

# UC Davis

## UC Davis Previously Published Works

### Title

A real-time CNC interpolator algorithm for trimming and filling planar offset curves

### Permalink

<https://escholarship.org/uc/item/60v7b269>

### Authors

Farouki, Rida T  
Srinathu, Jyothirmai

### Publication Date

2017-05-01

### DOI

10.1016/j.cad.2017.01.001

Peer reviewed

# A real-time CNC interpolator algorithm for trimming and filling planar offset curves

Rida T. Farouki and Jyothirmai Srinathu  
Department of Mechanical and Aerospace Engineering,  
University of California, Davis, CA 95616, USA.

## Abstract

Tool paths for CNC machines must be offset from the desired part shape, in order to compensate for the tool radius. To avoid gouging the part geometry and to ensure continuous paths, the offset construction employs *trimming* and *filling* operations at tangent discontinuities and regions of high concave curvature on the part shape. Typically, offset paths are constructed offline in a CAM system — the resulting paths are inherently approximate, and must be re-generated when a different tool size is selected. To circumvent these shortcomings, an interpolator algorithm for real-time offset curve trimming and filling is developed and verified herein. Since the algorithm uses the exact part geometry, offset path approximation errors are completely eliminated. Circular fill arcs of the appropriate angular extent are automatically executed at convex tangent-discontinuous junctures of adjacent part boundary segments, and offset trimming operations are triggered by real-time point/curve distance computations, which determine footpoints of the instantaneous tool position on the part boundary. The algorithm also accommodates feedrates that correspond to a constant speed of either the tool/part contact point, or the tool center. The practical feasibility of the method is demonstrated by an implementation on a 3-axis CNC mill governed by an open-architecture software controller.

**Keywords:** CNC machine, tool radius compensation, offset curve trimming and filling, point/curve distance function, footpoint, real-time interpolator, open-architecture controller.

e-mail addresses: farouki@ucdavis.edu, jsrinathu@ucdavis.edu

# 1 Introduction

Computer-aided design and manufacturing (CAD/CAM) technology plays a critical role in enhancing industrial productivity. Typically, geometric models designed within a CAD system are passed on to a CAM package, to generate part programs specifying the motion commands and other process parameters that govern a variety of computer-controlled part fabrication processes, such as CNC machining, 3D printing, and laser or waterjet cutting.

In certain respects, the CAM generation of process-specific part programs from CAD models is the weakest link of the chain from product conception to manufactured part. For example, tool path geometries are often incompatible with simple exact representations that machine controllers can interpret, and are consequently subject to data-intensive approximations. Discretized paths incur a loss of higher-order (curvature) information, essential to acceleration management, and severely impede the ability to smoothly execute high-speed motions. Tool paths are also specific to *a priori* choices of tooling, and part programs must be re-generated to accommodate tool changes.

The availability of fast processors for computer numerical control (CNC) machines now allows more-sophisticated computations, based upon the exact CAD geometry, to execute in real time. Although complete circumvention of the assumptions and approximations incurred by CAM part programs is an ambitious goal, important steps in this direction are nevertheless feasible and worthwhile. For example, a recent study [19] demonstrates the possibility of machining a versatile family of swept surface geometries directly from their high-level procedural definitions, yielding greater accuracy and versatility in tool choice, scallop height control, roughing/finishing passes, etc.

Within this context, the goal of the present study is to develop a real-time tool radius compensation strategy for machining of piecewise-analytic planar shapes with a cylindrical end mill. The usual CAM solution to this problem entails offline computation of the *offset curve* to the part shape, for a specified tool radius. In general, the computation of offset curves is a challenging task involving both local and global geometrical aspects of a given shape, in which the *point/curve distance function* plays a key role. However, the possibility of real-time computation of this function with a modest processor has recently been demonstrated [4] in the context of precise contour error measurement for the cross-coupled control of multi-axis CNC machines.

The real-time point/curve distance function computation, introduced in [4], is generalized herein to the case of piecewise-analytic curves and applied

to the real-time tool radius offset compensation problem. This methodology follows the STEP-NC philosophy [1, 13, 26] of relying more directly on precise CAD geometry data in CNC machining, and offers many advantages over the prevailing reliance on offline CAM-generated offset tool paths — for example:

- Modifications to the part shape or tool radius are easily accommodated, since they do not incur the need to re-generate a part program.
- Free-form curves do not, in general, admit exact (i.e., rational) offsets, so CAM-generated toolpaths are necessarily approximations. However, since the real-time offset compensation strategy uses only the original exact part geometry, the need for approximation is eliminated.
- Smoother execution of curvilinear motions, with accurate maintenance of feedrates, becomes possible when analytic path geometry is available in lieu of highly discretized piecewise-linear/circular approximations.
- Greater flexibility in specifying feedrates is possible — e.g., the feedrate may correspond to constant speed of either the tool/part contact point (yielding an approximately constant chip load), or of the tool center.
- Higher-order path information (e.g., curvature) can be computed from the exact part geometry in real time, and used for the management of axis accelerations, path tangent discontinuities, cutting forces, etc.

For brevity, the methodology is considered at present only in the context of machining planar shapes specified by closed piecewise-polynomial curves. Also, due to constraints on the complexity of feasible computations in real-time motion control, the present focus is on “local” offset trimming, in which the only necessary trimming operations result from high-curvature concave regions within a single boundary segment, or concave tangent-discontinuous junctures between adjacent segments. The “global” offset trimming problem for arbitrarily complex free-form shapes is computationally very demanding, and careful consideration must be given to issues of computational efficiency to extend the present methodology to accommodate interference of the offsets to arbitrary boundary segment pairs. A detailed treatment of the real-time global trimming problem is beyond the scope of the present study.

In general, local offset trimming accommodates the important “finish cut” motion, which must accurately achieve the desired part shape to a prescribed tolerance, with an appropriately smooth surface finish. The ability to drive

the finish machining operation directly from the exact analytic part geometry is a major advantage in this context. For a given part shape and tool radius, the validity of the real-time local offset trimming can be readily verified by an *a priori* off-line check. The practical feasibility of this approach to finish machining operations is demonstrated by its implementation and verification on a 3-axis CNC mill with an open-architecture software controller.

The authors are unaware of any prior investigations aimed at developing algorithms for real-time offset curve trimming and filling, and verifying their feasibility by implementation on a CNC machine, with performance assessed by the analysis of real-time position encoder data. The paper [20] describes a “locus tracing algorithm” for following offset paths to given smooth curves, without (as in the present study) explicit representations of their offsets, but does not directly address the critically important offset trimming and filling functions that are the main focus of the present investigation. Global offset trimming and filling can be achieved by the use of “level set” methods [2, 21]. However, these grid-based methods are computationally intensive, especially if grid resolutions compatible with CNC machine accuracy requirements are employed, and are thus unsuitable for real-time implementation.

The plan for the remainder of this paper is as follows. Sections 2 and 3 review some fundamental properties of the point/curve distance function and planar offset curves. These are employed in Section 4 to develop algorithms for “local” real-time trimming and filling of offset curves. An implementation of these algorithms on a 3-axis CNC mill governed by an open-architecture software controller, together with real-time performance data for several test curves, is presented in Section 5, and their extension to address the more-challenging problem of “global” offset curve trimming and filling is briefly discussed. Finally, Section 6 summarizes the contributions of this study, and identifies possible further developments of the methodology.

## 2 Point/curve distance function

The point/curve distance function plays a fundamental role in real-time offset curve trimming and filling. For any given point  $\mathbf{p} = (x_p, y_p)$  and parametric curve  $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ ,  $\xi \in [0, 1]$  this function is defined by

$$\text{distance}(\mathbf{p}, \mathbf{r}(\xi)) := \min_{\xi \in [0,1]} |\mathbf{p} - \mathbf{r}(\xi)| = \min_{0 \leq i \leq n+1} |\mathbf{p} - \mathbf{r}(\xi_i)|, \quad (1)$$

where  $\xi_0 = 0$  and  $\xi_{n+1} = 1$ , while  $\xi_1, \dots, \xi_n$  are the odd-multiplicity roots<sup>1</sup> on  $\xi \in (0, 1)$  of the function

$$F(x_p, y_p, \xi) := [x_p - x(\xi)]x'(\xi) + [y_p - y(\xi)]y'(\xi). \quad (2)$$

If  $\mathbf{r}(\xi)$  is a degree  $d$  polynomial curve,  $F(x_p, y_p, \xi)$  is of odd degree  $2d - 1$  in  $\xi$ , and thus has at least one real root. The real roots of (2) identify points of  $\mathbf{r}(\xi)$  where  $\mathbf{p}$  lies on the curve *normal line*. On traversing the curve,  $|\mathbf{p} - \mathbf{r}(\xi)|$  attains a local stationary value at  $\xi_1, \dots, \xi_n$  since

$$\frac{d}{d\xi} |\mathbf{p} - \mathbf{r}(\xi)| = - \frac{F(x_p, y_p, \xi)}{|\mathbf{p} - \mathbf{r}(\xi)|}.$$

