Simulating Multi-Axis Machining Using NURBS and Quaternions

David Yoon, Jie Shen, Evrard Ohou and Elsayed Orady

University of Michigan - Dearborn, {dhyoon|shen|oevfred|orady}@umich.edu

ABSTRACT

The activities of multi-axis machines, in particular, 5-axis machines are simulated using NURBS and quaternions. The translational activities of 3-axis machines may easily be simulated using NURBS, while the rotational activities are simulated using quaternions.

Keywords: NURBS, quaternions, cutter models, cutter swept surfaces.

1. INTRODUCTION

NC (Numerical Control) machines are classified in terms of the number of axes that are simultaneously and independently controlled. A 2-axis machine can simultaneously move an object on the cutting table along two axes and a 3-axis machine can do the same along three axes. These motions are essentially translational. However, NC machines of higher axes can rotate the object on the table in addition to the translational motions as in a 3-axis machine. For instance, a 5-axis machine can rotate the object with respect to two axes in addition to the translational motions as in a 3 axis machines.

Traditionally linear motions using small line segments are employed to generate translational motions, whereas circular motions using arcs have generated rotational motions. These techniques are *not* robust enough for the next generation NC controllers. We propose NURBS (Non-uniform rational B-spline) as the tool for translational motions and quaternions as the tool for rotational motions in 3-D space. Although NURBS have been employed in CAD extensively, they have not quite made their way into NC machining. In this paper cutter-paths and the profiles of cutters are modeled as NURBS curves and the surface swept by a cutter along the cutter-paths are modeled as NURBS surfaces. The motions of 3-axis machines are modeled and simulated primarily in terms of NURBS and those of 5-axis machines employ both NURBS and quaternions. In our earlier work, we presented the simulator [2] and simulated 2- and 3-axis machines [1]. In this paper, the emphasis is on 3- and 5- axis machines using NURBS and quaternions.

2. MATHEMATICAL PRELIMINARIES

The following Mathematical concepts are needed:

2.1 Non Uniform Rational B-Spline (NURBS)

A n^{th} -degree NURBS curve [3] is defined by

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad 0 \le u \le 1 \tag{1}$$

Where the { P_i } are the *control points*, the { w_i } are the *weights*, and the { $N_{i,p}(u)$ } are the pth-degree B-spline basis functions defined on the non-uniform knot vector

$$U = \{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_{m-p-1},\underbrace{1,\ldots,1}_{p+1}\}$$

A NURBS surface of degree n in the u direction and degree m in the v direction is defined by

$$S(u,v)=\frac{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)w_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m}N_{i,p}(u)N_{j,q}(v)w_{i,j}} \quad 0\le u,v\le 1 \tag{2}$$

The $\{P_{i,j}\}$ form a bi-directional control net, the $\{w_{i,j}\}$ are the weights, and the $\{N_{i,p}(u)\}$ and $\{N_{j,q}(v)\}$ are the nonrational B-spline basis function defined on the knot vectors

$U=\{\underbrace{0,\ldots 0}_{p+1},u_{p+1},\ldots,u_{r-p-1},\underbrace{1,\ldots 1}_{p+1}\}$ $\qquad V=\{\underbrace{0,\ldots 0}_{q+1},v_{q+1},\ldots,v_{s-q-1},\underbrace{1,\ldots 1}_{q+1}\}$

2.2 Quaternions

A quaternion is defined in terms of a 3-D vector with an extra scalar quantity. In animation the orientation of an object is often described using roll, pitch and yaw angles, which has some disadvantages such as developing a rotation from three separate rotations about the origin, and the interpolated path is hard to predict and generally produces non-linear movements. The quaternion approach provides a simple mechanism for rotating objects about axes. Given a sequence of quaternions, it is possible to interpolate them to create controlled complex rotations.

In general, a quaternion q is defined as a tuple, $q=[s,v]$, where s denotes the scalar and v denotes the three ordinary vectors in three-dimensional space, R^3 [4, 5].

Given two quaternions q_1 and q_2, $q_i=[s_i,v_i]$, where i=1,2,

$v_1 \bullet v_2$ is the dot product of v_1 and v_2, and $v_1\times v_2$ is the cross product. In both cases, the results are still quaternions.

The magnitude

$$|q|=\sqrt{x^2+y^2+z^2+s^2} \tag{3}$$

where s is the scalar term, and x,y,z are the vector components of v.

