Discrete Interpolation of G01 Codes in 2D Machining under Bounded Accelerations

Hongbo Li, Xiaoshan Gao, Lixian Zhang and Ruiyong Sun

Abstract. In this paper, a new algorithm is proposed to interpolate G01 codes in 2D milling. Without resorting to smooth curve approximation, the algorithm makes full use of the acceleration bounds to change directions with optimal velocities at the cutter location points designated by the G01 codes, and then makes global lookahead and optimization to generate time-optimal interpolation of the G01 codes. The algorithm is realized and tested on NC wood carving machine, showing that when compared with the classical equal-velocity corner-turning method, the current method can reduce the machining time by 14%-115%, while maintaining better details than the classical method.

Mathematics Subject Classification (2000). Primary 93C55; Secondary 49K30.

Keywords. G01 code interpolation; NC milling; Time-optimal control; Microlinear interpolation.

1. Introduction

G01 code interpolation is a basic task in NC machining [1], [2], [4], [8], [9], [12]. A series of G01 codes for 2D Cartesian machining gives a piecewise linear trajectory in the plane composed of edges and vertices. Under bounded accelerations in the x-axis and y-axis, for fixed interpolation period, feedrate bound and shape error tolerance, for a series of G01 codes with zero boundary velocity constraints, a *time-optimal interpolation* algorithm outputs a new series of G01 codes whose piecewise linear trajectory is within the shape error tolerance with the input trajectory, such that on the new trajectory, the motion on each new edge takes one interpolation period, the motion along the whole trajectory obey the acceleration, feedrate and boundary velocity constraints, and the whole motion time is minimal among all motions satisfying above constraints.

This work was partially supported by Program 2011CB302404, NSFC 10925105, 60821002/F02.

There are three kinds of algorithms generating time-optimal interpolations of G01 codes. The first kind is by smoothing the trajectory globally, making time-optimal interpolation on the smooth trajectory, and then sampling along the smooth trajectory for each interpolation period [6], [11], [14]. The second kind is by smoothing the trajectory locally, *i.e.*, at each vertex of the input piecewise linear trajectory, constructing a smooth curve blending the two edges of the vertex, then making time-optimal interpolation and sampling on the smooth trajectory [3], [16], [17].

The above two kinds of algorithms cause two shape errors, one from the approximation by smooth curve, and the other from sampling along the smooth curve to generate new G01 codes. The third kind of algorithm is by *discrete interpolation* and *direct corner turning* [5], [10], [15]. By discrete interpolation we mean generating new G01 codes without resorting to any smooth approximation of the input trajectory. By direct corner turning we mean at each vertex of the input piecewise linear trajectory, the motion from one edge to the other is within one interpolation period. As a consequence, the new G01 codes have all their vertices on the input trajectory. This kind of interpolation has the advantage of small shape error in that the new vertices approximate the old ones along the edges of the input trajectory.

At each vertex of a piecewise linear trajectory, the displacement $\mathbf{s}_a$ along one edge to the vertex in one interpolation period T gives the incoming velocity $\mathbf{s}_a/T$, and the displacement $\mathbf{s}_b$ from the vertex along the other edge in one interpolation period gives the outgoing velocity $\mathbf{s}_b/T$. The difference between the two velocities gives the acceleration of the corner turning: $\mathbf{a} = (\mathbf{s}_b - \mathbf{s}_a)/T^2$. Conversely, an acceleration vector at the vertex gives the incoming velocity and outgoing velocity by vector decomposition into the two edges. Obviously the two velocities need not be zero. The goal is to maximize them under given acceleration, feedrate and boundary velocity constraints.

The following *equal-velocity method* is classical in direct corner turning: at each vertex, the incoming velocity always equals the outgoing velocity as scalars [13]. Then the acceleration at each vertex is caused only by the change of direction of the motion, so it always follows the bisector direction of the angle formed by the two edges at the vertex.

It turns out that this choice of acceleration at corner turning does not maximize the allowable incoming velocity and outgoing velocity. This is the starting point of our research on discrete interpolation. We find out that optimizing the sum of incoming velocity and outgoing velocity is a problem of linear programing for linear object function, no matter if the constraints upon the dynamics of the two axes of the machine is acceleration, jerk, jounce and the like.

In this paper, we propose to adopt different velocities at different edges of a vertex. The purpose is to maximize the sum of incoming velocity and outgoing velocity for the motion to change direction from one edge to another in one interpolation period. This is our *different-velocity corner-turning method.*

To develop a time-optimal interpolation algorithm from the different-velocity method, the following problems need to be solved: (1) how to construct the corner-turning acceleration vector at each vertex under given acceleration, feedrate and boundary velocity constraints; (2) if the maximal allowable velocity at one edge of a vertex cannot be reached by the motion from the other end of the edge, how to make velocity-optimal corner turning under the same dynamic constraints?

In this paper, we first investigate the basic problem of time-optimal discrete interpolation within a line segment under bounded accelerations, feedrate and fixed endpoint velocities. The difficulty lies in minimizing the motion time which is an integer variable. This is the content of Section 2 and Section 3. We then consider velocity-optimal corner turning under given acceleration bounds. This is the content of Section 4. These are local optimal considerations of the interpolation problem. In particular, the velocity-optimal corner turnings provide upper bounds for each of the velocities at the ends of the edges in the input trajectory.

For a whole piecewise linear trajectory with zero boundary velocity constraints, in order for the motion starting from one end of the trajectory to reach the other end with zero terminal velocity, a *lookahead strategy* is necessary. This strategy updates the upper bounds for the velocities at the ends of the edges in the whole trajectory. With these endpoint velocity upper bounds, a forward chasing algorithm is designed to globally optimize the endpoint velocities (not their upper bounds!). Then an interpolation procedure is carried out to every edge and every corner of the trajectory, leading to a series of interpolated G01 codes that is time-optimal. These are the central part of the paper and is in Section 5.

The algorithms are implemented in C language and integrated with the Blue-sky NC system developed by Shenyang Blue-sky NC Co. Ltd. for wood carving machine. Real-time wood carving of different surface designs show that when compared with the equal-velocity corner turning method, the new different-velocity corner turning method can reduce the the milling time by 14%-115%, depending on the choice of acceleration and feedrate bounds. Meanwhile, the carved wood designs have much clearer details and fewer errors than by the equal-velocity corner turning method.

2. Continuous model and discrete model

Let there be a 2D cutter motion and an interpolation period T in NC machining. In this paper we usually set $T = 1$ for convenience, and call it the *sampling period.*

The *continuous model* of G01 codes refers to sampling the motion trajectory by assuming the time variable to be continuous. This model is currently used by many NC systems. For example, if a motion on a straight line follows a constant velocity v, then the position sequence is by sampling $x = vt + x_1$, and is

$$x_1, x_1 + v,\ x_1 + 2v, \ldots, x_1 + tv. \tag{2.1}$$

If a motion follows a constant acceleration a, then the velocity sequence and position sequence are by sampling $v = at + v_{x_1}$ and $x = at^2/2 + v_{x_1}t + x_1$, and for

$x_1 = 0$ and $v_{x_1} = 0$, are respectively

$$\begin{array}{l} 0,\ a,\ 2a, \ldots, ta, \\ 0, 0.5a,\ 2a,\ 4.5a, \ldots, 0.5t^2a. \end{array} \tag{2.2}$$

The *discrete model* of G01 codes refers to defining velocity and acceleration respectively by first-order and second-order differences of positions on the motion trajectory. In discrete geometry, the velocity vector $\mathbf{v}(t)$ and acceleration vector $\mathbf{a}(t)$ at instance t where t is an integer, are defined by the position vector $\mathbf{x}(t)$ at instance t by

$$\begin{array}{rcl} \mathbf{v}(t) & := & \mathbf{x}(t) - \mathbf{x}(t-1), \\ \mathbf{a}(t) & := & \mathbf{v}(t) - \mathbf{v}(t-1). \end{array} \tag{2.3}$$

The discrete model is also used frequently in CNC systems. In comparison, for a motion on a straight line with constant velocity, the motion plan based on the continuous model is identical to that based on the discrete model. However, for a motion of constant acceleration, the position sequence in the second row of (2.2) indicates that in the discrete sense, the motion no longer has constant acceleration, as by (2.3), the sequences of discrete velocity and acceleration are respectively

$$\begin{array}{l} 0,\ 0.5a,\ 1.5a,\ 2.5a, \ldots, ta - 0.5a, \\ 0,\ 0.5a,\ a,\ a, \ldots, a. \end{array} \tag{2.4}$$

The discrete velocity $v_d(t)$ in (2.4) is related to the sampling sequence of continuous velocity $v_c(t)$ in the first row of (2.2) by

$$v_d(t) = \frac{v_c(t) + v_c(t-1)}{2}. \tag{2.5}$$

Given a line segment $[x_1, x_2]$, if the motion from x_1 to x_2 has constant velocity, the motion time described by the discrete model is the same with that by the continuous model. If the motion has constant acceleration a and initial velocity v_{x_1}, then in the continuous model, the motion time t_c satisfies

$$x_2 - x_1 = t_c v_{x_1} + \frac{t_c^2}{2} a, \tag{2.6}$$

while in the discrete model, the motion time t_d satisfies

$$x_2 - x_1 = (v_{x_1} + a) + (v_{x_1} + 2a) + \cdots + (v_{x_1} + t_d a) = t_d v_{x_1} + \frac{t_d(t_d+1)}{2} a. \tag{2.7}$$

Hence $t_d < t_c$.

