Cubic Spline Trajectory Generation with Axis Jerk and Tracking Error Constraints

Ke Zhang, Jian-Xin Guo, Xiao-Shan Gao
KLMM, Institute of Systems Science, Chinese Academy of Sciences

Abstract

This paper presents a cubic spline trajectory generation algorithm that produces continuous position, velocity, and acceleration profiles for 3-axis CNC machines with confined axis jerk and tracking error. A series of reference knots are connected using cubic spline functions in time t for constructing axis trajectories. They are generated in such a way that continuity up to the second derivative is preserved along the overall composite curve. For CNC machines governed by a typical PID controller, the tracking error is also constrained as well as the velocity, acceleration, and jerk for each axis. The time intervals between each pair of adjacent knots are scheduled such that the machining time is minimized subject to all the constraints. Simulation results are presented to illustrate the application of the algorithm.

Keywords. Cubic spline trajectory, confined jerk, PID controller, tracking error, minimum time.

1 Introduction

The CNC of the machine tool has to create a feedrate profile to follow a given tool path generated by CAM software. Generally, G codes or spline curves are employed to describe the tool path. The aim of feedrate planning is to find an optimized feedrate profile which makes best use of the kinematical characteristics of the machine.

Feedrate planning can be solved using different methods and was first studied in robotics. Bobrow et al. [1] gave an algorithm to determine the minimum time motion for a robot manipulator along a specific path (smooth parametric curve) with actuator torque constraints. Neuman and Tourassis [2] introduced an inherently discrete time dynamic model for robotic manipulators. Tan and Potts [3] used the model of [2] in minimum time trajectory planning. The joint velocity, torque and jerk constraints are incorporated into the model. Macfarlane and Croft [4] 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.

The trajectory can also be planned by connecting a sequence of predefined via-points. Lin et al. [5] developed an algorithm to schedule the time intervals between each pair of adjacent via-points such that the total traveling time is minimized subject to the physical constraints on joint velocities, accelerations, and jerks. Cubic spline functions in time t are used for constructing joint trajectories. The physical constraints can be expressed as inequality constraints of time intervals. Then it becomes a nonlinear programming problem with the sum of time intervals as the objective function. Gasparetto and Zanotto [6] generalized the method in [5] by adding a term proportional to the integral of the squared jerk to the objective function. This new added term ensures that the resulting trajectory is smooth enough.

The problem has also received a significant amount of attention in the CNC machining literature. Altintas and Erkorkmaz [7] presented a quintic spline trajectory generation algorithm to connect a series of reference knots that produces continuous position, velocity, and acceleration profiles. They also developed a feedrate optimization technique in [8] for minimizing the cycle time in machining spline tool paths with axis velocity, torque and jerk limits. Feed modulation is achieved by manipulating segment durations which define the overall minimum jerk feed profile. Sencer et al. [9] expressed the variation of the feed along the five-axis tool-path in a cubic B-spline form. The feedrate is planned by iteratively modulating the feed control points of the B-spline to maximize the feed along the tool path without violating the programmed feed and the drives' physical limits. 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 feedrate trajectory generation through tangential jerk-confined acceleration profiles for the parametric curves. Lai et al. [11] further proposed a method using backtracking at each step with chord error, feedrate, acceleration and jerk limits. 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 with tangential jerk limit for the NURBS curves. Lee et al. [13] proposed an off-line feedrate scheduling method of CNC machines constrained by chord tolerance, acceleration and jerk limitations, which is realized as a pre-processor and releases the computational burden in real-time task. Zhang et al. [14] proposed a multi-period turning method to improve the feedrate at the junctions using the linear acceleration/deceleration mode for the G1 tool paths, which utilizes the maximal acceleration capabilities of the NC machine while satisfying the machining precision.

Most of the feedrate planning algorithms cannot find a real time-optimal solution since the mathematical description of this problem is not trivial when considering the limitation on each axis. Following the method in [1], Timar et al. [15] gave a piecewise-analytic expression of the time-optimal feedrate function with acceleration bounds on x, y, z axes. Zhang et al. [16] simplified the method in [15] for quadratic B-splines and realized real-time manufacturing on industrial CNC machines. Yuan et al. [17] provided a time-optimal feedrate planning method with tangential acceleration and chord error bounds. Zhang et al. [18] tried to extend the above methods to the case of jerk bounds and gave a greedy feedrate planning algorithm.

However, the computational expense of the analytical methods are high, especially for tool paths described by high order spline curves. Some discrete methods for the feedrate planning have been proposed and developed. Dong et al. [19] gave a discrete greedy algorithm for the problem with constraints of parametric velocity, acceleration and jerk based on a series of single variable optimization subproblems. Beudaert et al. [20] presented an algorithm that iteratively smoothes the joint motions to raise the real feedrate starting from a given 5-axis tool paths. They also proposed a velocity profile optimization method in [21] by intersecting all the constraints due to the drives in an iterative algorithm for linear interpolation (G1) and NURBS interpolation.

