Mathematics Mechanization Research Preprints
KLMM, Chinese Academy of Sciences
Vol. 29, 189–205, September 2010

A Greedy Algorithm for Feed-rate Planning of CNC Machines along Curved Tool Paths with Confined Jerk for Each Axis

Ke Zhang, Xiao-Shan Gao, Hongbo Li, Chun-Ming Yuan
KLMM, Institute of Systems Science, Chinese Academy of Sciences
E-mail: xgao@mmrc.iss.ac.cn

Abstract. The problem of optimal feed-rate planning along a curved tool path for 3-axis CNC machines with a jerk limit for each axis is addressed. We prove that the optimal feed-rate planning must use "Bang-Bang" control, that is, at least one of the axes reaches its jerk bound throughout the motion. As a consequence, the optimal parametric velocity can be expressed as a piecewise analytic function of the curve parameter u. We also give the explicit formula for the velocity function by solving a second order differential equation. Under a "greedy rule", an algorithm for optimal jerk confined feed-rate planning is presented, together with an example.

Keywords. Optimal feed-rate planning, confined jerk, velocity limit surface, parametric tool path, "Bang-Bang" control.

1 Introduction

The feed-rate optimization along curved tool paths is an important problem in CNC machining. In the feed-rate planning, the acceleration on each axis of the machine must be constrained, because the torque (or force) capabilities of the axes' drives are limited. Therefore, the problem is how to identify the feed-rate along a given path such that the machining time is minimal without exceeding the capabilities of the actuators.

Bobrow et al [1], Shiller and Lu [2] gave algorithms to determine the minimum-time motion for a robot manipulator along a specific path (at least a smooth curve) with acceleration bounds on x, y, z axes. Farouki and Timar [3, 4] planned the feed-rate for CNC machining, also with acceleration bounds on x, y, z axes, and gave a piecewise-analytic expression of the optimal velocity planning function. Yuan and Gao [5] provided a time optimal feed-rate planning method with tangential acceleration and chord error bounds. All of the methods mentioned above used the velocity limit curve and its switching points in the $u - \dot{u}$ phase plane to obtain an optimal solution which is a continuous time optimal velocity function along a specific path. Dong and Stori [6] gave a discrete greedy algorithm for the above problem with a acceleration bound on each axis. These methods are all based on the idea of "Bang-Bang" control, that is, at least one of the axes reaches its acceleration bound throughout the motion.

However, the acceleration profile obtained with the above methods has discontinuities, since the acceleration may change from the maximum A to the minimum $-A$ instantly. These

discontinuities correspond to step changes in the force output demanded of the drive, cause vibrations and then large contouring errors. One method to reduce vibrations is introducing jerk constraints along each axis to the original problem. Then we will obtain a feed-rate planning with continuous acceleration.

When jerk constraints are added, the analysis must be performed in the $u - \dot{u} - \ddot{u}$ phase space instead of the $u - \dot{u}$ phase plane. The new optimization problem becomes more difficult. However, it is much easier when considering the constraints of the tangential acceleration and jerk. Such problems have received much attention in the robotics and manufacturing literature. Altintas and Erkorkmaz [8] presented a quintic spline trajectory generation algorithm that produces continuous position, velocity, and acceleration profiles with confined tangential acceleration and jerk. Macfarlane and Croft [9] developed and implemented an online method to obtain smooth, jerk-bounded trajectories with fifth-order polynomials for industrial robot applications. Their method is near time optimal with confined tangential jerk and acceleration. Nam and Yang [10] presented a recursive trajectory generation method that estimates an admissible path increment and determines the initiation of the final deceleration stage according to the distance left to travel estimated at every sampling time, resulting in exact feed-rate trajectory generation through tangential jerk-confined acceleration profiles for the parametric curves. Lin et al [11] proposed a dynamics-based interpolator with real-time look-ahead algorithm to generate a smooth and tangential jerk-confined acceleration/deceleration feed-rate profile. Emami and Arezoo [12] introduced a look-ahead trajectory generation method which determines the deceleration stage according to the fast estimated arc length and the reverse interpolation of each curve at every sampling time. They obtained a feed-rate trajectory with tangential jerk-confined acceleration profiles for the NURBS curves. Lai et al [13] further proposed a method which can generate velocities with jerk limits as well as chord error, speed, and acceleration limits. The method uses a discrete model and satisfies all these constraints by backtracking at each step.

In order to make full use of the capabilities of the machine tool, it is desirable to solve the problem with jerk constraints on each axis, because the drivers of the axes of a CNC machine are controlled independently. Using a jerk limit on each axis will lead to a continuous acceleration curve for each axis. Dong et al [14] extended their discrete greedy algorithm [6] by adding jerk constraints on each axes and obtained a discrete optimal solution. However, none of these prior approaches have attempted to get an analytical solution with a continuous model with jerk constraints on each axis.

In this paper, we consider the problem of optimal feed-rate planning along a specific tool path $\vec{r}(u)$ with a confined jerk on each axis of a 3-axis machine. First, we prove that the time-optimal feed-rate planning must use "Bang-Bang" control, that is, at least one of the axes reaches its jerk bound throughout the motion. Then we give an optimal feed-rate planning algorithm under a "greedy rule": using the maximal jerk as much as possible.

Our algorithm has two key components, which are also the main contribution of this paper. The first one is how to compute the parametric velocity function after the control axis and maximal (or minimal) jerk are given. To compute the parametric velocity function, we need to solve a second-order differential equation, and the analytic solutions are given. We also introduce the CASS (*control axis switching surface*). The control axis should be changed when the velocity integration trajectory passes through a CASS. The second key

component is to introduce and use the VLS (*velocity limit surface*) for the feed-rate planning. It is similar to the VLC (*velocity limit curve*) in the feed-rate planning with acceleration constraints [1, 2, 3, 4]. The VLS is a surface in the $u - \dot{u} - \ddot{u}$ space which limits the parametric velocity and acceleration.

The general idea of our algorithm is to compute the integration trajectory forward from $(0, 0, 0)$ in the $u-\dot{u}-\ddot{u}$ space under the limit of VLS and our "greedy rule"; then to compute the integration trajectory backward from $(1, 0, 0)$ in a similar way; and finally to obtain a complete velocity integration trajectory with continuous acceleration by connecting the two integration trajectories.

The rest of our paper will be organized as follows. Section 2 gives the mathematical description and theoretical analysis of our feed-rate optimization problem. Section 3 gives our feed-rate planning algorithm. Section 4 gives an example. Section 5 concludes the paper.

