Int J Adv Manuf Technol (2011) 52:913–927
DOI 10.1007/s00170-010-2799-8

ORIGINAL ARTICLE

Vector model-based workpiece update in multi-axis milling by moving surface of revolution

Eyyup Aras · Hsi-Yung Feng

Received: 22 October 2009 / Accepted: 17 June 2010 / Published online: 11 July 2010

Abstract This paper presents a parametric approach to updating workpiece surfaces in a virtual environment. The workpiece surfaces are represented by a series of discrete vectors, which may be orientated in different directions. The methodology is developed for multi-axis machining in which a tool can be arbitrarily oriented in space. The cutter is modeled as a surface of revolution, which is a canal surface formed by sweeping a sphere with varying radius along a spine curve. To define the tool swept envelope, the cutter surfaces are decomposed into a set of characteristic circles which are generated by a two-parameter family of spheres. Then, the grazing points, at which the discrete vectors can intersect the tool envelope, are obtained by considering the relationships between these circles and feed vector of the cutter. From this, the envelope-vector intersections are transformed into a single-variable function. Examples of this technique are generated for typical milling tools with both linear and circular spine curves. The vector/tool envelope intersection calculations for cutters with linear spine curves can be performed analytically. However, the intersection calculations for cutters having circular spine curves require solving a system of nonlinear equations. For this purpose, a root-finding analysis is developed for guaranteeing the root(s) in the given interval. Finally, to improve the efficiency when updating the workpiece, a vector localization scheme is developed based on the Axis-aligned Bounding Box method.

Keywords Workpiece update · NC machining · Canal surface · Two-parameter family of spheres · Milling cutters · Localizing tool movements

E. Aras (✉)
Department of Mechanical Engineering, King Saud University,
P.O. Box 800, Riyadh 11421, Kingdom of Saudi Arabia
e-mail: eyyaras@hotmail.com

H.-Y. Feng
Department of Mechanical Engineering,
The University of British Columbia,
Vancouver, BC, Canada V6T 1Z4

1 Introduction

One of the techniques for advancing the productivity and quality of machining processes is to design, validate, and produce the parts in a virtual environment. Virtual machining is used for simulation of the machining process prior to actual machining, thereby avoiding costly test trials on the shop floor. The basic idea in NC simulation and verification is to remove the swept volume generated as a result of the tool motion from the in-process workpiece and thus to obtain the final machined surface. Several choices for modeling the in-process work piece exist. The two most common are mathematically accurate solid modeler-based methodologies that are used in CAD modeling and approximate modeler-based methodologies such as those used in computer graphics.

Solid modeling offers the best choice for highly accurate modeling of the geometric conditions. Currently, the most popular schemes used in solid modelers are the Boundary representation and Constructive Solid Geometry. Research into solid modeler-based methodologies has been reported [1–6]. In these approaches, in-process workpiece updating is performed using geometric and topological algorithms within the solid modeler kernel. Notwithstanding improvements in the efficiency and robustness of these algorithms in today's solid modeling kernels, the sheer number of such intersections that must be performed in simulating machin-

ing on a real part still makes this a difficult technology to deploy practically. Other limitations come from the size of the data structure that is necessary in particular for capturing relationships between topology. These relationships are preserved when using a solid modeler for small surface artifacts such as cusps (see Fig. 1) that are generated during machining [7]. This results in a data structure that is large and grows as the simulation progresses particularly when ball-end and fillet-end cutters are being used.

To increase the efficiency and robustness in cutting simulations, a number of approximate approaches have been developed. They can be classified into two main groups: the image-space and object-space approaches [8]. In the image-space approaches [9–14], a vector normal to the plane of the screen is drawn at each pixel. Then, using a scan-line algorithm, the intersections between these vectors and the tool swept envelopes are calculated. For each pixel, an extended z-buffer is maintained that stores the z-depths of the vectors' entries and exits to the workpiece. Then by performing the Boolean difference operations with the z-buffer of the tool swept volume, the workpiece z-buffer is modified. The limitations with the image space methods are [8]: they are view dependant and errors, not visible in the chosen viewing direction, are undetected. Also, small machining errors (e.g. less than 0.1 mm) are unlikely to be detected by a visual inspection of the computer graphics image.

Research into object-based in-process workpiece update methodologies has been reported [8, 15–21]. In these methodologies, the workpiece is broken into a set of evenly distributed parallel lines, called z-vectors, with the length of each vector representing the depth of the workpiece. During the cutting simulation, z-vectors are updated by the tool swept volume (see Fig. 2). These methodologies use similar calculation techniques from Chappel's [9] and Oliver's [10] approaches.

The object-based schemes are mainly for fast cutting simulation and verification [18]. Some of the advantageous of these methodologies are [8]: (a) The accuracy of the simulation can be adjusted by user, (b) by comparing the discrete vector lengths with the user-defined tolerances, the verification can be easily accomplished, and (c) simulation of the material removal can be efficiently accomplished by calculating the intersections between discrete vectors and tool movement envelopes. Choi and Jerard [15] described a three-axis machining methodology based on a discrete approximation of the object into a set of surface points. Vectors passed through these points are intersected with the tool movements. Later, in the extended z-buffer model [16] a three-axis approximation of the five-axis tool movement for the ball-end mill is used. Chung et al. [17] showed a three-axis milling procedure for representing the cutter swept surface of a generalized cutter in a single-valued form and later Baek et al. [18, 19] in three-axis milling expressed the intersections between z-vectors and tool swept envelopes as the solution of a system of nonlinear equations.

This paper presents a parametric approach of updating the in-process workpiece represented by discrete vectors, where all directions are not necessarily identical. In this approach, a cutter can have an arbitrary orientation in space. The cutter, modeled as a canal surface, is decomposed into a set of characteristic circles which are generated by a two-parameter family of spheres. Therefore, the presented methodology is more general than previous publications as it will handle a broader variety of tool shapes. Considering the relationships between the characteristic circles and cutter feed vector, the grazing points at which the discrete vectors can intersect the tool envelope are obtained. Then, the envelope-vector intersections are transformed into a single-variable function. Finally, for localizing the calculations by eliminating from consideration the vectors which have no possibility of intersecting the tool envelope, a methodology based on the Axis-aligned Bounding Box (AABB) is presented.

The rest of the paper is organized as follows: a closed-form solution for finding the locations of the grazing points generated by a surface of revolution is presented in Section 2. Later, in Section 3, the workpiece update methodology is presented. A vector localization technique in multi-axis machining and the examples from implementation of the methodology are shown in Section 4, followed by the discussion and conclusions in Section 5.

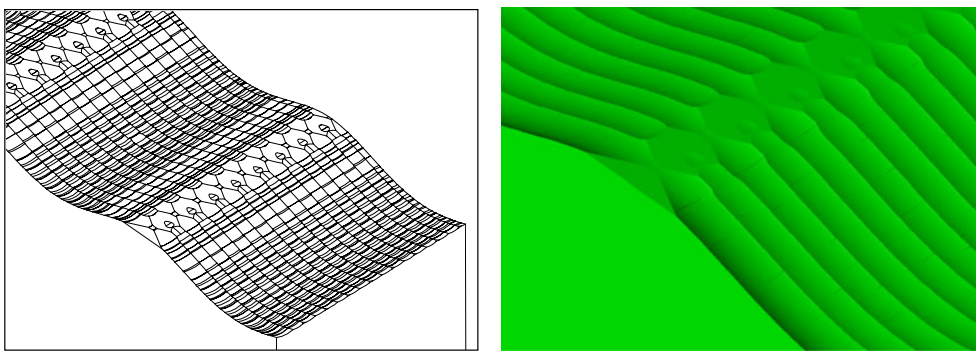

Fig. 1 Final machined surface with cusps

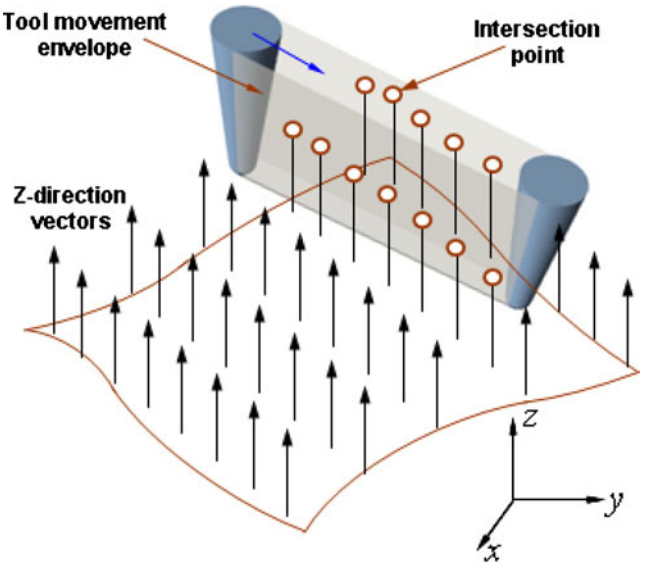

Fig. 2 Updating z-direction vectors with tool movement envelope

2 Generating tool swept envelopes

The major tool motions in milling operations are three-axis and five-axis tool motions. For some workpiece geometries, three-axis motions are not suitable for updating the surfaces. For example, as illustrated in Fig. 3a, if the part is deep three-axis motion is not sufficient enough for finishing. Also, milling a hard material with long cutters generates a bad surface finish and requires long machining time. On the other hand, exact five-axis tool motions are not preferable in simulations because of the extensive use of nonlinear root finding algorithms [16, 22]. Therefore, in this paper, the in-process workpiece update methodology is developed for up to 3½-axis tool motions in which the machine moves only its three linear axes while the two rotary axes are locked resulting in a fixed arbitrary cutter orientation in space [23] (see Fig. 3b). The authors believe that the presented in-process workpiece update methodology can be used to approximate five-axis tool motions and further information for these approximations can be found in [16].

