} = S_a^{i+1}$. And since

$$S_a^{i+1} = S(u_{i+1}) - S(u'_{mm}) \tag{9}$$

Then u'_{mm} can be figured out from its arc length and the corresponding feed rate is kept as V_{mm}.

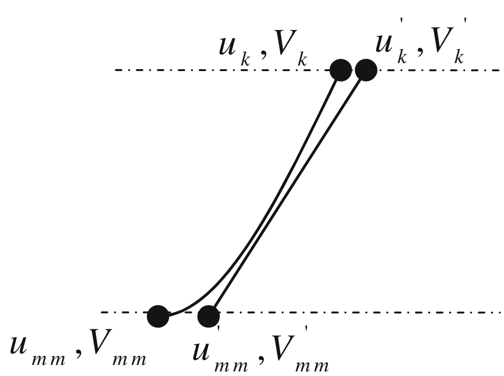


Fig. 3 Acceleration block

4. If u_{i+1} is not the end point of the acceleration block, perform step 2 with the next point. If it is the end point, the new start point of acceleration is obtained with its curve parameter being u'_{mm} and feed rate being V_{mm}.
5. After determining the new start point u'_{mm}, it is time to find out the new end point. Denote S as the distance required to accelerate from V_{mm} to V_k, thus

$$S = \frac{{V_k}^2 - V_{mm}^2}{2A} = S(u'_k) - S(u'_{mm}) \tag{10}$$

Then check if S is less than the curve length between u'_{mm} and u'_{k}. If the test is true, then

$$S(u'_k) = S + S(u'_{mm}) \tag{11}$$

so u'_{k} can be obtained and feed rate of the end point is $V'_{\mathrm{k}} = V_k$.
6. If not, the end point of the acceleration should be updated to $u'_{\mathrm{k}} = 1$. And from

$$S = \frac{{V'_k}^2 - V_{mm}^2}{2A} = S(u_N) - S(u'_{mm}) \tag{12}$$

the feed rate of the end point V'_{k} can be obtained.

Until now, the algorithm for determining the new start and end points of acceleration block has been finished. We store the information that specifies the acceleration block as the format below:

$$\begin{array}{ccc} 1 & & \\ S(u_{mm}) & u_{mm} & V_{mm} \\ S(u_k) & u_k & V_k \end{array}$$

where $S(u_{mm})$ and $S(u_k)$ denote the arc length between the initial point of the NURBS curve and point u_{mm} and u_k, respectively. u_{mm}, V_{mm}, u_k, and V_k, which have been updated by u'_{mm},V'_{mm},u'_{k}, andV'_{k}, denote the new start and end point of the acceleration and the corresponding feed rate.

3.4 Feed rate modification

Applying the algorithm introduced above, acceleration and deceleration blocks with a specified acceleration rate are obtained. However, since all the acceleration and deceleration blocks are considered separately, with regardless of the relations between any two blocks, some undesired results may occur. For instance, as shown in Fig. 4, the start point of certain deceleration block may appear before the end point or even the start point of another acceleration/deceleration block, thus intersection or discontinuous of the feed rate curve, with several different values of feed

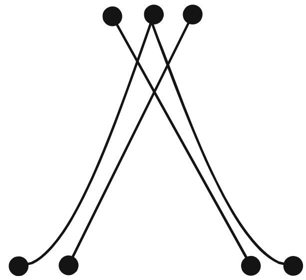


Fig. 4 Undesired results

rate corresponding to the same curve parameter u, would come forth and it is unacceptable. Therefore, in this section, an algorithm is proposed to modify the feed rate so that the feed rate curve would become continuous without jumping, besides there is one and only one feed rate value corresponding to the given curve parameter.

In the proposed algorithm, the acceleration/deceleration block is considered as input one by one, and we compare the current input block with the previous one to figure out the desired result, if the result still appears with intersection or discontinuous of feed rate with the more previous block, then another comparison is needed. Keep doing the comparison until the desired result is figured out.

In real world, there are any kinds of NURBS curve, consequently any kinds of feed rate curve would appear in the previous algorithm. There are two aspects that make the proposed algorithm be possible to apply to various NURBS curves. Firstly, it is the constraint generated by the algorithm determining the new start and end points of acceleration/deceleration blocks. Secondly, the acceleration/deceleration block is read in orderly.