2 Problem description and theoretical analysis

2.1 Problem description

For brevity, we just consider a plane piecewise parametric curve as the tool path:

$$\vec{r}(u) = (x(u), y(u)), 0 \leq u \leq 1,$$

where $x(u), y(u) \in C^2([0, 1])$. Furthermore, we assume that each segment of the curve is infinitely differentiable. For instance, a cubic B-spline curve satisfies the conditions. Also, we only consider the bounds on the x and y jerk components. The extension to spatial paths is relatively straightforward but more tedious. We denote the derivatives with respect to time t and the parameter u by dots and primes, respectively:

$$\dot{u} = du/dt, x' = dx/du.$$

Then, it is obvious that

$$\dot{u}' = \frac{\ddot{u}}{\dot{u}}, \tag{1}$$

$$\ddot{u}' = \frac{\dddot{u}}{\dot{u}}, \tag{2}$$

and

$$\dot{u}'' = (\frac{\ddot{u}}{\dot{u}})' = \frac{\dddot{u}}{\dot{u}^2} - \frac{\ddot{u}^2}{\dot{u}^3}. \tag{3}$$

The jerks on the x and y axes are:

$$\begin{cases} j_x = \dddot{x} = ((x'\dot{u})'\dot{u})'\dot{u} = x'''\dot{u}^3 + 3x''\dot{u}^2\dot{u}' + x'\dot{u}(\dot{u}')^2 + x'\dot{u}^2\dot{u}'', \\ j_y = \dddot{y} = ((y'\dot{u})'\dot{u})'\dot{u} = y'''\dot{u}^3 + 3y''\dot{u}^2\dot{u}' + y'\dot{u}(\dot{u}')^2 + y'\dot{u}^2\dot{u}''. \end{cases} \tag{4}$$

Substituting (1)(3) into (4), j_x, j_y can be expressed as

$$\begin{cases} j_x = x'''\dot{u}^3 + 3x''\dot{u}\ddot{u} + x'\dddot{u}, \\ j_y = y'''\dot{u}^3 + 3y''\dot{u}\ddot{u} + y'\dddot{u}. \end{cases} \tag{5}$$

We call $\dot{u}, \ddot{u}$, and $\dddot{u}$ *parametric velocity, parametric acceleration*, and *parametric jerk*, respectively. Then our feed-rate optimization problem becomes to plan the parametric velocity $\dot{u} \in C^1([0,1])$, such that the machining time is minimal:

$$\min T = \int_0^1 \frac{du}{\dot{u}} \tag{6}$$

under the following constraints:

$$\begin{cases} |j_x| \le J_x, \\ |j_y| \le J_y, \end{cases} \tag{7}$$

$$\begin{cases} \dot{u}|_{u=0,1} = 0, \\ \ddot{u}|_{u=0,1} = 0, \end{cases} \tag{8}$$

where J_x, J_y are positive constants, denoting maximal jerks of x, y axes, respectively.

2.2 Optimal solution is "Bang-Bang" control

In this section, we will prove that the solution to our optimal problem must be "Bang-Bang" control, that is, at least one of the axes reaches its jerk bound throughout the motion. In other words, $j_x = \pm J_x$ or $j_y = \pm J_y$ at every time. When there is an axis whose jerk reaches its bound, it is called the *control axis*.

We prove the claim by contradiction (see Fig.1). Assume that the optimal parametric velocity function is $\dot{u}$, and there exists an interval $[u_1, u_2]$ in $[0,1]$, such that neither j_x nor j_y reaches its bound for $u \in [u_1, u_2]$, i.e., the inequalities (7) are strict.

From (4), j_x, j_y can be expressed as functions in $u, \dot{u}, \dot{u}', \dot{u}''$, denoted by f, g, respectively:

$$\begin{cases} j_x = f(u, \dot{u}, \dot{u}', \dot{u}'') = x'''\dot{u}^3 + 3x''\dot{u}^2\dot{u}' + x'\dot{u}(\dot{u}')^2 + x'\dot{u}^2\dot{u}'', \\ j_y = g(u, \dot{u}, \dot{u}', \dot{u}'') = y'''\dot{u}^3 + 3y''\dot{u}^2\dot{u}' + y'\dot{u}(\dot{u}')^2 + y'\dot{u}^2\dot{u}''. \end{cases}$$

So, for every $u \in [0,1]$, f, g are polynomials in $\dot{u}, \dot{u}', \dot{u}''$. Using (7), there exist positive constants A_1 and A_2, such that

$$\begin{cases} |f(u, \dot{u}, \dot{u}', \dot{u}'')| \le A_1 < J_x, \\ |g(u, \dot{u}, \dot{u}', \dot{u}'')| \le A_2 < J_y \end{cases} \tag{9}$$

is established for $u \in [u_1, u_2]$.

For every positive ε, we construct

$$\Delta\dot{u} = \begin{cases} \varepsilon(1 + \cos\frac{\pi(2u - u_1 - u_2)}{u_2 - u_1}) & u_1 \le u \le u_2; \\ 0 & otherwise. \end{cases}$$

It is easy to show that

$$\begin{cases} \Delta\dot{u}|_{u_1,u_2} = 0, \\ (\Delta\dot{u})'|_{u_1,u_2} = 0, \end{cases} \tag{10}$$

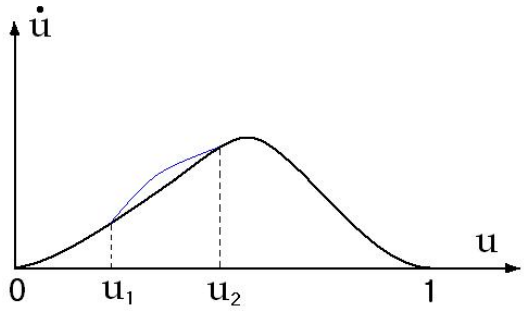


Fig. 1. optimal solution is "Bang-Bang" control

and

$$\begin{cases} 0 \le \Delta\dot{u} \le 2\varepsilon, \\ |\Delta\dot{u}'| \le B_1\varepsilon, \\ |\Delta\dot{u}''| \le B_2\varepsilon, \end{cases} \tag{11}$$

where B_1, B_2 are positive constants.

Let $\dot{u}^* = \Delta\dot{u} + \dot{u}$, from (10) we know that $\dot{u}^* \in C^1([0,1])$. For every $u \in [u_1, u_2]$, using first-order Taylor expansion of f, g to $\dot{u}, \dot{u}', \dot{u}''$, we obtain:

$$\begin{aligned} f(u, \dot{u}^*, \dot{u}^{*'}, \dot{u}^{*''}) = \; & f(u, \dot{u}, \dot{u}', \dot{u}'') + \Delta\dot{u}\frac{\partial f}{\partial \dot{u}}(u, \xi(u), \eta(u), \tau(u)) \\ & + \Delta\dot{u}'\frac{\partial f}{\partial \dot{u}'}(u, \xi(u), \eta(u), \tau(u)) \\ & + \Delta\dot{u}''\frac{\partial f}{\partial \dot{u}''}(u, \xi(u), \eta(u), \tau(u)), \end{aligned} \tag{12}$$

where $\xi(u)$ is between $\dot{u}$ and $\dot{u}^*$, $\eta(u)$ is between $\dot{u}'$ and $\dot{u}^{*'}$, and $\tau(u)$ is between $\dot{u}''$ and $\dot{u}^{*''}$. So $\xi(u), \eta(u), \tau(u)$ are bounded for $u \in [u_1, u_2]$. Because the partial derivatives of f in (12) are all polynomials in $\dot{u}, \dot{u}', \dot{u}''$, we have constants F_1, F_2, F_3 such that $\forall u \in [u_1, u_2]$:

$$\begin{cases} |\dfrac{\partial f}{\partial \dot{u}}(u, \xi(u), \eta(u), \tau(u))| \le F_1, \\ |\dfrac{\partial f}{\partial \dot{u}'}(u, \xi(u), \eta(u), \tau(u))| \le F_2, \\ |\dfrac{\partial f}{\partial \dot{u}''}(u, \xi(u), \eta(u), \tau(u))| \le F_3. \end{cases} \tag{13}$$