In multi-axis milling tool motions can be represented by the cutter location point $\mathbf{F}(u)$ and the instantaneous orientation of tool rotation axis $\mathbf{A}(u)$, $u\in[0,1]$. A complex tool path can be approximated as the combination of linear segments, where each segment can be expressed by

$$\mathbf{F}(u) = P_S + (P_E - P_S)\, u \tag{1}$$

where, P_S and P_E are the start and end positions of a linear segment respectively. Two sets of coordinate systems are used in this paper: a workpiece coordinate system (WCS) which is coincident with the Cartesian coordinate system denoted by the unit vectors **i**, **j** and **k**, and a Tool Coordinate System (TCS) denoted by a set of mutually orthogonal unit vectors **d**, **e**, and **n**. Up to 3½-axis machining using $\mathbf{F}(u)$ and $\mathbf{A}(u)$, one can define TCS on the tool as follows,

$$\begin{aligned} &\mathbf{n} = \mathbf{A}/|\mathbf{A}|, \\ &\mathbf{e} = (\mathbf{A}\times\mathbf{f})/|\mathbf{A}\times\mathbf{f}|, \quad \text{if } \mathbf{A}\times\mathbf{f}\neq 0 \\ &\mathbf{d} = \mathbf{e}\times\mathbf{n} \end{aligned} \tag{2}$$

Where $\mathbf{f}=(f_x\ f_y\ f_z)$, the derivative of $\mathbf{F}(u)$ with respect to u, represents the instantaneous feed vector in WCS. According to Eq. (2) it is not allowed for the cutter to plunge along its rotational axis and also it is assumed that **d** axis of TCS is aligned with the instantaneous feed vector.

2.1 Representing the moving surface of revolution in parametric form

A surface of revolution can be parameterized by using the properties of canal surfaces. A canal surface is defined as the envelope of a one-parameter family of spheres $\sum(t)$, centered at a spine curve $\mathbf{m}(t)$[24]. The radius of the spheres is given by the function $r(t)$, $t\in\mathrm{R}$. A surface of revolution Φ, considered as a canal surface, can be

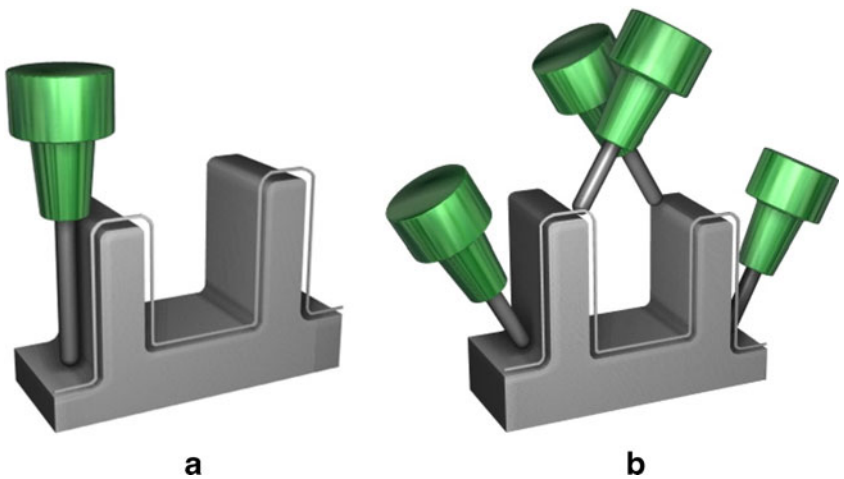

Fig. 3 Machining a deep workpiece with **a** 3-axis and **b** 3½-axis tool motions

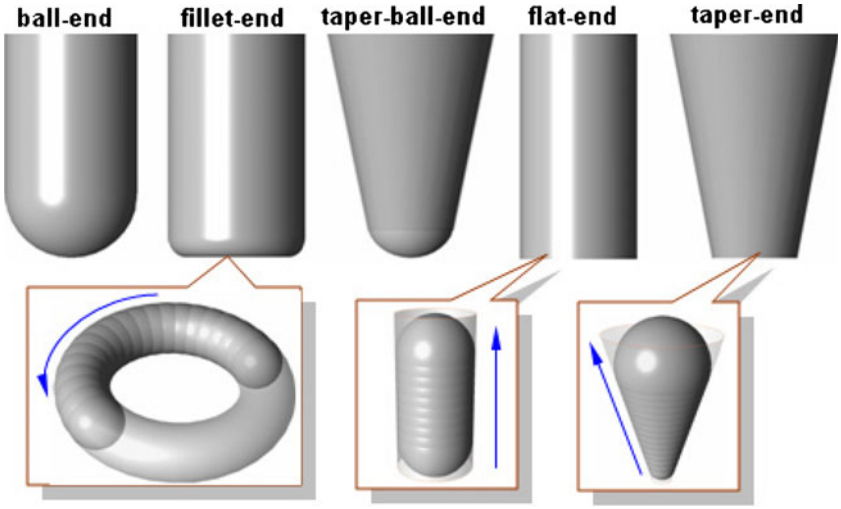

Fig. 4 Typical milling cutters as canal surfaces

constructed by using the one-parameter family of spheres, where each member is called a moving sphere. As shown in Fig. 4, the major NC cutter geometries can be constructed by using the moving spheres.

During NC simulation, a moving sphere within Φ follows two trajectories: the spine curve and tool path parameterized by t and u, respectively. Therefore in NC machining the moving spheres can be expressed by [25] (refer to Fig. 5)

$$\sum(t,u) : (P - \mathbf{m}(t,u))^2 - r(t)^2 = 0 \tag{3}$$

where, $\sum(t,u)$ represents the *two-parameter family of spheres*. Although in Eq. (3) the center of a sphere $\mathbf{m}(t,u)$ is a function of two parameters, the radius $r(t)$ is described by one parameter. This is because the radius of Φ only changes along the spine curve at a given cutter location. Taking the partial derivative of Eq. (3) with respect to t yields

$$\sum_t(t,u) : (P - \mathbf{m}(t,u)) \cdot \mathbf{m}_t(t,u) + r(t)\, r_t(t) = 0 \tag{4}$$

Where functions with the subscript t represent the corresponding partial derivatives and $\sum_t(t,u)$ is a plane with normal $\mathbf{m}_t(t,u)$. A characteristic circle K, embedded in $\sum(t, u)$, is a solution of the system of Eqs. (3) and (4) and it lies on $\sum_t(t,u)$ (see Fig. 5).

A member from the *two-parameter family of spheres* touches Φ at a characteristic circle. Therefore, Φ can be parameterized by using the characteristic circles described by their centers and radii. The center of the characteristic circle in Fig. 5 can be expressed by

$$\mathbf{C}(t,u) = \mathbf{m}(t,u) - r(t)\, \cos\varphi(t)\, \frac{\mathbf{m}_t(t,u)}{\| \mathbf{m}_t(t,u) \|} \tag{5}$$

where, $\varphi(t)$ satisfies the following dot product relationship (see Fig. 5)

$$(P - \mathbf{m}(t,u)) \cdot \mathbf{m}_t(t,u) = \| P - \mathbf{m}(t,u) \| \, \| \mathbf{m}_t(t,u) \| \, \cos(\pi - \varphi(t)) \tag{6}$$

Using Eqs. (3, 4, 6) $\cos\varphi(t)$ can be obtained as follows

$$\cos\varphi(t) = \frac{(P - \mathbf{m}(t,u)) \cdot \mathbf{m}_t(t,u)}{\| P - \mathbf{m}(t,u) \| \, \| \mathbf{m}_t(t,u) \|} = \frac{r_t(t)}{\| \mathbf{m}_t(t,u) \|} \tag{7}$$

Substituting $\cos\varphi(t)$ into Eq. (5) yields

$$\mathbf{C}(t,u) = \mathbf{m}(t,u) - r(t)\, r_t(t)\, \frac{\mathbf{m}_t(t,u)}{\| \mathbf{m}_t(t,u) \|^2} \tag{8}$$

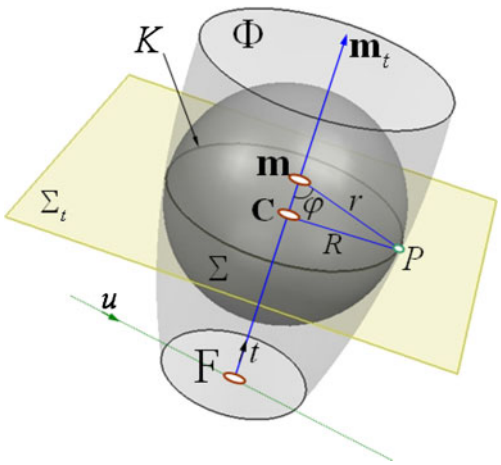

Fig. 5 Geometric description of a moving surface of revolution

Also from Fig. 5, the radius of the characteristic circle can be obtained as

$$R(t,u) = r(t)\ \sin\varphi(t) = r(t)\ \frac{\sqrt{\| \mathbf{m}_t(t,u)\|^2 - r_t(t)^2}}{\| \mathbf{m}_t(t,u)\|} \quad (9)$$

where, $\sin\varphi(t)$ is derived from Eq. (7). Using Eqs. (8) and (9), a moving surface of revolution can be expressed as

$$K(t,u,\theta) = \mathbf{C}(t,u) + R(t,u) \times (\cos\theta\ \mathbf{b}_1(t,u) + \sin\theta\ \mathbf{b}_2(t,u)) \quad (10)$$

where, $\theta\in[0,2\pi]$ and

$$\mathbf{b}_1(t,u) = \frac{\mathbf{m}_t(t,u)\times\mathbf{m}_{tt}(t,u)}{\|\mathbf{m}_t(t,u)\times\mathbf{m}_{tt}(t,u)\|}\ \mathbf{b}_2(t,u) = \frac{\mathbf{b}_1(t,u)\times\mathbf{m}_t(t,u)}{\|\mathbf{b}_1(t,u)\times\mathbf{m}_t(t,u)\|} \quad (11)$$

