Real-time CNC Tool Path Generation for Machining IGES Surfaces

Y.D. Chen
Graduate Student Research Assistant.

J. Ni
Assistant Research Scientist.

S. M. Wu
R. and P. Anderson Professor of Manufacturing Technology.

Department of Mechanical Engineering and Applied Mechanics,
The University of Michigan,
Ann Arbor, MI 48109

A real-time CNC tool path generation algorithm has been developed for machining IGES surfaces. IGES-based CAD data files can be directly inputted to the CNC system and the tool paths generated in real time can be passed to a motion controller during cutting via a Multibuss II backplane structure. The development of such a real-time tool path generation method has eliminated the need for a tradeoff between the desired surface accuracy and the required memory size for storing off-line generated NC codes. The real-time NC path generation algorithm can properly deal with issues such as trimming lines, gouging detection, and adaptive tool step adjustment. The developed algorithm has been implemented on a multi-processor CNC system and verified through actual cutting tests. The test results show that no violating conditions occurred on machined part surfaces, and the surface contour error of the cut part is less than the given tolerance, which was 0.02 mm in this particular test.

1 Introduction

Cutter paths for machining part surfaces are usually generated in advance either manually for simple part geometries or automatically after a part is designed in a CAD station. Many engineering parts, such as car bodies and airplane wings, have complex free-form surfaces. Since it is difficult to represent these surfaces by a single mathematical equation, they are in general represented by composite surfaces composed of many patches. Many kinds of surface patch equations have been developed, and the most popular one is the bicubic parametric patch: entity type 114 defined in the Initial Graphics Exchange Specification (IGES).

For such free-form surfaces, the current industrial practice is to use an off-line tool path generation procedure by which the tool path is first generated from the CAD data files (surface equations) and then to use a post-processor to generate and to download NC codes to a CNC controller. When a complex free-form surface is to be machined with a tight tolerance requirement, the off-line NC path generators have to generate the cutter locations in (X, Y, Z) format at very small steps in order to produce the required smooth surfaces. This in turn requires a very large memory in the CNC controller to store the NC tool path data. In many cases the memory of the available CNC controllers cannot accommodate this huge amount of data. For instance, to machine a die surface for an automobile quarter panel (as shown in Fig. 1) which contains more than 600 surfaces, the tool path data could take as much as 20 Mbytes of memory. As a result, the off-line generated tool path data have to be broken into many batches and downloaded one at a time after a previous batch has been executed.

NC tool path generation has evolved over the last 30 years, influenced by available technologies, market needs, and other forces outside of manufacturing. In the 1960s, NC machines had to be programmed manually. The programming process was very slow and error prone. The availability of computer aided methods and the need for speed and responsiveness continue to reduce reliance on this method. Since the 1970s, more research has been done in computer aided programming for NC machines. In general, there are three types of techniques for tool path generation.

- APT-based tool path generation method. Explicit tool guiding surfaces (drive surfaces) are introduced in this method. The basic idea is that the cutter is moved in one direction while maintaining contact with both the part surface and the drive surface (Faux and Pratt, 1981; Bobrow, 1985). At each step, numerical iteration searches are made to locate the cutter position within a specified tolerance limit. The disadvantage of this method is that the iterative computation is time consuming and there is no guarantee that the iterations will converge for irregularly curved sculptured surfaces.

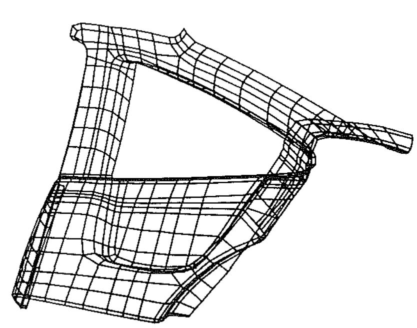


Fig. 1 An automobile quarter panel surface represented in IGES format

Contributed by the Production Engineering Division for publication in the JOURNAL OF ENGINEERING FOR INDUSTRY. Manuscript received Aug. 1991; revised June 1992. Associate Technical Editor: S. G. Kapoor.

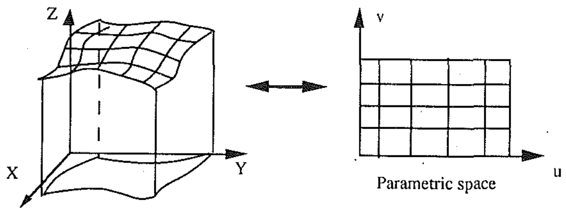


Fig. 2 Parametric Spline Surface (IGES Type 114)