No matter which algorithm is adopted for feedrate planning, there will be tracking errors in each axis, due to the inherent machine and feedback controller dynamics. A number of different approaches have been proposed by various researchers that attempt to maintain high tracking accuracy. Renton and Elbestawi [22] developed a method which results in reduced cycle time or reduced path error. A servo loop control law is developed that uses the axis performance envelope as well as instantaneous position, velocity, and acceleration information of the target path and machine axis to improve servo performance in the presence of disturbances. Dong and Stori [23] presented an algorithm with a structure modeled closely after that of [22] for generating a minimum time feedrate profile subject to the velocity, acceleration and contour error constraints. Altintas and Erkorkmaz [24] provided a systematic approach for designing a control law which provides a high tracking bandwidth as well as adequate disturbance rejection and parameter variation robustness, in order to minimize the following errors in each axis. Ernesto and Farouki [25] solved the problem of compensating for inertia and damping of the machine axes by a priori modifications to the commanded path geometry for CNC machines governed by typical feedback controllers. Conway et al. [26] presented a cross-coupled control scheme on a 3-axis CNC mill, which is based on essentially exact contour error computations for free-form curved paths. Huo et al. [27] proposed a generalized Taylor series expansion error compensation (GTSEEC), which is capable of compensating for the contour errors of arbitrary two-dimensional contours.

This work presents a scheme for generating C^2 cubic spline tool paths, using a nonlinear programming strategy that produces continuous position, velocity, and acceleration profiles for 3-axis CNC machines. Unlike most of the trajectory generation methods, the tool path is directly designed as a function in time t. A series of reference knots in the machine coordinate system are predefined. The objective is to connect the reference knots by using cubic spline functions in time t to construct axis trajectories. The velocity, acceleration and jerk constraints for each axis are considered. For the PID controller, the tracking error on each axis is approximated by a linear particular solution, which can be easily added to the constraints of the original problem. Then an algorithm is developed to schedule the time intervals between each pair of adjacent knots such that the machining time is minimized subject to all the constraints. Finally, the minimum time cubic spline trajectory generation is completed without a re-interpolation since the trajectory is expressed as a function in time t. The idea of using cubic splines is from [5]. The main contribution of this paper is to handle the tracking error constraints effectively.

The rest of this paper is organized as follows. Section 2 gives the formulation of the cubic spline trajectories. Section 3 gives the constraints and minimum time model. Section 4 gives the simulation results. Section 5 concludes the paper.

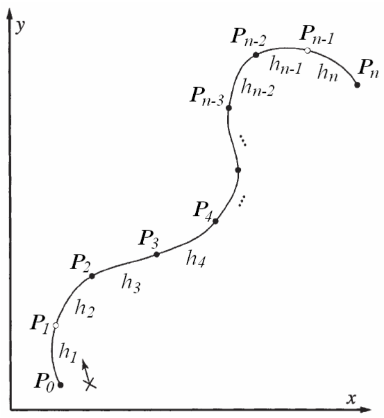

Figure 1: Composite cubic spline defined by $n+1$ reference knots.

Table 1: Nomenclature

$n+1$	number of reference knots
P_i	ith reference knot
$\vec{p}_i$	position vector of ith reference knot
$\vec{v}_i$	velocity vector at ith reference knot
t_i	time instance at ith reference knot
h_i	length of ith time interval
$\vec{C}_i(t)$	axes position function on ith time interval
$\dot{\vec{C}}_i(t)$	axes velocity function on ith time interval
$\ddot{\vec{C}}_i(t)$	axes acceleration function on ith time interval
$\dddot{\vec{C}}_i$	axes jerk on ith time interval
$\vec{V}_{max}$	axes velocity limits
$\vec{A}_{max}$	axes acceleration limits
$\vec{J}_{max}$	axes jerk limits
$\vec{e}(t)$	axes tracking error function
$\vec{\tilde{e}}(t)$	axes approximate tracking error function
$\vec{E}_{max}$	axes tracking error limits

2 Formulation of the trajectories

Suppose a series of $n+1$ reference knots $P_0, P_1, \ldots, P_n$ which describe the tool movements are already generated in the upper level. The knots are considered to be intensive enough for the accuracy. Now the position of P_1 and P_{n-1} are set to be free to give enough freedom for solving this problem. The objective of cubic spline tool path generation is to connect the $n+1$ reference knots with n cubic splines (see Fig. 1) which are functions in time t. The freedom of these cubic splines will be fully used such that the continuity up to the second derivative is preserved along the overall composite curve. Then a cubic spline is fit between successive knots such that position, as well as first and second derivative boundary conditions are met at both ends of the spline. Compared with other tool path parameters, there are two advantages for choosing time t to be the tool path parameter: the expressions of all the constraints are simpler and a re-interpolation is not needed after the feedrate planning.

The position vectors of the reference knots P_i are denoted as $\vec{p}_i, i = 0, \ldots, n$ (where $\vec{p}_1$ and $\vec{p}_{n-1}$ are free as mentioned above). The velocity and acceleration at the initial and terminal positions P_0, P_n are specified to be $\vec{v}_0, \vec{a}_0, \vec{v}_n, \vec{a}_n$ respectively.

Let

$$h_i = t_i - t_{i-1}, i = 1, \ldots, n,$$

where $t_0 < t_1 < \ldots < t_n$ is the sequence of time instants corresponding to the $n+1$ reference knots. Suppose the velocity vectors at reference knots $P_1, P_2, \ldots, P_{n-1}$ are $\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_{n-1}$ respectively. Using Hermite interpolation on each interval $[t_{i-1}, t_i], i = 1, \ldots, n$, the cubic spline can be expressed as:

$$\begin{aligned} \vec{C}_i(t) =& (\frac{t-t_i}{h_i})^2(2\frac{t-t_{i-1}}{h_i}+1)\vec{p}_{i-1} + (\frac{t-t_i}{h_i})^2(t-t_{i-1})\vec{v}_{i-1} \\ &+ (\frac{t-t_{i-1}}{h_i})^2(2\frac{t_i-t}{h_i}+1)\vec{p}_i + (\frac{t-t_{i-1}}{h_i})^2(t-t_i)\vec{v}_i. \end{aligned} \tag{1}$$

