

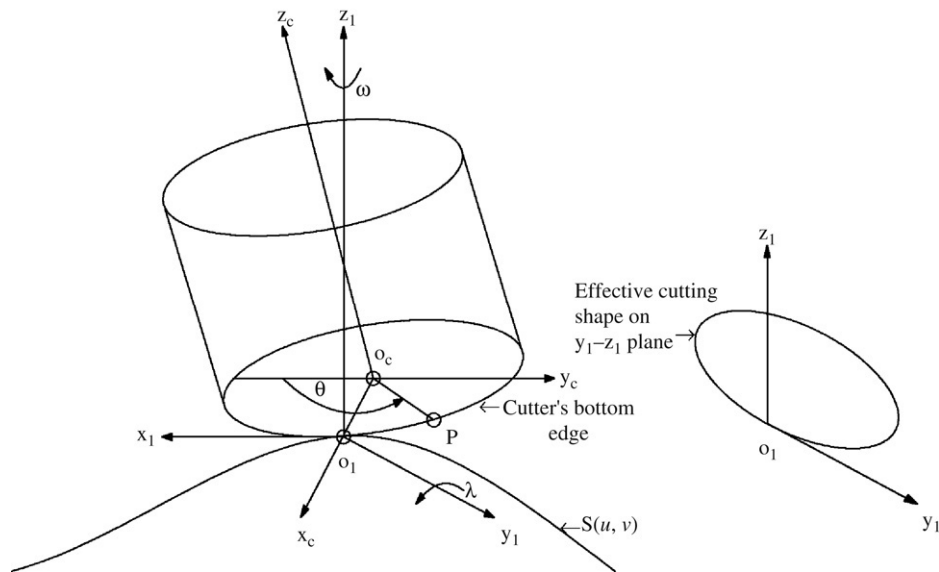


Fig. A.1. Geometrical analysis of the cutter.

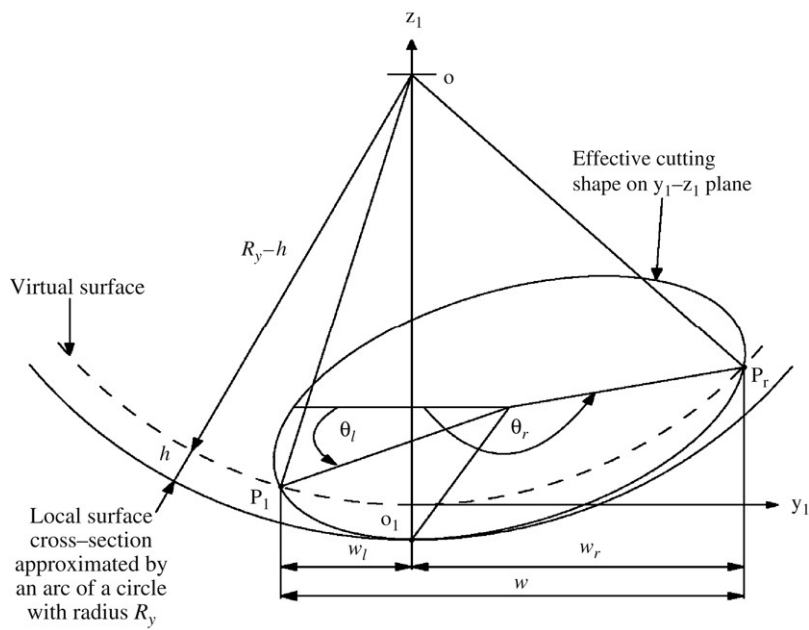


Fig. A.2. Machining strip width estimation.

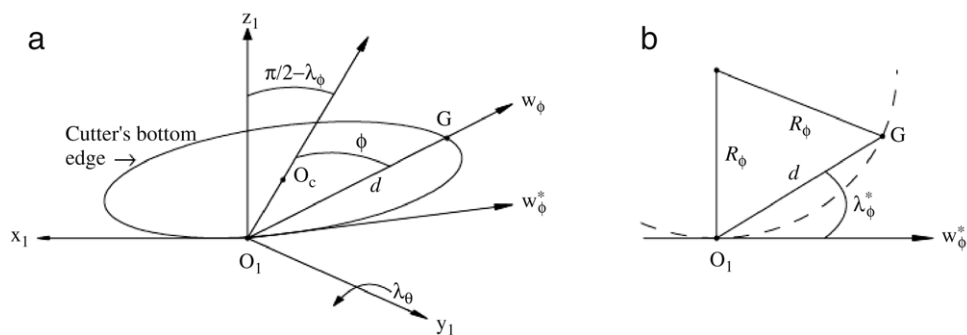


Fig. B.1. Tool gouging.

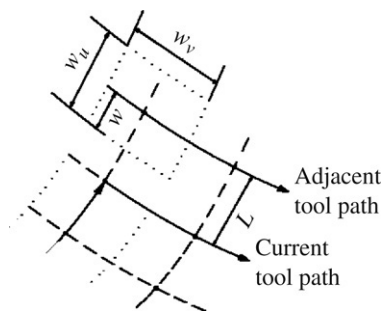


Fig. C.1. Machining strips (dashed lines) on adjacent tool paths generated by using space-filling curve.

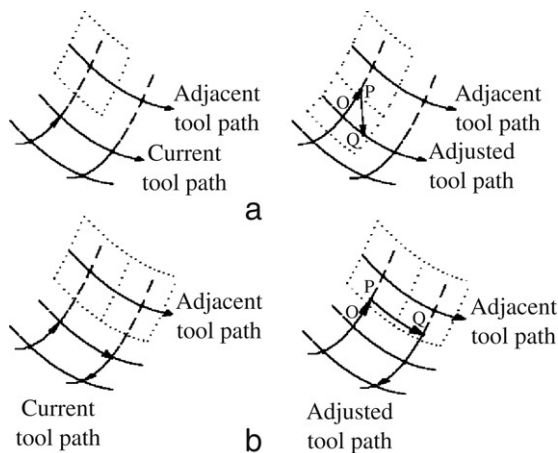


Fig. C.2. Tool path trajectory alteration at (a) corner turn (left or right) and (b) U turn.

where L and w denote the tool path interval and the side machining strip width (left or right), respectively, as shown in Fig. C.1

The second modification is applied to the tool orientation. The tool orientation is usually set by inclining the tool by λ in the tool cutting direction.² At sharp turns, the tool orientation changes abruptly creating large kinematics error. This kinematics error could not be reduced by merely inserting more points as usually done for tool path segmentation [4,22]. Additionally, the tool orientation of the newly inserted CC point needs to be adjusted by interpolating the tool orientations at the two adjacent CC points.

Consider the sharp turn $o-p-q$ shown in Fig. C.2. To make a correct turn, the feed direction at the turning point p is first aligned with the feed direction at the previous point o . To reduce the kinematics error when going from point p to point q , a new point p' is inserted and the feed direction is set to

$$\mathbf{f}_{p'} = \frac{\mathbf{f}_p + \mathbf{f}_q}{|\mathbf{f}_p + \mathbf{f}_q|},$$

where $\mathbf{f}_p$ is the feed direction at point p .

² The tool cutting direction is the direction from the current CC point to the next CC point.

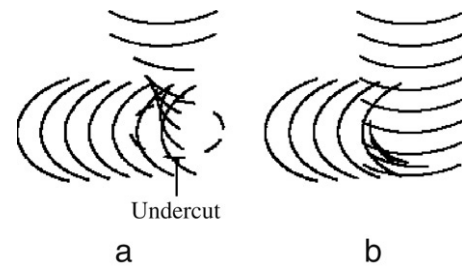


Fig. C.3. Trajectories of the cutter's effective cutting edge (projected onto the $x-y$ plane) (a) before and (b) after the tool path correction.

Fig. C.3 shows the trajectories of the effective cutting edge of the tool projected onto the $x-y$ plane before and after applying the tool path correction.

References

- [1] Anotaipaiboon W, Makhanov SS. Tool path generation for five-axis NC machining using space-filling curves. In: Proceeding of the third asian conference on industrial automation and robotics, vol. 1; 2003. p 23–8.
- [2] Anotaipaiboon W, Makhanov SS. Tool path generation for five-axis NC machining using adaptive space-filling curves. *Int J Prod Res* 2005; 1643–65.
- [3] Anotaipaiboon W, Makhanov SS, Bohez EJ. Optimal setup for 5-axis machining. *Int J Mach Tools Manu* 2006;964–77.
- [4] Bohez E, Makhanov SS, Sonthipermoon K. Adaptive nonlinear tool path optimization for five-axis machining. *Int J Prod Res* 2000;4329–43.
- [5] Brackbill JU, Saltzman JS. Adaptive zoning for singular problems in two dimensions. *J Comput Phys* 1982;342–68.
- [6] Bieterman MB, Sandstrom DR. A curvilinear tool-path method for pocket machining. *J Mater Process Technol* 2003;125(4):709–15.
- [7] Charakhch'yan AA, Ivanenko SA. A variational form of the Winslow grid generator. *J Comput Phys* 1997;136(2):385–98.
- [8] Cox JJ, Takezaki T, Ferguson HRP, Kohkonen KE, Mulkay EL. Space-filling curves in tool-path applications. *Comput Aided Design* 1994;26: 215–24.
- [9] Gao J, Chen X, Zheng D, Yilmaz O, Gindy N. Adaptive restoration of complex geometry parts through reverse engineering application. *Adv Eng Softw* 2006;37(9):592–600.
- [10] Griffiths JG. Tool path based on Hilbert's curve. *Comput-Aided Design* 1994;26:839–44.
- [11] Hopcroft JE, Ullman JD. Introduction to automata theory, languages, and computation. New York: Addison-Wesley; 1979.
- [12] Ivanenko SA. Generation of non-degenerate meshes. *USSR Comput Math Math Phys* 1988;28:141–6.
- [13] Ivanenko SA. Harmonic mappings. In: Thompson JF, Soni BK, Weatherill NP, editors. Handbook of grid generation, CRC Press LLC; 1999. pp. 8 (1–43).
- [14] Lam WM, Shapiro JH. A class of fast algorithms for the Peano–Hilbert space-filling curve. *Proceedings ICIP-94*, vol. 1. IEEE Computer Society; 1994. p. 638–41.
- [15] Lauwers B, Dejonghe P, Kruth JP. Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machining simulation. *Comput Aided Design* 2003;35:421–32.
- [16] Lee Y-S. Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. *Comput Aided Design* 1997;29:507–21.
- [17] Lo C-C. Efficient cutter-path planning for five-axis surface for machining with a flat-end cutter. *Comput Aided Design* 1999;31:557–66.
- [18] Rao A, Sarma R. On local gouging in five-axis sculptured surface machining using flat end tools. *Comput Aided Design* 2000;32:409–20.
- [19] Robert FS. Applied combinatorics. Englewood Cliffs (NJ): Prentice-Hall; 1984.
- [20] Dafner R, Cohen-Or D, Matias Y. Context-based space filling curves. *Comput Graph Forum* 2000;19:C209–C217.
- [21] Makhanov SS. An application of the grid generation techniques to optimize a tool-path of industrial milling robots. *J Comput Math Math Phys* 1999;39:1589–600.

- [22] Makhanov SS, Batanov D, Bohez E, Sonthipaumpoon K, Anotaipaiboon W, Tabucanon M. On the tool-path optimization of a milling robot. *Comput Ind Eng* 2002;43:455–72.
- [23] Makhanov SS, Ivanenko SA. Grid generation as applied to optimize cutting operations of a five-axis milling machine. *Appl Numer Math* 2003;46:353–77.
- [24] Thomson JF, Soni B, Weatherill N. *Handbook of grid generation*. CRC Press; 1999.
- [25] Winslow AM. Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J Comput Phys* 1966;1(2):149–72.
- [26] Yoon J-H. Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining. *Int J Prod Res* 2003;41:2125–42.

Space-filling curves in adaptive curvilinear coordinates for computer numerically controlled five-axis machining

S.S. Makhanov

Information Technology Program, School of Information, Computer and Communication Technology, Sirindhorn International Institute of Technology, Thammasat University, Thailand

Available online 7 February 2009

Abstract

The paper presents a concatenation of two methods for optimization of five-axis machining proposed earlier by the author. The first method is based on the grid generation techniques whereas the second method exploits the space filling curve technologies. Combination of the two techniques is superior with regard to the conventional methods and with regard to the case when the two methods are applied independently.

© 2009 IMACS. Published by Elsevier B.V. All rights reserved.

1. Introduction

Milling machines are programmable mechanisms for cutting industrial parts. The axes of the machine define the number of the degrees of freedom of the cutting device. Five axes provide that the cutting device (the tool) is capable of approaching the machined surface at a given point with a required orientation. The machines consist of several moving parts designed to establish the required coordinates and orientations of the tool during the cutting process (see [Figs. 1 and 2](#)). The movements of the machine parts are guided by a controller which is fed with a so-called NC program comprising commands carrying three spatial coordinates of the tool-tip and a pair of rotation angles needed to rotate the machine parts to establish the orientation of the tool.

1.1. Tool path generation

The tool path is a sequence of positions possibly arranged into a structured spatial pattern. The conventional engineering patterns are the zigzag and the spiral (see [Fig. 3](#)).

Tool path planning for five-axis machining requires a multi-criteria optimization governed by estimates of the difference between the required and the actual surface. Additionally, the criteria vector may include the length of the path, the negative of the machining strip (strip maximization), the machining time, etc. (see for instance [\[20,23,29\]](#)). Besides, the optimization could be subjected to constraints [\[22,39\]](#) the most important of which are

- The scallop height constraints. The scallops between the successive tool tracks must not exceed a prescribed tolerance.
- The local accessibility constraints. The constraint ensures against the removal of an excess material when the tool comes in contact with the desired surface due to the so-called curvature interference.

E-mail address: makhanov@siit.tu.ac.th.

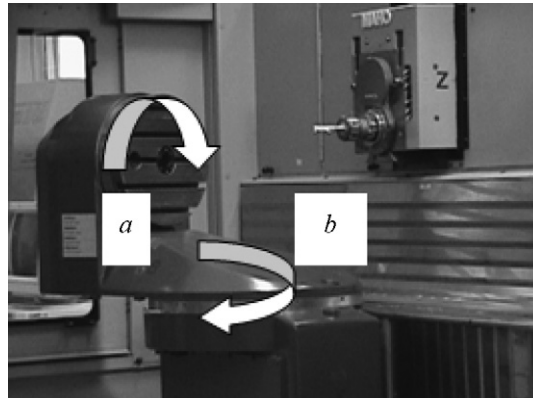


Fig. 1. Five-axis milling machine MAHO600E (Deckel Maho Gildemeister) *a* and *b* are the rotation axes.



Fig. 2. MAHO600E during cutting operations.

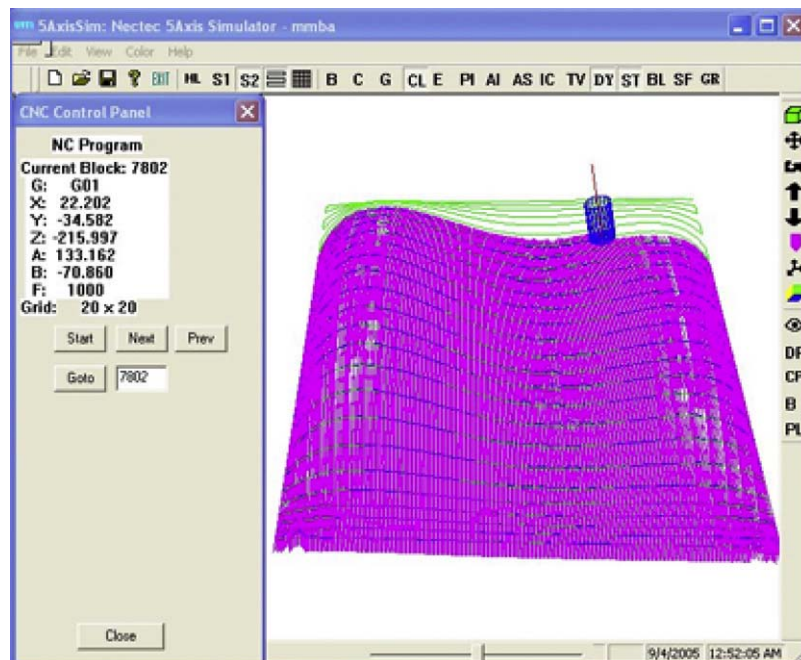


Fig. 3. Zigzag tool path and the machining strips in the workpiece coordinate system.

- The global accessibility constraints. The constraint ensures against the tool colliding with either machine parts or unwanted parts of the desired surface.

1.2. Space filling curves and their applications

Some formal definitions of the space filling curves (SFC) and their short history are presented below.

Definition 1. 1 A function f from a domain X to a codomain Y is said to be surjective if its values span its whole codomain; that is, for every y in Y , there is at least one x in X such that $f(x) = y$.

Definition 2. A function f from a domain X to a codomain Y is said to be bijective if for every y in Y there is exactly one x in X such that $f(x) = y$.

Definition 3. An N -dimensional SFC is a continuous, surjective (onto) function from the unit interval $[0, 1]$ to the N -dimensional unit hypercube $[0, 1]^N$. In particular, a 2-dimensional space-filling curve is a continuous curve that passes through every point of the unit square $[0, 1]^2$.

The history of space-filling curves started in 1878 when George Cantor (1845–1918) demonstrated that any two smooth manifolds of arbitrarily finite dimensions have the same cardinality. Cantor's finding implies that the unit line segment $[0, 1]$ can be mapped bijectively onto the unit square $[0, 1]^2$. In 1879, Eugen Netto (1848–1919) demonstrated that such mapping is necessarily discontinuous and cannot be called a curve. Given that the condition of bijectivity is neglected, in 1890 Giuseppe Peano (1858–1932) found a continuous map from an interval onto a square. This was the first example of a space-filling curve (see Fig. 4(a)). Further examples were introduced by D. Hilbert (in 1891, see Fig. 4(b)), E.H. Moore (in 1900), H. Lebesgue (in 1904), W. Sierpinski (in 1912), G. Polya (in 1913) (see ref. [31]). The SFCs are encountered in different fields of computer science, especially where it is important to linearize multidimensional data. Examples of multidimensional data are matrices, images, tables and computational grids resulting from the discretization of partial differential equations. Typical applications of SFCs are data indexing [21,27], data storing and retrieving [33], image processing [36,40], image scanning and coding [10,11,38], mesh partitioning and reordering [32], etc.

With the variety of space-filling curves and the wide spread of multidimensional applications, the selection of the appropriate space-filling curve for a certain application is not a trivial task. According to the classification in ref. [3] space-filling curves are classified into two categories: recursive and nonrecursive. Examples of recursive SFCs are the Peano's curve and the Hilbert's curve. Most of the existing applications employ the recursive SFCs which allow for the linearization of recursive hierarchical data structures. One of the most favorable properties of SFCs is their locality (SFC never leaves a region at any level of refinement before traversing all points of that region) and the fact that the linearization is easily computable.

1.3. Space-filling curves and tool paths

The most popular SFC for tool path planning is the recursive Hilbert's curve [19] considered for numerous applications including the tool path planning [14]. Cox et al. [9] used various forms of space-filling curves, such as the Moore's curve, for tool path generation. Nevertheless, Hilbert's curve is still particularly appealing in tool path planning as its local refinement property can be used to adaptively to increase the density of the path only where necessary. However, each local refinement of the tool path based on the Hilbert's curve increases the tool path density in the refined region by a factor of 2 resulting in lower machining efficiency due to the increased total path length. Besides, the Hilbert's curve has an undesirable property that it leads to a path, where the tool is frequently changing directions which slows down the machining process and produces large kinematics errors. To overcome these drawbacks Anotaipaiboon and Makhanov [1,2] proposed the use of an adaptive SFC characterized by the following features. First of all, the adaptive SFC always follows the local optimal direction. Second, as opposed to the conventional SFC, the adaptive SFC turns only when necessary, in other words, only when the optimal direction changes. Third, the adaptive SFC eliminates the large kinematics errors and the overcuts appearing due to the sharp angular turns.

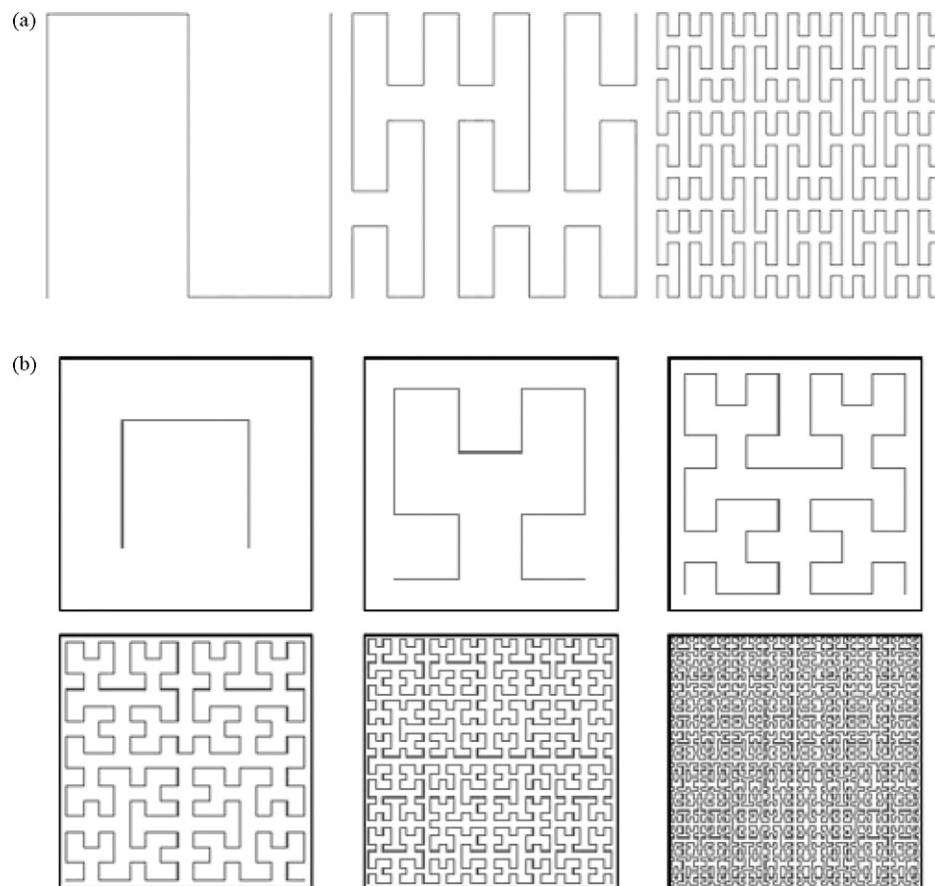


Fig. 4. (a) 3 iterations of the Peano's space-filling curve and (b) 6 iterations of the Hilbert's space-filling curve.

Finally, local refinement of the adaptive SFC is accomplished in exactly the same fashion that the conventional SFC is refined.

The proposed adaptable SFC tool path generation method requires four steps.

- Construction of a basic rectangular grid.
- Generation of the adaptive space-filling tool path on the grid.
- Correction of the tool path.
- Inserting additional points along the path to reduce the kinematics error. The SFC is constructed as a Hamiltonian path on a grid-like graph using a cover and merge algorithm [2,11].

1.4. The new curvilinear space filling curve approach for tool path generation

The basic rectangular grid used to construct the adaptable SFC in [2] is often inefficient since a small step between the tracks could be required only in certain areas. The grid is also inefficient in treating complex geometries appearing in the case of the so-called trimmed surfaces having the boundaries created by intersections with other surfaces.

On the other hand the above geometrical complexities and sharp variations of the surface curvature have been proven to be successfully treated by numerically generated curvilinear zigzag tool paths obtained from adaptive grids topologically equivalent to the rectangular grids. In refs. [24,25] a modification of a classic grid generation method based on the Euler–Lagrange equations for Winslow functional [37] has been adapted to the curvilinear zigzag tool path generation. The zigzag tool path is constructed by solving numerically Euler–Lagrange equations for a functional representing desired properties of the grid such as smoothness, adaptivity to the boundaries and to a certain weight

(control) function [7]. A similar idea to use a Laplacian curvilinear grid was suggested independently by Bieterman and Sandstrom in ref. [4]. In the framework, proposed in refs. [5,24–26] the grid is adapted to the kinematics error subject to constraints relevant to the heights of scallops between the successive tool tracks. However, these techniques have several major drawbacks. Chief among them is slow convergence for complicated constraints. Besides, the approach requires an equal number of the cutter contact points on each track of the tool. Therefore, if the kinematics error changes sharply from track to track, the method may require an excessive number of points.

This paper introduces a new modification of the grid refinement which fits better in the framework of tool path optimization and is designed specifically for the SFC generation. The method does not require equal number of points on each track. It automatically evaluates the number of the required grid lines. As opposed to the preceding approach, where the weight function represents either the kinematics error or an estimate of the kinematics error (such as the surface curvature or the rotation angles), the proposed algorithm iteratively constructs an adaptive control function designed to represent the scallop height constraints. Additionally, instead of the Winslow functional the new optimization is based on the harmonic functional derived from the theory of harmonic maps [16]. The functional not only provides the smoothness and the adaptivity but under certain conditions guarantees the numerical convergence [17]. Finally, this approach merges with the SFC techniques. In this case, the grid is not converted to the tool path directly. Instead, it becomes the basic grid required for the SFC generation. With this modification, the curvilinear space filling curve (CSFC) tool path can be constructed for surfaces with complex irregular boundaries, cuts off, pockets, islands, etc. Besides, the adaptive grid allows to efficiently treat complex spatial variability of the constraints in such a way that the SFC is created on a grid having the small cells only where necessary.

The combination of the two techniques is superior with regard to the case when the two methods are applied independently. A variety of examples is presented when the conventional methods are inefficient whereas the proposed algorithms allow constructing the required tool path with the length close to the minimal. The numerical experiments are complemented by the real machining as well as by the test simulations on the Unigraphics 18. Finally, although elegant and intellectually appealing, grid generation methods are computationally costly, in many cases, requiring many hours of computing. The use of such methods is well justified only for regional milling for complex shaped surfaces with sharp variations in curvature.

2. Grid generation method

Let $S \equiv S(u, v) \equiv (x(u, v), y(u, v), z(u, v))$ be a surface to be machined, where u and v are the parametric variables. Consider a set of cutter location points $\{u_{i,j}, v_{i,j}\}$ arranged as a curvilinear grid. Mathematically, it means that $(u_{i,j}, v_{i,j})$, $0 \leq i \leq N_\xi, 0 \leq j \leq N_\eta$ is a discrete analogy of a mapping from the computational region $\{0 \leq \xi \leq N_\xi, 0 \leq \eta \leq N_\eta\}$ onto a parametric region defined in the parametric coordinates u, v . In other words, there exists a pair of functions $\{u(\xi, \eta), v(\xi, \eta)\}$ such that the rectangular grid i, j being fed to $\{u(\xi, \eta), v(\xi, \eta)\}$ becomes $\{u_{i,j}, v_{i,j}\}$ (see Fig. 5).

2.1. The harmonic functional

The required grid is a discretized solution the following minimization problem:

$$\min_{u,v} I \equiv \min_{u,v} \int \frac{(u_\xi^2 + u_\eta^2)(1 + f_u^2) + (v_\xi^2 + v_\eta^2)(1 + f_v^2) + 2f_u f_v (u_\xi v_\eta + u_\eta v_\xi)}{(u_\xi v_\eta - u_\eta v_\xi) \sqrt{1 + f_u^2 + f_v^2}} d\xi d\eta, \quad (1)$$

where subscripts u, v, ξ, η denote partial derivatives and f is the control function. The harmonic functional I is a generalization the Winslow functional to the case of grids lying on the surface $f(u, v)$. The harmonic functional is derived from the theory of harmonic maps [16]. It has been proven that the functional minimizes an “energy of mapping” [35] and produces a grid adapted to the regions of large gradients of f . Note that if $f_u = f_v \equiv 0$, then the harmonic functional becomes the Winslow functional, however, it is important that I adapts the grid to the *gradients* of f rather than to f itself as in ref. [24].

It is known that minimization of Eq. (1) could be computationally expensive as compared with minimization of the Winslow functional [8]. However, it has many points in its favor. In particular, it is possible to construct a computational procedure which, under certain conditions, converges to a *non-degenerate grid* [35], that is, the grid without twisted or non convex cells. The constraint minimization of Eq. (1) can be performed by using efficient penalty type techniques

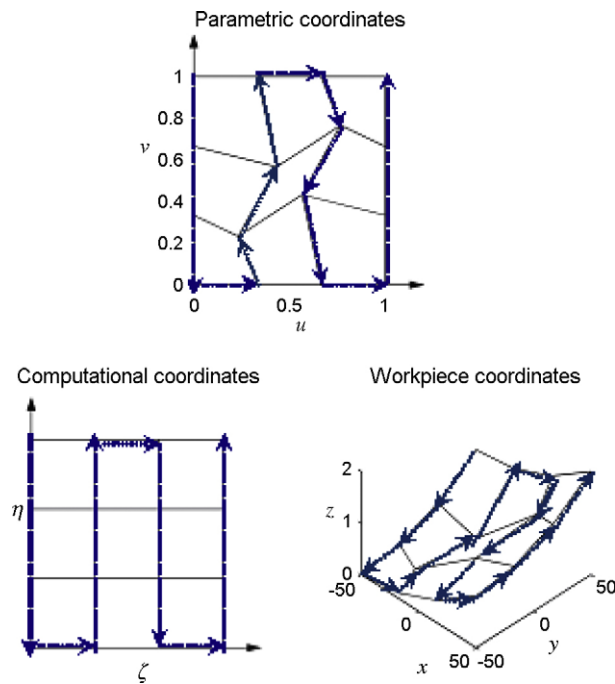


Fig. 5. Grids and the curvilinear zigzag tool path.

similar to those presented in ref. [24]. Finally, the algorithm based on Eq. (1) is more reliable and converges for sharp variations of the input data whereas the Winslow functional often produces degenerated grids.

2.2. Control function for tool path optimization

Since the tool path is a discrete set of points, the derivatives of the control functions $f_u(u, v)$ and $f_v(u, v)$ in (1) for a given surface can not be explicitly evaluated. Therefore, these derivatives are generated “artificially” as follows. First $(f_u)_n^0 \equiv (f_v)_n^0 \equiv 0$. Next,

$$(f_u)_n^{l+1} = \begin{cases} (f_u)_n^l + \lambda_+, & \text{if } s_u(n) > 0, \\ (f_u)_n^l - \lambda_-, & \text{otherwise,} \end{cases}$$

$$(f_v)_n^{l+1} = \begin{cases} (f_v)_n^l + \lambda_+, & \text{if } s_v(n) > 0, \\ (f_v)_n^l - \lambda_-, & \text{otherwise,} \end{cases}$$

where λ_+ and λ_- is the prescribed increment and decrement, respectively, n is the grid node number, l is the iteration number and $s_d(n)$ is the difference between the actual distance between the tracks and the machining strip defined by:

$$s_u(n) = \max_{d \in \{\text{left}, \text{right}\}} W(n, N'(n, d)) - T(n, N'(n, d)),$$

$$s_v(n) = \max_{d \in \{\text{up}, \text{down}\}} W(n, N'(n, d)) - T(n, N'(n, d)),$$

where $N'(n, d)$ is the set of neighboring nodes to n and $W(n, m)$ is the distance between nodes n and m given by

$$W(n, m) = |S((u, v)_n) - S((u, v)_m)|.$$

Finally, $T(n, m)$ is an estimate of the machining strip at midpoint $S(u_n + u_m/2, v_n + v_m/2)$ (see Sections 2.4 and 2.5).

2.3. Inserting additional tracks

The initial grid does not (and should not) satisfy the scallop height constraint. However, it is often the case that additional tracks must be inserted for convergence. For a structured grid having initially n_r rows and n_c columns, the number of rows and columns at the next step is evaluated as follows

$$n_{r,new} = n_r + n_{r,add},$$

$$n_{r,add} = \frac{\max_{1 \leq i \leq n_r} \sum_{j=1}^{n_c-1} (W(n_{i,j}, n_{i,j+1}) - T(n_{i,j}, n_{i,j+1}))}{2r}, \quad (2)$$

$$n_{c,new} = n_c + n_{c,add},$$

$$n_{c,add} = \frac{\max_{1 \leq i \leq n_c} \sum_{j=1}^{n_r-1} (W(n_{i,j}, n_{i,j+1}) - T(n_{i,j}, n_{i,j+1}))}{2r}, \quad (3)$$

where $n_{i,j}$ is the grid node and r is the tool radius. It should be noted that if the grid is constructed to produce a curvilinear zigzag tool path in one direction, then only one from the two formulas (2) and (3) must be applied. However, if the grid is needed for the CSFC generation, they must be applied in the both directions. Finally, (2) and (3) may overestimate the number of the required tracks. Consequently, it can be replaced by

$$n_{new} = n + n_{add}\alpha_{rel},$$

where $\alpha_{rel} < 1$ is a “the rate of release” of the additional curves. The “rate of release” is determined experimentally. Such a procedure may lead to a decrease in the number of the zigzag curves, thus, improving the efficiency of the machining. An inexperienced user is safe with $\alpha_{rel} = 1/n_{add}$ which, however, may lead to an increase in the computational cost.

2.4. Machining strip evaluation

Given the maximum allowable scallop height h_{max} , the distance between the tool tracks is found by computing the machining strip width.

Introduce a local coordinate system (O_l, x_l, y_l, z_l) at the CC (cutter contact) point O_l shown in Fig. 6, where x_l denotes the normalized projection of the tool cutting direction onto the tangent plane, z_l denotes the surface normal vector, and $y_l = z_l \times x_l$. The tool is rotated by an inclination angle λ about the y_l axis, then by a tilt angle ω about the z_l axis. The projected bottom edge of a flat-end cutter with radius r onto the (y_l, z_l) -plane becomes an ellipse called the effective cutting shape. In order to evaluate the machining strip, the surface cross-section perpendicular to the tool cutting direction x_l is approximated by a circular arc, for which radius R_y is equal to the radius of the normal curvature of the surface in the y_l direction as shown in Fig. 6. Suppose that $h = h_{max}$. The maximum machined surface error is represented by a virtual circular arc with radius $R_y - h$ as shown in Fig. 7. The machining strip width is then obtained by finding intersections of the effective cutting shape with the virtual arc.

Let P be an arbitrary point on the cutter bottom edge (see Fig. 7). Consider an angle θ required to turn the y_c -axis around the z_c axis in such a way that the negative y_c -axis passes through P . Furthermore, angles corresponding to the left and the right intersections P_l and P_r are denoted by θ_l and θ_r respectively. It is not hard to demonstrate that the left and the right machining strip w_l and w_r are then given by

$$w_\mu = r|\cos \omega \cos \theta_\mu - \cos \lambda \sin \omega(1 - \sin \theta_\mu)|,$$

where $\mu = l$ or $\mu = r$.

The entire machining strip width is then

$$w = w_l + w_r.$$

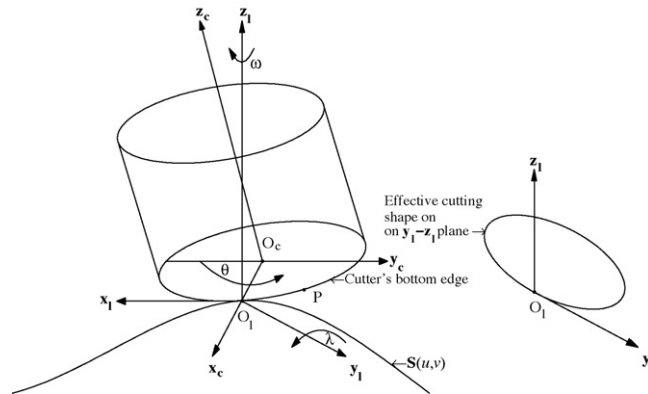


Fig. 6. Geometric analysis of the cutter.

2.5. Tool orientation and gouging

The effective cutter radius r_e is given by

$$r_e = ra^2 \left(\frac{1 + b^2}{a^2 + b^2} \right)^{3/2},$$

where $a = \sin \lambda \cos \omega$, $b = \tan \lambda \sin \omega$, (see ref. [23]).

To optimize the machining strip width, λ and ω are usually set so that r_e is the best match to the radius of curvature at the CC point. For convex or planar surfaces, the tool inclination angle λ is set to a small default angle or zero and the tilt angle ω is set to zero as well. If the surface is non-convex, a non-zero tool inclination angle λ is needed to avoid gouging. Consider a flat-end cutter shown in Fig. 8. Gouging occurs whenever a point on the circle touches or goes inside the surface. Let G be a gouging point (Fig. 8(a)). The line connecting the two points, O_l and G , forms a chord on the circle. Denote the angle between this line and $\overline{O_l O_2}$ by ϕ (see Fig. 8(a and b)). Let λ_ϕ be the tool inclination angle that corresponds to a specific ϕ . The minimum tool inclination angle to avoid gouging is then

$$\lambda_{\min} = \max_{-\pi/2 \leq \phi \leq \pi/2} \lambda_\phi.$$

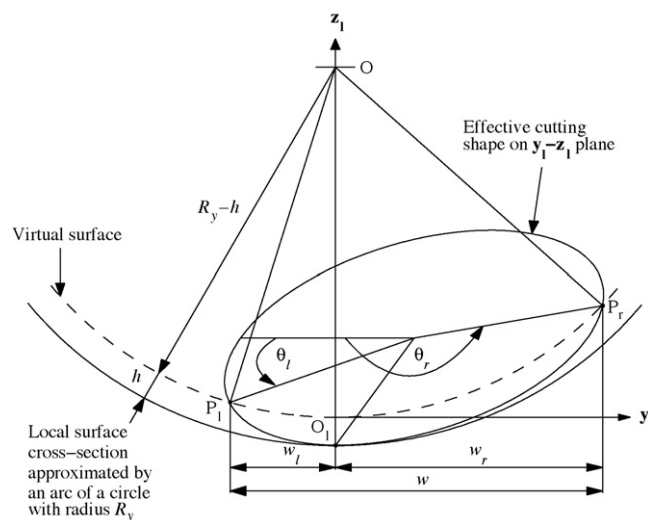


Fig. 7. Machining strip width estimation.

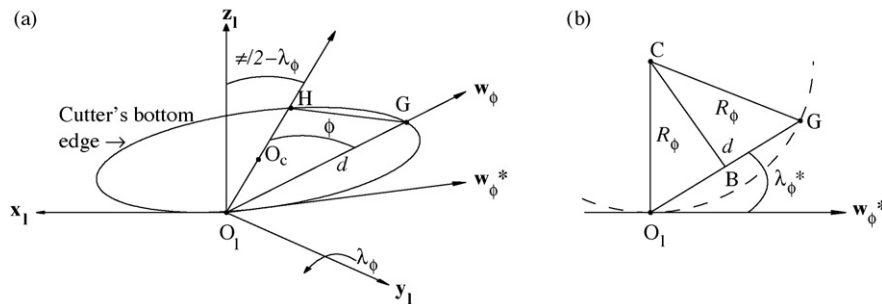


Fig. 8. Tool gouging.

It is not hard to demonstrate that for a non-convex surface

$$\lambda_{\min} = \sin^{-1}(rk_{\max}),$$

where $k_{\max}$ is the maximum surface curvature at the CC point. Clearly, for a convex surface an inclination is not required, so $\lambda_{\min} = 0$, however, from technological viewpoints a small inclination angle is still recommended. Furthermore, for $\forall \lambda < \lambda_{\min}$ the gouging will be avoided as well. It also can be shown [23] that for the flat end (cylindrical) cutter the orientation $(\lambda, \omega) = (\lambda_{\min}, 0)$ maximizes the machining strip. If gouging can not be eliminated by inclining the tool alone or the inclination angle λ requires rotations which exceed the limit of the machine, the tilt angle can be optimized or a smaller tool size must be used.

2.6. The algorithm

The grid generation algorithm consists of the following steps

- (1) Generate an initial convex grid. The grid is generated manually or by interpolating. Note that interpolation may generate a grid with the nodes outside the boundary of the region. In that case, several iterations can be performed by the classic Brackbill's and Saltzman's method [7] which will move the nodes back inside the region. The initial grid should *not* satisfy the scallop constraints, because, if it does, the adaptation is not necessary. It also means that the number of grid nodes could be reduced.
- (2) Insert additional nodes using (2) and (3).
- (3) Adapt the grid by numerically minimizing functional Eq. (1) until all the grid points satisfy the scallop constraint or until a prescribed number of iterations has been exceeded.
- (4) If the scallop constraint has not been satisfied for all the points, *goto the refinement stage 2*. Fig. 9 illustrates the adaptive harmonic grid generation applied to a simple surface characterized by a large gradients along a sinus shaped zone (Fig. 9(a)). The required small machining strips generate the control function depicted in Fig. 9(b) which in turn produces a grid adapted to the control function depicted in Fig. 9(c).

2.7. Composite surfaces

The techniques above work for a single parametric surface, however, the industrial parts are usually represented by surfaces composed from the Bezier or NURBS patches [12,28,6]. The NURBS are supported by one of the most popular formats called the IGES (the Initial Graphics Exchange Specification). There exists a variety of other CAD formats and representations such as STL, STEP, SLC, DXF, etc. However, the commercial CAD/CAM systems usually provide conversion between the major data formats (see, for instance, <http://www.actify.com/v2/products/Importers/formats.htm>). Usually, the compound NURBS surface is defined as multiple patches whose boundaries are generated in trimming and/or intersecting manipulations and which are joined together with C^0 , C^1 , or C^2 continuity. Special techniques exist to connect those patches smoothly and automatically (see, for instance ref. [18]). Fortunately, the CAD/CAM systems provide all information about each patch, including its external and internal boundaries, both in the part

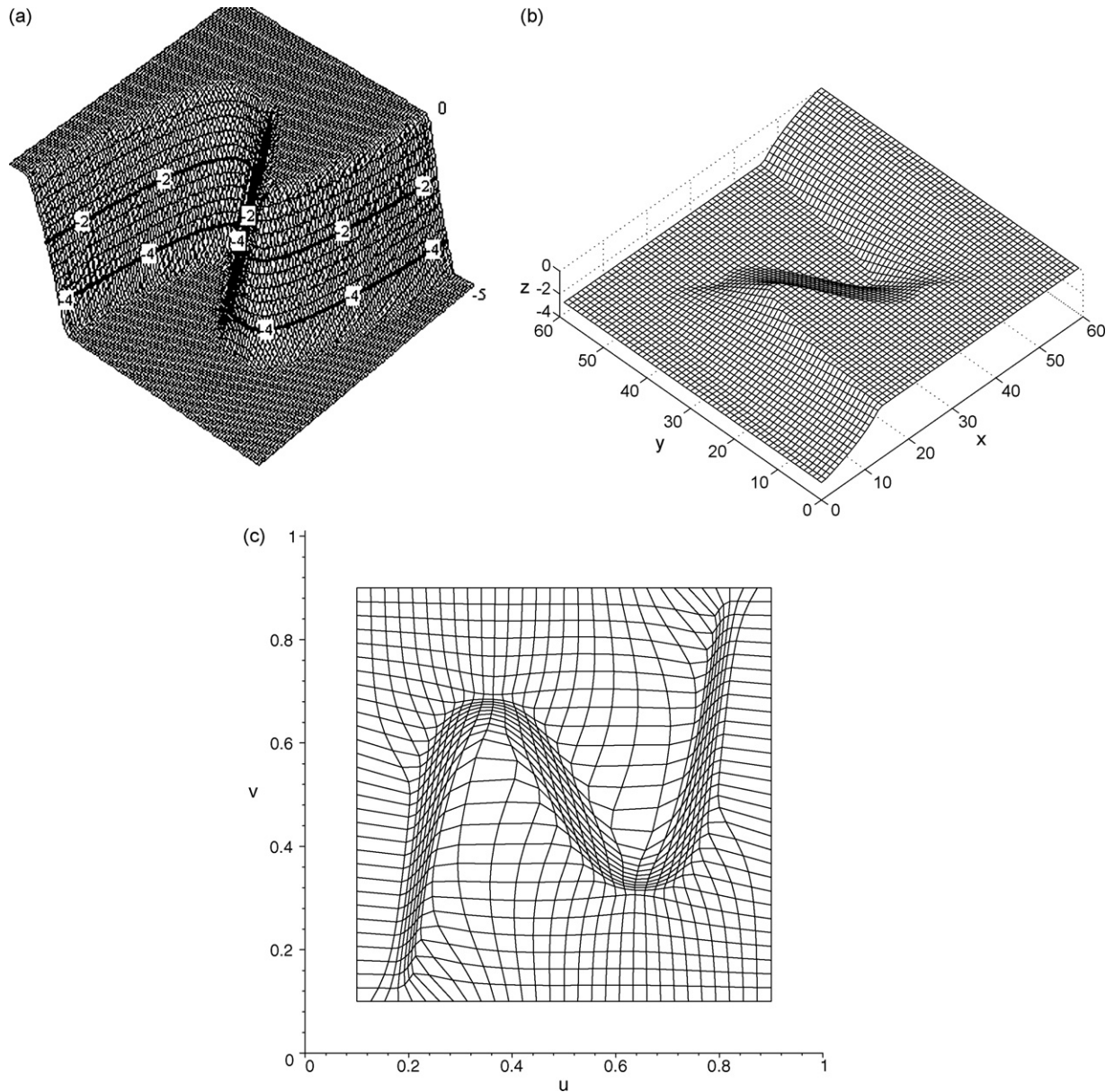


Fig. 9. Adaptive harmonic tool path generation (a) the surface, (b) the control function, (c) the grid.

coordinate system and the parametric space. In many cases, the part surface is composed from a structured grid of multi-block patches. In this case the composed surface can be re-parametrized in such a way that global parametric coordinates can be introduced across the entire surface. Alternatively, if the patch data is not structured, the commercial CAD/CAM systems often make it possible to reduce or increase the number of patches to create a structured multi-block geometry without a significant loss of accuracy. Finally, if such correction is not applicable, a variety of multi-block strategies developed for general purpose grid generation can be adapted, see for instance, ref. [35] or [34]. However, even in this case, in spite of obvious technical problems, the main ideas proposed in this paper, namely, the scallop based Dirichlet curvilinear grid, artificially generated derivatives of the control function and the Hamilton path for generation of the SFC apply irrespectively of an underlying surface representation.

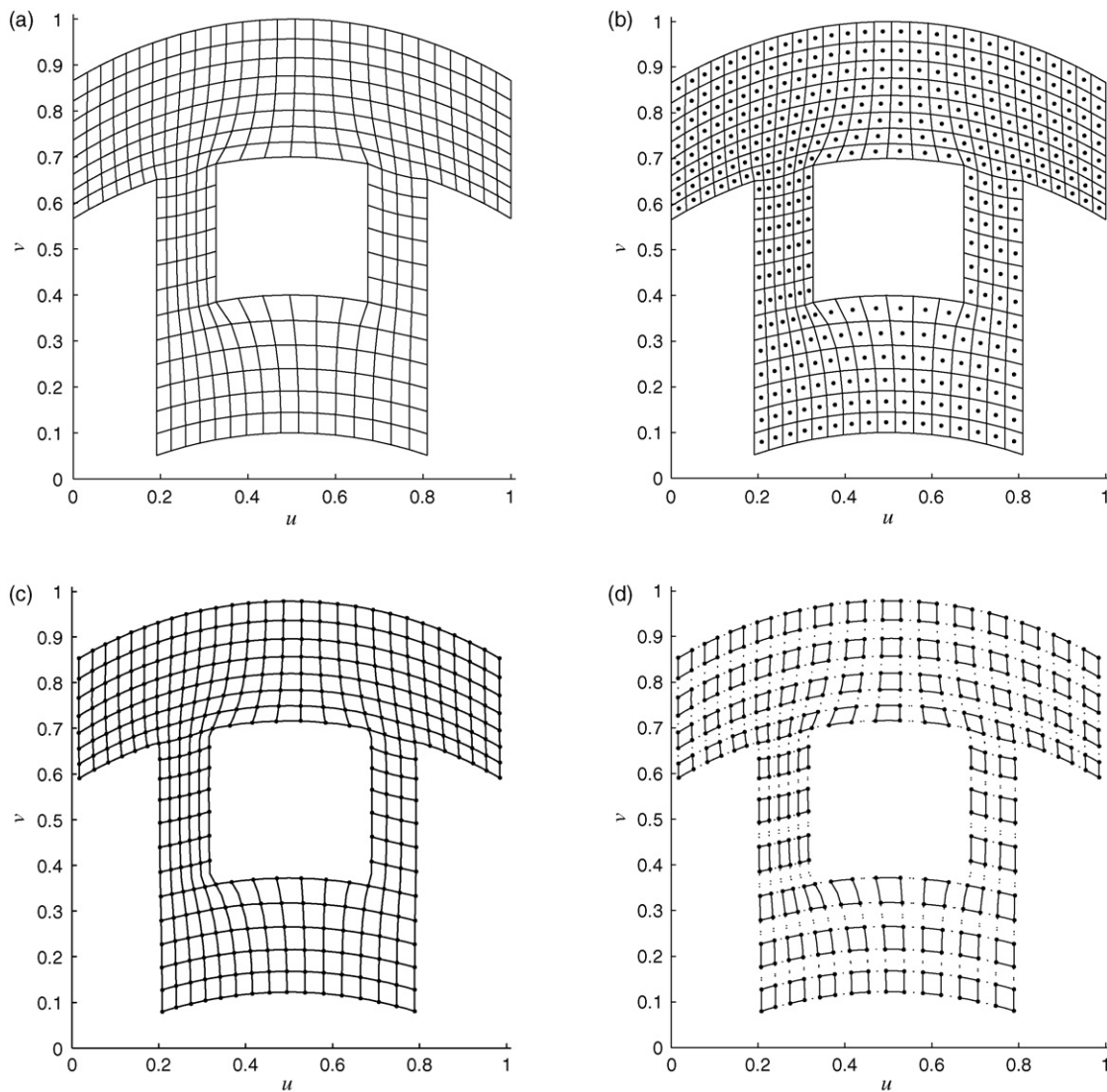


Fig. 10. (a–c) Construction of an undirected graph for the CSFC tool path generation. (d) Generation of an initial set of circuits for constructing the CSFC.

