Int J Adv Manuf Technol
DOI 10.1007/s00170-015-7969-2

ORIGINAL ARTICLE

Minimum-time cornering for CNC machines using an optimal control method with NURBS parameterization

Molong Duan[1] · Chinedum Okwudire[1]

Received: 5 June 2015 / Accepted: 12 October 2015

Abstract Speed and accuracy are conflicting demands that must be met by manufacturing machines. This conflict is very frequently encountered when executing sharp corners where, to achieve tight tolerances, machine axes have to slow down considerably thus sacrificing cycle time. The most common way of reducing this tradeoff between speed and accuracy is to smooth the corner using a pre-specified curve (e.g., a spline) that allows high-speed cornering subject to path tolerance and machine kinematic limits. This paper presents a different corner-smoothing approach where the smoothing curve is not specified a priori. Instead, an optimal control method is used to generate the best free-form curve that minimizes corner traversal time while adhering to path tolerance and kinematic constraints. The resulting optimal free-form curve is parameterized using an enhanced NURBS curve fitting process to allow its accurate execution in a CNC. Illustrative examples are presented to demonstrate the merits of the proposed approach.

Keywords CNC · Cycle time · Corner smoothing · Optimal control · NURBS

1 Introduction

The problem of accurate and time-optimal execution of sharp corners has received plenty of attention in robotics and manufacturing literature due to its influence on the overall accuracy and speed of robots and manufacturing machines [1–10]. To perfectly execute a sharp corner, a machine must come to a complete stop at the corner while adhering to specified kinematic limits. However, manufacturing machines seldom need to execute perfectly sharp corners. Instead, they are often required to traverse the corner as fast as possible subject to the tolerance constraints of the manufacturing process and the kinematic limits of the machine [2, 5]. It is therefore common practice to smooth a corner with a continuous curve that eliminates the need for the machine to come to a complete stop, thus increasing cornering speed while keeping to tolerance requirements.

A large number smoothing techniques in the literature are focused on short line segments (often less than 50 μm in length) that are artificially formed by interpolating smooth curves using numerous tiny straight lines (e.g., [7–9]). The key goal of such techniques is to “compress” the numerous tiny lines into a few smooth curves while keeping to path tolerance specifications. This paper is however focused on a different set of smoothing techniques that address corners formed by the intersection of bona fide straight lines, with lengths often greater than 1 mm (e.g., [1–6]). For such long line segments, smoothing is highly localized to a corner region that is typically a small fraction of the total length of the line. The standard procedure in existing corner smoothing techniques for long line segments is to first specify the geometry of the smoothing curve in parametric form and then optimize the motion along the curve. For instance, Jouaneh et al. [1] employed a pair of clothoid curves at sharp corners to facilitate smooth tracking motion in robots. Barre et al. [2] proposed a heuristic corner-smoothing curve based on race car piloting. Erkorkmaz and Altintas [3] used quintic splines, while Imani and Jahanpour [4] employed P-H curves at sharp corners to improve tracking accuracy in computerized numerical control (CNC) machine tools. Beudaert et al. [6] tackled the problem of smoothing sharp corners in five-axis motion profiles by

✉ Chinedum Okwudire
okwudire@umich.edu

[1] University of Michigan, Ann Arbor, MI, USA

Published online: 07 November 2015

using a pair of B-spline curves. Ernesto and Farouki [5] addressed the problem of traversing a sharp corner in minimum time subject to tolerance and machine kinematics constraints. They replaced the sharp corner by a conic parameterized as a rational quadratic Bezier curve and determined the feed rate profile along the curve that minimized corner traversal time. The resulting nonlinear constrained optimization problem was linearized so that near-optimal solutions could be obtained using well-established linear programming methods. In a similar vein, Sencer et al. [10] proposed smoothing sharp corners with B-splines whose largest curvatures were minimized to facilitate higher cornering speeds. The problem with existing techniques is that the geometric profile selection is separated from the feed rate optimization, which may unnecessarily limit the achievable cornering speed. Therefore, in this paper, we present a different approach where the smoothing curve is not parametrically specified before solving the minimum-time cornering problem. Instead, in Sect. 2, we make a useful simplification to the problem that allows us in Sect. 3 to re-cast it as a two-point boundary value problem (TPBVP) based on Pontryagin's minimum principle from optimal control theory. This approach removes the constraints imposed on the optimization problem by pre-specifying the smoothing curve. Section 4 tackles practical issues related to the solution of the TPBVP and proposes a novel technique for parameterizing the resulting optimal free-form curve using nonuniform rational B-spline (NURBS) so that it is usable in CNCs. In Sect. 5, illustrative examples are presented to compare the performance of the proposed method to unsmoothed corners as well as to Ernesto and Farouki's [5] smoothing approach, followed by discussions and conclusions.

2 Formulation of minimum-time cornering problem

2.1 Exact formulation

Consider the planar cornering scenario shown in Fig. 1, arising from the intersection of two line segments generated by perpendicular axes (x and y) of a manufacturing machine. $\mathbf{p}_1=\{0,0\}^T$ is the resulting sharp corner, while angles θ_1 and θ_2 respectively define the orientations of the incoming and outgoing line segments. Based on the unit vectors $\mathbf{t}_1=\{\cos\theta_1, \sin\theta_1\}^T$ and $\mathbf{t}_2=\{\cos\theta_2, \sin\theta_2\}^T$ along each line, two boundary points $\mathbf{p}_0=\mathbf{p}_1+l_1\mathbf{t}_1$ and $\mathbf{p}_2=\mathbf{p}_1+l_2\mathbf{t}_2$ are specified to mark the start and the end of the corner region surrounding $\mathbf{p}_1$. The exact corner path can therefore be defined, per Ernesto and Farouki [5], by the point set

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

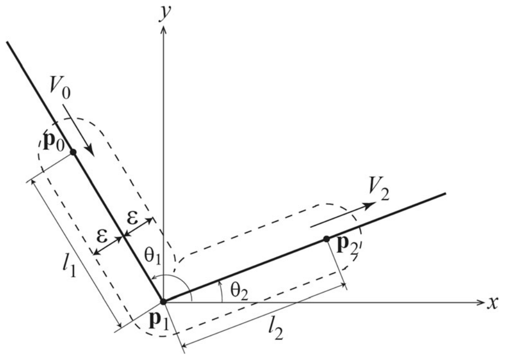


Fig. 1 Corner region of two intersecting lines

For a manufacturing machine to perfectly negotiate the sharp corner defined in Eq. (1), it would have to come to a complete stop at $\mathbf{p}_1$ before continuing on to $\mathbf{p}_2$, thus severely sacrificing cornering speed. However, having a perfectly sharp corner is often not required in manufacturing processes. It is instead permissible for the actual path to deviate from the exact corner within specified tolerance constraints. For this reason, it is customary to modify the corner trajectory such that the machine traverses it as fast as possible while keeping to the speed and tolerance constraints of the manufacturing process, as well as to the kinematic limits of the machine axes. Let the modified trajectory be defined by the point set

$$\mathbf{r}(t) = \{x(t), y(t)\}^T \quad \forall\ t\in[t_0, t_2] \tag{2}$$

where t_0 and t_2 are respectively the time instants at $\mathbf{p}_0$ and $\mathbf{p}_2$. To satisfy continuity requirements at $\mathbf{p}_0$ and $\mathbf{p}_2$, $\mathbf{r}(t)$ must satisfy the conditions

$$\mathbf{r}(t_0) = \mathbf{p}_0; \quad \mathbf{r}(t_2) = \mathbf{p}_2 \tag{3}$$

Similarly, $d\mathbf{r}(t)/dt$ must satisfy the end conditions

$$\frac{d\mathbf{r}(t_0)}{dt} = -V_0\mathbf{t}_1; \quad \frac{d\mathbf{r}(t_2)}{dt} = V_2\mathbf{t}_2 \tag{4}$$

where V_0 and V_2 are the speeds at $\mathbf{p}_0$ and $\mathbf{p}_2$, respectively. Given the maximum allowable corner error, ε, the tolerance constraint on $\mathbf{r}(t)$ can be expressed [5] as

$$d_H\left(\mathbf{r}, \hat{\mathbf{r}}\right) \leq \varepsilon \tag{5}$$

with d_H being the Hausdorff distance defined as

$$d_H\left(\mathbf{r}, \hat{\mathbf{r}}\right) = \max\left(\sup_{\mathbf{m}\in\hat{\mathbf{r}}} \inf_{\mathbf{n}\in\mathbf{r}} d(\mathbf{m},\mathbf{n}), \sup_{\mathbf{m}\in\mathbf{r}} \inf_{\mathbf{n}\in\hat{\mathbf{r}}} d(\mathbf{m},\mathbf{n})\right) \tag{6}$$

where d is the Euclidean distance. The constraint of Eq. (6) implies that the modified corner trajectory must lie within the dashed lines surrounding the exact corner, as shown in Fig. 1. Additionally, the trajectory must satisfy the kinematic constraints

$$\left|\frac{d\mathbf{r}(t)}{dt}\right| \leq V_{\max} \tag{7}$$

$$-\left\{\begin{matrix} V_x \\ V_y \end{matrix}\right\} \leq \frac{d\mathbf{r}(t)}{dt} \leq \left\{\begin{matrix} V_x \\ V_y \end{matrix}\right\} \tag{8}$$

and

$$-\left\{\begin{matrix} A_x \\ A_y \end{matrix}\right\} \leq \frac{d^2\mathbf{r}(t)}{dt^2} \leq \left\{\begin{matrix} A_x \\ A_y \end{matrix}\right\} \tag{9}$$

where $V_{\max}$ is the maximum allowable speed along the path, while V_x, V_y and A_x, A_y are the velocity and acceleration limits of the x and y axes, respectively. Notice that there is some redundancy in the velocity constraints of Eqs. (7) and (8). The axis velocity limits of Eq.(8) are typically governed by the velocity limits of the axis actuators, while the path velocity constraint of Eq.(7) is dictated by the feed rate specified in the numerical control (NC) code for the manufacturing process. If the feed rate in the NC code is properly planned, it would not exceed the axis velocity limits, meaning that only Eq. (7) would have to be satisfied. However, adding Eq. (8) provides a safeguard (i.e., redundancy) in case the feed rate in the NC code is improperly specified, causing the axes to exceed their velocity limits.

