일반적인 내벽을 가진 자유바닥 곡면 파켓의 NC 가공을 위한 단일화된 황삭과 정삭 알고리즘 - Part 2. Experiment

(An unified rough and finish cut algorithm for NC machining of free form pockets with general polygon - Part 2. Experiment)

최 용 훈*, 김 상 진**
(Yong-hoon, Choi, Sang-jin, Kim)

요 약 3축 CNC 공작기계상에서 자유형상의 벽을 가진 자유곡면을 가공할 때 최종 NC (Numerical Control) 코드를 만들기 위한 공구 경로는 효율적으로 결정 되어져야 한다. 단일화된 황삭과 정삭 알고리즘 및 공구경로가 이미 그래픽으로 Part 1에서 시뮬레이션 되었다. 본 연구에서는 Part 1에서 보여진 시뮬레이션 결과를 3축 CNC 공작기계를 이용하여 자유곡면 바닥을 가진 일반적인 파켓 밀링을 위한 직선보간 및 직선+아크를 혼합한 보간을 위한 NC 공구경로 데이터로 만들어서 3D 격자 항행 알고리즘을 실험하였고 그 효용성을 증명하였다.

핵심주제어 : NC 공구경로, 직선보간, 직선+아크보간, 3D 격자 항행 알고리즘

Abstract NC (Numerical Control) code for the tool path needs to be generated efficiently for machining of free form pockets with arbitrary wall geometry on a three axis CNC machine. The unified rough and finish cut algorithm and the tool motion is graphically simulated in Part 1. In this paper, a grid based 3D navigation algorithm simulated in Part 1 for generating NC tool path data for both linear interpolation and a combination of linear and circular interpolation for three-axis CNC milling of general pockets with sculptured bottom surfaces is experimentally performed and verified.

Key Words : NC tool path, linear interpolation, linear and circular interpolation, 3D navigation algorithm

1. INTRODUCTION

One of the most important features in determining CNC machining efficiency and productivity is the cutter path motion planning especially for complicated form & functional surface shapes. This cutter motion on compound-curvature surfaces determines the machihing time and surface

* 한국전자통신연구원 선임연구원
** 시립인천전문대학 컴퓨터제어과 교수

roughness(or cusp height). it is to be noted the surface roughness always exist because of the lack of geometry matching between the cutter and workpiece surface. Most CAD/CAM systems connect the data points with linear segments to generate a NC tool path. This method creates many short linear move commands with sudden changes in direction. Tool path created in this way can cause problems for the CNC machine during the machining process, and this can affect the finished quality of the part.

Moreover, machining with linear segments leads to rough surface due to interpolation errors and significantly longer machining times due to repeated *stop and go* operations. Curvilinear machining algorithms can reduce these problems. Moreover, with curvilinear machining, the cutter location file size can be reduced for machining the same amount of volume. This is possible since curved data only needs starting and ending points and arc center or intermediate control points while linear movements need all data points to specify the tool path.

The grid base unified rough and finish cut algorithm simulated in part 1 with OpenGL is verified experimentally [1].

The procedure in Figure 1 to make NC code for the experiment is the same as simulation process.

In this paper, a grid based navigation scheme is used to plan the cutter path. A rectangular grid is created inside the pocket. The grid size and the cutter size are chosen in an iterative fashion to ensure that a predefined tolerance for surface roughness is satisfied.

In the interior of the pocket, tool motion proceeds from one grid center point to the next, based on a navigation algorithm. The pocket wall is machined by determining a series of corner points. The length of the tool path is optimal in a "greedy heuristic" sense. Additional zigzag motions along the X and Y directions may be required at some grid points to satisfy surface roughness tolerances, and may be added optionally at other locations to obtain a smoother surface.

The grid points are later post-processed to generate a combined linear interpolation and arc interpolation sequence for machining. the circular arc-machining algorithm also results in less machining time and smaller CL files. It may be noted that for advanced NURB interpolators, grid points may be post-processed for representation as NURBS. NC code is automatically generated and experimentally validated.

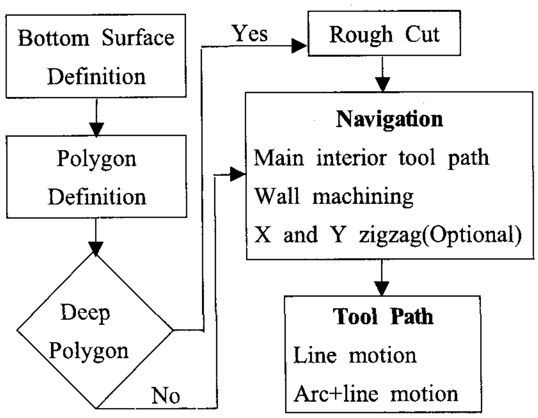


FIGURE 1. Schematic representation of the unified rough and finish cut algorithm for machining free form pockets with arbitrary wall machining.