3. Tool path generation using curvilinear space filling curve method

As an example consider a basic curvilinear grid generated using the algorithm developed in Section 2 (Fig. 10(a)).

In order to construct the CSFC, each grid cell is replaced by a vertex in the middle of the cell (Fig. 10(b)). Every pair of adjacent vertices is then connected by an edge as shown in Fig. 10(c and d) to create an initial set of small circuits. Note that vertices of the graph correspond to initial set of CC points on the required surface. Therefore, the distance between two connected vertices is defined as the distance between the corresponding CC (cutter contact) points on the surface in R^3 . A cut along the path between any two connected vertices satisfies the scallop height constraint. This feature allows for the tool path optimization by means of the SFC. The SFC tool path generation algorithm is presented next.

The tool path generation on the grid-like graph constructed is a particular case of the travelling salesman problem called the Hamiltonian path problem [30]. Since the problem is NP-hard [15], the algorithms for finding the optimal solution are slow and inefficient.

A simple and computationally efficient algorithm for producing the Hamiltonian path based on the cover and merge algorithm was developed by Dafner et al. [11] for 2-dimensional image scanning. This paper extends the algorithm for non rectangular domains and block structured grids and applies it to the CSFC tool path generation which works as follows. First, all vertices are covered by small disjoint circuits. The circuits are then merged into a single Hamiltonian circuit. The initial circuits are created by constructing small rectangular cyclic paths over every 4 adjacent vertices,

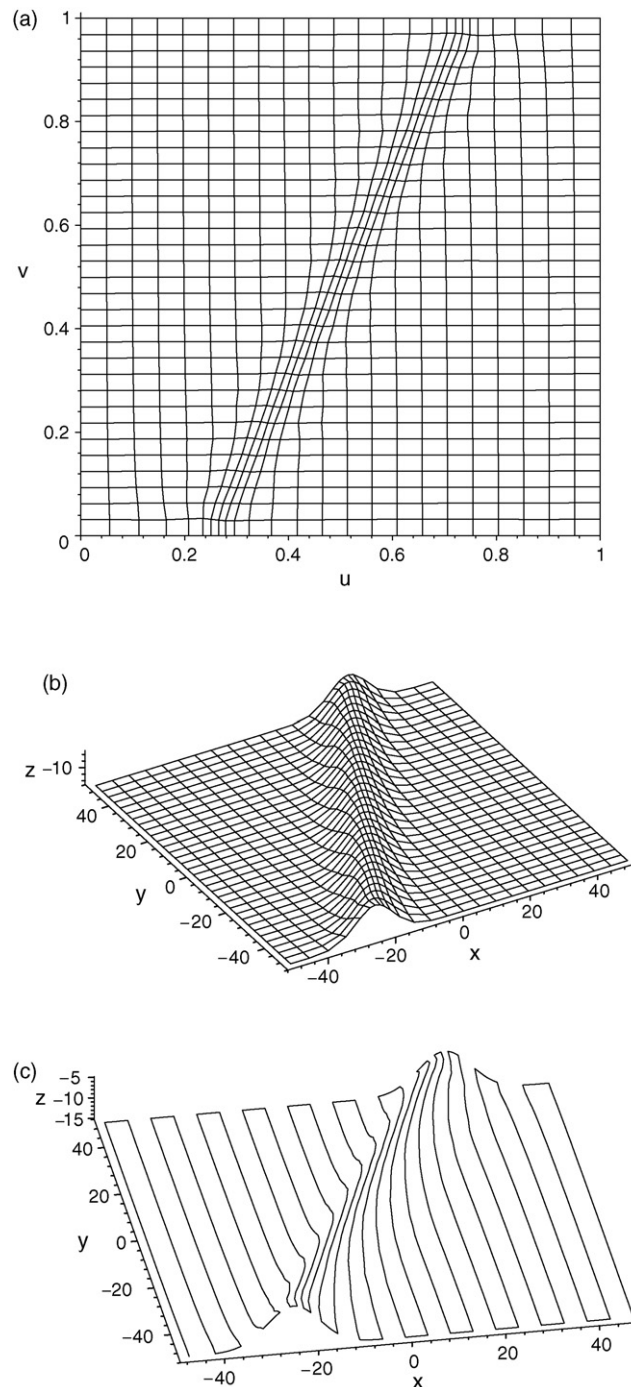


Fig. 11. Curvilinear grid adapted to the unimodal surface which exponential peak along a line in (a) $u-v$ domain and (b) grid on the surface in the workpiece coordinate systems, (c) the corresponding curvilinear space filling curve.

i.e., by connecting the vertices on even rows and columns with the vertices on odd rows and columns, respectively. For structured grid, if the grid size is odd, virtual circuits are constructed to cover the vertices along the boundaries. Any two adjacent circuits can be merged into one bigger circuit. The cost of merging is defined by

$$\text{Cost}(A, B) = |s| + |t| - |e| - |f|, \quad (4)$$

where $|e|$ denotes the distance between two vertices connected by edge e . The cost of merging of two virtual circuits is set to $-\infty$, i.e., all the virtual circuits are initially merged. This is to ensure that there is no discontinuity of the tool path after removing the virtual edges from the Hamiltonian path. To merge all small circuits, we construct a dual graph and a minimum spanning tree is constructed by iteratively merging circuits according to Eq. (4).

4. Examples and practical machining

This section demonstrates the efficiency and advantages of the use of the proposed CSFC tool path generation by examples and practical machining.

4.1. Example 1. A unimodal surface

The first example demonstrates the efficiency of the CSFC with the reference to the traditional iso-parametric tool path method. Consider a unimodal surface which exponential peak along a line in the parametric domain (u, v) given by (see Fig. 11)

$$x = 100u - 50,$$

$$y = 100v - 50,$$

$$z = 10 \exp -40(2u - 0.5 - v)^2 - 15.$$

For flat-end tool of radius 3 mm and machined surface tolerance of 0.1 mm, the final curvilinear grid is shown in Fig. 11. The comparison of the zigzag and SFC tool paths generated from traditional iso-parametric tool path and curvilinear grid is presented in Table 1. The length of the zigzag and SFC tool paths based on the adaptive grid are shorter by 45.76% and 17.84% respectively, when compared with the zigzag and SFC based on iso-parametric tool path method.

4.2. Example 2. Curvilinear boundaries and pocket milling

This example demonstrates the use of the CSFC to construct tool paths to machine surfaces with complex irregular boundaries, cuts off, and islands. Consider a surface shown in Fig. 12(a and b) shows the basic curvilinear grid constructed using the proposed method. Fig. 12(c) shows the CSFC and Fig. 12(d) the CSFC on the surface. Finally, Fig. 12(e) shows the machining result obtained with the use of the solid modeling engine of the Unigraphics. The surface has been machined by a flat-end tool of radius 3 mm and the machined surface tolerance of 0.05 mm. Consequently, the method is capable of creating tool path for surfaces with complex non rectangular boundaries and islands.

4.3. Example 3. Point milling of an impeller blade

Frequently, the blades of industrial impellers are produced by the so-called five-axis *swarf* milling made by a side of the tool. In this case the contact between the workpiece and the cutter is characterized by a contact line rather than a

Table 1
Comparison of the methods in terms of the tool path length.

Method used for constructing the basic grid	Total path length (mm)	
	Vertical zigzag	SFC
Iso-parametric tool path	4716.91	3066.87
Curvilinear grid generation	2557.74	2508.25

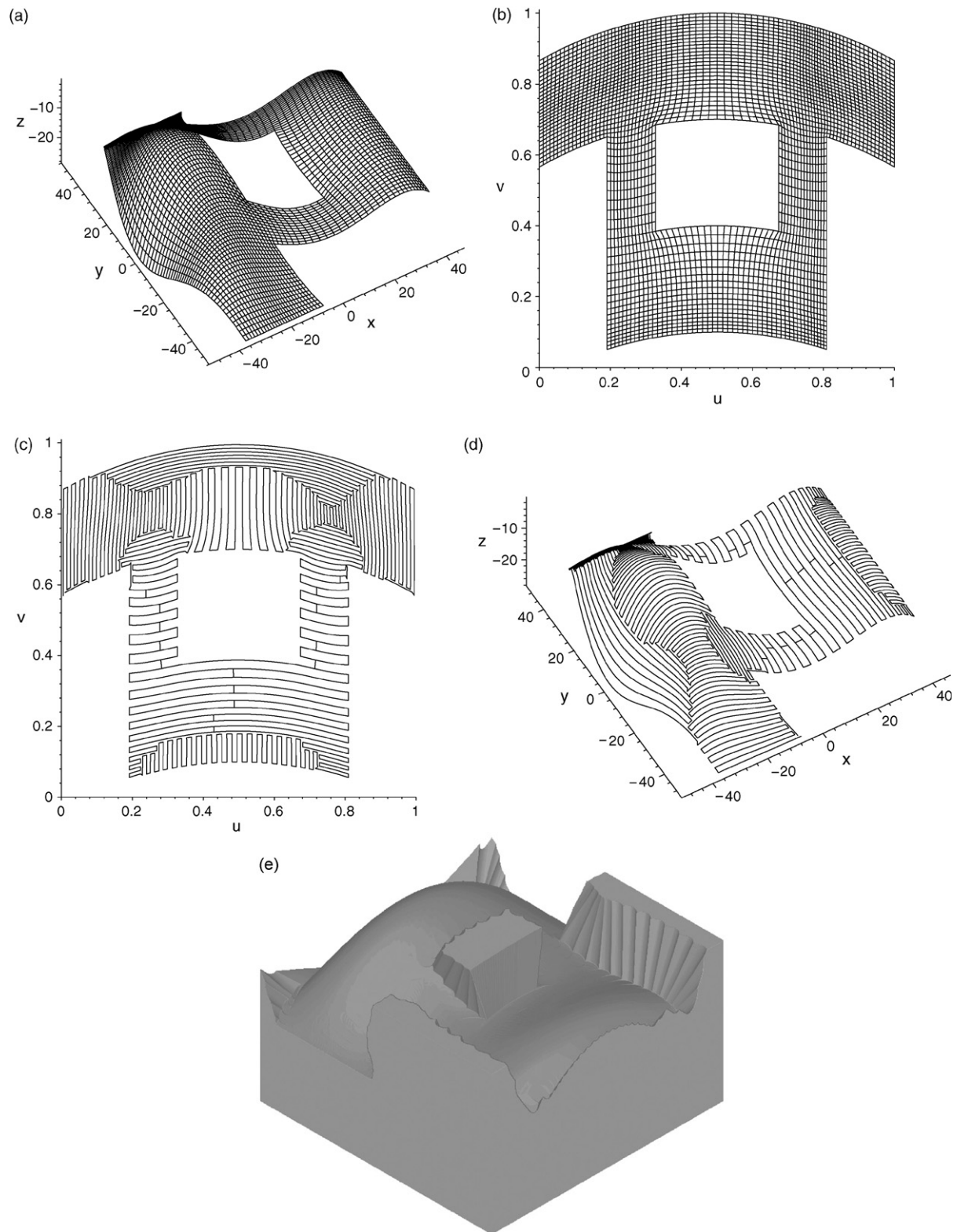


Fig. 12. Constructing the curvilinear space filling curve. (a) The surface, (b) the curvilinear grid, (c) the space filling curve, (d) the space filling curve on the surface and (e) virtual machining, Unigraphics.

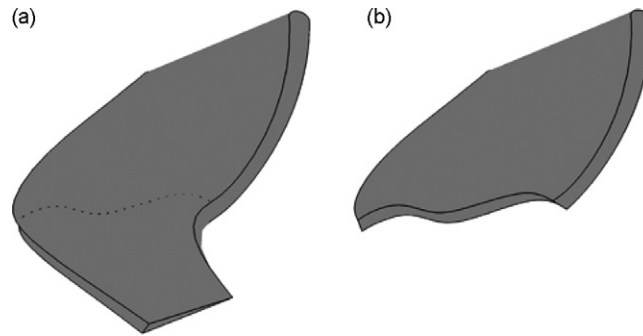


Fig. 13. (a) A broken blade and (b) the missing part of the blade.

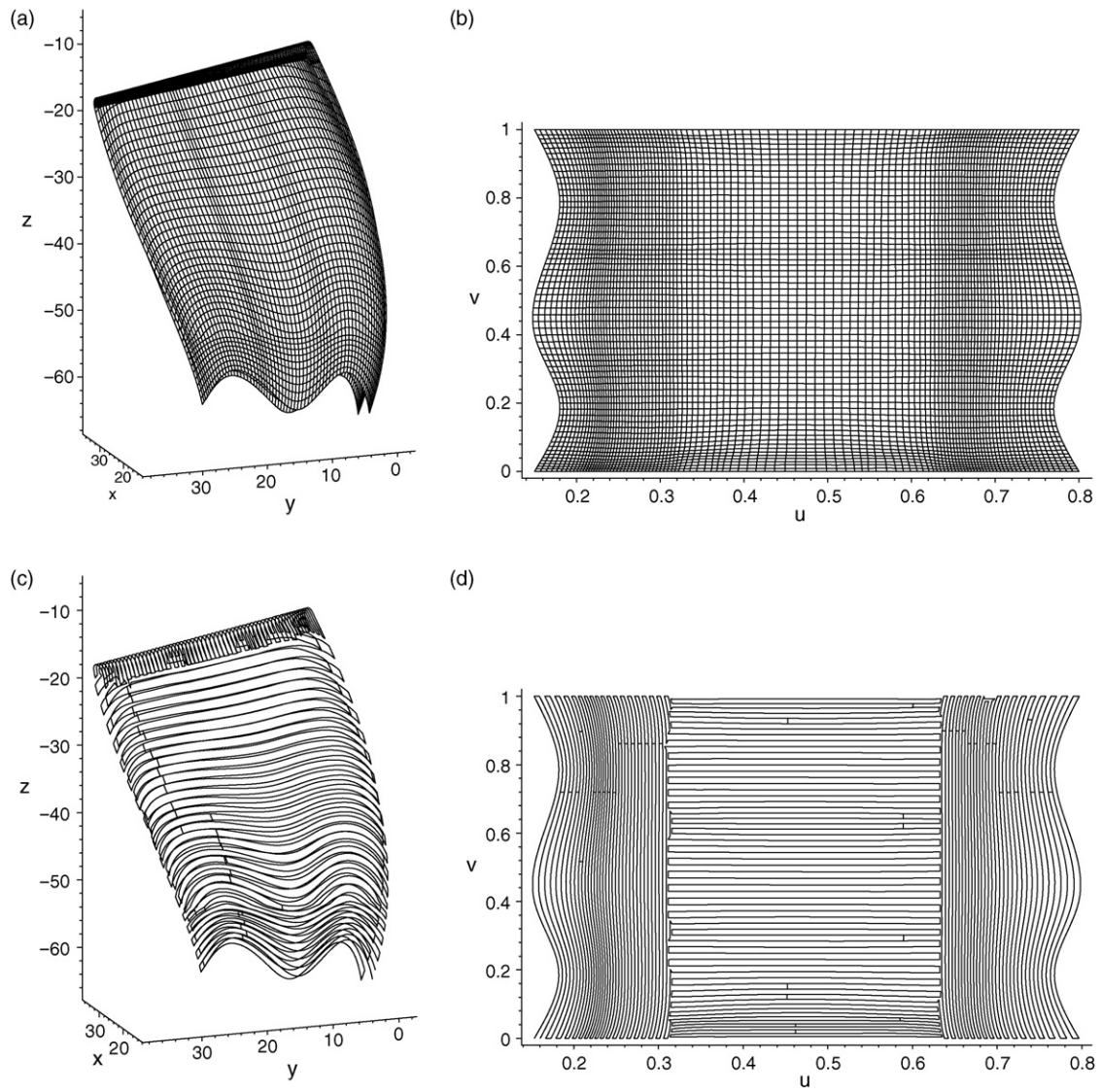


Fig. 14. Curvilinear grid adapted to part of a surface of the blade (a) on the surface (c) in the $u-v$ domain, (b) CSFC on the surface, (d) CSFC in the $u-v$ domain.

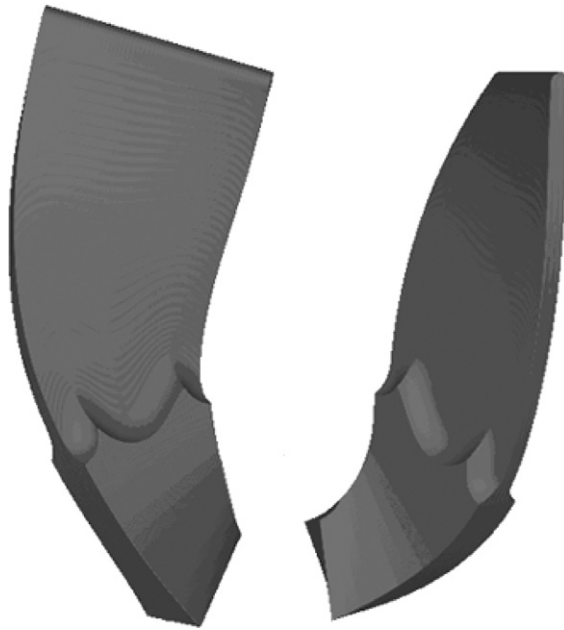


Fig. 15. The blade restoration. Simulation in the Unigraphics.



Fig. 16. The blade restoration. Actual machining.

contact point. However, this technique may lead to large errors. This example demonstrates machining of the impeller by five-axis *point milling* with the use of curvilinear adaptive space filling curves.

In order to demonstrate the advantages of the proposed method, the geometrical complexity of the example blade is increased as follows. Consider a blade depicted in Fig. 13(a). After long-hours serves in a harsh environment the blades may suffer from a variety of defects, such as distortion, cracks, nicks and dents. Suppose that the blade is broken as shown in Fig. 13(a) (the dashed line) and requires a restoration. The missing part is shown in Fig. 13(b). Note that similar (but smaller in size) restorations through the reverse engineering techniques are described, for instance, in ref. [13]. The tool path must be generated using the shape and the boundary of the repair volume to reduce the machining time. It will be shown that the CSFC method generates the tool path which follows exactly the boundary of the region being restored.

Our basic curvilinear grid adapted to the shape of the blade is shown in Fig. 14(a). For a ball-end tool of the radius 3 mm and the surface tolerance of 0.05 mm, the SFC tool path is shown in Fig. 13(b). The corresponding grid and the SFC tool path in the parametric region are shown in Fig. 14(c and d).

Finally, the virtual cutting using the proposed CSFC tool path is shown in Fig. 15 whereas a real prototype of the blade (wood) is shown in Fig. 16. For demonstration purposes the size of scallops has been chosen so that the CSFC is clearly visible on the surface.

5. Conclusions

Numerically generated adaptive curvilinear grid is introduced to replace the rectangular grid used for construction of the space filling tool path for five-axis machining. With this modification the SFC can be constructed for surfaces with complex irregular boundaries, cuts off, pockets, islands, etc. Besides, the adaptive grid allows to efficiently treat complex spatial variability of the constraints. The combination of the two techniques is superior with regard to the case when the two methods are applied independently.

Acknowledgement

This work is supported by Thailand Research Fund, grant BRG50800012.

References

- [1] W. Anotaipaiboon, S.S. Makhanov, Tool path generation for five-axis NC machining using space-filling curves, in: Proceeding of The Third Asian Conference on Industrial Automation and Robotics, vol. 1, 2003, pp. 23–28.
- [2] W. Anotaipaiboon, S.S. Makhanov, Tool path generation for five-axis NC machining using adaptive space-filling curves, International Journal of Production Research 43 (2005) 1643–1665.
- [3] T. Asano, D. Ranjan, T. Roos, E. Welzl, P. Widmayer, Space-filling curves and their use in the design of geometric data structures, Theoretical Computer Science 181 (1) (1997) 3–15.
- [4] M.B. Bieterman, D.R. Sandstrom, A curvilinear tool-path method for pocket machining, Journal of Materials Processing Technology 125 (4) (2003) 709–715.
- [5] E. Bohez, S.S. Makhanov, K. Sonthipermpon, Adaptive nonlinear tool path optimization for five-axis machining, International Journal of Production Research 38 (2000) 4329–4343.
- [6] C. De Boor, A Practical Guide to Splines, Springer, New York, USA, 2001.
- [7] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, Journal of Computational Physics 46 (1982) 342–368.
- [8] A.A. Charakhchyan, S.A. Ivanenko, A variational form of the Winslow grid generator, Journal of Computational Physics 136 (2) (1997) 385–398.
- [9] J.J. Cox, T. Takezaki, H.R.P. Ferguson, K.E. Kohkonen, E.L. Mulkay, Space-filling curves in tool-path applications, Computer Aided Design 26 (1994) 215–224.
- [10] A.J. Cole, Compaction techniques for raster scan graphics using space-filling curves, The Computer Journal 30 (1) (1987) 87–92.
- [11] R. Dafner, D. Cohen-Or, Y. Matias, Context-based space filling curves, Computer Graphics Forum 19 (3) (2000) 209–218.
- [12] G.E. Farin, NURBS: From Projective Geometry to Practical Use, A.K. Peters Ltd., Natick, MA, USA, 1999.
- [13] J. Gao, X. Chen, D. Zheng, O. Yilmaz, N. Gindy, Adaptive restoration of complex geometry parts through reverse engineering application, Advances in Engineering Software 37 (9) (2006) 592–600.
- [14] J.G. Griffiths, Toolpath based on Hilbert's curve, Advances in Engineering Software 26 (1994) 839–844.
- [15] J.E. Hopcroft, J.D. Ullman, Introduction to Automata Theory, Languages and Computation, Addison-Wesley, New York, 1979.
- [16] S.A. Ivanenko, Generation of non-degenerate meshes, USSR Computational Mathematics and Mathematical Physics 28 (1988) 141–146.

- [17] S.A. Ivanenko, J.F. Thompson, B.K. Soni, N.P. Weatherill, *Harmonic Mappings Handbook of Grid Generation*, vol. 8, CRC Press LLC, 1999, pp. 1–43.
- [18] K. Konno, Y. Yoshimasa Tokuyaman, H. Chiyokura, A G^1 connection around complicated curve meshes using C^1 NURBS Boundary Gregory Patches, *Computer Aided Design* 33 (2001) 293–306.
- [19] W.M. Lam, J.H. Shapiro, A class of fast algorithms for the Peano-Hilbert space-filling curve, in: *Proceedings ICIP-94*, IEEE Computer Society, vol. 1, 1994, pp. 638–641.
- [20] B. Lauwers, P. Dejonghe, J.P. Kruth, Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machining simulation, *Computer Aided Design* 35 (2003) 421–432.
- [21] J.K. Lawder, P.J.H. King, Querying multi-dimensional data indexed using the Hilbert space-filling curve, *ACM SIGMOD Record* 30 (1) (2001) 19–24.
- [22] Y-S. Lee, Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *Computer Aided Design* 29 (1997) 507–521.
- [23] C.-C. Lo, Efficient cutter-path planning for five-axis surface for machining with a flat-end cutter, *Computer Aided Design* 31 (1999) 557–566.
- [24] S.S. Makhanov, An application of the grid generation techniques to optimize a tool-path of industrial milling robots, *Journal of Computational Mathematics and Mathematical Physics* 39 (1999) 1589–1600.
- [25] S.S. Makhanov, S.A. Ivanenko, Grid generation as applied to optimize cutting operations of a five-axis milling machine, *Applied Numerical Mathematics* 46 (2003) 353–377.
- [26] S.S. Makhanov, D. Batanov, E. Bohez, K. Sonthipaumpoon, W. Anotaipaiboon, M. Tabucanon, On the tool-path optimization of a milling robot, *Computers & Industrial Engineering* 43 (2002) 455–472.
- [27] V. Pascucci, R.J. Frank, Global static indexing for real-time exploration of very large regular grids (CDROM), in: *Supercomputing'01*, *Proceedings of the 2001 ACM/IEEE Conference on Supercomputing*, 2001.
- [28] L. Piegl, W. Tiller, *The NURBS Book*, Springer-Verlag, London, UK, 1995.
- [29] A. Rao, R. Sarma, On local gouging in five-axis sculptured surface machining using flat end tools, *Computer Aided Design* (2000) 409–420.
- [30] F.S. Robert, *Applied Combinatorics*, Prentice-Hall, Englewood Cliffs, NJ, 1984, p. 32.
- [31] H. Sagan, *Space-Filling Curves*, Springer-Verlag, New York, 1994.
- [32] S. Schamberger, J.-M. Wierum, Partitioning finite element meshes using space-filling curves, *Future Generation Computer Systems* 21 (5) (2005) 759–766.
- [33] Z. Song, N. Roussopoulos, Using Hilbert curve in image storing and retrieving, in: *MULTIMEDIA'00*, *Proceedings of the 2000 ACM Workshops on Multimedia*, 2000, pp. 167–170.
- [34] B.K. Soni, R. Koomullil, D.S. Thompson, H. Thornburg, Solution adaptive grid strategies based on point redistribution, *Computer Methods in Applied Mechanics and Engineering* 189 (4) (2000) 1183–1204.
- [35] J.F. Thomson, B. Soni, N. Weatherill, *Handbook of Grid Generation*, CRC Press, 1999.
- [36] L. Velho, J. Gomes, Digital halftoning with space filling curves, in: *SIGGRAPH'91*, *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques*, 1991, pp. 81–90.
- [37] A.M. Winslow, Numerical solution of the quasilinear Poisson equation in a non-uniform triangle mesh, *Journal of Computational Physics* 1 (2) (1966) 149–172.
- [38] K.-M. Yang, L. Wu, M. Mills, Fractal based image coding scheme using Peano scanning, in: *Proceedings of 1988 IEEE International Symposium on Circuits and Systems*, 1988, pp. 2301–2304.
- [39] J.-H. Yoon, Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining, *International Journal of Production Research* 41 (2003) 2125–2142.
- [40] Y. Zhang, R.E. Webber, Space diffusion: an improved parallel halftoning technique using space-filling curves, in: *SIGGRAPH'93*, *Proceedings of the 20th Annual Conference on Computer graphics and Interactive Techniques*, 1993, pp. 305–312.

Adaptable geometric patterns for five-axis machining: a survey

Stanislav S. Makhanov

Received: 27 March 2007 / Accepted: 29 July 2009 / Published online: 26 September 2009
© Springer-Verlag London Limited 2009

Abstract The paper presents a survey of five-axis computer numerical controlled (CNC) machining optimization methods employing adaptable geometric patterns. First, the survey introduces evolution of CNC interpolators from the simplest Taylor series-based routines to sophisticated procedures based on constraint minimization from dynamic systems control theory. Furthermore, a variety of methods based on spline interpolation, NURBS interpolation and Farouki's Pythagorean-hodograph curves is presented and analyzed. Next, the survey deals with techniques to optimize the positions and orientations of the tool in a particular neighborhood of the part surface. The most important application of these techniques is cutting by a flat-end or a fillet mill while avoiding local overcuts or undercuts due to the curvature interference and rear gouging. This section is supplemented by detection of global interference using visibility cone schemes and their recent modifications and improvements. Solutions offered by solid modeling are presented as well. Finally, adaptable geometric patterns employed for tool path generation are considered and analyzed. The adaptation is performed using certain criteria of the tool path quality, such as kinematics error, scallops, possible undercuts or overcuts, and the continuity of the path. Also covered are complex pocket milling employing geometric patterns capable of following the boundary, such as the offset methods, regional milling, the potential path methods, and clustering. The chapter also presents tool path optimization based on the adaptable

curvilinear grids connecting the cutter location points. Finally, navigation approaches and the shortest-path schemes are considered, along with the adaptive space-filling curve algorithms and their combinations with grid generation.

Keywords Five-axis machines · Multi-axis machines · CNC interpolators · Tool orientation · Curvature interference · Rear gouging · Solid modeling · Pythagorean-hodograph curves · Pocket machining · Regional milling · Space-filling curves for machining · Curvilinear grids for machining · Navigation approach for machining

1 Introduction

Five-axis numerical controlled (NC) machines are becoming increasingly popular due to their ability to handle geometrically complex workpieces composed of raw material such as wood, wax, rubber, metal, stone, and plastic. Up-to-date five-axis NC machines are characterized by a high material removal rate and an efficient surface finish-up. Typically, manufacturing of the design surface by an NC machine comprises two stages, a rough cutting and a finish machining. During the rough cut, the raw material is removed as fast as possible while ensuring no excessive cutting or gouging. During the finish machining, the tool is placed at the maximum contact with the surface to remove the remaining excess and create a well-finished and accurate surface. After finishing, the remaining scallops which are inevitably generated on the machined surface must be removed by manual surface grinding and polishing. The finish machining and manual polishing stages require as much as 75% of the total machining time. Besides,

S. S. Makhanov (✉)
Sirindhorn International Institute of Technology,
Thammasat University,
Rangsit, Pathum Thani 12121, Thailand
e-mail: makhanov@siit.tu.ac.th

manual polishing is prone to error and undesirable irregularities.

The surfaces are usually characterized by complex geometries and variable curvatures. A single surface is usually composed of patches represented mathematically by parametric forms such as Bezier surfaces, B-splines, and NURBS [2]. Design and manufacture of the sculptured surface parts is an expensive and time-consuming process. First, a design surface is transformed into a computer model (possibly with the help of CAD [5]). The computer model is then imported into CAM software to generate commands to move the cutting tool of the machine. The resulting set of tool positions and orientations constitutes a tool path to machine the desired surface.

Five-axis machining offers an improvement in efficiency of the both the rough and the finish machining stage over the three- and four-axis counterparts. In five-axis machining, the tool orientation relative to the workpiece can be controlled by two additional degrees of freedom so as to achieve higher machining efficiency (see Figs. 1 and 2). With these advantages, a large number of tool path planning methods for five-axis machining have been developed and presented in the literature.

This survey of five-axis machining optimization methods is focused on adaptable geometric patterns, tool path interpolators, and methods for tool posture and gouging avoidance. We believe that the above components are the most important for constructing numerical methods for five-axis tool path optimization. The survey also includes three-axis methods so long as they can be extended to the five-axis case.

Finally, kinematics error is important for constructing efficient tool path generation methods. Therefore, the paper includes an appendix presenting kinematics equations for five-axis milling machines classified by the relative positions of the rotary axes.

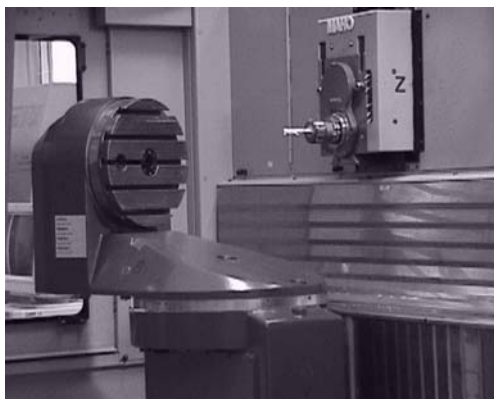


Fig. 1 Five-axis milling machine MAHO600E



Fig. 2 MAHO600E during cutting operations

2 CAD/CAM formats

Every CAD [5] or CAD/CAM software uses an internal format to represent and control the required part. When geometrical data is transferred from a CAD [5] system to a CAD [5] or CAM system, a neutral format for the data transfer is used. One of the most popular is the initial graphics exchange specification (IGES) format (see the history of the IGES format in B). Goldstein et al. [71]. The IGES format supports the use of surfaces defined by NURBS [2], see Farin [54]; Piegl and Tiller [161]; and De Boor [39].

There are also several free libraries designed to control and manipulate NURBS [1], such as NURBS++ (<http://libnurbs.sourceforge.net/index.shtml>) and Nurbs Toolbox (<http://www.aria.uklinux.net/nurbs.php3>). Some free IGES-file processing tools are collected at the NIST/IGES [3] webpage (<http://www.nist.gov/iges/>).

The STL files originally employed by layered manufacturing technologies such as rapid prototyping are now becoming more and more popular due to the simplicity of describing the part surfaces. Rather than a complex description of surfaces, lines and trimming boundaries employed by the IGES format, a collection of triangles each described by the coordinates of its three corners and a normal vector is being transferred. This technology provides an important platform for CAD/CAM applications due to the existence of many robust triangulation algorithms. Besides, the surface models are often composed of many patches. Therefore, by tessellating the patches and creating groups of triangles, one can use many well-established methods for treating intersections, trimming, shading, hidden surface removal, and gouge protection [67]. The machining is usually performed by creating contours obtained by slicing the STL-surface [140, 195]. Of course, slicing of the NURBS [2] surfaces is also possible, but it requires much more sophisticated techniques such as Ma et al. [132]. The contours are then saved using the SLC [4] format (see, for instance, <http://www-rp.me.vt.edu/bohn/rp/SLC.html>).

Other popular CAD/CAM formats include Standard for the Exchange of Product Data (STEP), The Drawing Exchange Format from AutoDesk and many others. As a matter of fact, the difference in data formats has created a large software industry specializing in transferring, adapting, and processing the CAD/CAM files. A table of compatibility of the CAD/CAM formats is at <http://www.cs.cmu.edu/People/unsal/research/rapid/cadcam.html> or <http://www.actify.com/v2/products/Importers/formats.htm>.

3 Optimization of the tool path and adaptable patterns

Optimization of tool paths for five-axis machining includes many features and multiple criteria, such as the accuracy, the length of the tool path, the machining time and the size of the remaining scallops. It may also include gouging avoidance, satisfying the machine axis limits, maximizing the volume of the removed material, and reducing the tool wear. The optimization may also take into account thermal characteristics of the cutting process, the tool bending, vibrations and jacks, workpiece positioning, and many other parameters. The criteria could also include the configuration of the machine or specific parts of the machine as well as the design of the clamping device.

Readers interested in citations before 1997 could use a fairly comprehensive survey Dragomatz and Mann [45]. Their survey presents a classification of research on three- to five-axis machining related to geometry of the tool paths and tool positioning. These categories include (1) systems, (2) iso-parametric paths, (3) non-iso-parametric paths, (4) planar pocketing paths, (5) sculptured surface pocketing paths, (6) roughing paths, (7) tool positioning, (8) offset surface methods, (9) five-axis machining, (10) mesh models, (11) pixel and point models, and (12) simulation and verification.

Of course, the above groups overlap. Techniques involved in one group could be also involved in another group. For example, systems for tool path generation may include all of the above mentioned techniques. Roughing paths may be generated by the iso-parametric or non-iso-parametric schemes and so on.

This survey is primarily focused on five-axis machining. It also includes three-axis methods so long as they can be extended to the five-axis case. We confine ourselves to techniques designed for cutting the part surface by the bottom edge of the tool, e.g., flat-end milling and fillet milling. Many interesting methods designed for five-axis grinding (flank milling) and plunge milling are not included.

The survey is focused on the following three categories: tool path interpolators, methods for tool posture and gouging avoidance, and adaptable geometric patterns.

3.1 Tool path interpolators

A conventional scheme for five-axis tool path planning and control is presented in Fig. 3. It includes generation of the cutter contact (CC) path, offsetting to create a cutter location (CL) path, and interpolation to create inputs for axial controllers of the milling machine.

Conventional servo control systems employ linear or circular interpolation techniques, see Koren [104]. Both methods may result in discontinuities in the velocity at the junctions of the segments. They may also result in high accelerations and subsequent surface inaccuracies as well as long machining time. Furthermore, modern high-speed machining may require feed rates up to 40 m/min with accelerations up to 2g. At such high speeds, small discontinuities in the reference tool path can result in undesirable high-frequency harmonics in the reference trajectory, which may excite natural modes of the mechanical structure and the servo control system.

Although the NC program cannot change the way the controller moves the machine parts, the cutter location points, the rotation angles, and the feed rates applied to cut the part surface can be changed in such a way that the errors are minimized or at least reduced on the tool path generation stage.

Optimization of the CC positions is often combined with feed rate adaptation and reprogramming the machine controller.

The way the controller operates may be changed in two different ways. The first method uses the cutter path curve, which is transformed from the part-based coordinate system to the machine coordinate system, maintaining a special representation of the curve (for instance, a spline). Then the

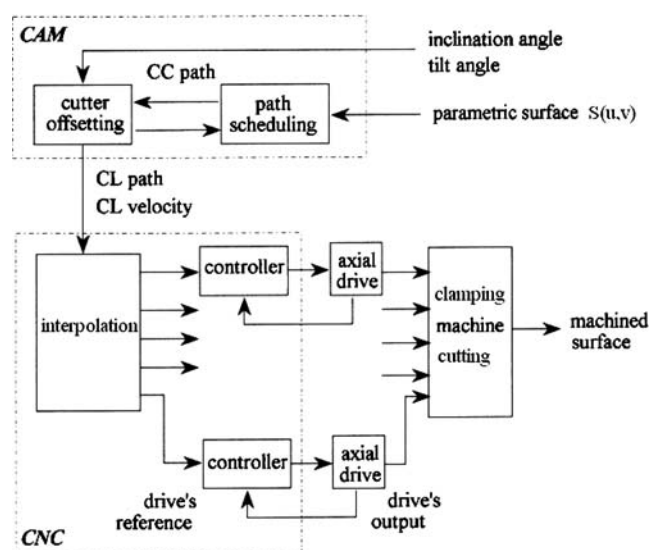


Fig. 3 A conventional scheme for tool path generation and control

part program is composed of axis-positioning commands given in terms of the spline representation. As the part program is run, the controller interprets the spline representation of the path and uses this to find a sequence of discrete time axis-positioning commands for the given feed rate. An example is the Siemens SINUMERIK 840-D controller.

The second approach uses a traditional part program, processed for certain cutter location points, but the controller interpolates between the points using a specified interpolation method to produce smoother sequences at the segment boundaries. Such policies could be implemented with an open CNC controller [see 159].

The task of generating such a sequence of points using a traditional or advanced controller is called interpolation. The method of moving the tool (actually the machine parts) between the prescribed positions or along the entire curve is called the tool path interpolator.

One of the major problems occurring in tool path interpolators design is the need to accommodate smooth accelerations to suppress machining force variations and to subdue feed rate fluctuations that arise from the discrete nature of the traditional piecewise linear or circular interpolation methods.

Conventional G-code part programming invokes a basic conflict between the accuracy with which the curved path is specified and the smoothness with which it is traversed at high feed rates. Indeed, a piecewise linear or circular tool path can be improved to any prescribed accuracy by increasing the number of approximating segments. However, if l is a typical segment length, V is the feed rate, and Δt is the sampling time of the controller, the feed rate accuracy is maintained only if $V\Delta t \ll l$, in other words, if many sampling intervals elapse while traversing a segment. Otherwise, one may expect a variety of deleterious consequences, including jerkiness of the tool motion and execution times longer than those expected from the specified paths and speeds.

Early interpolation schemes tackle this problem by smoothing the tool path at the corners [27] and using low-pass filters [208]. The low-pass filter scheme proposed by Weck and Ye, which they refer to as “the inverse compensation filter”, filters out the high-frequency components of reference trajectories and makes them easier to track. A similar approach was proposed by Tung and Tomizuka in [196].

However, the filter’s parameters depend on the drive dynamics, and for operation at a different feed rate, readjustments to the filter are necessary. Besides, the above filtering techniques ignore the basic difference between the chord and the arc lengths. That is, actually, why generating the tool positions by incrementing the chord length leads to feed rate instabilities.

In the controller, the interpolation points are generated with a constant frequency called the sampling rate of the controller loop. Consequently, the time between two consecutive points is constant during the interpolation. Therefore, the feed rate has to be controlled depending on the arc length distance between two consecutive points. The ideal fit curve must be parameterized with respect to the arc length. In this case, equal increments in the curve parameter yield equal increments in the arc length, and feed rate inaccuracies are eliminated.

Many modern interpolation schemes are focused on finding a suitable interpolation. Based on B-spline curves, a variety of free-form curve interpolators for CNC machining have been proposed [89, 179, 212, 217, 223, 224].

However, the major inconvenience of the B-splines is that the arc length along the curve has to be approximated using numerical integration leading to inevitable numerical errors.

As a matter of fact, arc length parameterizations are not analytically possible for general spline curves. Therefore, a number of approximate solutions have been proposed. Shpitalni et al. [179] present a numerical method based on a truncated Taylor series to estimate in real-time the next point along the spline for a given feed rate. Suppose that the spline functions representing the cutter path are given by $P(t)=(x(t), y(t), z(t))$, $R(t)=(a(t), b(t))$, where $P(t)$ represents the three spatial coordinates of the tool tip (the tool position vector) and $R(t)$ represents the rotation angles. Suppose that t is discretized as $\{t_0, t_1, \dots, t_n\}$. Using a first-order Taylor series yields: $t_{k+1} = t_k + t' \Delta t + o(\Delta t^2)$, where prime denotes the derivative and Δt the increment. Given a desired velocity V (the feed rate) and a sampling (servo update) period T_s our basic interpolation algorithm assigns the discrete values $t=t_k$ as follows:

$$t_{k+1} = t_k + \frac{V T_s}{\sqrt{x'^2 + y'^2 + z'^2}}, \quad (3.1.1)$$

This generates approximately equal arc-length steps for each round of interpolation.

A compensatory parameter for the first-order approximation Eq. 3.1.1 was proposed by Yeh and Hsu [216].

Of course, the first-order approximation is not always appropriate [101]. Therefore, the accuracy is enhanced by using a second-order Taylor series. In this case the curve parameter is incremented as follows:

$$t_{k+1} = t_k + \frac{V T_s}{\sqrt{x'^2 + y'^2 + z'^2}} - \frac{V^2 T_s^2 (x'x'' + y'y'' + z'z'')}{2(x'^2 + y'^2 + z'^2)^{3/2}}. \quad (3.1.2)$$

Wang and Yang [203] generate the trajectory by means of cubic and quintic splines using the chord length and a nearly arc length parameterization, implemented with the quintic spline interpolation. Assuming small curvatures, the points on the original curve can be (approximately) evenly distributed so that the resulting composite spline is closer to being arc-length parameterized with the reference to the cubic splines. The quintic splines are likely to be shaped close the original curves without unwanted high-order oscillations. With this property the next point along the spline is simply obtained by

$$t_{k+1} = t_k + V T_s.$$

The first stage of the algorithm is calculation of the near arc-length quintic position spline using a set of position vectors. The k -th segment of the position spline on $[t_k, t_{k+1}]$ is represented by $P_k = c_{0,k} + c_{1,k}t + c_{2,k}t^2 + c_{3,k}t^3 + c_{4,k}t^4 + c_{5,k}t^5$, where $t \in [0, l_k]$ and l_k is the range of t . Coefficients $c_{i,k}$ are found using the following procedure. First, a cubic spline $p(t)$ is constructed through the given set of points. Next, the first and second derivatives of the cubic spline at the data points are extracted and used to construct the quintic near arc-length parameterized spline. The first derivative is normalized so that the quintic spline is arc-length parameterized at the data points. Coefficients $c_{i,k}$ are then found by

$$\begin{aligned} c_{0,k} &= p_k, \quad c_{1,k} = p'_k, \quad c_{2,k} = \frac{p''_k}{2}, \\ c_{3,k} &= \frac{10(p_{k+1} - p_k)}{l_k^3} - \frac{2(p'_{k+1} + 3p'_k)}{l_k^2} + \frac{p''_{k+1} - 3p''_k}{2l_k}, \\ c_{4,k} &= \frac{15(p_{k+1} - p_k)}{l_k^4} + \frac{7p'_{k+1} + 8p'_k}{l_k^3} - \frac{2p''_{k+1} - 3p''_k}{2l_k^2}, \\ c_{5,k} &= \frac{6(p_{k+1} - p_k)}{l_k^5} - \frac{3(p'_{k+1} + p'_k)}{l_k^4} - \frac{p''_{k+1} - p''_k}{2l_k^3}, \end{aligned} \quad (3.1.3)$$

where $p_k = p(l_k)$. Finally, l_k are re-evaluated using the condition: $\frac{dp_k}{dt} \left(\frac{l_k}{2} \right) = 1$.

This model was followed by [202], who included an extra jerk (the rate of change of acceleration) continuity into the quintic spline interpolation. Although this improved the correctness of the resultant feed rate profile, significant fluctuations may still occur at high curvatures. Therefore, Wang and Wright recommended increasing the number of points at these segments to reduce the fluctuations due to the parameterization errors.

Fleisig and Spence [66] improved the quintic spline routine by finding the curvature of the cubic spline at the data points and applying these values to construct the quintic spline. This yields a spline that is significantly closer to arc-length parameterization. They also introduced

the so-called orientation spline which interpolates the orientation of the tool vectors. Since the orientation of the tool is defined by a unit vector, the orientation spline must lie on the surface of the unit sphere. The orientation spline must have both the near arc-length parameterization property and C^2 continuity. Therefore, it requires some adjustments. The first adjustment is made in the selection of the tangents and curvatures of the quintic spline by modifying the first and the second derivatives extracted from the cubic spline. The second improvement is made once the quintic spline control points are known, by forcing the unit tangency property at the middle of each segment.

Weck et al. [207] presented cubic spline interpolation, where feed rates at the spline segments are corrected based on the physical limitations of the drives using fourth-order acceleration profiles. This idea was then further developed by Erkorkmaz and Altintas [52] in the quintic spline trajectory generation algorithm to produce continuous position, velocity, and acceleration profiles. The reference trajectory generated with varying interpolation period is re-sampled at the servo loop closure period using fifth-order polynomials, which enables the desired kinematics profiles to be preserved.

The quintic spline is parameterized with respect to the chord length between two consecutive positions and the chord length is retained as the parameter. However, the increment is adjusted recursively to obtain a constant displacement magnitude at each step of interpolation, hence avoiding feed rate fluctuations. The step is selected so that the total length L is traveled at the highest feed rate $h_{\max}$. Therefore, the number of interpolation steps $N_i = \left\lceil \frac{L}{T_s V_{\max}} \right\rceil$ and $\Delta t = \frac{L}{N_i}$.

For simplicity, consider two axes. Suppose that the axes increments are given by Δx and Δy . The path increment is then given by

$$\Delta t = \frac{L}{N_i} = \sqrt{\Delta x^2 + \Delta y^2}.$$

Invoking the quintic representations yields

$$\Delta x = x_{i+1} - x_i = c_0^x + c_1^x t + c_2^x t^2 + c_3^x t^3 + c_4^x t^4 + c_5^x t^5 - x_i,$$

$$\Delta y = y_{i+1} - y_i = c_0^y + c_1^y t + c_2^y t^2 + c_3^y t^3 + c_4^y t^4 + c_5^y t^5 - y_i.$$

Substituting Δx , Δy into the equation for Δt produces a 10th order polynomial equation solved numerically by Newton–Raphson's iterations. An additional modification providing a smooth feed motion along the quintic spline is performed by modulating the interpolation period between evenly spaced reference points. The approach allows the feed rate to be easily modified on the fly, by rescaling the interpolation period with a desired override factor. As

the result the requested value of jerk is not larger than an achievable value for given acceleration, deceleration, and sampling period values. The kinematics profiles used in feed rate generation are characterized by trapezoidal acceleration profiles with pre-specified slopes (jerk values). The acceleration, feed rate, and displacement profiles are obtained by integrating the jerk profile.

Substituting specified jerk profiles, integrating and setting a desired jerk magnitude as well as desired acceleration and deceleration magnitudes makes it possible to evaluate feed rates at the end of each stage as well as the distances to be traveled at each stage. Finally, reference trajectories generated with varying interpolation period are reconstructed at the control loop closure period so that smooth feed rate and acceleration profiles are received by each digital axis controller.

Of course, the jerks are only limited rather than entirely eliminated. The approach requires a considerable offline computing effort. In particular, in order to generate the boundary conditions for the quintic spline, the method uses a preprocessing cubic spline interpolation. Another disadvantage is that since the chord length is used to parameterize the curve the method leads to good results only for small curvatures. Yet, this technique is an efficient combination of the kinematics profile analysis and quintic spline interpolation method. However, as far as this group of methods is considered, it is not clear how to implement them for a five-axis case and whether such technique is applicable to the jerks appearing in the rotation axis.

A similar approach was proposed by Nam and Yan [152]. The jerk-limited path is obtained by estimating an admissible range of accelerations at the consecutive points along the tool path and deciding the maximum possible value for achieving the final deceleration stage at every sampling period. Interestingly, four years earlier Red [171] took it one step further allowing for non-zero starting accelerations.

Some research has been focused on generating optimal or nearly optimal speed profiles for single moves to execute the process plan as quickly as possible, subject to process constraints, and dynamic limitations of a given machine.

Renton and Elbestawi [172] developed a two-pass algorithm to determine a minimum time speed profile subject to speed and acceleration constraints. The method is computationally expensive and evaluates the speed profile in the parametric domain.

