Computer-Aided Design 55 (2014) 37–48

Contents lists available at ScienceDirect

Computer-Aided Design

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

A fine-interpolation-based parametric interpolation method with a novel real-time look-ahead algorithm

Yu-an Jin, Yong He*, Jian-zhong Fu, Zhi-wei Lin, Wen-feng Gan

The State Key Lab of Fluid Power Transmission and Control Systems, Department of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China

HIGHLIGHTS

- Optimization in both of the two interpolation stages.
- Full consideration of various kinematical constraints.
- Methodology of parameters adjustments in fine interpolation.
- Application of a novel look-ahead algorithm in rough interpolation.
- Application in open architecture CNC.

ARTICLE INFO

Article history:
Received 29 November 2013
Accepted 9 May 2014

Keywords:
Parametric interpolation
Fine interpolation
Look-ahead
Open CNC

ABSTRACT

Parametric interpolation is presently supported by majority of CNC systems because of its various advantages over traditional linear/circular interpolation. Two stages (i.e. rough interpolation and fine interpolation) involved in parametric interpolation are complementary to each other in terms of affecting machining quality significantly. So far much work has been conducted to improve the machining process with various rough interpolation adjustments, while with little research on fine interpolation. To further alleviate the feedrate jump between two adjacent rough interpolation periods, a fine interpolating strategy implemented within one rough interpolation period can be utilized to make the feedrate alteration comparatively smooth. Meanwhile, an arc is adopted to substitute the linear path to reduce the chord errors caused by rough interpolation. Besides, as one of the major difficulties of parametric interpolation is the feedrate determination concerning a wide variety of technical parameters, a real-time look-ahead feedrate generation method which can determine the decelerating position rapidly and accurately is proposed in this paper. The look-ahead approach can generate the feedrate profile to satisfy the geometrical constraints and kinematical characteristics determined by machine tools. Finally, the proposed parametric interpolation method is performed in an open architecture CNC platform to machine parametric curves. The results are satisfactory and are able to verify the robustness and effectiveness of the proposed algorithm.

1. Introduction

1.1. Parametric interpolation

CNC machining has become a significant part in the manufacturing industry for ever growing demands for high-precision and high-efficiency machining. Since the goal for machining is to obtain more satisfying machined parts within a shorter period, quality and efficiency are basically two pursuits for research and efforts in this field. As for the interpolation stage of machining, high-accuracy requires errors induced by the approximation with a large amount of segments are limited to a tolerant range, while high-speed needs a smooth feedrate profile that keeps machining fast and accurate. Parametric interpolation owns many merits over the traditional linear and circular interpolation in terms of the model representation, feedrate smoothness and application range [1–4].

(a) The geometrical information of the machining contours can be totally as well as accurately transferred to the CNC systems without any approximation errors and data loss that may occur in the linear and circular interpolation.

(b) Parametric interpolator only needs some critical parameters of the machining contours (i.e. control points, knot vectors,

* Corresponding author.
E-mail address: yongqin@zju.edu.cn (Y. He).

http://dx.doi.org/10.1016/j.cad.2014.05.002
0010-4485/© 2014 Elsevier Ltd. All rights reserved.

weights), such a transmission mechanism can guarantee the efficiency of interaction between the host and the slave.

(c) Feedrate continuity is achieved effectively as the junctions between tiny segments in traditional interpolation methods which require repeated acceleration–deceleration processes are avoided.

(d) Parametric interpolation can still be used in the conventional CNC systems after some developments of the machining segments, such as approximating tiny parts into curves or transitional optimization with parametric curves between them.

In order to realize parametric interpolation, researchers developed a wide variety of methods to achieve better machining qualities. The initial approaches for interpolating parametric curves were based on Taylor's expansion [5–8]. The first-order and the second-order approximations of Taylor's expansion were the representative methods to determine the feedrate based on a certain value of chord error. To alleviate the feedrate fluctuation during machining, other algorithms had been introduced, specifically, optimizing feedrate profiles considering acceleration/deceleration (ACC/DEC) processes was feasible in this respect. Du et al. [9] presented an adaptive NURBS interpolator with the consideration of ACC/DEC control. A real-time flexible ACC/DEC control scheme was introduced to solve the sudden feedrate change around the corners with large curvature in their method. Lin and Tsai [8] proposed a real-time look-ahead NURBS interpolator using a servo dynamic feedrate modification technique to generate a jerk-limited feedrate profile. Heng and Erkorkmaz [10] presented a robust and numerically efficient NURBS interpolation strategy which applied an adaptive manner to avoid unwanted feed fluctuations and round-off errors, and a feedrate modulation strategy based on the trapezoidal ACC/DEC profile was developed to guarantee the final trajectory was jerk limited in all axis and the kinematic continuity.

Besides, Shen et al. [11] offered a new interpolation scheme for 2D NURBS curve which contained two steps: pre-processing and real-time interpolation. Besides the geometrical characteristics of the machining contours, the method also took the dynamic parameters limitations into consideration to achieve satisfied machining results. Lee et al. [12] proposed an off-line feedrate planning method of CNC machines constrained by chord error, acceleration and jerk limitations. The method adopted a pre-processor to process the machining NURBS in advance to release computation burden as well as to improve the feedrate profile. The feedrate profile out of their approach was suitable for CNC high-speed machining. Zhou et al. [13] presented a novel multiconstraints feedrate scheduling method for the parametric interpolation in five-axis machining. A feed optimization model was initially built with constrains of geometric error, the maximum feedrate and acceleration in their method. Then a linear programming algorithm was applied to achieve the optimal feedrate profile on the sampling positions. Beudaert et al. [14] put forward an algorithm to obtain an optimized feedrate profile to make the best use of kinematical characteristics of the machine. They pointed out the fundamental job of the feedrate interpolation which was limited by various kinematical parameters. Meanwhile, they proposed an iterative algorithm to compute the minimum time feedrate profile. Besides the full consideration of both tangential jerk and axis jerk, their algorithm could be applied to any articulated mechanical structures, which was quite meaningful. Sun et al. [15] proposed a novel adaptive feedrate interpolation method with drive constraints. They mainly adopted an iterative adjustment to generate a smooth feed profile. While Zhao et al. [16] introduced a feedback interpolator to eliminate the feedrate fluctuation. Besides all these researches, researchers mainly focused on the solution of feedrate determination or feedrate optimization with different kinds of algorithms, such as adaptive speed control algorithms and look-ahead algorithms by taking a wide range of kinematical and dynamical factors.

1.2. Look-ahead process

Look-ahead is a preprocessing on the contours before the real machining. Detecting the feedrate sensitive areas of the curve and determining the decelerating points to achieve a smooth feedrate profile are the main goals in the look-ahead process. Nam and Yang [17] developed a recursive trajectory generation method to estimate and determine the deceleration stage according to the distance left to travel; a look-ahead scheme with a jerk-limited acceleration was proposed for smoothing feedrate profile. Tsai et al. [18] applied a hybrid digital convolution technique to develop a look-ahead scheme that smoothed the feedrate between the joint of two curves. However, the algorithm was applied to multi-block NURBS curves, not to single NURBS curves. Tsai et al. [19] proposed an integrated look-ahead dynamics-based algorithm with the consideration of geometric and servo errors simultaneously. Emami and Arezoo [20] introduced a look-ahead trajectory generation to determine the acceleration stage according to the fast estimated arc length and the reverse interpolation of each curve for NURBS curves. Zhao et al. [2] presented a real-time look-ahead scheme which comprised of path-smoothing, bidirectional scanning and feedrate scheduling to acquire a feedrate profile with smooth acceleration. Based on the works referred above, look-ahead process is able to achieve the adaptive control of the feedrate according to the geometrical characteristics of the trajectory to be machined and assures that the contour errors of the interpolation trajectory are within the range of the appointed error [21].

There are two universal approaches for look-ahead strategy. One is off-line prediction, the other is on-line operation. Off-line strategy provides sufficient detection of the machining contours, and the feedrate profiles are obtained in a non-real-time process before machining procedure. Some complicated and comprehensive algorithms can be realized because the time-consuming computation can be accomplished in advance. On-line method is also called real-time look-ahead process, which requires predicting the following machining contours and determines the current feedrate within a limited time. Among majority of the research on look-ahead methods, the length of the tool path is known or can be calculated easily; however that is actually impractical for parametric curves because of an inaccurate mapping between the parametric u and the displacement S [6]. Besides, off-line look-ahead cannot be applied the to real-time systems effectively, so the expansibility of interpolation algorithms is limited. Hence, in this paper, a real-time look-ahead strategy for parametric interpolation which can enhance the effectiveness and efficiency of machining is presented.

1.3. Two stages of machining

To accomplish the machining of parametric curves, two stages are involved in the CNC interpolation process: rough interpolation and fine interpolation [22]. Rough interpolation generates tiny line segments for each rough interpolation period and sends them to the fine interpolator. The fine interpolator finishes the machining of these tiny parts with several sections based on the servo cycle. So, the rough interpolation period should be integral multiples of the fine interpolation period. A general architecture of the current parametric interpolation methods is shown in Fig. 1. Currently, most research which concentrates on the improvement of machining performance with various algorithms is implemented in the former stage, while quite a few works are involved with the fine interpolation stage that affects the machining results significantly. Therefore, some strategies are proposed in this paper to boost the machining accuracy and efficiency in the fine interpolation stage.

