Defence Science Journal, Vol. 56, No. 2, April 2006, pp. 117-127

Algorithm for Pocket Milling using Zig-zag Tool Path

P. Selvaraj
Aeronautical Development Agency, Bangalore–560 017

and

P. Radhakrishnan
Vellore Institute of Technology, Vellore–632 014

ABSTRACT

Pocket-milling operations are widely used for scooping out materials during the machining of aircraft components. This paper presents a tool-path planning algorithm for pocket-milling using zig-zag method. The algorithm consists of basically three modules, viz., generating tool-path elements using pocket geometry entities as input, finding out intersection points (edge points), and rearranging points in a zig-zag fashion. OPTPATH algorithm[1,2] is used for generating tool-path elements. These elements thus generated are used to find out the intersection points with all entities. The valid points are arranged in a zig-zag way, which are used for machining any pocket considered. This algorithm works satisfactorily for all the pocket boundaries having line-line, line-arc, and arc-arc geometry entities.

Keywords: Pocket milling, algorithm, tool-path elements, pocketing tool path, zig-zag tool path, tool-path planning, OPTPATH algorithm, milling pocket machining

1. INTRODUCTION

Milling is, one of the most widely used metal removal processes. Pocket milling clears an area bounded by a set of specified entities such as lines, arcs, and free-form curves, which constitute outer periphery with or without islands. The machining sequence may be either in the order of entities selected or in reverse order. Types of pockets include rectangular, circular, and inclined. A pocket-milling operation consists of the following:

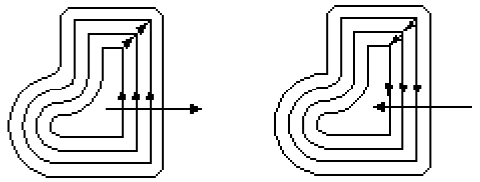
- Rough and finish operations
- Multiple islands capability
- Multiple rough and finish cuts
- Plunge, ramp or helical entry.

1.1 Types of Pocket Machining

Based on the contour shapes and machining methods[3], pocketing tool paths are classified into spiraling and zig-zag types.

To generate spiral tool path, the boundary profile are strinked inwards while the island profiles outwards using the appropriate steps. In spiral-out option, the tool paths track from the centre of pocket to the outer boundary of the pocket [Fig. 1(a)], whereas

Received 06 November 2005

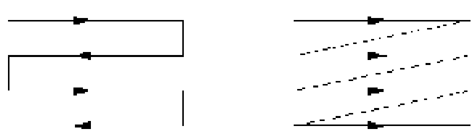


(a) (b)

Figure 1. Tool path: (a) spiral-out and (b) spiral-in

in spiral-in option, the tool-path track from the outer boundary of the pocket towards the centre inwardly, as in Fig. 1(b).

In the zig-zag method of pocketing, the tool paths generated are parallel to a predefined vector direction and the tool moves back and forth. This method is used when a machine tool has a preferred direction of cut, like along the major axis of machine or the grain direction of the material calls for a particular direction of machining. In successive zig-zag method, machining takes place bidirectionally parallel to a selected axis. In bidirectional milling, the cutting edge changes alternatively left and right sided, ie, up milling and climb milling as in Fig. 2(a). In retract-rapid-engage/unidirectional machining, the tool always cuts the material in one way, along or against the spindle direction in the



(a) (b)

Figure 2. In bidirectional milling, the cutting edge changes alternatively left and right sides: (a) zig-zag and (b) retract-rapid-engage.

entire process as in Fig. 2 (b). The offset chains of pocket entities are intersected with a sequence of equidistant parallel lines/curves/arcs, which are oriented along the selected direction of cut.

1.2 Pocket-milling Module

Automatic generation of NC codes, specially the NC tool path for pocket milling, is essential for achieving total manufacturing automation in the industries like aircraft, die and tool making[4]. Considerable amount of work has already been done in addressing various aspects of this pocket-milling operations. Many of the existing CAM packages automatically generate NC codes when geometry of part, tool, and machining parameters are given. The requirements that should be fulfilled by a pocketing software include the following:

The process shall guarantee the coverage of entire area of polygonal 2.5D surface without leaving any uncut projections.

The process shall ensure not overcutting the area of polygonal surface.

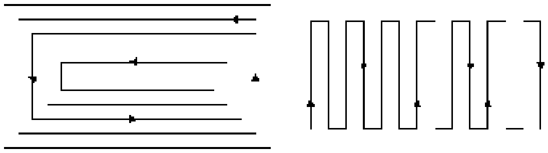
2. PROBLEM DESCRIPTION

Milling process assumes a unique position in the current manufacturing practice due to a variety of jobs that can be produced through this machining method. CNC milling produces about 80 per cent of all machined parts. Manufacturing of critical components such as tools, dies, and surface plates involve milling as a main process in the production cycle.