The value of distance( $\mathbf{p}, \mathbf{r}(\xi)$ ) is the smallest of the interior extremal distances *and* the distances to the curve endpoints  $\mathbf{r}(0)$  and  $\mathbf{r}(1)$ . Minima of  $|\mathbf{p} - \mathbf{r}(\xi)|$  can be distinguished from maxima by noting that the second derivative may be expressed as

$$\frac{d^2}{d\xi^2} |\mathbf{p} - \mathbf{r}(\xi)| = - \frac{F'(x_p, y_p, \xi)}{|\mathbf{p} - \mathbf{r}(\xi)|} - \frac{F^2(x_p, y_p, \xi)}{|\mathbf{p} - \mathbf{r}(\xi)|^3},$$

where

$$\begin{aligned} F'(x_p, y_p, \xi) &= [\mathbf{p} - \mathbf{r}(\xi)] \cdot \mathbf{r}''(\xi) - |\mathbf{r}'(\xi)|^2 \\ &= [x_p - x(\xi)]x''(\xi) + [y_p - y(\xi)]y''(\xi) - [x'^2(\xi) + y'^2(\xi)]. \end{aligned}$$

Since  $F(x_p, y_p, \xi_i) = 0$  for  $i = 1, \dots, n$  we see that  $\xi_i$  identifies a minimum of  $|\mathbf{p} - \mathbf{r}(\xi)|$ , with a positive second derivative, when  $F'(x_p, y_p, \xi_i) < 0$  — i.e.,

$$[x_p - x(\xi_i)]x''(\xi_i) + [y_p - y(\xi_i)]y''(\xi_i) < x'^2(\xi_i) + y'^2(\xi_i).$$

If the minimum in (1) is realized with  $i = k$ , we call  $\mathbf{r}(\xi_k)$  a *footpoint* of  $\mathbf{p}$  on the curve  $\mathbf{r}(\xi)$ . When  $1 \leq m \leq n$  it is called an *interior* footpoint, but if  $m = 0$  or  $n + 1$  it is called a *terminal* footpoint. Ordinarily,  $\mathbf{p}$  has a unique footpoint on  $\mathbf{r}(\xi)$ , but if the point  $\mathbf{p}$  lies on the *self-bisector* or *medial axis* of  $\mathbf{r}(\xi)$ , there can be more than one footpoint [10].

For boundary curves specified in the customary Bézier/B-spline form, it is natural to express the polynomial (2) in the Bernstein basis on  $\xi \in [0, 1]$ . Its real roots on this interval can then be accurately and efficiently computed

---

<sup>1</sup>Even-multiplicity roots identify stationary, but non-extremal, values of  $|\mathbf{p} - \mathbf{r}(\xi)|$ .

by exploiting the subdivision and variation–diminishing properties [7] of the Bernstein form. The method employed herein determines isolating intervals for each real root over which elementary conditions [15] for the guaranteed convergence of Newton–Raphson iterations hold.

The real–time offset curve trimming and filling algorithm is based on the following two basic functions:

- $\text{footpoint}(\mathbf{p}, \mathbf{r}(\xi))$  gives the parameter value of the footpoint of a point  $\mathbf{p}$  on the curve segment  $\mathbf{r}(\xi)$  — when there is more than one footpoint, the largest of their parameter values is returned;
- $\text{distance}(\mathbf{p}, \mathbf{r}(\xi))$  returns the distance of a point  $\mathbf{p}$  from segment  $\mathbf{r}(\xi)$ .

$\text{distance}(\mathbf{p}, \mathbf{r}(\xi))$  calls  $\text{footpoint}(\mathbf{p}, \mathbf{r}(\xi))$ , but it will also be necessary to call the latter independently of the former.

### 3 Untrimmed and trimmed offset curves

We consider a part shape defined by a closed planar contour  $C$ , comprising a sequence of curve segments  $\mathbf{r}_i(\xi) = (x_i(\xi), y_i(\xi))$ ,  $\xi \in [0, 1]$  for  $i = 1, \dots, N$ . These segments meet end–to–end with (at least) point continuity, i.e.,

$$\mathbf{r}_i(1) = \mathbf{r}_{i+1}(0), \quad i = 1, \dots, N - 1 \quad \text{and} \quad \mathbf{r}_N(1) = \mathbf{r}_1(0),$$

such that  $C$  bounds a simply–connected domain. The parameterizations are assumed to be such that  $C$  has an anti–clockwise orientation. The distance of a point  $\mathbf{p}$  from the entire contour  $C$  is defined by

$$\text{distance}(\mathbf{p}, C) := \min_{1 \leq i \leq N} \text{distance}(\mathbf{p}, \mathbf{r}_i(\xi)). \quad (3)$$

Along segment  $\mathbf{r}_i(\xi)$ , the *tangent* and *normal* vectors and the *curvature* are defined [25] by

$$\mathbf{t}_i(\xi) = \frac{\mathbf{r}'_i(\xi)}{|\mathbf{r}'_i(\xi)|}, \quad \mathbf{n}_i(\xi) = \mathbf{t}_i(\xi) \times \mathbf{z}, \quad \kappa_i(\xi) = \frac{[\mathbf{r}'_i(\xi) \times \mathbf{r}''_i(\xi)] \cdot \mathbf{z}}{|\mathbf{r}'_i(\xi)|^3}, \quad (4)$$

where  $\mathbf{z}$  is a unit vector orthogonal to the plane. Note that the normal  $\mathbf{n}_i(\xi)$  points locally to the *right* of segment  $\mathbf{r}_i(\xi)$  as it is traversed with increasing  $\xi$ , and  $\kappa_i(\xi)$  is negative or positive according to whether  $\mathbf{n}_i(\xi)$  points toward

or away from the center of curvature. The *nodal points*  $\mathbf{q}_1, \dots, \mathbf{q}_N$  of  $C$  are the junctures of its individual curve segments<sup>2</sup> — i.e.,

$$\mathbf{q}_i = \mathbf{r}_i(1) = \mathbf{r}_{i+1}(0), \quad i = 1, \dots, N-1 \quad \text{and} \quad \mathbf{q}_N = \mathbf{r}_N(1) = \mathbf{r}_1(0).$$

A nodal point  $\mathbf{q}_i$  is *tangent continuous* if  $\mathbf{t}_i(1) = \mathbf{t}_{i+1}(0)$ . A nodal point that is not tangent–continuous amounts to a curvature impulse, since the tangent and normal exhibit finite instantaneous rotations at such points.

The *untrimmed offset* at signed distance  $d$  to the curve segment  $\mathbf{r}_i(\xi)$  is the locus defined by

$$\mathbf{r}_{d,i}(\xi) = \mathbf{r}_i(\xi) + d\mathbf{n}_i(\xi), \quad \xi \in [0, 1]. \quad (5)$$

This defines an *interior* or *exterior* offset to  $\mathbf{r}_i(\xi)$  according to whether  $d$  is negative or positive. However, the untrimmed offset to the entire contour  $C$  is not simply the union of the offsets to all its individual curve segments. If  $\mathbf{q}_i$  is a tangent–continuous nodal point, the segments  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$  have contiguous untrimmed offsets, i.e.,  $\mathbf{r}_{d,i}(1) = \mathbf{r}_{d,i+1}(0)$ . However, if  $\mathbf{q}_i$  is not tangent–continuous, the untrimmed offsets are not contiguous.

The untrimmed offset can be regarded as the locus traced by the vector  $d\mathbf{n}$  along  $C$ , where  $\mathbf{n}$  is the normal along  $C$ . The variation of  $\mathbf{n}$  with arc length  $s$  may be expressed in terms of the tangent  $\mathbf{t}$  and curvature  $\kappa$  of  $C$  as

$$\frac{d\mathbf{n}}{ds} = \kappa \mathbf{t}.$$

Now at a nodal point  $\mathbf{q}_i$  that is not tangent–continuous, the normal  $\mathbf{n}$  to  $C$  is not uniquely determined. However, consistent with the interpretation of such a point as a curvature impulse,  $\mathbf{n}$  may be considered to undergo a finite instantaneous rotation  $\Delta\theta_i$  about that point, so that  $d\mathbf{n}$  traces a circular arc connecting  $\mathbf{r}_{d,i}(1)$  and  $\mathbf{r}_{d,i+1}(0)$ . If each tangent–discontinuous nodal point is treated in this manner, the untrimmed offset to  $C$  forms a continuous closed curve (although it may exhibit self–intersections).

The process of ensuring a continuous untrimmed offset, by introducing a circular arc associated with each tangent–discontinuous node of  $C$ , is called *offset filling* — Figure 1 illustrates this for a simple polygonal contour. Each point of the untrimmed offset constructed in this manner is precisely distance

---

<sup>2</sup>It should be understood that the index  $i$  will henceforth be interpreted cyclically — i.e.,  $i = N + 1$  is replaced by  $i = 1$ .

$|d|$  from some point  $C$ . However, its distance from the *entire* contour  $C$ , in the sense of the function (3), may be less than  $|d|$ . Portions of the untrimmed offset containing such points must be deleted, to obtain the *trimmed* or “true” offset to  $C$ . Figure 1 illustrates this *offset trimming* process.

