High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error

Charlie A. Ernesto and Rida T. Farouki
Department of Mechanical and Aerospace Engineering,
University of California, Davis, CA 95616, USA.

Abstract

To exactly execute a sharp corner in the toolpath, the feedrate of a CNC machine must instantaneously drop to zero at that point. This constraint is problematic in the context of high–speed machining, since it incurs very high deceleration/acceleration rates near sharp corners, that increase the total machining time, and may incur significant path deviations (contour errors) at these points. A strategy for negotiating sharp corners in high–speed machining is proposed herein, based upon *a priori* toolpath/feedrate modifications in their vicinity. Each corner is smoothed by replacing a subset of the path that contains it with a conic "splice" segment, deviating from the exact corner by no more than a prescribed tolerance ϵ, along which the square of the feedrate is specified as a Bernstein–form polynomial. Determining the fastest traversal of the conic segments under known axis acceleration bounds can then be formulated as a constrained optimization problem, and by exploiting some well–known properties of Bernstein–form polynomials this can be approximated by a simple linear programming task. Some computed examples are presented to illustrate the implementation and performance of the high–speed cornering strategy.

Keywords: CNC machine, high–speed machining, feedrate, contour error, cornering, path modification, constrained optimization, linear programming.

e–mail addresses: caernesto@ucdavis.edu, farouki@ucdavis.edu

1 Introduction

Each axis of a Cartesian CNC machine is subject to bounds on the magnitude of its acceleration, determined (at minimum) by the axis inertia and torque capacity of the drive motor. The machine must come to a complete stop when traversing a sharp corner in the toolpath, since an instantaneous change in direction of motion at finite speed implies infinite acceleration.[1] The handling of the deceleration/acceleration phases necessitated by corners significantly influences the total path execution time, and the fidelity of the executed path to the commanded path — i.e., the path *contour error*.

A sharp corner may be regarded as an impulse in the path curvature, of magnitude equal to the angular change of path direction at that corner. It is useful to consider the corner as the limit of a sequence of smooth paths with finite curvature "spikes" culminating in a delta function. If the corner is to be traversed in a time interval shorter than that implied by an exact traversal under the machine axis acceleration bounds, it is necessary to "smooth out" the geometry. Two aspects of the smoothing facilitate a faster traversal: (i) the smoothed path can be of shorter length than the corresponding portion of the exact path; and (ii) since the smoothed path has finite curvature, a non–zero feedrate can be maintained over its entire extent.

It should be noted that, because of the internal dynamic characteristics (inertia and damping) of the machine axes, there will be an inherent tendency of the physical motion to "round out" sharp path corners. However, this fact does not exempt one from the need for careful analysis of the exact manner in which corners are executed, if prescribed bounds on axis accelerations and path contour error are to be observed. In principle, it is possible to formulate an *inverse dynamics* problem, whose solution determines a modified machine input that, subject to the internal machine dynamics, exactly produces the desired output motion. This has been addressed [4] in the context of smooth paths and unbounded accelerations, but the problem is far more difficult for non–smooth paths subject to finite axis acceleration bounds.

The problem of negotiating sharp corners in toolpaths that one wishes to execute at very fast nominal feedrates is of particular concern in the context of *high–speed machining* [2, 15, 18, 19, 20] and some authors [3, 13, 14] have proposed smoothing of sharp corners by analytic curves with free parameters

[1] To execute a sharp corner at finite speed, each of the machine axes must in general achieve an instantaneous change of speed — i.e., an infinite acceleration.

that can be adjusted to minimize contour error. The present paper proposes a different approach to high–speed cornering, based on *a priori* modifications to the toolpath geometry and feedrate that are formulated to ensure satisfaction of prescribed bounds on the contour error and axis accelerations.

The proposed corner smoothing scheme is based on "splicing" the linear segments that meet at a corner with a G^1 conic segment. The conic segment is specified as a rational quadratic Bézier curve $\mathbf{r}(\xi)$ for $\xi \in [\,0,1\,]$ with control points $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ and scalar weight w_1 associated with $\mathbf{p}_1$. Taking $\mathbf{p}_1$ as the exact corner point, the lengths $\ell_1 = |\mathbf{p}_1 - \mathbf{p}_0|$, $\ell_2 = |\mathbf{p}_2 - \mathbf{p}_1|$ of the control–polygon legs and the weight w_1 are available as free parameters to adjust the geometry of the conic splice segment, and to provide a simple and intuitive means of subduing the contour error — which decreases monotonically as w_1 increases and ℓ_1, ℓ_2 decrease. Furthermore, it is possible to develop efficient and accurate real–time interpolators for Bézier conic segments [7], as natural extensions of simple linear/circular G code segments.

Once a conic splice segment $\mathbf{r}(\xi)$ satisfying a prescribed contour error has been constructed, a feedrate function $V(\xi)$ must be determined along it so as to mimimize the traversal time, consistent with the axis acceleration bounds. Formally, this feedrate is the solution of a calculus of variations problem with pointwise constraints. Since this does not admit simple closed–form solutions, we formulate an approximation scheme based on writing[2] $E(\xi) = \frac{1}{2}V^2(\xi)$ as a Bernstein–form polynomial on $\xi \in [\,0,1\,]$. The axis acceleration constraints can then be expressed as a set of polynomial inequalities, with coefficients linearly dependent on the Bernstein coefficients $E_0, \ldots, E_n$ of $E(\xi)$.

The polynomial inequalities arising from the acceleration constraints are not imposed directly. Instead, the inequalities are applied to their Bernstein coefficients (which are linear expressions in $E_0, \ldots, E_n$). Through recursive application of the de Casteljau subdivision algorithm the "control polygons" defined by these coefficients converge monotonically [5] to the graphs of the constraint polynomials, guaranteeing satisfaction of the exact acceleration constraints to any desired precision through a system of linear inequalities. The task is then to minimize the traversal time for the conic splice segment,

$$T \,=\, \int_0^1 \frac{|\mathbf{r}'(\xi)|}{\sqrt{2\,E(\xi)}}\,\mathrm{d}\xi\,, \tag{1}$$

[2] $E(\xi) = \frac{1}{2}V^2(\xi)$ is used to specify the feedrate since, unlike $V(\xi)$, its time derivatives are polynomials in the curve parameter ξ.

with respect to $E_0, \ldots, E_n$ — subject to linear constraints on them. Although the objective function (1) is not linear in $E_0, \ldots, E_n$, it is easily verified to be monotone with respect to each of these variables. Therefore, to facilitate solution of the constrained optimization problem, a linear approximation of (1) is used, which reduces the task to a classical linear programming problem — for which efficient solution procedures are well–known.

The plan for the remainder of this paper is as follows. The key parameters governing exact acceleration–limited execution of sharp corners by Cartesian CNC machines are first identified in §2. The problem of achieving fast corner execution, subject to specified contour error and axis acceleration bounds, is then introduced in §3, which describes an approach to smoothing sharp corners using conic "splice" segments that deviate from the exact corners by no more than a prescribed tolerance ϵ. In §4 we consider fastest traversal of smoothed corners, consistent with the axis acceleration bounds. The optimal feedrate is the solution of a calculus of variations problem under pointwise constraints, but near–optimal solutions can be obtained by approximating it with a simple linear programming task. Finally, examples from an implementation of the method are presented in §5, while §6 recapitulates key results of this study and identifies some issues that deserve further investigation.

2 Exact acceleration–limited cornering

We study first the optimal (fastest) exact execution of sharp corners by CNC machines with acceleration–limited axes, since this furnishes key parameters for the corner smoothing scheme. For brevity, we consider 2–axis Cartesian machines (the extension to 3–axis machines is straightforward). We assume that the desired motion before and after negotiation of the sharp corner point corresponds to a prescribed constant feedrate V_*.

