Int J Adv Manuf Technol (2007) 35:505–518
DOI 10.1007/s00170-006-0735-8

ORIGINAL ARTICLE

A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead

Fu-Yuan Luo · Yun-Fei Zhou · Juan Yin

Received: 5 April 2006 / Accepted: 25 July 2006 / Published online: 8 November 2006

Abstract Usually a complex contour-machining program generated by CAD/CAM systems composes a lot of small line segments. The intelligence of look-ahead is key to meeting the demand of high speed and high accuracy in the machining of these line segments. By now most researchers just utilize the simplest trapezoidal velocity profiles to deal with the look-ahead problem. Also the number of look-ahead line segments in most commercially available CNC systems is fixed, which cannot make full use of CNC processors. Therefore, a universal velocity profiles generation approach and corresponding optimal look-ahead algorithm based on dynamic back tracking along a doubly linked list are proposed in this paper. Two novel strategies, one for reducing feed rates fluctuation and the other for reducing move errors that come from digital integration and accumulated computation, are also presented. By using the proposed look-ahead techniques, arbitrary velocity profiles having the desired acceleration and deceleration characteristics for the movement of a lot of small line segments can be generated efficiently. Thus, the machining productivity can be dramatically increased without sacrifice of accuracy. The results of simulations and experiments showed the proposed approach was feasible and effective.

F.-Y. Luo (✉) · Y.-F. Zhou
National Technical Research Center-Numerical Control System,
School of Mechanical Science and Engineering,
Huazhong Science and Technology University,
Wuhan, Hubei, China
e-mail: fuyuanluo@126.com

J. Yin
School of Machinery and Automobile Engineering,
Hefei University of Technology,
Hefei, Anhui, China

Keywords Acceleration and deceleration · Back tracking · Error compensation · Look-ahead · Numerical integration · Velocity profile

1 Introduction

High speed and high accuracy are increasingly demanded in modern manufacturing. In order to make computer numerical control (CNC) machine tools perform given tasks more quickly and more accurately, many approaches have been proposed for generating velocity profiles that have the desired acceleration and deceleration characteristics. One of the most efficient approaches is digital convolution technique, which generates velocity profiles by using digital convolutions [1]. However, the digital convolution approach fails to generate various velocity profiles that are useful for CNC machine tools. In this case, Jeon and Ha proposed a coefficients-based method that can generate velocity profiles with arbitrary acceleration and deceleration characteristics [2]. In order to fully utilize the maximum acceleration for short distance movement, Jeon improved the approach by selecting appropriate acceleration and deceleration intervals according to a given distance [3]. Nevertheless, the approach only addresses the case that both starting point speed and end point speed of a movement are zero. So when it is used in CNC machining, the CNC has to stop at the end of one movement before going on to the next one. If the distance of the movement is too small, the CNC cannot accelerate to a high enough speed. Since a complex contour-machining program generated by CAD/CAM systems usually composes a lot of small line segments, actual feed rate is heavily reduced and the consequential machining efficiency becomes very low.

To achieve faster machining speed, along with required accuracy, CNC systems must take advantage of the intelligence of look-ahead. Look-ahead does just what its name implies: it looks ahead of a certain number of NC-code data blocks at any given moment to verify that the CNC will be able to handle the demands the path will put on the machine [4]. Look-ahead may be referred to as "automatic feed rate override". For example, if data check reveals that there is a sharp corner in front, CNC system will adjust the feed rate downward to insure accuracy; if the corner is not sharp, it will adjust the feed rate upward for less stopping and maximum average velocity. In recent years a few researchers have been addressing the so-called look-ahead intelligence. Cao and Chang proposed a kind of smooth control algorithm that has the function of judging the speed at the turning point and used it in CNC carving machines [5]. Hu, Xiao, Wang and Wu designed an optimal feed rate model and proposed an algorithm to seek the approximate optimal feed rate by evaluating the tool path ahead [6]. However, all authors in the two articles only consider the simplest linear acceleration and deceleration, i.e., trapezoidal velocity profile, which usually exhibits a jump in acceleration at the beginning and the end of velocity adaptation phase and results in disadvantageous impact on machining tools. They also neglect the fact that sampling periods must be integers in practical CNC systems. So their algorithm to find optimal feed rate is neither universal nor accurate. Taking both contour error and machining efficiency into account, Tsai, Cheng, Liu and Tsai proposed an improved digital convolution based multi-segments look-ahead approach [7]. But they disregarded the constraints on velocity at sharp corners and in short distance movements. Moreover, as mentioned earlier, the digital convolution approach cannot generate various velocity profiles that are useful for CNC machine tools.

Some commercially available CNC systems such as Fanuc 21iGE-series and Heidenhain iTNC 530 have provided the feature of look-ahead intelligence. However, most of them just use linear acceleration and deceleration characteristics. In addition, the number of line segments processed by these look-ahead systems is usually fixed, which cannot make full use of the potential of CNC processors. To a few of advanced CNC systems such as Heidenhain Atek with Cyclone Expert, the number of look-ahead lines is dynamic, which can make full use of the potential of CNC processors, but the related algorithm is unpublicized.

According to that mentioned above, a universal velocity profile generation approach and corresponding optimal look-ahead algorithm are proposed in this paper. The proposed look-ahead approach extends the work of Jeon greatly. It allows for a nonzero starting point speed and end point speed of every line segment to increase machining productivity. It is based on dynamic back tracking along a doubly linked list, so the number of look-ahead line segments is dynamic. Moreover, it uses two effective strategies to reduce feed rates fluctuation and the move errors that come from digital integration and accumulated computation, so the generated velocity profile gets smoother and the machining accuracy is improved. Accordingly, the proposed approach is more practical for high-speed and high-accuracy CNC machining.

This paper is organized as follows. In the next section, all constrains on feed rate are deduced according to the kinematics model with universal velocity profiles. In Section 3, a look-ahead algorithm for seeking optimal feed rates with given constrains is put forward. In Section 4, the resulting optimal feed rates are used to generate actual velocity profiles according to given distance of every small line. Two novel strategies, one for reducing feed rates fluctuation and the other for reducing move errors are also introduced. Simulations and experiments applying the previous approach and algorithm are given in Section 5. The paper concludes with Section 6.

2 Velocity profile formulation

2.1 Arbitrary acceleration and deceleration formulation based on coefficients

Following the idea of Jeon in [3], define a variable u ($0 \leq u \leq 1$) and two functions $f_a(u)$ and $f_d(u)$, which represent the desired acceleration and deceleration characteristics, respectively. Both $f_a(u)$ and $f_d(u)$ should be differential on $0<u<1$ and are continuous on $0 \leq u \leq 1$. Moreover, they must satisfy Eqs. (1) and (2).

$$f_a(0) = 0;\ f_a(1) = 1;\ f_d(0) = 1;\ f_d(1) = 0 \tag{1}$$

$$\max_{0 \leq u \leq 1} |f_a(u)| = 1;\quad \max_{0 \leq u \leq 1} |f_d(u)| = 1 \tag{2}$$

For example, for linear acceleration and deceleration, the two functions can be (3); for cosine acceleration and deceleration, the two functions can be (4). It is obvious that both of them satisfy Eqs. (1) and (2).

$$f_a(u) = u; f_d(u) = 1 - u \tag{3}$$

$$\begin{aligned} f_a(u) &= 0.5(1 - \cos(\pi u)) \\ f_d(u) &= 0.5(1 + \cos(\pi u)) \end{aligned} \tag{4}$$

Let V_{start} and V_{end} represent starting point speed and end point speed of a straight-line movement, respectively. If $V_{start}<V_{end}$, a single acceleration movement can be de-

scribed by Eq. (5); If $V_{start}>V_{end}$, a single deceleration movement can be described by Eq. (6).

$$V(u) = f_a(u)(V_{end} - V_{start}) + V_{start} \ \ for\, 0 \leq u \leq 1 \tag{5}$$

$$V(u) = f_d(u)(V_{start} - V_{end}) + V_{end} \ \ for\, 0 \leq u \leq 1 \tag{6}$$

According to Eq. (1), in both Eqs. (5) and (6), if u=0, then $V(u)=V_{start}$; if u=1, then $V(u)=V_{end}$. Let t_a and t_d represent the time of acceleration and deceleration, respectively. Then Eq. (5) can be rewritten as Eq. (7) by replacing u with t/t_a. Similarly, Eq. (6) can be rewritten as Eq. (8) by replacing u with t/t_d.