Dong and Stori [40] extended Renton and Elbestawi's method to account for and limit the effects of actuator limitations on the contour error. Timar et al. [44] developed a method for time-optimal speed profiles for a single move subjected to speed and acceleration constraints.

As far as more general control system approaches are concerned Shin and McKay [181] proposed a dynamic programming approach to solving the minimum time control

problem with jerk constraints. In the general framework, there are no limitations on the form of the constraints, cost function, or system model. However, the approach is computationally expensive since discretization of the phase plane requires high numerical accuracy and, therefore, a large number of points to obtain an acceptable feasible solution.

Dong et al. [43] suggested a general approach in the framework of the dynamic systems control theory. Given an arbitrary parameterized curve $S(t)$, the feed rate optimization problem is formulated as identification of a parametric velocity function $\dot{s}(t)$ such that $S(s(t))$ is time-optimal subject to a particular set of limitations and dynamics state constraints. The machine tool constraints are the velocity, acceleration/torque, and jerk limits of all active linear and rotary drives (Fig. 4). Since the feed must be assigned along the tool path in such a way that the constraints are not violated, the velocity, acceleration, and jerk become path-dependent as follows:

$$\begin{aligned} v_S(t) &= \frac{dS(s)}{ds} \dot{s} \\ a_S(t) &= \frac{d^2S(s)}{ds^2} \dot{s}(t)^2 + \frac{dS(s)}{ds} \ddot{s}(t) \\ j_S(t) &= \frac{d^3S(s)}{ds^3} \dot{s}(t)^3 + 3 \frac{d^2S(s)}{ds^2} \dot{s}(t) \ddot{s}(t) + \frac{dS(s)}{ds} \dddot{s}(t). \end{aligned} \quad (3.1.4)$$

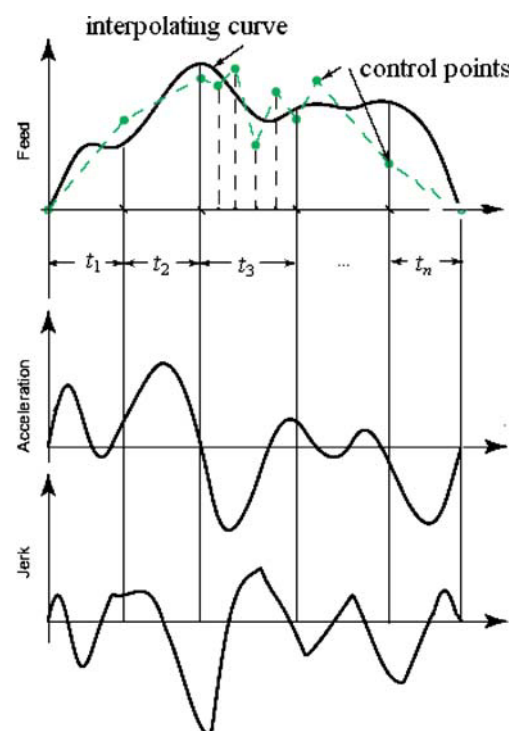


Fig. 4 The feed, acceleration, and jerk profiles along the tool path

With these relationships the system's states can be represented explicitly in terms of the parametric velocity, acceleration, and jerk. The problem is stated as

$$\underset{\dot{s}(t)}{\text{minimize}} \int_0^1 \frac{ds}{\dot{s}(t)} \quad (3.1.5)$$

subject to constraints representing limitations on the jerk, acceleration, and velocity given by

$$|\dot{s}(t)| \leq J_{\max},$$

$$|k_a a_S(t) + k_v v_S(t)| \leq F_{\max}(v_S(t)),$$

where $k_a, k_v, F_{\max}(v)$ represent the system capability constraints of each axis derived from the system's dynamics characteristic (see further details in [41, 61]). However, the methods are computationally expensive. Besides, they do not consider the actual kinematics of the machine, assuming that the tool follows a certain parameterized curve. Furthermore, [42] included into the formulation some tool capability constraints, and presented a two-pass algorithm to solve the corresponding optimization problem. Unfortunately, the resulting trajectories may still possess discontinuous acceleration and torque profiles, defeating the purpose of using a smooth tool path and leading to inaccurate cuts for high-speed machining.

Smoothing the trajectory and enhancing the tracking performance taking into account the jerk limits of the drives has been considered in Tarkiainen and Shiller [198] and Piazzzi and Visioli [163].

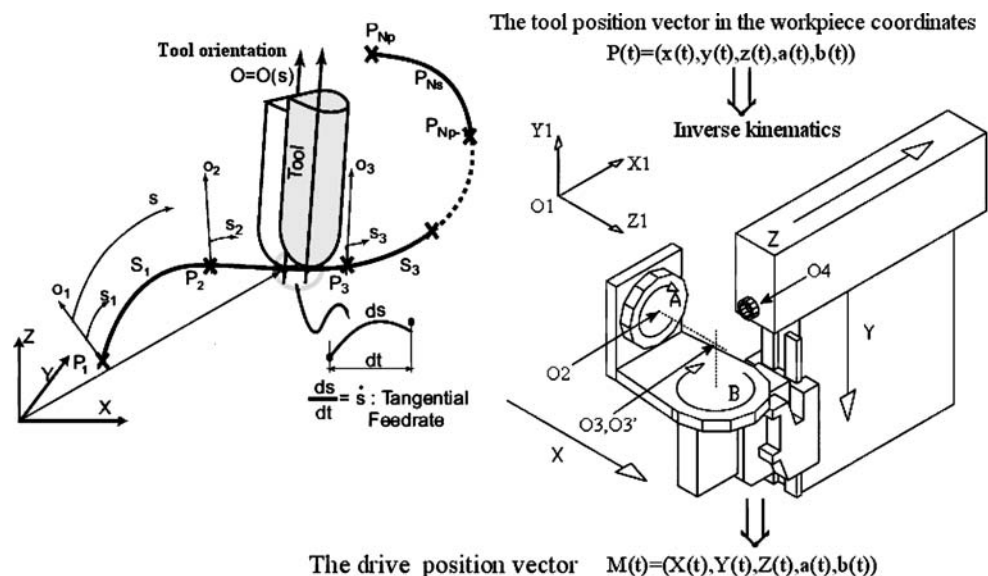
Constantinescu and Croft [37] used torque rate constraints, and iteratively optimized the spline trajectories for finding minimum time trajectories of robotic manipulators. In all those cases the constraints considerably increase the computational complexity, requiring iterative numerical methods.

Sencer et al. [185] minimize five-axis machining time subject to process and machine tool constraints. The first constraint is the vector feed along the tool path defined by the NC program and the machine tool constraints are the velocity, acceleration/torque, and the jerk limits of all active linear and rotary drives (see Eq. 3.1.5). In addition to that, the tangential feed, acceleration and jerk are fitted as to cubic or quintic splines to avoid discontinuities in the multi-axis trajectory. Furthermore, the drive constraints are adapted to the five-axis mode as follows: the tool tip position $P(t)=(x(t), y(t), z(t))$ and the orientations are transferred into the drive position vector $M(t)=(X(t), Y(t), Z(t), a(t), b(t))$ for which the profiles of velocity, acceleration and jerk can be derived similarly to (Eq. 3.1.4; see Fig. 5) but with regard to the drive trajectories produced by the five-axis kinematics. The velocity constraints are given by

$$-1 \leq \begin{pmatrix} \frac{X(t)}{v_{x,\max}} \\ \frac{Y(t)}{v_{y,\max}} \\ \frac{Z(t)}{v_{z,\max}} \\ \frac{a(t)}{v_{a,\max}} \\ \frac{b(t)}{v_{b,\max}} \end{pmatrix} \dot{s} \leq 1.$$

It is clear that the slowest drive, which has the maximum derivative of its displacement normalized with its velocity limit, defines the allowable feed. The acceleration and the jerk limits are determined in a fashion similar to Eqs. 3.1.4 and 3.1.5 for all five drives.

Fig. 5 Generation of the motions for five-axis drives



Note that in Eq. 3.1.5 Sencer et al. [185] replace $\dot{s}(t)$ by $\dot{S}(t)$, where S is an arbitrary splined tool path, as follows:

$$\underset{\dot{S}(t)}{\text{minimize}} \int_0^1 \frac{ds}{\dot{s}(t)}. \quad (3.1.6)$$

Furthermore, the feed profile is represented in a B-spline form as a function of the path length with modulated control points defined at some fixed path positions.

The optimization is performed with regard to these positions to minimize the machining time while respecting the axis constraints. The feed control points move up and down, altering the feed profile, during an iterative optimization process until optimal feed values, which do not violate the machine drive constraints, are obtained. Note that the optimization process can be computationally inefficient if all control points are considered simultaneously along a long tool path. Therefore, the optimization algorithm is applied in a suboptimal fashion, employing moving windows with a smaller number of control points.

Finally, in spite of a variety of research and experimental papers there is still no agreement on the best interpolation $S(t)$ in Eq. 3.1.6. Although NURBS [2] seems to present one of the best options, arc length parameterizations are not analytically possible for general spline curves.

In [55, 56], Farouki introduced Pythagorean–hodograph (PH) curves to solve the problem of feed rate control for three-axis machines. These curves provide a mathematically elegant solution to interpolation problems occurring in NC machining.

For simplicity consider a 2D case. The hodograph of a polynomial curve $S(t)=(x(t), y(t))$ is called a curve given by $h(t)=(x'(t), y'(t))$. A Bezier curve is called Pythagorean–hodograph (PH) if the length of its tangent vector depends in a (piecewise) polynomial way on the parameter t . In particular $S(t)=(x(t), y(t))$ is called the planar PH curve if there exists a polynomial $\sigma(t)$ such that

$$\sigma^2(t) = x'^2(t) + y'^2(t).$$

Furthermore, introducing a complex valued $\tilde{S}(t) = x(t) + iy(t)$ and $\tilde{h}(t) = x'(t) + iy'(t)$, entails that $S(t)$ is a PH if and only if there exists a complex polynomial $w(t)$ such that $\tilde{h}(t) = w(t)^2$.

In terms of real analysis it means that there exist three polynomials $\tilde{\alpha}(t), \tilde{\beta}(t), \tilde{\omega}(t)$ such that

$$\begin{aligned} x'(t) &= \tilde{\omega}(\tilde{\alpha}^2 - \tilde{\beta}^2), \quad y'(t) = 2\tilde{\omega}\tilde{\alpha}\tilde{\beta}, \quad \sigma(t) \\ &= \tilde{\omega}(\tilde{\alpha}^2 + \tilde{\beta}^2). \end{aligned} \quad (3.1.7)$$

A major advantage of the PH curves, compared to the ordinary polynomial curves, is that their arc length is a polynomial given by

$$s(u) = \int_{u_0}^u |\sigma(l)| dl.$$

Besides their offset curves have a closed form given by $o(d, u) = \frac{d}{\sigma(u)} (y'(u) - x'(u))$, where d is the oriented offset distance.

Suppose that $\tilde{\omega}(t) \equiv 1$. In this case the PH curve is constructed as follows. First, two polynomials $\tilde{\alpha}(t), \tilde{\beta}(t)$ are selected to define the so-called pre-image curve. Next, we integrate Eq. 3.1.7 to obtain a parametric representation of the PH curve. Since curves having the same hodograph differ only by translation, a regular planar PH curve is fully determined by the pre-image and by the location of its starting point.

In Farouki and Neff [57], a 2D Hermite interpolation combined with PH curves was proposed and analyzed. The ideas were further developed in Farouki et al. [60, 62] and Farouki et al. [61]. It has been demonstrated that since the arc length of the PH curves can be represented by a polynomial function of the curve parameter, they can be successfully used for NC interpolation. Consequently, a variety of planar PH curves given Hermite-type boundary data were developed and implemented [e.g. 97, 145, 184].

Finally, Aigner et al. [8] demonstrate the use of the PH curves for approximation of data, in the framework of an evolution-based least square approach. Some available PH solutions for the Hermite-type interpolation are given in Table 1 [184].

We outline the following advantages of the PH curves:

- the ability to specify smooth accelerations and decelerations along curved tool paths through the use of feed rate functions with linear or quadratic dependence on the arc length,
- the use of curvature dependent feed rates to reduce machining force variations due to varying material removal rates at fixed depth of cut along curved tool paths
- the ability to directly interpolate the offset at a given fixed distance from a curved path for tool radius compensation,
- suppression of feed rate fluctuations incurred by the incompatibility of discrete i. e. linear/circular tool path descriptions with smooth realization of high speeds and hence improved surface finish.

Figure 6 shows PH quintics (bold) compared with conventional cubic interpolants to a first-order Hermite

Table 1 Hermite-type interpolation of planar data by the PH curves

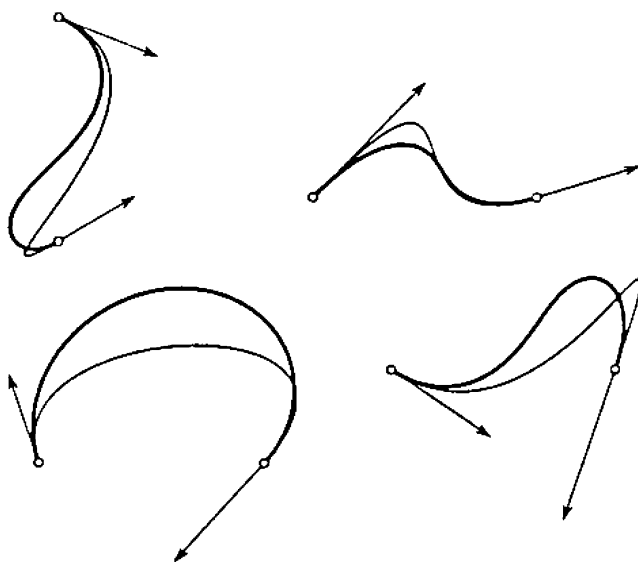
Continuity	Degree	Number of solutions	Remarks
G^1	3	2 solutions, a quadratic equation [142]	One of the solutions has the 4-th order approximation at generic points
C^1	5	4 solutions, quadratic equations [57]	The best solution can be identified via its rotation index [145]. One of the solutions has approximation order 4 [184]
G^2	5	System of 6 equations, the 2-d degree [58, 59]	
$G^2(C^1)$	7	8 solutions, quartic equations [97]	One of the solutions has the 6-th order of approximation at generic points. Inflections reduce the approximation order [97]
G^2	9	4 solutions, quadratic equations [58, 59]	The best solution can be found by visual inspection of the curves and their control points [58, 59]. One of the solutions has the 6-th order of approximation [184]

data. The PH curves have fairer shapes especially in cases with inflections [60, 62].

Among the disadvantages of PH curves are computational difficulties and the fact that they have less degrees of freedom than other polynomial curves of the same order. For example, a cubic PH curve has only one degree of freedom. At least five degrees of freedom are required for a PH curve to have an inflection point.

Besides an important step in the application of PH curves is their construction from the input data. Due to the special algebraic properties of PH curves, all constructions which are linear in the case of standard curves [such as the Bezier curves or 2] become nonlinear in the PH case. In particular, the applicability of global constructions, such as global spline interpolation and least squares fit is limited, since they lead to large systems of nonlinear equations.

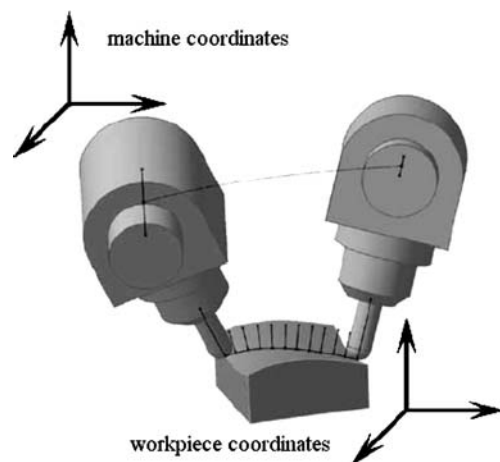
Furthermore, a common drawback of all the above interpolators is that they neglect the actual machine kinematics. Consequently, there is a need to merge the concept of tool path interpolation with kinematics of a particular milling machine (see the Appendix).

**Fig. 6** PH curves vs. cubics

The fact that, a linear trajectory in the workpiece coordinate system becomes nonlinear in the machine coordinates (if the rotary axes are involved) is not always well understood in the engineering practice. However, the errors induced by the kinematics of the machine in many cases become the major source of the inaccuracies. Moreover, the kinematics error may exceed the above mentioned errors related to the discontinuities of the velocities and accelerations.

The nonlinear effects can be observed on the real speed of the tool relatively to the part [111]. As opposed to considering the tool motions solely in the workpiece coordinates, the corresponding errors include kinematics of the particular machine making the optimization problem machine-dependent (Fig. 7).

Some classification of machine errors and demonstrations of the nonlinear effects are given in Bohez [25]. Coordinate transformations between the workpiece and machine coordinate systems for five-axis milling are incorporated in the interpolator by Lo [126, 128]. The task of the proposed interpolation algorithm is to assign a distribution of the sampled tool locations following a specified parametric curve with a desired velocity. The

**Fig. 7** A linear trajectory in the machine coordinates becomes nonlinear in the workpiece coordinates

algorithm utilizes a recursive approach based on Taylor series Eqs. 3.1.1 and 3.1.2 combined with inverse the kinematics transformations.

In [128] a problem of maintaining a desired cutter contact point velocity, rather than a cutter location point velocity, is considered. This task is combined with decreasing of the deviation from the desired curve along the cutter contact path. The proposed method includes spline functions for the CC location, the surface principal normal, the CC velocity, the tool inclination, and the rotation angles.

The CNC interpolator calculates the CL path in real-time so as to meet the desired CC velocity along the path. A global feedback loop is closed by the interpolator, which monitors the actual CC location in real-time and compensates for the deviations from the desired surface. In addition, an online adaptation of the feed rate is introduced.

Muller et al. [148] presented an algorithm for simultaneous five-axis spline interpolation which merges PH interpolation and an analytic solution of the inverse kinematics equations using a template equation method. The result is a time-dependent spline which represents a given tool path with a high accuracy. However, the technique is complicated and cumbersome. Some technological factors such as maximum allowed velocities and accelerations along each axis or altering material removal rates are not considered. The authors write that “the algorithm turned out to be very robust [only] if the following assumptions are true: (1) the given surface is smooth (no geometric discontinuities); (2) the position and tangent values can be calculated on the given surface and (3) a reasonable discretisation is given. Points (1) and (2) seems to be trivial but it turned out in practical applications that the surfaces obtained from commercial CAD [5] or CAM systems do not always correspond to what was expected from visualization. Besides, the problem of finding an optimal discretisation is excluded from the scope of this article although the issue is important for the robustness of the algorithm in practical applications.”

The machines with rotation axes on the table often have to turn around heavy workpieces. Therefore, they must support significant mechanical efforts during machining. As a result, these machines may have low capacities for acceleration. When such a machine has to slow down or stop, the speed reduction requires a considerable time for deceleration and re-acceleration. This effect significantly increases machining time, and is amplified in HSM when the rotation axes reach greater speeds. Therefore, it can even be economically justified to use three-axis at high speeds to reduce the number of decelerating axis.

As far as the five-axis machines are concerned, two basic problems can be considered. Problem A: evaluation of the tool path using a polynomial curve in the workpiece coordinates

and problem B: calculation of the tool path in the machine coordinates using inverse kinematics transformations.

Langeron et al. [111] note: “The essential difficulty concerns the relative position of these two problems within the general process. In other words, does the calculation of the tool path have to be carried out in the part coordinate system (problem A before problem B) or in the machine coordinate system (problem B before problem A)?”

In case of “problem A before problem B”, the CAM system fits a curve to a set or subset of prescribed tool path positions. Furthermore, the spatial coordinates as well as the rotation angles can be represented in the workpiece coordinates using a certain family of curves such as the B-splines PH curves, etc. At this stage, tool path discontinuities can be detected. The curve is then transmitted to the NC unit, which interprets it before performing the inverse kinematics transformation. Therefore, this case seems to be well adapted to five-axis polynomial curve machining.

Langeron et al. [111] suggest a polynomial B-spline interpolation method which takes into account the kinematics of the five-axis machine as follows: $P(t)=(x_1(t), y(t), z_2(t))$ defines the locus of the tool tip locations, whereas $Q(t)=(x_2(t), y_2(t), z_2(t))$ defines the locus of a second point belonging to the tool axis. This new five-axis format is generic and allows for representing the tool path for all types of machining such as three, four, or five-axis machining, including point, flank, or even plunge milling (Fig. 8).

Clearly, $\exists H : \forall t \quad \|P(t) - Q(t)\| = H$.

Moreover, the tool orientation is conveniently defined by $O(t) = \frac{P(t)-Q(t)}{\|P(t)-Q(t)\|}$.

It has been shown that the new format makes it possible to design efficient algorithms for elimination of gouging. As a matter of fact, with this format, one can easily quantify the maximum distance between the side of the tool and the surface or the distance between the back of the tool and the part. Furthermore, NURBS-based [2] $P(t)$ and $Q(t)$ are used to represent the tool movement in the part coordinate system. The C^2 continuity introduced by the interpolation format ensures continuous dynamics of the rotation axes, which often have lower capacities of acceleration and thus

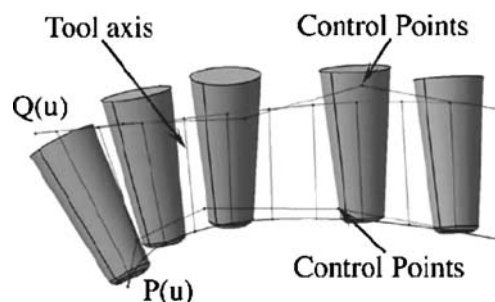


Fig. 8 Tool path definition using the two-curve format

are the main source of the kinematics errors. Although the errors due to the difference between the chord and the arc lengths are not analyzed with the reference to the machine kinematics, the proposed format appears to be suitable to be combined with the PH curves to represent $\{P(t), Q(t)\}$. However, such analysis remains an open problem.

Lo [129] introduced spline interpolators for iso-parametric, iso-scallop, and iso-planar machining methods (see the forthcoming sections). Besides that, the paper merges online interpolation with the benefits of offline tool path generation. The offline part of the interpolation approach is based on the CC path, followed by real-time tool offsetting that calculates CL points. Note that the older methods use cutter offsets computed in the CAM system before interpolation. As opposed to that, Lo's proposed procedure is executed at the sampling rate of the actual CNC unit. Therefore, at any sampling instant, the CC point is always located on the surface, whereas the older methods lead to CC position errors, except at the end points of the segments.

Sir et al. [183] presented a modification of the bi-arc interpolation technique based on spline curves composed of circular arcs and compared them with the PH curves. The paper concludes that PH curves are still one of the best options for maintaining a constant feed rate.

Furthermore, NURBS-based [2] interpolators have received considerable attention. By manipulating the weights, knot vectors, and control points, a variety of curves suitable for interpolation can be designed and implemented [161]. NURBS [2] have also been employed by many CAD/CAM systems as a basic geometry representation. Therefore, many NURBS-based [2] command generators were proposed [see, for instance, 32, 107, 224].

NURBS [2] command generators eliminate linear interpolation errors as well as errors due to converting from a CAD [5] representation to the CNC interpolating mode. Many NURBS [2] interpolators similar to those employed for cubic or quintic splines [16] are capable of achieving approximately constant velocity along the required curve. In fact, NURBS [2] have become the standard for many CAD/CAM and computer graphics systems. Their basics are considered in many text books such as, Piegl and Tiller [161]; Farin [54], etc.

Some CNCs have already incorporated the option to internally spline a point data in the conventional part program. However, better solution is to interpolate directly from the actual curves generated in CAM. For instance, GE Fanuc [6, 7] [GE FANUC, NURBS 2 Interpolation] control reads a G-code which includes the NURBS [2] data: the control points, weights, and knot vectors required to define the curve. The control builder asserts that this method of representing curved cutter paths "results in a reduction of program size of 1/10th to 1/100th of a comparable linear

interpolation part program and significantly improves the fundamental accuracy issues." [8] (Akino. Interpolating Curves).

Still there have been many improvements and modifications of the NURBS [2] based on interpolation technologies. Liu et al. [124] develops an interpolation scheme which, when applied to NURBS [2], not only meets the requirements of approximately constant feed rate and chord accuracy but also integrates dynamic factors, namely, sharp corners, feed rate-sensitive corners, components with high frequencies or frequencies matching the machine natural frequencies, and high jerks. A look-ahead module uses the Fast Fourier transform to detect unwanted frequencies. A notch-filtering or a time-spacing method is used to eliminate those components.

Interesting NURBS [2] interpolations, taking into account the volume of the removed material (which could be an important factor as far as the load on the machine tool is concerned), is considered by Tikhon et al. [143] and Ko et al. [102]. The research papers propose a feed rate evaluation algorithm that keeps an approximately constant material removal rate along the desired curve.

Many papers claim an improvement of the NURBS [2] based interpolation; however, it is often a modification of constraint minimization Eq. 3.1.6. For instance, Altintas and Erkorkmaz [12] perform a gradient-based search that leads to shorter time cycles. Erkorkmaz and Heng [53] present yet another heuristic approach to produce a shorter cycle time compared to the worst-case approach. The numerical algorithm converges to a feasible solution faster than a gradient-based method. Lei et al. [118] take an advantage of the numerical adaptive quadrature to evaluate the curve length dividing the parameter interval into subintervals with a fine or coarse grid spacing.

M. Korosec [105] integrates feed rate evaluation based on the constant instantaneous volumetric removal [204] with an adaptive least square NURBS [2] approximation to produce a minimal or close to the minimal number of control points. The method can be interpreted in the framework of grid generation methods comprehensively discussed in this paper Section (3.3).

A novel approach to NURBS [2] based interpolation is proposed by Tsai et al. [193], where a NURBS [2] interpolator has been integrated with a model of servo dynamics of the machine. A look-ahead code includes a corner detection module, a jerk-limited module, and a dynamics module to take into account chord errors, feed rate fluctuations, jerks, and servo-errors simultaneously. The experiments show that in many cases, the techniques achieves better accuracy while requiring less machining time with the reference to an adaptive-feed rate method of Tikhon [143] and curvature-feed rate interpolation proposed by Yeh and Hsu [216].

Finally, a number of papers introduce interpolators designed for high-speed milling (HSM). In HSM operations, the tool path frequently changes direction producing high accelerations and decelerations. Therefore, commercial CAM systems often cannot accurately predict the cycle time by assuming a constant feed rate. During high-speed machining, the actual average feed rate could be significantly lower than the programmed feed rate due to the physical restrictions of the machine tool and the block processing time of the CNC controller. In many cases, the machine tool hardly reaches the maximum feed rates offered by the manufacturer. This happens when the block processing time is longer than the block execution time and the machine reaches the end point of the segment before information required for the next movement is available. In this case some machines stop, producing a significant jerk. Others automatically reduce the programmed feed rate, which results in a lower real feed rate and, consequently, a longer machining time [130]. This relatively new HSM issue has been further discussed in Monreal and Rodriguez [144]. This paper demonstrates the phenomenon experimentally and presents some heuristics to evaluate the actual feed rate. However, a theoretical framework to evaluate the feed rate for HSM has not been proposed.

Siller et al. [182] suggest a mechanistic model based on the frequency distribution of linear interpolation path lengths in the CNC program and characterizations of the machine tool for brisk and smooth movements. The model is characterized by the standard advantages and disadvantages of statistical models. In particular, the model does not take into account the actual kinematics of the five-axis HSM machine. In addition, statistical data could be very different for different parts in which case, the proposed feed rate prediction techniques could be inaccurate.

Hu et al. [87] suggest a look-ahead algorithm to find an approximate “optimal” feed rate for high-speed machining. This paper successfully builds up a mathematical model for feed rates by considering only the key and representative factors, then proposes a feasible way to find the approximate feed rate by reading a preset number of line blocks ahead. Simulations show that this algorithm is able “to anticipate sudden direction changes and react accordingly. If the look-ahead check reveals that there may be a problem, feed rate for intermediate points along the tool path can slow down, otherwise it will maintain a high speed.” The algorithm considers constraints on accelerations assuming linear acceleration/deceleration profiles and constraints on the angular acceleration. However, the angle constraints are somewhat artificial since the tool is being moved by rotating the machine parts. Therefore, the angular accelerations should be represented in terms of the accelerations of the rotating axes rather than the tool. The authors do not differentiate between CL and CC points and

do not take into account the machine kinematics. Finally, the method has not been analyzed from the viewpoint of the optimal number of look-ahead blocks. The authors admit that “Generally the more look-ahead blocks there are, the greater productivity is. However, when the number of look-ahead blocks is set too large, the calculation time and memory cost is also increased. So more blocks do not necessarily mean better performance.”

Finally, surface quality and machining time are recognized as the most important machining criteria in the modern tool-shop industry. However, there are other important measures such as the tool wear, the machining costs and so forth, which can be significant as well. Therefore, the intelligent controller of the future should be capable of adaptation according to a variety of selected technological parameters and criteria.

3.2 Tool posture and gouging avoidance

This section deals with techniques to optimize the position and orientation of the tool in a particular neighborhood of the machined surface. The most important application of these techniques is cutting a part surface by a flat-end or a fillet mill.

In [141], Marciniak proved that in five-axis machining, the width of the machined strip on the surface can be maximized when the tool is moved on the surface approximately along the minimum curvature line or curve. The maximum strip width depends on the difference between the surface main curvatures at the contact point.

Some of the early research papers exploiting this idea are Kruth and Klewais, [108], Li and Jerard [120], and Gani et al. [68].

Furthermore, Kruth and Klewais [108] introduced an optimal milling direction parallel to the principal direction of the surface with the minimum curvature.

Elber [49] introduced a tool path optimization method which splits the surface into convex, concave, and saddle-like regions. Applying a flat-end cutting tool to convex regions and a ball-end tool to other regions leads to better material removal rates and smaller scallops. Of course, having irregular-shaped regions cut by different tools is not always a good idea due to difficulties in finding the boundaries of those regions and possible inaccuracies at the boundaries. On the other hand, the ball-end (ball-nose) tool is capable of changing the orientation without additional path correction which makes it useful for regions that are hard to reach (see Fig. 9)

Another important drawback of the ball-end cutters is that the cutting speed varies along the tool radius. Maximal cutting speed is reached along the tool circumference, and at the tool tip (the center) it is zero. This may lead to cutting edge chipping as well as poor surface roughness (see Fig. 10).

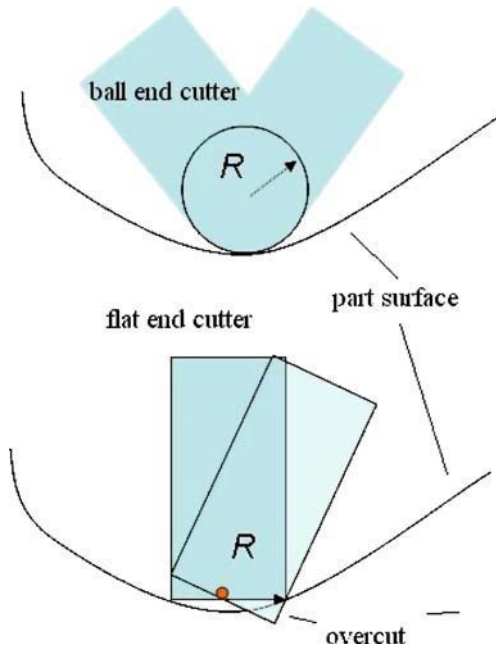


Fig. 9 Ball-end vs. flat-end cutter

Gani et al. [68] noticed that “One of the critical problems in five-axis milling is the positioning of the cutter in relation to the surfaces in order to machine without having overcut or undercut (gouging). Because of this problem, ball-end cutters are preferred because the tool path for the ball-nose cutters is mainly a problem of surface offset.”

Recall that in five-axis machining, the tool has five degrees of freedom relative to the surface. The three spatial degrees are used to locate the tool at the cutter location. The additional two rotational degrees establish the orientation of the tool represented by the inclination angle and the tilt angle [see, for instance, 178] or the tilt angle and the yaw angle [96]. The angles are evaluated in a local coordinate system usually defined by the feed direction, the surface normal and the corresponding cross-product vector.

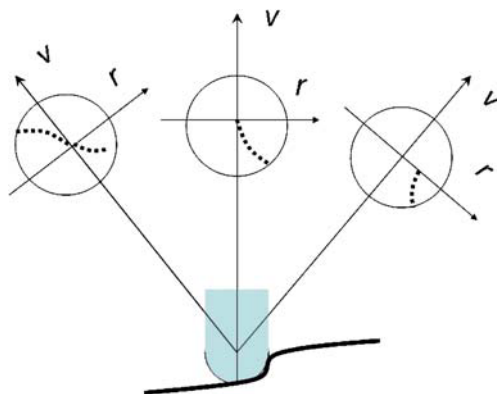


Fig. 10 Cutting speed profiles for the ball-nose cutter

Consider the flat-end mill. The boundary of the base of the tool, which is the part of the tool cylinder, is called the cutting circle of the tool.

The effective cutting shape, also referred to as the tool swept section, is defined as the projection of the base of the tool onto the plane normal to the feed direction. The projected edge of the flat-end mill becomes an ellipse called the effective cutting ellipse.

Let us introduce a local coordinate system (O_1, x_1, y_1, z_1) at CC point O_1 shown in Fig. 11, where x_1 denotes the normalized projection of the tool cutting direction onto the tangent plane, z_1 denotes the surface normal vector, and $y_1 = z_1 \times x_1$. The tool is rotated by an inclination angle λ about the y_1 axis and by a tilt angle ω about the z_1 axis. The projected bottom edge of a flat-end cutter with radius r onto the (y_1, z_1) -plane is the effective cutting ellipse.

In order to evaluate the machining strip, the surface cross-section perpendicular to the tool cutting direction x_1 is approximated by a circular arc, for which radius R_y is equal to the radius of the normal curvature of the surface in the y_1 direction as shown in Fig. 12. Suppose that h is the maximum allowed scallop height. The maximum machined surface error can be represented by a virtual circular arc with radius $R_y - h$ shown in Fig. 12. The machining strip width is then obtained by finding intersections of the effective cutting shape with the virtual arc as follows.

Let P be an arbitrary point on the cutter bottom edge (see Fig. 11). We consider an angle θ required to turn the y_c axis around the z_c axis in such a way that the negative y_c axis passes through P . Furthermore, angles corresponding to the left and the right intersections P_l and P_r are denoted by θ_l and θ_r , respectively (Fig. 12). It is not hard to demonstrate that the left and the right machining strip w_l and w_r are given by

$$w_\mu = r |\cos \omega \cos \theta_\mu - \cos \lambda \sin \omega (1 - \sin \theta_\mu)| \quad (3.2.1)$$

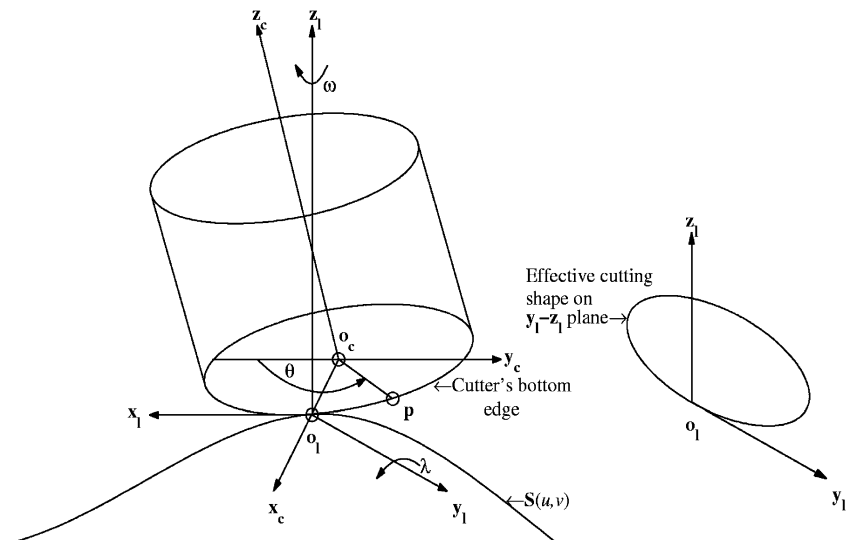
where $\mu = l$ or $\mu = r$. The entire machining strip width is then

$$w = w_l + w_r.$$

Note that the local surface cross-section at the CC point seems to be better approximated by a circular arc [14] than by the parabola suggested by Lee and Ji [115].

The local gouging or the curvature interference is defined as the excess material removal in the vicinity of the cutter contact point due to the mismatch in curvatures between the tool cutting edge and the desired surface. Detecting and avoiding local gouging requires matching up the curvature of the effective cutting shape (also referred to as the effective cutting curvature) with the normal curvature of the surface evaluated in the same plane. If the effective

Fig. 11 Geometric analysis of the flat-end cutter



cutting curvature is greater than the normal curvature of the surface then local gouging does not occur.

The effective cutter radius r_e is given by [127, 128].

$$r_e = ra^2 \left(\frac{1+b^2}{a^2+b^2} \right)^{3/2} \quad (3.2.2)$$

where $a = \sin \lambda \cos \omega$, $b = \tan \lambda \sin \omega$.

The curvature of the effective cutter shape is then given by

$$k_{\text{eff}} = \frac{\sin \lambda}{r \cos^2 \omega} \quad (3.2.3)$$

To optimize the machining strip width, λ and ω are usually set so that r_e is the best match to the radius of curvature at the CC point. For convex or planar surfaces, the tool inclination angle λ is set to a small default angle or zero and the tilt angle ω is set to zero as well. If the surface

is non-convex, a non-zero tool inclination angle λ is needed to avoid gouging.

Consider a flat-end cutter shown in Fig. 11. Gouging occurs whenever a point on the circle touches or goes inside the surface. Let G be a gouging point (see Fig. 13a). The line connecting the two points, O_1 and G , forms a chord on the circle. Denote the angle between this line and $\overline{O_1 O_c}$ by ϕ (see Fig. 13a, b). Let λ_ϕ be the tool inclination angle that corresponds to a specific ϕ . The minimum tool inclination angle to avoid gouging is then

$$\lambda_{\min} = \max_{-\pi/2 \leq \phi \leq \pi/2} \lambda_\phi. \quad (3.2.4)$$

It is not hard to demonstrate that for a non-convex surface

$$\lambda_{\min} = \sin^{-1}(rk_{\max}), \quad (3.2.5)$$

where $k_{\max}$ is the maximum surface curvature at the CC point. Clearly, for a convex surface an inclination is not

Fig. 12 Machining strip width estimation

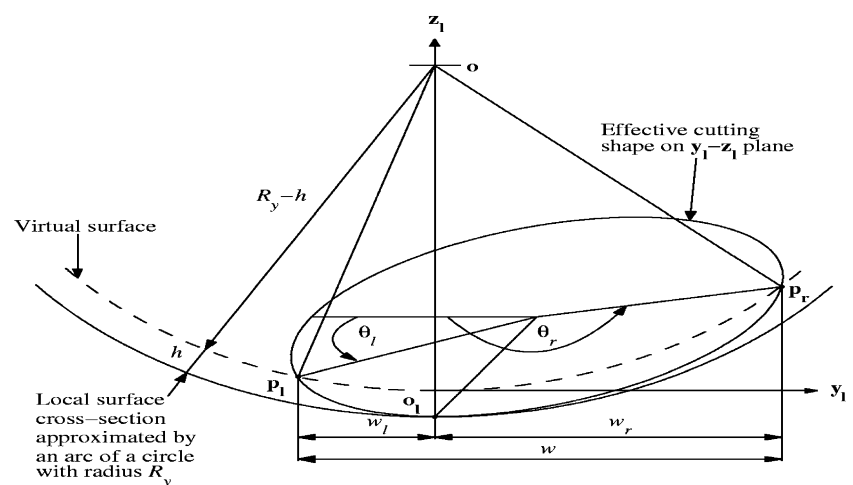
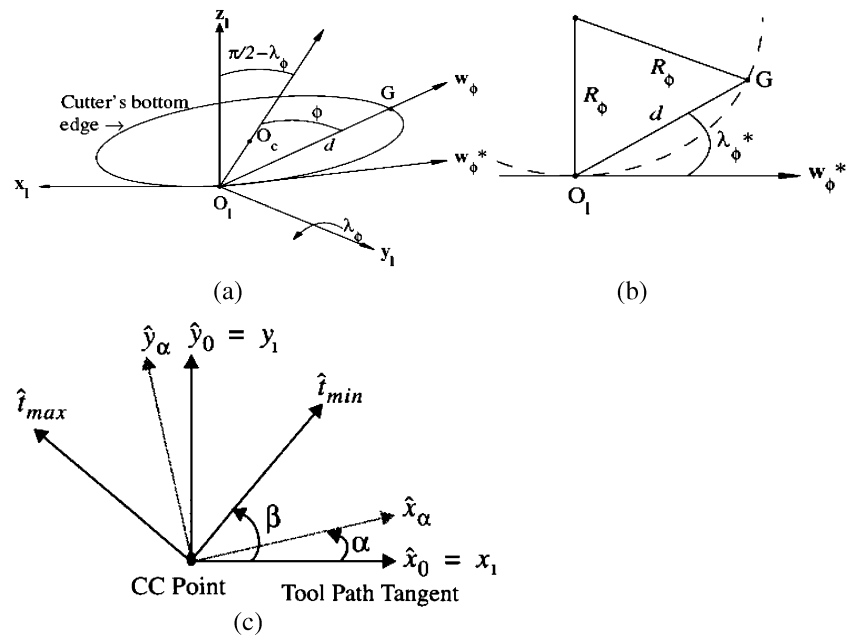


Fig. 13 **a** and **b** tool gouging, **c** principal directions at the CC point



required, so $\lambda_{\min}=0$ (from technological viewpoints a small inclination angle is still recommended). Furthermore, it can be shown that for flat-end cutters the orientation $(\lambda, \omega)=(\lambda_{\min}, 0)$ maximizes the machining strip. If gouging cannot be eliminated by inclining the tool alone or the inclination angle λ requires rotations which exceed the limit of the machine, the tilt angle can be optimized or a smaller tool size must be used.

Note that Eqs. 3.2.1–3.2.4 follow seminal works of Lee and Chang [215]; Lee and Ji [115]. Similar analysis of the fillet end mill has been published in Lee [114].

Of course, gouging is still possible, because the curvatures are compared only in one section. In order to eliminate this error, Lee and Ji [115] compare the curvature of the effective cutting shape evaluated in two planes: along the tool path and along the normal to the tool path. The effective cutting curvatures are compared with the normal curvatures of the surface in the respective planes and the inclination angle is computed as the maximum of the two minimal inclinations. Unfortunately, the method is not applicable to non-convex surfaces when the radius of the curvature of the part surface is negative in the both directions but the maximum principle curvature is positive. In these cases, the method produces a zero inclination. This “bug” often leads to local gouging. Lo [128] solves this problem by continuously checking for gouging in all directions. Some improvements and modifications of these techniques are also given in Anotaipaiboon and Makhanov [14].

Pottmann et al. [164] proposed a local millability criterion which also guarantees global millability (i.e. rear-gouge and collision-free milling) for three-axis machining using ball-end tools. The local millability criterion is based on curvature matching, using Dupin indicatrices in the

tangent plane at the CC point, between the designed surface and the tool swept surface [218]. A five-axis version of this method is presented in Yoon et al. [220].

Furthermore, Rao and Sarma [178] presented a general closed form, coordinate free method for the detection and elimination of local gouging using flat-end tools. The method is based on finding the curvatures of the tool swept surface at CC points along the tool path. Local gouging can then be detected and eliminated by sampling a finite set of points on the tool path, while comparing curvatures of the tool swept surface and the designed surface.

Consider a tangent plane (x,y) at a CC point (Fig. 13c). The effective surface curvature k_α for the direction $\hat{x}_\alpha$ is the normal curvature of the surface in the cutting plane $(\hat{x}_\alpha, \hat{z}_\alpha)$ at the CC point.

An expression for the effective surface curvature can be found by using Euler’s formula for normal curvatures as follows:

$$k_\alpha = k_{\max} \sin^2(\alpha - \beta) + k_{\min} \cos^2(\alpha - \beta), \quad (3.2.6)$$

where $k_{\min}$, $k_{\max}$ are the maximum and minimum principal curvatures with the corresponding principal directions $\hat{t}_{\min}$, $\hat{t}_{\max}$ and where β locates the principal curvatures in the tangent plane (see Figs. 11 and 13).

Elementary trigonometric relations yield

$$k_\alpha = k_{\pi/2} \sin^2(\alpha) - \frac{k_{\max} - k_{\min}}{2} \sin(2\alpha) \sin(2\beta) + k_0 \cos^2(\alpha), \quad (3.2.7)$$

where

$$k_0 = k_{\max} \sin^2(\beta) + k_{\min} \cos^2(\beta), \quad k_{\pi/2} = k_{\max} \cos^2(\beta) + k_{\min} \sin^2(\beta)$$

are the curvatures along the tool path tangent and along the normal to the tangent, respectively.

The tool swept surface in the global part surface coordinate system is represented by

$$V(t, \theta) = \begin{pmatrix} x_1 & y_1 & z_1 & S \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\omega) & -\sin(\omega) & 0 & 0 \\ \sin(\omega) & \cos(\omega) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \\ \begin{pmatrix} \cos(\lambda) & 0 & \sin(\lambda) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\lambda) & 0 & \cos(\lambda) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -r \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \\ \begin{pmatrix} r \cos(\theta) \\ r \sin(\theta) \\ 0 \\ 1 \end{pmatrix}. \quad (3.2.8)$$

where θ is the angle from y_c axis to point P on the cutter edge (see Fig. 11).

The effective cutting shape in the direction normal to the tool motion for a flat-end tool is calculated by assuming that the tool moves along a first-order approximation of the tool path (i.e. along the tangent of the tool path) at the CC point. With this assumption the effective cutting shape is given by [113]

$$E(\theta) = \begin{pmatrix} 0 \\ r \cos \lambda \sin \omega \cos \theta + r \cos \omega \sin \theta - r \cos \lambda \sin \omega \\ -r \sin \lambda \cos \theta + r \sin \lambda \end{pmatrix}. \quad (3.2.9)$$

The effective cutting curvature k_{eff} is the curvature of the effective cutting shape Eq. 3.2.8. It can be shown [169, 178] that

$$k_{\text{eff},\alpha} = \frac{\sin^2(\alpha)}{\cos^2(\omega)} \left(\frac{\sin \lambda}{r} - k_0 \right) + k_0 - \tan(\omega)(k_{\text{max}} - k_{\text{min}}) \sin(2\beta) \quad (3.2.10)$$

Note that Eq. 3.2.8 presents the curvature of the tool swept shape for an arbitrary $\hat{x}_\alpha$.

Taking $\alpha = \frac{\pi}{2}$ in Eq. 3.2.10 yields

$$k_{\text{eff},\pi/2} = \frac{1}{\cos^2(\omega)} \left(\frac{\sin \lambda}{r} - k_0 \right) + k_0 - \tan(\omega) (k_{\text{max}} - k_{\text{min}}) \sin(2\beta). \quad (3.2.11)$$

Comparing Eqs. 3.2.11 and 3.2.3 shows a significant difference when $\omega \neq 0$. However, when $\omega=0$, the both estimates produce the traditional $k_{\text{eff},\pi/2} = \frac{\sin \lambda}{r}$.

Finally, comparing Eqs. 3.2.11 and 3.2.7 implies that the local gouging occurs when

$$k_{\text{eff},\pi/2} < k_{\pi/2}. \quad (3.2.12)$$

This relationship is necessary and sufficient for detection of the local gouging.

Furthermore, the accuracy of the single-point gouging models may be insufficient. Li and Chen [121] write “Not only the parameters of the part of cutter body that pierces into the stock, but also the parameters of the area on the designed surface that may have relations to the cutter is yet to be studied. But the cutter location point, just as its name, is only the common point both on the cutter and the designed surface, any methods only based on the geometric properties of it will not obtain the best cutter positions.” As a matter of fact, the single-point methods do not guarantee gouge-free tool positions. This weakness is due to considering the geometric properties of the tool and the surface only at a single point and matching the curvature in a single plane or at selected set of planes. Consequently, these methods use secondary gouge check and avoidance strategies that complicate implementation.

Multipoint strategies [205, 206, 220] could be applied to further enhance the accuracy of tool positioning. In multipoint machining methods, the lead angle is computed by finding configurations where there is more than one contact point between the desired surface and the tool. However, the intersections are evaluated numerically, and the corresponding iterations in some cases may diverge [72]. As a matter of fact, the computational complexity of the multipoint method can be very high and still there is no absolute guarantee of a gouge-free position.

The arc intersect method [73] presents a certain compromise between curvature matching methods and multipoint methods. The tool is tilted along the feed direction since it leads to the widest machined strip. Therefore, the tool vector is constrained to lie in the tilting plane. The idea is to find the minimum tilt angle (about the corresponding cross-product vector) at which the tool contacts a second point on the surface and maintains its contact with the CC point without gouging the surface.

A new strategy is suggested in Kiswanto et al. [106]. For a constant tool orientation, the tool is lifted immediately when the specified inclination angle causes gouging with the part surface. In the case of varying optimal tool orientation, the minimum rotation angle (inclination angle) has to be found first to avoid gouging. If gouging still occurs (e.g. due to limited rotational axes of the milling machine), then the tool is lifted.

