Pergamon International Journal of Machine Tools & Manufacture 39 (1999) 123–142

INTERNATIONAL JOURNAL OF
MACHINE TOOLS
& MANUFACTURE
DESIGN, RESEARCH AND APPLICATION

G codes for the specification of Pythagorean-hodograph tool paths and associated feedrate functions on open-architecture CNC machines

Rida T. Farouki*, Jairam Manjunathaiah, Guo-Feng Yuan

Department of Mechanical Engineering and Applied Mechanics, University of Michigan, Ann Arbor, MI 48109, USA

Received 1 June 1997; in final form 1 December 1997

Abstract

A new class of machine codes for the specification of Pythagorean-hodograph (PH) curve tool paths, and associated feedrate functions, is proposed. The PH curves are a special family of free-form curves, compatible with the Bézier/B-spline representations of CAD systems, that are amenable to real-time interpolation at constant or variable feedrate directly from their exact analytic descriptions. The proposed codes are compatible with 'ordinary' (linear/circular) G codes, and may be regarded as extensions thereof. Compared to the customary approach of using piecewise-linear/circular approximations, they offer significant improvements in the accuracy, efficiency, data volume, and flexibility of part programs for free-form shapes. Experimental results from an implementation of these real-time PH curve interpolators on an 'open-architecture' CNC milling machine are also described. © 1998 Elsevier Science Ltd. All rights reserved.

Keywords: G codes; Pythagorean-hodograph curves; CNC interpolators; Feedrates

1. Introduction

G codes [23] have been used extensively in CNC part programming for many years, and suffice for precision machining of basic geometries. A fundamental limitation of their current embodiment, however, is that they only admit the interpolation of discrete tool positions along linear and circular—and, less frequently, parabolic—path segments. Moreover, these 'reference points' correspond to uniform time-sampling of the tool path at *fixed speed* (feedrate) along each segment.

* Corresponding author. E-mail: farouki@engin.umich.edu

PII: S0890-6955(98)00018-2

This means that complex free-form shapes, meticulously designed in a CAD system, are necessarily subject to rather crude and data-intensive approximations before they are machined; and feedrate variations (which may be required to control the cutting force, part accuracy, or surface finish) must always be specified in a *discontinuous* manner.

Thus, compared to the hardware capabilities of modern CNC systems, the currently-accepted method of communicating the geometrical and temporal information required to machine complex shapes offers scope for significant improvement. Our present intent is to show that a new approach to designing free-form shapes—based on *Pythagorean-hodograph* (PH) *curves* [6,16]—allows such improvements to be fully realized. The PH curves incorporate a special algebraic structure that greatly facilitates the problem of real-time CNC interpolation directly from their analytic definitions. Furthermore, this can be accomplished not only for constant feedrate, but also various feedrate dependencies on arc length or curvature [18] of practical interest.

Since they are entirely compatible with the Bézier/B-spline representation of existing CAD systems, geometrically intuitive methods for designing with PH curves can be provided in such systems by incorporating the appropriate software functions and graphical user interface. The CAD system then writes out part programs that *exactly* specify the desired free-form tool paths, and corresponding feedrate functions, according to conventions described in Section 4 below. These G code conventions for PH curves are designed for compatibility with existing standards [1] to allow maximum flexibility in part programming, and to encourage their adoption within the user community.

An extensive theory and collection of algorithms for PH curves has been developed [2,6–10,14,16], including the formulation and verification of real-time CNC interpolators [11,12,18]. These studies establish PH curves as a viable medium for practical design and manufacturing applications. We shall not delve into the mathematical details here—our purpose is rather to give, from the part programmer's perspective, a 'high-level' review of the unique capabilities and advantages of PH curves in CNC applications.

Our plan in this paper is as follows. In Section 2 we review some basic concepts of G code part programming, as currently defined for piecewise-linear/circular tool paths. The distinguishing properties of PH curves are then summarized in Section 3, together with a synopsis of available methods for their construction and the capabilities of PH-curve CNC interpolators. G code extensions that describe PH curve tool paths and associated feedrate functions are proposed in Section 4, and sample part programs illustrating their usage can be found in Section 5. Some experimental results from an implementation on an open-architecture CNC milling machine are given in Section 6, illustrating the realization of various feedrate forms, the control of machining forces by feedrate variation, and the improved surface finish obtained using PH-curve interpolators. Finally, Section 7 summarizes the advantages of PH curves in CNC machining, and identifies important issues for further research.

2. Traditional G codes

In the G code programming language, machining commands are composed of characters and numerical values built up into words and blocks. A 'word' comprises an address character followed by a numerical value, and a 'block' is a collection of words on a single line. A part program contains an ordered sequence of blocks that describe in detail the motions and ancillary functions a CNC machine must execute in order to fabricate a specified part. The part program is often generated 'automatically' by a CAD/CAM system, once the part geometry has been completely defined.

Although different G code 'dialects' exist, the typical syntax of a block consists of (a subset of) certain standard words, ordered according to their address characters in the following manner:

N G X Y Z U V W I J K A B C F S T M.

The numerical values following these characters are interpreted as follows:

Character	Address for:
N	sequence number
G	preparatory function
X, Y, Z	primary motion dimensions
U, V, W	secondary motion dimensions parallel to X, Y, Z
I, J, K	offsets to circle center in X, Y, Z directions
A, B, C	angular dimensions about X, Y, Z
F	feedrate
S	spindle speed
T	tool selection
M	miscellaneous (machine) function

Other, more specialized, addresses are also defined [1] but they do not concern us here. Elementary G code blocks specify linear and clockwise/anticlockwise circular motions, through the G01 and G02/G03 words, respectively. Thus,

```
N10 G01 X500 Y0 F100
```

specifies linear motion from the current position, (0,0) say, to position (500,0) at a constant feedrate of 100 (in suitable units). This might be followed by, say, an anticlockwise circular motion about (0,0) to (300, 400):

```
N15 G03 X300 Y400 I500 J0 F100.
```

Here I and J specify offsets (which are, by convention, non-negative values) from the start point to the circle center. It is common practice to approximate general free-form tool paths by sequences of such 'simple' motions.

3. Pythagorean-hodograph curves

Modern CAD systems offer great flexibility in the design of free-form shapes, typically through interpolation/approximation of discrete data or by means of 'control polygon' manipulations. Typically, these shapes are created and stored internally using the Bézier or B-spline format [5]. Although they are well-suited to design needs, these forms are much less convenient for use in a manufacturing context. Two basic limitations that preclude driving CNC machines directly from Bézier or B-spline curve definitions are:

1. their offsets, which describe centerline paths for finite-radius tools, are not *rational* loci, and must be approximated in the CAD system [13];
2. whereas the feedrate is the time derivative of arc length along a curve, the arc lengths of general Bézier/B-spline curves are integrals with no analytic reduction—they must be computed by numerical quadrature.

Pythagorean-hodograph (PH) curves were introduced [16] as a means of circumventing these basic limitations, and thus providing a more direct link between design and manufacturing of complex shapes. They are guaranteed, by their manner of construction, to have rational offset curves and arc lengths that are simple (polynomial) functions of the curve parameter.