Let $t_d = t_c - \tau$. Substituting it into (2.7), we get

$$\frac{a}{2}(t_c^2 - 2\tau t_c + \tau^2) + (\frac{a}{2} + v_{x_1})(t_c - \tau) - (x_2 - x_1) = 0. \tag{2.8}$$

By (2.6),

$$\frac{t_c + \tau^2}{2} = \tau(\frac{1}{2} + \frac{v_{x_1}}{a} + t_c). \tag{2.9}$$

So

$$\tau = \frac{1}{2} + \frac{v_{x_1}}{a} + t_c - \sqrt{(\frac{1}{2} + \frac{v_{x_1}}{a} + t_c)^2 - t_c} \leq \frac{2t_c}{1 + 2t_c + \sqrt{1 + 4t_c^2}}. \tag{2.10}$$

If $t_c \geq 1$, then t_d and t_c differ by less than half the sampling period. Because of this, for very small sampling period and constant acceleration or velocity motion on a line segment, the difference between the two models are negligible.

3. Discrete interpolation on a line segment

We start with the simplest 1D case of linear interpolation. Given a line segment $[x_1, x_2]$ on the x-axis, let the maximal velocity and maximal acceleration on the axis be v_m, a_m respectively, and let the initial velocity at point x_1 and the terminating velocity at point x_2 be v_{x_1}, v_{x_2} respectively. The motion interpolation problem contains three aspects:

1. Under the distance, speed and acceleration limits on the line segment, determine whether or not the initial speed can be changed into the terminating speed at a point before the terminal x_2. This is the *reachability problem.*
2. If $v_{x_1} < v_{x_2}$ and no motion can reach v_{x_2} at x_2, determine the maximal reachable velocity at x_2.
3. If the terminating speed can be reached, design a *motion plan* which is composed of a *velocity sequence*

$$v(0), v(1), v(2), \ldots, v(t), \tag{3.1}$$

where
- $t > 0$ is an integer,
- $0 \leq v(i) \leq v_m$ is the speed at time i,
- $v(0) = v_{x_1}$, $v(t) = v_{x_2}$,
- the acceleration from $v(i)$ to $v(i+1)$ is within a_m,

and a *position sequence*

$$x(0), x(1), x(2), \ldots, x(t), \tag{3.2}$$

where
- $x(0) = x_1$,
- $x(i)$ is the position at time i,
- $x(i) < x(i+1) \leq x_2$,
- if $x(t) < x_2$, then any motion from $x(t)$ to x_2 with initial and terminal velocities both equal to v_{x_2} takes time more than $T = 1$.

A motion plan is said to be *time-optimal* if either (a) $x(t) = x_2$ and t is the smallest among all the motions plans satisfying $x(t) = x_2$, or (b) $x(t) < x_2$, no motion plan satisfies $x(t+1) = x_2$ and $v(t+1) = v_{x_2}$ simultaneously, and t is the smallest among all motion plans.

3.1. Reachability

Given initial velocity and terminal velocity v_{x_1}, v_{x_2}, if $v_{x_1} = v_{x_2}$, the reachability is trivial. Without loss of generality, assume $v_{x_1} < v_{x_2}$.

Starting from x_1 and accelerating by constant a, the motion at integer instance t arrives at position x with velocity v. In the continuous model,

$$\begin{array}{rcl} v & = & v_{x_1} + at, \\ x & = & x_1 + tv_{x_1} + t^2\dfrac{a}{2}. \end{array} \tag{3.3}$$

Define a real function $\lceil\ \rceil$ on $\mathbb{R}$ as follows:

$$\lceil x \rceil := \begin{cases} x, & \text{if } x \text{ is an integer;} \\ [x]+1, & \text{otherwise.} \end{cases} \tag{3.4}$$

Define another function $\lfloor\ \rfloor$ taking values in $[0,1)$ as follows:

$$\lfloor x \rfloor := \lceil x \rceil - x. \tag{3.5}$$

For example, $[0.3] = 0,\ \lceil 0.3 \rceil = 1,\ \lfloor 0.3 \rfloor = 0.7$.

Eliminating t from (3.3), we get

$$v^2 = v_{x_1}^2 + 2a(x - x_1). \tag{3.6}$$

When $v = v_{x_2}$, solving for t from the first equality in (3.3), we get

$$t = \lceil \frac{v_{x_2} - v_{x_1}}{a} \rceil. \tag{3.7}$$

It is the minimal number of sampling periods to accelerate from v_{x_1} to v_{x_2}.

When $(v_{x_2} - v_{x_1})/a$ is an integer, as long as

$$x_2 - x_1 \geq \frac{v_{x_2}^2 - v_{x_1}^2}{2a}, \tag{3.8}$$

the velocity v_{x_1} at x_1 can reach v_{x_2} at x_2.

When $(v_{x_2} - v_{x_1})/a$ is not an integer, the non-decelerating motion of minimal displacement with initial and terminal velocities v_{x_1}, v_{x_2} in $1 + [(v_{x_2} - v_{x_1})/a]$ sampling periods is first moving with constant velocity v_{x_1} in time $\lfloor (v_{x_2} - v_{x_1})/a \rfloor$, then moving with constant acceleration a in time $(v_{x_2} - v_{x_1})/a$. So as long as

$$x_2 - x_1 \geq \lfloor \frac{v_{x_2} - v_{x_1}}{a} \rfloor\, v_{x_1} + \frac{v_{x_2}^2 - v_{x_1}^2}{2a}, \tag{3.9}$$

the velocity v_{x_1} at x_1 can reach v_{x_2} at x_2. Obviously (3.9) contains (3.8) as a special case.

In the discrete model, (3.7) is still valid, while (3.6) is replaced by

$$x - x_1 = \frac{v^2 - v_{x_1}^2}{2a} + \frac{v - v_{x_1}}{2}. \tag{3.10}$$

When $(v_{x_2} - v_{x_1})/a$ is an integer, the reachability is equivalent to

$$x_2 - x_1 \geq \frac{v_{x_2}^2 - v_{x_1}^2}{2a} + \frac{v_{x_2} - v_{x_1}}{2}. \tag{3.11}$$

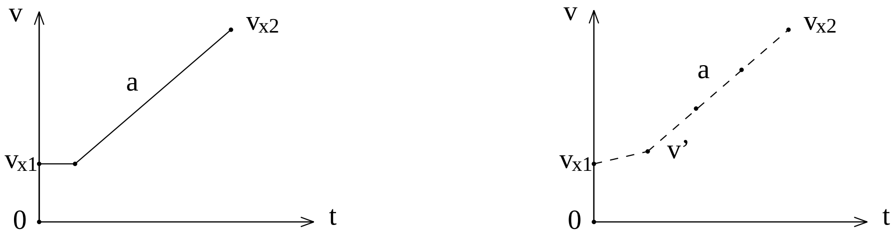

FIGURE 1. Reachability: continuous model (left), discrete model (right).

When $(v_{x_2}-v_{x_1})/a$ is not an integer, the velocity v_{x_2} takes time $[(v_{x_2}-v_{x_1})/a]$ to decrease with constant acceleration $-a$ to

$$v' = v_{x_2} - [\frac{v_{x_2}-v_{x_1}}{a}]a = v_{x_1} + (1 - \lfloor\frac{v_{x_2}-v_{x_1}}{a}\rfloor)a. \tag{3.12}$$

The motion of minimal displacement with initial and terminal velocities v_{x_1}, v_{x_2} in $1 + [(v_{x_2} - v_{x_1})/a]$ sampling periods is first accelerating v_{x_1} to v' in the first sampling period, then in $[(v_{x_2} - v_{x_1})/a]$ sampling periods accelerating v' to v_{x_2} with constant acceleration a. The reachability is equivalent to

$$\begin{aligned} x_2 - x_1 &\geq \frac{v_{x_2}^2 - v'^2}{2a} + \frac{v_{x_2} - v'}{2} + v' \\ &= \frac{v_{x_2}^2 - v_{x_1}^2}{2a} + \frac{v_{x_2} - v_{x_1}}{2} + \lfloor\frac{v_{x_2} - v_{x_1}}{a}\rfloor(v_{x_1} + \frac{a}{2}(1 - \lfloor\frac{v_{x_2} - v_{x_1}}{a}\rfloor)). \end{aligned} \tag{3.13}$$

Obviously, (3.13) contains (3.11) as a special case.