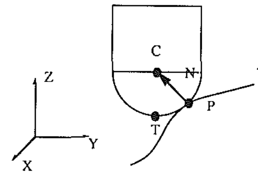


Fig. 3 Tool position definitions

- Cartesian machining method. Tool paths are planned on the XY-plane of a Cartesian coordinate system. The cutter path is the intersection of the part surface and a vertical plane perpendicular to the XY-plane (Choi et al., 1988, 1989; Bobrow, 1985). In this method, the cutter plane has to be a vertical plane. Because gouging points may not be on the vertical cutter plane, the search for gouging points must be performed on the entire part surface.
- Parametric machining method. Tool paths are planned on the parametric space. There are two ways to calculate the tool paths. (*1*) The tool is moved at equally spaced points on the u-line or v-line of the surface. Corresponding to each selected point on the u-v domain, a (x, y, z) point on the part surface can be calculated and used as tool paths. The drawback of this scheme is the difficulty of controlling the tolerance on a part surface based only on the u-v domain point interval. The equally spaced points in the parameter space may result in differently spaced points on the part surface (Bobrow, 1985). (*2*) The points and normal vectors on a part surface are first evaluated at predefined small intervals of both parameters u and v. The tool path is then generated by searching through this huge data array of part surface points in a given cutter plane. This method is widely used in commercial NC software. The operations of filters are performed in order to generate the smallest tool path point and reduce the tool path data size. The filler operations are also necessary to insert points in the interval if it is not small enough. The operations of filler and filter are very time consuming.

All these off-line tool path generation methods have a common drawback. There is always a tradeoff between the level of path precision, in terms of the coarseness of tool path points on a sculptured surface, and the memory requirement.

This paper presents a new method for the real-time generation of NC tool paths to produce free-form sculptured surfaces on a 3-axis milling machine. This method is intended to eliminate the need for the tradeoff between path precision and memory requirement. The unique feature of this real-time path generation method is that it can generate as many points as needed within each computational cycle time. The method is designed to generate NC tool paths directly from IGES 114 data files since the IGES is the main standard electronic data translator recognized by the ANSI.

A new CNC controller is being developed based on the Multibus II structure and the iRMX-III operating system. The controller is a true multi-processor and multi-tasking system. One of the processors in this controller is used as a Free-Form Surface Milling Tool Path Processor, which can read IGES data directly from a CAD data file and generate the CNC tool path during machining operations. The tool path data is then passed to motion controller processors through the Multibus in real time.

The definitions of IGES surfaces and the derivation of their offset surfaces are given in Section 2. The proposed tool path generation algorithms for free-form sculptured surfaces are presented in Section 3. The issues of tool path planning, tool step calculation, gouging detection, tool path interval, and trimmed surfaces are discussed in detail. These developed algorithms are verified through actual cutting tests, and the results are presented in Section 4.

2 IGES Surfaces and Their Offset Surfaces

IGES is a widely used CAD data file structure. All major CAD software packages support the IGES input and output format. The IGES data format is used as the input data format of this new real-time tool path generation method.

The focus of this paper is on the machining of free-form parametric surfaces on a 3-axis milling machine. It is assumed that the surfaces can be reached by the milling cutter in the positive Z direction. The entity 114 in IGES can represent these free-form parametric surfaces. IGES gives all information used to define the free-form parametric spline surfaces. A detailed definition of IGES surfaces can be obtained from Ref. NBS (1988). The free-form parametric spline surface shown in Fig. 2 can be expressed as:

$$\mathbf{S}(u,v) = x(u,v)\mathbf{i} + y(u,v)\mathbf{j} + z(u,v)\mathbf{k} \tag{1}$$

Let $\mathbf{S}(u,v)$ be a part surface parameterized and oriented by $\mathbf{N}(u, v)$, a differentiable normal vector defined on the whole surface. An offset surface to S is a parameterized surface $\mathbf{O}$ (u, v) given by:

$$\mathbf{O}(u,v) = \mathbf{S}(u,v) + f(d) \cdot \mathbf{N}(u,v), \tag{2}$$

where

$$\mathbf{N}(u,v) = \frac{\dfrac{\partial \mathbf{S}(u,v)}{\partial u} \times \dfrac{\partial \mathbf{S}(u,v)}{\partial v}}{\left| \dfrac{\partial \mathbf{S}(u,v)}{\partial u} \times \dfrac{\partial \mathbf{S}(u,v)}{\partial v} \right|},$$

and $f(d)$ is an offset distance function and a constant for machining with a ball-end cutter. With more general cutter geometry and 5-axis machining, further investigation is necessary to determine this offset distance function.

Cutting tools are not an idealized geometrical point and have a finite radius ball tip as shown in Fig. 3, where point P is called the cutter contact (CC) position and point C is the cutter location (CL) and T stands for the tip position. The tool path is often referred to as the cutter location (CL) data, which are to be generated from designed part surface data and used by the CNC controller to drive the cutting tool.

3 Proposed Tool Path Generation Algorithm for Free-Form Surfaces

The basic requirements for the real-time tool path generation are that (*1*) the tool path on a cutter plane should nave no gouging problem with points that are not in the cutter plane, (*2*) the length of tool steps should be adjustable based on the tolerance and condition of the surface to be machined, and (*3*) the algorithm should be able to handle the trimmed curves on the surfaces. The following issues should be addressed:

- Tool path planning
- Tool step length calculation
- Gouging problem
- Tool path interval
- Trimmed surface.

