Swept Tool Envelope and Machining Potential Field for 5-Axis Sculptured Surface Machining

Yuan-Shin Lee[1] and John C. J. Chiou[2]

[1]North Carolina State University, yslee@eos.ncsu.edu
[2]UniGraphics Solutions, mediaaccount@yahoo.com

ABSTRACT

This paper presents the machining potential field (MPF) and the explicit solutions of swept envelopes for 5-axis sculptured surface machining. The properties of the swept envelopes are analyzed. The results help us realize the geometric matching in tool-tip machining and understand the geometric machinability in tool-side machining. The complement of the swept envelope, which represents the in-process workpiece, is also addressed. It finally presents the swept envelope applications in tool orientation determination, machining interference avoidance, and run-time simulation.

Keywords: swept envelope, swept profiles, NC simulation.

1. INTRODUCTION

When a solid object moves, it creates a three-dimensional swept volume. This swept volume defines a space where is ever occupied by the object during its motion. Anything inside this occupied space is collided with the object, while anything outside the space is collision-free. The swept envelope defines the boundary between the collision and collision-free space. Therefore, the swept envelope plays an important role in a wide variety of geometric applications such as geometric construction, robot workspace configuration, and collision detection [1, 2].

Swept envelope computation is usually difficult because of the nonlinearity of the problem. It is generally impossible to obtain closed form expressions for the swept envelope created by objects with arbitrary geometric shapes [1, 3]. Fortunately the majority of tool geometry for robots and NC machining is axial symmetry and defined by surfaces of revolution. These tools usually can be decomposed into geometric primitives, such as spheres, tori, cylinders, and cones [4]. Due to their simplicity, the swept envelopes of the geometric primitives can be easily determined and explicitly presented. The swept envelopes of the original tools can then be integrated by the swept envelopes of their primitives [5].

This paper details the determination of the swept envelope of axial symmetric tools, and its applications. It first analyzes tool geometry and its decomposition of geometric primitives. Then it discusses the tool position and the tool motion. Based on the tool geometry and motion, the explicit solution of the swept envelope is derived. It further analyzes the properties of the swept envelope. The results deduce us how to match the tools with the part geometry, and reveals the machinability of the geometry. The construction of the complement of the swept envelope is also discussed. After illustrating examples in computer applications, we conclude in the final section.

2. TOOL GEOMETRY

Tools used in robots and NC machining industry are usually axial symmetry. The geometry can be represented by surfaces of revolution, like spheres, tori, cylinders, and cones. This research limits the object geometry to these four types of primitive geometry only. Without lose of generality and to facilitate the tool representation in the following sections, the tool origin is defined at its top point as Fig. 1 shown. A tool composed by a cone and a torus can be represented as follows [4]:

$$O(s)=\begin{bmatrix}(e+r_{tc}\cos\beta+(1-c_{cone})(l-r_{tc}(1-\sin\beta))\tan\beta)\cos\theta\\(e+r_{tc}\cos\beta+(1-c_{cone})(l-r_{tc}(1-\sin\beta))\tan\beta)\sin\theta\\c_{cone}(l-r_{tc}(1-\sin\beta))\\1\end{bmatrix}_{cone}\cup\begin{bmatrix}(e+r_{tc}\sin\phi)\cos\theta\\(e+r_{tc}\sin\phi)\sin\theta\\l-r_{tc}(1-\cos\phi)\\1\end{bmatrix}_{torus} \quad (1)$$

where s represents the parameters for the geometry description. $c_{cone}\in[0,1]$, $\phi\in[0,(\pi/2-\beta)]$, and $\theta\in[0,2\pi]$. By changing the parameters, a variety of tool geometry can be defined. When $\beta=0$, the cone becomes a cylinder, while

$e = 0$, the torus degenerates into a sphere. For a milling tool, the cone and the torus portions are the only sections that remove raw material. A cone or a plane may be added to the top and the bottom to make the surface of the tool closed. For simplicity, this paper only discusses the swept envelope of the tool in the cone and the torus portions.

The surface normal of the cone is determined as follows:

$$\vec{N}_{cone} = \begin{bmatrix} \cos\theta & \sin\theta & \tan\beta & 0 \end{bmatrix}^{T} \tag{2}$$

Similarly the surface normal of the torus is determined as follows:

$$\vec{N}_{torus} = \begin{bmatrix} \sin\phi\cos\theta & \sin\phi\sin\theta & \cos\phi & 0 \end{bmatrix}^{T} \tag{3}$$

If the tool can not be considered as a rigid body but deflection geometry due to the external forces, the deformed geometry can be determined by a deformation matrix as follows:

$$O_{def}(s) = M_{def}(z_c)\, O(s) \tag{4}$$

The surface normal for the deflection tool can also be determined similarly to Eq. (4). The following sections will use the tool geometry and the surface normal to determine the swept envelope.

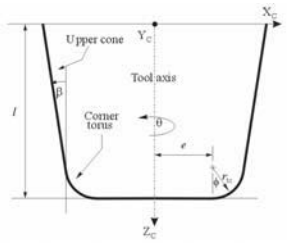

Fig. 1. Tool geometry.

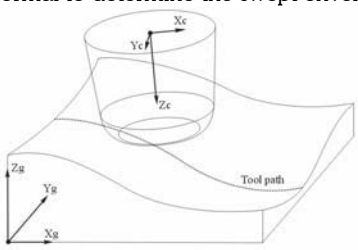

Fig. 2. Global coordinate system.

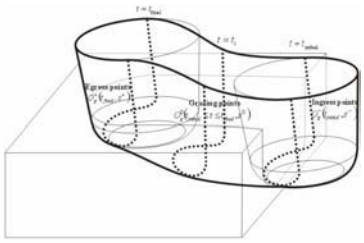
Fig. 3. Swept envelope.

