Efficient Intersection of Surface Normals With Milling Tool Swept Volumes for Discrete Three-Axis NC Verification

James H. Oliver
Iowa State University, oliver@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/me_pubs
Part of the Mechanical Engineering Commons

The complete bibliographic information for this item can be found at http://lib.dr.iastate.edu/me_pubs/53. For information on how to cite this item, please visit http://lib.dr.iastate.edu/howtocite.html.

This Article is brought to you for free and open access by the Mechanical Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Mechanical Engineering Publications by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
Efficient Intersection of Surface Normals With Milling Tool Swept Volumes for Discrete Three-Axis NC Verification

Abstract
An efficient algorithm is presented for intersecting vectors with swept solids which represent three-axis numerically controlled (NC) milling tool motions. The intersection calculation proceeds in hierarchical steps through a series of progressively more exact definitions of the shape of the tool swept volume. At each step, results of intermediate calculations are used to determine whether intersection with an exact representation of the solid is possible and, if so, where and how the swept volume model must be refined for the next step. This structure ensures that superfluous intersection calculations are minimized. This intersection technique has been successfully implemented as part of an algorithm for automatic verification of three-axis NC milling programs, and may also be useful for applications in robotics and factory automation.

Keywords
Intersections, Milling

Disciplines
Mechanical Engineering

Comments
This article is from Journal of Mechanical Design 114 (1992): 283–287, doi:10.1115/1.2916944. Posted with permission.
An efficient algorithm is presented for intersecting vectors with swept solids which represent three-axis numerically controlled (NC) milling tool motions. The intersection calculation proceeds in hierarchical steps through a series of progressively more exact definitions of the shape of the tool swept volume. At each step, results of intermediate calculations are used to determine whether intersection with an exact representation of the solid is possible and, if so, where and how the swept volume model must be refined for the next step. This structure ensures that superfluous intersection calculations are minimized. This intersection technique has been successfully implemented as part of an algorithm for automatic verification of three-axis NC milling programs, and may also be useful for applications in robotics and factory automation.

Vector/Solid Intersection

The logical flow of the vector/solid intersection algorithm is summarized schematically in Fig. 1. As input, the algorithm requires starting and ending tool positions which define a tool motion, and a list of design surface points and associated normals (vector list) which are in the vicinity of the given tool output is produced which depicts the desired part as shaded surfaces with out-of-tolerance areas highlighted. This technique can provide computationally efficient and dimensionally accurate NC verification for parts designed with either solids, sculptured surfaces, or both.

Algorithms for discrete NC verification are most easily described in terms of three major modules. The first involves a method for discretizing the desired part model into a sufficiently dense grid of surface points and normals. The second module provides a means for extracting a subset of eligible points and normals to be considered for each tool motion. Algorithms for these two procedures are presented elsewhere (Jerard and Drysdale, 1987; Oliver and Goodman, 1990). The focus of this paper is the third module; an efficient and robust technique for the calculation of vector/solid intersections for use in three-axis milling applications. The algorithm presented here is distinct from previously published intersection techniques (Jerard et al., 1989b) in its hierarchical structure which provides a means to quickly eliminate vectors that nearly intersect the swept volume from further, more involved intersection calculations. Although the algorithm presented here deals specifically with a spherical (ball-nosed) milling tool, the technique can be generalized to accommodate most common tool shapes.
surrounds a tool swept volume resulting from a typical three-

Parallelepiped Approximation

Figure 2 depicts the smallest bounding parallelepiped which

axis milling tool motion. The parallelepiped is composed of

six infinite planes represented in the standard form, i.e.,

\[ A_i x + B_i y + C_i z + D_i = 0 \quad \text{for} \quad i = 1, 2, ..., 6 \]  

where scalars \( A_i, B_i, \) and \( C_i \) define a unit vector normal to the

plane, and \( D_i \) is calculated by substituting the coordinates of

any point on the plane for \( x, y, \) and \( z. \)

The three independent vectors necessary to define the bound­

ing planes are generated from the direction of the mill tool

axis and the direction of tool travel. The tool path vector \( p \)

calculated as the difference between successive tool center po­

sitions (i.e., linear tool path interpolation is assumed). The

cross product of \( p \) with the mill axis vector \( z \) results in a vector

\( q \) which defines the side planes of the parallelepiped. The end

faces are defined by the vector \( s \) generated from the cross

product of \( q \) and \( p \), while the cross product of \( q \) and \( p \) yields

the vector \( t \) which is perpendicular to the top and bottom faces

of the parallelepiped. The necessary vectors and vector prod­

ucts are summarized below.