Using (9) (11) (12) (13), we have:

$$|f(u, \dot{u}^*, \dot{u}^{*'}, \dot{u}^{*''})| \le A_1 + C_1\varepsilon,$$

where $C_1 = 2F_1 + B_1F_2 + B_2F_3$. In a similar way, there exists a C_2 such that:

$$|g(u, \dot{u}^*, \dot{u}^{*'}, \dot{u}^{*''})| \le A_2 + C_2\varepsilon.$$

We just need to choose

$$\varepsilon = \min\{(J_x - A_1)/C_1, (J_y - A_2)/C_2\},$$

and then $\dot{u}^*$ also satisfies the constraints (7) (8) and the continuity condition. But we have $\dot{u}^* \geq \dot{u}$, and $\dot{u}^* > \dot{u}$ for $u \in (u_1, u_2)$, from (6) we know that $\dot{u}^*$ is a better solution, which contradicts the original claim of optimality of $\dot{u}$. So the optimal solution of our problem is "Bang-Bang" control.

3 Feed-rate planning algorithm

The key idea of our algorithm for feed-rate optimization along curved tool paths is: in the $u - \dot{u} - \ddot{u}$ space, using the jerk constraints to deduce a kind of velocity limit surfaces, then generating an integration trajectory with the maximal parametric jerk under the limit of such kind of surfaces. Before the integration trajectory reaches these surfaces, we use the minimal parametric jerk to generate an integration trajectory to keep the continuity of the acceleration curve.

3.1 Parametric jerk constraints

Using (5) (7), we can rewrite the jerk limits to be constraints of the parametric jerk $\dddot{u}$:

(a) When $x'y' \neq 0$, (7) is equivalent to:

$$\begin{cases} f_1(u, \dot{u}, \ddot{u}) \leq \dddot{u} \leq g_1(u, \dot{u}, \ddot{u}), \\ f_2(u, \dot{u}, \ddot{u}) \leq \dddot{u} \leq g_2(u, \dot{u}, \ddot{u}), \end{cases} \tag{14}$$

where

$$f_1(u, \dot{u}, \ddot{u}) = \begin{cases} (-J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x' & x' > 0; \\ (J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x' & x' < 0. \end{cases}$$

$$g_1(u, \dot{u}, \ddot{u}) = \begin{cases} (J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x' & x' > 0; \\ (-J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x' & x' < 0. \end{cases}$$

$$f_2(u, \dot{u}, \ddot{u}) = \begin{cases} (-J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y' & y' > 0; \\ (J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y' & y' < 0. \end{cases}$$

$$g_2(u, \dot{u}, \ddot{u}) = \begin{cases} (J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y' & y' > 0; \\ (-J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y' & y' < 0. \end{cases}$$

Let

$$\begin{cases} J_-(u, \dot{u}, \ddot{u}) = \max\{f_1, f_2\}, \\ J_+(u, \dot{u}, \ddot{u}) = \min\{g_1, g_2\}. \end{cases} \tag{15}$$

Then constraints (14) become

$$J_-(u, \dot{u}, \ddot{u}) \leq \dddot{u} \leq J_+(u, \dot{u}, \ddot{u}). \tag{16}$$

It shows that in every point of the $u - \dot{u} - \ddot{u}$ space, $\dddot{u}$ has upper and lower bounds: J_+, J_-.

(b) When $x' = 0$, (7) becomes:

$$\begin{cases} -J_x \le x'''\dot{u}^3 + 3x''\dot{u}\ddot{u} \le J_x, \\ f_2(u,\dot{u},\ddot{u}) \le \dddot{u} \le g_2(u,\dot{u},\ddot{u}). \end{cases} \tag{17}$$

The first equation of (17) indicates the range of $(\dot{u}, \ddot{u})$ on the u section where u satisfies $x' = 0$. The range is limited by two curves $x'''\dot{u}^3+3x''\dot{u}\ddot{u} = -J_x$ and $x'''\dot{u}^3+3x''\dot{u}\ddot{u} = J_x$ in the $u-\dot{u}-\ddot{u}$ space. We call these curves *type one velocity switching curve* (abbr. VSC_1). The second equation of (17) still shows the upper and lower bounds of $\dddot{u}$, where now $J_+ = g_2, J_- = f_2$.

(c) When $y' = 0$, the analysis to the following equations are similar:

$$\begin{cases} f_1(u,\dot{u},\ddot{u}) \le \dddot{u} \le g_1(u,\dot{u},\ddot{u}), \\ -J_y \le y'''\dot{u}^3 + 3y''\dot{u}\ddot{u} \le J_y. \end{cases} \tag{18}$$

The first equation of (18) shows the upper and lower bounds of $\dddot{u}$, where $J_+ = g_1, J_- = f_1$. The second equation of (18) indicates the range of $(\dot{u}, \ddot{u})$ on the u section where u satisfies $y' = 0$. It is limited by two curves $y'''\dot{u}^3 + 3y''\dot{u}\ddot{u} = -J_y$ and $y'''\dot{u}^3 + 3y''\dot{u}\ddot{u} = J_y$ in the $u - \dot{u} - \ddot{u}$ space. These curves are also VSC_1.

3.2 Integration trajectory and control axis switching surface

Since we have proven that the solution to our optimal problem uses"Bang-Bang" control, it is necessary to deduce the parametric velocity function $\dot{u}$ when any axis reaches its jerk bound. Using (1), it is easy to show that once the parametric velocity function $\dot{u}$ in u is known, the parametric acceleration function $\ddot{u}$ in u is determined. Then the two functions $\dot{u}$ and $\ddot{u}$ in u determine a curve in the $u-\dot{u}-\ddot{u}$ space. We call the curve *integration trajectory*. This subsection will discuss 1) how to compute the parametric velocity function when any axis reaches its jerk bound and 2) how to choose control axis and switch axis.