The optimal trajectory $\mathbf{r}(t)$ is the one that minimizes the objective function

$$J = \int_{t_0}^{t_2} dt \tag{10}$$

subject to the constraints defined in Eqs. (3) through (9). The solution of this optimization problem is extremely challenging, in large part due to the complicated point-wise geometric constraints imposed by the Hausdorff distance defined in Eq.(6), which requires the entire corner trajectory to be known beforehand. A simplification of the problem is therefore needed to facilitate its solution.

2.2 Simplified formulation

The standard approach for simplifying the minimum-time cornering problem defined above is to separate its geometric constraints (i.e., Eqs. (3) and (5)) from its kinematic constraints (i.e., Eqs. (4) and (7)–(9)). The geometric shape of the modified trajectory is pre-defined using a parametric curve $\mathbf{r}(\xi)$ (where $\xi \in [0, 1]$ is the curve parameter) such that it satisfies the geometric constraints. With the path now fixed, the objective function can be reduced to

$$J = \int_{t_0}^{t_2} dt = \int_0^1 \left|\frac{d\mathbf{r}(\xi)}{d\xi}\right| \frac{1}{V(\xi)} d\xi \tag{11}$$

from which the optimal speed profile $V(\xi)$ that satisfies the established kinematic constraints is determined. This approach, generally known as optimal feed rate scheduling, is widely used in trajectory planning for robotics and manufacturing applications, e.g., [11–14], and has recently been applied to the minimum-time cornering problem by Ernesto and Farouki [5].

While pre-defining the corner path greatly simplifies the optimization problem, it could lead to sub-optimal solutions because it does not guarantee that the selected curve is most favorable for the optimization. A less rigid simplification to the minimum-time cornering problem is proposed in this paper, based on preliminary work reported by the authors in [15]. It is assumed that if the tolerance condition of Eq.(5) is satisfied at the corner point $\mathbf{p}_1$, then it is satisfied everywhere else on the point set $\mathbf{r}(t)$. Therefore, the position and slope of $\mathbf{r}(t)$ at $\mathbf{p}_1$ are selected such that

$$\mathbf{r}(t_1) = \{\varepsilon\cos\beta, \varepsilon\sin\beta\}^T; \quad \beta \triangleq \frac{\theta_1 + \theta_2}{2} \tag{12}$$
$$\frac{\dot{\mathbf{r}}\,(t_1)}{|\dot{\mathbf{r}}\,(t_1)|} = dir\cdot\{\sin\beta, -\cos\beta\}^T$$

where dir=sgn($\{\sin\beta, -\cos\beta\}^T\{\cos\theta_2, \sin\theta_2\}$) and t_1 is the time instant at $\mathbf{p}_1$. Equation (12) implies that the velocity at the critical point must be perpendicular to the bisector of the corner and be directed towards $\mathbf{p}_2$. This allows $\mathbf{r}(t)$ to be split into two portions, $\mathbf{r}_1(t)$ for $t \in [t_0, t_1]$ and $\mathbf{r}_2(t)$ for $t \in [t_1, t_2]$, whose geometries are not pre-determined but instead are allowed to vary according to the dictates of the optimization. Note that Bosetti and Bertolazzi [16] have recently proposed a simplification approach for optimal trajectory generation that bears some similarity to the one described in Eq.(12). However, in their approach, which is not focused on the cornering problem addressed in this paper, $\mathbf{r}(t)$ is forced to pass through point $\mathbf{p}_1$ instead of the critical point $\mathbf{r}(t_1)$ defined in Eq.(12). It should be obvious to the reader that such an assumption is detrimental to minimum-time cornering compared to the assumption made in Eq.(12), even though it greatly simplifies the general problem formulation studied in [16].

We observe empirically that the length l_i (i=1, 2) is critical for the assumption made in Eq.(12) to be valid. The reason is that the optimization seeks to minimize the curvature along the corner path so that higher speeds can be achieved while keeping to the imposed acceleration and endpoint constraints. If l_i is small enough, the endpoint constraints combined with curvature restraints are sufficient to prevent the optimized path from violating the tolerance conditions at points other than the critical point defined in Eq.(12). Therefore, in the appendix, we provide an estimation of the maximum l_i (i.e., $l_{i,\max}$) that ensures that our assumption is valid for a given cornering problem.

3 Solution of minimum-time cornering problem

The field of optimal control deals with the solution of dynamic optimization problems like the one formulated in the previous section. Generally speaking, there are two classes of solution methods available: direct and indirect [17]. Direct methods discretize the problem and solve it numerically using nonlinear programming. Indirect methods, on the other hand, convert the optimal control problem to a boundary value problem by analytically providing conditions for optimality. Direct methods are in general more robust than indirect methods. However, they are well known to suffer from high computational costs, especially if a solution with high resolution is desired [18]. In our preliminary investigation [15] using dynamic programming (a direct method), we show that the computational burden of dynamic programming makes it impractical for solving the minimum-time cornering problem with sufficient resolution. In this paper, we propose a solution to the optimal cornering problem that combines the merits of direct and indirect methods. Pontryagin's minimum principle, the basis of all indirect methods, is applied to the minimum-time cornering problem thereby converting it into a TPBVP which is solved using a search-based shooting algorithm. To enhance the robustness of the TPBVP solution, a documentation of initial conditions using a direct method is proposed.

3.1 Theoretical overview of PMP

Given a dynamical system governed by the equation,

$$\dot{\mathbf{z}}\,(t) = \mathbf{f}(\mathbf{z}(t), \mathbf{u}(t), t), \quad t \in [t_0, t_f], \tag{13}$$

$$\mathbf{z}(t_0) = \mathbf{z}_0, \quad \mathbf{u}(t) \in U$$

where $\mathbf{z}$ is the state vector and U is the compact set of admissible control inputs $\mathbf{u}(t)$ that must be chosen to minimize the objective functional J given in general form by the equation

$$J = K(\mathbf{z}(t_f), t_f) + \int_{t_0}^{t_f} L(\mathbf{z}(t), \mathbf{u}(t), t)dt \tag{14}$$

where K and L are scalar functions that depend on the optimization problem to be solved. Let the Hamiltonian H be defined as

$$H \triangleq L(\mathbf{z}(t), \mathbf{u}(t), t) + \boldsymbol{\lambda}^T(t)\mathbf{f}(\mathbf{z}(t), \mathbf{u}(t), t) \tag{15}$$

where $\boldsymbol{\lambda}$ is the so-called co-state vector of the system. Pontryagin's minimum principle (PMP) states that the optimal state trajectory $\mathbf{z}^*(t)$, co-state trajectory $\boldsymbol{\lambda}^*(t)$, and control input $\mathbf{u}^*(t)$ must satisfy the conditions [17]

$$\dot{\mathbf{z}}\,^*(t) = \frac{\partial H}{\partial \mathbf{p}}(\mathbf{z}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{16}$$

$$\dot{\boldsymbol{\lambda}}\,^*(t) = -\frac{\partial H}{\partial \mathbf{z}}(\mathbf{z}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \tag{17}$$

$$H(\mathbf{z}^*(t), \mathbf{u}^*(t), \boldsymbol{\lambda}^*(t), t) \le H(\mathbf{z}^*(t), \mathbf{u}(t), \boldsymbol{\lambda}^*(t), t) \tag{18}$$

for all $t \in [t_0, t_f]$ and $\mathbf{u} \in U$. Additionally, the end conditions t_f and $\mathbf{z}(t_f)=\mathbf{z}_f$ should satisfy the condition

$$\left[\frac{dK}{d\mathbf{z}}(\mathbf{z}^*(t_f), t_f) - \boldsymbol{\lambda}^*(t_f)\right]^T \delta\mathbf{z}_f + \left[H(\mathbf{z}^*(t_f), \mathbf{u}^*(t_f), \boldsymbol{\lambda}^*(t_f), t_f) + \frac{dK}{dt}(\mathbf{z}^*(t_f), t_f)\right]\delta t_f = 0 \tag{19}$$

where $\delta\mathbf{z}_f$ and δt_f represent the variations of $\mathbf{z}_f$ and t_f, respectively [17]. $\delta\mathbf{z}_f$ and δt_f are zero if $\mathbf{z}_f$ and t_f are fixed (i.e., specified).