3.2. Maximal reachable terminal velocity

Consider the following problem: given an initial velocity v_{x_1} at x_1, for fixed $L = x_2 - x_1$, determine the maximal reachable velocity V at x_2, such that the time (integer) spent from (x_1, v_{x_1}) to (x_2, V) is minimal.

In the continuous model, the motion is described by (3.3), so

$$L = tv_{x_1} + t^2\frac{a}{2}, \tag{3.14}$$

where t is the motion time. When t is an integer, then by (3.6),

$$V^2 = v_{x_1}^2 + 2aL. \tag{3.15}$$

When t is not an integer, in our later application of linear interpolation, it is sufficient to take

$$V = \sqrt{v_{x_1}^2 + 2aL'}, \quad \text{where} \quad L' = [t]v_{x_1} + [t]^2\frac{a}{2}, \tag{3.16}$$

as the maximal reachable velocity.

Of course, (3.16) leaves room for improvement. When t is not an integer, the minimal motion time to arrive at x_2 is $[t] + 1$. The non-decelerating motion with maximal terminal velocity at x_2 is the following: first accelerate with some constant value $a' < a$ in some time $t' < t$, then accelerate with maximal value a in time $[t] + 1 - t'$. The t-x graph is similar to Figure 1 right. In the discrete model, computing integer variable t' symbolically is difficult.

Below we compute a', t' in the continuous model. At time t', let the velocity and traveled distance be v', L'. Then

$$\left\{\begin{array}{lll} v' & = & v_{x_1} + t'a', \\ L' & = & t'v_{x_1} + t'^2\dfrac{a'}{2}. \end{array}\right. \tag{3.17}$$

Similarly, at time $[t] + 1$,

$$\begin{array}{lll} V & = & v' + ([t] + 1 - t')a, \\ L - L' & = & ([t] + 1 - t')v' + ([t] + 1 - t')^2\dfrac{a}{2}. \end{array} \tag{3.18}$$

Solving for a', t' from (3.17) and (3.18), we get

$$\left\{\begin{array}{lll} a' & = & \dfrac{L - v_{x_1}([t] + 1) - \frac{a}{2}([t] + 1 - t')^2}{t'([t] + 1 - \frac{t'}{2})}, \\ V & = & v_{x_1} + a([t] + 1) + \dfrac{L - v_{x_1}([t] + 1) - \frac{a}{2}([t] + 1)^2}{[t] + 1 - \frac{t'}{2}}. \end{array}\right. \tag{3.19}$$

Since $L - v_{x_1}([t] + 1) - a([t] + 1)^2/2 < 0$, V becomes maximal if and only if t' is minimal. Although t' is bounded by $0 < t' < t$ and $0 \leq a' \leq a$. only $a' \geq 0$ is nontrivial for t'.

Parabola $f(t') = L - v_{x_1}([t]+1) - a(t' - [t] - 1)^2/2$ has the following properties: (1) $f(\infty) = -\infty$, (2) $f(t') < 0$ when $t' \leq 0$. As $a' \geq 0$ is equivalent to $f(t') \geq 0$, t' is bounded below by the left zero point of the parabola. So when

$$t' = [t] + 1 - \sqrt{\frac{2(L - v_{x_1}([t] + 1))}{a}}, \tag{3.20}$$

V becomes maximal.

3.3. Motion plan on a line segment

In this section we always assume the reachability from (x_1, v_{x_1}) to (x_2, v_{x_2}), but no longer assume $v_{x_1} < v_{x_2}$. If not considering the constraint that the motion time is an integer, then the time-optimal motion is the following [7]:

In the x-v plane, the motion to the right starting from (x_1, v_{x_1}) with constant acceleration a meets the motion to the left starting from (x_2, v_{x_2}) with constant acceleration a at a point (X, V). If $V \leq v_m$ then the whole motion is the concatenation of the first motion and the inverse of the second motion, else the two motions each can reach maximal velocity v_m, and the interval not passed by the two motions in $[x_1, x_2]$ is passed by the motion of constant velocity v_m.

Below we discretize the whole motion by sampling the time variable.

Continuous model:

(3.6) represents the x-v curve (hodograph) starting from (x_1, v_{x_1}). The hodograph starting from (x_2, v_{x_2}) is similar:

$$v^2 = v_{x_2}^2 - 2a(x - x_2). \tag{3.21}$$

Combining (3.6) and (3.21), we get the values of (X, V) as following:

$$\begin{aligned} X &= \frac{x_1 + x_2}{2} + \frac{v_{x_2}^2 - v_{x_1}^2}{4a}, \\ V^2 &= \frac{v_{x_1}^2 + v_{x_2}^2}{2} + a(x_2 - x_1). \end{aligned} \tag{3.22}$$

By (3.8), $V \geq \max(v_{x_1}, v_{x_2})$.

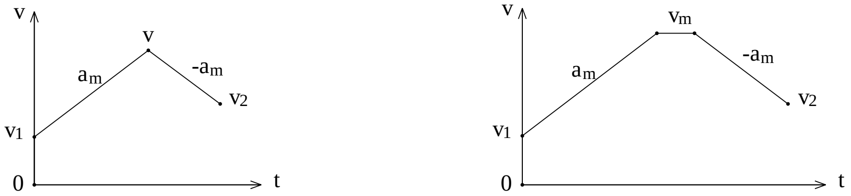

FIGURE 2. Time-optimal 1D motion in the continuous model. Left: v_m cannot be reached. Right: v_m can be reached.

If $V \leq v_m$, from (3.22) we get the time t_1, t_2 spent by starting from x_1, x_2 to arrive at velocity V:

$$t_1 = \frac{V - v_{x_1}}{a}, \qquad t_2 = \frac{V - v_{x_2}}{a}. \tag{3.23}$$

If $t_1 + t_2$ is an integer, the motion on $[x_1, x_2]$ is sampled as follows: for integer $0 \leq t \leq t_1 + t_2$,

$$x(t) = \begin{cases} x_1 + v_{x_1} t + \frac{a}{2} t^2, & \text{if} \quad t \leq t_1, \\ x_2 - v_{x_2}(t_1 + t_2 - t) - \frac{a}{2}(t_1 + t_2 - t)^2, & \text{if} \quad t > t_1. \end{cases} \tag{3.24}$$

If $t_1 + t_2$ is not an integer, then when $t = [t_1 + t_2]$, the motion (3.24) cannot reach speed v_{x_2}, while when $t = \lceil t_1 + t_2 \rceil$, the motion goes beyond x_2. If $v_{x_1} \geq v_{x_2}$, a simple method to solve this problem is to preserve an interval $[x', x_2]$ within $[x_1, x_2]$, where

$$x' = x_2 - v_{x_2} - \frac{a}{2}, \tag{3.25}$$

as long as (i) $x' \geq x_1$, (ii) (x', v_{x_2}) can be reached from (x_1, v_{x_1}). If $v_{x_1} < v_{x_2}$, the interval is replaced by $[x_1, x_1 + v_{x_1} + a/2]$. By symmetry, we consider only the case $v_{x_1} \geq v_{x_2}$.

If the above conditions (i), (ii) are satisfied, the motion on $[x_1, x']$ before sampling is planned according to the initial and terminal velocities v_{x_1}, v_{x_2}, so that if the arrival time at x' is not an integer, then with one of the accelerations $0, a$, the motion reaches speed v_{x_2} before arriving at x_2. If the two conditions are not both satisfied, the motion on $[x_1, x_2]$ before sampling is identical to the motion of minimal displacement in the reachability analysis.

After sampling, the motion from x_1 to a point $x_{12} < x_2$ reaches speed v_{x_2}. The motion of constant velocity from x_{12} to x_2 takes less than two sampling periods. Distance $x_2 - x_{12}$ is called the *residue* of velocity v_{x_2} at point x_2.