In [74], Gray et al. proposed another modification of the five-axis arc intersection method for the so-called 3 1/2 1/2-axis machining. This type of machining is characterized by

three linear axes and two temporarily locked rotary axes. The rotary axes are locked during the entire cut (resulting in a fixed tool orientation) or during a certain fraction of the cut. The rotary axes, represented by a high-precision indexing device, constitute a powerful and inexpensive alternative to five-axis machining. Interestingly enough, 3 $\frac{1}{2}$ $\frac{1}{2}$ -axis machining can be also considered as a way to solve problems associated with reduced and fluctuating feed rates commonly encountered in simultaneous five-axis machining as well as problems related to sharp angular variations near stationary points [149]. In 3 $\frac{1}{2}$ $\frac{1}{2}$ -axis machining, the rotary axes are locked while the tool is engaged with the workpiece. Therefore the machine only interpolates and moves linearly. The result is that cutting forces and surface finish are more consistent than in simultaneous five-axis machining.

Hosseinkhani et al. [85] proposed an interesting modification of the arc intersect method, applying it to each sampling point. After this transformation is done, the gouging intensity function for each sample point of the surface is calculated. The results of this calculation are saved in a two-dimensional matrix called the gouging matrix which is used to determine the minimal inclination angle.

The above methods are based on the properties of a single or several contact points and therefore errors are still unavoidable, for instance when wide strip precision machining is required. Besides, there is always a possibility of so-called rear gouging when the back side of the tool gouges the surface in an attempt to obtain a wider machining strip. When a gouge is detected the tool must be inclined further and checked for gouging again until it clears the part. This secondary check and gouge elimination can be performed using the rolling ball method suggested in Gray et al. [72]. The basic idea is to roll a varying radius ball along the tool path and position the tool inside the ball.

An original approach has been developed by Li and Chen [121]. The envelope surface created by the cutter movement is discretized into an infinite set of characteristic curves. Each of these curves exactly copies itself onto the stock. Then an analysis of the characteristic curves is performed to solve the problem of cutter positioning. The authors use the concept of instantaneous cutter position error employing the virtual cutting edge of the tool. The effective bandwidth of the cutting strip is calculated and used in the optimization algorithm.

Furthermore, even though improvement provided by the multipoint methods is undeniable, there are still many drawbacks to these strategies. For example, most methods lead to suboptimal results by overestimating the area that should be considered in the rear gouging elimination phase. Furthermore, they do not guarantee that the proposed inclination angle is the smallest. Finally, both the multipoint

and single-point methods treat gouging qualitatively without evaluating the damage made.

Finally, the above problem of millability of the part surface in the neighborhood of a particular cutter contact point is referred to as local millability. However, regardless of the tool orientation, there always exists a possibility of global interference of the workpiece with the tool holder, fixture, or other parts of the machine. There also exists a possibility that given any orientation, the tool still flank-mills an unwanted part of the surface. Moreover, it is often the case that there is no orientation of the workpiece providing access to each portion of the surface. Elber and Cohen [51] write “The problem of accessibility, or the ability to verify and possibly correct gouging into the machined surface or even into other surfaces, is apparently the most fundamental hindering factor in the broad use of five-axis machining.” Therefore, the global millability at a point is defined as the local millability combined with the absence of flank milling.

Solid modeling systems offer the possibility of doing both simulation and verification of tool paths offline. In particular, curvature interference and rear gouging can be verified, but the solid modeling approach is computationally expensive [11]. The cost of simulation using so-called constructive solid geometry is proportional to the fourth power of the number of the tool movements [26].

On the other hand, the solid model detects both local and global interference, including collisions with the clamping device and machine parts. Therefore, it is suitable for global millability checks. However, a typical G-code could contain more than 10,000 tool positions and orientations. Therefore, current solid modeling research focuses on efficient and fast algorithms to compute the swept volume of the tool and perform Boolean operations to subtract the intersection from the stock.

The partition into elements and the corresponding data structures are the most important components of these procedures. The Z-buffer structure [13, 189], ray representation [88], Octree method [155, 174], *K-D* trees [80], *BSP* trees [154], B-rep indices [200, 103], tetrahedral meshes [155], and regular grids [69] are examples of such spatial decomposition techniques. Each solid modeling algorithm has advantages and disadvantages in terms of accuracy, robustness, data structure, and computation time. However, it seems that the simplicity of the data structure required for the Z-buffer scheme and the possibility to generate and update the part model very quickly have influenced many commercial CAM programs to use the Z-buffer algorithm or its extensions for NC code verification and optimization [225].

Bohez et al. [26] presents a short introduction to solid modeling schemes such as the extended Z-buffer algorithm [82], line graphic simulation approaches [95].

Recent research papers include many improvements on the Z-buffer techniques such as the enhanced Z-buffer model [116, 225], the stencil buffer [26], the adaptive depth buffer [173], and the undo facilities for the Z-buffer scheme [21].

However, the above methods are not designed specifically for global interference detection. As a matter of fact, the solid model visualizes a general cut which may or may not include global interference. It may take hours of simulation and possibly an operator to visually detect collisions. It is undeniable that solid modeling systems such as Vericut or UG provide a very good simulation of real cutting. However, as is the case with the real system, after hours of virtual cutting, it is often hard to say what was the reason for a particular inaccuracy [84].

Elber and Cohen [51] note that “A first attempt at the global interference problem might be to expand the Z-buffer idea by one-dimension and create a volumetric representation of the entire stock that is encompassing the machined model”. However, “It would entail the use of a huge amount of memory.” For instance, consider a small 1-in. model of a cube approximated with a moderate tolerance of a thousandth of an inch. For this unit size stock and a resolution of one thousand, one is required to provide $1,000^3$ or one billion volumetric cells.

Therefore, methods based on closed-form mathematical solutions or their approximations are still valuable.

The problem of global gouging can be treated mathematically using the concept of accessibility. The accessibility of a point in a given direction is defined as follows: a surface point is accessible in a given direction if a ray can be drawn from it in the given direction without intersecting the interior of the surface.

The problem of accessibility in three-axis machining can be solved by a method of hidden surface removal of the same scene from a direction collinear with the tool axis [47, 83]. The fact that the tool has a finite thickness can be compensated for, by offsetting all the check surfaces by the radius of the tool.

Moreover, for three-axis machining, under certain conditions, the absence of local gouging implies a complete absence of collisions [165].

In 2000, Wallner and Pottmann [201] proved a global millability theorem for general workpieces see also [164]. They analyzed several possible configuration manifolds of tool positions relative to a workpiece under different aspects: the number of degrees of freedom of tool motion, the correspondence between the contact point and the tool position, and the presence or absence of unwanted collisions between the tool and the workpiece.

Unfortunately, the three-axis methods are not applicable to the five-axis case. For example, when the tool is oriented along the normal of the surface, two normals emanating

from two different points on the surface might intersect at the same point. Therefore, a point on a check surface could affect two different, disjoint points on the desired surface. Consequently, the collisions in five-axis mode require sophisticated mathematical methods and enhanced computer graphics.

Takeuchi et al. [190], Takeuchi et al. [191] and Takeuchi and Watanabe [192] proposed a method for computing collision-free CL data using a trial and error approach.

Morishige and Takeuchi [146] and Morishige et al. [147] used so-called *C*-space techniques to generate a smooth, continuously varying tool path without collisions. The *C* space is a general concept of robotics where the configuration of a mechanism is specified by a sequence of values. A rigid body, for example, can be located in space by specifying six parameters related to its six degrees of freedom. The configuration space (*C* space) of a mechanism is the space of these parameters, and a point in the *C* space specifies a particular configuration. Obstacles can be mapped to the *C* space as well, and the required collision-free access can theoretically be inferred by navigating the point in the *C* space around the obstacles. Unfortunately, though intuitive and intellectually appealing, the *C*-space approach could lead to computationally intractable tasks.

An interesting approach based on a physical analogy with electrically charged bodies was suggested by Cho et al. [34]. When the cutter and part surfaces are virtually charged with static electricity, the potential energy is stored. When the cutter approaches the part surface, the stored potential energy increases. In the case of a collision, the energy increases enormously. Meanwhile, a relatively small amount of energy is stored for collision-free cases. Although interesting, the method does not provide a way to efficiently threshold the energy to specify whether a local or global collision occurred or whether the cutter just approached the part surface.

Lee and Chang [113] utilize convex hulls to find a feasible set of tool orientations. In the case of a collision, a correction vector is calculated in the direction opposite to the surface normal vector at the point of interference.

Lauwers et al. [112], describe multi-axis tool path generation software in which the tool orientation is optimized to avoid machine collisions and at the same time to maximize the material removal rate along the tool track. To perform efficient collision avoidance, the tool path generation module, the post-processing module, and the machine simulation have been integrated into one system. Once a collision is detected, the collision vector is derived from two objects: the center of the collision curve and the curve of intersection of the machined part with the cylindrical approximation of the tool, in the direction perpendicular to the tool axis. This is used later to calculate the correction vector. Unfortunately, these techniques

assume a cylindrical approximation of the tool, which is not always the case.

[165] show that if in five-axis machining all axis positions pass through a fixed point and if all points of the workpiece surface can be seen from this point, then local millability implies global millability. However, first of all, this is only a necessary condition. Second, the result only applies to a special class of so-called star-shaped surfaces defined by the condition that there exists a point called a guard such that the guard can see every point on the surface.

However, the above tells us that accessibility can be represented by a somewhat simpler concept of visibility. A point on an object is visible from a point at infinity if there exists a straight line segment connecting the two points which does not intersect with the object. Visibility is a useful precursor for the accessibility computation because, for a certain class of tools, visibility is a necessary condition for accessibility.

Seminal theoretical results on visibility were obtained by Woo and Turkovich, [209], Woo and Gan [210], Woo [211], Elber and Cohen [48], and Elber and Zussman [50]. In particular, the concept of visibility maps derived from Gauss maps [46] to define mutual visibility of points on the surface was studied by Woo and Turkovich, [209] and by Woo [211]. The problem of finding suitable setups for composite surfaces is dealt with in Woo and Turkovich [209]. Relevant solution methods and computational issues are discussed in Chen and Woo [29].

Later research focuses on global accessibility using so-called visibility cones [see 98, 199]. The visibility cone is defined as the feasible range of the tool axis for a surface point. In other words, it is a set of directions along which the tool can approach a part of the workpiece without intersecting with another part of the surface as shown in Fig. 14. The visibility cone and the corresponding spherical map are measures of the surface complexity, that is, when the visibility cones are scattered on the sphere, the corresponding surface has a complex shape. Translating the apex of the local visibility cone to a fixed center gives a map on the unit sphere called the spherical map or visibility map (*V*-map). The *V*-map represents all the orientations

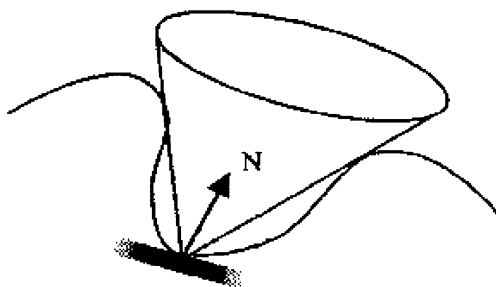


Fig. 14 Point visibility cone

along which a given tool can access and machine the point on the surface.

If a machine can produce a part surface in one setup without violating the workspace limits, we shall call this surface *machinable*. The work space is defined by the range of the linear and rotational axes. However for convenience it is often assumed that the motion ranges are confined to the rotational axes. This is based on the practical observation that it is the rotational axes that usually effect the machinability.

For a three-axis machine without a rotation axis a single tool motion is represented by a point on the sphere (Fig. 15a). A four-axis machine with one rotational axis is represented by an arc on the sphere (Fig. 15b) and the five-axis machine with two rotational axes is represented by a fan or a rectangle (depending on the machine configuration) as shown in Fig. 14c and d.

A point is accessible only from the directions that lie above the tangential plane at that point. In other words, the feasible tool direction vectors lie within 90° from the surface normal vector at the point. Each point visibility cone lies completely inside a hemisphere centered at the point and placed above the tangential plane at the point.

Furthermore, the size of the feasible tool motion is defined by the rotational axes range. A *k*-axis feasible tool motion depends on the position on the sphere *T* and the range of the rotational motions, Ω . In a four-axis configuration $\Omega = (O^-, O^+)$ whereas in the a five-axis version $\Omega = [(O^-, O^+) \times (\Phi^-, \Phi^+)]$ (see Fig. 15).

The visibility maps, the *k*-axis feasible tool motion and the point visibility cone are the fundamental means for solving the machinability problems.

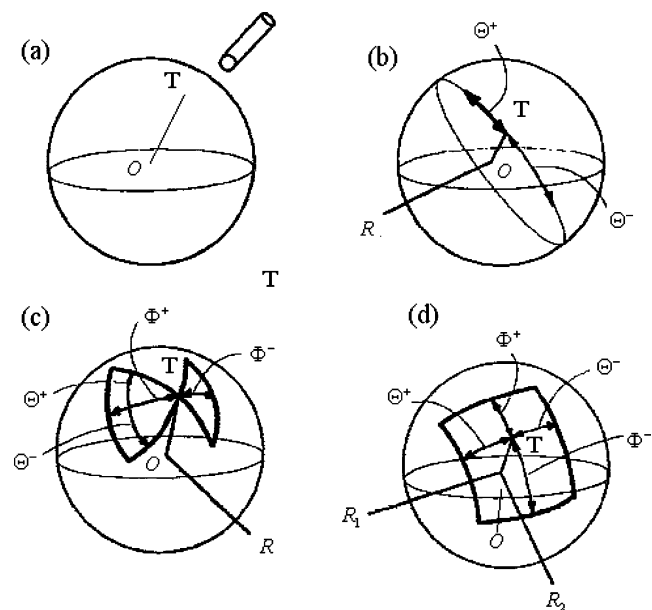


Fig. 15 Feasible tool motions of a three-axis machine, b four-axis machine, c and d two types of the five-axis machine motions

A point visibility cone can also be considered as subset of a complete sphere and hence a set of all accessible directions for a point can be represented by a sphere centered at the point, also called a binary spherical map (BSM). A BSM is a discretized unit sphere decomposed by tessellation into a finite number of triangles. Each triangle on the sphere can be labeled 0 or 1 depending upon whether a ray projected through the center of the patch to the centroid of the sphere provides access to the point or not [see 98]

Tseng and Joshi [194] use the visibility maps to determine the accessibility of a tool given its point location, exploiting the subdivision and convex hull properties of the free-form surface representations to compute directional bounds.

Kang and Suh [98] present visibility maps combined with the convex hull techniques designed for several (but not all) types of industrial five-axis machines. The algorithms are very complicated, including many “if-then” rules. Besides, they have not been fully verified by practical machining. Furthermore, in five-axis machining, the intersection of the local visibility cones derived from each subsurface of the part does not give global visibility. The planar projection approach proposed by Kang and Suh may be computationally intractable. Finally, the full variety of possible five-axis configurations is not considered.

Elber and Zussman [50] extend the idea of local visibility to the so-called α -sensitivity to express the sensor's physical constraints. A point is called α sensible if the angle between the corresponding normal on the Gauss map and a certain fixed direction is not less than α . The goal is to subdivide the freeform surface into regions that fit into a single visibility cone. The method decomposes the surface into regions, so that each region is α sensible from a selected direction. The selection of the directions guarantees complete coverage of the surface. However, the method is suboptimal and has been designed for laser scanning without considering the specific five-axis issues such as local millability, shape of the tool, and nonlinear kinematics.

It is also worthwhile to note that the orientation of the tool does not necessarily follow the normal of the surface. In the case of the flat-end tool the orientation is derived from the local millability constraints. Furthermore, the tool may follow orientations that go through a point to machine cavities with negative slopes. One might define machining operations in other ways, for example going through a curve with some specific orientation. In this sense, the regular hidden surface removal solution is a special global accessibility problem for which the prescribed vector field follows the viewing direction.

Elber and Cohen [51] formulate global accessibility as follows: given a part surface $S(u, v)$ a vector field that prescribes the orientations of the tool $O(u, v)$, and a check

surface $K(s, t)$, find all regions $\wp(u, v)$ accessible with respect to the check surface (or all regions $\wp'(u, v)$ inaccessible with respect to the check surface, see Fig. 16). Suppose that $\frac{\partial S(u, v)}{\partial u} \neq 0$.

Define

$$O_1(u, v) = O(u, v) \times \frac{\partial S(u, v)}{\partial u},$$

$$O_2(u, v) = O(u, v) \times O_1(u, v).$$

The vector fields above span the plane orthogonal to $O(u, v)$. Finally, a point lies on the boundary between $\wp(u, v)$ and $\wp'(u, v)$ if and only if

$$\begin{cases} (O_1(u, v), S(u, v) - K(s, t)) = 0 \\ (O_2(u, v), S(u, v) - K(s, t)) = 0 \\ (n_K(s, t), S(u, v) - K(s, t)) = 0, \end{cases} \quad (3.2.13)$$

where $n_K(s, t)$ is the normal to $K(s, t)$.

In other words we deal with three nonlinear equations in four unknowns. In particular when $O(u, v) = n(u, v)$, where $n(u, v)$ is the surface normal, we have

$$\begin{cases} \left(\frac{\partial S(u, v)}{\partial u}, S(u, v) - K(s, t) \right) = 0 \\ \left(\frac{\partial S(u, v)}{\partial u}, S(u, v) - K(s, t) \right) = 0 \\ (n_K(s, t), S(u, v) - K(s, t)) = 0 \end{cases} \quad (3.2.14)$$

Providing the global solution has many advantages over point-by-point local validation and correction approach.

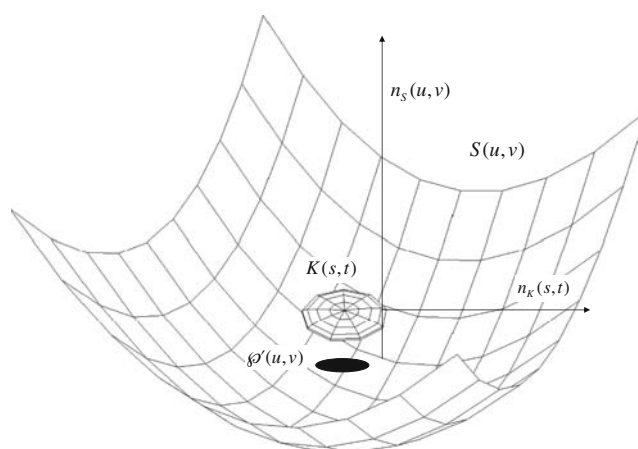


Fig. 16 Global accessibility

First, the solution is independent of the selected tool path, as it solves the general accessibility problem that is not specific to some given tool path. Furthermore, by considering the global picture, global algorithms can potentially be made more accurate and efficient. When a tool path is validated on a point-by-point basis, no guarantee can be made on possible gouging between the verified contact point locations. In contrast, this guarantee is fundamental to global algorithms and, in fact, is cost-free once a global algorithm is employed.

Equations 3.2.14 are capable of answering the fundamental global question: what regions of a surface can be machined from an arbitrarily prescribed orientation field and not gouge into the given check surface. Of course, an iterative numerical solution is needed. Consequently, the user may experience problems such as divergence or slow convergence.

As of yet, there have not been many examples of application of the proposed techniques. A possible reason is that conceptually, solid modeling can be easily understood by mechanical engineers whereas methods of differential geometry require sophisticated analysis.

There have been many interesting five-axis collision detection methods designed for particular parts. For instance, Gian et al. [70] developed a method to find open regions (the ones that can collide with the tool) using slicing, Boolean operations, and geometric analysis in the regions. The geometric relationship between the CC point and the contour of the open region is considered to obtain an initial orientation of the tool. With the orientation, collisions are inspected to ensure that the orientation can feasibly prevent interference between the cutting tool and the cavity surface.

Young et al. [222] presented a new parametric method with an approximate constant cutting depth for the rough machining of an impeller. The initial tool spindle axis is considered as the initial orientation to determine the cutting tool posture for which the variation of rotational axes of the five-axis machine tool will be reduced.

Kim et al. [99, 100] presented a cutter location (CL) surface deformation approach for a five-axis tool path generation. The proposed algorithm was applied for machining of a cooling pin (a small part used to remove the heat from the chips of the computer) and a tire mold. Lee and Lee [116] present an interference-free tool path generating method of a spatial cam. The cutting tool is confined within the meshing element and the motion of the cutting tool follows the meshing element so that collision problem can be avoided. Unfortunately, many elements of the above procedures are problem-dependent and cannot be generalized.

In [17], Balasubramaniam et al. developed a method for five-axis tool positioning that accounts for accessibility of

the tool using visibility maps of the triangulated data. Using this visibility data for finish machining, the authors show how it can be used to generate globally collision-free five-axis finishing tool paths while considering machine limits, tool tilt, cusp height, and tool pitch limits.

Ilushin et al. [90] exploited the axial symmetry inherent in the tool's rotational motion to construct a polygon/tool intersection algorithm for global collision detection. The algorithm allows testing for collisions between arbitrarily shaped tools and tool holders, clamping devices, and the rotating table. The workpiece and the parts of the NC machine are given in a polygonal representation. An arbitrary polyline representation of the tool is allowed. Given a potentially interfering set, the authors derive planar hyperbolic segments that originate from the radial projection of the triangles' points around the tool's axis onto a plane for each triangle in the set. These hyperbolic segments are then tested for intersections with the tool's profile and all other rotating parts such as the chucks and spindle. Such intersections identify collisions between the rotating parts of the NC machine and the workpiece or other stationary parts. A very important advantage of this approach is that the collision tests are independent of the initial orientation of the model. This dependency is a significant drawback of many contemporary voxel-based alternatives. On the other hand, the algorithm considers possible collisions only at the contact points. Obviously, collisions could occur between contact points and hence could be missed.

Hsueh et al. [86] propose to prevent collisions using two-stage techniques. The first stage is to obtain the tilting and collision-free angle range in the plane that is normal to the tool path obtained. Next, a checking cone generated from this collision-free tool axis range is used for the second collision check. The collision region is formed by the intersection of the neighboring surfaces.

Analyzing a proper sculptured surface orientation on the worktable of a multi-axis CNC machine, Radzevich and Goodman [167] proposed the so-called spherical indicatrix of the sculptured surface machinability. This characteristic curve indicates whether the sculptured surface is machinable under a known scenario. The theory is developed in connection with a sculptured surface orientation on the worktable of a multi-axis CNC machine.

Radzevich [168] presents an approach that enables us to detect regions of a sculptured surface which are not accessible for a cutting tool of a given design. Furthermore, if any unmachinable regions exist, the approach enables us to subdivide the sculptured surface into the cutter-accessible and the cutter-inaccessible regions.

Still it is not clear whether solid modeling- or differential geometry-based methods produce better results. Perhaps the solution is a suitable hybrid involving both approaches

combined with a specific CAD [5] representation of the part surface such as IGES or STL [106].

An interesting algorithm which exemplifies the above-mentioned combination of the two approaches has been proposed by Tang et al. [197]. The algorithm detects collisions between the tool and workpiece and also between the other parts of the CNC machine. Changing workpiece geometry is included in the detection process. The workpiece and machine bodies are represented by an octree of bounding spheres. Collision detection is then conducted between these spheres. When interference is detected between the spheres in the last octree level, the slices within these colliding spheres are further checked using the sweep plane algorithm.

Finally, CAM for five-axis machining should be able to analyze the machine limits, global and local collisions, rear gouging, and kinematics errors, thus providing an appropriate interpolation to generate feasible gouge-free tool positions and orientations. Several noncommercial systems such as Xu et al. [214] and Lauwers et al. [112] present a suitable integration of some these components. However, none of the up-to-date five-axis modeling tools is capable of providing a fully integrated solution for an arbitrary desired part.

3.3 Adaptable geometric patterns and grid generation methods

Five-axis machining offers an improvement in both rough and finish machining efficiency over three-axis machining. In five-axis mode, the tool orientation relative to the workpiece can be controlled by two additional degrees of freedom so as to achieve higher machining efficiency. A milling machine can be considered as a special type of robot, and for robots, optimal control is a well-known theory. However, the robot theory seeks to follow a given path, whereas milling machine paths are not pre-specified, but must be designed in order to cover the entire area. Besides, the tool must be inclined in a special way, machining strips should be maximized, and geometric errors should be minimized. These and other differences gave rise to a large number of tool path planning methods developed specifically for five-axis machining. Many of these methods adapt the tool positions offline to minimize machining errors. Note that the tool positions along a path cannot be assigned independently. They must follow a certain connecting pattern. The patterns usually are composed of curves along which the CC points are allowed to move. The simplest patterns are the zigzag and the spiral paths.

Furthermore, as opposed to the general robot trajectory planning problem, the connecting curves usually are not allowed to intersect. Although this requirement is not a must, the majority of tool path generation methods follow it. One of the main reasons is that paths including intersecting curves are likely to be longer than those without intersections.

Although in general this is not always the case, it has a heuristic value in not performing the same cut twice. Besides, self-intersecting tool paths could create unpredictable patterns of scallops which might be hard to eliminate.

Another desired feature of the designed tool path is that the number of tool retractions is minimal or zero in order to reduce cutting time and minimize the risk of tool marks due to jumps and plunge on the machined surface. Thus, the best solution is a single continuous curve running across the entire surface, even though as far as machining time or tool path length is considered, the single curve does not necessarily outperform machining with retractions.

This section presents adaptable geometric patterns for tool path generation. Adaptation is performed using criteria representing certain characteristics of tool path quality such as kinematics error, scallop heights, undercuts/overcuts, and the continuity of the path. The section partly covers methods for complex pocket milling employing geometric patterns capable of following the boundary of a pocket.

Conventional tool path planning employs structured zigzag or spiral patterns due to their simplicity and their ease of computation [see surveys by 45, 178]. Still, adaptation of the positions of the CC or CL points along the zigzag or spiral is needed.

The simplest adaptation deals with spacing between the points by using a variable forward step. In this approach we select the forward step with regard to geometric errors appearing due to the discrete nature of the prescribed tool path. We assume that the trajectories of the tool tip between the CC points are linear. We extract a reference curve from the desired surface and compare it with the linear trajectory. This is done one step forward at a time. In other words, given the initial CC point the forward step is selected by assigning the maximum allowable deviation between the desired curve and the straight line.

Choi and Banerjee [36] define the forward step as the maximum distance between CC points on the current tool path in which the chordal deviation does not exceed the given tolerance. The forward step found by bisection or another inexpensive numerical methods converted from the physical domain into the parametric domain.

The problem can be further simplified by introducing a certain approximation to the reference curve, for example, employing arcs or polynomials. In that case some closed-form solutions can be found in Li et al. [119].

A slightly more accurate technique evaluates errors produced by several selected points on the tool, for instance, Pi et al. [162] consider the CC points and the CL points at the same time.

Numerical procedures can also be replaced by a closed-form solution found by Choi and Banerjee [36] for the three-axis case under the assumption of the Lipschitz continuity of the part surface.

Observe that the above methods based on the chordal deviation are assumed by the most previous studies in five-axis machining as producing accurate results [125]. However, despite the simplicity of the chordal deviation measure, the approach may lead to considerable inaccuracies and, often, it is not recommended. Note that in five-axis machining, even the assumption that the error gets smaller as the CC points get closer is not actually correct at singularities such as ridges or stationary points [23, 134].

Fortunately, the forward step technique applies to the nonlinear case as well. All we have to do is to evaluate the error, taking into account the kinematics of the machine. The maximum forward step providing the maximum allowable tolerance can be found using a numerical approach. However, in the nonlinear case the solution becomes machine-dependent. Moreover, it depends not only on the machine configuration but on the particular setup of the part surface on the mounting table.

Furthermore, the solution is not unique. For example, the step forward approach may produce different results given different initial points along the path. Besides, the tool path usually includes some fixed points, such as the turning points of the zigzag, which contribute to the multiplicity of the forward step solutions.

It is important to notice the measurement of the kinematics error presents itself a computational problem. Recall that the tool path is defined by a sequence of cutter contact points and orientation vectors. In the machine coordinates the CC points and angles are linearly interpolated.

If $W_{p,p+1}^D(t) \equiv (x_{p,p+1}^D(t), y_{p,p+1}^D(t), z_{p,p+1}^D(t)) \in S(u, v)$ is a desired curve between two tool positions W_p and W_{p+1} , it is often extracted from the machined surface $S(u, v)$, where t is a parametric coordinate along the curve. The kinematics error [136] can be defined as a total distance between the desired trajectories $W_{p,p+1}^D(t)$ and the actual trajectories $W_{p,p+1}(t) \equiv (x_{p,p+1}(t), y_{p,p+1}(t), z_{p,p+1}(t))$ generated by the machine kinematics, namely,

$$\varepsilon = \sum_p \text{dist}(W_{p,p+1}^D, W_{p,p+1}), \quad (3.3.1)$$

where $\text{dist}(A, B)$ denotes an appropriate distance between space curves $A(t)$ and $B(t)$. The difference between the space curves can be evaluated by the generic Hausdorff distance [19] or the Fréchet distance [10]. However, these measures are computationally expensive and may lead to intractable optimization problems. Some computationally simple choices are

$$\begin{aligned} \text{dist}_2(A, B) &= \int_0^1 |A(t) - B(t)|_E dt \text{ and } \text{dist}_\infty(A, B) \\ &= \max_t |A(t) - B(t)|_E \end{aligned}$$

where $|\cdot|_E$ is the Euclidian distance and $A(t)$, $B(t)$ are parameterized with regard to $t \in [0, 1]$.

Although these distances depend on parameterization, they often produce good results when the actual trajectory is parameterized with regard to the fictitious time t and the surface curve with regard to a line segment between (u_p, v_p) and (u_{p+1}, v_{p+1}) in the parametric space (u, v) . This is because when the number of inserted points is large enough the compared curves are similar, arc-like segments with their arc length close to the corresponding chord length.

The parameterization-invariant Hausdorff distance is given by

$$\text{dist}_H(A, B) = \max \left\{ \max_{a \in A(t)} \min_{b \in B(t)} |a - b|_E, \max_{b \in B(t)} \min_{a \in A(t)} |a - b|_E \right\}. \quad (3.3.2)$$

Another option is the Fréchet distance defined by

$$\text{dist}_F(A, B) = \min_{\{\alpha(t), \beta(t)\}} \max_{t \in [0, 1]} |A(\alpha(t)) - B(\beta(t))|_E, \quad (3.3.3)$$

where minimum is considered over all continuous and increasing functions $\alpha(t)$ and $\beta(t)$.

However, dist_H and dist_F are computationally expensive.

Note that $\text{dist}_H(A, B) \leq \text{dist}_\infty(A, B)$. Therefore, minimization with regard to dist_∞ reduces the error measured by dist_H as well.

Recall summed Hausdorff metrics (such as the Lindstrom–Turks mean geometric distance $\text{dist}_{HS}(A, B)$) obtained by replacing “max” in Eq. 3.3.2 by summation or integration [123]. Since $\text{dist}_{HS}(A, B) \leq \text{dist}_2(A, B)$ the minimization with regard to $\text{dist}_2(A, B)$ reduces $\text{dist}_{HS}(A, B)$, but, of course, it does not guarantee a minimum in dist_{HS} .

Finally, the compared curves can be approximated by piecewise linear functions. In this case the Fréchet distance (which is essentially the minimum equal-parameter distance between A and all possible re-parameterizations of B) can be evaluated explicitly [9]. However, the algorithm is computationally expensive and may lead to hard optimization problems.

A good option is a distance based on the natural parameterization given by

$$\text{dist}_N(A, B) = \sqrt{\int_0^1 |A(l_A(t)) - B(l_B(t))|^2 dt}, \quad (3.3.4)$$

where $A(l_A(t))$ and $B(l_B(t))$ denote the corresponding arc-length parameterizations.

Unfortunately, it is not possible to find a closed-form parameterization for real rational curves [such as NURBS 2] represented by rational functions of its arc

length [63]. Therefore, such parameterizations must be evaluated numerically.

Finally, finding a distance between curve $W_{p,p+1}(t)$ and the entire part surface $S(u,v)$ rather than between $W_{p,p+1}(t)$ and $W_{p,p+1}^D(t)$ certainly increases the quality of the error evaluation. However, the problem often requires minimizing the distance between unknown curves. Therefore, computationally hard metrics may lead to intractable optimization problems.

Finally, the tool trajectory is compared with the desired trajectory which is in some way extracted from the machined part. In engineering practice the parts are defined by standard formats such STEP, IGES, etc. For instance, the IGES represents curvilinear NURBS [2] faces glued together along the boundary edges. Therefore, the method of extracting the trajectory should include the case of the multi-patch surfaces when the curve crosses the boundary or even several boundaries.

Once the distance and the error are defined, the points must be distributed along the tool path to minimize the error. A general approach based on grid generation technology was proposed in [133, 134]. The CL points are distributed along the required curve extracted from the part surface in such a way that the total interpolation error is minimized. The grid generation approach works either with a linear or with a nonlinear model of the machine kinematics. The kinematics error contributes to a weighting function which controls the grid spacing. These techniques are illustrated in Fig. 17.

The resulting grid is topologically equivalent to two tool paths in the v and the u direction. The grid generation technologies are capable of treating complex-shaped boundaries and islands for complex pocket milling. However, the adaptations of the grid generation methods to tool path generations are often slow and require additional assumptions to increase computational efficiency. For example, in Makhanov et al. [134] in order to simplify

the computational algorithm the authors made a heuristic assumption that the error is proportional to the curvature of the part surface. However, this assumption (although realistic) has not been proven theoretically.

Observe that we deal with the distribution of the CL or CC points along a desired set of curves extracted from the part surface. In turn, these curves are connected with each other into a complete tool path using certain patterns adapted to the prescribed quality criteria.

The most popular patterns are the iso-parametric zigzag and spiral (box) patterns [175]. The term iso-parametric means that the zigzag or the spiral are generated in the parametric space u,v along one of the coordinates, say, u . The v coordinate is then used to generate the forward steps so that the kinematics error is within the required tolerance. In the case of the spiral path the parametric space is represented in polar coordinates. The radial and angular variables are incremented so that the linear distance between the tool tracks and the distance between the CL points along the tracks are controlled. Alternatively, the increments can be measured in terms of the distance on the surfaces and then translated back into the steps in the parametric space.

However, in some cases these simple methods are not applicable. For example, so-called pocket milling requires special geometric patterns designed for parts with one or more complex shaped “islands” inside [78, 79]. Special patterns are also needed for so-called trimmed surfaces, whose boundaries are defined by intersections with other surfaces.

A method called iso-planar machining follows curves that are intersections of the part surface and a series of parallel planes (Fig. 18). In other words, the cutter follows a contour map of the surface. Generally speaking, the contour map can be constructed using planes parallel to an arbitrary reference plane. Once generated, the contours are saved using the SLC [4] format developed specifically for this type of machining (see Section 1). Early papers reporting

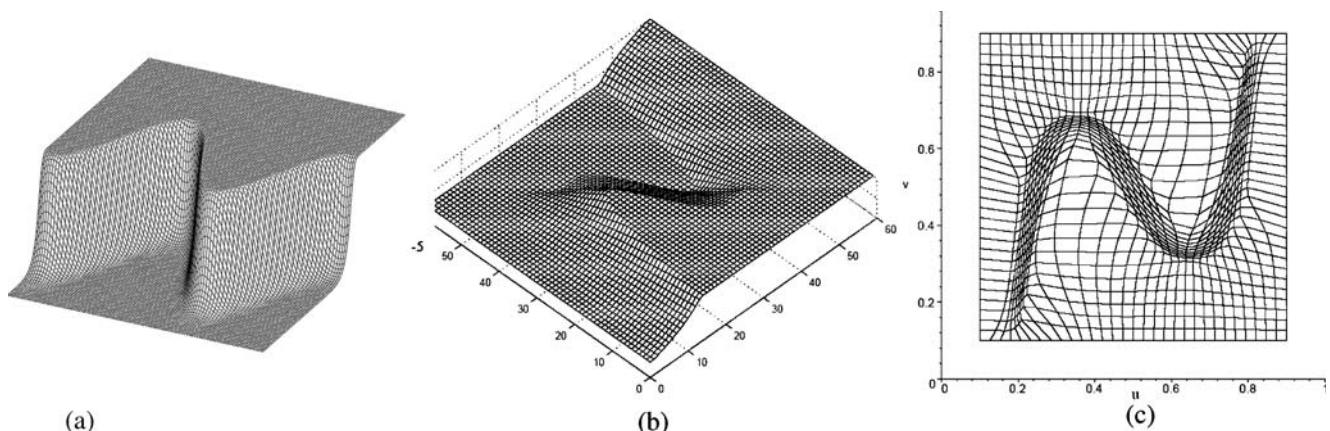


Fig. 17 Grid generation for tool path optimization **a** the part surface, **b** the control function, **c** the grid

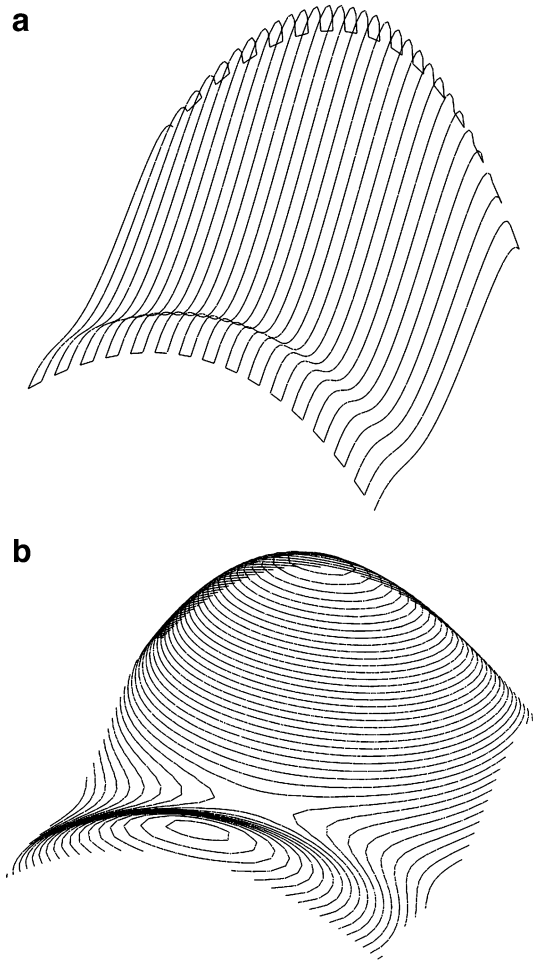


Fig. 18 Milling patterns: **a** zigzag, **b** iso-planar

these techniques are Loney and Ozsoy [131], Chen et al. [31], and Rao et al. [170].

In [35], Choi and Jerard introduced the term regional milling referring to machining operations in several regions

specified within the parametric space or directly on the part surface. The iso-planar methods are well adapted to regional milling. However, if the surface has several stationary points (Fig. 18b), the tool path consists of disconnected subpaths with possible retractions. Therefore, one of the most important problems is linking the contours in such a way that the number of tool retractions is minimized.

Furthermore, the idea of offsetting boundary or/and planar curves to obtain an appropriate tool path for pocket milling is very popular in “3+”-axis machining. As a matter of fact, contour-parallel machining is the most popular method for a two-dimensional pocket machining and three-dimensional machining of surfaces with islands. The method used successive offsets of the boundary curve as tool path elements. Offsetting the boundary could create a singly connected tool path even for a complex surface (Fig. 19). However, as opposed to the iso-planar method, the resulting curves may intersect, creating loops and ridges (Fig. 20). Besides offsetting complex boundaries can lead to disconnected paths requiring linking (Fig. 21).

One of early works on linking is Held et al. [81]. Their algorithm designed for these types of machining is based on proximity maps and Voronoi diagrams. A linking procedure requires a spanning tree of the planar graph of the monotonic pouches.

Park and Chung [157] develop a contour-parallel linking algorithm accommodating minimization of slotting, tool retractions, and drilling holes.

Park et al. [158] present a tool path linking algorithm, which guarantees zero tool retractions for contour-parallel milling. The algorithm employs the concept of a tool path element net providing information on the parent/child relationships between the regions.

Feng and Teng [65] present iso-planar piecewise linear NC tool paths for three-axis surface machining using ball-end milling generated directly from discrete measured data

Fig. 19 Simply connected boundary offset tool path

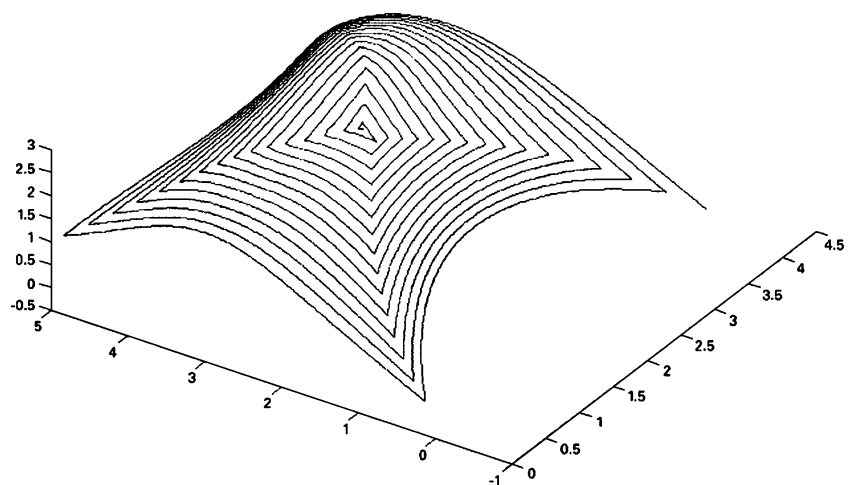
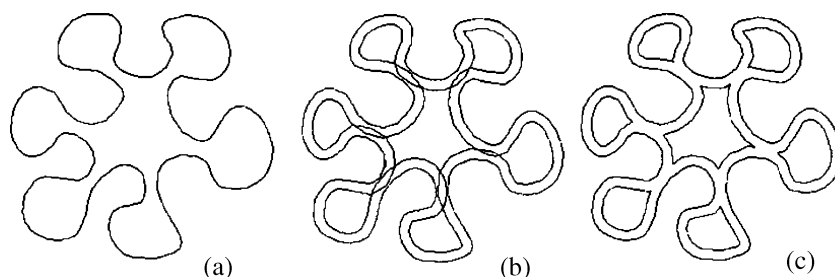


Fig. 20 Machining a pocket profile (a), offset profile (b), offset profile after the required loop removal (c)



points. However, generalizations to the four- and five-axis case are still not available.

In [156], Park and Choi proposed direction parallel machining routine which can handle multiple connected areas (Fig. 22). The input of the algorithm is a set of planar area curves, and the output is a linked set of direction-parallel tool paths. The tool path planning method for direction-parallel milling consists of three modules: finding the optimal inclination, calculating and storing tool path elements, and tool path linking solved by the so-called tool path element net traversing method. The authors claim simplicity and short computational time as compared to the contour parallel mode.

Kim and Choi [24] compare the direction-parallel path with contour parallel paths. The evaluation includes a model of the machining time based on the assumption that the velocity of the CN machine is a linear function of the distance in the acceleration, deceleration, and uniform stages. The study shows an advantage of the direction parallel smooth zigzag (C type and S type connections) regardless of the feed rates and the tool path offset. However, it is clear that the study cannot be generalized to an arbitrary part surface.

Jeong and Kim [93] and Lai et al. [109] present algorithms designed to offset boundary curves in a geometrically complex region using the Voronoi diagram. Each curve segment is offset within the corresponding Voronoi polygon to avoid the degeneracy problem; however, from the practical viewpoint it is not easy to implement a

reliable and efficient Voronoi algorithm applicable to an arbitrary boundary.

Jeong and Kim [94] introduce a distance map approach to effectively find characteristic points and self-intersection points of the offset curve segments and to eliminate such topological problems as loops, ridges, and cusps. An adaptation of these ideas to spiral high-speed machining is presented by Lee [117].

A forward locus tracing method is introduced in Lai et al. [110]. The algorithm searches for all intervals split by intersections of planar curves then maps the 2D transversal intersections onto 1D interval identifications. This proposed mapping simplifies the structure of tasks and can be integrated easily into CAD [5] systems.

A promising technique to remove possible loops appearing in the boundary offset methods is the so-called rolling ball method, similar to that used to avoid local gouging in the cutter positioning problem. However, this time gouging is defined as intersection of the rolling ball and the boundary contour. This idea is exploited in You et al. [219]. Furthermore, avoiding boundary gouging is crucial when converting CC data into CL data for arbitrary pockets involving islands.

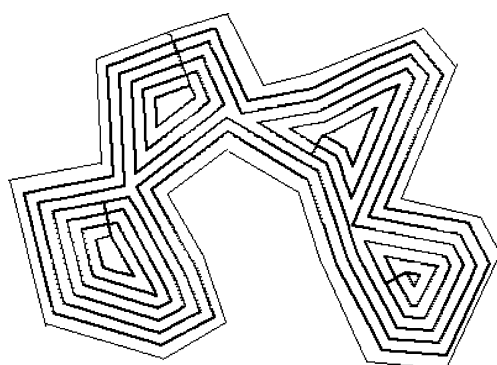


Fig. 21 Contour parallel tool path with links

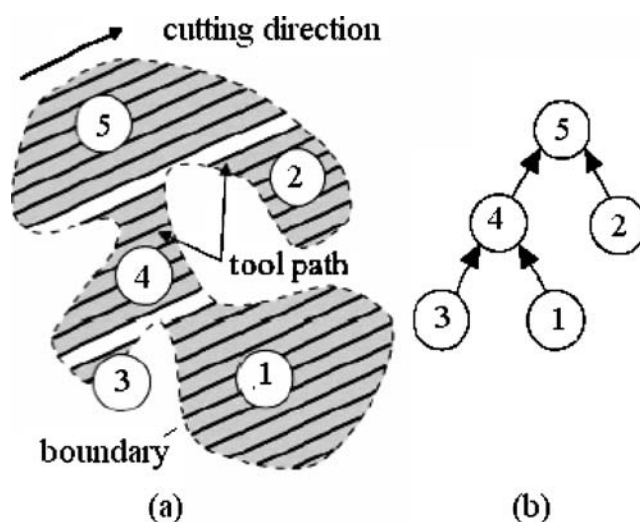


Fig. 22 Direction-parallel milling (a) tool path and the regions (b) region connectivity graph

Sheen and You [180] propose the use of so-called pair-wise bridges to merge pockets and islands. They show that the proposed pair-wise bridge method leaves very small-sized scallops. An alternative Boolean method is designed to solve the problem of pocket machining including numerous islands. However, the method is very hard to implement since it involves multiple complicated geometric constraints. There is still no guarantee that it applies to an arbitrary sculptured surface.

Observe that the above iso-parametric and iso-planar methods calculate the distance between consecutive tracks based on the maximum allowable scallop height. Therefore, a conservatively small interval for incrementing the cutter paths is selected. In other words, the minimal allowable step is taken from all the maximum allowable (point wise) distances. Otherwise the curves are no longer iso-parametric or iso-planar. Consequently, the method does not produce constant scallop heights across the entire surface. Thus, the machining strip is undervalued and as a result, machining efficiency is limited.

The above observations lead to the concept of iso-scallop tool path generation. Methods to maintain a constant scallop height, called iso-scallop machining methods, were first proposed in Suresh and Yang [188], Lin and Koren [122], and Sarma and Dutta [177] for three-axis ball-nose machining. Observe that the scallop depends on many factors for instance the direction of the tool motion. Therefore, scallop height is not a 2D surface, hence 2D contour map of scallops for an arbitrary sculptured surface cannot be defined. Consequently generation of the iso-scallop tool paths is more difficult problem than that of the iso-planar paths. Additionally as noted by Sarma and Dutta [177] the iso-scallop paths may lead to cusps and self-intersections when the variation of the surface curvature is large.

The iso-scallop methods often use swept sections of the tool along the tracks to calculate the tool path intervals. The swept sections of the tool are assumed to be coplanar or

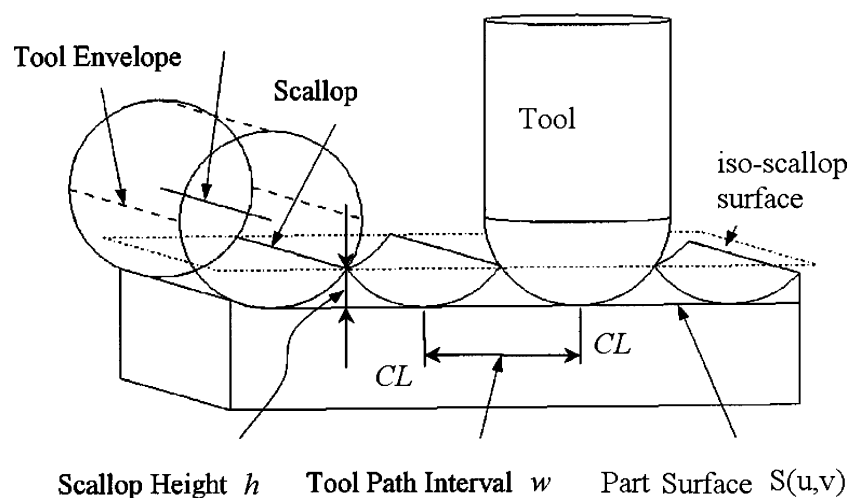
perpendicular to tangent vectors to a common scallop curve. This means that the 3D scallop evaluation problem has been reduced (under questionable assumptions) to 2D calculations in the corresponding section of the desired surface. Figure 23 depicts the elements of the scallop evaluation for ball nose and the inclined flat-end cutter. Figure 24 shows scallops for the flat-end cutter along the surface normal for a convex surface.

