Piecewise Arc-Length Parameterized NURBS Tool Paths Generation for 3-Axis CNC Machining of Accurate, Smooth Sculptured Surfaces

Maqsood Ahmed Khan

A Thesis

In the Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy at

Concordia University

Montreal Quebec, Canada

December 2010

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: **Maqsood Ahmed Khan**

Entitled: **Piecewise Arc-Length Parameterized NURBS Tool Paths Generation for 3-Axis CNC Machining of Accurate, Smooth Sculptured Surfaces**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. D. Qiu

_____ External Examiner
Dr. D. Xue

_____ External to Program
Dr. Z. Tian

_____ Examiner
Dr. M. Chen

_____ Examiner
Dr. A. Akgunduz

_____ Thesis Supervisor
Dr. Z. Chen

Approved by _____
Dr. W-F. Xie, Graduate Program Director

January 14, 2011

_____
Dr. Robin A.L. Drew, Dean
Faculty of Engineering & Computer Science

ABSTRACT

Piecewise Arc-Length Parameterized NURBS Tool Paths Generation for 3-Axis CNC Machining of Accurate, Smooth Sculptured Surfaces

Maqsood A. Khan, Ph.D.

Concordia University, 2010

In current industrial applications many engineering parts having complex shapes are designed using sculptured surfaces in CAD system. Due to the lack of smooth motions and accurate machining of these surfaces using standard linear and circular motions in conventional CNC machines, new commercial CNC systems are equipped with parametric curve interpolation function. However, in some applications these surfaces can be very complex that are susceptible to gouging and due to the approximation of; CL-path in CAM system and path parameter in real – time, high machining accuracy, smooth kinematic and feed-rate profiles, are difficult to achieve. This dissertation focuses on developing algorithms that generate tool paths in NURBS form for smooth, high speed and accurate sculptured surface machining. The first part of the research identifies and eliminates gouge cutter location (CL) point from the tool path. The proposed algorithm uses global optimization technique (Particle Swarm Optimization) to check all the CC-points along a tool-path with high accuracy, and only gouging free CC-points are used to generate the set of valid CL-points. Mathematical models have been developed and implemented to cover most of the cutter shapes, used in the industry.

In the second phase of the research, all valid CL-points along the tool-path are used to generate CL-path in B-spline form. The main contribution of this part is to formulate an error function of the offset approximation and to represent it in NURBS form to globally bound the approximation errors. Based on this error function, an algorithm is proposed to generate tool-paths in B-spline from with; globally controlled accuracy, fewer control points and low function degree, compared to its contemporaries. The proposed approach thus presents an error-bounded method for B-spline curve approximation to the ideal CL-path within the accuracy. This part of research has two components, one is for 2½- axis (pocket) and the other one is for 3-axis (surface) CNC machining.

The third part deals with the problem of CL-path parameter estimation during machining in real time. Once the gouging free CL-path in NURBS form with globally controlled accuracy is produced, it is re-parameterized with approximate arc-length in the off-line stage. The main features of this work are; (1) sampling points and calculating their approximate arc-lengths within error bound by decomposing the input path into Bezier curve segments, (2) fitting the NURBS curve with approximate arc-length parameter to the sample points until the path and parameterization errors are within the tolerance, and (3) segment the curve into pieces with different feed rates if during machining the cutter trajectory errors are beyond the tolerance at highly curved regions in the NURBS tool path.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and gratitude to my supervisor Prof. Chevy Chen for his invaluable guidance, support and enthusiasm throughout my PhD work. I have gained valuable research experience and significantly improved my technical skills under his supervision.

Thanks to my fellow graduate students, in past and present, for providing all sorts of help and creating an excellent working atmosphere. The assistance of the faculty and staff of the Mechanical and Industrial Engineering Department is also greatly appreciated.

I would also like to acknowledge the financial and moral support of NED University of Engineering and Technology, Karachi, Pakistan.

Finally, I dedicate this work to my family whose encouragement and help made this work possible.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Research Problems

Many products with complex shapes are designed with sculptured surfaces to enhance their aesthetic appeal for customer satisfaction, especially in the automobile and consumer electronics industries. Other products have sculptured shapes to meet functional requirements, such as; turbine blades, plastic injection molds and sheet metal dies. Usually, for these products, several surface patches are required to be connected (compound surface) to generate the desired shape. Mathematically geometric objects can be represented in three forms; *parametric*, *implicit* and *explicit*. Implicit and explicit forms are often referred as *nonparametric* forms. Usually, freeform or sculptured shapes are represented in parametric forms like Bezier, B-spline or non-uniform rational B-spline (NURBS) [1]. Among different representations, NURBS is a general form that includes non-rational B-splines, and rational/non-rational Bezier curves and surfaces.

1.1.1 NURBS curve

Mathematically a *p*th-degree NURBS curve is defined by [1]

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \qquad 0 \le u \le 1 \tag{1.1}$$

$\mathbf{P}_i$ denotes $(n+1)$ control points (forming a control polygon); $N_{i,p}(u)$ are the B-spline basis functions; p is the degree of $N_{i,p}(u)$; w_i are the corresponding weights of control points. Basis functions can be defined by a recursion equation:

$$N_{i,p}(u) = \begin{cases} 1 & \text{if } u_i \le u \le u_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad (p = 0)$$
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (1.2)$$

defined over the strictly non decreasing clamped knot vector:

$$U = \left\{ \underbrace{a,\ldots,a}_{p+1}, u_{p+1}, \ldots\ldots, u_{m-p-1}, \underbrace{b,\ldots,b}_{p+1} \right\}$$

There are $(m+1)$ elements of the vector, called knots where, $(m = n + p + 1)$. Polygonal curve defined by the control points is called control polygon. Following are some relevant properties of NURBS curves.

1. $\mathbf{C}(0) = \mathbf{P}_1$ and $\mathbf{C}(1) = \mathbf{P}_n$
2. Strong convex hull property: if $u \in [u_i, u_{i+1})$, then $\mathbf{C}(u)$ lies within the convex hull of the control points $\mathbf{P}_{i-p}, \ldots, \mathbf{P}_i$.
3. A NURBS curve with no interior knots is a rational Bezier curve. Furthermore if all weights w_i are equal to one, then NURBS curve contain non-rational B-spline and rational and non-rational Bezier curves as special cases.

4. Local approximation: If the control $\mathbf{P}_i$ is moved, or the weights w_i is changed, it affects only that portion of the curve on the interval $u \in \left[u_i, u_{i+p+1}\right)$

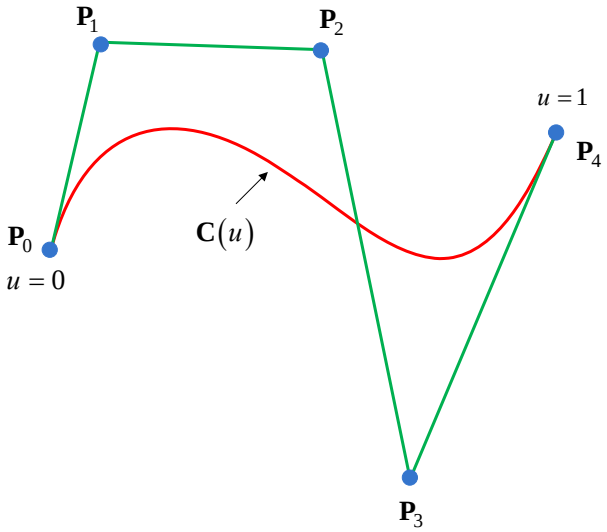


Figure 1.1 A cubic NURBS curve and its control polygon.

1.1.2 NURBS surface

A NURBS surface of degree p in the u direction and degree q in the v direction is a bivariate vector-valued piecewise rational function of the form

$$\mathbf{S}(u) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j} \mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) w_{i,j}} \qquad 0 \le u, v \le 1 \tag{1.3}$$

The $\{\mathbf{P}_{i,j}\}$ form a bidirectional control net (control polyhedron); the $\{w_{i,j}\}$ are the weights; and the $\{N_{i,p}(u)\}$ and $\{N_{j,q}(v)\}$ are the non-rational B-spline basis functions defined on the knot vectors

$$U = \left\{ \underbrace{0,\ldots,0}_{p+1}, u_{p+1}, \ldots\ldots, u_{m-p-1}, \underbrace{1,\ldots,1}_{p+1} \right\}$$

$$V = \left\{ \underbrace{0,\ldots,0}_{q+1}, u_{q+1}, \ldots\ldots, u_{s-q-1}, \underbrace{1,\ldots,1}_{q+1} \right\}$$

Where $r = n + p + 1$ and $s = m + q + 1$.

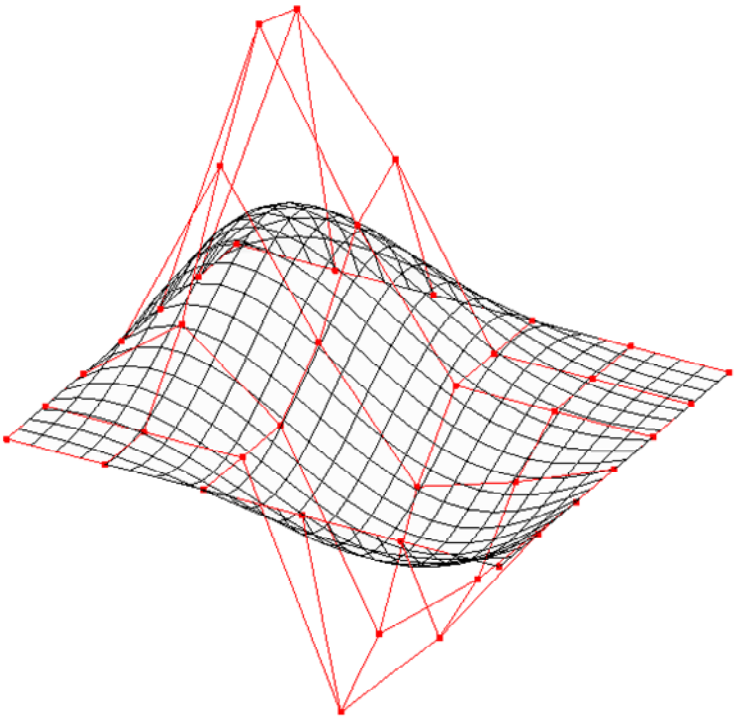


Figure 1.2 A NURBS surface and its control net.

1.1.3 Production of sculptured parts

While the sculptured surfaces are created using CAGD techniques, it is the role of manufacturing engineers to realize them in physical form. As an ever-increasing variety of products are being designed with sculptured surfaces, efficient machining of these surfaces is becoming increasingly important in many areas of

manufacturing. In order to machine sculptured surfaces by a cutting tool on a CNC milling machine, it is necessary to generate a series of points called cutter location points, and the trajectory of these is termed as NC tool path. The point of contact between the cutter and surface is denoted by CC (cutter contact) point while the corresponding reference point on the cutter is defined by CL (cutter location) point (Figure 1.3).

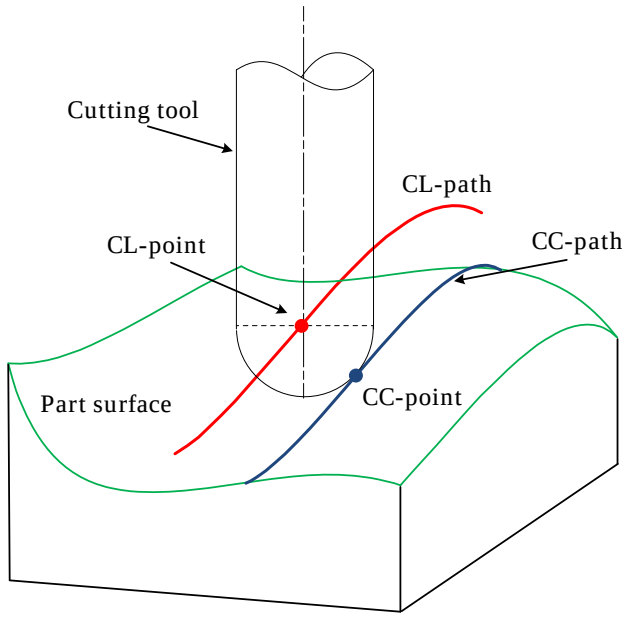


Figure 1.3 Illustration of CC-path and the corresponding CL-path.

For producing free form (sculptured surface) parts on a CNC machine following three phases are required [2]:

1. **Process planning phase:** machining-tolerance, cutter specification, tool-path topology and milling strategy (up/down milling and upward/downward milling).
2. **Tool-path generation phase:** tool-path planning and CL-data computation.

3. **Validation phase:** gouging detection and cutting simulation.

In tool-path generation phase the first step (tool-path planning), is to determine the values of step-length and path-interval based on the information from process planning phase and machining-tolerance requirements. The output of this step is a set of CC-points from the part surface, and then in the next step each CC-point is converted to a CL-point along with cutting conditions (federate and spindle speed). Once the CL-path is computed, in validation stage the gouging and interference is detected and eliminated and finally the cutting simulations are conducted to verify the final set of CL-paths.

1.1.4 NURBS tool path

For sculptured surface machining, conventional CNC machines provide only linear and circular arc interpolators. Therefore, CAD/CAM systems must segment the tool path into huge number of small line segments and transmit them to CNC machines [3]. However this approach causes several problems like; large NC data file, increased machining and data transmission time, and surface inaccuracies [4]. Furthermore, the modern high speed machining requires federates up to 40m/min with accelerations up to 2g [5] and the cutter needs to accelerate/decelerate at the beginning and end of each line segment which results in chatter and vibration. In addition to this, the real cutting speed is also lower than input speed; therefore the cutting time is prolonged.

Owing to the above reasons, the curve interpolator has recently been accepted as an alternative to replace these conventional linear and circular

interpolators. Since, they are potentially more suitable for high speed and high accuracy machining with smooth feed rate profiles; some of the CNC controller manufacturers (FANUC, SIEMENS, HEIDENHAIN and MITSUBISHI) have developed this curve interpolation function. Shiuh et al. [6] conducted real machining experiments and found that machining time was reduced about 23% when NURBS tool-path was used instead of conventional linear segments.

Among the techniques used for representing parametric curves and surfaces, the Non-Uniform Rational B-spline (NURBS) has several advantages [7]. Since late 1980s, NURBS has emerged as the standard free-form geometry representation tool because it is a superset of Bezier and B-spline, and can represent quadric curves and surfaces exactly. In Piegl's renowned survey paper [8], he pointed out that NURBS offers a common mathematical form for representing and designing both standard analytical shapes and free-form curves (surfaces). By manipulating the values of weights, knot vector and control points, a wide variety of shapes can be produced using NURBS.

In NC machining, when NURBS data is used to interpolate movements for the machine tool, it results in much finer and smoother NC movements. NURBS Interpolation can reduce cycle time by up to 30% in the manufacture of highly complex, curved dies and molds, while providing higher accuracy and improved surface finish. According to *GE Fanuc*, its NURBS Interpolation function results in a reduction of program size to 1/10th to 1/100th that of a comparable linear interpolation part program and significantly improves the fundamental accuracy issues. "NURBS program allows for faster machining" (*Makino*) [9].

Due to the superior geometric and manufacturing features of NURBS representation as compare to the other forms we have adopted NURBS curve to represent the final CL-path. The general approach for 3-axis CNC machining of sculptured parts using NURBS CL-path includes the following

1. 3D geometric model is generated in a CAD module
2. In CAM module the CC-points are generated for the given cutter size and shape, machining tolerance and geometric features of the part surface.
3. Gouging and interference is detected and eliminated for all CC-point.
4. The set of valid CC-points is used to generate the CL-points, based on the cutter and surface normal informations.
5. NURBS curve is fit to all CL-points within the specified curve fitting tolerance along a tool-path.
6. The CL-path in NURBS form is post processed for a particular CNC machine.
7. After post processing the NURBS tool-path in ISO codes is fed to the CNC machine controller.
8. The NURBS interpolator within the CNC controller generates reference positions along the path for the motion control (servo control) system to machine the part.

1.1.5 Error sources in NURBS machining of sculptured parts

Inaccuracies in CNC machining can be traced to several major sources like; cutting tool deflection, cutting tool wear, machine tool vibration, and improper coolant/lubricant. To improve part accuracy, much research has been done to

predict cutting forces and machining errors in different cutting condition. However, in milling of sculptured surface parts, due to their curved shapes, the geometries of the cutting tools do not match the part surface well. As a consequence, machining error is inevitable, even if there is no other source of errors in ideal machining conditions. Sometimes, geometric errors caused by this tool-surface mismatch along the tool-path are more harmful and result in surface gouging.

As mentioned before, the geometric model of the part with sculptured surface is designed in CAD system, and then corresponding tool paths for CNC machining are automatically generated in CAM module. Since the sculptured surfaces are complex and due to the tool-surface mismatch, close-form equations of the ideal tool paths for the surfaces do not exist. Therefore, discrete CC- points need to be sampled based on chordal deviation tolerance to approximate the ideal CC-path. Basically, this tolerance is a measure of how accurately the line segments joining the sampled CC-points approximate the shape of the ideal path (Figure 1.4). Smaller tolerances result in a path that follow the original shape more accurately, but often result in a larger G-code file.

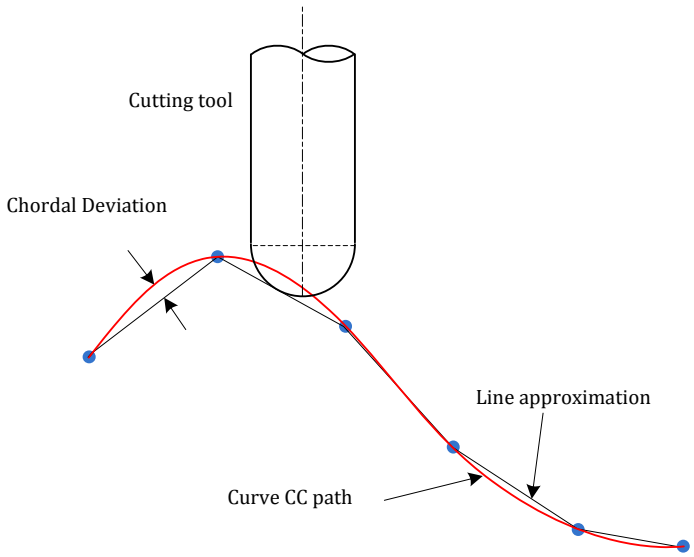


Figure 1.4 Illustration of Chordal deviation.