3.2 Application of PMP to the minimum-time cornering problem

3.2.1 Derivation of system dynamics

The first step in applying PMP to the minimum-time cornering problem defined in Sect. 2 is to formulate the system dynamics in the form of Eq.(13); i.e.,

$$\begin{aligned}\dot{\mathbf{z}} &= \{\dot{z}_1 \quad \dot{z}_2 \quad \dot{z}_3 \quad \dot{z}_4 \quad \dot{z}_5 \quad \dot{z}_6 \quad \dot{z}_7\}^T \\ &= \{\dot{x} \quad u_1 \quad \dot{y} \quad u_2 \quad h_1 \quad h_2 \quad h_3\}^T = \mathbf{f}(\mathbf{z}, \mathbf{u}, t)\end{aligned} \tag{20}$$

where $u_1 = \ddot{x}$ and $u_2 = \ddot{y}$ are the control inputs which are respectively bounded by the axis acceleration limits A_x and A_y, as defined in Eq.(9). The variable h_j (j=1, 2, 3) is defined by a single-sided penalty function [17] designed to enforce the velocity constraints of Eqs.(7) and (8). It is given by

$$h_j = \begin{cases} 0 & s_j < 0 \\ s_j^{\,3} & \text{otherwise} \end{cases}; \quad \begin{aligned} s_1 &\triangleq \dot{x}^2 + \dot{y}^2 - V_{\max}^2 \\ s_2 &\triangleq \dot{x}^2 - V_x^2 \\ s_3 &\triangleq \dot{y}^2 - V_y^2 \end{aligned} \tag{21}$$

Note that the choice of a single-sided cubic function for h_j helps ensure a stiff penalty for violating the velocity constraints.

Defining the co-state vector $\boldsymbol{\lambda}=\{\lambda_1, \lambda_2 ,..., \lambda_7\}^T$, the Hamiltonian is derived from Eq.(15) as

$$H = 1 + \lambda_1 z_2 + \lambda_2 u_1 + \lambda_3 z_4 + \lambda_4 u_2 + \lambda_5 h_1 + \lambda_6 h_2 + \lambda_7 h_3 \quad (22)$$

based on the objective function J defined in Eq.(10) for which K=0 and L=1. By invoking Eq.(17), the dynamics of the co-states (also known as the adjoint system dynamics) is given by

$$\begin{aligned}&\dot{\boldsymbol{\lambda}}= \left\{0 \quad -\left(\lambda_1 + 6\,\dot{x}\left(F_+^2(s_1)\lambda_5 + F_+^2(s_2)\lambda_6\right)\right) \quad 0 \quad -\left(\lambda_3 + 6\,\dot{y}\left(F_+^2(s_1)\lambda_5 + F_+^2(s_3)\lambda_7\right)\right) \quad 0 \quad 0 \quad 0\right\}^T \\ &F_+\left(s_j\right) \triangleq \begin{cases} 0 & s \ < \ 0 \\ s_j & \text{otherwise} \end{cases}\end{aligned} \quad (23)$$

3.2.2 Optimal control protocol

The optimal control inputs must be determined based on Eq.(18), which essentially states that the optimal control input must be chosen such that it minimizes the Hamiltonian. Judging from Eq.(22), the optimal control input is given by

$$\begin{aligned} u_1 &= -A_x \mathrm{sgn}(\lambda_2) \qquad \text{for } \lambda_2 \neq 0 \\ u_2 &= -A_y \mathrm{sgn}(\lambda_4) \qquad \text{for } \lambda_4 \neq 0 \end{aligned} \quad (24)$$

As expected for a minimum-time problem, u_1 and u_2 are bang-bang [11, 17]. However, there are two main problems with u_1 and u_2 as given by Eq.(24). First, they are not defined for any finite durations where λ_2=0 or λ_4=0. Over such durations, known as singular intervals, the optimal control cannot be determined directly from Eq.(22); it must be determined by investigating other properties of the dynamical system [17]. In the minimum-time cornering problem, singular intervals can occur only if the velocity constraints of Eqs.(7) and (8) become active. We observe that the velocity constraints are typically active in portions of the trajectory where the curvature is low thus allowing the trajectory to reach higher speeds without violating the acceleration limits. One can assume that low-curvature portions of the trajectory can be approximated as straight lines for which bang-coast-bang acceleration (i.e., trapezoidal velocity) profiles deliver minimum-time motion when velocity limits are activated [19]. It therefore makes sense to turn off the control input (i.e., make u_1=0 and/or u_2=0) during singular intervals so that the trajectory can coast. The second problem with Eq.(24) is that the control input is discontinuous, which can lead to numerical issues in the solution of the problem. It is therefore of practical benefit to replace it with a continuous curve having smoother transitions. Consequently, the control law is modified to

$$\begin{aligned} u_1 &= -A_x \mathrm{sgn}(\lambda_2) \cdot [a \tanh(k(|\lambda_2| - b)) + c] \\ u_2 &= -A_y \mathrm{sgn}(\lambda_4) \cdot [a \tanh(k(|\lambda_4| - b)) + c] \end{aligned} \quad (25)$$

where a and c must satisfy the conditions

$$\left.\begin{aligned} a \ + \ c \ = \ 1 \\ a\tanh(kb) - c = 0 \end{aligned}\right\} \Rightarrow a = \frac{1}{1+\tanh(kb)}; \quad c = \frac{\tanh(kb)}{1+\tanh(kb)} \quad (26)$$

To provide some intuition for the modified control input defined in Eq.(25), the relationship between u_1/A_x and λ_2 is plotted in Fig. 2. (Note that the plot is the same for u_2/A_y and λ_4). As can be seen, the control input is a smooth function which transitions gradually between the negative and positive acceleration limits. Additionally, around λ_2= 0, it has a relatively flat portion whose width is modulated by the variable b. The flat portion of the curve acts as a (small) boundary layer to ensure that $u_1 \approx 0$ can be maintained when $\lambda_2 \approx 0$ in order to allow coasting during singular intervals. The abruptness of the transition between the two control extremes is governed by the variable k; increasing k makes the transition more abrupt and vice versa. We recommend selecting k>10 so that the transition is not too sluggish, and b such that $kb \approx 2$ to ensure that a sufficiently large boundary layer is formed around λ_2=0. It must be noted that, even though the modified control input is more practical, it sacrifices optimality in the strictest sense, as is demonstrated in Sect. 5.

3.2.3 Specification of boundary conditions

With the system dynamics and optimal control protocol specified, the final step in defining the TPBVP is to specify its boundary conditions. In the simplified formulation presented in Sect. 2.2, the corner trajectory consists of two portions (i.e., $\mathbf{r}_1(t)$ for $t \in [t_0, t_1]$ and $\mathbf{r}_2(t)$ for $t \in [t_1, t_2]$). Therefore, there are actually two TPBVPs that must be solved simultaneously to obtain the optimal corner trajectory. For $\mathbf{r}_1(t)$, let t_{01} and t_{f1} correspond to t_0 and t_1, respectively. Consequently, its initial and final states, $\mathbf{z}(t_{01})$ and $\mathbf{z}(t_{f1})$, are given by

$$\begin{aligned} \mathbf{z}(t_{01}) &= \{l_1\cos\theta_1 \quad -V_0\cos\theta_1 \quad l_1\sin\theta_1 \quad -V_0\sin\theta_1 \quad 0 \quad 0 \quad 0\}^T \\ \mathbf{z}\left(t_{f1}\right) &= \left\{\varepsilon\cos\beta \quad z_2\left(t_{f1}\right) \quad \varepsilon\sin\beta \quad z_4\left(t_{f1}\right) \quad 0 \quad 0 \quad 0\right\}^T \end{aligned} \quad (27)$$

where $z_2(t_{f1})$ and $z_4(t_{f1})$ represent the unknown x and y velocities at $\mathbf{p}_1$. The boundary conditions for $\mathbf{r}_2(t)$ can be obtained in a similar manner by letting t_{02} and t_{f2} correspond to t_2 and t_1, respectively, so that the initial

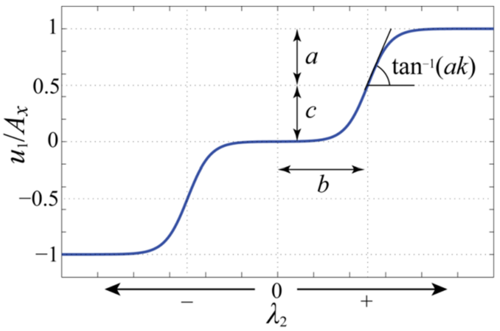


Fig. 2 Modified control input with boundary layer and smoothing

and final states, $\mathbf{z}(t_{02})$ and $\mathbf{z}(t_{f2})$, are given by

$$\begin{aligned}\mathbf{z}(t_{02}) &= \{\, l_2\cos\theta_2 \quad -V_2\cos\theta_2 \quad l_2\sin\theta_2 \quad -V_2\sin\theta_2 \quad 0 \quad 0 \quad 0\,\}^T \\ \mathbf{z}(t_{f2}) &= \{\, \varepsilon\cos\beta \quad -z_2(t_{f2}) \quad \varepsilon\sin\beta \quad -z_4(t_{f2}) \quad 0 \quad 0 \quad 0\,\}^T\end{aligned} \tag{28}$$

Note that the boundary conditions for $\mathbf{r}_2(t)$ are intentionally set up such that time evolves from t_2 to t_1 (opposite the direction of motion) so that the two TPBVPs coincide at time t_1. Also, notice that in Eqs. (27) and (28), the initial and final values of z_5, z_6, and z_7 are specified as zero. Since $\dot{z}_5(t)$, $\dot{z}_6(t)$, and $\dot{z}_7(t)$ are each≥0 ∀ t, forcing the boundary conditions of z_5, z_6, and z_7 to be zero ensures that $z_5(t)=z_6(t)=z_7(t)=0$ ∀ t, thus ensuring that the velocity constraints of Eqs. (7) and (8) are never violated [17]. Also, note from Eqs. (27) and (28) that V_0 and V_2 are assumed to be known. It is shown in the appendix that, providing l_i is not too short (i.e., providing $l_i \geq l_{i,\min}$), $V_0=V_1^*$ and $V_2=V_2^*$, where $V_i^*=\min\{|V_x/\cos\ \theta_i|,\ |V_y/\sin\ \theta_i|,\ V_{\max}\}$, i=1, 2. Selecting $l_i<l_{i,\min}$ is not reasonable for the proposed method because it severely reduces the wiggle room to optimize the corner trajectory, thus diminishing its advantage compared to methods with pre-specified paths. Therefore, for our approach, we stipulate that $l_i \geq l_{i,\min}$ so that $V_0=V_1^*$ and $V_2=V_2^*$ can be assumed. Fixing the initial and final velocities of the path also reduces the computational burden of the TPBVP solution by reducing the number of unknown states. Note that situations where the length of the line segment is less than $l_{i,\min}$ can be readily incorporated into the proposed method. However, such situations are not considered in this paper because for such cases, it is easier to use existing methods with pre-specified paths, given that the proposed method would not provide any significant cornering time advantages due to the lack of wiggle room for optimization.

