# Flank Millable Surfaces Generated via Polynomial Composition

Chenggang Li[1], Stephen Mann[2] and Sanjeev Bedi[3]

[1]University of Waterloo, cgli@engmail.uwaterloo.ca
[2] University of Waterloo, smann@uwaterloo.ca
[3]University of Waterloo, sbedi@uwaterloo.ca

**ABSTRACT**

Flank milling is a high efficiency machining method in 5-axis machining and is used in the machining of ruled surfaces. Li et al. [1, 2] developed methods to design double curved surfaces for flank milling. Those methods are restricted to cylindrical cutting tools. In this paper, Li's method is generalized to tools of revolution. Based on polynomial composition [3], a method to model a grazing curve with NURBS on a surface of revolution tool is developed and is tested with cylindrical tools, conical tools and barrel tools. The NURBS grazing curves are combined to generate the flank millable surface. Examples are given to demonstrate the proposed flank millable surface design method.

**Keywords:** curved surface design; flank milling; surface numerical analysis; surface error control.

## 1. INTRODUCTION

Flank milling is a 5-axis NC machining method. In flank milling, the side of cutter touches the surface, and the stock in front of the cutter is removed away. High material remove rate and good surface finish can be obtained by this machining technique. Today, flank milling is mainly used in aerospace and fluid flow application in mechanical engineering and focused on ruled surface machining. Different flank milling tool positioning methods have been developed [5-10], which attempt to reduce the deviation between the designed ruled surface and the generated machined surface.

In general engineering applications, curved surfaces are used and must be machined. Applying flank milling techniques in curved surface machining to increase the efficiency of machining has been proposed. Elber and Fish [11] developed a piecewise ruled surface approach to machine freeform surfaces. In their method, the freeform surface is divided into conjunctive ruled surfaces. Each small surface (a ruled surface) is machined with one of the developed flank milling methods. A long tool path is needed for their method. Li et al. [1-2], [12] also developed curved surface flank milling methods that are used in machining surfaces like impellers and turbine blades. In these methods, a machined surface that is described by a grazing surface (or a swept surface) is used as the design surface. Because the grazing surface is composed of a bundle of discrete points and is hard to use in engineering applications, a NURBS surface has been developed to represent the grazing surface. This NURBS surface is called a *flank millable* surface. Using this surface, the machined surface can match the designed flank millabe surface accurately. Three different ways to build the flank millable surface were presented in their papers.

Li et al.'s methods to generate the flank millable surfaces are based on cylindrical cutting tools and still need to be extended to conical tools, barrel tools and other tools of revolution. One such extension is probed in this paper, with polynomial composition [3] being used to generate the grazing curves in this study.

The process to generate the approximate grazing curve (a contact curve between the tool and the design surface) in the new method uses polynomial composition, while in the earlier methods the approximating grazing curve is created by extending a circle in 3D. In our new method, the polynomial curve at each tool position exactly lies on the tool surface, while in the NURBS approximation to the grazing curve in the earlier methods might not lie on the tool surface. Our new method can be used with any tool of revolution described by a polynomial profile curve, while the earlier methods only work for cylindrical tools.

The organization of this paper is as follows: The mathematical background for this paper is given in the next section, focusing on the ideas of grazing point generation, Bézier curves, Bézier surfaces and polynomial composition. The method to design the flank millable surface is presented in section 3. Examples to demonstrate the proposed flank millable surface design method is given in section 4. Some discussion of the method and variations on the method are given in section 5, and the paper concludes in section 6.

## 2. MATH BACKGROUND

In this section, we review some background material about grazing point generation, Bézier curves, tensor-product Bézier surfaces, and polynomial composition. For more details on Bézier curves and tensor product surfaces, see [13].

### 2.1 Grazing Point Generation

Grazing points are imprint points left on the machined surface when a cutting tool moves along the feed direction [10]. The net of evenly distributed grazing points on the machined surface can be used to describe the machined surface itself. In flank milling, the side of the cutter touches the surface and is rolled along guiding curve directions. The stock in front of the cutter is machined away. The effective contact between the cutter and the surface at each tool position is called a *grazing curve* and is composed of a sequence of grazing points. These grazing points are located not only on the machined surface, but also on the cutting tool surface. Based on this property, these grazing points can be easily computed at each tool position. The grazing points are the points on the cutting tool surface whose motion directions are perpendicular to the corresponding normals of the cutting tool surface.