The CC-points satisfying the chordal deviation tolerance may still damage the part surface because of the surface gouging. Gouging occurs when a cutting tool, contacting at a CC-point, interfere with other portions of the part surface more than tolerance value (Figure 1.5). Gouging damages the part surface and can break the cutter also, it is therefore necessary to detect and remove all CC-points which cause gouging. Most of the existing methods only focus on point sampling on the grid (therefore accuracy depends on grid density), and limited to only ball-end mill cutter. Normally, extreme curvature values are compared with cutter radius to detect gouging. It should be noted that curvature is a local geometric property and it cannot avoid global gouging around the cutting tool tip. Sculptured surface parts such as stamping dies and injection molds are designed with compound surfaces because of shape complexity. During machining of these surfaces, while removing material around a CC point, the tool may interfere with another point (gouging) on the neighboring surface patches. Detection and elimination of gouging from

compound surfaces is more challenging especially for cutters other than ball end mill. Another problem is faced, while machining a sculptured surface the tool body and/or the tool holder may also collide with the surface in addition to the bottom cutting edge of the cutter. Therefore, the gouging detection algorithm must (1) be able to detect and eliminate gouging from the tool path for compound surface machining with high accuracy and efficiency, and (2) consider the whole tooling system instead of only the bottom cutting edge of the tool (which has been the focus of many researchers). In three-axis machining ball end mills are more popular than flat and round end mills. It is easy to locate a ball end mill on a surface due to its simple geometric shape. The cusp (remaining material) produced on the surface after machining is also uniform; therefore the grinding or polishing efforts are reduced in the next operation. However, flat and round end mills offer higher cutting efficiency than ball end mill, because parts can always be cut by the periphery of these cutters at the maximum cutting efficiency. Therefore the algorithm should be applicable for different cutting tool shapes.

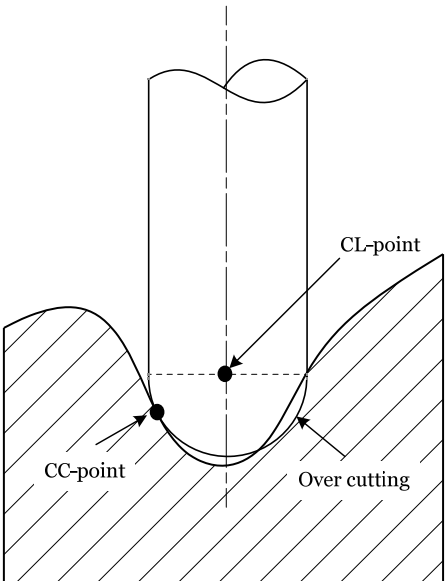
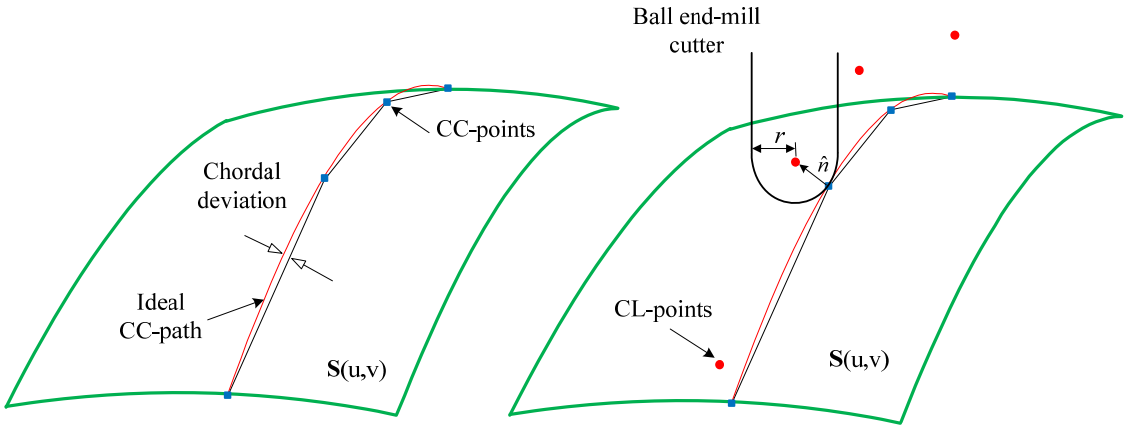


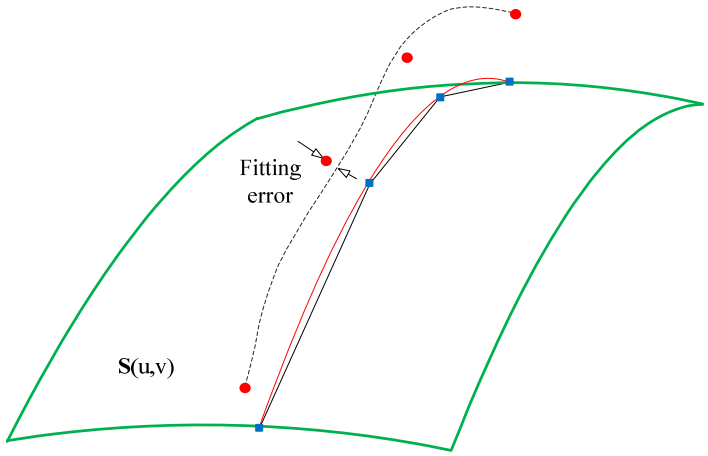
Figure 1.5 Illustration of gouging.

Once all the CC-points have been evaluated and corrected for gouging, they are converted to CL-points, and used to generate NURBS CL-path. For deep understanding of NURBS tool path generation process, a mathematical problem can be formulated as follows. Suppose that we want to cut surface $\mathbf{S}(u,v)$ using ball end mill cutter along the CC-path as shown in Figure 1.6(a). Since, the ideal CC-path is unknown a set of CC-points is sampled along the curve based on chordal deviation tolerance. Using the surface normal $\hat{n}$ at CC-points and cutter radius r, the corresponding set of CL-points is generated (Figure 1.6(b)). Fit a NURBS curve to all CL-points to approximate the ideal CL-path (Figure 1.6(c)). To evaluate the quality of the CL-path, compute the perpendicular distance to calculate the approximation error by projecting each CL-point on the curve. The approximated CL-path is said to be acceptable if the maximum approximation error is within the fitting tolerance. Sampling of CL-points needs one tolerance to respect allowable chordal deviation

and the generation of NURBS CL-path also needs curve fitting tolerance. However, this double-tolerance approach only makes sure that the approximated CL-curve respects the ideal tool-path at discrete CL-points. But, *it is possible that the approximated CL-path deviates from the ideal path more than allowable tolerance at locations other than discrete CL-points*. If it is happened, the usage of this CL-path for machining will damage the part surface.



(a) Ideal CC-path and sampled CC-points (b) generate set of CL-points



(c) Fit NURBS curve using CL-points to approximate ideal CL-path

Figure 1.6 CL-path generation process

To avoid this problem, it is essential to make sure that the approximated curve respects the ideal path along its whole length, instead of at only discrete points. There are some existing methods which guarantee the global control over the approximation error, but they need high curve degree, and large number of control points to generate the offset curves (CL-paths). Furthermore, the CL-paths generated by these methods are only planar curve, which can only be used for pocket machining. Therefore, in the second part of the research an algorithm is proposed to generate CL-path in NURBS form; (1) which approximates the ideal CL-path within the given tolerance along the whole curve length i.e. the error is globally bounded, (2) with flexibility to choose curve degree, (3) needs minimum number of control points to represent the curve, and (4) applicable for both 2½- axis (pocket) and 3-axis sculptured surface machining.

Once the CL-path is generated, it is ready to be send to the CNC machine for producing the sculptured part. Usually the free-form CL-path $\mathbf{C}(u) = \begin{bmatrix} x(u) & y(u) & z(u) \end{bmatrix}^T$ is represented via unit less parameter (u-parameter), which needs to be converted into time domain during machining. Tool positions in terms of time instead of u-parameter are called reference commands and become the input of position servo control in CNC machine. This process of command generation for curves is called parametric interpolation and is performed in the CNC controller during machining. Let V and T_s are the required machining feed-rate and sampling time of servo control respectively. The linear distance travelled by cutter along the tool path during time T_s is VT_s . The job of parametric interpolator is to generate a point on

$\mathbf{C}(u)$ by incrementing u within each interpolation period T_s. In simplest way, this increment can be a uniform value i.e. $\triangle u$ =constant. But, unfortunately uniform increment in parametric space does not correspond to uniform length of cutter movement in machining space due to the non-linearity between a curve and its parameter. Consequently, the travelling distance in each interpolation period will be different during the same T_s, which means values of actual feed-rate are different from the commanded feed-rate, resulting in feed rate fluctuations, which produces surface finish variations and unnecessary longer machining time.

Since, cutter moves in straight path between two consecutive interpolated points, the chordal error is unavoidable. Another problem of NURBS interpolation is that it is difficult to get analytical solutions for the chordal error between the interpolated curve and the ideal curve. If the extreme variations in curve derivatives among the knot spans and/or high curvature values are encountered, existing interpolators may produce cutter trajectory errors beyond the tolerance. Although, NC program cannot change the way the controller moves the cutter, the CL-path can be changed in such a way that these errors are minimized or at least decreased. Therefore, the last part of the research deals with the re parameterization of the CL-path, so that the real time parametric interpolator will be able to compute an appropriate value of parametric increment for each interpolation cycle, which, in turn, guarantees smooth feed rate profiles and cutter trajectory errors within the allowable limits.

1.2 Research Objectives

The first objective of this research is to develop efficient, and accurate generalized gouging detection algorithms for compound sculptured surface machining on 3-axis CNC milling. To achieve this objective, it is essential to develop a technique to check all CC-points accurately with high searching efficiency. One purpose of this part is to make the algorithm as generic as possible so that, the methodology can be applied to different cutting tool shapes (flat end mill to form milling cutters) and complex part surfaces (simple to compound surfaces).

After gouging detection and elimination from CL-points, the next main objective is to develop a methodology for generating smooth CL-path in NURBS form with fewer control points, lower base function degree, and higher geometric accuracy. To get this objective following tasks are required:

(1) Formulate a mathematical function to globally bound the approximation error.

(2) For global error control, it is necessary to know the maximum error (upper bound). Therefore, the error function needs to be converted into scalar NURBS form, and based on convex hull property the maximum coefficient of the NURBS curve can be used to estimate the maximum error.

(3) An algorithm is also required, to adjust the approximated CL-path at proper locations according to the positions of maximum coefficient of the NURBS error function, and finally produce the desired tool-path having

less function degree, fewer control points and guaranteed to be within the allowable approximation error.

Once the general NURBS CL-path with unit less parameter ($u \in [0\ 1]$) is developed the last objective is to convert it into piecewise NURBS tool paths with the arc length parameter ($s \in [0\ L]$). It is generated off-line by taking the NURBS interpolation into consideration. This type of tool path is essential for high speed and accuracy machining along the NURBS CL-paths with high curvatures. To generate this path

(1) A methodology is required to sample the cutter location points, estimation of their arc-lengths and fit a NURBS curve to these CL-points.
(2) The path and parameterization errors need to be within the prescribed tolerances.
(3) For accurate tool trajectory and consistent tool velocity in high speed machining the highly curved arc-length parameterized CL-path needs to broken into segments to generate piecewise NURBS tool path with different feed rates for different pieces.

Overall, the objective of this research is to develop algorithms for generating NURBS tool paths if

(1) The design surface has extreme curvature values and it needs to be machined with high accuracy and good surface finish

(2) The machining speed is very high and it is required to keep consistent feed rate profile to reduce machine vibrations.

1.3 Dissertation Organization

The remaining sections of this dissertation are organized as follows. Chapter 2 reviews the current technologies of gouging detection and elimination, NURBS NC tool-path for pocket machining and three axis surface machining and real time CNC interpolator. Chapter 3 presents a generalized optimization based gouging detection and elimination technique for three axis machining using different shapes of cutters.

Several examples are used to verify the correctness and effectiveness of the proposed method. Chapter 4 presents a new method to approximate 2D free form curve offsets for B-Spline NC tool-paths with offset error globally bounded. Simulation results are presented to show that our method is better than existing methods in terms of number of control points and order of the curve for a given tolerance value. Chapter 5 develops an integrated approach to generate cutter location path in B-Spline form for three axis surface machining with high accuracy. Finally chapter 6 introduces a new method to reparameterize the cutter location path with approximate arc length parameter. Chapter 7 contains the summary of this work.

Chapter 2 Literature Review

This chapter reviews the state-of-the-art technologies of gouging detection and elimination, 2D curve offsetting, 3D NURBS tool-paths and real-time CNC interpolation. The limitations of the current methods are also discussed.

2.1 Gouging Detection and Elimination

In surface machining, gouging and interference hinder the surface quality and production efficiency. To overcome these problems, extensive research has been conducted in this topic. However, there is no effective solution to them so far. Conceptually, there are two approaches by which gouging can be avoided. The first is a conservative approach in which a smaller cutter size is selected to cut the whole surface without gouging, but it results in low machining efficiency. In the second approach the machining efficiency is increased by selecting large cutter size for only those surface regions which are gouging free and then the remaining surface is machined with smaller cutter.

Choi and Jun [10] studied interference detection for CC-point tool path generation approach. First the CC points were converted into a triangular mesh and interference was detected by comparing each CC-point with its neighbor CC-points. Wenliang et al. [11] also used discrete definition of the surface using triangles to

detect gouging for ball end-mill cutter. Yang et al [12] found the optimal ball end-mill cutter size for gouging free machining of free form surface represented by point cloud data. Zhou [13] Checked gouging by evaluating the intersection of vertical lines (from the grid on xy-plan) with the part surface and the ball end mill cutter. Oliver et al. [14] termed the overcutting because of true chordal deviation as local gouging, for correcting this type of gouging they calculated true chordal deviation. If the local radius of curvature is less than the ball end mill radius, then this location will be gouged and it is termed as global gouging. George et al. [15] used surface offsets to detect interference, and checked for the self intersection curve. The tool path is then planned over the CC surface after removing the CC data that lies inside the self intersection curve. The key problems in this approach are to find offset surface first and then to identify its self intersections. This method is only applicable for ball-end mill cutter. Hatana et al. [16] isolated the interfering regions of the surface by comparing the radius of curvature at discrete points on the grid with cutter radius, Ding X. M. et al. [17] also used grid of points to identify interference regions in concave and saddle surfaces by comparing extreme curvature values with cutter. Yang et al. [18] proposed a methodology for local interference detection, by finding the total interference area (TIA) using extreme curvature values. Global interference is detected by checking the collision of the cutter with surface during its movement. All of these research studies, however, only focused on point sampling on the grid, and limited to ball-end mill cutter. Beside this curvature is a local geometric property and it cannot avoid global gouging around the cutting tool tip.

These methods cannot detect local and global gouging for compound continuous sculptured surface machining while using different cutter shapes with high accuracy. To incorporate different cutter shapes, Z.C. Chen and G. Liu [19] proposed a mathematical model (imaginary cutter size model) for a compound surface to determine the largest allowable cutter size using particle swarm optimization (PSO) method. But this method requires large computational time because it randomly searches the whole compound surface for a given CC-point.

Finding gouge locations for complex surface machining is a global optimization problem. Among several heuristic tools (genetic algorithm, ant colony, differential evolution) particle swam optimization (PSO) method, first introduced by Kennedy and Eberhart [20] is becoming more popular due to its simplicity and ability to quickly converge to a solution [21].

2.2 NURBS NC Tool Path

After gouging detection and elimination, all valid cutter location (CL) points are transferred to the CNC machine tool in the form of NC tool-paths. However, in case of parametric curve (NURBS, B-spline, Bezier, quintic splines etc) interpolation, these CL points need to be represented by curves. For NURBS machining, NURBS curves are used to approximate this CL-data. For pocket machining, planer curves which are generally the offsets to the design pocket profile are generated (Figure 2.1(a)). In sculptured surface machining 3D space curves are used to approximate the CL-points Figure 2.1(b).

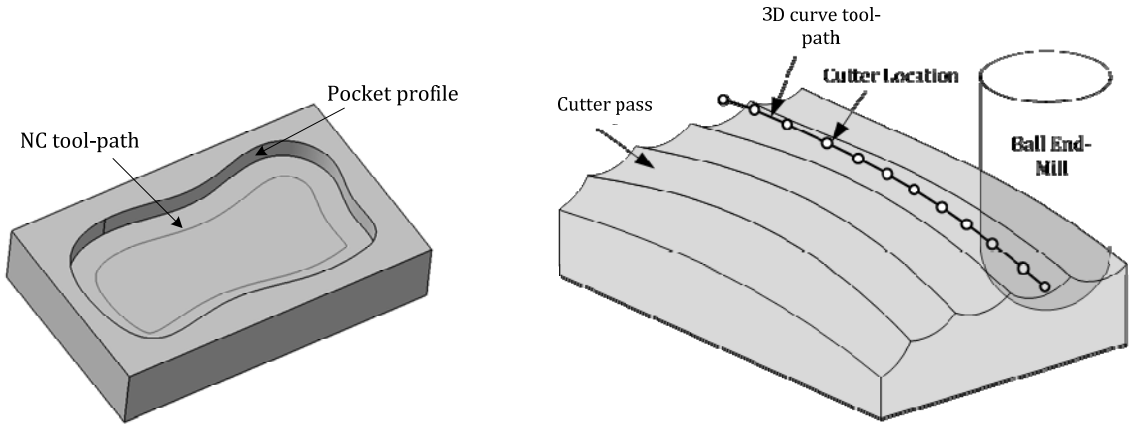


Figure 2.1 (a) Pocket profile to be machined and its corresponding planer B-spline offset as NC tool-path (b) surface machining along 3D B-spline curve tool-path.

2.2.1 Planar offsets as tool-paths for pocketing

In the past decade, besides the exact solutions (Farouki and Sakkalis [28]) to the offsets of special geometries, such as the analytical and the Pythagorean-hodographs (PH) curves, several researchers [24-27, 29-31] have proposed a number of methods of finding the offset of free-form curve approximately. The common way of evaluating the approximate offset accuracy in these methods is to calculate the deviations of the approximate from the exact offsets at the discrete sample points. But these deviations cannot demonstrate the overall maximum approximation error [35]. If the individual deviations are not less than the specified tolerance, the curve is subdivided at the midpoint regardless the actual maximum error location.

Some researchers [32, 33, 35-37, 39, 40] have rendered methods of approximating offsets with the errors globally bounded. To estimated the overall deviations between the approximate offset $\mathbf{C}_d^a(u)$ and the exact offset $\mathbf{C}_d(u)$ of a source curve $\mathbf{C}(u)$, Elber and Cohen [32] established a function $\varepsilon(u)=\left\|\mathbf{C}_d^a(u)-\mathbf{C}(u)\right\|^2-d^2$ (d is the offset distance) to represent the deviations. Then, this function $\varepsilon(u)$ is represented with a scalar NURBS curve. Based on a property of the NURBS curve, its largest coefficient is greater than the maximum value of $\varepsilon(u)$. The curve locations with unsatisfactory offsetting results are identified and new knots are simultaneously inserted, until the global error is within the allowable tolerance. Reference [35] mentioned the underestimation problem of $\varepsilon(u)$. To remedy this problem, [32] proposed angle function also (between $\delta(u)=\mathbf{C}_d^a(u)-\mathbf{C}(u)$ and unit normal vector $N(u)$) in addition to the difference function $\varepsilon(u)$.