Although there are many possible ways of planning a tool path in pocket-milling operation, traditionally contour augmentation (spiral) and zig-zag (or staircase) milling, have been the two standard procedures practiced. Zig-zag or staircase milling involves the movement of the tool in a number of parallel passes to cover an entire area of the polygon[5,6] to be mach

In other words, milling takes place along line segments to a specified reference line in alternate opposite directions between adjacent paths. On the other hand, contour augmentation milling involves movement of tool parallel to the contour of pocket boundary in a spiral like form, starting from point on the periphery of the polygon[7].

Each one of the above methods has its own advantages and disadvantages. Zig-zag (staircase) milling requires more number of stops and turns, requiring more machining time. Contour augmentation (spiral) milling method requires a relatively larger tool overlap between successive passes to avoid the undercut projections on the surface of the polygon. This results in an increased length of the tool path,



(a) (b)

Figure 3. Pocketing: (a) contour augmentation and (b) zig-zag.

and consequently, machining time. Figure 3 (a) illustrates pocketing (contour augmentation) method and Fig. 3(b) explains that of zig-zag milling method.

2.1 Pocket-milling (Zig-zag) Requirements

Pocket milling (zig-zag) module has the following sub-modules[8]:

- Inputing valid geometry data[9] (pocket boundary)
- Tool-path (zig-zag) generation
- Tool-path optimisations[10-11]
- Tool-movement simulation
- Cutter location data generation

Generally accepted user requirements of a zig-zag machining include efficient machining using minimum machining time, fine surface quality without tool marks, and no gouge against boundary curves.

Considering the above functional requirements, tool-path generation algorithm developed in this study includes the following:

(i) Minimisation of tool retractions

(ii) Minimisation of tool-path elements

(iii) Maximisation of average tool-path length

(iv) Technological requirements

(v) Motion along boundary curve

(vi) Interactive computation time.

- *Minimisation of tool retractions*: Tool retractions[12] cause non-cutting tool motions in the air and tool marks on the machined surface. These types of motions have to be minimised.
- *Minimisation of tool-path elements*: At the end of tool path elements, the feed rate should be slowed to avoid machining error caused by rapid change of feed direction. Minimising the number of tool-path elements improves both productivity and quality of machined parts.
- *Maximisation of average length of tool-path elements*: Tool-path elements having longer length allow constant feed rate and direction, which in turn improves surface quality.
- *Technological requirements*: The tool-path planning system should be able to adapt to the various technological requirements or constraints such as one way milling or zig-zag milling and up or down milling.
- *Motion along boundary curves*: Linear tool motion between tool-path elements may cause gouging at the sharp vertices of boundary curves[13] and which has to be checked.
- *Interactive computation time*: Tool-path planning should be efficient enough to support interactive computation time with real problems.

2.1.1 *Robustness of the Algorithm*

The algorithms should be designed to be insensitive to numerical errors, ie, system should be able to handle unpredictable errors caused by numerical computations.

2.1.2 *Process Parameters*

The module should handle both single and multi-level milling. The various process parameters to be considered are: (i) final depth (ii) step-over value, and (iii) direction of cut (parallel to X-axis or Y-axis).

2.1.3 *Cutter Location Data Generation*

The algorithm developed generates cutter location (CL) data based on zig-zag tool path for single and multiple depths. This CL file has to be post-processed to generate NC code for a given CNC control system.

The present study does not include any island. Pocket boundary entities, having line-line, arc-arc, and line-arc combinations, have been considered.

3. DEVELOPMENT OF ALGORITHM

The parameters such as spindle speed, feed rate, depth of cut, number of teeth on milling cutter, etc affect the machining time[14]. These parameters are fixed in a pocket milling and the machining time is proportional to the tool traverse time (T).

3.1 Determining Effective Tool-path Length

Tool traverse time (T) can be defined as

$$T = Lc/Vc + m*t + Lr/\mathrm{Vr}$$

where

Lc = Cutting tool-path length

Vc = Cutting feed rate

m = Total number of turns along tool path

t = Time required for the tool to make one turn along its path

Lr = Rapid traverse tool-path length and

Vr = Rapid traverse feed rate.

Tool traverse time (T) can be expressed as

$$T = L/Vc + m*t$$

where L is the effective tool-path length defined as $L = Lc + \gamma*Lr$, $\gamma = Vc/Vr$.