Consider a machine with acceleration/deceleration bounds A_x, A_y for the x, y axes. Let $\mathbf{p}_1 = (x_1, y_1)$ be a sharp corner point of the machine path, and let the angles θ_1, θ_2 define the orientation of the incoming and outgoing path segments that meet at $\mathbf{p}_1$, so that $\mathbf{t}_1 = (\cos\theta_1, \sin\theta_1)$ and $\mathbf{t}_2 = (\cos\theta_2, \sin\theta_2)$ are unit vectors along them. To exactly execute the corner, the machine must completely stop at $\mathbf{p}_1$ — i.e., the feedrate must decrease from V_* to 0 on the incoming segment, and increase from 0 to V_* on the outgoing segment.

According to the "bang–bang principle" [11, 16] the fastest traversal of the corner is realized when either the x or y axis is operating at its maximum acceleration/deceleration rate, A_x or A_y, on both the incoming and outgoing path segments. We take $\mathbf{p}_1 = (0,0)$ without loss of generality, and denote by d_1, d_2 and t_1, t_2 the extents and durations of the deceleration and acceleration phases. Then for the incoming segment we have

$$\begin{aligned} 0 - d_1 \cos\theta_1 &= \tfrac{1}{2} a_x t_1^2 \,, & 0 - V_* \cos\theta_1 &= a_x t_1 \,, \\ 0 - d_1 \sin\theta_1 &= \tfrac{1}{2} a_y t_1^2 \,, & 0 - V_* \sin\theta_1 &= a_y t_1 \,, \end{aligned}$$

where a_x, a_y are the x, y axis accelerations (assumed uniform). Similarly, for the outgoing segment

$$\begin{aligned} d_2 \cos\theta_2 - 0 &= \tfrac{1}{2} a_x t_2^2 \,, & V_* \cos\theta_2 - 0 &= a_x t_2 \,, \\ d_2 \sin\theta_2 - 0 &= \tfrac{1}{2} a_y t_2^2 \,, & V_* \sin\theta_2 - 0 &= a_y t_2 \,. \end{aligned}$$

Eliminating t_1 and t_2 among these equations, we obtain the axis accelerations

$$(a_x, a_y) \;=\; \frac{V_*^2(-\cos\theta_1, -\sin\theta_1)}{2d_1} \qquad \text{and} \qquad (a_x, a_y) \;=\; \frac{V_*^2(\cos\theta_2, \sin\theta_2)}{2d_2}$$

during the two phases of the corner traversal, and we must have $|a_x| \le A_x$, $|a_y| \le A_y$ for each phase. These constraints determine the *minimum* required extents d_1, d_2 for the deceleration/acceleration phases as

$$d_1 = V_*^2 \max\left[\frac{|\cos\theta_1|}{2A_x}, \frac{|\sin\theta_1|}{2A_y} \right], \quad d_2 = V_*^2 \max\left[\frac{|\cos\theta_2|}{2A_x}, \frac{|\sin\theta_2|}{2A_y} \right]. \tag{2}$$

Consequently, the minimum required time $T_{\min}$ for an exact corner traversal, under the given axis acceleration bounds, is

$$T_{\min} \;=\; \frac{2(d_1 + d_2)}{V_*} \,. \tag{3}$$

Note that the subset of the toolpath associated with the exact corner traversal can be parameterized as

$$\hat{\mathbf{r}}(\xi) \;=\; \begin{cases} \mathbf{p}_0(1 - 2\xi) + \mathbf{p}_1(2\xi - 0) & \text{for} \;\; \xi \in [\,0, \tfrac{1}{2}\,] \,, \\ \mathbf{p}_1(2 - 2\xi) + \mathbf{p}_2(2\xi - 1) & \text{for} \;\; \xi \in [\,\tfrac{1}{2}, 1\,] \,, \end{cases} \tag{4}$$

where $\mathbf{p}_0 = \mathbf{p}_1 - d_1\mathbf{t}_1$ and $\mathbf{p}_2 = \mathbf{p}_1 + d_2\mathbf{t}_2$.

For the exact corner traversal, the feedrate V decreases linearly with time from V_* to 0 along the incoming segment, and increases linearly with time from 0 to V_* along the outgoing segment. The dependence of V on the path parameter ξ, however, is not linear — it is given by

$$\hat{V}(\xi) \;=\; \begin{cases} V_*\sqrt{1 - 2\xi} & \text{for} \;\; \xi \in [\,0, \tfrac{1}{2}\,] \,, \\ V_*\sqrt{2\xi - 1} & \text{for} \;\; \xi \in [\,\tfrac{1}{2}, 1\,] \,. \end{cases} \tag{5}$$

3 High–speed cornering problem

An exact traversal of the corner (4) subject to the prescribed axis acceleration bounds is possible using the feedrate (5), and its duration is defined by (3). To achieve a faster traversal, it is necessary to "smooth out" the corner so a non–zero feedrate can be maintained throughout the motion.

3.1 Path modification scheme

The overall scheme for the corner toolpath/feedrate modification is as follows. Lengths L_1 and L_2 on the original path, before and after the corner point $\mathbf{p}_1$, are isolated and decomposed as $L_1 = a_1 + b_1 + \ell_1$ and $L_2 = \ell_2 + b_2 + a_2$. The lengths a_1 and a_2 define "coasting" phases on the original path, at constant feedrate V_*, while b_1 and b_2 define constant deceleration/acceleration phases, where the feedrate decreases from V_* to V_1 and increases from V_2 to V_* (the values V_1, V_2 are yet to be determined). Finally, the lengths ℓ_1 and ℓ_2 on the original path are replaced by a Bézier conic segment $\mathbf{r}(\xi)$ defined on $\xi \in [\,0,1\,]$ with control points $\mathbf{p}_0 = \mathbf{p}_1 - \ell_1\mathbf{t}_1$, $\mathbf{p}_1$, $\mathbf{p}_2 = \mathbf{p}_1 + \ell_2\mathbf{t}_2$ and scalar weight w_1 associated with control point $\mathbf{p}_1$ — see expression (12) below.

The feedrate variation $V(\xi)$ along the conic segment $\mathbf{r}(\xi)$, including the initial/final values $V_1 = V(0)$ and $V_2 = V(1)$, will be obtained by minimizing its traversal time under the axis acceleration bounds. To compare the total traversal time with the unmodified path, we also isolate lengths L_1 and L_2 on it, before and after $\mathbf{p}_1$, and decompose them as $L_1 = c_1 + d_1$ and $L_2 = d_2 + c_2$, where c_1, c_2 define coasting phases at feedrate V_* and d_1, d_2 are given by (2). The parameters that define corresponding modified and unmodified segments of the toolpath corner are illustrated in Figure 1.

Now depending on the values of $\ell_1, \ell_2, \theta_1, \theta_2, V_1, V_2$ and A_x, A_y it may be possible to omit the coasting phase on the incoming or outgoing parts of the modified or unmodified path (or both), to make L_1, L_2 as small as possible. The feed acceleration (i.e., time derivative of the feedrate $V = \mathrm{d}s/\mathrm{d}t$) is

$$A \,=\, \frac{\mathrm{d}V}{\mathrm{d}t} \,=\, V\,\frac{\mathrm{d}V}{\mathrm{d}s}\,,$$

where s is distance along the path. To satisfy the axis acceleration bounds, the maximum feed acceleration magnitude on the incoming/outgoing linear

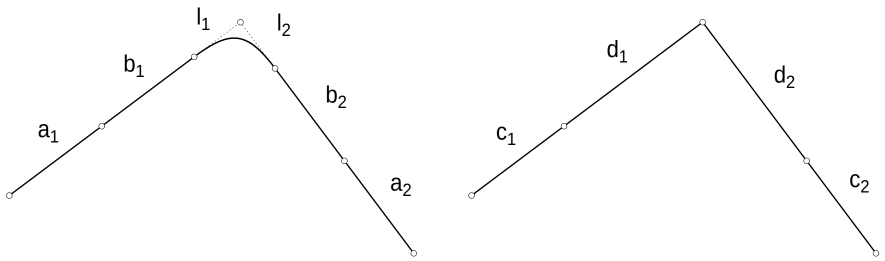

Figure 1: Left: modified path corner — a_1 and a_2 are "coasting" segments, at feedrate V_*; b_1 and b_2 are linear deceleration/acceleration segments, from V_* to V_1 and from V_2 and V_*; and ℓ_1, ℓ_2 are control polygon legs for the conic smoothing segment. Right: unmodified corner — c_1, c_2 are coasting segments, while d_1, d_2 are deceleration/acceleration segments defined by (2).