Lee, et al. [35] adopted a different approach based on unit circle approximation to bound the maximum offset error. Let $\mathbf{U}(s)$ denotes a unit circular arc such that $\mathbf{C}'(u)\|\mathbf{U}'(s(u))$.Then the exact offset curve can be computed as $\mathbf{C}_d(u)=\mathbf{C}(u)+d\cdot\mathbf{U}(s(u))$. They approximated $\mathbf{U}(s)$ with a quadratic polynomial curve segment $\mathbf{Q}(s)$ within the tolerance ε, and using the parallel constraint $\mathbf{C}'(u)\|\mathbf{Q}'(s(u))$, they achieved the unique solution of parameterization $s(u)$, which provides the approximated offset curve, bounded by $d.\varepsilon$, as follows

$$\mathbf{C}_d^a(u) = \mathbf{C}(u) + d \cdot \mathbf{Q}(s(u))$$

$\mathbf{C}_d^a(u)$ is a planar rational curve of degree $3p-2$, where p is the degree of $\mathbf{C}(u)$. In the work of [36], Lee, et al. approximated the re-parameterization $s(u)$ and represented $\mathbf{U}(s)$ exactly using a quadratic rational polynomial curve. In this method, the only error source is from the approximation of $s(u)$, which violates the parallel constraint $\mathbf{C}'(u) \| \mathbf{Q}'(s(u))$. In [37] they proposed 3 re-parameterization based offset approximation methods, LRC, TMC, SRC and this time the offset approximation error is bounded by $\varepsilon \approx d.\sin\theta$, where θ is the angle between the normal vector $\mathbf{N}(u)$ and the difference vector $\mathbf{C}_d^a(u) - \mathbf{C}(u)$. In 2007, Zhao et al. [41] improved the error bound from $d.\sin\theta$ to $d.\sin^2\theta$ in re-parameterization based approaches, for curve offsetting. They calculated the Hausdorff distance to estimate the approximation error, instead of the angular deviation θ.

2.2.2 NURBS tool-paths for surface machining

For generating curve CL-paths off-line, generally there are two approaches. In the first approach cubic and quintic splines are used to interpolate the CL-points with near arc length parameterization [42-44, 47, 48]. In the second approach the CL-points are interpolated or approximated by NURBS or B-spline curves [49-54] with unit less parameter. The main weakness of both approaches is the accuracy of CL-path can only be guaranteed at discrete CL-points. The CL-paths from first approach do not need any parameter approximation in real-time. In the second

approach, because of the discrepancy between path parameter and path length increments in real-time the parameter needs to be approximated. Therefore several interpolation techniques [66-72] have also been developed to adjust path parameter in real-time for non-arc-length-parameterized curve CL-paths.

Cubic splines have inherent problem of parameterization, if spline parameter is based on chord length between two consecutive sample points, and even if they are parameterized with arc-length, the performance is not better than quintic spline [42]. In quintic spline interpolation a set of CL points, say $\mathbf{P}_0, \mathbf{P}_1, \ldots, \mathbf{P}_n$ are fit to a set of (n-1) composite quintic spline with C2-continuity condition at the junctions of spline segments. Since only CL points are known, so the first and second derivatives at segment junctions are approximated using chord lengths. To reduce the contour and feed-rate errors, all spline segments should be parameterized with arc-length instead of chord length. To solve this problem [42] and [43] fit cubic spline with chord length parameterization to get unit tangent and second derivative at data points and approximated the arc-length between two consecutive data points by solving fourth degree polynomial. But their methods can only guarantee arc length parameterization at both ends and middle point of the quintic spline segment, which is termed as nearly arc-length parameterized quintic spline. Wang, et al. [44] improved above discussed method, by considering more locations, deviating from arc-length characteristic between two data points. Auxiliary data points are added and the quality of parameterization is checked, this iterative process is continued until the arc-length parameterization is achieved within some tolerance value. Again

in [47] Wang, et al. used same approach but added C3-continuity, and the spline is termed as approximately arc-length parameterized C3-quintic interpolator spline. Erkorkmaz, et al. [48] proposed two approaches for quintic spline interpolation. In the first part, derivatives at data points and arc lengths are estimated using optimization with an objective to improve arc length parameterization, instead of using C2-cubic spline for derivatives. But this may produce a geometrically different tool path from the original shape.

In the second approach Taosheng, et al. [49] approximated the cutter location path by interpolating the discrete CL points using B-spline segment. Their method is limited to only isoperimetric tool-path generation method. The largest deviation distance between the cutter location path and the corresponding isoperimetric path is calculated numerically at different parameter values. If the deviation is larger than tolerance then the tool path is divided into segments. This process is continued until the deviation is within the tolerance value.

Method proposed in [50] is limited to iso-planar tool path generation strategy and based on checking fitting errors at each mid-span of the two consecutive CL points. First, the sample points are generated by computing the intersection of offset surface and the tool driving plane (plane parallel to the axis of tool). These points are interpolated by a planar cubic B-spline curve. The accuracy of B-spline tool-path is evaluated and more sample points are added if the error is higher than the allowable fitting tolerance.

Yau and Kuo [51] proposed a post-processing approach to converting G1 NC tool-paths generated with the CAD/CAM systems to NURBS paths for high speed contour machining. Langeron, et al. [52] developed an approach to generating 5-axis NURBS tool-paths in the part coordinate system and introduced an interpolation format to ensure continuous evolution of the rotation axis speed. Chen and Yang [53, 54] improved the existent fitting method with rough and fine fitting to a group of sample cutter locations. Bey, et al. [55] generated NURBS tool-paths with the parallel-plane strategy by using three methods of computing given points' parameters in order to find paths with the minimum number of control points. Lin, et al. [56] converted G1/G2 paths of a NC program into NURBS paths in two steps: (1) dividing the CLs according to their geometric shape into several groups, and (2) fitting a NURBS tool-path to each group of CLs in a iterative process of reducing the deviations between the sample CLs and the path and the chord errors within the prescribed tolerances. Similarly, Li et al. [57] designed a NURBS pre-interpolator to convert G1/G2 tool-paths into NURBS tool-paths. Shih and Chuang [58] constructed a one-sided polygon to approximate a 2-D NURBS part profile, offset the polygon to establish a unilateral tolerance zone, and generated a C^1-continuous and interference-free planar NURBS tool-paths to cut the profile. Lai [59] optimized the 2-D offsets of planar NURBS curves, which are NURBS tool-paths, by introducing an evaluating bound error and a radiating web-like searching path. In all, these methods cannot generate high quality 3-D NURBS tool-paths for high speed-and-accuracy NURBS machining of sculptured surfaces. Now, it is important to understand that 3-D NURBS tool-paths should be spatial arc-length parameterized

curves that are gouging-and-interference free and approximate theoretical tool-paths within a prescribed tolerance.

The NURBS fitting technique is useful to NURBS tool-path generation, though it is not all. A number of articles of this topic are reviewed. Rogers and Fog [60] developed a technique for constrained B-spline fitting by modifying the parameter values of a group of given points in iterations. Weiyin and Kruth [61] presented a simple method to assign parameter values to randomly measured points for the least squares fitting of B-spline curves. Park et al. [62] proposed a method of repetitively fitting input curves on a common knot vector in order to have them evenly transited. Borges and Pastva [63] applied Gauss-Newton method to minimize the fitting error of a single Bezier segment to a set of ordered points and directly evaluated the Jacobian by implicitly differentiating a pseudo-inverse. Park and Lee [64] selected a number of dominant points among a group of given points, determined the knots using the dominant point parameters, and fit a B-spline with lease squares minimization. Brujic et al. [65] precisely defined the sparsity structures of the relevant matrices and fully exploited the structures to reduce computation time and storage.

Since the CL paths generated by using second approach are not parameterized by their arc lengths therefore during machining the appropriate path parameter value is approximated. Initially tool path in parametric form is represented in unit less form (u-parameter), which needs to be converted into time domain. Tool positions in terms of time instead of u-parameter are called reference

commands and become the input of position servo control in CNC machine. This process of command generation for curves is called parametric interpolation and is performed in the CNC controller during machining. Different strategies have been adopted in the development of parametric interpolators [66-72], and will be discussed in the next section.

2.3 Real Time CNC Interpolation

It was mentioned before that to cut smooth sculptured surfaces accurately and efficiently in CNC machining, NURBS (non-uniform rational B-spline) interpolation has recently been developed in some CNC controllers. As a reference, a pre-generated NURBS path of cutter location is fed into the controllers, and then the NURBS interpolator calculates in real-time cutter locations along the path to instantaneously execute finite motions of the tool NURBS trajectory in machining. Current research on NURBS machining includes two types: (1) NURBS tool paths generation prior to machining and (2) real-time NURBS interpolation for NURBS tool trajectory in CNC machining. Type 1 research is to generate accurate NURBS paths from geometric perspective, and type 2 is to develop real-time NURBS interpolation algorithms of CNC controllers to make sure that (a) the deviation between the tool trajectory and the path is less than the prescribed tolerance and (b) the tool consistently feeds in the prescribed rate within the acceptable fluctuation range. However, current CAM software usually generates NURBS paths with a unit free parameter, which could be highly non-linear between the parameter of a cutter location and its corresponding arc length of the path. Fed with these

paths, the interpolation methods are not able to properly process them, resulting in the real-time cutter trajectories with errors beyond the tolerance and much feed rate fluctuation. Even though many researchers are working for better NURBS interpolation algorithms to solve this problem, it is quite passive and challenging due to the in-appropriate tool paths generated prior to machining. Hence, an effective solution is to generate better reference paths in the phase of tool path planning. Moreover, upon increasing demand for high speed machining in industry, the feed rate can be very high, which is a new challenge to the NURBS interpolation algorithms and can cause large trajectory errors. To find a completed solution, review on papers of NURBS path generation and interpolation is carried out.

Bedi et al. [66] developed a uniform interpolator by fixing the parameter increment at a constant value. Although the constant parameter increment simplifies the computation of cutter locations in the online interpolation process, this method cannot control the cutter trajectory error, the feed rate error, and the feed acceleration /deceleration. Shpitalni et al. [67] represented the relationship of the parameters of two points using the first order Taylor's expansion, instead of using the constant parameter increment. This method provides better results of the feed rate, compared to Bedi's method. Yang and Kong [68] mentioned that the approximation error using the first order Taylor's expansion is proportional to the sampling time and path curvature, causing feed rate fluctuation; therefore, the second order Taylor's approximation was proposed for higher precision. Yeh and Hsu [69] proposed an adaptive feed rate interpolator to adjust the curve speed and to confine the contour error (depends on curvature) within the specified tolerance

during interpolation. Xu et al. [70] proposed a NURBS interpolation algorithm with variable feed rates. Lo [71] used an iterative method in real-time to calculate the parametric increment, instead of using Taylor's expansion, because of heavy computation; however, the convergence time of this method is not fixed [72]. A time-series forecasting model was proposed in [73] to predict the arc length change rate in terms of the parameter. Since this method is predictive, it can result in contour and feed-rate errors. The higher order terms of the Taylor's expansion are neglected in the first order approximation in order to reduce computation load; however, this causes feed-rate fluctuation. Thus, Yeh et al. [74] proposed a speed controlled algorithm to avoid approximation errors, and higher order terms were estimated by calculating a compensatory value. But their algorithm is unable to handle tool paths with high curvatures.

In case of abrupt change of curve derivative among knot spans the Taylor expansion interpolator may produce large contour and feed-rate errors. An algorithm was proposed in [76] to remedy this problem by making sure that the next estimated path parameter is in the current knot span. But this makes Taylor expansion interpolation method complex which is not suitable in real time. In [77] they used chord-length parameterization for quintic spline causing feed-rate and contour errors, which leads to high acceleration and jerk values. To avoid this problem a tenth degree polynomial is solved iteratively in real time, to get spline parameter for a fixed interpolation displacement. With the development in NC controllers, the sampling time is decreasing to achieve high speed and precision; therefore, the iterative methods are not suitable for real time interpolation because

the convergence time is not fixed [75]. Erkorkmaz et al. [48] in his second approach developed a seventh degree *feed correction polynomial* for each spline segment by applying a least square fit over the arc displacement and spline parameter values. For each spline segment a separate seventh degree polynomial is required, which have to be included along with the data points in NC program, therefore, in high precision applications the amount of information becomes huge. A quintic polynomial was proposed in [75] for the same purpose. Since error bound is calculated with the assumption that the curvature varies with parameter monotonically, during segmentation monoticity is checked, so that the estimated arc length differ from the actual one no more than the error bound, otherwise the segment is bisected. Although the monoticity of curvature is checked by comparing the end curvatures with the curvature at middle point, even then the calculated error bound may differ from the actual error bound because of uneven variation of curvature between two consecutive data points. Lei et al. [78] used Hermit curve for each sub-interval to relate the curve parameter u to path length l and termed it as *Inverse length function*. NURBS curve's length is calculated numerically using adaptive quadrature method and divided into segments until the length of each segment is within the tolerance. If the curve length is estimated numerically using the adaptive quadrature method, short knot spans can be skipped, and cause wrong length estimation. The same problem was mentioned by [76] in estimating the curve length numerically, if these special knot spans are encountered, which they termed as extreme knot spans. Sharpe and Thorne [79] introduced a numerical technique of representing spacing of points as function of arc length for any parameter curve.

Farouki [80] reformulated a degree n polynomial curve as a rational curve with a parameterization closest to the ideal arc length parameterization. Gil and Keren [81] developed a family of curves to construct arc length parameterized curves between two arbitrary points. Hernandez and Estrada [82] proposed an iterative algorithm to generate a sequence of a prescribed number of points on a parametric curve with control of their distribution.

2.4 Summary

In this chapter, we reviewed current techniques closely related to this research, including gouging detection and its elimination from the NC tool path, NURBS tool path generation methods for pocket and three axis sculptured surface machining and finally the NURBS interpolation techniques to generate the cutter trajectories during machining. Some of the current methods still suffer certain drawbacks. In the following chapters, new methods to; solve gouging related problems, generate accurate NURBS CL path for both pocket and three axis surface machining and reparameterize the NURBS path with arc length, are developed.

Chapter 3 Gouging Detection & Removal for 3-Axis Surface Machining

3.1 Introduction

Today, there is a great demand for sculptured surface parts, which are challenging in design and manufacturing. During machining, in some applications these surfaces can be very complex that are susceptible to gouging. The term 'gouging' is used here to mean overcutting the design surface. To check gouging, normally the part surface is discretized to form a grid, but the size of grid directly influences the accuracy of gouging detection. Furthermore, most of the researchers have only considered ball-end mill cutter, however flat and round end mills are also used in 3-axis milling because of higher cutting efficiency. It is also important to include the tool shank and holder during gouging check, instead of only the bottom cutting edge (which has been the focus of many researchers).

The aim of this chapter is to introduce a generalized methodology to detect gouging for complex compound surface. The proposed method uses global optimization technique to improve the accuracy, and a discrete hybrid PSO method is introduced for increased searching efficiency. Mathematical models have been developed and implemented to cover most of the cutter shapes including free-form

milling cutters. Current approach reduces the long computational time for evaluating the whole compound surface, by checking only the potential gouge surface patches. An effective method is used to calculate the CL-points and detect gouging for free-form milling cutter also. Different examples are used to demonstrate that the proposed method can be used to identify and eliminate gouging in complex compound surface machining using a variety of cutter shapes.

3.2 Gouging Types in 3-Axis Machining

Sculptured surface parts such as stamping dies and injection molds are designed with compound surfaces because of shape complexity. A compound surface, $\mathbf{S}$ is composed of several surface patches $\mathbf{S}_i(u,v)$, $(i = 0,1,...n)$ connected each other with different continuity conditions. During NC machining of these surfaces, while removing material around a CC point, the tool may interfere with another point (or points) on the same surface patch or the neighboring patches. If this interference (overcut) is beyond the machining accuracy, it is called gouging and damages the part surface. Therefore, gouging detection and its removal for the NC tool-path generation becomes crucial.

We consider a 3-axis milling machine and assume that the tool cuts surface patch $\mathbf{S}_0$ at a CC-point $\mathbf{P}_{CC} = [x_{CC} \quad y_{CC} \quad z_{CC}]^T$ and the corresponding CL-point is $\mathbf{P}_{CL} = [x_{CL} \quad y_{CL} \quad z_{CL}]^T$. In the current work, different cutter shapes; ball, flat, round and free form end mills are considered, which are used for sculptured surface machining. These cutters may gouge the surface in several ways. The most typical

gouges are illustrated in Figure 3.1 using ball end mill as an example:- **local gouging**: when the local radius of curvature is less than the tool radius, the bottom of the cutter interferes with the surface see Figure 3.1(b); **global gouging:** If anywhere the angle θ between the surface normal $\vec{n}$ and the z-axis exceeds 90°, the cutter body gouges the surface in that region, see Figure 3.1(a); **gouging with neighboring patch:** If two or more surface patches are connected each other in such a way that the cutter accessibility to a CC-point on one surface patch is restricted by the surrounding surface patches, then machining at that CC-point gouges the surface (Figure 3.1(c)).

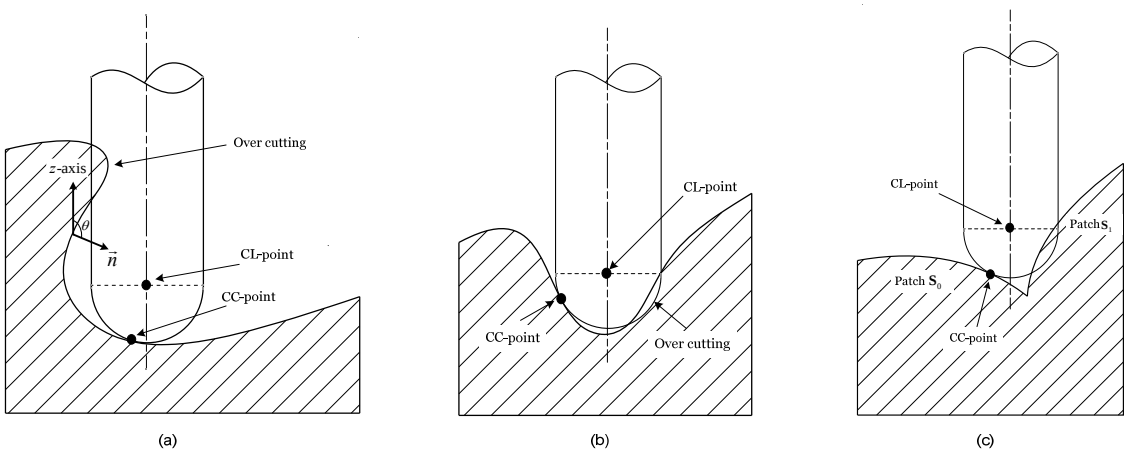


Figure 3.1 Three types of gouging: (a) global gouging, (b) local gouging and (c) gouging with neighboring patch.