Since the direction of the velocity at $\mathbf{p}_1$ is perpendicular to the bisector of the corner, $z_2(t_{f1})$ and $z_4(t_{f1})$ must satisfy the relationship

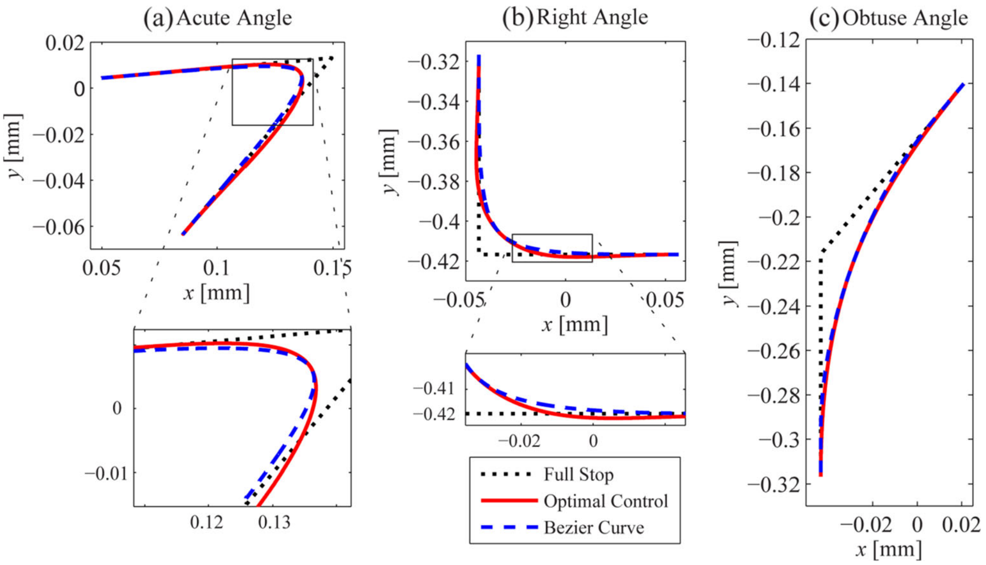
$$z_2(t_{f1})\cos\beta + z_4(t_{f1})\sin\beta = 0 \tag{29}$$

Equation (19) provides an additional condition that must be satisfied by the final value of the co-state vector. Since K=0 and, for the optimal system, H=0 [17], Eq. (19) reduces to

$$\boldsymbol{\lambda}^T\delta\mathbf{z}_f = 0 \Rightarrow \lambda_2(t_{f1})\sin\beta - \lambda_4(t_{f1})\cos\beta = 0 \tag{30}$$

The boundary conditions defined in Eqs. (27)–(30), combined with the state and co-state dynamics and the optimal control input (i.e., Eqs. (20), (23), and (25)), completely define the two TPBVPs for the minimum-time cornering problem. However, as previously noted, the two TPBVPs must be solved simultaneously. A problem that arises is that the durations, $|t_{f1}-t_{01}|$ and $|t_{f2}-t_{02}|$, of both TPBVPs are not necessarily the

Fig. 3 Corner paths of full stop, Bezier curve, and proposed optimal control methods for three corner angles



same. In other words, the TPBVPs could have different time scales, thus making it difficult to solve them simultaneously. To unify their time scales, we define parameter $\tau \in [0, 1]$ such that

$$t = \tau t_{fi} \quad (i = 1, 2) \tag{31}$$

where t_{0i}=0 has been arbitrarily assumed for convenience. Consequently, both TPBVPs can be re-parameterized as functions of τ via a straightforward variable change so that their time scales match. However, since t_{fi} are not known a priori, they are included as extra states in the TPBVPs by defining their dynamics as

$$\dot{t}_{fi} = 0 \tag{32}$$

To summarize, the optimal corner trajectory, $\mathbf{r}(t)$, is obtained using PMP by solving two TPBVPs, associated with $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$, simultaneously. Combined, the two TPBVPs have 30 dynamic equations (i.e., the seven state dynamics of Eq.(20) and seven co-state dynamics of Eq.(23) for each TPBVP, plus the two state equations for time as defined in Eq.(32)). They also have (combined) 30 boundary constraints, defined in Eqs.(27)–(30). The optimal solution is gotten by determining control inputs $u_1 = \ddot{x}$ and u_2=$\ddot{y}$ that minimize the Hamiltonian defined in Eq.(22). The optimal control input is the bang-bang control input of Eq.(24), which is approximated using the smoother profile given in Eq.(25) so as to obtain a solution during singular intervals and to improve the numerical solution process. In general, TPBVPs can be solved using so-called shooting algorithms [17], which can be readily programmed in commercial software with numerical solvers (e.g. MATLAB®). However, two practical issues must be addressed to enhance the convergence and practicality of the solution, as discussed in the next section.

4 Documentation of initial conditions and parameterization of optimal corner trajectory

4.1 Documentation of initial conditions

The key challenge with solving TPBVPs is that their boundary conditions are split. That is, they have to satisfy conditions at the beginning and end of a set of differential equations. Some TPBVPs, like the ones derived in the preceding section, have an incomplete set of initial conditions that are completed by end conditions. A shooting algorithm solves such TPBVPs by guessing the unknown initial conditions and iteratively solving the resulting initial value problem to obtain final solutions that match the known end conditions. Consequently, the convergence of shooting algorithms is highly sensitive to the reasonableness of the guessed initial conditions. To ensure a reasonable set of initial conditions, in this paper we propose using a direct method to generate low-resolution solutions for a characteristic set of configurations of the minimum-time cornering problem. The obtained solutions are then used to create a database of initial conditions for the characteristic set. Therefore, for any given cornering problem, the shooting algorithm can be provided with the closest initial guesses from the available solutions to facilitate its quick convergence to the desired high-resolution optimal curve.

To create the database, we propose normalizing the variables of the problem as follows

$$\bar{\mathbf{r}}(t) = \frac{1}{l_{\max}}\mathbf{r}(t); \quad \dot{\bar{\mathbf{r}}}(t) = \frac{1}{V_{\max}}\dot{\mathbf{r}}(t); \tag{33}$$

$$\ddot{\bar{\mathbf{r}}}(t) = \frac{l_{\max}}{V_{\max}^2}\ddot{\mathbf{r}}(t); \qquad \bar{t} = \frac{V_{\max}}{l_{\max}}t$$

where $l_{\max}$=max(l_1, l_2). Similarly, the parameters can be normalized as

$$\overline{V}_x = \frac{V_x}{V_{\max}}; \quad \overline{V}_y = \frac{V_y}{V_{\max}}; \quad \overline{A}_x = \frac{l_{\max}}{V_{\max}^2}A_x; \tag{34}$$

$$\overline{A}_y = \frac{l_{\max}}{V_{\max}^2}A_y; \quad \bar{\varepsilon} = \frac{\varepsilon}{l_{\max}}; \quad \bar{l}_i = \frac{l_i}{l_{\max}}$$

Note that θ_1 and θ_2 do not need to be normalized because they are only used inside trigonometric functions. The proposed normalization provides two important benefits. First, it eliminates $V_{\max}$ and the larger of l_1 and l_2, since their normalized values are unity. So we are left with eight parameters instead of ten. Secondly, the maximum values of all normalized variables and parameters are brought to the same order of magnitude (~1), which improves the performance of the numerical solvers.

The database for each TPBVP needs to provide a guess for the 14 initial co-state vectors (i.e., $\boldsymbol{\lambda}(t_{0i})$) and 2 final times t_{fi}. It can be in the form of a matrix of size $16N_s^8$, if each of the normalized parameters is discretized into N_s elements. We have observed that N_s=~20 provides a reasonable resolution for each parameter, as is demonstrated in illustrative examples presented in Sect. 5. With the database established, the initial conditions for a given minimum-time cornering problem can be estimated by interpolating between the documented initial conditions.

4.2 Parameterization of optimal trajectory using NURBS

The optimal free-form curve obtained by solving the TPBVPs needs to be accurately parameterized such that is usable in a CNC. The NURBS has become the de facto industry standard for parameterizing free-form curves in computer-aided design and manufacturing due to its intuitiveness and flexibility combined with its excellent mathematical and algorithmic properties [20, 21]; therefore, it is adopted in this paper for parameterization.