3.1 Tool Path Planning. The Cartesian and parametric machining methods mentioned above are combined in the development of the proposed new real-time path generation al-

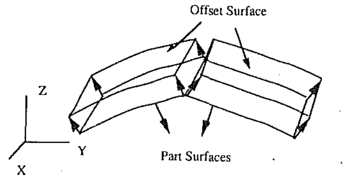


Fig. 4 Offset surfaces of IGES parametric spline surfaces

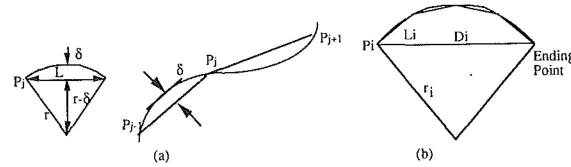
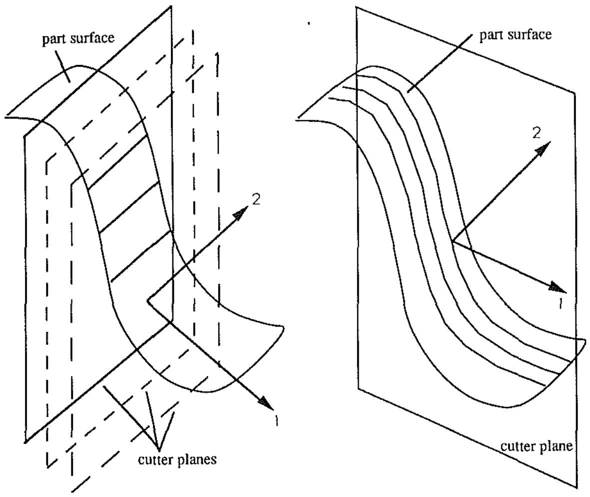


Fig. 6 Calculation of tool step length



(a) Cutting direction not recommended (b) Cutting direction recommended

Fig. 5 Selection of cutting directions for simple surfaces

gorithm. It is important that the offset surface shown in Fig. 4, *instead of* a part surface, be used and intersected by a cutter plane in the new algorithm. This feature ensures that there will be no gouging points outside of the cutter plane. It is only necessary to check the gouging with points on the cutter plane.

Unlike the Cartesian method, the cutter plane in this new algorithm will not necessarily be a vertical plane. It can be any general plane defined in the XYZ space:

$$AX+BY+CZ+D=0. \tag{3}$$

This cutter plane can be oriented and positioned in the XYZ space by varying the coefficients A, B, C, and D. The advantage of this selection of cutter planes compared with the Cartesian method lies in the fact that the new method can be better suited for the application to 5-axis contour cutting, where it is desireable to keep the cutter plane always normal to the part surfaces.

For simplicity in 3-axis machining, the coefficients A, B, C, and D of the cutter plane can be calculated from a predetermined cutting direction (i, j, k) and a starting point (X_0, Y_0, Z_0) on the part surface. This cutting direction should be chosen such that the produced surface has minimum errors. As illustrated in Fig. 5, the surface changes its surface normals drastically along axis 1 and has a relatively small change in surface normals along axis 2. Therefore, the cutting direction along axis 2 will be a better choice as compared to that along axis 1.

The cutter path will be on the intersection line between the cutter plane and the offset surface. The equation for this curve is:

$$AX(u,v)+BY(u,v)+CZ(u,v)+D=0, \tag{4}$$

where $X(u, v)$, $Y(u, v)$, and $Z(u, v)$ are the coordinate functions for the offset surface and are given in reference NBS (1988). Since $X(u, v)$, $Y(u, v)$, and $Z(u, v)$ are nonlinear functions of parameters (u, v), the above equation is a two-variable nonlinear equation.

The procedure to calculate the tool position is as follows. A cutter plane is first selected in the XYZ space, and the intersection curve between the cutter plane and the offset surface is found using Eq. (4). By mapping this intersection curve in the XYZ space into the u-v domain, a curve in the u-v domain can be obtained. For any given point in this u-v curve, a unique tool position can be determined in the intersection curve. Therefore, the task of tool path generation will become a task of finding this tool path in the u-v domain.

In order to solve for (u, v) in Eq. (4), i.e., mapping the intersection curve into a u-v curve, another equation or condition is required. The tool step length constraint is used to construct this equation, which can be represented as the following and will be discussed in detail in Section 3.2.

$$F(u,v)=0 \tag{5}$$

The solutions of Eq. (4) and Eq. (5) (tool step length constraint) will give the (u, v) values corresponding to the intersection curve.

3.2 Tool Step Calculation. The determination of the cutter step intervals depends on the tolerance specified for part surfaces and on the interpolation scheme to be employed subsequently. In real-time cutter path generation, the step size can be controlled to be small enough so that linear interpolation between different cutter positions will be sufficiently accurate.