A plane parametric curve $\mathbf{r}(\xi) = (x(\xi), y(\xi))$ for $\xi \in [0,1]$ is a PH curve if there exist polynomials $u(\xi)$ and $v(\xi)$ such that its derivative or 'hodograph' $\mathbf{r}'(\xi) = (x'(\xi), y'(\xi))$ satisfies[1]

$$x'(\xi) = u^2(\xi) - v^2(\xi),\ y'(\xi) = 2u(\xi)v(\xi),\ \sigma(\xi) = u^2(\xi) + v^2(\xi) \tag{1}$$

where $\sigma(\xi) = \sqrt{x'^2(\xi) + y'^2(\xi)}$ is the 'parametric speed'—the rate of change of arc length s with respect to the parameter ξ. The polynomials (1) form a *Pythagorean triple*, satisfying $x'^2(\xi) + y'^2(\xi) \equiv \sigma^2(\xi)$—this is the source of the unique advantages of PH curves over 'ordinary' polynomial curves.

A PH curve $\mathbf{r}(\xi)$ for $\xi \in [0,1]$ is defined by either a start or end point and two polynomials, $u(\xi)$ and $v(\xi)$. For numerical stability, it is preferable [15] that the latter be specified by their Bernstein coefficients—for PH quintics, they would be quadratics written in the form

$$\begin{aligned} u(\xi) &= u_0(1-\xi)^2 + u_1 2(1-\xi)\xi + u_2\xi^2, \\ v(\xi) &= v_0(1-\xi)^2 + v_1 2(1-\xi)\xi + v_2\xi^2. \end{aligned} \tag{2}$$

3.1. Construction procedures

Since the PH curves are actually 'special' polynomial curves, they can always be represented in Bézier form, and composite PH curves can be expressed in B-spline form—this ensures their compatibility with existing CAD systems. By using somewhat higher degrees, PH curves may be endowed with all the 'shape freedoms' we are accustomed to when designing with ordinary Bézier or B-spline curves. Although algorithms for constructing PH curves are more involved, the

[1] For brevity we focus on planar PH curves here; see [17] for details on space curves.

designer need not be exposed to their inner workings, and can be presented with intuitive 'shape handles' to communicate with them.

A number of PH curve construction methods, serving various geometrical design purposes, have already been developed—they include:

- *First-order Hermite interpolants* [14]—these are PH quintics matching given end points and derivatives: $\mathbf{r}(0)$, $\mathbf{r}'(0)$ and $\mathbf{r}(1)$, $\mathbf{r}'(1)$. Since such data defines a unique 'ordinary' cubic, the latter may be replaced on a one-for-one basis by PH quintics (for composite curves this maintains tangent, though not curvature, continuity)—see Fig. 1.
- *Second-order Hermite interpolants* [10]—to match specified end points with first and second derivatives, $\mathbf{r}(0)$, $\mathbf{r}'(0)$, $\mathbf{r}''(0)$ and $\mathbf{r}(1)$, $\mathbf{r}'(1)$, $\mathbf{r}''(1)$, degree-9 PH curves are required. These are appropriate for specialized applications, such as cam design (see Section 5.2 and Fig. 4 below).
- *C^2 interpolating splines* [2] are piecewise PH quintics that interpolate an ordered sequence $\mathbf{p}_0,\ldots, \mathbf{p}_N$ of points with second-order continuity and match specified end derivatives $\mathbf{d}_0$ and $\mathbf{d}_N$ (or, alternately, satisfy prescribed spline end conditions)—see Fig. 2.
- *Monotone-curvature segments* [9,32], for curvature-continuous splices of line and circle segments, can be constructed using the PH quintics.

There are doubtless other interesting methods to be explored. A remarkable serendipitous property of PH curves constructed by interpolating discrete data is that they yield 'fairer' loci, with better curvature distributions, than loci constructed using ordinary polynomial curves [8]—see Figs. 1 and 2.

3.2. CNC interpolators

CNC machines employ digital spatial/temporal representations in executing a programmed trajectory. The unit of time is called the *sampling interval*, while spatial coordinates are measured

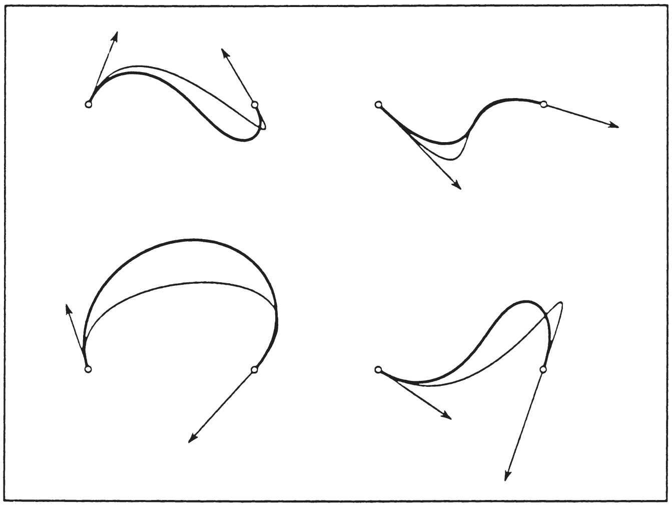

Fig. 1. PH quintic (bold) and 'ordinary' cubic (light) interpolants to first-order Hermite data, i.e., end points and derivatives. The former have fairer shapes (more even curvature variations), especially in cases with inflections.

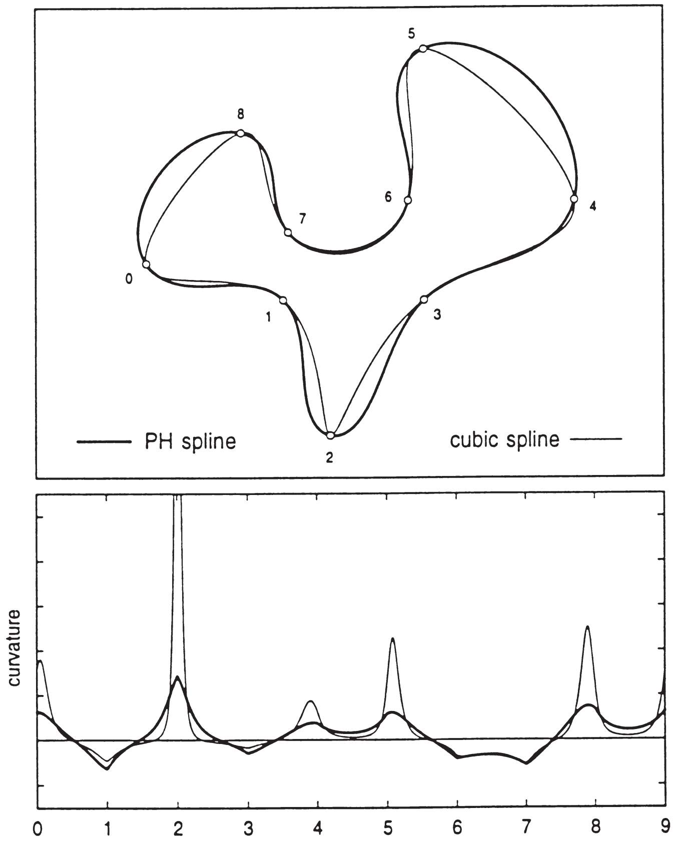

Fig. 2. C^2 PH quintic and 'ordinary' cubic splines that interpolate a set of points with periodic end conditions—the PH spline gives a much fairer locus, as is evident from a comparison of the corresponding curvature plots.

in *basic length units*[2] (or BLUs). The input to the control algorithm is a sequence of discrete position coordinates at each sampling time, and the function of the CNC *interpolator* is to compute these 'reference points' in real time from the given analytic path descriptions and corresponding traversal speeds or *feedrates*.

