Virtual CNC system. Part II. High speed contouring application

Article *in* International Journal of Machine Tools and Manufacture · August 2006

DOI: 10.1016/j.ijmachtools.2005.08.001

3 authors, including:

Kaan Erkorkmaz
University of Waterloo
49 PUBLICATIONS **1,268** CITATIONS

SEE PROFILE

Yusuf Altintas
University of British Columbia - Vancouver
231 PUBLICATIONS **12,917** CITATIONS

SEE PROFILE

Available online at www.sciencedirect.com
SCIENCE DIRECT
ELSEVIER
International Journal of Machine Tools & Manufacture 46 (2006) 1124–1138
INTERNATIONAL JOURNAL OF MACHINE TOOLS & MANUFACTURE DESIGN, RESEARCH AND APPLICATION
www.elsevier.com/locate/ijmactool

Virtual CNC system. Part II. High speed contouring application

Kaan Erkorkmaz[a,*], Chi-Ho Yeung[b,1], Yusuf Altintas[b,1]

[a]*Precision Controls Laboratory, Department of Mechanical Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ont., Canada N2L 3G1*
[b]*Manufacturing Automation Laboratory, Department of Mechanical Engineering, University of British Columbia, 2324 Main Mall, Vancouver, BC, Canada V6T 1Z4*

Received 8 July 2005; accepted 4 August 2005
Available online 3 October 2005

Abstract

This paper presents a trajectory planning strategy for maintaining the tool positioning accuracy in high speed cornering applications. A 3D contour error estimation algorithm is presented for determining the geometric deviation from arbitrarily shaped toolpaths. Two spline fitting strategies are developed for smoothening sharp corners. The under-corner approach reduces the toolpath length, and therefore the cornering time. This technique yields successful results when used with a high bandwidth servo controller (such as sliding mode control), capable of accurately tracking the commanded toolpath. The over-corner approach is based on stretching out the sharp corner with a smooth curve, which counteracts the 'undercut' caused by the large phase lag in low bandwidth servo controllers (such as P–PI control). The cornering feedrate is adjusted in the Virtual CNC platform, developed in the first part of this article, to ensure that contour error violation does not occur. The achieved cornering accuracy is verified in experiments, which are in close agreement with predictions obtained with the Virtual CNC.

Keywords: Virtual machining; CNC; Contour error; Spline interpolation

1. Introduction

Maintaining the contouring accuracy at high feedrates is vital to preserving the tolerance integrity of parts produced with high speed machining. Contour errors originate from servo tracking errors in the individual axes (Fig. 1), which become prevalent when trying to track reference trajectories with high frequency content and high torque demands beyond the actuators' limits. Examples of this are machining circular arcs with very small radius at high feedrates or turning around sharp corners, where large transients in position, velocity, and acceleration commands occur. The look-ahead functionality in CNC systems has been developed to alleviate this problem to a certain extent, by ensuring that acceleration commands in the interpolated trajectory never exceed the drives' dynamic capability [1], following similar research conducted in time-optimal trajectory generation for robots [2] and machine tools [3]. However, limiting the magnitude of acceleration or torque is not sufficient for guaranteeing the tracking, and therefore the contouring accuracy. To address this problem, Altintas and Erkorkmaz have proposed feedrate optimization that also considers jerk limits in the individual axes [4], helping indirectly maintain the tracking error within a certain tolerance while reducing the cycle time. Still, specifying torque and jerk limits are not sufficient on their own for guaranteeing the contouring accuracy of a drive system, as additional factors such as friction, backlash, and structural resonances play an important role in determining the overall dynamic performance [5]. In this paper, a contouring strategy is developed which takes advantage of the Virtual CNC's ability to accurately predict contour errors in the presence of the above mentioned factors, and applies corrective measures by modifying the toolpath and adjusting the feedrate, so that sharp corners can be traveled

* Corresponding author. Tel.: +1 519 885 1211 5214; fax: +1 519 885 5862.
E-mail addresses: kaane@mecheng1.uwaterloo.ca (K. Erkorkmaz), cyeung@mech.ubc.ca (C.-H. Yeung), altintas@mech.ubc.ca (Y. Altintas).
URL: http://www.pcl.uwaterloo.ca.
[1] Tel.: +1 604 822 8182; fax: +1 604 822 2403, http://www.mech.ubc.ca/~mal.

0890-6955/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmachtools.2005.08.001

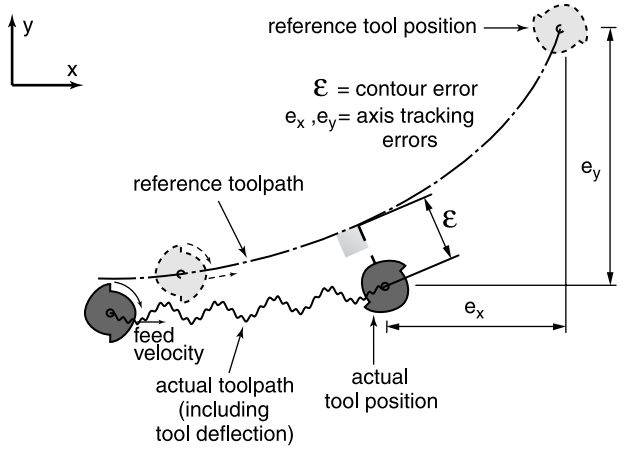

Fig. 1. Contour error originating from servo tracking errors.

in minimum time without violating the specified toolpath tolerance.

Accurate estimation of the contour error plays an important role in taking necessary corrective action either in the control law [6], or in feedrate planning [7], in order to improve the accuracy of high speed contouring operations. Koren [6,8] indicated that the contour error for tracking linear and circular toolpaths could be formulated as a function of the axis tracking errors, and based his cross-coupling controller, which works to directly reduce the contouring error, on this principle. For the contour error estimation technique to work with curved toolpaths, the analytical expression of the toolpath has to be known in advance. Later, Erkorkmaz and Altintas [9] developed a real-time contour error estimation method for arbitrarily shaped toolpaths, which does not require analytical knowledge of the toolpath. This technique is based on reconstructing the reference contour from commanded positions, and evaluating the shortest distance between the desired contour and the actual tool position. This method has been adopted in this article, for evaluating the contouring accuracy achieved with the proposed trajectory modification scheme.

One possible solution for tracking cornered toolpaths is to replace the sharp corner with a smooth curve, as has been applied in robotics literature. Jouaneh et al. [10] indicated that sharp cornered paths usually take more time to track, since the manipulator has to come to a full stop at the corner. They solved this problem by replacing the corner with a circular arc. Later, they improved the methodology by using a double clothoid curve instead of a circular arc, to further smoothen the motion transition [11,12]. Kanayama and Miyake [13] also used a double clothoid to smoothen the motion transition at the corners in trajectory planning for mobile robots. Another innovative idea for accurately turning sharp corners was proposed by Weck and Ye [14], which consisted of low pass filtering the motion commands before applying them to a high bandwidth servo controller with feedforward action. This has a similar effect to geometrically smoothening out the sharp corners. One disadvantage of this strategy is its close dependence on the drive dynamics, and the necessity of re-adjusting the low pass filter for different cornering feedrates.

In this paper, the smooth transition at sharp corners is achieved with a quintic spline instead of a clothoid curve. Spline trajectory generation techniques have been widely studied for machining complex sculptured surfaces such as aerospace parts, dies and molds. The nearly arc length parameterization scheme developed by Wang and Yang [15] has been used, so that the resulting curve closely follows the specified geometric tangent and normal (i.e. curvature) boundary conditions. The recursive interpolation scheme developed by Erkorkmaz and Altintas [16] has been used for smoothly interpolating the toolpath without feedrate fluctuation.

Henceforth, the paper is organized as follows. The contour error estimation technique implemented in the Virtual CNC is presented in Section 2. This is followed by the development of toolpath modification and feedrate adjustment schemes, presented in Section 3. Simulation and experimental results, verifying the effectiveness of the proposed contouring strategy are presented in Section 4. The paper is concluded in Section 5.

2. Contour error estimation

In order to evaluate the geometric deviation between the desired and actual toolpaths, an algorithm for accurately estimating the contour error in 3D Cartesian space is presented. This method does not pose any restriction on the type of toolpath geometry (such as linear, circular, or spline) and uses the basic idea of finding the shortest orthogonal distance between the reference toolpath and the actual tool position [9]. The nearest reference segment is used in estimating the contour error. Considering Fig. 2, where P is the actual tool position, R_i is the nearest reference point, the contour error ε is estimated by considering the three possible cases based on the region the actual tool position falls into:

Case 1: Contour error is estimated using the previous closest reference path segment $\overline{R_{i-1}R_i}$

Case 2: Contour error is estimated using the next closest reference path segment $\overline{R_iR_{i+1}}$

Case 3: Contour error is estimated using the closest reference point R_i

In regions 1 and 2, the contour error is obtained by computing the orthogonal distance between the actual tool position P, and the closest reference command segment ($\overline{R_{i-1}R_i}$ or $\overline{R_iR_{i+1}}$). In region 3, where the contour error is not orthogonal to either $\overline{R_{i-1}R_i}$ or $\overline{R_iR_{i+1}}$, the vector $\overrightarrow{R_iP}$ is taken as the contour error.