To calculate these points on a cylindrical surface, let the tool axis be designed with a vector $T_{axis}$ and consider a point on the tool axis $P$. The velocity of tool at $P$ is $V$. The radius of the cutting tool is $R$. The grazing point corresponding to the point $P$ is $G$ and is given by

$$G = P + \frac{V * T_{axis}}{| V * T_{axis} |} R \,. \tag{1}$$

To compute grazing points on a general surface of revolution, see [18]. Different grazing points along the tool axis direction are calculated and composed of a grazing curve. Different grazing curves at each tool position are combined in sequence to build a grazing surface. This surface can be used to represent the machined surface closely.

### 2.2 Bézier Curves

A parametric Bézier curve is defined by

$$B(t) = \sum_{i=0}^{n} P_i B_i^n(t) \,,$$

where the $P_i$ are points in space known as *control points*, and $B_i^t(t) = \binom{n}{i}(1-t)^{n-i} t^i$ are the *Bernstein polynomials*, which form a basis for degree n polynomials. The parameter $t$ is a real number in the domain; as we vary $t$ from 0 to 1, we trace a curve starting at $P_0$ and ending at $P_n$. An example of a cubic Bézier curve is shown in Fig.1(a).

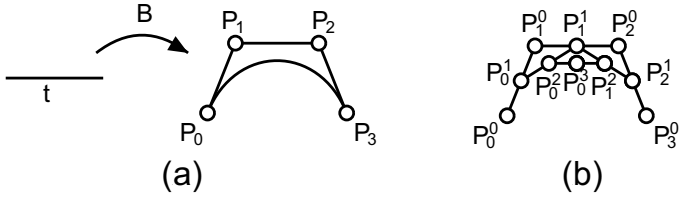


Fig. 1: (a) A cubic Bézier curve. (b) de Casteljau evaluation of Bézier curve.

An affine combination is a linear combination whose coefficients sum to 1; e.g., $aP_0 + bP_1$ is an affine combination of $P_0$ and $P_1$ if $a + b = 1$. Note that the degree $n$ Bernstein polynomials sum to 1, so a Bézier curve is an affine combination of its control points.

We can evaluate a Bézier curve using repeated affine combinations via de Casteljau's algorithm as follows. Let $P_i^0 = P_i$. Then to evaluate $B(t)$ at $t = \overline{t}$, we compute $P_i^1 = (1-\overline{t})P_i^0 + \overline{t}P_{i+1}^0$ for $i = 0,\cdots,n-1$. Repeating for $j = 2,\cdots,n$, we compute

$$P_i^j = (1-\overline{t})P_i^{j-1} + \overline{t}P_{i+1}^{j+1} \tag{2}$$

for $i = 0,\cdots,n-j$. We then have $B(\overline{t}) = P_0^n$. See Fig. 1 (b).

The blossom of a degree $n$ polynomial curve $B$ is the unique $n-$variate function $b(t_1,\cdots,t_n)$ where

- $b$ is symmetric:
$$b(t_1,\cdots,t_n) = b(t_{\sigma(1)},\cdots,t_{\sigma(n)}),$$
where $\sigma$ is any permutation of $(1,\cdots,n)$;
- $b$ is multi-affine:
$b(t_1,\cdots,t_{i-1},au_i+(1-a)v_i,t_{i+1},\cdots,t_n) = ab(t_1,\cdots,t_{i-1},u_i,t_{i+1},\cdots,t_n) + (1-a)b(t_1,\cdots,t_{i-1},v_i,t_{i+1},\cdots,t_n)$,
for all i.
- $b$ agrees with $B$ on the diagonal:
$$b(t,\cdots,t) = B(t).$$