3.3 Proposed Gouging Detection Method

The relationship between a cutter and a test point $\mathbf{P}_t$ on the surface for three cutter shapes is shown in Figure 3.2. The corresponding mathematical formulation of gouging detection is given by

$$f(z) = \begin{cases} \Rightarrow T^2 + \left(z(u,v) - z_o\right)^2 - R_1^2 & z_{CL} < z \le z_o \\ \quad where \quad T = \sqrt{\left(x(u,v) - x_{CL}\right)^2 + \left(y(u,v) - y_{CL}\right)^2} - R_2 & \\ \Rightarrow -1 \ ; \text{ if } T < 0 & \\ \Rightarrow \left(x(u,v) - x_{CL}\right)^2 + \left(y(u,v) - y_{CL}\right)^2 - \left(R_1 + R_2\right)^2 & z > z_o \end{cases} \text{Round end mill}$$

$$\Rightarrow \left(x(u,v) - x_{CL}\right)^2 + \left(y(u,v) - y_{CL}\right)^2 - R^2 \qquad z > z_{CL} \quad R = R_2; \ R_1 = 0 \quad \text{Flat end mill}$$

$$\begin{cases} \Rightarrow \left(x(u,v) - x_o\right)^2 + \left(y(u,v) - y_o\right)^2 + \left(z(u,v) - z_o\right)^2 - R^2 & z_{CL} < z \le z_o \\ \Rightarrow \left(x(u,v) - x_o\right)^2 + \left(y(u,v) - y_o\right)^2 - R^2 & z > z_o \end{cases} R = R_1; R_2 = 0 \quad \text{Ball end mill}$$

(3.1)

At a CL-point $\mathbf{P}_{CL}$ the minimum value of $f(z)$ is calculated using optimization. If Min. $f(z) < 0$ then, machining the surface at this location will cause gouging. The representation of $f(z)$ changes according to the cutter shapes and z-value of the test point $\mathbf{P} = [x \quad y \quad z]^T$ as shown in Eq. (3.1).

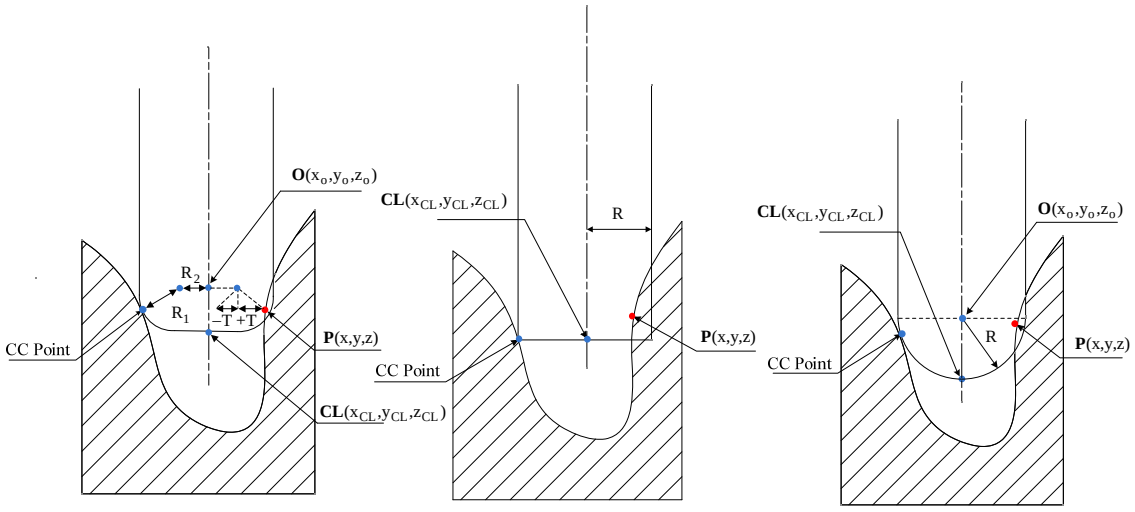


Figure 3.2 Three scenarios of gouging for different cutter shapes.

3.3.1 Global optimization procedure

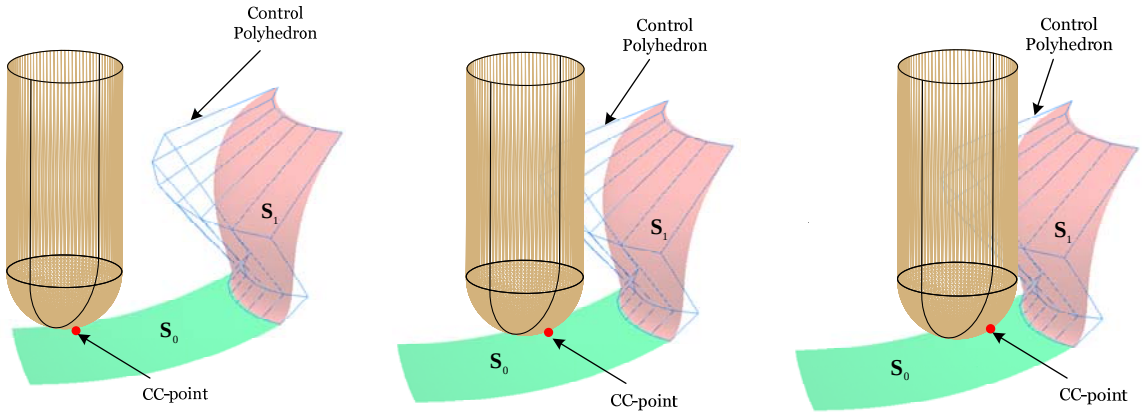
As it was mentioned that for gouging detection the minimum value of $f(z)$ is required, but for different shapes of cutting tools and complex part surface, large number of local optimum solutions are possible. Therefore at a CC point the detection of gouging becomes a global optimization problem. Finding an arbitrary local optimum is relatively straightforward by using local optimization methods. Finding the global maximum or minimum of a function is much more challenging. Several modern heuristic tools have evolved in the last two decades that facilitates solving global optimization problems like; genetic algorithm, ant colony, differential evolution and particle swarm optimization. Among these methods particle swarm optimization is becoming popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution [21]. Like other population-based optimization methods PSO also **randomly** searches the global best solution and it may take longer time to pinpoint the global optimum. On the other side gradient-based method can quickly find the local optimum solution, but they are unable to find the global solution if multiple local optimums exist. In order to further increase the searching efficiency of PSO, the parametric domain of the surface is discretized to form a two dimensional grid. We have combined this discrete domain approach of PSO with gradient method to reach the global solution quickly, and termed it as **discrete hybrid method** (will be discussed in chapter 5).

3.3.2 Algorithm of gouging detection and removal

In this section we explain all the steps required to find gouging for a compound surface. A compound surface has several patches, and for a CC-point only few of them may interfere with the cutting tool. Since, the computational cost for gouging detection is proportional to the number of patches; only intersecting neighbouring patches are scanned. Therefore, the first step is to find those patches by evaluating each patch against a criterion. The criterion is explained as follows:

Consider $\mathbf{S}$ as the compound surface; the cutter is positioned at a CC-point on the patch $\mathbf{S}_0$ and $\mathbf{S}_1$ is one of the neighboring surface patches, enclosed by its convex hull of the control polyhedron, denoted by ξ. According to the convex hull property of a NURBS surface [1], if the tool does not intersect with ξ, $\mathbf{S}_1$ will not be gouged by the cutter at that location (Figure 3.3(a)), which is necessary and sufficient condition. On the other side the intersection of ξ with the cutter envelope (Figure 3.3(c)) is a necessary condition for gouging the corresponding patch, but it is not sufficient as it can be seen from (Figure 3.3(b))

Once the intersecting patches are identified, the optimization tool discussed above is applied to the base patch (where the CC-point resides) and all intersecting patches one by one. The form of the objective function in Equation 3.1 depends on the shape of the cutter and the location of the test point $\mathbf{P}_t = \begin{bmatrix} x_t & y_t & z_t \end{bmatrix}^T$. Finally all CC-points which cause gouging are eliminated from the tool path.



(a) (b) (c)

Figure 3.3 Three types of interactions between a cutter and the neighboring patch at a CC-point: (a) $\mathbf{S}_1$ will not be gouged, (b) the cutter is intersecting with the control polyhedron of $\mathbf{S}_1$ but without gouging and (c) the cutter is intersecting with the control polyhedron of $\mathbf{S}_1$ and gouging it also.

Algorithm: Gouging detection

Define:

$\mathbf{S}_0(u_0, v_0)$, base surface patch where CL-point exists

$\mathbf{S}_i(u_i, v_i)(i = 1, 2, \ldots m)$, intersecting patches

$(\tilde{u}, \tilde{v})$, rough estimation of the global optimum position

(u^*, v^*), estimation of the global optimum position within tolerance

Input:

$\mathbf{S}$, the compound NURBS surface

$\mathbf{P}_{CL}(x_{CL}, y_{CL}, z_{CL})$, CL-point

τ, tolerance for gouging detection

Output:

Gouging free CL-points

Algorithm:

For a CL-point $\mathbf{P}_{CL}$

Find all intersecting patches $\mathbf{S}_i, i = 1, 2, \ldots m$

For $j = 0 \ \ to \ \ m$

1. Discretize $\mathbf{S}_j$
2. Run PSO to find $(\tilde{u}, \tilde{v})$
3. Use $(\tilde{u}, \tilde{v})$ as the input to the gradient method to find (u^*, v^*) within τ
4. If $f(u^*, v^*) < 0$ then eliminate this CL-point from the CL-path, and exit for the next CL-point

End

End

3.4 Free Form Cutters

Custom non-standard form cutters exhibit higher machining efficiency than the conventional end milling cutters for specific machining applications. Since the shape of the cutter is according to a particular part design, the profile of the cutter might be designed using free form curves; Figure 3.4 shows some examples of these cutters.

Figure 3.4 Form milling cutters with curve profile.

For conventional cutters the geometry is defined using some standard geometrical entities like lines and arcs; therefore the computation of a CL-point is straight forward. On the other side free form cutters require a curve to define their profile, therefore the calculation of a CL-point for this type of cutter is different. For this purpose we consider the profile curve and assume that it is in parametric form, denoted by $\mathbf{C}(u) = \begin{bmatrix} x(u) & y(u) & z(u) \end{bmatrix}^T$ (Figure 3.5(a)). To calculate a CL-point $\mathbf{P}_{CL}$ the parameter value t_{cc} on the profile curve is searched by matching the direction of cutter profile normal n_p and part surface normal n_s when the cutter is in contact with the part surface at $\mathbf{P}_{CC}$. Because of the strictly convex shape of the curve (curve tangent touches the curve only at a single point) there is only one unique value of t_{cc}; therefore, any gradient-based method can be used to quickly find it. Once t_{cc} is known the CL-point $\mathbf{P}_{CL}$ which is the summation of two translations $\mathbf{b}_1$ and $\mathbf{b}_2$ (Figure 3.5(b)) can be calculated as:

$$\mathbf{P}_{CL} = \mathbf{P}_{CC} + \mathbf{b}_1 + \mathbf{b}_2$$

The first translation is $\mathbf{b}_1$ which is from $\mathbf{P}_{CC}$ to point $\mathbf{o}$ and it is given by:

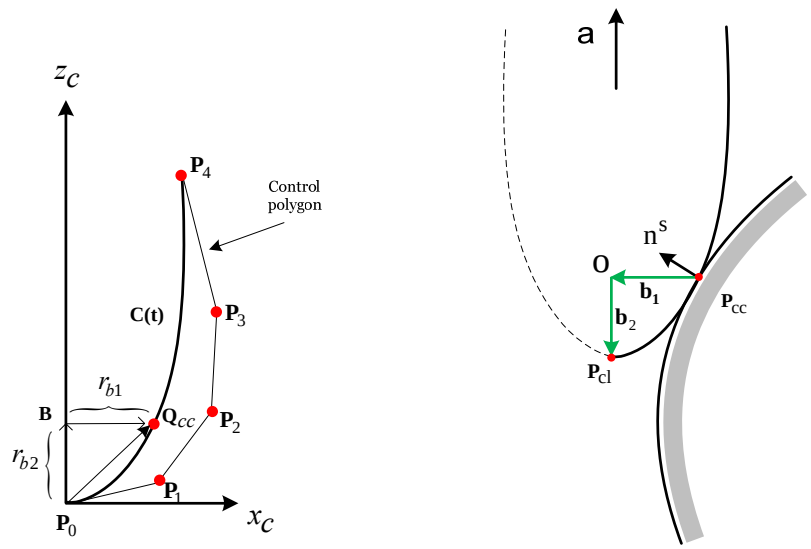
$$\mathbf{b}_1 = \left(\mathbf{a} \times (\mathbf{n}_s \times \mathbf{a}) \Big/ \left\| \mathbf{a} \times (\mathbf{n}_s \times \mathbf{a}) \right\| \right) . r_{b1}$$

where r_{b1} is the horizontal component of the vector from $\mathbf{P}_o$ to $\mathbf{Q}_{CC}$ (Figure 3.5(a))

The second translation is $\mathbf{b}_2$ which is from point $\mathbf{o}$ to $\mathbf{P}_{CL}$ and it is given by:

$$\mathbf{b}_1 = -\mathbf{a} . r_{b2}$$

and r_{b2} is the vertical component of the vector from $\mathbf{P}_o$ to $\mathbf{Q}_{CC}$



(a) Curve profile of the form milling cutter (b) CC-point and its corresponding CL-point

Figure 3.5 Computation of the cutter location point for a given CC-point and cutter profile curve.

Checking of a point on the surface to avoid gouging is more challenging when free form cutter is used to machine the surface. Say, $\mathbf{P}_t = [x_t \quad y_t \quad z_t]^T$ is a point on the part surface and we want to check that the point is outside the cutter envelope (no gouging) or inside (gouging). The first task is to find a point on the curve, that has the same z-elevation as of the test point $\mathbf{P}_t$, and call it as profile point $\mathbf{P}_p = [x_p \quad y_p \quad z_p]^T$ (shown in Figure 3.6(a)).

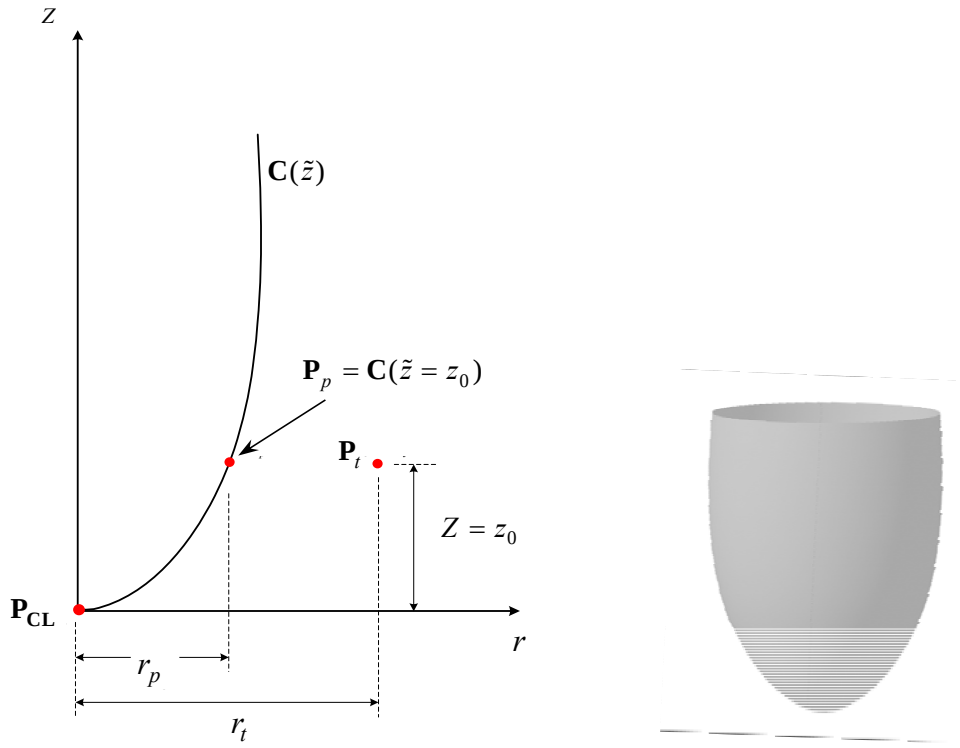


Figure 3.6(a) The Curve profile and the location of profile point according to the test point and (b) the corresponding envelope of the cutter.

Finding profile point $\mathbf{P}_p$ requires the estimation of curve parameter value u_p which gives a point on the curve having same z-elevation as of the test point

$\mathbf{P}_t$ i.e. $z_p \cong z_t$. For parametric curves the computation of such z_p is highly non linear problem therefore to facilitate the computation of $\mathbf{P}_p$ the profile curve $\mathbf{C}(u) = [x(u) \quad y(u) \quad z(u)] : [0 \quad 1] \rightarrow R^3$ is re-parameterized along z-axis. Since for arbitrary parametric curves the desired re-parameterization can only be approximated, therefore the objective is to approximate the function $f(m) = [x(m) \quad y(m) \quad m]$ by another curve $\mathbf{C}(\tilde{z})$. The feature of this approximated re-parameterized curve $\mathbf{C}(\tilde{z})$ is that $\mathbf{P}_p$ can be directly calculated from the z-elevation of the test point as the curve parameter value instead of searching the curve parameter. Let r_t and r_p denote the distances of $\mathbf{P}_t$ and $\mathbf{P}_p$ from $\mathbf{P}_{CL}$ respectively on xy-plan , if $r_t < r_p$, the test point $\mathbf{P}_t$ will be gouged.

3.5 Application

Example-1: Machining of a single surface patch using ball end mill.

A B-spline surface patch with high curvature variation, shown in Figure 3.7, is selected to show the effectivness of the proposed gouging detection algorithm. A ball end mill cutter of radius 0.5 inch is used to generate the gouge free CC-paths. The CC-points, shown in the Figure 3.7, are the gouging points, and should be eliminated from the tool-path. The gouge area associated to the cutter R = 0.5 inch can be machined after using smaller cutter size. Figure 3.8 shows the error for the given gouging allowance of 0.05 inch, and it is clear from the figure that all the CC-points are within the gouging allowance. The same surface was machined in CATIA

surface machining work bench, without changing the machining parameters and cutting tool. It was observed that the cutting tool gouged the surface at high curvature regions, which is shown in Figure 3.9.