When determining the new start and end points of acceleration/deceleration block, two constraints would be generated for the relations between the original start and end points and the new start and end points, as shown in Fig. 5

In the case of deceleration,

$$\begin{cases} u'_2 \le u_2 \\ u_2 \le u'_m \le u_m \end{cases}$$

is the constraint.

In the case of acceleration,

$$\begin{cases} u'_k \ge u_k \\ u_{mm} \le u'_{mm} \le u_k \end{cases}$$

is the constraint.

With these two constraints, read in the acceleration/deceleration block one by one and do comparison with previous blocks until the desired result is obtained. For simplification, denote the previous block, i.e., the previous accelerating/decelerating curve segments as SEG1, which has been stored but can be updated if undesired result appears; denote the current block, which is read in now, as SEG2. Suppose the start point of SEG1 is u_{11} and its feed rate is V_{11}, the end point of SEG1 is u_{12} and its feed rate is V_{12}; while the start point of SEG2 is u_{21} and its feed rate is V_{21}, the end point of SEG2 is u_{22} and its feed rate is V_{22}. All situations that may appear can be described as the following four cases: (1) SEG2 is a deceleration block and SEG1 is an acceleration one, (2) SEG2 is a deceleration block and so is to SEG1, (3) SEG2 is an acceleration block and SEG1 is a deceleration one, (4) SEG2 is an acceleration block and so is to SEG1.

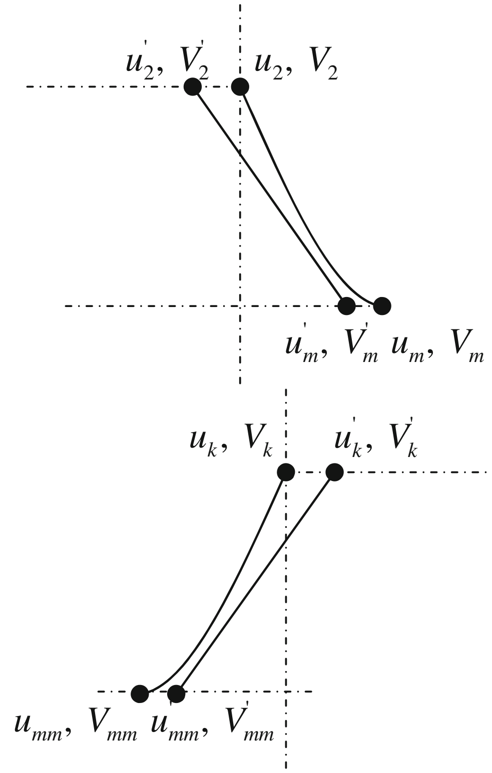


Fig. 5 Constraints of acceleration/deceleration

Case 1 SEG2 is a deceleration block and SEG1 is an acceleration one. As shown in Fig. 6, the different situation can be classed according to the relation between the start point of deceleration and the end point of acceleration.

Case 2 SEG2 is a deceleration block and so is to SEG1, as shown in Fig. 7.

Case 3 SEG2 is an acceleration block and SEG1 is a deceleration one, as shown in Fig. 8

Case 4 SEG2 is an acceleration block and so is to SEG1, as shown in Fig. 9

Next, a typical case is taken as an example to illuminate the principle of the feed rate modification algorithm. The method to deal with other cases is similar. As shown in