The maximum deviation, δ, between a true intersection curve and the chord connecting two successive cutter positions is measured normal to the chord, as shown in Fig. 6(a). For small steps, it is reasonable to approximate the curve by its osculating circle, and the local curvature of the intersection curve may be used to determine the step length. From geometries, it can be shown that:

$$L^2=4\delta(2r-\delta), \tag{6}$$

where L is the step length, and r is the radius of the curvature at point P_j. When the maximum allowable deviation δ is given as an input parameter and the curvature of the surface is calculated from the equation of the surface, the step length at this position can be calculated based on the above equation.

Given the condition of a surface curvature and the permissible interpolation error, δ, a recursive algorithm is proposed to calculate the tool step interval. First, the intersection points of the cutter plane and edges of the offset surface are calculated. The ending point is used as the reference point for this segment of the tool path, as shown in Fig. 6(b). At each tool position, P_i, the distance from this position to the reference point is denoted as D_i, and the radius of surface curvature at position P_i is r_i, which is approximated by the average of the two main components of the curvature radius at this position. L_i can be calculated using Eq. (6). Knowing D_i, r_i and L_i, the total number of tool steps can be calculated as:

$$N_i=\frac{\sin^{-1}\dfrac{D_i}{2r_i}}{\sin^{-1}\dfrac{L_i}{2r_i}} \tag{7}$$

where N_i represents the total number of linear segments required to approximate the intersection curve from position P_i to the ending point. This procedure is recursive in the sense that when the cutter proceeds to a new position, the parameters

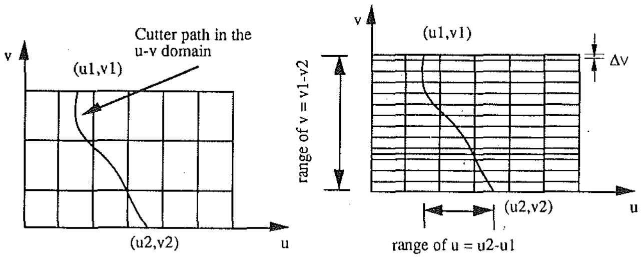


Fig. 7 Segmentation of tool path in parameter *u-v* domain

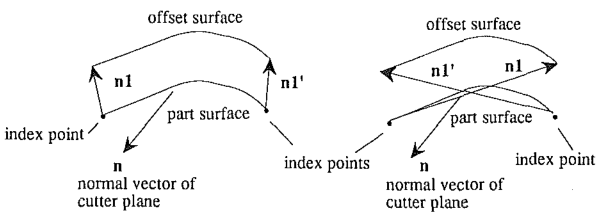


Fig. 8 Triple product of vectors for curve sorting

D_i, r_i and L_i will be updated to their new values at the new position, and Eq. (7) will be applied to give a new total number of tool steps, N_i.

After the total number of tool steps is determined, the tool path in the *u-v* domain will be divided into N_i segments. It is best to divide the tool path in the *u-v* domain along an axis that has a larger parameter range. For example, as shown in Fig. 7, the tool path from $(u1, v1)$ to $(u2, v2)$ has a relatively larger change along the v axis. Therefore, the division tool path will be made along the v axis.

$$\Delta v_i = \frac{(v_i - v_2)}{N_i}$$

or

$$\Delta u_i = \frac{(u_i - u_2)}{N_i} \tag{8}$$

This method is fast and easy to implement. It can adjust the parameter interval based on the curvature of the surface.

3.3 Gouging Problem. After calculating the intersection between the offset surface and the cutter planes, the tool path will become a series of curves that are in the cutter planes. These curves will discontinue at the joint section of the surfaces. Algorithms must be developed to deal with these discontinuities.

Curve Sorting. The order of the surfaces given in IGES format is not arranged by the geometric structure and the information about how to link these curves is not provided by IGES. A curve sorting algorithm is developed to find the connection between different intersection curves. The objective of the curve sorting algorithm is to determine which two curves ought to be connected and how the ends of these two curves are connected.

Two additional orientation index points are added to each intersection curve in this sorting process. These index points are the intersection points of the cutter plane and the part surface boundary, representing the starting and ending points of a part surface boundary. The ending point of one surface will be the starting point of another surface if they are connected to each other.

In order to determine whether the index points are put in the proper positions or not, the triple product of vectors **n**1, **n**1′, and **n** is used to check the relative positions of the starting and ending points of both part surface and offset surface, as shown in Fig. 8. Vector **n** is the surface normal of the cutter plane (which is the paper plane in the drawing), and vectors **n**1 and **n**1′ are the vectors connecting the starting and ending index points, respectively, to the edges of the offset surface. If $\mathbf{n}\cdot(\mathbf{n}1 \times \mathbf{n}1') < 0$, the index points on a part surface is properly connected to the starting and ending points on its offset surface. If $\mathbf{n}\cdot(\mathbf{n}1 \times \mathbf{n}1') > 0$, these points are not properly connected.