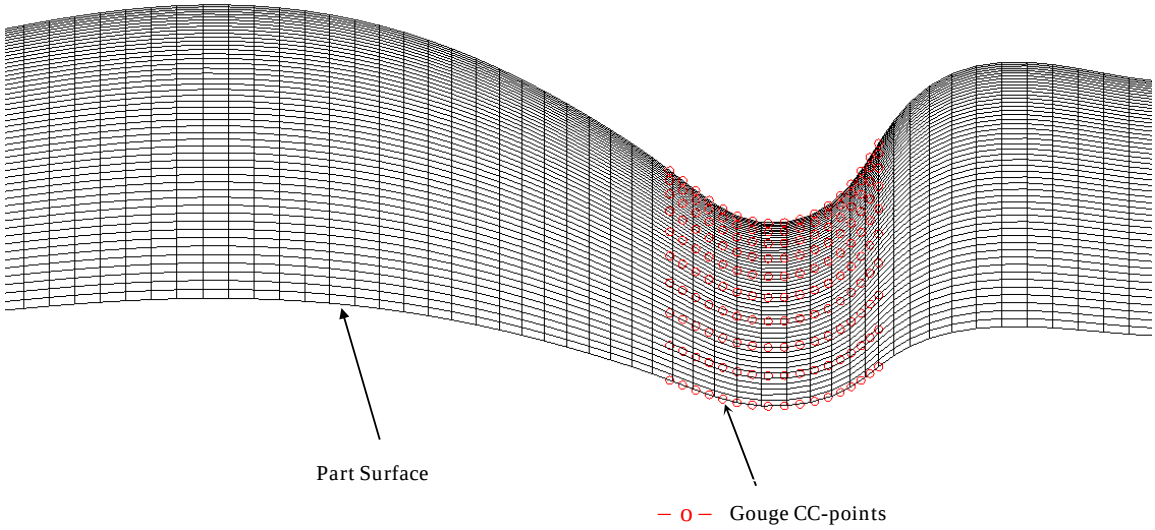


Figure 3.7 Gouge CC-points when using ball end mill cutter of radius 0.5 inch.

Example-2: Machining of a single surface patch using Round End Mill.

In 3-axis milling, ball endmills are more frequently used than other types of cutters because of its simple geometric shape. However, flat and round end mills offer higher cutting efficiency, because parts can always be cut by the periphery of

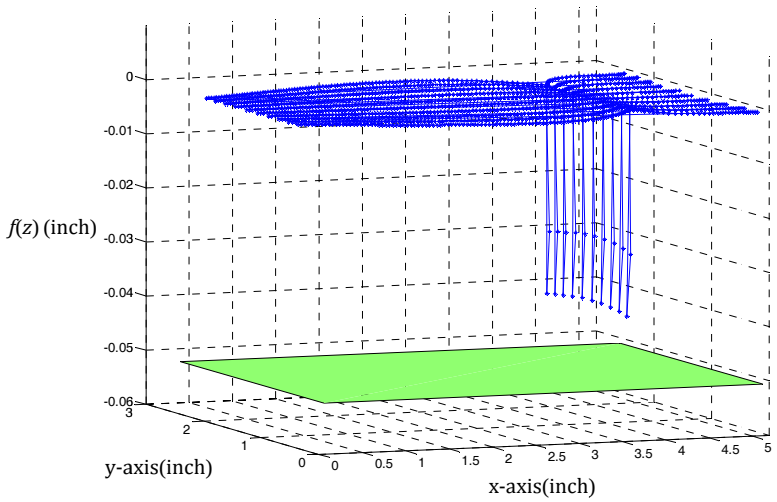


Figure 3.8 Error of gouging free tool path for ball end mill of 0.5 inch diameter and 0.05 gouging allowance

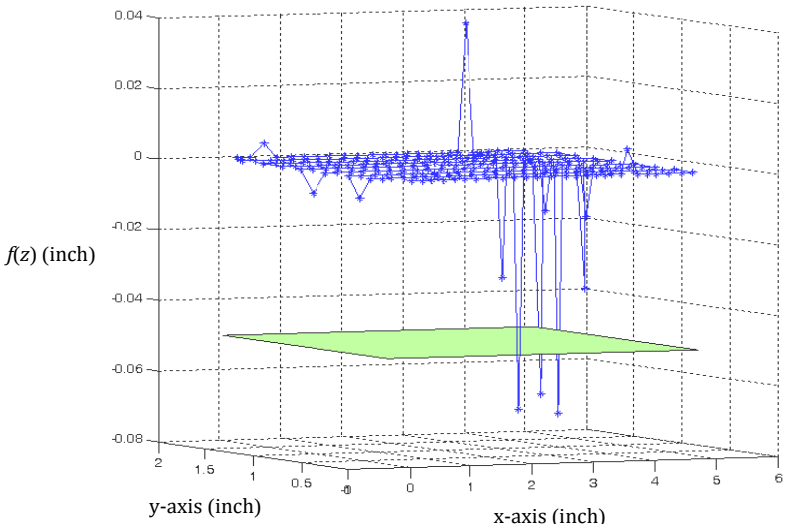
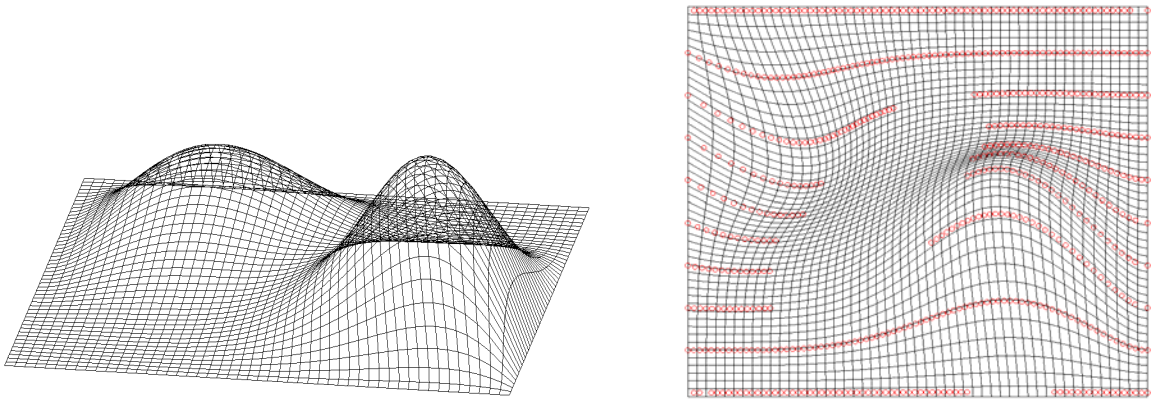


Figure 3.9 Error of gouging free tool path generated in CATIA for ball end mill of 0.5 inch diameter and 0.05 gouging allowance

these cutters at the maximum cutting efficiency [23]. Therfore roughing operation can be done with high machining efficiency using flat or round end mills, followed by fininshing operation with smaller size ball end mill cutter. In this example we have selected a surface, shown in Figure 3.10(a), with high curvature variations, and a round end mill cutter to show that the proposed method can handle complex

surfaces and a variety of cutter shaps. Figure 3.10(b), shows the gouging free CC-paths by using round end mill (R_1 = 0.2 inch, R_2 = 0.2 inch).



(a) (b)

Figure 3.10 (a) Surface for Example-2 and (b) gouging free CC-path using round end mill R1 = 0.2 inch, R2 = 0.2 inch.

Example-3: Machining of a Composite surface using ball end mill.

Surface models of industrial parts with complicated shape are usually composed of many surface patches, called compound surface. During machining of one patch the cutter may gouge the other neighboring patches, therefore it is important for gouging detection algorithm to check the suspected neighboring patches also. Figure 3.11, shows a compound surface which contains six surface patches. We have used three different sizes (Radius=0.2 inch, 0.55 inch, 0.7 inch) of ball end mill cutters, and the corresponding gouge free CC-points (squares) and gouge CC-points (circles) are shown in Figure 3.12.

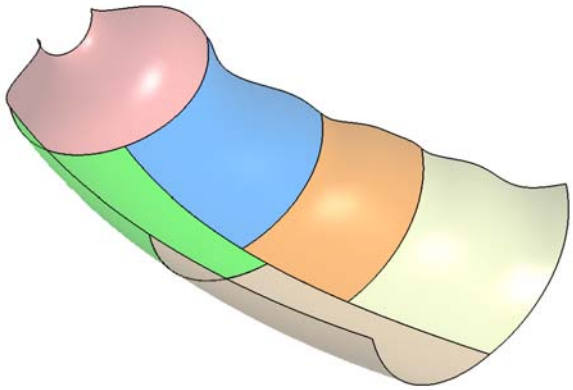


Figure 3.11 Composite NURBS surface with six surface patches.

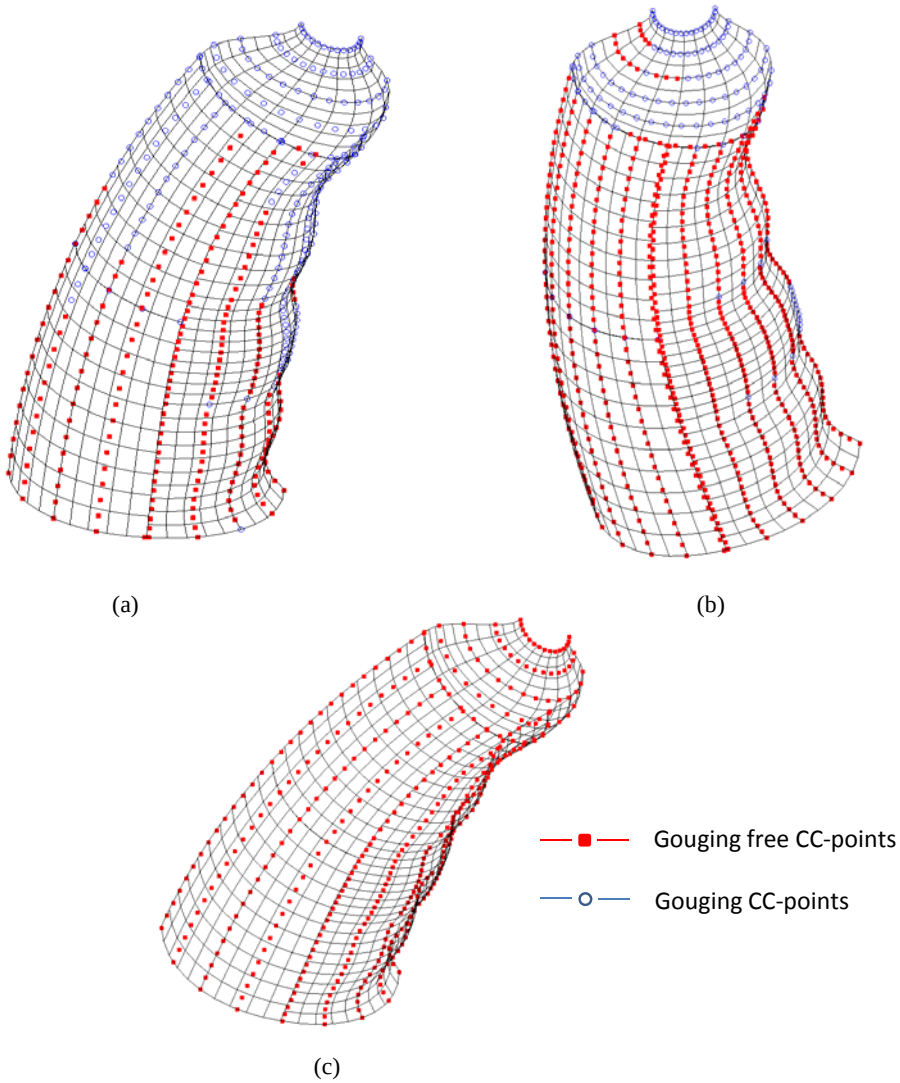


Figure 3.12 CC-points using ball end mill of (a) 0.7 inch diameter, (b) 0.5 inch diameter and (c) 0.2 inch diameter

3.6 Summary

In this paper a generalized gouging detection algorithm has been proposed for three axis compound surface machining. For higher accuracy the proposed method uses optimization technique and a discrete hybrid PSO method is developed to improve the computational efficiency. The cutter shapes mostly used in the industry have been discussed to check gouging for complex surface machining. To avoid searching across the whole surface, which is normally composed of several surface patches, a method for the query of only potential gouge patches has been developed. Different examples have shown that the proposed method is versatile and robust enough to be used from simple to complex surfaces for a variety of cutter shapes with higher accuracy and efficiency.

Chapter 4 Offset for B-Spline NC Tool-Path with Offset Error Globally Bounded

4.1 Introduction

The topic of representing the offset of a 2-D B-spline curve in the same form has been under research for a long time, and it has many industrial applications, especially, in NC tool path generation for pocketing. For B-spline tool paths, it is often required that the tool paths have fewer control points, lower base function degree, and higher geometric accuracy. However, the existing methods often generate the offsets of 2-D free-form curves in the form of B-spline curves with high function degree and many control points. Although these offsets are useful in computer-aided design, they are inappropriate for the use of CNC machining. To address the problems in order to generate high quality B-spline tool paths, this original work formulates an error function of the offset approximation and then constructs a NURBS curve to globally bound the errors. By checking the maximum coefficient of the bounding curve, the upper bound of all the approximated offset errors is found and the errors can be reduced by adding more offset points at the appropriate locations. The proposed new approach is more efficient, and the

resulting offsets in B-spline are more accurate with fewer control points and low function degree. It is useful to generate B-spline tool paths for CNC pocketing, and has potential for other applications in industry.

In this chapter we use a different approach from Elber and Cohen [32] to bound the error globally in curve offset approximation. They computed the difference of $\mathbf{C}_d^a(u)$ and $\mathbf{C}(u)$ to generate the difference function, which may underestimate the error bound (for details see [35]). In contrast, we estimate the difference function between the approximated offset and $\mathbf{C}_d^a(u)$ and the exact offset $\mathbf{C}_d(u)$ to develop the error function $\delta(u)$, which guarantees to avoid the underestimation problem, faced by [32]. Although the approximation methods based on circular approximation [35, 36, 41] give exact and sharper error bound than our method, but normally produce high degree approximate offset curve with larger number of control points (curve segments) compare to our method. The proposed method is robust against the presence of inflection points and loops in the exact and approximated offset curve, unlike the other methods [35, 36, and 41].

4.2 Formulation of Upper Bound Function $\delta(u)$ for Offset Error

For a given regular B-spline curve $\mathbf{C}(u)=\left[x(u),\, y(u)\right]^{\mathsf{T}}$, its true offset curve $\mathbf{C}_d(u)$ at distance *d* along its unit normal vector $\mathbf{N}(u)$ is given by

$$\mathbf{C}_d(u)=\begin{bmatrix} x_d(u) \\ y_d(u) \end{bmatrix}=\mathbf{C}(u)+d\cdot\mathbf{N}(u) \tag{4.1}$$

Where,

$$\mathbf{N}(u) = \frac{1}{\sqrt{x'(u)^2 + y'(u)^2}} \cdot \begin{bmatrix} -y'(u) \\ x'(u) \end{bmatrix}$$

or

$$\mathbf{N}(u) = \frac{1}{\sqrt{x'(u)^2 + y'(u)^2}} \cdot \begin{bmatrix} y'(u) \\ -x'(u) \end{bmatrix}$$

To represent the offset with a B-spline curve, the true offset $\mathbf{C}_d(u)$ is approximated by fitting a B-spline to it. Basically, the approximate offset $\mathbf{C}_d^a(u)$ is represented as $\mathbf{C}_d^a(u) = \left[x_d^a(u), y_d^a(u) \right]^{\mathbf{T}}$.

By considering Figure 4.1, the difference $\mathbf{R}(u)$ between the true offset and the original curve is

$$\begin{aligned} \mathbf{R}(u) = \begin{bmatrix} x_r(u) \\ y_r(u) \end{bmatrix} &= \mathbf{C}_d(u) - \mathbf{C}(u) \\ &= d \cdot \mathbf{N}(u) = \frac{d}{\sqrt{x'(u)^2 + y'(u)^2}} \cdot \begin{bmatrix} -y'(u) \\ x'(u) \end{bmatrix} \end{aligned}$$

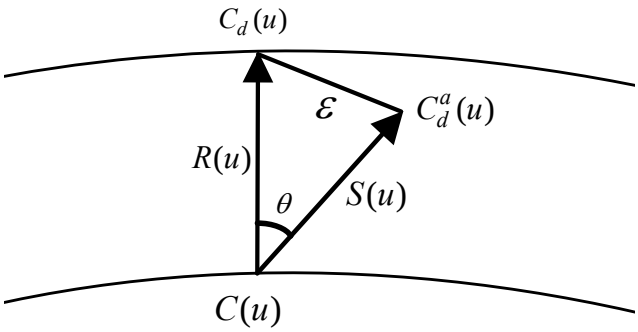


Figure 4.1 A B-spline curve its offset and error ε

and the difference $\mathrm{s}(u)$ between the approximate offset and the original curve is

$$\mathbf{S}(u)=\begin{bmatrix} x_s(u) \\ y_s(u) \end{bmatrix}=\mathbf{C}_d^a(u)-\mathbf{C}(u)=\begin{bmatrix} x_d^a(u)-x(u) \\ y_d^a(u)-y(u) \end{bmatrix}$$

The offset error $\varepsilon(u)$ is defined as the distance between the true offset $\mathbf{C}_d(u)$ and the approximate offset $\mathbf{C}_d^a(u)$. The representation of the offset error is

$$\varepsilon(u)=\left|\mathbf{R}(u)-\mathbf{S}(u)\right|=\left\|\begin{bmatrix} x_r(u)-x_s(u) & y_r(u)-y_s(u) \end{bmatrix}\right\|$$

Because of the presence of square root, $\varepsilon(u)$ cannot be represented in polynomial form. Therefore, the square of the offset error is introduced as

$$\varepsilon(u)^2=\left(x_r(u)-x_s(u)\right)^2+\left(y_r(u)-y_s(u)\right)^2 \tag{4.2}$$