Then, on interval $[t_{i-1}, t_i]$, the functions of velocity, acceleration and jerk are respectively:

$$\begin{aligned} \dot{\vec{C}}_i(t) &= \frac{6}{h_i^3}(t-t_i)(t-t_{i-1})\vec{p}_{i-1} + \frac{t-t_i}{h_i^2}(3(t-t_i)+2h_i)\vec{v}_{i-1} \\ &\quad -\frac{6}{h_i^3}(t-t_i)(t-t_{i-1})\vec{p}_i + \frac{t-t_{i-1}}{h_i^2}(3(t-t_{i-1})-2h_i)\vec{v}_i, \end{aligned} \tag{2}$$

$$\begin{aligned} \ddot{\vec{C}}_i(t) &= \frac{6}{h_i^3}(2(t-t_i)+h_i)\vec{p}_{i-1} + \frac{6}{h_i^2}(t-t_i+\frac{h_i}{3})\vec{v}_{i-1} \\ &\quad -\frac{6}{h_i^3}(2(t-t_{i-1})-h_i)\vec{p}_i + \frac{6}{h_i^2}(t-t_{i-1}-\frac{h_i}{3})\vec{v}_i, \end{aligned} \tag{3}$$

$$\dddot{\vec{C}}_i = \frac{12}{h_i^3}(\vec{p}_{i-1}-\vec{p}_i) + \frac{6}{h_i^2}(\vec{v}_{i-1}+\vec{v}_i). \tag{4}$$

In (1), (2), (3), and (4), $h_i, i = 1, \ldots, n$ and $\vec{v}_i, i = 1, \ldots, n-1$ are undetermined. For the second order continuity, the following equations must hold:

$$\ddot{\vec{C}}_i(t_i) = \ddot{\vec{C}}_{i+1}(t_i), i = 1, \ldots, n-1. \tag{5}$$

Using (3), (5) is equivalent to

$$h_{i+1}\vec{v}_{i-1} + 2(h_i + h_{i+1})\vec{v}_i + h_i\vec{v}_{i+1} = -\frac{3h_{i+1}}{h_i}\vec{p}_{i-1} + 3(\frac{h_{i+1}}{h_i} - \frac{h_i}{h_{i+1}})\vec{p}_i + \frac{3h_i}{h_{i+1}}\vec{p}_{i+1}. \tag{6}$$

The boundary conditions $\ddot{\vec{C}}_1(t_0) = \vec{a}_0, \ddot{\vec{C}}_n(t_n) = \vec{a}_n$ are equivalent to

$$\vec{p}_1 = \vec{p}_0 + \frac{2h_1}{3}\vec{v}_0 + \frac{h_1^2}{6}\vec{a}_0 + \frac{h_1}{3}\vec{v}_1, \tag{7}$$

$$\vec{p}_{n-1} = \vec{p}_n - \frac{2h_n}{3}\vec{v}_n + \frac{h_n^2}{6}\vec{a}_n - \frac{h_n}{3}\vec{v}_{n-1}. \tag{8}$$

Using (6), (7), and (8), for each axis component of $\vec{v}_i$ (for brevity, omit the arrow of a vector to denote one of its component), the velocities $v_1, \ldots, v_{n-1}$ will be determined by the time intervals $h_1, \ldots, h_n$ according to

$$H_1(v_1, \ldots, v_{n-1})^T = H_2, \tag{9}$$

where

$$H_1 = \begin{pmatrix} 2h_1 + h_2 + \frac{h_1^2}{h_2} & h_1 & & & & \\ h_3 + \frac{h_1 h_3}{h_2} & 2(h_2 + h_3) & h_2 & & & \\ & h_4 & 2(h_3 + h_4) & h_3 & & \\ & \ddots & \ddots & \ddots & & \\ & & h_{n-2} & 2(h_{n-3} + h_{n-2}) & h_{n-3} & \\ & & & h_{n-1} & 2(h_{n-2} + h_{n-1}) & h_{n-2} + \frac{h_{n-2}h_n}{h_{n-1}} \\ & & & & h_n & h_{n-1} + 2h_n + \frac{h_n^2}{h_{n-1}} \end{pmatrix}_{(n-1)^2}$$

$$H_2 = \begin{pmatrix} -\frac{3h_1}{h_2}p_0 + (h_2 - \frac{2h_1^2}{h_2})v_0 + (\frac{h_1h_2}{2} - \frac{h_1^3}{2h_2})a_0 + \frac{3h_1}{h_2}p_2 \\ -\frac{3h_3}{h_2}p_0 - \frac{2h_1h_3}{h_2}v_0 - \frac{h_1^2h_3}{2h_2}a_0 + 3(\frac{h_3}{h_2} - \frac{h_2}{h_3})p_2 + \frac{3h_2}{h_3}p_3 \\ -\frac{3h_4}{h_3}p_2 + 3(\frac{h_4}{h_3} - \frac{h_3}{h_4})p_3 + \frac{3h_3}{h_4}p_4 \\ \vdots \\ -\frac{3h_{n-2}}{h_{n-3}}p_{n-4} + 3(\frac{h_{n-2}}{h_{n-3}} - \frac{h_{n-3}}{h_{n-2}})p_{n-3} + \frac{3h_{n-3}}{h_{n-2}}p_{n-2} \\ -\frac{3h_{n-1}}{h_{n-2}}p_{n-3} + 3(\frac{h_{n-1}}{h_{n-2}} - \frac{h_{n-2}}{h_{n-1}})p_{n-2} + \frac{3h_{n-2}}{h_{n-1}}p_n - \frac{2h_{n-2}h_n}{h_{n-1}}v_n + \frac{h_{n-2}h_n^2}{2h_{n-1}}a_n \\ -\frac{3h_n}{h_{n-1}}p_{n-2} + \frac{3h_n}{h_{n-1}}p_n + (h_{n-1} - \frac{2h_n^2}{h_{n-1}})v_n + (\frac{h_n^3}{2h_{n-1}} - \frac{h_{n-1}h_n}{2})a_n \end{pmatrix}_{(n-1)\times 1}$$