path segments is defined for $i = 1, 2$ by

$$A_i = \min\left[\frac{A_x}{|\cos\theta_i|}, \frac{A_y}{|\sin\theta_i|}\right] . \tag{6}$$

Since, for a constant feed acceleration A over linear distance Δs, the change in $\frac{1}{2}V^2$ is equal to $A\Delta s$, we have

$$\tfrac{1}{2}V_*^2 - \tfrac{1}{2}V_i^2 \,=\, A_i b_i \qquad \text{and} \qquad \tfrac{1}{2}V_*^2 \,=\, A_i d_i$$

for $i = 1, 2$. Consequently,

$$b_i \,=\, \frac{V_*^2 - V_i^2}{2A_i} \qquad \text{and} \qquad d_i \,=\, \frac{V_*^2}{2A_i} , \tag{7}$$

while $a_i = L_i - b_i - \ell_i$ and $c_i = L_i - d_i$ are given by

$$a_i \,=\, L_i \,-\, \frac{V_*^2 - V_i^2}{2A_i} \,-\, \ell_i \qquad \text{and} \qquad c_i \,=\, L_i \,-\, \frac{V_*^2}{2A_i} . \tag{8}$$

Hence, comparing $b_i + \ell_i$ with d_i (see Figure 1) for $i = 1, 2$ we may choose

$$a_i = 0\,, \quad c_i = \ell_i - \frac{V_i^2}{2A_i} \qquad \text{if} \;\; \ell_i > \frac{V_i^2}{2A_i} \,,$$

$$a_i = 0\,, \quad c_i = 0 \qquad \text{if} \;\; \ell_i = \frac{V_i^2}{2A_i} \,,$$

$$a_i = \frac{V_i^2}{2A_i} - \ell_i\,, \quad c_i = 0 \qquad \text{if} \;\; \ell_i < \frac{V_i^2}{2A_i} \,,$$

to make L_1, L_2 as small as possible.

For the unmodified corner, the total traversal time is simply

$$T_c \,=\, \frac{c_1}{V_*} + \frac{V_*}{A_1} + \frac{V_*}{A_2} + \frac{c_2}{V_*}\,, \tag{9}$$

while for the smoothed corner, it is

$$T_s \,=\, \frac{a_1}{V_*} + \frac{V_* - V_1}{A_1} + T_C + \frac{V_* - V_2}{A_2} + \frac{a_2}{V_*}\,, \tag{10}$$

with T_C being the traversal time for the conic segment, obtained from the feedrate optimization process (see §4 below).

3.2 Conic smoothing segments

The Bernstein basis for polynomials of degree n on $\xi \in [\,0,1\,]$ is defined by

$$b_k^n(\xi) \,=\, \binom{n}{k}(1-\xi)^{n-k}\xi^k\,, \quad k = 0,\ldots,n\,. \tag{11}$$

On account of its inherent numerical stability and available useful algorithms [6, 9, 10, 22] this basis is employed in all subsequent computations. A conic segment can be defined [5] in the "standard" rational quadratic Bézier form

$$\mathbf{r}(\xi) \,=\, \frac{\mathbf{p}_0\, b_0^2(\xi) \,+\, w_1\mathbf{p}_1\, b_1^2(\xi) \,+\, \mathbf{p}_2\, b_2^2(\xi)}{b_0^2(\xi) \,+\, w_1\, b_1^2(\xi) \,+\, b_2^2(\xi)} \tag{12}$$

by the control points $\mathbf{p}_0 = \mathbf{p}_1 - \ell_1\mathbf{t}_1$, $\mathbf{p}_1$, $\mathbf{p}_2 = \mathbf{p}_1 + \ell_2\mathbf{t}_2$ and the associated weights $w_0 = 1, w_1, w_2 = 1$. Expression (12) specifies an ellipse, parabola, or hyperbola segment according to whether $w_1 < 1$, $w_1 = 1$, or $w_1 > 1$. As $w_1 \to \infty$, it converges monotonically to the control polygon defined by $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2$ (this is the limiting case of a hyperbola as a pair of intersecting lines) — see Figure 2. Note also that higher values of w_1 yield a smoother transition between that linear and conic path segments.

For $w_1 = 1$, expression (12) reduces to

$$\mathbf{r}(t) \,=\, \mathbf{p}_0\, b_0^2(\xi) \,+\, \mathbf{p}_1\, b_1^2(\xi) \,+\, \mathbf{p}_2\, b_2^2(\xi)\,, \tag{13}$$

i.e., parabolas admit *polynomial* parameterizations. For brevity, we now set

$$\mathbf{l}_1 \,=\, \mathbf{p}_1 - \mathbf{p}_0\,,\;\; \ell_1 \,=\, |\mathbf{l}_1| \qquad \text{and} \qquad \mathbf{l}_2 \,=\, \mathbf{p}_2 - \mathbf{p}_1\,,\;\; \ell_2 \,=\, |\mathbf{l}_2|\,.$$

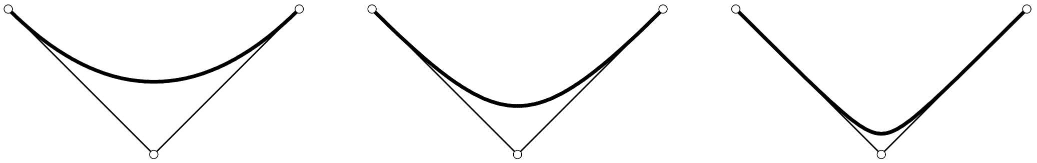

Figure 2: Conic segments represented as rational quadratic Bézier curves of the form (12) with weights $w_1 = 1$ (left), $w_1 = 2$ (center), and $w_1 = 6$ (right).

When $w_1 \neq 1$ the center of the ellipse/hyperbola is situated [17] at

$$\mathbf{c} \,=\, \mathbf{p}_1 \,+\, \frac{\mathbf{l}_2 - \mathbf{l}_1}{2(1-w_1^2)}\,, \tag{14}$$

while for $w_1 = 1$ the focus of the parabola is given [17] by

$$\mathbf{f} \,=\, \mathbf{p}_1 \,+\, \frac{\ell_1^2\,\mathbf{l}_2 - \ell_2^2\,\mathbf{l}_1}{|\,\mathbf{l}_2 - \mathbf{l}_1|^2}\,. \tag{15}$$

The derivative of expression (12) can be written as

$$\mathbf{r}'(\xi) \,=\, \frac{\mathbf{d}(\xi)}{w^2(\xi)}\,, \tag{16}$$

where

$$w(\xi) \,=\, b_0^2(\xi) \,+\, w_1\,b_1^2(\xi) \,+\, b_2^2(\xi)\,, \tag{17}$$

and we define

$$\mathbf{d}(\xi) \,=\, \mathbf{d}_0\,b_0^2(\xi) \,+\, \mathbf{d}_1\,b_1^2(\xi) \,+\, \mathbf{d}_2\,b_2^2(\xi) \tag{18}$$

with

$$\mathbf{d}_0 \,=\, 2w_1\mathbf{l}_1\,, \quad \mathbf{d}_1 \,=\, \mathbf{l}_1 + \mathbf{l}_2\,, \quad \mathbf{d}_2 \,=\, 2w_1\mathbf{l}_2\,.$$

In the case $w_1 = 1$, we have simply

$$\mathbf{r}'(\xi) \,=\, 2\,\mathbf{l}_1\,b_0^1(\xi) + 2\,\mathbf{l}_2\,b_1^1(\xi)\,.$$