The value of $m*t$ in the above equation can be treated as negligible when compared with L/Vc. In such a case, T becomes a linear function of L and optimisation of L is equivalent to optimisation of T. Hence, the development of algorithm for generating effective tool-path length, L has been discussed.

3.2 Methodology of Pocket Milling (Zig-zag)

Pocket milling (zig-zag) consists of the following major steps:

(i) *Getting valid geometry input*: The pocket boundary geometry data is obtained by digitising in sequence of a CAD model entities, and a check is made if the boundary forms a closed profile, and all the selected entities are in sequence[15,16].

If yes, store the entities data in a file (input_geometry). If line, store start and end points of line. If arc, store the centre point, start angle, end angle, and radius of the arc.

The system displays an error message if the profile is not a closed one or if the entities selected are not in sequence.

(ii) *Identifying input parameters*: Input parameters for this software module include: pocket geometry entities[15] (lines and arcs), diameter of tool, direction of tool-path element, step-over value, stock allowance, work plane, and tolerance. Work plane indicates the z-value on which milling takes place.

In multi-level pocketing, both the depth of pocket and the incremental depth have to be specified as an additional input data.

(iii) *Offset of the profile (inner) by the sum of radius of tool and stock allowance*: Consider this new offset profile as input for further step-over calculations, as zig-zag pocketing cut will be made on this profile.

(iv) *Store the geometric data related to lines and arc*: Store the geometric data related to these entities (lines and arcs) in the offset profile in 'line.txt' and 'arc.txt' files for line and arc, respectively. If the entity is a line, start and end points of the line are stored. Similarly, for an arc, centre point, start angle, end angle, and radius of the arc are calculated and stored[17]. Mid points of the arc are also computed and stored.

(v) *Determination of maximum and minimum X/Y of profile*: It consists of the following steps:

- If tool path is parallel to X-axis, the tool-path element line equation is given by

$$Y = Y_{min} + (k - ½) * d \text{ until } Y < Y_{max}$$

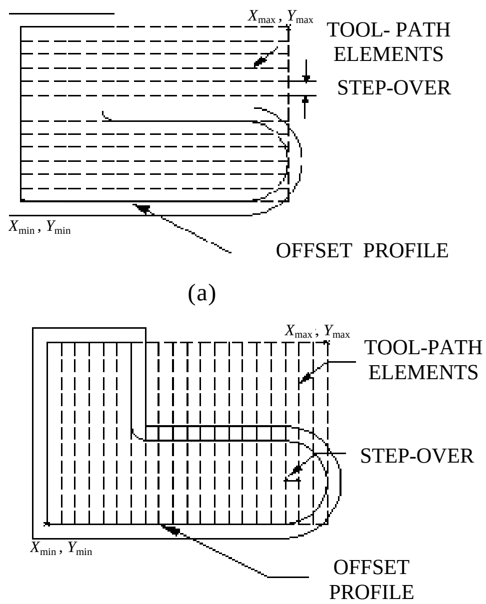


Figure 4. Zig-zag tool path: (a) parallel to X-axis and (b) parallel to Y-axis.

where k is pass number, d is the diameter of tool, and $Y = Y_{max} - d/2$ for the last pass.

- If tool path is parallel to Y-axis, the tool-path element line equation is given by $X = X_{min} + (k - ½) * d$, until $X < X_{max}$ and $X = X_{max} - d/2$ for the last pass.

These equations have been derived by Kamarathi, Bukkapatnam and Hsieh[1,2].

Figures 4 (a) and 4 (b) illustrate findings of maximum and minimum limits of x and y.

(vi) *Calculate maximum and minimum values of x and y for the profile*: Maximum and minimum values of x and y for the profile can be calculated by opening 'line.txt' and 'arc.txt' files after considering all geometrical data entities.

(vii) *Decide the pocketing algorithm*: Pocketing algorithm can be decided, such that minimum 'tool retract-rapid-engage' operations are involved[18]. Figure 5 illustrates the concept of segmenting the pockets into regions of milling to ensure minimum tool retractions.

(viii) *Generation of tool-path elements*: The direction for tool-path element is input by the user. If the direction of milling is parallel to X-axis, then for minimum and maximum of y value, tool-path elements are generated using the equations given in Section 3.1 and the same are stored in 'toolpath.txt'.

(ix) *Calculation of intersection point with each entity*: The intersection point with every entity is calculated for all the tool path line equations generated in the last step. These details are brought out in Section 3.3.