H_1 is a tridiagonal matrix. It is easy to show that H_1 is equivalent to a strictly diagonal-dominant matrix, when using the first row of H_1 to eliminate the first element in the second row (symmetrically to the last two rows), since $h_i, i = 1, \ldots, n$ are all positive. Then H_1 is nonsingular. When the time intervals $h_1, \ldots, h_n$ are determined, the velocity vectors $\vec{v}_i, i = 1, \ldots, n-1$ are uniquely determined according to (9), so do the expressions of $\vec{C}_i(t), i = 1, \ldots, n$. As a consequence, the only undetermined variables are $h_i, i = 1, \ldots, n$. The next section will show how to determine $h_1, \ldots, h_n$ under two kinds of constraints.

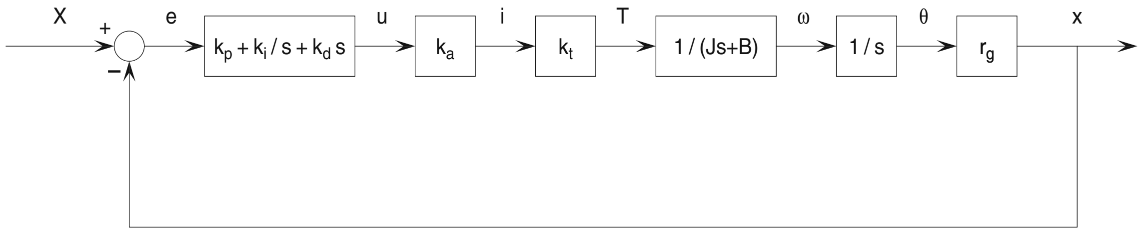

Figure 2: Block diagram for the x-axis drive and PID controller.

3 Constraints and minimum time model

3.1 Drive kinematic constraints

Because of the physical limitations of the drives, the velocity, acceleration, and jerk of each individual drive have to be constrained. All the kinematic constraints are set to be symmetrical and denoted as $\vec{V}_{max}, \vec{A}_{max}, \vec{J}_{max}$ respectively.

Now the discussion will be focus on one axis (the analysis for any other axis is exactly the same). Omit the arrows of the symbols defined above to denote an axis component.

From (2), the axis velocity function on each time interval is quadratic in t. Then it just needs to consider the velocity constraints at two ends and one extremal point on each time interval:

$$\begin{cases} |v_i| \leq V_{max}, i = 1, \ldots, n-1, \\ |\dot{C}_i(t_i^*)| \leq V_{max}, when\ t_{i-1} < t_i^* < t_i, i = 1, \ldots, n, \end{cases} \tag{10}$$

where $t_i^* = \frac{3(t_{i-1}+t_i)(p_i-p_{i-1})+h_i((t_{i-1}+2t_i)v_{i-1}+(2t_{i-1}+t_i)v_i)}{6(p_{i-1}-p_i)+3h_i(v_{i-1}+v_i)}$ is the extremal point of $\dot{C}_i(t)$ on $[t_{i-1}, t_i]$.

From (3), the axis acceleration function on each interval is linear in t. It just needs to consider the acceleration constraints at two ends of each interval:

$$|\ddot{C}_i(t_i)| \leq A_{max}, i = 1, \ldots, n-1. \tag{11}$$

From (4), the axis jerk function on each interval is a constant. The jerk constraints become:

$$|\dddot{C}_i| \leq J_{max}, i = 1, \ldots, n. \tag{12}$$

It is obviously that the constraints (10), (11), and (12) are all polynomial inequality constraints about $h_1, \ldots, h_n$ (and $v_1, \ldots, v_{n-1}$) from (2), (3), (4), and (9).

3.2 Tracking error constraints

For commonly used controllers (e.g., PID control) in industrial CNCs, there will be tracking errors in each axis as the closed loop control system is not able to follow the rapidly varying position commands. So it is necessary to reduce the tracking error in each axis, which can improve machining accuracy.

In this paper, the PID control with proportional, integral, derivative gains k_p, k_i, and k_d is considered (see Fig. 2). In fact, the following analysis of tracking error approximation are also appropriate for other types of controllers. Here, only the intrinsic machine dynamics is considered, without cutting forces, external disturbances. Same as above, the discussion is focused on one axis dynamics (take x-axis for example). In Fig. 2, the *tracking error* $e = X - x$ is the difference between the commanded and actual axis locations. The current amplifier k_a converts the actuating signal u into a current i to the motor, which produces a torque T through the motor torque gain k_t. The torque T determines the angular speed ω through the system inertia J and damping B. The motor shaft angle θ, obtained by integration of ω, determines the axis linear position x through the transmission ratio r_g. For brevity, set $K = k_a k_t r_g$ since the three parameters often occur in the form of this product.