Feng and Li [64] extend the swept section framework to the 3D case. As the tool moves along the tool path, the tool envelope surface is created by sweeping a circle which represents the cutter. The horizontal distance between the adjacent tracks is referred to as the side step or tool path interval. The scallop curve is defined as the 3D curve along which the machined scallops are equal. The curve satisfying this definition is not unique since it depends on the shape of next track of the tool path. Feng and Li's iso-scallop machining method is based on the fact that the scallop curve is the intersection of the two adjacent envelope surfaces and the scallop surface. The method improves on the accuracy of iso-scallop machining because it works directly with 3D envelope surfaces rather than with the projections of the effective cutting shape. However, since the two adjacent envelope surfaces are not defined uniquely, the problem has multiple solutions.

Yoon [221] reports an iso-scallop height method for three-axis ball-end milling similar to Feng and Li [64]. As opposed to the previous approach, which uses the bisection method to search for scallop curves and tool center curves, the author applies Newton's iterative algorithm, which converges faster. The derivatives of the functions are represented by their Taylor approximations and the initial guess is obtained by considering the local machining geometry.

Finally, the above algorithms are suitable only for ball-nose three-axis machining. An efficient algorithm for searching the iso-scallop cutter paths that extends the algorithm to five-axis machining with a flat-end cutter

Fig. 23 A simple scallop evaluation procedure



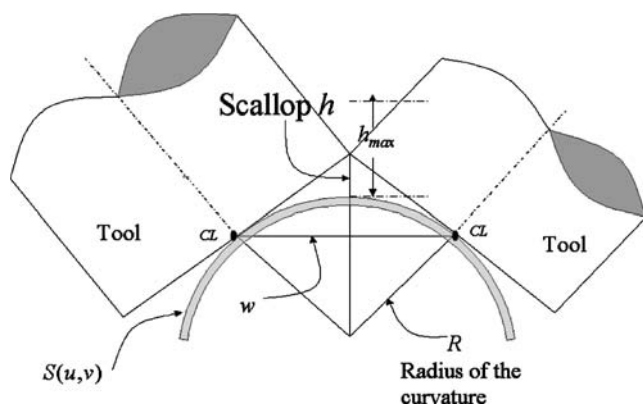


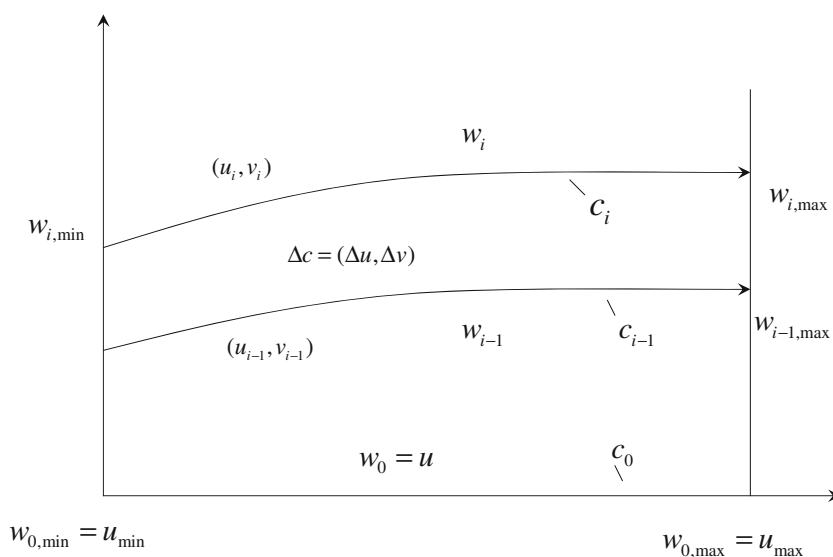
Fig. 24 A simple scallop evaluation for a flat-end cutter positioned along the normal.

was proposed by Lo [127]. The method is suitable for the three-axis case as well. The algorithm starts with an initial curve in the parametric domain and calculates offset curves so that the scallop height remains approximately constant. A tool path in the Cartesian 3D domain corresponds to a specific curve $c(w)=(x(w), y(w))$ in the parametric $u-v$ domain. This technique in parametric domain is illustrated in Fig. 25. The i -th $u-v$ is obtained by an increment of the $i-1$ -th $u-v$ curve. For iso-scallop paths, the increments $\Delta u, \Delta v$ correspond to a fixed scallop height. For a given scallop height, the path interval, of which the distance l can be calculated in the side-step direction b by using.

$$\frac{\partial S}{\partial u} \Delta u + \frac{\partial S}{\partial v} \Delta v = bl. \quad (3.3.5)$$

Note that Eq. 3.3.5 is a vector equation which provides three individual component equations; however, only two axial components need to be evaluated so as to calculate $\Delta u, \Delta v$.

Fig. 25 Iso-scallop tool path generation



The algorithm is designed for flat-end cutters and includes adaptive inclination which maximizes the machining strip and avoids local gouging (see Section 2). A boundary curve, considered as the initial track, is repeatedly offset, propagating into the parametric region. At each point the offset depends on the tool inclination and therefore varies from point to point. Consequently, the next track is a curve which may have an entirely different shape as compared with the initial track. Eventually, a zigzag-like set of curves is constructed. Lo [129] incorporates these ideas into an interpolator (see Section 1) designed for iso-parametric, iso-scallop, and iso-planar machining by ball-nose cutters.

However, there are a number of open problems. In particular, the solution depends on the initial curve and the choice of this curve is not trivial. It is not clear how to proceed if the next curve intersects the boundary of the region or another curve. Furthermore, none of the above tool path generation methods provide optimal cutting conditions at each CC point.

Evaluation of the machining strip versus the inclination and the direction appears to entail very complicated tool path topologies. For each tool position on the surface there exists at least one direction which maximizes the machining strip. Consider the corresponding vector field. For simplicity let us map the vector field onto the parametric space (u,v) so that the field becomes two-dimensional. Clearly, a continuous tool path which visits every point and follows the optimal direction at every point constitutes the optimal tool path which maximizes the machining strip in the global sense. However, such a path can rarely be found in practice due to the complexity of the resulting vector field. An algorithm to find a suboptimal solution of this problem is presented in Chiou and Lee [33]. The idea is similar to Lo [127], but the optimization is considered in more global

sense. The entire surface is discretized using a rectangular grid in the parametric space and then covered by potential machining patches each characterized by the optimal direction producing the maximum machining strip (Fig. 26). The optimal direction vector field can be used to find the an approximate optimal direction in an arbitrary point without employing an expensive direct evaluation of the machining strips similar to Eq. 3.2.1.

Next, the authors introduce an “initial” path which has the largest average machining strip. Next, the entire tool path is constructed by offsetting the initial path and propagating the offsets inside the region. These offset paths are modified if they substantially deviate from the streamlines of the optimal direction vector field (Figs. 26 and 27).

Unfortunately, many surfaces produce a complicated, non-uniform vector field and although the above algorithm allows one to decompose the surface into sub-surfaces, the decomposition is not very well motivated from the optimization viewpoint. In particular, after the first propagation step, the algorithm searches for a new “initial tool path” such that the ratio between the length of the path and the average machining strip is less than a certain threshold. It is not hard to show that such analysis is not always accurate from the viewpoint of global optimization. It may also be sensitive to local variations of the optimization criteria. More importantly, finding the initial tool path is a computationally expensive, NP hard problem.

My et al. [150, 151] and Makhanov [137, 138] introduce a partition of the surface into clusters having vector field streamlines similar to the conventional zigzag or spiral patterns (Fig. 27). The advantage of this approach is that within a cluster the tool follows a nearly optimal path. Clustering optimizes global criteria of the decomposition and makes it possible not only to decompose the surface but also to recognize local similarity to the conventional tool path patterns such as the zigzag and the spiral. An appropriate linking of the clusters can be performed using the ideas introduced in Park and Chung [157].

Furthermore, a few papers explore other “iso” methods such as the iso-distance and iso-curvature methods [77]. In Kim et al. [99, 100] additional tool path segments are appended to the basic tool path in order to achieve constant cutting forces and to avoid chatter vibrations in the entire machining area.

Suppose that the part is partitioned into a grid of cells each of them being a curvilinear triangle or a curvilinear rectangle. The optimization can then be considered as constructing a path which visits each cell, does not have intersections, requires a minimal number of tool retractions, and satisfies some error-related criteria. Using this idea, a pocket machining technique using staircase or window frame patterns was proposed in Persson [160].

Hansen and Arbab [76] developed a similar scan line algorithm for generating NC tool paths for arbitrarily shaped flat-bottom pockets with islands. Flat pocket machining based on grids was suggested by Bao and Yim [18]. High speed pocket machining was analyzed in [166].

Treating tool path generation as a navigation problem on grids leads to exploiting shortest-path optimization and related techniques. Suh and Shin [186] developed a neural network model to obtain the tool path in rough pocket machining as a solution to the traveling salesperson problem. A rigorous mathematical analysis of such strategies is given in Arkin et al. [15]. The problem is formulated as follows: given a region in the plane, and given the shape of a cutter, find a shortest tour/path for the cutter such that every point within the region is covered by the cutter at some position along the tour (tool path). Additionally, the cutter could be constrained to stay within a certain region.

Narayanaswami and Choi [153] present an extension of the grid-based navigation approach to the 3D case. The approach can be extended to the five-axis case.

Space-filling curves (SFC) having been applied in computer graphics, image processing, and information systems, can be also seen as a suitable navigation pattern

Fig. 26 Optimal direction vector field

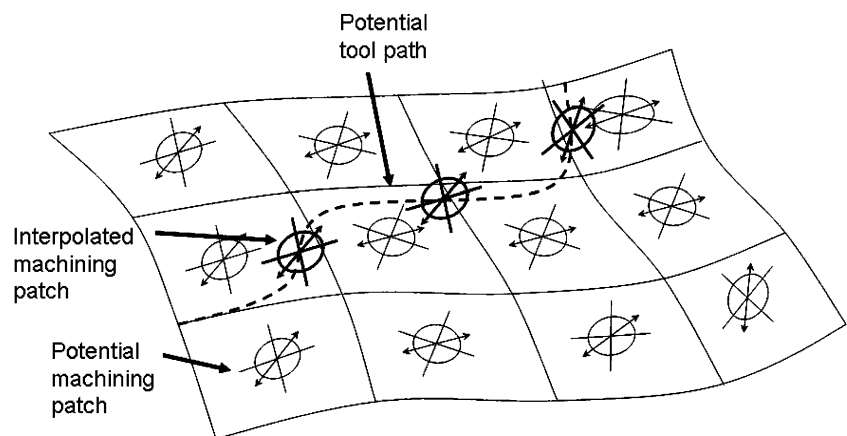
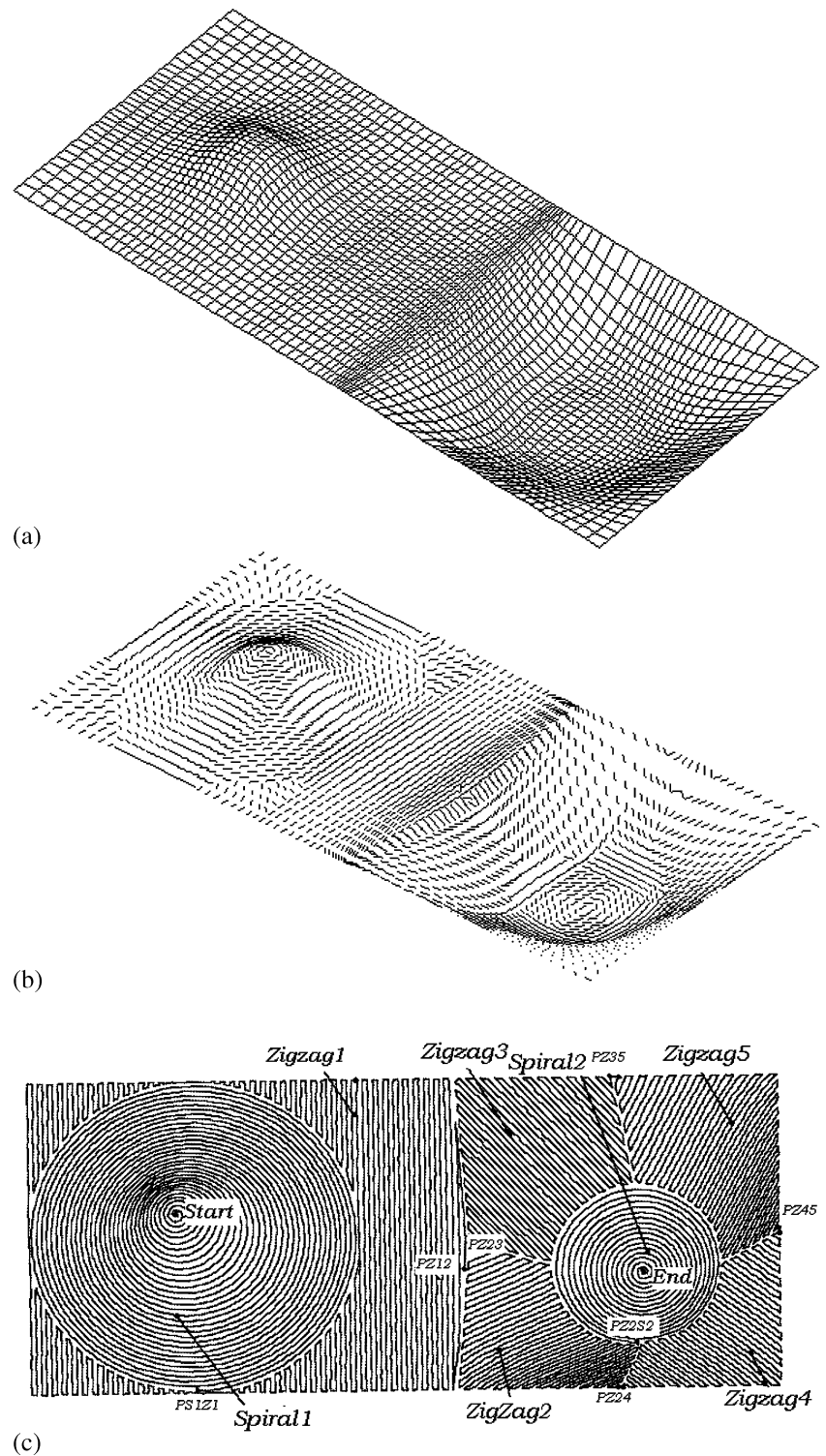


Fig. 27 Vector field clustering approach: **a** the part surface, **b** the optimal direction vector field, **c** the resulting clusters with the spiral and zigzag type of the motion

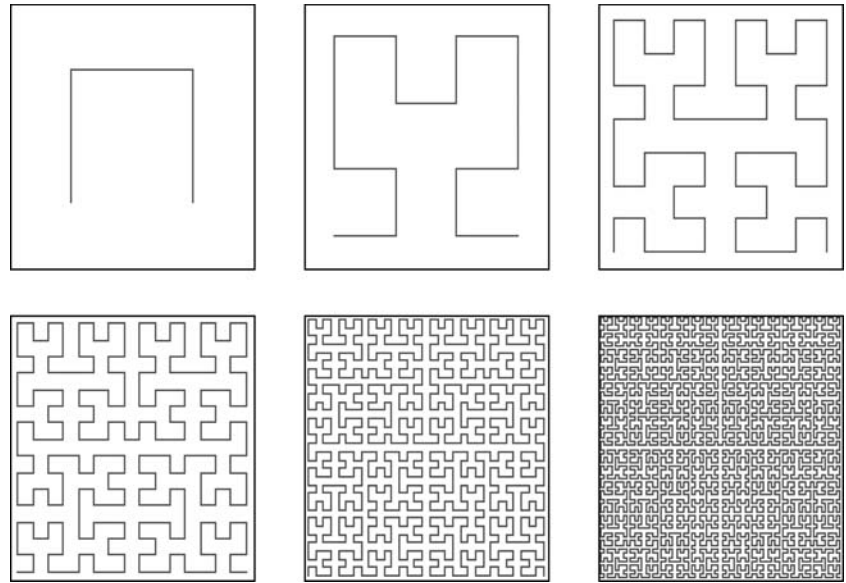


for generation of machining paths. The first application of SFC to NC tool path generation was reported in Griffiths [75] and Cox et al. [38].

Griffiths proposed the use of Hilbert's curve as a tool path (Fig. 28), while Cox et al. used various forms of space-filling curves such as Moore's curve.

Fractal-based techniques were suggested by Chen et al. [30].

The Hilbert curve is in particular appealing in tool path planning as its local refinement property can be used to adaptively increase the density of the path only where necessary. However, each local refinement of the tool path

Fig. 28 Six iterations of the Hilbert space-filling curve

based on the Hilbert's curve increases the tool path density in the refined region by a factor of 2 resulting in lower machining efficiency due to the increased total path length. Besides, the Hilbert's curve has an undesirable property that it leads to a path, where the tool is frequently changing directions which slows down the machining process and produces large kinematics errors. This is why neither SFCs nor fractals have ever been very popular in the five-axis machining community due to a large number of sharp turns produced by conventional SFC's.

The concept of adaptive space-filling curves for tool path planning for five-axis NC machining was proposed in Anotaipaiboon and Makhanov [14]. Space-filling curves, adapted to the local optimal cutting direction, produce shorter tool paths. Besides, a tool path correction stage makes it possible to eliminate the effect of the sharp angular turns that characterize the standard SFC patterns (Fig. 29).

Note that even if the SFC provides the shortest path in five-axis machining, that path does not necessarily provide the shortest time. A slow angular feed rate may lead to time longer than that provided by a zigzag grid. This follows from a simple estimate of the required machining time based on the slowest (for this step) axis. Given the spatial and angular increments $\Delta x_M, \Delta y_M, \Delta z_M, \Delta a, \Delta b$ and the feed rate F , the machining time t_M is calculated by $t_M = \max\{t_0, t_x, t_y, t_z, t_A, t_B\}$, where $t_0 = \frac{\Delta L}{F}$, where $\Delta L = \sqrt{(\Delta x_M)^2 + (\Delta y_M)^2 + (\Delta z_M)^2}$, and where $t_x = \frac{\Delta x_M}{v_{x,\max}}, t_y = \frac{\Delta y_M}{v_{y,\max}}, t_z = \frac{\Delta z_M}{v_{z,\max}}, t_A = \frac{\Delta a}{v_{A,\max}}, t_B = \frac{\Delta b}{v_{B,\max}}$, where $v_{\max}$ denotes the maximum speed in the corresponding axis.

However, for fast angular feed rates and short, slow linear motions, SFCs are a very good choice.

Finally, the direction of the adaptive SFC at a given point can be selected using a variety of criteria such as the

speed of the tool or the acceleration of the linear and/or rotation axes. In this case the definition of the distance between two cutter location points must be modified accordingly. Of course, having angular accelerations constrained could turn the SFC into a simple zigzag. In some cases it is possible to perform optimization directly with regard to the machining time; however, for general machining tools, it is still an open problem.

Furthermore, the entire tool path can be considered in the framework of grid generation technologies. The concept was first introduced in Makhanov [133] and developed in Bohez et al. [23] and Makhanov et al. [134]. Grid generation techniques are surprisingly well adapted to tool path optimizations. As a matter of fact, the concept of grid refinement contains almost all the main ingredients for tool path planning, such as grid adaptation to the regions of large milling errors, the possibility of easily constructing curvilinear versions of the conventional zigzag and spiral patterns, and adaptation to constraints related to the tool diameter and the scallop height. Moreover, in contrast to the standard techniques characterized by a local error estimate, grid generation deals with a global spatial error and consequently adapts all the CL points simultaneously.

These ideas were developed further in Makhanov and Ivanenko [135] and Makhanov [137] specifically for five-axis machining. Bieterman and Sandstrom [20] suggested a similar approach independently. Finally, Sun et al. [187] presented a spiral version of the grid generation algorithm applied to tool path generation.

Consider a set of cutter location points $\{u_{ij}, v_{ij}\}$ arranged as a curvilinear grid. Mathematically, it means that (u_{ij}, v_{ij}) is a discrete analogy of a mapping from the computational region $\{0 \leq \xi \leq 1, 0 \leq \eta \leq 1\}$ onto a parametric

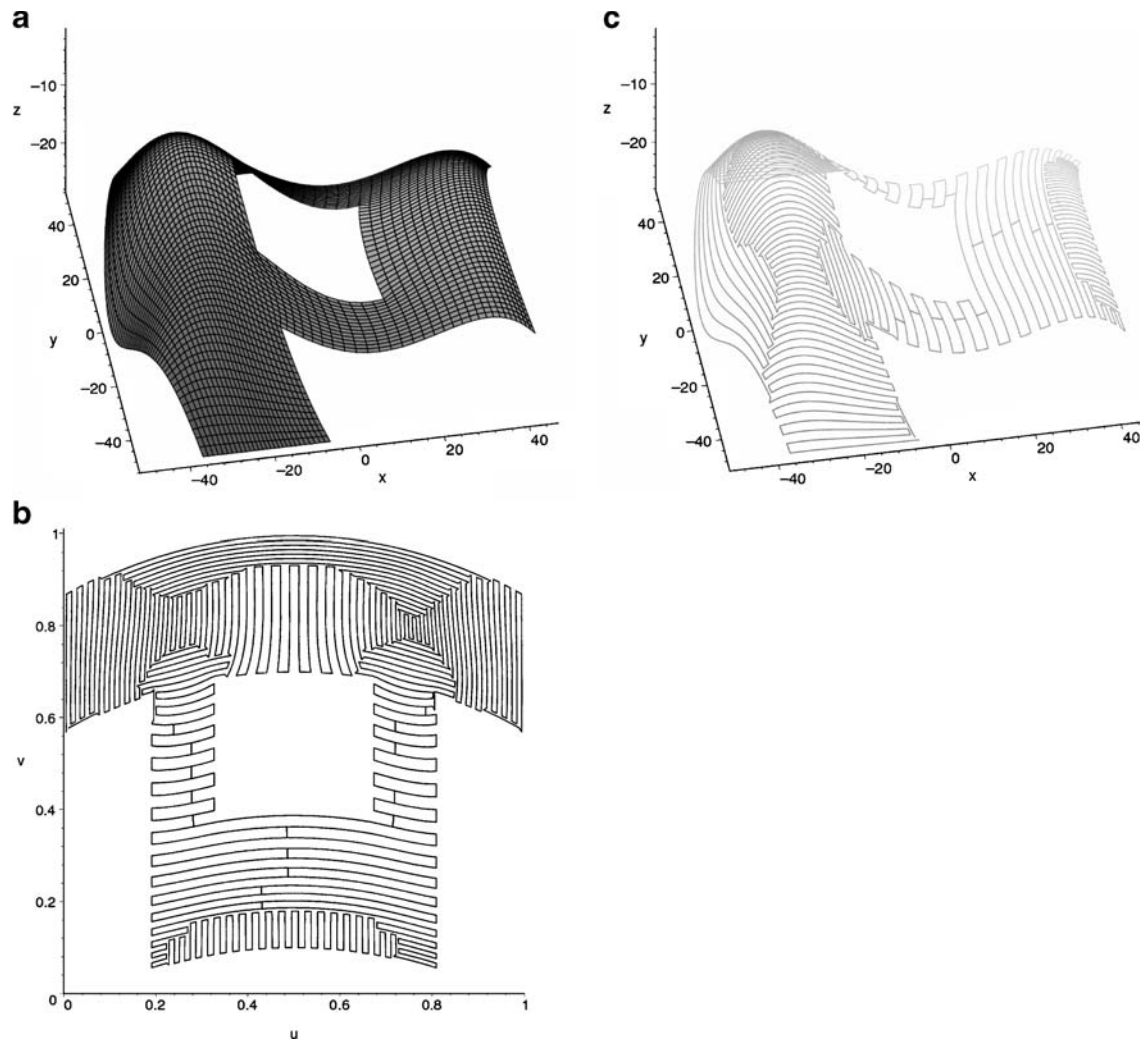


Fig. 29 Adaptive space-filling curve **a** part surface with an island, **b** space-filling curve in the parametric space, **c** space-filling curve on the surface

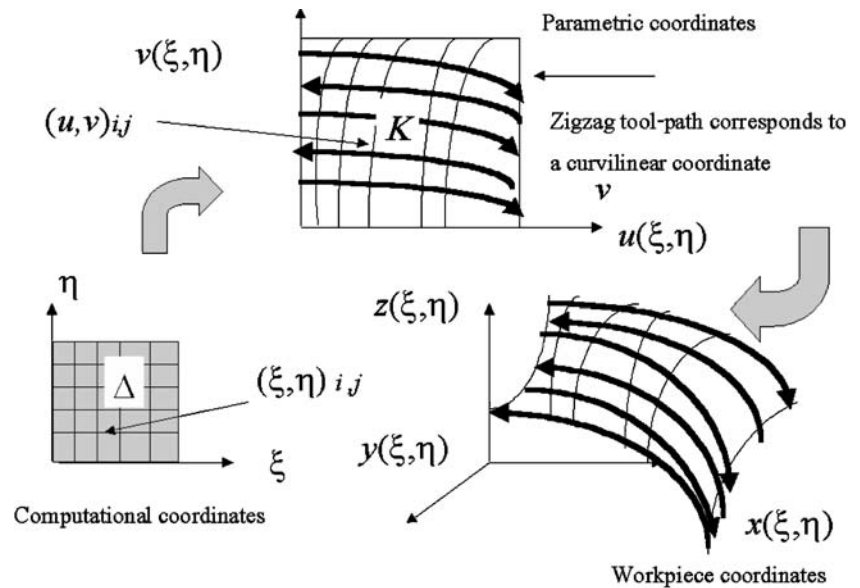
region defined in the parametric coordinates u, v . In other words, there exists a pair of functions $\{u(\xi, \eta), v(\xi, \eta)\}$ such that the rectangular grid $\{i, j\}$ being fed to $\{u(\xi, \eta), v(\xi, \eta)\}$

becomes $\{u_{i,j}, v_{i,j}\}$, see Fig. 30. The required (unknown) grid is a discretized solution the following minimization problem:

$$\min_{u,v} \int_0^1 \int_0^1 \frac{(u_\xi^2 + u_\eta^2)(1 + f_u^2) + (v_\xi^2 + v_\eta^2)(1 + f_v^2) + 2f_u f_v (u_\xi v_\eta + u_\eta v_\xi)}{(u_\xi v_\eta - u_\eta v_\xi) \sqrt{1 + f_u^2 + f_v^2}} d\xi d\eta, \quad (3.3.6)$$

where subscripts u, v, ξ, η denote partial derivatives and f is the control function. The harmonic functional I is a generalization the Winslow functional [22, 213] to the case of the grids lying on the surface $f(u, v)$. The harmonic functional is derived from the theory of harmonic maps [91, 92]. It has been proven that the functional minimizes an “energy of mapping” and produces a grid adapted to

the regions of large gradients of f . Note that if $f_u = f_v \equiv 0$, then the harmonic functional becomes the Winslow functional; however, it is important that I adapts the grid to the gradients of f rather than to f itself as in the previous version [133]. It is known that minimization of Eq. 3.3.6 could be computationally expensive as compared with minimization of the Winslow functional [28].

Fig. 30 Tool path in the framework of grid generation

However, it has many points in its favor. In particular, it is possible to construct a computational procedure which, under certain conditions, converges to a non-degenerate grid, that is, the grid without twisted or non-convex cells. The constraint minimization of Eq. 3.3.6 can be performed by using efficient penalty type techniques similar to those presented in [133]. Finally, Eq. 3.3.6 is more reliable and converges for sharp variations of the input data whereas the Winslow functional often produces degenerated grids.

However, the above techniques have a number of drawbacks. In particular, they may converge slowly for complicated constraints. Besides, the approach requires equal numbers of cutter contact points on each track of the tool. Therefore, if the kinematics error changes sharply from track to track, the method may require an excessive number of points.

A new modification of grid refinement which better fits within the framework of tool path optimization and is designed specifically for SFC generation is introduced by Anotaipaiboon and Makhanov [139]. The method does not require equal number of points on each track since the grid generation does not try to minimize or to reduce the kinematics error along the tracks as in Makhanov [133]. At the grid generation stage, the kinematics error is ignored and the grid is generated with regard to scallop height constraints. In fact, the proposed techniques present a combination of the iso-scallop method, the boundary offset method, and the grid navigation approach.

The resulting grid can be converted into a curvilinear zigzag path or into a space-filling curve. After that, the cutter location points along the resulting curve are distributed using a standard method such as bisection.

Additionally, the algorithm evaluates the number of required grid lines. As opposed to the preceding approach, where the weight function represents either the kinematics error or an estimate of the kinematics error (such as the surface curvature or the rotation angles), the proposed algorithm iteratively constructs an adaptive control function designed to represent the scallop height constraints. This important modification makes it possible to consider an arbitrary number of points along the tool tracks.

The kinematics error is reduced by means of inserting additional points along the resulting curvilinear coordinates. In other words, the approach replaces the scallop constraints by a weight function and then treats the kinematics error independently. Additionally, instead of the Winslow functional proposed in [133] the new optimization is based on a functional derived from the theory of harmonic maps [28]. The functional not only provides smoothness and adaptivity but under certain conditions also guarantees numerical convergence.

Finally, Anotaipaiboon and Makhanov [139] merge the grid generation technique with SFC techniques (Fig. 31). In this case, the grid is not converted to a tool path directly. Instead, it becomes the basic grid required for SFC generation. With this modification, the SFC tool path can be constructed for surfaces with complex irregular boundaries, cutoffs, pockets, islands, etc. Besides, the adaptive grid is an efficient way to handle complex spatial variability because it allows the SFC to be created on a grid having small cells only where necessary. The combination of the two techniques is superior to the cases where the two methods are applied independently.

Finally, although elegant and intellectually appealing, grid generation methods are computationally costly. In

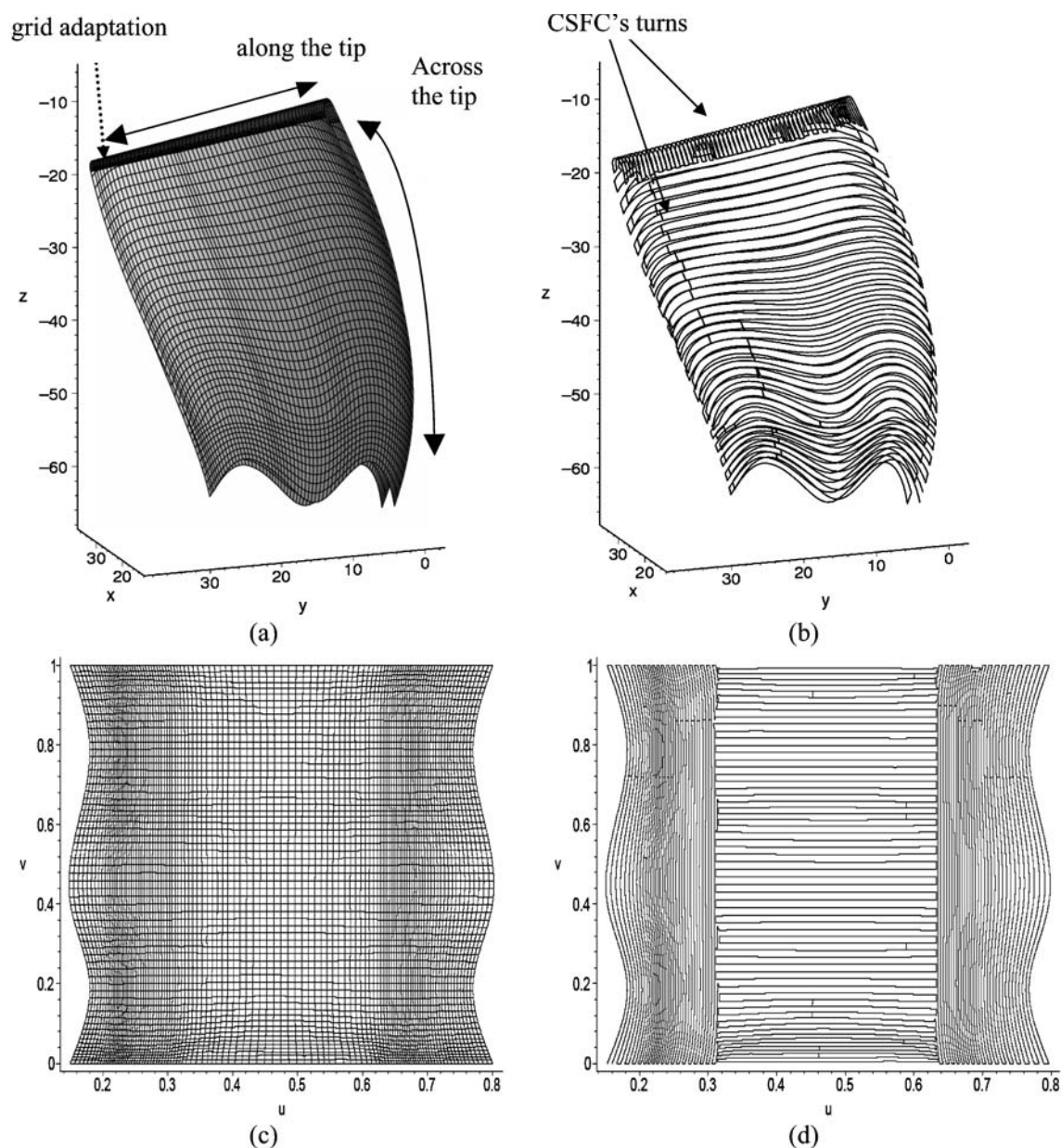


Fig. 31 Curvilinear SFC for the blade of an industrial impeller **a** curvilinear grid on the surface, **b** curvilinear SFC on the surface, **c** curvilinear grid in the parametric domain, **d** curvilinear SFC in the parametric domain

many cases they require many hours of computing time. Sometimes the computation time is larger than the cutting time. Therefore, the use of such methods is well justified for regional milling for complex-shaped surfaces with sharp variations in curvature.

4 Conclusions

A survey on tool path interpolators, methods for tool posture, gouging avoidance, and adaptable geometric patterns has been presented. The most important numerical

optimization procedures for constructing efficient adaptable tool paths for five-axis machining are outlined and analyzed. The survey includes 225 references dated from 1978 to 2009.

The survey shows that tool path generation for five-axis milling machines is a multi-criteria problem which includes minimization of the machining time, the length of the tool path, kinematics error, scallops, undercuts/overcuts due to the curvature interference, and rear gouging. The optimization is subjected to a number of constraints such as the machine axis limits, global gouging, acceleration, and jerk limits.

The adaptable patterns such as the navigated paths, space-filling curves, cluster-based or region-based methods, and curvilinear grids constitute a promising trend in modern tool path generation. However, as of yet, there is not a single approach applicable to an arbitrary part. Therefore, intelligent adaptive tool path generation of the future should include a variety of the quality criteria combined with technological parameters and the machine kinematics. Besides, the tool path generator must be able to select an appropriate geometric pattern based on the machining features and the quality requirements. There is no doubt that the adaptable geometric patterns will become such industry standards as NURBS [2] and solid modeling.

Acknowledgement I acknowledge a sponsorship of Thailand Research Fund. I also wish to thank Matthew N. Dailey for technical edition, attention to details and many useful comments.

Appendix. Kinematics equations of the five-axis machines

In order to classify the machine kinematics, we introduce the following coordinate systems: the workpiece coordinate system O_1 , a coordinate system of the first rotary part O_2 , a coordinate system of the second rotary part O_3 , and a coordinate system of the spindle O_4 . We shall call the first rotary axis the *A*-axis and the second rotary axis the *B*-axis.

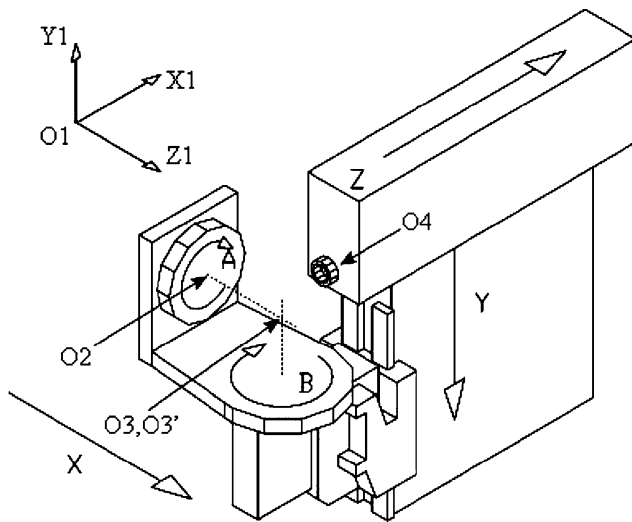


Fig. 32 The 1-0 machine and the reference coordinate systems

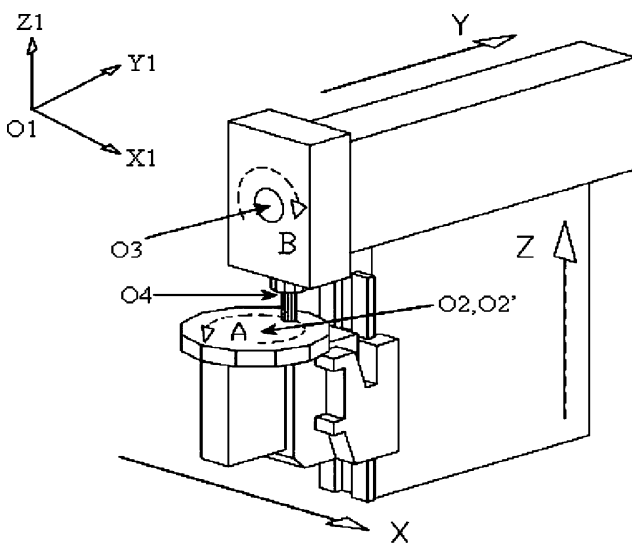
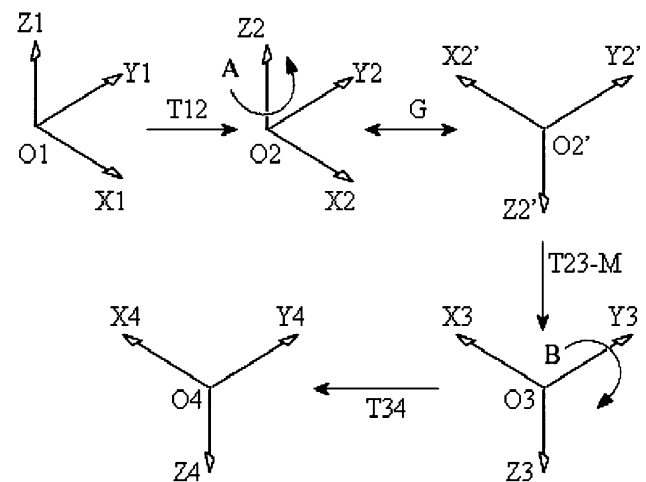
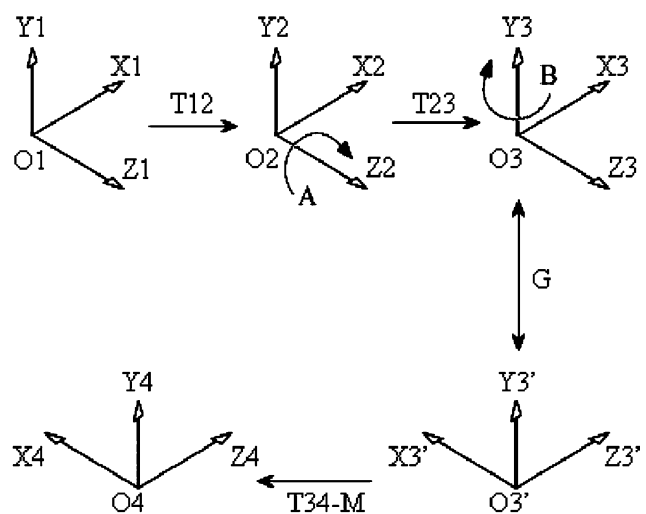


Fig. 33 The 1-1 machine and the reference coordinate systems



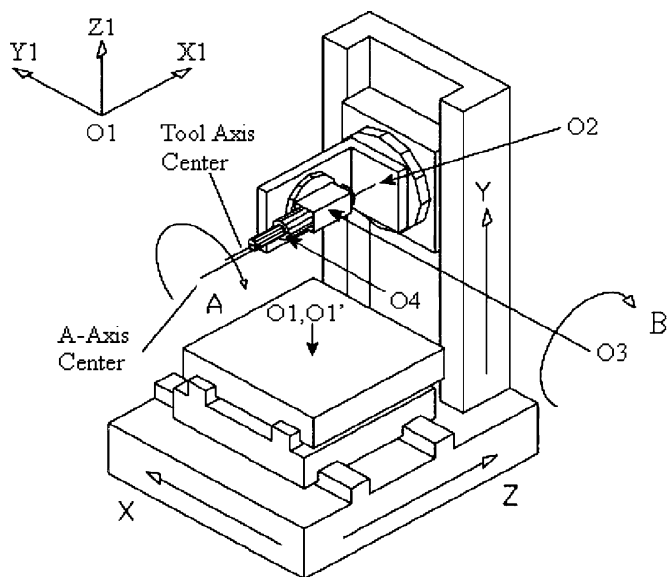


Fig. 34 The 0-2 machine and the reference coordinate systems

where T_{23} is the coordinate of O_2 in O_3 .

Consider three important types of the machine kinematics categorized by the positions of the rotational joints in the kinematics chain.

The 2-0 machine. Two rotary axes on the table, Fig. 32. In this case the kinematics is the 1-1 machine. One rotary axis on the table and one on the tool, Fig. 33.

The 0-2 machine. Two rotary axes on the tool, Fig. 34. Let us deduce the corresponding kinematics equations.

The 2-0 machine

The kinematics equations are obtained by the following coordinate transformations (see Fig. 32).

Step 1: Coordinate translation $O_1 \rightarrow O_2$

$$P_2 = W + T_{12},$$

where T_{12} is the coordinate of O_1 in O_2 .

Step 2: Rotation around A -axis in O_2 by a

$$P_{2A} = A[a]P_2 = A[a](W + T_{12}),$$

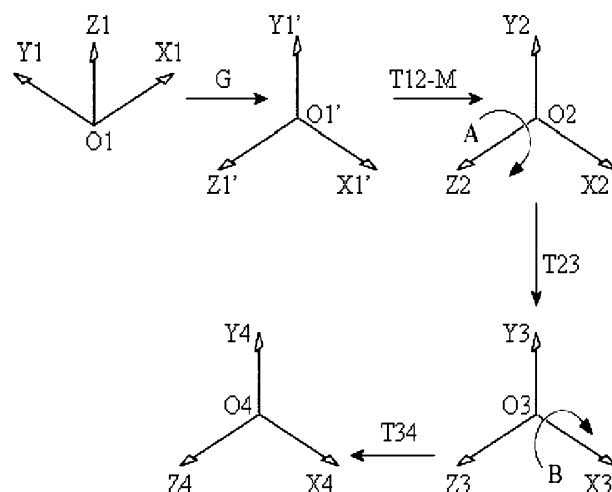
where $A[a] = \begin{bmatrix} \cos a & \sin a & 0 \\ -\sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the rotation matrix around the A -axis.

Step 3: Coordinate translation $O_2 \rightarrow O_3$

$$P_3 = P_{2A} + T_{23} = A[a](W + T_{12}) + T_{23},$$

Step 4: Rotation around B -axis in O_3 by b

$$P_{3B} = B[b]P_3 = B[b](A[a](W + T_{12}) + T_{23}),$$



where $B[b] = \begin{bmatrix} \cos b & 0 & -\sin b \\ 0 & 1 & 0 \\ \sin b & 0 & \cos b \end{bmatrix}$ is the rotation matrix around B -axis.

Step 5: Coordinate rotation (axis alignment) $O_3 \rightarrow O'_3$

$$P'_3 = GP_{3B} = GB[b](A[a](W + T_{12}) + T_{23}),$$

where $G = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$ is the axis alignment matrix.

Step 6: Coordinate translation $O'_3 \rightarrow O_4$ with machine slide translation M

$$P_4 = P'_3 + T_{34} - M = GB[b](A[a](W + T_{12}) + T_{23}) + T_{34} - M,$$

where T_{34} is the coordinate of O'_3 in O_4 with respect to the machine zero point $M = (0, 0, 0)$.

Equating P_4 and T_4 yields

$$T_4 = P_4 = GB[b](A[a](W + T_{12}) + T_{23}) + T_{34} - M.$$

After rearrangement,

$$M = GB[b](A[a](W + T_{12}) + T_{23}) + T_{34} - T_4, \\ W = A^{-1}[a](B^{-1}[b]G^{-1}(M - T_{34} + T_4) - T_{23}) - T_{12},$$

where $T_4 = (0, 0, -L)$ is the coordinate of the tool tip in O_4 and L is the tool length.

