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.

0378-4754/$36.00 © 2009 IMACS. Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.matcom.2009.01.019
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.
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.** 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.
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.
2. Grid generation method

Surfaces with sharp variations in curvature.
many hours of computing. The use of such methods is well justified only for regional milling for complex shaped
are complemented by the real machining as well as by the test simulations on the Unigraphics 18. Finally, although
algorithms allow constructing the required tool path with the length close to the minimal. The numerical experiments
independently. A variety of examples is presented when the conventional methods are inefficient whereas the proposed
besides, the adaptive grid allows to efficiently treat complex spatial variability of the constraints in such a way that the
(SFCS) 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 \( (u_{i,j}, v_{i,j}), 0 \leq i \leq N_x, 0 \leq j \leq N_y \) is a discrete analogy of a mapping from the computational region \( \{0 \leq \xi \leq N_x, 0 \leq \eta \leq N_y\} \) 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:

\[
\min I \equiv \min_{u,v} \int \frac{(u_x^2 + u_y^2)(1 + f_u^2) + (v_x^2 + v_y^2)(1 + f_v^2) + 2 f_u f_v (u_x v_y + u_y v_x)}{(u_x v_y - u_y v_x) \sqrt{1 + f_u^2 + f_v^2}} \, d\xi \, d\eta, \tag{1}
\]

where subscripts \( u, v, \xi, \eta \) 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 \( f_u = f_v = 0 \), then the harmonic functional becomes the Winslow functional, however, it is important that \( I \) adapts the grid to the \( \text{gradients of } f \) 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 \textit{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
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 $(f_u)_n^0 = (f_v)_n^0 = 0$. 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 $\lambda_+$ and $\lambda_-$ 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:

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

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

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

\[W(n, m) = |S((u_n, v_n) - S((u_m, v_m))|.\]

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:

\[
n_{r,\text{new}} = n_r + n_{r,\text{add}}, \\
\max_{1 \leq i \leq n_r} \sum_{j=1}^{n_c-1} (W(n_{i,j}, n_{i,j+1}) - T(n_{i,j}, n_{i,j+1})) = \frac{n_{r,\text{add}}}{2r}, \\
n_{c,\text{new}} = n_c + n_{c,\text{add}}, \\
\max_{1 \leq j \leq n_c} \sum_{i=1}^{n_r-1} (W(n_{i,j}, n_{i,j+1}) - T(n_{i,j}, n_{i,j+1})) = \frac{n_{r,\text{add}}}{2r},
\]

(2)

(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 both directions. Finally, (2) and (3) may overestimate the number of the required tracks. Consequently, it can be replaced by

\[
n_{\text{new}} = n + n_{\text{add}}\alpha_{\text{rel}},
\]

where \( \alpha_{\text{rel}} < 1 \) 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 \( \alpha_{\text{rel}} = 1/n_{\text{add}} \) which, however, may lead to an increase in the computational cost.

2.4. Machining strip evaluation

Given the maximum allowable scallop height \( h_{\text{max}} \), 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 \( \lambda \) about the \( y_l \) axis, then by a tilt angle \( \omega \) 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 \( h = h_{\text{max}} \). 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 \( \theta \) 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 \( \theta_l \) and \( \theta_r \) 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

\[w = w_l + w_r.\]
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, \( \lambda \) and \( \omega \) 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 \( \lambda \) is set to a small default angle or zero and the tilt angle \( \omega \) is set to zero as well. If the surface is non-convex, a non-zero tool inclination angle \( \lambda \) 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, \( O_1 G \), forms a chord on the circle. Denote the angle between this line and \( O_1 O_2 \) by \( \phi \) (see Fig. 8(a and b)). Let \( \lambda_\phi \) be the tool inclination angle that corresponds to a specific \( \phi \). The minimum tool inclination angle to avoid gouging is then

\[
\lambda_{\min} = \max_{-\pi/2 \leq \phi \leq \pi/2} \lambda_\phi.
\]
It is not hard to demonstrate that for a non-convex surface

\[ \lambda_{\text{min}} = \sin^{-1}(rk_{\text{max}}), \]

where \( k_{\text{max}} \) is the maximum surface curvature at the CC point. Clearly, for a convex surface an inclination is not required, so \( \lambda_{\text{min}} = 0 \), however, from technological viewpoints a small inclination angle is still recommended. Furthermore, for \( \forall \lambda < \lambda_{\text{min}} \) the gouging will be avoided as well. It also can be shown [23] that for the flat end (cylindrical) cutter the orientation \( (\lambda, \omega) = (\lambda_{\text{min}}, 0) \) maximizes the machining strip. If gouging can not be eliminated by inclining the tool alone or the inclination angle \( \lambda \) 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 than 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
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, inspite 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.
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 $\mathbb{R}^3$. 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

\[
\text{Cost}(A, B) = |s| + |t| - |e| - |f|,
\]

where \(|e|\) denotes the distance between two vertices connected by edge \(e\). The cost of merging 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)

\[
\begin{align*}
  x &= 100u - 50, \\
  y &= 100v - 50, \\
  z &= 10 \exp(-40(2u - 0.5 - v)^2 - 15).
\end{align*}
\]

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>
<thead>
<tr>
<th>Method used for constructing the basic grid</th>
<th>Total path length (mm)</th>
<th>Vertical zigzag</th>
<th>SFC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iso-parametric tool path</td>
<td>4716.91</td>
<td>3066.87</td>
<td></td>
</tr>
<tr>
<td>Curvilinear grid generation</td>
<td>2557.74</td>
<td>2508.25</td>
<td></td>
</tr>
</tbody>
</table>
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


