Adaptive Tool Path Planning Strategy for Freeform Surface Machining using Point Cloud

Mandeep Dhanda[1] and S.S. Pande[2]

[1]Indian Institute of Technology Bombay, India, mandeepdhanda25@gmail.com
[2] Indian Institute of Technology Bombay, India, s.s.pande@iitb.ac.in

ABSTRACT

This paper reports the development of an efficient tool path planning strategy for machining of freeform surfaces directly from their representation in the form of point cloud without interim surface fitting. A grid based adaptive planar strategy has been designed and implemented for 3-axis CNC machining using ball end mill. Uniform CL grid points are initially computed by using tool inverse offset algorithm. Estimating the chordal and scallop errors likely to be produced, adaptive grid refinement is carried out iteratively till the errors converge below the prescribed tolerance limits. Adaptive grid (CL) points are subsequently processed to generate the final CNC part programs in the ISO format. The part programs were extensively tested for various case studies using the commercial CNC simulator as well on the actual CNC machine. Effects of variation of grid size and cloud point density were studied for grid optimality and their effect on the quality of CNC programs generated. The results were compared with those from the commercial software. Our system was found to generate more efficient tool paths in terms of enhanced productivity, part quality and reduced memory requirement.

Keywords: Point cloud, Adaptive tool path planning, 3-axis CNC machining.
DOI: https://doi.org/10.14733/cadaps.2019.289-307

1 INTRODUCTION

Today complex freeform surfaces are widely used on parts in various automotive, aerospace, die/mold and medical implant manufacturing industries to meet various functional and aesthetic requirements of the products [1]. Multi-axis (3-5 axis) CNC machines are widely used to manufacture the free form surface parts owing to their programmable flexibility and ability to machine diverse materials. Efficiency of CNC machining is dictated to a large extent by the availability of error free, efficient part programs. Researchers worldwide are therefore, attempting to develop various tool path planning strategies to efficiently machine free form surfaces on multi-axis CNC machines. Majority of these research efforts are focused on using CAD part models represented in parametric or mesh based (STL) formats [4],[6-8],[17],[19].

Of late, reverse engineering is being widely used to replicate an existing part and/or modify it to suit part functional and aesthetic requirements. It forms an important alternative route for product development. Literature reveals that major research efforts are focused on reconstructing the surface from scanned data points (point cloud) and using this CAD model subsequently for planning the CNC tool path [14],[22]. This procedure is however, computationally expensive as well as less accurate (error prone) [2]. A need thus, exists to develop efficient algorithms for CNC tool path planning for complex free form surfaces represented as point cloud without the need of surface fitting. Direct tool path generation from point cloud would significantly shorten the reverse engineering cycle of a component and in turn, reduce the costs by eliminating the scan processing costs [2]. The challenging task is to compute differential geometric properties of the surface model such as normal and curvature, which are readily available from the mathematical expression of a continuous surface, can now only, be estimated with unknown estimation error [5].This derives the motivation to develop fundamental algorithm for direct generation of efficient CNC part programme from unorganized point cloud which represents the CAD part models. In this paper, we present an approach that produces high productivity and quality NC tool paths directly from massive point data without laborious CAD model reconstruction.

The rest of the paper is organized as under. Section 2 presents the literature review for different attempts made in this direction and scope of work related to tool path generation directly from point cloud data. Section 3 presents the grid based adaptive algorithm for tool path planning and the associated functional modules in the system developed. Results from various case studies are presented in section 4. Summarized important conclusions from the work are in Section 5.

2 REVIEW OF LITERATURE

Literature reports three main tool path planning strategies (iso-parametric [17], iso-planar [8] and iso-scallop [19] for the machining of freeform surfaces. Each has its own advantages and limitations [13]. Iso-parametric strategy is computationally simple but produces varying scallop heights. Iso-planar is most widely used but generates redundant machining. Though iso-scallop generates constant scallop, it is computationally complex and time consuming [13]. The main drawback of all these strategies is that they are feature blind. Of late, attempts are being made to address this issue using surface properties like normal (iso-phote) angle and curvature variations [7],[13]. Han et al. [7] reported that the iso-phote based tool paths were shorter than the iso-planar but longer than the iso-parametric. Ding et al. [4] divided the parametric surface into two regions using iso-photes and observed that the global path planning strategy gives better surface finish at steeper regions but generates redundant tool paths. Using surface curvature, Giri et al. [6] observed that number of tool paths could be reduced significantly if they lied along convex directions. Tourier and Duc [21] observed that the tool paths are more spaced when curvature is concave than when it is convex.

All the above tool path planning strategies focus on the surfaces defined by explicit/implicit surface equations. Compared to the work on parametric or mesh based CAD models, very few attempts have been reported on directly using point cloud for tool path planning which circumvents the surface reconstruction issue and the associated problems. Feng et al. [5] proposed a CL-net cell approach to cluster points in each cell. CL points are established by taking a weighted average of the CL-net node positions to meet the machining error and maximum scallop height requirements. They, however, reported that bounding the machining error is not really guaranteed by this approach. Teng et al. [20] introduced a machining area segmentation approach based on the measure of geometric complexity of data points. Regions of high and low complexities are segregated and machined separately using the CL-net cell algorithm. Result shows that high machining efficiency is achieved with machining area segmentation. However, challenge is sharp corner of convex hull enclosing data points. Liu et al. [16] reported an approach for iso-scallop/iso-cusp tool path planning using one boundaries of the surface as the initial tool path. The methodology claimed to provide constant scallop height. However, the density of points governed the accuracy of tool paths generated. Chui et al. [3] partitioned the scanned raw 3D data into bands. Intersections between the spheres located at each data point were computed. By joining the points with circular arcs, the tool path is generated. No details have been provided on the accuracy of finish cut machining.

A new approach Inverse offset method (IOM) was proposed by Kishinami et al. [11], to generate cutter paths from 3D regular/irregular point data. It uses the concept of machining a single point in space using an arbitrarily shaped tool to generate self-interference free cutter paths on a forward and side step grid projected onto X-Y plane. Several authors [9],[10],[15] have reported generation of cutter paths using the IOM with constant grid sizes. Kayal [9] used IOM to adaptively refined grid based on the surface slope and curvatures to limit the surface error within tolerance. Though the machining efficiency significantly improved, accuracy of the machined surface was dependent on the estimation of curvature value and its use for adaptive grid generation. If the entire surface is machined with the smallest grid sizes in the forward and transverse directions, there will be redundant CL-point data in some regions and the machining efficiency will be reduced drastically. Therefore, it is critical to recognize the geometric features of the point-based surface within cutter path planning in order to achieve the machined surface with the desired accuracy, and with reduction of the machining time [5].

The present work is an attempt in this direction. It aims to develop a grid based adaptive planar tool path strategy directly from the point-cloud data with the adaptive grid sizes determined based on the surface slopes and curvatures to maintain the machined surface error (chordal deviation and cusp) within tolerance. In what follows, the grid based algorithm for tool path planning for 3-axis CNC machining is presented in detail.

3 GRID BASED ADAPTIVE TOOL PATH PLANNING SYSTEM

The developed system has three functional modules - Data pre-processor, Tool path generation and Post-processor. The system takes four inputs from the user viz; part model point cloud data, permissible chordal step error, permissible scallop height and the diameter of the ball end mill. It automatically generates the final post-processed CNC program (G/M format) for 3 axis CNC machine in ISO format. Various functional modules in the system are discussed one by one.