3. TOOL MOTION

The tool geometry is initially defined in its coordinate system (Xc-Yc-Zc). To describe the tool location and orientation during the motion, a global coordinate system (Xg-Yg-Zg) is defined, as shown in Fig. 2. The tool is transformed from its coordinate system to the global coordinate system as follows:

$$O_{global}(t,s) = M_{global}(t)\, M_{def}(z_c)\, O(s) = \begin{bmatrix} R_{rot}(t) & T_{trans}(t) \end{bmatrix} M_{def}(z_c)\, O(s) \tag{5}$$

The transformation matrix includes two parts - rotation and translation - to describe the orientation and the position of the tool at any point in time. Using the same transformation matrix, the surface normal of the tool can also be transformed to the global coordinate system as follows:

$$\vec{N}(t,s) = M_{global}(t) M_{def}(z_c) \vec{N}_{tool}(s) \tag{6}$$

The $\vec{N}_{tool}$ represents the surface normal of the tool in its coordinate system. When the tool moves, the speed vector of any point on its surface is determined as follows [20]:

$$\vec{V}(t,s) = \frac{\partial O_{global}(t,s)}{\partial t} = \frac{\partial \begin{bmatrix} R_{rot}(t) & T_{trans}(t) \end{bmatrix}}{\partial t} M_{def}(z_c) O(s) = \frac{\partial T_{trans}}{\partial t} + \Omega \times \left(\begin{bmatrix} R_{rot} & 0 \end{bmatrix} M_{def}(z_c) O(s) \right) \tag{7}$$

A general motion of a tool includes simultaneously translational speed and rotational speed. The $\partial T_{trans}/\partial t$ is the translational speed vector of the tool. The Ω is the rotational speed vector of the tool. The surface normal represented in Eq. (6) and the speed vector in Eq. (7) will be used to determine the tool swept envelopes. Details are discussed in the following sections.

4. SWEPT ENVELOPE

The swept volume of a moving tool is considered as the totality of the points that belongs to the trace of the moving tool [6, 7]. The swept envelope is the surface that encloses the swept volume. Material inside the swept volume will collide with the tool and be removed during the machining processes. The machined part geometry is the complement of the swept volume from the raw material. When gouge occurs or interference takes place, the gouge and the interface can be detected by the intersection between the swept envelope and the part surface [8].

The key idea of the swept envelopes is that, at any point in time, the moving tool is always tangent to the swept envelope along a contact curve (or curves), as shown in Fig. 3. Thus the speed vector $\vec{V}$ of the moving tool along the contact curve is tangent to the envelope surface. In addition, the normal $\vec{N}$ of the moving tool along the contact curve is parallel to the normal of the envelope surface. Therefore, any point on the moving tool satisfying $\vec{V} \cdot \vec{N} = 0$ is on the

envelope surface. The swept envelope also consists of the rear surface of the tool at the initial location, where $\vec{V}\cdot\vec{N}<0$, and the front surface of the tool at the final location, where $\vec{V}\cdot\vec{N}>0$. Therefore, to determine the swept envelope of a moving tool, a tangency function *F* is defined as follows [9]:

$$F(t,s)=\vec{V}(t,s)\cdot\vec{N}(t,s) \tag{8}$$

By using the tangency function *F*, the swept envelope *SE* of a moving tool is determined as follows [9, 10]:

$$SE\left(O_{global}\left(t,s^{*}\right)\right)=\left\{O^{-}_{global}\left(t=t_{initial},s^{-}\right)\cup O^{0}_{global}\left(t_{initial}\le t\le t_{final},s^{0}\right)\cup O^{+}_{global}\left(t=t_{final},s^{+}\right)\right\} \tag{9}$$

As shown in Fig. 3, The $O^{-}_{global}\left(t_{initial},s^{-}\right)$ are the ingress points, which satisfy $F\left(t_{initial},s^{-}\right)<0$. The $O^{0}_{global}\left(t_{initial}\le t\le t_{final},s^{0}\right)$ are the grazing points, which satisfy $F\left(t,s^{0}\right)=0$. The $O^{+}_{global}\left(t_{final},s^{+}\right)$ are the egress points, which satisfy $F\left(t_{final},s^{+}\right)>0$. Note that s* is a subset of *s*, which satisfies above-mentioned conditions. By setting the tangency function *F* equal to zero, the grazing points of the swept envelope *SE* of the tool is determined by the following equation:

$$F=\vec{V}\left(t,s^{0}\right)\cdot\vec{N}\left(t,s^{0}\right)=0 \tag{10}$$

The swept envelope of the cone is determined by substituting the speed vector and the surface normal of the cone into Eq. (10) and simplifying it as follows:

$$\left(\left(-A_{\mathrm{c}}\tan\beta+B_{\mathrm{c}}\right)\Omega_{y}+V_{x}\right)\cos\theta^{0}_{cone}+\left(\left(A_{\mathrm{c}}\tan\beta-B_{\mathrm{c}}\right)\Omega_{x}+V_{y}\right)\sin\theta^{0}_{cone}+V_{z}\tan\beta=0 \tag{11}$$

where $A_{\mathrm{c}}=e+r_{tc}\cos\beta+\left(1-c_{cone}\right)\left(l-r_{tc}\left(1-\sin\beta\right)\right)\tan\beta$, $B_{\mathrm{c}}=c_{cone}\left(l-r_{tc}\left(1-\sin\beta\right)\right)$, $\left[V_x\ V_y\ V_z\ 0\right]^{\mathrm{T}}=\left[M^{-1}_{def}\left[R_{rot}\ 0\right]^{-1}\partial T_{trans}/\partial t\right]^{\mathrm{T}}$, and $\left[\Omega_x\ \Omega_y\ \Omega_z\ 0\right]^{\mathrm{T}}=\left[M^{-1}_{def}\left[R_{rot}\ 0\right]^{-1}\Omega\right]^{\mathrm{T}}$. The Ω can be determined by Eq. (7). Eq. (11) is a simple trigonometric function. For non-degenerate cases, at a moment *t*, we can determine a corresponding θ^{0}_{cone} for each c_{cone} as follows:

$$\theta^{0}_{cone}=\sin^{-1}\left(c_1\Big/\sqrt{a_1^2+b_1^2}\right)-\xi_1 \tag{12}$$

where $a_1=\left(-A_{\mathrm{c}}\tan\beta+B_{\mathrm{c}}\right)\Omega_{y}+V_{x}$, $b_1=\left(A_{\mathrm{c}}\tan\beta-B_{\mathrm{c}}\right)\Omega_{x}+V_{y}$, $c_1=-V_{z}\tan\beta$, $\sin\xi_1=a_1\Big/\sqrt{a_1^2+b_1^2}$, and $\cos\xi_1=b_1\Big/\sqrt{a_1^2+b_1^2}$. Similarly, we use Eq. (10) to determine the swept envelope of the torus. We have