In the sampling sequence of positions from x_1 to x_{12}, if there exists three neighboring positions $x_{i_1} < x_{i_2} < x_{i_3}$ such that

$$\min(x_{i_3} - x_{i_2}, x_{i_2} - x_{i_1}) \leq x_2 - x_{12} \leq \max(x_{i_3} - x_{i_2}, x_{i_2} - x_{i_1}), \tag{3.26}$$

then the subsequence $x_{i_1}, x_{i_2}, x_{i_3}$ can be replaced by $x_{i_1}, x_{i_2}, x_{i_2} + (x_2 - x_{12}), x_{i_3} + (x_2 - x_{12})$, so that when arriving at position x_2, the residue of speed v_{x_2} becomes zero. This replacement is called *residue absorption*.

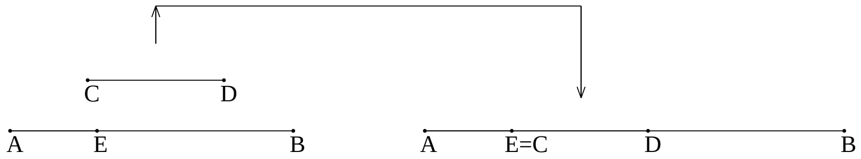

FIGURE 3. Assimilation of residue: displacement CD is assimilated between displacements AE and EB in the sequence of displacements of a motion plan.

If the residue at x_2 cannot be assimilated as above by the motion on $[x_1, x_{12}]$, then starting from x_{12}, the motion with constant velocity v_{x_2} continues $[(x_2 - x_{12})/v_{x_2}]$ sampling periods. If the motion still does not reach x_2, then

$$\epsilon = \lfloor \frac{x_2 - x_{12}}{v_{x_2}} \rfloor \tag{3.27}$$

is called the *residue ratio* of velocity v_{x_2} at x_2. In the next section, it will be shown how the residue ratio can be propagated to the next line segment.

If $V > v_m$, the time t_{m1}, t_{m2} spent by starting from v_{x_1}, v_{x_2} separately to reach velocity v_m is

$$t_{m1} = \frac{v_m - v_{x_1}}{a}, \qquad t_{m2} = \frac{v_m - v_{x_2}}{a}. \tag{3.28}$$

The total displacements are respectively $(v_m^2 - v_{x_1}^2)/(2a)$, $(v_m^2 - v_{x_2}^2)/(2a)$. The middle segment on $[x_1, x_2]$ that is not covered by the displacements has length

$$L_m = x_2 - x_1 - \frac{v_m^2}{a} + \frac{v_{x_1}^2 + v_{x_2}^2}{2a}. \tag{3.29}$$

It can only be passed with constant velocity v_m, and the time taken is L_m/v_m. Below we sample the motion composed of the three stages.

Let

$$t_{sum} = t_{m1} + t_{m2} + \frac{L_m}{v_m}. \tag{3.30}$$

If t_{sum} is an integer, the sampling sequence of positions is the following: for $0 \leq t \leq t_{sum}$,

$$x(t) = \begin{cases} x_1 + v_{x_1}t + \frac{a}{2}t^2, & \text{if} \quad t \leq t_{m1}, \\ (x_1 + v_{x_1}t_{m1} + \frac{a}{2}t_{m1}^2) + v_m(t - t_{m1}), & \text{if} \quad t_{m1} < t \leq t_{m1} + L_m/v_m, \\ x_2 - v_{x_2}(t_{sum} - t) - \frac{a}{2}(t_{sum} - t)^2, & \text{if} \quad t > t_{m1} + L_m/v_m. \end{cases} \tag{3.31}$$

If t_{sum} is not an integer, then similar to the case $V \leq v_m$, the motion can be modified by either preserving a line segment $[x', x_2]$ if $v_{x_2} \leq v_{x_1}$, or preserving a line segment $[x_1, x'']$ if $v_{x_2} > v_{x_1}$, or adopting the motion of minimal displacement on $[x_1, x_2]$ in the reachability analysis. The occurrence, assimilation and propagation of residue at x_2 are similar to those in the case $V \leq v_m$.

Discrete model:

The motion plan in the discrete model does not rely on discrete sampling of a continuous curve, and the time spent to arrive at the destination is always an integer. In the discrete model, the motion plan relies on the v-x curve, while in the continuous model, the motion plan relies on both the v-x curve and the v-t curve.

(3.10) is the hodograph of the motion accelerating from (x_1, v_{x_1}). The hodograph of the motion accelerating from (x_2, v_{x_2}) is obtained from

$$\begin{array}{rcl} v & = & v_{x_2} + at, \\ x & = & x_2 - v_{x_2}t - \frac{a}{2}t(t-1) \end{array} \tag{3.32}$$

by eliminating t, and is

$$x_2 - x = \frac{v^2 - v_{x_2}^2}{2a} - \frac{v - v_{x_2}}{2}. \tag{3.33}$$

Combining (3.6) and (3.33), we get the maximal velocity $v = V$ as following:

$$V^2 = \frac{v_{x_1}^2 + v_{x_2}^2}{2} + \frac{a(v_{x_1} - v_{x_2})}{2} + a(x_2 - x_1). \tag{3.34}$$

By (3.11), $V \geq \max(v_{x_1}, v_{x_2})$.

If $V \leq v_m$, from (3.34) we get the time t_1, t_2 spent from v_{x_1}, v_{x_2} separately to reach speed V, which is just (3.23). Hence the first subsequence of displacements starting from v_{x_1} is

$$v_{x_1} + at, \qquad t = 1, 2, \ldots, [t_1], \tag{3.35}$$

and the last subsequence of displacements finishing at v_{x_2} is

$$v_{x_2} - at, \qquad t = [t_2], \ldots, 2, 1, 0. \tag{3.36}$$

For $i = 1, 2$, assume that when $t = [t_i]$, the motion starting from (x_i, v_{x_i}) reaches (X_i, V_i). Then $X_1 \leq X_2$, and $V_i \leq V < V_i + a$. Hence $|V_1 - V_2| < a$, and

$$X_2 - X_1 < V_1 + V_2 + a. \tag{3.37}$$

The interval $[X_1, X_2]$ can always be passed in two sampling periods, and in particular,

$$\frac{X_2 - X_1}{\min(V_1, V_2) + a} < 2. \tag{3.38}$$

If the left side of (3.38) is greater or equal to 1, then a new displacement of length $\min(V_1, V_2) + a$ can be inserted between the two subsequences (3.35) and (3.36), forming a new sequence $\mathcal{S}$ together with the two subsequences. The leftover

$$X_2 - X_1 - [\frac{X_2 - X_1}{\min(V_1, V_2) + a}](\min(V_1, V_2) + a) \tag{3.39}$$

can either be assimilated by the sequence $\mathcal{S}$, or form a residue of v_{x_2} at x_2.

If $V > v_m$, let t'_{m1}, t'_{m2} be respectively the time taken by accelerating from v_{x_1}, v_{x_2} to v_m. For $i = 1, 2$, let the motion starting from (x_i, v_{x_i}) arrives at (X_{mi}, V_{mi}) at instance $t = [t_{mi}]$. Then $X_{m1} < X_{m2}$, and $|V_{m1} - V_{m2}| < a$.

If $X_{m2} - X_{m1} < V_{m2}$, then it is the residue of V_{m2} at X_{m2}, and may be assimilated by the sequence of displacements starting from x_1 or x_2. If $X_{m2} - X_{m1} \geq V_{m2}$, then the displacement from $X_{m2} - V_{m2}$ to X_{m2} takes one sampling period, while the motion of constant velocity v_m on $[X_{m1}, X_{m2} - V_{m2}]$ takes $[(X_{m2} - X_{m1} - V_{m2})/v_m]$ sampling periods; the residue

$$X_{m2} - X_{m1} - V_{m2} - [\frac{X_{m2} - X_{m1} - V_{m2}}{v_m}]v_m \tag{3.40}$$

of v_m, if nonzero, may be assimilated by the sequence of displacements starting from x_1 or x_2, or form a residue ratio of v_{x_2} at x_2.

4. Corner turning in 2D motion

A G01 code series gives a 2D motion whose trajectory is a piecewise linear curve in the x-y plane composed of vertices and edges. Locally, the motion plan of the curve is composed of that on each line segment and that at each vertex. The former has been considered in the previous section. The latter is the content of this section.

In the continuous model, if the velocity of a motion at a vertex is nonzero, then since the two edges containing the vertex have different directions, the acceleration at the vertex is infinite. If the velocity at each vertex is zero, then the motion plan of the whole piecewise linear curve is simply the union of the motion plans each on a different edge. This motion plan does not have good performance in improving the machining time. So only the discrete model is valid in order for the velocities at the vertices to be nonzero.