Two orthogonal unit vectors $\mathbf{b}_1(t,u)$ and $\mathbf{b}_2(t,u)$, located on the plane $\sum_t(t,u)$, determine the orientation of the characteristic circle. For the Eq. (11), it is assumed that the spine curve is biregular, i.e., $\mathbf{m}_t(t,u) \times \mathbf{m}_{tt}(t,u) \neq 0$. In NC milling, a member from the two-parameter family of spheres can follow either linear or circular spine curve. For example, conical and cylindrical surfaces have linear spine curves and the toroidal surface has circular spine curve (refer to Fig. 4). For the circular spine curve, $\mathbf{b}_1(t,u)$ and $\mathbf{b}_2(t,u)$ represent respectively the normal $\mathbf{M}(t,u)$ and binormal $\mathbf{B}(t,u)$ unit vectors of the Frenet frame formed by

$$\begin{aligned}\mathbf{T}(t,u) &= \mathbf{m}_t(t,u)/\|\mathbf{m}_t(t,u)\| \\ \mathbf{M}(t,u) &= \mathbf{T}_t(t,u)/\|\mathbf{T}_t(t,u)\| \\ \mathbf{B}(t,u) &= \mathbf{T}(t,u)\times\mathbf{M}(t,u)\end{aligned} \quad (12)$$

where, $\mathbf{T}(t,u)$ is the unit tangent vector along the spine curve $\mathbf{m}(t,u)$. On the other hand because the curvature vanishes in the linear spine curve, the Frenet frame components are not applicable. For this case, $\mathbf{b}_1(t,u)$ and $\mathbf{b}_2(t,u)$ represent respectively **d** and **e** components of TCS defined in Eq. (2).

2.2 Locating grazing points by characteristic circles

When NC simulation is performed on a workpiece represented by a collection of discrete vectors, it is necessary to compute the intersections between the discrete vectors and the cutter swept volume. In NC machining, a swept volume can be considered as a set of points on the moving cutter that lie on the machined surface. However, there is no closed form expression for the swept volume of a moving cutter [26]. One way around this problem is to use the swept envelope for defining the swept volume. To determine the swept envelope of a moving cutter, a tangency function is defined as follows [27]

$$f(t,u,\theta) = \mathbf{N}(t,u,\theta)\ \cdot\ \mathbf{V}(t,u,\theta) \quad (13)$$

where, $\mathbf{N}(t,u,\theta)$ and $\mathbf{V}(t,u,\theta)$ are the normal and velocity vectors at a point located on the tool surface respectively. By using the tangency function, the swept envelope of a moving surface of revolution can be determined as [28]

$$\begin{aligned}\mathrm{SE}(K(t,u,\theta)) = {} & K^-(t,u=0,\theta) \\ & \cup\ K^0(t,0\le u\le 1,\theta) \\ & \cup\ K^+(t,u=1,\theta)\end{aligned} \quad (14)$$

where K^- are the *ingress points* with $f(t,u,\theta)<0$, K^0 are the *grazing points* with $f(t,u,\theta)>0$ and K^+ are the e*gress points* with $f(t,u,\theta)>0$. Ingress and egress points can be easily obtained by using the tool surface geometry at the initial and final tool locations with the constraints $f(t,u,\theta)<0$ and $f(t,u,\theta)>0$, respectively. Therefore, in this section, by using the characteristic circles of surface of revolution the locations of the grazing points K^0 will be addressed. All points on the surface of revolution Φ are embedded in the characteristic circles $K(t,u,\theta)$. Therefore a normal vector on Φ can be expressed for the circular or linear spine curves as (see Fig. 6a, b)

$$\mathbf{N}(t,u,\theta) = K(t,u,\theta) - \mathbf{m}(t,u) \quad (15)$$

Up to 3½-axis machining the velocity vector at the center of a member from the two-parameter family of spheres is aligned with the instantaneous feed vector **f** (refer to Fig. 6a, b). Therefore, for an arbitrary point on K^0 the

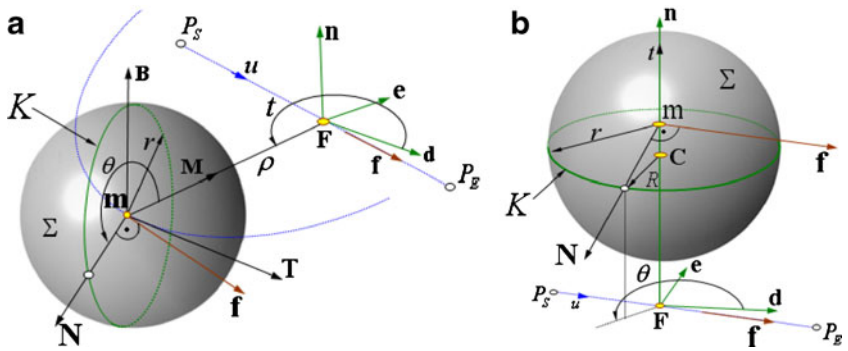

Fig. 6 Expressing the surface normals in a **a** circular and **b** linear spine curve

tangency condition $\mathbf{N}(t,u,\theta) \cdot \mathbf{f} = 0$ holds, where replacing $\mathbf{N}(t,u,\theta)$ from Eq. (15) gives

$$(K(t,u,\theta) - \mathbf{m}(t,u)) \cdot \mathbf{f} = 0 \tag{16}$$

Introducing Eq. (10) into Eq. (16) and taking the dot product yields

$$\sqrt{A_1^2 + A_2^2}\ \sin(\theta + \beta) - A_0 = 0 \tag{17}$$

Where $\beta = \tan_2^{-1}(A_2, A_1)$ and

$$\begin{aligned} A_2 &= R(t,u)(\mathbf{b}_1(t,u) \cdot \mathbf{f}) \\ A_1 &= R(t,u)(\mathbf{b}_2(t,u) \cdot \mathbf{f}) \\ A_0 &= \tfrac{r(t)r_t(t)}{\|\mathbf{m}_t(t,u)\|^2}(\mathbf{m}_t(t,u) \cdot \mathbf{f}) \end{aligned} \tag{18}$$

Solving Eq. (17) for θ yields

$$\begin{cases} \theta_1 = \sin^{-1}\left(\frac{A_0}{\sqrt{A_1^2+A_2^2}}\right) - \beta \\ \theta_2 = \pi - \sin^{-1}\left(\frac{A_0}{\sqrt{A_1^2+A_2^2}}\right) - \beta \end{cases} \tag{19}$$

For the real solutions of $\theta_{1,2}$, the condition $A_1^2 + A_2^2 \geq A_0^2$ must hold. Substituting $\theta_{1,2}$ back into Eq. (10) yields the following closed form solution for the grazing points on K^0

$$K^0(t,u,\theta_{1,2}) = \mathbf{C}(t,u) + R(t,u) \times \left(\cos\theta_{1,2}\ \mathbf{b}_1(t,u) + \sin\theta_{1,2}\ \mathbf{b}_2(t,u)\right) \tag{20}$$

By definition, a *pipe surface* is an envelope of the family of spheres with a constant radius. For example, the circular cylinder and toroidal surfaces are pipe surfaces. In pipe surfaces, the centers of the characteristic circle and moving sphere are equal to each other. If a cutter is defined by the pipe surface, then there is an alternative simpler solution for the grazing points. The following property motivates this solution

Property 1 A surface normal of a moving sphere which is in the direction of the vector product of $\mathbf{f}$ and $\mathbf{m}_t$ passes through a grazing point [25].

According to property 1, the surface normal of a moving sphere must be perpendicular to both $\mathbf{m}_t$ and $\mathbf{f}$ to pass through a grazing point, i.e., $\mathbf{N}(t,u) = \mathbf{f} \times \mathbf{m}_t(t,u)$. Thus, for the NC cutters defined by pipe surfaces, Eq. (20) can be written as

$$K^0(t,u,\theta_{1,2}) = P_{1,2} = \mathbf{m}(t,u) \mp r\frac{\mathbf{N}(t,u)}{||\mathbf{N}(t,u)||} \tag{21}$$

Given the t and u, the above equation generates two grazing points P_1 and P_2, where the angle between them is 180°.