3.1 Data Pre-processor

Input to this system module is the part model point cloud data (X, Y, Z coordinates).This could be the data from a 3D CMM/Laser scanner as an unorganized point data set. The Pre-Processor module carries out two functional tasks viz. Uniform grid generation and (CL) grid point computation as outlined below

3.1.1 Uniform 2D grid generation

Input data points for the surface model are projected onto X-Y plane. A 2D mesh (grid) of points is prepared by computing extents (minimum, maximum) of the projected coordinates in the X-Y plane. Considering the Forward and Transverse (side step) directions to be X and Y, constant grid sizes g^x and g^y are computed in relation to the tool radius (Fig. 1).

Various grid spacing (g^x, g^y) based on the tool radius (R) were tried out. To test the efficacy of grid size (spacing), concept of tool shadow was used i.e. how many grid cells are present in the projected tool diameter (circle).The grid spacing (g^x, g^y), called as the step-forward and side step-intervals in cutter-path planning, are critical for ensuring the accuracy and efficiency of the machining process [9]. The step-forward distance controls local gouging in the forward (X) direction whereas the side step-interval distance governs cusps (scallops) in the transverse (Y) direction.

3.1.2 CL grid point Computation

For the uniform grid, CL grid points (Z values) are calculated using the Inverse tool offset algorithm (IOM) [9]. Considering each grid point of the mesh(X-Y) as center, a circle with a radius equal to tool radius (R) (tool shadow) is constructed and is moved successively over the whole grid surface.

Through the grid points which lie inside the circle CC-region (tool shadow area), grid lines running parallel to the tool axis (+Z) are constructed (Fig. 2).

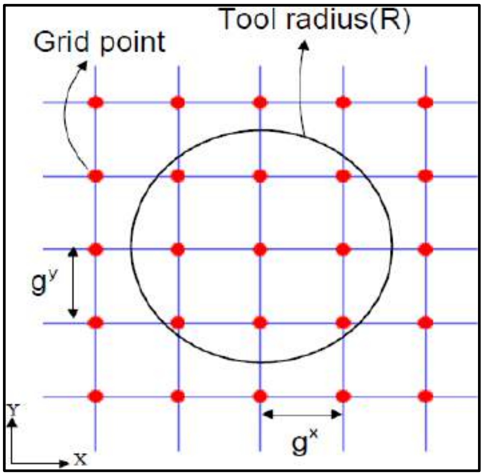


Figure 1: Uniform 2D grid cells.

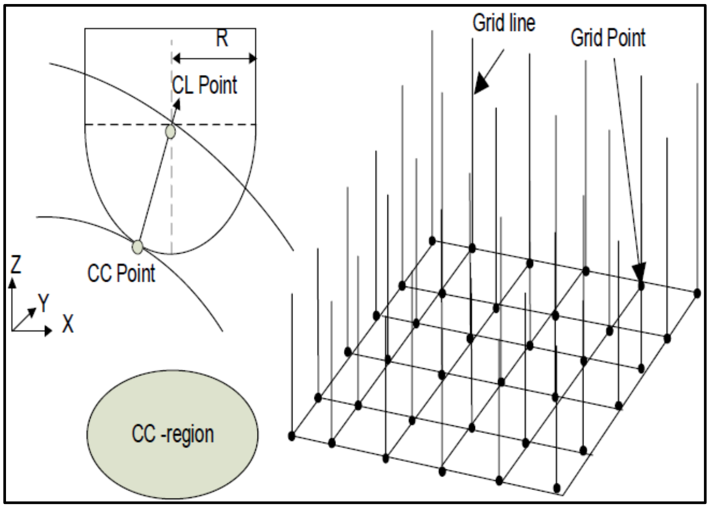
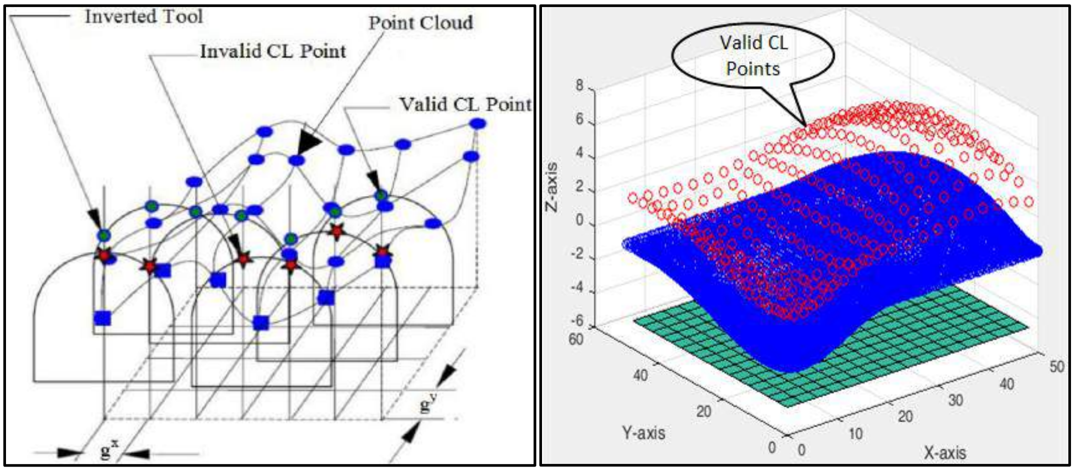


Figure 2: Tool shadow (CC) region with grid points and Z lines.

The cutter center point of the inverted ball end mill (Sphere) is placed at each point of the point cloud which lie inside the tool circle (CC-region).The grid line selected at the center of tool circle intersects with each inverted tool and produces intersection points. The highest intersection point (Z-height) is considered as the valid CL-point for each grid line and others are discarded (Fig. 3a). The same procedure is repeated at each grid point by successively moving the tool shadow circle. The point cloud is thus, mapped to generate valid CL points (Fig. 3b). The main advantage of this algorithm is in generating gouge free offset points.



(a) (b)

Figure 3: (a) Uniform CL point generation using IOM, (b) Valid CL points.

The toolpaths is defined as a sequence of valid CL-points with raster (Zigzag) planar path using linear interpolation. The initial (uniform) grid based tool path so generated is analyzed for chordal and scallop errors likely to be produced and adaptively refined to control the error values. The uniform CL point tool path generated can be used for roughing strategy with proper choice of flat end mill cutter.

3.2 Tool Path Generation

The tool path generation is divided into various functional submodules as under.

3.2.1 Curvature Computation in Forward and Side Step directions

To estimate the value of the chordal deviation (Forward error) in the X-Z plane and cusp (scallop) height (side step error) in the Y-Z plane, curvatures (ρ_1,ρ_2) are calculated in both X and Y directions from the initially generated uniform CL grid points. Curvature computation in both the forward and side step directions were carried out in Matlab [12]. The procedure first fits polygons to the set of 2D points and then calculates the analytical curvature from the polygons, using the neighboring points of each line/contour points. This algorithm was implemented and validated with the analytical curvature procedure for various case studies. It was found to give error less than 0.01% and was thus, chosen for this work. Using the curvature value, the chordal deviation due to the forward step and the cusp height due to the side step were calculated.