In the discrete model, a *corner turning* at a vertex $\mathbf{P}_2$ refers to moving from a point $\mathbf{P}_1$ on one edge to its vertex $\mathbf{P}_2$ in one sampling period, and then moving from $\mathbf{P}_2$ to another point $\mathbf{P}_3$ on the other edge in another sampling period. $\mathbf{v}_a = \mathbf{P}_2 - \mathbf{P}_1$ is called the *input velocity*, and $\mathbf{v}_b = \mathbf{P}_3 - \mathbf{P}_2$ is called the *output* *velocity*. Their lengths and directions are denoted by

$$v_a = |\mathbf{P}_2 - \mathbf{P}_1|, \qquad v_b = |\mathbf{P}_3 - \mathbf{P}_2|, \qquad \mathbf{e}_a = \frac{\mathbf{v}_a}{v_a}, \qquad \mathbf{e}_b = \frac{\mathbf{v}_b}{v_b}. \tag{4.1}$$

Let a_x, a_y be respectively the maximal accelerations in the x, y directions. The acceleration reaches the maximum at the diagonal directions of the rectangle bounded by the limits, and is

$$a_m = \sqrt{a_x^2 + a_y^2}. \tag{4.2}$$

There are all together four vertices of the rectangle:

$$\begin{array}{rcl} \mathbf{a}_I & = & a_x\mathbf{e}_x + a_y\mathbf{e}_y, \\ \mathbf{a}_{II} & = & -a_x\mathbf{e}_x + a_y\mathbf{e}_y, \\ \mathbf{a}_{III} & = & -a_x\mathbf{e}_x - a_y\mathbf{e}_y, \\ \mathbf{a}_{IV} & = & a_x\mathbf{e}_x - a_y\mathbf{e}_y. \end{array} \tag{4.3}$$

At a corner turning, the acceleration is related to the input and output velocities as following:

$$\mathbf{a} = \mathbf{v}_b - \mathbf{v}_a = v_b\mathbf{e}_b - v_a\mathbf{e}_a, \tag{4.4}$$

from which we get

$$\begin{array}{rcl} v_a & = & \dfrac{\det(\mathbf{e}_b, \mathbf{a})}{\det(\mathbf{e}_a, \mathbf{e}_b)}, \\ v_b & = & \dfrac{\det(\mathbf{e}_a, \mathbf{a})}{\det(\mathbf{e}_a, \mathbf{e}_b)}. \end{array} \tag{4.5}$$

Since $v_a \geq 0$, $v_b \geq 0$, the *zone of allowable accelerations* of the corner turning is the intersection of the rectangle bound with the wedge area bounded by the two rays $-\mathbf{e}_a, \mathbf{e}_b$, as shown in Figure 4.

By the bang-bang control principle [11], in a time-optimal interpolation and with the absence of velocity bound, the acceleration at every instant must be on the boundary of the box $[-a_x, a_x] \times [-a_y, a_y]$. We first analyze the relationship between $\mathbf{a}, v_a, v_b$ when $\mathbf{a}$ varies along a boundary line segment of the box.

It turns out that there are two kinds of boundary line segments of $\mathbf{a}$:

Leading boundary: When $\mathbf{a}$ moves along the boundary line segment, v_a, v_b are either non-increasing simultaneously, or non-decreasing simultaneously. For example, in the allowable zone of the top-left corner turning in Figure 4, all the three line segments $\mathbf{A}'\mathbf{a}_{II}$, $\mathbf{a}_{II}\mathbf{a}_{III}$, $\mathbf{a}_{III}\mathbf{B}'$ are leading boundaries.

Blocking boundary: When $\mathbf{a}$ moves along the boundary line segment, one of v_a, v_b increases strictly, while the other decreases strictly. For example, in the allowable zone of the bottom-right corner turning in Figure 4, line segment $\mathbf{a}_I\mathbf{a}_{II}$ is a blocking boundary, while $\mathbf{A}'\mathbf{a}_{II}$, $\mathbf{a}_I\mathbf{B}'$ are still leading boundaries.

Definition 4.1. A corner turning is said to be *leading*, if all the boundary line segments of the acceleration in the allowable zone are leading, otherwise it is said to be *blocking*.

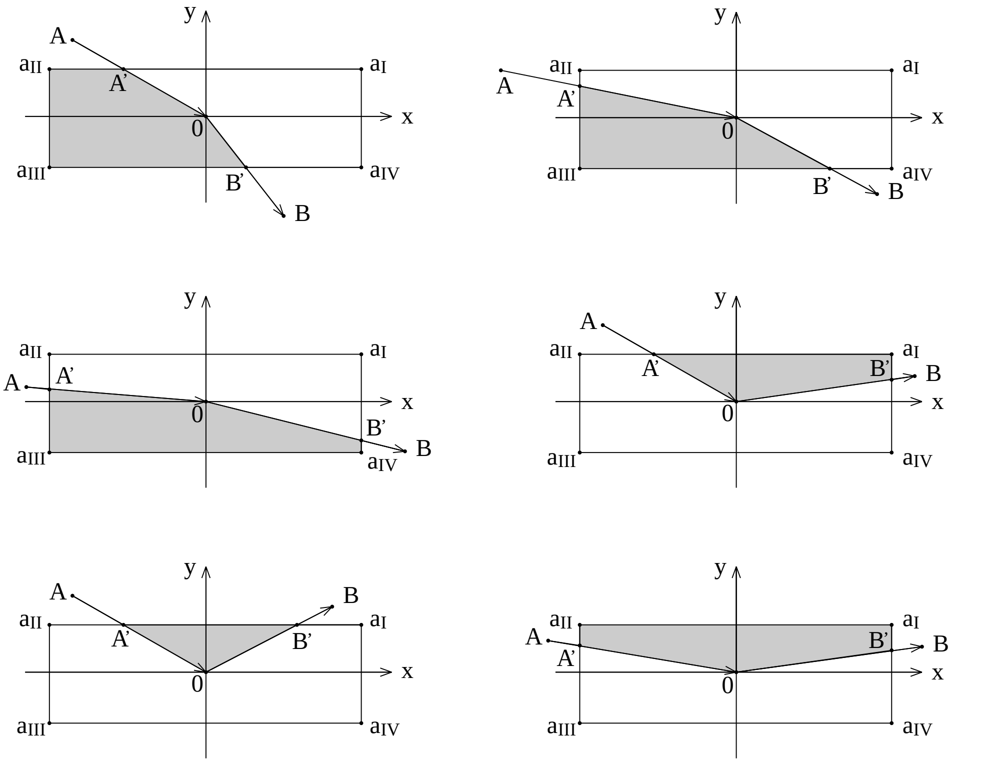

Figure 4. Zone of allowable accelerations of a corner turning.

For example in Figure 4, the first three corner turnings are leading, while the rest are blocking.

There are two problems in the design of a corner-turning motion:

- **Free corner turning:** Given $\mathbf{e}_a, \mathbf{e}_b, \mathbf{P}_2$, determine $\mathbf{a}, v_a, v_b$, or equivalently, $\mathbf{a}, \mathbf{P}_1, \mathbf{P}_3$, so that the total velocity $v_a + v_b$ is maximal.
- **Semifree corner turning:** Given $\mathbf{e}_a, \mathbf{e}_b, \mathbf{P}_2, v_a, \mathbf{P}_1$, determine $\mathbf{a}$ and v_b, or equivalently, $\mathbf{a}$ and $\mathbf{P}_3$, so that v_b is maximized when $\mathbf{a}$ takes values in the allowable zone.

Theorem 4.2. [Free corner turning] Let directions $\mathbf{e}_a, \mathbf{e}_b$ be given, and v_a, v_b be unbounded.

1. If the corner turning is leading, then $v_a + v_b$ is maximized only when the acceleration vector takes the unique vertex of the rectangle bound that is in the same quadrant with neither $-\mathbf{e}_a$ nor $\mathbf{e}_b$.
2. If the corner turning is blocking, then $v_a + v_b$ is maximized when the acceleration vector takes one of the endpoints of the unique blocking boundary within the allowable zone.

Proof. Let the unit vectors along the x-axis and y-axis be $\mathbf{e}_x, \mathbf{e}_y$ respectively. The inequalities on variable $\mathbf{a}$ are

$$\begin{array}{l} 0 \le v_a = \frac{\det(\mathbf{e}_b,\mathbf{a})}{\det(\mathbf{e}_a,\mathbf{e}_b)} \le v_m, \\ 0 \le v_b = \frac{\det(\mathbf{e}_a,\mathbf{a})}{\det(\mathbf{e}_a,\mathbf{e}_b)} \le v_m, \\ -a_x \le v_b \mathbf{e}_x \cdot \mathbf{e}_b - v_a \mathbf{e}_x \cdot \mathbf{e}_a \le a_x, \\ -a_y \le v_b \mathbf{e}_y \cdot \mathbf{e}_b - v_a \mathbf{e}_y \cdot \mathbf{e}_a \le a_y. \end{array} \tag{4.6}$$

They are all linear. Since the object function $v_a + v_b$ is also linear in $\mathbf{a}$, the optimization occurs at the vertices of the convex set defined by the inequalities.

