Modelling & Tool Path Generation of Complex Bezier Surface for Reverse Engineering Applications

Vishal Saxena
Department of Mechanical Engineering
Uttar Pradesh Technical University, India

Abstract— The present work is to develop a model of three-dimensional surface for reverse engineering. A generalized surface modelling for free form surfaces has been presented in this work. Surface scanning re-modelling and tool path generation to produce the required surface are being used very frequently to develop upon an already existing product within least possible time. In this paper, complex Bezier surface patch has considered a free form surface for modelling as well as generating tool path. For Bezier surface patch, the control points have defined for the patch and using this single patch, control vertices have been created. Then within the range of Bezier parameters u and v between 0 and 1, Bezier boundary/smoothening points have been developed for these defined vertices, and by plotting these points for various rows and columns, smoothening Bezier surface patch has been generated and tool path planning is done with the help of smoothening points.

Key words: Reverse Engineering, Surface Scanning Re-Modelling, Complex Bezier Surface Patch, Smoothening Points, Tool Path Planning

I. Introduction

A. *Free Form Surface: Bezier Surface Patch*

Free form surfaces do not have fixed linear dimensions, unlike regular surfaces such as planes, cylinders and conic surfaces. The form of complex free form surfaces or curves are not stored or defined in CAD software in terms of polynomial equations but by their pole, degree and number of patches. The degree of a surface determines the shape by a polynomial in terms of mathematical properties with variables to the power of the degree value. For example, a surface would be a degree of 1, would be a flat cross section surface, a surface with a degree of 2 would be curved in one direction and a surface with degree of 3 could change once from concave to convex curvature (but does not necessarily).

In a complex Bezier surface patch, there is one more pole than the degree values of the surface. The smoothness between patches is known as continuity, which mostly referred in terms of C type continuity:

1) C0: just touching, two curves simply meet.
2) C1: tangent of two curves, but could have sudden change in radius of curvature.
3) C2: the patches are continuous curvature to one another.

Bezier surface patches are a part of mathematical spline commonly used in computer graphics, computer aided design and finite element modelling. Like Bezier Curve, a Bezier surface is also defined by a set of control points. Similar to interpolation in every field, the major difference is that the surface in general, does not pass through the central control points.

A given complex Bezier surface of degree (m, n) is defined by a set of (n+1), (m+1) control points $k_{i,j}$. It maps the unit square into a smooth continuous surface, embedded within the space of same dimensionality as $\{k_{i,j}\}$. For example, if k are all points in a four dimensional space, then the surface will be within a four dimensional space.

A two-dimensional Bezier surface patch can be defined as a parametric surface, where the position of a point p as a function of the parametric coordinates u, which is given by:

$$p(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u) B_j^m(v) k_{ij} \quad (1.1)$$

evaluated over the unit square, where

$$B_i^n(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad (1.2)$$

is a Bernstein polynomial, and

$$\binom{n}{i} = \frac{n!}{(n-i)!} \quad (1.3)$$

is the binomial coefficient.

Some properties of Bezier surfaces are as follows:

1) Bezier surface will convert like its control points under all linear transformations and translations.
2) All u = constant and v= constant lines in the parametric space, and, in particular, all the four edges of deformed (u, v) unit square are Bezier curves.
3) A Bezier surface will completely inside the convex hull of its control points, and therefore will completely inside the boundary box of its control points in any given Cartesian coordinate system. The points in the patch corresponding to the corners of the deformed unit square coincide with four of the control points.
4) A Bezier surface does not generally pass through its other control points. Generally, the most common use of Bezier surfaces is as nets of bi-cubic patches (where m = n = 3). The geometry of a single bi-cubic patch is thus completely defined by a set of 16 control points.
5) Bezier patches meshes are superior to meshes of triangles as a representation of smooth surfaces. It is easy to manipulate and have much better continuity properties.
6) Bezier patch of degree (m, n) may be constructed out of two Bezier triangles of degree (m + n), or out of a single Bezier triangle of degree (m + n), with the input domain as a square, instead of a triangle.