3.2.2 Computation of Chordal and Cusp Errors

To estimate value of the chordal deviation in the X-Z plane and cusp height in the Y-Z plane, the slope $\left(\alpha_{i,j}^{x},\alpha_{i,j}^{y}\right)$ and curvatures $\left(\rho_1,\rho_2\right)$ are calculated from the initially generated uniform CL points. Given a set of CL points (Fig. 4) slopes angles $\left(\alpha_{i,j}^{x},\alpha_{i,j}^{y}\right)$can be computed in the X and Y directions respectively as

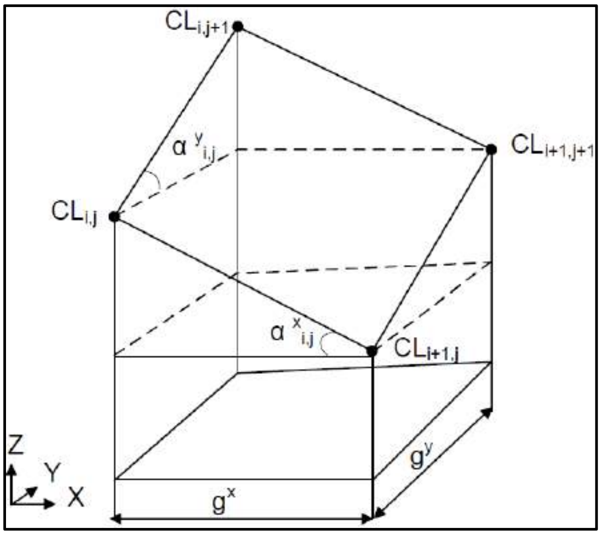


Figure 4: Slope angles for CL-segments.

$$\alpha^{x}_{i,j} = \cos^{-1}\frac{g^{x}}{\left\| CL_{i,j}\, CL_{i+1,j} \right\|} \tag{3.1}$$

$$\alpha^{y}_{i,j} = \cos^{-1}\frac{g^{y}}{\left\| CL_{i,j}\, CL_{i,j+1} \right\|} \tag{3.2}$$

3.2.3 Computation of Chordal Error

When the cutter path is approximated by connecting adjacent CL-points by straight line segments, the contour error is the deviation of the path from the corresponding CL-curve. This error $\left(e^{x}_{i,j}\right)$ is defined as the chordal deviation in X-Z (Fig. 5). Given a step-forward distance (g^x), the chordal deviation $\left(e^{x}_{i,j}\right)$ is calculated in terms of the slope $\left(\alpha^{x}_{i,j}\right)$ and curvature $\left(\rho_1\right)$ of the CL curve segment in the forward direction (X) [9].

For Convex region ($\rho_1 > 0$)

$$e^{x}_{i,j} = \rho_1 - \sqrt{\rho_1^2 - \left(\frac{g^{x}}{2cos\alpha^{x}_{i,j}}\right)^2} \tag{3.3}$$

For Concave region ($\rho_1 < 0$)

$$e^{x}_{i,j} = \rho_1 + \sqrt{\rho_1^2 - \left(\frac{g^{x}}{2cos\alpha^{x}_{i,j}}\right)^2} \tag{3.4}$$

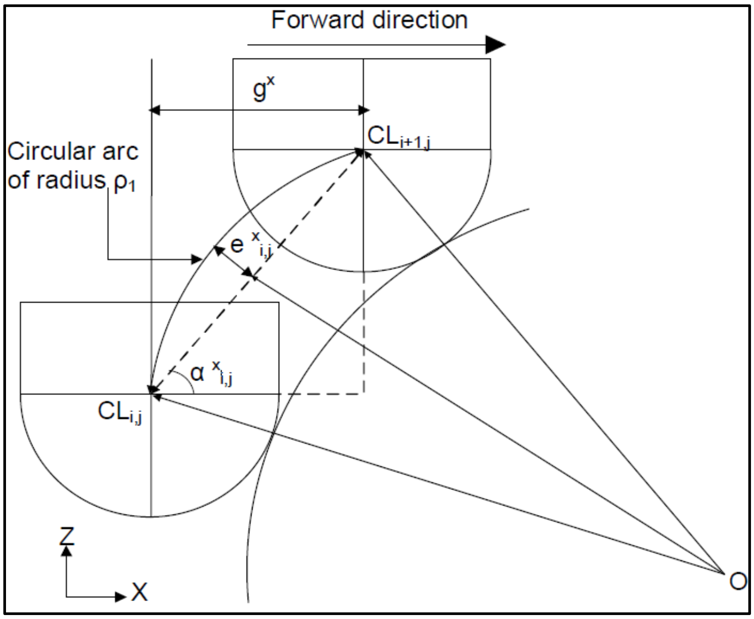


Figure 5: Chordal deviation in forward direction.

3.2.4 Computation of Cusp height

The region left un-machined between consecutive cutter paths in the transverse (side step) direction (Y-Z plane) is called the scallop or cusp, the upper limit being termed as the Cusp height $\left(e^y_{i,j}\right)$ (Fig. 6). Given a step-interval distance (g^y) and cutter radius R, the cusp height $\left(e^y_{i,j}\right)$ is derived in terms of the slope $\left(\alpha^y_{i,j}\right)$ and curvature (*ρ2*) of the CL-curve segment in the transverse (Y) direction [9].

For Convex region (*ρ2* > 0)

$$e^y_{i,j} = \sqrt{\rho_2^2 - \left(\frac{g^y}{2cos\alpha^y_{i,j}}\right)^2} - \sqrt{R^2 - \left(\frac{g^y}{2cos\alpha^y_{i,j}}\right)^2} - (\rho_2 - R) \qquad (3.5)$$

For Concave region (*ρ2* < 0)

$$e^y_{i,j} = \sqrt{\rho_2^2 - \left(\frac{g^y}{2cos\alpha^y_{i,j}}\right)^2} + \sqrt{R^2 - \left(\frac{g^y}{2cos\alpha^y_{i,j}}\right)^2} + (\rho_2 - R) \qquad (3.6)$$

These calculations are done at each grid point of the mesh cell.

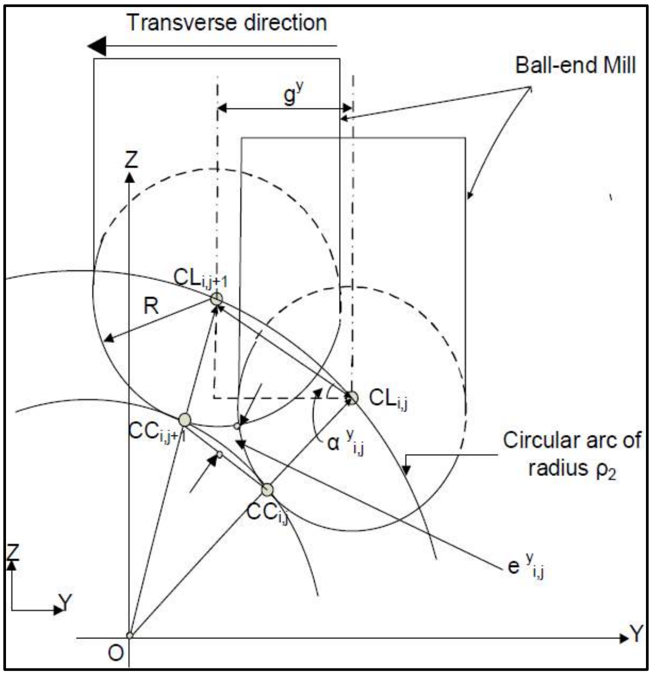


Figure 6: Cusp height in transverse direction.