1.4. Research objective

This paper proposes a parametric interpolation method integrated to a novel real-time look-ahead algorithm as well as

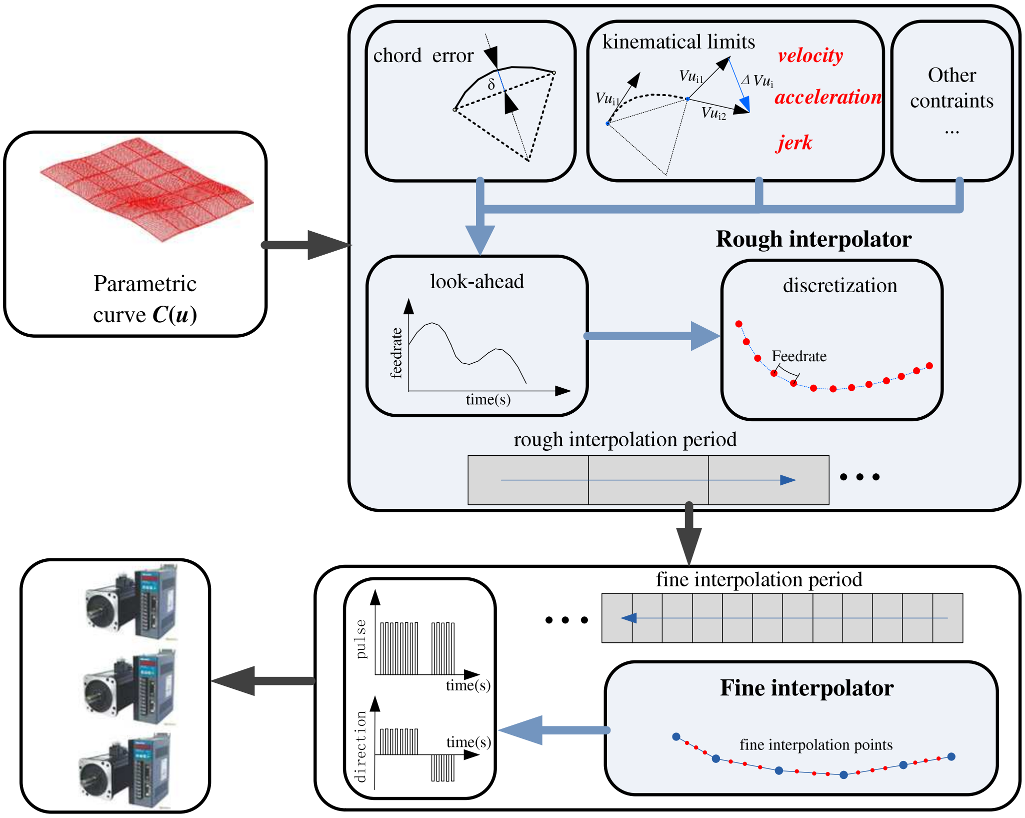


Fig. 1. Architecture of parametric interpolation.

presents some improvements in the fine interpolation stage to further enhance the machining accuracy. As the fine interpolation is directly associated with the actual machining process, any adjustment of the tool paths in this stage may impose a significant influence on the final machining. Algorithms applied in fine interpolator may prolong the period of a fine interpolation cycle, so the rough interpolator has to implement some adjustments accordingly. The proposed look-ahead algorithm in this paper can realize this objective of a high-speed and smooth machining. In order to verify the performance of the algorithm, an open architecture CNC system is developed, as shown in Fig. 2. With the connection bridge of Ethernet, the PC host with high computation ability and the slave motion controller realize the rough interpolation and fine interpolation respectively. During machining, tiny segments after rough interpolation acquired by the PC host are sent down to the slave motion controller for transforming into control commands with some algorithms in each fine interpolation period. Meanwhile, the machining contours are machined accordingly by the movements of the tools controlled by the motors.

Section 2 gives some related knowledge on parametric interpolation. The look-ahead algorithm implemented in the rough interpolator for parametric interpolation is presented in Section 3, while some strategies in the fine interpolator are proposed in Section 4. And then, experiments and simulations are carried out in the experimental platform to demonstrate the efficiency and feasibility of the proposed method in Section 5. Finally, a conclusion is drawn in Section 6.

2. Background and related work

According to the whole procedure of parametric interpolation in Fig. 1, the rough interpolation mainly obtains some necessary motion parameters (e.g. position, velocity, acceleration, and jerk) of each rough interpolation cycle. In order to achieve a high-quality machining, some operations (e.g. look-ahead algorithm and tool paths preprocessing) are integrated into this process. Generally a look-ahead algorithm mainly considering chord errors and some kinematical limitations is employed to determine several critical motion parameters of machining contours. The input contour paths are then segmented into line segments with allowable tolerance; the line segment lengths depend on the current feedrate and the interpolation period [23]. After rough discretization of the machining contours, the corresponding data is sent to the fine interpolator for further procedures. Until now, the details of fine interpolation have not attracted enough attention because it is completely accomplished by the motion controller itself which possesses built-in fine interpolating algorithms.

2.1. Chord error and kinematical limits

The discretization of the machining contours inevitably contributes to the contour errors between two interpolated points [24]. As shown in Fig. 3, there are two common methods to define the contour errors brought by rough interpolation. A direct approach is shown by Eq. (1) with the assumption that the maximal contour error is δ_i' located in the middle point of two adjacent interpolated points. The other method using an approximation to calculate the contour error δ_i is given by Eq. (2). The second method is a simple but more effective one to evaluate the accuracy of interpolation.

$$\delta_i' = \left| C\left(\frac{u_i + u_{i+1}}{2}\right) - \frac{C(u_i) + C(u_{i+1})}{2} \right| \tag{1}$$

$$\delta_i = \rho_i - \sqrt{\rho_i^2 - \left(\frac{L_i}{2}\right)^2}. \tag{2}$$

Besides the contour error, some other factors should be taken into consideration for determining the feedrate of each interpolation period, such as maximum velocity allowed by machine tools,

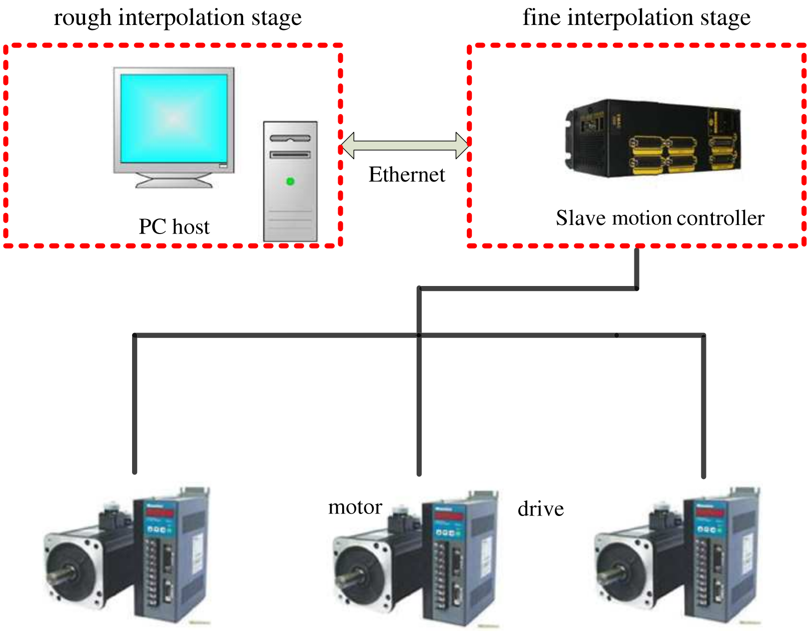


Fig. 2. Open architecture CNC system.

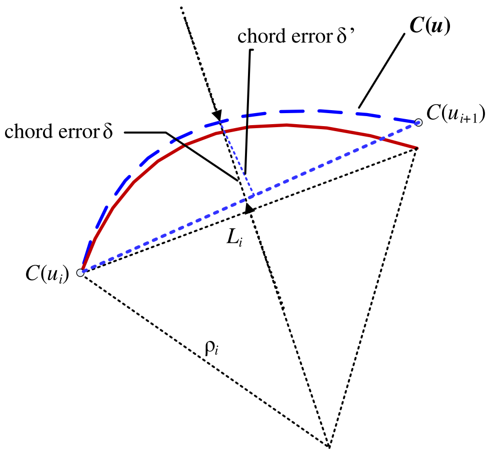


Fig. 3. Chord error between two interpolated points.