2. RELEVANT LITERATURE

Tool path planning strategy and algorithm to machine complex surfaces has been developed about iso-parametric machining of free form surfaces[2], where the tool is moved along a fixed parameter, parallel plane section machining, where the tool is moved on a curve determined by the intersecting plane and the surface[3], and the pocket machining using window frame or stair case style patterns[4]. Good surveys on tool path planning and numerical control machining algorithms may be found in [5-7].

By using iso-parametric curves as tool paths, costly surface-surface intersection algorithms can be avoided. However, the spacing between iso-parametric curves, in general, will be non-uniform, causing over machining or under machining. Non iso-parametric curves[8] and constant scallop height machining[9] may be used to provide more optimal and valid coverage of the surfaces. These techniques are computationally intensive.

Linear interpolation based curved surface machining can result in thousands of small linear motions and is very inefficient for complex surfaces. Circular interpolation is available on current three-axis CNC systems, and spline and NURBS based interpolation schemes have been proposed[10].

An algorithm to generate final NC codes for machining a general polygon with a sculptured bottom surface was presented and simulated in part 1 [1] and experimental verification of the grid-based navigation strategy for NC machining of free form pockets with arbitrary wall geometry is presented in this paper.

The tool moves along the coordinates on the workpiece, not along the parameters on the surface to prevent the over- and under-machining which happens in iso-parametric machining. Additional zigzag moves may be required or can be optionally added for a smoother surface. the strategy presented here can also be used in applications where NC tool path planning is required for reverse engineered data.

3. POCKET MACHINING ALGORITHM

3.1 Surface, Polygon, & Grid Definition

As described in Part 1, B-spline surface is used as an example to illustrate the algorithm. the B-spline surface formulation uses a control net of 3D points and the B-spline polynomial basis functions for blending. Points on the B-spline surface are specified by the tensor product

$$P(t,u)=\sum_{i=0}^{m}\sum_{j=0}^{n} P_{i,j}N_{i,k}(t)N_{j,l}(u)$$

$$P = P(t,u) = [1/6]^2 [t^3 \quad t^2 \quad t^1 \quad 1] M^T B M \begin{bmatrix} u^3 \\ u^2 \\ u \\ 1 \end{bmatrix}$$

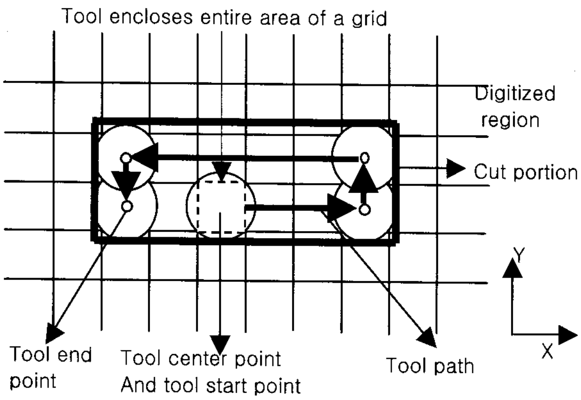
$$M = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 0 & 4 \\ -3 & 3 & 3 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} p_{00} & p_{01} & p_{02} & p_{03} \\ p_{10} & p_{11} & p_{12} & p_{13} \\ p_{20} & p_{21} & p_{22} & p_{23} \\ p_{30} & p_{31} & p_{32} & p_{33} \end{bmatrix}$$

where, $P_{i,j}$ are the vertices of the control net and $N_{i,k}$ and $N_{j,l}$ are the basis functions. B-spline surfaces do not necessary pass though the corner points of the characteristic polygon, and have greater local control.

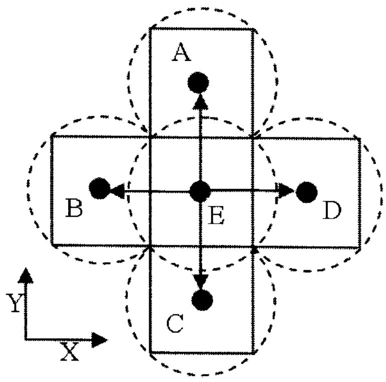
Any general polygon can be defined for pocketing. A general polygon means that it includes both convex and concave shapes. The pocket wall shape is specified by the polygon and the bottom surface of the pocket is specified by the B-spline surface.