Traditional G codes accommodate only linear and circular segments with constant feedrates, due to computational difficulties that arise in real-time generation of reference points for free-form segments and variable feedrates. Several authors [3,4,21,22,26,27,29,30,33] have recently described free-form curve interpolators. However, these are based on approximate (linear) Taylor series expansions: they are inaccurate for strongly curved or unevenly parameterized loci, and cannot accommodate variable feedrates.

By contrast, the form (1) of PH curves allows an analytic reduction of the integrals expressing distance travelled along them, at either fixed feedrate or various arc length/curvature-dependent

[2] The BLU for a given CNC machine is determined by its position encoders; it represents the smallest distance the machine can resolve.

feedrates, in each sampling interval [12,18]—this yields an essentially exact computation of the corresponding reference points in real time. The interpolation of PH curves at fixed feedrate is particularly simple to implement, and yields reference points distributed uniformly with respect to true arc length along the curve.

In specifying a feedrate dependence on the distance or arc length s along a curve, it is natural to use the fractional arc length $\lambda = s/S$, where S is the total arc length. Thus, a feedrate increasing linearly (with distance) from an initial value V_0 to a final value V_1 is given by

$$V(\lambda) = V_0(1 - \lambda) + V_1\lambda \tag{3}$$

with $0 \leq \lambda \leq 1$. Similarly, a quadratic feedrate dependence on fractional arc length is specified in Bernstein form as

$$V(\lambda) = V_0(1 - \lambda)^2 + V_1 2(1 - \lambda)\lambda + V_2\lambda^2, \tag{4}$$

where V_0 and V_2 are initial and final feedrates, and V_1 controls whether V varies in a ‘concave’ or ‘convex’ manner between them, compared to a linear variation, according to whether it is smaller or greater than $\frac{1}{2}(V_0 + V_2)$.

PH curve interpolators corresponding to the linear and quadratic feedrate functions (3) and (4) are also simple to implement; complete details may be found in [18]. They provide a repertoire of basic forms which can approximate arbitrary feedrate variations, designed to maintain the dimensional accuracy and surface finish of machined parts—such variations might be determined from, for example, a physical model of the machining process [20].

Another approach is to make the feedrate V depend on the local curvature κ of the PH curve. A particular form of such a feedrate function, for a tool of radius d and a fixed depth of cut δ, is given [12] by

$$V(\kappa) = \frac{V_0}{1 + \kappa(d - \frac{1}{2}\delta)}\,, \tag{5}$$

where V_0 is a nominal (zero-curvature) feedrate. This function compensates for variations in the material removal rate (at a fixed depth of cut) incurred by the tool path curvature, thus giving a more nearly constant mean cutting force than is obtained with fixed feedrate; see Figs. 3 and 8 below.

Thus far we have assumed that the PH curves describe tool paths, and (for a finish cut, at least) the part shape is therefore an offset by the tool radius d from those curves. The usual practice is to define the part geometry in a CAD system, and then compute offset tool paths from it. This necessarily entails approximation, since segments other than lines and circles do not, in general, have rational offsets. One can formulate PH-curve CNC interpolators that bypass this problem [18]—one supplies a part shape (defined by PH curves) and tool radius d directly to an interpolator that computes reference points along the offset

$$\mathbf{r}_d(\xi) = \mathbf{r}(\xi) + d\mathbf{n}(\xi) \tag{6}$$

at distance d to the PH curve segment $\mathbf{r}(\xi)$, with unit normal vector $\mathbf{n}(\xi)$.

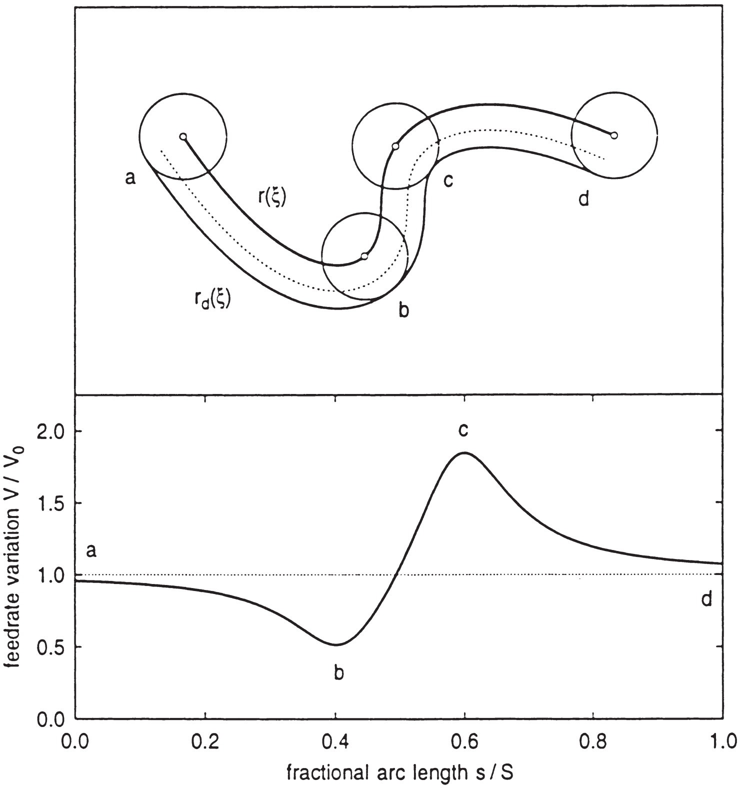

Fig. 3. Curvature-dependent feedrate along a PH quintic $\mathbf{r}(\xi)$ to maintain a constant material removal rate (see Part Program 1). The desired part shape is the offset $\mathbf{r}_d(\xi)$ at the tool radius d, and the dotted curve shows the depth of cut δ above it. In accordance with equation Section 5, the feedrate at the 'concave' position b is lower than the nominal value, and higher in the 'convex' position c, to compensate for increased/diminished overlap of the tool and workpiece.

4. G codes for PH curves

Since G05 is 'unassigned' in the EIA RS-274-D standard [1], we adopt it to specify PH-curve tool paths and associated feedrate functions. A block citing the preparatory function G05 contains one or more words that communicate information on the geometry and traversal rate of a PH curve. Specifically, we adopt the identifiers listed in Table 1 within G05 blocks.