$$\left(-A_{\mathrm{ts}}\Omega_{\mathrm{y}}\cos\phi+B_{\mathrm{ts}}\Omega_{\mathrm{y}}\sin\phi+V_{\mathrm{x}}\sin\phi\right)\cos\theta^{0}_{torus}+\left(A_{\mathrm{ts}}\Omega_{\mathrm{x}}\cos\phi-B_{\mathrm{ts}}\Omega_{\mathrm{x}}\sin\phi+V_{\mathrm{y}}\sin\phi\right)\sin\theta^{0}_{torus}+V_{z}\cos\phi=0 \tag{13}$$

where $A_{\mathrm{ts}}=e+r_{\mathrm{ts}}\sin\phi$ and $B_{\mathrm{ts}}=l-r_{\mathrm{tc}}\left(1-\cos\phi\right)$. The corresponding θ^{0}_{torus} for each ϕ can be determined as follows:

$$\theta^{0}_{torus}=\sin^{-1}\left(c_2\Big/\sqrt{a_2^2+b_2^2}\right)-\zeta_2 \tag{14}$$

where $a_2=-A_{\mathrm{ts}}\Omega_{\mathrm{y}}\cos\phi+B_{\mathrm{ts}}\Omega_{\mathrm{y}}\sin\phi+V_{\mathrm{x}}\sin\phi$, $b_2=A_{\mathrm{ts}}\Omega_{\mathrm{x}}\cos\phi-B_{\mathrm{ts}}\Omega_{\mathrm{x}}\sin\phi+V_{\mathrm{y}}\sin\phi$, $c_2=-V_{z}\cos\phi$, $\sin\xi_2=a_2\Big/\sqrt{a_2^2+b_2^2}$, and $\cos\xi_2=b_2\Big/\sqrt{a_2^2+b_2^2}$. After determining the corresponding θ^{0}_{cone} and θ^{0}_{torus} at a point in time *t*, substituting the θ^{0}_{cone} and θ^{0}_{torus} into Eq. (1), we obtain the instantaneous grazing points at *t*. The grazing points of the swept envelope in between is represented as follows:

$$SE\left(t_{initial}\le t\le t_{final},s^{0}\right)=O_{global}\left(t,s^{0}\right) \tag{15}$$

5. ENVELOPE PROPERTY ANALYSIS AND MACHINING POTENTIAL FIELD (MPF)

At a point in time, $t_{initial}\le t_0\le t_{final}$, the instantaneous grazing points of the swept envelope degenerate into a curve (or curves), and is called *Swept Profile* of the tool at $t=t_0$ [11, 12]

$$SP\left(t_0,s^{0}\right)=SE\left(t_0,s^{0}\right) \tag{16}$$

Since the swept envelope is constructed by the swept profiles, we here analyze the swept profiles to understand the characteristics of the swept envelope.

5.1 Geometry Matching for Optimization Tool Orientation

When a tool touches the part surface at a contact point, the fundamental requirement for the surface point to be locally machinable is that the tool should not penetrate the part surface in the neighborhood of the point [13]. The local properties of the part and the tool near the contact point can be represented by their curvatures. The curvature of the swept profile *SP* of the tool at the contact point is represented as follows [14] (also shown in Fig 4):

$$\kappa_{SP}(t_0, s^0) = \left| \frac{\partial SP}{\partial s^0} \times \frac{\partial^2 SP}{\partial s^{0^2}} \right| \Big/ \left| \frac{\partial SP}{\partial s^0} \right|^3 \tag{17}$$

The osculating plane of the *SP* at the contact point is defined by two vectors, $\vec{N}_{SP}$ and $\vec{T}_{SP}$. Where $\vec{T}_{SP} = \frac{\partial SP}{\partial s^0} \Big/ \left| \frac{\partial SP}{\partial s^0} \right|$, $\kappa_{SP} \vec{B}_{SP} = \frac{\partial SP}{\partial s^0} \times \frac{\partial^2 SP}{\partial s^{0^2}} \Big/ \left| \frac{\partial SP}{\partial s^0} \right|^3$, and $\vec{N}_{SP} = \vec{B}_{SP} \times \vec{T}_{SP}$. The curvature of the part surface at the contact point along the osculating plane is determined as follows [Faux] (also shown in Figs 4 and 5):

$$\kappa_p = \frac{\kappa_n}{\vec{N}_{SP} \cdot \vec{n}_p} \tag{18}$$

where κ_n is the normal curvature of the part surface along $\vec{T}_{SP}$. The $\vec{n}_p$ is the surface normal at the contact point. By adjusting the tool orientation, we can make $\kappa_{SP} \geq \kappa_p$ to avoid the tool penetrating into the part surface in the neighborhood of the contact point. Furthermore, when $\kappa_{SP} = \kappa_p$, there is a higher chance that the swept profile *SP* is close to the part surface. Hence, we have a wider machining strip width. This orientation, which makes $\kappa_{SP} = \kappa_p$, is considered as the optimization tool orientation $\vec{A}_{optimal}^{orient}$. It is presented as follows:

$$\vec{A}_{optimal}^{orient}(t_0, s) = M_{global}(t_0) M_{def}(z_c) [0 \;\; 0 \;\; 1 \;\; 0]^T \tag{19}$$

where $\kappa_{SP}(t_0, s^0) = \kappa_p(t_0, s^0)$

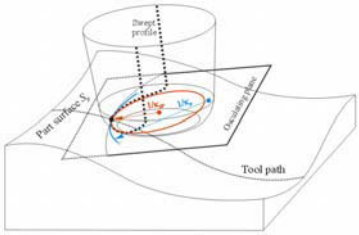

Fig. 4. Curvature matching.

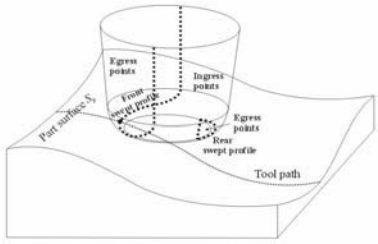

Fig. 5. Rear swept profile.