Now let ϵ be a prescribed bound on the contour error between $\mathbf{r}(\xi)$ and the exact corner $\hat{\mathbf{r}}(\xi)$ defined by (4) with $\mathbf{p}_0 = \mathbf{p}_1 - \ell_1\mathbf{t}_1$, $\mathbf{p}_2 = \mathbf{p}_1 + \ell_2\mathbf{t}_2$. A suitable measure "distance$(\mathbf{r}(\xi),\hat{\mathbf{r}}(\xi))$" for the deviation of the conic segment from the exact path must be defined, to enforce the contour error bound ϵ. Perhaps the simplest definition,

$$\text{distance}(\mathbf{r}(\xi),\hat{\mathbf{r}}(\xi)) \,=\, \max_{\xi\in[\,0,1\,]}|\,\mathbf{r}(\xi) - \hat{\mathbf{r}}(\xi)\,|\,,$$

is based on a correspondence of the points on $\mathbf{r}(\xi)$ and $\hat{\mathbf{r}}(\xi)$ imposed by their parameterizations. A more rigorous measure, satisfying all the requirements of a metric function, is the *Hausdorff distance* defined [12] by

$$\text{distance}(\mathbf{r}(\xi),\hat{\mathbf{r}}(\xi)) \,=\, \min_{u\in[\,0,1\,]}\,\max_{v\in[\,0,1\,]}|\,\mathbf{r}(u)-\hat{\mathbf{r}}(v)\,|\,.$$

We adopt a simpler measure of the deviation of $\mathbf{r}(\xi)$ from $\hat{\mathbf{r}}(\xi)$ here — namely, the distance between the corner point $\mathbf{p}_1$ and the point on the conic segment closest to it. The point of (12) nearest to $\mathbf{p}_1$ must satisfy the condition

$$[\,\mathbf{p}_1-\mathbf{r}(\xi)\,]\cdot\mathbf{r}'(\xi) \,=\, 0\,, \tag{19}$$

i.e., the displacement $\mathbf{p}_1-\mathbf{r}(\xi)$ is orthogonal to the curve tangent direction, defined by $\mathbf{r}'(\xi)$. For an ellipse or hyperbola with $w_1\neq 1$, this can be reduced using (16) to a quartic equation with Bernstein coefficients

$$c_0 \,=\, 2w_1\ell_1^2\,,\qquad c_1 \,=\, \tfrac{1}{4}(1-w_1)(\ell_1^2+\mathbf{l}_1\cdot\mathbf{l}_2)\,,\qquad c_2 \,=\, 0\,,$$

$$c_3 \,=\, -\,\tfrac{1}{4}(1-w_1)(\ell_2^2+\mathbf{l}_1\cdot\mathbf{l}_2)\,,\qquad c_4 \,=\, -\,2w_1\ell_2^2\,.$$

For the parabola with $w_1=1$, on the other hand, it reduces to a cubic with Bernstein coefficients

$$c_0 \,=\, \ell_1^2\,,\qquad c_1 \,=\, -\,c_2 \,=\, \mathbf{l}_1\cdot\mathbf{l}_2\,,\qquad c_3 \,=\, \ell_2^2\,.$$

Standard root–finding methods for Bernstein–form polynomials can be used to compute the real roots of the cubic or quartic on $\xi\in[\,0,1\,]$.

The above procedure allows computation of the tolerance between the exact corner and the conic segment defined by w_1 for given points $\mathbf{p}_0,\mathbf{p}_1,\mathbf{p}_2$. To find the w_1 value required to achieve a specified tolerance ϵ, the procedure can be invoked in a standard numerical scheme — such as the bisection or secant method [1]. Alternatively, note that ℓ_1, ℓ_2 can be adjusted to achieve a desired tolerance ϵ, since uniform scaling of $\mathbf{r}(\xi)$ by a positive factor k does not alter the roots of (19) but changes the tolerance from ϵ to $k\,\epsilon$.

3.3 Acceleration on conic segment

The parametric speed and feedrate along the conic segment are

$$\sigma(\xi) \,=\, |\mathbf{r}'(\xi)| \,=\, \frac{\mathrm{d}s}{\mathrm{d}\xi}\qquad\text{and}\qquad V(\xi) \,=\, \frac{\mathrm{d}s}{\mathrm{d}t}\,,$$

where s denotes the arc length of $\mathbf{r}(\xi)$ measured from $\xi = 0$. Thus, derivatives with respect to time t and the curve parameter ξ are related by

$$\frac{\mathrm{d}}{\mathrm{d}t} \;=\; \frac{\mathrm{d}s}{\mathrm{d}t}\,\frac{\mathrm{d}\xi}{\mathrm{d}s}\,\frac{\mathrm{d}}{\mathrm{d}\xi} \;=\; \frac{V}{\sigma}\,\frac{\mathrm{d}}{\mathrm{d}\xi}\,. \tag{20}$$

Hence, using dots for time derivaties and primes for parametric derivatives, we have

$$\mathbf{v} \,=\, \dot{\mathbf{r}} \,=\, \frac{V}{\sigma}\,\mathbf{r}' \qquad \text{and} \qquad \mathbf{a} \,=\, \ddot{\mathbf{r}} \,=\, \frac{V}{\sigma}\left[\frac{\sigma V' - \sigma' V}{\sigma^2}\,\mathbf{r}' \,+\, \frac{V}{\sigma}\,\mathbf{r}''\right].$$

and the x, y components of velocity $\mathbf{v}$ and acceleration $\mathbf{a}$ along the conic path segment $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ are

$$\begin{aligned} \dot{x} \,&=\, \frac{V}{\sigma}\,x'\,, \qquad & \ddot{x} \,&=\, \frac{V}{\sigma}\left[\frac{\sigma V' - \sigma' V}{\sigma^2}\,x' + \frac{V}{\sigma}\,x''\right], \\ \dot{y} \,&=\, \frac{V}{\sigma}\,y'\,, \qquad & \ddot{y} \,&=\, \frac{V}{\sigma}\left[\frac{\sigma V' - \sigma' V}{\sigma^2}\,y' + \frac{V}{\sigma}\,y''\right], \end{aligned}$$

where

$$\sigma = \sqrt{\mathbf{r}'\cdot\mathbf{r}'} = \sqrt{x'^2 + y'^2}\,, \qquad \sigma' \,=\, \frac{\mathbf{r}'\cdot\mathbf{r}''}{\sigma} \,=\, \frac{x'x'' + y'y''}{\sigma}\,. \tag{21}$$

Substituting from (21), the acceleration components can be written in terms of the path $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ and feedrate $V(\xi)$ as

$$\begin{aligned} \ddot{x} \,&=\, \frac{V}{x'^2 + y'^2}\left[\left(V' - \frac{x'x'' + y'y''}{x'^2 + y'^2}\,V\right)x' \,+\, Vx''\right], \\ \ddot{y} \,&=\, \frac{1}{x'^2 + y'^2}\left[\left(V' - \frac{x'x'' + y'y''}{x'^2 + y'^2}\,V\right)y' \,+\, Vy''\right]. \end{aligned}$$

The acceleration constraints $|\ddot{x}| \le A_x$, $|\ddot{y}| \le A_y$ can thus be phrased as

$$\begin{aligned} (x'^2 + y'^2)(x''V^2 + x'VV') - (x'x'' + y'y'')x'V^2 - A_x(x'^2 + y'^2)^2 \,&\le\, 0\,, \\ (x'^2 + y'^2)(x''V^2 + x'VV') - (x'x'' + y'y'')x'V^2 + A_x(x'^2 + y'^2)^2 \,&\le\, 0\,, \\ (x'^2 + y'^2)(y''V^2 + y'VV') - (x'x'' + y'y'')y'V^2 - A_y(x'^2 + y'^2)^2 \,&\le\, 0\,, \\ (x'^2 + y'^2)(y''V^2 + y'VV') - (x'x'' + y'y'')y'V^2 + A_y(x'^2 + y'^2)^2 \,&\le\, 0\,. \end{aligned} \tag{22}$$

When $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ and $V(\xi)$ are polynomial or rational functions of ξ, these are polynomial or rational inequalities.