3 In-process workpiece update

The machining process can be simulated through calculating the intersections between the discrete vectors and swept envelope SE. As given in Eq. (14), SE has three portions. For finding the intersections in K^- and K^+, the discrete vectors are intersected with the cutter surface at the initial and final positions of the tool path segment, respectively. When an intersection satisfies the tangency function $f(t,u,\theta)<0$ (in the ingress portion) or $f(t,u,\theta)>0$ (in the egress portion), then the corresponding vector is updated. In this section, the intersection calculations will be shown for the K^0 portion of *SE*. A discrete vector with an arbitrary orientation in a Cartesian coordinate system can be defined as follows

$$I(v) = I_a + (I_b - I_a)v,\ v = [0,1] \tag{22}$$

where $I_a = (x_a, y_a, z_a)$ and $I_b = (x_b, y_b, z_b)$ are the start and end positions of the vector respectively. For updating the workpiece the discrete vectors must pass through the grazing points. Therefore Eq. (22) is set to Eq. (20) as follows

$$I(v) = K^0(t,u,\theta_{1,2}) \tag{23}$$

The above equation yields a system of three equations with three variables t, u and v. If the solutions of Eq. (23) are within the parameter ranges of those variables, then the corresponding vector is updated. Because the focus is on NC machining, in this paper, the workpiece update methodology is presented by using the major milling cutter surfaces which are sub-geometries of the general surface of revolution. Those cutter geometries are represented by the natural quadrics and toroidal surfaces (see Fig. 4). Natural quadrics consist of the sphere, circular cylinder, and the cone. Together with the plane (a degenerate quadric) and torus, these constitute the surface geometries found on the majority of cutters used in milling. As given in the following property, the number of intersections obtained from Eq. (23) for the above mentioned NC cutter surfaces may vary with the degree of cutter surface.

Property 2 Given an implicitly represented cutter surface geometry $S(x,y,z,u)$ which performs linear motion under (up to) 3½-axis machining: The maximum number of intersections between a discrete vector and the envelope generated by $S(x,y,z,u)$ is equal to the degree of $S(x,y,z,u)$.

The natural quadrics are the degree of two surfaces and the torus is a degree of four surfaces [7]. Therefore,

according to property 2, a discrete vector may intersect with the envelopes of the natural quadrics at most two points and with the envelopes of the torus at most four points. In the following subsections, the application of the workpiece update methodology will be shown for the cutter surfaces having linear and circular spine curves. For this purpose, the frustum of a cone and the toroidal surfaces are chosen.

3.1 Update with the frustum of a cone

A frustum of a cone is used in the body of the taper end and the taper-ball-end mills (see Fig. 4). As illustrated in Fig. 7, the radius of a moving sphere can be defined by $r(t) = t\sin\alpha + R_b/\cos\alpha$, where R_b and α are the radius of lower base and cone half angle, respectively, and $t\in[0, h/\cos^2\alpha]$. Also from Fig. 7, the center of a moving sphere can be obtained in WCS by

$$\mathbf{m}(t,u) = \mathbf{F}(u) + t\,\mathbf{n} \tag{24}$$

where, $\mathbf{F}(u)$ has been defined in Eq. (1).

A frustum of a cone which is a surface of revolution can be represented parametrically by using the characteristic circles.

Substituting $\mathbf{m}(t,u)$ $\mathrm{r}(t)$, r_t=sin α and $\mathbf{m}_t(t,u)$=$\mathbf{n}$ into Eqs. (8) and (9) gives respectively the center and the radius of the characteristic circle as

$$\mathbf{C}(t,u) = \mathbf{F}(u) + \left(t\cos^2\alpha - R_b\tan\alpha\right)\mathbf{n} \tag{25}$$

$$R(t,u) = t\ \sin\alpha\ \cos\alpha + R_b \tag{26}$$

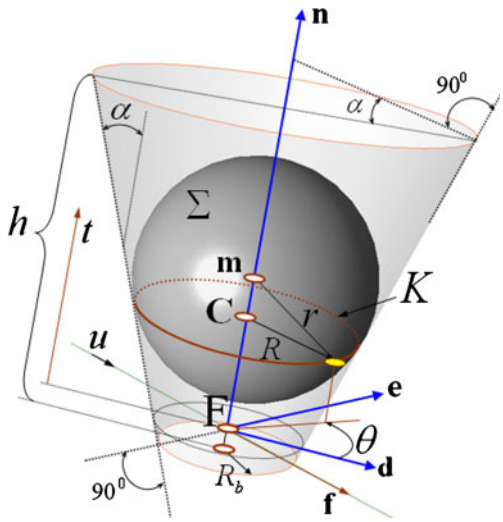

Fig. 7 Geometric description of a frustum of a cone

Then substituting $r(t)$, $r_t(t)$, $\mathbf{m}_t(t,u)$ and $R(t,u)$ into Eq. (18) yields,

$$\begin{aligned} A_2 &= (t\sin\alpha\cos\alpha + R_b)(\mathbf{d}\cdot\mathbf{f}) \\ A_1 &= 0, \\ A_0 &= \tan\alpha(t\sin\alpha\cos\alpha + R_b)(\mathbf{n}\cdot\mathbf{f}) \end{aligned} \tag{27}$$

where, **e·f=0** (refer to Eq. (2)). For finding the values of θ, the coefficients given in Eq. (27) are plugged into Eq. (19). This yields,

$$\begin{aligned} \theta_1 &= \sin^{-1}(\tan\alpha(\mathbf{n}\cdot\mathbf{f}/\mathbf{d}\cdot\mathbf{f})) - \pi/2 \\ \theta_2 &= \pi/2 - \sin^{-1}(\tan\alpha(\mathbf{n}\cdot\mathbf{f}/\mathbf{d}\cdot\mathbf{f})) \end{aligned} \tag{28}$$

Then finally substituting C(*t*,*u*), *R*(*t*,*u*), and $\theta_{1,2}$ back into Eq. (20) gives the following closed-form expression for locating the grazing points

$$\begin{aligned} K^0(t,u,\theta_{1,2}) = P_S &+ (P_E - P_S)u \\ &+ (t\sin\alpha\cos\alpha + R_b)\cos\theta_{1,2}\ \mathbf{d} \\ &+ (t\sin\alpha\cos\alpha + R_b)\sin\theta_{1,2}\ \mathbf{e} \\ &+ \left(t\cos^2\alpha - R_b\tan\alpha\right)\mathbf{n} \end{aligned} \tag{29}$$

The above equation can be rearranged as follows,

$$K^0(t,u,\theta_{1,2}) = \mathbf{O}_{1,2} + u\,\mathbf{f} + k\,\mathbf{S}_{1,2} \tag{30}$$

Where $\mathbf{f}$=P_E–P_S, k=t cos α and

$$\mathbf{O}_{1,2} = P_S + R_b\cos\theta_{1,2}\ \mathbf{d} + R_b\sin\theta_{1,2}\ \mathbf{e} - R_b\tan\alpha\ \mathbf{n}$$

$$\mathbf{S}_{1,2} = \sin\alpha\cos\theta_{1,2}\ \mathbf{d} + \sin\alpha\sin\theta_{1,2}\ \mathbf{e} + \cos\alpha\ \mathbf{n}$$

Eq. (30) generates two planar surfaces for $k\in[0,h/\cos\alpha]$ and $u\in[0,1]$ (refer to Fig. 8). These planar surfaces, spanned by linearly independent vectors $\mathbf{f}$ and $\mathrm{S}_{1,2}$, have their origins at $\mathrm{O}_{1,2}$.

To update the in-process workpiece, a discrete vector must pass through K^0. Therefore Eqs.(22) and (30) are substituted into Eq. (23). This yields a linear system of three equations in three variables as follows

$$\begin{bmatrix} u \\ v \\ k \end{bmatrix} = Q^{-1}\ U \tag{31}$$

where, $U = [I_a - \mathbf{O}_{1,2}]_{3\times1}$ and $Q = [\mathbf{f}\ (I_a - I_b)\ \mathbf{S}_{1,2}]_{3\times3}$. When a discrete vector intersects K^0, Eq. (31) gives real solutions. If these solutions satisfy the parameter ranges $k\in[0,h/\cos\alpha]$, $u\in[0,1]$ and $v\in[0,1]$ then the corresponding vector is updated. As mentioned in property 2, a discrete vector may intersect with these planar surfaces at most 2 points. Please note that when a discrete vector is parallel to K^0 or it lies on K^0 there will be no real solution for Eq. (31) because Q becomes a singular matrix.

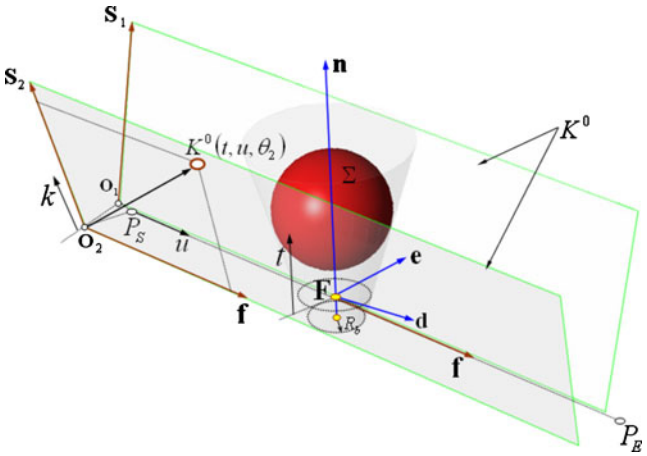

Fig. 8 Envelope surfaces generated by the frustum of a cone

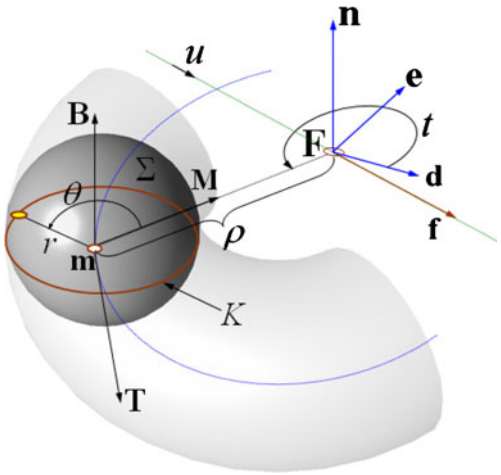

Fig. 9 Geometric description of a toroidal surface

3.2 Update with the toroidal surface

In NC machining, typically, a toroidal surface is used as one of the constituent parts of a fillet-end mill (see Fig. 4). A fillet-end mill is widely used for roughing and finishing free form surfaces. The relative orientations of the tool axis and a discrete vector affect the number of intersections generated by a toroidal section of the fillet-end mill. For example, if the axis of a fillet-end mill and a discrete vector have the same orientations then it can be said that "*intersecting a discrete vector with the envelope surface of the toroidal section always gives a maximum of one intersection point*" [15, 17-19]. For this case, one of the numerical root finding methods, i.e., bisection [29], can be applied to the nonlinear function obtained from the vector-envelope intersections. On the other hand, if the orientations of the fillet-end mill and the discrete vectors are different then the number of intersections can be more than one and therefore the numerical root finding methods cannot be applied directly. In this section, a method for addressing these multiple intersections a nonlinear root finding analysis will be developed. In Fig. 9, a moving sphere on the circular spine curve is illustrated. From this figure, the sphere center in WCS can be obtained by