To simplify the offset error function, set

$$f_1(u)=\left|\mathbf{R}(u)\right|^2+\left|\mathbf{S}(u)\right|^2=d^2+x_s(u)^2+y_s(u)^2$$

and

$$f_2(u) = \left|\mathbf{R}(u)\right| \cdot \left|\mathbf{S}(u)\right| \cdot \cos\theta = x_r(u) \cdot x_s(u) + y_r(u) \cdot y_s(u)$$

where θ is the angle between $\mathbf{R}(u)$ and $\mathbf{S}(u)$, and Eq. (4.2) can be simplified as

$$\varepsilon(u)^2 - f_1(u) = -2 \cdot f_2(u) \tag{4.3}$$

Taking square on both sides

$$\left(\varepsilon(u)^2 - f_1(u)\right)^2 = \left(-2 \cdot f_2(u)\right)^2$$

$$\varepsilon(u)^4 - 2\varepsilon(u)^2 f_1(u) + f_1(u)^2 = 4f_2(u)^2$$

Since ε is normally very small therefore; $\varepsilon(u)^4 \approx 0$

$$-2\varepsilon^a(u)^2 f_1(u) + f_1(u)^2 = 4f_2(u)^2$$

where $\varepsilon^a(u)$ is the approximated offset error function

$$\varepsilon^a(u)^2 = \frac{f_1(u)^2 - 4f_2(u)^2}{2f_1(u)} \tag{4.4}$$

$$\varepsilon^a(u)^2 = \frac{\left(f_1(u) + 2f_2(u)\right)\left(f_1(u) - 2f_2(u)\right)}{2f_1(u)}$$

$$\varepsilon^a(u)^2 = \left(0.5 + \frac{f_2(u)}{f_1(u)}\right)\left(f_1(u) - 2f_2(u)\right)$$

using Eq. (4.3)

$$\varepsilon^a(u)^2 = \left(\frac{1}{2} + \frac{f_2(u)}{f_1(u)}\right) \cdot \varepsilon(u)^2 \tag{4.5}$$

Lemma: For a given B-spline curve $\mathbf{C}(u)$ and its offset distance d, the true offset curve $\mathbf{C}_d(u)$ cannot be represented with a B-spline curve. To represent the true offset curve, a B-spline curve $\mathbf{C}_d^a(u)$ is fit to sample points on $\mathbf{C}_d(u)$ to approximate it. The offset error function $\varepsilon^a(u)$ is defined above. A function $\delta(u)$ of an upper bound of the offset error is found as

$$\delta(u) = \frac{\varepsilon^a(u)^2}{0.5 + 2 \cdot \left(\frac{f_2(u)}{f_1(u)}\right)^2} \geq \varepsilon(u)^2$$

Proof:

The module of the difference $\mathbf{R}(u)$ between the true offset and the original curve is $\left|\mathbf{R}(u)\right|$, and the module of the difference $\mathbf{S}(u)$ between the true offset and the approximate curve is $\left|\mathbf{S}(u)\right|$. The following relationship always holds,

$$\left(\left|\mathbf{R}(u)\right| - \left|\mathbf{S}(u)\right|\right)^2 \geq 0$$

Then we can get

$$0 \le \frac{\left|\mathbf{R}(u)\right| \cdot \left|\mathbf{S}(u)\right|}{\left|\mathbf{R}(u)\right|^2 + \left|\mathbf{S}(u)\right|^2} \le \frac{1}{2}$$

Since angle θ is the angle between $\mathbf{R}(u)$ and $\mathbf{S}(u)$, generally, the in-equation $0 \le \cos\theta \le 1$ is true. By multiplying this in-equation with the above in-equation, we get

$$0 \le \frac{\left|\mathbf{R}(u)\right| \cdot \left|\mathbf{S}(u)\right| \cdot \cos\theta}{\left|\mathbf{R}(u)\right|^2 + \left|\mathbf{S}(u)\right|^2} \le \frac{1}{2}$$

which is

$$0 \le \frac{f_2(u)}{f_1(u)} \le \frac{1}{2}$$

meanwhile, we can get an in-equation

$$\frac{f_2(u)}{f_1(u)} \ge 2 \cdot \left(\frac{f_2(u)}{f_1(u)}\right)^2$$

therefore,

$$\frac{1}{2} + \frac{f_2(u)}{f_1(u)} \ge \frac{1}{2} + 2 \cdot \frac{f_2(u)^2}{f_1(u)^2}$$

using Eq. (4.5),

$$\varepsilon^{a}(u)^{2}=\left(\frac{1}{2}+\frac{f_2(u)}{f_1(u)}\right)\cdot\varepsilon(u)^{2}\geq\left(\frac{1}{2}+2\cdot\frac{f_2(u)^{2}}{f_1(u)^{2}}\right)\cdot\varepsilon(u)^{2}$$

and

$$\delta(u)=\frac{\varepsilon^{a}(u)^{2}}{\frac{1}{2}+2\cdot\frac{f_2(u)^{2}}{f_1(u)^{2}}}=\frac{2\cdot\varepsilon^{a}(u)^{2}}{1+4\cdot\frac{f_2(u)^{2}}{f_1(u)^{2}}}\geq\varepsilon(u)^{2}$$

[End]

The function can be further simplified as

$$\delta(u)=\frac{f_1(u)^{2}-4\cdot f_2(u)^{2}}{f_1(u)^{2}+4\cdot f_2(u)^{2}}\cdot f_1(u)$$

Since the function is always equal to or larger than the square of the offset error function, it is an upper bound function of the offset error function. In other words, the square root of the maximum value of the bound function is guaranteed to be larger than all the error along the offset curve. The function is a rational function; and its numerator and denominator are polynomials without square root. Thus, it can be re-represented as a NURBS curve function, which is crucial to have offset error globally bounded in the offset approximation.

4.3 Globally Bounded Offset Error in the Offset Approximation

The main feature of this work is to effectively control the maximum offset error within a specified tolerance in the process of approximating the true offset with a B-spline curve. The resulting offset in B-spline form is accurate with less control points and lower function degree. This feature is superior for the purpose of generating B-spline tool paths, compared to the current offset methods. In the current work of finding approximate offsets in B-spline form, the offset errors between the approximate and the true offsets are not globally checked. It is always a concern about the errors of the approximate offset between the sample points. This is why some researchers have proposed offset approximation methods with the errors bounded all along the offset curve.

To globally control the offset error, it is necessary to know the maximum error of an approximate offset or the upper bound of this error. For this purpose upper bound function of the offset error function is represented in NURBS form as

$$\delta(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u)\ w_i . F_i}{\sum_{i=0}^{n} N_{i,p}(u)\ w_i} \tag{4.6}$$

Where F_i are the control points, w_i are the weights, and $N_{i,p}(u)$ are the $p\text{th}-\text{degree}$ basis functions. Among the scalar coefficients F_i, we can find the maximum coefficient σ ($\sigma = \max(F_i)$) and it can be proved that $\sigma \geq \max(\varepsilon(u)^2)$ by assuming all the coefficients in Eq. (4.6) equal to σ

$$\delta(u) \le \frac{\sum_{i=0}^{n} N_{i,p}(u)\ w_i.\sigma}{\sum_{i=0}^{n} N_{i,p}(u)\ w_i}$$

and

$$\delta(u) \le \frac{N_{1,p}(u)w_1.\sigma + N_{2,p}(u)w_2.\sigma +,\ldots,N_{n,p}(u)w_n.\sigma}{N_{1,p}(u)w_1 + N_{2,p}(u)w_2 +,\ldots,N_{n,p}(u)w_n}$$

which leads to

$$\delta(u) \le \frac{\sigma.\left(N_{1,p}(u)w_1 + N_{2,p}(u)w_2 +,\ldots,N_{n,p}(u)w_n\right)}{\left(N_{1,p}(u)w_1 + N_{2,p}(u)w_2 +,\ldots,N_{n,p}(u)w_n\right)}$$

then, we have

$$\delta(u) \le \sigma$$

This result implies that

$$\sigma \ge \max(\delta(u)) \ge \max(\varepsilon(u)^2) \tag{4.7}$$

Therefore, $\sqrt{\sigma}$ is an upper bound of the approximate offset errors, and it can be used to estimate the maximum offset error. Hence, the principle of approximating a true offset by controlling its error under the tolerance is to construct an appropriate offset error function and represent it with a NURBS curve, if the upper bound $\sqrt{\sigma}$ is larger than the specified tolerance φ, the original B-spline curve is subdivided at a location of a domain where the maximum coefficient controls. This process is continued until the error along the whole curve length is within the tolerance. The

main procedure of finding an approximate offset with the error globally bounded is outlined as follows.

1. Sample points on the offset curve $\mathbf{C}_d(u)$ using Eq. (4.1), Figure 4.2(b).
2. Apply the least square method to approximate the exact offset $\mathbf{C}_d(u)$ by a B-spline curve $\mathbf{C}_d^a(u)$ with any suitable degree, Figure 4.3(a).
3. Find the corresponding upper bound function $\delta(u)$ of the offset error, Figure. 4.3(b).
4. Find the maximum value of the coefficients of $\delta(u)$, $\max(F_i)$ in Eq. (4.6) to bound the maximum offset error according to Eq. (4.7).
5. If $\sqrt{\max(F_i)} > \tau$, break the base curve into segments at u_i and u_{i+p+1} knot values of $\mathbf{C}_d^a(u)$.
6. Repeat steps 2-5 for each segment of the base curve, until the offset approximation errors for all segments are within the tolerance, Figure 4.5.
7. Eliminate removable control points from the approximated offset curve segments generated in the above steps so long as the errors are within the tolerance.
8. Output the composite approximated offset in B-spline form with offset error globally bounded and relatively small number of control points.

To demonstrate the advantages of the proposed method, it is applied to a planar B-spline curve of degree five, termed as original curve (see Figure 4.2(a)) in

order to approximate its exact offset. The control points of the base curve are listed in Table-4.1, the offset distance d is 0.5 inch and the offset approximation tolerance τ is 10^{-2}. Figures 4.2 to 4.5 demonstrate graphically the process of generating offset approximations. First, points are sampled on the offset curve $\mathbf{C}_d(u)$, Figure.4 2(b). There are several techniques of curve fitting to the sample points; we have chosen global least square approximation technique and parameterization from the base curve to get offset approximation $\mathbf{C}_d^a(u)$ using a 3rd degree B-spline curve with 28 control points, Figure 4.3(a). To determine the accuracy of $\mathbf{C}_d^a(u)$, the proposed error bound function $\delta(u)$ is computed. To represent $\delta(u)$ as a NURBS curve, let the numerator is $A(u)$

$$(f_1(u)^2 - 4 \cdot f_2(u)^2) \cdot f_1(u) = A(u) = \sum_{i=0}^{n} N_{i,p}(u)\, P_i$$

and the denominator is $B(u)$

$$f_1(u)^2 + 4 \cdot f_2(u)^2 = B(u) = \sum_{j=0}^{m} N_{j,q}(u)\, Q_j$$

Where P_i and p are the control points and degree of $A(u)$ and similarly Q_j and q are the control points and degree of $B(u)$. We have used Matlab spline tool box to solve for $A(u)$ and $B(u)$ and found as follows

$$\delta(u) = \frac{A(u)}{B(u)} = \frac{\sum_{i=0}^{1000} N_{i,38}(u)\, P_i}{\sum_{j=0}^{689} N_{j,28}(u)\, Q_j} \tag{4.8}$$

Since $q < p$ and $m < n$, therefore to match the degree and number of control points of $B(u)$ to $A(u)$ we used degree elevation and knot insertion processes which converts $B(u)$ into $D(u) = \sum_{j=0}^{1000} N_{j,38}(u) w_j$ (note that degree elevation and knot insertion does not change the original curve). Now Eq. (4.8) becomes

$$\delta(u) = \frac{A(u)}{D(u)} = \frac{\sum_{i=0}^{1000} N_{i,38}(u)\, P_i}{\sum_{i=0}^{1000} N_{i,38}(u)\, w_i} \tag{4.9}$$

Assume $F_i = \frac{P_i}{w_i}$ $(i = 0,\ldots,1000)$ and $\delta(u)$ is written as

$$\delta(u) = \frac{A(u)}{D(u)} = \frac{\sum_{i=0}^{1000} N_{i,38}(u)\, w_i.F_i}{\sum_{i=0}^{1000} N_{i,38}(u)\, w_i} \tag{4.10}$$

Eq. (4.10) is the standard NURBS representation of $\delta(u)$ (see Figure 4.3(b)). To find $\delta(u)$ in NURBS form using Matlab spline tool box on duel core processor of 2.40 GHz, the total computation time was 3.581 seconds.

We have already proved that offset error can be globally bounded by the maximum value of the coefficients of $\delta(u)$. In this example the 129th coefficient of $\delta(u)$ is the extreme value and it is found to be $F_{129} = 6.021 \times 10^{-4}$. Since, $\sqrt{F_{129}} > \tau$ the approximated offset is segmented with the objective to decrease the offset error. Using the local property of NURBS curve the base curve $\mathbf{C}(u)$ is broken at u_{129} and u_{133} to convert it into three segments. For all segments, new offset curves are approximated and upper bound functions are generated which are within the tolerance as shown in Figure 4.5.

Initially, 28 control points were used to approximate the offset and the errors were higher than the prescribed tolerance, Figure 4.3(b). When the proposed upper bound function is used to find the maximum error and its location; after segmentation only 23 control points are required and the maximum offset error is lower than the tolerance. Figure 4.4(a) shows the square of exact offset error $\varepsilon(u)^2$ to compare with the error estimated by the proposed upper bound function $\delta(u)$, Figure 4.3(b). For further clarification the difference of $\delta(u)$ and $\varepsilon(u)^2$ is depicted in Figure 4.4(b), which reveals two important facts; (1) the upper bound function is always larger than the exact offset error as it was proved by the lemma in section-4.2, the difference is very small, means the over estimation cost is low.

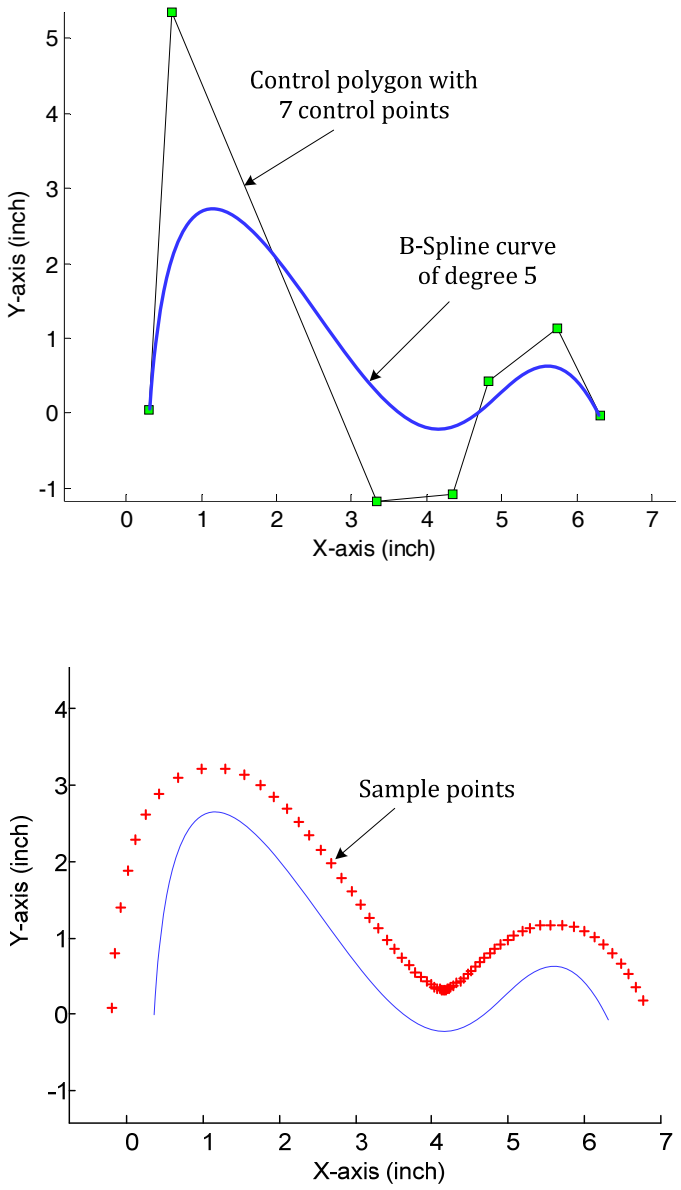


Figure 4.2(a) A planar B-spline curve of degree 5 and its control polygon; (b) sample points on the offset curve.

Table 4.1 Control points of the B-spline curve shown in Figure 4.2(a).

Control points	**1**	**2**	**3**	**4**	**5**	**6**	**7**
x (inch)	0.3002	0.6007	3.3409	4.3524	4.8219	5.7489	6.3163
y (inch)	0.0479	5.3615	-1.1651	-1.0703	0.4364	1.1416	-0.0375

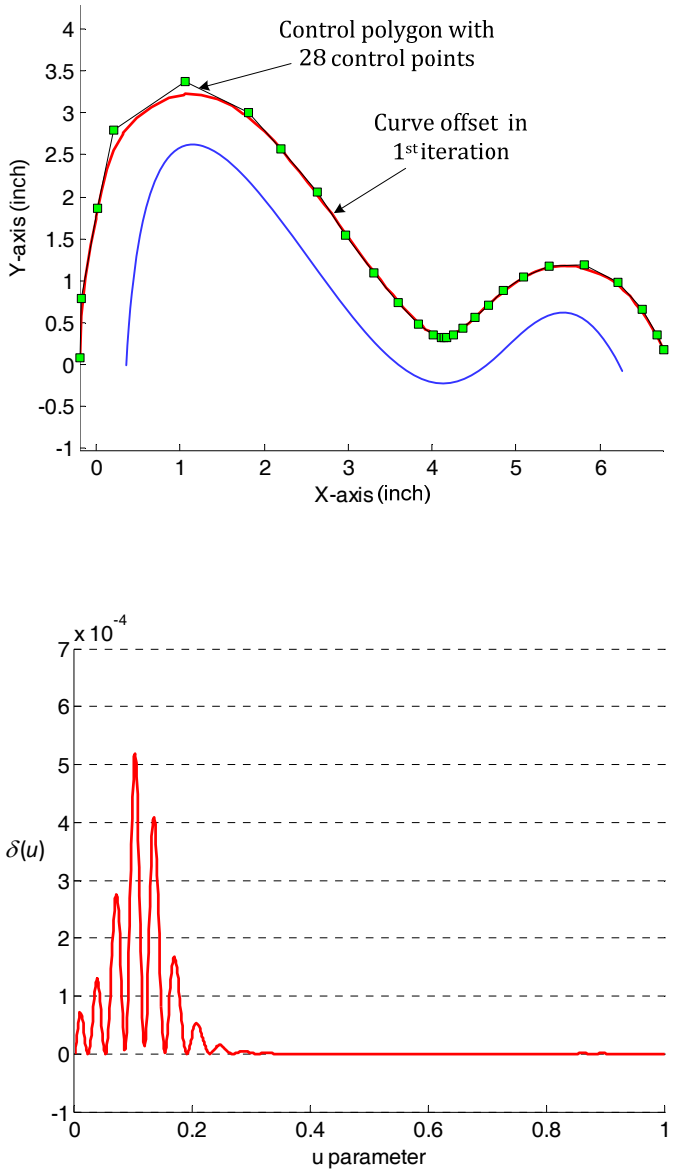


Figure 4.3(a) Offset approximation in the first iteration and its control polygon; (b) upper bound function of the offset error $\delta(u)$.