$$V(t) = f_a(t/t_a)(V_{end} - V_{start}) + V_{start} \ \ for\, 0 \leq t \leq t_a \tag{7}$$

$$V(t) = f_d(t/t_d)(V_{start} - V_{end}) + V_{end} \ \ for\, 0 \leq t \leq t_d \tag{8}$$

Then, the position increment during each sampling period T_s in the acceleration interval can be represented as

$$\begin{aligned} {}^aP(k) &= \int_{(k-1)T_s}^{kT_s} [f_a(t/t_a)(V_{end} - V_{start}) + V_{start}]dt \\ &= V_{start}T_s + (V_{end} - V_{start})\int_{(k-1)T_s}^{kT_s} f_a(t/t_a)dt \\ &= V_{start}T_s + (V_{end} - V_{start})t_a\int_{(k-1)/n_a}^{k/n_a} f_a(u)du \end{aligned} \tag{9}$$

And it can be written as:

$${}^aP(k) = T_s[V_{start} + (V_{end} - V_{start})n_a{}^a\gamma(k)] \tag{10}$$

where $n_a=t_a/T_s$, ${}^a\gamma(k)$are the coefficients which can be calculated from

$${}^a\gamma(k) = \int_{(k-1)/n_a}^{k/n_a} f_a(u)du \text{ for } k = 1, 2, \ldots, n_a \tag{11}$$

Similarly, the position increments during each sampling time in the deceleration interval can be represented as:

$${}^dP(k) = T_s\left[V_{end} + (V_{start} - V_{end})n_d{}^d\gamma(k)\right] \tag{12}$$

where $n_d=t_d/T_s$, ${}^d\gamma(k)$ are the coefficients which can be calculated from

$${}^d\gamma(k) = \int_{(k-1)/n_d}^{k/n_d} f_d(u)du \ \text{ for } k = 1, 2, \ldots, n_d \tag{13}$$

Coefficients in Eq. (11) can be calculated from n_a and $f_a(u)$, coefficients in Eq. (13) can be calculated from n_d and $f_d(u)$. All these coefficients can be stored in advance. Further, defining

$$R_a = \max_{0\leq u\leq 1} \left|\frac{d}{du}f_a(u)\right|; \quad R_d = \max_{0\leq u\leq 1} \left|\frac{d}{du}f_d(u)\right| \tag{14}$$

$$\alpha_a = \int_0^1 f_a(u)du; \ \alpha_d = \int_0^1 f_d(u)du \tag{15}$$

If the acceleration characteristic represented by $f_a(u)$ and the deceleration characteristic represented by $f_d(u)$ are symmetrical, $R_a=R_d$, $\alpha_a=\alpha_d$. As in the previous examples, for linear acceleration and deceleration, according to (3), $R_a=R_d=1$, $\alpha_a=\alpha_d=1/2$; for cosine acceleration and deceleration, according to Eq. (4), $R_a=R_d=\pi/2$, $\alpha_a=\alpha_d=1/2$. However, in asymmetrical case, it is possible that $R_a\neq R_d$, $\alpha_a\neq\alpha_d$. For example, in the case that the cosine acceleration characteristic combines with parabolic deceleration characteristic such that

$$\begin{aligned} f_a(u) &= 0.5(1 - \cos(\pi u)) \\ f_d(u) &= 1 - u^2 \end{aligned} \tag{16}$$

Then, $R_a=\pi/2$, $R_d=2$, $\alpha_a=1/2$, $\alpha_d=2/3$, so $R_a\neq R_d$, $\alpha_a\neq\alpha_d$. To simplify formulation and save storage space, symmetrical acceleration and deceleration characteristics is used in this paper.

Defining

$$R_a = R_d = R \qquad \alpha_a = \alpha_d = \alpha \tag{17}$$

Then R and α can be stored in advance and directly used in the further calculations without considering the difference between R_a and R_d and the difference between α_a and α_d. Otherwise, some formulas maybe very complicated.

2.2 Maximum speed constraint

In actual machining, the feed rate is subject to many physical factors, such as torque/power of motors, tool material, tool design, tool flank angles, flexure, breakage, chip removal, heat generation, and so on. Usually the system model would be too complicated to include all these factors. In most cases, however, they only influence the choosing of maximum allowable speed (i.e., feed rate). Let V_{max1}, V_{max2}, V_{max3}, etc., represent the maximum speed allowed by torque/power of motors, the maximum speed allowed by the properties of the tool material, the maximum speed allowed by chip removal, etc., respectively. Thus, a safe maximum speed V_{max} allowing for all of the physical factors can be defined as

$$V_{\max} = \{V_{\max 1}, V_{\max 2}, V_{\max 3}, \ldots\} \tag{18}$$

In any case, the actual speed of the machine should not exceed V_{max}, that is

$$V_i \leq V_{\max} \tag{19}$$

2.3 Context speed and moving distance constraints

Assume L_i is the moving distance of the ith line segment. V_{i-1}, V_i are the starting point speed and end point speed of the movement covering distance L_i, respectively. Besides, V_m is the actual maximum speed; A_{max} is the maximum allowable acceleration. As mentioned above, V_{max} is the maximum allowable speed.

In the case of speeding up from given V_{i-1} to V_i, there are three types of velocity profile that cover the distance L_i, as shown in Fig. 1(a). One is to fully accelerate to speed V_m without constant speed phase and deceleration phase, $V_i = V_m$. The second type is to accelerate to speed V' and keep moving in this speed until end point is arrived, $V_i = V'$. The last one is to accelerate to a certain speed, then change to deceleration phase and reach the end point at speed V'', $V_i = V''$. Obviously, since the moving distance must be equal to L_i, there is a relationship that

$$V' < V_m, \quad V'' < V_m \tag{20}$$

Hence

$$V_i \leq V_m \tag{21}$$

According to Eqs. (15), (17) and Fig. 1(a),

$$\begin{aligned} L_i &= \int_0^{t_a} (V_m - V_{i-1}) f_a(u) du + V_{i-1} t_a \\ &= (V_m - V_{i-1}) \alpha t_a + V_{i-1} t_a \end{aligned} \tag{22}$$

where t_a is the acceleration time. Suppose during the acceleration phase, maximum level of acceleration is fully utilized, such that

$$\max_{0 \leq t \leq t_a} \left| \frac{d}{dt} (V_m - V_{i-1}) f_a(t/t_a) \right| = A_{\max} \tag{23}$$

According to Eqs. (14) and (17),

$$\max_{0 \leq t \leq t_a} \left| \frac{d}{dt} (V_m - V_{i-1}) f_a(t/t_a) \right| = \frac{(V_m - V_{i-1}) R}{t_a} = A_{\max} \tag{24}$$

Thus

$$t_a = (V_m - V_{i-1}) R / A_{\max} \tag{25}$$

Combining Eqs. (21), (22), and (25) yields

$$V_i \leq V_m = \frac{-(1 - 2\alpha) V_{i-1} + \sqrt{V_{i-1}^2 + 4\alpha L_i A_{\max} / R}}{2\alpha} \tag{26}$$

In the case of deceleration from given V_i to V_{i+1}, there are also three types of velocity profile that covers the distance L_{i+1}, as shown in Fig. 1(b), where t_d is the deceleration time. By similar deduction, one can get

$$V_i \leq V_m = \frac{-(1 - 2\alpha) V_{i+1} + \sqrt{V_{i+1}^2 + 4\alpha L_{i+1} A_{\max} / R}}{2\alpha} \tag{27}$$

Equations (26) and (27) show that the maximum end point speed of a movement of a small line segment is subject to not only starting point speed and moving distance of itself but also end point speed and moving distance of the next movement.