Note that many of these identifiers have established meanings within 'conventional' G code part programs [1], e.g., angular dimensions about the coordinate axes for A, B, C; spindle speed for S; tool selection for T; feedrate for F; and secondary dimensions parallel to the coordinate axes for U, V, W. It is understood, therefore, that the interpretations in Table 1 apply only within G05 blocks, and the 'usual' interpretations prevail in all other blocks. With this convention, no conflict or ambiguity may arise, and PH-curve G codes can be freely combined with 'ordinary' (linear/circular) G codes.

The information needed to precisely specify a PH curve $\mathbf{r}(\xi)$ and feedrate function $V(\xi)$, for $\xi \in [0,1]$, may be incorporated within a single G05 block. However, since shorter blocks are more readable, we allow such specifications to be broken down into sequences of blocks. For this

Table 1
Proposed use of address characters for the specification of PH-curve tool paths, and associated feedrate functions, within G code part programs

Character	Address for:
N	sequence number
G	preparatory function
H	degree of PH curve
X, Y	end-point coordinates
A, B, C, D, E	Bernstein coefficients of $u(\xi)$
P, Q, R, S, T	Bernstein coefficients of $\nu\xi)$
F	feedrate function type
U, V, W	feedrate function parameters

purpose, it is important that the H, X, Y, A, B, C, D, E, P, Q, R, S, T, F, U, V, W words of G05 blocks be modal—i.e., they specify values that will remain in effect until superseded by words of the same type in subsequent G05 blocks. The software is responsible for checking that data given in this manner are both sufficient and consistent in specifying a PH-curve tool path and feedrate function. Further details on the interpretation of words within G05 blocks are as follows:

H: integer specifying degree of PH curve (usually 5, occasionally 9 for special applications—see Section 5.2)
X, Y: coordinates of the PH curve end point
A, B, C... numerical values defining shape of PH curve (A, B, C suffice for PH quintics—D, E are also required for degree-9 PH curves)
P, Q, R... numerical values defining shape of PH curve (P, Q, R suffice for PH quintics—S, T are also required for degree-9 PH curves)
F: an integer specifying the feedrate functional form—the following interpolator types have been developed:
0 = constant feedrate
1 = linear feedrate variation with arc length
2 = quadratic feedrate variation with arc length
3 = curvature-dependent feedrate for constant material removal rate at fixed depth of cut
4 = constant feedrate along an offset curve
U, V, W: numerical values instantiating the feedrate function
F0: U = constant feedrate
F1: U and V = initial and final feedrates
F2: U and W = initial and final feedrates, V = value that controls quadratic variation between U and W values
F3: U = nominal feedrate, V = tool radius, W = depth of cut
F4: U = constant feedrate, V = offset distance

Notes:

- Quintic PH curves (H5) suffice for most design applications; their shape flexibility is similar to that of ‘ordinary’ cubics. Degree-9 PH curves (H9) are intended for specialized uses, such as design of cams [10].

- It is perhaps more natural to define a PH curve by its initial point and the polynomials $u(\xi)$ and $v(\xi)$. We use the X, Y words to specify the final point, however, for compatibility with linear/circular G codes.
- The curve coefficients A, B, C,... and P, Q, R,... do not have intuitive geometrical interpretations—the part programmer should not attempt to ‘guess’ their values, since this yields meaningless results. It is the responsibility of the CAD system to compute them by algorithms for basic geometrical constructions (Hermite interpolants, C^2 splines, etc.).
- A, B, C,... and P, Q, R,... must be specified with sufficient precision to ensure correct start point coordinates—especially for paths composed of contiguous PH curves. The CAD system is also responsible for this.

5. Example part programs

We now give some part programs illustrating the compactness and flexibility of G code specifications for PH tool paths and associated feedrate functions, and how they can be freely combined with ordinary (linear/circular) G codes. In these programs, coordinates are integer values in BLUs and feedrates are in BLU/min; for the machine on which the CNC interpolators were tested, 1 BLU = 0.01 mm. The coefficients of the polynomials $u(\xi)$ and $v(\xi)$ are in decimal form, with enough digits to guarantee that the computed coordinates of the curve start point $\mathbf{r}(0)$ are exact when rounded to integer BLU values. The part programs assume that the tool is initially positioned at (0,0).

5.1. PH quintic with various feedrates

The program below defines a single PH quintic segment, traversed with the curvature-dependent feedrate (5) that compensates for tool-path curvature to yield a constant material removal rate at fixed depth of cut.

Part Program 1

```
N05 G05 F3 U30000 V635 W476
N10 G05 H5 X5080 Y0
N15 G05 A130.712 B-51.811 C138.385 P-69.955 Q128.872 R-29.367
```

The word F3 in block N05 specifies a curvature-dependent feedrate function for PH curves, with the following parameters: a nominal feedrate (U) of 30,000 BLU/min (5 mm/s), a tool radius (V) of 635 BLU (6.35 mm or 1/4 in), and a depth of cut (W) of 476 BLU (4.76 mm or 3/16 in). Blocks N10 and N15 specify the actual PH quintic tool path, with end point (5080, 0) and coefficients for $u(\xi)$ and $v(\xi)$ given by A, B, C and P, Q, R. The three blocks could be readily combined, but separating them makes for a more readable program.

Fig. 3 illustrates the tool path and associated feedrate function defined by Part Program 1. Other feedrate functions could have been specified in lieu of that defined by Eq. (5). For example, the block

N05 G05 F1 U24000 V36000

defines—through the F1 word—a feedrate that is linearly increasing (with respect to arc length) from 4 mm/s to 6 mm/s.

5.2. Cam design and manufacture

A context in which the PH curves can offer significant advantages over current practice is the problem of design and manufacture of cams [10]. For a planar dwell-rise-dwell-return cam, with a zero-offset roller follower, the cam pitch curve comprises two circular dwell arcs of specified radii and angular extents, connected in a 'sufficiently smooth' manner (i.e., curvature-continuous if the follower displacement, velocity, and acceleration are to be continuous) by the rise and return profiles. This can be accomplished using second-order PH Hermite interpolants of degree 9—see [10] for complete details.

Fig. 4 shows a cam pitch curve designed in this manner; this is an offset from the actual cam shape by a distance equal to the follower radius—it determines the follower motion. We assume the cam will be cut with a tool of the same radius as the follower, so the pitch curve represents the desired tool path. The part program for this path is shown below: it comprises two degree-9 PH curves (the 'rise' and 'return' portions) and four circular arcs (each of the dwell circular arcs must be split in two, since traditional G codes only define arcs within one quadrant of the coordinate system).