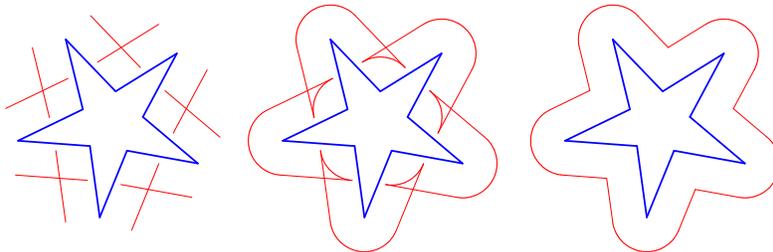


Figure 1: Left: the untrimmed offsets to each individual segment of a simple polygonal curve  $C$ . Center: filling the untrimmed offset by the insertion of circular arcs associated with each tangent–discontinuous nodal point. Right: trimmed offset obtained by removing all segments at distance  $< |d|$  from  $C$ .

Let  $\Delta\theta_i \in (-\pi, +\pi)$  be the finite rotation mapping  $\mathbf{n}_i(1)$  onto  $\mathbf{n}_{i+1}(0)$  at a tangent–discontinuous nodal point  $\mathbf{q}_i$ , satisfying

$$\cos \Delta\theta_i = \mathbf{n}_i(1) \cdot \mathbf{n}_{i+1}(0) \quad \text{and} \quad \sin \Delta\theta_i = [\mathbf{n}_i(1) \times \mathbf{n}_{i+1}(0)] \cdot \mathbf{z},$$

with angles considered positive anti–clockwise and negative clockwise. Then  $\mathbf{q}_i$  is *convex* or *concave*, relative to the offset distance  $d$ , according to whether the product  $d \Delta\theta_i$  is positive or negative. As seen in Figure 1, a convex nodal point  $\mathbf{q}_i$  incurs a “gap” between the untrimmed offset segments  $\mathbf{r}_{d,i}(\xi)$  and  $\mathbf{r}_{d,i+1}(\xi)$ , that must be *filled* by the circular arc corresponding to the rotation of  $d \mathbf{n}_i(1)$  onto  $d \mathbf{n}_{i+1}(0)$  through angle  $\Delta\theta_i$ . On the other hand, a concave nodal point  $\mathbf{q}_i$  incurs an “overlap” between  $\mathbf{r}_{d,i}(\xi)$  and  $\mathbf{r}_{d,i+1}(\xi)$ , resulting in a self–intersection loop of the untrimmed offset that must be *trimmed*.

In a trimming operation incurred by a concave nodal point  $\mathbf{q}_i$ , it is not necessary to insert the circular arc generated by the rotation of  $d \mathbf{n}_i(1)$  onto  $d \mathbf{n}_{i+1}(0)$  — it suffices to cut away the portions  $\xi \in [\xi_l, 1]$  and  $\xi \in [0, \xi_r]$  of  $\mathbf{r}_{d,i}(\xi)$  and  $\mathbf{r}_{d,i+1}(\xi)$  where  $\xi_l$  and  $\xi_r$  are, respectively, the smallest and largest parameter values such that  $\mathbf{r}_{d,i}(\xi_l) = \mathbf{r}_{d,i+1}(\xi_r)$  — i.e.,  $\xi_l$  and  $\xi_r$  identify an intersection point of consecutive untrimmed offset segments.

In addition to the filling/trimming operations associated with the tangent–discontinuous nodal of points of  $C$ , the untrimmed offset may require further

trimming operations to obtain the trimmed offset. Every portion  $[\xi_l, \xi_r] \subseteq [0, 1]$  of an untrimmed offset segment  $\mathbf{r}_{d,i}(\xi)$  such that

$$\text{distance}(\mathbf{r}_{d,i}(\xi), C) < |d| \quad \text{for } \xi \in (\xi_l, \xi_r)$$

should be trimmed away. Such portions are delineated by untrimmed offset points that have more than one footpoint on  $C$ , and these footpoints may be on a single segment or on completely unrelated segments of  $C$ . For planar contours  $C$  that comprise  $N$  free-form curve segments, with nodal points that are not necessarily tangent-continuous, we distinguish between two regimes of the general offset curve trimming problem, as follows.

In *local trimming*, all required trimming operations are incurred by either (a) concave tangent-discontinuous nodes between consecutive segments of  $C$ ; or (b) untrimmed offset segments delineated by self-intersections that have footpoints on a single segment of  $C$ . If  $\xi_l$  and  $\xi_r$  are the smallest and largest distinct values in  $[0, 1]$  such that the offset to a single segment  $\mathbf{r}_i(\xi)$  satisfies  $\mathbf{r}_{d,i}(\xi_l) = \mathbf{r}_{d,i}(\xi_r)$ , a (local) *interior trim* of the subsegment  $[\xi_l, \xi_r]$  is required.

*Global trimming*, on the other hand, deals with untrimmed offset segments at distance less than  $|d|$  from  $C$ , that are delineated by self-intersections with footpoints on unrelated segments of  $C$  (see Figure 2). Global offset trimming is a much more challenging and computation-intensive problem. We focus at present on developing a comprehensive solution to the local real-time offset trimming problem, and outline preliminary steps towards a solution of the global trimming problem (see Section 5.3 below). Notwithstanding its more-limited scope, the local solution offers an accurate and versatile scheme for the important task of finish-machining complex free-form planar shapes.

It should be noted that *any* trimming of the untrimmed offset reflects an inability to precisely machine the desired shape with a tool of radius  $r = |d|$ . To avoid an “overcut” (or *gouging*), the trimmed offset ensures an “undercut” — concave portions of the shape where the radius of curvature is less than  $r$  will be rounded out by circular fillet arcs of radius  $r$  on the machined part.

## 4 Local offset trimming and filling

For brevity, only local offset trimming (which is especially valuable in finish machining) is considered in this initial study, although some key requirements for the extension to global trimming are briefly discussed in Section 5.3. This

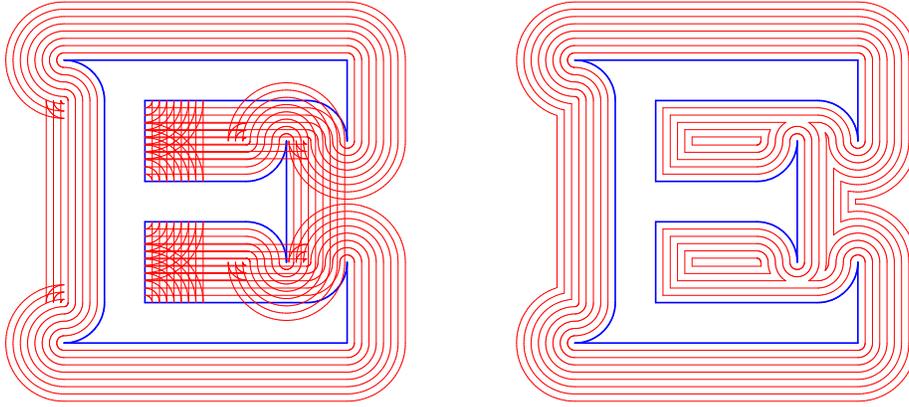


Figure 2: Left: several untrimmed offsets to a piecewise–linear/circular curve  $C$ , illustrating the need for global trimming. Right: trimmed offsets obtained by deletion of all untrimmed offset segments at global distance  $<|d|$  from  $C$ .

section describes the basic real–time interpolator, and the various functions required to implement real–time local offset curve trimming and filling.

#### 4.1 Real–time interpolator

The function of the real–time interpolator in a CNC machine is to compute a *reference point* (commanded machine position) in each sampling interval  $\delta t$  of the servo system, from prescribed path geometry and feedrate information. The reference point is compared with the actual machine location, measured by axis position encoders, to generate the instantaneous axis position errors that serve as the basic input to the control algorithm. The focus here is on piecewise–analytic paths, rather than discretized G code approximations.

The parametric speed  $\sigma(\xi) = |\mathbf{r}'(\xi)| = ds/d\xi$  of a curve  $\mathbf{r}(\xi)$  specifies the rate of change of arc length  $s$  with the parameter  $\xi$ . Since the feedrate  $V$  is the rate of change  $ds/dt$  of arc length with time, the derivative of the parameter  $\xi$  with respect to time is

$$\frac{d\xi}{dt} = \frac{ds}{dt} \frac{d\xi}{ds} = \frac{V}{\sigma}. \quad (6)$$

Thus, for each sampling interval  $\delta t$ , the reference point parameter value is

updated according to

$$\xi \leftarrow \xi + \frac{V\delta t}{\sigma(\xi)}. \quad (7)$$

The first-order approximation (7) is commonly employed [3, 16, 18, 28] and is sufficiently accurate in practice for modest feedrates  $V$  and small sampling intervals  $\delta t$ . The accuracy may be improved, if desired, by use of a higher-order Taylor series expansion [12], or analytic integration of the relation (6) in the case of Pythagorean-hodograph curves [11].