(x) *Sorting of the intersection points*: After getting the intersection points, their sorting is the next step to get the sequenced tool paths. This is described in Section 3.4.

3.3 Algorithm for Computation of Intersection Points

The intersection point with each entity is calculated for the tool-path elements generated in the last module. The algorithm will not compute intersection point if the entities is 'line' and parallel to each other. On the other hand, intersection points computed between line entities and tool-path element are first checked for whether these interaction points

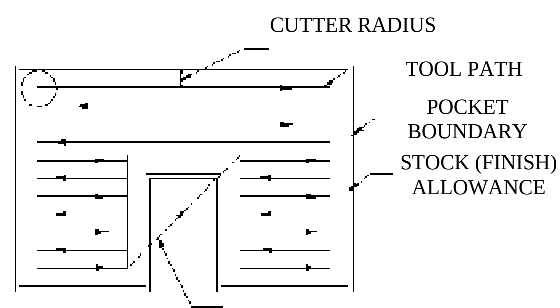


Figure 5. Minimum tool retractions

are in between the start and the end points of respective tool-path element, and also in between the start and the end points of the respective line entity, and those points are stored in 'intersec.txt'. If the entity is an arc, intersection points are calculated and checked whether these intersection points lie between start and end points of the arc. These points are saved in 'intersec.txt'.

3.4 Algorithm for Sorting Points in Zig-zag Path

The points which are repeating in ‘intersec.txt’ are removed and once again copied into ‘inter.txt’. If the direction is parallel to X-axis, the points are sorted first wrt *y* value in ascending order and then wrt *x* value in ascending order with the same *y* value. If the direction is parallel to Y-axis, then the points are sorted *vice versa*, followed by sorting of intersection points. If tool path is parallel to X-axis, the minimum value of *x* corresponding to the first pass *y* value is considered from a set of intersection points and is assigned the first point in the tool path. The next higher *x* value with the same *y* value is assigned for the second point. The value of *x* nearest to the last *x* value with the following *y* value is taken as third point. The fourth point contains the next minimum of *x* value with the same *y* value. This loop is executed likewise until the last pass is reached.

The above procedure is repeated for the left-over points until all the points have been considered. This will generate the required zig-zag tool paths[19,20] and these points are used for generating cutter location (CL) data file[21,22].

The sequence of operations for sorting of points in a zig-zag fashion is as follows:

(i) Open the file ‘intersec.txt’

(ii) Check whether any pair of points are repeated. If yes, delete one of the points and store the new set of points in the file ‘inter.txt’.

(iii) Open the file ‘inter.txt’

(iv) If the direction of motion is parallel to the X-axis, sort the points with respect to Y-axis

(v) Sort the points having the same *y* value wrt X-axis

(vi) Fix the point having minimum *x* and *y* values as the first point from the sorted point

(vii) Fix the point having the same *y* value and next higher *x* value as the next point

(viii) Point having next higher *y* value with the value of *x* being near to last *x* value is fixed as the next point

(ix) Point having next lower *x* value with the same *y* value is taken as the successive point

(x) Go to *Step* (vi) until maximum *y* value is reached.

(xi) Repeat the above process until all the points are fixed and the zig-zag points are stored in the file ‘copy.txt’.

The same procedure is repeated for *y* direction also, wherein the points are sorted wrt X-axis and Y-axis.

In case of multiple depths, the following computation is done to determine the actual number of steps:

Final depth = d_f

Incremental depth = d_{inc}

No. of steps, $n_{steps} = d_f / d_{inc}$

If n_{steps} is not an integer, take the next highest integer (n_{step_act} –> actual number of passes).