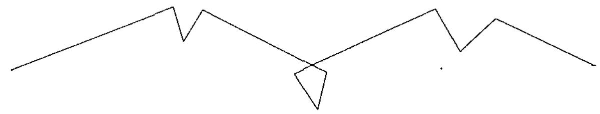


Fig. 9 Connection of tool paths in the cutter plane

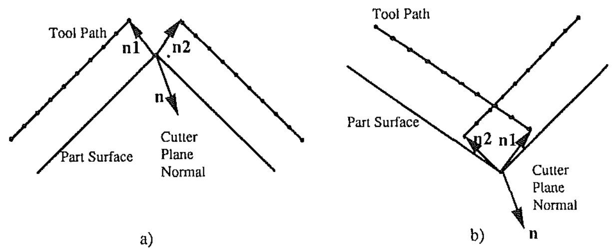


Fig. 10 Determination of filleting or trimming conditions between two tool path segments

After the index points associated with each tool path segment are properly labeled with a starting or ending mark, they are used to identify the connection between different tool path segments. If the ending index point of a tool path segment is determined to be equal to the starting index point of another segment, these two segments should be connected together.

Another faster method to sort the order of the curves is to sort them based on the distances from their starting and ending points to a given fixed reference point. These distances are calculated using Eq. (9).

$$DIS_s(i) = \sqrt{(X_s(i) - X_r)^2 + (Y_s(i) - Y_r)^2 + (Z_s(i) - Z_r)^2}$$

$$DIS_e(i) = \sqrt{(X_e(i) - X_r)^2 + (Y_e(i) - Y_r)^2 + (Z_e(i) - Z_r)^2}, \tag{9}$$

where subscript s stands for starting point, subscript e stands for ending point, and subscript r stands for the reference point.

For instance, let us assume that the fixed reference point is on the far left side of all curves and tool path segments starting from left to right when properly oriented. Then the distance from the starting point of the tool path segment to the reference point should be smaller than that from the ending point. Otherwise, the tool path segment is placed in a wrong orientation. By checking the distances from both the starting and ending points of each tool path segment, its orientation can be determined.

After sorting the curves, all calculated tool path curves can be linked from the ending index point of one curve to the starting index point of the next curve, as shown in Fig. 9.

Connecting Surfaces. It can be seen from Fig. 9 that for two surfaces with a convex joint there is a gap between their offset surfaces and for two surfaces with a concave joint there is overlap between their offset surfaces. In order to connect these tool paths, corner fillets and/or edge removals are necessary. To check if the tool path segment needs to be removed or filleted, another triple product is used. If $\mathbf{n}\cdot(\mathbf{n}1 \times \mathbf{n}2) < 0$, an interpolation or fillet is needed between these two tool path segments, as shown in Fig. 10(*a*). If $\mathbf{n}\cdot(\mathbf{n}1 \times \mathbf{n}2) > 0$, these two tool path segments are gouged, and a removal operation is needed, as shown in Fig. 10(*b*).

In either case, an intersection point, P_0, should be determined, as shown in Fig. 11. If a removal operation is required,

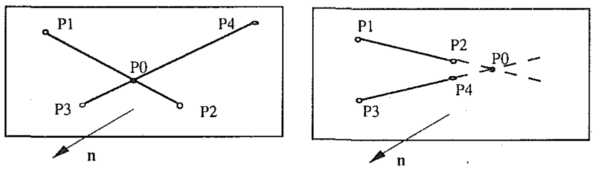


Fig. 11 Corner fillet and gouging removal between two tool path segments

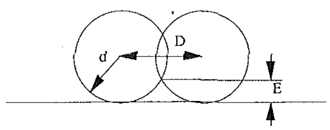


Fig. 12 Calculation of tool pass interval

the segments between the intersection point, P_0, and the two close end points will be removed. Otherwise, when a filleting operation is performed, the segments between the P_0 and the two close end points will be added to the final tool path.

For a given size of tool radius, any sharp corner with a radius smaller than the tool radius cannot be machined. This is true for both on-line and off-line tool path generation methods. However, the proposed system has a nice feature. If a sharp corner is detected, some portion of the tool path will be removed, as shown in Fig. 11. The removed portion of the tool path will be stored in a separate file so that the stored information can be used in a final finish cut performed using a tool with a smaller radius.

3.4 Tool Pass Interval. Tool pass interval is the interval between two subsequent passes of a cutter on the surface. It is the distance between the two cutter planes. The cutter plane is moved in parallel by a distance D, which can be calculated from a specified cusp error, as shown in Fig. 12. If E is the specified cusp error and d is the diameter of a ball-end milling cutter, the distance between the two cutter planes can be calculated from

$$D_i = 2\sqrt{2dE - E^2}. \tag{10}$$

3.5 Trimmed Surface. The above discussion on corner fillet and surface trimming is mainly for conditions resulting from offsetting a free-form surface. In the IGES data format, there are specific entities, such as entities 142, 144, and 106, that are used to specify the trim curves in the given parametric surfaces.