The main modification of the basic real-time interpolator algorithm, in the present context, is to omit certain portions of the  $\xi \in [0, 1]$  parameter domain when offset trimming operations become necessary. In the case of a single segment interior trim, a subset  $[\xi_l, \xi_r] \subset [0, 1]$  will be omitted. In the case of adjacent segments  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$  meeting at a concave node, the trimming amounts to using truncated parameter domains of the form<sup>3</sup>  $[0, \xi_l]$  and  $[\xi_r, 1]$ . In both cases, the trim parameter values  $\xi_l$  and  $\xi_r$  will be automatically determined in real time, to within a certain resolution.

For a segment  $\mathbf{r}_i(\xi)$  with offset  $\mathbf{r}_{d,i}(\xi)$ , two interpretations of the feedrate  $V$  in equation (7) are of interest. When  $V$  is interpreted as specifying the tool center speed along the offset path, the parametric speed  $\sigma_{d,i}(\xi) = |\mathbf{r}'_{d,i}(\xi)|$  of the offset must be used in (7). By differentiating (5), one can verify [8] that

$$\mathbf{r}'_{d,i}(\xi) = [1 + \kappa_i(\xi)d] \mathbf{r}'_i(\xi),$$

and thus  $\sigma_{d,i}(\xi) = |1 + \kappa_i(\xi)d| \sigma_i(\xi)$ . However, this incurs an uneven spacing of the tool/part contact points, and hence a variable chip load — the contact points are more closely spaced and more widely spaced in convex ( $\kappa_i(\xi)d > 0$ ) and concave ( $\kappa_i(\xi)d < 0$ ) regions of  $\mathbf{r}_i(\xi)$ , respectively.

Since varying chip loads can incur tool chatter or degrade the machined surface finish, an alternative interpretation of (7) may be preferable, in which  $V$  is viewed as the speed of the tool/part contact point. This yields a uniform chip load, and is implemented by employing the parametric speed  $\sigma_i(\xi)$  of the boundary segment  $\mathbf{r}_i(\xi)$  in (7), in lieu of the offset parametric speed  $\sigma_{d,i}(\xi)$ . The disadvantage of this approach is that, in the case of strongly-curved shapes, it incurs substantial variation of the tool center speed.

Both interpretations of the feedrate  $V$  have been implemented, and results for a variety of test curves will be presented in Section 5 below.

---

<sup>3</sup>We assume here that no trim between  $\mathbf{r}_{i-1}(\xi)$ ,  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$ ,  $\mathbf{r}_{i+2}(\xi)$  is necessary.

## 4.2 Single segment interior offset trim

An interior offset trim of a single segment  $\mathbf{r}_i(\xi)$  is required [8] whenever its curvature satisfies

$$\kappa_i(\xi) d < -1 \quad (8)$$

over a subinterval  $[\xi_l, \xi_r]$  of the parameter domain  $\xi \in [0, 1]$ . This condition indicates that the tool radius exceeds the radius of curvature of  $\mathbf{r}_i(\xi)$  over the concave segment  $\xi \in [\xi_l, \xi_r]$ , so the tool will gouge the desired shape specified by that segment. To avoid this, an “interior trim” is required, amounting to a deletion of the untrimmed offset segment  $\xi \in [\xi_l, \xi_r]$ .

An explicit curvature test is not necessary to execute the interior offset trim, since the footpoint function (see Section 2) can be used for this purpose. Before reaching  $\xi_l$ , the real-time interpolator increments the parameter  $\xi$  along the curve using (7). The footpoint function is then called with  $\mathbf{p} = \mathbf{r}_{d,i}(\xi + \delta\xi)$ , the updated offset point. Once  $\xi + \delta\xi$  has passed  $\xi_l$ , the footpoint function will detect a new closest point, with a parameter value close to  $\xi_r$ , and return it as the new footpoint parameter. Commencing with this new parameter value, the interpolation scheme (7) then resumes.

The interior parameter interval  $[\xi_l, \xi_r]$  is thus (approximately) removed from the untrimmed offset in real time. The trimming is only approximate, because the non-zero sampling time  $\delta t$  and feedrate  $V$  imply that the machine travels a finite distance  $\delta s = V\delta t$  in each sampling interval, so the reference points generated by the real-time interpolator will never exactly possess the footpoint parameter values  $\xi_l$  and  $\xi_r$ . For feedrate  $V = 100$  ipm and sampling frequency  $f = 1/\delta t = 1024$  Hz, as employed in the experiments described in Section 5 below, the maximum geometrical error of the offset trim points is

$$\epsilon = V\delta t \approx 0.0016 \text{ in.} \quad (9)$$

However, by adjusting the feedrate — e.g., by using an extrapolation method to detect an impending trim point, and reducing  $V$  in its vicinity — it is possible to identify the offset trim points as accurately as desired. In general, trimming introduces offset path tangent discontinuities, and suppressing the feedrate in their vicinity has the additional benefit of minimizing the impact of the corresponding machine velocity and acceleration discontinuities.

### 4.3 Offset trimming at concave nodes

Recall from Section 3 that *concave* and *convex* junctures between successive segments  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$  are characterized by *negative* and *positive* values for the product  $d\Delta\theta_i$  of the offset distance  $d$  and rotation angle  $\Delta\theta_i$  between their final and initial normals,  $\mathbf{n}_i(1)$  and  $\mathbf{n}_{i+1}(0)$ . To detect a concave node  $\mathbf{q}_i = \mathbf{r}_i(1) = \mathbf{r}_{i+1}(0)$ , with  $d\Delta\theta_i < 0$ , and execute the proper offset trimming it incurs, it is necessary to monitor the footpoints of the machine position  $\mathbf{p}$  on *both*  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$ , as it follows the untrimmed offset  $\mathbf{r}_{d,i}(\xi)$ .

This is accomplished by calling the footpoint and point/curve distance functions *twice*, with the instantaneous machine location  $\mathbf{p}$  and consecutive segments  $\mathbf{r}_i(\xi)$ ,  $\mathbf{r}_{i+1}(\xi)$  as arguments. The necessity of an offset trim between these segments is detected in real time when a machine location  $\mathbf{p}$  such that

$$\text{distance}(\mathbf{p}, \mathbf{r}_{i+1}(\xi)) < \text{distance}(\mathbf{p}, \mathbf{r}_i(\xi)) \quad (10)$$

arises. If  $\xi_l$ ,  $\xi_r$  are the footpoint parameter values of  $\mathbf{p}$  on  $\mathbf{r}_i(\xi)$ ,  $\mathbf{r}_{i+1}(\xi)$  when this condition arises, execution of the offset  $\mathbf{r}_{d,i}(\xi)$  is terminated at  $\xi = \xi_l$  and execution of the offset  $\mathbf{r}_{d,i+1}(\xi)$  commences at  $\xi = \xi_r$ , i.e., only the parameter domains  $\xi \in [0, \xi_l]$  and  $\xi \in [\xi_r, 0]$  of these offset paths are traversed.

As with the interior offset trim of a single segment, the offset trimming for two adjacent segments that meet at a concave node is approximate. An exact trim would require determination of the precise machine location  $\mathbf{p}$  such that the footpoint parameters  $\xi_l$  and  $\xi_r$  yield satisfaction of the condition (10) with equality, but the discrete sequence of reference points generated by the real-time interpolator for a given sampling time  $\delta t$  and feedrate  $V$  do not exactly identify this point. The value  $V\delta t$  bounds the error in the trim point location, although it is typically much smaller. For small  $V\delta t$ , the trimming accuracy is high, and the need for complicated self-intersection calculations [9] — which must be performed offline — is avoided.

### 4.4 Circular fill arcs at convex nodes

Only a convex node  $\mathbf{q}_i = \mathbf{r}_i(1) = \mathbf{r}_{i+1}(0)$  between consecutive curve segments, with  $d\Delta\theta_i > 0$ , necessitates the insertion of a circular fill arc to guarantee a continuous tool path. Let  $\theta_e$  and  $\theta_s$  denote the angular orientations<sup>4</sup> of the end and start point normals,  $\mathbf{n}_i(1)$  and  $\mathbf{n}_{i+1}(0)$ , of the segments incident at a

---

<sup>4</sup>Here  $\theta_e, \theta_s \in [0, 2\pi)$  and  $\theta_s > \theta_e$  or  $\theta_s < \theta_e$  according to whether  $d > 0$  or  $d < 0$ .

convex node  $\mathbf{q}_i$ . On completion of segment  $\mathbf{r}_i(\xi)$ , satisfaction of the condition

$$d [\mathbf{n}_i(1) \times \mathbf{n}_{i+1}(0)] \cdot \mathbf{z} \geq 0 \quad (11)$$

indicates that a fill arc with center  $\mathbf{q}_i$  and radius  $|d|$ , between the angular limits  $\theta_e$  and  $\theta_s$ , must be executed. This arc is of length  $S = |d \Delta\theta_i|$ , where  $\Delta\theta_i = \theta_s - \theta_e$ , and the distance travelled along it during one sampling interval  $\delta t$  at feedrate  $V$  is  $\delta s = V\delta t$ , corresponding to a (signed) angular increment  $\delta\theta = V\delta t/d$  along the fill arc. Thus, the real-time interpolator generates the reference point

$$\mathbf{p} = \mathbf{q}_i + |d| (\cos \theta, \sin \theta), \quad \theta = \theta_e + k\delta\theta \quad (12)$$

at time  $t = k\delta t$  since commencement of the fill arc. Execution of the fill arc terminates when  $\theta_e + k\delta\theta$  becomes greater than or less than  $\theta_s$ , according to whether  $d > 0$  or  $d < 0$ . In general, the total execution time  $T = |d \Delta\theta_i|/V$  for the circular fill arc is not an exact integer multiple of the sampling interval  $\delta t$ , and this incurs a jump in feedrate during the final sampling interval. This can be mitigated by slightly perturbing the feedrate along the circular fill arc from the nominal value  $V$  to  $\tilde{V} = |d \Delta\theta_i|/N\delta t$ , where  $N = \lfloor |\Delta\theta_i/\delta\theta| \rfloor$ . Once traversal of the circular fill arc has been completed, the real-time interpolator can proceed to execute the offset to segment  $\mathbf{r}_{i+1}(\xi)$ .