The transfer function of the output x and the input X in the Laplace domain can be written as [25, 28, 29],

$$\frac{x(s)}{X(s)} = \frac{K(k_d s^2 + k_p s + k_i)}{Js^3 + (B + Kk_d)s^2 + Kk_p s + Kk_i}. \tag{13}$$

Denote the three poles of the transfer function by λ_1, λ_2 and λ_3, which are assumed to be all different. (13) can also be written as

$$\frac{e(s)}{X(s)} = \frac{Js^3 + Bs^2}{Js^3 + (B + Kk_d)s^2 + Kk_p s + Kk_i}. \tag{14}$$

The equivalent differential equation of (14) in time domain is

$$\dddot{e}(t) + \frac{B + Kk_d}{J}\ddot{e}(t) + \frac{Kk_p}{J}\dot{e}(t) + \frac{Kk_i}{J}e(t) = \dddot{X}(t) + \frac{B}{J}\ddot{X}(t). \tag{15}$$

Remember that the input $X(t)$ is a cubic polynomial on each time interval. Then the right hand side of (15) is a piecewise linear function in time t. As a consequence, the following piecewise linear function in t

$$\hat{e}(t) = (\frac{J}{Kk_i} - \frac{Bk_p}{Kk_i^2})\dddot{X}(t) + \frac{B}{Kk_i}\ddot{X}(t), \tag{16}$$

is a particular solution for the differential equation (15) of $e(t)$. Then, the general solution of (15) with initial conditions $e(0) = \dot{e}(0) = \ddot{e}(0) = 0$ can be expressed as

$$e(t) = \hat{e}(t) + \mu_1 e^{\lambda_1 t} + \mu_2 e^{\lambda_2 t} + \mu_3 e^{\lambda_3 t}, \tag{17}$$

where μ_1, μ_2 and μ_3 are integration constants. For preserving continuity up to the second derivative of $e(t)$, μ_1, μ_2 and μ_3 may be different on different time intervals. Denote them as $\mu_{1,i}$, $\mu_{2,i}$ and $\mu_{3,i}$ respectively on $[t_{i-1}, t_i], i = 1, \ldots, n$. The following equations should hold for $i = 1, \ldots, n - 1$:

$$\begin{cases} e(t_i^-) = e(t_i^+), \\ \dot{e}(t_i^-) = \dot{e}(t_i^+), \\ \ddot{e}(t_i^-) = \ddot{e}(t_i^+). \end{cases} \tag{18}$$

Substituting (17) into (18), the equations become

$$\begin{pmatrix} \mu_{1,i} e^{\lambda_1 t_{i-1}} \\ \mu_{2,i} e^{\lambda_2 t_{i-1}} \\ \mu_{3,i} e^{\lambda_3 t_{i-1}} \end{pmatrix} = \begin{pmatrix} \mu_{1,i-1} e^{\lambda_1 t_{i-1}} \\ \mu_{2,i-1} e^{\lambda_2 t_{i-1}} \\ \mu_{3,i-1} e^{\lambda_3 t_{i-1}} \end{pmatrix} + \begin{pmatrix} 1 & 1 & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}^{-1} \begin{pmatrix} \hat{e}(t_{i-1}^-) - \hat{e}(t_{i-1}^+) \\ \dot{\hat{e}}(t_{i-1}^-) - \dot{\hat{e}}(t_{i-1}^+) \\ 0 \end{pmatrix}, \tag{19}$$

for $i = 1, \ldots, n$, where $\mu_{1,0} = \mu_{2,0} = \mu_{3,0} = \hat{e}(0^-) = \dot{\hat{e}}(0^-) = 0$. Denote

$$\Lambda = \begin{pmatrix} 1 & 1 & 1 \\ \lambda_1 & \lambda_2 & \lambda_3 \\ \lambda_1^2 & \lambda_2^2 & \lambda_3^2 \end{pmatrix}. \tag{20}$$

Now it will be shown how to reduce $e(t)$ by limiting $\hat{e}(t)$ as

$$|\hat{e}(t)| \leq E_{max}, \tag{21}$$

where E_{max} is an adjustable positive number. For the stability of the system, the real parts of λ_1, λ_2, and λ_3 should be negative. Denote $\alpha = \min\{|\Re\lambda_1|, |\Re\lambda_2|, |\Re\lambda_3|\}$ and $\dot{E}_{max} = \frac{BJ_{max}}{Kk_i}$. The length of each time interval h_i has a lower bound since the axis velocities are constrained:

$$h_i \geq \max_{\text{for each axis}} \frac{|p_i - p_{i-1}|}{V_{max}} \triangleq l_i, i = 1, \ldots, n. \tag{22}$$

Denote $h = \min_{i=1,\ldots,n} l_i$. From (17) and (19), it can be easily deduced that on each $[t_{i-1}, t_i]$:

$$\begin{aligned}
|e(t)| &\leq E_{max} + |\mu_{1,i} e^{\lambda_1 t_{i-1}}| + |\mu_{2,i} e^{\lambda_2 t_{i-1}}| + |\mu_{3,i} e^{\lambda_3 t_{i-1}}| \\
&\leq E_{max} + |\mu_{1,i-1} e^{\lambda_1 t_{i-1}}| + |\mu_{2,i-1} e^{\lambda_2 t_{i-1}}| + |\mu_{3,i-1} e^{\lambda_3 t_{i-1}}| + 2||\Lambda^{-1}||_1 (E_{max} + \dot{E}_{max}) \\
&\leq E_{max} + (|\mu_{1,i-1} e^{\lambda_1 t_{i-2}}| + |\mu_{2,i-1} e^{\lambda_2 t_{i-2}}| + |\mu_{3,i-1} e^{\lambda_3 t_{i-2}}|) e^{-\alpha h_{i-1}} + 2||\Lambda^{-1}||_1 (E_{max} + \dot{E}_{max}) \\
&\leq \cdots \\
&\leq E_{max} + 2||\Lambda^{-1}||_1 (E_{max} + \dot{E}_{max})(1 + e^{-\alpha h_{i-1}} + \cdots + e^{-\alpha(h_1 + \cdots + h_{i-1})}) \\
&\leq E_{max} + 2||\Lambda^{-1}||_1 (E_{max} + \dot{E}_{max})(1 + e^{-\alpha h} + \cdots + e^{-(i-1)\alpha h}) \\
&\leq E_{max} + \frac{2||\Lambda^{-1}||_1}{1 - e^{-\alpha h}} (E_{max} + \dot{E}_{max}).
\end{aligned} \tag{23}$$

Then with a proper choice of k_p, k_i and k_d, $e(t)$ can also be suitably constrained by E_{max}. As a consequence, it just needs to add the constraints on $\hat{e}(t)$.