The inverse of quaternion

$$q^{-1}=\frac{[s,-v]}{|q|^2} \tag{4}$$

$$q^{-1}q=[1,[0,0,0]] \tag{5}$$

A vector v can be rotated about an arbitrary axis q by the operation, $v^{'}=qvq^{-1}$, where $v^{'}$ is the vector after the rotation. Here we define a position vector A that passes through the origin. Then any point on the vector can be represented in quaternion form as $[0,A]$. Rotate the point by an angle about an axis by applying the quaternion rotation operation is:

$$q[0,A]q^{-1} \tag{6}$$

As a special case, when the unit quaternion is used i.e. $|q|$ =1, the unit quaternion equals to:

$$q=[s,\sin(\theta/2)u] \tag{7}$$

where $s=\cos(\theta/2)$, u is a unit vector, which is also the direction of the rotation axis passing through the origin; θ is the angle of rotation.

2.3 Z-map

The Z-map [6] representation of a surface patch is the discrete version of its explicit representation, z=f(x,y): z((i,j) = (x(i),y(j)), where x=x_0 + d_1*i, y=y_0 +d_2*j, where d_1, d_2 are real values.

In this century, the parametric representation of a surface s(u,v) =(x(u,v), y(u,v), z(u,v)), has been popular and implemented in most commercial CAD systems. However, most CNC machines are based on the 3-dimensional Euclidean space. Hence the parametric surface must be embedded into the 3-dimensional space R^3. The transformation $T : I^2 \rightarrow R^3$ such that T(s(u,v)) = (x(u,v), y(u,v),z(u,v)) needs to be devised. It can be represented as z=f(x,y) for some function f. The continuous surface, z=f(x,y), may be discretized as z(i,j) =(x(i),y(j)), where x(i) and y(j) are computed as above.

Representing the *cut surface* is a direct concern of this project. Model surfaces are most often well defined mathematically, usually through parametric equations and control points (e.g. Bezier patches, NURBS. Unfortunately, while the surface is being cut, it is generally not so well defined. The z-vector's terminate at the cut part's surface. This method is used by [7,8]. An enhanced version of the z-map method has also been developed using non-uniformly spaced z-vectors.

This method of surface representation is convenient for cutter/surface intersection methods, especially for the design of the real-time work-piece database for 3- and 5-axis and robot-based machining.

$$z = CSS(u,v) = \begin{cases} S_z + CBS(u,v) & for \quad v \le R \\ CBS(u, SC(u)) + slope * (u - SC(u)) & for \quad SC(u) < v < SC(u) + d \\ E_z + CBS(u,v) & for \quad v \ge SC(u) + d \end{cases} \tag{8}$$

3. THE SIMULATOR

The simulator consists of four modules: The cutter-modeler, the cutter-swept surface (CSS) modeler, the workpiece modeler, and the communicator [2]. The simulator receives design data from remote CAD systems and simulates the material removal from workpieces. The cutter modeler models various cutters based on the specifications of APT cutters. In the following section, the cutter modeler is briefly summarized.

3.1 The Cutter Modeler

There are numerous cutter shapes on the market. However, in this project the classical APT cutter models are employed. In Ref [7] the authors propose explicit z-value equations for ball-end mill and rounded end-mill CSS's (cutter-swept surface). Recently, Chung *et al* [10] provided an analytical method for finding the z-value of a CSS for any *general* APT-like cutter.

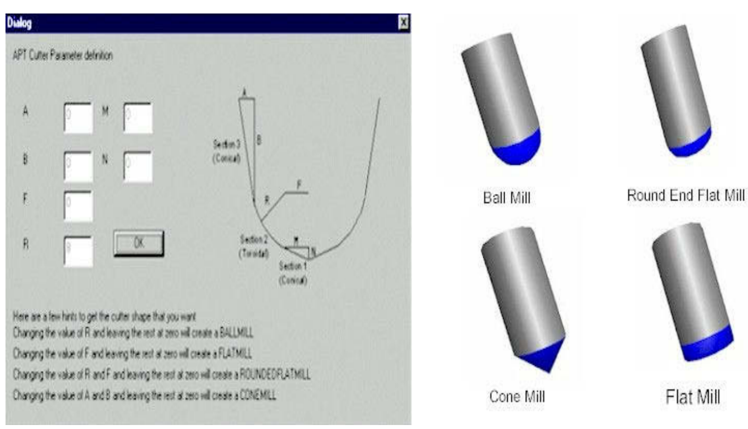

Fig. 1: APT cutter shapes generated by the cutter modeler.