## 4.5 Algorithm outline

The following outline summarizes the real-time offset curve trimming/filling algorithm. The input is a set of polynomial curve segments  $\mathbf{r}_i(\xi)$ ,  $i = 1, \dots, N$  with parameter domain  $\xi \in [0, 1]$  that meet end-to-end to form a simple closed contour  $C$  with anticlockwise orientation, and the offset distance  $d$ . Note that the segment index  $i$  is always reduced modulo  $N$ , i.e.,  $i+1$  becomes 1 when  $i = N$ . The algorithm employs the basic functions  $\text{footpoint}(\mathbf{p}, \mathbf{r}_i(\xi))$  and  $\text{distance}(\mathbf{p}, \mathbf{r}_i(\xi))$ , described in Section 2. For brevity, the outline below considers only the case where no adjacent offset trimming between the final and initial segments,  $i = N$  and  $i = 1$ , is required — this case can be easily accommodated through some technical modifications.

1. initialize: set  $i \leftarrow 1$ ,  $\xi \leftarrow 0$
2. while (  $\xi < 1$  )
  - {

```

increment  $\xi$  using (7), set  $\mathbf{p} \leftarrow \mathbf{r}_{d,i}(\xi)$ 

// interior segment trimming
set  $\xi_r = \text{footpoint}(\mathbf{p}, \mathbf{r}_i(\xi))$ 
if (  $\xi_r > \xi$  ) set  $\xi \leftarrow \xi_r$  and  $\mathbf{p} \leftarrow \mathbf{r}_{d,i}(\xi)$ 

// adjacent segment trimming
set  $\xi_r = \text{footpoint}(\mathbf{p}, \mathbf{r}_{i+1}(\xi))$ 
if ( distance( $\mathbf{p}, \mathbf{r}_{i+1}(\xi)$ ) < distance( $\mathbf{p}, \mathbf{r}_i(\xi)$ ) )
    { set  $i \leftarrow i + 1$ ,  $\xi \leftarrow \xi_r$ ,  $\mathbf{p} \leftarrow \mathbf{r}_{d,i}(\xi)$  }
}

```

```

3. if ( condition (11) is satisfied )
    {
        // execute circular fill arc
        compute angular limits  $\theta_e, \theta_s$ 
        set  $\theta \leftarrow \theta_e$  and  $\delta\theta \leftarrow V\delta t/d$ 
        while ( sign( $d(\theta - \theta_e)$ ) > 0 )
            { compute reference points  $\mathbf{p}$  using (12) }
        if (  $i < N$  )
            { set  $i \leftarrow i + 1$ ,  $\xi \leftarrow 0$ , go to 2 }
        else
            { go to 4 }
    }
else
    {
        // proceed to next segment
        set  $i \leftarrow i + 1$ ,  $\xi \leftarrow 0$ , go to 2
    }
}

```

4. stop

## 4.6 Kinematical considerations

If a path is executed at constant feedrate (speed), the local path smoothness will determine the continuity of velocity and acceleration along it. An interior offset trim of a single segment, or an offset trim between adjacent segments associated with a concave node, generically incurs a point with discontinuous

tangent  $\mathbf{t}$  and normal  $\mathbf{n}$  on the trimmed offset. At the junctures of a circular fill arc with the offsets to two boundary segments  $\mathbf{r}_i(\xi)$  and  $\mathbf{r}_{i+1}(\xi)$  that meet at a convex node, the tangent and normal are continuous but in general the curvature  $\kappa$  is not continuous. For a constant feedrate  $V$ , the velocity  $\mathbf{v}$  and acceleration  $\mathbf{a}$  along a curved path are defined by

$$\mathbf{v} = V\mathbf{t} \quad \text{and} \quad \mathbf{a} = \kappa V^2\mathbf{n}.$$

Thus, velocity and acceleration are both discontinuous at offset trim points. When moving from a curve offset segment to a circular fill arc (or vice-versa), on the other hand, velocity is continuous but acceleration is discontinuous.

Commanded motions involving discontinuities in velocity or acceleration are obviously inconsistent with smooth execution, especially at high speeds, and may incur machine vibration or significant path contour error. Since the offset path geometry is fixed by the requirement of a continuous gouge-free tool path, the only way to mitigate the effects of the discontinuities — which are of special concern in high-speed machining [17, 24, 27] — is to modulate the feedrate in accordance with the path geometry. However, algorithms for real-time feedrate modulation to accommodate impending offset trim or fill actions must employ a look-ahead strategy, and are not trivial to formulate. This aspect of the problem is deferred to a future study — at present, we consider only modest constant feedrates.

## 5 Implementation and experimental results

To establish its practical feasibility, the real-time local offset trimming/filling algorithm was implemented on a 3-axis CNC mill, and run on a selection of piecewise-polynomial test curves. The implementation details and real-time performance results from the test runs are described below.

### 5.1 Open-architecture software controller

Figure 3 shows the table-top CNC milling machine used in the experiments. The machine is governed by the MDSI OpenCNC open-architecture software controller, that allows incorporation of customized real-time motion control algorithms. The controller runs on a commercial PC with a modest 500 MHz CPU and employs a sampling frequency  $f = 1024$  Hz, which corresponds to a sampling interval  $\delta t = 1/f \approx 0.001$  seconds.

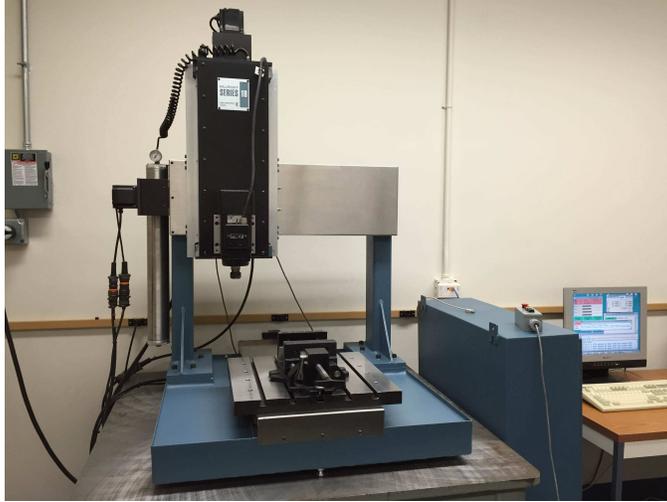


Figure 3: 3-axis CNC mill governed by the OpenCNC software controller.

The required software modifications were accomplished entirely within the real-time interpolator module, the function that generates a reference point (commanded machine position) in each sampling interval from the specified path geometry and feedrate. During each test run, real-time position encoder data are stored in memory for an offline analysis of the machine performance. Real-time axis velocities and accelerations are obtained by first- and second-order differencing of the position data, and the feedrate along the path is then determined as the magnitude of the velocity  $\mathbf{v}$  with components specified by the individual-axis velocities  $(v_x, v_y)$ . The acceleration magnitude is likewise obtained from the acceleration vector  $\mathbf{a} = (a_x, a_y)$ . From the position encoder data and the known exact path geometry, it is possible to compute the actual real-time machine contour error — i.e., the normal deviation of the actual machine position from the commanded path.

## 5.2 Experimental results from test curves

For the test curves described below, the indicated dimensions are in inches, and in each case an offset distance  $d = 1$  in and nominal feedrate  $V = 100$  ipm is specified. These test curves employ Pythagorean-hodograph (PH) quintic segments [6], but the algorithm can accommodate general polynomial curves. During each run, the machine real-time position encoder data is recorded in

memory, for subsequent analysis to assess: (1) correct offset curve trimming and filling; (2) contour error (i.e., path accuracy); (3) feedrate variation; and (4) acceleration magnitude.