3.2.5 Analysis of Error and Grid Segmentation

Analysis of machined surface error is carried out independently in both the forward and step directions from the CL-points generated with constant grid size g^x and g^y. If the machined surface error exceeds the specified tolerance, then the grid of points is refined through segmentation. The segmentation in the forward direction (X) is related to the number of linear segments required to complete the tool path. In practice, it governs the size of CNC program rather than the total distance (or the machining time) of the cutter path. Segmentation in the transverse direction, however, increases the total distance travelled by the cutter and consequently increases the machining time. As a result, to meet the desired tolerance segmentation is first done in the forward direction followed by the segmentation in the transverse (side step) direction. Important steps of algorithm are as under.

Given:A ($m\times n$) grid of CL-points generated with constant grid sizes g^x (X-direction) and g^y (Y-direction).

1. The chordal deviations and cusp heights are estimated using Eqns. (3.1-3.6).
2. If Error > user specified tolerance, choose
   $g^x = g^x/M$, where M =[1,……..10] and $g^y = g^y/N$, where N = [1,……..10]
3. The segmentation is performed by choosing the grid sizes $g^x/2, g^x/3 \ldots\ldots g^x/M$ (M is an integer-max 10) in the X direction and similarly in the Y direction respectively until the error is within specified tolerance. Store g^x and g^y value in new matrix corresponding to the calculated value of M and N. The radii of curvature and slope angles are unchanged during segmentation.
4. Store a new matrix (gx_new2) of $g^x[m\times n]$ in forward direction after segmentation
5. Store a new grid matrix (gy_new2) of $g^y[m\times n]$ after taking the maximum value of segments (grid points) between each two successive side steps(Y planes).
6. These new grid matrix gx and gy are input to the meshgrid which now becomes an adaptive meshgrid.

7. The entire procedure of computing new CL points, tool path, error computation and analysis (steps 1-6) is repeated to generate adaptive CL points till both chordal and scallop errors are in limit.

3.2.6 Adaptive Grid Refinement

Once all grid point segments are determined in both directions, adaptive grid is generated as under.

1. First store the new grid points for forward direction. Find the maximum value of segments (grid points) between each two successive side steps (Y) and store these new grid points in Y direction.
2. Generate adaptive grid of 2D points in the X-Y plane as shown in Fig. 7 using the number of segments at Step 1 in the X and Y directions respectively.

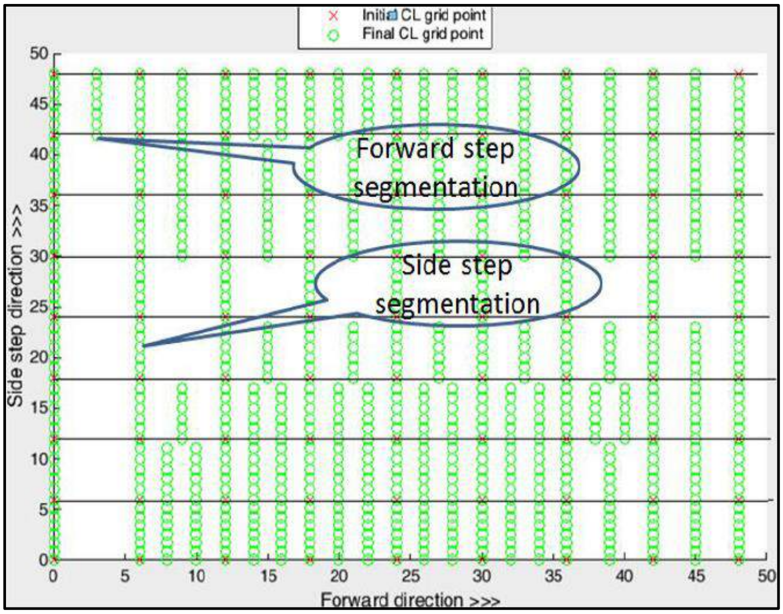


Figure 7: Adaptive Grid refinement.

These refined grid points are finally used as input for the adaptive CL point generation.

3.2.7 Generation of Adaptive CL point Tool path

The adaptive grid so generated, is used to generate the CL-points using the IOM algorithm outlined earlier (Section3.1). One more iteration is carried out for the generated adaptive grid which ensures that the new adaptive CL points provide tool path with estimated errors in limit. The new adaptive CL point's matrix [mxn] will be input to the postprocessor to generate the final CNC program. Fig. 8 shows typical adaptive CL points generated for the point based freeform surface model.

3.3 Post-Processing

The CL points generated finally are connected to create a continuous zig zag pattern for tool motion. Additional points are added at the start as well as the end of each tool path (Ramp On/Off) to ensure smooth entry and exit of tool into the workpiece. The CL path is post processed to ISO neutral as well as for FANUC (G/M) format to create the final CNC part program.

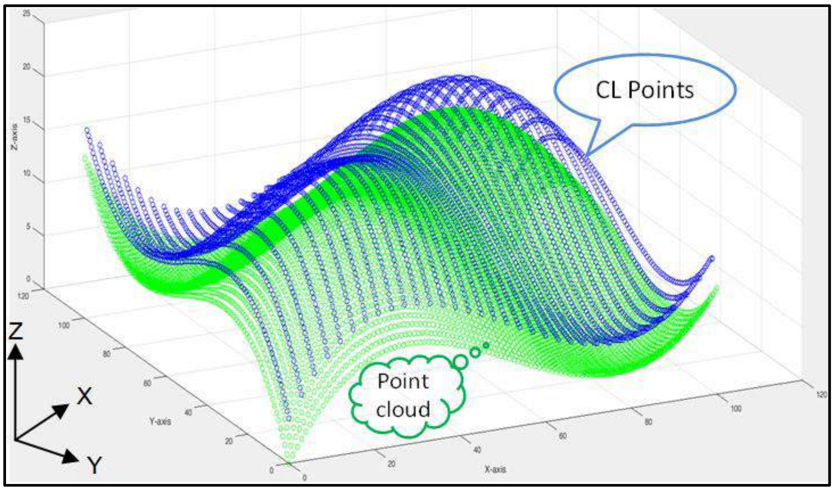


Figure 8: Adaptive CL points generated.

4 IMPLEMENTATION AND TESTING

The algorithms were implemented in MATLAB R2014b on Windows platform with Intel(R) Core(TM) i7-4790 CPU, 3.60GHz processor and 16GB RAM. The system was extensively tested for various part models with varying complexities. For the purpose of testing the algorithms, simulated point data sets were created by extraction of points from STL files and parametric surface models of complex freeform surfaces. Effect of varying the initial grid size in the algorithm and the point cloud density were studied. Results are presented in the sections to follow.

4.1 Effect of variation of grid sizes

Grid size is an important parameter in the algorithm which governs the scallop height, machining time and code length (memory) of the CNC program. To test the validity of the system, initial grid size equal to tool radius ($g^x = g^y = R$) was used. The results generated by our system were compared with those from commercial software (Mastercam X5).The post processed CNC program (G/M code) was simulated in Vericut 8 software, and the performance parameters such as machining time and scallop height were estimated (measured) using Vericut software and our developed data processing software (Appendix A). Results for two typical case studies are presented here.