For instance, IGES entity 106 specifies a trim line by a series of (u, v) data in u-v domain. These data can be mapped to the XYZ space using Eq. (4) to form a trim curve, as shown in Fig. 13.

The mapped points on the trim curve will be used to find the trim points in the cutter plane. For each given mapped trim point, its distance to the cutter plane can be treated as an index and calculated using

Distance (from trim point i to cutterplane)

$$= \frac{AX(i) + BY(i) + CZ(i) + D}{\sqrt{A^2 + B^2 + C^2}} \tag{11}$$

for i = 1, . . . , number of trim line points.

This calculated distance index has different signs for points at different sides of the cutter plane. If the points are on the positive side of the cutter plane, the distances will be a positive number, otherwise they will have a negative sign. The zero-cross points are detected by checking the signs of the distances. The intersection point between a line linking two adjacent zero-cross points and the cutter plane is found to be the trim point in the cutter plane.

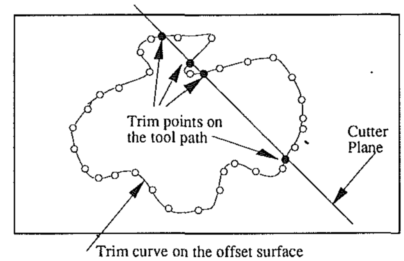


Fig. 13 A trim curve specified using entity 106

The tool path calculated in Section 3.2 is first trimmed by these trim points and is then used to analyze the connection condition by the algorithms discussed in Section 3.3.

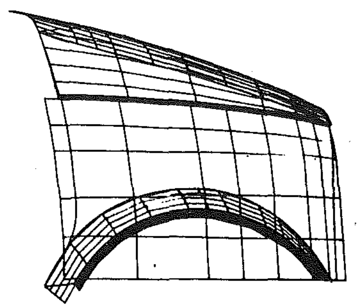
4 Experimental Results

The real-time tool path generation algorithms developed in this paper have been implemented in a Multibus II based CNC controller. The algorithms reside on a 80386 single board computer, called the Free-Form Surface Milling Tool Path Processor, with a clock rate of 25 Mhz. CAD data files in the IGES format can be read by the processor, and 3D milling tool paths can be generated in real time during cutting.

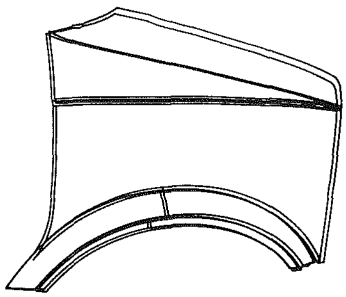
A procedure is sketched below to illustrate the execution of this real-time NC tool path generator:

Tool_Path_Generation
Input Parameter:
  IGES File Name
  Cutter Size
  Cutter Direction (i, j, k)
  Tolerance for Chord Error
  Tolerance for Cusp Error
Main {
  Load IGES CAD data file/* **read IGES to memory** */
  Calculate boundaries for all surfaces;
  Do { **Put Cutter Platter at a start position;**
    For (Each Surfaces of IGES 114) {
      If (The Cutter Plane pass:
        —Trim Curve if surface trimmed;
        —Surface Boundary if surface not trimmed) {
      Find the start and end position of part surface;
      Find the start and end position of normal offset surface;
      Calculate cutter location data on normal offset surface;
      Trim the cutter location data by trim curve;
      not_pass_surface = = FALSE;
      } /* end of if */

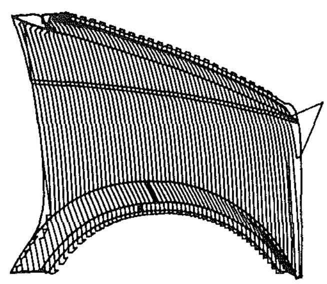
      else{ **Cutter plane do not pass this surface**
        not_pass_surface = = TRUE;
      } /* end of else */
    } /* end of for */
    For (All surface passed by the cutter plane) {
      Sort the cutter location data of all surface;
      If (ith segment index point = = i + 1st segment index point) {
        If (ith segment gouged with i + 1st segment) {
          Find Intersect Point;
          Trim cutter location data by Intersect Point;}
        else {/* **ith segment not gouged with i + 1st segment***/
          Find Interpolation Point between these segments}
      else {/*ith segment index point ≠ i + 1st segment index point*/