Since $\hat{e}(t)$ is piecewise linear, (21) becomes

$$\begin{cases} |\hat{e}(t_i^-)| \leq E_{max}, i = 1, \ldots, n, \\ |\hat{e}(t_i^+)| \leq E_{max}, i = 0, \ldots, n-1, \end{cases} \tag{24}$$

which are also polynomial inequality constraints of $h_1, \ldots, h_n$ (and $v_1, \ldots, v_{n-1}$) from (3), (4), (9), and (16).

3.3 Minimum time scheduling problem

The total machining time should be minimized by scheduling the time intervals h_i under all the constraints. Together with (9), (10), (11), (12), and (24), the optimization problem is:

$$\min_{h_i > 0} (h_1 + \cdots + h_n)$$

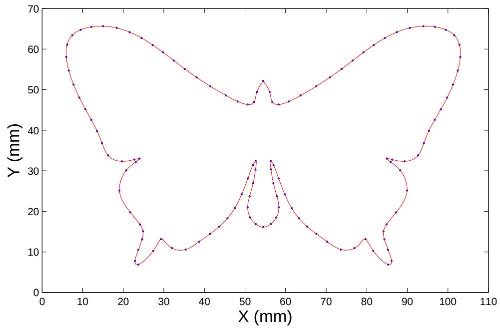

Figure 3: The reference knots and cubic spline trajectory.

s.t.

$$\begin{cases} H_1(v_1,\ldots,v_{n-1})^T = H_2, \text{ for each axis,} \\ |\vec{v}_i| \le \vec{V}_{max}, i=1,\ldots,n-1, \\ |\dot{\vec{C}}_i(t_i^*)| \le \vec{V}_{max}, \text{when } t_{i-1} < t_i^* < t_i, i=1,\ldots,n, \\ |\ddot{\vec{C}}_i(t_i)| \le \vec{A}_{max}, i=1,\ldots,n-1, \\ |\dddot{\vec{C}}_i| \le \vec{J}_{max}, i=1,\ldots,n, \\ |\vec{\hat{e}}(t_i^-)| \le \vec{E}_{max}, i=1,\ldots,n, \\ |\vec{\hat{e}}(t_i^+)| \le \vec{E}_{max}, i=0,\ldots,n-1. \end{cases} \tag{25}$$

The notation $|\cdot|$ in (25) stands for the absolute value of each scalar term.

It is obvious that the solution of this problem exists by multiplying a large enough positive number to any initial $(h_1,\ldots,h_n)$ whose components are all positive (similar to [5, 6]). The lower bounds of $h_1,\ldots,h_n$ in (22) can be used and an initial guess for solving this nonlinear optimization problem. The problem has a linear objective function and polynomial constraints, which can be solved by using sequential quadratic programming techniques [30, 31].

When the optimal values of $h_1,\ldots,h_n$ are known, the trajectory is determined by (1) and then the interpolation points are directly obtained.

4 Simulation results

A plane butterfly curve in [32] is used to generate the reference knots for the simulation. The original curve is discretized to be more than 100 points which are close enough to be the reference knots (see the points in Fig. 3).

The drive constraints are set to be $\vec{V}_{max} = (100, 100)$ mm/s, $\vec{A}_{max} = (1000, 1000)$ mm/s^2, $\vec{J}_{max} = (10000, 10000)$ mm/s^3. The initial and terminal velocities and accelerations are all zero. The sampling period is 1 ms. The same physical parameters are assumed for both the x and y axes [25]: $k_a = 8$ A/V, $k_t = 0.5$ Nm/A, $r_g = 0.002$ m/rad, $J = 0.01$ kgm^2, $B = 0.025$ kgm^2/s, so $K = k_a k_t r_g = 0.008$ Nm^2/V. The proportional, integral, derivative

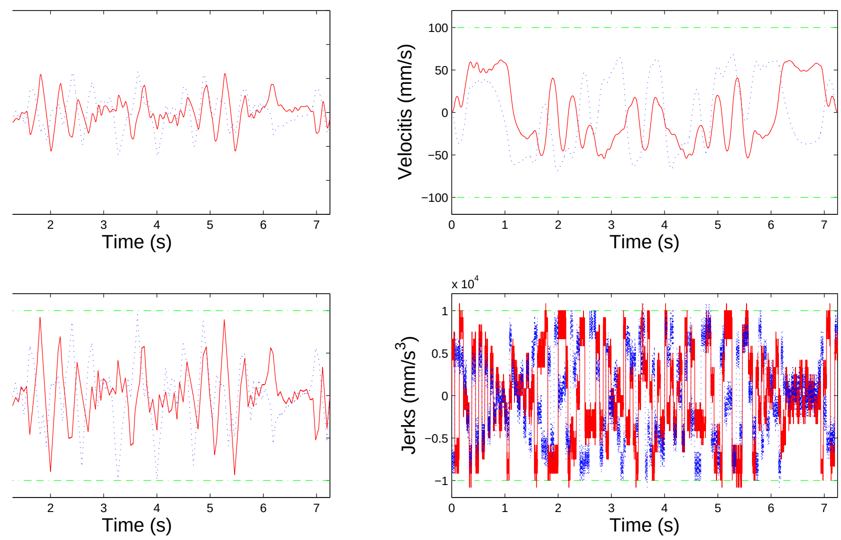

Figure 4: Simulation results without tracking error constraints.

gains are chosen as $k_p = 80$ V/mm, $k_i = 800$ V/(mm·s) and $k_d = 1$ V/(mm/s) for both axes. For comparison, E_{max} is set to be null, 100 μm and 50 μm respectively for both axes.

