Adaptive Tool Path Planning Applied in Manufacturing Optimization

J. Radej, L. Budin and Z. Mihajlovic
University of Zagreb
Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Croatia
Phone: (385) 1 612 99 35, Fax: (385) 1 612 96 53, E-mail: josko.radej@fer.hr

Abstract - **Paper describes research done on algorithms for the generation of gouge-free nonisoparametric tool paths across surface plane segments obtained through triangulation partitioning. Along with contiguous trajectory sequences, adaptive tool path planning introduces partially discontinuous tool trajectories. Algorithm achieves the necessary trajectory continuity by the insertion of the auxiliary tool-orientation trajectories. The tool passes over the path gaps are accomplished through rapid tool traverses. The evaluation of surface machining optimization is provided by simulation of computed tool motion trajectories applied to triangulation models. Tool tracks and tool trajectories obtained with different surface models are used for presentation and examination of the optimized tool path modeling and algorithm computing performance issues.**

I. Adaptive Tool Path Planning

The adaptive tool path planning is selected as a convenient general method in approaching the tool-path optimization. Method is based on an adaptive spatial partitioning technique, which incorporates proximity estimation between the adjacent tool trajectories. Elber and Cohen [1] applied this method to surface models with parametric representation. Tool path generation was carried out through an adaptive extraction of isoparametric curves in the free-form surface parametric domain. The extracted isocurves were applied as valid tool trajectories and declared as boundaries of the newly created surface regions. In order to optimize the coverage of the region, successive tool path generation was carried out by interpolating the regions with adaptively extracted isocurves or partial subisocurves. The recursive process of adaptive-isocurve extraction and the successive partitioning of the region were accomplished when the full coverage of the complete surface was achieved. Thus, the path generating control was performed through an extensive trajectory proximity examination in the Euclidean surface domain. The employment of partial tool trajectories, which appropriately follow the computation of subisocurves, represents a specific, almost new approach in surface machining practice. The method ensures complete surface coverage and largely eliminates tool trajectory redundancies.

In any case, the application of the tool path planning method, which introduces piecewise broken tool trajectories and subsequently intermittent machine tool cutting motions, requires additional technological justification. Any manufacturing process optimization should somehow contribute to the improvement of manufacturing effectiveness. The improvement is usually the result of the corresponding reduction in the total machining time. Concerning the free-form surface machining, where the applied feed-motion rates are restricted by the machining technology reasons, the increase of machining effectiveness can be achieved only through an effective reduction of the tool path length.

The part program, which contains the commands for controlling the machining of the entire surface, should be equipped with some additional traversing commands for overstepping the gaps between successive cutting trajectories. The traversing motions are usually accomplished with the highest machine speed at the appropriate safety distance. Thus, the traversing speeds are considerably higher than the cutting tool-motion speeds. Supposing that all traverse motions contribute very little to the total machining time, the applied tool path optimization finds a reasonable justification.

The employment of the adaptive tool path planning method for path generation over the triangulated surfaces requires, in the first place, the preparation of an algorithm for effective surface partitioning and the establishment of convenient surface regions [2]. In addition, the region triangle structures should conform to computing requirements for the continual filling of region interiors through adaptive tool path generation. Accordingly, the adaptive region filling with complete and partial tool trajectories must be provided by an optimized coverage of each of the region triangles.

II. Adaptive Path Filling

Algorithm for adaptive path filling refers to the adaptive tool path generation required for optimized machining of the region interiors. It is subsequently invoked upon the completion of the region boundaries. Inside the given region, the algorithm provides an adaptive path filling by generating complete or partial tool paths. Thus, path generation depends on variations of the real bandwidth along the region. Each region is established as a one-fold string of triangles. Triangles are mutually concatenated by common side-edges in a closed annular structure. Accordingly, region boundaries are built of triangle vertices and triangle base-edges, allowing the triangles to be of different size and form. With respect to the adaptive path filling, mapping of the path segments and control of the triangle coverage is obtained by generating straight-line path segments across the triangle and parallel to the triangle base. Therefore, every straight line cuts the triangle height and side edges in equal proportions. This characteristic triangle property makes possible the recursive interpolation of succeeding tool tracks between the boundaries of the region by a direct computation of trajectory spacing in every triangle. Generalized procedural steps applied in programming of the optimized tool track generation along the sequence of the region triangles are presented in *the adaptive tool path-generating algorithm* (Appendix).

The algorithm is accommodated for the employment of the ball-end tools. It requires the specification of the cutter radius (R) and parameter (δ), the greatest admissible path spacing obtained from the given scallop height (h) [1,3]. The construction of tool trajectories, which employ the cutter location (CL) points and enable the final specification of machine tool motion statements, requires the subsequent calculation of tool orientation. In addition, an exact determination of the current trajectory-segment end-point requires an anticipative extraction of geometry parameters from the subsequent in-region triangle. Besides, the computation of additional tool motions required for successive change of tool orientations depends on local relations between the triangle planes and differs for crossing either concave or convex triangle edges.