The three regions are bounded by the normal surfaces of the path segment vectors, and the bi-normal surface of

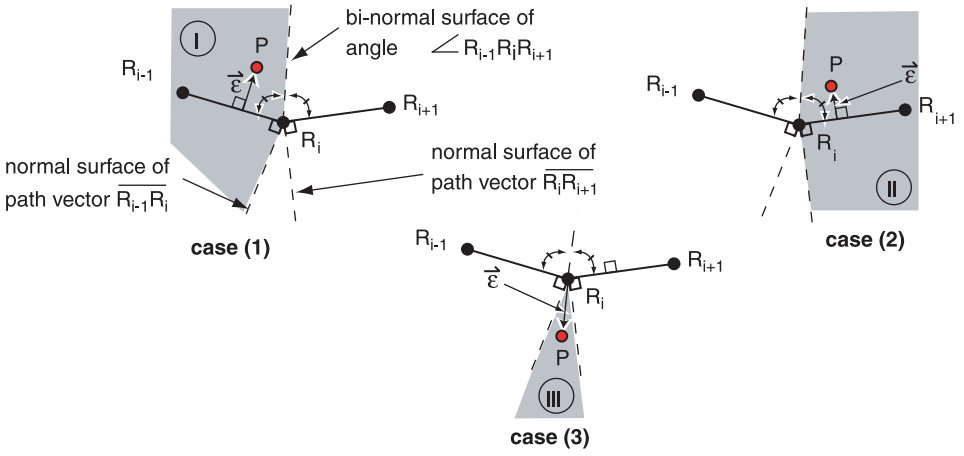

Fig. 2. Different cases for contour error estimation.

the angle $\angle R_{i-1}R_iR_{i+1}$. The reference points $R_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$, $R_i(x_i, y_i, z_i)$, and $R_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$ are generated by the position command interpolator, hence are available for contour error estimation. The normalized path segment vectors are defined as,

$$\left.\begin{aligned}\hat{\mathbf{r}}_i &= \frac{\overrightarrow{R_{i-1}R_i}}{||\overrightarrow{R_{i-1}R_i}||} = \frac{(x_i - x_{i-1})\cdot i + (y_i - y_{i-1})\cdot j + (z_i - z_{i-1})\cdot \mathbf{k}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2 + (z_i - z_{i-1})^2}} = r_{x,i}\cdot\mathbf{i} + r_{y,i}\cdot\mathbf{j} + r_{z,i}\cdot\mathbf{k} \\ \hat{\mathbf{r}}_{i+1} &= \frac{\overrightarrow{R_iR_{i+1}}}{||\overrightarrow{R_iR_{i+1}}||} = \frac{(x_{i+1} - x_i)\cdot i + (y_{i+1} - y_i)\cdot j + (z_{i+1} - z_i)\cdot \mathbf{k}}{\sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2}} = r_{x,i+1}\cdot\mathbf{i} + r_{y,i+1}\cdot\mathbf{j} + r_{z,i+1}\cdot\mathbf{k}\end{aligned}\right\} \quad (1)$$

The position vector $\mathbf{p}$ from the closest reference point $R_i(x_i, y_i, z_i)$ to the actual tool position $P(x_p, y_p, z_p)$ is,

$$\begin{aligned}\mathbf{p} &= \overrightarrow{R_iP} = (x_p - x_i)\cdot\mathbf{i} + (y_p - y_i)\cdot\mathbf{j} + (z_p - z_i)\cdot\mathbf{k} \\ &= p_x\cdot\mathbf{i} + p_y\cdot\mathbf{j} + p_z\cdot\mathbf{k}\end{aligned} \quad (2)$$

The normal surfaces N_i and N_{i+1}, perpendicular to the path segment vectors $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{r}}_{i+1}$ respectively, are defined as,

$$\left.\begin{aligned}N_i(x, y, z) &= r_{x,i}\cdot(x - x_i) + r_{y,i}\cdot(y - y_i) \\ &\quad + r_{z,i}\cdot(z - z_i) = 0 \\ N_{i+1}(x, y, z) &= r_{x,i+1}\cdot(x - x_i) + r_{y,i+1}\cdot(y - y_i) \\ &\quad + r_{z,i+1}\cdot(z - z_i) = 0\end{aligned}\right\} \quad (3)$$

The bi-normal surface B_i of the angle $\angle R_{i-1}R_iR_{i+1}$ is defined as follows,

$$\begin{aligned}B_i(x, y, z) &= (r_{x,i} + r_{x,i+1})\cdot(x - x_i) + (r_{y,i} + r_{y,i+1}) \\ &\quad \times (y - y_i) + (r_{z,i} + r_{z,i+1})\cdot(z - z_i) = 0\end{aligned} \quad (4)$$

Hence, the bounds of the three regions are identified in Eqs. (3) and (4). The actual tool position P is classified into the given regions as follows:

$$\left.\begin{aligned}&\text{if } \angle R_{i-1}R_iR_{i+1} \neq 180^\circ \text{ (i.e. } \hat{\mathbf{r}}_i + \hat{\mathbf{r}}_{i+1} \neq 0) \text{ then :} \\ &\left\{\begin{aligned}&\text{Region I} \Leftarrow \text{if } B_i(x_p, y_p, z_p) < 0 \,\&\, N_i(x_p, y_p, z_p) \leq 0 \\ &\text{Region II} \Leftarrow \text{if } B_i(x_p, y_p, z_p) \geq 0 \,\&\, N_{i+1}(x_p, y_p, z_p) \geq 0 \\ &\text{Region III} \Leftarrow \text{if } N_i(x_p, y_p, z_p) > 0 \,\&\, N_{i+1}(x_p, y_p, z_p) < 0\end{aligned}\right. \\ &\text{otherwise :} \\ &\left\{\begin{aligned}&\text{Region III} \Leftarrow \text{if } N_i(x_p, y_p, z_p) > 0 \,\&\, N_{i+1}(x_p, y_p, z_p) < 0 \\ &\text{Region II} \Leftarrow \text{otherwise}\end{aligned}\right.\end{aligned}\right\} \quad (5)$$

2.1. Regions I and II

If the actual tool position falls into Region I or II, the contour error is estimated as follows

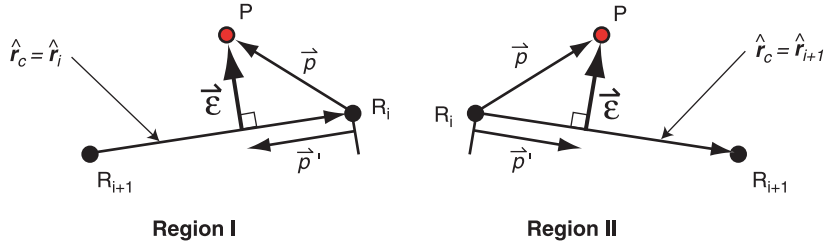

Fig. 3. Contour error estimation in Regions I and II.

Considering Fig. 3, the projection $\mathbf{p}'$ of the position vector $\mathbf{p}$ onto the closest normalized path segment $\hat{\mathbf{r}}_c$ is obtained as,

$$\text{Projection} \Rightarrow \mathbf{p}'(\hat{\mathbf{r}}_c \cdot \mathbf{p})\hat{\mathbf{r}}_c = (r_{xc}p_x + r_{yc}p_y + r_{zc}p_z)(r_{xc}\mathbf{i} + r_{yc}\mathbf{j} + r_{zc}\mathbf{k})$$
$$\text{where} \begin{cases} \hat{r}_c = \hat{r}_i & \Leftarrow \text{inRegionI} \\ \hat{\mathbf{r}}_c = \hat{r}_{i+1} & \Leftarrow \text{inRegionII} \end{cases} \tag{6}$$

Following the projection, the contour error vector $\vec{\varepsilon}$ is obtained by subtracting the projection $\mathbf{p}'$ from the position vector $\mathbf{p}$ as,

$$\varepsilon = \varepsilon_x \cdot \mathbf{i} + \varepsilon_y \cdot \mathbf{j} + \varepsilon_z \cdot \mathbf{k} = \mathbf{p} - \mathbf{p}' \tag{7}$$

and its magnitude is computed as,

$$\text{contour error} = ||\varepsilon|| = \sqrt{\varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2} \tag{8}$$

2.2. Region III

If the actual tool position P falls into Region III, the contour error is estimated as the vector $\vec{R_iP}$:

$$\left.\begin{aligned} &\text{For Region III} \Rightarrow \varepsilon = \varepsilon_x \cdot \mathbf{i} + \varepsilon_y \cdot \mathbf{j} + \varepsilon_z \cdot \mathbf{k} \cong \vec{R_iP} = \mathbf{p} = p_x \cdot \mathbf{i} + p_y \cdot \mathbf{j} + p_z \cdot \mathbf{k} \\ &\text{where} \Rightarrow \text{contouring error} = ||\varepsilon|| = \sqrt{p_x^2 + p_y^2 + p_z^2} \end{aligned}\right\} \tag{9}$$

3. Smoothening of sharp corners

The proposed cornering technique consists of smoothening out sharp transitions in the toolpath with an add-in spline (Fig. 4), which allows corners to be turned while maintaining a continuous feedrate profile. This feature is exploited for improving either the accuracy or the speed of cornering operations. The quintic spline has been chosen as the basis curve, due to its smoothness and near arc-length parameterization capability. General formulation for quintic toolpath generation is presented in Section 3.1. This is followed by parameterization methods developed for determining the add-in spline (Section 3.2), consisting of the under and over-corner approaches. Scheduling of feedrate in order to avoid contour error violation is explained in Section 3.3.

3.1. Quintic spline toolpath generation

In quintic spline toolpath generation, the objective is to connect a series of reference knots (i.e. control knots) with a series of fifth-order polynomial segments. These segments are constructed in such a way that continuity of the first and second derivatives are preserved at the knots, resulting in an overall C^2 composite curve. The spline toolpath generation process consists of parameterization and interpolation steps.

3.1.1. Parameterization

Considering Fig. 5, a quintic spline segment $\mathbf{S}_i$ connecting the two control knots $R_i(x_i,y_i,z_i)$ and $R_{i+1}(x_{i+1}, y_{i+1},z_{i+1})$ is expressed in Cartesian space by a fifth-order polynomial as follows,

$$\left.\begin{aligned} \boldsymbol{S}_i(u) =& \boldsymbol{A}_i u^5 + \boldsymbol{B}_i u^4 + \boldsymbol{C}_i u^3 + \boldsymbol{D}_i u^2 + \boldsymbol{E}_i u + \boldsymbol{F}_i \Leftarrow \\ &\text{where } u \in [0, l_i] \text{ and} \end{aligned}\right. \tag{10}$$

$$\left.\begin{aligned} \mathbf{S}_i(u) = \begin{bmatrix} S_{xi}(u) \\ S_{yi}(u) \\ S_{zi}(u) \end{bmatrix}, \quad \mathbf{A}_i = \begin{bmatrix} A_{xi} \\ A_{yi} \\ A_{zi} \end{bmatrix}, \quad \mathbf{B}_i = \begin{bmatrix} B_{xi} \\ B_{yi} \\ B_{zi} \end{bmatrix}, \quad \mathbf{C}_i = \begin{bmatrix} C_{xi} \\ C_{yi} \\ C_{zi} \end{bmatrix}, \quad \mathbf{D}_i = \begin{bmatrix} D_{xi} \\ D_{yi} \\ D_{zi} \end{bmatrix}, \quad \mathbf{E}_i = \begin{bmatrix} E_{xi} \\ E_{yi} \\ E_{zi} \end{bmatrix}, \quad \mathbf{F}_i = \begin{bmatrix} F_{xi} \\ F_{yi} \\ F_{zi} \end{bmatrix} \end{aligned}\right\} \tag{11}$$

S_{xi}, S_{yi}, and S_{zi} are the axis position polynomials for the ith spline segment. $\mathbf{A}_i$, $\mathbf{B}_i$,…,$\mathbf{F}_i$ are the coefficient vectors defining the spline toolpath in the x, y, and z coordinates. u

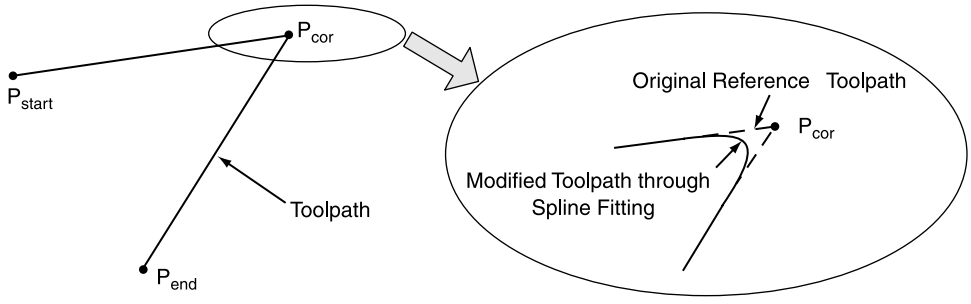

Fig. 4. Smoothening out sharp corners with spline fitting.