When $v_m = \infty$, in a leading corner turning, v_a and v_b increases or decreases simultaneously, and it is easy to show that the optimization occurs at a unique vertex of the rectangle bound. In a blocking corner turning, it is easy to show that the optimization occurs on the unique blocking boundary. □

For example, in Figure 4, for the first three corner turnings, $v_a + v_b$ is maximized only when $\mathbf{a} = \mathbf{a}_{III}$. For the rest three corner turnings, the maximization occurs at one of $\mathbf{A}', \mathbf{a}_I$, one of $\mathbf{A}', \mathbf{B}'$, and one of $\mathbf{a}_I, \mathbf{a}_{II}$ respectively.

Theorem 4.2 gives an upper bound of $v_a + v_b$ at a corner turning. When the corner turning is leading, the theorem gives upper bounds for v_a, v_b. When the corner turning is blocking, by fixing an acceleration vector optimizing $v_a + v_b$ and extracting the corresponding values of v_a, v_b, one can take them as *optional upper bounds* for v_a, v_b.

The theorem provides a different-velocity approach to corner turning. Locally and for a leading corner turning, this approach can improve the speed of corner turning to about three times when compared with the equal-velocity approach. For example, in a leading corner turning,

- let the incoming ray cuts the horizontal direction by 30°, and the outgoing ray cuts the horizontal direction by 40°. In the equal-velocity approach, the maximal corner turning velocity is 7.003 times a_y, while in the different-velocity approach, the incoming and outgoing velocities are 22.92 and 19.38 times a_y respectively.
- If the horizontal angle of incoming ray is changed to 60° while the outgoing ray is unchanged, the maximal corner turning velocity in the equal-velocity approach is 2.732 a_y, while in the different-velocity approach, it is 9.66 and 6.732 times a_y respectively.

Next consider semifree corner turning. The constraint $\mathbf{v}_a = \mathbf{P}_2 - \mathbf{P}_1$ defines a linear equation in $\mathbf{a}$. In the $\mathbf{a}$-plane, the straight line described by this equation meets the allowable zone at a line segment. The optimization of v_b occurs at one endpoint. In the case where v_b is bounded, the allowable zone is reduced accordingly.

In the above discussion, we assume that the point of corner turning $\mathbf{P}_2$ is exactly the one given by a G01 code. In general this cannot be guaranteed. Suppose we have a sequence of displacements $\mathbf{A}_0, \mathbf{A}_1, \mathbf{A}_2, \mathbf{A}_3$, where $\mathbf{A}_0, \mathbf{A}_1$ are on the input ray of the corner, and $\mathbf{A}_2, \mathbf{A}_3$ are on the output ray, such that $|\mathbf{A}_1\mathbf{A}_2| = v_a$ is the designed input velocity at the vertex, $\mathbf{A}_1 \neq \mathbf{P}_2$ because of error, but $\mathbf{A}_1$ is sufficiently close to the vertex, in the sense that traveling along line segment $\mathbf{A}_1\mathbf{P}_2$ takes less than one sampling period.

The ratio $\epsilon = |\mathbf{A}_1\mathbf{P}_2|/v_a$ is called the *residue ratio* of velocity v_a. In general, for a nonzero velocity V at one end $\mathbf{B}$ of a line segment, if a motion starting from the other end $\mathbf{A}$ of the line segment reaches V at a point $\mathbf{C}$ within the line segment, and continuing the motion with constant velocity V to reach the endpoint $\mathbf{B}$ takes less than one sampling period, then

$$\epsilon = \frac{|\mathbf{CB}|}{V} < 1 \tag{4.7}$$

is called the *residue ratio* of velocity V. If the residue $|\mathbf{CB}|$ cannot be assimilated by the motion sequence on the line segment, then the residue ratio ϵ has to be *propagated* to the next line segment.

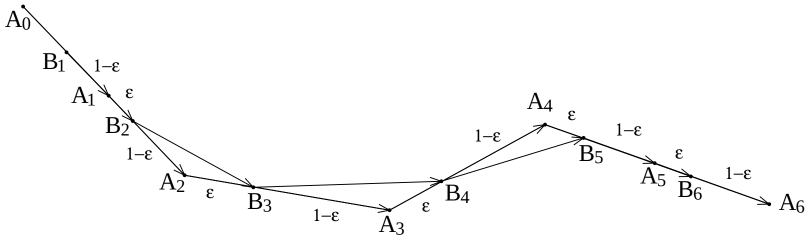

FIGURE 5. Proportional propagation of residue ratio.

As shown in Figure 5, let there be a sequence of positions $\mathbf{A}_0, \mathbf{A}_1, \ldots$, whose velocities and accelerations satisfy linear constraints. For $i \geq 1$, let $\mathbf{v}_i = \mathbf{A}_i - \mathbf{A}_{i-1}$. For $i \geq 2$, let $\mathbf{a}_i = \mathbf{v}_i - \mathbf{v}_{i-1}$. Now a residue ratio $1-\epsilon$ of v_2 occurs at $\mathbf{A}_2$, leading to the following new position of $\mathbf{A}_1$ outside line segment $\mathbf{A}_1\mathbf{A}_2$:

$$\mathbf{B}_1 = \mathbf{A}_1 + (1-\epsilon)(\mathbf{A}_1 - \mathbf{A}_2) = (2-\epsilon)\mathbf{A}_1 - (1-\epsilon)\mathbf{A}_2. \tag{4.8}$$

Accordingly, the motion towards $\mathbf{A}_2$ is advanced to new position

$$\mathbf{B}_2 = \mathbf{A}_1 + \epsilon(\mathbf{A}_2 - \mathbf{A}_1). \tag{4.9}$$

If the residue ratio is *propagated by proportion*, *i.e.*, the new sequence of positions are

$$\mathbf{B}_i = \mathbf{A}_{i-1} + \epsilon(\mathbf{A}_i - \mathbf{A}_{i-1}), \quad i \geq 2, \tag{4.10}$$

then the new velocity sequence is

$$\begin{array}{rclcl} \mathbf{v}'_1 & = & \mathbf{v}_1, & & \\ \mathbf{v}'_2 & = & \mathbf{B}_2 - \mathbf{B}_1 & = & \mathbf{v}_2, \\ \mathbf{v}'_i & = & \mathbf{B}_i - \mathbf{B}_{i-1} & = & (1-\epsilon)\mathbf{v}_{i-1} + \epsilon\mathbf{v}_i, \quad i \geq 3. \end{array} \tag{4.11}$$

As they are all convex combination, all the velocities obey the velocity linear-inequality constraints.

The new acceleration sequence is

$$\begin{array}{lclcl} \mathbf{a}'_2 & = & \mathbf{a}_2, & & \\ \mathbf{a}'_3 & = & \mathbf{v}'_3 - \mathbf{v}'_2 & = & \epsilon\mathbf{a}_3, \\ \mathbf{a}'_i & = & \mathbf{v}'_i - \mathbf{v}'_{i-1} & = & (1-\epsilon)\mathbf{a}_{i-1} + \epsilon\mathbf{a}_i, \quad i \geq 4. \end{array} \tag{4.12}$$

Similarly, these accelerations obey the acceleration linear-inequality constraints.

Under bounded accelerations, if the sum of (a) the residue ratio propagated to a line segment, and (b) the residue ratio occurred in the motion plan on the line segment, is greater than or equal to 1, then the integer part of the sum can be assimilated as single or multiple sampling-period displacement, the rest can be taken as new residue ratio to continue propagation. If their sum is less than 1, then the two residue ratios can be combined to form a new residue ratio for propagation.

Now that the residues are caused by time discretization, a natural idea is to consider concatenating continuous motions on different line segments, combining the residue time $\lfloor t \rfloor$ where $t < T = 1$ with the motion time of the next line segment, and sampling the continuous time variable. In the following, we show that this idea does not work.