4.1.1 Case Studies

Fig. 9 shows the point cloud part model C1-10201 points (work size 50x50x10mm), CL points generated and the planar meshgrid with grid size equal to tool radius (R). For all the case studies, following parameters were set for the generation of tool paths and the NC program: Allowable forward step error: 0.05mm; Allowable Scallop Height: 0.05mm; Tool diameter: 6.35mm (0.2 inches); Feed: 127.28 mmpm (5 inches/min); Spindle Speed: 1591rpm.

CNC program generated with our system was simulated in Vericut 8.0 software (Fig. 10) for visualization and estimation of parameters such as machining time and scallop heights.

Table 1 shows the effect of variation of initial grid size from coarser ($g^x = g^y = 2R$) to finer ($g^x = g^y = R/3$) on code length (size) and the estimated machining time (path length) for all the case studies together with the results obtained with Matercam. It is seen that the side step is a significant factor in path planning which governs the machining time. For example in C1(Table 1), comparing the final adaptive grid size for R/2 - (51x67) and R/3 –(51x84), Forward grid size 51 remains same while the side step grid size 67 increases to 84, thereby increases the machining time from 28:36min

(R/2) to 35:31 min (R/3). This is because each additional row (grid point in Y side step) is a new tool path which will add to the machining time. Insertion of points in forward step controls only the chordal error and does not affect the machining time much. It however, increases the code length.

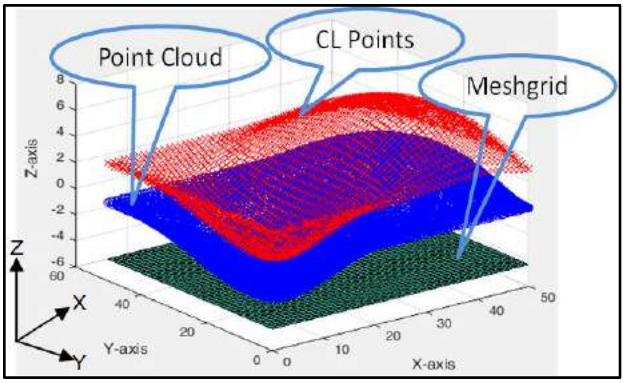


Figure 9: Case study–C1.

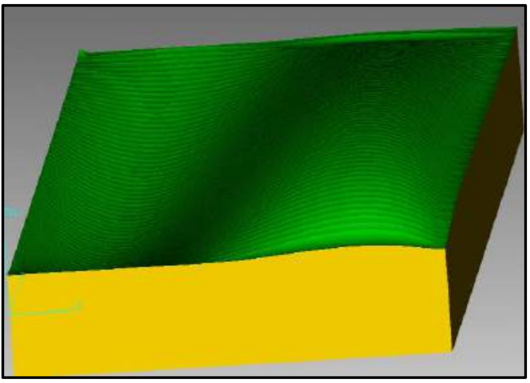


Figure 10: Simulated machined surface in Vericut(C1).

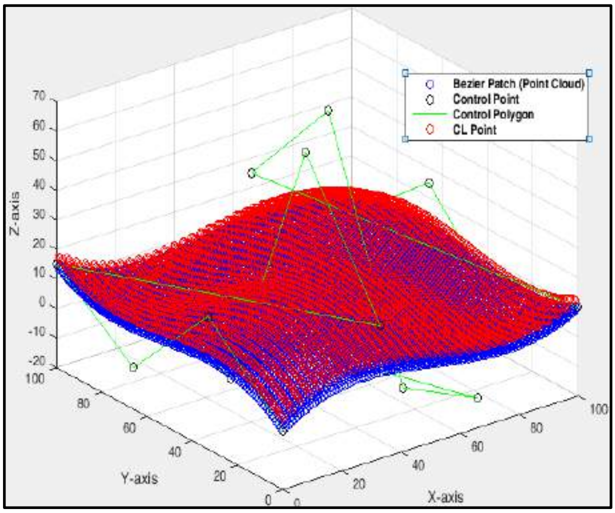
Case Study	Grid size $g^x = g^y$	Starting grid size	Final grid size	Code length	Machining time(min)	Mastercam results
C1	2R	9x9	21x50	885	20:41	Code length-4334, M/c time-25:13 min
	R	17x17	28x54	1085	22:16	
	R/2	34x34	51x67	2356	28:36	
	R/3	51x51	51x84	4294	35:31	
C2	2R	17x17	30x100	2542	1h:18min:53s	Code length-1708, M/c time-1h:27min:26s
	R	34x34	34x108	3682	1h:27min:56s	
	R/2	67x67	73x133	8945	1h:48min:03s	
	R/3	101x101	110x144	14590	1h:58min:22s	

Table 1: Effect of variation of grid size.

To test the optimality of grid size and adaptive nature of the algorithm, a complex Bicubic Bezier surface patch (Case study- C2) was specifically designed with 16 control points. The patch was rectangular in both the Parametric (u,v) and Cartesian domain (X-Y projection). Equation of Bezier Bicubic Surface patch is given as [18]

$$P(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3} B_{i,3}(u)B_{j,3}(v)P_{i,j}\,,\ 0 \le u \le 1 \text{ and } 0 \le v \le 1 \qquad (4.1)$$

Where $B_{i,3}$ (u) and $B_{j,3}$ (v) are the Bernstein basis functions in the u and v parametric directions. This case study was conceived as an endurance test for a complex profile (part -100x100x80mm - 1002001 points). Fig. 11 shows the control points and the CL points generated over the point cloud model. All the parameters were kept same for the generation of tool path as in case study C1. The system successfully generated the CNC program with 3896 lines. Fig. 12 shows the machined surface simulated in Vericut. Trends of results in case study C2 (Table1) were similar to Case Study C1. For example comparing final grid size 2R – (30x100) and for R – (34x108), it is seen that the machining time increases from 1:18:53min (2R) to 1:27:56 min(R). The side step thus, governs the machining time. The Case studies (Table1) suggest that from the consideration of machining time only, grid size [2R or R] can be chosen as an optimal starting grid, the values being less than those from Mastercam.



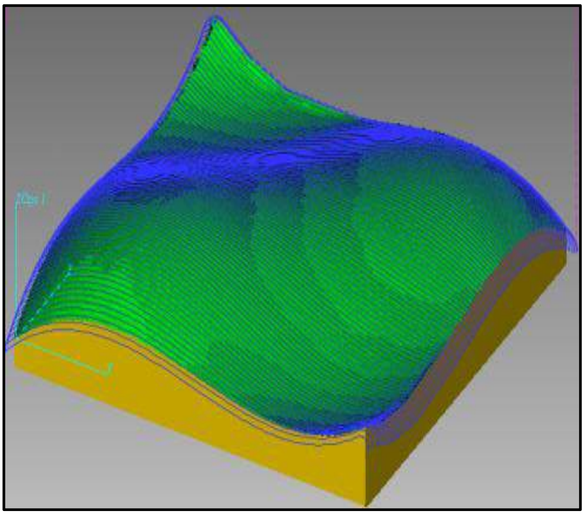


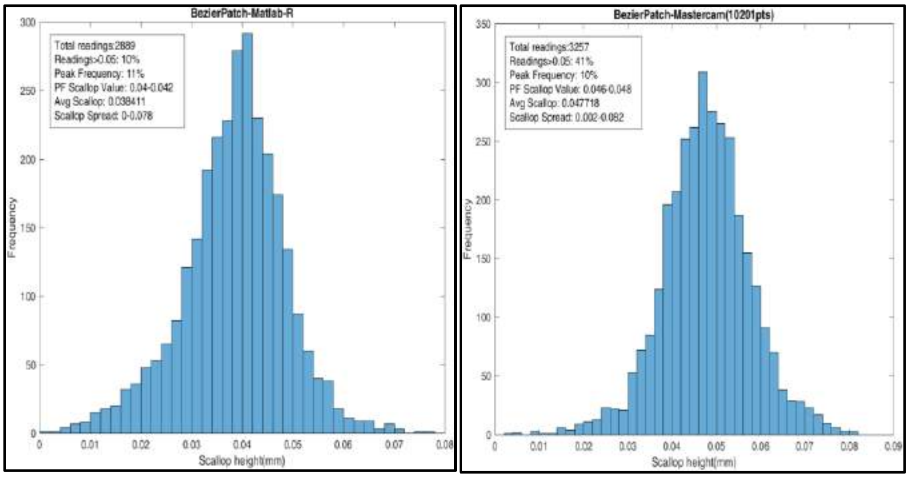
Figure 11: Case study–C2.