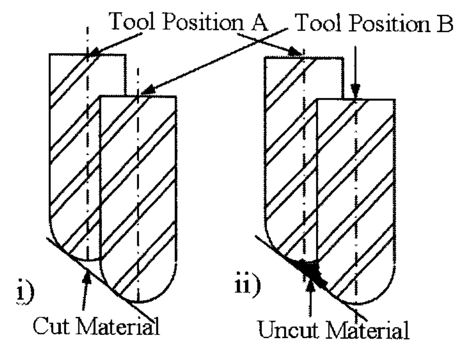
The region to be machined is divided into a grid with size equal to, or less than, $D/\sqrt{2}$, where, D is the diameter of the tool, as shown in Figure 2. from each of the grid center-points, the tool moves to one of four adjacent grid centers. When the pocket has an arbitrarily shaped bottom surface, uncut material may still remain because of the geometry mismatch between the desired surface and the ball-end mill. Figure 2 also shows an uncut region because the tool center did not pass through the uncut region. For three-axis machining, this situation will occur whenever there are any regions that the tool center does not pass through in the X- or Y-directions.

To reduce the surface roughness, additional X and Y zigzag motions may be required for later execution. The selection of grid size based on the specified tool size is important since it determines the machining time and surface roughness. A larger grid leads to less machining time and larger surface roughness, and vice versa.



(a) the tool encloses the entire area of the grid





(b) (i) The tool moves directly from A to B

(ii) The tool does not move directly from A to B

FIGURE 2. Grid definition and uncut material in ball-end milling

3.2 Predefined Surface Roughness

Surface roughness (scallop height) is proportional to the relation between tool diameter and grid size. As the tool diameter becomes larger for a given grid size, the roughness is reduced. The relation between CL(cutter location) and CC(cutter contact) of the tool must be considered to determine surface error and surface roughness. The tool moves along the center points of the grids (CL). For a flat bottom surface the CL and CC points would be identical.

However, for an inclined bottom surface, these points are different as can be seen in Figure 3. The magnitude of the difference between CL and CC points is also proportional to the tool diameter and grid size. This should be considered when a complex and more accurate surface is required. The following relations can be deduced from Figure 3.

$$\theta = \tan^{-1}(Z\,inc\,/\,unit\,grid\,size)$$

$$Surface\,Error \cong R(1-\cos\theta)$$

$$\frac{AB}{AD} = \frac{EB}{EC} \text{ as } \Delta\text{'s ABD and EBC are nearly similar}$$

$$\therefore \frac{\sqrt{unit\,grid\,size^2 + (R - Zinc)^2}}{R\cos\theta}$$

$$\cong \frac{\sqrt{unit\,grid\,size^2 + (R - Zinc)^2} - R}{Scallop\,Height}$$

$$\therefore Scallop\,Height$$

$$\cong \frac{R\cos\theta(\sqrt{unit\,grid\,size^2 + (R - Zinc)^2} - R)}{\sqrt{unit\,grid\,size^2 + (R - Zinc)^2}}$$

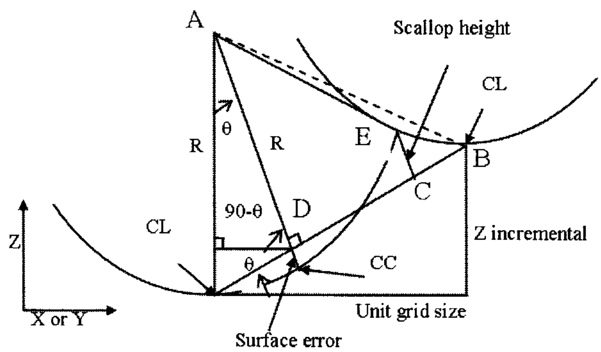


FIGURE 3. The relation of cutter location (CL) and cutter contact (CC) points to surface error and roughness.

A detailed iterative algorithm for calculating the grid size based on a computation of surface roughness for a predefined when the surface roughness tolerance is exceeded. It is possible that the grid is extra fine at some locations, and at these points, zigzag motion may not be needed and is omitted.

3.3 Navigation

The navigation strategy plans a sequence of legal moves from one grid center-point to the next as shown in Part 1. A set of priorities governs the choice of move. The priority depends upon the previous or precedent moves. Generally, the direction in which to go is always the first permissible direction in a counterclockwise direction.

3.4 Machining by Linear Interpolation

Linear interpolation proceeds by line movement from one grid center-point to the next. However, there are a large number of points, which causes machining delay and large CL files.