2.4 Corner angle constraint

Let β represent the corner angle between two consecutive line segments, as shown in Fig. 2. In the simplest case, if the next point simply extends the current path further along a line and there is no deviation, i.e., β=180°, then no deceleration is required, the two consecutive line segments can be reassembled to a single line segment. Unlike that, if the next point is at a 180-degree deviation from the current path, the axes must be stopped completely. So there must be certain constraints due to the angle. Let V_A represent the end point velocity of the ith line segment and V_B be the starting

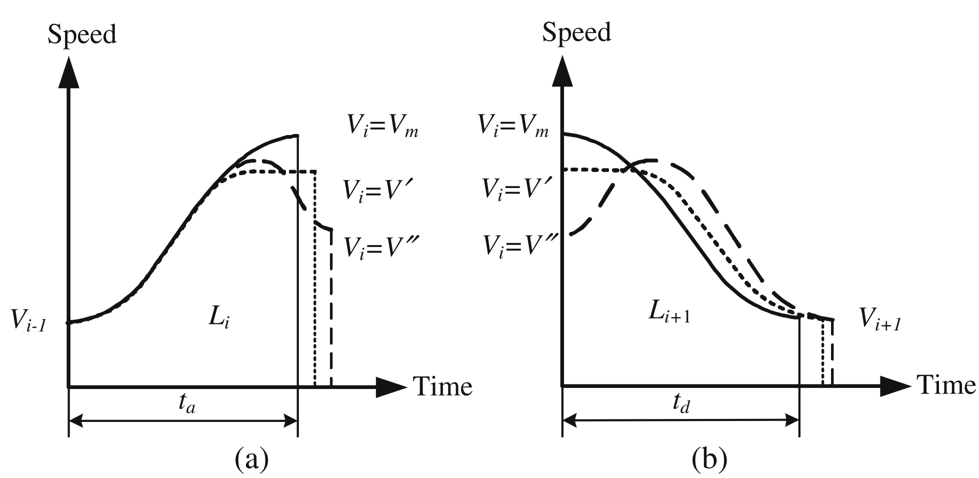


Fig. 1 Various velocity profiles for movement of specified distance

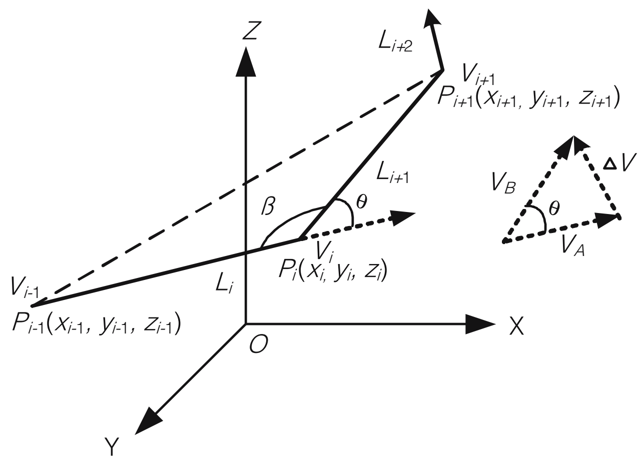


Fig. 2 Speed constraints due to deviation of consecutive line blocks

point velocity of the i+1th line segment. Note that velocity is a vector quantity that has both magnitude and direction. Machine tools must start moving along the i+1th line segment as soon as it finishes moving along the ith line segment. During this process, the velocity vector of the machine is constant in magnitude but changing in direction. That is to say, the magnitude of V_A is as large as V_B, i.e., $|V_A|=|V_B|=V_i$, but the direction of V_A is different from that of V_B (except that there is no deviation), so the machine experiences a velocity change and therefore an acceleration. According to Fig. 2, the magnitude of the acceleration is:

$$a=\frac{|\Delta V|}{T_s}=\frac{|V_B-V_A|}{T_s}=\frac{2V_i\sin(\theta/2)}{T_s}$$
$$=\frac{2V_i\cos(\beta/2)}{T_s}\leq A_{\max} \tag{28}$$

where θ is the angle of deviation, $\theta=\pi-\beta$, T_s is sampling period. The inequality Eq. (28) can be rewritten as

$$V_i\leq\frac{A_{\max}T_s}{2\cos(\beta/2)} \tag{29}$$

The value of cos(β/2) can be got by cosine law as the following:

$$\cos\frac{\beta}{2}=\sqrt{\frac{1+\cos\beta}{2}}=\sqrt{\frac{1}{2}+\frac{L_i^2+L_{i+1}^2-\left|\overline{P_{i+1}P_{i-1}}\right|}{4L_iL_{i+1}}} \tag{30}$$

where, P_{i+1} is the end point of the i+1th line segment, P_{i-1} is the starting point of the ith line segment. Their coordinates are $(x_{i+1}, y_{i+1}, z_{i+1})$ and $(x_{i-1}, y_{i-1}, z_{i-1})$, respectively. $\left|\overline{P_{i+1}P_{i-1}}\right|$ is the distance between the two points, i.e.,

$$\left|\overline{P_{i+1}P_{i-1}}\right|=\sqrt{(x_{i+1}-x_{i-1})^2+(y_{i+1}-y_{i-1})^2+(z_{i+1}-z_{i-1})^2} \tag{31}$$

According to Eq. (29), $V_i\leq+\infty$ for β=180°, which means no constraint is needed when the next point is in the extension of the current line segment; $V_i\leq A_{max}T_s/2$ (usually a very small number) for β=0°, which means the feed rate should be lowered to almost zero when the next point is at a 180-degree deviation from the current line segments.

However, as mentioned before, if the next point is at a 180-degree deviation from the current path, i.e., β=0°, the axes must be stopped completely. So inequality Eq. (29) cannot fully explain the constraints due to deviation of consecutive line segments. Someone even claimed the axes must be stopped completely if $\beta\leq$90° because at least one axis must experience a direction change if $\beta\leq$90° and the axis must experience a speed of zero during the direction change. In fact, it is possible that an axis experience a direction change even if β>90°. So the condition $\beta\leq$90° is not sufficient to include all the cases of direction change. Hence, a new method for determining when zero speed is demanded is proposed below.

In a movement from point A to point B, if the coordinates of one axis does not change, mark 0; if the coordinates of one axis increases, mark 1; if the coordinates of one axis decreases, mark –1. In the next movement from point B to point C, make a mark in a same manner. Now it can be determined whether or not the axis experiences a direction change at turning point B: if the marks change from 1 to -1 or from –1 to 1, i.e., the first mark multiplied by the second mark is –1, the axis must experience a direction change and its speed at point B must be zero. Further, if the mark changes from 1 to 0 or from –1 to 0, the speed at point B of this axis should be zero, too. According to the principle of linear interpolation, for a three-axis machine tool, if any axis (x, y, or z axis) is demanded to stop completely at turning point, the resultant velocity should be zero at turning point.

As a result, each time to determine the speed at a corner, the above simple calculation should be made before using Eq. (29). That is to say,

$$V_i\leq\begin{cases}0 & (\textit{if need stop})\\ A_{\max}T_s/[2\cos(\beta/2)]\end{cases} \tag{32}$$

This method ensures the resultant velocity is zero in the case of not only $\beta\leq$90° but also β>90° while an axis experiences a direction change. This method also ensures the resultant velocity keeps in a reasonable value when no axis experiences a direction change.

3 Optimal look-ahead algorithm

In summary, the constraints on the end point speed of a movement of a small line segment can be formulated as:

$$\begin{cases} (\mathrm{a})V_i \le \dfrac{-(1-2\alpha)V_{i-1} + \sqrt{V_{i-1}^2 + 4\alpha L_i A_{\max}/R}}{2\alpha} \\ (\mathrm{b})V_i \le \dfrac{-(1-2\alpha)V_{i+1} + \sqrt{V_{i+1}^2 + 4\alpha L_{i+1} A_{\max}/R}}{2\alpha} \\ (\mathrm{c})V_i \le \begin{cases} 0 & (\textit{if need stop}) \\ A_{\max}T_s/[2\cos(\beta/2)] \end{cases} \\ (\mathrm{d})V_i \le V_{\max} \end{cases} \tag{33}$$

Obviously, the look-ahead algorithm for seeking the optimal end point speed, which determines optimal feed rate in CNC machining, becomes the problem of finding out the maximum value of V_i that satisfies the group of inequalities with initial state $V_0=0$. The traditional algorithm deals with these inequalities disorderly. Suppose the maximum number of look-ahead line segments is N. In the traditional way, each time to determine V_i, the CNC processor has to read the coordinates of the next $(i+j)$th $(1 \le j \le N)$ line segment and calculate the maximum value of each V_{i+j} according to Eq. (33) on the assumption $V_{i+N+1}=0$. Hence, to get an optimal V_i, every right half of the inequality in Eq. (33) is calculated for N times. To get the next optimal V_i, a calculation task as heavy as the previous one is imposed on CNC processors.