$$\mathbf{m}(t,u) = \mathbf{F}(u) + \rho\cos t\ \mathbf{d} + \rho\sin t\ \mathbf{e}, \quad t \in [0,\ 2\pi] \qquad (32)$$

Where ρ is the radius of the circular spine curve. The partial derivative of Eq. (32) with respect to t is obtained by

$$\mathbf{m}_t(t,u) = -\rho\sin t\ \mathbf{d} + \rho\cos t\ \mathbf{e} \qquad (33)$$

As mentioned in Section 2.2, the toroidal surface is a pipe surface. Therefore, in the following derivations for simplicity, Eq. (21) will be used. Given the t and u Eq. (21) generates two grazing points P_1 and P_2, where the angle between them equals to 180° (see Fig. 10).

There are three types of tool movements up to 3½-axis machining with respect to the tool axis **n** and feed **f** vectors (refer to Eq. (2)): an ascending movement ($\mathbf{n}\cdot\mathbf{f} > 0$), a horizontal movement ($\mathbf{n}\cdot\mathbf{f} = 0$) and a descending movement ($\mathbf{n}\cdot\mathbf{f} < 0$). During the descending or ascending motions of the toroidal section, P_2 always lies within the cutter body and therefore it has no effect on the workpiece update process. Therefore, during the workpiece update process, a discrete vector can intersect with toroidal section at P_1 which corresponds to the (+) sign under a descending motion or the (–) sign under an ascending motion in Eq. (21). For the following calculations, without loss of generality, it is assumed that the tool has a descending motion. For the ascending and horizontal motions, similar steps can be

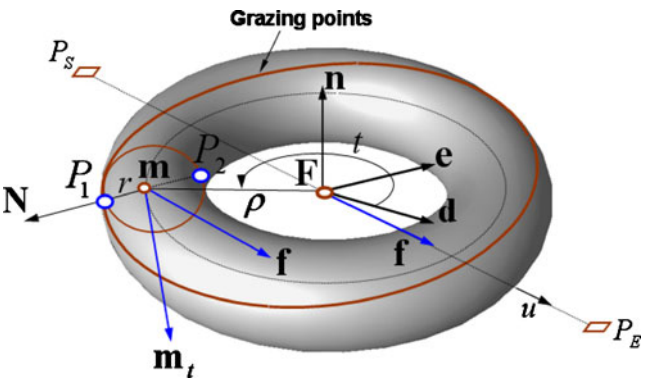

Fig. 10 Grazing points P_1 and P_2 on the torus during a descending motion

applied. Substituting $\mathbf{m}(t,u)$ and $\mathbf{m}_t(t,u)$, into Eq. (21) for the (+) sign yields

$$P_1 = \mathbf{F}(u) + \rho\cos t\,\mathbf{d} + \rho\sin t\,\mathbf{e} + r\,\frac{-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t\,(\mathbf{f}\times\mathbf{e})}{\|-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t\,(\mathbf{f}\times\mathbf{e})\|} \tag{34}$$

For updating the workpiece, a discrete vector must pass through P_1. By setting Eq. (34) equal to Eq. (22), a nonlinear system of three equations in three variables u, v, and t is obtained

$$G_i(u,v,t) = \mathbf{F}(u) + \rho\cos t\,\mathbf{d} + \rho\sin t\,\mathbf{e} + r\,\frac{-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t\,(\mathbf{f}\times\mathbf{e})}{\|-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t\,(\mathbf{f}\times\mathbf{e})\|} - I(v) = 0 \tag{35}$$

Where i=1,2,3. For solving the above system of equations, the equivalent of v from $G_1(u,v,t)$=0 is substituted into $G_{2,3}(u,v,t)$=0. Thus, two nonlinear equations with the variables u and t are obtained. Then, the equivalent of u from one of these two equations is substituted into the other for obtaining the final nonlinear equation with respect to t. For the following calculations without loss of generality we assume that, the given discrete vectors have orientations along the $\mathbf{z}$=(0,0,1) axis of the Cartesian coordinate system. In this case, the equations $G_{1,2}(u,v,t)$=0 will depend on the variables t and u only. When the equivalent of u from $G_2(u,v,t)$=0 is substituted into $G_1(u,v,t)$=0 the following equation in a single variable is obtained

$$g(t) = A(t)\cos t + B(t)\sin t + C = 0 \tag{36}$$

Where

$$A(t) = \rho\,(y_E - y_S)\,d_x - \rho\,(x_E - x_S)\,d_y + r\,\frac{(f_y e_z - f_z e_y)(y_E - y_S) - (f_z e_x - f_x e_z)(x_E - x_S)}{\|-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t(\mathbf{f}\times\mathbf{e})\|}$$

$$B(t) = \rho\,(y_E - y_S)\,e_x - \rho\,(x_E - x_S)\,e_y + r\,\frac{(f_z d_x - f_x d_z)(x_E - x_S) - (f_y d_z - f_z d_y)(y_E - y_S)}{\|-\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t(\mathbf{f}\times\mathbf{e})\|}$$

$$C = (x_E - x_S)\,(y_a - y_S) + (y_E - y_S)\,(x_S - x_a)$$

When the parameters $t\in[0,2\pi]$ and $u\in[0,1]$ are plugged into Eq. (34), a two-part envelope surface is obtained as follows (see Fig. 11)

$$K^0(P_1) = K^0_{\text{Lower}} \cup K^0_{\text{Upper}} \tag{37}$$

where K^0_{Lower} is defined in $t \in [\pi/2,\ 3\pi/2]$ and K^0_{Upper} is defined in $t \in [-\pi/2,\ \pi/2]$.

Please note that in this research the toroidal surface is considered as one of the constituent parts of a fillet-end mill. Therefore, the $K^0_{\text{Upper}}$is enclosed within the envelope of the cylindrical part of the fillet-end mill (refer to Fig. 4). Thus the workpiece is updated by the K^0_{Lower}. As mentioned in property 2, there can be up to four intersection points between a discrete vector and the envelopes of a toroidal surface. Because we eliminate P_2 from calculations, the maximum number of vector-envelope intersections are reduced to two (refer to Fig. 11). Therefore it can be concluded that the nonlinear function given in Eq. (36) has maximum two real roots which correspond to maximum two intersections between a discrete vector and the K^0_{Lower}(see Fig. 11).

The first step for solving the Eq. (36) is to check the signs of $g(t)$ at $\pi/2$ and $3\pi/2$.

- If $g(\pi/2)g(3\pi/2)<0$ then the number of intersections between a discrete vector and the K^0_{Lower} is one (refer to Fig. 12a). In this case, to find the single root of $g(t)$ in $t\in[\pi/2,3\pi/2]$ one of the numerical root finding methods, i.e., bisection, can be applied.
- On the other hand, if the signs of $g(t)$ at $\pi/2$ and $3\pi/2$ are the same, i.e., $g(\pi/2)g(3\pi/2)>0$ then the number of vector-envelope intersections are either two or zero (refer to Fig. 12b). Therefore, the numerical root finding methods cannot be applied directly. For this case, the following theorem motivates finding the roots.