Figure 12: Simulated machined surface in Vericut(C2).

Study of variation of scallop for these grid sizes gives further insight into the optimality of grid. Scallop computation of various grid sizes for the case studies (C1, C2) was carried out on the simulated machined surface. Vericut software provides the user a CAD measurement tool to find scallop height at particular points. This system of measurement however, is very approximate, cumbersome, error-prone and can even leave some of the areas unobserved. Hence, data processing software was designed and developed for the computation and evaluation of the surface quality/Scallop from the simulated CNC program. Appendix A presents the salient features of this scallop height calculation module in detail.

Scallop estimation for the two case studies was done on simulated machined surface with the help of the developed scallop computation system (Appendix A) and histograms of Scallop variation Vs Frequency were generated for all cases. Histogram for one typical case study (C2) with grid size=R and Mastercam is presented here (Fig. 13).

Fig. 13(a) shows that scallop is well controlled within the chosen tolerance limits (0.05mm) over the surface indicating the effectiveness of the adaptive nature of the tool path generated by our system. About 90-95% of scallop values are under tolerance (<0.05mm) with very few readings going beyond the prescribed tolerance limit. This could also be due to the approximations resulting from the STL file from Vericut simulated machined surface, curve fitting and the choice of spacing between cross sectional planes for computation. Histogram (Fig. 13(a)) shows that the % of scallop values which exceed the tolerance value (0.05mm) are 5-10% for our system compared to about 41% for Mastercam (Fig. 13(b)).



(a) (b)

Figure 13: Scallop-Frequency Histogram plots (C2): (a) grid size = R; (b) Mastercam.

Table 2 shows the consolidated results of the scallop-frequency histogram generated for the two cases. It is seen that, the performance parameters given by our algorithm are superior compared to those from Mastercam. From the scallop consideration, there is no perceptible difference between grid size 2R, R and R/2 in terms of Average scallop height and % exceeding tolerance value. From product accuracy point of view, grid size R/2 is better while from the machining time considerations choice of 2R seems good. As a tradeoff between part surface accuracy and the machining time, initial grid size R would, thus, be a good choice. Peak frequency of scallops is increasing from coarser grid size 2R (10%) to denser R/3(13%) with reduction in the scallop spread which also suggest that initial grid size R would be optimal.

Case Study	Grid size $g^x = g^y =$	Reading>0.05mm	Peak Frequency	PF Scallop Range	Avg. Scallop	Scallop Spread
C1	2R	2%	25%	0.039-0.042	0.0391	0.024-0.06
	R	0%	29%	0.035-0.04	0.0354	0.015-0.05
	R/2	0%	24%	0.024-0.026	0.0231	0.008-0.032
	R/3	0%	42%	0.01-0.015	0.0174	0-0.04
	Mastercam	21%	22%	0.048-0.051	0.0452	0.024-0.057
C2	2R	11%	10%	0.04-0.042	0.0389	0.001-0.074
	R	10%	11%	0.04-0.042	0.0384	0.001-0.078
	R/2	5%	12%	0.027-0.03	0.0302	0.001-0.075
	R/3	6%	13%	0.039-0.042	0.0336	0.001-0.069
	Mastercam	41%	10%	0.046-0.048	0.0477	0.002-0.082

Table 2: Scallop Frequency Histogram- consolidated result.

Table 3 presents the variation of average scallop for various grid sizes chosen for the case studies. It can be seen that average scallop height depends significantly on the increase in grid size in the side step direction(Y direction) to meet the scallop tolerance requirement. Using Average scallop to rank and select the optimum grid size, it looks best to have coarsest possible grid size which meets the tolerance value as this will minimize both the code size and machining time. In general, the Y grid gets finer to meet the tolerance (scallop) requirement. Further refining the grid would not improve part quality but would only increase no. of CL points, the code length as well as the machining time. On the whole, choice of the starting grid spacing ($g^x = g^y$ = R) looks logical. This would provide a good trade-off between amount of path redundancy and the fineness of scallop. We have thus, chosen the optimal starting grid size for our system to be $g^x = g^y$ = R.

Grid Size	Case Study (C1)			Case Study (C2)		
$g^x = g^y =$	Initial Grid Size	Final Grid Size	Average Scallop (mm)	Initial Grid Size	Final Grid Size	Average Scallop (mm)
2R	9x9	21x50	0.0391	17x17	30x100	0.0389
R	17x17	28x54	0.0354	34x34	34x108	0.0384
R/2	34x34	51x67	0.0231	67x67	73x133	0.0302
R/3	51x51	51x84	0.0174	101x101	110x144	0.0336

Table 3: Effect of variation of grid size on average scallop.

4.2 Effect of variation of point density

To study the effect of variation of point density, the CAD models were simplified in MeshLab and points were extracted. Table 4 shows the effect of point density on the quality of CNC program in terms of % scallop exceeding tolerance, average scallop height and the spread. It is seen that reducing the point cloud density, % scallop readings exceeding tolerance limit (50μm) increases eventually.

Case Study	No. of Cloud Points	Scallop Reading>0.05mm	Avg Scallop(mm)	Scallop Spread(mm)
C1	10201	0%	0.0354	0.015-0.05
	5177	3%	0.0364	0.015-0.06
	2686	30%	0.0456	0.02-0.09
C2	111556	6%	0.0369	0.002-0.068
	63001	6%	0.0368	0.002-0.064
	40401	7%	0.0372	0.002-0.068
	10201	10%	0.0384	0.001-0.078
	8281	15%	0.0393	0.001-0.084
	4489	61%	0.0574	0-0.12

Table 4: Effect of variation of point density.

Complexity of the surface governs the minimum number of points required to control scallop within tolerance. For the case 1(Table 4), approximately 5000 points seems to minimum required while for case 2, approximately 10000 points seems to minimum required. Increasing point density beyond these values does not improve the part quality and results in the redundancy of points. Effect of sparse points seems to be very significant. Since most scanners today produce dense point cloud, the quality of CNC program generated by our algorithm will not be affected by the point cloud in terms of scallop control.

4.3 Comparison with commercial software

The CNC programs generated by our system were compared with those by Mastercam X5 for the same part surface. Since, Mastercam does not take point cloud as an input for toolpath generation; it was converted to an STL file and input to Mastercam. Iso-planar Zigzag strategy was employed in Mastercam for scallop tolerance limit 0.05mm (with a side step of 1.09mm). NC files generated both from our system and Mastercam were simulated in Vericut for comparing performance parameters like code length, cutting distance, machining cycle time, MRR (material removal rate) and average scallop. MRR is estimated as the ratio of volume removal/machining time. The scallop tolerance limit was given as an input and fixed side step was chosen automatically in Mastercam as it does not provide any guidance to choose the step-over. In comparison the side step value in our system is automatically calculated and is seen to vary adaptively from 0.75 to 1mm for the cases. Optimal initial grid size was chosen equal to R for all the cases. Table 5 shows the results of the comparison.