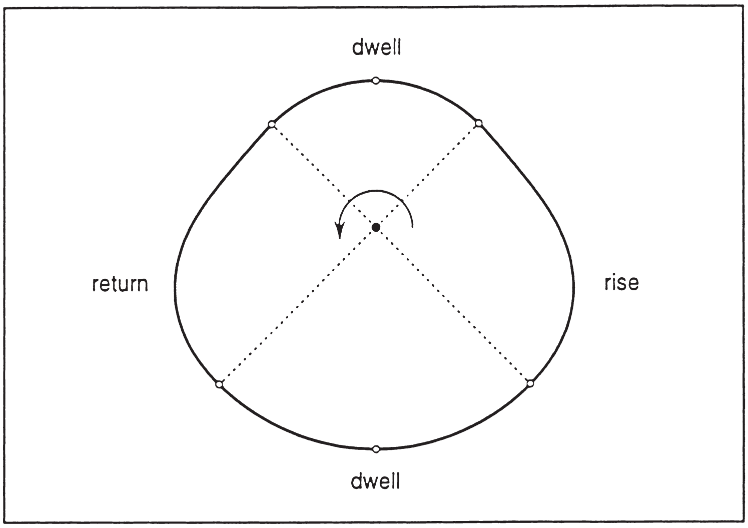

Fig. 4. Cam pitch curve comprising two circular dwell arcs with C^2 splices to rise/return profiles defined by degree-9 PH curves (see Part Program 2).

Part Program 2

```
N05 G03 X2694 Y1116 I0 J3810 F24000
N10 G05 H9 X1796 Y5606 F0 U24000
N15 G05 A68.432 B78.556 C12.213 D36.348 E23.463
N20 G05 P28.345 Q46.970 R81.956 S58.111 T56.645
N25 G03 X0 Y6350 I1796 J1796
N30 G03 X-1796 Y5606 I0 J2540
N35 G05 H9 X-2694 Y1116
N40 G05 A-23.463 B-36.348 C-12.213 D-78.556 E-68.432
N45 G05 P56.645 Q58.111 R81.956 S46.970 T28.345
N50 G03 X0 Y0 I2694 J2694
```

The follower displacement, velocity, and acceleration functions obtained from PH-curve cam designs compare favorably with ‘standard’ forms, such as the cycloidal and 3-4-5 polynomial functions [28]. Moreover, unlike the latter, PH cams and their pitch curves are rational—they can be downloaded to an open-architecture CNC machine and cut directly from their analytic definitions, bypassing the need for linear/circular G code approximations.

5.3. C^2 PH quintic spline curve

The PH spline illustrated in Fig. 2, interpolating nine distinct points with periodic end conditions, was constructed using a complex-variable homotopy method [2]. It comprises a sequence of nine PH quintic segments that meet with C^2 continuity to form a closed loop. Each segment is describable by one G-code block. A single feedrate function may be assigned to all the segments through an initial G-code block. Alternately, an individual feedrate function may be appended to the G-code block for each segment.

Part Program 3

```
N05 G05 H5 F0 U37200
N10 G05 X1092 Y-294 A-31.026 B-38.537 C-31.481 P16.934 Q-16.436 R13.062
N15 G05 X1470 Y-1386 A-31.481 B-24.426 C-28.476 P13.062 Q42.560 R2.794
N20 G05 X2226 Y-294 A-28.476 B-32.526 C-36.896 P2.794 Q-36.972 R-14.887
N25 G05 X3444 Y504 A-36.896 B-41.267 C-32.579 P-14.887 Q7.197 R-25.365
N30 G05 X2226 Y1722 A-32.579 B-23.892 C8.378 P-25.365 Q-57.927 R-36.389
N35 G05 X2100 Y504 A8.378 B40.647 C21.987 P-36.389 Q-14.851 R-28.356
N40 G05 X1134 Y252 A21.987 B3.326 C-13.761 P-28.356 Q-41.861 R-28.525
N45 G05 X756 Y1050 A-13.761 B-30.849 C-3.668 P-28.525 Q-15.189 R-32.746
N50 G05 X0 Y0 A-3.668 B23.514 C31.026 P-32.746 Q-50.304 R-16.934
```

In the above part program, the word F0 in block N05 specifies a constant feedrate with a value of 37200 BLU/min (6.2 mm/s). Since the feedrate-function assignment for PH curves is modal—i.e., it remains in effect until superseded by a new assignment—it is understood that the entire curve is to be traversed at this fixed feedrate. Each of the remaining blocks N10-N50 specifies a PH quintic segment of the spline curve—X and Y are the segment end-point coordinates, corresponding to the discrete points that the spline must interpolate, while A, B, C and P, Q, R specify the Bernstein coefficients of the two quadratic polynomials $u(\xi)$ and $v(\xi)$.

For comparison, Part Program 4 is a G code approximation (with 1 BLU tolerance) of the PH spline. This was generated by reading exact descriptions of the nine segments, as quintic Bézier curves, into the Unigraphics CAD system and using its NC capability. Coordinates have been rounded to integer BLU units—a total of 115 linear/circular segments were needed to achieve the 1 BLU accuracy (i.e., 0.01 mm for our machine).

Part Program 4

```
N20 G01 X0 Y0 F37200
N25 X-23 Y41
N30 X-41 Y87
N35 X-54 Y136
N40 X-62 Y189
N45 G02 X-65 Y249 I651 J59
N50 G02 X-23 Y478 I654 J0
N55 G02 X422 Y992 I881 J314
N60 G01 X474 Y1015
N65 X525 Y1034
N70 X575 Y1047
N75 X624 Y1055
```

... 100 lines omitted...

```
N580 X100 Y-101
N585 X63 Y-73
N590 X29 Y-39
N595 X0 Y0
```

6. Experimental results

Experiments with the PH curve part programs were performed on a 3 hp open-architecture CNC milling machine, with interpolators implemented for two axes. The original machine control system has been replaced by a 33 MHz 80486-based PC incorporating our own software interpolator and controller, and custom-made hardware components such as a pulse-width-modulation (PWM) board.[3] In addition, incremental linear encoders and tachometers are used to provide table position and motor speed feedback. The hardware components and control computer are interfaced through a digital I/O board, an analog-to-digital converter (ADC), and a quadrature decoder board.

The system has a controller sampling interval of 0.01 s and basic length unit (BLU), as defined by the position encoders, of 0.01 mm. The controller compares instantaneous positions measured by the encoders with reference points computed by the interpolator. To emphasize the influence

[3] Earlier experiments with this CNC system have been reported in [11] and [26].

of the interpolation scheme (rather than the control algorithm) on the machine performance, a simple proportional (P) controller was adopted.

6.1. Feedrate functions

The feedrate along a curve is the magnitude of the velocity vector whose (x,y) components are measured by the tachometers. Fig. 5 compares measured feedrates with Part Programs 3 and 4, i.e., the PH spline shown in Fig. 2 traversed at 6.2 mm/s, using both the exact description and a piecewise-linear/circular G code approximation (with 1 BLU tolerance). The feedrate fluctuations are clearly much smaller using the analytic curve interpolator, corresponding to a smoother execution of the trajectory.