**Example 1** As seen in Figure 4, the first example involves a path comprising three smooth segments with tangent-discontinuous junctures. Since the path tangent suffers a complete reversal at these nodes, semi-circular fill arcs are required to ensure a continuous offset path. However, no interior or adjacent segment trimming is needed in this case. Figure 4 compares the “raw” offset (without filling or trimming) and the “true” offset, plotted from the real-time position encoder data generated by the offset trimming/filling algorithm.

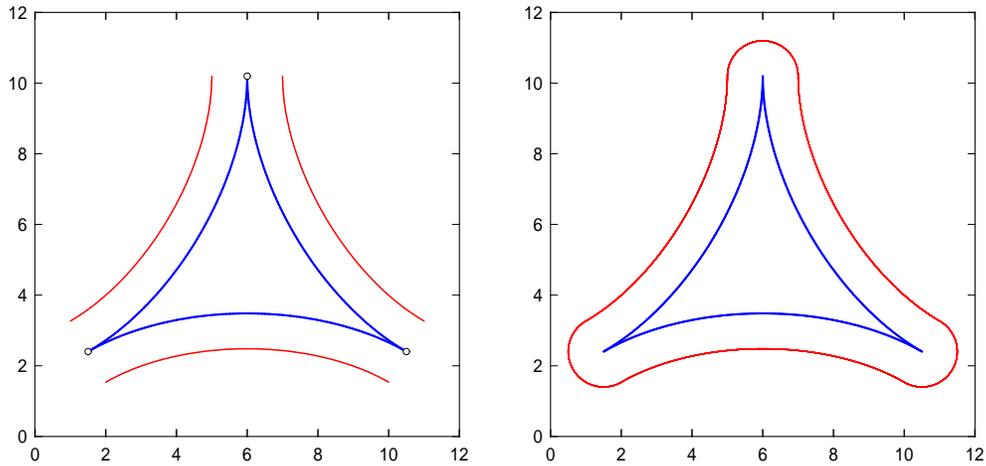


Figure 4: The raw offset (left) and true offset (right), computed by the real-time interpolator algorithm, for a case involving only the insertion of circular fill arcs at each convex tangent-discontinuous point of the part geometry.

Figure 5 illustrates the actual tool center speed obtained from the real-time position encoder data<sup>5</sup> for a constant 100 ipm specified speed of both the tool center and the tool/part contact point. It is seen that the real-time interpolator performs very well in the case of a constant 100 ipm speed of the tool center. For the case of a constant speed of the tool/part contact point, the tool center speed is reduced below the nominal 100 ipm value along the

<sup>5</sup>All velocity and acceleration plots shown herein are based on directly differencing the position encoder data: no smoothing filters have been applied.

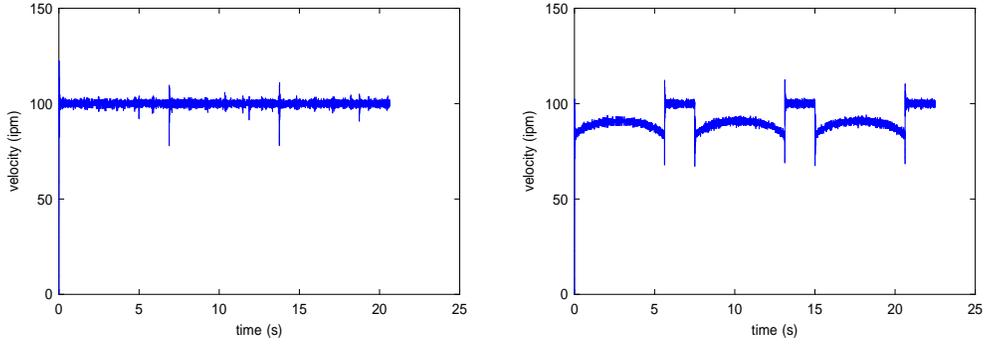


Figure 5: Actual tool speed, computed from real-time position encoder data along the path shown in Figure 4, for (left) a constant speed of the tool center on the offset path, and (right) a constant speed of the tool/part contact point.

offsets to the concave segment boundaries, but maintains the 100 ipm value along the circular fill arcs (on these arcs, however the tool/part contact point is actually stationary rather than moving at constant speed).

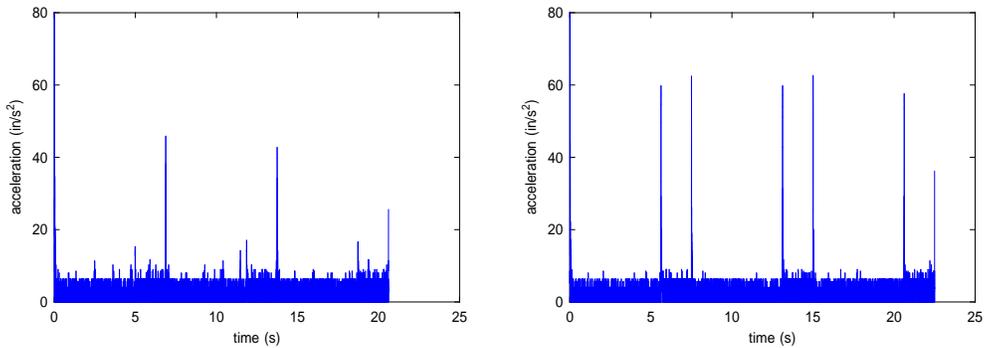


Figure 6: Actual tool acceleration, computed from real-time position encoder data along the path in Figure 4, for (left) a constant speed of the tool center on the offset path, and (right) constant speed of the tool/part contact point.

Analogous plots for the tool acceleration magnitude are shown in Figure 6. The acceleration spikes visible in these plots are attributable to the curvature discontinuities between the boundary segment offsets and the circular fill arcs.

**Example 2** The second test curve involves eight segments with concave radii

of curvature smaller than the tool radius, and convex tangent-discontinuous nodes, necessitating both interior trimming and filling operations. Figure 7 shows the “raw” and “true” offsets in this case. The actual variation of the tool center speed, for both types of specified feedrate, is shown in Figure 8. Apart from the spikes in the measured feedrate, resulting from the trimmed offset tangent discontinuities, the measured feedrate conforms closely to the prescribed 100 ipm value for both feedrate types. The similarity of the plots in this case results from the mild curvature of the trimmed offset paths. As can be seen in Figure 9, the tangent-discontinuous offset trim points incur quite severe spikes in the measured acceleration magnitude.

The measured contour error, plotted in Figure 10, is seen to be similar for the two feedrate variations, and its overall magnitude is consistent with the bound (9). In fact, the root-mean-square contour error over the entire path is comparable to (9), so positional errors incurred by the finite resolution of the real-time offset trimming algorithm are not especially pronounced.

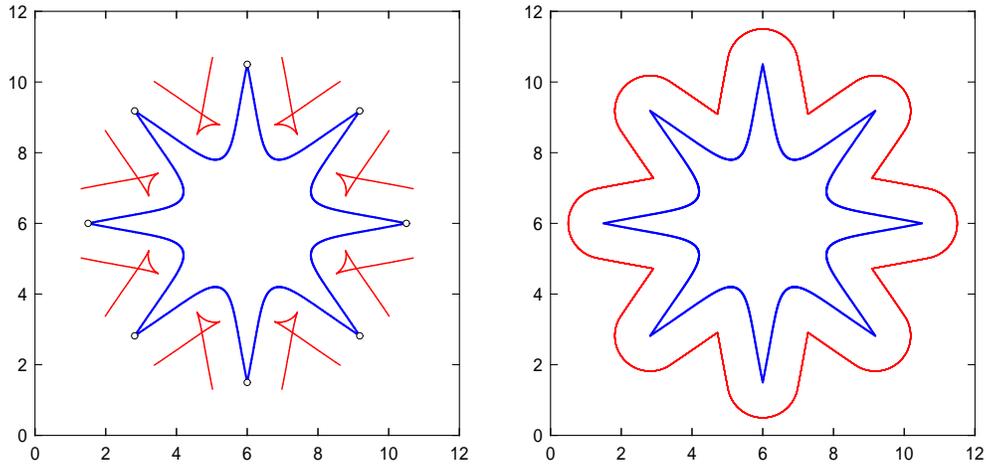


Figure 7: The raw offset (left) and true offset (right), computed by the real-time interpolator, for a case that requires both insertion of circular fill arcs and internal segment trimming due to violation of the curvature bound (8).