B. *Modelling*

During the late 1970s, surface modelling was developed in the automotive industries to design and manufacture complicated shapes. A new approach to interactive modelling of free form surfaces has been discussed. Instead of a fixed mesh of control points, the model presented to the user is that of an infinitely malleable surface, with no fixed controls. The user is free to apply control points and curves which are then available as handles for direct manipulation. The complexity of the surface's shape may be increased by adding more control points and curves, without apparent limit. Within the constraints described by the controls, the shape of the surface is fully determined by one simple criteria, such as smoothness. The previous discussed method which is used to solve the resulting constrained variation optimization problems rests on a surface representation scheme allow no uniform subdivision of B-spline surfaces. Automatic subdivision is used to ensure that constraints are met, and to enforce error bounds. Efficient numerical solutions are obtained by exploiting linearities in the problem formulation and the representation. [1]

Lizhuang Ma and Qunsheng Peng [2] presented smoothing of free form surfaces with Bezier patches. They proposed a unified method for smoothing free form surfaces with Bezier patches is presented. A non-smooth surface consisting of bi-cubic patches is turned into a G1 surface by trimming each patch along user-defined parameter lines using subdivision. Attach eight patches (one to each edge and one to each vertex) to each trimmed patch to blend between it and its neighbour patches. Using this method, solid objects bounded with bi-cubic Bezier patches can be G1 smoothed with bi-cubic patches. The operation is local. It provides designers with the facility to adjust the shape of a smoothed object interactively. Based on the theory of geometric continuity, they proposed a new method for smoothing shapes bounded by Bezier patches has been presented. It allows one to model the fillet and rounding surfaces using bi-cubic patches flexibly.

Most of the proposed solutions for reverse engineering are realized in two steps (Fig.1). During the first step [3], the physical model is measured or digitized by a measuring device, e.g. a co-ordinate measuring machine (CMM) and surface points are captured in 3-D coordinates. Depending upon the measuring hardware used and measuring strategies applied, the measured points can be distributed according to a regular or irregular grid, following sectional lines, or even randomly. During the second step, a CAD model is reconstructed from the measured points.

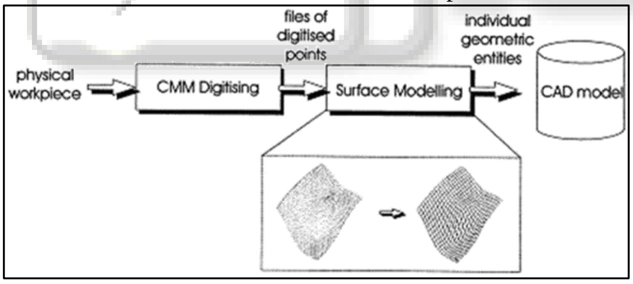

Fig. 1: Two Steps Reverse Engineering Approach [3]

Interval Bezier curves are new representation forms of parametric curves that can embody a complete description of coefficient errors. Using this new representation, the problem of lack of robustness in all state of all CAD systems can be largely overcome. In this work [4], the problem of bounding interval Bezier curves with lower degree interval Bezier curve has been discussed. It includes two different methods- Linear Programming and Optimal Approximation to solve this problem and provide several examples to demonstrate the algorithms.

C. Tool Path Generation

The tool path generation can be classified into three methods [5] namely, iso-parametric, iso-planning, and iso-scallop height.

In iso-parametric method, Loney and Ozsoy (1987) [6] and Broomhead and Edkins (1986) [7] have studied the tool path generation using iso-parametric curve. They approximately generated tool path into surface using iso-parametric curve on the surfaces.

In iso-planar machining, the tool paths are along with the series of planes on the part. Bobrow (1985) [8] and Huang and Oliver (1994) [9] have developed iso-planar NC tool path generation. A part which consists of CSG (constructive solid geometry) is divided by a series of planes to obtain cutter contact points.

The last approach is iso-scallop machining, Suresh and Yang (1994) [10] and Lin and Koren (1996) [11] have studied scallop height machining. They proposed algorithm to generate tool path based on desired accuracy on the manufactured part. Using this method, the accuracy of the manufactured part is controlled by user directly. In this method, the primary goal is to generate shortest overall length of tool paths with predetermined tolerance and scallop height of the manufactured part.

Computer aided manufacturing (CAM) as well as computer numerical control (CNC) is widely used in today's manufacturing industries. To produce the desired part, the CAM system generates the tool path for the CNC machine tool so that it can cut the surfaces of the part. In the current approach, the CAM system first computes the cutter contact (CC) points and then offsets these points to get the cutter location (CL) points. The CC points are a set of points located on the sculptured surface so that the edge of the tool is scheduled to pass through, while the CL points are a set of points that the centre (or the bottom tip) is scheduled to pass through. In this work [12], the CC path and the CL paths are defined as the paths passing through the CC points and CL points respectively.