Firstly, we deal with the solutions of parametric velocity function $\dot{u}$. For example, if the x-axis reaches its jerk bound J_x, we need to solve the following second-order differential equation:

$$((x'\dot{u})'\dot{u})'\dot{u} = J_x. \tag{19}$$

Let $f = x'\dot{u}$. The differential equation becomes

$$\frac{d}{dx}(\frac{df}{dx}f)f = J_x.$$

Let $g = \frac{df}{dx}$. Then we have

$$g^2 f + g f^2 \frac{dg}{df} = J_x.$$

Let $h = g^2$. The equation above becomes

$$\frac{dh}{df} = \frac{2J_x}{f^2} - \frac{2h}{f}.$$

Solving the differential equation, we obtain

$$h = \frac{2J_x}{f} - \frac{C_1}{f^2}, \tag{20}$$

where C_1 is an integration constant. The above equation can be rewritten as

$$\frac{df}{dx} = \pm\frac{\sqrt{2J_x f - C_1}}{f}.$$

We solve it to obtain

$$x - C_2 = \pm \int \frac{f df}{\sqrt{2J_x f - C_1}} = \pm(C_1\sqrt{2J_x f - C_1} + \frac{1}{3}\sqrt{2J_x f - C_1}^3)/2J_x^2, \tag{21}$$

where C_2 is an integration constant. We solve the equation above to obtain

$$\dot{u} = \frac{1}{2J_x x'}[\omega(U + \sqrt{U^2 + C_1^3}\ )^{\frac{2}{3}} + \omega^2(U - \sqrt{U^2 + C_1^3}\ )^{\frac{2}{3}} - C_1], \tag{22}$$

where $U = 3J_x^2(x - C_2)$, $\omega^3 = 1$.

Now we deduce the expressions of these integration constants C_1, C_2 in $u, \dot{u}, \ddot{u}$ for our later algorithm. We have

$$h = (\frac{df}{dx})^2 = (\frac{x''\dot{u}^2 + x'\ddot{u}}{x'\dot{u}})^2.$$

Substituting it into (20), we get

$$C_1 = 2J_x f - hf^2 = 2J_x x'\dot{u} - (x''\dot{u}^2 + x'\ddot{u})^2. \tag{23}$$

Use (21) (23) to obtain

$$C_2 = x \pm ((x''\dot{u}^2 + x'\ddot{u})^3 - 3J_x x'\dot{u}(x''\dot{u}^2 + x'\ddot{u}))/3J_x^2. \tag{24}$$

Then from (23) (24), the integration constants C_1, C_2 are determined by specifying a known point on the integration trajectory in the $u - \dot{u} - \ddot{u}$ space.

If y-axis reaches its jerk bound J_y, we solve the parametric velocity function in the same way to get

$$\dot{u} = \frac{1}{2J_y y'}[\omega(U + \sqrt{U^2 + C_1^3}\ )^{\frac{2}{3}} + \omega^2(U - \sqrt{U^2 + C_1^3}\ )^{\frac{2}{3}} - C_1], \tag{25}$$

where $U = 3J_y^2(y - C_2)$, $\omega^3 = 1$. We also have

$$C_1 = 2J_y y'\dot{u} - (y''\dot{u}^2 + y'\ddot{u})^2, \tag{26}$$

$$C_2 = y \pm ((y''\dot{u}^2 + y'\ddot{u})^3 - 3J_y y'\dot{u}(y''\dot{u}^2 + y'\ddot{u}))/3J_y^2. \tag{27}$$

In (22) or (25), if $U^2+C_1^3$ is negative in some interval of u, the expression of $\dot{u}$ should be converted, for the convenience of our computation. Taking (25) for example, we substitute ω by $e^{\frac{2}{3}ik\pi}$ $(k=0,1,2)$ to obtain

$$\begin{aligned}\dot{u} &= \frac{-C_1}{2J_y y'}[e^{\frac{2}{3}ik\pi}(\frac{U}{(-C_1)^{3/2}}+i\sqrt{1-\frac{U^2}{(-C_1)^3}})^{2/3}+e^{-\frac{2}{3}ik\pi}(\frac{U}{(-C_1)^{3/2}}-i\sqrt{1-\frac{U^2}{(-C_1)^3}})^{2/3}+1]\\ &= \frac{-C_1}{2J_y y'}[e^{\frac{2}{3}ik\pi}e^{\frac{2}{3}i\arccos\frac{U}{(-C_1)^{3/2}}}+e^{-\frac{2}{3}ik\pi}e^{-\frac{2}{3}i\arccos\frac{U}{(-C_1)^{3/2}}}+1]\\ &= \frac{-C_1}{2J_y y'}[2\cos\frac{2}{3}(\arccos\frac{U}{(-C_1)^{3/2}}+k\pi)+1].\end{aligned}$$

If x (or y)-axis reaches its jerk bound $-J_x$ (or $-J_y$), we just need to replace J_x (or J_y) by $-J_x$ (or $-J_y$) in the solutions above.

Now we turn to deal with how to decide the control axis. There are two problems: determining control axis at the starting point and axis switching during the motion.

If we integrate $\dddot{u}=J_+(u,\dot{u},\ddot{u})$ forward from $(0,0,0)$ in the $u-\dot{u}-\ddot{u}$ space as the current integration trajectory, it is easy to determine the control axis from $(u,\dot{u},\ddot{u})=(0,0,0)$ and the three cases (a),(b),(c) in section 3.1. For example, when $x'(0)>0, y'(0)>0$, x-axis is the control axis if and only if $g_1(0,0,0)=J_x/x'<g_2(0,0,0)=J_y/y'$.

From (15), when we integrate $\dddot{u}=J_+(u,\dot{u},\ddot{u})$, the expression of the parametric velocity may change if the values of g_1,g_2 vary. It means that the control axis should be switched. So we call $g_1=g_2$ the *control axis switching surface* (abbr. CASS). For example, if the integration trajectory passes through a CASS from the region $g_1<g_2$ to the region $g_1>g_2$, the control axis should be switched from x to y, and vice versa.

The situation is similar when integrating $\dddot{u}=J_-(u,\dot{u},\ddot{u})$, and the CASS is then $f_1=f_2$. When the integration trajectory passes through the CASS from the region $f_1>f_2$ to the region $f_1<f_2$, the control axis should be switched from x to y, and vice versa. We will not mention about how to deal with the CASS when integrating $\dddot{u}=J_+(u,\dot{u},\ddot{u})$ or $\dddot{u}=J_-(u,\dot{u},\ddot{u})$ in our later algorithm.

3.3 Velocity limit surface and velocity switching curve

Let $J_-(u,\dot{u},\ddot{u})$ and $J_+(u,\dot{u},\ddot{u})$ be the expressions defined in (15). We call $J_-(u,\dot{u},\ddot{u})=J_+(u,\dot{u},\ddot{u})$ the *velocity limit surface* (abbr. VLS), which is an algebraic surface in the following region of the $u-\dot{u}-\ddot{u}$ space:

$$D=\{(u,\dot{u},\ddot{u})|0\le u\le 1,\dot{u}\ge 0\}.$$

Obviously, the integration trajectories cannot go beyond the VLS, that is, we can only plan the integration trajectories in the region where $J_-(u,\dot{u},\ddot{u})\le J_+(u,\dot{u},\ddot{u})$ is valid. Intuitively, this region is just a part of D divided by the VLS, which contains $(0,0,0),(1,0,0)$ (see Fig.2).