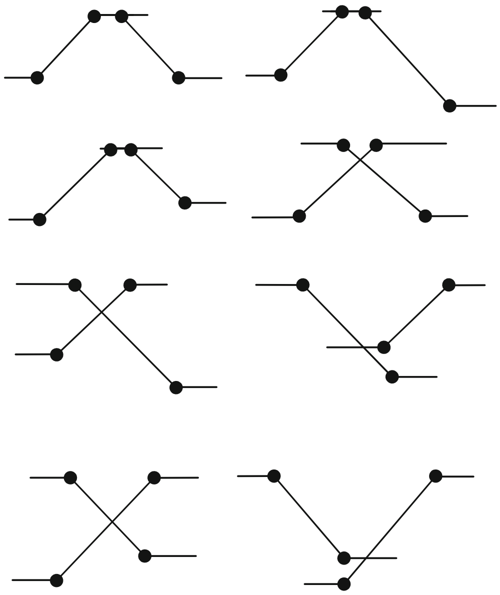


Fig. 6 Case 1

Fig. 10, intersection appears between an acceleration block and a deceleration block after the acceleration/deceleration disposal module. Suppose the start point of the acceleration block is u_{11} and its feed rate is V_{11}, the end point is u_{12} and its feed rate is V_{12}; while the start point of the deceleration block is u_{21} and its feed rate is V_{21}, the end point is u_{22} and its feed rate is V_{22}.

Assuming the intersection point is u_c and its corresponding feed rate is V_c, by constructing the relation between the distance required to accelerate from the initial feed rate of the acceleration block V_{11} to V_c with the specified acceleration rate and the arc length between u_{11} and u_c, then Eq. 13 can be obtained. Similarly, by constructing the relation between the distance required to decelerate from the initial feed rate of the deceleration block V_{21} to V_c and the arc length between u_{21} and u_c, then Eq. 14 can be obtained.

$$S(u_c) - S(u_{11}) = \frac{{V_c}^2 - {V_{11}}^2}{2A} \tag{13}$$

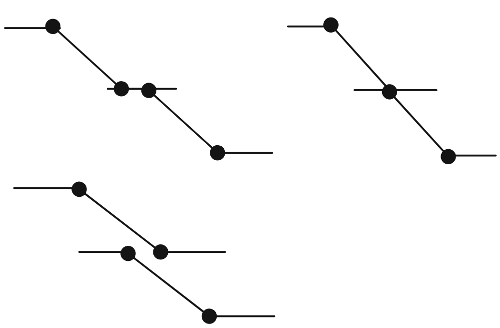
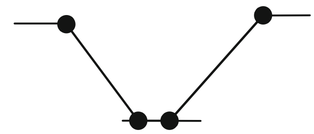


Fig. 7 Case 2

Fig. 8 Case 3



$$S(u_c) - S(u_{21}) = \frac{{V_{21}}^2 - {V_c}^2}{2A} \tag{14}$$

Then the feed rate V_c and the arc length of u_c can be obtained by solving these two equations.

$$V_c = \sqrt{\frac{{V_{11}}^2 + V_{21}^2}{2} + A(S(u_{21}) - S(u_{11}))} \tag{15}$$

$$S(u_c) = \frac{V_{21}^2 - V_{11}^2}{4A} + \frac{1}{2}(S(u_{11}) + S(u_{21})) \tag{16}$$

By Eq. 16, we get u_c from its arc length. And then the original acceleration block is updated to a block accelerating from u_{11}, V_{11} to u_c, V_c, and the original deceleration block can be saved as a new deceleration block from u_c, V_c to u_{22}, V_{22}. The final modification result is shown as Fig. 11.

After applying the algorithm, all cases shown above can be modified, with the results that there is no any intersection or discontinuous. The modification results can be shown as Fig. 12.

Finally, all the information of the feedrate curve and be described as the format below:

Flag of block (0, 1, −1)
Curve parameter of the block start point
Feed rate of the block start point
Curve parameter of the block end point
Feed rate of the block end point

where, 0 denotes the uniform feed rate block, 1 denotes the acceleration block, and −1 denote the deceleration block.