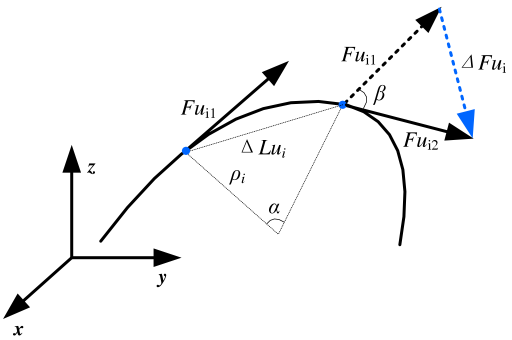


Fig. 4. Speed change within one interpolation period.

maximum acceleration and the lengths of each segment [25]. These factors are also termed as kinematical limitations originated from the hardware conditions of machine tools. Assume the maximum values of velocity, acceleration, jerk of axis are $v_{\max}, a_{\max}, j_{\max}$, respectively. So the corresponding parameters of one interpolation period should satisfy Eqs. (3)–(5).

$$\max(v_i^s, v_i^e) \le v_{\max} \tag{3}$$

$$\max(a_i^s, a_i^e) \le a_{\max} = \frac{\Delta v_{\max}}{T_r} \tag{4}$$

$$\max(j_i^s, j_i^e) \le j_{\max} = \frac{\Delta a_{\max}}{T_r} \tag{5}$$

where Λ_i^s ($\Lambda = v, a, j$) and Λ_i^e ($\Lambda = v, a, j$) denote the parameters at the starting point and the end point of the current machining segment, respectively. Actually, these kinematical requirements are not only just met by each axis, but also need to be abided by the normal kinematical parameters (i.e. total values). For example, the maximum normal acceleration in Fig. 4 should satisfy the condition $\Delta F_{ui}/T_r \le a'_{\max}$, $a'_{\max}$ denotes the maximal total acceleration.

2.2. Discretization process during rough interpolation

Considering the mentioned constraints, the optimal feedrate of each interpolation period is sought to achieve a smooth feedrate profile for a high-speed and high-accuracy machining by using a look-ahead strategy. The universal look-ahead algorithms are summarized in Section 1. Because of the difficulty and complexity to obtain an analytic expression for feedrate determination in parametric interpolation, an iterative calculation is available to get the optimal feedrate with an adaptive algorithm.

Generally, a look-ahead process includes two modules as shown in Fig. 5. The first is the feedrate constraints acquisition module, which determines the maximum feedrate at any point of the machining contour. The feedrate limitations are determined by various constrains including the chord error, tangential feedrate/acceleration/jerk of each axis, total feedrate/acceleration/jerk, command feedrate, axis loads, etc. The second module plans the feedrate profile based on the mentioned feedrate limitations and the ACC/DEC strategy. With different ACC/DEC approaches, the feedrate profile varies differently [26].

After the look-ahead process, the feedrate of each interpolation period can be determined subsequently according to the feedrate profile. Meanwhile, the machining contours are segmented into a large amount of lines which are for the fine interpolation stage. The

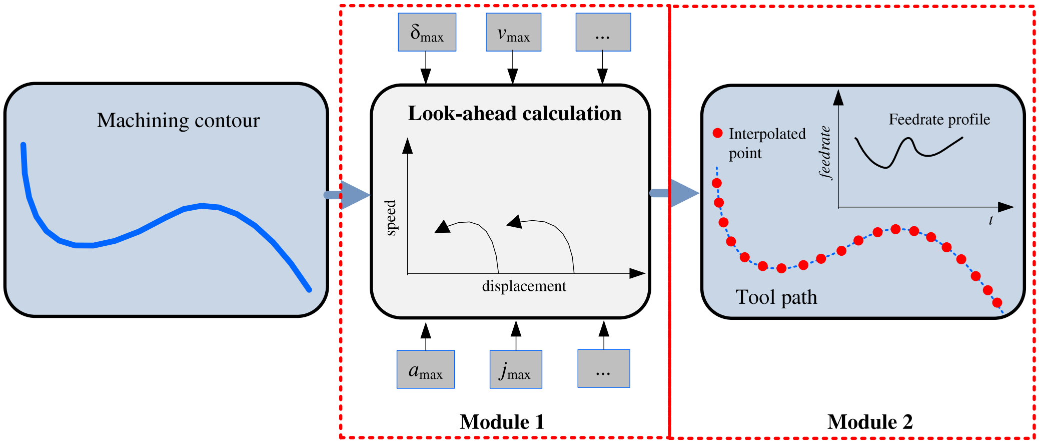


Fig. 5. Scheme of look-ahead process.

tool paths after rough interpolation are a polygonal line containing a point set. This point set is sent to the fine interpolator for final real-time machining. This can be demonstrated in *Module* 2 in Fig. 5.

3. A novel real-time look-ahead algorithm for parametric interpolation

In the interpolators with sampled data method, parameters of tool motion need to be achieved based on the feedrate profile at every sampling period (i.e. fine interpolation cycle); then these parameters are sent to the servo control system for the axis control. Hence, a continuous and smooth feedrate profile should be acquired in the interpolation algorithm before machining procedure. Meanwhile, the feedrate profile is supposed to satisfy a wide range of feedrate constraints. In order to meet these requirements, a look-ahead process is employed in the interpolation algorithm.

According to the description on the look-ahead in Section 1, it can be concluded that the on-line look-ahead takes priority over the off-line one in terms of effectiveness and quick-response. Currently there are two common on-line look-ahead strategies. The first one is realized in the rough interpolation stage as shown in Fig. 1 which has advantage in relieving heavy computing burdens on motion controller. This kind of real-time look-ahead method, however, poses a demanding requirement on data transmission between PC host and motion controller since the feedrate profile acquired in the rough interpolation needs to be sent to the motion controller simultaneously and completely. The second one is finished in the fine interpolation stage which has a weakness inherently with respect to the accuracy of the algorithm. Because of the storage limitation in the motion controller, it would be impractical to send numerous data block groups to the controller, which means the look-ahead algorithm can only be performed in a limited number of segments. This method can definitely alleviate the data transmitting burden as well as avoid several obstacles during the transmission. Actually, with the growing advanced transmission technology, interaction of vast number of data between PC host and motion controller has become no longer a challenging task. The Ethernet communication utilized in this paper is capable of realizing the data exchange between host and slave controller effectively. These two real-time look-ahead methods are illustrated in Fig. 6 and the former one is applied in this paper.

Essentially, the look-ahead algorithm employed in either stage is to obtain a smooth and effective feedrate profile concerning various kinematical limitations of machine tools and given error tolerance. A necessary factor that should be taken into account when planning the feedrate profile is the ACC/DEC strategy mentioned in Section 2.2. Different ACC/DEC methods need different lengths of path for transition between two interpolated points with distinct feedrate. At present, there are various ACC/DEC control strategies, such as linear ACC/DEC strategy, S curve ACC/DEC strategy, and exponential ACC/DEC strategy. A balance between computing complexity and effectiveness should be drawn with respect to the ACC/DEC method. A confined jounce (the derivative of jerk) ACC/DEC strategy is employed in the proposed look-ahead algorithm [27]. As the real-time machining is rigorous in continuous of the data flow, some accurate but time-consuming algorithms are not suitable for real-time look-ahead process. Therefore, the proposed look-ahead method tries to simplify the algorithm without compromising the accuracy and availability.

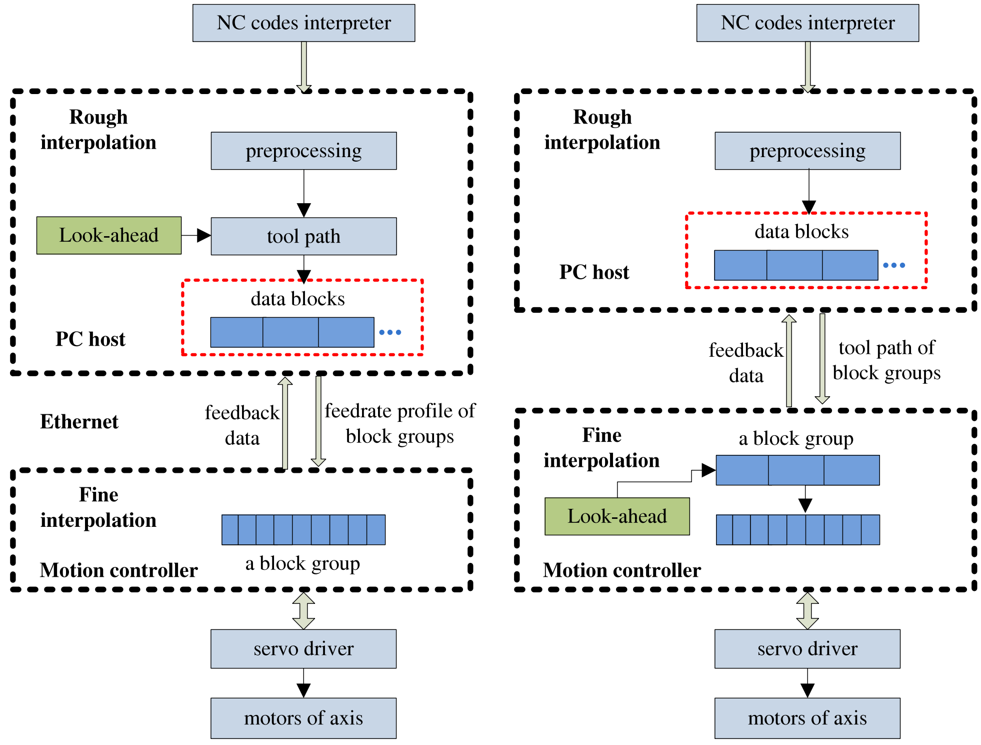
Fig. 7 shows the kinematic profiles during the acceleration stage of the proposed ACC/DEC strategy. The characteristic of jounce-limited can enhance the stationarity of tool movement in comparison with the S curve ACC/DEC strategy by avoiding the slight change of jerk which may lead to vibration in the machining. The period of constant jerk being assumed as zero is to simplify this algorithm. It can be concluded that acceleration and jerk are continuously changing during acceleration stage and velocity changes slowly at the terminal part, which benefits the transition between segments.