From $f_1<g_1, f_2<g_2$, and (14) (15), we know that the VLS has two components:

$$f_1(u,\dot{u},\ddot{u})=g_2(u,\dot{u},\ddot{u}),$$

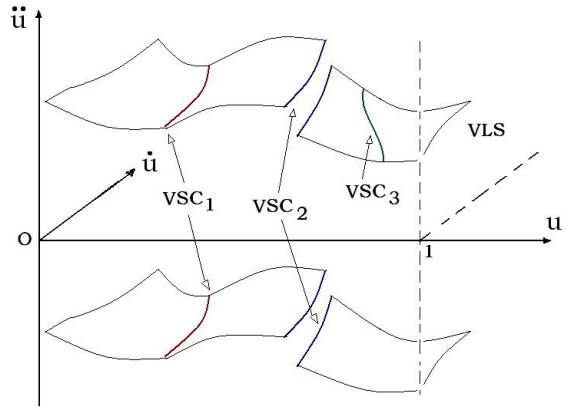


Fig. 2. VLS and three kinds of VSC

$$f_2(u, \dot{u}, \ddot{u}) = g_1(u, \dot{u}, \ddot{u}),$$

and these two components do not intersect. Assuming the contrary, we list the simultaneous equations (consider the situation of $x' > 0, y' > 0$):

$$\begin{cases} (-J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x' = (J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y', \\ (-J_y - y'''\dot{u}^3 - 3y''\dot{u}\ddot{u})/y' = (J_x - x'''\dot{u}^3 - 3x''\dot{u}\ddot{u})/x'. \end{cases}$$

Add the equations above together to obtain

$$-J_x/x' - J_y/y' = J_x/x' + J_y/y',$$

which is obviously not true.

It is easy to show that the VSC_1 introduced in section 3.1 is also divided into two parts on the VLS (see Fig.2). Besides VSC_1, there are two kinds of *velocity switching curves* VSC_2 and VSC_3.

Since we assume the tool path to be a piecewise C^2 curve, there may exist discontinuities for x''' or y'''. From (14) (15), they will cause discontinuities of the VLS along certain curves, which are called VSC_2. Because we assume that each segment of the piecewise parametric curve is infinitely differentiable, the discontinuities for x''' or y''' can only occur in the nodes or connection points of the piecewise parametric curve.

Besides, we call the set of points (in fact, curves) where the integration trajectories are tangent to the VLS to be VSC_3. For $i = 1, j = 2$ or $i = 2, j = 1$, the integration trajectories which are tangent to $f_i = g_j$ are just the solutions of $\dddot{u} = f_i$ or $\dddot{u} = g_j$. Differentiating $f_i - g_j = 0$ with respect to u, and using (1) (2), we obtain:

$$\frac{\partial}{\partial u}(f_i - g_j) + \frac{\ddot{u}}{\dot{u}}\frac{\partial}{\partial \dot{u}}(f_i - g_j) + \frac{\dddot{u}}{\dot{u}}\frac{\partial}{\partial \ddot{u}}(f_i - g_j) = 0.$$

Substituting $\dddot{u} = f_i$ into the equation above, we have:

$$\dot{u}\frac{\partial}{\partial u}(f_i - g_j) + \ddot{u}\frac{\partial}{\partial \dot{u}}(f_i - g_j) + f_i\frac{\partial}{\partial \ddot{u}}(f_i - g_j) = 0. \tag{28}$$

The intersection of (28) and the VLS: $f_i = g_j$ is the VSC_3.

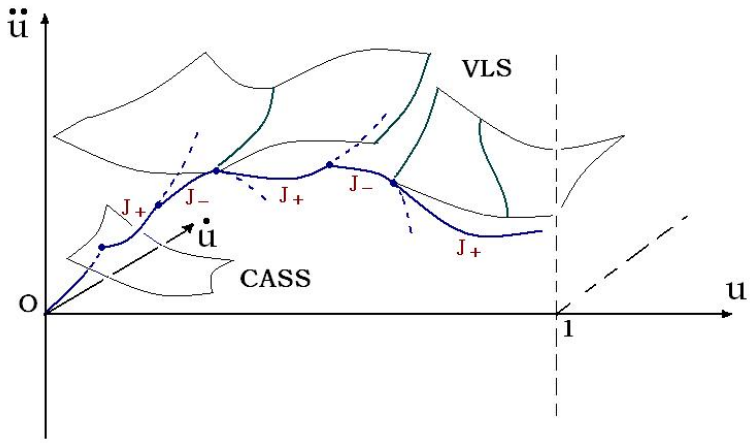


Fig. 3. CASS and integration trajectory of $\dot{u}_f$

3.4 Feed-rate planning algorithm

Our feed-rate planning algorithm is designed under a "greedy rule": using the maximal parametric jerk $\dddot{u}$ as much as possible, that is, we use the minimal parametric jerk only when it has to decelerate. The optimal feed-rate planning problem with acceleration bounds [1, 2, 3, 4] uses a similar rule, but the difference is that we cannot prove our "greedy rule" generates a globally optimal solution for our problem. We will discuss about this in the conclusion.

Firstly, we give the framework of our feed-rate planning algorithm. The specific computational methods in the algorithm will be given later.

Algorithm. Feed-rate planning with jerk constraints.
Input: $\vec{r}(u) = (x(u), y(u)), 0 \le u \le 1$; J_x, J_y.
Output: The integration trajectory for $u \in [0, 1]$.
step0: Let $S = (0, 0, 0)$.
step1: Generate a J_+ trajectory by integrating $\dddot{u} = J_+(u, \dot{u}, \ddot{u})$ forward from S, until the J_+ integration trajectory (if it passes through the CASS first, then change the control axis as previously mentioned) intersects the VLS. Denote the parametric velocity function of the J_+ trajectory obtained by $\dot{u}_1$; if it does not intersect the VLS before $u = 1$, then we have obtained an integration trajectory for $u \in [0, 1]$. Denote its parametric velocity function by $\dot{u}_f$, and go to step4.
step2: Generate a J_- trajectory by integrating $\dddot{u} = J_-(u, \dot{u}, \ddot{u})$ backward from each point on the VSC. If the J_- trajectory starting from point Q on a VSC intersects the J_+ trajectory $\dot{u}_1$ obtained in step1 at point P, then update the integration trajectory between P and Q to be the J_- trajectory from P to Q (see Fig.3).
step3: Let $S = Q$, iterate the process of steps 1-3 until $u = 1$. Denote the parametric velocity function of the whole forward integration trajectory by $\dot{u}_f$.
step4: Generate a backward integration trajectory starting from $(1, 0, 0)$ in a similar

way as step1-step3 until $u = 0$. Denote the parametric velocity function of the backward integration trajectory by $\dot{u}_b$.

step5: Connect the two integration trajectories of $\dot{u}_f$ and $\dot{u}_b$ by J_- trajectories. We obtain a complete integration trajectory for $u \in [0, 1]$.