Experiments were also performed using variable-feedrate interpolators. Fig. 6 shows tachometer measurements from traversal of a degree-9 PH curve, with the feedrate increasing linearly (with respect to arc length) by a factor of ~ 2. To compare specified and realized feedrates, the former must be expressed in terms of time: a linear variation of V with arc length s, from V_0 to V_1, actually corresponds to the exponential time-dependence

$$V(t) = V_0 \exp \frac{(V_1 - V_0)t}{S} ,$$

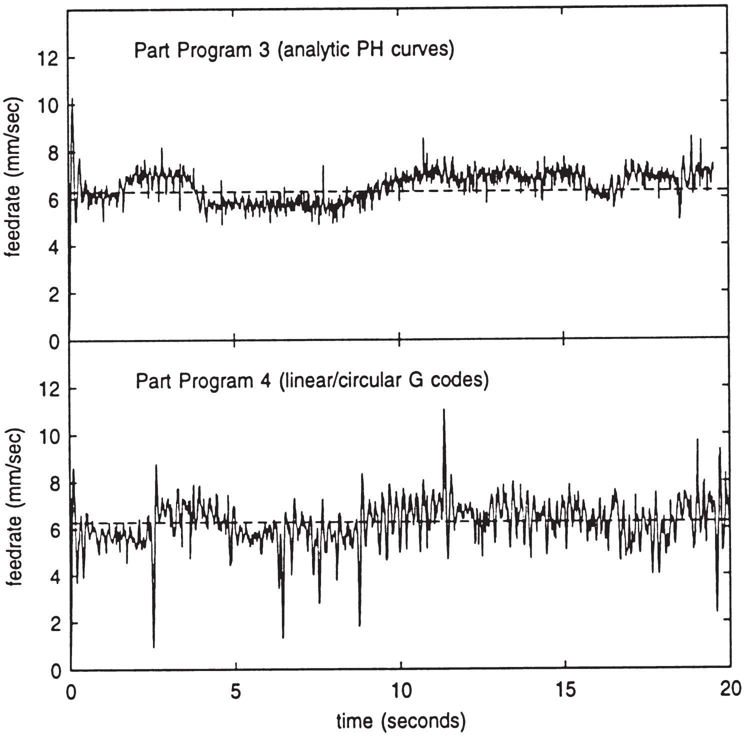

Fig. 5. Measured feedrates for the example in Fig. 2, based on the exact PH-curve description (upper) and a linear/circular G code approximation (lower) with a 1 BLU tolerance. The PH-curve interpolator exhibits smaller deviations from the nominally-specified constant feedrate of 6.2 mm/s.

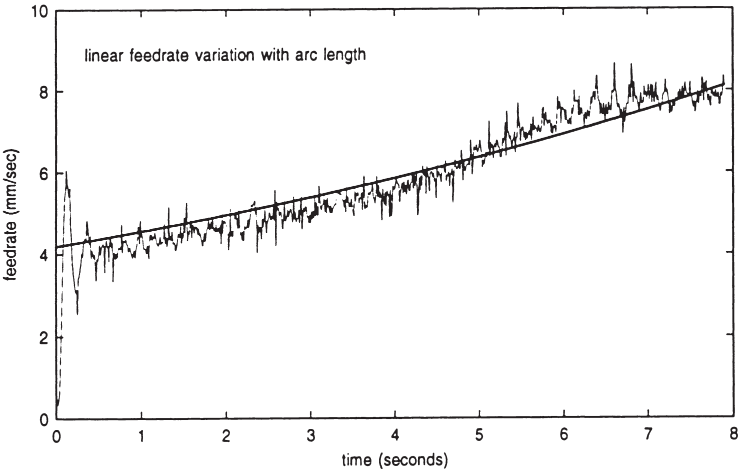

Fig. 6. Tachometer measurements from traversal of a degree-9 PH curve with feedrate function specifying a linear increase (with respect to arc length) from 4.2 to 8.2 mm/s, as shown by the solid line; this is somewhat curved since linear dependence on arc length implies an exponential time variation.

S being the total arc length; the traversal time is $T = (V_1 - V_0)^{-1} S \ln V_1/V_0$. In Fig. 6 we see that, apart from the initial acceleration (and overshoot) from rest, the interpolator accurately realizes the feedrate function (3).

In Fig. 7 we show results from a run on a PH quintic using the curvature-dependent feedrate (5). For this case, there is no simple explicit dependence of feedrate V on time t, and to compare the specified and realized feedrates we must compute discrete values of V and t corresponding to the parameter value ξ of each reference point, and then plot the former against the latter—this is shown as the solid curve in Fig. 7. The agreement of the actual feedrate with the variation (5)

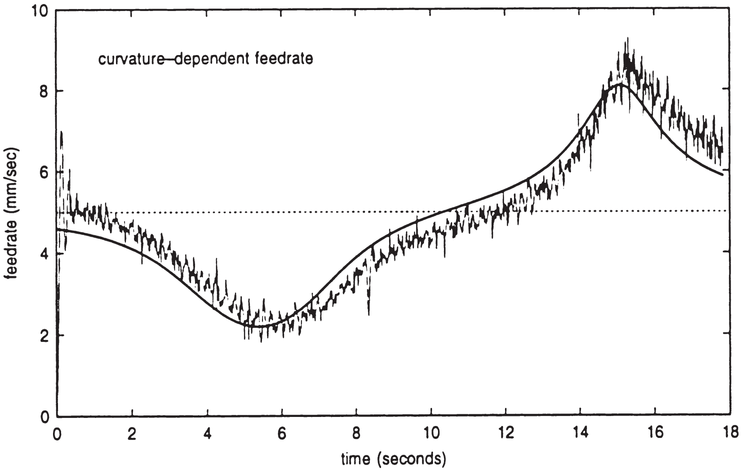

Fig. 7. Tachometer measurements along a PH quintic with the curvature-dependent feedrate for uniform material removal rate at a fixed depth of cut (shown by the solid line). The nominal (zero curvature) feedrate is 5 mm/s.

is again excellent, apart from a slight 'lag' that may due to the control algorithm or the initial acceleration from rest.

6.2. Machining forces

Since the primary goal of the curvature-dependent feedrate function (5) is to minimize variations in the material removal rate (and thus the mean cutting force), the instantaneous force was monitored by mounting a Kistler piezo-electric dynamometer on the machine table while machining aluminum at a fixed depth of cut along the PH quintic shown in Fig. 3. Part Program 1 was employed for this purpose, as shown in Section 5 above with the curvature-dependent feedrate, and also with a constant feedrate specified by

```
N05 G05 F0 U30000
```

in lieu of the first line in Part Program 1.

Fig. 8 compares force measurements from the constant and curvature-dependent feedrate runs. The graph of the *instantaneous* cutting force shows prominent 'spikes' that correspond to successive engagements of the tool cutting edges with the workpiece. Thus, to discern broad trends in the force data, we present 1 s moving averages of it in Fig. 8. A force sampling rate of 250 Hz was employed, much higher than the 46.7 Hz frequency of the force 'spikes' incurred by the two-flute cutter and 1400 rpm spindle speed, to avoid biasing of the averaged data by aliasing effects.

To provide better insight into its variation with location along the curve, we plot the averaged force against *arc length*, rather than *time*, in Fig. 8 (these variables are proportional for a constant feedrate, but have a rather complex non-linear relationship for the curvature-dependent feedrate). Also, the initial and final ~ 6 mm of curve arc length, where there are insufficient data to compute moving averages, have been omitted from Fig. 8.