The proposed look-ahead method is realized by calculating the length of deceleration form the current feedrate to zero based on the jounce-limited ACC/DEC method and looking forward the same length of the machining curve repeatedly at every sampling period in the rough interpolation stage. This method allows the computing load to decrease to minimum. Assuming the current feedrate is f_t and the length of deceleration S_{de} according to the jounce-limited ACC/DEC strategy is shown in Eq. (6). The proof of Eq. (6) is given in the Appendix.

$$S_{dec} = \frac{97}{24} J \cdot t_0^{\frac{4}{3}} \tag{6}$$

where J is the jounce which is determined by the property of machine tools. The time constant t_0 is determined by four kinematical parameters (i.e. current feedrate, maximal acceleration, maximal jerk and jounce), which are discussed in detail in the Appendix.

The kinematical parameters that are critical in judging whether the generated feedrate profiles meet all the requirements are determined according to the formulas in the Appendix. These parameters of each servo cycle are generated to control the motors which drive the axis to realize precision and stable movements. In the proposed look-ahead scheme, various kinematical constraints from both machine tools and workpieces are taken into consideration and the feedrate profiles are adapted to the required conditions.



(a) Look-ahead in rough interpolation.

(b) Look-ahead in fine interpolation.

Fig. 6. Schemes of two different real-time look-ahead methods.

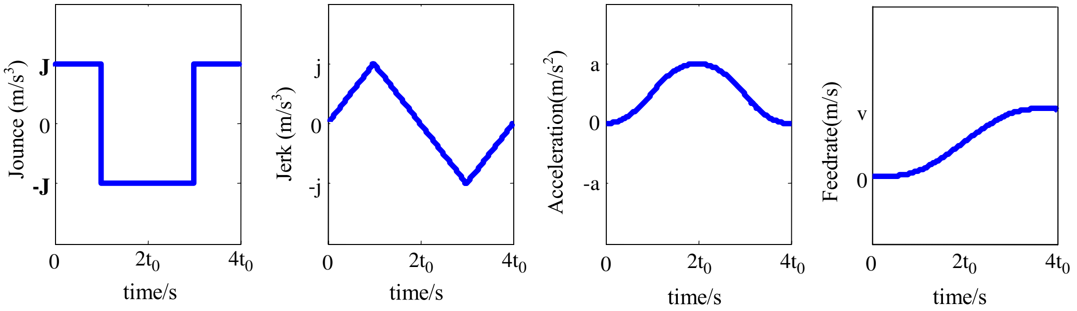


Fig. 7. Kinematic profiles with jounce-limited acceleration.

The look-ahead algorithm is performed on the machining curves in the rough interpolation stage just as shown in Fig. 8. When the machining process begins, the look-ahead module also starts to work. According to the current feedrate f_t and other three kinematical parameters, the length of looking forward is achieved as long as f_t is not equal to zero. If there exists a sensitive area where a lower feedrate is required based on various constraints, an inverse computation is performed to gain the corresponding decelerating point. After the determination of the decelerating point, a decelerating procedure starts to smooth the feedrate profile when machining process arrives at this point. During the decelerating process, the look-ahead algorithm still looks forward the machining path based on the current feedrate. If another sensitive area is detected during deceleration, the original feedrate profile may be modified according to the inverse computation. In the proposed look-ahead algorithm, there are two possible terminating conditions. Besides reaching the end of the machining curve, current feedrate equaling to zero is another condition to terminate the look-ahead process. As long as the current feedrate is not zero, the look-ahead procedure is working to monitor the sensitive areas in the machining contours.

4. Improvements in the fine interpolation stage for parametric interpolation

The controllable shortest time unit in most current parametric interpolation algorithms is a rough interpolation period which contains several fine interpolation cycles. Theoretically, the velocity changes continuously, but in practice, it follows a step change after an interpolation interval as a cycle [28]. The feedrate of each rough interpolation period is decided by the corresponding chord errors and various kinematical limitations as described in Section 2. The feedrate variation during each cycle is given by

$$f_i = f_{i-1} + a_i T_r \tag{7}$$

where f_i is the current period's feedrate, f_{i-1} is the former period's feedrate, a_i is the current acceleration which is decided by the maximal chord error $\delta_{\max}$ and the kinematical limitation of machine

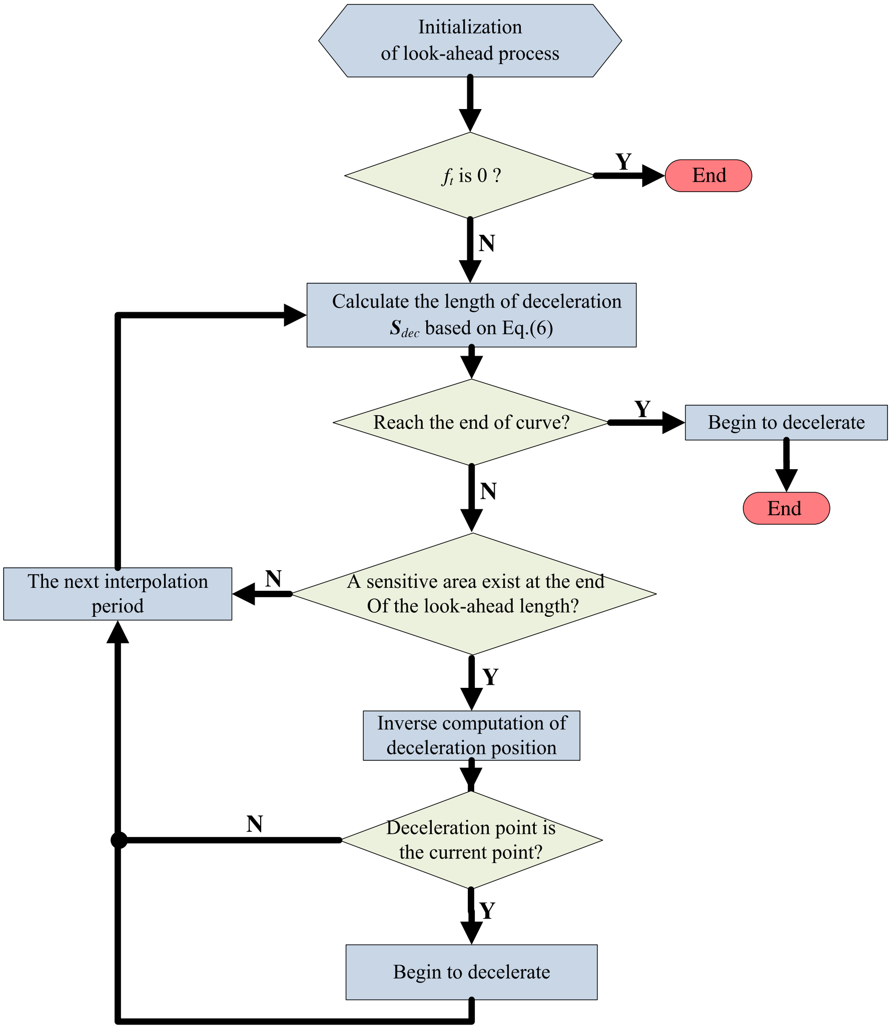

Initialization of look-ahead process
f_t is 0 ?
Y
End
N
Calculate the length of deceleration S_{dec} based on Eq.(6)
Reach the end of curve?
Y
Begin to decelerate
End
N
A sensitive area exist at the end Of the look-ahead length?
N
The next interpolation period
Y
Inverse computation of deceleration position
Deceleration point is the current point?
N
Y
Begin to decelerate

Fig. 8. Flow chart of the real-time look-ahead algorithm.

tools, and T_r is the rough interpolation period which is related to the resolution of CNC system. A constant feedrate is maintained within one period, while a sharp saltus appears at the junction between two adjacent periods especially when the acceleration is large. Just as shown in Fig. 9, the feedrate profile seems smooth as a whole, but obvious alternations at the beginning of each period may cause a mechanical vibration on machine tools.

Furthermore, contour errors caused by approximating with tiny line segments in the fine interpolation stage are inevitable since the fine interpolation process implemented at each servo cycle is accomplished based on the parameters from rough interpolation. As shown in Fig. 10, the hollow points are fine interpolated points which are derived from the rough interpolated points (the solid points shown in figure). As the period of fine interpolation is shorter than that of rough interpolation, the intermediate path between two adjacent rough interpolated points deviates from the desired path definitely due to the existence of curvature.