Surface protection against gouges [4,5], which are likely to occur at the concave areas, is accomplished by employment of *the tool interference detection and avoidance algorithm*. The algorithm provides surface protection using the method of computing *the safe CL points*, rather then providing the complete trajectory safety [6]. Every computed CL point, which completes an effective tool trajectory, is successively checked by the three-step procedure on tool interference with surface vertices, triangle edges and triangle segment planes [7]. The detection of possible interference through any of the distinct steps requires the tool location modification by virtually retracting the tool into a safe position. The successive multi-step retraction accomplishes the final calculation of the appropriate safe CL point, which can than be applied for generating the safe tool trajectory.

According to this, the implementation of the complete algorithm for surface manufacture optimization enables the successive command generation for optimized machine tool motions. Thus, all surface parameters, that the algorithm requires for generating an optimized tool path, are obtained from the tabulated function description of the relevant triangulation, which already approximates the surface within the range of machining tolerance (±ε). The user should provide the parameter which determines the required surface quality, i. e. the critical path spacing (δ) respectively. The only parameter, which is left for consideration, is the tool size. Even if the surface is protected against gouges by the interference-avoiding algorithm, the resultant surface will not meet the required precision if the tool size is overestimated. Presuming that the triangles obtained through triangulation optimization are by size and form optimally adapted to the desired approximation form, the minimal triangle size can be considered as a guiding criterion for selecting the appropriate tool size.

III. Case Study

In order to examine path-generation performances, algorithm is evaluated with respect to the tool path planning and accuracy of the tool trajectory geometry.

Referring to the case described in [2] where an initial triangulation of four triangles was appropriately reconstructed into ten triangles, allowing the formation of region boundaries, processing of phase two of the algorithm provides the region with adaptive path filling. By generating contiguous sequences of tool tracks across the region triangles, the number of tool passes across the region interior is appropriately adapted to the parameter δ within every triangle (Fig.1). By minimizing the tool track redundancy, the algorithm ensures the full coverage of all triangle interiors. Under certain conditions that depend on the applied parameterization and corresponding triangle geometry, some limited areas along the triangle side edges can be left partially uncovered. The algorithm resolves this insufficient covering of triangle side edges by inserting additional *short tool tracks*. In addition, the algorithm provides the calculation and insertion of possible *short tracks* after the completion of all regular passes over any of the region triangles and before any of triangles is declared as fully covered (Appendix).

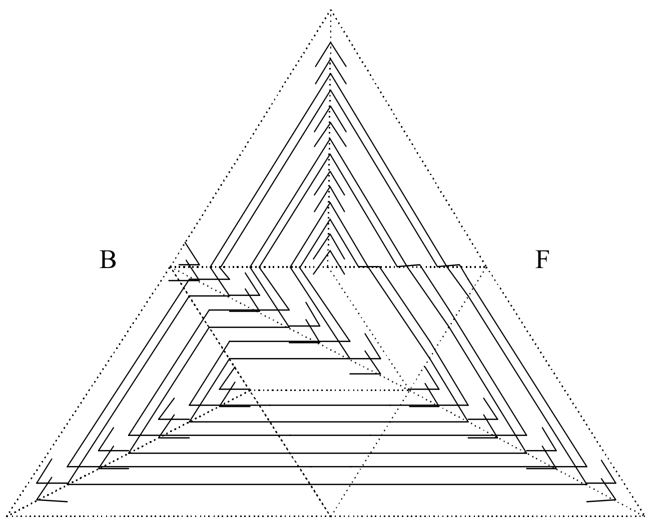

Figure 1. Adaptive path filling by calculating tool tracks across region triangles. Resulting tool trajectories are accommodated to triangle plane orientations.

The calculation of tool tracks represents only the necessary procedural step required for the calculation of tool trajectories. Geometrical parameters needed for generating machine tool motion commands (i.e. tool trajectory segments) are accomplished through the calculation of CL points and the specification of the appropriate motion function. To calculate CL points, the algorithm employs the procedures of computational geometry applied to data retrieved from the local surface segments (triangles), constructed tool tracks and actual tool parameters. The algorithm obtains tool orientation change by generating a spatial circular arc motion around the convex type side-edges or linear edge parallel motion by approaching the contiguous track of the next triangle over the concave type edges. Thereupon, fig. 1 shows tool trajectories obtained with the tool-radius distance from the corresponding triangle planes. Four triangles made up the basic configuration of the observed surface model, where one of them lies in a plane relatively inclined to the common plane orientation of all other triangles. This inclination introduces a concave type edge (BF) within the model and consequently causes the corresponding adaptation of the tool trajectory at the edge crossing. Particularly, the crossing of this type of triangle edge causes the insertion of a linear path segment. In this way, the length of the path segment strongly depends on the combination of three parameters: plane-tilting angle, tool track approaching direction (i.e. triangle shape) and ball-end tool radius. Their effects can be sometimes mutually compensated, as it can be seen by comparing two different tool-crossing courses over the BF triangle side edge.