Let T'_1 be the coordinate of $T'_4 = (0, 0, -L + 1)$ in O_1 and is given by

$$T'_1 = A^{-1}[a](B^{-1}[b]G^{-1}(M - T_{34} + T'_4) - T_{23}) - T_{12}.$$

The relationship between the tool orientation $I = (I_x, I_y, I_z)$ and the rotation angles, a and b , are then given by

$$I = T'_1 - W = A^{-1}[a]B^{-1}[b]G^{-1}(T'_4 - T_4) \\ = \begin{bmatrix} \cos(a) \cos(b) \\ \sin(a) \cos(b) \\ -\sin(b) \end{bmatrix}.$$

Inverting with regard to a and b yields

$$a = \begin{cases} \tan^{-1}\left(\frac{I_y}{I_x}\right) & \text{if } I_x > 0 \text{ and } I_y > 0, \\ \tan^{-1}\left(\frac{I_y}{I_x}\right) + \pi & \text{if } I_x < 0, \\ \tan^{-1}\left(\frac{I_y}{I_x}\right) + 2\pi & \text{otherwise,} \end{cases} \\ b = -\sin^{-1}I_z.$$

The 1-1 machine

The above procedure applies to an arbitrary machine configuration. For the 1-1 machine shown in Fig. 33, the coordinate of the tool tip in O_4 is given by

$$T_4 = B[b](GA[a](W + T_{12}) + T_{23} - M) + T_{34}.$$

Clearly,

$$M = GA[a](W + T_{12}) + T_{23} + B^{-1}[b](T_{34} - T_4), \\ W = A^{-1}[a]G^{-1}(M - T_{23} - B^{-1}[b](T_{34} - T_4)) - T_{12}, \\ I = T'_1 - W = A^{-1}[a]G^{-1}B^{-1}[b](T'_4 - T_4) = \begin{bmatrix} \cos(a) \sin(b) \\ -\sin(a) \sin(b) \\ \cos(b) \end{bmatrix}, \\ a = \begin{cases} -\tan^{-1}\left(\frac{I_y}{I_x}\right) & \text{if } I_x > 0 \text{ and } I_y < 0, \\ -\tan^{-1}\left(\frac{I_y}{I_x}\right) + \pi & \text{if } I_x < 0, \\ -\tan^{-1}\left(\frac{I_y}{I_x}\right) + 2\pi & \text{otherwise,} \end{cases} \\ b = \cos^{-1}I_z,$$

where

$$A[a] = \begin{bmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}, B[b] = \begin{bmatrix} \cos b & 0 & -\sin b \\ 0 & 1 & 0 \\ \sin b & 0 & \cos b \end{bmatrix}, \\ G = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \\ T_4 = (0, 0, L).$$

The 0-2 machine

Applying the same procedure to the for 0-2 machine (Fig. 34) yields

$$T_4 = B[b](A[a](GW + T_{12} - M) + T_{23}) + T_{34}.$$

Clearly,

$$M = GW + T_{12} + A^{-1}[a](T_{23} + B^{-1}[b](T_{34} - T_4)), \\ W = G^{-1}(M - T_{12} - A^{-1}[a](T_{23} + B^{-1}[b](T_{34} - T_4))), \\ I = T'_1 - W = G^{-1}A^{-1}[a]B^{-1}[b](T'_4 - T_4) = \begin{bmatrix} \cos(b) \\ \sin(a) \sin(b) \\ -\cos(a) \sin(b) \end{bmatrix}, \\ a = \begin{cases} -\tan^{-1}\left(\frac{I_y}{I_z}\right) & \text{if } I_y < 0 \text{ and } I_z > 0, \\ -\tan^{-1}\left(\frac{I_y}{I_z}\right) + \pi & \text{if } I_z < 0, \\ -\tan^{-1}\left(\frac{I_y}{I_z}\right) + 2\pi & \text{otherwise,} \end{cases} \\ b = -\cos^{-1}I_x.$$

where

$$A[a] = \begin{bmatrix} \cos a & -\sin a & 0 \\ \sin a & \cos a & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\ B[b] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos b & -\sin b \\ 0 & \sin b & \cos b \end{bmatrix}, \\ G = \begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{bmatrix}, \\ T_4 = (0, 0, L).$$

References

1. The NURBS++ package, <http://libnurbs.sourceforge.net/index.shtml>.
2. NURBS Toolbox, <http://www.aria.uklinux.net/nurbs.php3>.
3. NIST/IGES, <http://www.nist.gov/iges/>.
4. SLC file format, <http://www-rp.me.vt.edu/bohn/rp/SLC.html>. Info Page: CAD/CAM, <http://www.cs.cmu.edu/People/unsal/research/rapid/cadcam.html>.
5. CAD Importer File Formats, <http://www.actify.com/v2/products/Importers/formats.htm>.
6. Makino. GE FANUC, NURBS Interpolation, http://www.makino.com/about/article/2-1-2008/GE_Fanuc
7. Makino. Interpolating Curves http://www.makino.com/about/article/2-1-2008/Interpolating_Curves
8. Aigner M, Šir Z, Jüttler B (2007) Evolution-based least-squares fitting using Pythagorean hodograph spline curves. *Comput Aided Geom Des* 24:310–322
9. Alt H, Godau M (1995) Computing the Fréchet distance between two polygonal curves. *Int. J. Comput. Geom. Appl.* 5:75–91
10. Alt H, Knauer C, Wenk C (2001) Matching polygonal curves with respect to the Fréchet distance, *Lecture Notes in Computer*

- Science, 18th Annual Symposium on Theoretical Aspects of Computer Science Dresden, Germany, pp. 63–74, February 15–17
11. Altintas Y, Merdol SD (2007) Virtual high performance milling. *CIRP Ann-Manufacturing Technology* 56(1):81–84
12. Altintas Y, Erkorkmaz K (2003) Feedrate optimization for spline interpolation in high speed machine tools. *CIRP ann* 52(1):297–302
13. Anderson RO (1978) Detecting and eliminating collisions in NC machining. *Computer-Aided Design* 10(4):231–237
14. Anotaipaboon W, Makhanov SS (2005) Tool path generation for five-axis NC machining using adaptive space-filling curves. *Int. J. Prod. Res.* 43(8):1643–1665
15. Arkin EM, Fekete SP, Mitchell JSB (2000) Approximation algorithms for lawn mowing and milling. *Computational Geometry: Theory and Applications* 17(1–2):25–50
16. Bahr B, Xiao X, Krishnan K (2001) A real time scheme of cubic parametric curve interpolations for CNC systems. *Comput. Ind.* 45:309–317
17. Balasubramaniam M, Sarma SE, Marciniak K (2003) Collision free finishing tool paths from visibility data. *Comput Aided Des* 35(4):359–374
18. Bao HP, Yim H (1992) Tool path determination for end milling of non-convex shaped polygons. *NAMRI Transactions*, 151–158
19. Belogay E, Cabrelli C, Molter U, Shonkwiler R (1997) Calculating the Hausdorff distance between curves. *Inf. Process. Lett.* 64:17–22
20. Bieterman MB, Sandstrom DR (2003) A curvilinear tool-path method for pocket machining. *J. Mater. Process. Technol.* 125(4):709–715
21. Blasquez I, Poiraudau J-F (2004) Undo facilities for the extended z-buffer in NC machining simulation. *Comput. Ind.* 53(2):193–204
22. Brackbill JU, Saltzman JS (1982) Adaptive zoning for singular problems in two dimensions. *J Comput Phys* 46:342–368
23. Bohez ELJ, Makhanov SS, Sonthipermoon K (2000) Adaptive nonlinear tool path optimization for 5-axis machining. *Int. J. Prod. Res.* 38(17):4329–4343
24. Kim BH, Choi KB (2002) Machining efficiency comparison direction-parallel path with contour parallel path. *Comput. Aided Des* 34:89–95
25. Bohez ELJ (2002) Compensating for systematic errors in 5-axis NC machining. *Comput. Aided Des* 34(5):505–520
26. Bohez ELJ, Minh NTH, Kiatsrithanakorn B, Natasukon P, Ruei-Yun H, Son LT (2003) The stencil buffer sweep plane algorithm for 5-axis CNC tool path verification. *Comput Aided Des* 35(12):1129–1142
27. Butler J, Haack B, Tomizuka M (1988) Reference input generation for high speed coordinated motion of a two axis system. In *Symposium on Robotics, Winter Annual Meeting of the American Society of Mechanical Engineers*, pp. 457–470
28. Charakhch'yan AA, Ivanenko SA (1997) A variational form of the Winslow grid generator. *J Comput Phys* 136(2):385–398
29. Chen L, Woo T (1992) Computational geometry on the sphere with application to automated machining. *J. Mech. Des.* 114:288–295
30. Chen C-CA, Juang Y-S, Lin W-Z (2002) Generation of fractal tool paths for irregular shapes of surface finishing areas. *J. Mater. Process. Technol.* 127(2):146–150
31. Chen YD, Ni J, Wu SM (1993) Real-time CNC tool path generation for machining IGES surfaces. *ASME Journal of Engineering for Industry* 115(4):480–486
32. Cheng MY, Tsai MC, Kuo JC (2002) Real-time NURBS command generators for CNC servo controllers. *Int. J. Mach. Tools Manuf.* 42:801–803
33. Chiou C-J, Lee Y-S (2002) A machining potential field approach to tool path generation for multi-axis sculptured surface machining. *Comput. Aided Des.* 34(5):357–371
34. Cho I, Lee K, Kim J (1997) Generation of collision-free cutter location data in five-axis milling using the potential energy method. *Int J Adv Manuf Tech* 13(8):523–529
35. Choi BK, Jerard RB (1998) *Computer aided machining—the z-map way: sculptured surface machining—theory and applications*. Kluwer Academic Publishers, Dordrecht
36. Choi Y-K, Banerjee A (2007) Tool path generation and tolerance analysis for free-form surfaces. *Int. J. Mach. Tools Manuf.* 47(3–4):689–696
37. Constantinescu D, Croft EA (2000) Smooth and time optimal trajectory planning for industrial manipulators along specified paths. *J. Robot. Syst.* 17:233–249
38. Cox JJ, Takezaki Y, Ferguson HRP, Kohkonen KE, Mulkay EL (1994) Space-filling curves in tool-path applications. *Comput. Aided Des.* 26(3):215–224
39. De Boor C (2001) *A practical guide to splines*. Springer, New York
40. Dong J, Stori JA (2003) Optimal feed-rate scheduling for high speed contouring. *ASME International Mechanical Engineering Congress*. Washington, D.C. pp. 497–513, November 15–21.
41. Dong J, Stori JA (2006a) A generalized time-optimal bi-directional scan algorithm for constrained feedrate optimization. *ASME Journal of Dynamic Systems, Measurement, and Control* 128:379–390
42. Dong J, Stori JA (2006b) A generalized time-optimal bidirectional scan algorithm for constrained feedrate optimization. *Journal of Dynamic Systems, Measurement and Control, Transactions of the ASME* 128(2):379–390
43. Dong J, Ferreira PM, Stori JA (2007) Feed-rate optimization with jerk constraints for generating minimum-time trajectories. *Int. J. Mach. Tools Manuf.* 47:1941–1955
44. Timar SD, Farouki RT, Smith TS, Boyadjieff CL (2005) Algorithms for time-optimal control of CNC machines along curved tool paths. *Robot. Comput.-Integr. Manuf* 21:37–53
45. Dragomatz D, Mann S (1997) A classified bibliography of literature on NC milling path generation. *Comput. Aided Des.* 29(3):239–247
46. DoCarmo MP (1976) *Differential geometry of curves and surfaces*. Prentice-Hall, Englewood Cliffs
47. Elber G, Cohen E (1990) Hidden curve removal for free form surfaces. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pp. 95–104
48. Elber G, Cohen E (1995) Arbitrarily precise computation of gauss maps and visibility sets for freeform surfaces. In *SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications*, pages 271–279
49. Elber G (1995) Freeform surface region optimization for 3-axis and 5-axis milling. *Comput. Aided Des* 27(6):465–70
50. Elber G, Zussman E (1998) Cone visibility decomposition of freeform surface. *Comput. Aided Des.* 30(4):315–320
51. Elber G, Cohen E (1999) A unified approach to verification in 5-axis freeform milling environments. *Comput. Aided Des.* 31(13):795–804
52. Erkorkmaz K, Altintas Y (2001) High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. *Int. J. Mach. Tools Manuf.* 41(9):1323–1345
53. Erkorkmaz K, Heng M (2008) A heuristic feedrate optimization strategy for NURBS toolpaths. *CIRP Annals-Manufacturing Technology* 57:407–410
54. Farin GE (1999) *NURBS: from projective geometry to practical use*. A. K. Peters, Natick
55. Farouki RT (1994) The conformal map $z \rightarrow z^2$ of the hodograph plane. *Comput. Aided Geom. Des.* 11:363–390
56. Farouki RT, Sakkalis T (1994) Pythagorean-hodograph space curves. *Adv. Comput. Math.* 2(1):41–66

57. Farouki RT, Neff CA (1995) Hermite interpolation by Pythagorean hodograph quintics. *Math Comput* 64(212):1589–1609
58. Farouki R, Saitou K, Tsai Y-F (1998) Least-square tool path approximation with Pythagorean-hodograph curves for high-speed CNC machining. In Cripps R (ed) *Proceedings of the IMA Mathematics of Surfaces VIII Conference*, Information Geometers Press, p. 245–264.
59. Farouki RT, Manjunathaiah J, Jee S (1998) Design of rational cam profiles with Pythagorean-hodograph curves. *Mech Mach Theory* 33(6):669–682
60. Farouki RT, Manjunathaiah J, Yang GF (1999) G codes for the specification of Pythagorean-hodograph tool paths and associated feedrate functions on open-architecture CNC machines. *Int. J. Mach. Tools Manuf.* 39:123–142
61. Farouki RT, Tsai Y-F, Wilson CS (2000) Physical constraints on feedrates and feed accelerations along curved tool paths. *Comput Aided Geom Des* 17(4):337–359
62. Farouki RT, Tsai Y-F, Yuan G-F (1999) Contour machining of free-form surfaces with real-time PH curve CNC interpolators. *Comput Aided Geom Des* 16(1):61–76
63. Farouki RT, Sakkalis T, Vaserstein L (2009) Non-existence of rational arc-length parametrizations for curves in R^n . *J. Comput. Appl. Math* 228(1):494–497
64. Feng H-Y, Li H (2002) Constant scallop height tool path generation for three axis sculptured machining. *Comput. Aided Des* 34:647–654
65. Feng H-Y, Teng Z (2005) Iso-planar piecewise linear NC tool path generation from discrete measured data points. *Comput. Aided Des.* 37:55–64
66. Fleisig RV, Spence AD (2001) A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining. *Comput. Aided Des.* 33:1–15
67. Flutter A, Todd J (2001) A machining strategy for tool making. *Comput. Aided Des.* 33(13):1009–1022
68. Gani EA, Kruth JP, Vanherck P, Lauwers B (1997) A geometrical model of the cut in five-axis milling accounting for the influence of tool orientation. *Int J Adv Manuf Tech* 13(10):677–684
69. Garcia-Alonso A, Serrano N, Flaquer J (1994) Solving the collision detection problem. *IEEE Comput. Graph. Appl.* 14(3):36–43
70. Gian R, Lin T, Lin AC (2003) Planning of tool orientation for five-axis cavity machining. *Int J Adv Manuf Tech* 22(1–2):150–160
71. Goldstein BL, Kemmerer SJ, Parks CH (1998) A brief history of early product data exchange standards-NISTIR 6221
72. Gray PJ, Bedi S, Ismail F (2003) Rolling ball method for 5-axis surface machining. *Comput. Aided Des.* 35(4):347–357
73. Gray PJ, Bedi S, Ismail F (2005) Arc-intersect method for 5-axis tool positioning. *Comput. Aided Des.* 37(7):663–674
74. Gray PJ, Ismail F, Bedi S (2007) Arc-intersect method for 311/22-axis tool paths on a 5-axis machine. *Int. J. Mach. Tools Manuf.* 47(1):182–190
75. Griffiths JG (1994) Tool path based on Hilbert's curve. *Comput. Aided Des.* 26(11):839–844
76. Hansen A, Arbab F (1992) An algorithm for generating NC tool paths for arbitrarily shaped pockets with islands. *ACM Trans. Graph* 11(2):152–182
77. Hatna A, Grieve B (2000) Cartesian machining versus parametric machining: a comparative study. *Int. J. Prod. Res.* 38(13):3043–3065
78. Held M (1991) A geometry-based investigation of the tool path generation for zigzag pocket machining. *Vis. Comput* 7(5–6):296–308
79. Held M (1991) *On the computational geometry of pocket machining*. Springer, New York
80. Held M, Klosowski J, Mitchell JSB (1995) Evaluation of collision detection methods for virtual reality fly-throughs. In *proceedings Seventh Canadian Conference on Computational Geometry*, 205–210
81. Held M, Luk'acs G, Andor L (1994) Pocket machining based on contour-parallel tool paths generated by means of proximity maps. *Comput. Aided Des.* 26(3):189–203
82. Hook TV (1986) Real-time shaded nc milling display. In *SIGGRAPH '86: Proceedings of the 13th annual conference on computer graphics and interactive techniques*, 15–20
83. Hornung C, Lellek W, Rehwald P, Straßer W (1985) An area oriented analytical visibility method for displaying parametrically defined tensor-product surfaces. *Comput Aided Geom Des* 2(1–3):197–205
84. Ho S, Sarma S, Adachi Y (2001) Real-time interference analysis between a tool and an environment. *Comput. Aided Des* 33(13):935–47
85. Hosseinkhani Y, Akbaria J, Vafaeseefat A (2007) Penetration-elimination method for five-axis CNC machining of sculptured surfaces. *Int. J. Mach. Tools Manuf.* 47:1625–1635
86. Hsueh Y-W, Hsueh M-H, Lien H-C (2007) Automatic selection of cutter orientation for preventing the collision problem on a five-axis machining. *Int J Adv Manuf Tech* 33(9–10):994–1000
87. Hu J, Xiao L, Wang Y, Wu Z (2006) An optimal feed rate model and solution algorithm for a high-speed machine of small line blocks with look-ahead. *Int J Adv Manuf Tech* 28(9): 930–935
88. Huang Y, Oliver JH (1995) Integrated simulation, error assessment, and tool path correction for five-axis NC machining. *J. Manuf. Syst.* 14(5):331–334
89. Huang J-T, Yang DCH (1992) A generalized interpolator for command generation of parametric curves in computer controlled machines. *Proceedings of the Japan/USA Symposium on Flexible Automation*, ASME 1:393–399
90. Ilushin O, Elber G, Halperinc D, Weinc R, Kim MC (2005) Precise global collision detection in multi-axis NC-machining. *Comput. Aided Des.* 37:909–920
91. Ivanenko SA (1988) Generation of non-degenerate meshes. *USSR Computational Mathematics and Mathematical Physics* 28:141–146
92. Ivanenko SA (1999) Harmonic mappings. In: Thompson JF, Soni BK, Weatherill NP (eds) *Handbook of grid generation* 8. CRC, New York, pp 1–43
93. Jeong J, Kim K (1999) Generating tool paths for free-form pocket machining using z-buffer-based Voronoi diagrams. *Int J Adv Manuf Tech* 15(3):182–187
94. Jeong J, Kim K (1999) Generation of tool paths for machining free-form pockets with islands using distance maps. *Int J Adv Manuf Tech* 15(5):311–316
95. Jerard RB, Drysdale RL (1989) Methods for geometric modeling, simulation and spatial verification of NC machining programs. In: Wozny MJ, Turner JU, Pegna J (eds) *Product modeling for computer-aided design and manufacturing*. Elsevier/North-Holland, New York
96. Jun C-S, Cha K, Lee Y-S (2003) Optimizing tool orientations for 5-axis machining by configuration-space search method. *Comput. Aided Des.* 35(6):549–566
97. Juttler B (2001) Hermite interpolation by Pythagorean hodograph curves of degree seven. *Math Comput* 70(235):1089–1111
98. Kang J-K, Suh S-H (1997) Machinability and set-up orientation for five-axis numerically controlled machining of free surfaces. *Int J Adv Manuf Tech* 13(5):311–325
99. Kim H-C, Lee S-G, Yang M-Y (2006) An optimized contour parallel tool path for 2D milling with flat end mill. *Int J Adv Manuf Tech* 31:567–573
100. Kim SJ, Lee D-Y, Kim H-C, Lee S-G, Yang M-Y (2006) CL surface deformation approach for a 5-axis tool path generation. *Int J Adv Manuf Tech* 28(5–6):509–517

101. Kiritsis D (1994) High precision interpolation algorithm for 3D parametric curve generation. *Comput. Aided Des.* 26(11):850–856
102. Ko TJ, Kim HS, Park SH (2005) Machineability in NURBS interpolator considering constant material removal rate. *Int. J. Mach. Tools Manuf.* 45(6):665–671
103. Kondo M (1994) Decomposition of complex geometry for a manufacturing application. *Comput. Aided Des.* 26(3):244–252
104. Koren Y (1976) Interpolator for a computer numerical control system. *IEEE Trans Comput* 25(1):32–37
105. Korosec M. (2009) Feed rate optimization in free form machining using NURBS approximation, *Computers & Industrial Engineering*
106. Kiswanto G, Lauwers B, Kruth J-P (2007) Gouging elimination through tool lifting in tool path generation for five-axis milling based on faceted models. *Int J Adv Manuf Tech* 32(3–4):293–309
107. Krishnan KK, Kappen J, Bahr B (2001) Calculation of variable federate and spindle speed for NURBS based CNC machining. *Transactions of NAMRI/SME* 24:429–435
108. Kruth J-P, Klewais P (1994) Optimization and dynamic adaptation of the cutter inclination during five-axis milling of sculptured surfaces. *CIRP ann* 43(1):443–448
109. Lai W, Faddis T, Sorem R (2000) Incremental algorithms for finding the offset distance and minimum passage width in a pocket machining toolpath using the Voronoi technique. *Material Processing Technology* 100:30–35
110. Lai Y-L, Wu JS-S, Hung J-P, Chen J-H (2006) A simple method for invalid loops removal of planar offset curves. *Int J Adv Manuf Tech* 27(11–12):1153–1162
111. Langeron JM, Duc E, Lartigue C, Bourdet P (2004) A new format for 5-axis tool path computation, using B-spline curves. *Comput. Aided Des.* 36(12):1219–1229
112. Lauwers B, Dejonghe P, Kruth JP (2003) Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation. *Comput. Aided Des.* 35(5):421–432
113. Lee Y-S, Chang T-C (1995) 2-phase approach to global tool interference avoidance in 5-axis machining. *Comput. Aided Des* 27(10):715–29
114. Lee Y-S (1997) Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. *Comput. Aided Des.* 29(7):507–521
115. Lee Y-S, Ji H (1997) Surface interrogation and machining strip evaluation for 5-axis CNC die and mold machining. *Int. J. Prod. Res.* 35(1):225–252
116. Lee JN, Lee RS (2007) Interference-free toolpath generation using enveloping element for five-axis machining of spatial cam. *J. Mater. Process. Technol.* 187–188:10–13
117. Lee E (2003) Contour offset approach to spiral toolpath generation with constant scallop height. *Comput. Aided Des.* 35:511–518
118. Lei WT, Sung MP, Lin LY, Huang JJ (2007) Fast real-time NURBS path interpolation for CNC machine tools. *Int. J. Mach. Tools Manuf.* 47:1530–1541
119. Li F, Wang XC, Ghosh SK, Kong DZ, Lai TQ, Wu XT (1995) Tool-path generation for machining sculptured surface. *J. Mater. Process. Technol.* 48(1–4):811–816
120. Li SX, Jerard RB (1994) 5-axis machining of sculptured surfaces with a flat-end cutter. *Comput. Aided Des.* 26(3):165–178
121. Li Z, Chen W (2006) A global cutter positioning method for multi axis machining of sculptured surfaces. *Int. J. Mach. Tools Manuf.* 46(12–13):1428–1434
122. Lin R-S, Koren Y (1996) Efficient tool-path planning for machining free-form surfaces. *ASME Journal of Engineering for Industry* 118(1):20–28
123. Lindstrom P, Turk G (2000) Image-driven simplification. *ACM Trans. Graph* 19(3):204–241
124. Liu X, Ahmad F, Yamazaki K, Mori M (2005) Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics. *Int. J. Mach. Tools Manuf.* 45:433–444
125. Liang H, Hong H, Svoboda J (2002) A combined 3D linear and circular interpolation technique for multi-axis CNC machining. *ASME Journal of Manufacturing Science and Engineering* 124:305–312
126. Lo CC (1997) Feedback interpolator for CNC machine tool. *ASME. Journal of Manufacturing Science and Engineering* 119 (4):587–592
127. Lo CC (1999) Efficient cutter-path planning for five-axis surface machining with a flat-end cutter. *Comput. Aided Des.* 31 (9):557–566
128. Lo CC (1999) Real-time generation and control of cutter path for 5-axis CNC machining. *Int. J. Mach. Tools Manuf.* 39(3):471–488
129. Lo CC (2000) CNC machine tool surface interpolator for ball-end milling of free-form surfaces. *Int. J. Mach. Tools Manuf.* 40(3):307–326
130. De Lacalle LNL, Lamikiz A, Salgado MA, Herranz S, Rivero A (2002) Process planning for reliable high-speed machining of moulds. *Int. J. Prod. Res.* 40(12):2789–2809
131. Loney GC, Ozsoy TM (1987) NC machining of free form surfaces. *Comput Aided Des* 9(2):85–90
132. Ma W, But W-C, He P (2004) NURBS-based adaptive slicing for efficient rapid prototyping. *Comput. Aided Des.* 36(13):1309–1325
133. Makhanov SS (1999) An application of variational grid generation techniques to the tool-path optimization of industrial milling robots. *Comput. Math. Math. Phys.* 39(9):1524–1535
134. Makhanov SS, Batanov D, Bohez E, Sonthipaumpoon K, Anotaipaiboon W, Tabucanon M (2002) On the tool-path optimization of a milling robot. *Computers and Industrial Engineering* 43(3):455–472
135. Makhanov SS, Ivanenko SA (2003) Grid generation as applied to optimize cutting operations of the five-axis milling machine. *Appl. Numer. Math.* 46(3–4):331–351
136. Makhanov SS, Munlin M (2007) Optimal sequencing of rotation angles for five-axis machining. *Int J Adv Manuf Tech* 35 (10):41–54
137. Makhanov SS (2007) Optimization and correction of the tool path of the five-axis milling machine: Part 1. Spatial optimization, *Mathematics and Computers in Simulation* 75(5–6):210–230
138. Makhanov SS (2007) Optimization and correction of the tool path of the five-axis milling machine: Part 2: rotations and setup. *Math. Comput. Simul.* 75(5–6):231–250
139. Anotaipaiboon W, Makhanov SS (2008) Curvilinear space-filling curves for five-axis machining. *Comput. Aided Des.* 40(3):350–367
140. Mani K, Kulkarni P, Dutta D (1999) Region-based adaptive slicing. *Comput. Aided Des.* 31(5):317–333
141. Marciniak K (1987) Influence of surface shape in admissible tool positions in 5-axis face milling. *Comput. Aided Des.* 19(5):233–236
142. Meek DS, Walton DJ (1997) Geometric Hermite interpolation with Tschirnhausen cubics. *J. Comput. Appl. Math* 81:299–309
143. Tikhon M, Ko TJ, Lee SH, Kim HS (2004) NURBS interpolator for constant material removal rate in open NC machine tools. *Int. J. Mach. Tools Manuf.* 44:237–245
144. Monreal M, Rodríguez CA (2003) Influence of tool path strategy on the cycle time of high-speed milling. *Comput. Aided Des.* 35(4):395–401
145. Moon HP, Farouki RT, Choi HI (2001) Construction and shape analysis of PH quintic Hermite interpolants. *Comput Aided Geom Des* 18(2):93–115

146. Morishige K, Takeuchi Y (1995) 5 axis control rough cutting of an impeller with an efficiency and accuracy. *Proceedings 1997 IEEE International Conference on Robotics and Automatics*, 1241–1247
147. Morishige K, Takeuchi Y, Kase K (1999) Tool path generation using C-space for 5-axis control machining. *Journal of Manufacturing Science and Engineering* 121(1):144–149
148. Muller M, Erds G, Xirouchakis PC (2004) High accuracy spline interpolation for 5-axis machining. *Comput. Aided Des.* 36(13):1379–1393
149. Munlin M, Makhanov SS, Bohez ELJ (2004) Optimization of rotations of a five-axis milling machine near stationary points. *Comput. Aided Des.* 36(12):1117–1128
150. My CA, Bohez ELJ, Makhanov SS (2005) Critical point analysis of 3D vector field for 5-axis tool path optimization. In *Proceedings of the 4th Asian Conference on Industrial Automation and Robotics, ACIAR 2005*, 11–13 May 2005, Bangkok, Thailand, 11
151. My CA, Makhanov SS, Munlin M, Bohez ELJ, Phien HN, Tabucanon MT (2005) On 5-axis freeform surface machining optimization: vector field clustering approach. *International Journal of CAD/CAM* 5:1–14
152. Nam S-H, Yang M-Y (2004) A study on a generalized parametric interpolator with real-time jerk-limited acceleration. *Comput. Aided Des.* 36:27–36
153. Narayanaswami R, Choi Y (2001) NC machining of freeform pockets with arbitrary wall geometry using a grid-based navigation approach. *Int J Adv Manuf Tech* 18(10):708–716
154. Naylor B, Amanatides J, Thibault W (1990) Merging BSP trees yields polyhedral set operations. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 115–124
155. Noborio H, Fukuda S, Arimoto S (1989) Fast interference check method using octree representation. *Adv Robot* 3(3):193–212
156. Park SC, Choi BK (2000) Tool-path planning for direction parallel area milling. *Comput. Aided Des.* 32(1):17–25
157. Park SC, Chung YC (2002) Offset tool-path linking for pocket machining. *Comput. Aided Des.* 34(4):299–308
158. Park SC, Chung YC, Choi BK (2003) Contour-parallel offset machining without tool-retractions. *Comput. Aided Des.* 35(9):841–849
159. Park S, Kim SH, Cho H (2006) Kernel software for efficiently building, reconfiguring, and distributing an open CNC controller. *Int J Adv Manuf Tech* 27:788–796
160. Persson H (1978) NC machining of arbitrarily shaped pockets. *Comput Aided Des* 10(3):169–174
161. Piegel L, Tiller W (1995) *The NURBS book*. Springer, London
162. Pi J, Red E, Jensen G (1998) Grind-free tool path generation for five-axis surface machining. *Computer Integrated Manufacturing Systems* 11(4):337–350
163. Piazza A, Visioli A (1998) Global minimum-time trajectory planning of mechanical manipulators using interval analysis. *Int. J. control* 71(4):631–652
164. Pottmann H, Wallner J, Glaeser G, Ravani B (1999) Geometric criteria for gouge-free three-axis milling of sculptured surfaces. *ASME J. Mech. Des* 121(2):241–248
165. Pottmann H, Ravani B (2000) Singularities of motions constrained by contacting surfaces. *Mech Mach Theory* 35(7):963–984
166. Pateloup V, Chanal H, Duc E, Ray P (2006) HSM-adapted tool path calculation for pocketing. *Mach. Sci. Technol* 10(2):181–196
167. Radzevich SP, Goodman ED (2002) Computation of optimal workpiece orientation for multi-axis NC machining of sculptured part surfaces. *ASME J. Mech. Des* 124(2):201–212
168. Radzevich SP (2005) A cutting-tool-dependent approach for partitioning of sculptured surface. *Comput. Aided Des.* 37(7):767–778
169. Rao A, Sarma R (2000) On local gouging in five-axis sculptured surface machining using flat-end tools. *Comput. Aided Des.* 32(7):409–420
170. Rao N, Ismail F, Bedi S (1997) Tool path planning for five-axis machining using the principal axis method. *Int. J. Mach. Tools Manuf.* 37(7):1025–1040
171. Red EW (2000) A dynamic optimal trajectory generator for Cartesian path following. *Robotica* 18:451–458
172. Renton D, Elbestawi MA (2000) High speed servo control of multi-axis machine tools. *Int. J. Mach. Tools Manuf.* 40:539–559
173. Roth D, Ismail F, Bedi S (2003) Mechanistic modeling of the milling process using an adaptive depth buffer. *Comput. Aided Des.* 35(14):1287–1303
174. Roy U, Xu Y (1999) Computation of a geometric model of a machined part from its NC machining programs. *Comput. Aided Des.* 31(6):401–411
175. Sata T, Kimura F, Okada N, Hosaka M (1981) A new method of NC interpolation for machining the sculptured surface. *CIRP Ann* 30(1):369–372
176. Sang-Kyu Lee S-LK (2002) Development of simulation system for machining process using enhanced Z map model. *J. Mater. Process. Technol.* 130–131:608–617
177. Sarma R, Dutta D (1997) The geometry and generation of NC tool paths. *ASME J. Mech. Des.* 119:253–258
178. Sarma R (2000) An assessment of geometric methods in trajectory synthesis for shape-creating manufacturing operations. *J. Manuf. Syst.* 19(1):59–72
179. Shpitalni M, Koren Y, Lo CC (1994) Realtime curve interpolators. *Comput. Aided Des* 26:832–838
180. Sheen B-T, You C-F (2006) Tool path generation for arbitrary pockets with islands. *J Intell Manuf* 17:275–283
181. Shin K, McKay N (1986) A dynamic programming approach to trajectory planning of robotic manipulators. *IEEE Transaction on Automatic Control* 31:491–500
182. Siller H, Rodriguez CA, Ahuett H (2006) Cycle time prediction in high-speed milling operations for sculptured surface finishing. *J. Mater. Process. Technol.* 174(1–3):355–362
183. S'ir Z, Feichtinger R, J'uttler B (2006) Approximating curves and their offsets using biarcs and Pythagorean hodograph quintics. *Comput. Aided Des.* 38(6):608–618
184. S'ir Z, J'uttler B (2005) Constructing acceleration continuous tool paths using Pythagorean Hodograph curves. *Mech Mach Theory* 40(11):1258–1272
185. Sencer B, Altintas Y, Croft Y (2008) Feed optimization for five-axis CNC machine tools with drive constraints. *Int. J. Mach. Tools Manuf.* 48:733–745
186. Suh SH, Shin YS (1996) Neural network modeling for tool path planning of rough cut in complex pocket milling. *J. Manuf. Syst.* 15(5):295–304
187. Sun Y-W, Guoa D-M, Jia Z-Y (2006) Spiral cutting operation strategy for machining of sculptured surfaces by conformal map approach. *J. Mater. Process. Technol.* 180(1–3):74–82
188. Suresh K, Yang DCH (1994) Constant scallop-height machining of free-form surfaces. *ASME Journal of Engineering for Industry* 116(2):253–259
189. Takata S (1989) A cutting simulation system for machinability evaluation using a workpiece model. *CIRP ann* 38(1):417–420
190. Takeuchi Y, Shimizu H, Idemura T, Watanabe T, Ito T (1990) 5-axis control machining based on solid model. *Journal of the Japan Society for Precision Engineering* 56(1):111–116
191. Takeuchi Y, Idemura T, Sata T (1991) 5-Axis Control Machining and Grinding Based on Solid Model. *CIRP Annals-Manufacturing Technology* 40(1):455–458
192. Takeuchi Y, Watanabe T (1992) Generation of 5-axis control collision-free tool path and postprocessing for NC data. *CIRP Annals-Manufacturing Technology* 41(1):539–542

193. Tsai M-S, Nien H-W, Yau H-T (2008) Development of an integrated look-ahead dynamics-based NURBS interpolator for high precision machinery. *Comput. Aided Des.* 40:554–566
194. Tseng YJ, Joshi S (1991) Determining feasible tool-approach directions for machining Bezier curves and surfaces. *Comput. Aided Des.* 23(5):367–378
195. Tata K, Fadel G, Bagchi A, Aziz N (1998) Efficient slicing for layered manufacturing. *Rapid Prototyping J* 4(4):151–167
196. Tung ED, Tomizuka M (1993) Feedforward tracking controller design based on the identification of low frequency dynamics. *J. Dyn. Syst. Meas. Control* 115(3):348–356
197. Tang TD, Bohez ELJ, Koomsap P (2007) The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis NC machining. *Comput. Aided Des.* 39:1012–102
198. Tarkkainen M, Shiller Z (1993) Time optimal motions of manipulators with actuator dynamics, in: IEEE International Conference on Robotics and Automation, vol. 2, Atlanta, GA, May 2–6, IEEE Comput. Soc. Press, Los Alamitos, CA: 725–730
199. Vafaeseefa A, El Maraghy HA (1998) Accessibility analysis in 5-axis machining of sculptured surfaces. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation, pages 2464–2469
200. Van'cek G Jr (1991) Brep-index: a multidimensional space partitioning tree. In SMA '91: Proceedings of the first ACM symposium on Solid modeling foundations and CAD/CAM applications, pp. 35–44
201. Wallner J, Pottmann H (2000) On the geometry of sculptured surface machining. In: Laurent P-J, Sablonni'ere P, Schumaker LL (eds) *Curve and Surface Design*. Vanderbilt University Press, Nashville
202. Wang F-C, Wright PK (1998) Open architecture controllers for machine tools. Part 2: a real time quintic spline interpolator. *Journal of Manufacturing Science and Engineering* 120(2):425–432
203. Wang F-C, Yang DCH (1993) Nearly arc-length parameterized quintic-spline interpolation for precision machining. *Comput. Aided Des.* 25(5):281–288
204. Wang K (2003) Solid modeling for optimizing metal removal rate of three dimensional NC end milling. *J. Manuf. Syst.* 7 (1):57–65
205. Warkentin A, Ismail F, Bedi S (1998) Intersection approach to multi-point machining of sculptured surfaces. *Comput Aided Geom Des* 15(6):567–584
206. Warkentin A, Ismail F, Bedi S (2000) Multi-point tool positioning strategy for 5-axis machining of sculptured surfaces. *Comput. Aided Geom. Des* 17(1):83–100
207. Weck M, Meylahn A, Hardebusch C (1999) Innovative algorithms for spline-based CNC controller, production engineering research and development in Germany. *Annals of the German Academic Society for Production Engineering* VI(1):83–86
208. Weck M, Ye GH (1990) Sharp corner tracking using the IKF control strategy. *CIRP ann* 39(1):437–441
209. Woo T, Turkovich B (1990) Visibility map and its application to numerical control. *CIRP anna* 39(1):451–454
210. Woo T, Gan J (1992) Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining. *J. Mech. Des* 114:477–485
211. Woo TC (1994) Visibility maps and spherical algorithms. *Comput Aided Des* 26(1):6–16
212. Wings E, Juttler B (2004) Generating tool paths on surfaces for a numerically controlled calotte cutting system. *Comput. Aided Des.* 36:325–331
213. Winslow AM (1966) Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh. *J Comput Phys* 1(2):149–172
214. Xu XJ, Bradley C, Zhang YF, Loh HT, Wong YS (2002) Tool-path generation for five-axis machining of free-form surfaces based on accessibility analysis. *Int. J. Prod. Res.* 40(14):3253–3274
215. Lee Y-S, Chang T-C (1996) Automatic cutter selection for five-axis sculptured surface machining. *Int J Prod Res* 34(4):977–998
216. Yeh SS, Hsu PL (1999) The speed-controlled interpolator for machining parametric curves. *Comput. Aided Des.* 31: 349–357
217. Yeung MK, Walton DJ (1994) Curve fitting with arc splines for NC tool path generation. *Comput. Aided Des.* 26(11):845–849
218. Yoon J-H (1997) Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining. *Int. J. Prod. Res.* 41(10):2125–2142
219. You CF, Sheen BT, Lin TK (2001) Robust spiral tool-path generation for arbitrary pockets. *Adv. Manuf. Technol.* 17:181–188
220. Yoon J-H, Pottmann H, Lee Y-S (2003) Locally optimal cutting positions for 5-axis sculptured surface machining. *Comput. Aided Des.* 35(1):69–81
221. Yoon J-H (2005) Fast tool path generation by the iso-scallop height method for ball-end milling of sculptured surfaces. *Int. J. Prod. Res.* 43:5061–5070
222. Young H-T, Chuang L-C, Gerschweiler K, Kamps S (2004) A five-axis rough machining approach for a centrifugal impeller. *Int J Adv Manuf Tech* 23(3–4):233–239
223. Yang DCH, Kong T (1994) Parametric interpolator versus linear interpolator for precision CNC machining. *Comput Aided Des* 26:225–234
224. Zhang QG, Greenway RB (1998) Development and implementation of a NURBS curve motion interpolator. *Robot. Comput.-Integr. Manuf* 14(1):27–36
225. Kang M, Lee S-K, Ko S-L (2002) Optimization of Cutting Conditions using Enhanced Z Map Model. *CIRP Annals - Manufacturing Technology* 51(1):429–432

Optimization of rotations for six-axis machining

Ahmarn Mudcharoen · Stanislav S. Makhanov

Received: 23 September 2009 / Accepted: 28 July 2010
© Springer-Verlag London Limited 2010

Abstract Five-axis machines with three translational and two rotation axes are becoming increasingly popular in serving the needs of the mass production industry due to their ability to handle geometrically complex workpieces using the rotational axes. Theoretically, the combination of the five axes offers a minimal number of the degrees of freedom required to transport the tool into a prescribed spatial position and establish a required orientation. However, the rotation axes lead to an inevitable nonlinearity of the tool tip trajectory and the so-called kinematics errors appearing due to the specific kinematics of the machine. Eventually, one arrives at an interesting question. Is it possible to compensate this error by introducing an additional rotation axis? In other words, “does an additional rotation axis offer any optimization benefits in the sense of the above mentioned error?” In this paper, we answer this question positively by analyzing a hypothetical six-axis milling machine with two rotation axes on the table and one additional rotation axis on the tool. The sixth axis is built on the top of the existing five-axis machine MAHO600E by Deckel Gildemeister. We present an extension of an optimization algorithm developed earlier by the authors for five-axis machining based on an optimal angle sequencing (the shortest path optimization). The extension is a combination of the shortest path strategy and the use of the additional axis. The algorithm leads to an increase in the machining accuracy, in particular, for rough milling. Numerical experiments and cutting by a virtual six-axis machine built in Vericut 5.0 validates the results of the

optimization. The proposed optimization procedure is capable of upgrading the existing five-axis G-codes to the case of six-axis machine.

Keywords Multi-axis machining · Optimization · Kinematics of milling machines

1 Introduction

Milling machines are programmable mechanisms for cutting complex industrial parts. The machine consists of several moving bodies designed to establish the required coordinates and orientations of the tool during the cutting process. The axes of the machine define the number of the degrees of freedom of the cutting device. The movements of the machine are guided by a controller which is fed with a so-called NC program comprising commands carrying spatial coordinates of the tool-tip and angles needed to rotate the machine parts to establish the orientation of the tool.

One of the most popular five-axis configurations consists of three translation and two rotation axes. This combination offers a minimal number of the degrees of freedom required to transport the tool into a spatial position and to establish the required orientation.

The machines with the rotation axes on the table often have to turn around heavy workpieces. Therefore, they must support significant mechanical efforts during machining. As a result, the machines may have low capacities for acceleration. When the machine has to slow down or stop, the speed reduction requires a considerable time for deceleration and re-acceleration. This effect significantly increases the machining time and can be even amplified in high-speed machining when the rotation axes must reach

A. Mudcharoen · S. S. Makhanov (✉)
School of Information, Computer and Communication
Technology, Sirindhorn International Institute of Technology,
Thammasat University,
Bangkok, Thailand
e-mail: makhanov@siit.tu.ac.th

greater speeds. As a matter of fact, it can be often financially justified to use only three linear axes at high speeds to reduce the number of decelerating axes. This is also because the rotation axes invoke an inevitable nonlinearity of the tool trajectory and, consequently, errors appearing due to specific kinematics of the machine (the kinematics errors).

Therefore, we arrive at an interesting question. Does an additional sixth rotational axis offer any optimization benefits in the sense of the abovementioned kinematics error? In other words, can the rotations (not the tool positions!) be optimized so that the kinematics errors are reduced as compared to the five-axis machining?

Note that the six-axis configuration is usually being discussed with the reference to the Stewart platform mechanisms. Only a few papers deal with kinematics of conventional six-axis machines. The conventional six-axis configurations have been recently analyzed with regard to non-rotational tool in [1, 2] and with regard to six-axis grinding in [3]. However, to the best of our knowledge, the kinematics errors produced by a hypothetical or a real six-axis point milling and possible benefits of sixth axis have not been presented in the literature.

This paper offers a positive view regarding benefits of the sixth axis; however, it also shows that the additional axis cannot be used in an ad hoc fashion. Moreover, in some cases, it is not efficient. For demonstration, we present an analysis of a hypothetical six-axis milling machine with two rotation axes on the table and an additional rotation axis on the tool. The sixth axis is built on the top of the existing five-axis machine MAHO600E by Deckel Gildemeister.

The analysis shows that the six-axis machine requires special optimization algorithms. Otherwise, the result can be even worse than that produced by five-axis machining.

Therefore, a new optimization procedure has been proposed and analyzed. The new algorithm is an extension of the shortest path optimization developed earlier by the authors for five-axis machining [4, 5]. The algorithm is based on minimization of the total distance traveled by the tool in the angular space.

The algorithm leads to a significant increase in the machining accuracy, in particular, for rough milling displaying clearly optimization benefits of the additional axis.

The algorithm does not change the prescribed tool path in the part surface coordinate system and changes only the way the orientations are being achieved. The proposed procedure is capable of upgrading and optimizing the existing five-axis G-codes to the case of six axes.

Finally, the basic virtual five-axis machine has been successfully tested and compared with the actual machine Maho600E at the CIM Lab of Asian Institute of Technology of Thailand (see, for instance [4, 5]). The sixth axis has

been built at the top of this well-known and verified virtual machine. Virtual machining using Vericut has been proven to correctly represent a variety of multi-axis machines in numerous reports and publications. Therefore, the virtual machine is a practical way to verify the proposed algorithm.

2 Kinematics of six-axis milling machines

Consider a six-axis machine with two rotary axes on the table and an additional axis on the tool (Fig. 1). The machine is guided by axial commands carrying three spatial coordinates of the tool-tip in the machine coordinate system M and three rotation angles A , B , and C . A successive set of coordinates (cutter location points or CL points) in the workpiece (part surface) coordinate system W is transformed into the M system to provide reference inputs for the servo-controllers of the machine.

Define the kinematics transformation of the machine $\mathcal{R}(M, \mathcal{R}) : R^6 \rightarrow R^3$. For every point $M \equiv (X, Y, Z)$ in the machine coordinates and every triple of the rotation angles $\mathcal{R} \equiv (A, B, C)$ $\mathcal{R}(M, \mathcal{R})$ generates a corresponding point $W \equiv (x, y, z)$ in the workpiece coordinates. Examples of the kinematics transformations for some particular six-axis machines are presented in the Appendix.

Let $\mathcal{R}^{-1}(W, \mathcal{R})$ be the inverse transformation such that $\forall W, M, \mathcal{R}, \mathcal{R}^{-1}(\mathcal{R}(M, \mathcal{R}), \mathcal{R}) \equiv M$ and $\mathcal{R}^{-1}(\mathcal{R}(W, \mathcal{R}), \mathcal{R}) \equiv W$. Let W_p and W_{p+1} be two successive spatial positions in the part-surface coordinates and let $M_p \equiv \mathcal{R}^{-1}(\mathcal{R}_p, W_p)$, $M_{p+1} \equiv \mathcal{R}^{-1}(\mathcal{R}_{p+1}, W_{p+1})$. The machine coordinates and the

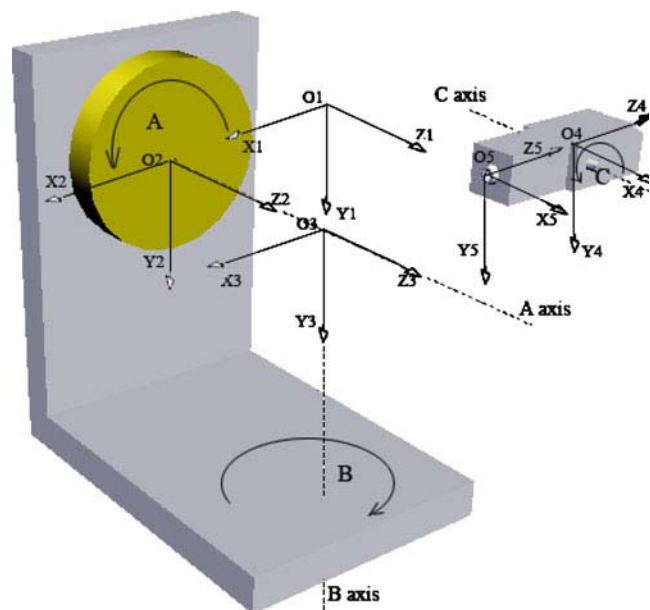


Fig. 1 Hypothetical six-axis machine build on the top of MAHOO600E

rotation angles between p and $p+1$ are assumed to change linearly, namely, $M(t) = tM_{p+1} + (1-t)M_p$, $\mathcal{R}(t) = t\mathcal{R}_{p+1} + (1-t)\mathcal{R}_p$, where t is the fictitious time coordinate ($0 \leq t \leq 1$). Transforming machine coordinates M back to workpiece coordinates W for every t yields

$$W_{p,p+1}(t) = \mathcal{R}(\mathcal{R}(t), M(t)) = \mathcal{R}(t\mathcal{R}_{p+1} + (1-t)\mathcal{R}_p, tM_{p+1} + (1-t)M_p).$$

The above represents a tool tip trajectory between p and $p+1$ as a function of the workpiece coordinates and the tool orientation. Kinematics transformations for the specific six-axis machine shown in Fig. 1 are obtained

$$A \equiv A(C) = \begin{cases} \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))}, & \text{if } (-j \tan(C) + i \sin(B)) > 0 \text{ and } (j \sin(B) + i \tan(C)) > 0, \\ \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))} + \pi, & \text{if } (-j \tan(C) + i \sin(B)) < 0, \\ \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))} + 2\pi, & \text{otherwise.} \end{cases} \quad (2)$$

$$B \equiv B(C) = \cos^{-1} \left(\frac{k}{\cos(C)} \right). \quad (3)$$

Note that (3) implies $\frac{k}{\cos(C)} \leq 1$. Also, note that any angle can be selected as the independent (free) variable; however, the selection of C has some benefits because the C axis is at the end of the kinematics chain [8].

3 Optimization with regard to the sixth axes

Introduce a kinematics error defined as a norm of the difference between the desired the actual trajectory $W_{p,p+1}(t) \equiv (x_{p,p+1}(t), y_{p,p+1}(t), z_{p,p+1}(t))$, namely,

$$\varepsilon = \sum_p \text{dist}(W_{p,p+1}^D, W_{p,p+1}), \quad (4)$$

where $W_{p,p+1}^D(t) \equiv (x_{p,p+1}^D(t), y_{p,p+1}^D(t), z_{p,p+1}^D(t)) \in S(u, v)$ is a curve between tool positions W_p and W_{p+1} extracted

using intermediate reference coordinate systems shown in Fig. 2 (see [6, 7]).

The kinematics of the machine works in such a way that the tool vector $(1, 0, 0)$ in the machine coordinate system O_5 is transformed into the tool orientation vector (i, j, k) in O_1 . Introducing the rotation angles A , B , and C and applying the corresponding rotation matrices to $(1, 0, 0)$ yields

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} \sin(A) \sin(C) + \cos(A) \sin(B) \cos(C) \\ -\cos(A) \sin(C) + \sin(A) \sin(B) \cos(C) \\ \cos(B) \cos(C) \end{pmatrix}. \quad (1)$$

Solving with regard to A , B and leaving C as a free variable yields

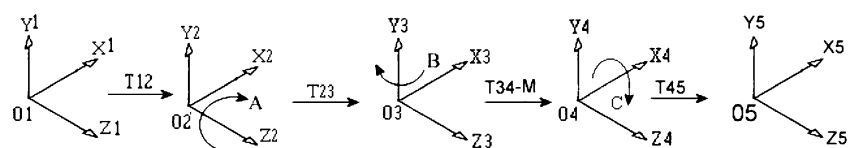
from part surface $S(u, v)$, t is the parametric coordinate along the curve and dist is an appropriate distance. The curve is extracted in such a way that it represents the desired tool trajectory. For every cutting step $p, p+1$ the tool tip trajectory must be as close as possible to the desired surface.

The actual trajectory $W_{p,p+1}(t)$ is generated using inverse transformations derived for the specific machine configuration (see, for instance, [2]). Note that ε is nothing else than a numerical approximation of the deviation between the machined and the actual surface. As a matter of fact, the actual error might even exceed ε ; however, ε is a conventional error estimate which should be minimized first.

Furthermore, we follow an assumption validated by numerical simulations and cutting experiments in [4, 5] and [6] that the kinematics error ε is proportional to the variation of the rotation angles given by

$$f = \sum_p \int \sqrt{\left(\frac{dA_p}{ds}\right)^2 + \left(\frac{dB_p}{ds}\right)^2 + \left(\frac{dC_p}{ds}\right)^2} ds. \quad (5)$$

Fig. 2 Intermediate reference coordinate systems for the six-axis machine



Approximating (5), using tool positions $(\mathcal{R}_p, W_p)$ for discretization yields

$$f \approx f_C \equiv f_C(C_1, C_2, \dots, C_N) = \frac{1}{N-1} \sum_{p=2}^N (A_p(C_p) - A_{p-1}(C_{p-1}))^2 + (B_p(C_p) - B_{p-1}(C_{p-1}))^2 + (C_p - C_{p-1})^2,$$

where N is the total number of the cutter location points. Note that f_C can be interpreted as the average (per step) distance traveled by the tip of the tool in the angular space (A, B, C) .