For Bézier curves, the control points have "nice" blossom value:

$$P_i = b(\underbrace{0,\cdots 0}_{n-i},\underbrace{1,\cdots,1}_{i}).$$

We can evaluate the blossom using a variation of the de Casteljau algorithm, where we use a different blossom argument at each level of the de Casteljau evaluation. I.e., to evaluation $b(t_1,\cdots,t_n)$, in place of (2) we use

$$P_i^j = (1-t_j)P_i^{j-1} + t_jP_{i+1}^{j+1}.$$

### 2.3 Tensor-product Bézier Surfaces

A tensor-product Bézier surface is defined by

$$S(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} P_{i,j}B_i^n(u)B_j^m(v), \tag{3}$$

where the $P_{i,j}$ are the control points for the surface and the $B_i^n(u)$ and $B_j^m(v)$ are Bernstein polynomials. As we vary $u$ and $v$ over the $[0,1]\times[0,1]$ domain, a surface patch is traced out. An example of a bicubic tensor-product Bézier patch is shown in Fig. 2.

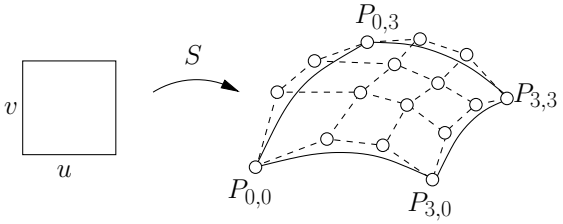


Fig. 2: A tensor-product Bézier patch.

We can also blossom a tensor-product surface $S(u,v)$, giving an $n+m$ variate function $s(u_1,\cdots,u_n;v_1,\cdots v_m)$ where

- $s$ is symmetric in the $u_i$ and $v_i$, but interchanging a $u_i$ argument with a $v_i$ will in general change the value of $s$.
- $s$ is multi-affine in both the $u_i$ and $v_i$.
- $s$ agrees with $S$ on the diagonal:
$$s(u,\cdots,u;v,\cdots v) = S(u,v).$$

### 2.4 Polynomial Composition

Given a Bézier curve $F$ whose range is the domain of a tensor product Bézier surface $G$, we can compute the control points of $H(t) = G(F(t))$. Various methods for efficient computing the control points of $H$ are available [4], [5]. Here we rederive the special case of a curve composed with a tensor-product surface.

**Theorem**: Let $F : R \to R^2$ be a degree $n$ Bézier curve and let $G : R^2 \to R^3$ be a degree $l \times m$ tensor-product surface patch with blossom $g(u, v)$. Here we are considering the range $R^2$ of $F$ to be the domain of $G$. Let $F_i$ be the control points of $F$, with $F_i = (F_i^u, F_i^v)$ giving the decomposition of $F$'s control points into coordinates in the range. Then the control points of $H(t) = G(F(t))$ are

$$H_i = \sum_{I,J,|I|+|J|=i} C(i_1,\cdots,i_l,j_1,\cdots,j_m) B^{(l+m)n}_{|I|+|J|}(t) g(F^u_{i_1},\cdots,F^u_{i_l};F^v_{j_1},\cdots,F^v_{j_m}) . \quad (4)$$

**Proof:** Our goal is to find the control points of

$$H(t) = \sum H_i B_i^{l+m}(t) = G(F(t)) = G(\sum F_i^u B_i^n(t); \sum F_i^v B_i^n(t)) . \quad (5)$$

Using the blossom of $G$, we find

$$H(t) = g(\sum F^u_{i_1} B^n_{i_1}(t), \sum F^u_{i_2} B^n_{i_2}(t), \cdots, \sum F^u_{i_l} B^n_{i_l}(t); \sum F^v_{j_1} B^n_{j_1}(t), \sum F^v_{j_2} B^n_{j_2}(t), \cdots, \sum F^v_{j_m} B^n_{j_m}(t)) .$$

Since $g$ is multiaffine and since the Bernstein polynomials sum to 1, each of the sums can be brought to the outside:

$$H(t) = \sum_{i_1,\cdots,i_l,j_1,\cdots,j_m} B^n_{i_1}(t)\cdots B^n_{i_l}(t) B^n_{j_1}(t)\cdots B^n_{j_m}(t) g(F^u_{i_1},\cdots,F^u_{i_l};F^v_{j_1},\cdots,F^v_{j_m}) .$$