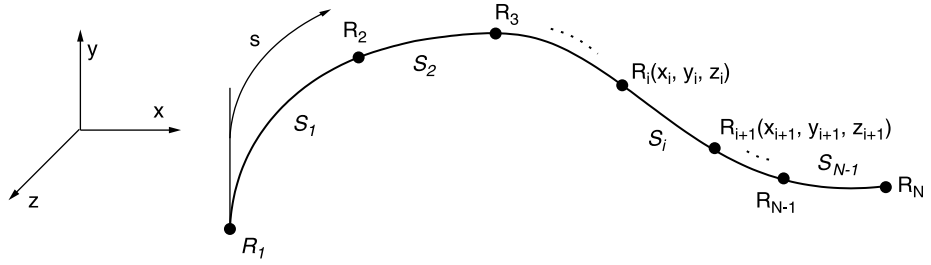

Fig. 5. Composite quintic spline defined by N control knots.

is the spline parameter and l_i is the spline parameter range, that inclusively bounds the maximum value u can have in the ith segment. Since there are six unknown coefficients in the quintic toolpath definition in Eq. (10), six boundary conditions are required to evaluate them. These boundary conditions are the vector coordinates ($\mathbf{p}_i$, $\mathbf{p}_{i+1}$) of the two consecutive knots, and the first and second derivatives, ($\mathbf{t}_i$, $\mathbf{t}_{i+1}$) and ($\mathbf{n}_i$, $\mathbf{n}_{i+1}$) in the mentioned order. Differentiating Eq. (10) with respect to the spline parameter u gives the first and second derivative profiles of the quintic spline as,

$$\left.\begin{aligned}\frac{d\mathbf{S}_i(u)}{du} &= 5\mathbf{A}_i u^4 + 4\mathbf{B}_i u^3 + 3\mathbf{C}_i u^2 + 2\mathbf{D}_i u + \mathbf{E}_i \\ \frac{d^2\mathbf{S}_i(u)}{du^2} &= 20\mathbf{A}_i u^3 + 12\mathbf{B}_i u^2 + 6\mathbf{C}_i u + 2\mathbf{D}_i\end{aligned}\right\} \tag{12}$$

By substituting the boundary conditions into Eqs. (10) and (12), the spline coefficients $\mathbf{A}_i$, $\mathbf{B}_i$,…,$\mathbf{F}_i$, are analytically solved as [17]

$$\left.\begin{aligned}\mathbf{A}_i &= \frac{1}{l_i^5}\left[6(\mathbf{p}_{i+1}-\mathbf{p}_i) - 3(\mathbf{t}_{i+1}+\mathbf{t}_i)\cdot l_i + \frac{1}{2}(\mathbf{n}_{i+1}-\mathbf{n}_i)\cdot l_i^2\right] \\ \mathbf{B}_i &= \frac{1}{l_i^4}\left[-15(\mathbf{p}_{i+1}-\mathbf{p}_i) + (7\mathbf{t}_{i+1}+8\mathbf{t}_i)\cdot l_i - \left(\mathbf{n}_{i+1}-\frac{3}{2}\mathbf{n}_i\right)\cdot l_i^2\right] \\ \mathbf{C}_i &= \frac{1}{l_i^3}\left[10(\mathbf{p}_{i+1}-\mathbf{p}_i) - (4\mathbf{t}_{i+1}+6\mathbf{t}_i)\cdot l_i + \left(\frac{1}{2}\mathbf{n}_{i+1}-\frac{3}{2}\mathbf{n}_i\right)\cdot l_i^2\right] \\ \mathbf{D}_i &= \frac{1}{2}\mathbf{n}_i \\ \mathbf{E}_i &= \mathbf{t}_i \\ \mathbf{F}_i &= \mathbf{p}_i\end{aligned}\right\} \tag{13}$$

For the first and second derivatives ($d\mathbf{S}_i/du$ and $d^2\mathbf{S}_i/du^2$) on the spline to correspond to the geometric tangent and normal vectors, the spline parameter u has to be parameterized as closely as possible to the real arc length s shown in Fig. 5. In order to achieve this, the nearly arc-length parameterization scheme devised by Wang and Yang [15] is adopted, which is obtained by setting the geometric arc increment ds equal to the parameter increment du at the mid-point of each segment:

$$\left|\frac{ds}{du}\left(u=\frac{l_i}{2}\right)\right| = 1, \quad \text{where}: \frac{ds}{du} = \sqrt{(5A_{xi}u^4 + \ldots + E_{xi})^2 + (5A_{yi}u^4 + \ldots + E_{yi})^2 + (5A_{zi}u^4 + \ldots + E_{zi})^2} \tag{14}$$

Substituting the toolpath coefficients from Eq. (13) and re-arranging Eq. (14) yields the below quartic expression for the spline parameter range l_i:

$$a_0 l_i^4 + a_1 l_i^3 + a_2 l_i^2 + a_3 l_i + a_4 = 0$$

where

$$\left.\begin{aligned}a_0 &= (\boldsymbol{n}_{i+1}-\boldsymbol{n}_i)^{\mathrm{T}}(\boldsymbol{n}_{i+1}-\boldsymbol{n}_i) \\ a_1 &= -28(\boldsymbol{n}_{i+1}-\boldsymbol{n}_i)^{\mathrm{T}}(\boldsymbol{t}_{i+1}+\boldsymbol{t}_i) \\ a_2 &= -1024 + 196(\boldsymbol{t}_{i+1}+\boldsymbol{t}_i)^{\mathrm{T}}(\boldsymbol{t}_{i+1}+\boldsymbol{t}_i) \\ &\quad +120(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)^{\mathrm{T}}(\boldsymbol{n}_{i+1}-\boldsymbol{n}_i) \\ a_3 &= -1680(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)^{\mathrm{T}}(\boldsymbol{t}_{i+1}+\boldsymbol{t}_i) \\ a_4 &= 3600(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)^{\mathrm{T}}(\boldsymbol{p}_{i+1}-\boldsymbol{p}_i)\end{aligned}\right\} \tag{15}$$

Given the boundary conditions of $\mathbf{p}_i$, $\mathbf{p}_{i+1}$, $\mathbf{t}_i$, $\mathbf{t}_{i+1}$, and $\mathbf{n}_i$, $\mathbf{n}_{i+1}$, the above polynomial is efficiently solved using Newton–Raphson iteration. The initial guess for l_i is taken as the chord length between consecutive control knots, i.e. $l_i^{\text{init}} = \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2}$. Once the parameter range l_i converges, the spline coefficients are calculated using Eq. (13). In order to schedule the feed motion correctly on top of the spline toolpath, the arc length S_k of each segment is integrated numerically. This is achieved by dividing the segment into M_k (≥ 100) subdivisions, proportional to its parameter length, computing the corresponding axis positions:

$$\left.\begin{aligned}x_{k,j} &= A_{xk}(j\cdot\Delta u)^5 + B_{xk}(j\cdot\Delta u)^4 + \ldots + F_{xk} \\ y_{k,j} &= A_{yk}(j\cdot\Delta u)^5 + B_{yk}(j\cdot\Delta u)^4 + \ldots + F_{yk} \\ z_{k,j} &= A_{zk}(j\cdot\Delta u)^5 + B_{zk}(j\cdot\Delta u)^4 + \ldots + F_{zk}\end{aligned}\right\}, \tag{16}$$

$$\Delta u = \frac{l_k}{M_k} \quad j = 0, 1, \ldots, M_k$$

and summing up the resulting arc increments:

$$S_k = \sum_{j=1}^{M_k} \Delta s_{k,j} = \sum_{j=1}^{M_k} \sqrt{(x_{k,j}-x_{k,j-1})^2 + (y_{k,j}-y_{k,j-1})^2 + (z_{k,j}-z_{k,j-1})^2} \tag{17}$$