The *fmincon* function in MATLAB is used for solving (25). The cubic spline trajectory without tracking error constraints is shown in Fig. 3. PID controller and drive system are simulated by using Simulink in MATLAB for each axis. In Fig. 4-6, the simulation results of tracking error, velocity, acceleration and jerk for x-axis (solid curves) and y-axis (dotted curves) with different tracking error constraints are shown, respectively. The comparison of tracking results (include maximum value and root-mean-square (rms) of tracking errors for each axis) and machining time with different E_{max} are listed in Table. 2. It shows that the tracking errors can be significantly reduced with the costs of a small increase of machining time.

From (23), the tracking error constraint on each axis can be shown in this example:

$$|e(t)| \leq 29.62E_{max} + 1.12,$$

where the unit is mm. From the results shown in Table. 2, it can be seen that the bound E_{max} does not directly decide the value of tracking errors. However, given a smaller E_{max}, the tracking errors also become smaller.

5 Conclusions

In this study, a scheme for generating C^2 cubic spline trajectory with confined jerk and tracking error for 3-axis CNC machines is presented. Unlike most of the feedrate planning and interpolation methods, the tool path is directly designed as the function in time t, which can directly generate the interpolation points. For the PID controller, the tracking error on

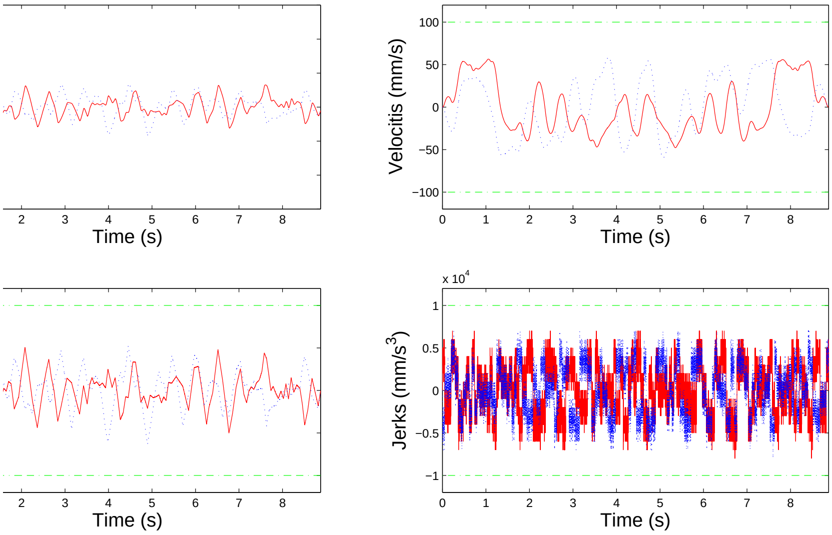

Figure 5: Simulation results with $E_{max} = 100$ μm.

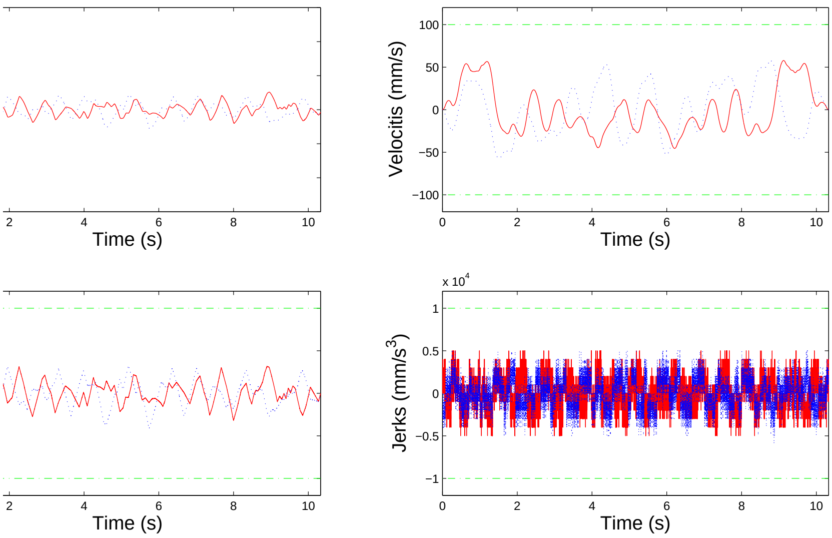

Figure 6: Simulation results with $E_{max} = 50$ μm.

Table 2: Summary of simulated tracking results and machining time.

E_{max} (μm)	X axis tracking error		Y axis tracking error		Machining time (s)
	max. (μm)	rms (μm)	max. (μm)	rms (μm)	
Null	11.5	3.0	12.8	3.6	7.251
100	6.5	2.0	8.4	2.5	8.875
50	5.2	1.5	5.6	1.8	10.328

each axis is approximated by a piecewise linear particular solution. All the constraints can be easily expressed as polynomial constraints of $h_1, \ldots, h_n$, which are then scheduled such that the machining time is minimized. Simulation results show that with a proper choice of E_{max}, the tracking errors can be significantly reduced with the costs of a small increase of machining time.