The CL path is in practice the tool path. Conventionally, the CL path is represented by a set of consecutive linear segments (that connect the CL points), and then fed to the CNC machine that adopts a linear interpolator.

II. PROBLEM DEFINITION & METHODOLOGY

A. Problem Definition

In the present work a problem is defined in two parts as modelling of free form surfaces and generation of tool path. Both are the parts of Reverse Engineering (RE) and now a day's RE has been used as a tool for CAD/CAM applications. Basically the present work includes the following objectives:

1) To model a free form surface from given control points.
2) To generate tool paths for the surfaces developed.
3) To interpolate the control points for better accuracy in Bezier surface and its tool path.

Bezier surface has been considered for developing a complex free form surface and also for generating tool path.

B. Methodology

In the present work, we use MATLAB a programming language for generating surfaces and their tool paths which is also technical software and provides interactive tools for various engineering fields.

In general, when we start the MATLAB program, it displays the MATLAB desktop, a set of tools (graphical user interfaces or GUIs) for managing files, variables and applications associated with MATLAB.

1) Workspace in MATLAB

Firstly the control points have to be saved in the workspace of MATLAB. In workspace the files have to be prepared in which stores the value in matrix form and there are three files prepared in workspace as per the values of x, y, z coordinates. The Fig.2 shows workspace window.

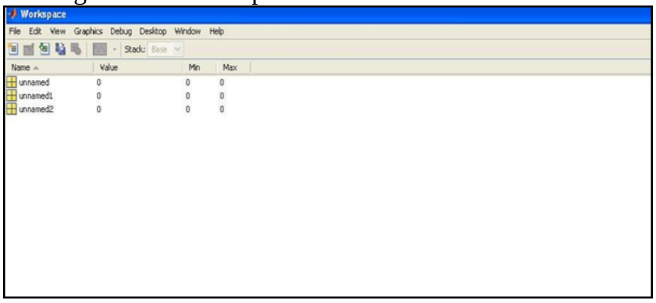

Fig. 2: Workspace in MATLAB

2) Algorithms of Surface Modelling & Tool Path

The Bezier surface generation algorithm has been discussed as below:

1) Define the function for Bezier surface.
2) Input the cutter contact points i.e. initial given control points in the form of x, y and z which has been saved in the workspace of MATLAB.
3) Create a patch with the help of control points x, y and z in matrix form such that patch in the form of [x y z 1].
4) Construct the control vertices X, Y and Z in the form of patch (m, n) in such a way that m is related to vertices X, Y and Z and it is 1, 2 and 3, and n is related to control points . The control points lie in the patch (m, n).
5) Calculate the Bezier boundary points P_X, P_Y and P_Z with the help of control vertices X, Y and Z.
6) Plot the Bezier boundary points P_X, P_Y and P_Z for various rows and columns, smoothening Bezier surface patch has been developed.

The Bezier surface tool path generation algorithm has been discussed as follows:

1) Define the function for Bezier surface.
2) Input the cutter contact points i.e. initial given control points in the form of x, y and z which has been saved in the workspace of MATLAB.
3) Create a patch with the help of control points x, y and z in matrix form such that patch in the form of [x y z 1].
4) Construct the control vertices X, Y and Z in the form of patch (m, n) in such a way that m is related to vertices X, Y and Z and it is 1, 2 and 3, and n is related to control points . The control points lie in the patch (m, n).
5) Calculate the Bezier boundary points P_X, P_Y and P_Z with the help of control vertices X, Y and Z.
6) The Bezier boundary points P_X, P_Y and P_Z are shaped using shaping algorithm shape it and generate the new variables x1, y1 and z1 with the help of grid data function and then generate the tool paths of Bezier surface with given tolerance.
7) The above tool path generated may vary according to the desired value of given tolerance.