5.2 Machining Potential Field (MPF) and Optimal Cutting Direction

Some cut patterns, like window-frame or zig-zag, are mainly parallel to the surface boundary. But, some cut patterns, like *machining potential field* (*MPF*), are independent from the surface boundary but dependent on the surface geometry [21]. The directions of the tool paths follow the optimal cutting direction at each contact point. Thus, the optimal cutting direction as well as the optimal tool orientation needs to be determined for such boundary-independent cut patterns.

When $\kappa_{SP} = \kappa_p$, the tool can match the part geometry well, and obtains a wider machining strip width. However, rear gouge may hinder the adjustment of the tool orientation. Fig. 6(a) shows a tool is machining a spherical surface, whose principal curvatures are equal. The swept profile, which can be determined by Eq. (13), is a circle, which matches with the part surface on the osculating plane. If we reduce the curvature in the traversal direction, as shown in Fig. 6(b), the surface normal curvature κ_n along the traversal direction also reduces. However, since the curvature in the cutting direction does not change, the tool orientation, which is defined related to the part surface along the cutting direction, also does not change. Thus, the κ_{SP} does not change, either. Therefore, from Eq. (18) we know, to achieve $\kappa_n / (\vec{N}_{SP} \cdot \vec{n}_p) = \kappa_{SP}$, the tool (torus) should be moved closer to the part surface to make $\vec{N}_{SP} \cdot \vec{n}_p$ smaller. However, due to the unchanged of curvature in the cutting direction, the tool (torus) is unable to be moved closer to the part surface. (Otherwise, the tool will gouge the part surface in the rear portion.) Thus, it is usually impossible to make $\kappa_{SP} = \kappa_p$.

On the other hand, if we enlarge the curvature in the traversal direction, as shown in Fig. 6(c), the tool (torus) should be move away from the part surface to match the swept profile with the part surface. Since the curvature in the cutting direction does not change, it is possible to move the tool (torus) away from the part surface such that $\kappa_{SP} = \kappa_p$.

Therefore as shown in Fig. 6(b), to machine a surface point by generally following its maximum curvature direction usually has a high risk that the tool will gouge the part surface in its rear cutting edges. On the other hand, as shown in Fig. 6(c), to machine a surface point by following its minimum curvature direction is usually possible to make the tool

swept profile match with the part surface without the rear gouge. Thus, the machining strip width along the minimum curvature direction is larger than that along the maximum curvature direction.

Therefore, the minimum curvature direction can be considered as the optimal cutting direction $\vec{D}_{optimal}^{cut}$. The corresponding tool orientation that makes $\kappa_{SP} = \kappa_p$ becomes the optimal tool orientation $\vec{A}_{optimal}^{orient}$, i.e.,

$$\begin{aligned} &\vec{D}_{optimal}^{cut} = \text{minimum curvature direction} \\ &\vec{A}_{optimal}^{orient}(t_0, s) = M_{global}(t_0) M_{def}(z_c)[0 \;\; 0 \;\; 1 \;\; 0]^T \\ &\text{where } \kappa_{SP}(t_0, s^0, \vec{D}_{optimal}^{cut}) = \kappa_p(t_0, s^0, \vec{D}_{optimal}^{cut}) \end{aligned} \tag{20}$$

Please notice that Eq. (20) does not promise the result machining width is always the largest. However, it defines the best cutting direction and the best tool orientation such that the tool has the least risk to gouge the part surface and has the largest chance to match the part surface and to result the biggest machining width.

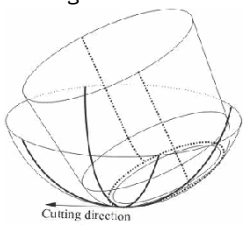

Fig. 6(a) Spherical surface.

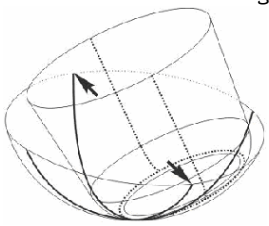
Fig. 6(b) Reducing traversal curvature.

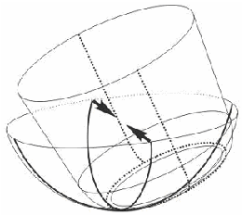
Fig. 6(c) Enlarging traversal curvature.

5.3 Rear Gouge Detection

Curvature matching only avoids the geometric interference near the contact point. The tool may gouge the part surface in the rear portion [15]. The machined geometry is the complement of the swept envelope of the tool and the swept envelope is constructed of the instantaneous swept profiles. When the tool gouges the part surface, the collision can be detected by the intersection between the swept profiles and the part surface. Therefore, the rear gouge only needs to be checked when the rear swept profiles exist. By observing the tool geometry, the rear swept profiles may happen only in the torus portion. We here, therefore, only analyze the properties of the swept profiles of the torus. The angle θ_{torus}^0 of the instantaneous swept profile is determined by Eq. (14). The θ_{torus}^0 has two different solutions for each ϕ^0 when $c_2/\sqrt{a_2^2+b_2^2} < 1$, no solution when $c_2/\sqrt{a_2^2+b_2^2} > 1$, and a doubled-solution when $c_2/\sqrt{a_2^2+b_2^2} = 1$. The doubled-solution $(\phi^0, \theta_{torus}^0)$ where $c_2/\sqrt{a_2^2+b_2^2} = 1$ can be considered as the extreme points of the instantaneous swept profiles. By substituting a_2, b_2, and c_2 into $c_2/\sqrt{a_2^2+b_2^2} = 1$, we obtain:

$$\begin{aligned} &\left(\left((l-r_{tc})\Omega_y + V_x\right)^2 + \left(-(l-r_{tc})\Omega_x + V_y\right)^2\right)\tan^2\phi^0 \\ &+ 2e\left(-(l-r_{tc})\Omega_y^2 - V_x\Omega_y - (l-r_{tc})\Omega_x^2 + V_y\Omega_x\right)\tan\phi^0 + \left(e^2\Omega_y^2 + e^2\Omega_x^2 - V_z^2\right) = 0 \end{aligned} \tag{21}$$