A NURBS curve of degree m, defined by n+1 control points, $\mathbf{b}_0$, $\mathbf{b}_1$,…, $\mathbf{b}_n$, with corresponding weights, w_0, w_1,…, w_n, and a knot vector, $\{g_0, g_1,\ldots, g_{m+n+1}\}^T$, is expressed as [20]

$$\mathbf{r}(\xi) = \frac{\sum_{k=0}^{n} N_{k,m}(\xi) w_k \mathbf{b}_k}{\sum_{k=0}^{n} N_{k,m}(\xi) w_k} = \sum_{k=0}^{n} R_{k,m}(\xi)\mathbf{b}_k \tag{35}$$

where $\xi \in [0, 1]$ is the spline parameter and $N_{k,m}(\xi)$ is the basis function of degree m given by

$$\begin{aligned} N_{k,m}(\xi) &= \frac{\xi - g_k}{g_{k+m} - g_k} N_{k,m-1}(\xi) \\ &\quad + \frac{g_{k+m+1} - \xi}{g_{k+m+1} - g_{k+1}} N_{k+1,m-1}(\xi) \end{aligned} \tag{36}$$

and

$$N_{k,0}(\xi) = \begin{cases} 1 & g_k \le \xi < g_{k+1} \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

In the specific application of NURBS considered in this paper, $\mathbf{r}(\xi)$ in Eq.(35) represents the optimal free-form corner trajectory calculated by solving the TPBVPs with parameter ξ representing time $t \in [t_0, t_2]$ rescaled such that t_0=0 and t_2=1. The traditional way of fitting a NURBS curve to $\mathbf{r}(\xi)$ is to select an initial set of weights w_0, w_1,…, w_n (e.g., uniform weights, for simplicity) and a uniform knot vector of the form [21]

$$g_j = \begin{cases} 0 & 0 \le j \le m \\ \dfrac{j-m}{n-m+1} & m+1 \le j \le n \\ 1 & n+1 \le j \le m+n+1 \end{cases} \tag{38}$$

Consequently, if $\xi \in [0, 1]$ is discretized into E+1 uniformly spaced points such that $\xi=\{\xi_0, \xi_1,\ldots, \xi_E\}$, Eq.(35) can be re-written in matrix form as

$$\underbrace{\begin{bmatrix} R_{0,m}(\xi_0) & R_{1,m}(\xi_0) & \cdots & R_{n,m}(\xi_0) \\ R_{0,m}(\xi_1) & R_{1,m}(\xi_1) & \cdots & R_{n,m}(\xi_1) \\ \vdots & \vdots & \ddots & \vdots \\ R_{0,m}(\xi_E) & R_{1,m}(\xi_E) & \cdots & R_{n,m}(\xi_E) \end{bmatrix}}_{\mathbf{R}} \cdot \underbrace{\begin{bmatrix} \mathbf{b}_0^T \\ \mathbf{b}_1^T \\ \vdots \\ \mathbf{b}_n^T \end{bmatrix}}_{\mathbf{B}} = \underbrace{\begin{bmatrix} \mathbf{r}(\xi_0)^T \\ \mathbf{r}(\xi_1)^T \\ \vdots \\ \mathbf{r}(\xi_E)^T \end{bmatrix}}_{\mathbf{\Gamma}} \tag{39}$$

from which the matrix $\mathbf{B}$ of control points can be solved for using the Least Squares equation

$$\mathbf{B} = \left(\left(\mathbf{R}^T\mathbf{R}\right)^{-1}\mathbf{R}^T\right)\cdot\mathbf{\Gamma} \tag{40}$$

A major problem with the traditional fitting process described by Eq.(40) is that even though it gives very good fits for $\mathbf{r}(t)$, it has been observed to give poorer fits for d$\mathbf{r}(t)$/dt and d$^2\mathbf{r}(t)$/dt^2 which are very important for smooth cornering motions. The quality of the fit for higher order derivatives of $\mathbf{r}(t)$ could be improved by adjusting the weights of the control points corresponding to sections with poor fits through an optimization process [22]. As an alternative, in this paper, we propose an enhanced NURBS fitting process that systematically incorporates higher order derivatives of $\mathbf{r}(t)$ thus reducing the need to optimize weights to improve fitting accuracy. Let us assume that the weights w_0, w_1,…, w_n are uniform, such that $R_{k,m}(\xi)$= $N_{k,m}(\xi)$ in Eq.(35). Accordingly, $\mathbf{r}'(\xi)$ is a NURBS curve with unity weights defined by n control points, degree $m-1$ and knot vector $[g_1,g_2,\ldots\ g_{m+n}]^T$, obtained by dropping the first and last knots from the knot vector of $\mathbf{r}(\xi)$ [21]; note that the accent   denotes derivative w.r.t. ξ. The control points of $\mathbf{r}$ (ξ), denoted by $\mathbf{b}_\mathrm{V}$, are related to the control points $\mathbf{b}$ of $\mathbf{r}(\xi)$ by [21]

$$\mathbf{b}_{\mathrm{V},k} = m\frac{\mathbf{b}_{k+1} - \mathbf{b}_k}{g_{k+m+1} - g_{k+1}}; \quad k = 0, 1\ldots n \tag{41}$$

Let us define $N_{\mathrm{d},k,m\text{-}1}(\xi)$ as the basis function defined over the reduced knot vector $[g_1,g_2,\ldots\ g_{m+n}]^T$. Accordingly, the time derivative d$\mathbf{r}(t)$/dt, due to its proportional relationship with $\mathbf{r}$ (ξ), can be written as

$$\begin{aligned} \dot{\mathbf{r}}\,(\xi) &= \frac{1}{t_{f1} + t_{f2}}\mathbf{r}'(\xi) \\ &= \frac{1}{t_{f1} + t_{f2}} \sum_{k=0}^{n-1} N_{\mathrm{d},k,m-1}(\xi) \frac{m(\mathbf{b}_{k+1} - \mathbf{b}_k)}{g_{k+m+1} - g_{k+1}} \end{aligned} \tag{42}$$

whose discretized form can be put in the compact format

$$\begin{aligned} \dot{\mathbf{\Gamma}} &= \left[\dot{\mathbf{r}}\,(\xi_0) \quad \ldots \quad \dot{\mathbf{r}}\,(\xi_E)\right]^T = \frac{\left[\mathbf{r}'\,(\xi_0)\ldots\mathbf{r}'(\xi_E)\right]^T}{t_{f1} + t_{f2}} \\ &= \frac{m}{t_{f1} + t_{f2}}\mathbf{N}_\mathrm{d}\mathbf{D}_1\mathbf{S}_n\mathbf{B} \triangleq \mathbf{N}_\mathrm{V}\mathbf{B} \end{aligned} \tag{43}$$

The matrix $\mathbf{N}_\mathrm{d}$ is obtained by discretizing $\{N_{\mathrm{d},k,m-1}(\xi)\}_{k=0,1,\ldots,n-1}$ over sampling indices $\xi=[\xi_0, \xi_1,\ldots, \xi_E]$, while $\mathbf{D}_i$ (i=1, 2) and $\mathbf{S}_n$ are defined as

$$\mathbf{S}_n = \begin{bmatrix} -1 & 1 & 0 & \ldots & 0 \\ 0 & -1 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}_{n\times(n+1)}, \tag{44}$$

$$\mathbf{D}_i = \begin{bmatrix} g_{m+1} - g_i & 0 & \cdots & 0 \\ 0 & g_{m+2} - g_{i+1} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & g_{m+n+1-i} - g_n \end{bmatrix}^{-1}$$

Table 1 Comparison of cornering times of the full stop, Bezier curve, and proposed optimal control methods for three corners

Corner	Cornering time [ms]		
	Full stop	Bezier curve	Optimal control
Acute angle (45°)	13.51	13.00	11.89
Right angle	14.25	10.05	9.56
Obtuse angle (140°)	13.52	7.72	7.81

Following the same procedure, $d^2\mathbf{r}(t)/dt^2$ can be written in the form

$$\ddot{\boldsymbol{\Gamma}} = \begin{bmatrix} \ddot{\mathbf{r}}(\xi_0) & \dots & \ddot{\mathbf{r}}(\xi_E) \end{bmatrix}^T = \frac{m(m-1)}{\left(t_{f1}+t_{f2}\right)^2}\mathbf{N}_{\mathrm{dd}}\mathbf{D}_2\mathbf{S}_{n-1}\mathbf{D}_1\mathbf{S}_n\mathbf{B} \triangleq \mathbf{N}_{\mathrm{A}}\mathbf{B} \tag{45}$$

where $\mathbf{N}_{\mathrm{dd}}$ is obtained by discretizing $\{N_{\mathrm{dd},k,m-2}(\xi)\}_{k=0,1,\dots,n-2}$, computed over the reduced knot vector $[g_2, g_3, \dots g_{m+n-1}]^T$ using $\xi = [\xi_0, \xi_1, \dots, \xi_E]$. The enhanced fitting can therefore be performed by solving for $\mathbf{B}$ as follows