The main idea of the proposed method is to make some improvements in the fine interpolation stage to alleviate the above problems and subsequently to enhance the machining accuracy. Since the high-performance of motion controller, some simple but effective strategies implemented in slave motion controller can function well on the machined segments during fine interpolation. Normally, there are several fine interpolation cycles between two rough interpolated points and this paper takes 4 as an example for demonstration.

Fine interpolation process is accomplished within a very short duration during which command values are sent to servo system and transferred to the corresponding pulse/voltage. In the majority of motion controller, parameters of each servo cycle are calculated with some fixed algorithms which are beyond user's options. In order to achieve a better machining, some approaches are applied on the rough interpolated points in the fine interpolation process.

If all the fine interpolated points travel through the machining contours, contour errors can be limited to the minimum. However, accurate calculation of each point on machining contours is time-consuming and complicated as well, so it is impossible for real-time fine interpolation. Some fast and comparatively accurate algorithms can be introduced for the determination of each fine interpolated point. Besides, the step change of the velocity at the junction of two adjacent segments shown in Fig. 6 can also be alleviated by the proposed method. If the feedrate changes from f_i^0 to f_i^1 more smoothly inside a rough interpolation period T_r, the average feedrate should be equal to v_i. And by doing this, the velocity of the last period's terminal point is the same as that of the current period's starting point. As shown in Fig. 11, the solid line is the actual feedrate profile which connects the velocity of two adjacent segments.

A rough interpolation period is divided into n fine interpolation periods (n is 4 in this paper, determined by the precision limitation of the controller) to realize the machining. Assume the feedrate of the former rough interpolation period, the current period

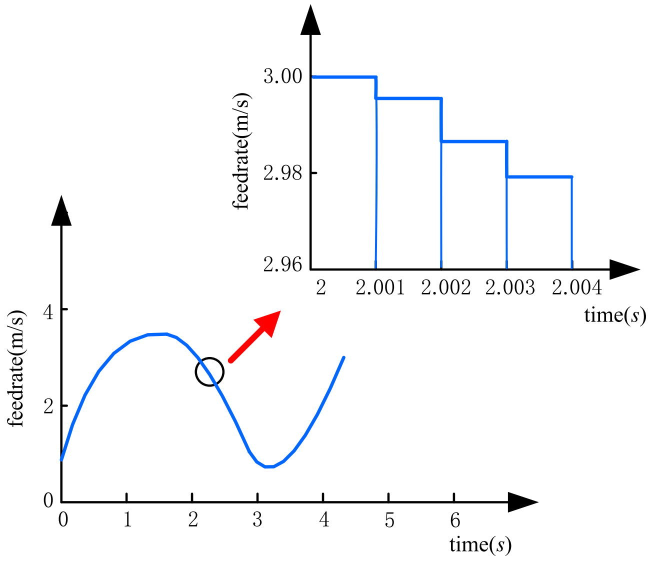


Fig. 9. Feedrate profile of interpolation.

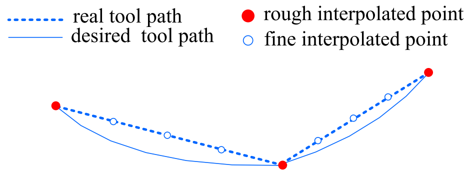


Fig. 10. Schema of fine interpolation.

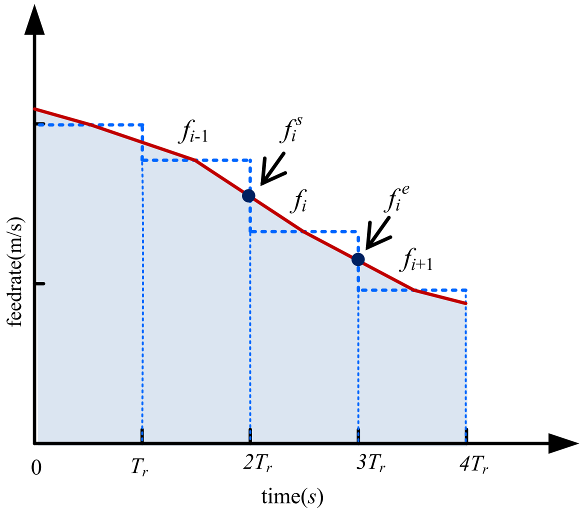


Fig. 11. Feedrate profile.

and the following period are f_{i-1}, f_i and f_{i+1} respectively according to the feedrate profile acquired from rough interpolation. The displacement of machining is the area under the feedrate curve in the figure. So when the area under feedrate curve is fixed, there is no extra error brought by the algorithm. The feedrate during one interpolation period changes from f_i^s to f_i^e, which are given by Eq. (8).

$$f_i^{s(e)} = \begin{cases} f_i - \dfrac{f_i - f_{i-1(i+1)}}{2} & f_i \geq f_{i-1(i+1)} \\ f_i + \dfrac{f_{i-1(i+1)} - f_i}{2} & f_i < f_{i-1(i+1)}. \end{cases} \tag{8}$$

The average feedrate between iT_r and $(i+1)T_r$ is decided by f_{i-1}, f_i and f_{i+1}. The detailed feedrate of each fine interpolation period within one rough interpolation period changes linearly.

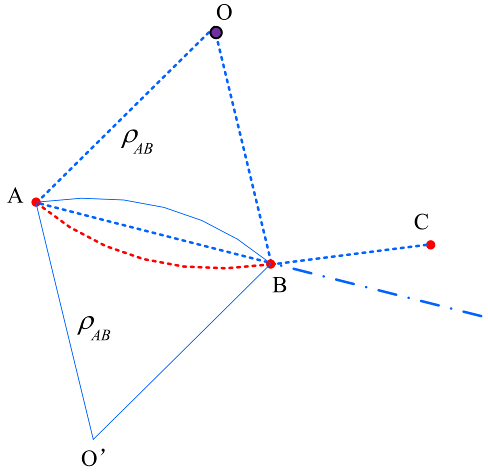


Fig. 12. Generation of interpolating arc.

Assume the current machining line segment is $\Delta \boldsymbol{L}_i$ which is confined with two adjacent rough interpolated pints $\boldsymbol{C}(u_i)$ and $\boldsymbol{C}(u_{i+1})$, the next rough interpolated point is $\boldsymbol{C}(u_{i+2})$. These three points are used to determine the fine interpolated points. The curvature of each point $(\boldsymbol{C}(u_i), \boldsymbol{C}(u_{i+1}), \boldsymbol{C}(u_{i+2}))$ which is calculated by Eq. (9) is also required.

$$K_i = K(u_i) = \left. \frac{\left\| \dfrac{dC(u)}{du} \times \dfrac{dC^2(u)}{du^2} \right\|}{\left\| \dfrac{dC(u)}{du} \right\|^3} \right|_{u=u_i}. \tag{9}$$

In order to determine the exact position of each fine interpolation period after the feedrate of each cycle is known, an interpolating curve is proposed for its fast-computation and effectiveness. As such, two stages are involved: the generation of a curve that travels along the rough interpolated points and the location determination of each fine interpolation period.

To generate the interpolating curve based on the original rough interpolated points, a circular arc is used to fit through these points. Just as shown in Fig. 9, the generating arc AB is derived from three rough interpolated points A, B and C. The radius of the interpolating arc is calculated with the average curvature in points A and B of the parametric curve as given by Eq. (11), and the concavity or convexity of the arc is determined by the relationship among the position of points A, B and C as illustrated in Fig. 12. According to the curvature of points A and B, there are two possibilities for the interpolating arc (as shown in Fig. 12), if the point C is located in the left of the line travelling through A and B, then the center point of the interpolating arc is located in the left, and vice visa. With judgments, one of the both candidates of center point O is defined as the exact one.

$$\rho_{AB} = \frac{1}{2}(\rho_A + \rho_B) = \frac{1}{2}\left(\frac{1}{K_A} + \frac{1}{K_B}\right). \tag{10}$$

After the interpolating arc has been determined, the fine interpolated points are calculated based on the determined feedrate of each cycle $(f_{i1}, f_{i2}, f_{i3}, f_{i4})$. The real feedrate is proportional to the angular speed which is the main theoretical basis for calculating points. Suppose the angle of two adjacent rough interpolation points is θ, and it should be divided into four parts $(\theta_{i1}, \theta_{i2}, \theta_{i3}, \theta_{i4})$ as shown in Fig. 13. The relationship between velocities and angles is given by Eq. (11). So the displacement of each fine interpolation is proportional to the feedrate determined by Eq. (8).

$$\frac{f_{i1}}{\theta_{i1}} = \frac{f_{i2}}{\theta_{i2}} = \frac{f_{i3}}{\theta_{i3}} = \frac{f_{i4}}{\theta_{i4}} = k = \frac{4\rho_{AB}}{T_r}. \tag{11}$$

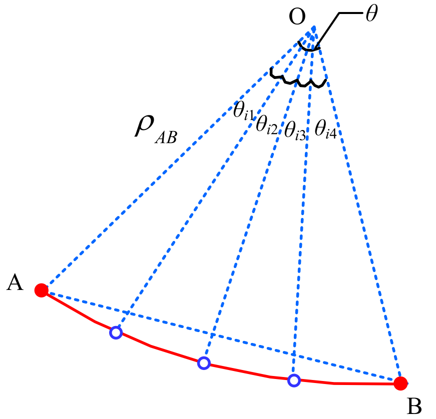


Fig. 13. Generation of fine interpolated points.