The product of Bernstein polynomials $B^n_{i_1}(t)\cdots B^n_{i_k}(t)$ is equal to $C(i_1,\cdots,i_k)B^{kn}(t)$, where

$$C(i_1,\cdots,i_k) = \binom{n}{i_1}\binom{n}{i_2}\cdots\binom{n}{i_k} \Big/ \binom{k_n}{\sum_{j=1}^{k_n} i_j} .$$

Let $I = (i_1,\cdots,i_l)$ and $|\, I \,| = \sum i_k$ and let $J = (j_1,\cdots,j_m)$ and $|\, J \,| = \sum j_k$. We can write

$$H(t) = \sum_{I,J} C(i_1,\cdots,i_l,j_1,\cdots,j_m) B^{(l+m)n}_{|I|+|J|}(t) g(F^u_{i_1},\cdots,F^u_{i_l};F^v_{j_1},\cdots,F^v_{j_m}) . \quad (6)$$

Since the Bernstein polynomials form a basis for degree $n$ polynomials, we can now find an expression for the control points of $H$ by equating the coefficients of the Bernstein polynomials in (6) with (5), giving us the desired result, (4), which concludes the proof.

Rather than compute each control point of $H$ independently, because of the indexing it is easier to write code that computes all of the control points at once. Fig. 3 is pseudo-code for doing so. While this isn't the most efficient way to compute the control points of $H$, it suffices for the purposes described in this paper. In this code, the two for loops are iterating over the sums in (6); the function TensorProductBlossom evaluates the blossom of a tensor-product surface, using the two arrays of values, $F_I^u$ and $F_J^v$, as the parameters to the blossom, where $F_I^u$ is the array consisting of the $u$ values of the control points of $F$ indexed by the elements of the array $I$. Note that the values of $C(I,J)$ are such that the resulting expressions for the $H_i$ are affine combinations.

```
function H = tpcompose ( l , m , G , n , F )
    For I = (i_1,...,i_l) = (0,...,0) to (n,...,n)
        For J = (j_1,...,j_m) = (0,...,0) to (n,...,n)
            R = TensorProductBlossom ( l , m , G , F_I^u , F_J^v );
            H_{|I|+|J|} = H_{|I|+|J|} + R * C(I,J);
        End
    End
```

Fig. 3: Pseudo-code for computing the control points of the composition $H = G(F)$.

## 3. GENERATION OF FLANK MILLABLE SURFACE

The design of the flank millable surface starts from two guiding curves and a given cutting tool. Our method positions the tool at several locations along the guiding curves. Different tool positioning methods result in different machined surfaces. In this study, Bedi et al's tool positioning method [10] is used to generate the tool path, and polynomial composition [3] is used to approximate each grazing curve with a B-Spline curve that lies on the tool surface. A few selected B-Spline curves are used to build the flank millable surface. The parametric error measurement method [14] is used to evaluate the error between the generated flank millable surface and the given grazing surface. The steps to achieve this flank millable surface with tool of revolution are given below.

At each tool position, the tool axis can be sampled with discrete points $P_i$ as Fig. 4 (Step 1) shows. The velocity, $V_i$, of each point $P_i$ can be calculated [10] and be used in surface design. The following steps give our algorithm

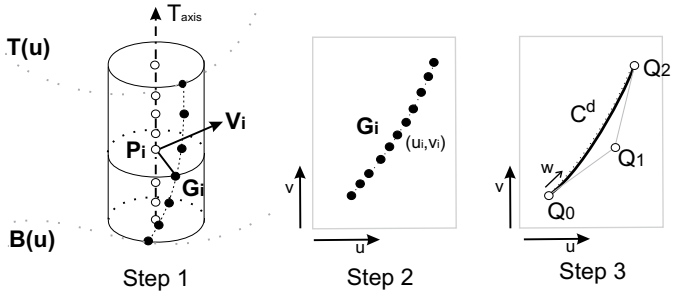


Fig. 4: Grazing points and Bézier curve.

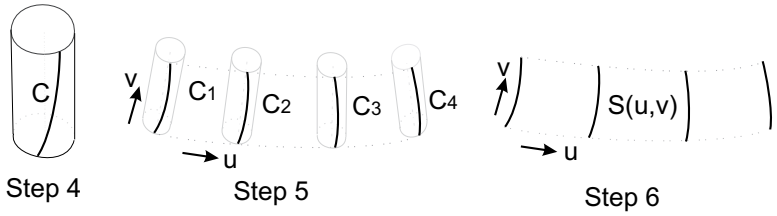


Fig. 5: Bézier curves and NURBS surface.