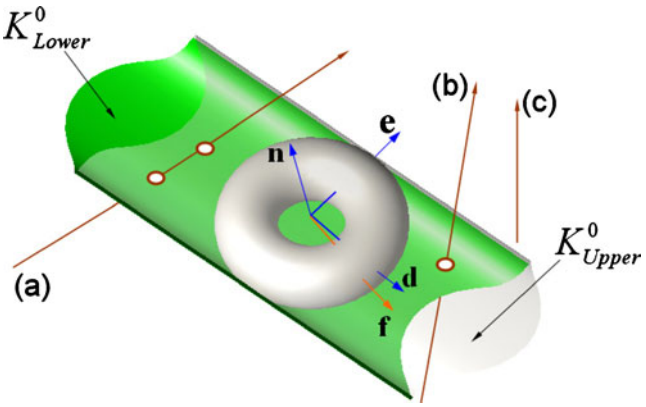

Fig. 11 A discrete vector intersects K^0_{Lower} **a** at two points, **b** at one point, and **c** no intersection

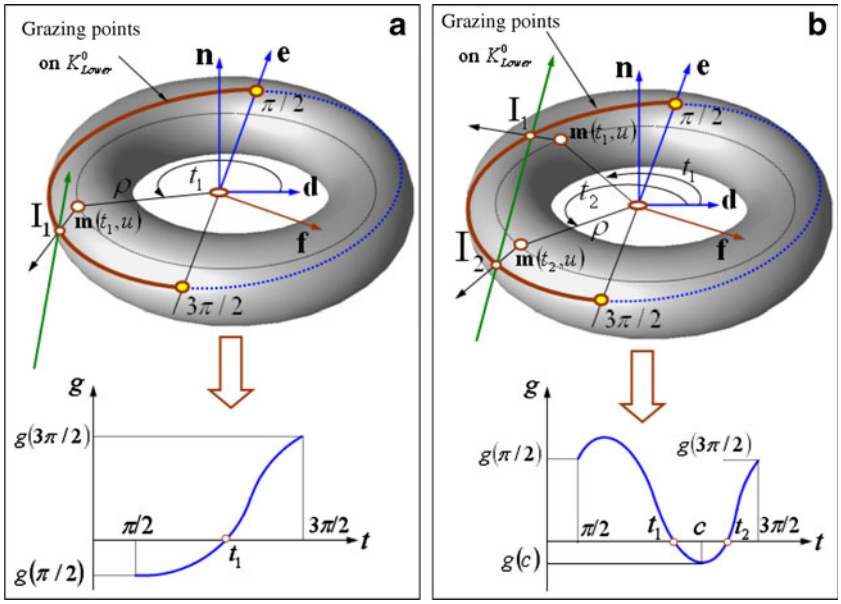

Fig. 12 Intersection cases with their corresponding roots: **a** one-point, and **b** two-point intersections

Theorem 1 *(Rolle's theorem) Let the function g be continuous on* [*a*, *b*], *and differentiable on* (*a*, *b*), *and suppose that* $g(a)=g(b)$. *Then there is some c with* $a<c<b$ *such that* $g'(c)=0$.

The denominators $\| -\sin t\,(\mathbf{f}\times\mathbf{d}) + \cos t(\mathbf{f}\times\mathbf{e}) \|$ of $A(t)$ and $B(t)$ in Eq. (36) cannot be zero. Therefore, $g(t)$ is a continuous function on $[\pi/2,3\pi/2]$. In Eq. (36), because the constituent parts $A(t)$, cos*t*, $B(t)$, sin*t* and *C* are differentiable on $[\pi/2,3\pi/2]$ then $g(t)$ is also differentiable on this interval. Furthermore, if $g(t)$ has two roots t_1 and t_2 on $[\pi/2,3\pi/2]$ then $g(t_1)=g(t_2)$ (refer to Fig. 12b). Therefore, the Rolle's theorem holds for $g(t)$ and thus there is a *c* with $t_1<c<t_2$ such that $g'(c)=0$. Using the *c* the interval $t\in[\pi/2,3\pi/2]$ is partitioned into two-sub intervals, i.e., $[\pi/2,c]$ and $[c,3\pi/2]$, where $g(\pi/2)g(c)<0$ and $g(c)g(3\pi/2)<0$, respectively. Now applying the bisection method in each sub interval, the roots t_1 and t_2 of $g(t)$ can be found. In this research for finding the *c*, a Matlab function *fminbnd* which finds the minimum of a function of one variable within a fixed interval is utilized. The algorithm used in *fminbnd* is based on the golden section search and parabolic interpolation. Please note that if the sign of $g(c)$ is the same as those of $g(\pi/2)$ and $g(3\pi/2)$, then it can be concluded that the $g(t)$ does not have any real root. Also note that when $g(\pi/2)g(3\pi/2)>0$, the signs of $g(t)$ at $\pi/2$ and $3\pi/2$ are either negative or positive. If both signs are negative then for applying *fminbnd* the $(-g(t))$ is used. This process only reverses $g(t)$ with respect to the *t*-axis and does not change the locations of the roots. The whole process of finding the roots of $g(t)$ is described in Algorithm (1).

Algorithm 1 Finding the roots of $g(t)=0$

Inputs The function $g(t)$ on $[\pi/2,3\pi/2]$

IF the signs of $g(\pi/2)$ and $g(3\pi/2)$ are different **THEN**

$t_1 \leftarrow$ **bisection**$(g, \pi/2, 3\pi/2)$

ELSE /*signs are the same*/

IF signs of $g(\pi/2)$ and $g(3\pi/2)$ are negative **THEN**

$g(t) \leftarrow -g(t)$ /*reverse with respect to *t*-axis*/

ENDIF

$c \leftarrow$ **fminbnd**$(g, \pi/2, 3\pi/2)$

IF $g(c)$ is negative **THEN**

$t_1 \leftarrow$ **bisection**$(g, \pi/2, c)$

$t_2 \leftarrow$ **bisection**$(g, c, 3\pi/2)$

ELSE

/*There is no intersection*/

ENDIF

ENDIF

The roots of $g(t)$ obtained from Algorithm 1 are substituted into $G_1(u,v,t)=0$, given in Eq. (35), for finding $u_{1,2}$. If $u_{1,2}\in[0,1]$ then $t_{1,2}$ and $u_{1,2}$ are plugged into $G_3(u,v,t)=0$ for obtaining the $v_{1,2}$. If $v_{1,2}\in[0,1]$ then the corresponding discrete vector is updated. The flowchart in

Fig. 13 shows the steps used in the proposed in-process workpiece update methodology.

4 Implementation

For implementing the described methodology, the workpiece surfaces must be represented by discrete vectors. There are two main representations: discrete normal vector (DNV) and discrete vertical vector (DVV) [30]. In DNV approach, a workpice is made up of discrete vectors whose directions are surface normals of the workpiece. In this approach, the workpiece surfaces can be represented well with respect to a given tolerance. But because the directions of discrete vectors are not identical, localizing the vectors for the given tool motion becomes difficult. During the machining simulation, the most time-consuming process is the localization of the vectors. On the other hand, in DVV approach, all vectors lie along the vertical z-axis of the Cartesian coordinate system. Although the DVV approach gives less accurate results the localization of the vectors is much easier. In DVV representation, the accuracy can be increased by closely spacing the vectors.

Although the workpiece update methodology, given in Section 3, can be used in both DNV and DVV representations for the localization purposes, the machining examples will be generated using the DVV representation. For this purpose, a localization methodology, based on the AABB, will be developed. The AABB, a rectangular four-sided box in 2D, is one of the most common bounding boxes. The best feature of the AABB is its fast overlap check which simply involves direct comparison of individual coordinate values. For localizing the discrete vectors the following steps are used (refer to Fig. 14).

Step 1 First the bounding cylinder of a cutter, located at P_S with an arbitrary orientation $\mathbf{n}=(n_x,n_y,n_z)$, is found. Then a symmetry plane, which contains the tool axis, passes through the c_{top} and c_{buttom} circles of the bounding cylinder. The normal of the symmetry plane is calculated by $\mathbf{N}_{plane} = \mathbf{n} \times [0 \quad 0 \quad 1]$. Thus, the intersection points $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$, and $\mathbf{I}_4$ are generated in the 3D space [31].

Step 2 Assuming $\mathbf{I}_1$ and $\mathbf{I}_4$ are the outermost points, the circles c_{top}, c_{buttom}, and $\mathbf{I}_1$, $\mathbf{I}_2$, $\mathbf{I}_3$, $\mathbf{I}_4$ are projected onto the Oxy coordinate plane. This generates a bounding box of the cutter projection with vertices P_1, P_2, P_3, and P_4. These vertices can be calculated as follows [31]

$$\mathbf{P}_{1,2} = \mathbf{I}_1 \mp r \begin{bmatrix} n_y/\sqrt{n_x^2+n_y^2} \\ -n_x/\sqrt{n_x^2+n_y^2} \\ 0 \end{bmatrix}, \quad \mathbf{P}_{3,4} = \mathbf{I}_4 \mp r \begin{bmatrix} n_y/\sqrt{n_x^2+n_y^2} \\ -n_x/\sqrt{n_x^2+n_y^2} \\ 0 \end{bmatrix} \tag{38}$$

Step 3 Steps 1 and 2 are applied to the cutter located at P_E with orientation $\mathbf{n}$. This yields a cutter projection on Oxy with vertices P_5, P_6, P_7 and P_8 (refer to