Remark. The "greedy" rule is used in step 2. In step 1, the J_+ integration trajectory $\dot{u}_1$ meets the VLS at a point. If we continue to use the same jerk, $\dot{u}_1$ will pass through the VLS and violate the jerk limits. In other words, we must decelerate before this happens. According to the definition, the integration trajectory can meet the VSC at a point in a VSC. That is why we generate a J_- integration trajectory starting from a point in the next VSC and try to use this trajectory as the decelerate part of the whole trajectory. In other words, we use the maximal parametric jerk to accelerate as long as possible and then use the minimal parametric jerk to decelerate so that the jerk limits (VLS) are not violated.

Concrete computational methods of step2 and step5 in our algorithm are given below. We will treat step5 first.

For step5, because the control axes of the forward and backward integration trajectories may have been switched for several times, $\dot{u}_f$ and $\dot{u}_b$ are both piecewise-analytic functions (see Fig.3). We need to traverse and choose each analytic segment of the forward and backward integration trajectories respectively, and to connect these two segments by a J_- trajectory if there exists such a solution (if we connect them by a J_+ trajectory, the two segments which are connected can only be J_- trajectories, this will violate our "greedy rule"). After choosing one segment in $\dot{u}_f$ and $\dot{u}_b$ respectively, there are two cases:

1) The J_- trajectory for connection does not pass through CASS. Assume the J_- trajectory starts from point $(u_1, \dot{u}_f(u_1), \ddot{u}_f(u_1))$ on the forward trajectory to point $(u_2, \dot{u}_b(u_2), \ddot{u}_b(u_2))$ on the backward trajectory in the $u-\dot{u}-\ddot{u}$ space. From (23) (24) or (26) (27), the integration constants of the J_- trajectory can be expressed as $C_1(u, \dot{u}, \ddot{u}), C_2(u, \dot{u}, \ddot{u})$. We need to solve the following algebraic equation system

$$\begin{cases} C_1(u_1, \dot{u}_f(u_1), \ddot{u}_f(u_1)) = C_1(u_2, \dot{u}_b(u_2), \ddot{u}_b(u_2)), \\ C_2(u_1, \dot{u}_f(u_1), \ddot{u}_f(u_1)) = C_2(u_2, \dot{u}_b(u_2), \ddot{u}_b(u_2)) \end{cases} \tag{29}$$

to obtain u_1, u_2. Then the integration constants of the J_- connection trajectory are $C_1(\bar{u}_1, \dot{u}_f(\bar{u}_1), \ddot{u}_f(\bar{u}_1))$, $C_2(\bar{u}_1, \dot{u}_f(\bar{u}_1), \ddot{u}_f(\bar{u}_1))$, where $\bar{u}_1$ is a solution of equation (29). Then we obtain the J_- trajectory for the connection in step5.

2) The J_- trajectory for connection passes through an CASS. Now the expressions of the J_- trajectory and its integration constants are different in the two sides of the CASS. Suppose the left side is controlled by $j_x = -J_x$ and the right side is controlled by $j_y = -J_y$. Denote the integration constants of the $j_x = -J_x$ trajectory by C_1^x, C_2^x and the integration constants of $j_y = -J_y$ trajectory by C_1^y, C_2^y. Assume the J_- trajectory for connection passes through the CASS at the point $(u_c, \dot{u}_c, \ddot{u}_c)$, and it starts from the point $(u_l, \dot{u}_f(u_l), \ddot{u}_f(u_l))$ on the forward trajectory to the point $(u_r, \dot{u}_b(u_r), \ddot{u}_b(u_r))$ on the backward trajectory. Then,

we need to solve the following algebraic equation system

$$\begin{cases} f_1(u_c,\dot{u}_c,\ddot{u}_c) = f_2(u_c,\dot{u}_c,\ddot{u}_c), \\ C_1^x(u_l,\dot{u}_f(u_l),\ddot{u}_f(u_l)) = C_1^x(u_c,\dot{u}_c,\ddot{u}_c), \\ C_2^x(u_l,\dot{u}_f(u_l),\ddot{u}_f(u_l)) = C_2^x(u_c,\dot{u}_c,\ddot{u}_c), \\ C_1^y(u_r,\dot{u}_b(u_r),\ddot{u}_b(u_r)) = C_1^y(u_c,\dot{u}_c,\ddot{u}_c), \\ C_2^y(u_r,\dot{u}_b(u_r),\ddot{u}_b(u_r)) = C_2^y(u_c,\dot{u}_c,\ddot{u}_c) \end{cases} \tag{30}$$

to obtain $u_l, u_c, u_r, \dot{u}_c, \ddot{u}_c$ and the two sets of integration constants of the J_- trajectory for the connection: $C_1^x(u_c,\dot{u}_c,\ddot{u}_c), C_2^x(u_c,\dot{u}_c,\ddot{u}_c)$ and $C_1^y(u_c,\dot{u}_c,\ddot{u}_c), C_2^y(u_c,\dot{u}_c,\ddot{u}_c)$. It is similar to deal with the case when the J_- trajectory passes through the CASS more than once.

In general, the solutions of the above equation systems are finite. We just need to compare these solutions to get an optimal one according to the machining time in (6).

For step2, there are two cases:

1) Point Q is on a VSC_1 or a VSC_2. If $u = u_0$ at Q, we assume the coordinate of Q is $(u,\dot{u},\ddot{u}) = (u_0, b, c)$ and denote the expression of VSC_1 or VSC_2 on the u_0 section by $h_1(\dot{u},\ddot{u}) = 0$ as previously mentioned. Assume $u = a$ at P. Then the coordinate of P is $(a, \dot{u}_1(a), \ddot{u}_1(a))$. The integration constants of the J_- trajectory are $C_1(u,\dot{u},\ddot{u}), C_2(u,\dot{u},\ddot{u})$ as above. We just need to solve the following algebraic equation system

$$\begin{cases} C_1(u_0,b,c) = C_1(a,\dot{u}_1(a),\ddot{u}_1(a)), \\ C_2(u_0,b,c) = C_2(a,\dot{u}_1(a),\ddot{u}_1(a)), \\ h_1(b,c) = 0 \end{cases} \tag{31}$$

to obtain a, b, c. If the equation system has more than one solutions or there are more VSCs, we should choose the solution with maximal parametric a according to our "greedy rule" . We deal with equation systems occurring later in the same way. The integration constants of the J_- trajectory can be computed by $C_1(u_0,b,c), C_2(u_0,b,c)$. Then we obtain the J_- trajectory in step2.