In Figure 5, suppose all the sampling points after $\mathbf{A}_3$ are on line segment $\mathbf{A}_2\mathbf{A}_3$. Let the motion starting from $\mathbf{A}_3$ on the line segment have constant acceleration $\mathbf{a}$. Then in the discrete model,

$$\mathbf{x}(t) = (1-t)\mathbf{A}_2 + t\mathbf{A}_3 + \frac{t(t-1)}{2}\mathbf{a}. \tag{4.13}$$

Now on line $\mathbf{A}_0\mathbf{A}_2$, the motion with constant velocity $|\mathbf{A}_1\mathbf{A}_2|$ has residue $(1-\epsilon)|\mathbf{A}_1\mathbf{A}_2|$, and is concatenated with the motion on line $\mathbf{A}_2\mathbf{A}_3$ in motion time ϵ. The new sampling points are

$$\mathbf{B}'_i = (1-\epsilon+3-i)\mathbf{A}_2 + (\epsilon+i-3)\mathbf{A}_3 + \frac{(\epsilon+i-3)(\epsilon+i-4)}{2}\mathbf{a}, \quad i \geq 3. \tag{4.14}$$

In particular,

$$\begin{array}{lcl} \mathbf{B}'_3 & = & (1-\epsilon)\mathbf{A}_2 + \epsilon\mathbf{A}_3 + \dfrac{\epsilon(\epsilon-1)}{2}\mathbf{a}, \\ \mathbf{B}'_4 & = & -\epsilon\mathbf{A}_2 + (1+\epsilon)\mathbf{A}_3 + \dfrac{\epsilon(\epsilon+1)}{2}\mathbf{a}. \end{array} \tag{4.15}$$

The new velocities are

$$\begin{array}{lcl} \mathbf{v}'_2 & = & \mathbf{v}_2, \\ \mathbf{v}'_3 & = & (1-\epsilon)\mathbf{v}_2 + \epsilon\mathbf{v}_3 + \dfrac{\epsilon(\epsilon-1)}{2}\mathbf{a}, \\ \mathbf{v}'_4 & = & \mathbf{v}_3 + \epsilon\mathbf{a}. \end{array} \tag{4.16}$$

The new accelerations are

$$\begin{array}{lcl} \mathbf{a}'_3 & = & \epsilon\mathbf{a}_3 + (1-\epsilon)(-\dfrac{\epsilon}{2}\mathbf{a}), \\ \mathbf{a}'_4 & = & (1-\epsilon)\mathbf{a}_3 + \epsilon(\dfrac{3-\epsilon}{2}\mathbf{a}), \end{array} \tag{4.17}$$

where $\mathbf{a}'_4$ definitely can go beyond the acceleration bounds.

Of course, residue ratio propagation at a vertex induces shape error when compared with the input trajectory. By restricting the lengths of the two sides of the triangle formed by residue ratio proportional propagation, the shape error can be controlled within given tolerance.

5. Lookahead technique and global optimization

In the previous section, at each endpoint of an edge in the input trajectory, a velocity upper bound is generated by local optimization of the corner turning velocities at the endpoint. For a corner turning with incoming ray $\mathbf{e}_a$ and outgoing ray $\mathbf{e}_b$, the upper bound v_{am} of the incoming velocity v_a is also the upper bound of the terminal velocity of the motion on the incoming edge; the upper bound v_{bm} of the outgoing velocity v_b is also the upper bound of the initial velocity of the motion on the outgoing edge. These upper bounds do not take into consideration the availability of the zero boundary velocities of the whole trajectory.

First consider the terminal zero boundary velocity constraint. In order for it to be satisfied we need to update the velocity upper bound at every end of the edges in the trajectory. This strategy is called *lookahead.* The input is a piecewise linear trajectory with vertices $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$, the acceleration bounds a_x, a_y and feedrate bound v_m, and endpoint velocity upper bounds $v_{bm,i}, v_{am,i+1}$ for every edge $\mathbf{P}_i\mathbf{P}_{i+1}$. For the initial and terminal points of the trajectory, the velocities are zero, so $v_{bm,0} = v_{am,N} = 0$. The output is an update of the endpoint velocity upper bounds for the edges.

In the lookahead strategy, the input trajectory is chased inversely in that the motion on an edge $\mathbf{P}_i\mathbf{P}_{i+1}$ always starts from the terminal $\mathbf{P}_{i+1}$ and finishes at the beginning $\mathbf{P}_i$, and at corner $\mathbf{P}_i$ with incoming ray $\mathbf{e}_{a,i}$ and outgoing ray $\mathbf{e}_{b,i}$, the motion is from $-\mathbf{e}_{b,i}$ to $-\mathbf{e}_{a,i}$.

Lookahead strategy: Start from the last edge of the trajectory, do the following:

Step 1: Let the edge be $\mathbf{P}_i\mathbf{P}_{i+1}$. Set the initial velocity at $\mathbf{P}_{i+1}$ to be $v_{am,i+1}$, and accelerate on the edge from $\mathbf{P}_{i+1}$ to $\mathbf{P}_i$ to obtain a maximal reachable velocity V_i at $\mathbf{P}_i$.

Step 2: V_i is the outgoing velocity of the current motion at corner $\mathbf{P}_i$. If $V_i \geq v_{bm,i}$, then do nothing.

Step 3: If $V_i < v_{bm,i}$, then set $v_{am,i} = \infty$ and $v_{bm,i} = V_i$. By semifree corner turning at $\mathbf{P}_i$ from ray $-\mathbf{e}_{b,i}$ to ray $-\mathbf{e}_{a,i}$, compute the new upper bound $v_{am,i}$.

When the above steps are finished for the current edge, continue to the next edge in the inverse order of edges, and do the above steps to the new edge. The loop finishes when the first edge is reached.

Next consider the initial zero boundary velocity constraint. This time the chasing of the trajectory is from the first edge to the last, and on every edge

$\mathbf{P}_i\mathbf{P}_{i+1}$, it is from $\mathbf{P}_i$ to $\mathbf{P}_{i+1}$. The chasing purposes to fix the incoming velocity and outgoing velocity at every corner instead of updating their upper bounds. The procedure is called *global optimization*. The input is the same as in the lookahead procedure, except that the endpoint velocity upper bounds for the edges are updated by the lookhead procedure. The output is the endpoint velocities for edges.

Global optimization: Start from the first edge of the trajectory, do the following:

Step 1: Let the edge be $\mathbf{P}_i\mathbf{P}_{i+1}$. Set the initial velocity at $\mathbf{P}_i$ and terminal velocity at $\mathbf{P}_{i+1}$ be $v_{bm,i}$ and $v_{am,i+1}$ respectively. Determine if the initial velocity at $\mathbf{P}_i$ can reach the terminal velocity at $\mathbf{P}_{i+1}$.

Step 2: If reachable, then do nothing.

Step 3: If not reachable, then compute a maximal reachable velocity V_i from $(\mathbf{P}_i, v_{bm,i})$ along the edge to $\mathbf{P}_{i+1}$. It is the incoming velocity at corner $\mathbf{P}_{i+1}$. By semifree corner turning at $\mathbf{P}_{i+1}$, compute the maximal outgoing velocity $V' \leq v_{bm,i+1}$. Then set $v_{bm,i+1} = V'$.

When the above steps are finished for the current edge, continue to the next edge in the input trajectory, and do the above steps to the new edge. The loop finishes when no edge is left.

Once the endpoint velocities are obtained for every edge, then linear interpolations can be carried out within every edge, with residue ratio assimilation and propagation taken into account. In this way a global linear interpolation algorithm is obtained. All the points described by the new G01 codes are on the input trajectory. In particular, if all residue ratios can be assimilated, then the old G01 codes are subsequences of the new ones.

In the global optimization procedure, either the acceleration vector or the feedrate reaches its bound except at some blocking corner turnings. When there is no blocking corner turning, then the global interpolation is time-optimal. In this case, the result is also *symmetric* in the sense that if the input trajectory is reversed, then the output G01 codes are exactly the reverse of the G01 codes obtained by interpolating the trajectory in the original order.

When blocking corner turning occurs, the current method fails to achieve global time optimality, nor does it guarantee symmetry of the result. so far it is still not clear how to do lookahead and global optimization in the presence of blocking corner turning.

Summing up, the linear interpolation method described above consists of four procedures: (1) local optimization of corner turning velocities, the output is upper bounds of the endpoint velocities for every edge in the input trajectory; (2) lookahead to guarantee zero velocity at the terminal of the trajectory, the output is updated upper bounds of the endpoint velocities for the edges; (3) global optimization to fix the endpoint velocities for the edges, at the same time guarantee zero velocity at the beginning of the trajectory; (4) global linear interpolation by linear interpolation within every edge, and taking into account residue ratio assimilation and propagation.

6. Experiments

The linear interpolation algorithm of the previous section is implemented in C language and integrated with Shenyang Blue-Sky NC System. A 3D milling machine for wood carving equipped with this system is shown in Figure 6 left. The maximal velocity for the $X/Y/Z$ axes are 64/50/10 m/min.