In fact, there is a lot of redundant calculation in the above algorithm. Repetitive reading coordinates of the same point should be avoided absolutely. Calculation of the moving distance L_i of ith line segment and the speed limited to corner angle in Eq. (33)(c) should be taken only once. Moreover, in most cases, the calculation results of the right half of Eqs. (33)(a) and (33)(b) are used for comparing instead of getting actual feed rate, so some techniques for fast comparison can be introduced. An improved look-ahead algorithm is described below and illustrated by Fig. 3 where the number of total line segments to be read is m and $N=3$.

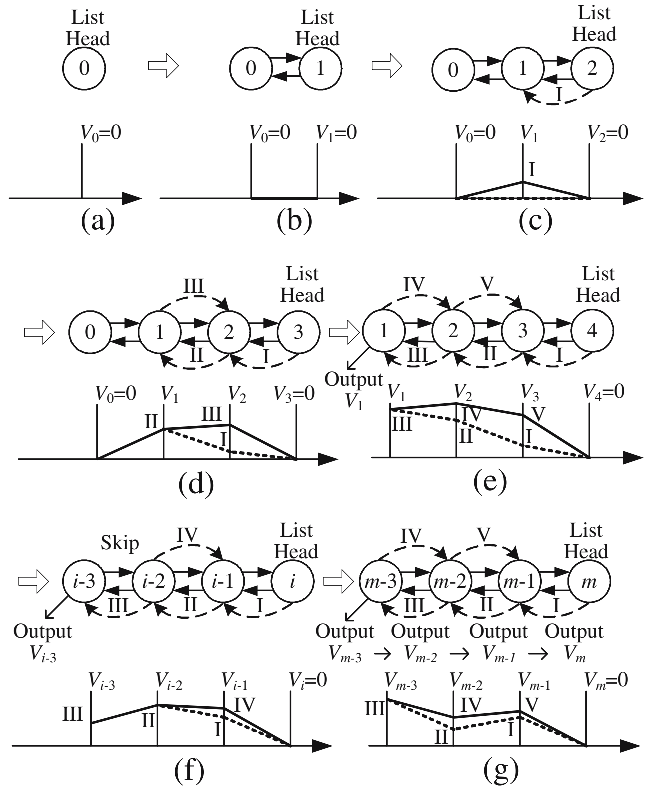


Fig. 3 Illustration of optimal look-ahead algorithm

1. Construct a doubly linked list of which each node includes the following properties. {
   - Coordinates of end point: P,
   - Moving distance: L,
   - Speed limited to corner angle: V_{lim_corner},
   - Speed limited to previous line: $V_{lim\_previous}$,
   - Need to improve by previous line: $YesNo_{V\_previous}$,
   - Speed limited to next line: V_{lim_next},
   - Need to improve by next line: $YesNo_{V\_next}$,
   - Current minimum speed: V_{min}

   }

   In initialization, the list has only a head node, as illustrated in Fig. 3(a). V_{min} of the head node is set to zero. The property $YesNo_{V\_next}$ of the node is set to NO to prevent back tracking described in step 4.
2. Set up a new node for newly read line segment, record P and calculate L. In addition, V_{min} is set to zero; the property $YesNo_{V\_previous}$ is set to NO. Then insert this node into the list before its head node. Thus, the old head node becomes the second node.
3. If only one line segment has been read, continue to read next line segment by step 2. Otherwise, it means that more than two line segments has been read, as illustrated in Fig. 3(c). Now, for the second node of the list, all the context constraints is available, i.e., all properties of this node can be determined. P and L have been registered before. V_{lim_corner} can be calculated by Eq. (33)(c). If $V_{lim\_corner} > V_{max}$, then let $V_{lim\_corner} = V_{max.}$ The left five properties are determined by the following rules:

If L is greater than S_{max}, then $V_{lim\_previous}=V_{max}$, else $V_{lim\_previous}$ is calculated by Eq. (33)(a). If $V_{lim\_previous}$ is greater than V_{lim_corner}, it means the speed is not limited to $V_{lim\_previous}$ instead of V_{lim_corner}, so set $YesNo_{V\_previous}$=NO; Otherwise $YesNo_{V\_previous}$=YES. Where, S_{max} is the minimum distance for CNC to accelerate from zero to V_{max} speed or decelerate from V_{max} to zero speed. S_{max} is calculated by Eq. (34) that comes from replacing V_m, V_{i-1} with V_{max} and zero in Eqs. (22) and (25) respectively. S_{max} can be stored in advance.

$$S_{\max} = V_{\max}^2 R\alpha / A_{\max} \tag{34}$$

Similarly, if L of the next node is greater than S_{max}, then $V_{lim\_next}=V_{max}$, else V_{lim_next} is calculated by Eq. (33)(b); If V_{lim_next} is greater than V_{lim_corner}, set $YesNo_{V\_next}$=NO; Otherwise $YesNo_{V\_next}$=YES.

At last, calculate V_{min} by

$$V_{\min} = \min\left\{V_{\lim\_corner}, V_{\lim\_previous}, V_{\lim\_next}\right\} \tag{35}$$

An exceptional case is if the next point simply extends the current path further along a line and there is no deviation, the two consecutive line segments should be reassembled to a single line. To make this feature into realization, the second node is deleted from the list in such an exceptional case. The work left to do is just to recalculate L of the head node and all above calculations of this step become unnecessary.

4. Try to increase V_{min} of all the nodes in the rest of the list by back tracking. If the property $YesNo_{V\_next}$ of next node is YES, then recalculate its V_{lim_next} and determine V_{min} by Eq. (35) again. If V_{min} is bigger than before, jump to next node and go on doing that. If the recalculated V_{min} is not bigger than before or tail node is arrived, skip back to the node after previous node along the old path, as illustrated in Fig. 3(f). If the property $YesNo_{V\_previous}$ of previous node is YES, then recalculate its $V_{lim\_previous}$ and determine V_{min} again. If recalculated V_{min} is bigger than before, go back to previous node and go on doing that until head node is arrived, else skip back to the node after previous node and go on doing that until head node is arrived. By now, back tracking is finished. Figure 3(b)–(e) illustrate how back tracking optimizes feed rate gradually. Note that each head node, including the original head node V_0, does not need back tracking for its speed is always zero.
5. Since the maximum number of look-ahead line segments is N, the list needs at most N+1 nodes. Each time when back tracking is finished, the V_{min} of the tail node of the list has been determined. Output this speed and delete this node. If there are remaining line segments to be read, go to step 2. Otherwise, V_{min} of each node in the list is output from tail to head one by one, as illustrated in Fig. 3(g).

Obviously, this well-designed look-ahead algorithm cost less computation time than the traditional algorithm previously mentioned for unnecessary and redundant calculation is greatly reduced. If select $\alpha = 0.5$ in Eq. (33) and use square value V_i^2 for comparison, the times of square root calculations will be decreased and computation time can be saved further.

In addition, one or two or even 20 look-ahead line segments is not enough in most applications. The number of look-ahead line segments needed varies according to contours, feed rates, and the performance of machine tools. In general, it cannot be limited to any arbitrary value, because conditions are constantly changing. On the other hand, it is not necessary to set a very large number of look-ahead line segments for there is no obvious effect on the productivity and more computation time and more memory are needed instead as it becomes too large [6]. Ideally, the number of look-ahead line segments should be dynamic [4]. Because the proposed look-ahead algorithm is based on a doubly linked list, the number of look-ahead line segments must not always be N but greater or less than N dynamically. If CNC processor is busy or a long linear contour is being machined, it looks ahead fewer line segments; otherwise it looks ahead more line segments. As a result, the potential of the CNC processor can be fully used.