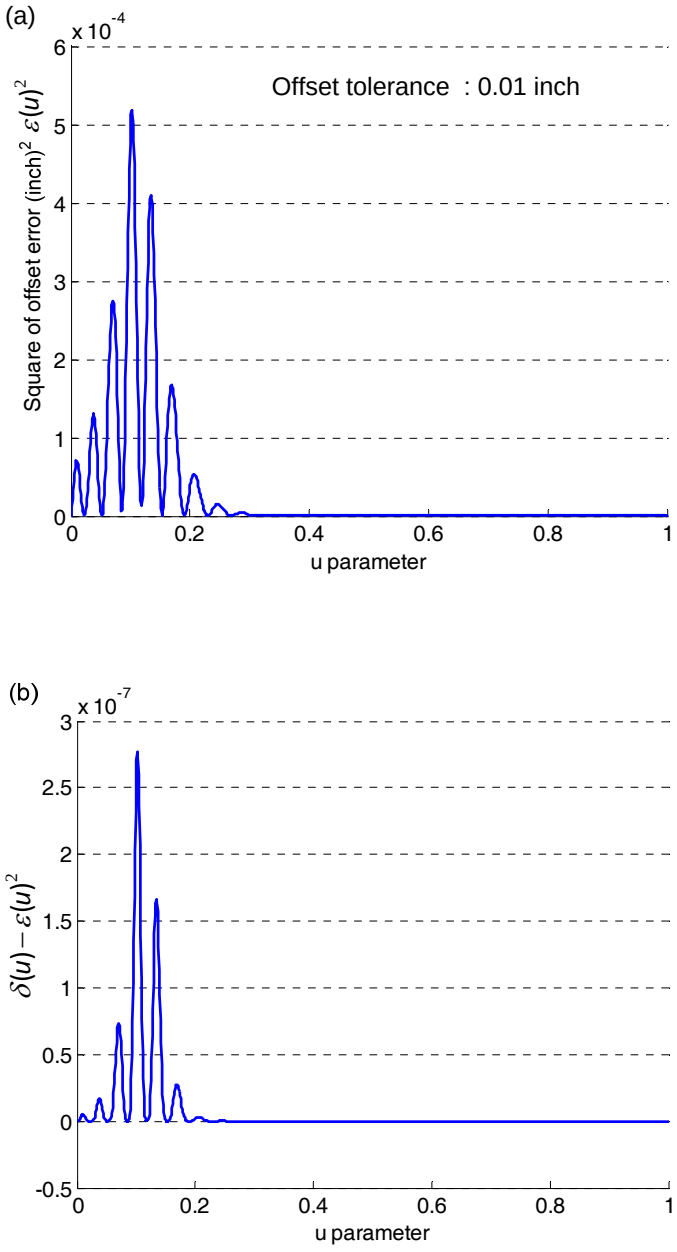
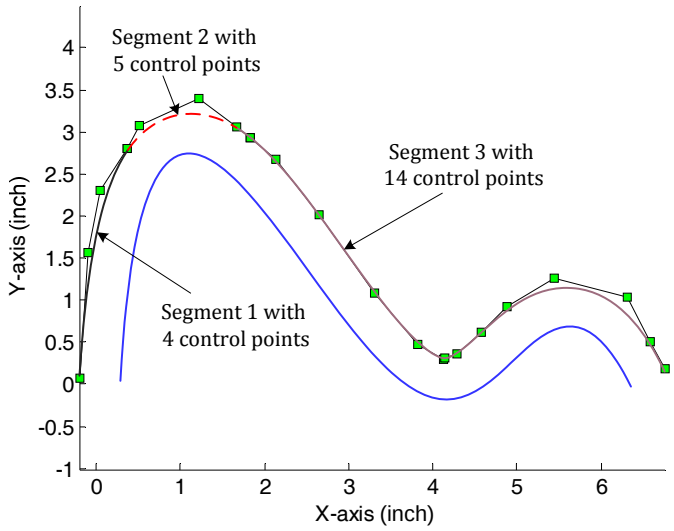
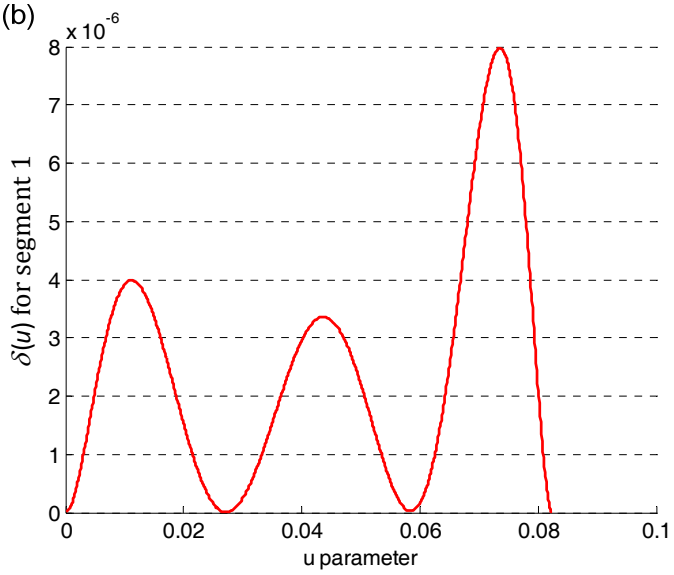


Figure 4.4 (a) Square of the exact offset error; (b) difference between the estimated and exact errors.

(a)


Segment 2 with
5 control points
Segment 3 with
14 control points
Segment 1 with
4 control points
Y-axis (inch)
X-axis (inch)
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
1
2
3
4
5
6

(b)


x 10^-6
δ(u) for segment 1
u parameter
8
7
6
5
4
3
2
1
0
0
0.02
0.04
0.06
0.08
0.1

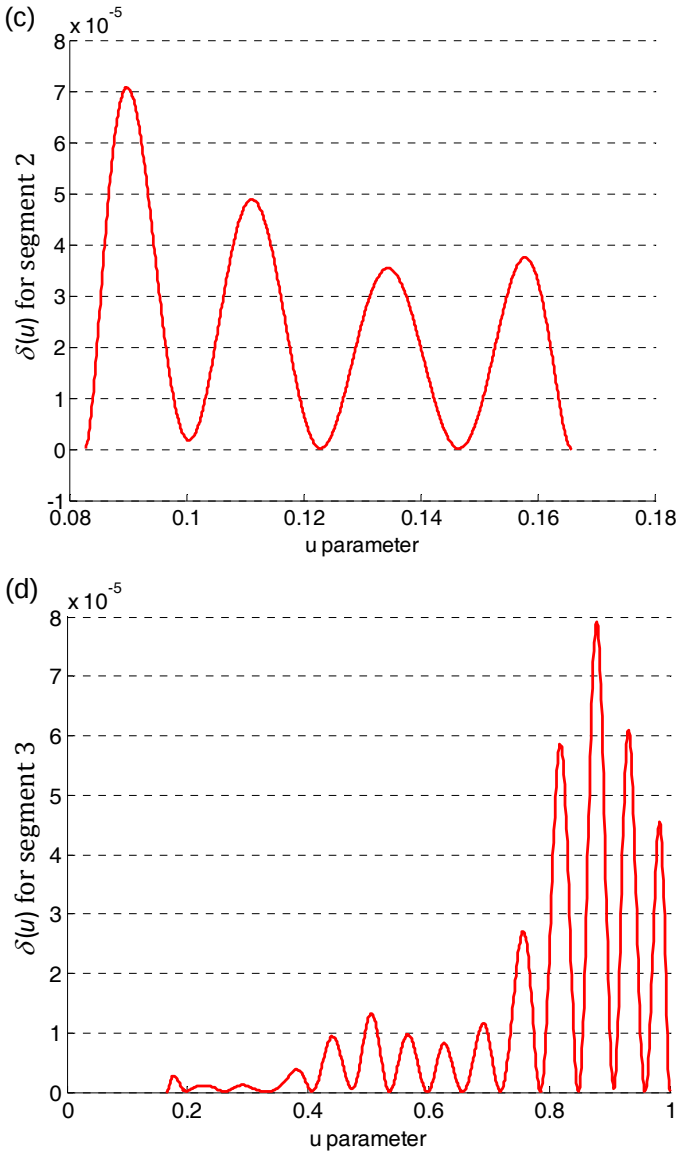


Figure 4.5. Offset approximation consists of three B-spline curve segments and the corresponding $\delta(u)$ for 3 segments.

4.4 Application and Comparisons

In this section, we give the comparison of our method based on error function with other offset approximation methods. Lee et al. [35] gave the comparison of the existing curve offsetting methods with their method in terms of accuracy and number of control points using different curve shapes. We have

selected the same curve shapes to demonstrate the effectiveness of our method in terms of the number of control points and order of the curve for a given tolerance value. The first example in Figure 4.6 is a cubic Bezier curve with four control points: (-0.7859, 0.8918), (-0.9933, -0.5969), (0.3000, -2.5000) and (0.9000, -0.2000). An approximate offset curve $\mathbf{C}_d^a(u)$ at radius 1.0 is generated using the least square method and the distance between $\mathbf{C}_d(u)$ and $\mathbf{C}_d^a(u)$ is estimated using the error function $\delta(u)$. The maximum deviation of $\mathbf{C}_d^a(u)$ form $\mathbf{C}(u)$ is bounded by $(d \pm \sqrt{\sigma})$ as shown in Figure 4.7(a). The corresponding error $\delta(u)$ is shown in Figure 4.7(b). The offset approximations for different tolerance values along the upper bound functions $\delta(u)$ are shown in Figure 4.8(a)-(e).

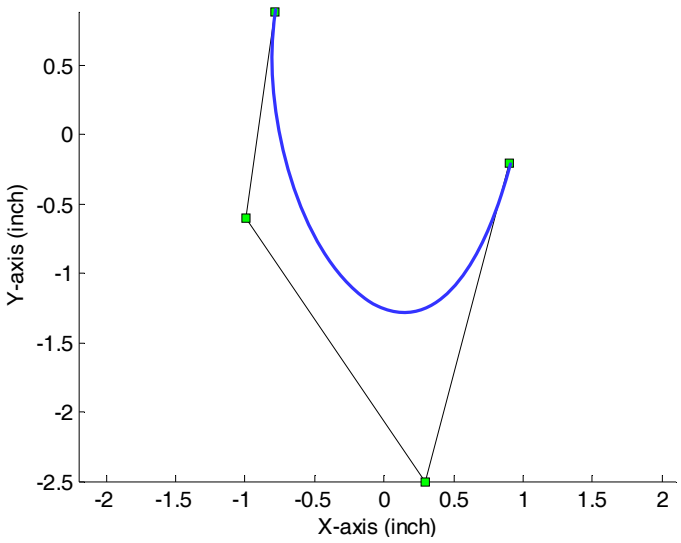


Figure 4.6 Cubic Bezier curve with control polygon.

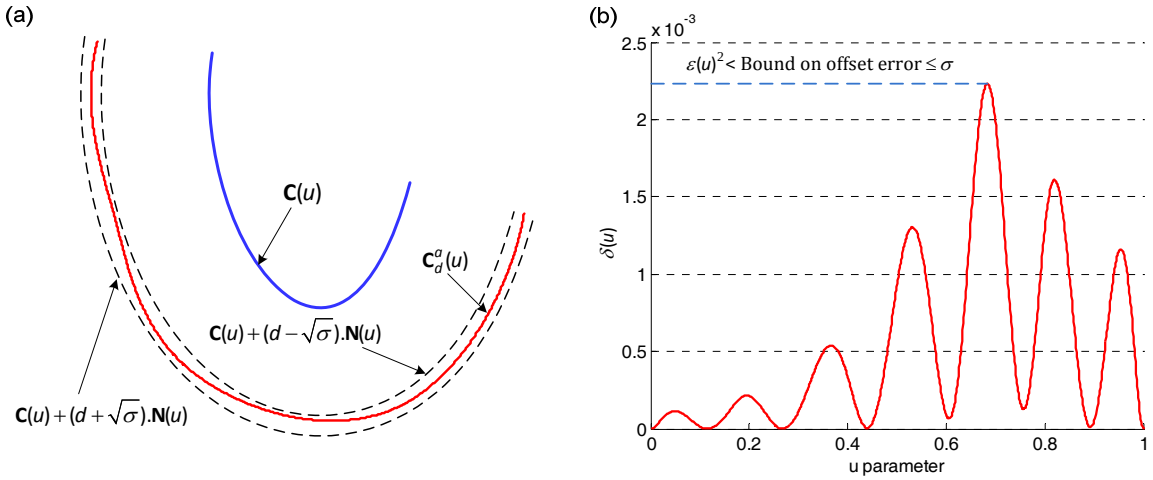
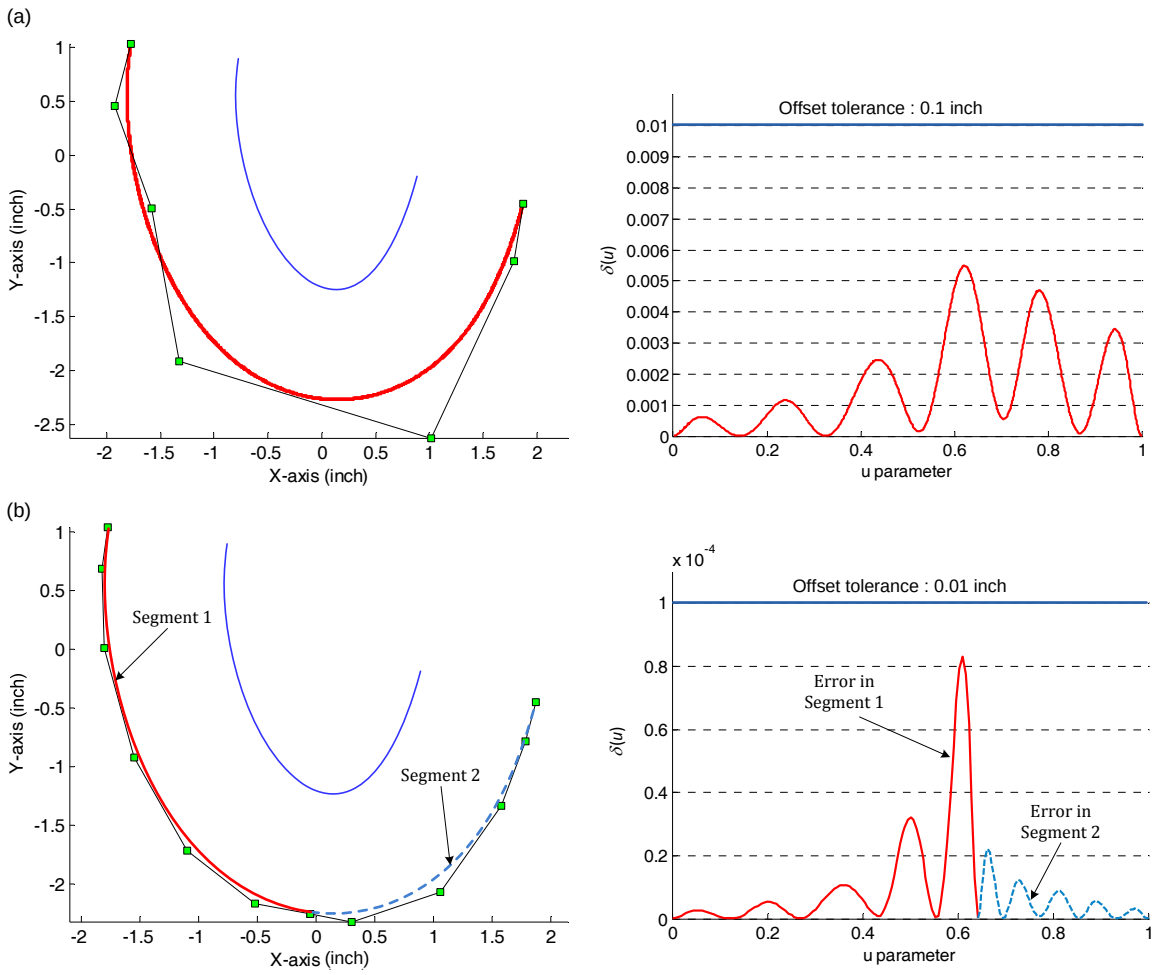


Figure 4.7(a) Approximate offset to cubic Bezier curve with error bound curves; (b) the corresponding error function.



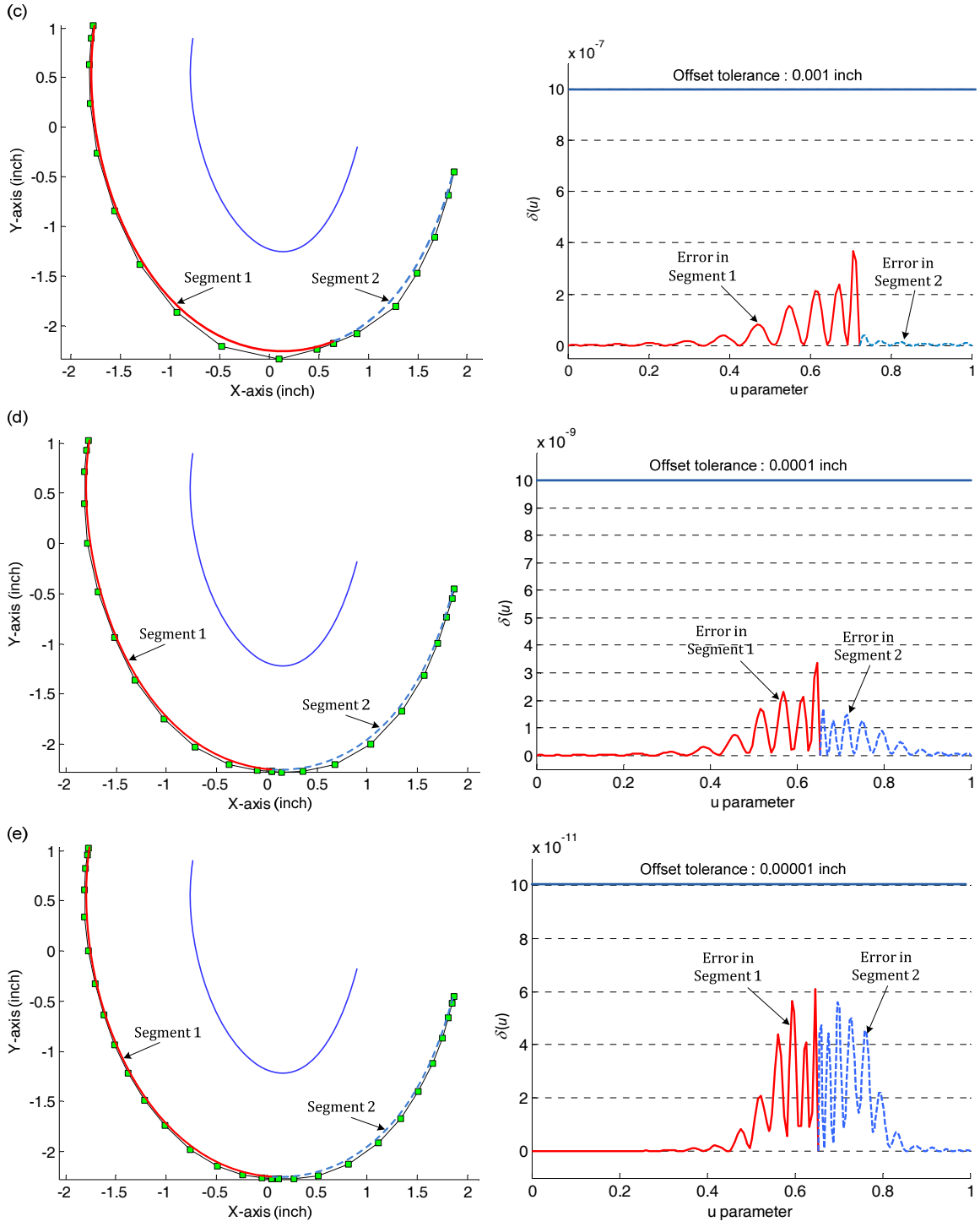


Figure 4.8 Offset approximations to the cubic Bezier curve and the corresponding upper bound functions for different tolerance values.