3.1.2. Interpolation:

Although the spline is parameterized very closely to its arc length, there will still be feedrate fluctuation if it is interpolated by applying constant parameter increments (Δu), which are proportional to the desired arc increment (Δs) in the form:

$$\Delta u = \left(\frac{l_k}{S_k}\right) \cdot \Delta s \tag{18}$$

Furthermore, if the ratio of parameter range to segment arc length differs as small as 5% between consecutive segments (i.e. $l_k/S_k \neq l_{k+1}/S_{k+1}$), this will also cause a discontinuity in the feed profile where spline segments connect. Such discontinuity will result in high frequency acceleration and jerk harmonics, and will cause unwanted vibrations by exciting the structural modes of the drive system. In order to avoid such problems, the iterative interpolation technique devised by Erkorkmaz and Altintas [16] is applied. The basic idea is to solve the value of the spline parameter u iteratively in real-time to realize the desired arc displacement Δs as accurately as possible. This is done by constructing the arc increment Δs_{test} corresponding to the tested spline parameter value u_{test} as:

$$\Delta s_{\text{test}} = \sqrt{(\Delta x_{\text{test}})^2 + (\Delta y_{\text{test}})^2 + (\Delta z_{\text{test}})^2}$$

where

$$\begin{aligned} &\Delta x_{\text{test}} = x_{\text{test}} - x_{\text{prev}}, \quad \Delta y_{\text{test}} = y_{\text{test}} - y_{\text{prev}}, \\ &\Delta z_{\text{test}} = z_{\text{test}} - z_{\text{prev}} \\ &x_{\text{test}} = A_{xk}u_{\text{test}}^5 + B_{xk}u_{\text{test}}^4 + \ldots + F_{xk} \\ &y_{\text{test}} = A_{yk}u_{\text{test}}^5 + B_{yk}u_{\text{test}}^4 + \ldots + F_{yk} \\ &z_{\text{test}} = A_{zk}u_{\text{test}}^5 + B_{zk}u_{\text{test}}^4 + \ldots + F_{zk} \end{aligned} \tag{19}$$

where x_{prev}, y_{prev}, and z_{prev} are the axis commands applied in the previous control sample. Then the error between the desired and tested arc increment is evaluated as:

$$e = \Delta s_{\text{des}} - \Delta s_{\text{test}} \tag{20}$$

The gradient of the error with respect to the tested spline parameter is [16]:

$$\frac{de}{du_{\text{test}}} = \frac{\Delta x_{\text{test}}(dx/du)_{\text{test}} + \Delta y_{\text{test}}(dy/du)_{\text{test}} + \Delta z_{\text{test}}(dz/du)_{\text{test}}}{\Delta s_{\text{test}}} \tag{21}$$

where $(dx/du)_{\text{test}}$, $(dy/du)_{\text{test}}$, $(dz/du)_{\text{test}}$ are obtained by substituting the value of u_{test} into the first derivative expressions in Eq. (12). The correct value for the spline parameter is obtained by performing the below iteration:

$$(u_{\text{test}})_{i+1} = (u_{\text{test}})_i - \frac{e_i}{(de/du_{\text{test}})_i}, \tag{22}$$

i : iteration number

until the magnitude of the arc increment error e is below 10^{-12}. The initial guess for u_{test} is obtained through approximation of the nonlinear relationship between the arc length S and spline parameter u with a feed correction polynomial, constructed while estimating the total arc length S_k for each segment through Eq. (17). Availability of a close initial guess and an analytical gradient enables Eq. (22) to converge reliably, within three iterations, at each trajectory interpolation cycle [16].

3.2. Parameterization of add-in quintic spline

The add-in spline is parameterized by imposing additional control knots, and first and second order derivatives at these knots. Two different methods are proposed: the under-corner approach and over-corner approach, shown in Fig. 6(a) and (b), respectively. The under-corner approach is suitable for machining sharp corners in shorter time, while remaining within a given toolpath tolerance (Δ). This technique helps reduce the cycle time, and is applicable for feed drive systems with high control bandwidth, which are capable of accurately tracking the reference toolpath. Examples of such high bandwidth servo controllers are feedforward control [18] and adaptive sliding mode control [19].

The over-corner approach is based on smoothening the toolpath outside the original corner, while still remaining within the given tolerance (Δ). When low bandwidth servo controllers such as P, P–PI or PID are used, the contour error typically occurs as an undercut due to the large phase lag inherent in the controller. The reference trajectory is significantly ahead of the actual tool position, hence the commanded position turns around the corner before the tool is actually able to reach it. This results in the tool trajectory being diverted away from the sharp corner, towards the instantaneous reference position, which causes the undercut. The over-corner approach is proposed for correcting this problem, by slightly stretching the out reference toolpath, and bringing the actual tool trajectory closer to the originally intended sharp corner.

Both methods allow the corner to be machined without stopping, while maintaining continuous position, velocity, and acceleration profiles. In the following, they are presented in detail.

3.2.1. Under-corner approach

The toolpath geometry for the under-corner spline approach is shown in Fig. 6(a). Δ is the deviation from the original toolpath, which is bounded by a given tolerance

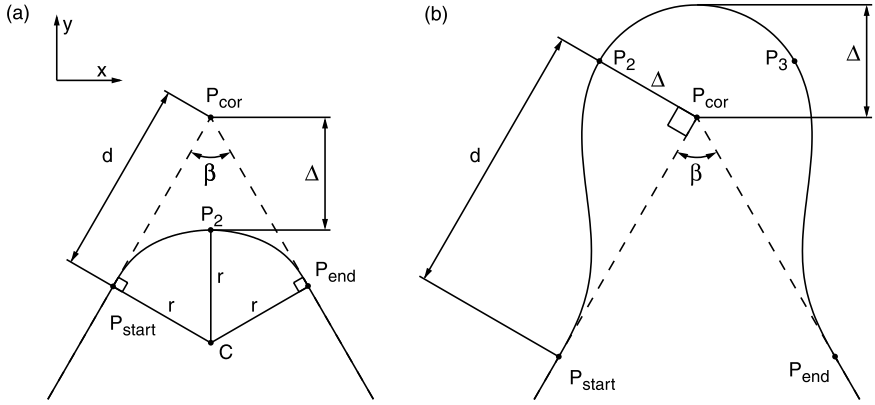

Fig. 6. Methodology for fitting a spline at the corner: (a) under-corner approach, (b) over-corner approach.

($\Delta \le \Delta_{\max}$) in order to ensure the dimensional integrity of the produced part. d is the clearance left in connecting the linear segments before the corner point $P_{\text{cor}}(x_c,y_c)$. β is the corner angle, $P_{\text{start}}(x_s,y_s)$ is the starting point, and $P_{\text{end}}(x_e,y_e)$ is the end point of the add-in spline. $P_2(x_2,y_2)$ is a midpoint control knot, introduced to ensure the shape of the add-in spline toolpath. r is the radius of curvature at P_2, and the distance from the curvature center C to P_{start} and P_{end}. The toolpath is generated by fitting two quintic spline segments, from P_{start} to P_2, and from P_2 to P_{end}. From geometry, it can be verified that:

$$r = d\cdot\tan(\beta/2),$$
$$\Delta = \frac{d}{\cos(\beta/2)} - r = \frac{1-\sin(\beta/2)}{\cos(\beta/2)}\cdot d \tag{23}$$

Given a desired linear clearance (d), the deviation from the original toolpath (Δ) is calculated using Eq. (23). If this violates the given toolpath tolerance (i.e. $\Delta > \Delta_{\max}$), then the clearance d is reduced so that the deviation from the original toolpath is within the given limit:

$$d' = \frac{\cos(\beta/2)}{1-\sin(\beta/2)}\cdot\Delta_{\max}, \qquad r' = d'\cdot\tan(\beta/2) \tag{24}$$

ensuring that $\Delta = \Delta_{\max}$. Considering Fig. 7(a), the unit tangent vectors $\boldsymbol{t}_1$ and $\boldsymbol{t}_3$ at the start and end points are expressed as:

$$\left.\begin{aligned}\mathbf{t}_1 &= \frac{\overrightarrow{P_{\text{start}}P_{\text{cor}}}}{|\overrightarrow{P_{\text{start}}P_{\text{cor}}}|} = \frac{(x_c-x_s)\mathbf{i}+(y_c-y_s)\mathbf{j}}{\sqrt{(x_c-x_s)^2+(y_c-y_s)^2}} = t_{1x}\cdot\mathbf{i}+t_{1y}\cdot\mathbf{j}\\ \mathbf{t}_3 &= \frac{\overrightarrow{P_{\text{cor}}P_{\text{end}}}}{|\overrightarrow{P_{\text{cor}}P_{\text{end}}}|} = \frac{(x_e-x_c)\mathbf{i}+(y_e-y_c)\mathbf{j}}{\sqrt{(x_e-x_c)^2+(y_e-y_c)^2}} = t_{3x}\cdot\mathbf{i}+t_{3y}\cdot\mathbf{j}\end{aligned}\right\} \tag{25}$$

The unit bisector vector $\boldsymbol{b}$ which divides the corner angle β in half is,

$$\mathbf{b} = \frac{-\mathbf{t}_1+\mathbf{t}_3}{|-\mathbf{t}_1+\mathbf{t}_3|} = \frac{(t_{3x}-t_{1x})\mathbf{i}+(t_{3y}-t_{1y})\mathbf{j}}{\sqrt{(t_{3x}-t_{1x})^2+(t_{3y}-t_{1y})^2}}$$
$$= b_x\cdot\mathbf{i}+b_y\cdot\mathbf{j} \tag{26}$$

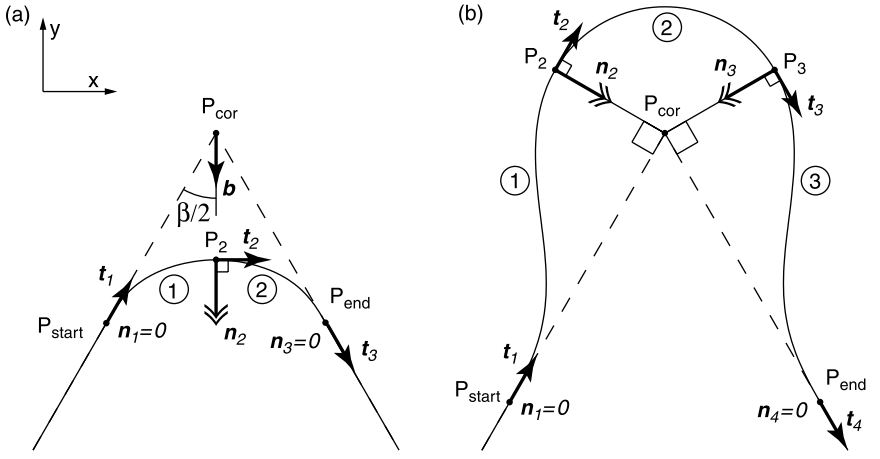

Fig. 7. Identification of first and second derivative boundary conditions (i.e. tangents and normals) in parameterizing (a) under-corner, (b) over-corner spline toolpath.