We seek

$$\arg \min f_C(C_1, C_2, \dots, C_N) \quad (6)$$

subject to

$$\frac{k_p}{\cos(C_p)} \leq 1, p = 1, \dots, N. \quad (7)$$

Note that (7) defines the C range for a given tool position. For example, if $k=1$ (the tool is parallel to the z -axes), the only possible value is $C=0$.

The constraint minimization (6) and (7) can be performed by many ways, for instance, employing a penalty function

$$\tilde{f} = f_C(C_1, C_2, \dots, C_N) + \lambda \sum_p L(\xi_p), \quad (8)$$

where and $L(\xi) = (\min(0, \xi))^2$, $\xi = k - \cos(C)$.

Unfortunately, $f_C(C_1, C_2, \dots, C_N)$ is not always differentiable. It may include corners and sharp variations near stationary points of the part surface [5] and where the components of the surface normal change the sign (see formula (2)). As an example, consider a test surface depicted in Fig. 3. Two triples of cutter location points are indicated by 1, 2, 3 and 4, 5, 6. The cost function $\tilde{f}(C_2) =$

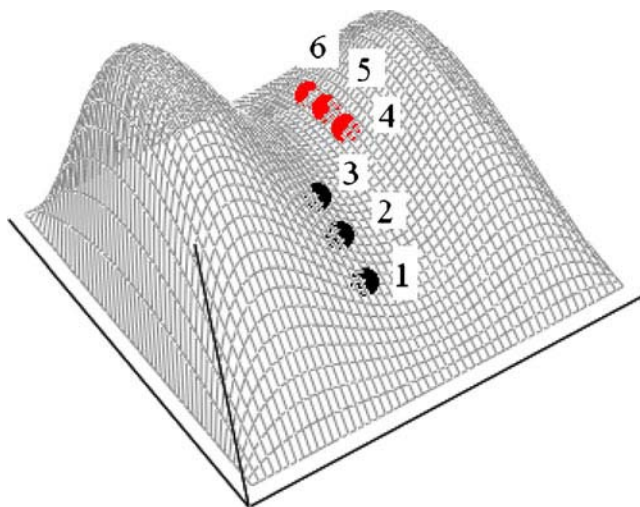


Fig. 3 A test surface and cutter location points 1,2 and 3 and 4,5,6

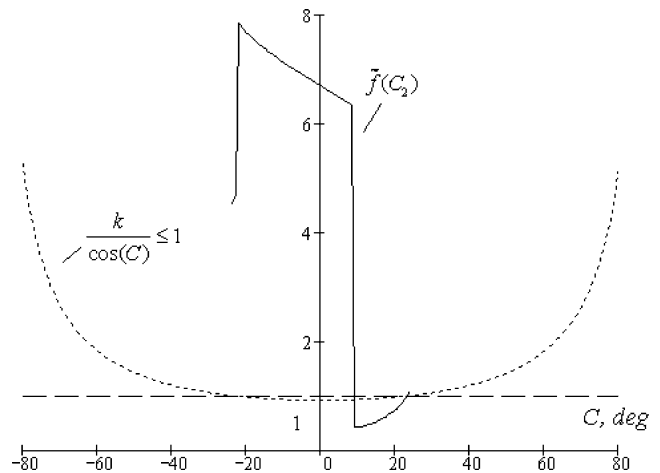


Fig. 4 The cost function and the constraints, point 3 in Fig. 3

$f_C(C_1, C_2, C_3)$ for some fixed C_1 and C_2 is depicted in Fig. 4 (solid line). Figure 5 shows the cost function for the second triple, namely, $\tilde{f}(C_5) = f_C(C_4, C_5, C_6)$ for fixed C_4 and C_6 (solid line). The function $\frac{k}{\cos(C)}$ which defines constraint (6) is depicted by a dotted curve. Note that $\tilde{f}(C_2)$ and $\tilde{f}(C_5)$ are defined only when $\frac{k}{\cos(C)} \leq 1$.

These curves represent typical profiles of the cost function when all the variables, except the one corresponding to the current point, are fixed. The cost function in Fig. 4 has three corners generated by different branches of $A(C)$.

A smooth monotone profile is shown in Fig. 5. Clearly, for this component of the cost function, finding the minimum is a trivial task whereas the function in Fig. 4 may require a combination of continuous and discrete minimization to treat the singularities.

We minimize $\tilde{f}$ using a combination of the gradient based method (a MATLAB built-in routine) and a standard bisection. First, a certain number the gradient minimization

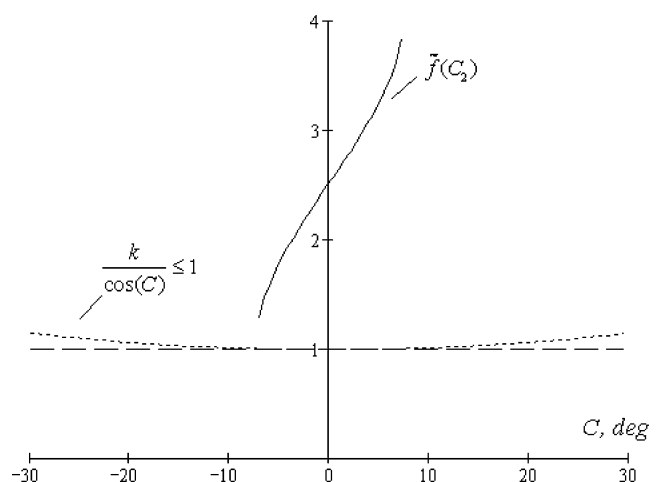


Fig. 5 The cost function and the constraints, point 5 in Fig. 3

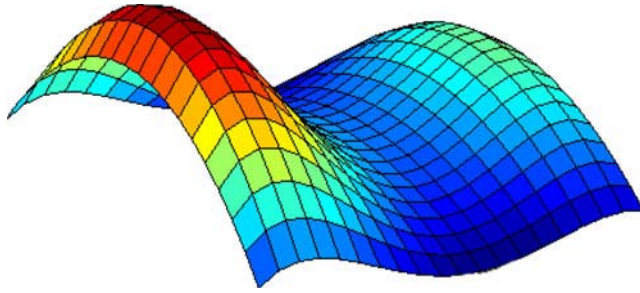


Fig. 6 Test surface 1

steps is performed. Then, if a corner is detected around the local minimum, the method switches to bisection. Otherwise, the gradient minimization continues. Our numerical experiments show that such combination of methods usually converges; however, it does not guarantee the global minimum.

4 Errors induced by an additional axis and other sources of errors

Apart from providing a possibility to optimize the tool path introduced in this paper, the sixth axis may entail a variety of degrading effects on the dynamics of the machine, the stability, and other critical parameters of the machining.

As far as the dynamics are concerned, the additional axis inevitably introduces an additional mass which may reduce the maximum achievable velocity and acceleration. This is, in particular, important when the additional axis is rotational since it is typically slower than the Cartesian (linear) axes. The sixth axis also requires a new controller and an additional servomotor which leads to a higher cost of the machine and an extra power consumption.

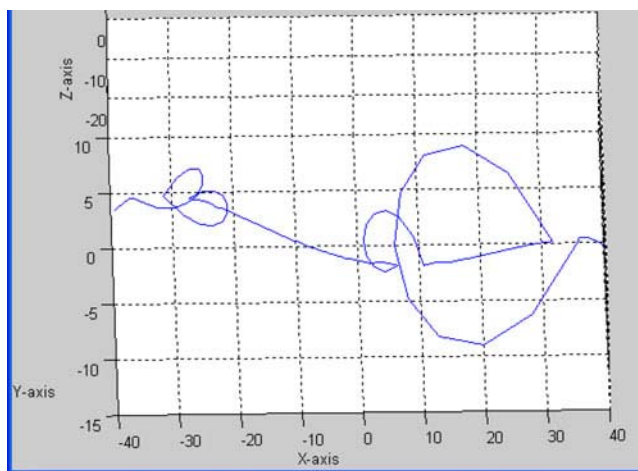


Fig. 7 A trajectory with loops (large errors)

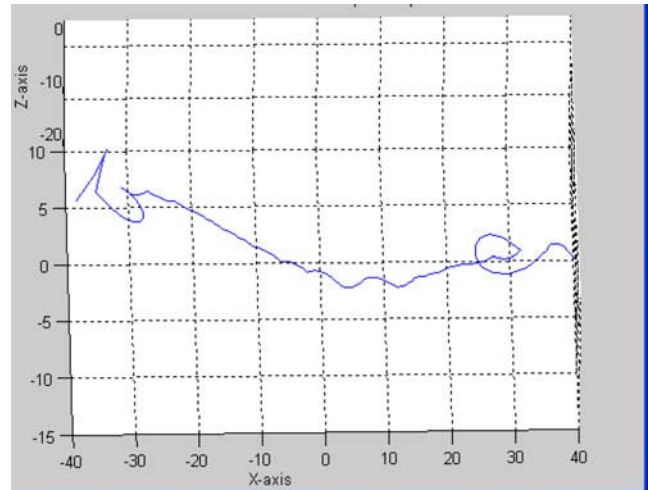


Fig. 8 Loops have been removed by a six-axis optimization

On the other hand, if high-performance servomotors are employed, the redundant axis can be utilized to compensate for velocity or/and acceleration saturation (see for instance [9] on the redundant manipulators). Nevertheless, benefits of the sixth axis should be weighted against possible changes in dynamics as well as additional energy costs. Such analysis presents an interesting optimization problem which, however, lies out of the scope of this paper.

The additional rotational joints may change the stability conditions of the cutting operations (the natural frequencies of the system) which may lead to an unexpected chatter (self-excited vibrations) and unwanted waviness of the part surface.

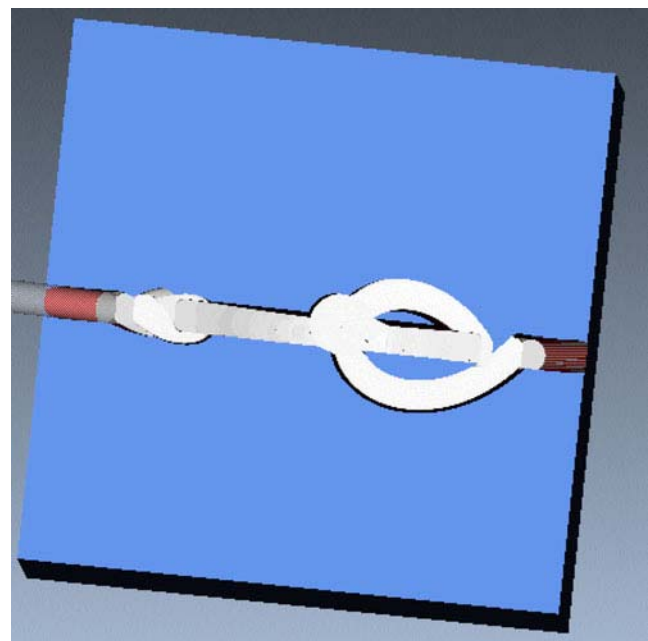


Fig. 9 Non optimized virtual cut. A single curve

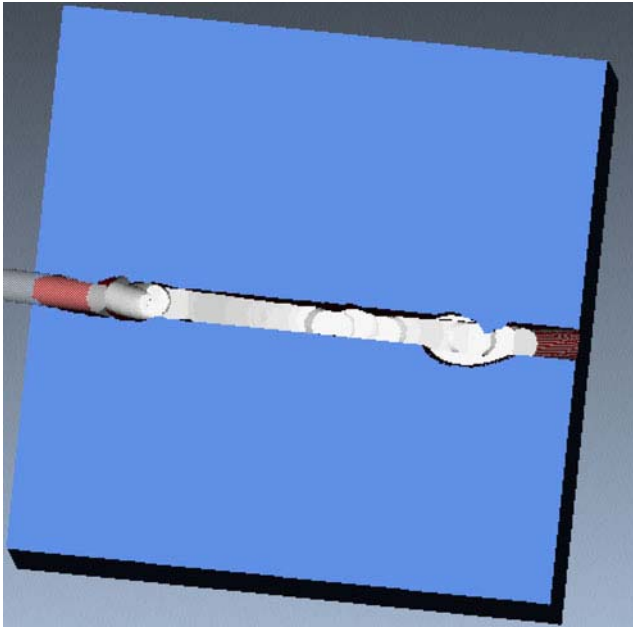


Fig. 10 Optimized virtual cut. A single curve

Even under stable, chatter-free conditions, the tool experiences periodic forced vibrations leading to overcuts or undercuts. In some cases [10], these effects may amount to as much as 75% of all contributions of the error sources. Clearly, the sixth axis can change the characteristics of the forced vibrations and, therefore, increase these inaccuracies.

Among other sources of errors neglected in our model are thermal deformations. It has been demonstrated [11] that additional degrees of freedom may increase these errors since under certain conditions, the thermal deviations associated with different axes add up. A variety of interesting thermal effects appearing during five-axis machining including high-speed micro-milling are analyzed in [11–14]. The effects of the temperature of the main building blocks of the machine on the tool errors are presented in [15, 16].

Furthermore, the tool deflection errors are one of the most prominent [17]. Milling a complex shape may be

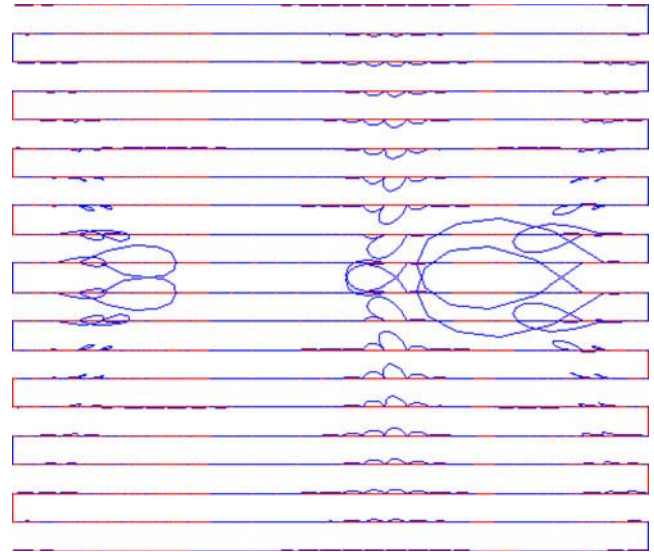


Fig. 11 Five -axis tool trajectories with large loops (errors) before optimization

affected by deflections of the end mills caused by variation of the cutting forces especially when corner cutting is involved [18, 19]. The end milling force and deflection models based on the input machining data (tool geometry, cutting conditions, and workpiece material properties) have been proven to be applicable for generating accurate and reliable cutting force and deflection errors [20, 21]. However, it is difficult to entirely eliminate these errors although a considerable increase in the accuracy of the machining can be achieved by various compensating strategies [22].

Machines with three linear axes have a total of 21 linear independent geometric error components [23]. A five-axis milling machine regarded as a set of rigid bodies has 42 independent geometric error components [24].

The number of the geometric errors for six-axis machine can be evaluated as follows: a rigid solid body has six degrees of freedom specifying uniquely its position in the

Table 1 Cost function and the kinematics error for five-axis and six-axis machining of surface 1

Number of the CL points	The cost function $\tilde{f}$			Kinematics error ϵ		
	Five-axis	Five-axis with sequencing	Six-axis (cost reduction)	Five-axis	Five-axis with sequencing	Six-axis (error reduction)
20×20	19.94	17.87	13.71 (31.24,23.28)%	0.0356	0.0217	0.0160 (55.06,26.27)%
30×20	13.37	12.52	9.69 (27.5,22.60)%	0.0173	0.0112	0.0086 (50.29,35.26)%
40×20	10.05	9.73	7.52 (25.17,22.71)%	0.0089	0.0076	0.0048 (46.07,36.84)%
50×20	8.05	7.76	6.16 (23.48,20.61)%	0.0075	0.0050	0.0037 (50.67,26.00)%
60×20	6.71	6.48	5.23 (22.06,19.29)%	0.0062	0.0037	0.0025 (59.68,32.43)%
70×20	5.75	5.61	4.55 (20.87,18.89)%	0.0046	0.0030	0.0022 (52.17,23.33)%

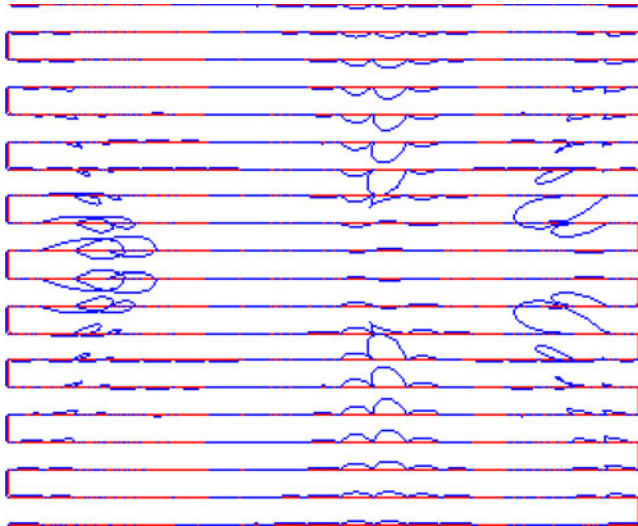


Fig. 12 Five-axis with sequencing. Large loops have been reduced

three-dimensional space. A six-axis machine has six slides that can move relative to each other. Two other bodies fixed to the machine are the tool and the workpiece. The total number of the positioning errors is $8 \times 6 = 48$.

Furthermore, when eight bodies move relative to each other, the number of independent error components can be found by connecting the eight bodies by a minimum number of rigid bars to form a single rigid body. Three rigid bodies can be connected by three rigid bars in a rigid triangle. Four bodies require six rigid bars (tetrahedron), etc. Eight bodies require 18 rigid bars. Therefore, the total number of independent error components is $48 + 18 = 64$. A variety of efficient procedures based on polynomial models to identify and compensate for these errors in the

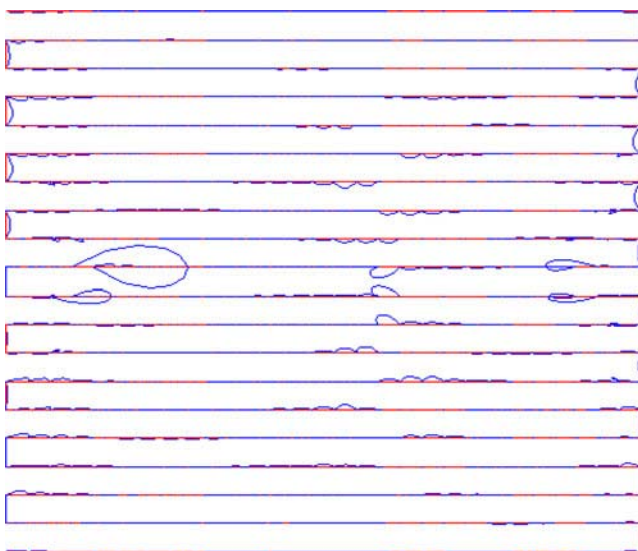


Fig. 13 Six-axis tool trajectories after optimization. Many loops have been eliminated.

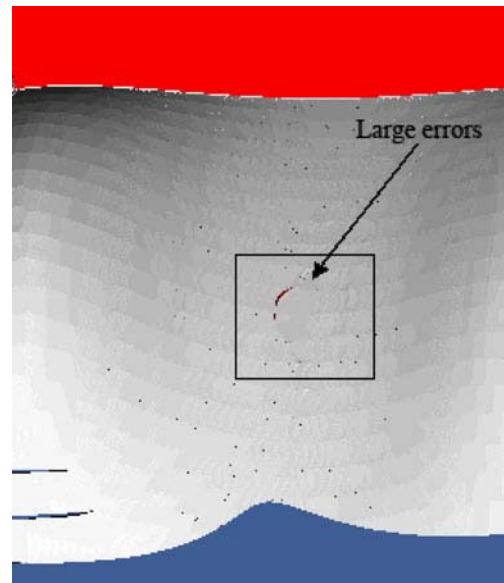


Fig. 14 Fragment of a non optimized five-axis virtual cut with a large error

five-axis mode have been developed and implemented (see a literature survey in [24]). Many of these methods can be generalized to the six-axis case.

Furthermore, some other less prominent sources of errors are: machine operating conditions such as the material removal rate, wet or dry cutting, clamping conditions, the tool wear and other tool imperfections, errors due to configuration of the machine, errors due to singular points of the part surface, etc.

Recall that the tool path is defined by a sequence of cutter contact points and orientation vectors. The positions

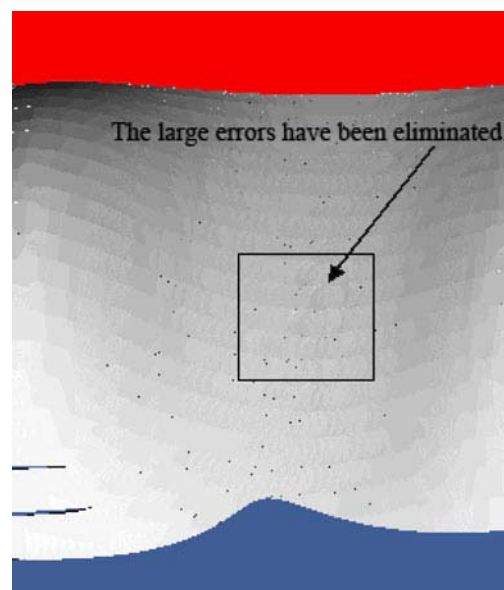


Fig. 15 Fragment of the virtual cut, six-axis machine. The error has been eliminated

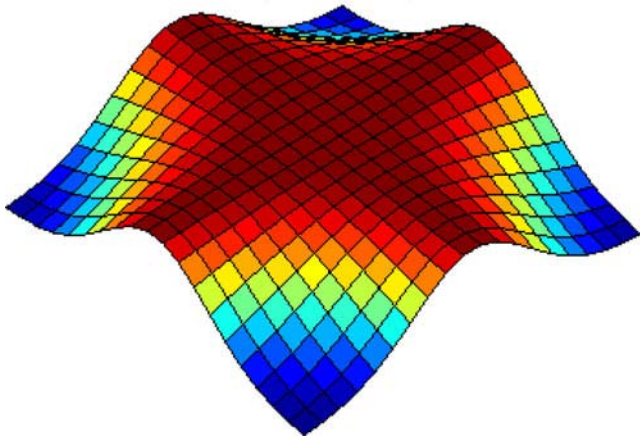


Fig. 16 Test surface 1

of translation and rotary axes are linearly interpolated. Due to the prescribed transformations, the machine tool tip follows curves which represent a continuous piecewise-smooth nonlinear three-dimensional interpolation of the desired trajectories. The deviation of the interpolating function from the actual trajectory defined by (4) is called the kinematics error. However, the definition is not complete. For instance, it does not include the scallops appearing between the tool tracks.

Moreover, the distance involved in (4) is an open problem. The difference can be evaluated by the generic parameter invariant Hausdorff or Fréchet distance given, respectively, by $\text{dist}_H(A, B) = \max\{\max_{a \in A(t)} \min_{b \in B(t)} |a - b|_E, \max_{b \in B(t)} \min_{a \in A(t)} |a - b|_E\}$ and $\text{dist}_F(A, B) = \min_{\{\alpha(t), \beta(t)\}} \max_{t \in [0, 1]} |A(\alpha(t)) - B(\beta(t))|_E$, where minimum in the second formula is considered over all continuous and increasing functions $\alpha(t)$ and $\beta(t)$. However, these measures are computationally expensive and may lead to intractable optimization problems. Some computationally simple choices are the root mean square (rms) given by $\text{dist}_2(A, B) = \| |A(t) - B(t)|_E \|_2$ and the max-norm given by $\text{dist}_\infty(A, B) = \| |A(t) - B(t)|_E \|_\infty$, where $\|_E$ is the Eu-

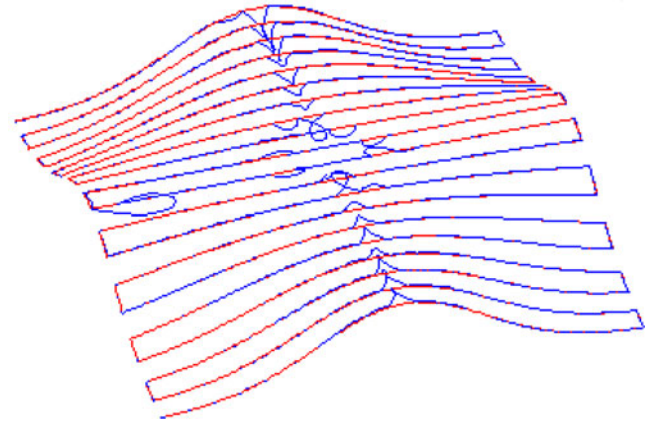


Fig. 18 Six-axis trajectory with reduced loops

clidian distance and $A(t)$, $B(t)$ are parameterized with regard to $t \in [0, 1]$.

A good option is an rms-distance based on a natural parameterization given by

$$\text{dist}_N(A, B) = \sqrt{\int_0^1 |A(l_A(t)) - B(l_B(t))|^2 dt},$$

where $A(l_A(t))$ and $B(l_B(t))$ denote the corresponding arc-length parameterizations.

Unfortunately, it is not possible to find a closed-form parameterization for real rational curves (such as NURBS) by rational functions of its arc length [25]. Therefore, such parameterizations are evaluated numerically.

Finally, finding a distance between curve $W_{p,p+1}(t)$ and the entire part surface $S(u, v)$, rather than between $W_{p,p+1}(t)$ and $W_{p,p+1}^D(t)$, is certainly a better option. However, this solution is computationally hard and often impractical.

The proposed compensation algorithm can be constructed for each of the abovementioned estimates; however, finding an appropriate distance for a particular

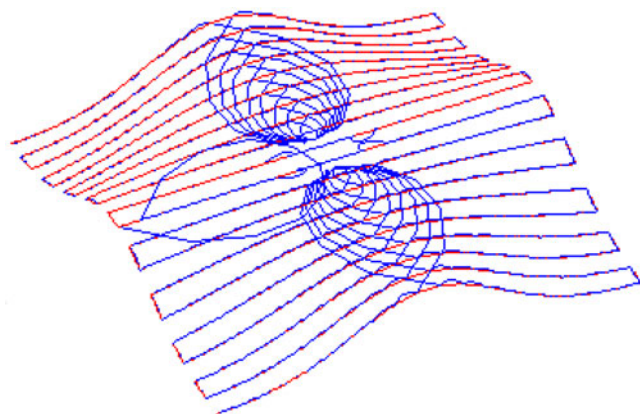


Fig. 17 Five-axis trajectory with loops

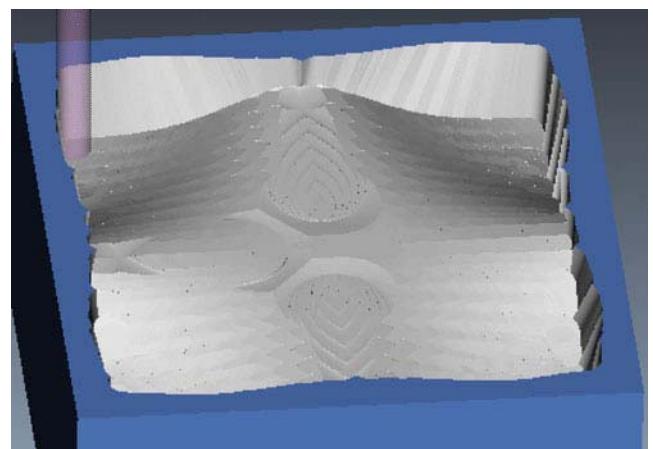


Fig. 19 Five-axis virtual cut

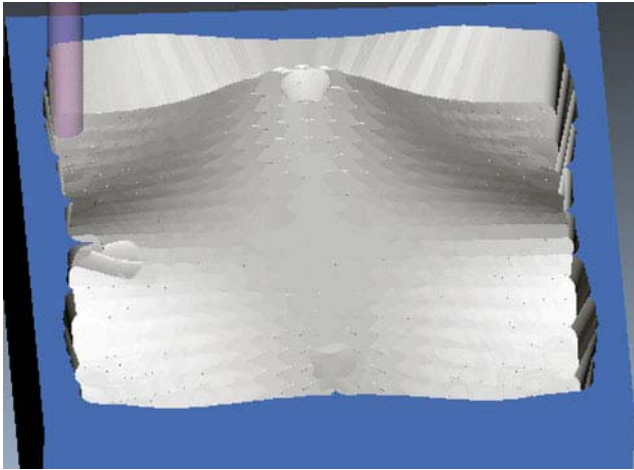


Fig. 20 Six-axis virtual cut with reduced loops

application is often problem dependent. In this paper, the problem of finding the appropriate error estimate is solved under a practical assumption that the error is proportional to the variation of the rotation angles no matter what distance is being considered. However, this assumption is heuristic and has been proven only experimentally [5, 26].

In summary, the proposed method invokes many assumptions. Moreover, in case of real six-axis machining, many types of errors could exceed kinematics error. However, a design of a new machine (which inevitably requires a prototype) lies out of the scope of this presentation. The goal of the paper is to find

$$S(u, v) = \begin{pmatrix} 100u - 50 \\ 100v - 50 \\ -80v(v-1)(3.5u - 14.8u^2 + 21.15u^3 - 9.9u^4) - 28 \end{pmatrix}, u, v \in [0, 1].$$

Our optimization performed on a single curve extracted from the test surface is illustrated in Figs. 7 and 8. A non-optimized tool trajectory characterized by many unwanted loops (large errors) induced by large differences of the

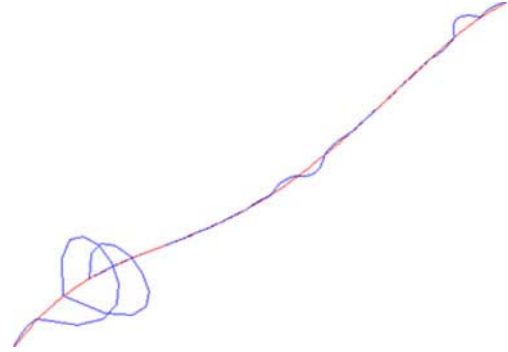


Fig. 21 A trajectory after five-axis sequencing. The large loops at the beginning can not be eliminated. The cost function and the kinematics error are 17.39 and 0.1613 respectively

benefits of the additional rotation axis in the sense of the abovementioned error (4). Consequently, an algorithm to improve the accuracy of the machining using the redundant sixth axis (without changing positions and orientations of the cutting tool!) has been introduced and analyzed.

4.1 Six-axis optimization: numerical experiments

Example 1 Six-axis optimization outperforms five-axis with sequencing

We demonstrate results of optimization (6) and (7) using a test surface given by (Fig. 6)

rotation angles is shown in Fig. 7. The loops produce considerable errors and may even lead to collisions of the machine tool and other parts of the machine. However, the proposed angle optimization procedure eliminates many

Table 2 Cost function and the kinematics error for five-axis and six-axis machining of surface 2

Number of the CL points	The cost function $\tilde{f}$			Kinematics error ε		
	Five-axis	Five-axis with sequencing	Six-axis	Five-axis	Five-axis with sequencing	Six-axis
20×20	14.79	7.68	11.21 (24.21,−31.49)%	0.0573	0.0045	0.0278 (51.48,−83.81)%
30×20	10.31	5.17	8.06 (21.82,−35.85)%	0.0399	0.0013	0.0212 (46.87,−93.87)%
40×20	7.92	3.91	6.94 (12.37,−43.66)%	0.0306	0.000732	0.0170 (44.44,−56.94)%
50×20	5.73	2.87	4.59 (19.90,−37.47)%	0.0287	0.000703	0.0162 (43.55,−95.66)%
60×20	4.66	2.24	3.01 (35.41,−25.58)%	0.0201	0.00065	0.0157 (21.89,−95.86)%
70×20	3.77	2.04	2.56 (32.09,−20.31)%	0.0188	0.00059	0.0115 (38.83,−94.87)%

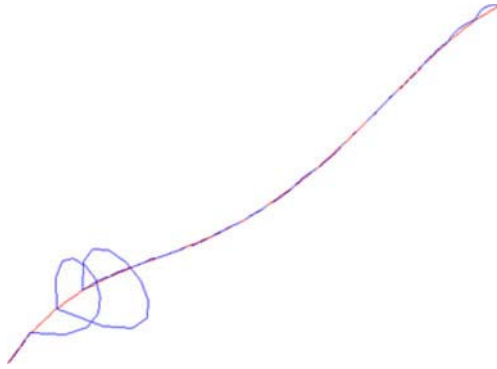


Fig. 22 The trajectory after six-axis sequencing. The large loops become smaller and some errors have been reduced. The cost function and the kinematics error are 15.98 and 0.1542 respectively.

unwanted loops, leading to an improved trajectory shown in Fig. 8.

Virtual cutting using our virtual six-axis machine build in Vericut 5 [27] (Figs. 9 and 10) validates the optimization. Vericut makes it possible to entirely design the milling machine: define the dimensions and the shape of every part and set up the controller. Furthermore, the software makes it possible to perform virtual cutting using a built-in solid modeling engine.

Table 1 shows that the proposed six-axis angle optimization applied to the entire surface leads to a considerable error reduction. For instance, optimization of the tool paths consisting of 400 (20×20) points leads to a cost reduction of about 31.24/23.28%, respectively, with the reference to five-axis and five-axis with sequencing proposed in [3, 4]. For 800 (40×20) points, it is 25.17/22.71%. The kinematics error has been reduced by 55.06/26.27% and 46.07/36.84%.

The tool trajectories from a sample cut (70×20 CL points) for non-optimized and optimized (six-axis) code are shown in Figs. 11, 12, and 13. Finally, fragments of a non-optimized virtual cut Figs. 14 and 15, respectively.

Example 2 Five-axis with sequencing may outperform six-axis optimization

Consider a test surface given by (Fig. 16)

$$S(u, v) = \begin{pmatrix} 100u - 50 \\ 100v - 50 \\ 10.0e^{-10(2.0u-1.0)^2(2.0v-1.0)^2} - 15.0 \end{pmatrix}, u, v \in [0, 1],$$

In order to analyze the efficiency of the six-axis optimization, the results obtained by five-axis machining and by five-axis with sequencing [3] are compared with the proposed six-axis optimization (Figs. 17, 18, 19, 20). The results presented in Table 2 show that five-axis optimization with the optimal choice of rotations [3, 4] outperforms optimization with regard to the sixth axis. Note that six-axis error reduction is close to the five-axis shortest path optimization [3]. However, it has been shown in [4] than inserting new point into the optimized shortest path sequence is not a trivial task, whereas optimization with regard to the six-axis does not present this problem. Next, the six-axis optimization is combined with sequencing proposed in [3]. However, this combination is not as trivial as it may seem. As a matter of fact, it leads to a much more sophisticated mathematical problem presented in the next section.

5 Six-axis optimization with sequencing: numerical experiments

In order to introduce six-axis sequencing, note that (1) has multiple solutions. Suppose A_0, B_0 is a solution (2) and (3) obtained for an arbitrary but fixed C , which satisfies $\frac{k}{\cos(C)} \leq 1$. It is not hard to demonstrate that if $-\frac{\pi}{2} \leq C \leq \frac{\pi}{2}$, then the following triples

$$\begin{aligned} A_1(C) &= 2\pi - A_0(C), B_1(C) = B_0(C), C_1(C) = C, \\ A_2(C) &= A_0(C) + \pi, B_2(C) = -B_0(C), C_2(C) = -C, \\ A_3(C) &= A_0(C) - \pi, B_3(C) = -B_0(C), C_3(C) = -C \end{aligned}$$

are also solutions of (1).

Table 3 Cost function and the kinematics errors: five-axis and six-axis machining of surface 1

Number of the CL points	The cost function $\tilde{f}$			Kinematics error ε		
	Five-axis with sequencing	Six-axis	Six-axis with sequencing	Five-axis with sequencing	Six-axis	Six-axis with sequencing
20×20	17.87	13.71	12.15 (32.01,11.38)%	0.0217	0.0160	0.0145 (33.18,9.36)%
30×20	12.52	9.69	9.01 (28.04,7.02)%	0.0112	0.0086	0.0080 (28.57,6.98)%
40×20	9.73	7.52	7.14 (26.62,5.05)%	0.0076	0.0048	0.0047 (38.16,2.08)%
50×20	7.76	6.16	5.98 (22.94,2.92)%	0.0050	0.0037	0.0036 (28.00,2.70)%
60×20	6.48	5.23	5.17 (20.22,1.15)%	0.0037	0.0025	0.0024 (35.14,4.00)%
70×20	5.61	4.55	4.50 (19.77,1.10)%	0.0030	0.0022	0.0019 (36.67,13.63)%

Table 4 Cost function and the kinematics error: five-axis and six-axis of surface 2

Number of the CL points	The cost function $\tilde{f}$			Kinematics error ε		
	Five-axis with sequencing	Six-axis	Six-axis with sequencing	Five-axis with sequencing	Six-axis	Six-axis with sequencing
20×20	7.68	11.21	6.23 (18.88,44.42)%	0.0045	0.0278	0.0037 (17.78,86.69)%
30×20	5.17	8.06	4.78 (7.54,40.69)%	0.0013	0.0212	0.0009 (30.77,95.75)%
40×20	3.91	6.94	3.11 (20.46,55.19)%	0.000732	0.0170	0.00061 (16.67,96.41)%
50×20	2.87	4.59	2.63 (8.36,42.70)%	0.000703	0.0162	0.000698 (0.71,95.69)%
60×20	2.24	3.01	2.01 (10.27,25.58)%	0.00065	0.0157	0.00060 (7.69,96.17)%
70×20	2.04	2.56	1.99 (2.45,22.27)%	0.00059	0.0115	0.00052 (11.86,95.48)%

Consequently, our new optimization problem is formulated as follows:

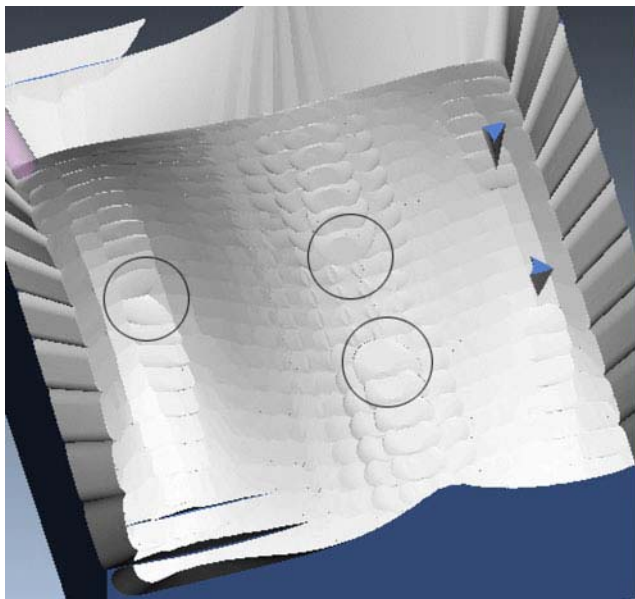
Define

$$\begin{aligned}\tilde{A}(C, d) &= \sum_{k=0}^3 d_k A_k(C), \tilde{B}(C, d) = \sum_{k=0}^3 d_k B_k(C), \tilde{C}(C, d) \\ &= \sum_{k=0}^3 d_k C_k(C),\end{aligned}$$

where $d=(d_0, d_1, d_2, d_3)$ is a binary vector, $d_i=0$ or $d_i=1$.

The minimization problem becomes

$$\begin{aligned}&\underset{C_1, C_2, \dots, C_N, D_1, \dots, D_N}{\text{minimize}} \quad \frac{1}{N-1} \sum_{p=2}^N (\tilde{A}(C_p, D_p) \\ &\quad - \tilde{A}(C_{p-1}, D_{p-1}))^2 \\ &\quad + (\tilde{B}(C_p, D_p) - \tilde{B}(C_{p-1}, D_{p-1}))^2 \\ &\quad + (\tilde{C}(C_p, D_p) - \tilde{C}(C_{p-1}, D_{p-1}))^2,\end{aligned}\quad (9)$$

**Fig. 23** Five-axis with sequencing virtual cut (20×20 CL points)

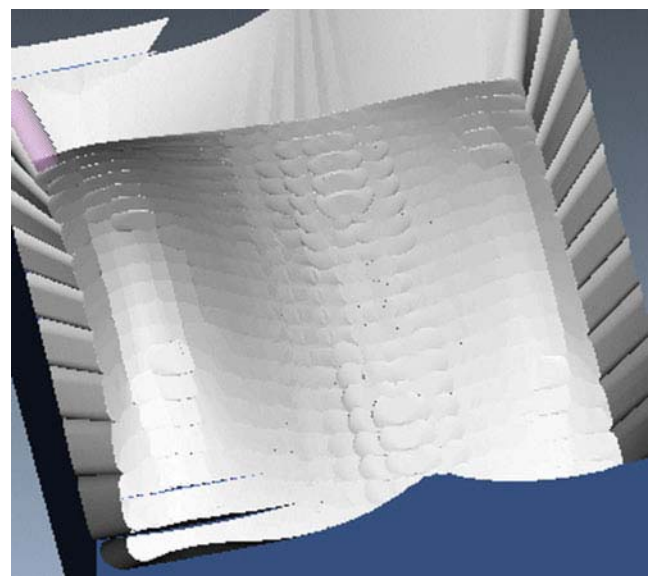
where C_i are continuous variables and $D_i = (d_{0,i}, d_{1,i}, d_{2,i}, d_{3,i})$ are the discrete optimization variables satisfying $d_{0,p} + d_{1,p} + d_{2,p} + d_{3,p} = 1$.

The minimization is subjected to a condition given by

$$\frac{k_p(s)}{\cos(C_p(s))} \leq 1, \forall s, \forall p. \quad (10)$$

It should be noted that this paper does not find the best optimization method for the hard mixed nonlinear optimization problem (9) and (10). Rather than that, a formulation of the problem is presented and a possibility of finding a suboptimal solution using a simple heuristic approach is demonstrated.

The heuristic method is based on the following iterations. First, the integer variables D_i are fixed and the corresponding nonlinear constraint minimization problem is solved with regard to C_i . Next, the continuous variables get “frozen” and a shortest path routine is applied to the binary variables D_i . With the new set of the binary variables and the initial values for C_i taken from the previous step, the

**Fig. 24** six-axis with sequencing virtual cut

problem is solved with regard to the continuous variables again. The procedure repeats until convergence.

Although this method does not guarantee the optimal solution, the engineers are often satisfied with a suboptimal solution as long as it provides that the objective function has been improved sufficiently. In our case, a lower bound on the optimal value is available from the solution of the five-axis shortest path routine which presents a fairly good idea regarding the quality of the solution.

Example 3 Six-axis optimization with sequencing, Surface 1

Let us use test surface 1 to demonstrate results of the six-axis sequencing.

The results obtained for a single curve are shown in Figs. 21 and 22. The errors analysis for the entire surface is given in Table 3. Clearly, the six-axis sequencing makes it possible to achieve an impressive reduction of the error, with regard to five-axis sequencing of about 30%.

Example 4 Six-axis optimization with sequencing, Surface 2

These experiments extend analysis presented by Example 2. Recall that for surface 2, five-axis optimization with optimal choice of rotations outperforms six-axis optimization. Table 4 shows that six-axis optimization (9) and (10) solved by the proposed procedure always improves the result. The error reduction ranges from 19% to 32% in terms of the cost function and 28–36% in terms of the kinematics error. Finally, a rough virtual cut (20×20) shown in Figs. 23 and 24 shows visually large errors produced by five-axis with sequencing compared to a six-axis rough cut with sequencing where these errors have been partially eliminated.

6 Conclusions

The additional rotation axis provides certain optimization benefits. The kinematics error can be substantially reduced at the expense of the correctional rotations around the additional axes. However, special minimization algorithms should be employed. The optimization formulated in terms of a functional representing the total distance traveled by the tool tip in the angular space makes it possible to reduce the kinematics error by 25–30%. The proposed six-axis optimization could be coupled with other schemes to insert additional points or to distribute the existing points in a desirable fashion. Finally, the optimal design of the six-axis machine is still an open problem.

Acknowledgment We acknowledge sponsorship of Thailand Research Fund, grant BRG 50800012.

Appendix: possible configurations of six-axis machines

The presented configurations differ by the number of the rotary axes on the tool and the table.

1. The 2–1 machine

The kinematics equations are obtained by the following coordinate transformations (see Figs. 25 and 26).

Step 1: Coordinate translation $O_1 \rightarrow O_2$

$$P_2 = W + T_{12}, \quad (11)$$

where T_{12} is the coordinate of O_1 in O_2 .

Step 2: Rotation around the A -axis in O_2 by A

$$P_{2A} = R_Z[A]P_2 = R_Z[A](W + T_{12}), \quad (12)$$

where $R_Z[A] = \begin{bmatrix} \cos A & \sin A & 0 \\ -\sin A & \cos A & 0 \\ 0 & 0 & 1 \end{bmatrix}$ is the rotation matrix around the A -axis.

Step 3: Coordinate translation $O_2 \rightarrow O_3$

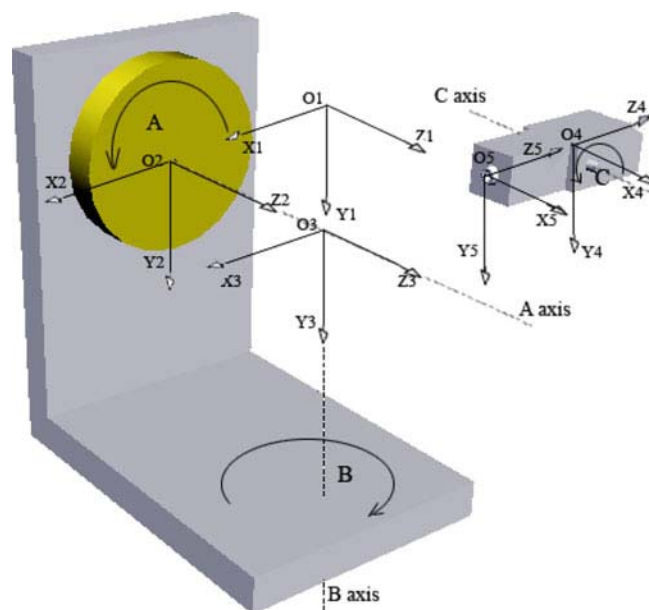
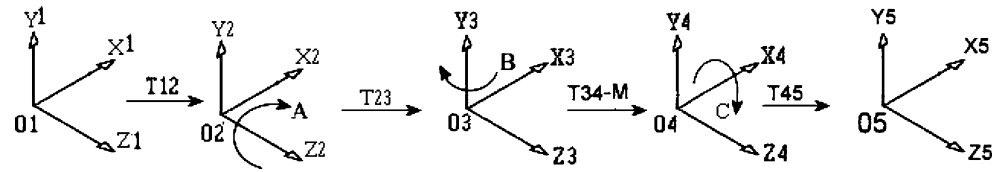


Fig. 25 Hypothetical six-axis machine with 2 axis on the table and one on the tool

Fig. 26 Intermediate reference coordinate systems for the 2-1 six-axis machine



$$P_3 = P_{2A} + T_{23} = R_Z[A](W + T_{12}) + T_{23}, \quad (13)$$

where T_{23} is the coordinate of O_2 in O_3 .

Step 4: Rotation around the B -axis in O_3

$$P_{3B} = R_Y[B]P_3 = R_Y[B](R_Z[A](W + T_{12}) + T_{23}), \quad (14)$$

where $R_Y[B] = \begin{bmatrix} \cos B & 0 & -\sin B \\ 0 & 1 & 0 \\ \sin B & 0 & \cos B \end{bmatrix}$ is the rotation matrix around the B -axis.

Step 5: Coordinate translation $O_3 \rightarrow O_4$ with the machine slide translation M .

$$P_4 = P_{3B} + T_{34} - M = R_Y[B](R_Z[A](W + T_{12}) + T_{23}) + T_{34} - M, \quad (15)$$

where T_{34} is the coordinate of O_3 in O_4 with respect to the machine zero point $M=(0, 0, 0)$.

Step 6: Rotation around the C -axis in O_4 by C

$$P_{4C} = R_X[C]P_4 = R_X[C](R_Y[B](R_Z[A](W + T_{12}) + T_{23}) + T_{34} - M), \quad (16)$$

$$\text{where } R_X[C] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos C & \sin C \\ 0 & -\sin C & \cos C \end{bmatrix}.$$

Step 7: Coordinate translation $O_4 \rightarrow O_5$ yields

$$P_5 = P_{4C} + T_{45} = R_X[C](R_Y[B](R_Z[A](W + T_{12}) + T_{23}) + T_{34} - M) + T_{45}, \quad (17)$$

where T_{45} is the coordinate of O_4 in O_5 .