$$\underbrace{\begin{bmatrix} \boldsymbol{\Gamma} \\ \rho_{\mathrm{V}}\dot{\boldsymbol{\Gamma}} \\ \rho_{\mathrm{A}}\ddot{\boldsymbol{\Gamma}} \end{bmatrix}}_{\overline{\boldsymbol{\Gamma}}} = \underbrace{\begin{bmatrix} \mathbf{N} \\ \rho_{\mathrm{V}}\mathbf{N}_{\mathrm{V}} \\ \rho_{\mathrm{A}}\mathbf{N}_{\mathrm{A}} \end{bmatrix}}_{\overline{\mathbf{N}}}\mathbf{B} \Rightarrow \mathbf{B} = \left(\overline{\mathbf{N}}^T\overline{\mathbf{N}}\right)^{-1}\overline{\mathbf{N}}^T\overline{\boldsymbol{\Gamma}} \tag{46}$$

where the matrix $\mathbf{N}$ is the same as $\mathbf{R}$ defined in Eq.(39), with $R_{k,m}(\xi)$ replaced by $N_{k,m}(\xi)$; ρ_{V} and ρ_{A} are weights attached to $d\mathbf{r}(t)/dt$ and $d^2\mathbf{r}(t)/dt^2$, respectively, to bring them to the same order of magnitude as $\mathbf{r}(t)$. They can be estimated as

$$\begin{aligned} \rho_{\mathrm{V}} &= \sigma \big/ \sigma_{\mathrm{V}} \\ \rho_{\mathrm{V}} &= \sigma \big/ \sigma_{\mathrm{A}} \end{aligned} \tag{47}$$

where σ, σ_{V}, and σ_{A} are the maximum singular values (i.e., worst-case magnitudes) [23] of the matrices $\mathbf{N}$, $\mathbf{N}_{\mathrm{V}}$, and $\mathbf{N}_{\mathrm{A}}$, respectively.

5 Illustrative examples

Simulation-based examples, similar to those employed by Ernesto and Farouki [5], are presented in this section to highlight the merits and limitations of the proposed optimal cornering and enhanced NURBS-based parameterization methods. Three examples consisting of an acute-angled (θ_1= −175°, θ_2=−130°), a right-angled (θ_1=90°, θ_2=0), and an obtuse-angled (θ_1=50°, θ_2=−90°) corner, all with $V_{\max}$= 25 mm/s, V_x=V_y=100 mm/s, A_x=A_y=4000 mm/s², l_1=l_2= 0.1 mm, and ε=0.015 mm are used to compare the following approaches:

1. *Full Stop*: Exact sharp corner with no smoothing curve.
2. *Bezier curve*: The cornering time is optimized using the method proposed by Ernesto and Farouki [5] based on a pre-specified rational quadratic Bezier curve (a type of B-spline) with optimal feed rate scheduling.
3. *Optimal control*: The cornering time is optimized using the PMP-based optimal control method proposed in this paper.

Note that for the Bezier curve method, the lengths l_1 and l_2 defined in this paper respectively correspond to lengths a_1+ b_1+l_1 and a_2+b_2+l_2 defined in [5]. Therefore, there is a slight difference in notation. To avoid confusion, let us rename the l_i in Ernesto and Farouki's paper [5] as f_i here so that l_i=a_i+b_i+f_i (where i=1, 2). With this clarification made, f_1=f_2=0.08 mm is used to determine the shape (i.e., weight w_2) of the Bezier curve specified by Ernesto and Farouki that satisfies the tolerance constraint ε. The parameters a_i and b_i are then determined (following the procedure outlined in [5]) such that a_i+ b_i=0.02 mm. It is also important to note that, in their formulation, Ernesto and Farouki [5] did not consider the velocity constraints of Eqs. (7) and (8). Therefore, to make the comparison consistent, the equation $E_k \leq ½ V^2_{\max}$ has been added to their method to enforce the path velocity constraint of Eq. (7). Note that the axis velocity constraints of Eq.(8) are not taken into account in the examples presented in this section because V_x and V_y are greater than $V_{\max}$, as is typically the case for properly planned manufacturing processes.

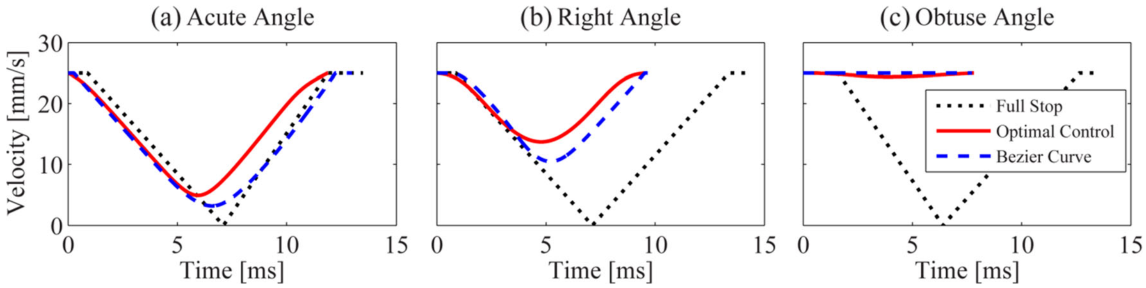


Fig. 4 Speed profiles of full stop, Bezier curve, and proposed optimal control methods for three corner angles

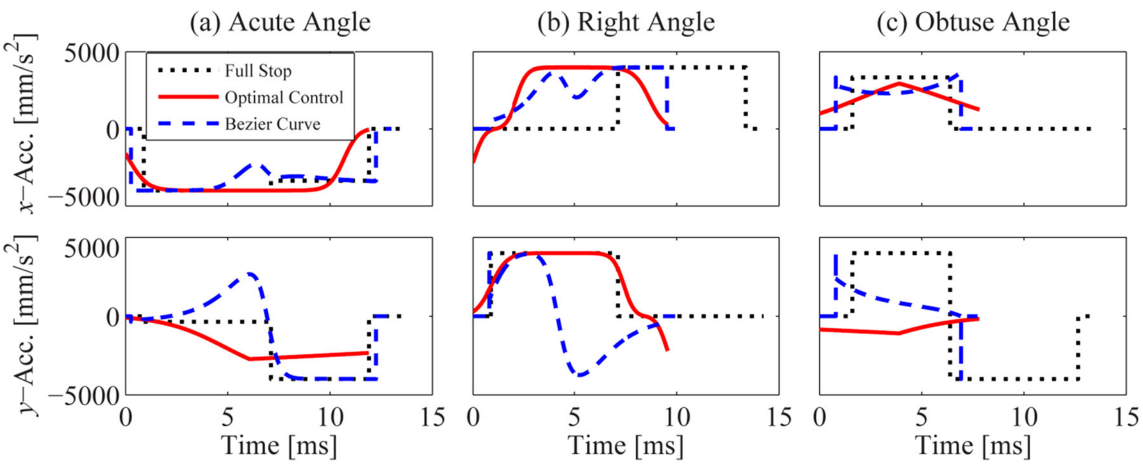


Fig. 5 Axes acceleration profiles of full stop, Bezier curve, and proposed optimal control methods for three corner angles

In the proposed approach, the initial condition documentation is performed using GPOPS-II, a MATLAB®-based optimal control solver (direct method), and the resolution of θ_1 and θ_2 in the database is 5°. The TPBVPs are solved using a shooting algorithm programmed in MATLAB® using its *fsolve* nonlinear equation and *ode45* ordinary differential equation solvers. k=16 and b=0.125 are used for the control protocol defined in Eq.(25).

Table 1 compares the cornering times, while Figs. 3, 4, and 5 compare the corner paths, path velocities, and axis accelerations of the full stop, Bezier curve, and optimal control methods applied to the three corners investigated in this paper. As can be seen from the figures, all three approaches satisfy the specified tolerance and kinematic constraints. Both the optimal control and Bezier curve-based methods perform significantly better than does the full stop, especially as the included angle of the corner increases. Table 1 indicates that, for the acute angle, the cornering times for the Bezier curve and optimal control methods are 4 and 12 % shorter, respectively, compared to the full stop case, while for the right angle, their cornering times are shorter by 29 and 33 %, respectively. From Figs. 3 and 4, one observes that the speed of the Bezier curve method is impeded by its relatively tight curvature, especially for the acute and right-angled corners. The optimal control approach is able to achieve relatively higher speeds by smartly modifying its shape (curvature) autonomously. Notice that the path the optimal curve travels is not symmetrical about the bisector of the acute-angled corner. This is because the acute-angled corner is not symmetrically positioned with respect to the x or y axis. Consequently, the optimal control method adjusts its shape to best utilize the directions of maximum acceleration. It is interesting to note that the path of the proposed method is similar to the time-optimal path followed by race car pilots when negotiating right-angled corners [2]. Even though the path is longer, its curvatures are wider, allowing for higher speeds.