The general problem involves finding the path $\mathbf{r}(\xi)$ and feedrate $V(\xi)$ that minimize the traversal time,

$$T \,=\, \int_0^1 \frac{\sigma}{V}\,\mathrm{d}\xi\,,$$

subject to satisfaction of the above system of acceleration constraints by $\mathbf{r}(\xi)$ and $V(\xi)$ for $\xi \in [\,0,1\,]$ and the tolerance ϵ on the deviation of $\mathbf{r}(\xi)$ from $\hat{\mathbf{r}}(\xi)$. This is an extremely challenging problem in the calculus of variations, and some assumptions/simplifications are necessary to render it more tractable. If the path $\mathbf{r}(\xi)$ is specified *a priori*, solution methods based on the principles of bang–bang control are available [21], but these are also quite complicated. Our goal is to formulate a relatively simple scheme suited to the high–speed cornering problem — yielding improvements in cornering time that, although not truly optimal, are nevertheless significant and worthwhile. This scheme uses conic arcs as smoothing segments (see §3.2) and a linear programming approximation (see §4) to the constrained optimization problem.

4 Feedrate function determination

The parametric speed of the conic segment (12) — i.e., the derivative $\mathrm{d}s/\mathrm{d}\xi$ of the arc length s with respect to the curve parameter ξ — can be expressed in terms of (17) and (18) as

$$\sigma(\xi) \,=\, |\mathbf{r}'(\xi)| \,=\, \frac{|\mathbf{d}(\xi)|}{w^2(\xi)}\,.$$

Applying (20) to (12), and using dots and primes to denote derivatives with respect to time t and the curve parameter ξ, the velocity $\mathbf{v}$ and acceleration $\mathbf{a}$ along $\mathbf{r}(\xi)$ can be written as

$$\mathbf{v} \,=\, \dot{\mathbf{r}} \,=\, V\frac{\mathbf{d}}{|\mathbf{d}|}\,, \qquad \mathbf{a} \,=\, \ddot{\mathbf{r}} \,=\, \frac{w^2}{|\mathbf{d}|^4}\,[\,VV'\,|\mathbf{d}|^2\,\mathbf{d} \,+\, V^2\,\mathbf{d}\times(\mathbf{d}'\times\mathbf{d})\,]\,.$$

Instead of defining the feedrate V as a polynomial in the curve parameter ξ, it is convenient to define $E = \frac{1}{2}V^2$ in the Bernstein basis (11) as

$$E(\xi) \,=\, \sum_{k=0}^{n} E_k b_k^n(\xi)\,, \quad \xi \in [\,0,1\,]\,. \tag{23}$$

The coefficients of $E(\xi)$ will be determined by minimizing the traversal time for the conic segment, subject to the axis acceleration constraints.

4.1 Axis acceleration constraints

In terms of E, the acceleration can be written as

$$\mathbf{a} \,=\, \ddot{\mathbf{r}} \,=\, \frac{w^2}{|\mathbf{d}|^4}\,[\,E'\,|\mathbf{d}|^2\,\mathbf{d} \,+\, 2E\,\mathbf{d}\times(\mathbf{d}'\times\mathbf{d})\,]\,.$$

We note that this depends *linearly* on the coefficients $E_0,\ldots,E_n$. Now let (a_x,a_y) and (d_x,d_y) be the components of $\mathbf{a}$ and $\mathbf{d}$, and suppose the motion is subject to the axis acceleration bounds $|a_x|\le A_x$ and $|a_y|\le A_y$. Clearing denominators then gives the polynomial inequalities

$$\begin{aligned} w^2\,|\,(d_x^2+d_y^2)d_xE'+2(d_yd_x'-d_xd_y')d_yE\,| \,&\le\, A_x(d_x^2+d_y^2)^2\,,\\ w^2\,|\,(d_x^2+d_y^2)d_yE'+2(d_xd_y'-d_yd_x')d_xE\,| \,&\le\, A_y(d_x^2+d_y^2)^2\,, \end{aligned}$$

which we cast in the form

$$|\,P_x(\xi)E'(\xi)+Q_x(\xi)E(\xi)\,| \,\le\, R_x(\xi)\,, \quad |\,P_y(\xi)E'(\xi)+Q_y(\xi)E(\xi)\,| \,\le\, R_y(\xi)\,,$$

by introducing the polynomials

$$\begin{aligned} (P_x,P_y) \,&=\, w^2(d_x^2+d_y^2)\,(d_x,d_y)\,,\\ (Q_x,Q_y) \,&=\, 2w^2(d_yd_x'-d_xd_y')(d_y,d_x)\,,\\ (R_x,R_y) \,&=\, (d_x^2+d_y^2)^2(A_x,A_y)\,. \end{aligned} \tag{24}$$

These polynomials may be constructed from (17) and (18) using the standard arithmetic procedures [6, 9, 10, 22] for polynomials in Bernstein form. Note that $\deg(P_x,P_y)=10$, $\deg(Q_x,Q_y)=9$, and $\deg(R_x,R_y)=8$.

The derivative of (23) may be expressed as in Bernstein form as

$$E'(\xi) \,=\, \sum_{k=0}^{n-1} n(E_{k+1}-E_k)\,b_k^{n-1}(\xi)\,. \tag{25}$$

Having constructed the polynomials (24), the arithmetic rules for Bernstein–form polynomials may be invoked to express $G_x(\xi)=P_x(\xi)E'(\xi)+Q_x(\xi)E(\xi)$ and $G_y(\xi)=P_y(\xi)E'(\xi)+Q_y(\xi)E(\xi)$ as polynomials of the form

$$G(\xi) \,=\, \sum_{k=0}^{n+9} g_k(E_0,\ldots,E_n)\,b_k^{n+9}(\xi)\,, \tag{26}$$

of degree $n+9$ in ξ, with coefficients that are homogeneous linear expressions,

$$g_k(E_0,\ldots,E_n) \;=\; \sum_{i=0}^{n} g_{ki}E_i\,,$$

in $E_0,\ldots,E_n$. One can verify [10] that the coefficients g_{ki} are specified by the general form

$$g_{ki} \;=\; n\,\frac{\binom{10}{k+1-i}\binom{n-1}{i-1}}{\binom{9+n}{k}}\,P_{k+1-i} \;-\; n\,\frac{\binom{10}{k-i}\binom{n-1}{i}}{\binom{9+n}{k}}\,P_{k-i} \;+\; \frac{\binom{9}{k-i}\binom{n}{i}}{\binom{9+n}{k}}\,Q_{k-i}\,, \tag{27}$$

$P_0,\ldots\, P_{10}$ and $Q_0,\ldots,Q_9$ being the Bernstein coefficients of $P_x(\xi)$, $Q_x(\xi)$ or $P_y(\xi)$, $Q_y(\xi)$ according to whether (26) represents $G_x(\xi)$ or $G_y(\xi)$. However, for any given k, the three terms in (27) occur only for restricted ranges in i — $\max(1,k-9)\le i\le\min(n,k+1)$ for the first term; $\max(0,k-10)\le i\le\min(n-1,k)$ for the second; and $\max(0,k-9)\le i\le\min(n,k)$ for the third.

Now $R_x(\xi)$ and $R_y(\xi)$ in (24) are nominally degree 8, but can be degree–elevated to obtain their coefficients in the Bernstein basis of degree $n+9$ on $\xi\in[\,0,1\,]$. Using these degree–elevated coefficients and (26), the acceleration constraints for each axis can be written as

$$\sum_{k=0}^{n+9}\left(\sum_{i=0}^{n} g_{ki}E_i - R_k\right) b_k^{n+9}(\xi) \;\le\; 0\,, \quad \sum_{k=0}^{n+9}\left(\sum_{i=0}^{n} g_{ki}E_i + R_k\right) b_k^{n+9}(\xi) \;\ge\; 0\,. \tag{28}$$

Such inequalities hold for both the x and the y axis, using the coefficients $P_0,\ldots,P_{10}$ and $Q_0,\ldots,Q_9$ in (27) and $R_0,\ldots,R_{n+9}$ appropriate to that axis. Since these constraints are difficult to enforce in a pointwise manner — i.e., for each $\xi\in[\,0,1\,]$ — we invoke the attractive properties of Bernstein–form polynomials to apply them in a rapidly–convergent approximate manner.