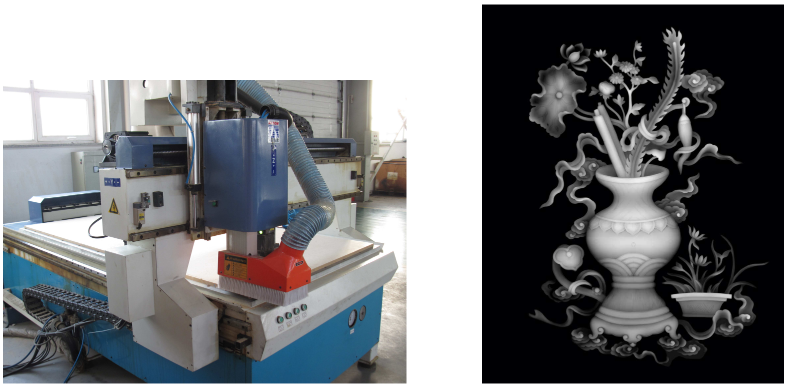

Figure 6. Left: Wood carving milling machine equipped with Shenyang Blue-Sky NC System. Right: a typical surface with a vase design.

A typical surface design with an embedded vase is shown in Figure 6 right. The tool paths of the designed surface all lie in planes parallel to the x-z plane, so the milling along each path is a 2D task. Some typical tool paths are shown in Figure 7. Each paths consists of 400-700 G01 codes with many micro-lines.

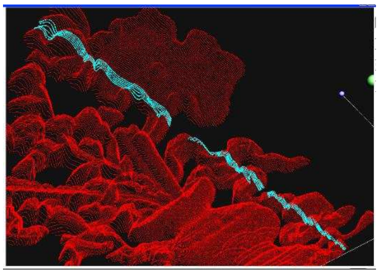

Figure 7. Some tool paths on the surface with vase design (highlighted).

The generation of endpoint velocities for the edges in the input trajectory can be done in real time. For the surface with vase design, depending on the setting of acceleration bounds and feedrate bound, when compared with the classical equal-velocity corner-turning method, the machining time by the current different-velocity corner-turning method can be reduced to 13.91%-81.91%.

Figure 8 shows pictures of the carved vase designs of Figure 6. The left is by the classical equal-velocity corner-turning method, while the right is by our different-velocity corner-turning method. The lotus figures are clearly shown in the right picture, but rather obscure in the left figure. Our method leads to significantly better machining quality.

Figure 8. Carved wood designs. Left: result by classical equal-velocity corner-turning method. Right: result by our different-velocity corner-turning method. Many details are preserved in the right picture, but are missing or obscure in the left picture.

The following table is a comparison of machining time by the two methods. The figure is a circle in the x-y plane. The input is a series of G01 codes obtained by uniform sampling in arclength.

accelerations (mm/s^2)	feedrate (m/min)	machining time Diff-V method (s)	machining time Equal-V method (s)	improved by (%)
1000/1000	12	0.139	0.191	37.41
3000/1000	12	0.129	0.184	42.64
3000/3000	12	0.08	0.143	78.75
6000/6000	12	0.057	0.123	115.79

The following table is another comparison of the two methods. The figure is parabola $y = x^2$ in the plane. The input is a series of G01 codes after uniform sampling on the x-axis.

accelerations (mm/s^2)	feedrate (m/min)	machining time by Diff-V method (s)	machining time by Equal-V method (s)	improved by (%)
1000/1000	12	0.08	0.094	17.5
3000/1000	12	0.068	0.08	17.64
3000/3000	12	0.046	0.064	39.13
6000/6000	12	0.033	0.05	51.51

7. Conclusion

In this paper, the problem of time-optimal G01 code interpolation within a line segment by taking into consideration different motion models is first investigated. Then the problem of 2D corner turning is probed for local optimization of corner turning velocities, and a method of different-velocity corner turning is established. After developing techniques for preserving the zero boundary velocity constraints on the input trajectory, an algorithm is proposed for time-optimal interpolation without resorting to smooth curve approximation or blending. Real wood carving experiments show significant improvements in both machining time and machining quality.

The extension of the method to 3D Cartesian motion is obvious in that a corner turning in any dimension is a 2D problem. Extending to 5-axis machining under bounded accelerations seems much the same. Extending to bounded jerk control can be done when the input and output accelerations at a corner turning are both zero, but otherwise difficult.

References

[1] Du, D., Liu, Y., Yan, C., and Li, C. An accurate adaptive parametric curve interpolator for NURBS curve interpolation. *International Journal of Advanced Manufacturing Technology* (2007) **32**: 999-1008.

[2] Du, D., Liu Y., Guo, X., Yamazaki, K., and Fujishima M. An accurate adaptive NURBS curve interpolator with real-time flexible acceleration/deceleration control. *Robotics and Computer-Integrated Manufacturing* (2010) **26**: 273-281.

[3] Erkorkmaz, K., Yeung, C.H., and Altintas, Y. Virtual CNC system. Part II. High speed contouring application. *Int. J. Machine Tools Manuf.* (2006) **46**(10): 1124-1138.

[4] Feng, J., Li, Y., Wang, Y., and Chen, M. Design of a real-time adaptive NURBS interpolator with axis acceleration limit. *International Journal of Advanced Manufacturing Technology* (2010) **48**: 227-241.

[5] Hu, J., Xiao, L., and Wang, Y. An optimal feedrate model and solution algorithm for a high-speed machine of small line blocks with look-ahead. *Int. J. Adv. Manuf. Technol.*, (2006) **27**: 930-935.

[6] Jahanpour, J. and Imani, B.M. Real-time P-H curve CNC interpolators for high speed cornering, *Int. J. Adv. Manuf. Technol.* (2008) **39**: 302-316.

[7] LaSalle, J. The time optimal control problem. In: *Contributions to the Theory of Nonlinear Oscillations*, vol 5, Cesari, L. et al. (eds), Princeton, NJ, Princeton University Press, 1960.

[8] Lee, A.C., Lin, M.T., Pan, Y.R., and Lin, W.Y. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Computer-Aided Design* (2011) **43**: 612-628.

[9] Lin, M.T., Tsai, M.S., and Yau, H.T. Development of real-time look-ahead algorithm for NURBS interpolator with consideration of servo dynamics. In: *Proceedings of the 46th IEEE conference on decision and control*, 2007, pp. 1862-1867.

[10] Luo, F.Y., Zhou, Y.F., and Yin, J. A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead. *Int. J. Adv. Manuf. Technol.* (2007) **35**: 505-518.

[11] Timar, S.D. and Farouki, R.T. Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds. *Robot CIM-Int. Manuf.* (2007) **23**(2): 563-579.

[12] Tsai, M.S., Nien, H.W., and Yau, H.T. Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. *Computer-Aided Design* (2008) **40**: 554-566.

[13] Wang, A., Zhang, J., and Wu, Y. *Modern Numerical Control Principle and Control Systems*, National Defense Engineering Publishers, Beijing, 2005.

[14] Wang, J.B. and Yau, H.T. Real-time NURBS interpolator: Application to short linear segments. *Int. J. Adv. Manuf. Technol.* (2009) **41**: 1169-1185.

[15] Xu, H.F. and Wang, Y.H. Research on feedrate model of look-ahead and solution algorithm for high-speed machining of small line segments. *Mech. Eng.* (2005) **9**: 9-13.

[16] Zhang, L., Sun, R., Li, H., and Gao, X. Local optimization based on motion curve and global optimization based on lookahead control in NC machining. *J. Sys. Sci. and Math.* (2011) **11**: 1548-1561.

[17] Zhang, L., Sun, R., Gao, X., and Li, H. Micro-line interpolation algorithm in high-speed NC machining and adaptive lookahead manipulation. *Science in China* (E): Engineering Science and Material Science (2011) **6**(41): 744-789.

Hongbo Li
KLMM, AMSS & NCMIS
Chinese Academy of Sciences
Beijing 100190, China
e-mail: `hli@mmrc.iss.ac.cn`

Xiaoshan Gao
KLMM, AMSS & NCMIS
Chinese Academy of Sciences
Beijing 100190, China
e-mail: `xgao@mmrc.iss.ac.cn`

Lixian Zhang
KLMM, AMSS & NCMIS
Chinese Academy of Sciences
Beijing 100190, China
e-mail: `shadowfly12@126.com`

Ruiyong Sun
KLMM, AMSS
Chinese Academy of Sciences
Beijing 100190, China
e-mail: `sunruiyong@126.com`

Submitted: February 20, 2012

Received: February 20, 2012
Revised: June 10, 2012
Accepted: June 25, 2012