For the obtuse angle, the optimal control method performs slightly (1 %) poorer than does the Bezier curve with regard to cornering time. This slight loss of performance is caused by the approximations made in the control law of Eq.(25) to accommodate singular intervals and reduce abrupt transitions. As explained in Sect. 3.2.2, wider angles are more likely to have singular intervals and thus be adversely influenced by the approximation. This fact is demonstrated in Fig. 6 which plots the cornering times of the three methods as functions of the included angle of the corner; it also plots the boundary beyond which no feasible solution can be found. The boundary is calculated using the GPOPS-II solver described earlier in this section. One can see that the effect of the approximations made in Sect. 3.2.2 is very small at smaller included angles but gradually increases as the included angle widens. Notice that the Bezier curve and optimal control methods almost

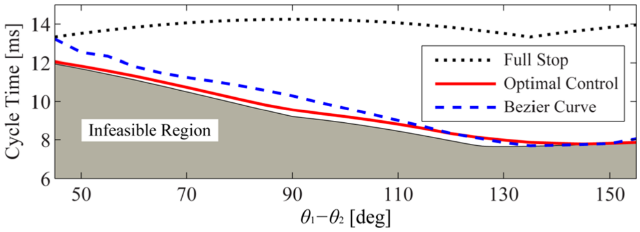


Fig. 6 Cycle time comparison for various corners as a function of the included angle

Table 2 Minimum and maximum values of l_i for three corners

Corner	Minimum and maximum values of l_i [mm]			
	$l_{1,\min}$	$l_{1,\max}$	$l_{2,\min}$	$l_{2,\max}$
Acute angle (45°)	0.0956	0.1427	0.0787	0.1321
Right angle	0.0962	0.1826	0.0962	0.1826
Obtuse angle (140°)	0.0851	0.1459	0.0851	0.1615

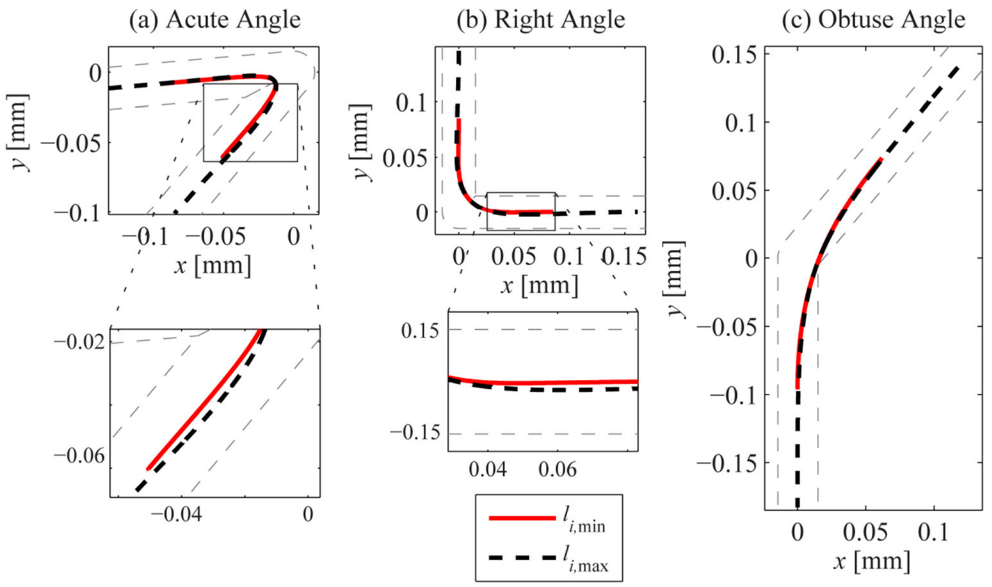


Fig. 7 Optimized Profile with $l_{i,\min}$ and $l_{i,\max}$

converge to the boundary of the feasible region at very large angles. This is because the curvatures are so low that the corner is almost entirely traversed at the maximum speed, $V_{\max}$, thus leaving very little room for optimization.

From a practical standpoint, it must be pointed out that a typical manufacturing process may have to traverse hundreds of thousands of sharp corners to produce one part. Therefore, even though the time savings of the Bezier and optimal control methods are in the order of a few milliseconds per corner, they can quickly add up to significant savings in throughput. Note also that the computation time of the TPBVPs for the acute, right, and obtuse angles are 1.4, 0.5, and 0.87 s, respectively, on a PC equipped with an Intel® Core™ i5-2400 processor at 3.10 GHz and 8 GB RAM. The implication is that the proposed method is much more practical for offline optimization of cornering time than is the dynamic programming approach investigated by the authors in [15], which required several minutes to converge to solutions of poorer resolution than the ones obtained using the proposed approach.

To validate the simplifying assumptions in estimating $l_{i,\min}$ and $l_{i,\max}$ in the Appendix, Table 2 shows the $l_{i,\min}$ and $l_{i,\max}$ values for the three corners. As can be seen, the selected value of $l_1=l_2=0.1$ mm falls within the minimum and maximum l_i values for all three corners. Figure 7 shows the corner profiles for the three corners, generated by using the $l_{i,\min}$ and $l_{i,\max}$ values reported in Table 2. As seen, the proposed estimation of $l_{i,\max}$ is conservative, ensuring that the assumption made in Sect. 2.2 is not violated as long as $l_i \leq l_{i,\max}$. Furthermore, $V_0=V_1^*$ and $V_2=V_2^*$ have been used for generating each corner trajectory, meaning that the estimation for $l_{i,\min}$ is reasonable.

The enhanced NURBS-based parameterization process proposed in this paper is validated using a NURBS curve with $n=30$ and $m=4$ to fit the optimal free-form curves obtained for the three corners. The weights $\rho_V=0.0188$ and $\rho_A=1.45\times10^{-4}$ are calculated according to Eq.(47). Three fitting schemes are compared:

1. *Traditional NURBS fitting with uniform weights*: NURBS fitting based on position (according to Eq.(40)) with uniform weights.
2. *Traditional NURBS fitting with optimized weights*: NURBS fitting based on position (according to Eq. (40)) with nonuniform weights determined by optimization (i.e., using a brute-force realization of the problem setup in [22]).
3. *Enhanced NURBS fitting*: Proposed fitting scheme (with uniform weights) which systematically

Table 3 Comparison of maximum fitting errors of three NURBS-based fitting methods

	Max. pos. err. [nm]			Max. vel. err. [μm/s]			Max. acc. err. [mm/s^2]		
	Acute	Right	Obtuse	Acute	Right	Obtuse	Acute	Right	Obtuse
Uniform weights	0.208	0.087	0.034	2.58	1.92	0.61	33.1	49.3	20.5
Optimized weights	0.193	0.087	0.034	8.94	1.92	0.61	50.5	49.3	20.5
Enhanced (proposed)	0.215	0.089	0.032	2.18	0.96	0.60	28.4	25.1	20.3

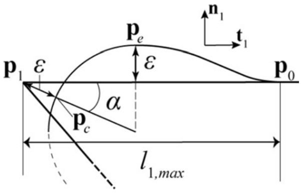


Fig. 8 Estimation of $l_{i,\max}$

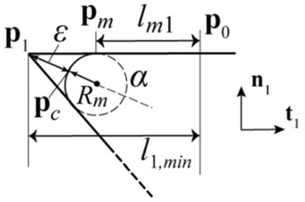


Fig. 10 Estimation of $l_{i,\min}$

incorporates position, velocity, and acceleration profiles as given by Eq.(46).

Table 3 reports the maximum position fitting errors for the three corners using the three fitting algorithms under investigation. The maximum position fitting errors of all three methods are within 5 % of each other and are negligible (in the sub-nanometer range). However, in terms of maximum velocity and acceleration fitting errors, the enhanced NURBS method outperforms the other two by up to 50 %.

6 Conclusion

This paper has presented a different approach for minimizing the time needed to execute sharp corners, subject to tolerance constraints and kinematic limits. Instead of pre-specifying a corner smoothing curve and optimizing the feed profile along the curve, an optimal control method is used to generate the best free-form curve that minimizes corner traversal time while adhering to path tolerance and kinematic constraints. An enhanced NURBS-fitting technique which systematically incorporates position, velocity, and acceleration profiles is proposed for accurately parameterizing the free-form curve resulting from the optimization to allow its execution in a CNC. Illustrative examples are used to demonstrate the ability of the proposed optimal control method to improve cornering times compared to unsmoothed corners and to a recently proposed technique that uses a pre-specified B-spline for high-speed cornering. The computational burden of the proposed method is also shown to be practical for offline optimization of cornering times in robotics or manufacturing applications. The enhanced NURBS-fitting technique is shown to be very elegant and significantly more accurate than are standard and optimized NURBS-fitting schemes in parameterizing time-based free-form curves like the one resulting from the proposed optimal control method. A nontrivial extension of this work could be the global smoothing and optimization of multiple short line segments subject to path error constraints, which is currently performed using pre-specified curves.

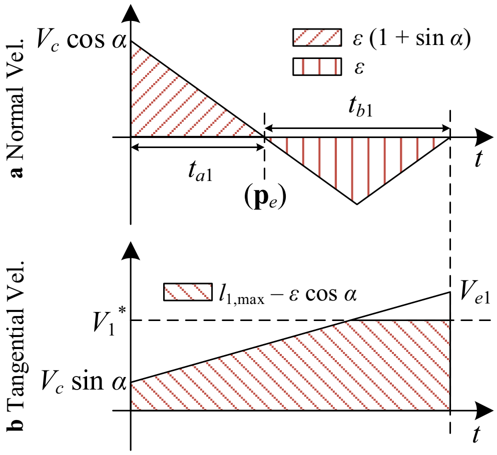


Fig. 9 **a** Normal and **b** tangential velocity profiles of half corner used for estimating $l_{i,\max}$

Acknowledgments This work was partially supported by the Rackham Centennial Fellowship awarded to Mr. Molong Duan by the University of Michigan.

Appendix: estimation of $l_{i,\max}$ and $l_{i,\min}$

The lengths l_i of the corner region have to satisfy the condition $l_{i,\min} \le l_i \le l_{i,\max}$ to ensure that

1. The simplifying assumption that the point defined in Eq.(12) is the critical point for corner tolerance violation holds.
2. The boundary condition $V_0 = V_1^*$ and $V_2 = V_2^*$ assumed in Sect. 3.2.3 is valid.

An estimation of $l_{i,\min}$ and $l_{i,\max}$ is presented below.