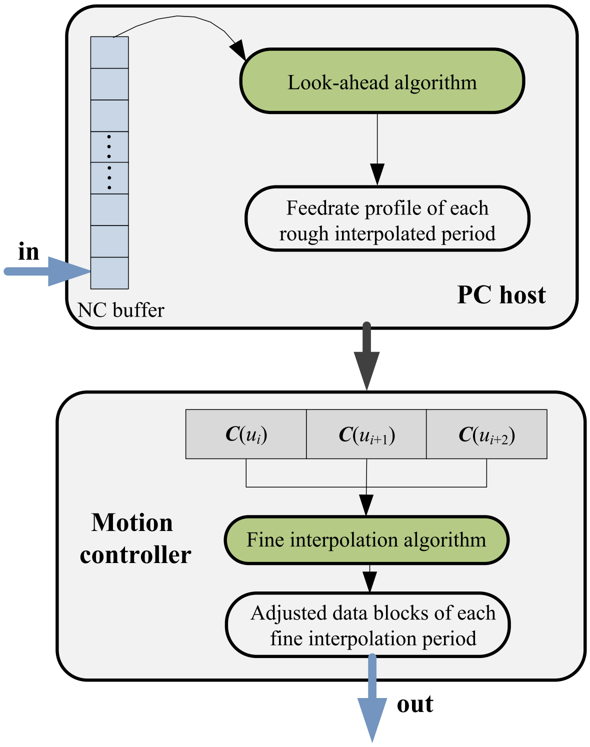

Look-ahead algorithm
Feedrate profile of each rough interpolated period
in
NC buffer
PC host
C(u_i)
C(u_i+1)
C(u_i+2)
Motion controller
Fine interpolation algorithm
Adjusted data blocks of each fine interpolation period
out

Fig. 14. Scheme of machining procedure.

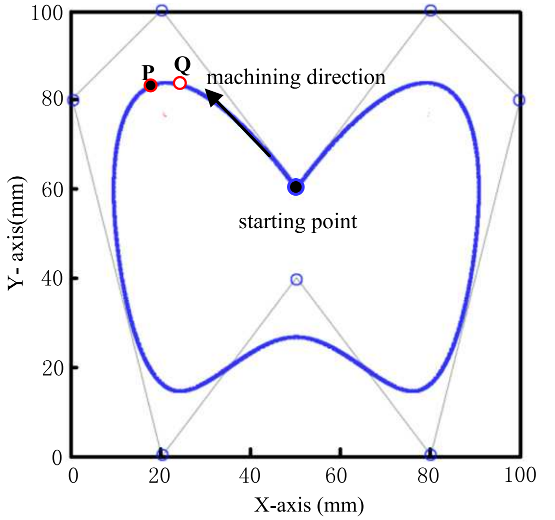


Fig. 15. Machining NURBS curve.

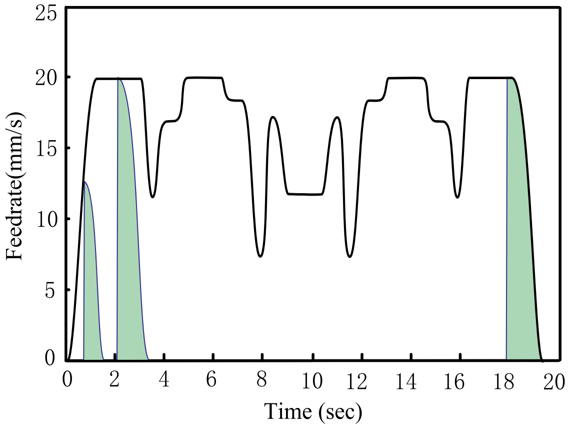


Fig. 16. Feedrate profile 1.

5. Simulation and experimentation of the algorithm

In order to verify the proposed algorithm, two models are adopted to machine in an open architecture CNC system shown in Fig. 2 which owns the capability of being redeveloped in the fine interpolation stage. The real-time look-ahead function is validated by the simulation of the first model. The kinematical characteristics during real machining are illustrated to verify the effectiveness of the proposed strategy in the fine interpolation stage. The fine interpolation algorithm in this paper is realized with C language and converted into hex file for burning into the motion controller. The real-time look-ahead algorithm is implemented in PC host with C++. The servo period of control loop is 1 ms set in the motion controller, so the interval for PC host sending rough interpolated data should not be larger than 4 ms as a data block from rough interpolation contains 4 units for fine interpolation.

The interpolating scheme for experimentation is illustrated in Fig. 14. After the interpretation of NC files, the data blocks of machining contours are stored in a buffer of PC host. The look-ahead algorithm can determine the feedrate of each rough interpolated cycle within the time of a sampling period (4 ms), while the fine interpolation algorithm generates the discrete parameters of every fine interpolation period (1 ms) based on the feedrate profile from buffer. The command signals of each servo cycle are subsequently sent to servo systems for semi-closed-loop control. Meanwhile, the feedback information is sampled by the motion controller to send back to PC host for data analysis.

A curve shown in Fig. 15 is employed as an experimental case to prove the efficiency and availability of the developed look-ahead scheme. Parameters are set as follows: the command feedrate is 20 mm/s, the maximal acceleration, jerk and jounce are 500 m/s^2, 10 000 m/s^3, 200 000 m/s^4, respectively, the maximal chord error $\delta_{\max}$ is 5 μm. The rough interpolation period T_r is set to 4 ms, while the fine interpolation period is 1 ms.

In the rough interpolation stage, the look-ahead process mentioned in Section 3 is implemented for the machined tool path. Firstly, the length of the tool path which needs to look forward is determined by Eq. (6), then the maximal feedrate along this tool path is determined according to various specified feedrate constraints. The length of the looking-forward path is changing continually with the current feedrate. More specifically, when the real-time look-ahead procedure just starts, the length of the looking-forward path is zero. Afterwards, the length increases with the feedrate going up. This can be seen from Fig. 16: the shadow areas at each point mean the looking-forward lengths which are

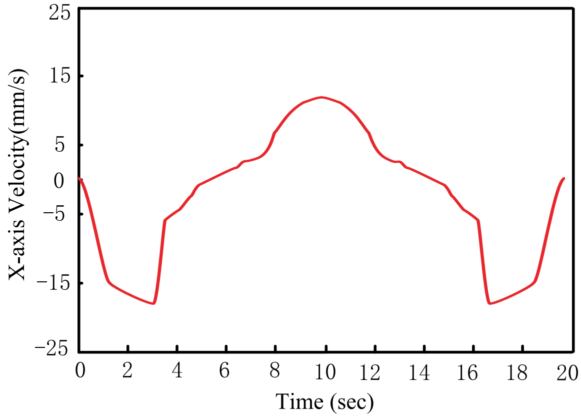


Fig. 17. *X*-axis velocity profile.

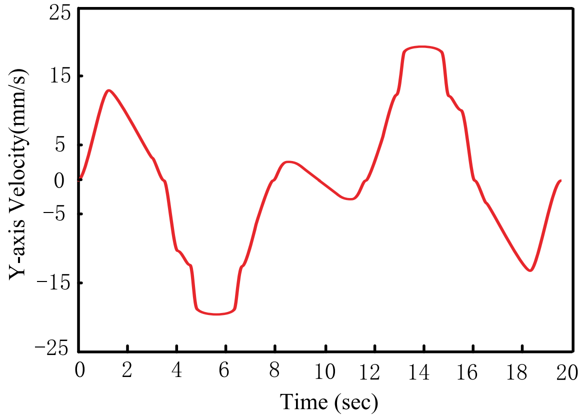


Fig. 18. *Y*-axis velocity profile.

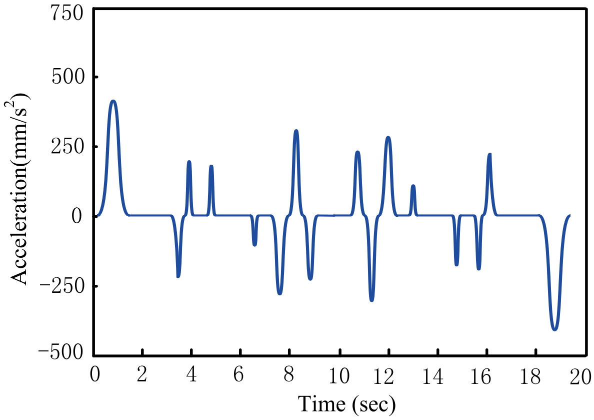


Fig. 19. Acceleration profile.

equal to the decelerating distance from the feedrate at those moments. For example, the area $\mathbf{S}_1$ represents the looking-forward path at t_1; similarly, the area $\mathbf{S}_2$ accounts for the corresponding path at time of t_2. At the moment of t_2, the end of the looking-forward path is a sensitive area (point **P** in Fig. 15) which means the feedrate at this position should be slowed down to satisfy the feedrate constraints. An inverse computation is adopted to determine the decelerating point (point **Q** in Fig. 15) based on the allowed feedrate at **P** and the applying ACC/DEC method. When the tool travels to **Q**, it begins to decelerate until it passes through **P** safely. During the deceleration process, the look-ahead scheme continues to perform on the following tool paths. From the feedrate profile, it is clear that the look-ahead algorithm can keep the feedrate changing smoothly through high-curvature areas.