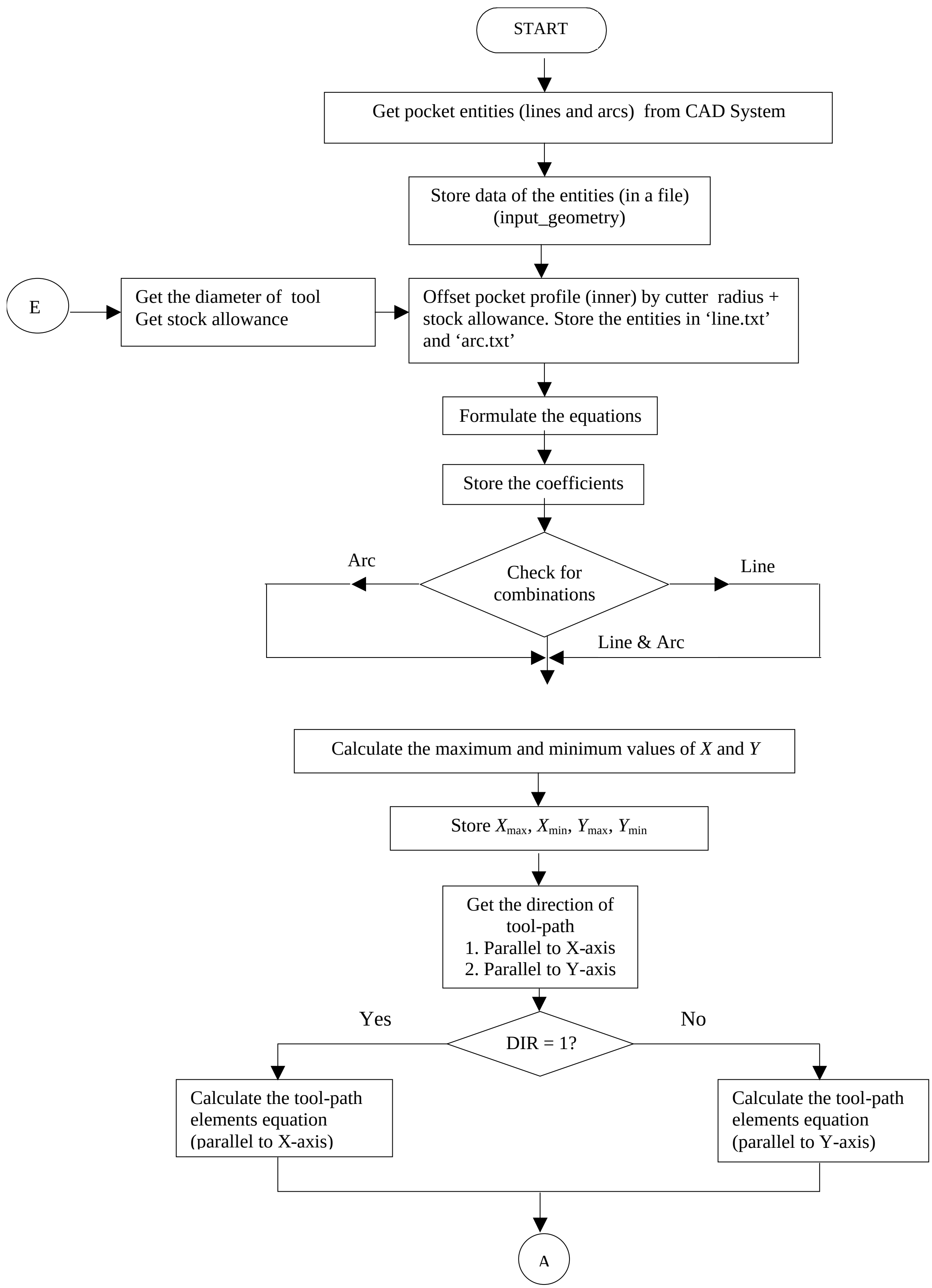
Actual depth of cut $d_{inc\_act} = d_f / n_{step\_act}$

For each pass, the tool plunges by d_{inc_act} incrementally. The CL data generated earlier are copied with a new set of Z coordinates corresponding to each depth of cut. Flow chart for pocket milling with zig-zag option is shown in Fig. 6.

The pockets are finish-machined by a profile-milling operation for which stock allowance has already been provided. The profile-milling algorithm is not covered in this paper.

4. TEST CASES

Typical pocket profiles having line-arc entity combinations have been tried for testing the methodology and algorithm. The test cases have been selected to solve typical situations. For example, Fig. 7 shows a pocket with curved boundaries. Another


START
Get pocket entities (lines and arcs) from CAD System
Store data of the entities (in a file) (input_geometry)
E
Get the diameter of tool
Get stock allowance
Offset pocket profile (inner) by cutter radius + stock allowance. Store the entities in 'line.txt' and 'arc.txt'
Formulate the equations
Store the coefficients
Arc
Check for combinations
Line
Line & Arc
Calculate the maximum and minimum values of X and Y
Store X_{max}, X_{min}, Y_{max}, Y_{min}
Get the direction of tool-path
1. Parallel to X-axis
2. Parallel to Y-axis
Yes
DIR = 1?
No
Calculate the tool-path elements equation (parallel to X-axis)
Calculate the tool-path elements equation (parallel to Y-axis)
A