Estimation of $l_{i,\max}$

Let us assume, as shown in Fig. 8, that the corner path passing through the critical point $\mathbf{p}_c$ defined in Eq.(12) is on the verge of violating the tolerance constraint, ε, at a point $\mathbf{p}_e$ different from $\mathbf{p}_c$. Let V_c represent the speed at $\mathbf{p}_c$ and let angle $\alpha = |\theta_1 - \theta_2|/2$. Considering the l_1 leg of the corner, let $\mathbf{t}_1$ and $\mathbf{n}_1$ respectively represent unit vectors in the tangential and normal directions of the line from $\mathbf{p}_1$ to $\mathbf{p}_0$. Accordingly, the acceleration limits, A_{t1} and A_{n1}, in the tangential and normal directions are given by

$$A_{t1} = \min\left\{\frac{A_x}{|\cos\theta_1|}, \frac{A_y}{|\sin\theta_1|}\right\}; \tag{48}$$

$$A_{n1} = \min\left\{\frac{A_x}{|\sin\theta_1|}, \frac{A_y}{|\cos\theta_1|}\right\}$$

As shown in Fig. 9, in the normal direction, the corner trajectory must first decelerate from speed $V_c \cos\alpha$ at $\mathbf{p}_c$ to 0 at $\mathbf{p}_e$, covering distance $(1+\sin\alpha)\,\varepsilon$ and then travel distance ε from rest at $\mathbf{p}_e$ to rest at $\mathbf{p}_0$. Due to the relatively short distances traveled in the normal direction, it is assumed that the minimum-time motion in the normal direction is "bang-bang" (i.e., it accelerates and decelerates using A_{n1}, without hitting the velocity limit in the normal direction). Therefore, V_c and t_{a1}, the travel time between $\mathbf{p}_c$ and $\mathbf{p}_e$, can be determined as

$$(1+\sin\alpha)\varepsilon = \frac{(V_c\cos\alpha)^2}{2A_{n1}} = \frac{1}{2}A_{n1}{t_{a1}}^2 \tag{49}$$

$$\Rightarrow \begin{cases} V_c = \sqrt{2A_{n1}(1+\sin\alpha)\varepsilon}\Big/\cos\alpha \\ t_{a1} = \sqrt{2(1+\sin\alpha)\varepsilon\Big/A_{n1}} \end{cases}$$

Similarly, the travel time, t_{b1}, from $\mathbf{p}_e$ to $\mathbf{p}_0$ can be calculated as

$$A_{n1}\left(\frac{t_{b1}}{2}\right)^2 = \varepsilon \Rightarrow t_{b1} = 2\sqrt{\frac{\varepsilon}{A_{n1}}} \tag{50}$$

Figure 9 also shows the tangential velocity profile from $\mathbf{p}_c$ to $\mathbf{p}_0$. It is assumed to utilize the maximum tangential acceleration, A_{t1}, starting from velocity $V_c \sin\alpha$ at $\mathbf{p}_c$. If it travels the entire distance from $\mathbf{p}_c$ to $\mathbf{p}_0$ without hitting the velocity limit, the tangential velocity V_{e1} is given by

$$V_{e1} = V_c\sin\alpha + A_{t1}(t_{a1}+t_{b1}) \tag{51}$$

However, due to the longer distance traveled in the tangential direction, the motion could encounter the tangential velocity limit, which is approximated as ${V_1}^*$, based on the assumption that the corner path is mostly aligned with the $\mathbf{t}_1$ direction. Accordingly, the distance $l_{1,\max}$ is derived as

$$l_{1,\max} = \begin{cases} l_{1,0} & V_{e1} \le V_1^* \\ l_{1,0} - \dfrac{\left(V_1^* - V_{e1}\right)^2}{2A_{t1}} & V_{e1} > V_1^* \end{cases} \tag{52}$$

$$l_{1,0} \triangleq \varepsilon\cos\alpha + (t_{a1}+t_{b1})(V_{e1}+V_c\sin\alpha)\Big/2$$

A similar procedure can be used to calculate $l_{2,\max}$.

Estimation of $l_{i,\min}$

To determine $l_{i,\min}$, let us assume, as shown in Fig. 10, that the optimized corner path consists, in part, of a circular arc from the point $\mathbf{p}_c$ to a point $\mathbf{p}_m$. If l_1 is very short, the circular arc is forced to be tangent to the line joining $\mathbf{p}_0$ and $\mathbf{p}_1$. Consequently, the radius R_m of the circular arc is expressed as

$$R_m = \frac{\sin\alpha}{1-\sin\alpha}\varepsilon \tag{53}$$

If R_m is assumed to be equal to the smallest radius of curvature along the optimized path, the maximum allowable constant speed V_m along the arc is given by

$$V_m = \min\left(V^*, \sqrt{A_c R_m}\right); \tag{54}$$

$$V^* \triangleq \min\{V_x, V_y, V_{\max}\}; \quad A_c \triangleq \min\{A_x, A_y\}$$

Furthermore, if we assume that the path between $\mathbf{p}_0$ and $\mathbf{p}_m$ is traveled with uniform acceleration/deceleration A_{t1} along $\mathbf{t}_1$, the shortest distance l_{m1} in Fig. 10 that will permit the motion from velocity ${V_1}^*$ at $\mathbf{p}_0$ to V_{m} at $\mathbf{p}_m$ without violating the acceleration limits is given by

$$l_{m1} = \frac{V_1^{*2} - V_m^2}{2A_{t1}} \tag{55}$$

where A_{t1} is defined in Eq.(48). Therefore, $l_{1,\min}$ can be calculated as

$$l_{1,\min} = l_{m1} + R_m\cot\alpha \tag{56}$$

The same procedure can be followed to calculate $l_{2,\min}$.

References

1. Jouaneh MK, Wang Z, Dornfeld DA (1990) Trajectory planning for coordinated motion of a robot and a positioning table—I: path specification. IEEE Trans Robot Autom 6:735–45. doi:10.1109/70.63274
2. Barre P, Dieulot J, Bearee R, Bouzidi A (2005) A heuristic path-planning method for enhancing machine-tool contour-following. J Syst Sci Syst Eng 14:85–96
3. Erkorkmaz K, Yeung C-H, Altintas Y (2006) Virtual CNC system. Part II. High speed contouring application. Int J Mach Tools Manuf 46:1124–38. doi:10.1016/j.ijmachtools.2005.08.001
4. Imani B, Jahanpour J (2008) High-speed contouring enhanced with PH curves. Int J Adv Manuf Technol 37:747–59. doi:10.1007/s00170-007-1006-z
5. Ernesto C, Farouki R (2012) High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error. Int J Adv Manuf Technol 58:327–38. doi:10.1007/s00170-011-3394-3
6. Beudaert X, Lavernhe S, Tournier C (2013) 5-axis local corner rounding of linear tool path discontinuities. Int J Mach Tools Manuf 73:9–16. doi:10.1016/j.ijmachtools.2013.05.008

7. Ye P, Shi C, Yang K, Lv Q (2008) Interpolation of continuous micro line segment trajectories based on look-ahead algorithm in high-speed machining. Int J Adv Manuf Technol 37:881–97. doi:10.1007/s00170-007-1041-9
8. Zhang LB, You YP, He J, Yang XF (2011) The transition algorithm based on parametric spline curve for high-speed machining of continuous short line segments. Int J Adv Manuf Technol 52:245–54. doi:10.1007/s00170-010-2718-z
9. Zhao H, Zhu L, Ding H (2013) A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. Int J Mach Tools Manuf 65:88–98. doi:10.1016/j.ijmachtools.2012.10.005
10. Sencer B, Ishizaki K, Shamoto E (2014) A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. Int J Adv Manuf Technol 76:1977–92. doi:10.1007/s00170-014-6386-2
11. Bobrow J, Dubowsky S, Gibson J (1985) Time-optimal control of robotic manipulators along specified paths. Int. J. Rob. Res
12. Constantinescu D, Croft E (2000) Smooth and time-optimal trajectory planning for industrial manipulators along specified paths. J Robot Syst 17:233–49. doi:10.1002/(SICI)1097-4563(200005)17:5<233::AID-ROB1>3.0.CO;2-Y
13. Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. CIRP Ann Technol 1–6
14. Sencer B, Altintas Y, Croft E (2008) Feed optimization for five-axis CNC machine tools with drive constraints. Int J Mach Tools Manuf 48:733–45. doi:10.1016/j.ijmachtools.2008.01.002
15. Duan M, Okwudire C (2014) Minimum-Time Corning for Manufacturing Machines Using Optimal Control. ASME Dyn. Syst. Control Conf.
16. Bosetti P, Bertolazzi E (2014) Feed-rate and trajectory optimization for CNC machine tools. Robot Comput Integr Manuf 30:667–77. doi:10.1016/j.rcim.2014.03.009
17. Kirk D (2012) Optimal control theory: an introduction
18. Pytlak R (1999) Numerical methods for optimal control problems with state constraints. Springer
19. Jamhour E, André P (1996) Planning smooth trajectories along parametric paths. Math. Comput. Simul. 55
20. Piegl L (1991) On NURBS: a survey. IEEE Comput Graph Appl 11:55–71
21. Piegl L, Tiller W (1997) The NURBS Book. Berlin
22. Ulker E (2012) NURBS curve fitting using artificial immune system. Int J Innov Comput Inf Control 8:2875–87
23. Golub G, Reinsch C (1970) Singular value decomposition and least squares solutions. Numer Math 14:403–20. doi:10.1007/BF02163027