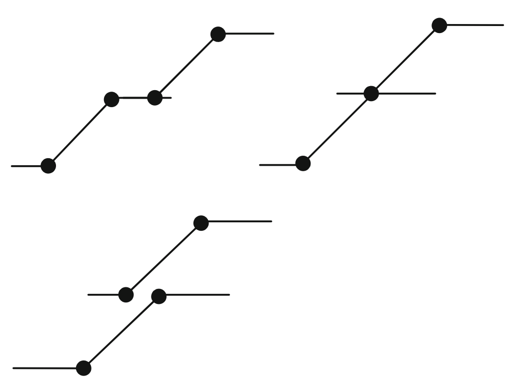


Fig. 9 Case 4

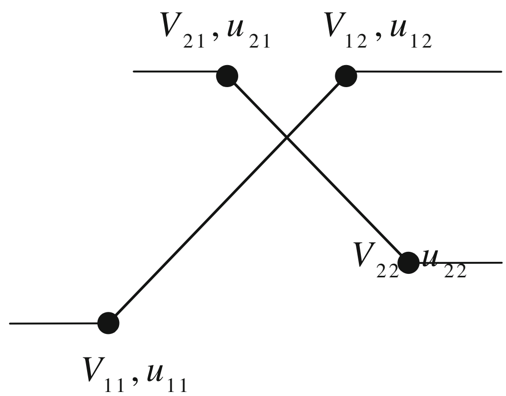


Fig. 10 Case of intersection

3.5 S-type velocity generation

From the information obtained by the proposed algorithm, the feed rate curve has been segmented into three kinds of block, the acceleration block, the deceleration block and the block with uniform feed rate. Then the S-type velocity profile can be generated now.

Take the acceleration block as an example, shown as Fig. 13. With the start velocity v_{st}, the end velocity v_{end}, and the acceleration of T (trapezoidal) type profile a_T, in order to ensure the same efficiency with the T-type profile, let us set

$$a_m = ka_T \tag{17}$$

where a_m is the max acceleration of S-type velocity profile, and $k \geq 1$.

According to the acceleration plot shown in Fig. 14, the equations can be obtained as following.