4 Velocity profile generation

4.1 Calculation of actual maximum speed Vm

Since moving distance may be long enough for the CNC to accelerate to the maximum speed V_{max} or too short to reach V_{max}, in order to determine the type of velocity profile, actual maximum speed V_m must be calculated first. Assume V_{i-1}, V_i are the optimal starting point speed and end point speed to cover given distance L_i. S_1, S_2, S_3 are distances in the acceleration phase, constant speed phase and deceleration phase, respectively, t_a, t_c, t_d are the time for each corresponding move phase. According to Fig. 4(a),

$$\begin{aligned} S_1 &= \int_0^{t_a} (V_m - V_{i-1}) f_a(u) du + V_{i-1} t_a \\ &= (V_m - V_{i-1})\alpha t_a + V_{i-1} t_a \end{aligned} \tag{36}$$

$$\begin{aligned} S_3 &= \int_{t_a}^{t_a+t_d} (V_m - V_i) f_d(u) du + V_i t_d \\ &= (V_m - V_i)\alpha t_d + V_i t_d \end{aligned} \tag{37}$$

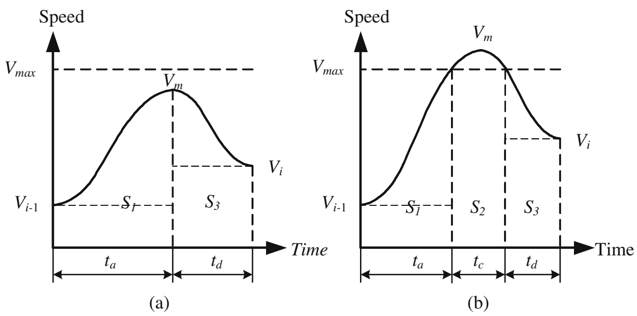


Fig. 4 Calculation V_m for determination of velocity profile type

$$L_i = S_1 + S_3 = (V_m - V_{i-1})\alpha t_a + V_{i-1}t_a + (V_m - V_i)\alpha t_d + V_i t_d \tag{38}$$

With reference to Eq. (25), one can get

$$t_a = \frac{(V_m - V_{i-1})R}{A_{\max}} \tag{39}$$

$$t_d = \frac{(V_m - V_i)R}{A_{\max}} \tag{40}$$

Combining Eqs. (38)–(40), results in

$$V_m = \frac{-F + \sqrt{F^2 + 8R\alpha\left[L_i A_{\max} + \left(V_{i-1}^2 + V_i^2\right)R(1-\alpha)\right]}}{4R\alpha} \tag{41}$$

where $F = (V_{i-1} + V_i)(1 - 2\alpha)R$

4.2 Determination of velocity profile type

Thus, the velocity profile can be determined as follows.

1. If $V_m \le V_{max}$, there is not a constant speed phase in velocity profile, i.e., t_c=0. Further, if $V_m > V_{i-1}$ and $V_m > V_i$, there are both acceleration phase and deceleration phase, the corresponding velocity profile type is named ACC_DEC; If $V_m = V_{i-1}$, there is only deceleration phase, t_a=0, t_d is determined by Eq. (40), $S_3 = L_i$, the corresponding velocity profile type is named DEC_ONLY; If $V_m = V_i$, there is only acceleration phase, t_d=0, t_a is determined by Eq. (39), $S_1 = L_i$, the profile type is named ACC_ONLY.
2. If $V_m > V_{max}$, there is always constant speed phase in velocity profile. As shown in Fig. 4(b), both time of acceleration and deceleration must be recalculated. Further, if $V_{i-1} = V_i = V_{max}$, there is only constant speed phase, $t_a = t_d = 0$, t_c is determined by Eq. (42), $S_2 = L_i$, the profile type is named CONST_ONLY; Else if $V_{i-1} = V_{max}$, there are constant speed phase and following deceleration phase, the profile type is named CONST_DEC, t_a=0, t_d, S_3 and t_c are determined by Eqs. (43)–(45), respectively.

$$t_c = L_i / V_i \tag{42}$$

$$t_d = (V_{\max} - V_i)R / A_{\max} \tag{43}$$

$$S_3 = (V_{\max} - V_i)\alpha t_d + V_i t_d \tag{44}$$

$$t_c = (L_i - S_3)/V_{\max} \tag{45}$$

Else if $V_i = V_{max}$, there are acceleration phase and following constant speed phase, the profile type is named ACC_CONST, t_d=0, t_a, S_1 and t_c are determined by Eqs. (46)–(48), respectively.

$$t_a = (V_{\max} - V_{i-1})R / A_{\max} \tag{46}$$

$$S_1 = (V_{\max} - V_{i-1})\alpha t_a + V_{i-1} t_a \tag{47}$$

$$t_c = (L_i - S_1)/V_{\max} \tag{48}$$

Otherwise, the velocity profile must begin with acceleration phase and then turn to constant speed phase and end with deceleration phase, the profile type is named ACC_CONST_DEC. t_a is determined by Eq. (46), t_d is determined by Eq. (43), t_c is determined by Eqs. (44), (47) and the following

$$t_c = S_2 / V_{\max} = (L_i - S_1 - S_3)/V_{\max} \tag{49}$$

4.3 Strategy for reducing feed rates fluctuation

The maximum allowable acceleration is always used in the previous process. Though such a method is efficient for machining, it will cause high jerk and large feed rates fluctuation when it deals with small line segments. To overcome this drawback, the previously determined velocity profile type should be changed by the following rules.

1. At first, calculate the actual maximum jerk J_{act}. Defining

$$\max_{0\le u\le 1}\left|\frac{d^2}{du^2}f_a(u)\right| = Q_a \quad \max_{0\le u\le 1}\left|\frac{d^2}{du^2}f_d(u)\right| = Q_d \tag{50}$$

Since symmetrical acceleration and deceleration characteristics are used, $Q_a = Q_d$. Defining

$$Q_a = Q_d = Q \tag{51}$$

During an acceleration phase, actual maximum jerk can be calculated by Eq. (52) where $V_{most}=\min\{V_m, V_{max}\}$.

$$\begin{aligned} J_{act} &= \max_{0\le t\le t_a}\left|\frac{d^2}{dt^2}(V_{most}-V_{i-1})f_a\left(\frac{t}{t_a}\right)\right| \\ &= \frac{V_{most}-V_{i-1}}{t_a^2}\max_{0\le u\le 1}\left|\frac{d^2}{du^2}f_a(u)\right| = (V_{most}-V_{i-1})Q/t_a^2 \end{aligned} \tag{52}$$

Similarly, during a deceleration phase, actual maximum jerk can be calculated by Eq. (53)

$$J_{act} = (V_{most}-V_i)Q/t_d^2 \tag{53}$$

Then, let J_{max} represent the maximum allowable jerk, if $J_{act} > J_{max}$, the previously determined profile type should be changed as the following.

2. If previously determined type is ACC_DEC and $V_{i-1} < V_i$, it is changed to ACC_ONLY type, t_a is determined by Eq. (54), which can be rewritten as Eq. (55), $t_c=t_d=0$. After this change, the total time spent in covering the distance of the line segment becomes longer than before, as illustrated in Fig. 5, but the maximum acceleration and jerk are greatly reduced as t_a increases, thus the velocity profile gets smoother than before. This example implies the actual maximum acceleration A_{act} can be less than A_{max} to obtain a smooth velocity profile.

$$\begin{aligned} L_i &= \int_0^{t_a}(V_i - V_{i-1})f_a(u)du + V_{i-1}t_a \\ &= (V_i - V_{i-1})\alpha t_a + V_{i-1}t_a \end{aligned} \tag{54}$$

$$t_a = L_i/[(V_i - V_{i-1})\alpha + V_{i-1}] \tag{55}$$

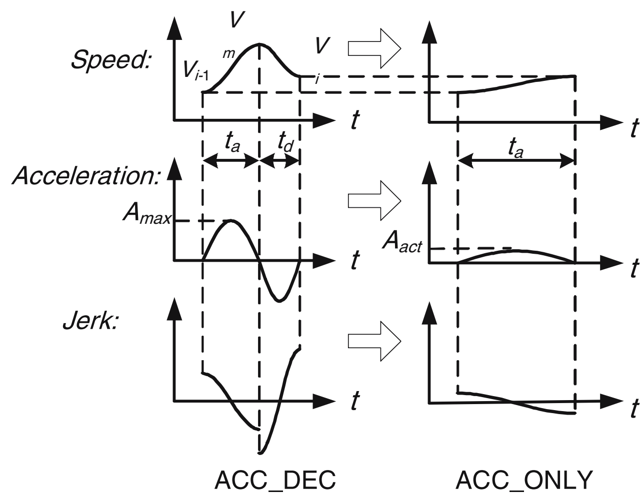


Fig. 5 An example of changing velocity profile type

Similarly, if the previously determined type is ACC_DEC and $V_{i-1} > V_i$, it is changed to DEC_ONLY type, t_d is determined by Eq. (56), $t_a=t_c=0$; otherwise, i.e., $V_{i-1}=V_i$, it is changed to CONST_ONLY type, t_c is determined by Eq. (27), $t_a=t_d=0$.