**Example 3** The third test curve comprises eight segments, and necessitates use of all three of the basic trim/fill functions: interior trimming of individual segments; trimming at concave nodes of adjacent segments; and filling at the

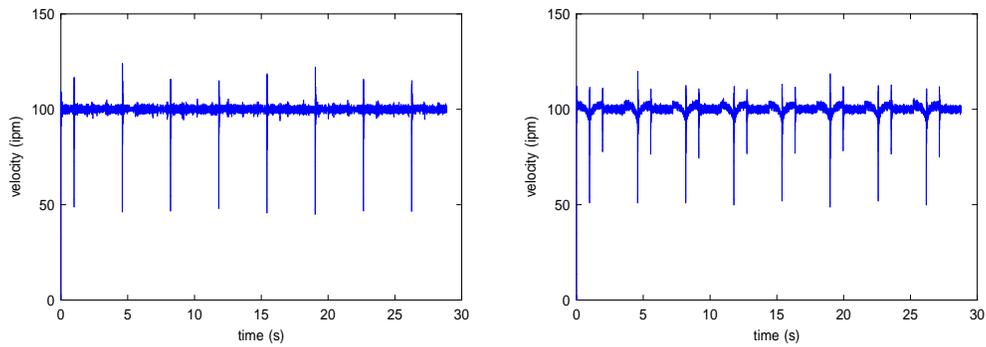


Figure 8: Actual tool speed, computed from real-time position encoder data along the path shown in Figure 7, for (left) a constant speed of the tool center on the offset path, and (right) a constant speed of the tool/part contact point.

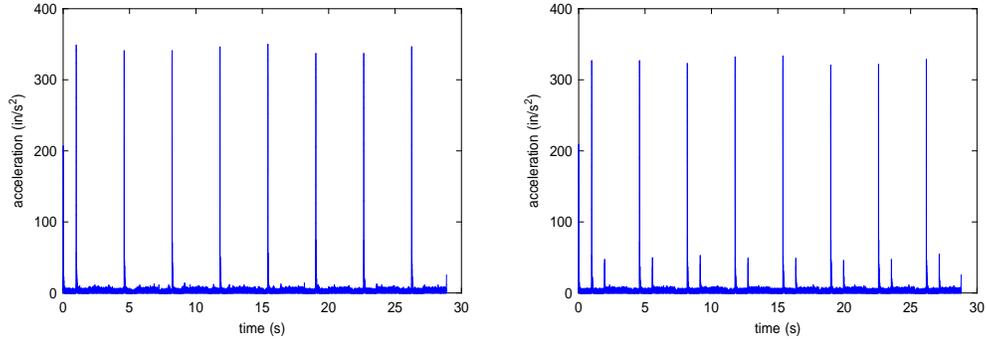


Figure 9: Actual tool acceleration, computed from real-time position encoder data along the path in Figure 7, for (left) a constant speed of the tool center on the offset path, and (right) constant speed of the tool/part contact point.

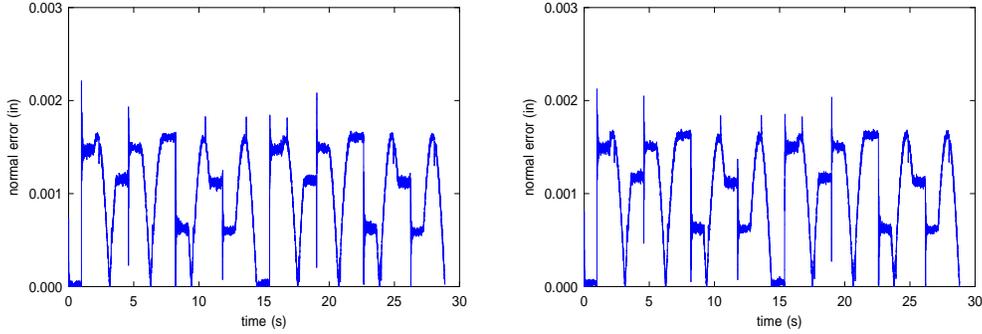


Figure 10: Contour error along the path in Figure 7 for (left) a fixed speed of the tool center on the offset path, and (right) of the tool/part contact point.

convex nodes of adjacent segments. The “raw” and “true” offsets in this case are illustrated in Figure 11 — it is seen that the trimmed offset involves strong convex and concave curvatures. Consequently, there is a marked difference between the observed behavior of the two feedrate types, shown in Figure 12. The real-time interpolator performs very well in maintaining the 100 ipm tool center speed, when this type of feedrate is specified. However, specifying a constant speed of the tool/part contact point incurs pronounced modulation of the tool center speed, above and below the nominal 100 ipm value, due to the strong curvature variation of the part shape. As seen in Figure 13, both feedrate types incur prominent spikes in the acceleration magnitude, due to the tangent-discontinuous interior and adjacent segment offset trim points. However, both types yield similar results (Figure 14) for the measured contour error, whose magnitude is again consistent with the bound (9).

### 5.3 Extension to global trimming

Although the real-time computational load imposed by the local offset curve trimming/filling procedure is not insignificant, no difficulties were observed in running it on the modest (500 MHz) control computer processor. Inability of the computer to complete the necessary computations within each sampling interval would cause an unacceptable following error to accumulate, resulting in automatic shutdown of the servo system, but this problem never occurred in practice. In extending the method to accommodate *global* real-time offset curve trimming/filling, however, the computational load will be substantially

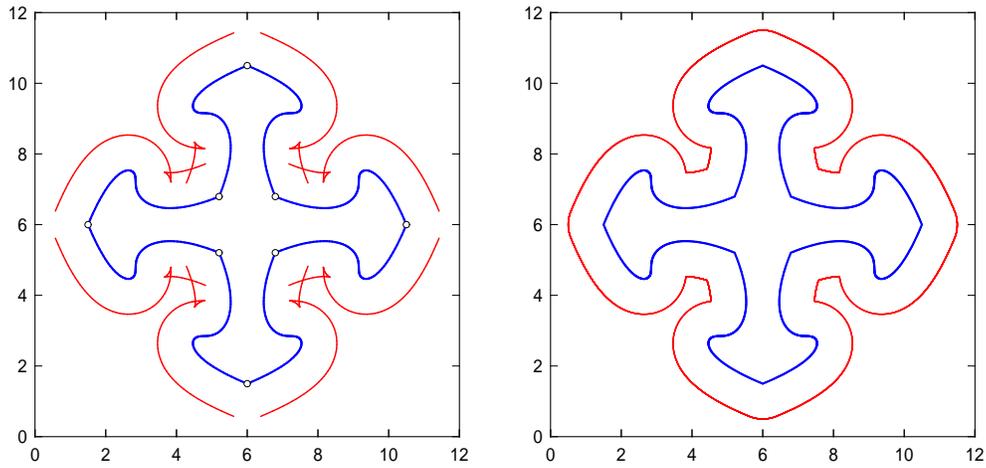


Figure 11: The raw offset (left) and true offset (right) for a case involving all three situations — insertion of circular fill arcs; internal segment trimming; and trimming at the concave nodes between pairs of adjacent segments.

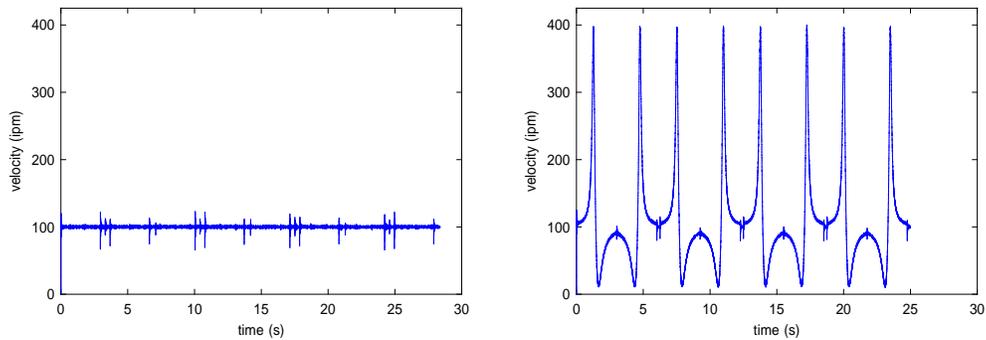


Figure 12: Actual tool speed, computed from real-time position encoder data along the path in Figure 11, for (left) a constant speed of the tool center on the offset path, and (right) constant speed of the tool/part contact point.

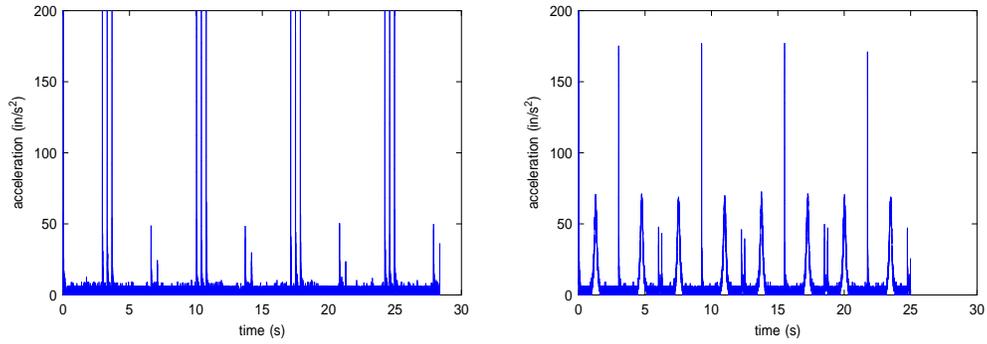


Figure 13: Actual tool acceleration, determined from position encoder data along the path in Figure 11, for (left) a constant speed of the tool center on the offset path, and (right) a constant speed of the tool/part contact point.