References

[1] Bobrow JE, Dubowsky S, Gibson JS. Time-optimal control of robotic manipulators along specified paths. *International Journal of Robotics Research* 1985; 4(3):3-17.

[2] Neuman CP, Tourassis VD. Discrete dynamic robot models. *IEEE Transactions on Systems, Man and Cybernetics* 1985; 15(2):193-204.

[3] Tan HH, Potts RB. Minimum time trajectory planner for the discrete dynamic robot model with dynamic constraints. *IEEE Journal of Robotics and Automation* 1988; 4:174-185.

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

[5] Lin CS, Chang PR, Luh JYS. Formulation and optimization of cubic polynomial joint trajectories for industrial robots. *IEEE Transactions on Automatic Control* 1983; 28:1066-1074.

[6] Gasparetto A, Zanotto V. A technique for time-jerk optimal planning of robot trajectories. *Robotics and Computer-Integrated Manufacturing* 2008; 24:415-426.

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

[8] Altintas Y, Erkorkmaz K. Feedrate Optimization for Spline Interpolation In High Speed Machine Tools. *CIRP Annals - Manufacturing Technology* 2003; 52(1):297-302.

[9] Sencer B, Altintas Y, Croft E. Feed optimization for five-axis CNC machine tools with drive constraints. *International Journal of Machine Tools and Manufacture* 2008; 48:733-745.

[10] Nam SH, Yang MY. A study on a generalized parametric interpolator with real-time jerk limited acceleration. *Computer-Aided Design* 2004; 36:27-36.

[11] Lai JY, Lin KY, Tseng SJ, Ueng WD. On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk. *International Journal of Advanced Manufacturing Technology* 2008; 37(1-2):104-121.

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

[13] Lee AC, Lin MT, Pan YR, Lin WY. The feedrate scheduling of NURBS interpolator for CNC machine tools. *Computer-Aided Design* 2011; 43:612-628.

[14] Zhang LX, Sun RY, Gao XS, Li HB. High speed interpolation for micro-line trajectory and adaptive real-time look-ahead in CNC machining. *Science China - Technological Sciences* 2011; 54(6):1481-1495.

[15] Timar SD, Farouki RT, Smith TS, Boyadjieff CL. Algorithms for time-optimal control of CNC machines along curved tool paths. *Robotics and Computer-Integrated Manufacturing* 2005; 21(1):37-53.

[16] Zhang M, Yan W, Yuan CM, Wang D, Gao, XS. Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. *Science China - Information Sciences* 2011; 54(7):1407-1418.

[17] Yuan CM, Zhang K, Fan W, Gao XS. Time-optimal interpolation for CNC machining along curved tool paths with confined chord error. *MM Research Preprints* 2011; 30:57-89.

[18] Zhang K, Yuan CM, Gao XS, Li HB. A greedy algorithm for feedrate planning of CNC machines along curved tool paths with confined jerk. *Robotics and Computer-Integrated Manufacturing* 2012; 28:472-483.

[19] Dong JY, Ferreiraa PM, Stori JA. Feedrate optimization with jerk constraints for generating minimum-time trajectories. *International Journal of Machine Tools and Manufacture* 2007; 47:1941-1955.

[20] Beudaert X, Pechard PY, Tournier C. 5-Axis tool path smoothing based on drive constraints. *International Journal of Machine Tools and Manufacture* 2011; 51:958-965.

[21] Beudaert X, Lavernhe S, Tournier C. Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path. *International Journal of Machine Tools and Manufacture* 2012; 57:73-82.

[22] Renton D, Elbestawi MA. High speed servo control of multi-axis machine tools. *International Journal of Machine Tools and Manufacture* 2000; 40:539-559.

[23] Dong JY, Stori JA. Optimal feed-rate scheduling for high-speed contouring. *Journal of Manufacturing Science and engineering* 2007; 129(1):63-76.

[24] Erkorkmaz K, Altintas Y. High speed CNC system design. Part III: high speed tracking and contouring control of feed drives. *International Journal of Machine Tools and Manufacture* 2001; 41:1637-1658.

[25] Ernesto CA, Farouki RT. Solution of inverse dynamics problems for contour error minimization in CNC machines. *International Journal of Advanced Manufacturing Technology* 2010; 49:589-604.

[26] Conway JR, Ernesto CA, Farouki RT, Zhang M. Performance analysis of cross-coupled controllers for CNC machines based upon precise real-time contour error measurement. *International Journal of Machine Tools and Manufacture* 2012; 52:30-39.

[27] Huo F, Xi XC, Poo AN. Generalized Taylor series expansion for free-form two-dimensional contour error compensation. *International Journal of Machine Tools and Manufacture* 2012; 53:91-99.

[28] Altintas Y. Manufacturing automation: metal cutting mechanics, machine tool vibrations, and CNC design. Cambridge University Press, Cambridge, 2000.

[29] Yeung CH, Altintas Y, Erkorkmaz K. Virtual CNC system. Part I. System architecture. *International Journal of Machine Tools and Manufacture* 2006; 46:1107-1123.

[30] Nocedal J, Wright SJ. Numerical optimization. Springer, Berlin Heidelberg New York, 1999.

[31] Yang WY, Cao WW, Chung TS, Morris J. Applied numerical methods using MATLAB. John Wiley and Sons, New Jersey, 2005.

[32] Yau HT, Lin MT, Tsai MS. Real-time NURBS interpolation using FPGA for high speed motion control. *Computer-Aided Design* 2006; 38:1123-1133.