\[
\begin{align*}
\mathbf{z} & \text{ mill axis} \\
\mathbf{p} & = \mathbf{p}_0 - \mathbf{p}_t \quad \text{tool path} \\
\mathbf{q} & = \mathbf{p} \times \mathbf{z}/|\mathbf{p} \times \mathbf{z}| \bot \text{ to parallelepiped (right-hand) side face} \\
\mathbf{s} & = \mathbf{q} \times \mathbf{z}/|\mathbf{q} \times \mathbf{z}| \bot \text{ to parallelepiped front face} \\
\mathbf{t} & = \mathbf{q} \times \mathbf{p}/|\mathbf{q} \times \mathbf{p}| \bot \text{ to parallelepiped bottom face}
\end{align*}
\]

The planes are generated such that all (plane) normals point

toward the inside of the volume. The points on the planes are

generated by considering the tool radius \( R \), and height \( H. \) For

example, the right-hand side of a parallelepiped is defined by

the vector \( q \) and the point \( p_t = p_0 + Rq, \) while the left-hand side

is defined by \( -q \) and the point \( p_t = p_0 - Rq. \)

Vector/Plane Intersection

The next step in the algorithm involves intersection of a

surface normal vector with the six infinite planes of the par­

allelepiped. Denoting a plane normal as \( \mathbf{n}_i = [A_i, B_i, C_i]^T, \)

design surface point \( p_i, \) and corresponding unit normal \( \mathbf{n}_i, \)
the signed distance \( L \) from \( p_i \) along \( \mathbf{n}_i \) to intersection with the

plane is given by,

\[
L = (\mathbf{n}_i^* \cdot \mathbf{p}_i + D_i)/|\mathbf{n}_i^* \cdot \mathbf{n}_i| 
\]

(2)

Note that the absolute value of the numerator in the above

formula is simply the normal distance from \( p_i \) to the plane.

The sign of \( L \) indicates the direction toward intersection relative to

\( \mathbf{n}_i. \) Intersection points are calculated for each of the six

bounding planes as,

\[
p_i = p_i + L_i \mathbf{n}_i \quad \text{for} \quad i = 1, 2, ..., 6 
\]

(3)

After \( \mathbf{n}_i \) has been intersected with each of the six planes of the

parallelepiped, the intersection points \( p_i \) are sorted, based on

\( L_i, \) from smallest (most negative) to largest.
Point/Region Classification

The first intersection point which falls on a parallelepiped face is processed further to determine whether refinement of the swept volume approximation is necessary, and if so, the nature and location of the refinement as well. This is accomplished with another point classification; this time with bounding planes moved toward the inside of the swept volume to represent the boundaries on its surface generated by the motion of the tool. These boundaries are the focus of swept volume points whose normals are perpendicular to the direction of the tool motion. In the terminology of Ganter (1985) these boundaries represent the “motion silhouettes” of the swept volume. In this application, they indicate transitions between planar and cylindrical surfaces, or between cylindrical and spherical surfaces of the precise milling tool swept volume. These region boundaries are indicated by dashed lines in both Figs. 2 and 4.

Figure 4 shows a side face of an approximating parallelepiped, divided into regions labeled 1 through 8. First, the intersection point is classified (as described above) with respect to the planes which bound region 1. If the intersection point falls within this region, processing for the point ends, since the planar intersection is already exact. If it is not within subregion 1, the elements of the vector product are interpreted to determine which region should be examined next. For instance, if the plane separating region 1 and region 4 yields a negative value, then another point classification, with the planes which bound region 4, determines whether region 4 or 5 should be considered next. At most, two point/region classification calculations are sufficient to determine the region of possible vector intersection for this (side) face of the parallelepiped.

Referring again to Fig. 4, if region 2, 3, or 4 contain the intersection point, then the normal vector may intersect a cylindrical surface. If regions 5 or 6 contain the point, a spherical surface must be considered. If the normal intersects the plane in region 7, the user is warned that the part may have been cut by the rearward facing top edge of the tool if the mill is lifting, or forward facing top edge if the mill is diving, and processing for the point is terminated. Similarly, if region 8 contains the point, the normal has missed the actual swept volume and processing terminates. Note that regions 7 and 8 do not exist if the mill does not change its Z coordinate (height) during a motion.

An analogous, although somewhat simpler, procedure is applied if the intersecting plane is an end face or the bottom face of the parallelepiped, except that only cylindrical or spherical regions are possible.