$$a_t = j_m t_1 \tag{18}$$

$$a_m = j_m(t_1 + t_2) \tag{19}$$

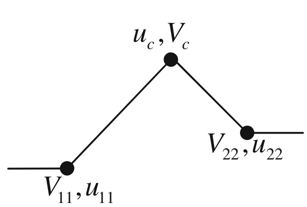


Fig. 11 Modification result

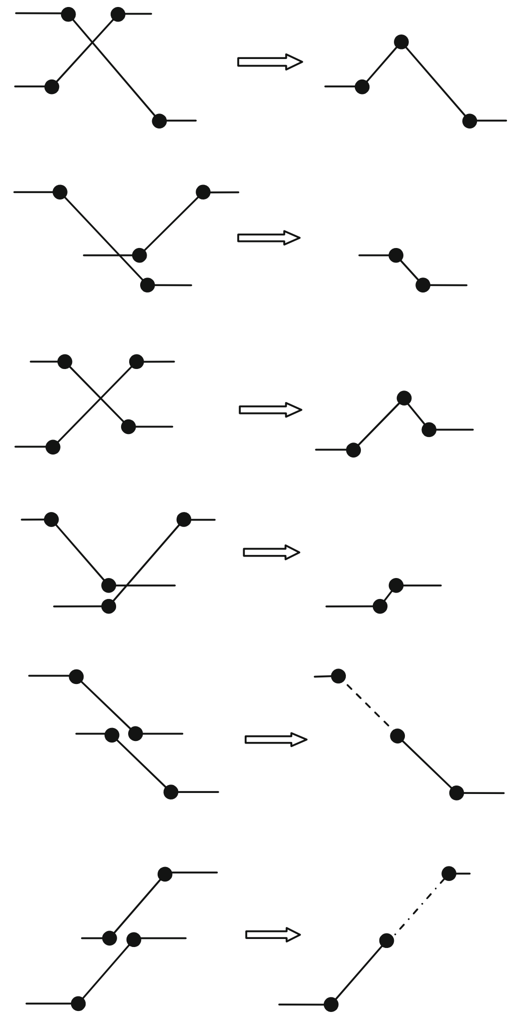


Fig. 12 Modification results

$$\frac{1}{2}a_t t_1 = \frac{1}{2}(a_m - a_t)(t2 + 2t3) \tag{20}$$

$$t_1 + t_2 + t_3 = T \tag{21}$$

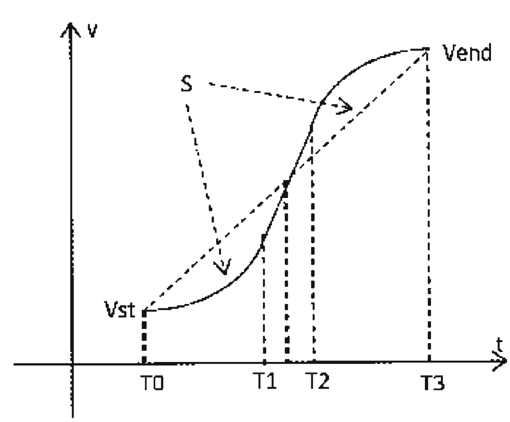


Fig. 13 Velocity–time plot

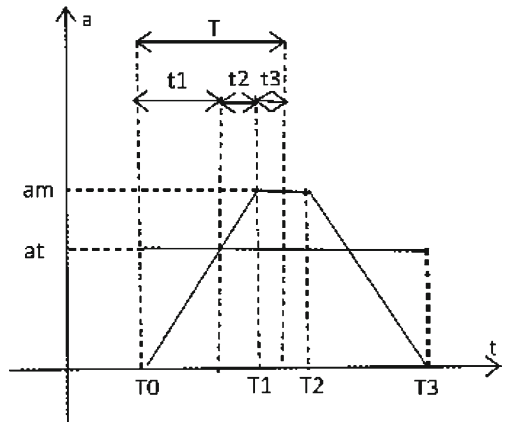


Fig. 14 Acceleration–time plot

where

$$T = \frac{v_{end} - v_{st}}{2a_T}$$

From Eqs. 17, 18, and 19, Eq. 22 can be obtained.

$$t_2 = (k-1)t_1 \tag{22}$$

And from Eqs. 20 and 22, Eq. 23 can be obtained.

$$t_3 = \frac{2k - k^2}{2(k-1)} t_1 \tag{23}$$

Then from Eqs. 21, 22, and 23, t_1 and the jerk j_m can be obtained as follows.

$$t_1 = \frac{2(k-1)}{k^2} T \tag{24}$$

$$j_m = \frac{k^2 a_T}{2(k-1)T} \tag{25}$$

Until now, the velocity of S type of the acceleration block can be get as follow with the max acceleration a_m and the jerk j_m shown as Eqs. 17 and 24.

$$V_{up}(t) = \begin{cases} V_{st} + \frac{1}{2} j_m (t - T_0)^2 & T_0 \le t \le T_1 \\ V_1 + a_m (t - T_1) & T_1 \le t \le T_2 \\ -\frac{1}{2} j_m (t - T_3)^2 + V_{end} & T_2 \le t \le T_3 \end{cases} \tag{26}$$

where

$$\begin{cases} T_1 = T_0 + t_1 + t_2 \\ T_2 = T_1 + 2t_3 \\ T_3 = T_0 + 2T \end{cases}$$

Another restriction of a_m can be obtained from the following equation and Fig. 13.

$$t_1 + t_2 \le T$$

The restriction $k \le 2$ can be achieved with Eqs. 22 and 24.

Thus, the conclusion is that in order to keep the same efficiency between S-type and T-type velocity profile, the following condition should be satisfied.

$$a_m = k a_T$$

$$j_m = \frac{k^2 a_T}{2(k-1)T}$$

where a_T is the acceleration of T-type profile, and $T = \frac{v_{end} - v_{st}}{2a_T}$, $1 \le k \le 2$.

The processing method of deceleration block is similar to the acceleration block. From the information obtained by the proposed algorithm, the second interpolation can be done according to the final feed rate curve. Since the final feed rate can not only keep the chord error fall within the set tolerance, but also limit the acc/dec rate to the specified value within the capability of the machine tool, as a result, the machine tool can work more fluent and finally get better machining quality.