Using the given corner position $\boldsymbol{p}_c(x_c,y_c)$, the location of the control knot P_2 is determined as:

$$\mathbf{p}_2 = \mathbf{p}_{cor} + \Delta\cdot\mathbf{b} = (x_c + \Delta b_x)\mathbf{i} + (y_c + \Delta b_y)\mathbf{j} = p_{2x}\cdot\mathbf{i} + p_{2y}\cdot\mathbf{j} \quad (27)$$

The unit tangent at P_2 is obtained from the tangents $\boldsymbol{t}_1$ and $\boldsymbol{t}_3$ as,

$$\mathbf{t}_2 = \frac{\mathbf{t}_1+\mathbf{t}_3}{|\mathbf{t}_1+\mathbf{t}_3|} = \frac{(t_{1x}+t_{3x})\mathbf{i}+(t_{1y}+t_{3y})\mathbf{j}}{\sqrt{(t_{1x}+t_{3x})^2+(t_{1y}+t_{3y})^2}} = t_{2x}\cdot\mathbf{i}+t_{2y}\cdot\mathbf{j} \quad (28)$$

Since the corner spline connects two linear segments, the second derivative (i.e. normal) vectors, which indicate curvature at P_{start} and P_{end} have to be zero:

$$\mathbf{n}_1 = \mathbf{n}_3 = 0 \quad (29)$$

On the other hand, the normal vector at P_2 needs to impose a curvature of $1/r$, in the direction of the bisector $\boldsymbol{b}$:

$$\mathbf{n}_2 = \frac{1}{r}\mathbf{b} = \frac{1}{r}(b_x\cdot\mathbf{i}+b_y\cdot\mathbf{j}) = n_{2x}\cdot\mathbf{i}+n_{2y}\cdot\mathbf{j} \quad (30)$$

thus controlling the shape of the cornering spline. Using the position boundary conditions of P_{start}, P_2, P_{end}, tangent vectors $\boldsymbol{t}_1$, $\boldsymbol{t}_2$, $\boldsymbol{t}_3$, and normal vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$, $\boldsymbol{n}_3$ defined in Eqs. (25), (27)–(30), the two segment add-in quintic spline is parameterized and interpolated as described in Section 3.1.

3.2.2. Over-corner approach

The toolpath geometry for the over-corner spline is shown in Fig. 6(b). Similar to the under-corner approach, Δ is the deviation from the original toolpath where $\Delta \le \Delta_{max}$ has to hold. d is the linear segment clearance which can be in the range: $d=(1.5\text{–}6.0)\Delta$. From experience, it was seen that choosing $d\cong(5.0\text{–}6.0)\Delta$ resulted in the least contour error, providing a smooth transition into and out of the cornering spline. Two control knots $p_2(x_2,y_2)$ and $P_3(x_3,y_3)$ are generated by fitting three quintic segments: $P_{start}\to P_2$, $P_2\to P_3$, and $P_3\to P_{end}$.

$$\left.\begin{aligned}\mathbf{p}_2 &= \mathbf{p}_c + \kappa\cdot\Delta\cdot\mathrm{Rot}(90^\circ)\cdot\mathbf{t}_1 \to \begin{bmatrix}p_{2x}\\p_{2y}\end{bmatrix} = \begin{bmatrix}p_{cx}\\p_{cy}\end{bmatrix} + \kappa\cdot\Delta\cdot\begin{bmatrix}0&-1\\1&0\end{bmatrix}\cdot\begin{bmatrix}t_{1x}\\t_{1y}\end{bmatrix} = \begin{bmatrix}p_{cx}-\kappa\cdot\Delta\cdot t_{1y}\\p_{cy}+\kappa\cdot\Delta\cdot t_{1x}\end{bmatrix}\\ \mathbf{p}_3 &= \mathbf{p}_c + \kappa\cdot\Delta\cdot\mathrm{Rot}(90^\circ)\cdot\mathbf{t}_4 \to \begin{bmatrix}p_{3x}\\p_{3y}\end{bmatrix} = \begin{bmatrix}p_{cx}\\p_{cy}\end{bmatrix} + \kappa\cdot\Delta\cdot\begin{bmatrix}0&-1\\1&0\end{bmatrix}\cdot\begin{bmatrix}t_{4x}\\t_{4y}\end{bmatrix} = \begin{bmatrix}p_{cx}-\kappa\cdot\Delta\cdot t_{4y}\\p_{cy}+\kappa\cdot\Delta\cdot t_{4x}\end{bmatrix}\end{aligned}\right\} \quad (34)$$

Where $\mathrm{Rot}(\alpha)$is the rotation matrix around the z-axis by angle α [17]. Since the add-in spline connects two linear segments, which have no curvature, the normal vectors at P_{start} and P_{end} have to be zero:

$$\mathbf{n}_1 = \mathbf{n}_4 = 0 \quad (35)$$

The normal vectors at P_2 and P_3 are determined to impose a curvature of $1/\Delta$, pointing towards the corner point P_{cor}:

$$\left.\begin{aligned}\mathbf{n}_2 &= \frac{\kappa}{\Delta}\cdot\mathrm{Rot}(-90^\circ)\cdot\mathbf{t}_1 \to \begin{bmatrix}n_{2x}\\n_{2y}\end{bmatrix} = \frac{\kappa}{\Delta}\cdot\begin{bmatrix}0&1\\-1&0\end{bmatrix}\cdot\begin{bmatrix}t_{1x}\\t_{1y}\end{bmatrix} = \begin{bmatrix}\kappa\cdot t_{1y}/\Delta\\-\kappa\cdot t_{1x}/\Delta\end{bmatrix}\\ \mathbf{n}_3 &= \frac{\kappa}{\Delta}\cdot\mathrm{Rot}(-90^\circ)\cdot\mathbf{t}_4 \to \begin{bmatrix}n_{3x}\\n_{3y}\end{bmatrix} = \frac{\kappa}{\Delta}\cdot\begin{bmatrix}0&1\\-1&0\end{bmatrix}\cdot\begin{bmatrix}t_{4x}\\t_{4y}\end{bmatrix} = \begin{bmatrix}\kappa\cdot t_{4y}/\Delta\\-\kappa\cdot t_{4x}/\Delta\end{bmatrix}\end{aligned}\right\} \quad (36)$$

required for imposing the desired curvature, which determines the toolpath shape. The add-in spline is

Given the start, corner, and end points $P_{start}(x_s,y_s)$, $P_{cor}(x_c,y_c)$, and $P_{end}(x_e,y_e)$, the unit tangents $\mathbf{t}_1$ and $\mathbf{t}_4$ at points P_{start} and P_{end} are obtained as:

$$\left.\begin{aligned}\mathbf{t}_1 &= \frac{\overrightarrow{P_{start}P_{cor}}}{|\overrightarrow{P_{start}P_{cor}}|} = \frac{(x_c-x_s)\mathbf{i}+(y_c-y_s)\mathbf{j}}{\sqrt{(x_c-x_s)^2+(y_c-y_s)^2}} = t_{1x}\cdot\mathbf{i}+t_{1y}\cdot\mathbf{j}\\ \mathbf{t}_4 &= \frac{\overrightarrow{P_{cor}P_{end}}}{|\overrightarrow{P_{cor}P_{end}}|} = \frac{(x_e-x_c)\mathbf{i}+(y_e-y_c)\mathbf{j}}{\sqrt{(x_e-x_c)^2+(y_e-y_c)^2}} = t_{4x}\cdot\mathbf{i}+t_{4y}\cdot\mathbf{j}\end{aligned}\right\} \quad (31)$$

Considering the geometry in Fig. 7(b), the tangents $\mathbf{t}_2$ and $\mathbf{t}_3$ at points P_2 and P_3 are identical to $\mathbf{t}_1$ and $\mathbf{t}_3$, respectively:

$$\left.\begin{aligned}\mathbf{t}_2 = \mathbf{t}_1 \to t_{2x}\cdot\mathbf{i}+t_{2y}\cdot\mathbf{j} = t_{1x}\cdot\mathbf{i}+t_{1y}\cdot\mathbf{j}\\ \mathbf{t}_3 = \mathbf{t}_4 \to t_{3x}\cdot\mathbf{i}+t_{3y}\cdot\mathbf{j} = t_{4x}\cdot\mathbf{i}+t_{4y}\cdot\mathbf{j}\end{aligned}\right\} \quad (32)$$

In determining the position of $P_2(x_2,y_2)$ and $P_3(x_3,y_3)$, the direction of motion along the corner spline has to be known. Defining the direction variable κ, where $\kappa=1$ for clockwise (CW), and $\kappa=-1$ for counter-clockwise (CCW) motion, κ is obtained through the cross product of the tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_4$:

$$\kappa = -\mathrm{sign}(\mathbf{t}_1\times\mathbf{t}_4) = -\mathrm{sign}(t_{1x}t_{4y}-t_{1y}t_{4x}) = \begin{cases}1, & \text{CW}\\ -1, & \text{CCW}\end{cases} \quad (33)$$

The position vectors $\mathbf{p}_2(x_2,y_2)$ and $\mathbf{p}_3(x_3,y_3)$ are obtained in relation to the original corner location $\mathbf{p}_c(x_c,y_c)$ as:

Using the position boundary conditions of P_{start}, P_2, P_3, P_{end}, tangent vectors $\mathbf{t}_1$, $\mathbf{t}_2$, $\mathbf{t}_3$, $\mathbf{t}_4$ and normal vectors $\mathbf{n}_1$,$\mathbf{n}_2$, $\mathbf{n}_3$,$\mathbf{n}_4$ defined in Eqs. (31), (32), (34)–(36), the three segment

add-in spline is parameterized and interpolated as described in Section 3.1.

3.3. Determination of cornering feedrate along add-in spline

The add-in spline is traveled at a constant feedrate, with deceleration and acceleration being realized while the tool is traveling on the linear path segments. In order to keep the cornering time to a minimum, the cornering feedrate has to be chosen as high as possible, while ensuring that the actual tool motion does not violate the given tolerance Δ_{max}. The Virtual CNC, developed in Part I of this article is used for determining the feedrate, so that the predicted contour error always remains below the set tolerance.

An initial guess for the cornering feedrate is chosen and the contouring performance around the corner is simulated. The contour error profile is evaluated with respect to the original sharp toolpath, using the technique described in Section 2. A safety margin of 10% below the admissible deviation Δ_{max} is considered, in order to account for minor discrepancies between the predicted and experimental contour error profiles. The transition feedrate is adjusted iteratively [20] until the maximum value of the predicted contour error is just below $0.9\times\Delta_{max}$. The main advantage of using the Virtual CNC for feedrate adjustment is that it takes into account all major factors that affect the toolpath accuracy; including the dynamics of the drive system and control law, quality of the feedback instrumentation, nonlinearities such as stick-slip friction, actuator saturation and backlash, and smoothness of the reference trajectory. In return, this enables accurate tuning of the cornering feedrate, in order to travel the modified corner in minimum time while ensuring that the toolpath tolerance is not violated.