(*a*) A truck fender in IGES representation



(*b*) Fender surface and its normal offset surface



(*c*) Tool path for fender rough cutting

Fig. 14 Surface of truck fender and NC tool path

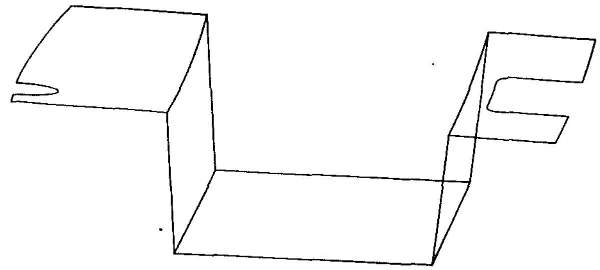


Fig. 15 Computer representation of the test part

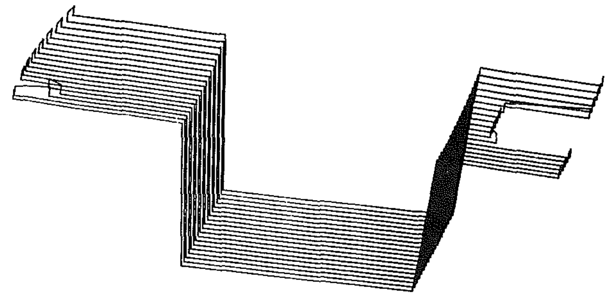


Fig. 16 Tool paths for the test part generated by the developed algorithm

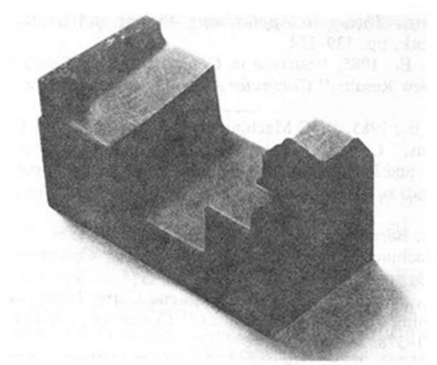


Fig. 17 The photograph of the machined test part

Move tool to safe zone between these two segments;
{/* end of for */
Pass Cutter Location data to Motion Processor through Multibus;
Move cutterplane to another position based on cusp error tolerance;
{ **while (not_pass_all_surface = = FALSE)** /* end of do */
}/* end of main */

A fender of a truck is used to generate the NC tool path based on the algorithm proposed in this paper. Figure 14(*a*) shows the IGES definition data. Figure 14(*b*) shows the surface representation and the normal offset surface of this fender. The NC tool path for rough cutting is calculated, as shown in Fig. 14(*c*).

Another part shown in Fig. 15 is used in a real cutting experiment since this part contains more gougings, which the real-time algorithm has to process correctly. When cutting a 90-degree corner with a ball-end milling cutter, one would encounter the gouging problem. Using the above tool path generation algorithm, the tool paths have been generated in real time and are stored for illustration purpose. It can be seen that no overcut situations have occurred. In order to display properly, only a rough cutting tool path has been shown in Fig. 16.

Actual machining has been conducted based on the following experimental conditions: (*1*) The radius of the ball-end milling cutter is 3.175 mm. (*2*) The rough cutting is conducted with 2 mm liftoff, 0.02 mm chord error, and 0.5 mm cusp error. (*3*) Tolerable cusp and chord errors for finishing cutting are specified to be 0.02 mm. (*4*) Spindle speed is 2000 rpm and feedrate is 150 ipm. (*5*) Workpiece material is wax. It takes less than 1 second to generate one tool path across the entire work piece surface,which contains on average more than 100 cutter location points. The workpiece has a dimension of 4 inches by 6 inches.

Figure 17 shows the part machined using the real-time tool path generation algorithm. It can be concluded from the machined part that the developed algorithm can properly handle problems such as trimming lines, gouging detection, and control of tool step sizes in real time. The surface contouring accuracy of the machined part is also checked on a coordinate measuring machine. It is found that the surface contour error of the cut part is less than 0.02 mm, which is within the specified tolerance.

5 Conclusion

A real-time NC tool path generation algorithm has been developed. Features of this new algorithm include:

(*1*) Three-dimensional NC tool paths can be generated in real time during cutting, which eliminates the compromise between desired surface contouring tolerances and the required memory sizes for storing off-line generated NC tool path data. This provides greatly increased surface contouring accuracy without lengthy NC code download time and huge memory storage requirements.

(2) This new algorithm directly uses IGES based CAD data files as its input, which makes it compatible with a wide range of CAD software packages.

(3) This new algorithm can automatically adjust its tool step size according to the change in surface curvatures, and, thus, can yield better surface smoothness and uniformity.

(4) Through computer simulation and actual machining, it has been verified that the developed real-time NC tool path generation algorithm can deal adequately with issues such as trimming lines, gouging, and adaptive tool step adjustment. Cutting results have shown that no violating conditions occurred on the machined part surfaces and the surface contour error of the cut part is less than the specified tolerance, 0.02 mm.

References

Anderson, R. O., 1978, "Detecting and Eliminating Collisions in NC Machining," *Computer Aided Design*, Vol. 10, pp. 231–238.

Armstrong, G. T., Garey, G. G., and Pennington, A. D., 1984, "Numerical Code Generation from A Geometric Modeling System," *Solid Modeling By Computers, from Theory to Applications*, Pickett and Boyse, eds., Plenum Press, New York, pp. 139–154.