The solutions of ϕ^0 for the extreme points are determined as follows:

$$\left\{\phi_{\text{front}}^0, \phi_{\text{rear}}^0\right\} = \tan^{-1}\left(\frac{-b_{\tan} \pm \sqrt{Det}}{2a_{\tan}}\right) \tag{22}$$

where the determinant $Det = b_{\tan}^2 - 4a_{\tan}c_{\tan}$, $a_{\tan} = \left(\left((l-r_{tc})\Omega_y + V_x\right)^2 + \left(-(l-r_{tc})\Omega_x + V_y\right)^2\right)$, $b_{\tan} = 2e\left(-(l-r_{tc})\Omega_y^2 - V_x\Omega_y - (l-r_{tc})\Omega_x^2 + V_y\Omega_x\right)$, and $c_{\tan} = \left(e^2\Omega_y^2 + e^2\Omega_x^2 - V_z^2\right)$. When $Det \geq 0$, the ϕ_{front}^0 and ϕ_{rear}^0 coexist. By observing the tool motion, one of the solutions, $\left(\phi_{\text{front}}^0, \theta_{torus}^0\right)$, must be in the front side. If the other solution $\left(\phi_{\text{rear}}^0, \theta_{torus}^0\right)$ exists in its feasible domain, it must be in the rear side of the tool, as shown earlier in Fig 5.

On the other hand, when $Det < 0$, there is no extreme point for the instantaneous swept profiles. The front swept profile of the torus appears from the lower bound to the upper bound. However, there exist either two or no θ_{torus}^0 for Eq. (14). Therefore, there must exist one front swept profile in the front cutting edges and one rear swept profile in the rear cutting edges. Therefore, when the following condition holds, there exists the rear swept profile.

$$\begin{cases} Det < 0, \text{ or} \\ Det \geq 0 \text{ and } 0 \leq \left\{\phi_{\text{front}}, \phi_{\text{rear}}\right\} \leq (\pi/2 - \beta) \end{cases} \tag{23}$$

When the rear swept profile exists, it implies the rear egress points also exist. For any feasible tool orientation, the rear swept profile and the rear egress point can not intersect the part surface. Fortunately the existence of the rear swept profile and the rear egress points can be determined by Eq. (23). Therefore, the computational-expensive algorithms for checking the intersection between the tool and the part surface do not need to be executed unless the determinant *Det* satisfies either condition in Eq. (23).

5.4 Machinability of a Ruled Surface

The blades of impellers are widely represented by ruled surfaces in turbo machinery industry [16]. These blades are machined by the tool-side expectedly to finish machining by a single cutting pass. In tool-side machining, the cutting edges in the cone portion are designated to contact the part surface and remove the raw material. Therefore, we analyze the properties of the swept envelope of the cone here. The swept profile of the cone is presented in Eq. (12). The derivative of the tangential angle θ^0_{cone} with respect to c_{cone} is determined as follows:

$$\frac{\partial \theta^0_{cone}}{\partial c_{cone}} = -\frac{1}{\left(a_1^2 + b_1^2\right)}\left(\frac{c_1\left(a_1\dot{a}_1 + b_1\dot{b}_1\right)}{\sqrt{a_1^2 + b_1^2 - c_1^2}} + \left(b_1\dot{a}_1 - a_1\dot{b}_1\right)\right) \tag{24}$$

If $\left(\Omega_x, \Omega_y\right) = 0$, then $\dot{a} = 0$ and $\dot{b} = 0$. Therefore $\partial\theta^0_{cone}/\partial c_{cone} = 0$. If $V_z = 0$, then $c = 0$. The derivative is simplified as follows:

$$\frac{\partial \theta^0_{cone}}{\partial c_{cone}} = -\frac{b_1\dot{a}_1 - a_1\dot{b}_1}{a_1^2 + b_1^2} = \frac{\left(l - r_{tc}\left(1 - \sin\beta\right)\right)\sec^2\beta}{a_1^2 + b_1^2}\left(V_x\Omega_x + V_y\Omega_y\right) \tag{25}$$

When $\left(V_x, V_y\right)\cdot\left(\Omega_x, \Omega_y\right) = 0$, it follows that $V_x\Omega_x + V_y\Omega_y = 0$. Therefore the derivative $\partial\theta^0_{cone}/\partial c_{cone} = 0$. By observing Eqs. (24) and (25), we have the following conclusions. The z-rotational component Ω_z does not affect the value of the tangential angle θ^0_{cone}. When $\left(\Omega_x, \Omega_y\right) = 0$, the tangential angle θ^0_{cone} is a constant. When $V_z = 0$ and $\left(V_x, V_y\right)\cdot\left(\Omega_x, \Omega_y\right) = 0$, the tangential angle θ^0_{cone} is also a constant.

The tool motion, $\left(\Omega_x, \Omega_y\right) = 0$, happens in 3-axis machining. The swept profile on one side of the cone is a straight line. The swept envelope constructed of the intermediate swept profiles is a ruled surface. The machined surface, which is the complement of the swept envelope, is also a ruled surface. The tool motion, when $V_z = 0$ and $\left(V_x, V_y\right)\cdot\left(\Omega_x, \Omega_y\right) = 0$, is like machining a right circular cone. The swept profile of the tool cone is a straight line. The swept envelope and the machined surface are both ruled surfaces. These two tool motion conditions happen when a tool is side machining a developable ruled surface.

However, in general 5-axis ruled surface machining, the above-mentioned two conditions are generally not true. The tangential angle θ^0_{cone} varies along with different c_{cone} values. Thus, the swept profile of the cone is not a straight line. The swept envelope and the machined surface are not ruled surfaces. This runs counter to our expectation since turbo machinery industries commonly use tool-side to machine ruled surfaces. From our discussion above, however, it results in non-ruled machined surfaces.

6. COMPLEMENT OF SWEPT ENVELOPE FOR 5-AXIS TOOL PATH GENERATION

Previous sections discuss the swept envelope of a tool. Although the swept envelope defines the space that is occupied by the tool during the motion, many applications are more interested in the complement space of the swept volume, which represents the in-process workpiece. Here we develop an enhanced marching cubes algorithm with 3D G-buffer method to determine and represent the complement geometry from the swept envelope.