$$t_d = L_i/[(V_{i-1} - V_i)\alpha + V_i] \tag{56}$$

3. If the previously determined type is ACC_CONST, it is changed to ACC_ONLY type, t_a is determined by Eq. (55), $t_c=t_d=0$; If previously determined type is CONST_DEC, it is changed to DEC_ONLY type, t_d is determined by Eq. (56), $t_a=t_c=0$.
4. If previously determined type is ACC_CONST_DEC, it is changed to ACC_CONST type if $V_{i-1}<V_i$, t_a is determined by Eq. (57), t_c is determined by Eq. (58), $t_d=0$; if $V_{i-1}>V_i$, it is changed to CONST_DEC type, $t_a=0$, t_d is determined by Eq. (59), t_c is determined by Eq. (60); otherwise, it is changed to CONST_ONLY type, t_c is determined by Eq. (42), $t_a=t_d=0$.

$$t_a = (V_i - V_{i-1})R/A_{\max} \tag{57}$$

$$\begin{aligned} S_1 &= (V_i - V_{i-1})\alpha t_a + V_{i-1}t_a \\ t_c &= (L_i - S_1)/V_{i-1} \end{aligned} \tag{58}$$

$$t_d = (V_{i-1} - V_i)R/A_{\max} \tag{59}$$

$$\begin{aligned} S_3 &= (V_{i-1} - V_i)\alpha t_d + V_i t_d \\ t_c &= (L_i - S_3)/V_i \end{aligned} \tag{60}$$

Table 1 Speed increment and constant speed in different types of velocity profile

Profile type	ACC_DEC	DEC_ONLY	CONST_ONLY	AEC_ONLY	CONST_DEC	ACC_CONST	ACC_CONST_DEC
ΔV_a	$V_m - V_{i-1}$	/	/	$V_i - V_{i-1}$	/	$V_i - V_{i-1}$	$V_{max} - V_{i-1}$
V_c	/	/	V_i	/	V_{i-1}	V_i	V_{max}
ΔV_d	$V_m - V_i$	$V_{i-1} - V_i$	/	/	$V_{i-1} - V_i$	/	$V_{max} - V_i$

4.4 Interpolation and error compensation strategy

By use of the determined t_a, t_c, t_d and Eqs. (10), (12), the position increment during each sampling period of corresponding move phase can be determined by

$$n_a = \lceil t_a/T_s \rceil; n_c = \lceil t_c/T_s \rceil; n_d = \lceil t_d/T_s \rceil \tag{61}$$

$$\begin{aligned}
&\text{If } n_a \geq 1, {}^aP(k) = [{}^a\gamma(k)(\Delta V_a)n_a + V_{i-1}]T_s(k=1,2,\ldots,n_a)\\
&\text{If } n_c \geq 1, {}^cP(k) = V_cT_s \qquad (k=1,2,\ldots,n_c)\\
&\text{If } n_d \geq 1, {}^dP(k) = \left[{}^d\gamma(k)(\Delta V_d)n_d + V_i\right]T_s(k=1,2,\ldots,n_d)
\end{aligned} \tag{62}$$

where n_a, n_c and n_d are the numbers of sampling periods of acceleration phase, constant speed phase and deceleration phase, respectively. ${}^a\gamma(k)$ and ${}^d\gamma(k)$ are defined by Eqs. (11) and (13), respectively. ΔV_a and ΔV_d denote the speed increment during acceleration phase and deceleration phase, respectively. V_c denotes constant speed. Their values are listed in Table 1, in which "/" means the related value is not used in the type of velocity profile.

However, there are rounding errors, integral calculation errors, accumulation errors in the computational results of interpolation because n_a, n_c, n_d must be integers and the word size of computer processor is limited. These errors may result in move error. In the movement of the ith line segment, the move error can be defined as the difference between the desired moving distance and the sum of the position increment during each sampling period of corresponding move phase, that is

$$E_{move} = \left| L_i - \left[\sum_{k=1}^{n_a} {}^aP(k) + \sum_{k=1}^{n_c} {}^cP(k) + \sum_{k=1}^{n_d} {}^dP(k) \right] \right| \tag{63}$$

In order to improve the accuracy of interpolation, a novel strategy for error compensation is developed in this paper. At first, define

$$E_a = S_1 - \sum_{k=1}^{n_a} {}^aP_k; E_c = S_2 - \sum_{k=1}^{n_c} {}^cP_k; E_d = S_3 - \sum_{k=1}^{n_d} {}^dP_k \tag{64}$$

where S_1, S_2 and S_3 have been determined in previous Sections 4.1, 4.2, and 4.3. (If t_a=0, then S_1=0; If t_c=0, then S_2=0; If t_d=0, then S_3=0.) They always satisfy $S_1+S_2+S_3=L_i$. For an acceleration phase that has n_a sampling periods, E_a is divided into n_a sections according to the following form.

$$\underbrace{0\ d\ 2d\ \ldots\ (N-1)d}_{N}\ \underbrace{(N-1)d\ \ldots 2d\ d\ 0}_{N}\ \underbrace{(0)}_{n_a-2N} \tag{65}$$

$$N = \lfloor n_a/2 \rfloor \tag{66}$$

There are two arithmetic progressions in the above sequence of numbers. Both the first term of the first arithmetic progression and the last term of the second arithmetic progression are zero in order to avoid abrupt change in speed after error compensation. If n_a is an odd number, n_a-2N=1, a zero is appended to the tail of Eq. (65). Otherwise n_a-2N=0, it means no zero is appended. Because E_a is equal to the sum of the two arithmetic progressions, the common difference d can be obtained by:

$$d = E_a/[N(N-1)] \tag{67}$$

Thus, every section of progression (65) can be determined. The position increment during each sampling period can be modified by:

$$\begin{cases}
{}^aP'_k = {}^aP_k + (k-1)d & for\ 1 \leq k \leq N\\
{}^aP'_k = {}^aP_k + (2N-k)d & for\ N+1 \leq k < n_a\\
{}^aP'_k = {}^aP_k & for\ k = n_a
\end{cases} \tag{68}$$

Table 2 Coordinates of line segments to be machined (mm)

No.	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	L11	L12
X	3.020	6.000	18.56	29.60	41.80	42.90	41.80	40.20	38.00	35.50	33.20	23.30
Y	5.010	10.02	30.92	49.30	69.70	51.50	50.00	47.50	43.50	38.10	32.10	0.00

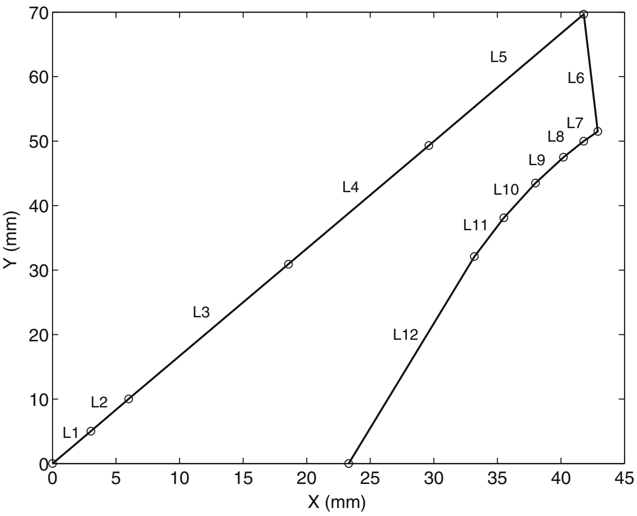


Fig. 6 A 2D curve used in testing the developed look-ahead approach

The same error compensation method can also be applied in constant phase and deceleration phase so that ${}^{c}P_k^{'}$ and ${}^{d}P_k^{'}$ are calculated. After such compensation, the move error of the ith line segment becomes

$$E_{move} = \left| L_i - \left[\sum_{k=1}^{n_a} {}^{a}P^{'}(k) + \sum_{k=1}^{n_c} {}^{c}P^{'}(k) + \sum_{k=1}^{n_d} {}^{d}P^{'}(k) \right] \right| \tag{69}$$

The resultant velocities during each sampling period of corresponding move phase are

$${}^{a}V^{'}(k) = {}^{a}P^{'}(k)/T_s;\ {}^{c}V^{'}(k) = {}^{c}P^{'}(k)/T_s;\ {}^{d}V^{'}(k) = {}^{d}P^{'}(k)/T_s \tag{70}$$

In a 3D CNC machine tool that has x, y and z axes, the resultant velocities are composed of three component velocities, so the final output speed of each axis during each sampling period can be calculated by

$$\begin{aligned} V_x(k) &= (x_i - x_{i-1})V^{'}(k)/L_i \\ V_y(k) &= (y_i - y_{i-1})V^{'}(k)/L_i \\ V_z(k) &= (z_i - z_{i-1})V^{'}(k)/L_i \end{aligned} \tag{71}$$

where $V(k)$ represents ${}^{a}V'(k)$, ${}^{c}V'(k)$ or ${}^{d}V'(k)$. (x_i, y_i, z_i) and $(x_{i-1}, y_{i-1}, z_{i-1})$ are the coordinates of starting point and end point of the ith line segment, respectively.