The method proposed in Ref. [10] is based on the special cases of the ball-end mill and the cone mill. The general APT cutter is then analyzed in three sections (Fig. 1) each of which is the shape of either a toroid or a cone. The method extrapolates the CSS from the cutter shape by calculating the cutter/surface intersection curve (silhouette curve) and assuming a linear motion. The CSS itself can then be thought of as the bottom of the cutter at the beginning and end of the cut (cutter bottom surface (CBS)) joined with the extrusion of the cutter/surface intersection curve along the cut direction. Mathematically we can write:

Here S_z is the z-value of the start point, E_z is the z-value of the end point, (u, v) is the local coordinate of the current z-vector being tested, d is the projected (onto the xy plane) distance the cutter moved, CBS(u,v) is the cutter bottom surface z-value given the local coordinates (u, v), and slope is the slope of the cut: $(S_z – E_z)/d$.

SC(u) is the silhouette curve equation that indicates the value, v, given u that lies along the curve of intersection between the cutter and the surface.

The calculation of CBS(u,v) is straightforward given the cutter parameters: a, b, r, f, m, n, and the calculation is based on the segmentation of the general cutter into the aforementioned conical and toroidal sections.

The calculation of SC(u) (the silhouette curve) is not as straightforward. To get the intersection between the cutter and the surface, the cutter normal and the cut slope vector should be normal. This condition leads to a quartic equation for the toroidal surface that can be solved algebraically at the expense of 35 multiplications, 2 divisions, 4 square roots and 2 cube roots in the worst case. Unless an estimation method is used, this calculation must be done (in the worst case of a ball mill) for every z-vector under the shadow of the CSS.

Though it is somewhat slow, we use the Chung *et al* method currently in our system because it is efficient enough for fast machines (e.g. P-266 and up) and because it can give us an exact solution to the intersection problem (if deflection is ignored). The cutter modeler is based on the model by Chung et al [10] and capable of modeling and rendering APT-type cutters. The left figures of Fig. 1 illustrates the window which allows the user to interact with the cutter modeler, the rest are the sample cutter shapes generated by the cutter modeler.

3.2 The Cutter Swept Surface (CSS) Modeler

A swept surface is a surface generated by sweeping an object in the 3-dimensional space. A fair amount of work has been done using the concept of an envelope. In this paper, following Piegl/Tiller [3], a swept surface is defined as follow:

Let S(u,v) = T(v) + M(v)C(u) be a swept surface, where $0 \leq u, v \leq 1$, T(v) = an arbitrary trajectory, C(u) = a profile curve of the object, M(v) = a 4x4 matrix representing transformations on the profile curve C(u) at v. There are essentially two cases to consider:

Case 1: M(v) is an identity matrix.

In this case S(u,v) = T(v) + C(u) and the surface is obtained by sweeping the profile curve along the trajectory T(v). The trajectory T(v) and the profile curve C(u) are specified in terms of NURBS (non-uniform rational B-spline). This forms the basis for CSS for 3-axis machining involving only translational motions.

Case 2: M(v) includes rotations.

This forms the basis for CSS for 5-axis machining which involves both translational and rotational motions. 3-dimensional rotations are modeled in terms of quaternions.

3.2.1 CSS for 3-axis Machining

The cutter path curve $P_c(v)$ is defined as a NURBS curve as follows:

$$P_c(v) = \frac{\sum_{i=0}^{n} N_{i,p}(v)\, w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(v)\, w_i} \quad a \leq v \leq b \tag{9}$$

where $K = \{ a, \dots, a, k_{p+1}, \dots, k_{m-p-1}, b, \dots, b \}$ is the *knot vector* (in general, non-uniform, i.e. $k_{i+1} - k_i \neq$ Constant, for i=0,,n-1) on which the curve is defined and the $N_{i,p}(v)$ are NURBS basis functions as defined previously.

Equation (9) defines a rational piecewise continuous curve of n+1 curve segments. These curves have many interesting properties including C^p continuity (except at multiple knots, i.e. $k_{i+1} = k_i$), affine invariance, strong convex hull property and more.