Although tool tracks represent only an intermediate result, their introduction into graphical presentation contributes to the overall clarity of trajectories obtained within the complex model forms. Tool tracks pass precisely over the model surface and break up exclusively at triangle side edges. In case of any possible change of the view direction, they maintain their mutual geometrical relations to the elements of the 3D surface segments in the 2D display. In this way, the graphical presentation of tool tracks enables easier identification of corresponding tool trajectories and their appropriate discrimination during trajectory visual analysis. Thus, every corresponding tool trajectory appears as a sequence of concatenated line segments spatially distant from the surface plane. The composition of path segments reflects the edge crossing particularities. The calculation of CL points and the construction of tool trajectories directly contribute to the increase in machining accuracy and prevent local interference's inside the region domain. An early prevention of all local in-region interference's accelerates computations required for tool interference detection and its subsequent avoidance.

The concave side of a hemisphere surface, realized through rough polyhedral approximation and appropriately partitioned into several regions [2], was taken as the next illustrative model for tool-path analysis. Fig. 2 shows the resulting tool trajectories obtained over the interior of surface regions. Every tool trajectory is accompanied by the corresponding tool track that traverses the triangle plane segments all along the established surface region. The tool trajectory is constructed of concatenated lines, which connect the corresponding CL points. By that, CL points are preliminary evaluated as the locally safe points by examining a local geometry and eliminating in-region tool interference. To ensure the global safety of trajectory, every CL point is already checked against the global interference. The specific construction of

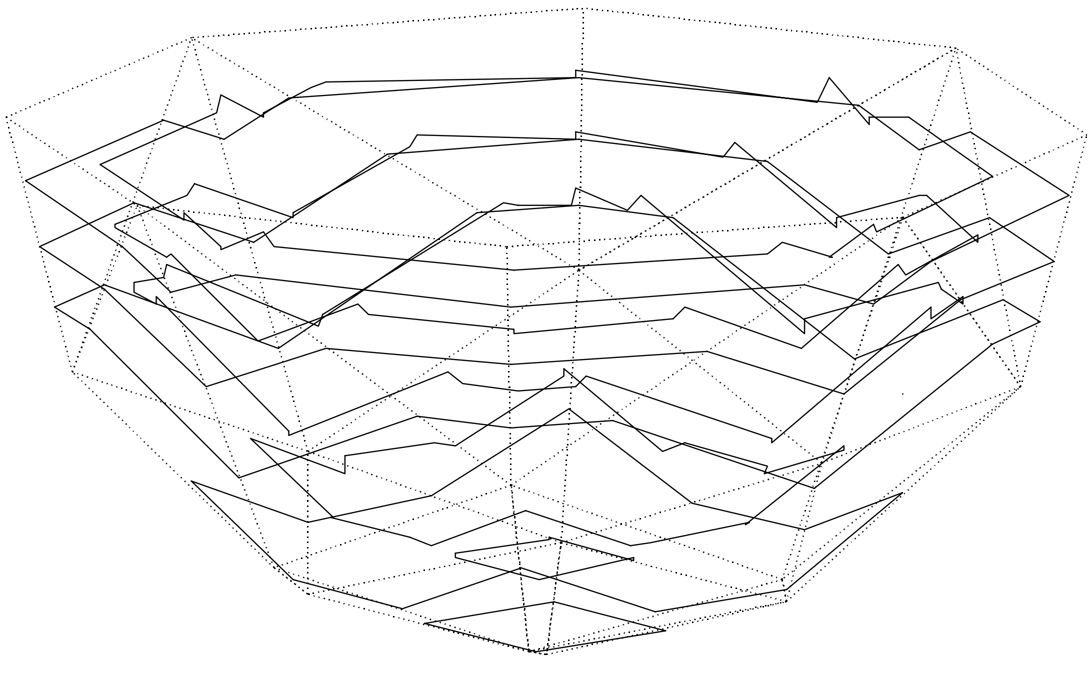

Figure 2: Concave surface of the hemisphere triangulation with tool tracks and tool trajectories obtained for filling the region partitions (R=20, δ=18)

trajectories and effects of the local, in-region geometry can be surveyed by comparing the adjacent trajectory segments and local surface forms. The effects of the trajectory corrections caused by the global tool interference avoidance are rather difficult to perceive. Inside the concave surface area, significant corrections emerge as an outcome of increasing tool size. In such circumstances, the corrected trajectory lifts up the tool sufficiently high above the cavity area.