4 Simulation results and discussion

Given a NURBS curve, shown in Fig. 15, the simulation results are shown in Fig. 16a–i. The control points, weight vector, and knot vector of the NURBS curve are given as

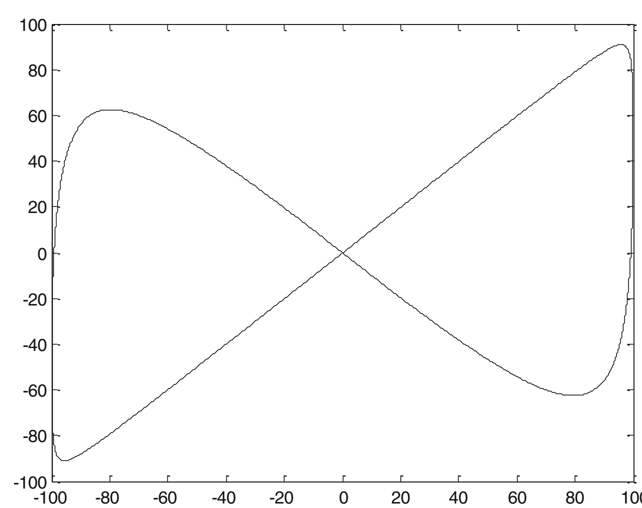


Fig. 15 NURBS curve

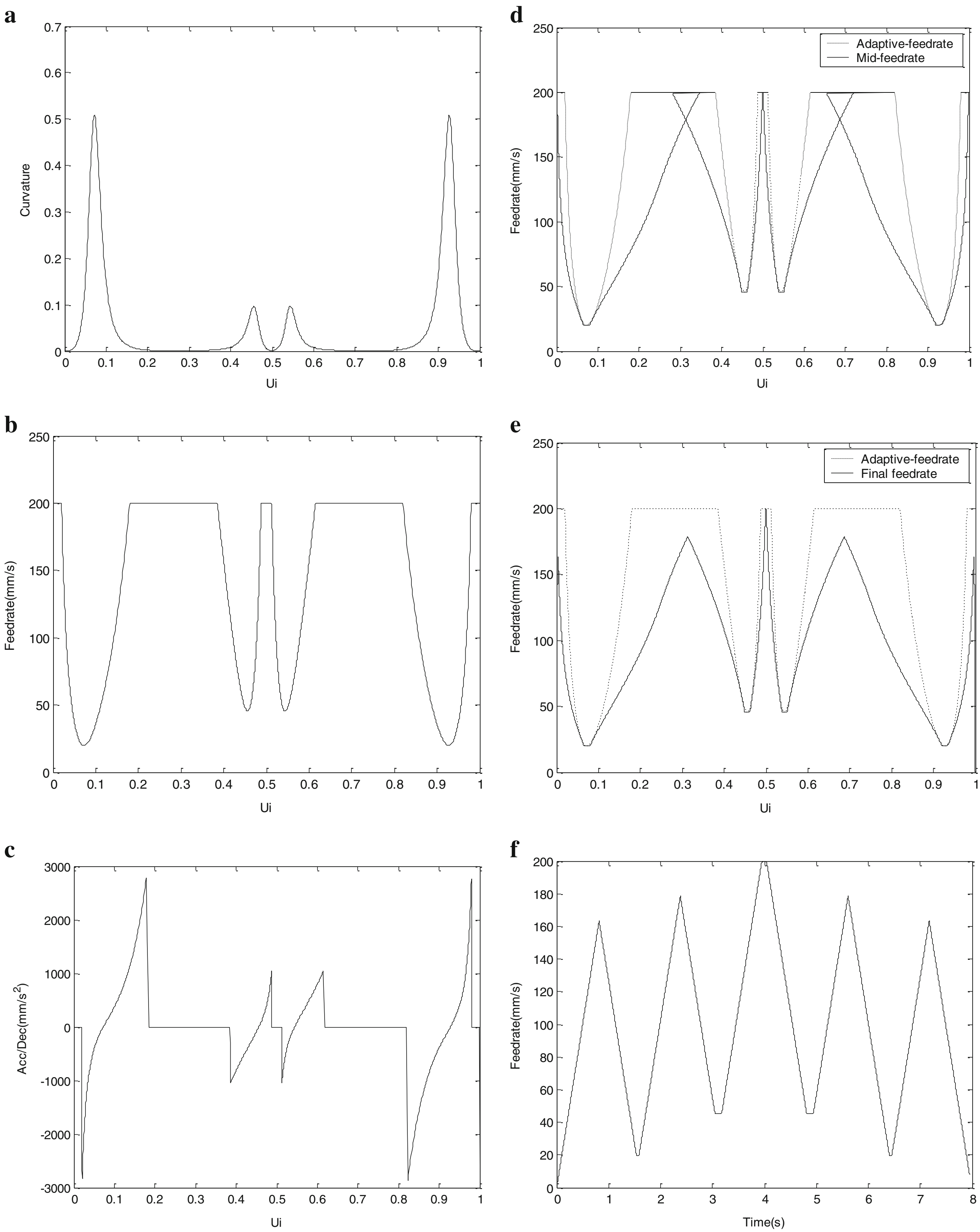


Fig. 16 **a** Curvature plot. **b** Feed rate plot after adaptive feed rate