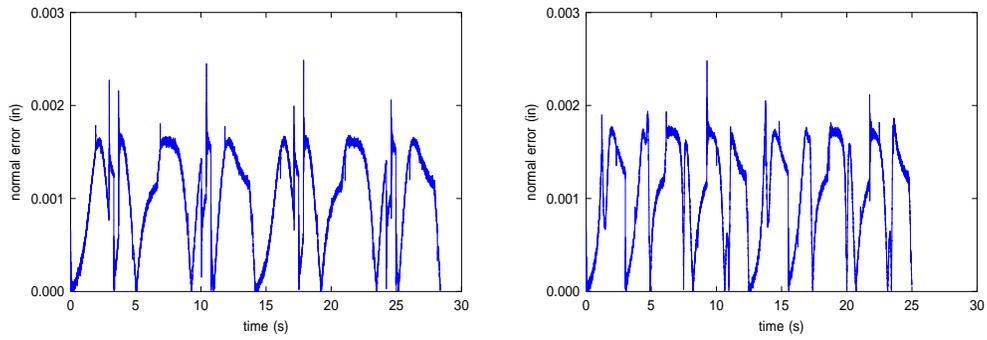


Figure 14: Contour error on the path in Figure 11 for a fixed speed of (left) the tool center along the offset path, and (right) the tool/part contact point.

increased, and the formulation of a viable real-time algorithm will necessitate careful attention to considerations of computational efficiency.

A global algorithm must incorporate trimming operations incurred by the interference of untrimmed offsets to general pairs of boundary segments, not just the interior trimming of a single segment, or interference due to adjacent boundary segments. Furthermore, topological complexities may arise from the fact that the offset to a single closed contour  $C$  can consist of several disjoint closed loops, requiring tool retraction and re-positioning procedures to ensure complete traversal. A standard approach to global offset trimming is through pre-computation of the *Voronoi diagram* or *medial axis* of the domain bounded by the contour  $C$ . In general, this is a non-trivial problem, incompatible with real-time implementation. Although an exact solution is possible [14] for a piecewise-linear contour  $C$ , approximations are necessary for contours composed of free-form curve segments [22, 23].

In principle, global offset trimming may be accomplished by calling the point/curve distance function with every boundary segment  $\mathbf{r}_i(\xi)$  in each sampling interval. However, a more efficient approach is needed for a viable real-time implementation, that will also faithfully capture the topology of the trimmed offset. A more thorough treatment of these issues is deferred to a subsequent study. At present, we only note that the convex hull property [5] of the Bézier form of polynomial curves can be used to “triage” the segments of  $C$ , in assessing likelihood of interference of their untrimmed offsets with that of a given segment, for global offset trimming operations.

## 6 Conclusion

An algorithm has been developed for the real-time generation and execution of continuous, gouge-free paths for machining planar shapes with piecewise-analytic free-form boundary curves using a tool of given radius. The method automatically executes the necessary offset path trimming/filling operations in real time, and its practical feasibility has been verified by implementation on a milling machine with an open-architecture controller, yielding excellent results. The algorithm offers many useful benefits — including (1) the ability to accommodate different tool sizes without re-generating the part program; (2) the elimination of errors associated with approximating the offsets to free-form curves; (3) the ability to modulate the feedrate to achieve a constant speed for either the tool center, or the tool/part contact point; and (4) the

possibility of utilizing analytic geometry data (e.g., path curvature) so as to achieve smoother motions through acceleration management.

The method described herein is intended for use on parts with boundaries defined by a reasonable number of extended analytic curve segments, rather than voluminous discretized approximations. As currently implemented, the focus is on *local* trimming, in which all required offset trim operations arise from concave high-curvature interior portions of single segments, or concave tangent-discontinuous nodes between adjacent segments. This is appropriate for finish machining, and also roughing with contour tool paths for parts of reasonable geometrical complexity. The extension to *global* offset trimming to accommodate trim operations incurred by general boundary segment pairs is more challenging, since computational complexity becomes a key concern.

Another problem that deserves further investigation is the use of feedrate modulation to mitigate the velocity and acceleration discontinuities arising at offset trim points when a constant feedrate is specified along a trimmed offset (this is important to guarantee smooth motions and suppress contour error). Detailed investigations of these problems are deferred to future studies.

## References

- [1] J. G. Campos and M. Hardwick (2006), A traceability information model for CNC manufacturing, *Comput. Aided Design* **38**, 540–551.
- [2] C-S. Chiang, C. M. Hoffmann, and R. E. Lynch (1991), How to compute offsets without self-intersections, SPIE Proceedings Vol. 1610: Curves and Surfaces in Computer Vision and Graphics II, 76–87.
- [3] J.-J. Chou and D. C. H. Yang (1991), Command generation for three-axis CNC machining, *ASME J. Eng. Indus.* **113**, 305–310.
- [4] J. R. Conway, C. A. Ernesto, R. T. Farouki, and M. Zhang (2012), Performance analysis of cross-coupled controllers for CNC machines based upon precise real-time contour error measurement, *Inter. J. Mach. Tools Manuf.* **52**, 30–39.
- [5] G. Farin (1997), *Curves and Surfaces for Computer Aided Geometric Design*, 4th edition, Academic Press, San Diego.

- [6] R. T. Farouki (2008), *Pythagorean–Hodograph Curves: Algebra and Geometry Inseparable*, Springer, Berlin.
- [7] R. T. Farouki (2012), The Bernstein polynomial basis: a centennial retrospective, *Comput. Aided Geom. Design* **29**, 379–419.
- [8] R. T. Farouki and C. A. Neff (1990), Analytic properties of plane offset curves, *Comput. Aided Geom. Design* **7**, 83–99.
- [9] R. T. Farouki and C. A. Neff (1990), Algebraic properties of plane offset curves, *Comput. Aided Geom. Design* **7**, 101–127.
- [10] R. T. Farouki and R. Ramamurthy (1998), Degenerate point/curve and curve/curve bisectors arising in medial axis computations for planar domains with curved boundaries, *Comput. Aided Geom. Design* **15**, 615–635.
- [11] R. T. Farouki and S. Shah (1996), Real–time CNC interpolators for Pythagorean–hodograph curves, *Comput. Aided Geom. Design* **13**, 583–600.
- [12] R. T. Farouki and Y–F. Tsai (2001), Exact Taylor series coefficients for variable–feedrate CNC curve interpolators, *Comput. Aided Design* **33**, 155–165.
- [13] M. Hardwick and D. Loffredo (2006), Lessons learned implementing STEP–NC AP–238, *Int. J. Comput. Integ. Manuf.* **19**, 523–532.
- [14] M. Held (1991), *On the Computational Geometry of Pocket Machining*, Springer, Berlin.
- [15] P. Henrici (1964), *Elements of Numerical Analysis*, Wiley, New York.
- [16] J.–T. Huang and D. C. H. Yang (1992), A generalized interpolator for command generation of parametric curves in computer–controlled machines, *Proc. Japan/USA Symposium on Flexible Automation*, Vol. 1, ASME, 393–399.
- [17] R. Komanduri, K. Subramanian, and B. F. von Turkovich (eds.) (1984), *High Speed Machining*, PED–Vol. 12, ASME, New York.

- [18] R-S. Lin and Y. Koren (1996), Real-time interpolators for multi-axis CNC machine tools, *Manuf. Syst.* **25**, 145–149.
- [19] K. M. Nittler and R. T. Farouki (2016), A real-time surface interpolator methodology for precision CNC machining of swept surfaces, *Int. J. Adv. Manuf. Technol.* **83**, 561–574.
- [20] S. L. Omirou (2004), A locus tracing algorithm for cutter offsetting in CNC machining, *Robotics. Comput. Integ. Manuf.* **20**, 49–55.
- [21] S. Osher and J. A. Sethian (1988), Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12–49.
- [22] R. Ramamurthy and R. T. Farouki (1999), Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. Theoretical foundations, *J. Comput. Appl. Math.* **102**, 119–141.
- [23] R. Ramamurthy and R. T. Farouki (1999), Voronoi diagram and medial axis algorithm for planar domains with curved boundaries II. Detailed algorithm description, *J. Comput. Appl. Math.* **102**, 253–277.
- [24] S. Smith and J. Tlustý (1997), Current trends in high-speed machining, *ASME J. Manuf. Sci. Eng.* **119**, 664–666.
- [25] D. J. Struik (1961), *Lectures on Classical Differential Geometry*, Dover Publications (reprint), New York.
- [26] S-H. Suh, J-H. Cho, and H-D. Hong (2002), On the architecture of intelligent STEP-compliant CNC, *Int. J. Comput. Integ. Manuf.* **15**, 168–177.
- [27] J. Tlustý (1993), High-speed machining, *CIRP Ann.* **42**, 733–738.
- [28] D. C. H. Yang and T. Kong (1994), Parametric interpolator versus linear interpolator for precision CNC machining, *Comput. Aided Design* **26**, 225–234.