4. Contour machining with actual and Virtual CNC machines

The contouring accuracy improvement and cycle time reduction, realized with the proposed toolpath modification strategies, are demonstrated in experiments accompanied by predictions obtained with the Virtual CNC. A Fadal VMC 2216 three axis machining center, controlled with an in-house developed open architecture CNC [21] has been used for conducting the experiments. The virtual model of this machine was developed in Part I of this article [5]. The under-corner approach has been tested using adaptive sliding mode control (SMC) [19], which was designed to have a sliding surface bandwidth of $\lambda = 200$ rad/s ($\cong 32$ Hz). This bandwidth corresponds approximately to the disturbance rejection frequency of the control system. The SMC also has inherent feedforward action, which effectively

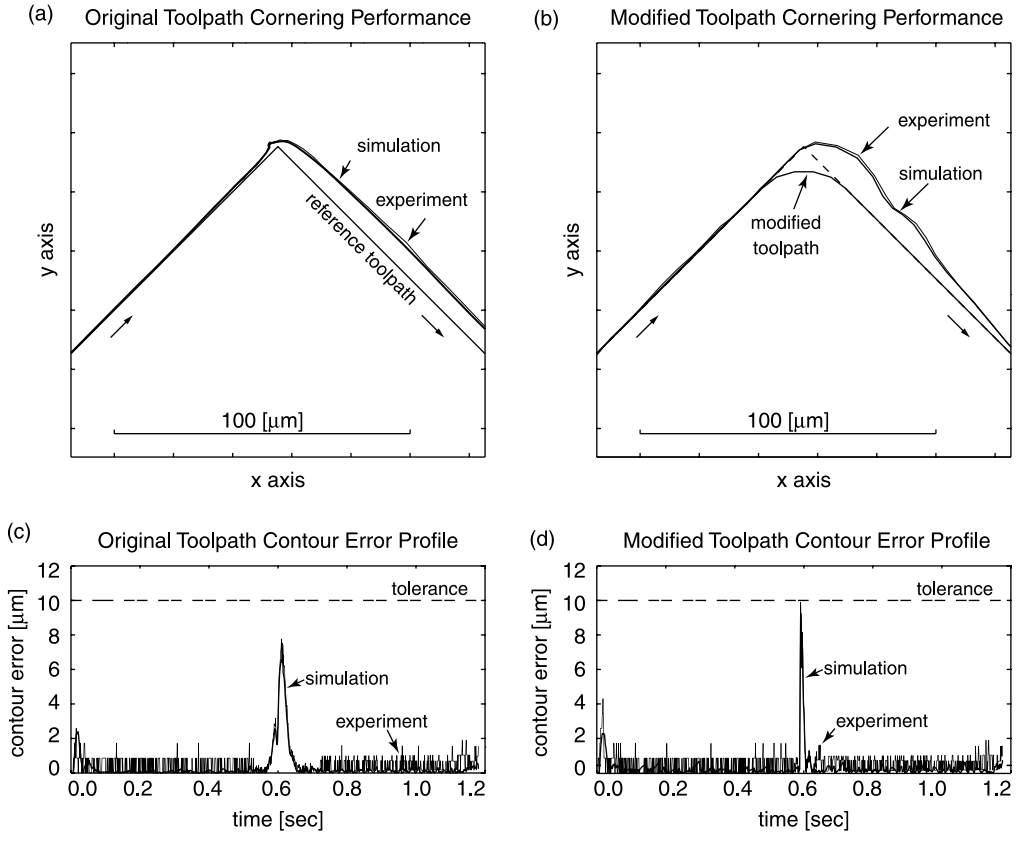

Fig. 8. Contouring performance for 90° under-corner spline with sliding mode control.

increases its command following bandwidth to 44 Hz, making it suitable for accurately tracking the commanded toolpath.

The over-corner approach was tested using P–PI control, which is a commonly used structure in CNC drive systems. The velocity loop is closed using PI control and the position loop is closed in cascade using only proportional (P) control. P–PI control possesses lower bandwidth, and therefore, turns around corners with a larger phase lag. This results in under-cut type of contouring errors, which will be compensated by the over-corner approach. The P–PI controller was tuned to yield an overall position tracking bandwidth of 15 Hz, and has no feedforward action. Both controllers have been implemented digitally at a sampling period of 1.0 ms. The encoder feedback resolution in both axes is 1.22 μm.

Experiments were conducted for turning around 90, 60, and 30° corners. The original toolpaths consisted of two 50 mm long linear segments connecting at a sharp corner at one of the mentioned angles. A constant feedrate of 100 mm/s is used for traveling along the linear toolpaths. Trapezoidal acceleration profiling [5] with 2000 mm/s^2 acceleration and 50000 mm/s^3 jerk magnitude is used for achieving smooth feedrate transitions. The original feed profile consists of starting from rest, traveling along the first linear segment at constant feed, decelerating to a full stop at the corner, accelerating and traveling at constant feed along the second linear segment, and decelerating to a full stop at the end of the toolpath. In the splined toolpaths, the speed is decelerated to the cornering feedrate before the tool enters the spline, and is accelerated back to the constant feedrate of 100 mm/s as the tool resumes traveling along the linear portion of the toolpath.

The under and over-corner approaches were first tested in the Virtual CNC. The transition feedrate was adjusted to ensure that the predicted contour error would remain within the given tolerance. The same tests were then replicated on the actual machine, which produced results in close agreement with the predictions. Each experiment was repeated three times, to ensure reliability of the results.

Simulation and experimental results for contouring with the under-corner approach are shown in Figs. 8–10. In testing the under-corner approach, a toolpath tolerance of $\Delta_{\max} = 10$ μm was used. The contouring accuracy and cornering time for each case has been summarized in Table 1. The cornering time was determined by removing the durations required to execute identical parts between the original and modified trajectories, from the total cycle time of each trajectory. These identical parts consisted of accelerating from rest at the beginning of the first linear segment, approaching the corner at a constant feedrate, leaving the corner at a constant feedrate, and decelerating to

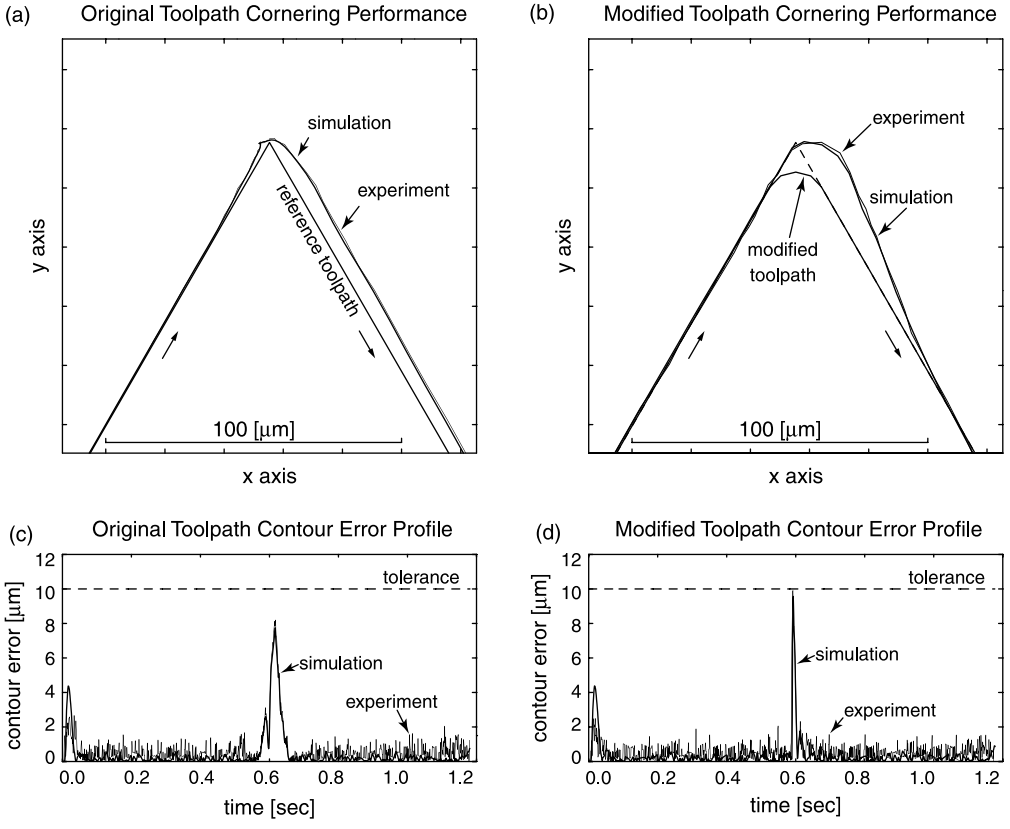

Fig. 9. Contouring performance for 60° under-corner spline with sliding mode control.

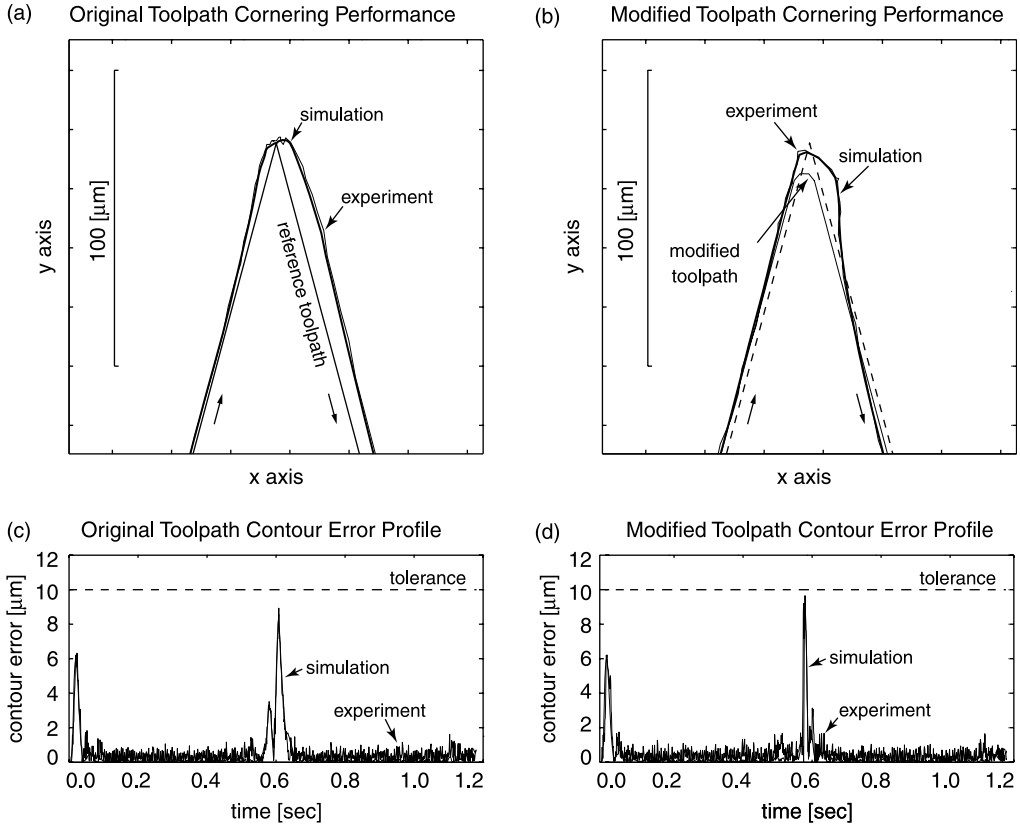

Fig. 10. Contouring performance for 30° under-corner spline with sliding mode control.

a stop at the end of the second linear segment. These durations were estimated by analyzing the reference trajectory profiles in each case.