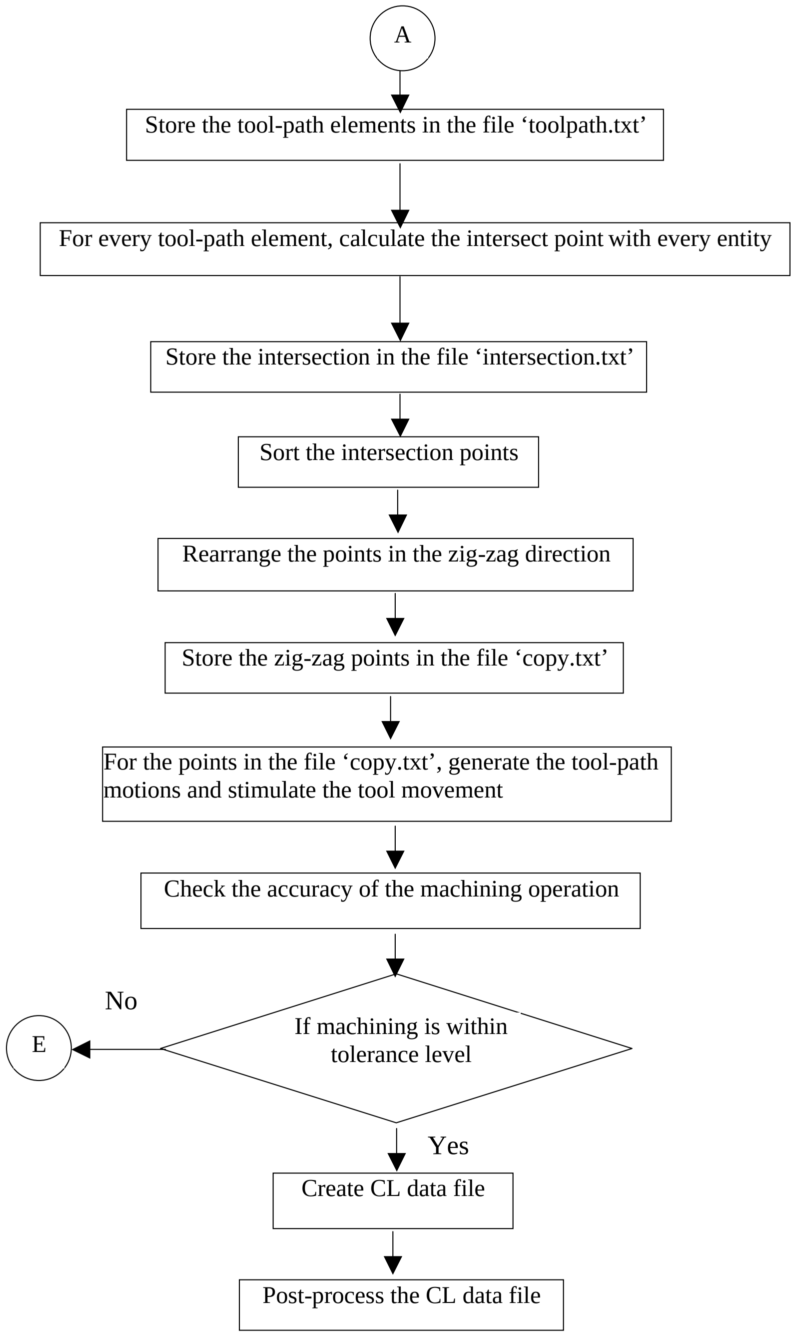


Figure 6. Flow chart for pocket milling (zig-zag)

complex segmented pocket with an additional convex curved profile is shown in Fig. 8. Figure 9 is a pocket with convex profile. However, many profiles of pockets can be considered to be the combinations of straight, convex, and concave profiles. Two such cases, tested using this software are shown in Figs 10 and 11.

5. CONCLUSION

An algorithm for the pocket milling (zig-zag) method has been developed using the recent techniques available and the coding has been done in 'C' language. The software has been tested for many complex profiles (both concave and convex shapes). The bugs and pitfalls in the program have been rectified