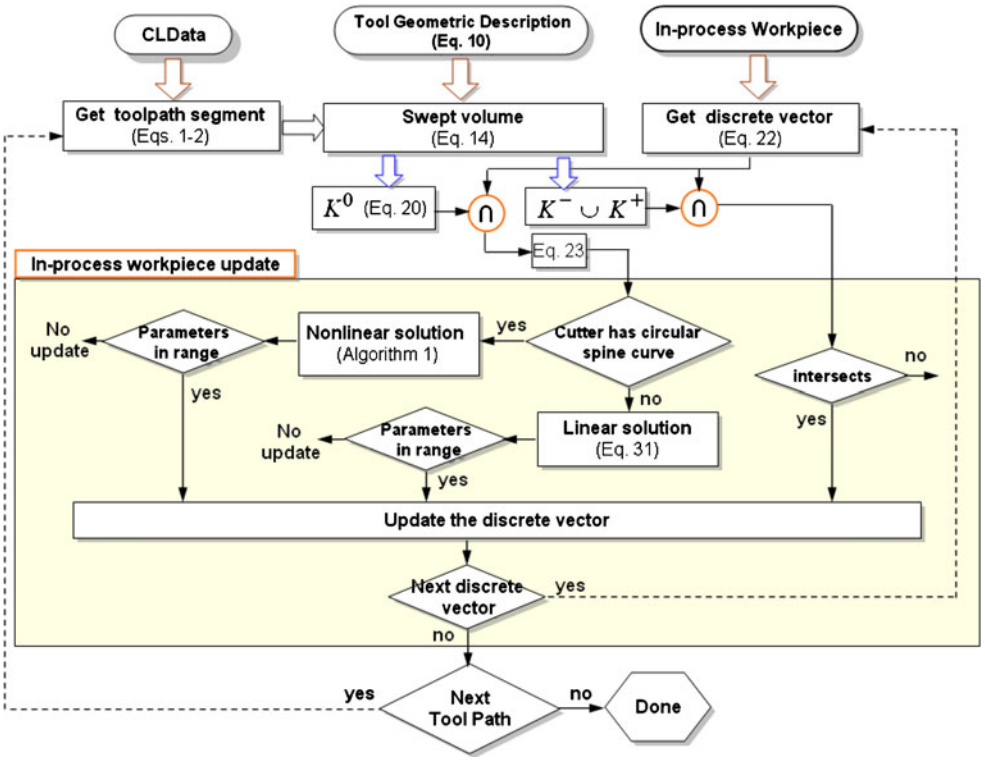

Fig. 13 Flowchart of the proposed in-process workpiece update methodology

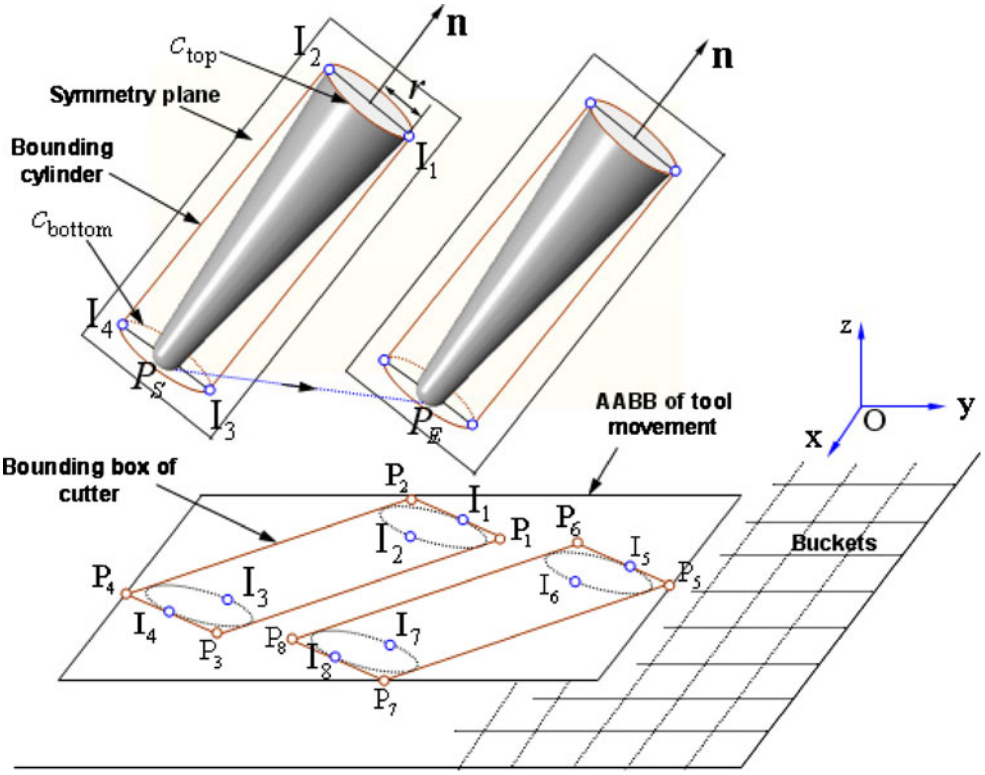

Fig. 14 Bounding box of a tool movement in Oxy coordinate plane

Fig. 14). Then, using the minimum and maximum x and y coordinates of the vertices (P_1-P_8), the AABB of the tool movement is obtained (refer to Fig. 14). Please note that because the tool orientation does not change along the toolpath segment, the projections of the intermediate tool motions between P_S and P_E on the Oxy plane are also located in the same AABB of the tool movement.

Step 4 The buckets in the Oxy coordinate plane can be thought as AABBs which contain the discrete vectors. In this step, the overlap check is performed between the buckets and the AABB of the tool movement. The algorithm for the AABB–AABB intersection can be found in [32]. If an overlap is found then the discrete vectors in the overlapping bucket are used in the workpiece update process described in Section 3.

The described methodology has been implemented using C++ codes. Figure 15 shows the simulation result of an automobile hood. The NC program consists of 16,768 steps for one layer of toolpath. For this example, a ball-end mill with 25.4 mm radius is used. The workpice dimension is 1,694×890×870 mm and it is represented using DVVs with 2 mm resolution. The toolpath file for this example is taken from VERICUT which uses image space approach in machining simulations.

The next example is for machining a gear box cover (Fig. 16a). The roughing is performed with a fillet-end mill of diameter 10 cm and corner radius 1 cm (Fig. 16b), the semi-finishing is performed with a ball-end mill of diameter 10 cm (Fig. 16c), and finally the finishing is done with a ball-end mill of diameter 5 cm (Fig. 16d). The workpiece has the dimension of 100 ×80 ×44 cm represented using DVVs with 2 mm resolution and the NC programs for roughing, semi-finishing, and finishing consist of 2,456 steps, 4,612 steps and 5,320 steps, respectively. The toolpath files for the gear box cover example are generated by using Unigraphics NX. In these examples, the final workpiece surfaces are tessellated by using Matlab.

5 Discussion and conclusions

The methodology presented in this paper targets the important problem of updating the workpiece surfaces in milling simulations. In this methodology, the workpiece is represented by a series of discrete vectors, which may be orientated in different directions and the cutter is modeled as a surface of revolution, which can be considered as a canal surface formed by sweeping a sphere with varying radius along a spine curve. Therefore, the presented methodology is more general than previous publications as it will handle a broader variety of tool shapes.

In the most recently published, the object space-based methodologies [18, 19], the cutter can follow only three-axis toolpaths, in which the cutter axis is fixed along the

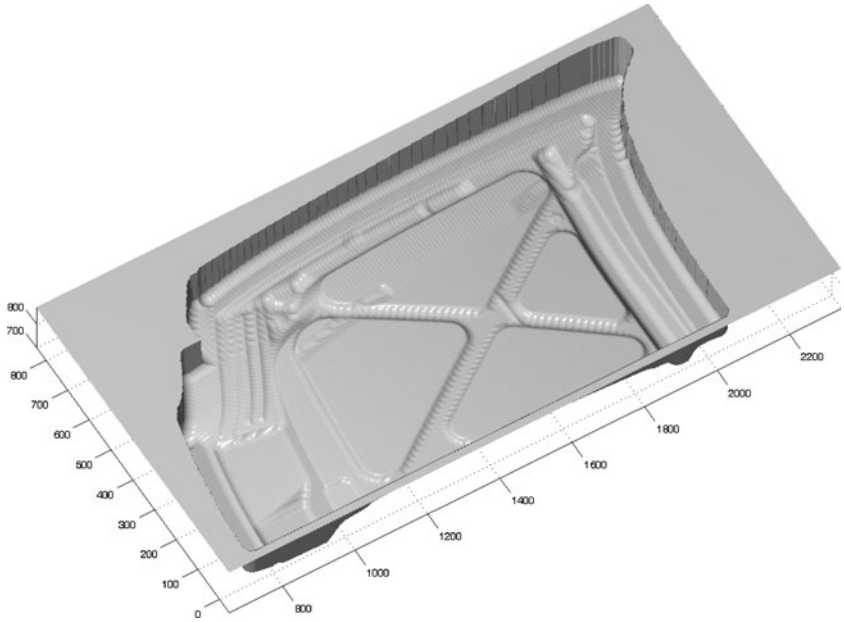

Fig. 15 NC milling simulation result for an automobile hood

z-axis of the Euclidian coordinate system. Because the three-axis tool paths are not sufficient for finishing some part geometries and the exact five-axis toolpaths are not preferable for simulations because of the extensive nonlinear calculations, the proposed in-process workpiece update methodology has been developed for up to 3½-axis toolpaths in which a cutter can have an arbitrary orientation in space. The authors believe that the presented methodology can be used to approximate five-axis tool motions and further information for these approximations can be found in [16].