To construct a CSS (cutter-swept surface) from a given path curve, $P_c(v)$, we also need a cutter orientation function, O(v), defining the "up" direction of the tool. This orientation function need not restrict the cutter to the positive z direction (i.e. O(v) = [0, 0, 1]) . In the 3-axis case, the orientation function as defined in

Eq(10) sets the cutter orientation to be normal to the cutter motion. The cutter velocity is given by $P_c^{'}(v)$ (the first derivative), then we construct our cutter orientation vector as:

$$O(v) = P_c^{'}(v) \times \left(\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \times P_c^{'}(v) \right) \tag{10}$$

From O(v) we can generate a local frame given by the coordinate basis vectors

[x(v) y(v) z(v)] as:

$$z(v) = \frac{O(v)}{|O(v)|} \qquad y(v) = \frac{P_c^{'}(v)}{|P_c^{'}(v)|}, \quad x(v) = \frac{z(v) \times y(v)}{|z(v) \times y(v)|} \tag{11}$$

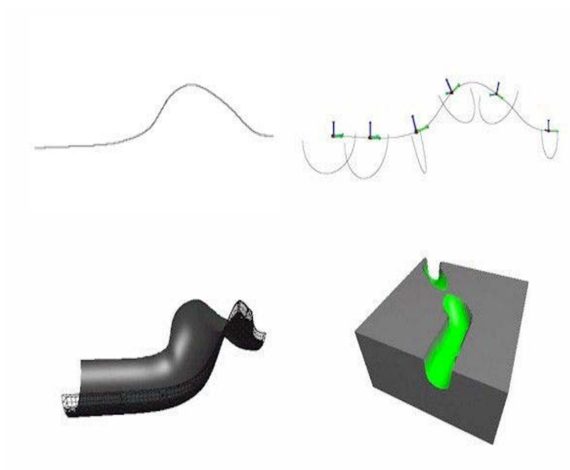

Fig. 2: Top row: a cutter path NURBS curve and path curve with local frames; bottom row: swept surface and the cutter swept surface in raw stock.

This local frame (see Fig. 2) is valid along the entire path curve, $P_c(v)$ and avoids the sudden twists and flipping associated with the Frenet frame [11]. In Fig. 2, the top left is the cutter path as a NURBS curve, the top right shows the cutter path with local coordinates along with the profile curves of the cutter, the bottom left is the shaded cutter swept surface and the bottom right is the raw stock with excessive material removed by the cutter.

3.2.2 CSS for 5-axis Machining

The CSS modeler for 5-axis machining is based on Case 2 of section 3.2 in which the transformation M(v) includes rotations. For each rotation, there is the axis and angle of rotation associated with it. They are represented as a quaternion q as defined in section 2.2. The algebraic properties of quaternions are summarized in section 2.2. and the further detailed can be found in [4,5]

To maintain continuous rotation, the application uses spherical linear interpolation (slerp) to calculate most of the quaternions. The implementation is based directly on the formula:

$$\text{Slerp}\ (q_0, q_1 : t) = q_0 \frac{\sin a(1-t)}{\sin a} + q_1 \frac{\sin at}{\sin a} \quad \text{and}$$

$$\cos a = q_0\, q_1 \tag{12}$$

Linear interpolation can be employed when the quaternions are very close and if omega is < 0 the signs need to be switched. The following algorithm implements the above interpolation:

```
GQuat GQuat::slerp(GQuat *in_to, float t) {
    GQuat to = *in_to;
        GQuat from= *this;
    double omega, cosine, sine, A, B;
```

Some of the special cases include the following: If r> 0 and a=b=f=m=n=0, it is a ball end-mill,

If f>0, b>0, and a=r=m=n=0, it is a round end-mill, and finally if a>0, b>0 and f=r=m=r=0, it is a cone end-mill (see Fig. 1). The GUI window is for 3- and 5-axis machining. The basic assumption is that rotations are performed between –45 and 45 degrees along the X- and Y-axis.

Fig. 4 illustrates the view menu that allows the user to animate the cutter, to step through motions frame by frame, and to view the cutter, CSSs, skeleton of the cutter paths, the current status of the raw stock, and the guide curve. The window in a way summarizes what has been done in this project. A sub-window will be included for robot-based machining.