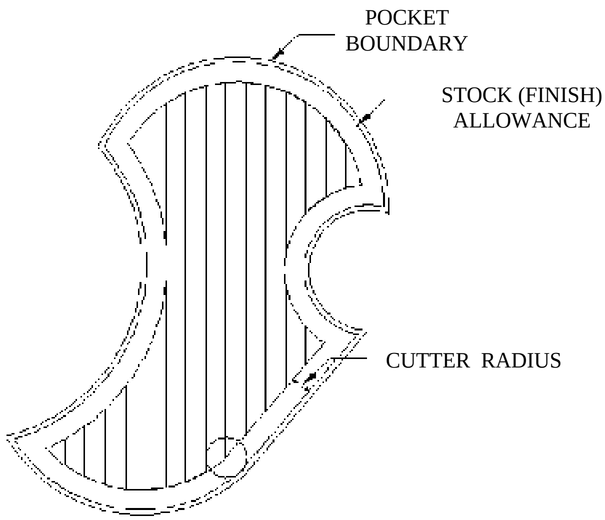


Figure 7. Pocket with curved boundaries

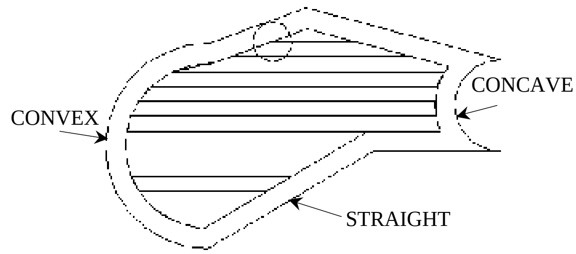


Figure 10. Pocket with straight, convex, and concave profiles

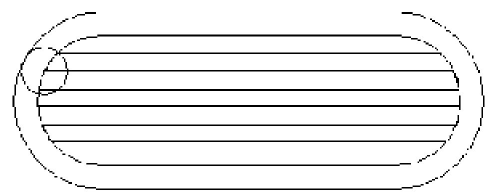


Figure 11. Pocket with convex profile

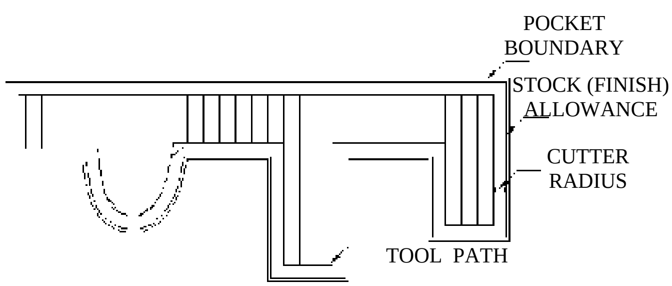


Figure 8. A segmented rectangular pocket with convex curve and rectangular slots.

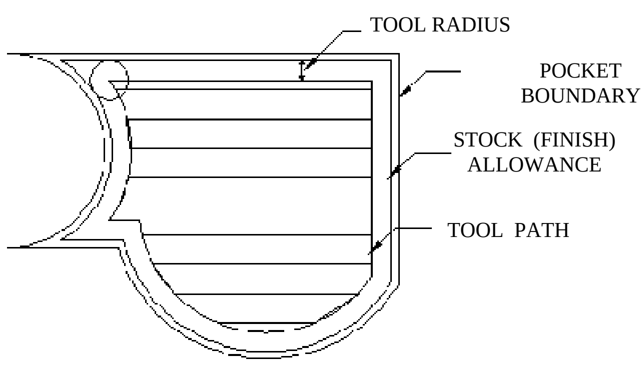


Figure 9. Combination of convex and concave profiles in a pocket.

and the program now supports any type of profile and generates the zig-zag tool path for the same with a minimum number of tool retractions and a minimum tool-path length.

ACKNOWLEDGEMENTS

The authors wish to acknowledge Dr T.G. Pai, Project Director (Technology Development) and Dr R.K. Ramanathan, Group Director (CAD and Computers) of Aeronautical Development Agency, Bangalore, for the encouragement to takeup this project jointly with PSG College of Technology, Coimbatore. The authors also wish to place it on record that the Management of PSG College of Technology, Coimbatore, came forward to work jointly on this project. The authors also wish to acknowledge Mr D. Shreeram, a postgraduate student in Computer Integrated Manufacturing for jointly developing this pocket machining (zig-zag) algorithm.

REFERENCES

1. Kamarathi, S.V.; Bukkaptnam, S.T.S. & Hsieh, S. An algorithm for a near-optimal NC path generation in staircase traversal of convex polygon surfaces. *Transactions ASME*, 2000, **122**, 182-90.

2. Kamarathi, S.V.; Pittner, S. & Bukkapatnam, S.T.S. Foundations for analytical models of staircase traversal of convex polygon surfaces. *Int. J. Prod. Res.*, 1997, **35**(7), 2009-024.