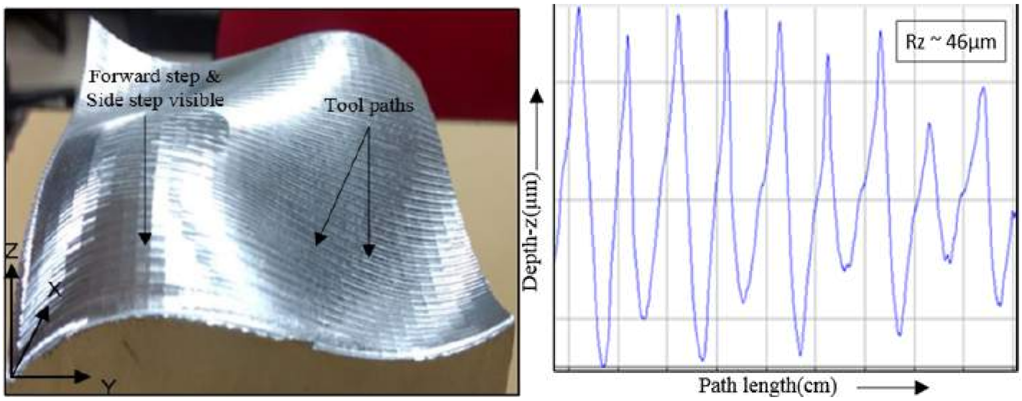
Case Study	Parameter	Our System	MastercamX5 (Iso-planar)
C1	Code Length	1085	4334
	Cutting Distance (mm)	2879.30	4243.63
	Machining Time (min)	22:16	25:13
	Volume Removal (mm^3)	17017.49	24619.50
	MRR (mm^3/min)	767.93	979.69
	Average Scallop (mm)	0.0354	0.0452
C2	Code Length	3682	1708
	Cutting Distance (mm)	11414.32	12193.67
	Machining Time (hr:min:sec)	1:27:56	1:27:26
	Volume Removal (mm^3)	105767.15	135553.44
	MRR (mm^3/min)	1207.94	1553.44
	Average Scallop (mm)	0.0384	0.0477

Table 5: Comparison of developed strategy with Mastercam X5.

It is seen that the performance parameters (code length, cutting distance, machining time, MRR, average scallop) for case study C1 and C2 given by our system are less than their corresponding values given by Mastercam X5-Isoplanar strategy. In particular, for Case C1 the reductions in values are: Code length (75%), cutting distance (32%), machining time (12%), MRR (21%) and average scallop (22%). For Case C2, the measures are cutting distance (6%), MRR (22%) and average scallop (19%) less than Mastercam with comparable machining time. To control the chordal error, more segmentation seems to have occurred in forward direction which possibly increased the code length. The machining time however, has remained unchanged. From productivity point of view for case C1, our algorithm is much superior whereas for C2 it is closer to Mastercam values. But from quality (scallop) consideration our algorithm for both cases (C1 and C2) performs better than Mastercam. This shows that the adaptive grid based planar strategy developed in this work is much superior to the iso-planar strategy in the commercial software.

4.4 Machining Trial

To validate the CNC program generated by our system, machining trial was carried out on a 3-axis Hardinge VMC 600II Siemens (Sinumerik controller) CNC Machine available at IIT Bombay. NC file for case study C2 (Bezier Patch) was chosen for this trial. Machining was carried out on Aluminium block (100x100x80mm) in two stages viz. Roughing followed by Finishing. Rough machining was done with Flat end Mill (Ø6mm) leaving 1mm stock on drive for finish cut. Following parameters are chosen for finish machining: Tool: Ball end Mill (Ø6mm), Feed rate: 127.8 mmpm, Spindle speed: 1500 rpm, Total tolerance: 0.05mm, Finish depth of cut: 1mm, Cutting Method: Zigzag Adaptive planar.



(a) (b)

Figure 14: (a) Finishing Metal Component (C2), (b) Surface roughness measurement.

Fig. 14(a) shows the finished metal component. The part looks very similar to the CAD model (Bezier freeform surface) designed and had a good surface finish. The finishing tool path (forward and side step) with fine scallop and the complex shape produced are clearly seen.

Surface roughness (Scallop height) measurement was carried out on the machined component with the help of non-contact optical Surface Profilometer (Alicona- Infinitefocus G4) available at IIT Bombay. Fig. 14(b) shows the result of one of the Alicona images part. Roughness measurements perpendicular to the lay direction were analyzed. Results show that Rz (peak to valley height roughness) values for the surface varies between 20µm to 46µm and are within tolerance limit (50µm) specified. This indicates the adaptive nature of the developed system.

5 CONCLUSIONS

The grid based adaptive tool path planning strategy developed in this work generates efficient CNC program directly from the point cloud without the need of surface reconstruction. The developed strategy reduces path redundancy and keeps variation in scallop height under control due to the adaptive nature of the tool path. Selection of grid sizes for roughing and finishing is governed by whether to achieve high productivity(less machining time) or superior part quality (high surface finish). Our algorithm provides a trade-off between redundant path planning and the requirement of finer scallop. Grid size (segmentation) in side step direction governs the average scallop height and machining time. Sparse point density affects the quality of CNC code in terms of scallop height generated. However for dense point sets, part quality provided by our algorithm is invariant. The system offers significant advantages over the popular commercial iso-planar strategy in terms of enhanced productivity and product quality. In the scope of the case studies considered, the strategy gives significant reduction in NC program code length, cutting distance, machining time, MRR and average scallop depending on the complexity of the surface. The algorithm is robust and computationally efficient in generating error free CNC program.

ACKNOLEDGEMENT

The financial support received by the first author from the NCAIR (National Centre for Aerospace Innovation and Research) of IIT Bombay-India is gratefully acknowledged.

Mandeep Dhanda, https://orcid.org/0000-0003-1168-9299
S.S. Pande, https://orcid.org/0000-0001-8805-1854

REFERENCES

[1] Choi, B.K.; Jerard, R.B.: Scupltured surface machining: theory and applications, Dordrecht: Kluwer Academic Publishers, the University of Michigan, USA, 1998.