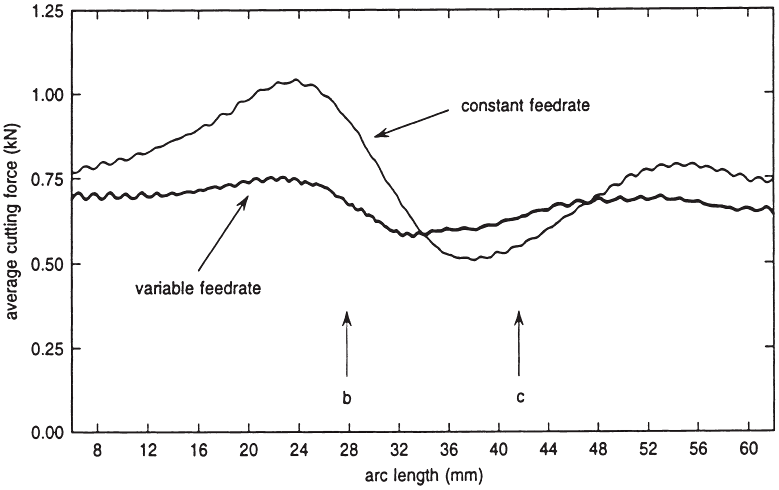

Fig. 8. Comparison of mean cutting force (with 1-second smoothing) along the PH quintic in Fig. 3, using both constant feedrate and the curvature-dependent feedrate (5) for a fixed depth of cut. The arrows *b* and *c* identify points of greatest positive and negative tool path curvature; see Fig. 3.

The data clearly show that, at a fixed depth of cut, machining the curve in Fig. 3 with constant feedrate incurs substantial variations of the mean cutting force (by a factor of ~ 2). By using the PH-curve CNC interpolator corresponding to the curvature-dependent feedrate (5), these variations are suppressed to minimal levels ($\lesssim \pm 10\%$). This 'force control'—which can offer improvements in dimensional accuracy, surface finish, and process time—is accomplished merely by low-level system software modifications, based on 'intelligence' of the exact tool path geometry. By contrast, the customary piecewise-linear/circular G code approximations need expensive sensors and sophisticated control algorithms to provide an equivalent functionality.

The arrows labelled b and c in Fig. 8 identify positions of most positive and negative curvature along the tool path illustrated in Fig. 3. Compared to these positions, a slight 'lead'—comparable to the tool radius d—in the occurrence of the maximum and minimum force (at fixed feedrate) is evident. The feedrate function (5) is based upon an osculating-circle approximation to the tool path at each point [12], and yields a truly constant material removal rate only to the extent that this approximation is accurate over lengths $\sim d$. This explains the residual force variations in the variable feedrate case.

Although the feedrate function (5) is designed specifically to compensate for variations in material removal rate incurred by the 'curvature effect' at fixed depth of cut, it is possible to formulate PH curve interpolators that also accommodate certain variable depth-of-cut functions. These interpolators offer a simpler and cheaper means of controlling machining force variations than adaptive control strategies [24,25,31] in cases where those variations are known a priori to be primarily due to volumetric effects.

6.3. Surface finish

We have found that the PH curve interpolators provide significantly smoother surface finish on machined parts than can be obtained by use of 'ordinary' linear/circular G code tool path approximations, even if the nominal accuracy of the latter exceeds the machine resolution (< 1 BLU). This is apparently due to the fact that the analytic interpolator yields a 'smooth' execution of the trajectory, compared to the inherently rather 'jerky' motion associated with discrete linear/circular G code approximations—see Fig. 5.

Fig. 9 gives a comparison of the PH quintic in Fig. 3, when machined (at a constant feedrate) in wax using both the analytic PH curve interpolator and a 1 BLU G code approximation. The difference is most clearly apparent on the bottom surface, where the end points of the short G code segments are evident as a sequence of consecutive 'rings' along the tool path—with the analytic interpolator, on the other hand, the path is described by a single smooth PH segment, and these rings have been eliminated.

A difference in surface finish is also present (though not quite as obvious) on the vertical 'side walls' of the cut: a more prominent 'ribbing' effect can be seen on these surfaces in the case of the linear/circular G code run. We do not currently have the means to provide quantitative measurements of surface roughness for parts machined using PH curve and 'ordinary' linear/circular G code interpolators, but we hope to report on this in a future study.

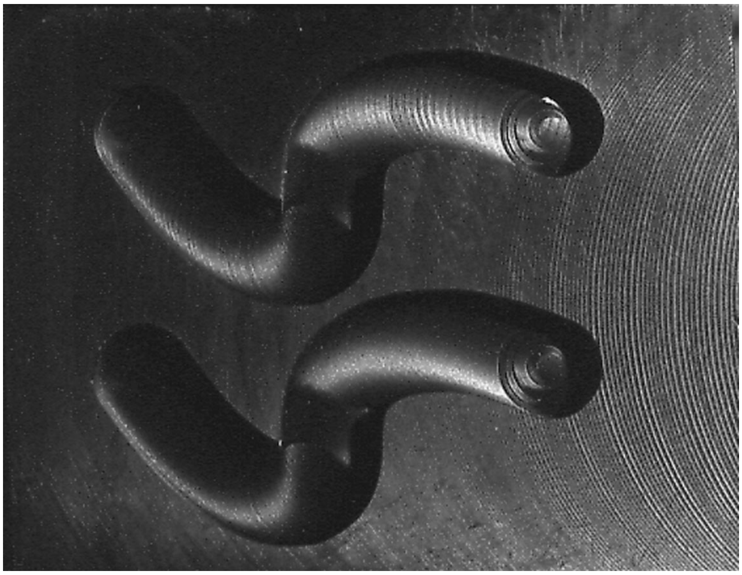

Fig. 9. Comparison of surface finish for cuts in wax obtained when using the analytic PH curve interpolator (lower cut) and a linear/circular G code approximation (upper cut) with a prescribed tolerance of 1 BLU.

6.4. Dimensional accuracy

It is expected that the analytic PH curve interpolators will also yield better dimensional accuracy of machined parts than linear/circular G code approximations. Unfortunately, we cannot report direct empirical evidence for this at present, since experiments indicate that the contouring accuracy of our machine is constrained mainly by substantial backlash (up to ~ 10 BLU) in the drives, regardless of the interpolator or tool path representation used—see [11] for further details. We are interested in hearing from readers with access to open-architecture CNC machines that employ more accurate drive systems, allowing meaningful experiments to be conducted on the influence of interpolators and tool path representations on machining accuracy.

6.5. Machining of free-form surfaces

For brevity we have concentrated on planar PH curve tool paths, although the ability to execute spatial paths is important in machining of free-form surfaces. The PH space curves are described in [17], and one may formulate CNC interpolators for them by straightforward extensions of the planar PH curve algorithms. Some preliminary results concerning their use in contour machining of free-form surfaces with a ball-end mill are described in a forthcoming paper [19]. In this context, PH space curves are used to approximate tool-center loci on the offset to the part surface, such that the tool contacts the surface along planar sections of it at successive z heights. There is scope for further useful work on approximating surface tool paths by PH curves so as to control scallop height, optimize the total machining time, etc.