For the 90° corner angle, the resulting clearance length and corner radius are $d=24.14$ μm and $r=24.14$ μm. The adjusted transition feedrate is $f=6.31$ mm/s. The original and modified toolpaths, and their corresponding contour error profiles are shown in Fig. 8. Considering Fig. 8(a), it is seen that the predicted and actual toolpaths are very close to each other, and both of them follow the commanded corner closely, with some overshoot after turning around the corner. In the case of the modified toolpath in Fig. 8(b), the actual tool position is also close to the original toolpath. The overshoot, although slightly larger in amplitude, dies out more rapidly. Considering the contour error profiles presented in Fig. 8(c) and (d), the maximum experimental contour error for the unmodified toolpath is 7.8 and 9.9 μm for the modified case, both of which are within the tolerance of 10 μm. Considering Table 1, the original cornering time of 185 ms is reduced by 2.16% to 181 ms by splining the toolpath, while preserving the contouring tolerance.

In case of the 60° corner angle, the clearance length and corner radius are $d=17.32$ and $r=10.00$ μm. Due to the smaller radius, the transition feedrate is also lower ($f=5.27$ mm/s). The contouring results are shown in Fig. 9. Considering Fig. 9(a), the predicted and actual toolpaths are again very close to each other, and show a similar trend to the 90° corner case. In the contour error profiles in Fig. 9(c) and (d), the maximum experimental contour error for the unmodified and modified toolpaths are 8.2 and 9.9 μm in the mentioned order, which are within the given tolerance. As indicated in Table 1, the original cornering time of 185 ms

Table 1
Summary of contouring results for under-corner spline with sliding mode servo control—toolpath tolerance: 10 μm

Corner toolpath	Maximum contour error (μm)		Cornering time (ms)	Relative cornering time (%)
	Simulation	Experiment		
90°—Sharp corner	7.5	7.8	185	100.00
90°—Splined toolpath	9.2	9.9	181	97.84
60°—Sharp corner	7.7	8.2	185	100.00
60°—Splined toolpath	9.6	9.9	182	98.38
30°—Sharp corner	8.6	8.9	183	100.00
30°—Splined toolpath	9.0	9.7	181	98.91

has been reduced by 1.62% to 182 ms by modifying the toolpath.

For the 30° corner, the clearance length and corner radius are $d=13.03$ μm and $r=3.49$ μm. The transition feedrate is $f=3.06$ mm/s, to account for the high curvature. The contouring results are shown in Fig. 10. Considering Fig. 10(a) and (b), it is seen that the predicted and actual toolpaths are very close, and the overshoot exhibits similarity to the 90 and 60° cases. In the contour error profiles in Fig. 10(c) and (d), the maximum experimental contour error for the unmodified and modified toolpaths are 8.9 and 9.7 μm, respectively, which are within the 10 μm limit. Considering Table 1, the cornering time reduction achieved by modifying the toolpath is 1.09%.

In the under-corner tests, it was demonstrated that around 1.0–2.5% reduction in cornering time can be achieved, by splining the toolpath to draw a small arc instead of traveling to the end of the sharp corner. This was done while ensuring that the actual tool motion remained within the given tolerance around the original toolpath, as verified experimentally. Although the cornering time reduction is not very large, its effect would still be evident in reducing the machining cycle time for complex parts, such as dies and molds.

Simulation and experimental results for contouring with the over-corner approach are shown in Figs. 11–13. In testing the over-corner approach, a toolpath tolerance of Δ_{max} 30 μm was used. The larger tolerance was chosen to account for the lower tracking accuracy of the P–PI controller, due to its low bandwidth compared to the sliding mode controller. The clearance length was chosen as $d=6\Delta_{max}=180$ μm in all cases. The smallest radius of curvature on the toolpath was $r=\Delta_{max}=30$ μm. The contouring accuracy and cornering time for each case has been summarized in Table 2.

The simulated and experimental contouring results for the 90° angle are shown Fig. 11. Considering Fig. 11(a), it is seen that the P–PI controller is not able to turn the sharp corner accurately, and results in a sizeable under-cut due to its large phase lag. Contour error violation points, which have been marked on the toolpath, indicate a geometric deviation of up to 288.9 μm from the original trajectory, which was determined by evaluating the contour error profile in Fig. 11(c). The simulated and experimentally recorded profiles are in close agreement, indicating reliable prediction by the Virtual CNC. The contouring profile with the modified toolpath is shown in Fig. 11(b). The transition feedrate in this case was adjusted to $f=2.66$ mm/s, in order to satisfy the contouring accuracy requirement. Considering Fig. 11(d), it is seen that the maximum value of the experimental contour error is 29.2 μm, which is within the specified tolerance of 30 μm. Although the over-corner approach helps maintain the toolpath tolerance by counteracting the under-cut, the additional travel length and low

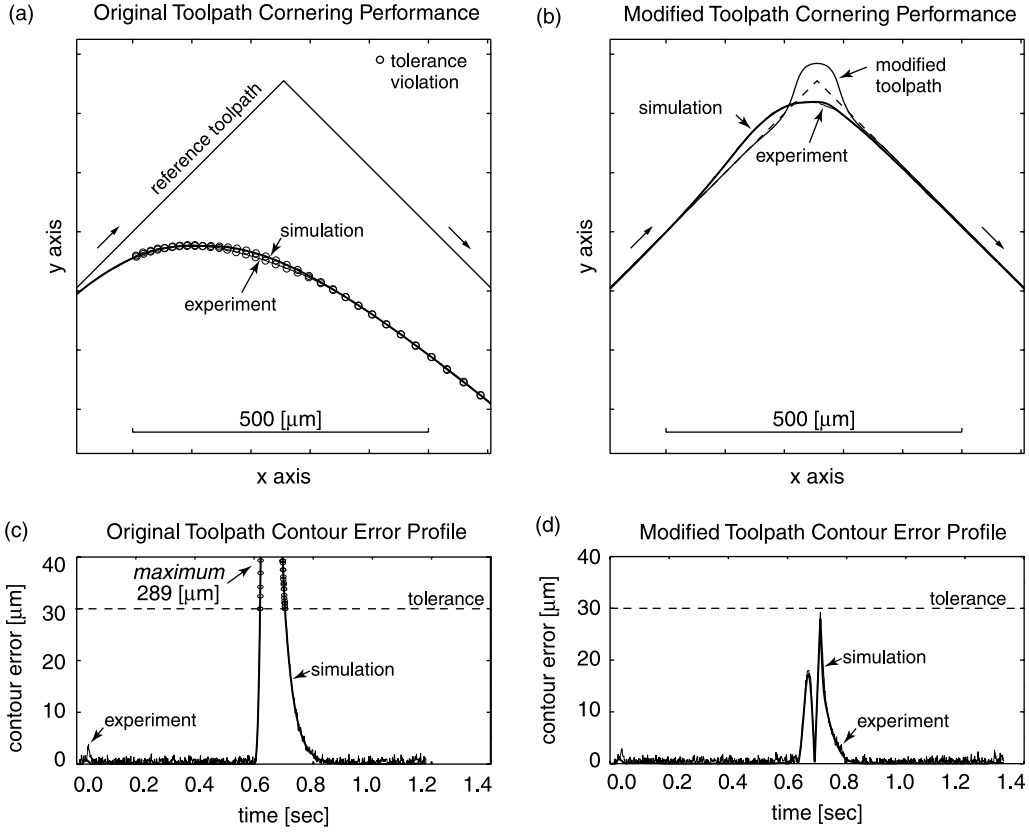

Fig. 11. Contouring performance for 90° over-corner spline with P–PI servo control.

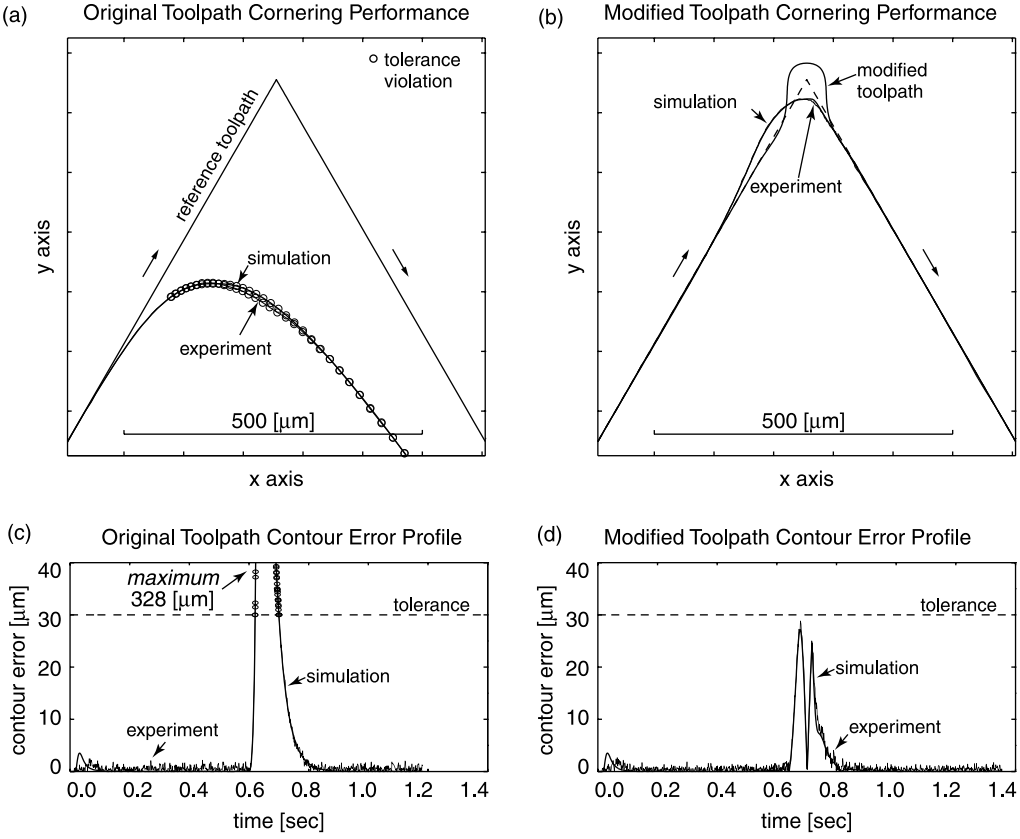

Fig. 12. Contouring performance for 60° over-corner spline with P–PI servo control.