3.5 Machining by Curvilinear Interpolation

The navigation grid points can be post-processed for curvilinear machining. A recursive approach to approximate a curve by a series of arcs and lines is used. Threshold values for the arc radius are used to obtain valid arcs for circular interpolation. If the threshold radius is exceeded, then linear interpolation is used.

A curve can thus be represented by a series of arcs and lines, as shown in Figure 4. If switching from arcs in the XZ to the YZ plane, the active machining axis must be changed.

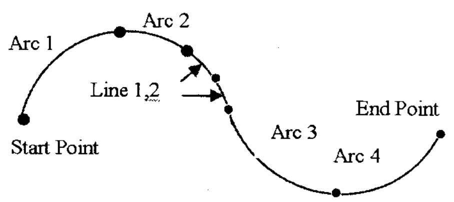


FIGURE 4. Illustration of Arc + Line Movements

3.6 Complete Pocketing Algorithm

The entire pocketing algorithm is now summarized.

Input values

1. Vertex points of pocket.
2. Tool diameter
3. Grid size in X- and Y- directions.
4. Predefined tolerance of surface roughness.

Interior navigation for main tool path

1. Model sculptured bottom B-spline surface.
2. Define pocket polygon and overlay X, Y grid
3. For each grid point obtain corresponding Z value. Call this ordered set G.
4. Calculate maximum surface roughness. Redesign input values if predefined tolerance is exceeded. Recalculate set G.
5. Find a set of cutter edge extreme points at each pocket vertex. Call this polygonal set H.
6. Determine the set of all grid-centers inside H. Call this new set S.
7. Find the navigation path(in G and not in S) from the point that has the left-most minimum Y value. Call this set L. Store X and Y zigzag motions separately in a set N.
8. Reverse the navigation path (L). Call this set M (main tool path).

Wall machining and X zigzag and Y zigzag tool paths

1. Find the nearest neighborhood point within set H to begin wall machining.
2. Calculate tool centers between each pair of vertices in set H. Call this set F.
3. Sort H and F in CCW order. Call this new set W(Pocket wall path).
4. Final navigation path is given by the set $T=\{M \cup W \cup N\}$.
5. Determine line movements or arc + line movements for machining.

4. CASE STUDY

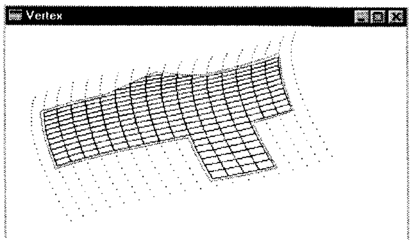
The proposed pocketing algorithm is implemented in Visual C++ and OpenGL for graphical simulation in Part 1. NC code is then automatically generated and machined on a three-axis vertical machining center.

4.1 Machining of Free Form Pocket

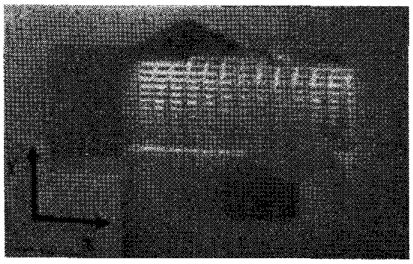
Input
1. Vertex points of pocket.
2. Tool diameter: 6.35 mm.
3. Grid size along X: 3.18 mm
4. Grid size in Y: 1.59 mm
5. Predefined tolerance : 1.02 mm

Output
1. Output is a set of pocket cutter paths.
2. Drawing of sculptured pocket surface.

The B-spline surface used to describe the bottom of the pocket has a mountain in the Y-direction and a mountain and a valley in the X-direction. The Simulation result in isometric view of the pocket along with the defined grid is shown in Figure 5(a).



(a) Pocket simulation



(b) Machined pocket sample

FIGURE 5. Free Form Pocket

Simulation is performed to obtain the tool path to machine the pocket. Initially, the interior grid navigation is carried out, then wall machining is performed, and finally zigzag motions are added. The grid navigation points may also be post processed to obtain arc interpolation. Experimentally machined samples are shown in Figure 5(b). The sequence of figures shows the main tool path, wall machining and the addition of the X and Y zigzag motions.