The *control polygon* of a polynomial $f(\xi)$ with coefficients $c_0,\ldots,c_m$ in the degree–m Bernstein basis on $\xi\in[\,0,1\,]$ is defined by connecting the control points $\mathbf{c}_k=(k/m,c_k)$ for $k=0,\ldots,m$ in order, and the graph of $f(\xi)$ always lies within the *convex hull* of its control points. This is a consequence of the fact that the basis functions (11) are *non–negative* on $\xi\in[\,0,1\,]$ and they form a *partition of unity.* The *de Casteljau algorithm* subdivides $f(\xi)$ at any point $\xi_*\in(0,1)$ to yield "left" and "right" control polygons, representing $f(\xi)$ on the sub–intervals $[\,0,\xi_*\,]$ and $[\,\xi_*,1\,]$. Setting $c_j^0=c_j$ for $j=0,\ldots,m$ this computes a triangular array of values c_j^r through the interpolations

$$c_j^r \;=\; (1-\xi_*)\,c_{j-1}^{r-1} \;+\; \xi_*\,c_j^{r-1} \tag{29}$$

for $j = r, \ldots, m$ and $r = 1, \ldots, m$. Then $f(\xi_*) = c_m^m$, while $c_0^0, c_1^1, c_2^2, \ldots, c_m^m$ and $c_m^m, c_m^{m-1}, c_m^{m-2}, \ldots, c_m^0$ are the coefficients of $f(\xi)$ on $[\,0, \xi_*\,]$ and $[\,\xi_*, 1\,]$. See [5] for further details on the properties of Bernstein–form polynomials.

4.2 Linearization of cost function

Although the constraint inequalities are linear in $E_0, \ldots, E_n$ the cost function T defined by (1) is not. Nevertheless, T has some useful properties that help simplify the optimization. From (1) and (23), the first and second partial derivatives of this cost function are

$$\frac{\partial T}{\partial E_i} \,=\, -\int_0^1 \frac{\sigma(\xi)\, b_i^n(\xi)}{[\,2\,E(\xi)\,]^{3/2}}\,\mathrm{d}\xi\,, \qquad 0 \le i \le n\,, \tag{30}$$

and

$$\frac{\partial^2 T}{\partial E_i \partial E_j} \,=\, 3\int_0^1 \frac{\sigma(\xi)\, b_i^n(\xi)\, b_j^n(\xi)}{[\,2\,E(\xi)\,]^{5/2}}\,\mathrm{d}\xi\,, \qquad 0 \le i, j \le n\,.$$

Since $\sigma(\xi)$ and $E(\xi)$ are by definition non–negative on $[\,0, 1\,]$ the first partial derivatives are always non–positive, while the second partial derivatives are always non–negative (i.e., the Hessian matrix of T has non–negative entries). Thus, T is monotonically decreasing with respect to each of $E_0, \ldots, E_n$, which implies that its global minimum lies on a constraint boundary. Furthermore, the non–negative nature of the second derivatives indicates that T is a *convex* function of $E_0, \ldots, E_n$ implying that a local minimum is a global minimum. Minimization of a monotone convex cost function is a less difficult task than minimization of a general non–linear function.

The linear constraints determine a convex polytope in the Euclidean space $\mathbb{R}^{n+1}$ with coordinates $E_0, \ldots, E_n$ containing all instances of the function (23) consistent with the axis acceleration constraints. Under iterated subdivision, the control polygons converge monotonically to the graph of $E(\xi)$ on $\xi[\,0, 1\,]$. The number of planar facets defining the feasible polytope essentially doubles with each subdivision step, which generates new coefficients specifying $E(\xi)$ that are linear combinations of the original coefficients $E_0, \ldots, E_n$.

The monotonicity of T with respect to the coefficients of (23) ensures that its minimum occurs on the boundary of the feasible polytope. Formally, this point can be identified as the solution of a *convex optimization* problem, since (1) is a convex function and the feasible region is convex. For simplicity, we take the additional step here of making a linearized approximation of (1), allowing standard *linear programming* (LP) methods to be used — for an LP problem, the minimum occurs at a vertex of the feasible polytope. If $\tilde{T}$ is the value of (1) at a particular point $(\tilde{E}_0,\ldots,\tilde{E}_n)$ about which it is to be linearized, we may write

$$T(E_0,\ldots,E_n) \,=\, T(\tilde{E}_0,\ldots,\tilde{E}_n) \,+\, \sum_{i=0}^{n} \frac{\partial T}{\partial E_i}(E_i-\tilde{E}_i)\,, \tag{31}$$

where the derivatives $\partial T/\partial E_i$ specified by (30) are evaluated at $(\tilde{E}_0,\ldots,\tilde{E}_n)$ — these derivatives must be evaluated by numerical quadrature.

Linearization about a point other than the true solution may cause the LP method to identify a vertex somewhat different from the true minimum. Since the cost function is monotone, the sign of the derivative of (31) with respect to each of $E_0,\ldots,E_N$ is always be consistent with that of the exact cost function (1). Also, since (1) is a convex function, the true minimum may not be at a vertex. The LP method will always guarantee a feasible solution, typically close to but not necessarily identical with the true optimum.

4.3 Formulation of LP constraints

Each axis incurs two acceleration constraints of the form (28). Instead of attempting to impose these pointwise constraints exactly, we take advantage of the convex hull and subdivision properties of the Bernstein form to obtain a convergent sequence of approximations to them, that are linear in $E_0,\ldots,E_n$. Moreover, these approximations are conservative, so one can be certain that the exact acceleration bounds are never exceeded.

Prior to any subdivision of the $[\,0,1\,]$ domain, imposing the conditions

$$\sum_{i=0}^{n} g_{ki}E_i \,\le\, R_k \quad \text{and} \quad \sum_{i=0}^{n} g_{ki}E_i \,\ge\, -\,R_k \quad \text{for} \;\; k=0,\ldots,n+9$$

on the coefficients of the constraints (28) ensures their satisfaction *a fortiori*. These conditions can be expressed in the standard LP form $\mathbf{A}\,\mathbf{x} \le \mathbf{b}$, where $\mathbf{x}$ is the vector of free parameters, the matrix $\mathbf{A}$ specifies their coefficients in the linear constraints, and $\mathbf{b}$ is the right–hand–side vector of constants.

These "initial" constraints are generally too restrictive, and several stages of subdivision will typically be required to identify a feasible solution giving a significant reduction of the traversal time T. Under repeated subdivision, the control polygons converge to the graphs of the exact polynomial constraints (28). Subdivision may be implemented through the de Castlejau algorithm. Alternatively, the vector $\bar{\mathbf{c}}$ of Bernstein coefficients on a sub–interval $[\,\xi_1, \xi_2\,]$ may be obtained from the coefficients $\mathbf{c}$ on $[\,0, 1\,]$ by a matrix multiplication $\bar{\mathbf{c}} = \mathbf{M}\mathbf{c}$, the elements of the subdivision matrix $\mathbf{M}$ being defined [8] by

$$M_{jk} \,=\, \sum_{i=\max(0,j+k-n)}^{\min(j,k)} b^i_j(\xi_2)\, b^{n-j}_{k-i}(\xi_1)\,, \qquad 0 \le j, k \le n\,.$$

Now if $\mathbf{G}$ is $(n+10) \times (n+1)$ matrix of the coefficients g_{ki} on $[\,0, 1\,]$, then

$$\bar{\mathbf{G}} \,=\, \mathbf{M}\,\mathbf{G} \tag{32}$$

is the corresponding matrix for the sub–interval $[\,\xi_1, \xi_2\,]$.