6.1 Three Dimensional G-buffer

When a tool moves, it creates a swept envelope in 3D space. If the swept envelope intersects the boundary of the in-process workpiece, the tool is removing the material. The geometric simulation and monitoring systems have to update the intersection information to G-buffer and reconstruct the in-process workpiece boundary.

How the G-buffer represents the in-process workpiece is first described here. Fig. 7 shows the relation of a line and the in-process workpiece. When the Line-A hits the in-process workpiece, the line intersects the boundary and enters the in-process workpiece. The line then intersects the boundary again and exits the in-process workpiece. If the in-process workpiece is more complicated, the line may intersect it more times. However, the intersection points are always paired-up. The space between the enter point and the exit point along the line represents the material-occupied

space. If we store the intersection points to an associated element of a G-buffer array by their order, the material-occupied space is the space between any paired-up intersection points [17].

Similarly when a line intersects the swept envelope created by a tool, the space between the paired-up points represents the material-removal space. We then check whether the material-removal space overlaps the material-occupied space or not. Then update the intersection information according to the relationship between the material-occupied space and the material-removal space. As Line-B shown in Fig. 7, when the material removal space is fully overlapped in the material occupied space, the enter point $P_{S\text{-}Ent}$ and the exit point $P_{S\text{-}Ext}$ of the swept envelope have to be added to the element between the enter point $P_{M\text{-}Ent}$ and the exit point $P_{M\text{-}Ext}$ of the in-process workpiece. The space between the paired-up points from $P_{M\text{-}Ent}$ to $P_{S\text{-}Ent}$ and from $P_{S\text{-}Ext}$ to $P_{M\text{-}Ext}$ becomes the new material-occupied space. When the material removal space is partially overlapped with the material occupied space, the new material-occupied space needs to be updated according to the material-removal space. The systems have to update the intersection information to the G-buffer in this way along x-, y-, and z- directions.

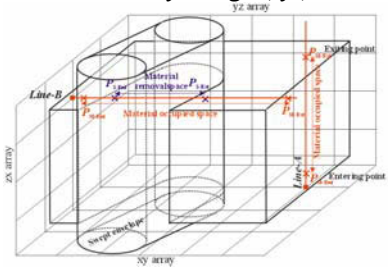

Fig. 7. G-buffer and material representation.

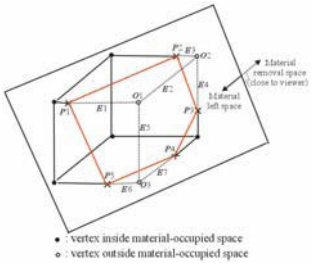

Fig. 8. Boundary construction.

6.2 Boundary Construction

The arrays in x-, y-, and z- directions virtually slice the in-process workpiece into small cubes. Some cubes are totally inside or outside the in-process workpiece. Some are in between and have intersection with the real boundary of the in-process workpiece. We are interested in these cubes and want to construct polygons to represent the boundary for such cubes.

A cube has 12 edges. Each edge corresponds to an element on G-buffer arrays, as shown in Fig. 8. By searching for the 12 corresponding elements, we can obtain the intersection points that range within the cubical edges. The intersection points on these 12 edges are used to construct the boundary of the in-process workpiece [18]. Note, there may exist none, one, or multiple intersection points on each cubical edge. The current marching cubes algorithms cannot construct a polygon when multiple intersection points exist on a cubical edge [18, 19]. Therefore, we here develop an enhanced algorithm to overcome such limitation.

When a tool removes a portion of material from a small cube, it is similar to using a knife straightly cut through cubical jell-O, as shown in Fig. 8. When any jell-O is cut out, the knife must cut through the cube and separate at least one corner (vertex) from the others. The intersection points are always edge-connected to the corner(s) that is in the material-removal space. Therefore, the intersection points are accessible from the corner(s) that is in the material-removal space. This statement also is true when the jell-O is straightly cut more than one time.

Hence, when we construct the polygons for a cube, we first identify the cubical corners that are in the material-removal space. Then, we depart from such corners, and travel along the connected edges. If the traveling reaches to the end of an edge because there is no intersection point on the edge, we continue to travel to the two connected edges. The traveling keeps continuous until it hits either intersection points on the edges or the corners that have been reached before. All intersection points that are hit from the same departing corner construct a polygon.

For instance, as shown in Fig. 8, the corners C1, C2, and C3 are in the material-removal space. We arbitrarily depart from C1 and travel along edges E1, E2, and E5. In E1, we hit P1 and stop. In E2, we hit another corner C2 and continuously travel along E3 and E4. In E3, we hit P2 and stop. In E4, we hit P3 and stop. When travel along E5, we hit another corner C3 and continuously travel along E6 and E7. In E6, we hit P5 and stop. In E7, we hit P4 and stop. Therefore, departing from C1, we totally hit P1, P2, P3, P4, and P5. These five intersection points are used to construct a polygon, as Fig 8 shown.

If there are any corners of a cube, which are in the material-removal space and are not hit yet, we similarly depart from these corners and travel along the connected edges to search for intersection points. Then, we construct a polygon by the found intersection points. This procedure continues until there is no non-hit corner left in the material-removal space. The procedure is summarized as follows:

Polygon Construction Algorithm:

Begin
Step_0 {Corners} ← Identify corners in material-removal space
Step_1 WHILE {Corners} ≠ {∅}
Empty {Intersection Points}
Pick a corner from {Corners}, mark the corner as hit, and remove it from {Corners}
Depart from the corner and travel along the connected edges
Step_2 IF hit an intersection point on the traveling edge
Put the hit intersection point in {Intersection Points}; STOP.
ELSE IF {the corner has been hit before} then STOP.
ELSE Mark the corner as hit and remove it from {Corners}
Travel along the connected edges
GOTO Step_2:
End-IF
End-IF
Step_3 Construct a polygon by {Intersection Points}
End-WHILE
End-Begin.