7. Concluding remarks

We have shown that PH curves can be machined directly from their analytic descriptions, by means of real-time CNC interpolators that offer considerable flexibility in specifying feedrate variations along tool paths. Compared to the usual approach of approximating free-form paths by linear/circular segments, the use of PH curves provides: (1) exact specifications for tool paths or part shapes; (2) more concise part programs; (3) control over machining forces due to variable depth of cut, or tool path curvature at fixed depth of cut; and (4) a smoother finish and better dimensional accuracy of machined surfaces. This is accomplished merely by software changes in open-architecture CNC systems, without the need for expensive hardware modifications.

The main contribution of this paper has been to propose a system of G codes for communicating PH curve tool paths and feedrate functions to CNC machines. These new G codes were designed to be compatible with existing conventions for ‘ordinary’ (linear/circular) G codes. The key requirement in transferring the PH curve methods to practical use is the provision of CAD software that can be used to design part shapes directly in terms of PH curves, and then output the appropriate PH curve G codes to machine them. Many of the necessary algorithms are already available, and can be incorporated into commercial CAD systems or customized stand-alone packages.

Finally we mention that, although we have emphasized CNC machining, the methods described herein are expected to be of value in other application domains—for example, in programming coordinate measuring machines and in compensating for ‘beam overlap’ in laser cutting along curved paths.

Acknowledgements

We are grateful to Professor Yoram Koren for advice and encouragement. This work was supported by the National Science Foundation (CCR-9530741) and the Office of Naval Research (N00014-95-1-0767).

References

[1] EIA Standard RS-274-D, Interchangeable variable block data format for positioning, contouring, and contouring/positioning numerically controlled machines, Electronic Industries Association, Engineering Dept., Washington, D.C., 1979.

[2] G. Albrecht, R.T. Farouki, Construction of C^2 Pythagorean-hodograph interpolating splines by the homotopy method, Advances in Computational Mathematics 5 (1996) 417–442.

[3] J.-J. Chou, D.C.H. Yang, Command axis generation for three-axis CNC machining, ASME Journal of Engineering for Industry 113 (August) (1991) 305–310.

[4] J.-J. Chou, D.C.H. Yang, On the generation of coordinated motion of five-axis CNC/CMM machines, ASME Journal of Engineering for Industry 114 (February) (1992) 15–22.

[5] G. Farin, Curves and Surfaces for Computer Aided Geometric Design, 3rd ed., Academic Press, Boston, 1993.

[6] R.T. Farouki, Pythagorean-hodograph curves in practical use, in: R.E. Barnhill (Ed.), Geometry Processing for Design and Manufacturing, SIAM, Philadelphia, 1992, pp. 3–33.

[7] R.T. Farouki, The conformal map $z \rightarrow z^2$ of the hodograph plane, Computer Aided Geometric Design 11 (1994) 363–390.

[8] R.T. Farouki, The elastic bending energy of Pythagorean-hodograph curves, Computer Aided Geometric Design 13 (1996) 227–241.

[9] R.T. Farouki, Pythagorean-hodograph quintic transition curves of monotone curvature, Computer Aided Design 29 (1997) 601–606.

[10] R.T. Farouki, S. Jee, J. Manjunathaiah, Design of rational cam profiles with Pythagorean-hodograph curves, Mechanism and Machine Theory 33 (1998) 669–682.

[11] R.T. Farouki, S. Jee, J. Manjunathaiah, Implementation and performance analysis of a real-time CNC interpolator for precision machining of Pythagorean-hodograph curves, ASME Journal of Manufacturing Science and Engineering, submitted.

[12] R.T. Farouki, S. Jee, J. Manjunathaiah, D. Nicholas, G.-F. Yuan, Variable feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves, Computer Aided Design, in press.

[13] R.T. Farouki, C.A. Neff, Analytic properties of plane offset curves and Algebraic properties of plane offset curves, Computer Aided Geometric Design 7 (1990) 83–127.

[14] R.T. Farouki, C.A. Neff, Hermite interpolation by Pythagorean-hodograph quintics, Mathematics of Computation 64 (1995) 1589–1609.

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

[16] R.T. Farouki, T. Sakkalis, Pythagorean hodographs, IBM Journal of Research and Development 34 (1990) 736–752.

[17] R.T. Farouki, T. Sakkalis, Pythagorean-hodograph space curves, Advances in Computational Mathematics 2 (1994) 41–66.

[18] R.T. Farouki, S. Shah, Real-time CNC interpolators for Pythagorean-hodograph curves, Computer Aided Geometric Design 13 (1996) 583–600.

[19] R.T. Farouki, Y.-F. Tsai, G.-F. Yuan, Contour machining of free-form surfaces with real-time PH curve CNC interpolators, Computer Aided Geometric Design, in press.

[20] B.K. Fussel, C. Ersoy, J.B. Jerard, Computer generated CNC machining feed rates, Proceedings of the JAPAN/USA Symposium on Flexible Automation, vol. 1, ASME, 1992, pp. 377–383.

[21] J.-T. Huang, D.C.H. Yang, A generalized interpolator for command generation of parametric curves in computer-controlled machines, Proceedings of the Japan/USA Symposium on Flexible Automation, vol. 1, ASME, 1992, pp. 393–399.

[22] Y. Koren, Interpolator for a computer numerical control system, IEEE Transactions on Computers C-25 (1976) 32–37.

[23] Y. Koren, Computer Control of Manufacturing Systems, McGraw-Hill, New York, 1983.

[24] Y. Koren, A.G. Ulsoy, Adaptive control, in: Metals Handbook, vol. 16: Machining, ASM, Cleveland, 1989, pp. 618–626.

[25] L.K. Lauderbaugh, A.G. Ulsoy, Model reference adaptive force control in milling, ASME Journal of Engineering for Industry 111 (1989) 13–21.

[26] R.-S. Lin, Y. Koren, Real-time interpolators for multi-axis CNC machine tools, Manufacturing Systems 25 (1996) 145–149.

[27] T. Sata, F. Kimura, N. Okada, M. Hosaka, A new method of NC interpolation for machining the sculptured surface, Annals of the CIRP 30 (1981) 369–372.

[28] J.E. Shigley, J.J. Uicker, Theory of Machines and Mechanisms, McGraw-Hill, New York, 1980.

[29] A. Shima, T. Sasaki, T. Ohtsuki, Y. Wakinotani, 64-bit RISC-based Series 15 NURBS interpolation, FANUC Technical Review 9 (1) (1996) 23–28.

[30] M. Shpitalni, Y. Koren, C.C. Lo, Realtime curve interpolators, Computer Aided Design 26 (1994) 832–838.

[31] A.G. Ulsoy, Y. Koren, Control of machining processes, ASME Journal of Dynamic Systems, Measurement, and Control 115 (1993) 301–308.

[32] D.J. Walton, D.S. Meek, A Pythagorean-hodograph quintic spiral, Computer Aided Design 28 (1996) 943–950.

[33] D.C.H. Yang, T. Kong, Parametric interpolator versus linear interpolator for precision CNC machining, Computer Aided Design 26 (1994) 225–234.