When $[\,0, 1\,]$ is subdivided into a set of m contiguous sub–intervals $[\,0, \xi_1\,]$, $[\,\xi_1, \xi_2\,]$, . . ., $[\xi_{m-1}, 1]$ there is an essentially m–fold increase in the number of constraints. However, the constraint corresponding to the initial coefficient on $[\,\xi_k, \xi_{k+1}\,]$ is identical to that corresponding to the final coefficient on the preceding interval $[\,\xi_{k-1}, \xi_k\,]$, and should be omitted for $k = 1, \ldots, m-1$. Hence, each of the two constraints (28) yields $(n+9)m+1$ conditions for each of the two axes, or $4\,[\,(n+9)m+1\,]$ linear inequalities altogether on the coefficients $E_0, \ldots, E_n$.

5 Illustrative examples

The following examples illustrate representative results from a preliminary implementation of the high–speed cornering scheme. In both examples, the degree of the polynomial (23) used in optimizing the feedrate over the conic smoothing segment is $n = 16$, and two levels of subdivision (through the de Casteljau algorithm) were invoked.

Example 1 Consider the corner traversal specified by the parameters $\theta_1 = 200°$, $\theta_2 = -90°$, $V_* = 25$ mm/s, $A_x = A_y = 2000$ mm/s^2, $\epsilon = 0.015$ mm. The conic smoothing segment is defined by values $w_1 = 2$, $\ell_1 = 0.07607$ mm, $\ell_2 = 0.08095$ mm. The feedrate optimization process for this segment yields entry and exit values $V_1 = 17.703$ mm/s, $V_2 = 19.289$ mm/s and hence from (7) we obtain $b_1 = 0.07320$ mm, $b_2 = 0.06323$ mm and $d_1 = 0.14683$ mm, $d_2 = 0.15625$ mm. In accordance with the criteria presented in Section 3.1, we choose overall dimensions $L_1 = 0.14927$ mm, $L_2 = 0.15625$ mm to isolate the path corner, corresponding to the values $a_1 = 0$ mm, $c_1 = 0.00244$ mm, $a_2 = 0.01207$ mm, $c_2 = 0$ mm specified through (8).

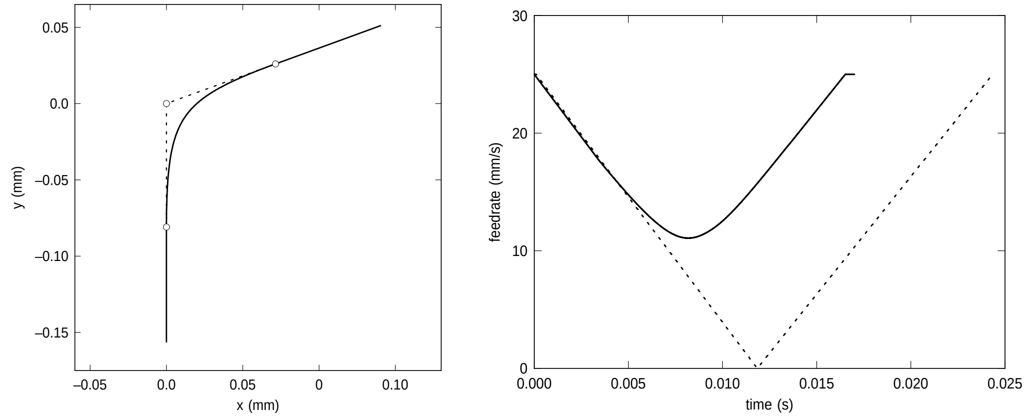

Figure 3: Left: comparison of exact path corner (dashed line) and smoothed corner (solid line) for Example 1 — the entire extents L_1, L_2 of the isolated incoming/outgoing corner segments are shown. Right: the feedrate variation along the original corner (dashed line) and smoothed corner (solid line) — note that the traversal time for the latter is $\sim 30\%$ less than for the former.

Figure 3 compares the original and smoothed path corners, together with the corresponding feedrate variations. The overall corner traversal times, as determined from expressions (9) and (10), are $T_c = 0.02434$ s for the exact corner, and $T_s = 0.01700$ s for the smoothed corner. The smoothed corner thus gives a reduction of $\sim 30\%$ in the overall cornering time.

Figure 4 shows that the specifed bound of 2000 mm/s^2 on the magnitude of the x and y axis accelerations is observed throughout the execution of the smoothed corner. Figure 4 also shows the "second hodograph" — i.e., the locus in the (x, y) plane traced by the acceleration vector — during execution of the conic segment. This is seen to conform closely to the perimeter of the rectangular region defined by $|a_x| \le 2000$ mm/s^2 and $|a_y| \le 2000$ mm/s^2, as would be expected for the exact "bang–bang" solution to the acceleration–constrained time–optimal control problem.

Example 2 The parameters specifying the second corner traversal are $\theta_1 = 20°$, $\theta_2 = -105°$, $V_* = 20$ mm/s, $A_x = A_y = 2000$ mm/s^2, $\epsilon = 0.02$ mm.

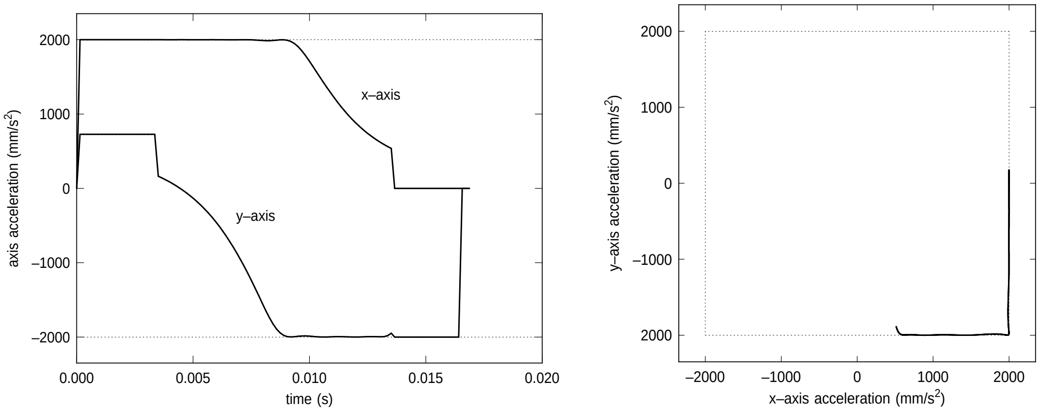

Figure 4: Left: x and y axis accelerations during execution of the smoothed corner in Example 1, in conformity with the prescribed bounds $\pm 2000\,\mathrm{mm/s}^2$. Right: the second hodograph (i.e., the locus traced by the acceleration vector) during traversal of the conic segment in Example 1, showing that the feedrate optimization yields a result closely approximating the "bang–bang" solution.

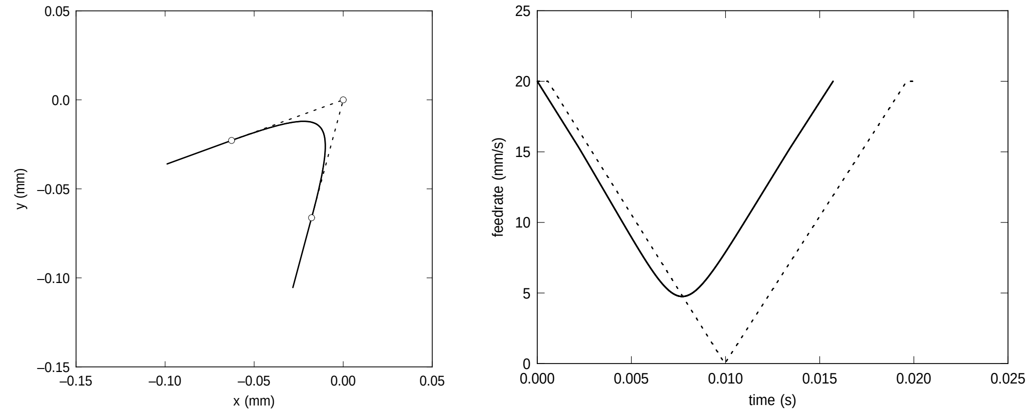

Figure 5: Left: comparison of exact path corner (dashed line) and smoothed corner (solid line) for Example 2 — the entire extents L_1, L_2 of the isolated incoming/outgoing corner segments are shown. Right: the feedrate variation along the original corner (dashed line) and smoothed corner (solid line) — in this case, the reduction in the overall corner traversal time is only $\sim 8\%$.