Step 1: For a given tool position and for each $P_i$ on the tool axis with velocity $V_i$, use equation (1) to compute and find its corresponding grazing point $G_i$. (Equation (1) is for a cylinder; for general surface of revolution, see [18].)

Step 2: For each $G_i$, compute $(u_i, v_i) = R^{-1}(G_i)$, where $R$ is the surface of the revolution tool, and $R^{-1}(G_i)$ represents the inverse of $G_i$ in the domain of the given tool of revolution.

Step 3: Use least squares curve fitting to find a Bézier curve $C^d$ to match points $(u_i, v_i)$.

Step 4: Using polynomial composition (section 2.4) to find a Bézier curve $C$ on the tool surface corresponding to the curve $C^d$.

Step 5: Repeat steps 1~4 to compute $N+1$ curves $C_j$ at $N+1$ tool positions.

Step 6: Construct a tensor product B-Spline surface $S(u,v)$ with $S(u_j, v) = C_j(v)$, where $u_j = j/N$.

Although the surface of the revolution $R$ is exactly described by a rational tensor product surface, $R$ can still be approximated with a high order polynomial using a tensor product Bézier surface, with the control points in one parametric direction representing the curve being revolved, and the control points in the other parametric direction

forming a cubic approximation to a circle [17]. By using a poly approximate the surface of revolution $R$, we can use polynomial composition in step 4.

Using the steps given above, the flank millable $S(u,v)$ can be obtained.

A few notes. $N$ is the number of tool positions that are used in flank millable surface generation. The selection of $N$ is based on surface error checking. By using more tool positions, the surface error can be effectively controlled.

Step 6 is the last step to build the flank millable surface. The details to apply this step to generate the flank millable surface are described in [1-2].

$u_j$ is the parametric value assigned to each tool position along the guiding curve direction. In this paper, we equally spaced the $u_j$. Other methods, like the chord length and the centripetal spacing [16], can also be used to compute $u_j$.

Although the proposed method is demonstrated with Bedi et al.'s tool positioning method, it can also be used with other tool positioning methods.

## 4. EXAMPLES

Examples are given in this section to test and demonstrate the proposed flank millable surface generation method. The design starts from two guiding curves and a cutting tool. The guiding curves used in this flank millable surface design are taken from Li et al.'s paper [1]. The control points of the guiding curves are listed in Table 1. The knot vector for both curves is [0,0,0,1,1,1] and the degree of the curves is 2.

| | T0 | T1 | T2 | B0 | B1 | B2 |
|---|---|---|---|---|---|---|
| $x$ | 75 | 30 | 0 | 60 | 30 | 15 |
| $y$ | 15 | 30 | 60 | 0 | 30 | 75 |
| $z$ | -5 | -5 | -5 | -45 | -45 | -45 |
| $w$ (weight) | 1 | 1 | 1 | 1 | 1 | 1 |

Tab. 1: Control points for guiding curves [mm].

Three types of cutting tools, a cylindrical tool, a conical tool and a barrel tool, are used to generate the flank millable surface for our tests. The geometries of these tools are given in Fig. 6.

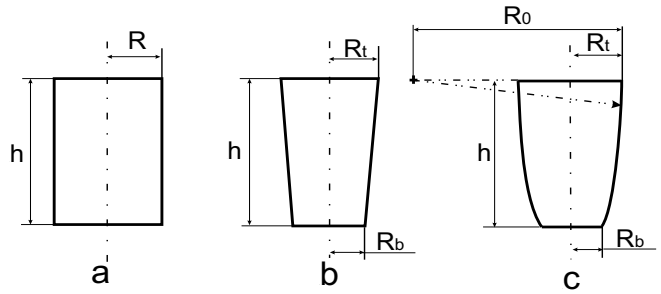


Fig. 6: Geometries of cutting tools. $R = 5mm$, $h = 50mm$, $R_t = 7mm$, $R_b = 4mm$, $R_0 = 418.17mm$. a). a cylindrical tool. b). a conical tool. c). a barrel tool.