3. PRO/E manufacturing: User's guide. Parametric Technology Corporation, USA, 1996.

4. Radhakrishnan, P. & Subramanian, S. CAD/CAM/CIM. New Age International Ltd, Wiley Eastern Ltd, New Delhi, 1995.

5. Chang, Tien-chien; Wysk, Richard A. & Hsu-pinwang, Computer-aided manufacturing. Prentice-Hall Inc, New Jersey, 1991.

6. Production technology. HMT Ltd, Tata McGraw Hill Publishing Co, New Delhi, 1980.

7. Chih-Chih. A new approach to CNC tool-path generation. *Computer-aided Design*, **30**(8), 619-55.

8. Park, S.C. & Choi, B.K. Tool-path planning for direction parallel area milling.*Computer-aided Design*, 2000, **32**, 17-25.

9. Hinduja, S. & Chen, S.J. Geometry of 2D Components. *Computer-aided Design,* 1987, **19**, 323-28.

10. Prabhu, P.V.; Gramopadhye, A.K. & Wang, H.P. A general mathematical model for optimizing NC tool path for face milling of flat convex polygonal surfaces. *Int. J. Prod. Res.*, 1990, **28**(1), 101-30.

11. Deshmukh, A. & Wang, H.P. Tool-path planning for NC milling of convex polygonal faces: Minimisation of non-cutting area, *Int. J. Adv. Manufac. Technol.*, 1993, **8**(1), 17-24.

12. Kai Tang, Shuo; Chou, Yan & Chen, Lin-Lin. An algorithm for reducing tool retractions in zig-zag pocket machining. *Computer-aided Design*, 1998, **30**(2), 123-29.

13. Persson, H. NC machining of arbitrary shaped pockets. *Computer Aided Design*, 1978, **10**(3), 169-75.

14. Radhakrishnan, P. Computer numerical control machines. New Central Book Agency Ltd, Kolkota, 1998.

15. Preparata, F. & Shamos, M. Computational geometry: An introduction. Springer-New York, 1985.

16. Wozny, M.J.; McLaughlin, H.W. & Encarnacao, J.L. Geometric modelling for CAD applications. Prentice Hall Publishing Ltd, North Holland, 1986.

17. Groover, Mickel P. Automation production systems and computer integrated manufacturing. Prentice Hall of India Ltd, New Delhi, 1998.

18. Arkin, E.M.; Held, M. & Smith, C.L. Optimisation related to zig-zag pocket machining. http://www.ics.uci.edu/- eppstein/gina/cam.html

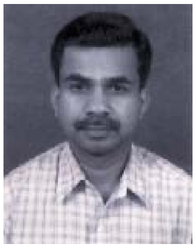
19. Foley, James D.; Dam, Andries Van; Fernier, Steven K. & Huges, John. Computer graphics. Ed. 2, Addision Wesley Publishing Co, Massachuttes, 1995.

20. Hearn, Donald & Baker, M. Pauline. Computer graphics. Prentice Hall of India, New Delhi, 1998.

21. Zeid, Ibraheim. CAD/CAM theory and practice. McGraw Hill Publishers, New Delhi, 1991.

22. Piegel, L. Curve fitting algorithm for rough cutting. *Computer-aided Des*ign, 1986, **18**(2), 79-82.

Contributors



Mr P. Selvaraj obtained in BE (Mech Engg) from the University of Madras, Chennai, in 1982 and ME from the PSG College of Technology, Coimbatore, in 1984. He joined Aeronautical Development Agency (ADA), Bangalore, as Project Engineer in 1987, and since then, he has been involved in the design and development of LCA-*Tejas* aircraft. Areas of his research include CAD/CAM, development of manufacturing application software, five-axis post-processor solutions for CNC machining, and sheet metal manufacturing automation. He has published a number of technical papers in the areas of CAD/CAM. At present he is working as Scientist/Engineer F in CAD & Computers Group of ADA, Bangalore.

Dr P. Radhakrishnan obtained his PhD (Mech Engg) from the University of Madras, Chennai, in 1976. He is presently the Vice-Chancellor of the Vellore Institute of Technology. Before this, he was the Principal of PSG College of Technology for three decades. He has carried out several consultancy projects in the areas of CAD/CAM, reverse engineering and technology development in rapid prototyping, etc.Organisations for which he carried out projects include ISRO, ADA, and several industries. He was associated with the GSLV project of ISRO and LCA project of ADA. He has successfully executed many funded and sponsored research projects through the Govt agencies like AICTE, ARDB, BARC, DRDO, DST, and MHRD. He is the author of four books and published more than 100 technical papers. He is a fellow of Institution of Engineers, a member of the Society of Manufacturing Engineers, and American Society for Engineering Education. He has been the Chairman of the Coimbatore Chapter of Computer Society of India and a member of Syndicate of Bharathiar University, Coimbatore. He is also a member of the Board for Industry-institute Interaction and a member of the UG Review Committee of AICTE.