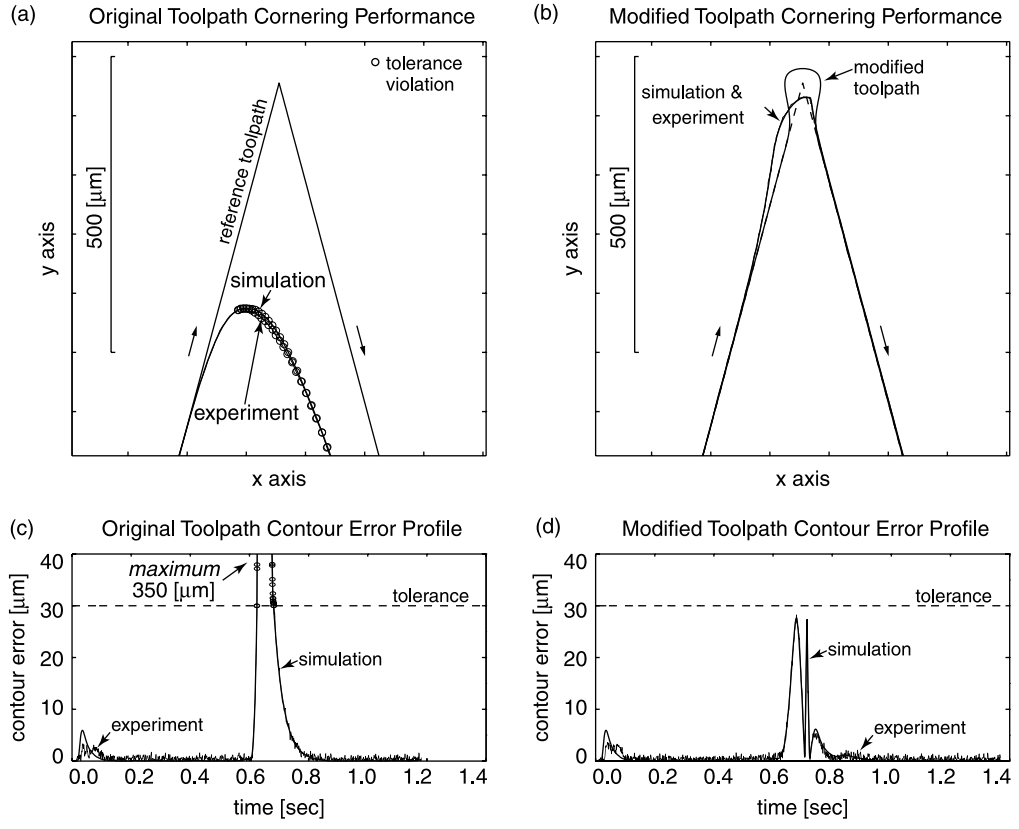

Fig. 13. Contouring performance for 30° over-corner spline with P–PI servo control.

Table 2
Summary of contouring results for over-corner spline with P–PI servo control—toolpath tolerance: 30 μm

Corner toolpath	Maximum contour error (μm)		Cornering time (ms)	Relative cornering time (%)
	Simulation	Experiment		
90°—Sharp corner	279.5	288.9	187	100.00
90°—Splined toolpath	27.9	29.2	338	180.75
60°—Sharp corner	318.3	327.9	185	100.00
60°—Splined toolpath	27.2	28.8	359	194.05
30°—Sharp corner	342.1	349.7	185	100.00
30°—Splined toolpath	27.7	28.3	373	201.62

cornering feedrate that it requires significantly increase the cornering time. Considering the results summarized in Table 2, it is seen that the original cornering time of 187 ms increases by 80.75% to 338 ms while realizing the over-corner toolpath for the 90° corner.

Contouring results for the 60° angle are shown in Fig. 12. The maximum experimental contour error for the original toolpath is 327.9 μm. By splining the corner and adjusting the transition feedrate to $f=2.42$ mm/s, the maximum value of the contour error is reduced to 28.8 μm, which is within the given tolerance. This is done at the expense of increasing the cornering time from 185 to 359 ms, which corresponds to a 94.05% increase.

The results for the 30° angle are shown in Fig. 13, which carry a similar trend to those obtained for the 90 and 60° cases. In this case, the maximum experimental contour error is 349.7 μm for the unmodified sharp corner. By modifying the toolpath with the over-corner spline, which is traveled at $f=1.98$ mm/s, the maximum value of the contour error is reduced to 28.3 μm. The increase in cornering time, paid as the trade-off for reducing the contour error, is 101.62% as indicated in Table 2.

The over-corner test results indicate the possibility of improving the contouring accuracy at sharp corners by stretching out the original corner with a smooth curve, in order to counteract the undercut effect of low bandwidth controllers. By planning the corner spline to remain within the toolpath tolerance, and adjusting the transition feedrate in the Virtual CNC environment, the possibility of ensuring a desired contouring accuracy has been experimentally demonstrated. Naturally, the contouring tolerance has to be specified within a reasonable range, which can be attained with the combination of the given drive system, servo controller, feedback device, and trajectory generation technique. The trade-off of the over-corner approach is the increase in cornering time, which was observed to be around 80–100%. Both the under-corner and over-corner methods were demonstrated to yield successful results for different corner angles ranging from 30 to 90°.

5. Conclusion

A trajectory planning strategy has been developed for maintaining the tool positioning accuracy in corner machining applications, by taking advantage of the accurate contour error prediction capability of the Virtual CNC. Two spline fitting techniques have been developed. The under-corner approach reduces the toolpath length, and therefore the cornering time, and is suitable for drives with high bandwidth. The over-corner approach helps counteract the 'under-cut' caused by the phase lag of low bandwidth controllers. Adjustment of the transition feedrate on the Virtual CNC ensures that contour error violation does not occur on the real machine. Both methods were demonstrated to yield successful results with close agreement between experiments conducted on real and Virtual CNC machines.

Acknowledgements

This research is sponsored by NSERC and Pratt & Whitney Canada under research chair and strategic grant agreements.

References

[1] M. Weck, A. Meylahn, C. Hardebusch, Innovative algorithms for spline-based CNC controller, Production engineering research and development in Germany, Annals of the German Academic Society for Production Engineering 6 (1) (1999) 83–86.
[2] J.E. Bobrow, S. Dubowsky, J.S. Gibson, Time-optimal control of robotic manipulators along specified paths, The International Journal of Robotics Research 4 (3) (1985) 3–17.
[3] J. Butler, B. Haack, M. Tomizuka, Reference generation for high speed coordinated motion of a two axis system, Symposium on Robotics, ASME Winter Annual Meeting, Chicago, IL, USA, DSC-11, 1988, pp. 457–470.
[4] Y. Altintas, K. Erkorkmaz, Feedrate optimization for spline interpolation in high speed machine tools, Annals of CIRP 52 (1) (2003) 297–302.
[5] C.-H. Yeung, Y. Altintas, K. Erkorkmaz, Virtual CNC system—Part I. System architecture, International Journal of Machine Tools and Manufacture, doi: 10.1016/j.ijmachtools.2005.08.002.
[6] Y. Koren, Cross-coupled biaxial computer control for manufacturing systems, Journal of Dynamic Systems, Measurement, and Control 102 (1980) 265–272.
[7] J. Dong, J.A. Stori, Optimal feed-rate scheduling for high-speed contouring, Proceedings of ASME-IMECE'03, MED-42357, 2003.
[8] Y. Koren, C.C. Lo, Variable-gain cross-coupling controller for contouring, Annals of CIRP 40 (1) (1991) 371–374.

[9] K. Erkorkmaz, Y. Altintas, High speed contouring control algorithm for CNC machine tools, Proceedings of ASME Dynamic Systems and Control Division, IMECE'98, DSC-64 1998, pp. 463–469.
[10] M.K. Jouaneh, Z. Wang, D.A. Dornfeld, Tracking of sharp corners using a robot and a table manipulator, Proceedings of USA–Japan Symposium on Flexible Automation, 1988, pp. 271–278.
[11] M.K. Jouaneh, Z. Wang, D.A. Dornfeld, Trajectory planning for coordinated motion of a robot and a positioning table. Part 1. Path specification, IEEE Transactions on Robotics and Automation 6 (6) (1990) 735–745.
[12] M.K. Jouaneh, D.A. Dornfeld, M. Tomizuka, Trajectory planning for coordinated motion of a robot and a positioning table. Part 2. Optimal trajectory specification, IEEE Transactions on Robotics and Automation 6 (6) (1990) 746–759.
[13] Y. Kanayama, N. Miyake, Trajectory generation for mobile Robots, Third International Symposium on Robotics Research, Cambridge, MA, 1986, pp. 333–340.
[14] M. Weck, G. Ye, Sharp corner tracking using the IKF control strategy, Annals of CIRP 39 (1) (1990) 437–441.
[15] F.-C. Wang, D.C.H. Yang, Nearly arc-length parameterized quintic-spline interpolation for precision machining, Computer Aided Design 25 (5) (1993) 281–288.
[16] K. Erkorkmaz, Y. Altintas, Quintic spline interpolation with minimal feed fluctuation, ASME Journal of Manufacturing Science and Engineering 127 (2) (2005) 339–349.
[17] T. Yoshikawa, Foundations of Robotics, The MIT Press, Cambridge, MA, 1990.
[18] M. Tomizuka, Zero phase error tracking algorithm for digital control, ASME Journal of Dynamic Systems, Measurement, and Control 109 (1987) 65–68.
[19] Y. Altintas, K. Erkorkmaz, W.-H. Zhu, Sliding mode controller design for high speed drives, Annals of CIRP 49 (1) (2000) 265–270.
[20] C.-H. Yeung, MASc thesis: a three-axis virtual computer numerical-controlled (CNC) System, University of British Columbia, Department of Mechanical Engineering, BC, Canada, 2004.
[21] Y. Altintas, N.A. Erol, Open architecture modular tool kit for motion and machining process control, Annals of CIRP 47 (1) (1998) 295–300.