For each cutting tool, Bedi et al.'s tool positioning method is used to determine the tool path and the method described in section 3 is used to construct the flank millable surface. Three grazing curves ( $N = 2$ ) located at $u = 0$, $u = 0.5$ and $u = 1$ along the guiding curve directions are selected to build the surface. For each grazing curve, the domain of

each grazing point, $(u_i, v_i)$, is calculated (Step 2). A quadratic 2D Bézier curve is fit to these discrete points (Step 3, $C^d$). In Step 4, a Bézier curve with 9 control points results for the cylindrical tool and the conical tool, and 13 control points for the barrel tool (curve $C$). This Bézier curve is used to approximate its corresponding grazing curve. By the same procedure, the remaining Bézier curves can be constructed (Step 5). For the estimates, three Bézier curves have been obtained after Step 5. With these Bézier curves, Step 6 creates a tensor product B-Spline surface with 9 by 3 control points (degree 8 by 2) for the cylindrical and conical tools, and 13 by 3 control points (degree is 12 by 2) for the barrel tool. This surface is used to approximate the grazing surface (or the machined surface). The deviation between the B-Spline surface and the grazing surface is evaluated with the parametrical error measurement method [16], and the evaluation results for different cutting tools are plotted in Fig. 7.

From Fig. 7, it is seen that the maximum surface error for the cylindrical tool is around 0.0235mm, for the conical tool is around 0.0244mm and for the barrel tool is around 0.024mm. These errors satisfy most of engineering applications. If any of the maximum errors exceeds the tolerance specified by the user, more control points can be added to the grazing curve direction and/or additional grazing curves can be used in the feed direction to reduce the surface error.

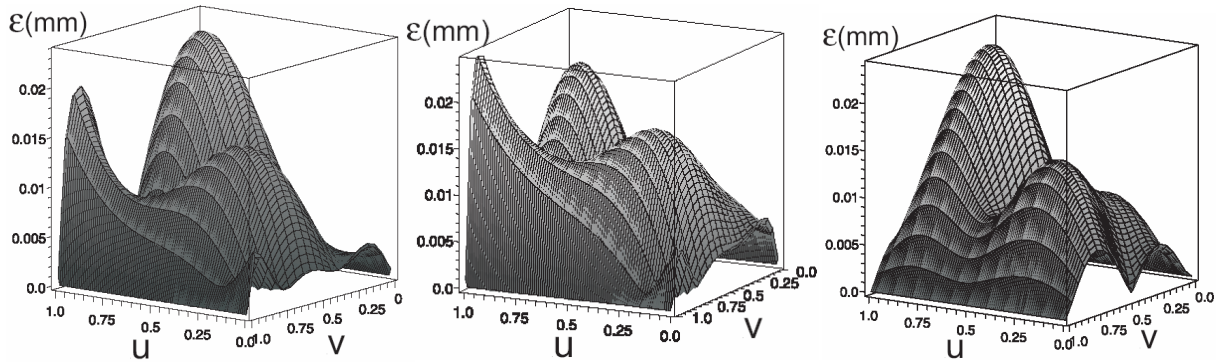


Fig. 7: Surface error for different cutting tools. a). a cylindrical cutter. b). a conical cutter. c). a barrel cutter.

We can also compare the solution of the cylindrical tool to the solution with the same surface and cutter obtained from Li's paper [1]. The maximum surface errors of the both method are similar (around 0.022mm in Li's paper), but the proposed method needs more control points (9 by 3 vs. 3 by 3) and higher degree (8 by 2 vs. 2 by 2) to generate the surface. However, the proposed method can be applied to any tool of revolution, broadening the application of flank millable surfaces.

## 5. DISSCUSSION AND VARIATIONS

In step 3, to find the curve $C^d$ to match points $(u_i, v_i)$, a Bézier curve is used to describe $C^d$. Under some situations, a B-Spline curve is is needed to more accurately approximate the given points. If a B-Spline curve is used in this step, we need initially to decompose the piecewise polynomial B-Spline curve into a sequence of Bézier curves using knot insertion [15], and each Bézier curve can then be used in step 4 to get its corresponding 3D Bézier curve. After all the 3D Bézier curves are obtained, knot removal is used to convert this curve sequence into a single B-Spline curve. This B-Spline curve is used in the following flank millable surface design.