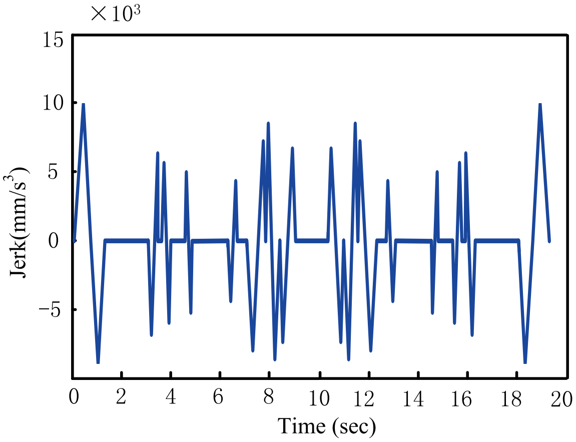


Fig. 20. Jerk profile.

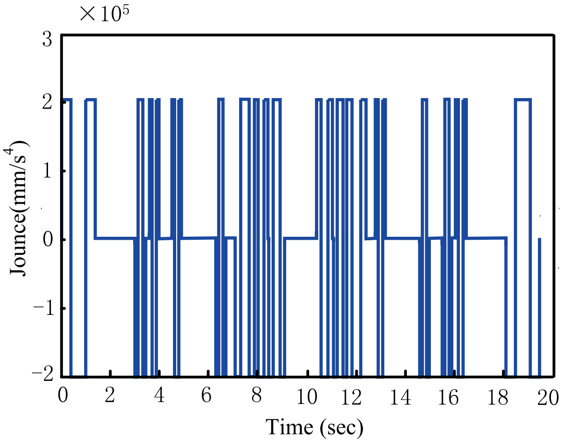


Fig. 21. Jounce profile.

The feedrate profiles generated from the rough interpolation are sent to the motion controller for fine interpolation. It is clear that the feedrate profile gained from the rough interpolation is comparatively smooth for machining. The feedrate profile changes gradually along the tool-path based on several constraints. The velocity profiles of each axis are also shown in Figs. 17 and 18, as well as the acceleration, jerk and jounce profiles that can be obtained from the rough interpolation are shown in Figs. 19–21. These simulation results prove the effectiveness and robustness of the proposed look-ahead scheme to determine the feedrate profile for CNC machining.

Besides, to verify the advantage of the applied jounce-limited ACC/DEC pattern, the acceleration and the jerk profiles based on the universally adopted S-shape ACC/DEC of the machining contours are shown in Figs. 23 and 24, which can be used to compare with the acceleration and the jerk profiles based on the proposed jounce-limited ACC/DEC pattern shown in Figs. 19 and 20. It is clear that the changes of the acceleration and the jerk are smoother which means the load caused by the acceleration process on tools and workpieces with our method changes smoothly. Therefore, the proposed ACC/DEC pattern is more suitable for high-speed machining.

Theoretically, the staircase of feedrate profile after fine interpolation process can be alleviated significantly with the proposed fine interpolation strategy. Meanwhile, the contour errors are limited to a smaller range as shown in Fig. 22 in comparison with those from the original interpolation method without the developed adjusting strategy in the fine interpolation stage. It is worth noticing that the chord error brought by the proposed interpolating method

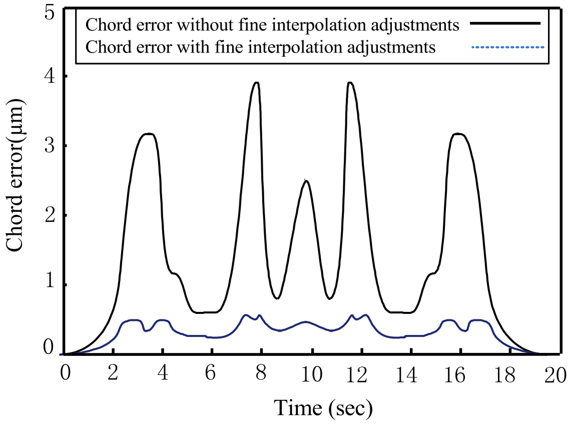


Fig. 22. Chord error.

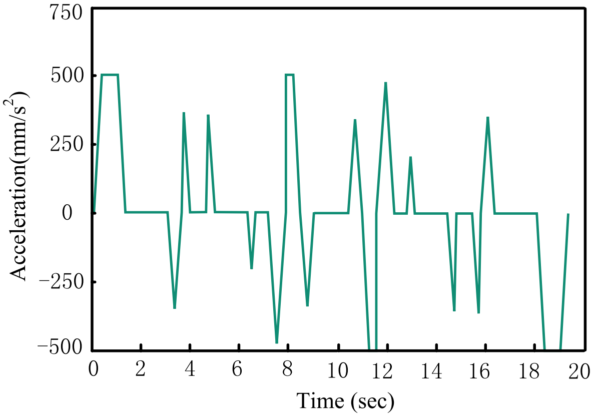


Fig. 23. Acceleration profile.

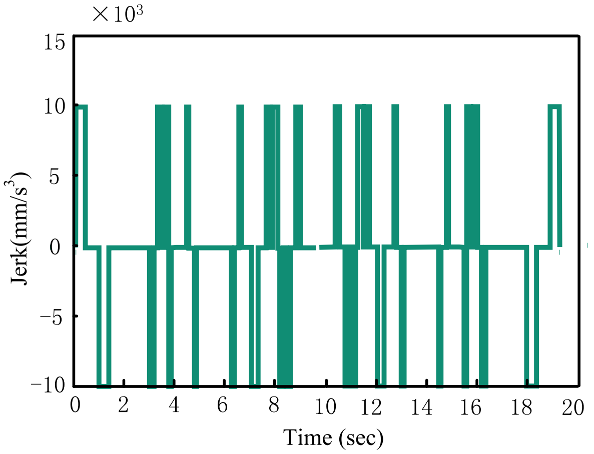


Fig. 24. Jerk profile.

is not just affected by the curvatures of the machining contours, but also influenced by the deviation between the interpolating arcs and the desired contours.

Another complex NURBS contour shown in Fig. 25 is also adopted to validate the proposed algorithm. The feedrate profile with and without the proposed fine interpolation strategy are shown in Fig. 26. Obviously, the tracking performance and accuracy for the machined contour with the introduced interpolators are improved.

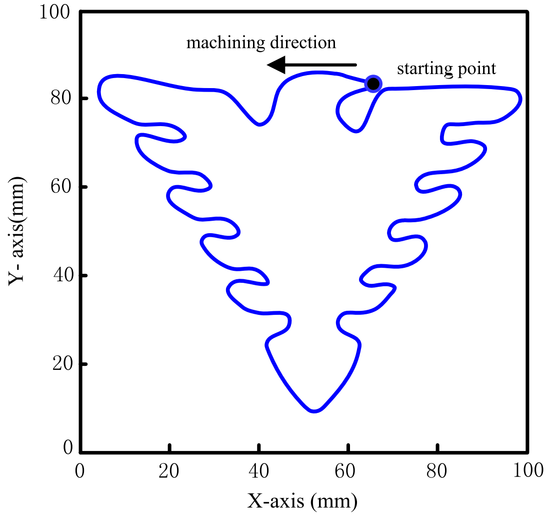


Fig. 25. Machining NURBS curve.

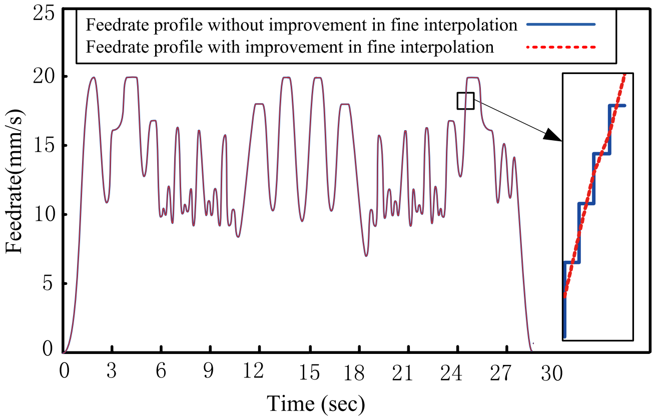


Fig. 26. Feedrate profile 2.

6. Conclusion

A real-time look-ahead scheme is proposed to achieve a quite smooth feedrate profile considering the confined chord error and a wide range of kinematical characteristics, as well as a strategy during fine interpolation stage. The look-ahead scheme is implemented in the rough interpolation stage in PC host, while the fine interpolation stage is performed in salve motion controller.