5 Simulations and experiments

Several simulations and experiments applying the proposed approach were made on a 3D CNC milling machine, in which the A_{max}=0.10 m/s^2, V_{max}=0.06 m/s, T_s=0.01s.

Table 2 shows the absolute coordinates of each end point of 12 small line segments depicting a piece of 2D curve

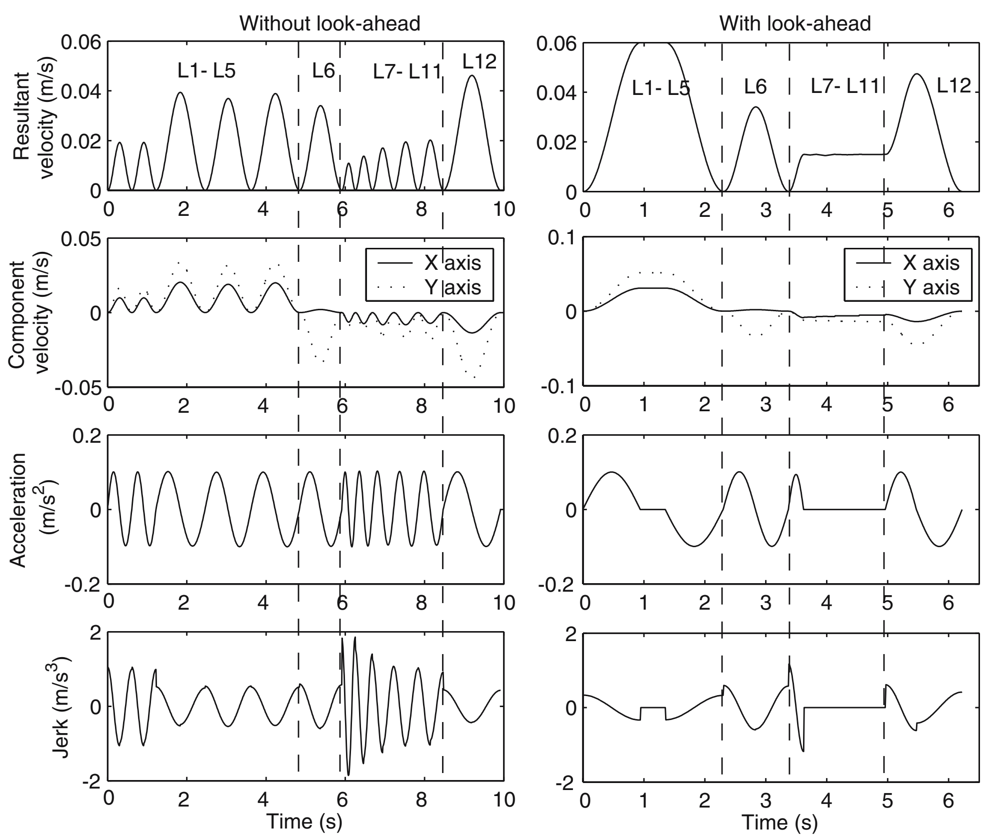


Fig. 7 Velocity profiles of cosine acc/dec characteristics without/with look-ahead

Table 3 The optimal end point speed (m/s) and improved move errors (m)

Line No.	L1–5	L6	L7	L8	L9	L10	L11	L12
Optimal end speed (m/s)	0.0000	0.0000	0.01539	0.01505	0.01440	0.01481	0.01495	0.0000
Desired distance (m)	0.08127	0.01823	0.001860	0.002968	0.004565	0.005951	0.006426	0.03359
Move error (m) (without compensation)	2.68E-5	5.78E-6	1.34E-5	7.55E-5	5.48E-6	4.62E-5	2.7E-5	1.05E-4
Move error (m) (by compensation)	2.74E-8	1.98E-9	3.33E-13	5.97E-10	1.02E-10	1.19E-9	1.89E-9	1.58E-8

which is deliberately chosen to test the proposed look-ahead approach. As shown in Fig. 6, the first five line segments are in a straight line but there are two sharp corners from the fifth line segment to the seventh line segment. Next to the sharp corners, there is an arc that is composed of five line segments. The last line segment is a tangent line of the arc.

In the first comparative simulation, cosine acceleration and deceleration characteristics specified by Eq. (4) are used. The simulated velocity profiles are shown in Fig. 7, from which it can be seen that if the look-ahead approach is not applied, acceleration and deceleration phases are always present in each line segment and the machine tool never reaches the maximum allowable speed. While the proposed look-ahead approach is adopted, the velocity profile takes on a new look that the number of acceleration and deceleration phases decreases dramatically because the first five line segments have been reassembled to a single line and the speed is kept almost constant from the seventh line segment to the 11th line segment. In addition, as shown in the first row of Table 3, the optimal end point speed of the fifth (or sixth) line segment is zero. The reason is that y-axis (or x-axis) experiences a direction change at the turning point, which can be seen from Fig. 6 and the component velocity profile of Fig. 7. In other turning points where no sharp corners are present, the optimal speed is a reasonable value rather than zero. As a result, the total time spent on the curve is shortened from 9.9 to 6.2s and the velocity profile is smoother than before.

Table 3 also shows the move errors defined by Eqs. (63) and (69). It is clear that move errors have been greatly improved, which proves the effectiveness of the proposed error compensation strategy.

In another simulation, linear acceleration and deceleration characteristics specified by Eq. (3) are used. As shown in Fig. 8, the total time spent on the same curve is shortened

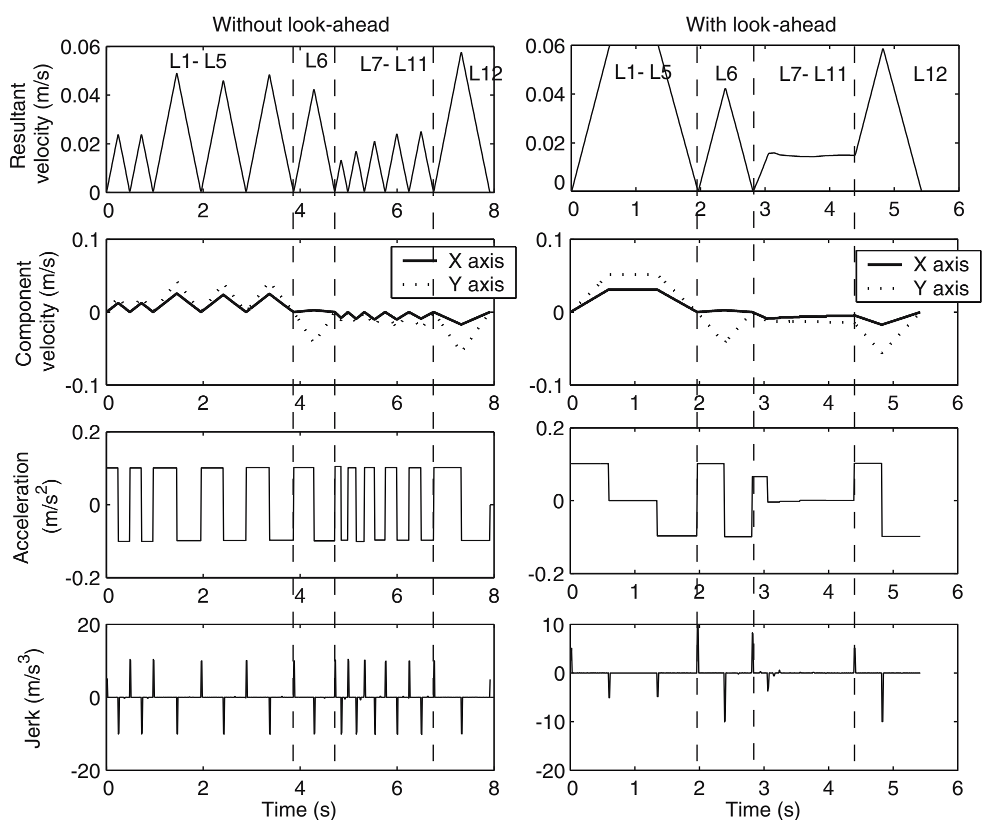


Fig. 8 Velocity profiles of linear acc/dec characteristics without/with look-ahead