We use a polynomial approximation to the surface of revolution. While this approximation should be more than sufficient for most engineering applications, if an exact representation of the surface of revolution is desired, then a rational tensor product surface could be used instead. One advantage of using a rational tensor product representation of the circles is that degree 2 curves could be used instead of the degree 3 polynomials curves we used. However, use of rationals would require finding settings of the weights, which is a non-trivial problem.

One of the key benefits of using polynomial composition in our method is that the approximations to the grazing curves lie exactly on our representation of the tool (i.e., the surface of revolution, R). The downside of using polynomial composition is that it leads to high degree representations of the grazing surfaces. Whether having the approximations to the grazing curves lying exactly on the tool is worth the high degree result is unclear; alternative methods of constructing approximations to the grazing surface may give better results.

## 6. CONCLUSIONS

In this paper, a method to generate the flank millable surfaces with polynomial composition was developed. The main feature of note in our method is that it can generate flank millabe surfaces for any tool of revolution. The examples show that the composed surface closely matches the true grazing surface.

## 7. REFERENCES


[1] Li, C. G.; Bedi, S.; Mann, S.: Surface design for flank milling, submitted to CAD journal in July, 2006, under review.

[2] Li, C. G.; Bedi, S.; Mann, S.: Accuracy improvement method for flank milling surface design, submitted to CAD journal in September, 2006, under review.

[3] DeRose, T. D.; Goldman, R. N.; Hagen, H.; Mann, S.: Functional composition algorithms via blossoming, TOG, 1993, 113-135.

[4] Liu, W.; Mann, S.: An optimal algorithm for expanding the composition of polynomials, ACM Transactions on Graphics, 16, 1997, 155-178.

[5] Liu, X.: Five-axis NC cylindrical milling of sculptured surfaces, Computer-Aided Design, 27(12), 1995, 887-894.

[6] Tsay, D. M.; Her, M. J.: Accurate 5-axis machining of twisted ruled surfaces, Journal of Manufacturing Science and Engineering, 123, 2001, 734-738.

[7] Redonnet, J.-M.; Rubio, W.; Dessein, G.: Side milling of ruled surfaces: optimum positioning of the milling cutter and calculation of interference, Advanced Manufacturing Technology, 14(7), 1998, 459-465.

[8] Li, C. G.; Bedi, S.; Mann, S.: Flank milling of ruled surface with conical tools-an optimization approach, The International Journal of Advanced Manufacturing Technology, 29(11-12), 2006, 1115-1124.

[9] Rubio, W.; Lagarrigue, P.; Dessein, G.; Pastor, F.: Calculation of tool paths for a torus mill on free-form surfaces on five-axis machines with detection and elimination of interference, The International Journal of Advanced Manufacturing Technology, 14(l), 1998, 13-20.

[10] Bedi, S.; Mann, S.; Menzel C.: Flank milling with flat end cutters, Computer-Aided Design, 35, 2003, 293-300.

[11] Elber, G.; Fish, R.: 5-axis freeform surface milling using piecewise ruled surface approximation, ASME, Journal of Engineering for industry, 119, 1997, 383-387.

[12] Li, C. G.; Bedi, S.; Mann, S.: Flank milling surface design with the least square approach, WSEAS TRANSACTIONS ON MATHMATICS. 5(7), 2006, 844-852.

[13] Mann, S.: A Blossoming development of splines, Synthesis Lectures on Computer Graphics, Morgan-Claypool, 2006.

[14] Li, C. G.; Bedi, S.; Mann, S.: Error measurements for flank milling, Computer-Aided Design, 37, 2005, 1459-1468.

[15] Farin, G.: Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide, Academic Press, 2002.

[16] Piegl, L.; Tiller, W.: The NURBS Book, Springer, 1997.

[17] Dokken. T.; Daehlen, M.: Good approximation of circles by curvature-continuous Bézier curves, Computer-Aided Geometric Design, 7, 1990, 33-41.

[18] Mann, S.; Bedi, S.: Generalization of the imprint method to general surfaces of revolution for NC machining, Computer-Aided Design, 34, 2002, 373-378.