To update the in-process workpiece surfaces the discrete vectors are intersected with the cutter swept volume which

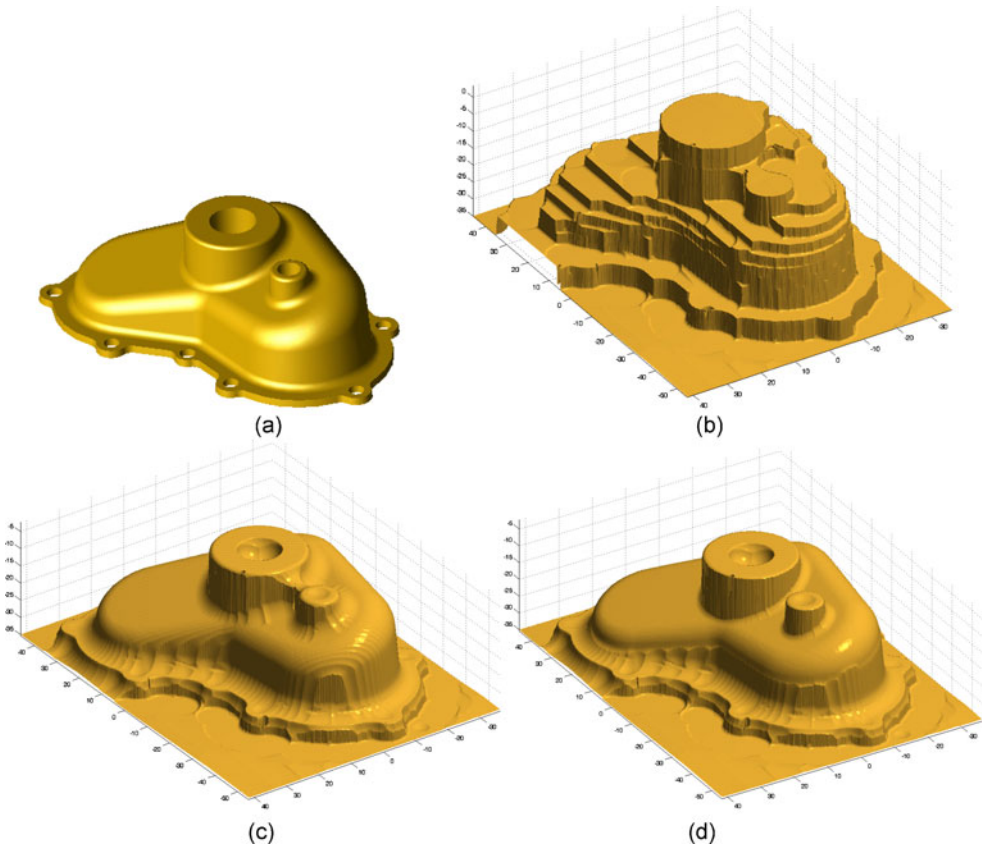

Fig. 16 NC milling simulation results for a gear box cover

contains the ingress, grazing, and the egress regions. Using the cutter surface geometry at the initial and final positions of a tool movement the intersection points at the ingress and egress regions can be obtained, respectively. Therefore, in this paper, the in-process workpiece update methodology has been developed for finding the intersections at the grazing region. For addressing the grazing region, at which the discrete vectors can intersect the tool envelopes, the cutter surfaces have been decomposed into a set of characteristic circles which are generated by the two-parameter family of spheres.

Because the focus is on NC machining, the implementation of the methodology has been shown for the subgroups of the general surface of revolution which have linear and circular spine curves. As shown in Section 3.1 using the frustum of a cone surface, the vector-tool envelope intersections can be found analytically for the cutters with linear spine curves. However, the intersection calculations for cutters having circular spine curves require solving a system of nonlinear equations. The implementation of the methodology for the circular spine curve has been shown using the toroidal surface of a fillet-end mill. Because the toroidal surface is a pipe surface, the calculations can be simplified. If the cutter axis and a discrete vector have the same orientations, then the number of intersections between the discrete vector and the swept envelope of a toroidal section of the fillet-end mill is maximum one. For this case, one of the numerical root-finding methods, i.e., bisection, can be applied. In the developed in-process workpiece update methodology, both the discrete vectors and the tool axis may have arbitrary orientations. Therefore, to find the intersections between discrete vectors and swept envelopes generated by the torus becomes more complicated. In this case, the number of intersections can be more than one and therefore the numerical root finding methods cannot be applied directly. For solving this problem, the system of nonlinear equations have been transformed into a single variable nonlinear function and then a root finding analysis has been developed to guarantee finding the root(s) in the given interval.

A typical milling tool path contains thousands of tool movements and during the machining simulation for calculating the intersections only a small percentage of all the discrete vectors are needed. To improve the computational efficiency when updating the workpiece, a vector localization scheme has been developed based on the AABB method. The best feature of the AABB is its fast overlap check, which simply involves direct comparison of individual coordinate values. Although the workpiece update methodology has been developed for the DNV representation, for efficiency reasons in the vector localization, the examples have been generated by using the DVV representation. The localization of the cutter envelope surfaces in DNV representations will be investigated. The described technique has been developed as part of a virtual milling system that combines the geometric modeling aspects of milling material removal with the modeling of the physics of the machining process.

Acknowledgment This research was funded in part by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would also like to acknowledge the significant support and encouragement of Professor Yusuf Altintas in the Department of Mechanical Engineering at the University of British Columbia throughout the course of this work.

References

1. Voelcker H. B, Hunt W. A. Role of solid modeling in machining-process modeling and NC verification. SAE Preprints 1981; n: 810195, 10p.
2. Wang WP (1988) Solid modeling for optimizing metal removal of three-dimensional NC end milling. J Manuf Syst 7(1):57–65
3. Spence AD, Altintas YA (1994) Solid modeler-based milling process simulation and planning system. J of Eng for Ind, Transactions of the ASME 116(1):61–69
4. El Mounayri H, Spence AD, Elbestawi MA (1998) Milling process simulation—a generic solid modeler based paradigm. Journal of Manufacturing Science and Engineering, Transactions of the ASME 120(2):213–221
5. Gupta SK, Saini SK, Brent WS, Yao Z (2005) Geometric algorithms for computing cutter engagement functions in 2.5D milling operations. Comput-Aided Des 37:1469–1480
6. Yip-Hoi D, Huang X (2006) Cutter/Workpiece engagement feature extraction from solid models for end milling. Journal of Manufacturing Science and Engineering, Transactions of the ASME 128(1):249–260
7. Aras E, Yip-Hoi D. (2008) Geometric Modeling of Cutter/Workpiece engagements in three-axis milling using polyhedral representations. Journal of Computing and Information Science in Engineering, Transactions of the ASME 8(3).
8. Jerard RB, Drysdale RL, Hauck KE, Schaudt B, Magewick J (1989) Methods for detecting errors in numerically controlled machining of sculptured surfaces. Computer Graphics and Applications, IEEE 9(1):26–39
9. Chappel IT (1983) The use of vectors to simulate material removed by numerically controlled milling. Comput-Aided Des 15(3):156–158
10. Oliver JH, Goodman ED (1986) Color graphic verification of N/C milling programs for sculptured surface parts. First Symposium on Integrated Intelligent Manufacturing ASME
11. Wang WP, Wang KK (1986) Real-time verification of multi-axis NC programs with raster graphics. IEEE Proc. Int'l Conf. Robotics and Automation, CS Press; 166-171.
12. Wang WP, Wang KK (1986) Geometric modeling for swept volume of moving solids. CG&A 6(12):8–17
13. Van Hook T (1986) Real-time shaded NC milling display. Computer Graphics (Proc SIGGRAPH) 20(4):15–20
14. Atherton P, Earl C, Fred C. (1987) A graphical simulation system for dynamic five-axis NC verification. Proc. Auto fact. SME (2-1 to 2-12).
15. Choi B.K., Jerard R.B. (1998). Sculptured surface machining: Theory and Applications. Kluwer Academic Publishers
16. Fussel BK, Jerard RB, Hemmet JG (2003) Modeling of cutting geometry and forces for 5-axis sculptured surface machining. Comput-Aided Des 35:333–346
17. Chung YC, Park JW, Shin H, Choi BK (1998) Modeling the surface swept by a generalized cutter for NC verification. Comput-Aided Des 30(8):587–594

18. Baek N, Maeng SR, Shin SY, Choi BK (2003) A Z-map update method for linearly moving tools. Comput-Aided Des 35:995–1009
19. Baek N, Maeng SR, Shin SY, Choi BK (2004) A fast NC simulation method for circularly moving tools in the Z-map environment. Proceeding of the Geometric Modeling and Processing
20. Jerard RB, Hussaini SZ, Drysdale RL, Schaudt B (1989) "Approximate Methods for Simulation and Verification of Numerically Controlled Machining Programs", The Visual Computer, vol. 5, no. 6
21. Drysdale RL, Jerard RB, Schaudt B, Hauck K (1989) Discrete simulation of NC machining. Algorithmica Special Issue on Computational Geometry 4(1):33–60
22. Quinn J (1993) Accurate verification of five-axis numerically controlled machining. PhD Thesis, Department of Computer Science, Dartmouth College, Tech Report PCS-TR93-191
23. Chen CZ, Dong Z, Vickers WG (2003) Automated surface subdivision and toolpath generation for 3½½ axis CNC machining of sculptured parts. Computers in Industry 50:319–331
24. Peternell M, Pottmann H (1997) Computing rational parameterizations of canal surfaces. Journal of Symbolic Computation 23:255–266
25. Aras E (2009) Generating cutter swept envelopes in five-axis milling by two-parameter families of spheres. Comput-Aided Des 41(2):95–105
26. Frey DD, Otto KN, Pflager W (1997) Swept envelopes of cutting tools in integrated machine and workpiece error budgeting. Ann CIRP 46(1):475–480
27. Wang WP, Wang KK (1986) Geometric modeling for swept volume of moving solids. IEEE Comput Graph Appl 6(12):8–17
28. Blackmore D, Leu MC, Wang LP (1997) The sweep-envelope differential equation algorithm and its application to NC-machining verification. Comput-Aided Des 29(9):629–637
29. William HP, Saul AT, William TV, Brian PF (2007) Numerical Recipes 3rd Edn. The Art of Scientific Computing
30. Jung WP, Yang HS, Yun CC (2005) Hybrid cutting simulation via discrete vector model. Comput-Aided Des 37:419–430
31. You CF, Chu CH (1997) Toolpath verification in five-axis machining of sculptured surfaces. Int J Adv Manuf Technol 13 (4):248–255
32. Ericson C (2005) Real-time collision detection. The Morgan Kaufmann Series in Interactive 3-D Technology