algorithm. **c** Acceleration plot after adaptive feed rate algorithm. **d** Comparison of adaptive feed rate and middle feed rate. **e** Comparison of adaptive feed rate and modified feed rate. **f** Feed rate of T-type profile. **g** Acceleration of T-type profile. **h** Feed rate of S-type profile. **i** Acceleration of S-type profile

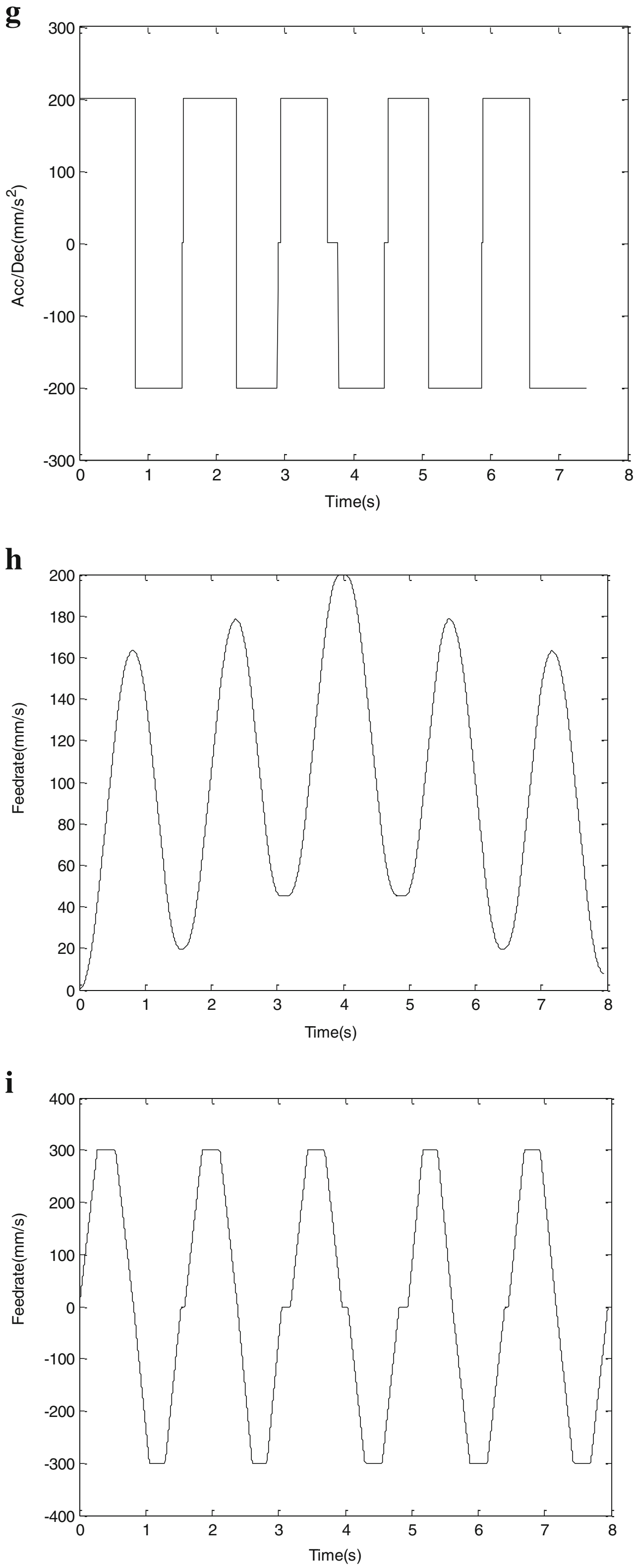


Fig. 16 (continued)