The conic smoothing segment is defined by $w_1 = 2$, $\ell_1 = 0.06672$ mm, $\ell_2 = 0.06858$ mm. From the feedrate optimization, we obtain $V_1 = 15.365$ mm/s, $V_2 = 15.232$ mm/s and (7) thus gives $b_1 = 0.03851$ mm, $b_2 = 0.04057$ mm and $d_1 = 0.09397$ mm, $d_2 = 0.09659$ mm. Invoking the criteria of Section 3.1, we choose overall dimensions $L_1 = 0.10523$ mm, $L_2 = 0.10915$ mm for the path corner, corresponding to values $a_1 = 0$ mm, $c_1 = 0.01126$ mm, $a_2 = 0$ mm, $c_2 = 0.01256$ mm specified through (8). In this example, the overall traversal times defined by (9) and (10) are $T_c = 0.02025$ s for the exact corner, and $T_s = 0.01571$ s for the smoothed corner, so the reduction in the cornering time is more modest ($\sim 22\%$) than in the previous example.

Figure 5 compares the original and smoothed corners, and corresponding feedrate variations. Again, the axis accelerations along the smoothed path, shown in Figure 6, lie within the ±2000 mm/s bounds. Figure 6 also shows the "second hodograph" for the conic smoothing segment of the path.

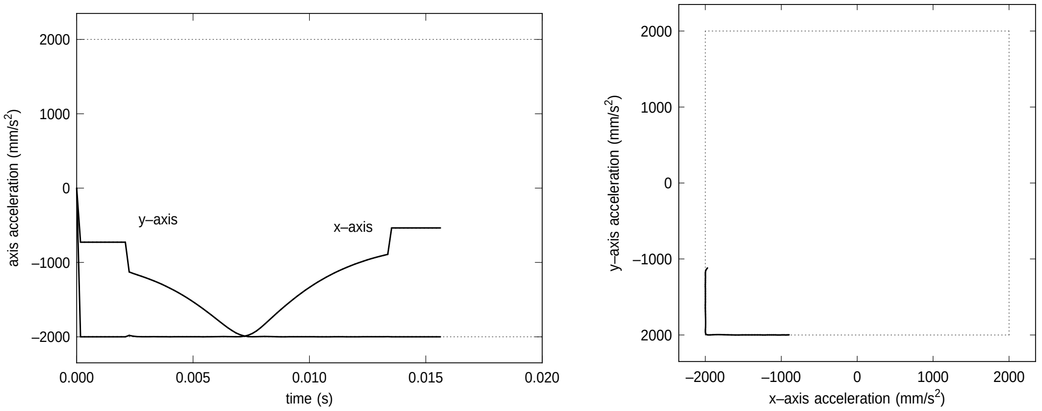

Figure 6: Left: x and y axis accelerations during execution of the smoothed corner in Example 2, in conformity with the prescribed bounds ±2000 mm/s^2. Right: the second hodograph (i.e., the locus traced by the acceleration vector) during traversal of the conic segment in Example 2, showing that the feedrate optimization yields a result closely approximating the "bang–bang" solution.

6 Closure

A method for computing *a priori* modifications to CNC part programs that include sharp toolpath corners, so as to enable faster corner execution while observing a prescribed geometrical contour error and bounds on the machine axis accelerations, has been proposed. The sharp corner is smoothed by the insertion of a conic smoothing segment, and the feedrate variation on this segment is determined by minimizing the traversal time, under the constraint of the axis acceleration bounds. By exploiting the advantageous properties of Bernstein–form poylnomials, solutions can be obtained using standard linear programming methods. Computed examples were presented to illustrate the implementation of the method, and its performance in typical examples.

The corner modification scheme is intended as on off–line pre–processing application, to modify the geometry of the toolpath and associated feedrates prior to real–time execution on a CNC machine. By cataloging the results for various corner geometries, nominal feedrates, acceleration bounds, contour error tolerances, etc., it should be possible to determine suitable parameters for arbitrary cornering tasks by an interpolation scheme, rather than running the optimization algorithm for each new cornering problem.

References

[1] G. Dahlquist and A. Björck (1974), *Numerical Methods*, Prentice–Hall, Englewood Cliffs, NJ.

[2] A. F. de Souza and R. T. Coelho (2007), Experimental investigation of feed rate limitations on high speed milling aimed at industrial applications, *International Journal of Advanced Manufacturing Technology* **32**, 1104–1114.

[3] K. Erkorkmaz, C–H. Yeung, and Y. Altintas (2006), Virtual CNC system. Part II. High speed contouring application, *International Journal of Machine Tools and Manufacture* **46**, 1124–1138.

[4] C. A. Ernesto and R. T. Farouki (2010), Solution of inverse dynamics problems for contour error minimization in CNC machines, *International Journal of Advanced Manufacturing Technology* **49**, 589–604.

[5] G. Farin (1997), *Curves and Surfaces for Computer Aided Geometric Design* (4th Edition), Academic Press, San Diego.

[6] R. T. Farouki and T. N. T. Goodman (1996), On the optimal stability of the Bernstein basis, *Mathematics of Computation* **65**, 1553–1566.

[7] R. T. Farouki, C. Manni, and A. Sestini (2001), Real–time CNC interpolators for Bézier conics, *Computer Aided Geometric Design* **18**, 639–655.

[8] R. T. Farouki and C. A. Neff (1990), On the numerical condition of Bernstein–Bézier subdivision processes, *Mathematics of Computation* **55**, 637–647.

[9] R. T. Farouki and V. T. Rajan (1987), On the numerical condition of polynomials in Bernstein form, *Computer Aided Geometric Design* **4**, 191–216.

[10] R. T. Farouki and V. T. Rajan (1988), Algorithms for polynomials in Bernstein form, *Computer Aided Geometric Design* **5**, 1–26.

[11] H. Halkin (1965), A generalization of LaSalle's "bang–bang" principle, *SIAM Journal on Control* **2**, 199–202.

[12] F. Hausdorff (1957), *Set Theory* (translated by J. R. Aumann et al.), Chelsea, New York.

[13] B. M. Imani and J. Jahanpour (2008), High–speed contouring enhanced with PH curves, *International Journal of Advanced Manufacturing Technology* **37**, 747–759.

[14] J. Jahanpour and B. M. Imani (2008), Real–time PH curve CNC interpolators for high speed cornering, *International Journal of Advanced Manufacturing Technology* **39**, 302–316.

[15] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High Speed Machining*, PED–Vol. 12, ASME, New York.

[16] J. P. LaSalle (1960), The time optimal control problem, in *Contributions to the Theory of Nonlinear Oscillations* (L. Cesari, J. P. LaSalle, and S. Lefschetz, eds.), Vol. 5, Princeton Univ. Press.

[17] E. T. Y. Lee (1987), The rational Bézier representation for conics, in *Geometric Modeling: Algorithms and New Trends* (G. E. Farin, ed.), SIAM, Philadelphia.

[18] H. Schultz and T. Moriwaki (1992), High–speed machining, *Annals of the CIRP* **41**, 637–643.

[19] S. Smith and J. Tlusty (1997), Current trends in high–speed machining, *ASME Journal of Manufacturing Science and Engineering* **119**, 664–666.

[20] J. Tlusty (1993), High–speed machining, *CIRP Annals* **42**, 733–738.

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

[22] Y–F. Tsai and R. T. Farouki (2001), Algorithm 812: BPOLY: An object–oriented library of numerical algorithms for polynomials in Bernstein form, *ACM Transactions on Mathematical Software* **27**, 267–296.