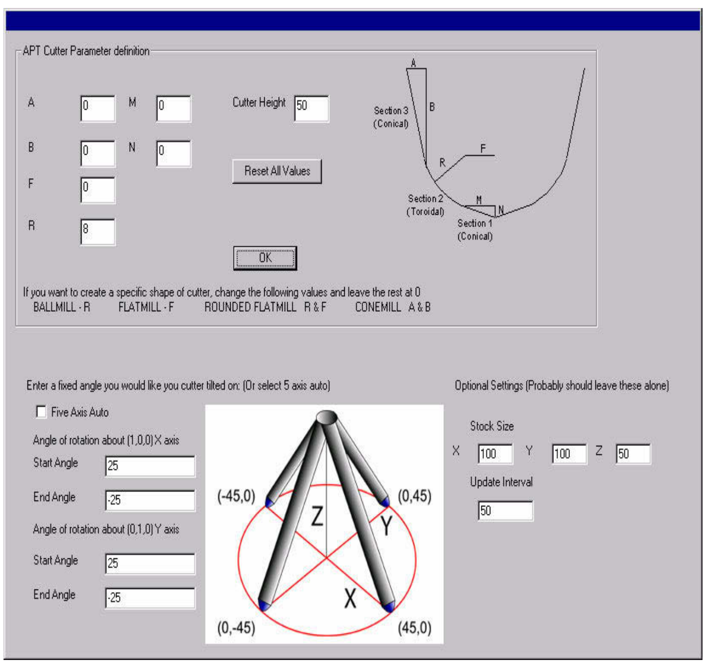

Fig. 3: GUI Window.

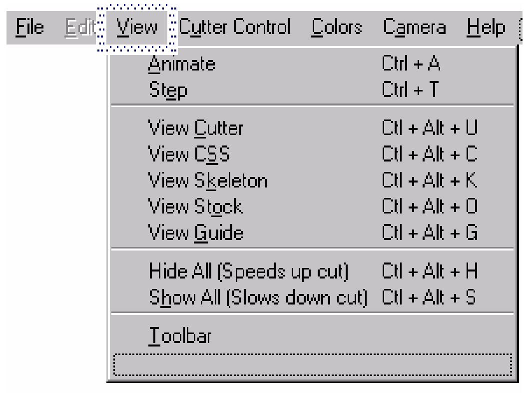

Fig. 4: View sub-window.

4. CONCLUSION

The simulator is an on-going project at the University of Michigan - Dearborn. In this paper the cutter swept surfaces (CSS) for 3- and 5-axis machining using ball-end cutters are presented. NURBS have been extensively used to model the cutter, the raw-stock, and the cutter paths. Quaternions have been employed to model rotational motions in 5-axis machining. Robot machining based on the articulated robot model is currently under development and will be covered in a separate paper.

5. REFERENCES

[1] Crossman, J. A. Yoon, D. H.: Cutter swept surface construction using B-spline surface interpolation, Int. J. of Modelling & Simulation, 21(4), 2001, 292-300

[2] Crossman, J. A, Yoon, D. H.: A cutter motion simulation system, J. of Integrated Design and Process Science, 4(1), 2000, 25-35.

[3] Piegl, L, Tiller, W.: The NURBS Book, 2nd ed., Springer-Verlag, New York, N.Y. 1997

[4] Hanson, A. J.: Visualizing Quaternions, Morgan Kaufman, Boston, 2006

[5] Shoemake, K.: Animating rotation with quaternion curves, SIGGRAPH'85, 19(3), 1985, 245-254

[6] Choi, B. K.; Jerard, R. B.: Sculptured Surface Machining, Kluwer, Boston, 1998, 255-258

[7] Choi, B. K,.; Chung, Y. C.; Park, J. W.: Application and extension of Z-map model, in SY Sin, T Kunii ed Computer Graphics and Applciations, World Scientific, New Jersey, 1995

[8] Chou, J. J.; Yang, D. C. H.: Command generation for three-axis CNC machining, Journal of Engineering for Industry, 113, August 1991

[9] Jerard, R. B.; Drysdale, R. L.: Geometric simulation of numerical control machining, ASME International Computers in Engineering Conference and Exhibition. San Francisco, CA. 1988, 129-136.

[10] Chung, Y. C.; Park, J. W.; Shin, H.; Choi, B. K.: Modeling the surface swept by a generalized cutter for NC verification. Computer Aided Design, 30(8), 1998, 587-593

[11] Do Carmo, M. P.: Differential Geometry of Curves and Surfaces, Prentice Hall, N.J. 1976