3) Matrix Representation of Bezier Surface

The Bezier surface can be represented in matrix form as follows:

P (u, v) = [U][M][B][M]T[V]

where,

$$[U] = [u^n \quad u^{n-1} \ldots\ldots\ldots..1]$$

$$[V] = [v^m \quad v^{m-1} \ldots\ldots\ldots..1]$$

$$[M] = \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$[B] = \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix}$$

For cubic Bezier surface (n=3 & m =3), we have

P (u, v)

$$= [u^3 \quad u^2 \quad u \quad 1] \begin{bmatrix} 1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} B_{0,0} & B_{0,1} & B_{0,2} & B_{0,3} \\ B_{1,0} & B_{1,1} & B_{1,2} & B_{1,3} \\ B_{2,0} & B_{2,1} & B_{2,2} & B_{2,3} \\ B_{3,0} & B_{3,1} & B_{3,2} & B_{3,3} \end{bmatrix} \begin{bmatrix} v^3 \\ v^2 \\ v \\ 1 \end{bmatrix}$$

Note: It is necessary to represent the control points in matrix form. As it has been constructed the Bezier surface patch, patch must be in the matrix form since the cubic Bezier surface patch requires 16 control points in matrix form.

III. RESULTS & DISCUSSIONS

A. Bezier Surface Modelling

The control points taken in present work for complex Bezier surface patch are same as of Young-Keun Choi [5] and are given below as:

(0,0,1.5); (0,1,1.2); (0,2,1.2); (0,3,1.5);
(0.7,0,1.2); (0.7,1,0.7); (0.7,2,0.7), (0.7,3,1.2);
(1.4,0,1.2); (1.4,1,0.7); (1.4,2,0.7); (1.4,3,1.2);
(2,0,1.5); (2,1,1.2); (2,2,1.2); (2,3,1.5)

The program developed using MATLAB (version 9.0.0, R2016a) generates the complex Bezier surface patch as shown in Fig.3 given below:

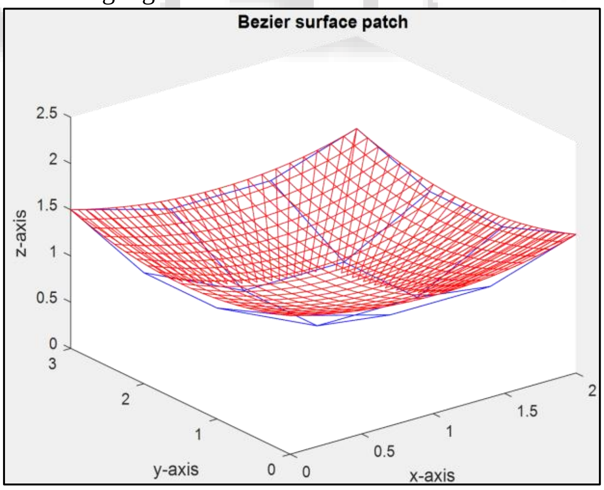

Fig. 3: Bezier Surface Patch

B. Bezier Surface Tool Path Generation

The tool path is generated for different tolerances using the Bezier boundary points P_X, P_Y and P_Z for given control points defined above of Bezier surface patch. This method is one of the simplest tool path generation algorithm in which the cutter contact points are specified as constant parametric points.

The program developed using MATLAB (version 9.0.0, R2016a) generates the tool path of complex Bezier surface patch for different tolerances as follows:

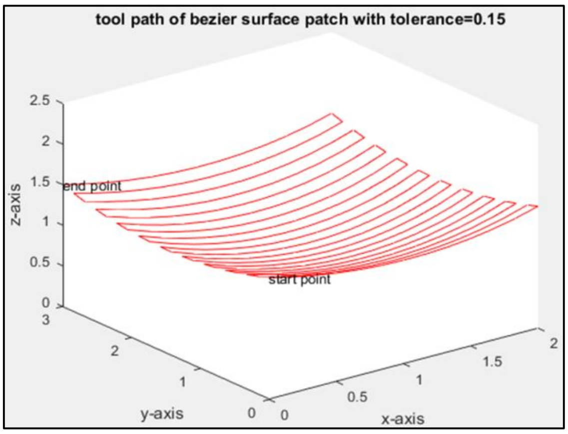

Fig. 4(a): Tool Path of Bezier Patch for Tolerance = 0.15

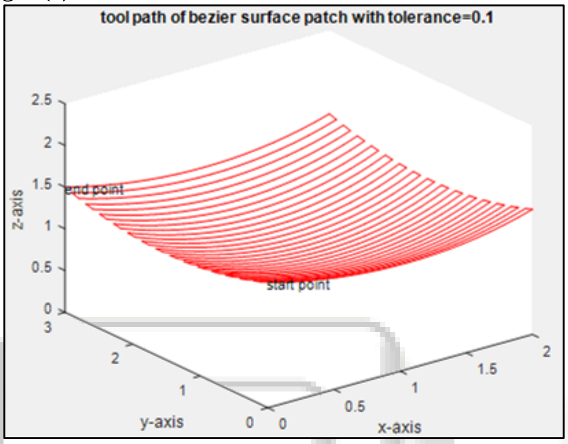

Fig. 4(b): Tool Path Generation of Bezier Patch for Tolerance = 0.1

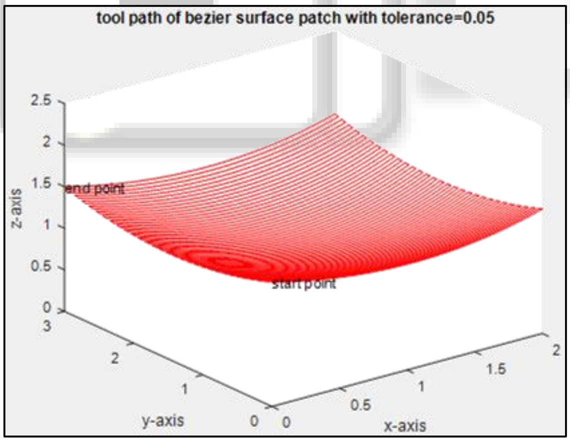

Fig. 4(c): Tool Path Generation of Bezier Patch for Tolerance = 0.05

From the above fig 4, it is cleared that decreasing the tolerance from 0.15 to 0.05 makes the tool path of Bezier surface patch smoother to its surface.

IV. CONCLUSIONS

Both surface modelling and tool path generation are the parts of reverse engineering and also an integral part for CAD/CAM applications now a day. Surface modelling has wide area of applications in reverse engineering through Computer Aided Design (CAD) while tool path generation is the most important aspect of Computer Aided Manufacturing (CAM) of intricate surface. In the present work, it should be concluded that modelling of complex Bezier patch has been successfully achieved through MATLAB programming and also tool path planning has been achieved through MATLAB for different tolerances.

REFERENCES

[1] William Welch and Andrew Witkin, “Variation Surface Modeling”, School of computer science Carnegie Mellon University Pitsburgh, PA 15213.

[2] Lizhuang Ma, Ounsheng Peng, “Smoothing of free from surfaces with Bezier patches”, Computer Aided Geometric Design, Vol.12, Issue 3, May 1995, pp. 231-249.

[3] J.P.Kruth, A.Kerstens, “Reverse engineering modelling of free form surfaces from point cloud surface to boundary conditions”, Journals of materials processing technology, Vol.76, Issue 1-3, April 1998, pp. 120-127.

[4] Falai Chen, Wenping Lou, “Degree reduction of internal Bezier curves”, Computer Aided Design, Vol.32, Issue 10, 1 September 2000, pp. 571-582.

[5] Young-Keun Choi, A.Bannerjee, “Tool path generation for free form surfaces using Bezier curves / surfaces”, Computer and industrial engineering, Vol. 52, Issue 4, May 2007, pp. 486-501.

[6] Loney, G.C., & Ozsoy, T.M., “Machining of free form surface”, Computer Aided Design, Vol.19, No.2, 1987, pp. 85-89.

[7] Broomhead, P., & Edkins. M, “Generating NC data at the machine tool for the manufacture of free form surface”, International Journal of Production Research, Vol.24, No.1, 1986, pp. 1-14.

[8] Bobrow, J.E., “Tool path generation by NC machines by the part representation of CSG”, Computer Aided Geometric Design, Vol.17, No.2, 1985, pp. 69-76.

[9] Huang.Y., & Oliver J., “ Tool path generation on sculptured surfaces using non constant parameter”, International Journal of Advanced Manufacturing Technology, Vol.9, No.2, 1994, pp. 281-290.

[10] Suresh.K, & Yang, D. C. H., “Constant scallop height machining of free form surfaces”, Journal of Engineering for Industry, Vol.116, 1994, pp. 253-259.

[11] Rong-Shine Lin (graduate student), Y. Koren (Professor) ASME fellow, “Efficient tool path planning for machining free form surfaces”, Journal of Engineering for Industry, Vol.118, February 1996, pp. 20-28.

[12] Chih Ching Lo, “A new approach to CNC tool path generation”, Computer Aided Design, Vol.30, No.8, 1998, pp. 649-655.