Regardless of the built in security feature for preserving the surface safety, the correct employment of the algorithm and best performance issues relevant to the tool path optimization can be obtained only by a proper selection of the tool size. Thus, the selection of the tool size should be referred to the engineer's practical experience and observation of critical surface areas, i.e. observation of the critical triangle sizes throughout the triangulation. Also, there exist some more objective methods based on measuring tool path corrections during machining simulation [8]. Methods employ the systematic tool interference examination and computation of cutter location corrections. Tool parameterization is derived from the acceptability and/or tolerability of the obtained tool-path corrections.

IV. CONCLUSION

Algorithm performs a systematic coverage of all of the region triangles. Thus, the systematic coverage understands the full coverage of triangle interiors, as well as the coverage of the triangle side-edge areas.

The major objective of the algorithm relates to tool-path optimization. It is introduced by the adaptive region surface coverage and carried out by a recursive tool-path-computing procedures. The adaptive tool-path generation simultaneously fulfills the requirements for complete surface coverage and largely eliminates the tool-path redundancy at covering the surface regions. Thus, the beneficial effects of tool-path optimization provided by the algorithm significantly depend upon the diversity of machining claims. Presuming that the claim of machining precision is already implied by the given triangulation through an adequate depth of surface approximation, a call for improved surface quality or increased tool-track density will contribute to the entire gain of the algorithm. The algorithm achieves the best effects in tool-path planning optimization through an optimized selection of the tool-size parameter.

REFERENCES

[1] G. Elber and E. Cohen, "Toolpath generation for freeform surface models" *Computer-Aided Design* Vol. 26 No. 6 (1994) pp. 490-496

[2] J. Radej, L. Budin and A. Milanovic, "Triangulation Partitioning Applied in Tool Path Optimization" *Proc. of Melecon'2004 Conference* Dubrovnik, Croatia, May 12-15 (2004)

[3] B. K. Choi, C. S. Lee, J. S. Hwang and C. S. Jun, "Compound surface modeling and machining" *Computer-Aided Design* Vol. 20 No. 3 (1988) pp.127-136

[4] B. K. Choi and C. S. Jun, "Ball-end cutter interference avoidance in NC machining of sculptured surfaces" *Computer-Aided Design* Vol. 21 No. 6 (1989) pp.371-378

[5] J. S. Hwang, "Interference-free tool-path generation in the NC machining of parametric compound surfaces" *Computer-Aided Design* Vol. 24 No. 12 (1992) pp. 667-676

[6] R. L. Drysdale, I I I, J. L. Quin, K. Ozair, R. B. Jerard, "Discrete Surface Representations for Simulation, Verification and Generation of Numerical Control Programs" *Proc. 1991 NSF Design & Manufac. System Conf.* Austin, USA, Jan 9-11, (1991)

[7] S. X. Li and R. B. Jerard, "5-axis machining of sculptured surfaces with a flat-end cutter" *Computer-Aided Design* Vol. 26 No. 3 (1994) pp. 165-178

[8] J. Radej and V. Sruk, "Validating the Tool Trajectories at Machining of Free-form Surfaces" *Proc. Computers in Technical and Intelligent Systems (MIPRO'02)* Opatija, Croatia, May 20-24, 2002 pp. 55-59

APPENDIX

Definition of parameters used for calculation of tool tracks:

$\delta = \delta(h,R)$	greatest allowed gap between adjacent tracks;
h	greatest allowed scallop height;
R	ball-end tool radius;
Δ	actual distance between tracks in triangle;
α, φ	angles between triangle height and side-edges;

Algorithm 2: **Determination of tool tracks across the region of triangles**

<u>Given:</u> δ, T_n ;

set i := 1; (sequence of tool passage round the region)

do while all region triangles are fully covered
  m_1: find segmentation coefficient <u>k</u> (based on sequence of tool passage);
    determine the starting point (P_m, T_n);

  m_2: **if** triangle coverage is completed
    then continue at m_3;
    else define Δ;
      if $(\Delta > \delta)$
        then perform tool pass across triangle (*feed motion*);
          continue at m_3;
        else if $(2\Delta > \delta\,(1+\cos\alpha))$
          then perform commencing short pass (*feed motion*);
            if $(2\Delta > \delta\,(1+\cos\varphi))$
              then perform finishing short pass;
                continue at m_3;
              else continue at m_3;
          else if $(2\Delta > \delta\,(1+\cos\varphi))$
            then perform finishing short pass (*feed motion*);
              continue at m_3;
            else mark triangle being finished;
              continue at m_3;

  m_3: cross over to the next triangle;

    if this is commencing triangle
      then i := i+1; (increment sequence of tool passage)
        continue at m_1;

      else continue at m_2;