7. COMPUTER IMPLEMENTATION AND EXAMPLES

The developed algorithm has been implemented on personal computers by using the Matlab, visual C++ programming languages and OpenGL for result calculation and display. Figure 9 shows the swept profiles and the swept envelope of a tool that is machining a sculptured surface along a tool path with 9 forward steps. In Fig. 9, only few intermediate swept profiles are displayed. In addition, the ingress surface in the initial location, the egress surface in the final location, and half of the swept envelope in between are illustrated. Figure 10 shows the machined part geometry that is determined by the complement of the swept envelope from the raw material.

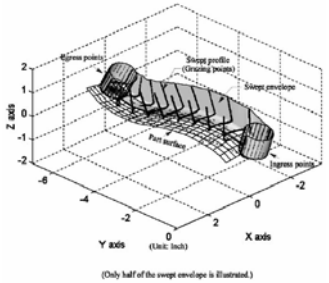

Fig. 9. Swept envelope of a tool.

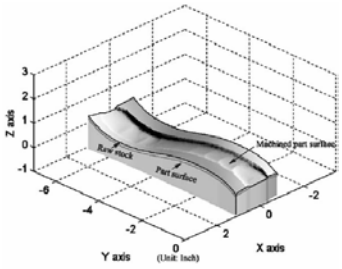

Fig. 10. Machined geometry.

Figure 11 shows a tool is machining a cylindrical surface along different cutting directions. Figures 11(a), (b), and (c) show the swept profiles when the tool is machining a surface point deviating away from the minimum curvature direction with 0°, 60° and 90°, respectively. The tool is oriented such that the curvature of the swept profile matches with the curvature of the part surface on the osculating plane. Due to the dramatic tool rotation, the rear swept profiles occur when cutting along 60° and 90° direction, as Figs. 11 (b) and 11(c) show. The tool motion in Fig. 11(b) satisfies the first determinant condition in Eq. (23). The tool motion in Fig. 11(c) satisfies the second determinant condition.

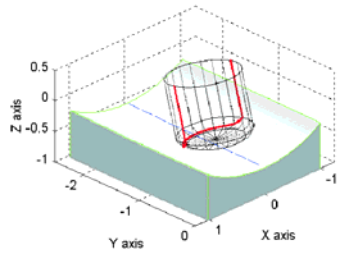

Fig. 11(a). Swept profile.

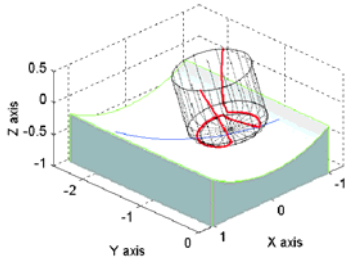

Fig. 11(b). Rear swept profile.

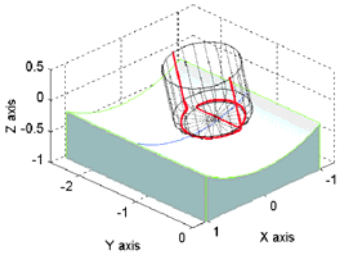

Fig. 11(c). Front and rear swept profile.

Figure 12 shows a free-form surface, which will be machined by the machining potential field (MPF) method [21]. The circular patches represent the relation of the machining strip width and the cutting direction at each surface point, which is determined by Eq. (19). The arrows represent the optimal cutting directions, which are determined by Eq. (20). Machining potential field (MPF) method generates tool paths by following the optimal cutting directions as much as possible till the machining efficiency becomes less than a given threshold. Then repeat the same procedures on remain areas until the generated tool paths cover the whole part surface. Refer readers to [21] for details. Figure 13 shows the generated tool paths. Due to the tool paths generally following the optimal cutting direction, the total tool path length by MPF method is 50% shorter than by the traditional zig-zag methods, even both methods generate almost the same numbers of the tool paths [21].

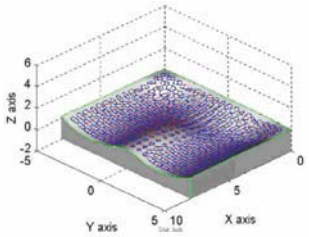

Fig. 12. Machining Potential Field.

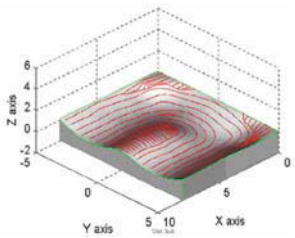

Fig. 13. MPF tool paths.

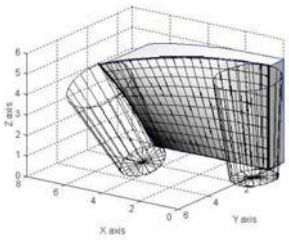

Fig. 14. Tool-side machining.

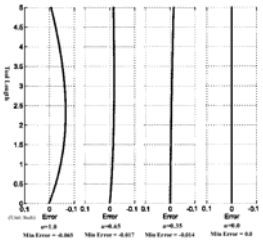
Fig. 15. Machining errors.

Figure 14 shows a tool is machining a ruled surface. The tool is traditionally positioned to contact with the two directrices of the ruled surface. Due to the simultaneous translational and rotational movement, the swept profiles of the tool are not straight lines and do intersect with the ruled surface. Figure 15 shows the machining errors between the swept profiles and the ruled surface. By observing Fig. 15, the minimum machining error is proportional to the twist of the ruled surface. The results from Fig. 15 also show that it is generally impossible to machine a ruled surface by a single cutting pass.

Fig 16 shows a milling tool that is cutting a raw block. The simulation system first initializes the boundary of the raw block by setting paired-up enter and exit points on G-buffer arrays equal to the raw block dimension along x-, y-, z-directions. The system then constructs the boundary of the raw block by using the developed algorithm.

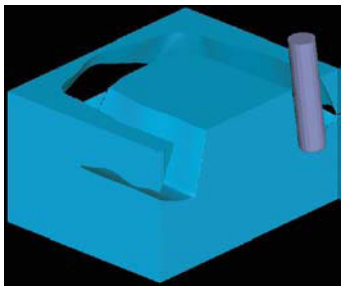
Fig. 16. Simulated in-process workpiece.

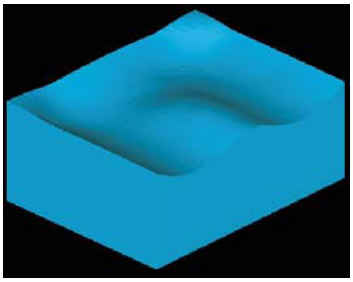
Fig 17. Simulated machined surface.