In the presented look-ahead algorithm, the length of the tool path that needs to look-ahead is mainly determined by the current feedrate and several kinematical constraints. The main task for the look-ahead process is to search sensitive areas that should reduce feedrate to an appropriate value which means deceleration processes before these critical areas are required. The proposed look-ahead algorithm has the advantage of speed as well as accuracy in comparison with other algorithms.

Most current interpolation algorithms achieve the final feedrate profile in the rough interpolation level, while the fine interpolation stage has not been involved. In this paper, several approaches in fine interpolation level are applied on the data block from rough interpolation to achieve a higher accuracy machining. An arc is used to substitute the original line path between two adjacent rough interpolated points and the chord errors are alleviated accordingly. Besides, the staircase of the feedrate profile can be improved with the consideration of feedrate variation during the fine interpolation stage.

According to the simulation and experimental results, the proposed real-time look-ahead method is robust as well as effective and the fluctuation in feedrate profile has been improved significantly. The proposed strategies in both the interpolation stages suit high-speed and high-accuracy CNC machining of parametric interpolation.

Acknowledgments

This paper is sponsored by the Science Fund for Creative Research Groups of National Natural Science Foundation of China (no. 51221004), National Natural Science Foundation of China (no. 51375440), Fundamental Research Funds for the Central Universities (no. 2013QNA4007), and Zhejiang Provincial Natural Science Foundation of China (no. LY12E05018).

Appendix. Proof of Eq. (6)

Proof. Kinetic parameters (e.g. jerk, acceleration, feedrate and displacement) during acceleration based on the jounce-limited ACC/DEC strategy are shown in Eqs. (A.1)–(A.4), where$J$ is jounce and t_0 is the time constant. When the change of feedrate is Δf, the constant t_0 is obtained from Eq. (A.5) based on Eqs. (A.1)–(A.3) when the jounce limit, jerk limit and acceleration limit are $J_{\max}$ $Jerk_{\max}$ and $Acc_{\max}$. Then the corresponding displacement during accelerating process can be acquired as shown in Eq. (A.6) based on the above formulas.

$$\text{Jerk} = \begin{cases} Jt & (0, t_0] \\ 2Jt_0 - Jt & (t_0, 3t_0] \\ Jt - 4Jt_0 & (3t_0, 4t_0] \end{cases} \tag{A.1}$$

$$\text{Acceleration} = \begin{cases} \frac{1}{2}Jt^2 & (0, t_0] \\ -\frac{1}{2}Jt^2 + 2Jt_0t - Jt_0^2 & (t_0, 3t_0] \\ \frac{1}{2}Jt^2 - 4Jt_0t + 8Jt_0^2 & (3t_0, 4t_0] \end{cases} \tag{A.2}$$

$$\text{Feedrate} = \begin{cases} \frac{1}{6}Jt^3 & (0, t_0] \\ \frac{1}{6}Jt^3 + Jt_0t^2 - Jt_0^2t + \frac{1}{3}Jt_0^3 & (t_0, 3t_0] \\ \frac{1}{6}Jt^3 - 2Jt_0t^2 + 8Jt_0^2t - \frac{26}{3}Jt_0^3 & (3t_0, 4t_0] \end{cases} \tag{A.3}$$

$$\text{Length} = \begin{cases} \frac{1}{24}Jt^4 & (0, t_0] \\ -\frac{1}{24}Jt^4 + \frac{1}{3}Jt_0t^3 - \frac{1}{2}Jt_0^2t^2 + \frac{1}{3}Jt_0^3t \\ \quad -\frac{1}{12}Jt_0^4 & (t_0, 3t_0] \\ \frac{1}{24}Jt^4 - \frac{2}{3}Jt_0t^3 + 4Jt_0^2t^2 - \frac{26}{3}Jt_0^3t \\ \quad +\frac{161}{24}Jt_0^4 & (3t_0, 4t_0] \end{cases} \tag{A.4}$$

$$t_0 = \min\left(\frac{\text{Jerk}_{\max}}{J_{\max}}, \left(\frac{\text{Acc}_{\max}}{J_{\max}}\right)^{\frac{1}{2}}, \left(\frac{\Delta f}{2J_{\max}}\right)^{\frac{1}{3}}\right) \tag{A.5}$$

$$\text{Length}\ (4t_0) = \frac{97}{24}J \cdot t_0^{\frac{4}{3}}. \tag{A.6}$$

As the decelerating process is a symmetry of the accelerating process, the length of the tool path for decelerating (S_{dec}) is shown in Eq. (A.7).

$$S_{dec} = \frac{97}{24}J \cdot t_0^{\frac{4}{3}}. \tag{A.7}$$

References

[1] Yeh S-S, Hsu P-L. Adaptive-feedrate interpolation for parametric curves with a confined chord error. Comput Aided Des 2002;34(3):229–37.

[2] Zhao H, Zhu LM, Ding H. A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. Int J Mach Tools Manuf 2013;65:88–98.

[3] Yau H-T, Wang J-B. Fast Bezier interpolator with real-time lookahead function for high-accuracy machining.. Int J Mach Tools Manuf 2007;47(10):1518–29.

[4] Baek D-K, Ko T-J, Yang S-H. Fast and precision NURBS interpolator for CNC systems. Int J Precis Eng Manuf 2012;13(6):955–61.

[5] Yau H-T, et al. PC-based controller with real-time look-ahead NURBS interpolator. Comput Aided Des Appl 2007;4(1–4):331–40.

[6] Wu J, et al. A NURBS interpolation algorithm with continuous feedrate. Int J Adv Manuf Technol 2012;59(5–8):623–32.

[7] Tikhon M, et al. NURBS interpolator for constant material removal rate in open NC machine tools. Int J Mach Tools Manuf 2004;44(2–3):237–45.

[8] Lin M-T, Tsai M-S, Yau H-T. Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. Int J Mach Tools Manuf 2007;47(15):2246–62.

[9] Du D, et al. An accurate adaptive NURBS curve interpolator with real-time flexible acceleration/deceleration control. Robot Comput-Integr Manuf 2010; 26(4):273–81.

[10] Heng M, Erkorkmaz K. Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability. Int J Mach Tools Manuf 2010;50(3):281–93.

[11] Shen HY, Fu JZ, Fan YQ. A new adaptive interpolation scheme of NURBS based on axis dynamics. Int J Adv Manuf Technol 2011;56(1–4):215–21.

[12] Lee A-C, et al. The feedrate scheduling of NURBS interpolator for CNC machine tools. Comput Aided Design 2011;43(6):612–28.

[13] Zhou J, Sun Y, Guo D. Adaptive feedrate interpolation with multiconstraints for five-axis parametric toolpath. Int J Adv Manuf Technol 2014;71(9–12): 1873–82.

[14] Beudaert X, Lavernhe S, Tournier C. Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path. Int J Machine Tools Manuf 2012; 57:73–82.

[15] Sun YW, et al. A novel adaptive-feedrate interpolation method for NURBS tool path with drive constraints. Int J Machine Tools Manuf 2014;77:74–81.

[16] Zhao H, Zhu LM, Ding H. A parametric interpolator with minimal feed fluctuation for CNC machine tools using arc-length compensation and feedback correction. Int J Machine Tools Manuf 2013;75:1–8.

[17] Nam S-H, Yang M-Y. A study on a generalized parametric interpolator with real-time jerk-limited acceleration. Comput Aided Design 2004;36(1):27–36.

[18] Tsai M-C, et al. On acceleration/deceleration before interpolation for CNC motion control. In: ICM'05. IEEE International Conference on Mechatronics, 2005. IEEE; 2005.

[19] Tsai M-S, Nien H-W, Yau H-T. Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. Comput Aided Design 2008;40(5):554–66.

[20] Emami MM, Arezoo B. A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. Comput Aided Design 2010;42(7):625–32.

[21] Ye P, et al. Interpolation of continuous micro line segment trajectories based on look-ahead algorithm in high-speed machining. Int J Adv Manuf Technol 2008;37(9–10):881–97.

[22] Shen H, et al. On-line asynchronous compensation methods for static/quasi-static error implemented on CNC machine tools. Int J Machine Tools Manuf 2012;60(0):14–26.

[23] Suh S-H, et al. Theory and Design of CNC Systems. Springer; 2008.

[24] Yong T, Narayanaswami R. A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining. Comput Aided Design 2003;35(13):1249–59.

[25] Wang L, Cao J. A look-ahead and adaptive speed control algorithm for high-speed CNC equipment. Int J Adv Manuf Technol 2012;63(5–8):705–17.

[26] Jin Y-a, He Y, Fu J-z. A look-ahead and adaptive speed control algorithm for parametric interpolation. Int J Adv Manuf Technol 2013;69(9–12):2613–20.

[27] Jin Y-a, et al. An interpolation method for the open CNC system based on EPM.. Int J Adv Manuf Technol 2013;69(1–4):405–16.

[28] Ji S, et al. A parametric hardware fine acceleration/deceleration algorithm and its implementation. Int J Adv Manuf Technol 2012;63(9–12):1109–15.