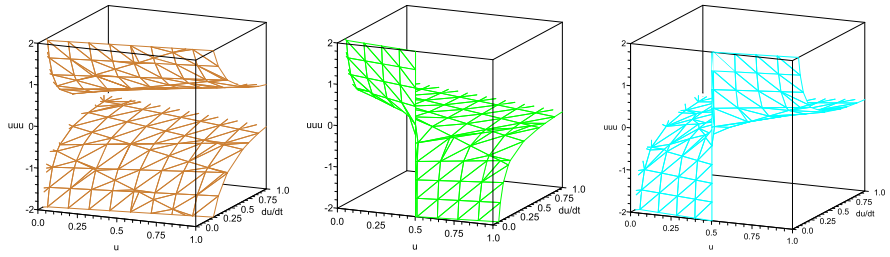
2) Point Q is on a VSC_3. Assume the coordinate of Q to be (d_0, b_0, c_0). From (28), denote the VSC_3 by $\{(u,\dot{u},\ddot{u}) | h_2(u,\dot{u},\ddot{u}) = 0, h_3(u,\dot{u},\ddot{u}) = 0\}$. Assume $u = a_0$ at P. Then the coordinate of P is $(a_0, \dot{u}_1(a_0), \ddot{u}_1(a_0))$. The integration constants of the J_- trajectory can also be expressed as $C_1(u,\dot{u},\ddot{u})$ and $C_2(u,\dot{u},\ddot{u})$. We just need to solve the following algebraic equation system

$$\begin{cases} C_1(d_0,b_0,c_0) = C_1(a_0,\dot{u}_1(a_0),\ddot{u}_1(a_0)), \\ C_2(d_0,b_0,c_0) = C_2(a_0,\dot{u}_1(a_0),\ddot{u}_1(a_0)), \\ h_2(d_0,b_0,c_0) = 0, \\ h_3(d_0,b_0,c_0) = 0 \end{cases} \tag{32}$$

to obtain a_0, d_0, b_0, c_0. The integration constants of the J_- trajectory are $C_1(d_0,b_0,c_0)$ and $C_2(d_0,b_0,c_0)$. If the J_- trajectory passes through a CASS between P and Q, we use the method mentioned above for case 2) of step5 to deal with this situation.

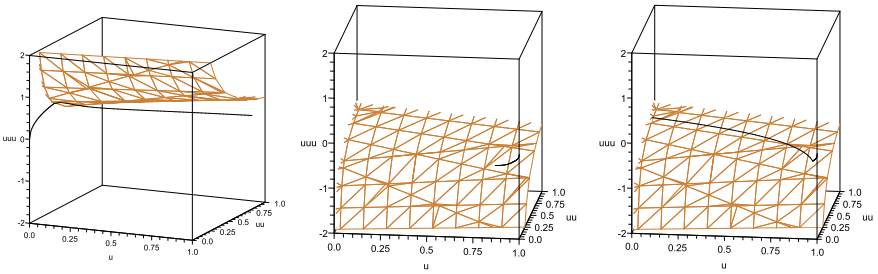
So far, we have obtained a complete integration trajectory, its parametric velocity function satisfies (7) (8) and our "greedy rule".

With the algorithm, we obviously obtain a unique and optimal feed-rate planning along specific tool paths with jerk constraints on each axis under the "greedy rule".



(a) VLS (b) CASS (c) CASS

Fig. 4. VLS and CASS



(a) (b) (c)

Fig. 5. Forward integration trajectory: (a); backward integration trajectory: (b) and (c)

4 An example

We use the following example to illustrate our algorithm:

$$\vec{r}(u) = (u, u^2), 0 \le u \le 1,$$

$$J_x = J_y = 1.$$

The algorithm has the following steps:

1) Firstly, compute the VLS and the CASS :

Fig.4(a): VLS $J_-(u, \dot{u}, \ddot{u}) = J_+(u, \dot{u}, \ddot{u})$;

Fig.4(b): CASS of maximal parametric jerk $g_1(u, \dot{u}, \ddot{u}) = g_2(u, \dot{u}, \ddot{u})$;

Fig.4(c): CASS of minimal parametric jerk $f_1(u, \dot{u}, \ddot{u}) = f_2(u, \dot{u}, \ddot{u})$.

Then, compute the three kinds of VSC :

VSC_1: $\{(0, \dot{u}, \ddot{u}) \mid 1 - 6\dot{u}\ddot{u} = 0\}$ and $\{(0, \dot{u}, \ddot{u}) \mid 1 + 6\dot{u}\ddot{u} = 0\}$;

VSC_3: $\{(u, \dot{u}, \ddot{u}) \mid 1 - 6\dot{u}\ddot{u} + 2u = 0, 4\dot{u} - 3\ddot{u}^2 = 0\}$ and $\{(u, \dot{u}, \ddot{u}) \mid 1 + 6\dot{u}\ddot{u} + 2u = 0, 4\dot{u} + 3\ddot{u}^2 = 0\}$;

VSC_2 dose not exist here.

2) Generate a J_+ trajectory forward from $(0, 0, 0)$. The trajectory is controlled by $j_x = J_x$ in the beginning, then intersects the CASS $g_1 = g_2$ at $u = 0.05$ and switches to the control of $j_y = J_y$. It will not intersect the VLS or the CASS before reaching $u = 1$ (Fig.5(a)). We

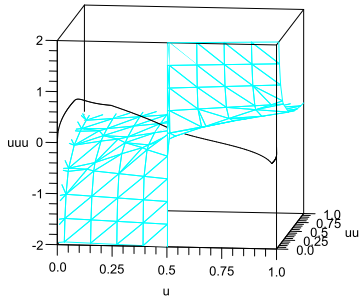


Fig. 6. Connect the forward and backward trajectories

obtain the parametric velocity function of the forward integration trajectory (Fig.5(a)):

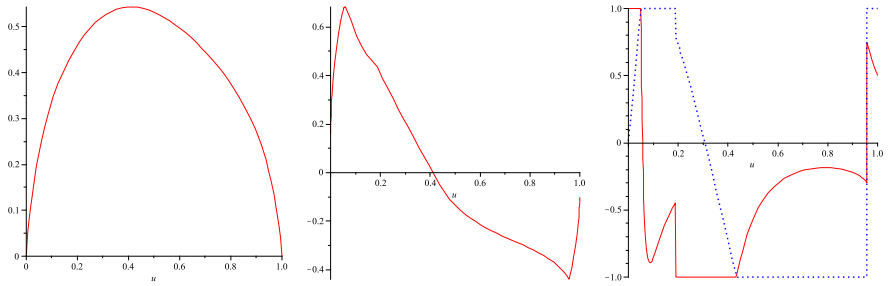
$$\dot{u}_f = \begin{cases} \frac{1}{2}(6u)^{\frac{2}{3}}, & 0 \le u \le 0.05; \\ \frac{1}{4u}((\sqrt{9u^4 - 0.5625 \cdot 10^{-2}u^2 + 0.5625 \cdot 10^{-5}} + 3u^2 - 0.9375 \cdot 10^{-3})^{\frac{2}{3}} \\ +(\sqrt{9u^4 - 0.5625 \cdot 10^{-2}u^2 + 0.5625 \cdot 10^{-5}} - 3u^2 + 0.9375 \cdot 10^{-3})^{\frac{2}{3}} \\ -0.0168), & 0.05 \le u \le 1. \end{cases}$$