Let $T_5 = (0, 0, -L)$ denote the coordinate of the tool tip in coordinate system O_5 where L is the tool length. Equating P_5 and T_5 yields

$$T_5 = P_5 = R_X[C](R_Y[B](R_Z[A](W + T_{12}) + T_{23}) + T_{34} - M) + T_{45}, \quad (18)$$

After a rearrangement

$$M = R_Y[B](R_Z[A](W + T_{12}) + T_{23} + T_{34} - R_X[C]^{-1}(T_5 - T_{45})), \quad (19)$$

$$W = R_Z[A]^{-1}(R_Y[B]^{-1}(M - R_X[C]^{-1}(T_{45} - T_5) - T_{34}) - T_{23}) - T_{12}. \quad (20)$$

Let T'_1 be the vector corresponding to $T'_5 = (0, 0, -L + 1)$ in O_1 so that

$$T'_1 = R_Z^{-1}[A](R_Y^{-1}[B](M - R_X^{-1}[C](T_{45} - T'_5) - T_{34}) - T_{23}) - T_{12}. \quad (21)$$

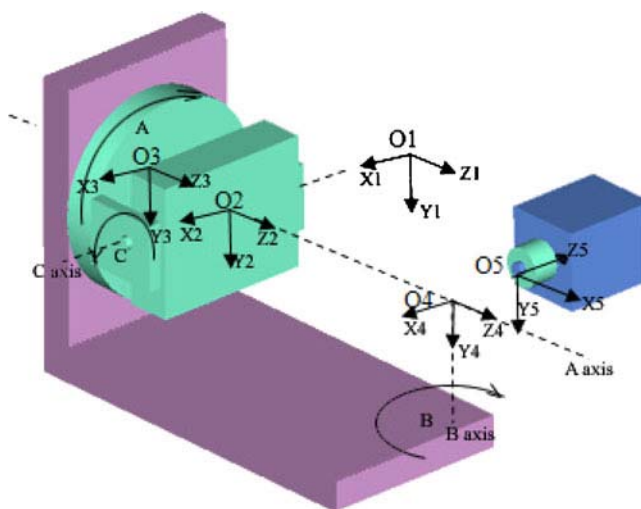
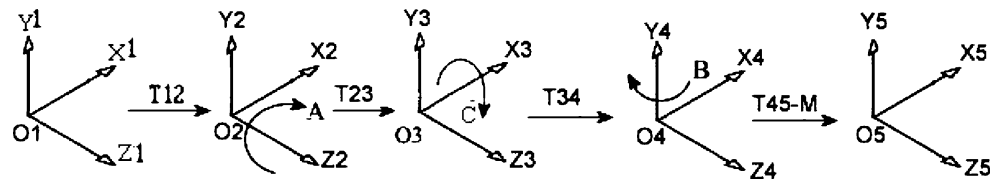


Fig. 27 3-0 six-axis machine with 3 axis on the table and none on the tool

Fig. 28 Intermediate reference coordinate systems for the 3-0 six-axis machine



The unit vector $T'_5 - T_5 = (0, 0, 1)$ in the machine coordinates represents the tool orientation $I = (i, j, k)$ in O_1 .

Clearly,

$$I = T'_1 - W = R_Z[A]^{-1} R_Y[B]^{-1} R_X[C](T'_5 - T_5). \quad (22)$$

After the matrix multiplications, we have

$$\begin{pmatrix} i \\ j \\ k \end{pmatrix} = \begin{pmatrix} \sin(A) \sin(C) + \cos(A) \sin(B) \cos(C) \\ -\cos(A) \sin(C) + \sin(A) \sin(B) \cos(C) \\ \cos(B) \cos(C) \end{pmatrix}. \quad (23)$$

Solving (23) with regard to A and B yields

$$A = \begin{cases} \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))}, & \text{if } (-j \tan(C) + i \sin(B)) > 0 \text{ and } (j \sin(B) + i \tan(C)) > 0, \\ \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))} + \pi, & \text{if } (-j \tan(C) + i \sin(B)) < 0, \\ \tan^{-1} \frac{(j \sin(B) + i \tan(C))}{(-j \tan(C) + i \sin(B))} + 2\pi, & \text{otherwise.} \end{cases} \quad (24)$$

$$B = \cos^{-1} \left(\frac{k}{\cos(C)} \right). \quad (25)$$

2. The 3-0 machine

The 3-0 machine configuration shown in Figs. 27 and 28 implies that the coordinates of the tool tip in O_5 are given by

$$T_5 = R_Y[B](R_X[C](R_Z[A](W + T_{12}) + T_{23}) + T_{34}) + T_{45} - M. \quad (26)$$

Clearly,

$$M = R_Y[B](R_X[C](R_Z[A](W + T_{12}) + T_{23}) + T_{34}) + T_{45} - T_5, \quad (27)$$

$$W = R_Z[A]^{-1}(R_X[C]^{-1}(R_Y[B]^{-1}(M - T_{45} + T_5) - T_{34}) - T_{23}) - T_{12}, \quad (28)$$

$$I = T'_1 - W = R_Z[A]^{-1} R_X[C]^{-1} R_Y[B]^{-1}(T'_5 - T_5) = \begin{pmatrix} \cos(A) \sin(B) + \sin(A) \sin(C) \cos(B) \\ \sin(A) \sin(B) - \cos(A) \sin(C) \cos(B) \\ \cos(C) \cos(B) \end{pmatrix}. \quad (29)$$

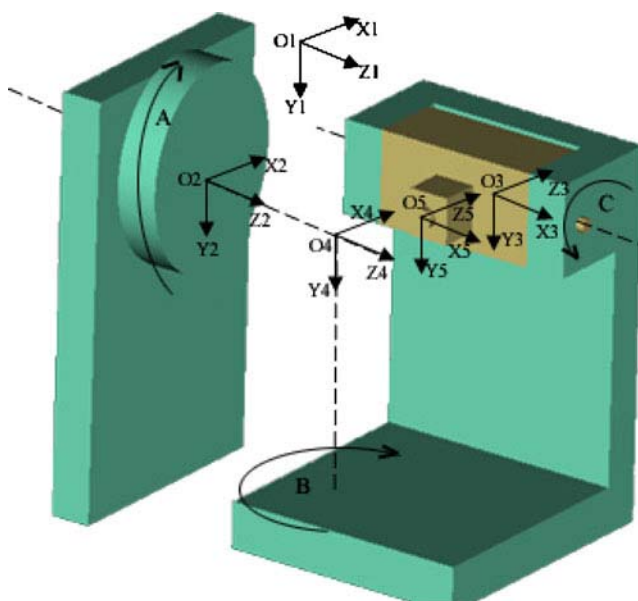
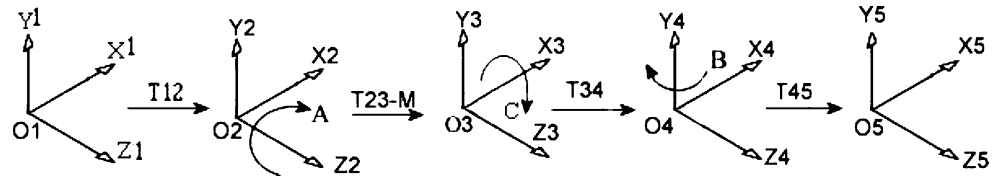


Fig. 29 1-2 six-axis machine with 1 axis on the table and 2 on the tool

Fig. 30 Intermediate reference coordinate systems for the 1-2 six-axis machine



Solving with regard to A and B yields

$$A = \begin{cases} \tan^{-1} \frac{(j \sin(B) + i \sin(C) \cos(B))}{(-j \sin(C) \cos(B) + i \sin(B))}, & \text{if } ((-j \sin(C) \cos(B) + i \sin(B)) > 0 \text{ and } (j \sin(B) + i \sin(C) \cos(B)) > 0, \\ \tan^{-1} \frac{(j \sin(B) + i \sin(C) \cos(B))}{(-j \sin(C) \cos(B) + i \sin(B))} + \pi, & \text{if } (-j \sin(C) \cos(B) + i \sin(B)) < 0, \\ \tan^{-1} \frac{(j \sin(B) + i \sin(C) \cos(B))}{(-j \sin(C) \cos(B) + i \sin(B))} + 2\pi & \text{otherwise.} \end{cases} \quad (30)$$

$$B = \cos^{-1} \left(\frac{k}{\cos(c)} \right).$$

$$(31) \quad W = R_z[A]^{-1} (M + R_x[C] (R_y[B] (T_5 - T_{45}) - T_{34})$$

$$- T_{23}) - T_{12}, \quad (34)$$

3. The 1-2 machine

Applying the same procedure to the 1-2 machine (Figs. 29 and 30) yields

$$T_5 = R_y[B]^{-1} (R_x[C]^{-1} (R_z[A] (W + T_{12}) + T_{23} - M) + T_{34}) + T_{45}. \quad (32)$$

Clearly,

$$M = R_z[A] (W + T_{12}) + T_{23} - R_x[C] (R_y[B] (T_5 - T_{45}) - T_{34}), \quad (33)$$

and

$$I = T_1' - W = R_z[A]^{-1} R_x[C] R_y[B] (T_5' - T_5) = \begin{pmatrix} -\cos(A) \sin(B) + \sin(A) \sin(C) \cos(B) \\ -\sin(A) \sin(B) - \cos(A) \sin(C) \cos(B) \\ \cos(C) \cos(B) \end{pmatrix}, \quad (35)$$

Fig. 31 0-3 six-axis machine with 0 axis on the table and 3 on the tool

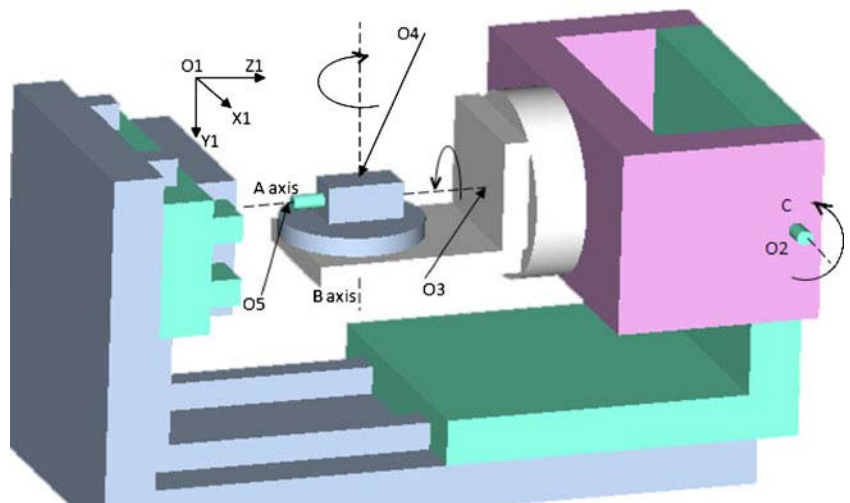
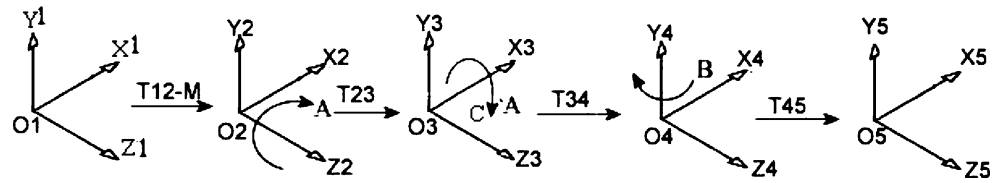


Fig. 32 Intermediate reference coordinate systems for the 1-2 six-axis machine



where

$$A = \begin{cases} -\tan^{-1}\left(\frac{i \sin(C) - j \tan(B)}{i \tan(B) + j \sin(C)}\right), & \text{if } (i \tan(B) + j \sin(C)) > 0 \text{ and } (i \sin(C) - j \tan(B)) > 0 \\ -\tan^{-1}\left(\frac{i \sin(C) - j \tan(B)}{i \tan(B) + j \sin(C)}\right) + \pi, & \text{if } (i \tan(B) + j \sin(C)) < 0, \\ -\tan^{-1}\left(\frac{i \sin(C) - j \tan(B)}{i \tan(B) + j \sin(C)}\right) + 2\pi, & \text{otherwise.} \end{cases} \quad (36)$$

$$B = \cos^{-1}\left(\frac{k}{\cos(C)}\right). \quad (37)$$

$$W = M - T_{12} + R_z[A](R_x[C](R_y[B](T_5 - T_{45}) - T_{34}) - T_{23}), \quad (40)$$

and

4. The 0–3 machine

The kinematics of the 0–3 machine is illustrated in Figs. 31 and 32. It is not hard to show that

$$T_5 = P_5 = R_y[B](R_x[C](R_z[A](W + T_{12} - M) + T_{23}) + T_{34}) + T_{45}, \quad (38)$$

$$I = R_z[A](R_x[C](R_y[B](T_5' - T_5))) = \begin{pmatrix} -\cos(A) \sin(B) - \sin(A) \sin(C) \cos(b) \\ \sin(A) \sin(B) - \cos(A) \sin(C) \cos(B) \\ \cos(C) \cos(B) \end{pmatrix}, \quad (41)$$

where

$$M = W + T_{12} - R_z[A](R_x[C](R_y[B](T_5 - T_{45}) - T_{34}) - T_{23}), \quad (39)$$

$$A = \begin{cases} \tan^{-1}\left(\frac{-j \sin(B) + i \sin(C) \cos(B)}{j \sin(C) \cos(b) + i \sin(B)}\right), & \text{if } (j \sin(C) \cos(b) + i \sin(B)) > 0 \text{ and } (-j \sin(B) + i \sin(C) \cos(B)) > 0 \\ \tan^{-1}\left(\frac{-j \sin(B) + i \sin(C) \cos(B)}{j \sin(C) \cos(b) + i \sin(B)}\right) + \pi, & \text{if } (j \sin(C) \cos(b) + i \sin(B)) < 0, \\ \tan^{-1}\left(\frac{-j \sin(B) + i \sin(C) \cos(B)}{j \sin(C) \cos(b) + i \sin(B)}\right) + 2\pi, & \text{otherwise.} \end{cases} \quad (42)$$

and

References

- Japitana FH, Morishige K, Takeuchi Y (2004) 6-Axis control cutting of overhanging curved grooves by means of non-rotational tool with application of ultrasonic vibrations. *Int J Mach Tools Manuf* 44:479–486

$$B = \cos^{-1}\left(\frac{k}{\cos(C)}\right). \quad (43)$$

2. Takeuchi Y, Yoneyama Y, Ishida T, Kawai T (2009) 6-Axis control ultra precision micro grooving on sculptured surfaces with non-rotational cutting tool. *CIRP Annals-Manufacturing Technology* 58:53–56
3. Hsieh J-F, Lin PD (2003) Production of multi-fluted drills on six-axis CNC tool-grinding machine. *Int J Mach Tools Manuf* 43:1117–1127
4. Munlin M, Makhnov SS, Bohez ELJ (2004) Optimization of rotations of a five-axis milling machine near stationary points. *Comput-Aided Des* 36:1117–1128
5. Makhnov SS, Munlin M (2005) Optimal sequencing of rotation angles for five-axis machining. *Int J Adv Manuf Technol* 35:41–54
6. Makhnov SS, Anotaipaboon W (2007) Advanced numerical methods to optimize cutting operations of five axis milling machines. Springer, New York
7. She C-H, Chang C-C (2007) Design of a generic five-axis postprocessor based on generalized kinematics model of machine tool. *Int J Mach Tools Manuf* 47:537–545
8. Bohez ELJ (2002) Five-axis milling machine tool kinematics chain design and analysis. *Int J Mach Tools Manuf* 42:505–520
9. Omrcen D, Zlajpah L, Nemec B (2007) Compensation of velocity and/or acceleration joint saturation redundant manipulator. *Rob Autom Syst* 55:337–344
10. Schmitz TL, Ziegert JC, Canning JS, Zapata R (2008) Case study: A comparison of error sources in high-speed milling. *Precis Eng* 32:126–133
11. Honegger CA, Tulsian A, Mukhopadhyay D (2010) Analysis of thermal errors in a high-speed micro-milling spindle. *Int J Mach Tools Manuf* 50:386–393
12. Vanherck P, Dehaes J, Nuttin M (1997) Compensation of thermal deformations in machine tools with neural nets. *Comput Ind* 33:119–125
13. Donmez M, Hahn M, Soons J (2007) A novel cooling system to reduce thermally-induced errors of machine tools. *CIRP Ann* 56(1):521–524
14. Haitao Z, Jianguo Y, Jinhua S (2007) Simulation of thermal behavior of a CNC machine tool spindle. *Int J Mach Tools Manuf* 47:1003–1010
15. Ramesh R, Mannan MA, Poo AN (2003) Thermal error measurement and modeling in machine tools. Part I. Influence of varying operating conditions. *Int J Mach Tools Manuf* 43(4):391–404
16. Ramesh R, Mannan MA, Poo AN, Keerthi SS (2003) Thermal error measurement and modeling in machine tools. Part II. Hybrid Bayesian Network-support vector machine model. *Int J Mach Tools Manuf* 3(4):405–419
17. Meng Lim EE, Meng C-H (1995) The prediction of dimensional error for sculptured surface productions using the ball-end milling process. Part 2. Surface generation model and experimental verification. *Int J Mach Tools Manuf* 35(8):1171–1185
18. Matsubara T, Tanaka H, Mizumoto H (1991) Study on accuracy in end milling operations. *Int J Jpn Soc Precis Eng* 25(4):291–296
19. Iwabe H, Fujii Y, Saito K, Kishinami T (1994) Study on corner cut by end mill. *Int J Jpn Soc Precis Eng* 28(8):218–223
20. Law KMY (1998) Process design for error compensation in end milling of pockets. M. Phil. Thesis, City University of Hong Kong
21. Law KMY, Geddam A, Ostafiev VA (1999) A process-design approach to error compensation in the end milling of pockets. *J Mater Process Tech* 89–90:238–244
22. Law KMY, Geddam A (2003) Error compensation in the end milling of pockets: a methodology. *J Mater Process Technol* 139:21–27
23. Weck M, Bibring H (1984) Metrological analysis and performance tests: handbook of machine tools, vol 4. Wiley, Chichester
24. Bohez ELJ, Ariyajunya B, Sinlapeecheewa C, Shein TMM, Lapd DT, Belforte G (2007) Systematic geometric rigid body error identification of 5-axis milling machines. *Comput-Aided Des* 39:229–244
25. Farouki RT, Sakkalis T, Vaserstein L (2009) Non-existence of rational arc-length parametrizations for curves in R^n . *J Comput Appl Math* 228(1):494–497
26. Makhnov SS (2010) Adaptable geometric patterns for five-axis machining: a survey. *Int J Adv Manuf Technol* 47(9–12):1167–1208
27. Vericut® V5.4 User Manual, URL: <http://www.cgtech.com>

Springer Series in Advanced Manufacturing

Stanislav S. Makhanov

Weerachai Anotaipaboon

Advanced Numerical Methods to

Optimize Cutting Operations of Five Axis Milling Machines

Makhanov
Anotaipaboon

Stanislav S. Makhanov
Weerachai Anotaipaboon

The book presents new optimization algorithms designed to improve the efficiency of tool paths for five-axis NC machining of sculptured surfaces. The book introduces the reader to fundamental issues involved in the tool path planning such as the kinematics of five-axis machines, types of 5 axis machines, part surface representation, machining strips, optimal tool orientation, gouging avoidance and forward step error. It also introduces new methods of optimization based on research conducted by the authors, including schemes performed in the spatial domain, angular domain as well as procedures to optimize the initial setup. The book can be used by undergraduate and graduate students and researchers in the field of NC machining and CAD/CAM as well as by the corporate research groups for advanced optimization of cutting operations.

ISBN 978-3-540-71120-9



9 783540 711209

> springer.com



Advanced Numerical Methods to Optimize
Cutting Operations of Five Axis Milling Machines

Advanced Numerical Methods to Optimize Cutting Operations of Five Axis Milling Machines

 Springer

Springer Series in
Advanced Manufacturing

Stanislav Makhanov

*Sirindhorn International Institute of Technology, Thammasat University,
Thailand*

Weerachai Anotaipaiboon

National Science and Technology Development Agency, Thailand

Advanced Numerical Methods to Optimize Cutting Operations of Five-Axis Milling Machines

January 4, 2007

Springer

To our families

Preface

Tool path generation and optimization for multi-axis milling machines is a new application area for computational mathematics requiring knowledge of numerical methods, differential and computational geometry, solid modeling, and optimization methods, as well as mechanical engineering and NC (Numerical Control) programming.

This book presents a computational framework for designing efficient computational algorithms for tool path optimization. We focus on five-axis machining, but the methods can also be applied to three- and four-axis machines.

The first three chapters of the book are tutorial, exposing readers to the basic knowledge necessary for five-axis cutting, such as G-code programming, differential geometry, and fundamental issues of tool path planning. These chapters can be used as an introduction to five-axis machining in the framework of an undergraduate or graduate course in computer aided manufacturing.

Advanced numerical methods introduced in the subsequent chapters include grid generation methods, space-filling curves, shortest path optimization, and the ramifications of each technique. This part of the book can be used for undergraduate and graduate research as well for corporate research and development. The authors have first-hand experience in designing, programming, and verifying five-axis optimization algorithms. Building on this experience, we present the numerical methods in detail and provide all of the basic computational formulas. Moreover, we illustrate and analyze the methods through numerous practical numerical examples, so that readers can understand, reproduce, and program any of them.

The book also includes a review of modern tool path optimization methods (130 references) and short historical notes on grid generation and space-filling curves.

We would like to thank Erik Bohez, Mud-Ameen Munlin, Bert Lauwers, and Than Lin for fruitful discussions and suggestions.

We also wish to acknowledge the National Science and Technology Agency, the National Electronic and Computer Technology Center, and the Thailand

VIII Preface

Research Fund, whose sponsorship made it possible to conduct many machining experiments and measurements.

Bangkok,
December 2006

Stanislav Makhanov, Ph.D.
Weerachai Anotaiapaiboon, Ph.D.

Contents

List of Figures	X
List of Tables	XV
1 Introduction	1
1.1 Motivation and Structure of the Book	1
1.2 CAD/CAM Formats	3
1.3 Short Literature Survey	4
References	15
2 Introduction to Five-Axis NC Machining	25
2.1 Five-Axis NC Machining Concepts	25
2.2 NC Part Programming	28
2.3 Classification of Five-Axis Machines	34
2.4 Five-Axis Machine Kinematics	37
2.5 Five-Axis Machining Example	43
References	48
3 Fundamental Issues in Tool Path Planning	51
3.1 Surface Representation	51
3.2 Machining Strip Width Estimation	53
3.3 Optimal Tool Orientation and Gouging Avoidance	60
3.4 Kinematics Error	63
3.5 Tool Path Generation	66
References	68
4 Space-Filling Curve Tool Paths	73
4.1 A Brief History of Space-Filling Curves and Their Applications	73
4.2 Tool Path Optimization	75
4.3 Tool Path Generation using Adaptive Space-filling Curves	77
4.3.1 Grid Construction	77

4.3.2	Space-Filling Curve Generation	78
4.3.3	Tool Path Correction	80
4.4	Examples and Discussion	83
	References	94
5	Tool Paths in Adaptive Curvilinear Coordinates	97
5.1	Introduction	97
5.2	A Historical Note on Grid Generation	98
5.3	Variational Grid Generation for Tool Path Optimization	101
5.3.1	Preliminary Examples	101
5.3.2	Variational Method and Functionals	102
5.3.3	The Harmonic Functional	109
5.3.4	Examples of the Tool Path Optimization	110
5.4	Application of Harmonic Functional to Tool Path Generation	116
5.5	Space-Filling Curve Generation on Block Structured Grid	124
5.6	Examples and Discussion	125
	Appendix: Derivation of Computational Formulas for Adaptive-Harmonic Grid Generation	133
	References	144
6	Optimization of Rotations	151
6.1	Introduction	151
6.2	Kinematics Error and Angle Variation	155
6.3	Optimization Problem	160
6.4	Optimization of Rotations: Examples and Practical Machining	162
6.5	Uniform Angular Grids	171
6.6	Uniform Angular Grids: Numerical and Machining Experiments	176
	Appendix: The APT cutter	180
	References	183
7	Theory of Optimal Setup for Five-Axis NC Machining	185
7.1	Introduction	185
7.2	Tool Trajectory Analysis	189
7.2.1	Invariant Parameters	189
7.2.2	Workpiece Setup and the Tool Trajectory	190
7.3	Least-Squares Optimization and Dependent Variables	192
7.3.1	Least-Squares Optimization	192
7.3.2	Dependent Variables	193
7.4	Examples and Discussion	194
7.4.1	Numerical Method	194
7.4.2	Examples	195
	References	198
	Index	205

List of Figures

2.1a	MAHO 600E milling machine	26
2.1b	Workpiece, clamping device, and rotary tables of MAHO 600E .	26
2.1c	HERMLE UWF902H milling machine.....	27
2.2	Zigzag and spiral tool paths	28
2.3	Kinematic chain diagram of machine in Fig. 2.1a	35
2.4	Example of 2-0 machine	36
2.5	Example of 1-1 machine	37
2.6	Example of 0-2 machine	38
2.7	Simple shape parametric surface.....	44
2.8	Zigzag tool path for surface (2.12) (Fig. 2.7)	44
2.9	Cutting simulation of surface (2.12) (Fig. 2.7) with 10×10 tool path	46
2.10	Cutting simulation of surface (2.12) (Fig. 2.7) with 20×20 tool path	46
2.11	Bezier surface	48
2.12	Tool trajectories with loops of 20×20 tool path for surface in Fig. 2.11	48
2.13	Cutting simulation of surface in Fig. 2.11 with 20×20 tool path	49
3.1	Examples of cutting tools	54
3.2	Geometric analysis of the cutting operations	55
3.3	Machining strip width estimation	55
3.4	Machining strip width estimation method of Lee and Ji	59
3.5	Tool gouging	61
3.6	Kinematics error between two cutter contact points	64
3.7	Angle adjustment for the 2-0 machine	65
3.8	Kinematics error reduction by angle adjustment	67
3.9	Overlapping of machining strips on adjacent tool paths.....	68
3.10	Example of isoparametric tool path	69
4.1	3 iterations of the Peano's space-filling curve	74

4.2	6 iterations of the Hilbert's space-filling curve	74
4.3	Grid construction for SFC tool path	78
4.4	Small circuits and dual graph construction.....	79
4.5	Illustration of SFC tool path construction	81
4.6	Machining strips on adjacent tool paths generated by using SFC	82
4.7	Modification of tool path trajectory at the turns	82
4.8	Trajectories of the cutter's effective cutting edge before and after applying the tool path correction	83
4.9	Overlaying of two isoparametric tool paths for the surface in Example 4.1	85
4.10	SFC tool path for the surface in Example 4.1	86
4.11	SFC tool paths in Example 4.1 generated with and without merging dependency	87
4.12	Practical machining using the SFC tool path without correction	87
4.13	Practical machining using the SFC tool path with correction ...	88
4.14	Overlaying of two isoparametric tool paths for the surface in Example 4.2	89
4.15	SFC tool path for the surface in Example 4.2	90
4.16	Overlaying of two isoparametric tool paths for the surface in Example 4.3	91
4.17	SFC tool path for the surface in Example 4.3	92
4.18	Simulation result of five-axis machining with SFC tool path in Unigraphics 18	93
5.1	Surface with a curvilinear zone of large gradients	101
5.2	Curvilinear grid which can be converted into a tool path to machine the surface in Fig. 5.1	102
5.3	A grid which can be converted into a tool path for a complex shaped region	103
5.4	Tool path as a mapping from the computational to the parametric (physical) domain	104
5.5	Constructing surface $T(\xi, \eta)$ from patches	105
5.6	The scallop height evaluation for convex surfaces	108
5.7	Tool path generation with adaptation to a weight function and the boundary	111
5.8	Grids obtained by constraint minimization and unconstrained minimization	112
5.9	A test Bezier surface.....	113
5.10	The machined Bezier surface.....	113
5.11	Conventional tool path and tool paths from grid generation....	114
5.12	Example of regular grid	117
5.13	Examples of block-structured grids	118
5.14a	Example of adaptive-harmonic grid	119
5.14b	Control function	120

5.15	Correspondence of node numbers for a mapping of the unit square in the (ξ, η) plane on to the quadrilateral cell 1 of the grid in the (u, v) plane	121
5.16	Partitioning of block-structured grid for refinement	124
5.17	Undirected graph construction for SFC tool path generation ...	126
5.18	Covering of grid in Fig. 5.17 by small circuits	127
5.19	Unimodal surface with exponential peak along a line	127
5.20	Curvilinear grid adapted to the unimodal surface with exponential peak along a line	128
5.21	Zigzag and SFC tool paths based on isoparametric and grid generation methods	129
5.22	Convergence rate of the grid generation technique in term of the number of iterations for Example 5.4	131
5.23	Grid refinement	132
5.24a	Curvilinear grid adapted to the surface in Example 5.6 in (u, v) domain	133
5.24b	Curvilinear grid adapted to the surface in Example 5.6 in workpiece coordinate system	134
5.25a	SFC tool path for the surface in Example 5.6 in (u, v) domain ..	135
5.25b	SFC tool path for the surface in Example 5.6 in workpiece coordinate system	136
5.26	Simulation result of five-axis machining with SFC tool path in Unigraphics 18 for the surface in Example 5.6	137
5.27	Convergence rate of the grid generation technique in term of the number of iterations for Example 5.6	138
5.28	Single blade of an impeller	139
5.29	Curvilinear grid adapted to part of a surface of the blade in (u, v) domain	140
5.30	Curvilinear grid adapted to part of a surface of the blade in workpiece coordinate system	141
5.31	SFC tool path for milling of the broken blade in (u, v) domain ..	142
5.32	SFC tool path for milling of the broken blade in workpiece coordinate system	143
5.33	Virtual cutting of the blade in Unigraphics 18	144
6.1	An experimental part surface S_1	152
6.2	Conventional tool path simulated by the virtual machine and surface machined by HERMLE UWF902H	153
6.3	Optimized tool path simulated by the virtual milling machine and optimized surface machined by HERMLE UWF902H	154
6.4	Nonlinearity of the tool path due to rotations in the workpiece coordinates	155
6.5	A loop-like trajectory induced by large gradients of the rotation angles	156
6.6	The “repaired” trajectory	157

6.7	The set of feasible rotations	162
6.8	A graph corresponding to the set of the feasible rotations	163
6.9	Trajectories corresponding to (1) b_{base} and (2) $\pi - b_{\text{base}}$ and the optimized trajectory composed from (1) and (2)	163
6.10	Trajectories corresponding to (1) a_{base} (2) $a_{\text{base}} - \pi$ (3) $a_{\text{base}} + \pi$ (4) $a_{\text{base}} - 2\pi$ and the optimized trajectory composed from the trajectories (1) (2) and (4)	164
6.11	Around or across the hill?	165
6.12	The kinematic error as a function of a_2 and b_2	166
6.13a	Conventional tool path for S_2 on HERMLE UWF902H showing large overcuts	171
6.13b	The tool path optimized with regard to the total error	172
6.13c	The tool path optimized with regard to the overcut error	173
6.14a	Conventional tool path for S_2 on MAHO 600E	173
6.14b	Optimization with regard to the total error	174
6.14c	Optimization with regard to the overcut error, S_2 on MAHO 600E	174
6.15a	Without optimization, S_2 on MAHO 600E (corresponds to Fig. 6.14a)	175
6.15b	Optimization with regard to the total error, S_2 on MAHO 600E (corresponds to Fig. 6.14b)	175
6.15c	Optimization with regard to the overcut error, S_2 on MAHO 600E (corresponds to Fig. 6.14c)	176
6.16a	Tool path and tool orientations, S_2 on MAHO 600E, before insertion	177
6.16b	Tool path and tool orientations, S_2 on MAHO 600E, conventional point insertion	177
6.16c	Tool path and tool orientations, S_2 on MAHO 600E, angular grid insertion	178
6.17a	Spatial grid for S_2 on MAHO 600E (corresponds to Fig. 6.16b) .	180
6.17b	Angular grid insertion for S_2 on MAHO 600E (corresponds to Fig. 6.16c)	181
6.18	Tool path for surface S_2 on MAHO 600E after the optimal sequencing	181
6.19	Tool path for surface S_2 on MAHO 600E after the optimal sequencing and inserting one point into the largest loop	182
6.20	APT cutter geometry	182
7.1	Changing an initial workpiece setup by rotating around the $\mathbf{z}_1$ -axis and the $\mathbf{y}_1$ -axis	186
7.2	Introductory example of optimal setup	188
7.3	Dependent parameters $T_{12,z} + T_{23,z} = c$ of the 2-0 machine . . .	194
7.4	Two test surfaces for optimal setup	196
7.5	Comparison of tool path trajectories in various optimal setups for sweep surface on the 2-0 machines	199

7.6	Comparison of tool path trajectories in various optimal setups for sweep surface on the 1-1 machines	200
7.7	Comparison of tool path trajectories in various optimal setups for sweep surface on the 0-2 machines	201
7.8	The sweep surface machined with different setups	202

List of Tables

2.1	List of word address codes	30
2.2	List of G-code functions	31
2.2	List of G-code functions (cont.)	32
2.3	List of M-code functions	33
2.4	List of CL points for tool path in Fig. 2.8	45
2.5	G-codes for tool path in Fig. 2.8	47
4.1	Performance of the SFC tool paths versus the isoparametric tool paths in terms of tool path length	84
4.2	Performance of the SFC tool paths versus the isoparametric tool paths in terms of estimated machining time	94
5.1	Convergence of the method λ_p versus θ	111
5.2	Accuracy of the machined surface	115
5.3	Convergence of the algorithm h versus ϵ	116
5.4	Comparison of tool paths based on variational grid generation techniques versus the isoparametric tool path scheme	130
6.1	Kinematics error for the optimized and non-optimized tool path, surface S_1 on MAHO 600E	167
6.2	Overcut error optimization, surface S_2 on HERMLE UWF902H	169
6.3	Overcut error optimization, surface S_2 on MAHO 600E	170
6.4	Error versus number of inserted points, the basic grid size 15×20178	
7.1	Minimal sets of optimization parameters	194
7.2	Performance of the optimization method, the sweep surface	197
7.3	Performance of the optimization method, the two-bell surface	197
7.4	Performance of the optimization measured by the required number of CC points	198

Introduction

1.1 Motivation and Structure of the Book

A variety of on-going research is focused on the development and analysis of methods to decrease the time required to progress from the computer modeling of the design surface to the machining while maintaining or improving the quality of the surface. One of the most important areas is tool path planning for numerical control (NC) machining. The main goal is obtaining the cutter location and orientation data that allow for an efficient surface milling within an allowed machining error.

Five-axis NC machines are becoming increasingly popular due to their ability to handle geometrically complex workpieces composed of raw material such as wood, wax, rubber, metal, stone, plastic, etc. Moreover, up-to-date five-axis NC machines are characterized by a high material removal rate and an efficient surface finish up.

Typically, manufacturing of the design surface by an NC machine comprises two stages, a rough cutting and a finish machining. During the rough cutting, the raw material is removed as fast as possible while ensuring no excessive cutting or gouging. During the finish machining, the tool is placed at the maximum contact with the surface to remove the remaining excess and create a well-finished and accurate surface. After finishing, the remaining scallops which are inevitably generated on the machined surface must be removed by manual surface grinding and polishing. The finish machining and manual polishing stages require as much as 75% of the total machining time. Besides, manual polishing is prone to error and undesirable irregularities.

Five-axis machining offers an improvement in efficiency of both the rough and finish machining stages over the three-axis counterpart. In five-axis machining, the tool orientation relative to the workpiece can be controlled by two additional degrees of freedom so as to achieve higher machining efficiency. With these advantages, a large number of tool path planning methods for five-axis machining has been developed and presented in the literature.

In chapter 1 we present the most popular CAD/CAM data formats and give a short literature survey on mathematical methods for optimization of five-axis machining. The survey has been focused on tool path interpolators, adaptable geometric patterns and methods for tool posture and gouging avoidance.

Chapter 2 exposes the readers to basic knowledge required to perform five-axis cutting. The G-code programming, examples of five-axis machining of simple shapes and verification of the cut using solid modeling software is presented and discussed. The chapter can be used as a short introduction into five-axis machining in the framework of an undergraduate course in Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM).

Chapter 3 introduces theories required to embrace the concepts of the tool path optimization for five-axis machining. The chapter presents such fundamental issues as kinematics of the five-axis machines, part surface representation, machining strip, tool orientation and gouging avoidance as well as the forward step error. A variety of configurations of the five-axis machines is also discussed and analyzed. This chapter can also be used at an undergraduate or a postgraduate level for CAD/CAM related studies.

Chapters 4 and 5 present advanced optimization schemes based on the adaptable geometric patterns, namely, the space-filling curves (SFC) and adaptive curvilinear grids. The SFC tool path has a number of attractive features such as the possibility to locally adapt the curve in such a way that the cutting device travels along the optimal direction. In addition, the entire surface is cut in one path eliminating the need of tool retractions. The use of the concept of curvilinear grids allows to simultaneously adapt the points on the tool paths to create more efficient zigzag, spiral or even SFC structures. The combination of the SFC and grid generation allows for tool paths on surfaces with complex irregular boundaries, cuts off, pockets, islands, etc.

Three-axis machines are often thought of as three dimensional plotters. However the five-axis machine is more like a big bore machine. That is why changing from three-axis to five-axis programming is not an easy task. In particular, the idea of optimizing rotations may seem totally foreign from the viewpoint of the three-axis machining. Therefore, chapter 6 presents the theory and practice of optimization of rotations for five-axis machining. Several optimization algorithms based on the shortest path techniques are presented and discussed.

Chapter 7 presents a theoretical background developed to construct numerical algorithms to minimize kinematics error introduced by the initial setup of five-axis milling machines. The initial setup consists of the position and orientation of the workpiece with respect to the mounting table and, optionally, the machine's initial configuration. Given a set of cutter contact points and tool orientations, a least-squares optimization procedure finds the optimal setup parameters.

1.2 CAD/CAM Formats

Sculptured or free-form surfaces are widely used in today's manufacturing industries for a variety of applications such as the production of dies, molds, aerospace and automotive parts, etc. The surfaces are usually characterized by complex geometries and variable curvatures. A single surface is usually composed of patches represented mathematically by parametric forms such as the Bezier surfaces, B-splines and NURBS. A design and manufacture of the sculptured surface parts is an expensive and time-consuming process. First, a design surface is transformed into a computer model (possibly with the help of CAD programs). The computer model is then used by the CAM programs to generate commands to move the cutting tool of the machine. The resulting set of tool positions and orientations constitutes a tool path to machine the desired surface.

Every CAD or CAD/CAM software uses an internal format to represent and control the required part. When the geometrical data is transferred from a CAD system to a CAD or CAM system, a neutral format for the data transfer is used. One of the most popular is the IGES (Initial Graphics Exchange Specification) format (see the history of the IGES format in [40]). The IGES format supports the use of surfaces defined by NURBS (Non Uniform Rational B-Splines) or derivatives of these representation. A good book for a beginner in NURBS is [31]. An advanced reader could use [98] and [24]. There are also several free libraries designed to control and manipulate the NURBS, such as NURBS++ package [1] and NURBS Toolbox [2]. Some free IGES-file processing tools are collected at the NIST/IGES web page [3].

The STL files, originally employed by the so-called layered manufacturing technologies such as the rapid prototyping, are now becoming more and more popular due to the simplicity of describing the part surfaces. As opposed to the complex description of surfaces employed by the IGES format, the STL format defines the surface as a collection of triangles each described by the coordinates of its three corners and a normal-vector. This technology provides an important platform for CAD/CAM applications due to the existence of many robust triangulation algorithms. Besides, the surface models are often composed of many patches. Therefore, by tessellating the patches and creating groups of triangles one can use many well established methods for treating intersections, trimming, shading, hidden surface removal and gouge protection [36]. The machining is usually performed by creating contours obtained by slicing the STL-surface [85, 115]. Of course, slicing of the NURBS surfaces is also possible, but it requires much more sophisticated techniques such as [81]. The contours are then saved using the SLC format (see [4] for instance).

Other popular CAD/CAM formats include STEP (Standard for the Exchange of Product Data), DXF (The Drawing Exchange Format from AutoDesk) and many others. As a matter of fact, the difference in data formats has created a large software industry specializing in transferring, adapting and

processing the CAD/CAM files. A table of compatibility of the CAD/CAM formats can be found, for instance, at [5] or [6].

1.3 Short Literature Survey

Optimization of tool paths for five-axis machining may include many features and multiple criteria such as the accuracy, the length of the tool path, the machining time, the size of the remaining scallops, etc. It may also include gouging avoidance, satisfying the machine axis limits, maximizing the volume of the removed material, reducing the tool wear. The optimization may also take into account the thermal characteristics of the cutting process, the tool bending, the vibrations and jacks, the workpiece positioning and many other parameters. The criteria could also include the configuration of the machine or specific parts of the machine as well as the design of the clamping device. Readers interested in citations before 1997 could use a fairly comprehensive survey by Dragomatz and Mann [25]. The survey presents a classification of research papers on three-axis and five-axis machining related to geometries of the tool paths and tool positioning. These categories include: 1) systems, 2) isoparametric paths, 3) non-isoparametric paths, 4) planar pocketing paths, 5) sculptured surface pocketing paths, 6) roughing paths, 7) tool positioning, 8) offset surface methods, 9) five-axis machining, 10) mesh models, 11) pixel and point models, 12) simulation and verification. Of course, the above groups overlap. Techniques involved in one group could be also involved in another group. For example, systems for tool path generation may include all of the above mentioned techniques. Roughing paths may be generated by the isoparametric or non-isoparametric schemes and so on.

Our survey is focused on five-axis machining. It also includes the three-axis methods but as long as they can be extended to the five-axis case. Besides, we confine ourselves by techniques designed for cutting the part surface by bottom-edge of the tool, e.g., flat-end milling and fillet milling. Many interesting methods designed for five-axis grinding (flank milling) and plunge milling have not been included.

The survey has been focused on the following categories: 1) *tool path interpolators*, 2) *adaptable geometric patterns* and 3) *methods for tool posture and gouging avoidance*. We believe that the above procedures are the most important part for efficient design of the numerical methods for five-axis machining.

Tool path interpolators

In the CNC machines, the tool motion is controlled by a sequence of reference points that are fed to the servo control system. The NC controllers employ linear interpolation techniques [66] or a circular interpolation which may result in discontinuities of the velocity at the junctions of the segments. They may

also result in high accelerations, subsequent surface inaccuracies and long machining time required to eliminate them. Furthermore, the modern high speed machining requires feedrates up to 40 m/min with accelerations up to 2 g. At such high speeds, small discontinuities in the reference tool path can result in undesirable high frequency harmonics in the reference trajectory, which may end up exciting the natural modes of the mechanical structure and the servo control system.

Although the NC program cannot change the way the controller moves the machine parts, the cutter location points and the rotation angles required to cut the prescribed curve can be changed in such a way that these errors are minimized or at least decreased. The task of generating such a sequence of points is called interpolation. Early interpolation schemes solved the problem of discontinuities by smoothing the tool path at the corners [17] and using low pass filters [126]. However, the problem is due to the difference between the chord and the arc lengths. That is why, generating the tool positions by incrementing the chord length leads to the feedrate instabilities. Therefore, if the fit curve is parameterized with respect to the arc length, these type of the inaccuracies will be eliminated. Therefore, many modern interpolation schemes are focused on finding a suitable polynomial interpolation (such as the B-splines) parameterized by means of the arc length.

Unfortunately, such parameterization is not analytically possible for general spline curves. Therefore, a number of approximate solutions were proposed. For instance, Wang and Yang [122] generate the trajectory by means of cubic and quintic splines using the chord length and a *nearly arc length parameterization*, Zhang and Greenway [133] implemented a similar B-spline based interpolation. Coordinate transformations between the workpiece and machine coordinate systems for five-axis milling were incorporated in the interpolator by Lo [77, 79] and Bohez et al. [14]. An extra jerk continuity condition has been included into the solution in [121].

Furthermore, the limitations of the machine tool drivers may cause failure in maintaining the commanded feedrate which in turn may lead to the tool chatter or breakage. Therefore, Weck et al. [125] have implemented cubic spline interpolation where adaptation of the feedrate was based on the physical limitations of the drives. The smooth transitions were obtained using fourth order acceleration profiles. Erkorkmaz and Altintas [30] presented a quintic spline trajectory generation algorithm that produced continuous position, velocity, and acceleration profiles. Smooth accelerations and decelerations were provided by imposing constraints on the first and second time derivatives of the feedrate.

In 1994, Farouki and Sakkalis [33] introduced the Pythagorean-Hodograph (PH) curves to solve the problem of feedrate control for three-axis machines. The curves provide a mathematically elegant solution to the above mentioned problems occurring in NC machining. In particular, the arc length was represented by a polynomial function of the curve parameter. In [32], a 2D Hermite interpolation combined with the PH was proposed and analyzed. The ideas

were further developed in [34, 35]. It was shown that since the arc length of the PH curves can be represented by a polynomial function of the curve parameter, they can be successfully used for the interpolation. Consequently, a variety of planar PH curves matching given Hermite type boundary data were developed (see, for instance, [61, 88, 119]).

Müller et al. [90] presented an algorithm for simultaneous five-axis spline interpolation which merges the PH interpolation and the analytic solution of the inverse kinematics problem using the template equation method. The result is a time-dependent spline which represents the given tool path with a high accuracy. Langeron et al. [69] suggested a polynomial B-spline interpolation which took into account the kinematics of the five-axis machine. The B-spline interpolation of the tool path in the part coordinate system includes the accuracy requirements and describes a five-axis tool path in a format adapted to the communication between the CAM software and the NC unit. The CAM output is directly expressed through the B-spline curves. Lo [80] introduced spline interpolators for *isoparametric*, *iso-scallop* and *iso-planar* machining methods (see the forthcoming section). Šír et al. [118] presented biarc interpolation techniques based on spline curves composed of circular arcs and compared them with the PH curves.

Finally, a number of papers introduce interpolators designed for high speed milling. During the high speed machining the actual average feedrate could be significantly lower than the programmed feedrate due to the physical restrictions of the machine tool and the block processing time of the CNC controller. In many cases the machine tool hardly reaches the maximum feedrates offered by the manufacturer. This happens when the block processing time is longer than the block execution time and the machine reaches the end point of the segment before information required for the next movement is available. In this case modern CNCs automatically reduce the programmed feedrate which results in a lower real feedrate and, consequently, a longer machining time. This relatively new issue has been discussed in [55, 64, 87, 109].

Adaptable geometric patterns

This section surveys research aimed to construct geometric patterns adaptable to a criteria which represents a certain estimate of the quality of the tool path such as the kinematics error, scallop heights, undercuts, overcuts, etc. It also includes methods for complex pocket milling since they often require special geometric patterns. Finally, the construction of the geometric patterns might or might not take into account the actual machine kinematics. In many cases, patterns employed for three-axis machining are also applicable, with certain modifications, to the five-axis machining.

The simplest tool path planning algorithms employ structured zigzag or spiral patterns due to their simplicity and the ease of computation (see [25, 108]). The zigzag and spiral motions employ uniform steps along a coordinate which parametrizes the desired curve extracted from the part surface.

The early adaptable methods replace the uniform spacing in favor of distributing the cutter location points by analogy with interpolation characterized by a variable step. First, the trajectories were assumed to be linear. Next, the desired curve was approximated using a certain technique, for example, employing arcs. Next, the forward step was selected by considering the deviation between the approximation and the straight line (see, for instance, [73]). The choice of the forward step can be performed by bisection or another inexpensive method. Some of the recent developments of these ideas are presented in [22, 73].

As mentioned before the most popular geometric solutions are the *zigzag* and the *spiral isoparametric* patterns constructed for single-patch or multi-patch parametric surfaces $S(u, v)$. In this context, the term *isoparametric* means that the zigzag tool path is generated in the parametric space $u - v$ along one of the coordinates, say, u . The v coordinate is then used to generate the forward steps.

Another approach is the *contour based* or *iso-planar* machining. In this case the cutter path follows intersection curves of the parametric surface and a series of vertical planes. One of the first papers reporting such techniques is [19], see also [104].

The both methods calculate the maximum allowable distance between the consecutive tracks using a scallop height limitation. However, if the maximum allowable distance is calculated globally, that is, the minimal allowable distance is taken from all the maximum allowable (pointwise) distances, then the method does not produce a constant scallop height. As the result, the machining efficiency is limited. Methods to maintain the constant scallop height called the *iso-scallop* machining methods were first proposed in [76, 112].

Lo [78] developed these approach and adapted it to five-axis machining. His algorithm starts with an initial curve in the parametric domain and calculates offset curves so that the scallop height remains approximately constant. The algorithm is designed for flat-end cutter and includes adaptive inclination which maximizes the machining strip. The algorithm also includes a *local gouging avoidance*. The local gouging refers to the removal of an excess material in the vicinity of the cutter contact point (CC point) due to the mismatch in curvatures between the tool as it is carried along the tool path and the desired surface (see the forthcoming section for details). Rao and Sarma [103] introduced similar *local gouging* avoidance algorithms applicable to surfaces characterized by low curvature and cut by the flat-end cutter. Finally, Lo [80] presents an iso-scallop tool path for *ball nose* cutters.

Evaluation of the machining strip versus the inclination and the direction could lead to complicated tool path topologies. For each tool position on the surface there exists at least one direction which maximizes the machining strip. The corresponding set of vectors mapped onto the parametric space (u, v) constitutes a 2D vector field which could be further analyzed. A continuous tool path which visits every point and follows the optimal direction at every point constitutes the optimal tool path which will maximize the machining