The control points: {(0,0,0), (−100,−100,0), (−100,100,0), (25,0,0), (100,−100,0), (100,100,0), (0,0,0)}
The weight: W={1 30 10 1 10 30 1}
The knot vector: U={0 0 0 0.25 0.5 0.5 0.75 1 1 1}
The interpolation period: Ts=2 ms
The maximum feed rate command: F=200 mm/s
The acceleration of T-type profile: A=200 mm/s^2
The max acceleration of S-type profile: A_{max}=300 mm/s^2

Figure 16a shows the curvature of the NURBS curve, and Fig. 16b shows the feed rate curve computed by adaptive feed rate algorithm. It can be seen that the feed rate decreases as the curvature increases. But from Fig. 16c, we can see that the acc/dec would go beyond the capability of the machine tool

Figure 16d shows the comparison of the feed rate curve after acceleration and deceleration disposal module, the intersection areas appear and it is unacceptable. By applying the feed rate modification algorithm, the comparison between modified feed rate and feed rate obtained by adaptive feed rate algorithm with respect to u is shown in Fig. 16e. It is clear that the intersection areas are eliminated. The modified feed rate is less than the adaptive feed rate, which means that the chord error can be kept within the tolerance. The feed rate of T-type profile with constant acceleration and the acceleration profile are shown in Fig. 16f and g, respectively. The final S-type feed rate and the acceleration with respect to time are shown in Fig. 16h and i, respectively. We can see clearly that the acc/dec change continuously and is kept to be below the specified value in the interpolation process, thus the acc/dec capability of machine tool can be guaranteed.

5 Conclusion

NURBS interpolator is a key component in future CNC system. By applying the NURBS interpolator, disadvantages of the conventional machining approach, such as large data size, feed rate and acceleration discontinuity, and shock of the machine tool, can be overcome. It offers significant benefits to CNC users with high speed, high accuracy machining.

In the paper, an adaptive off-line interpolator was proposed for NURBS curves, which can be applied to any kinds of NURBS curves. Essentially, chord error can be limited within tolerance by carefully designing the velocity profile. Moreover, continuous acc/dec profile ensure the generated interpolation command be limited within the capability of machine tool. The smooth interpolation command guarantees high-quality machining process with high speed. The proposed method can be implemented on the host of CNC system, which is our further work.

References

1. Tsai MC, Cheng CW (2003) A real-time predictor-corrector interpolator for CNC machining. J Manuf Sci Eng 125 (3):449–460
2. Yeh S, Hsu P (2002) Adaptive-feedrate interpolation for parametric curves with a confined chord error. Comput Aided Des 34:229–237
3. Narayanaswami YR (2003) A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining. Comput Aided Des 35:1249–1259
4. Zhang LY, Wang KS, Bian YC, Chen H (2008) A real-time NURBS Interpolator with feed rate adjustment. Advanced Intelligent Computing Theories and Applications with Aspects of Theoretical and Methodological Issues 1064–1071
5. Zhang LY, Bian YC, Chen H, Wang KS (2009) Implementation of a CNC NURBS curve interpolator based on control of speed and precision. Int J Prod Res 47(6):1505–1519
6. Piegl LA, Rajab K, Smarodzinava V, Valavanis KP (2009) Using a biarc filter to compute curvature extremes of NURBS curves. Eng Comput 25(4):379–387
7. Feng JC, Li YH, Wang YH, Chen M (2010) Design of a real-time adaptive NURBS interpolator with axis acceleration limit. Int J Adv Manuf Technol 48(1):227–241
8. Wang JB, Yau HT (2009) Real-time NURBS interpolator: application to short linear segments. Int J Adv Manuf Technol 41:1169–1185
9. Shen HY, Fu JZ, Fan YQ (2011) A new adaptive interpolation scheme of NURBS based on axis dynamics. Int J Adv Manuf Technol 56:215–221
10. Piegl L, Tiller W (1997) The NURBS Book, 2nd edn. Springer, Berlin