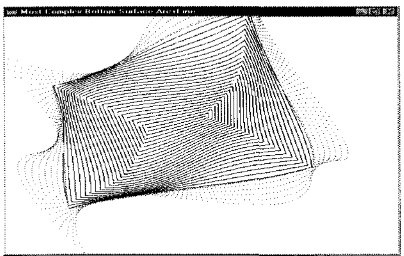
4.2 Machining of Complex Bottom Surface

1. Input

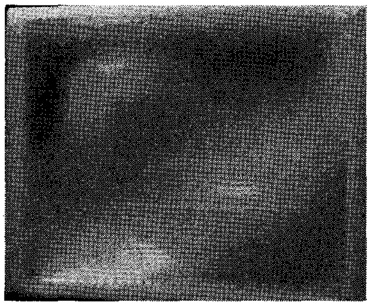
a. Vertex points of a rectangular
b. Tool diameter: 0.25 in
c. Grid size in X and Y direction: 0.05 in each
e. Predefined tolerance: 0.01in

2. Output

Output is a set of cutter path to remove inside area of a rectangular and drawing of free form bottom surface as shown in Figure 6. The pocket is machined using arc and linear interpolation and the simulated path and the corresponding machined sample are shown in each case.



(a) Simulation of complex bottom surface



(b) Machined pocket sample

FIGURE 6. Complex surface pocket
(Max. surface roughness: 0.0089in)

A comparison between linear machining and a combination of linear and curvilinear interpolation is made with variables of machining time and file size. the machining time of linear machining was 7 min 44 sec with the file size of 120 KB while curvilinear interpolation shows 7 min 3 sec machining time and 39 KB file size. Spindle speed was set at s=7500 r.p.m. and 25 inch/min of feed rate.

5. Conclusion

Simulations showed in Part 1 for the machining of general polygon pocket with a free form bottom surface was verified with the machined examples since it reduces the required machining time and NC code file size. The algorithm presented and proved in this paper lays a framework for the implementation of advanced curvilinear interpolation.

In this paper, final NC code data have been generated automatically for both linear and curvilinear movement. The curvilinear algorithms improve the accuracy of machining, and reduce the machining time and file size. the number of tool retractions is expected to be proportional to the number of non-convex vertices in the polygon. If zigzag machining is used, the number of tool retractions would be higher. the algorithm developed in this paper can also be directly applied to machine reverse engineered objects, as it is based on a grid of points.

Reference

[1] Choi, Y., Cho, C., and Kim, S. (2004), "A unified rough and finish cut algorithm for NC machining of free form pockets with general polygon - Part 1. Simulation", Journal of the Korea Industrial Information Systems Society, Vol. 9, No. 1, pp. 7-16.

[2] Loney, G. C. and Ozsoy, T. M. (1987), "NC machining of free form surfaces", Computer-Aided Design, 19(2), pp. 85-90.

[3] Bobrow, J. E. (1985), "Solid modelers improve NC machine tool path generation techniques", Computers in Engineering, 1, pp. 439-444.

[4] Persson, H. (1978), "NC machining of arbitrary shaped pockets", Computer-Aided Design, 10(3), pp. 169-174.

[5] Marshall S. and Griffiths. J. G. (1994), "A survey of cutter path construction techniques for milling machines", International Journal of Production Research, 32(12), pp. 2861-2877.

[6] Jensen, C. G. and Anderson, D. C. (1996), "A review of numerically controlled methods for finish sculptured - surface machining", IIE Transactions, 28, pp. 30-39.

[7] Dragomatz, D. and Mann, S. (1997), "A classified bibliography of literature in NC milling path generation", Computer Aided Design, 29(3), pp. 239-247.

[8] Huang, Y. and Oliver, J. H. (1994), "Non-constant Parameter NC tool path generation of Sculptured surfaces", International Journal of Advanced Manufacturing Technology, 9, pp. 281-290.

[9] Suresh, K. and Yang, D. C. H. (1994)

"Constant scallop-height machining of free-form surfaces", Journal of Engineering for Industry, 116(2), pp. 253-259.
[10] Wang, F. and Wright, P. K. (1998), "Open architecture controllers for machine tools: a real time quintic spline interpolator", Journal of Manufacturing Science and Engineering, 120(2), pp. 425-432.

최 용 훈 (Yong-hoon, Choi)

- 울산대 기계공학과 학사(97년)
- Iowa State University 산업공학 석사(99년)
- Iowa State University 산업공학 박사(02년)
- UC-Davis기계공학과 Intelligent Manufacturing Systems Mechatronics Lab 연구원 (03년)
  현대자동차 현대-기아생산개발총괄본부
  생산기술개발팀 과장 (03 - 04년)
- 현재 한국전자통신연구원 선임연구원
- 관심분야 : 생산공정설계/모니터링, 정밀가공, CAD/CAM, 신호처리

김 상 진 (Sang-jin, Kim)

- 한양대 전기공학과 학사(73년)
- 한양대 전기공학과 석사(75년)
- Kazakh National Academy of Sciences 전기공학과 박사(97년)
- 현재 시립인천전문대학 컴퓨터제어과 교수(81년~)
- 관심분야 : 산업제어공학, 센서공학