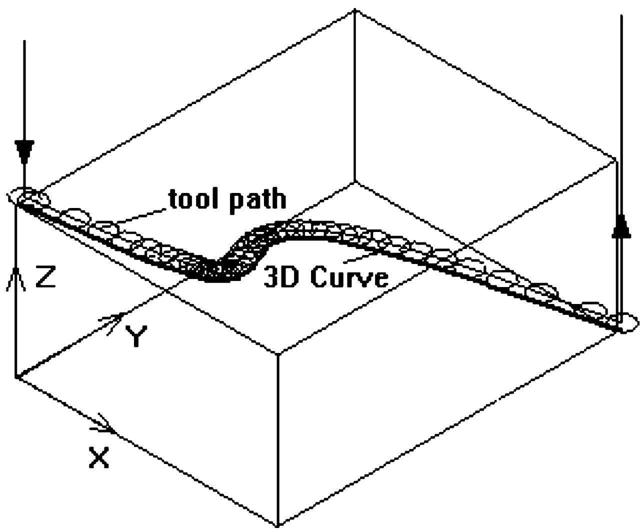


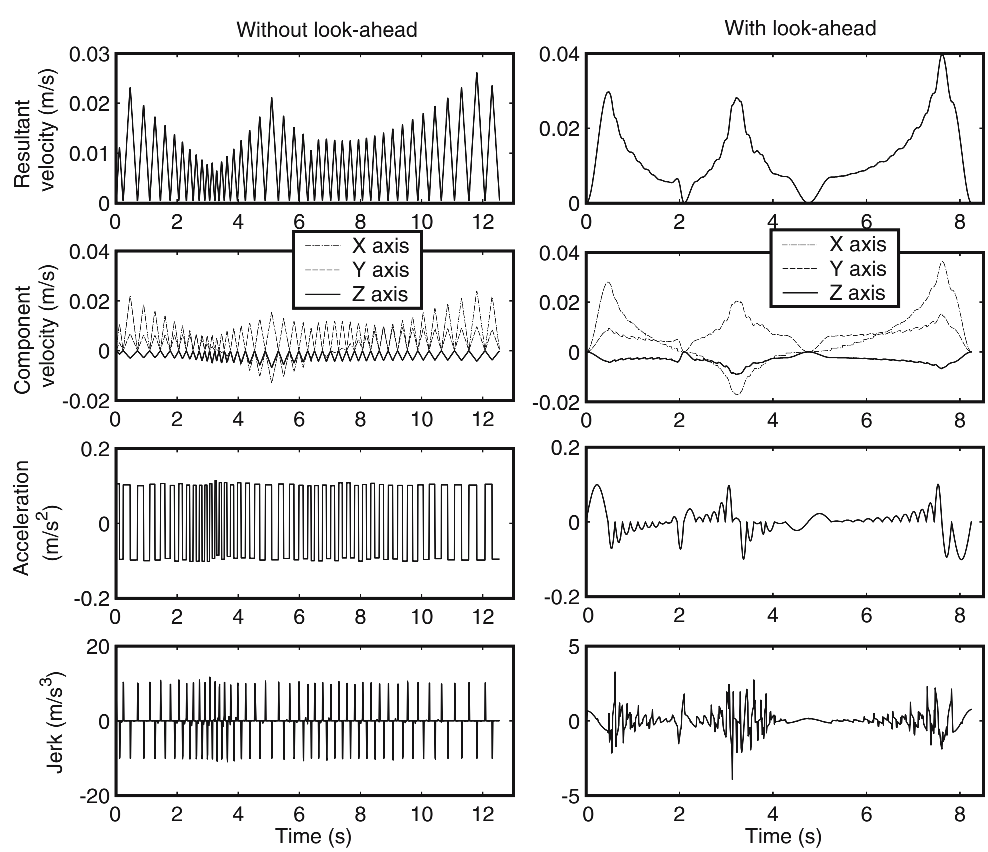
Fig. 9 A 3D curve used in testing the developed look-ahead approach

from 7.9 to 5.4s and the number of acceleration and deceleration phases is reduced, which also verifies the effectiveness of the proposed look-ahead approach. However, there is still a large jerk in the velocity profile for the intrinsic drawback of linear acceleration and deceleration characteristics. By contrast with Fig. 7, the finish time in Fig. 6 is a bit latish, but the velocity profile is much smoother than that in Fig. 7. The reason is that the acceleration profile is continuous and abrupt changes of acceleration can be avoided when cosine acceleration and deceleration characteristics are used.

An open architecture CNC system with a Pentium III 1Ghz CPU is used in the experiments of milling actual parts. The experimental results showed the machining time was dramatically shortened when the proposed approach is applied. Moreover, the feed rates are smoother than those without look-ahead approach and the final contour accuracy of the milled parts is satisfying.

Figure 9 illustrates a randomly selected 3D contour of one part used in experiment. The 3D curve was designed by a CAM software named MasterCAM, whose post process program broke the curve into 39 consecutive line segments in terms of confined chord tolerance. In the comparative experiment, traditional CNC interpolator and the newly designed CNC interpolator applying the proposed look-ahead approach processed these line segments, respectively. As shown in Fig. 10, it cost only 12.1 s to finish all the line segments for the former. While for the latter, 8.1 s instead. What's more, being benefit from cosine acceleration and deceleration characteristics and strategy for reducing feed rates fluctuation, the acceleration changed in a relaxed

Fig. 10 Velocity profiles in machining a 3D contour without/with look-ahead



manner and the magnitudes of jerks became smaller. In brief, the velocity profile with look-ahead was smoother than that without look-ahead.

6 Conclusions

A universal velocity profile generation approach with the feature of look-ahead for improving the machining of small line segments is proposed in this paper. The proposed approach makes it practical to generate desired velocity profiles of smooth acceleration and deceleration characteristics and the highest possible move speed. The proposed look-ahead algorithm is capable of making full use of the potential of CNC processors for it is based on a dynamic doubly linked list structure and timesaving back tracking. Two novel strategies, one for reducing feed rates fluctuation and the other for reducing move errors that come from digital integration and accumulated computation, are also proposed. The former smoothes out the generated velocity profile and the latter ensures required machining accuracy while increasing productivity. Both computer simulations and the experiments conducted on milling machine showed the proposed approach was feasible and effective. Though the approach is designed for CNC machine tools, it can also be applied in fast-motion robotics or other position control systems.

Acknowledgements The authors acknowledge the financial supports from National Natural Science Foundation of China (No. 50245011) and the Major State Basic Research Development Program of China (973 Program) (No. 2003CB716206). The authors also wish to thank Shanghai Nanpre Mechanics Co., Ltd. who provided valuable suggestions and experimental assistance.

References

1. Liu YJ, Sun LN, An ZW (2005) An approach for generating high velocity and high acceleration trajectories of industrial robots. Proc of 2005 IEEE Int Symp Computational Intelligence in Robotics and Automation, Espoo, Finland, pp 199–204
2. Jeon JW, Ha YY (2000) A generalized approach for the acceleration and deceleration of industrial robots and CNC machine tools. IEEE Trans Ind Electron 47(1):133–139
3. Jeon JW (2000) Efficient acceleration and deceleration technique for short distance movement in industrial robots and CNC machine tools. Electron Lett 36(8):766–768
4. Schuett T (2003) A closer look at look-ahead. Modern machine shop. http://www.mmsonline.com/articals/039603.html
5. CAO WG, Chang QX (2005) A kind of arithmetic having the function of "look ahead" in smoothly controlling. Modular Mach Tools Auto Mach Tech 5(9):56–59
6. Hu J, Xiao LJ, W YH, Wu ZY (2006) An optimal feedrate model and solution algorithm for a high–speed machine of small line blocks with look-ahead. Int J Adv Manuf Technol 28(9):930–935
7. Tsai MC, Cheng MY, Lin KF, Tsai NC (2005) On acceleration / deceleration before interpolation for CNC motion control. Proc of IEEE Int Conf Mechatronics, Taipei, TW, pp 382–387