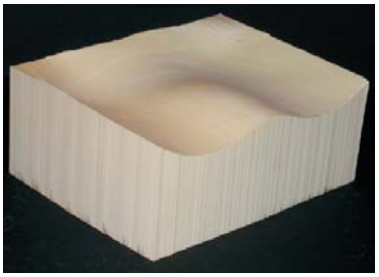
Fig. 18. Machined surface.

During simulation, the tool swept envelope is first determined. The system updates the intersection points between the in-process workpiece and the swept envelope to G-buffer arrays. Then, the simulation system reconstructs the boundary based on the intersection information stored on the G-buffer arrays along x-, y-, and z-direction. Fig 17 shows the finished simulated surface by MPF tool paths. Fig. 18 shows the real machined part.

8. CONCLUSIONS

This paper presents the swept envelopes for milling tools. Tool geometry and tool motion (including translation and rotation movements) are analyzed to support the determination of the swept envelopes. It further addresses the construction of the complement of the swept envelope, which represents the in-process workpiece during machining processes, and discusses the application of the swept envelope in geometric simulation systems. The results show that the closed form solutions of the swept envelopes are possible to be expressed for axial symmetric tools. By analyzing the swept envelope, we further conclude that the optimal tool orientation takes place when the curvature of the swept profile matches with the part surface. Rear gouge needs to be detected only when the rear swept profile occurs. The

determinant conditions are also derived. The properties of the swept envelope also deduce that it is generally impossible to tool-side machining a ruled-surface by a single cutting pass.

9. ACKNOWLEDGEMENT
This work was partially supported by the NSF Grant (DMI-0300297, DMI-0553310) and ARO (Grant #W911NF-04-D-0003) to North Carolina State University. Their support is greatly appreciated.

10. REFERENCES

[1] Pottmann, H. and Peternell, M., Envelopes - computational theory and applications, In B. Falcidieno, editor, *Spring Conference on Computer Graphics* 2000, pp 3-23.
[2] Abdel-Malek, K., Blackmore, D., and Joy, K., Swept volumes: Foundations, perspectives, and applications. *International Journal of Shape Modeling*, to appear. http://www.engineering.uiowa.edu/~amalek/papers/swept-volume-review.pdf
[3] Abrams, S. and Allen, P. K., Computing swept volumes, *Journal of Visualization and Computer Animation*, Vol. 11, No. 2, 2000, pp 69-82.
[4] Kral, I.H., *Numerical Control Programming in APT*, Prentice-Hall, 1986.
[5] Chiou, C. J. and Lee, Y.-S., Swept surface determination for five-axis numerical control machining, *International Journal of Machine Tools and Manufacturing*, Vol. 42, 2002, pp 1497-1507.
[6] Wang, W. P. and Wang, K. K., Geometric modeling for swept volume of moving solids, *IEEE Computer Graphics and Applications*, Vol. 6, No. 12, 1986, pp 8-17.
[7] Dhande, S. G., Karunakaran, K. P. and Misra, B. K., Geometric modeling of manufacturing processes using symmetric and computational conjugate geometry, *Journal of Engineering for Industry*, Vol. 117, No. 3, 1995, pp 288-296.
[8] Chiou, C. J., and Lee, Y. S., Optimal tool orientation for five-axis tool-end machining by swept envelope approach, *Journal of Manufacturing Science and Engineering*, Vol.127, No. 4, 2005, pp 810-818.
[9] Blackmore, D., Leu, M.C. and Wang, L. P., The swept-envelope differential equation algorithm and its application to NC machining verification, *Computer-Aided Design*, Vol. 29, No. 9, 1997, pp 629-638.
[10] Chiou, C. J. and Lee, Y. S., A shape-generating approach for multi-axis machining G-buffer models, *Computer-Aided Design*, Vol. 31, No. 12, 1999, pp 761-776.
[11] Roth, D., Bedi, S., Ismail, F. and Mann, S., Surface swept by a toroidal cutter during 5-axis machining, *Computer-Aided Design*, Vol. 33, No. 1, 2001, pp 57-63.
[12] Chung, Y. C., Park, J. W., Shin, H. and Choi, B. K., Modeling the surface swept by a generalized cutter for NC verification, *Computer-Aided Design*, Vol. 30, No. 8, 1998, pp 587-594.
[13] Yoon, J. H., Pottmann, H, and Lee, Y. S., Locally optimal cutting position for 5-axis sculptured surface machining, *Computer-Aided Design*, Vol. 35, No. 1, 2003, pp 68-81.
[14] Faux, I. D. and Pratt, M. J., *Computational Geometry for Design and Manufacturing*, John Wiley & Sons, New York, 1979.
[15] Lee, Y.S., Admissible Tool Orientation Control of Gouging Avoidance for 5-Axis Complex Surface Machining, *Computer-Aided Design*, Vol. 29, No. 7, 1997, pp. 507-521.
[16] Chiou, C. J., Accurate tool position for 5-axis ruled surface machining by swept envelope approach, *Computer-Aided Design*, Vol. 36, No. 10, 2004, pp 967-975.
[17] Fussell, B., Jerard, R., Hemmet, J., Modeling of Cutting Geometry and Forces for 5-axis Sculptured Surface Machining, *Computer-Aided Design*, Vol. 35, No. 4, 2003, pp333-346.
[18] Lorensen, W., Cline, H., Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *ACM Computer Graphics*, Vol. 21, No. 4, 1987, pp 163-170. (Siggraph '87 Conf. Proc.)
[19] Allamandri, F., Cignoni, P., Montani, C. and Scopigno, R., Adaptively Adjusting Marching Cubes Output to Fit a Trilinear Reconstruction Filter, *Visualization in Scientific Computing '98*, pp 25-34.
[20] Hibbeler, R.C., *Engineering Mechanics - Dynamics*, Macmillan Publishing Company, New York, 1989.
[21] Chiou, C. J. and Lee, Y.-S., A machining potential field approach to tool path generation for multi-axis sculptured surface machining, *Computer-Aided Design*, Vol. 34, No. 5, 2002, pp 357-371.