Vector/Sphere Intersection

Figure 5 depicts the intersection of a design surface normal vector with a spherical surface of the precise milling tool swept volume. The following terminology defines the geometry of this situation.

\[ p_i \] design surface point
\[ \mathbf{n}_i \] corresponding unit normal vector
\[ c \] center of sphere
\[ \mathbf{v} = c - p_i \] vector position of \( c \) relative to \( p_i \)

It is quite possible for a normal vector to pierce the parallelepiped approximation in a spherical region yet miss the actual swept volume completely. This condition is checked first by calculating the distance from \( c \) to \( p_i \) as,

\[ L_n = |\mathbf{v} \times \mathbf{n}_i| \]

If \( L_n \) is greater than the tool radius \( R \), then the vector misses...
where:

\[ \text{for coplanar case} \]

\[ m = u + L_n e \quad \text{for noncoplanar case} \]

\[ m = u \quad \text{for coplanar case} \]

Finally, the directed (signed) distance from \( p_s \) along \( n_s \) to intersection with the cylindrical surface is simply, \( L_c = L_s - L_{in} \).

**Implementation**

Several practical concerns must be considered in the implementation of this algorithm. The first is the pathological condition in which two vectors, generally assumed to be independent, are in certain special cases parallel. For instance, in calculating the basis vectors for the parallelepiped, if the tool path vector \( p \) is parallel to the mill axis vector \( z \) (i.e., \( p \times z = 0 \)) then vectors \( s \) and \( t \) must be arbitrarily selected as mutually orthogonal vectors in the plane defined by \( z \). Similarly, in calculating vector/cylinder intersections, a special case must be accommodated when \( n_c \) and \( c_p \) are parallel. A related problem occurs in calculating vector/face intersections; if the vector is nearly parallel with the face, a very large positive or negative value can result. This can be handled by setting a maximum magnitude for directed distance to intersection. An intersection point with distance magnitude greater than this limit can be eliminated from the list during sorting.

Another concern is the limitation of the point/region classification procedure. This scheme is generally successful in directing the intersection calculation toward the refinement most likely to produce intersection with precise bounding surfaces. However a straightforward implementation may produce misleading results. For example, a vector could be classified as piercing a region of the parallelepiped which suggests refinement with a spherical surface. The subsequent vector/cylinder intersection calculation could yield a miss while the vector actually proceeds, inside of the parallelepiped, to intersect a cylindrical surface.

This limitation can be ameliorated by careful consideration of the direction of the normal vector relative to the parallelepiped. For example, referring again to Fig. 4, if the vector pierces region 5, yet does not intersect the corresponding sphere...
ical surface, the scalar products \((\mathbf{n} \cdot \mathbf{x})\) and \((\mathbf{n} \cdot \mathbf{p})\) can be interrogated to determine whether cylindrical regions 2 or 4 should be considered, or if the vector does indeed miss the swept volume.

**Results**

This algorithm has been implemented successfully as part of a larger program for automated three-axis NC verification (Oliver and Goodman, 1990). In actual operation, the algorithm efficiently eliminates many vectors from detailed intersection analysis; only those which have a high likelihood of intersecting the precise quadric surfaces of the swept volume are subjected to the more complex intersection calculations. Of course, a straightforward alternative to the approach presented here is to simply intersect the normal with all of the quadric surfaces that bound the actual swept volume and accept the intersection yielding the smallest value. This approach, however, would require that for each normal an attempt be made to intersect it with three cylinders, two spheres, and two planes. We have found that the small additional computational burden associated with constructing the interior bounding planes, dividing the surfaces of the parallelepiped into regions, and classifying intersection points with respect to these regions, is far out-weighed by the substantial reduction in the number of actual quadric surface intersections. Since the intersection of normals with swept volume models is the most computationally intensive part of the discrete NC verification procedure, this technique provides excellent overall computational savings.

Although designed primarily for application to NC verification, this algorithm may also be useful for analysis of robotic path planning systems, automated assembly operations, and other applications in which the precise distance between a stationary part and another, translating object is of interest. The algorithm, however is not directly extensible to five-axis NC tool motions (i.e., machines which provide two rotational as well as three translational degrees of freedom). A new vector/solid intersection technique for general five-axis tool sweeps is currently under study; preliminary results are presented in (Narvekar, 1991). This technique will be employed in an ongoing study toward automated methods for accurate and efficient generation of general five-axis NC programs.

**References**