3) Generate a J_+ trajectory backward from $(1, 0, 0)$. The trajectory $\dot{u}_b$ is controlled by $j_y = J_y$ in the beginning. It intersects the the CASS $g_1 = g_2$ at $u = 0.9253$, then switches to the control of $j_x = J_x$. Then it intersects the VLS at $u = 0.8104$ (Fig.5(b)). Now, we need to execute step2 of the algorithm by solving the equation system (31). The only solution is a $j_y = -J_y$ trajectory from the point $(u, \dot{u}, \ddot{u}) = (0, 0.7528, -0.2214)$ on the VSC_1 at $u = 0$ to the point $(0.9561, 0.1679, -0.4486)$ on trajectory $\dot{u}_b$ (Fig.5(c)). Then we obtain the the backward integration trajectory:

$$\dot{u}_b = \begin{cases} \frac{1}{4u}(6(1-u^2))^{\frac{2}{3}}, & 0.9561 \le u \le 1; \\ \frac{0.3211}{u}(2\sin(\frac{2}{3}\arccos(2.0607u^2 - 1) + \frac{1}{6}\pi) - 1), & 0 \le u \le 0.9561. \end{cases}$$

4) Connect the integration trajectories of $\dot{u}_f$ and $\dot{u}_b$ by J_- trajectories. Solving the previous equation system (30), the only solution is that the J_- trajectory connects the integration trajectories of the second segment of $\dot{u}_f$ and the first segment of $\dot{u}_b$, and it intersects the CASS at $u = 0.4336$ (see Fig.6). It is controlled by $j_x = -J_x$ for $u \in [0.1893, 0.4336]$ and by $j_y = -J_y$ for $u \in [0.4336, 0.9580]$. Then the parametric velocity function of the complete integration trajectory is (see Fig.7(a)):

$$\dot{u} = \begin{cases} \frac{1}{2}(6u)^{\frac{2}{3}}, & 0 \le u \le 0.05; \\ \frac{1}{4u}((\sqrt{9u^4 - 0.5625 \cdot 10^{-2}u^2 + 0.5625 \cdot 10^{-5}} + 3u^2 - 0.9375 \cdot 10^{-3})^{\frac{2}{3}} \\ +(\sqrt{9u^4 - 0.5625 \cdot 10^{-2}u^2 + 0.5625 \cdot 10^{-5}} - 3u^2 + 0.9375 \cdot 10^{-3})^{\frac{2}{3}} \\ -0.0168), & 0.05 \le u \le 0.1893; \\ 0.5440(2\sin(\frac{2}{3}\arccos(2.6437u - 1.0877) + \frac{1}{6}\pi) - 1), & 0.1893 \le u \le 0.4336; \\ \frac{0.3124}{u}(2\sin(\frac{2}{3}\arccos(2.1476u^2 - 1.0869) + \frac{1}{6}\pi) - 1), & 0.4336 \le u \le 0.9580; \\ \frac{1}{4u}(6(1-u^2))^{\frac{2}{3}}, & 0.9580 \le u \le 1. \end{cases}$$



(a) $\dot{u}$ (b) $\ddot{u}$ (c) Solid: j_x; Dotted: j_y

Fig. 7. parametric velocity, parametric acceleration and x, y jerks in u

From Fig.7(b), the acceleration is continuous and from Fig.7(c), the jerk control is "Bang-Bang". The five segments of the integration trajectory are respectively controlled by J_x, J_y, $-J_x$, $-J_y$, and J_y in the u_+ direction.

5 Conclusion

Our algorithm in this paper is based on the idea of "greedy rule", that is, to use maximal parametric jerk as much as possible. Under the "greedy rule", we plan the feed-rate with a "Bang-Bang" control strategy and give an optimal jerk confined solution.

It is a significant open problem that how to prove the algorithm is globally optimal (or not) without the "greedy rule". The main difficulty is that, for second-order differential equations, we have no results similar to the "comparison theorem" for first-order differential equations (p.25, [7]).

Besides, in CNC real-time interpolator after the feed-rate planning, we need to compute a "reference point" indicating the commanded machine position. If u_k is the reference point parameter value at time $k\Delta t$, the equation that determines this value is:

$$k\Delta t = \int_0^{u_k} \frac{du}{\dot{u}}.$$

From (19), it admits closed-form integration in our algorithm. This helps to enhance CNC machining accuracy.

References

[1] J.E. Bobrow, S. Dubowsky, J.S. Gibson. Time-optimal control of robotic manipulators along specified paths. *Int J Robotics Res*, 4(3), 3-17, 1985.

[2] Z. Shiller, H.H. Lu. Robust computation of path constrained time optimal motions. *IEEE Inter. Conf. on Robotics and Automation*, Cincinnati, OH, 144-149, 1990.

[3] S.D. Timar, R.T. Farouki, T.S. Smith, C.L. Boyadjieff. Algorithms for time-optimal control of CNC machines along curved tool paths. *Robotics and Computer-Integrated Manufacturing*, 21, 37-53, 2005.

[4] S.D. Timar, R.T. Farouki. Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds. *Robotics and Computer-Integrated Manufacturing*, 23(2), 563-579, 2007.

[5] C.M. Yuan, X.S. Gao. Time-optimal interpolation of CNC machines along parametric path with chord error and tangential acceleration bounds. *MM-preprints*, 29, 165-188, 2010.

[6] J. Dong, J.A. Stori. A generalized time-optimal bi-directional scan algorithm for constrained feedrate optimization. *ASME Journal of Dynamic Systems, Measurement, and Control*, 128, 379-390, 2006.

[7] G. Birkhoff, G.C. Rota. *Ordinary differential equations*. John Wiley, New York, 1969.

[8] K. Erkorkmaz, Y. Altintas. High speed CNC system design Part I: jerk limited trajectory generation and quintic spline interpolation. *International Journal of Machine Tools and Manufacture*, 41, 1323-1345, 2001.

[9] S. Macfarlane, E.A. Croft. Jerk-bounded manipulator trajectory planning: design for real-time applications. *IEEE Transactions on Robotics and Automation*. 19, 42-52, 2003.

[10] S.H. Nam, M.Y. Yang. A study on a generalized parametric interpolator with real-time jerklimited acceleration. *Computer-Aided Design*, 36, 27-36, 2004.

[11] M.T. Lin, M.S. Tsai, H.T. Yau. Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm. *International Journal of Machine Tools and Manufacture*, 47(15), 2246-2262, 2007.

[12] M.M. Emami, B. Arezoo. A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length. *Computer-Aided Design*, 42, 625-632, 2010.

[13] J.Y. Lai, K.Y. Lin, S.J. Tseng, W.D. Ueng. On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk *Int J Adv Manuf Technol*, 37(1-2), 104-121,2008

[14] J. Dong, P.M. Ferreiraa, J.A. Stori. Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture*, 47, 1941-1955, 2007.