[2] Chui, K.L.; Chiu, W.K.; Yu, K.M.: Direct 5-axis tool-path generation from point cloud input using 3D biarc fitting, Robotics and Computer-Integrated Manufacturing, 24(2) ,2008, 270–286. https://doi.org/10.1016/j.rcim.2006.11.004

[3] Chui, K.L.; Yu, K.M.; Lee, T. C.: Direct tool-path generation from massive point input, Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 216(2), 2002, 199-206. https://doi.org/10.1243/0954405021519843

[4] Ding, S.; Mannan, M.A.; Poo, A.N.; Yang, D.C.H.; Han, Z.: Adaptive iso-planar tool path generation for machining of free-form surfaces, Computer-Aided Design, 35(2), 2003, 141-153. https://doi.org/10.1016/S0010-4485(02)00048-9

[5] Feng, H.-Y.; Teng, Z.: Iso-planar piecewise linear NC tool path generation from discrete measured data points, Computer-Aided Design, 37(1), 2005, 55–64. https://doi.org/10.1016/j.cad.2004.04.001

[6] Giri, V.; Bezbaruah, D.; Bubna, P.; Choudhury, A. R.: Selection of master cutter paths in sculptured surface machining by employing curvature principle, International Journal of Machine Tools and Manufacture, 45(10), 2005, 1202–1209. https://doi.org/10.1016/j.ijmachtools.2004.12.008

[7] Han, Z.; Yang, D. C. H.: Iso-phote Based Tool-path Generation for Machining Free-form Surfaces, Journal of Manufacturing Science and Engineering, 121(4), 1999, 656-664. https://doi.org/10.1115/1.2833094

[8] Huang, Y.; Oliver, J.H.: Non-constant parameter NC tool path generation on sculptured surfaces, The International Journal of Advanced Manufacturing Technology, 9(5), 1994, 281–290. https://doi.org/10.1007/BF01781282

[9] Kayal, P.: Inverse offset method for adaptive cutter path generation from point-based surface, International journal of CAD/CAM, 7, 2007, 1-18.

[10] Kayal, P.: Offset error analysis of ball-end mill for cutter path generation from point-based Surfaces, The International Journal of Advanced Manufacturing Technology, 36(11-12), 2008, 1133-1144. https://doi.org/10.1007/s00170-007-0933-z

[11] Kishinami, T.; Kondo, T.; Satio, K.: Inverse offset method for cutter path generation, Proceedings of the 6th International Conference on Production Engineering, Osaka, 1987, 807-812.

[12] Kroon, D.-J.: Accurate Curvature and Normals of a line/contour consisting of 2D points, Matlab Function File Exchange-Line Curvature 2D, University of Twente, Netherlands, 2011.

[13] Lasemi, A.; Xue, D.; Gu, P.: Recent development in CNC machining of freeform surfaces-A state-of-art-review, Computer-Aided Design, 42(7), 2010, 641-654. https://doi.org/10.1016/j.cad.2010.04.002

[14] Lee, A.-C.; Chen, D.-P.; Lin, C.-L.: A CAD/CAM system from 3D coordinate measuring data, International Journal of Production Research, 28(12), 1990, 2353–2371. https://doi.org/10.1080/00207549008942871

[15] Lin, A.C.; Liu, H.-T.: Automatic generation of NC cutter path from massive data points, Computer-Aided Design, 30(1), 1998, 77-90. https://doi.org/10.1016/S0010-4485(97)00066-3

[16] Liu, W.; Zhou, L.-S.; An, L.-L.: Constant scallop-height tool path generation for three-axis discrete data points machining, International Journal of Advanced Manufacturing Technology, 63(1-4), 2012, 137–146. https://doi.org/10.1007/s00170-011-3892-3

[17] Loney, G.C; Ozsoy, T.M: NC machining of free form surfaces, Computer -Aided Design, 19(2), 1987, 85–90. https://doi.org/10.1016/S0010-4485(87)80050-7

[18] Pande, S.S.: Computer Graphics and Product Modeling for CAD/CAM, Alpha Science International Limited, India, 2011.

[19] Suresh, K.; Yang, D.C.H.: Constant Scallop-height Machining of Free-form Surfaces, ASME Journal of Engineering for Industry, 116(2), 1994, 253–259. https://doi.org/10.1115/1.2901938

[20] Teng, Z.; Feng, H.-Y..; Azeem, A.: Generating efficient tool paths from point cloud data via machining area segmentation, The International Journal of Advanced Manufacturing Technology 30(3–4), 2006, 254–260. https://doi.org/10.1007/s00170-005-0081-2

[21] Tournier, C.; Duc, E.: A Surface Based Approach for Constant Scallop Height Tool-Path Generation, The International Journal of Advanced Manufacturing Technology, 19(5), 2002, 318-324. https://doi.org/10.1007/s001700200019

[22] Zhongwei, Y.; Shouwei, J.: Iso-phote based adaptive surface fitting to digitized points and its applications in region-based tool path generation, slicing and surface triangulation, Computers in Industry, 55(1), 2004, 15–28. https://doi.org/10.1016/j.compind.2004.01.002

APPENDIX A

ALGORITHM FOR SCALLOP HEIGHT CALCULATION

NC code from the developed system is simulated in Vericut 8.0 and machined surface is exported as STL file. The STL file is sliced using parallel planes that are perpendicular to the lay direction. The intersection of the planes and STL model would give points which, in turn, form the surface topology-pattern with waviness and roughness. Due to the direct slicing of STL facets, the scallop curve is well defined with very less noise. After getting the point data, next task is to remove waviness and calculate scallop heights from the shorter frequency point curves. The approach used here is the polynomial curve fitting to the points. Polynomial models for curves are given by

$$y = \sum_{k=0}^{n} p_i x^{n+1-i} \tag{A.1}$$

Where n +1 is the order of the polynomial, n is the degree of the polynomial chosen to be 1 ≤ n ≤ 9. Fig. A.1 shows the fitted curve and the residual (scallop curve) for the same fit.
Residual is calculated by

$$R = y(xi) - yi \tag{A.2}$$

Where R = residual at the point no. i, np is the total no. of points in the sliced plane/projected points on a plane, and 1 ≤ i ≤ np.

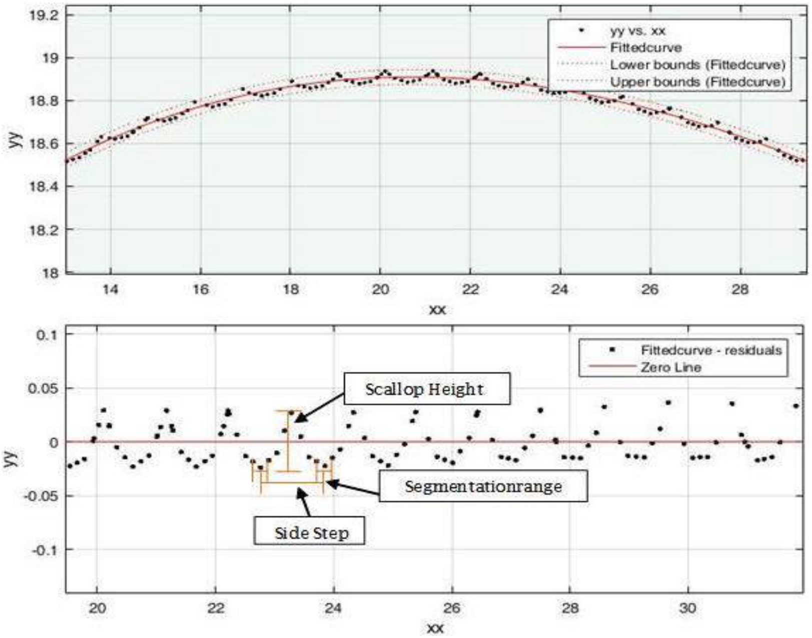


Fig. A.1: Polynomial curve fitting and scallop pattern plot

Scallop pattern can be clearly seen (Fig. A.1) after the residual calculation. The area is then segmented based on the side step that is used for tool paths. Scallop height can be calculated by

$$S = \max(yi) - \min(yi) \qquad \text{Where S is scallop height} \tag{A.3}$$

Scallop values computed at various points in each plane are stored to calculate Average scallop height and Histogram.