Barnhill, R. E., 1985, "Surface in Computer Aided Geometric Design; A Survey with New Results," *Computer Aided Geometric Design*, Vol. 2, No. 1, pp. 1–17.

Bobrow, J. E., 1985, "NC Machine Tool Path Generation From CSG Part Representations," *Computer-Aided Design*, Vol. 17, No. 2, pp. 69–76.

Boehm, W., and Farin, G., and Kahmann, J., 1984, "A Survey of Curve and Surface Methods in CAGD," *Computer Aided Geometric Design*, Vol. 1, No. 1, pp. 1–60.

Choi, B. K., Barash, M. M., and Anderson, D. C., 1984, "Automatic Recognition of Machined Surface from 3D Solid Model," *Computer Aided Design*, Vol. 16, No. 2, pp. 81–86.

Choi, B. K., and Jun, C. S., 1989, "Ball-end Cutter Interference Avoidance in NC Machining of Sculptured Surfaces," *Computer Aided Design*, Vol. 21, No. 6, pp. 371–378.

Choi, B. K., Lee, C. S., Hwang, J. S., and Jun, C. S., 1988, "Compound Surface Modeling and Machining," *Computer Aided Design*, Vol. 20, No. 3, pp. 127–136.

Culbreth, D. N., 1989, "Manufacturing Model: an Integrated Approach to Planning, Design and Managing Industrial Facilities," *Computer Aided Design*, Vol. 21, No. 1, pp. 49–53.

Drysdale, R. L. and Jerard, R. B., 1987, "Discrete Simulation of NC Machining," *Proc. ACM Symp. Computational Geometry*, June.

Duncan, J. P., and Mair, S. G., 1983, *Sculptured Surfaces in Engineering and Medicine*, Cambridge University Press.

Eversheim, W., and Holz, B., 1982, "Computer Aided Programming of NC-Machine Tools by Using the System AUTAP-NC," Annals of the CRIP, Vol. 31, p. 323.

Farouki, R. T., 1987, "The Approximation of Non-degenerate Offset Surfaces," *Computer Aided Design*, Vol. 4, pp. 3–16.

Faux, I. D., and Pratt, M. J., 1981, *Computational Geometry for Design and Manufacture*, John Wiley.

Ferstenberg, R., Wang, K. K., and Muchstadt, J., 1986, *Automatic Generation of Optimized 3-Axis NC Programs Using Boundary Files*, IEEE.

Gregory, C. L., and Ozsoy, T. M., 1987, "NC Machining of Free Form Surfaces," *Computer Aided Design*, Vol. 19, No. 2, pp. 84–89.

NBS, National Bureau of Standards, U.S. Department of Commerce, 1988, "Initial Graphics Exchange Specification (IGES) Version 4.0, NBSIR 88-3813.

Jayaram, S., and Myklebust, A., 1990, "Automatic Generation of Geometry Interfaces Between Applications Programs and CADCAM Systems," *Computer Aided Design*, Vol. 22, No. 1, pp. 50–56.

Jerard, R. B., Drysdale, R. L., Hauck, K., Schaudt, B., and Madgewick, J., 1989, "Methods for Detecting Errors in Numerically Controlled Machining of Sculptured Surfaces," *IEEE Computer Graph. Appl.*, Vol. 9, No. 1.

Kakazu, Y., Okino, N., and Hoshi, K., 1974, "The Penalty Method of Determining Automatically a NC Cutter Path," *CIRP Conference*.

Mills, R. B., 1984, "Computer Graphics for NC Programming," *CAE*, Nov., pp. 32–42.

Oliver, J. H., and Goodman, E. D., 1990, "Direct Dimensional NC Verification," *Computer Aided Design*, Vol. 22, No. 1, pp. 3–10.

Persson, H., 1978, "NC Machining of Arbitrarily Shaped Pockets," *CAD*, Vol. 10, No. 3, May.

Press, K., and Kaplansky, E., "Automated Part Programming for CNC Milling By Artificial Intelligence Techniques," *Journal of Manufacturing System*, Vol. 4, No. 1, pp. 51–63.

Suh, Y. S., and Lee, K., 1990, "NC Milling Tool Path Generation for Arbitrary Pockets Defined by Sculptured Surfaces," *Computer Aided Design*, Vol. 22, No. 5, pp. 274–284.

Thomas, S. W., 1990, "Scanline Rendering for 3-Axis NC Toolpath Generation, Simulation, and Verification," CSE-TR-43-90, The University of Michigan.

Wang, K. K., and Wang, W. P., 1981, "A PADL Based Numerical Control Machining Data Generation Program," *Annals of the CIRP*, Vol. 30, pp. 359–362.

Wang, W. P., and Wang, K. K., 1986, "Real-Time Verification of Multiaxis NC Programs with Raster Graphics," *IEEE Proc. Int'l. Conf Robotics and Automation*, Los Alamitos, CA, CS Press, pp. 166–171.