The last column (proposed method) of Tabel-4.2 shows the number of control points to approximate offset curve of degree 5 for different values of tolerances, and it is clear that the proposed method requires least number of control points as compare to other methods (we have used the same abbreviations, M2 is the method proposed by [37], for the details of methods and symbols see [37]).

Table 4.2: Comparison of the proposed method with current methods for the number of control points.

φ	Cobb	Elb	Elb-2	Til	Lst	Lst-2	M-2	Prop.
10^{-1}	10	11	13	10	7	10	22	**7**
10^{-2}	31	24	25	31	13	19	29	**12**
10^{-3}	94	74	97	97	19	31	43	**18**
10^{-4}	316	216	322	322	31	46	71	**23**
10^{-5}	865	974	769	886	50	88	127	**29**

The second example in Figure 4.9 is a cubic uniform B-spline curve with 7 control points (-3.0162, 2.3414), (-3.9719, 2.2084), (-1.0705, 0.07228), (0.3196, -2.7752), (-0.1528, 2.299), (2.9242, -0.9399) and (2.8027, 3.0278). The approximate offset curves $\mathbf{C}_d^a(u)$ of degree 5 at radius 0.5 are generated for different tolerance values, shown in Figure 4.10(a)-(e) along with deviations $\delta(u)$ from the exact offset.

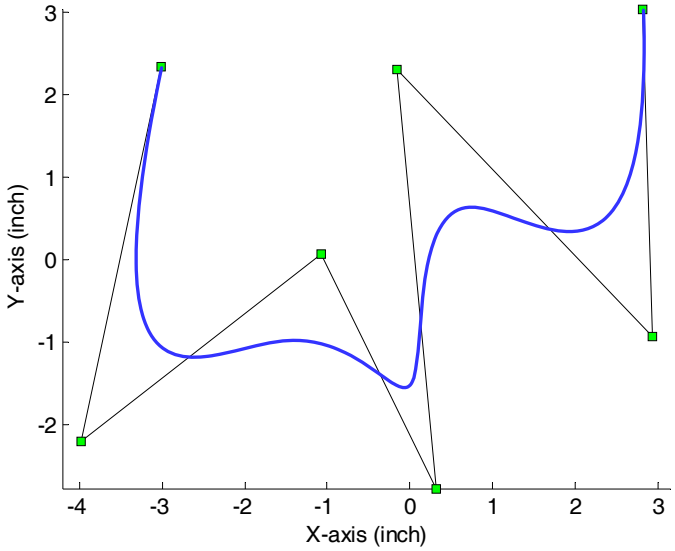
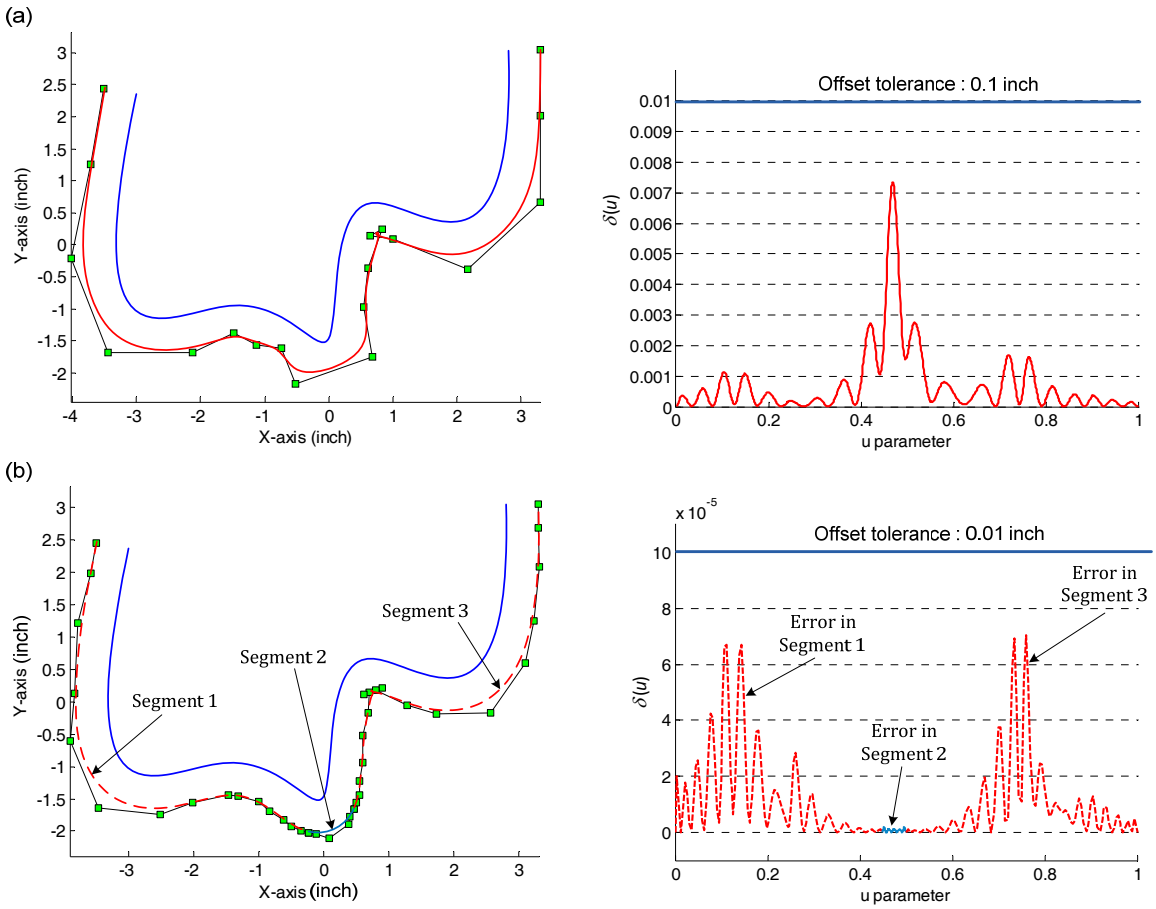


Figure 4.9 Cubic B-spline curve and its control polygon.



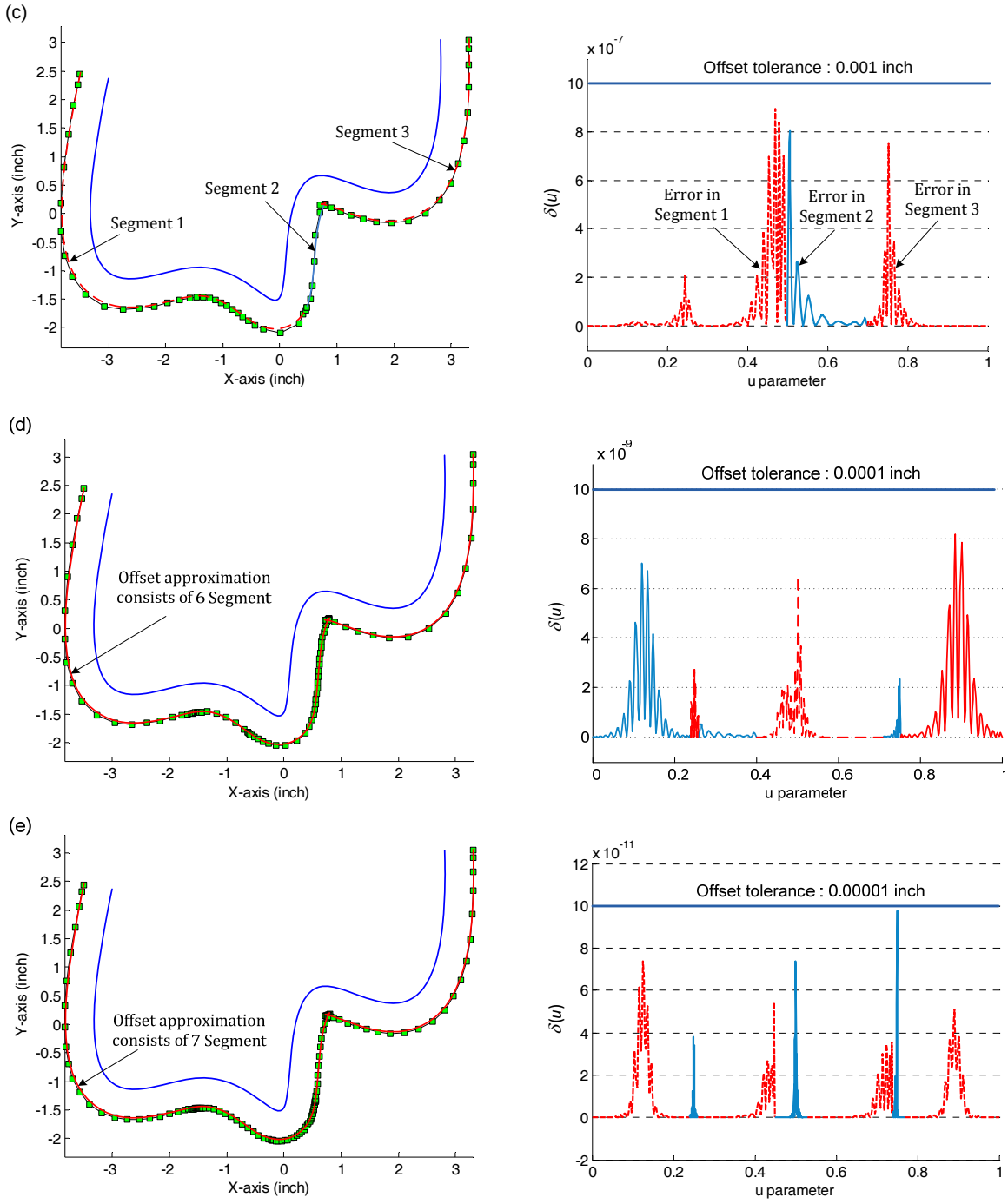


Figure 4.10 (a) Approximate offset to cubic B-spline curve with error bounds; (b) the corresponding error function

For higher tolerance values the curve was divided into segments at proper knot values as discussed in section-3. Table-4.3 shows the comparison of our method (last column) in terms of total number of control points required to meet

the prescribed tolerances for offset approximation to the cubic B-spline curve in Figure 4.9.

Table 4.3: Comparison of the proposed method with current methods for the number of control points.

φ	Cobb	Elb	Elb-2	Til	Lst	Lst-2	M-2	Prop.
10^{-1}	28	19	22	25	16	31	78	**19**
10^{-2}	73	57	55	67	48	49	92	**39**
10^{-3}	208	174	190	202	84	94	120	**65**
10^{-4}	637	417	550	640	138	166	176	**94**
10^{-5}	1846	1357	1690	1918	240	277	302	**146**

4.5 Summary

This work has proposed a new approach to approximating B-spline offsets with curves in the same form, which are appropriate for CNC pocketing. The main feature of this approach is that the offset error is globally bounded during the approximation process. The research contribution is the upper bound function of the offset error function constructed in this work, which can be used to easily estimate the maximum offset error in order to globally control the offset error. Many tests have demonstrated the effectiveness of the upper bound function. The approximate offsets in B-spline form basically have fewer control points, less function degree, and all their offset errors are within the specified tolerance. Therefore, the offsets are appropriate for the use of CNC pocket machining. This new approach has great potential to generate high precision NC tool paths for pocketing.

Chapter 5 Arc-Length Parameterized NURBS Tool-Paths with Global Error Control for Smooth and Accurate Sculptured Surfaces in 3-Axis Machining

5.1 Introduction

Non-uniform rational B-spline (NURBS) tool-paths have already been accepted in industry for machining sculptured surfaces of aerospace, aeronautical, mould, and automotive parts. Compared to free-form surface machining of the conventional linear and circular tool-paths, the advantages of using NURBS tool-paths in surface machining [43, 44] include: (a) the cutters can move faster with better kinematics, (b) the machined surfaces are smoother, and (c) the NC programs are smaller in size. To ensure these advantages, high quality NURBS tool-paths are essential, which should be arc-length accurately parameterized [45, 46], gouging-and-interference completely free, and machining-error globally bounded. However, people often regard that NURBS tool-path generation simply is to fit a NURBS curve to a group of discrete cutter locations. This prevailing opinion is wrong and hinders

research on high quality NURBS tool-path generation. Unfortunately, current tool-path generation methods and commercial CAD/CAM software cannot generate the high quality NURBS tool-paths; as a result, NURBS machining is not widely used in industry. Technically, the current methods of generating a NURBS tool-path are in a similar procedure: (1) finding a group of cutter contact (CC) points on a sculptured surface according to a path planning strategy, (2) calculating the corresponding cutter locations, (3) fitting a NURBS path to these locations to ensure the deviation between them is with a prescribed tolerance, and (4) conducting gouging detection to delete path segments with gouging. Although the methods can work on simple shaped parts, the resulting paths have the problems: (a) the path is accurate to a number of discrete CL points, but not to the theoretical CL path, which means the machining error is not globally bounded; (b) the path is not arc-length parameterized NURBS path, which is required by the NURBS interpolation of the CNC controller; and (c) the detection gouging along a NURBS path is difficult and deleting the NURBS path segments with gouging is troublesome.

To address the problems in order to promote this machining, NURBS tool-paths with the arc-length parameter is proposed, and a new approach is established to ensure the path accuracy in terms of the theoretical smooth CL path, keep good tool kinematics, and effectively eliminate gouging and interference. Several cutting tests have clearly demonstrated that the arc-length parameterized NURBS CL paths generated in this work can cut accurate, smooth sculptured surfaces. The approach can be implemented into current CAD/CAM software to benefit the manufacturing industry.

First, the arc-length parameterized NURBS tool-paths are introduced in Section 5.2. Second, the theoretical cutter location path is defined and samples cutter locations and their arc-lengths are computed in Section 5.3. Third, gouging- and interference detection is applied to the discrete cutter locations to eliminate the defective locations in Section 5.4. Then, a NURBS curve is fit to the sample cutter locations with the parameterization and path errors are globally bounded in Section 5.5. Finally, two practical examples are provided to demonstrate the effectiveness of this new approach in Section 5.6.

5.2 Arc-Length Parameterized NURBS Tool Paths

With development of NURBS interpolation in CNC controllers for CNC machining of accurate, smooth sculptured surfaces, NURBS tool-paths are necessary in order to use this new function. As the core of this emerging high technique, the NURBS interpolation and the NURBS tool-path generation have not been well-established yet and are under extensive research in academia and industry. In machining, the NURBS tool-path is input to the interpolation algorithm as a reference; and it is expected that the controller guides the tool in real-time moving along a NURBS trajectory with the prescribed feed rate, thus, they are closely related. It is well known that, if the NURBS tool-path is arc-length parameterized, the interpolation algorithm can easily generate accurate tool trajectory complying with the path and ensure good tool kinematics. Although many researchers are developing advanced NUBRS interpolation algorithms to handle non-linear NURBS tool-paths with a unit free parameter real-time, it is important and feasible to

generate accurate NURBS paths with the arc-length parameter prior to machining. This type of tool-path is an efficient solution to the problems of the NURBS interpolation.

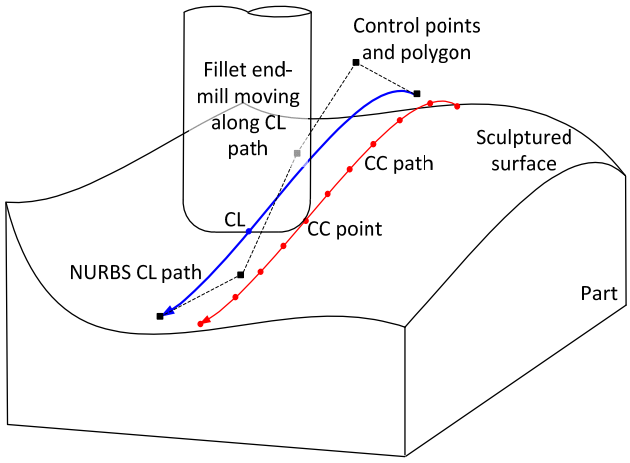


Figure 5.1 Illustration of a smooth NURBS CL path for machining a sculptured surface.

Mathematically, arc-length parameterized NURBS paths and general NURBS curves are formulated with similar parametric equations; however, they are essentially different. The NURBS curves are a new type of curve model in computer-aided design to represent free-form geometry. A general NURBS curve can be attained by fitting a group of points with the least square method; by taking the chord lengths as the point parameters, an approximate arc-length NURBS curve can be found. The major concern is the deviations between the points and the curve should be less than a tolerance. A NURBS path can be generated by fitting a group of cutter locations of a theoretical path. Compared to the general NURBS curves, the major difference of the NURBS paths includes that (1) the paths should be gouging free, and (2) the overall deviations between the NURBS path and the theoretical CL

path should be with the prescribed tolerance. Unfortunately, the existent approach to the approximate arc-length NURBS curve cannot be used for arc-length parameterized NURBS tool-paths.

As a new type of NC tool-path, the arc-length parameterized NURBS cutter location (CL) path is better than the conventional linear cutter location path in free-form curve and surface machining. Since the NURBS CL path is represented with the NURBS equation, and the linear CL path is represented with the polygon of a group of discrete cutter locations. The form of the NURBS CL path is simpler than that of the linear CL path. The other advantages of the NURBS CL paths include that (1) the NURBS CL path is smoother than the linear CL path, and (2) the tool kinematics of cutting along the NURBS CL path in high feed rate is better than that along the linear CL path. To generate CL paths for a surface milling, first, a group of discrete points on the surface are calculated as cutter contact (CC) points, according to a tool-path planning strategy, such as the parallel-plane, the iso-cusp and the steepest ascending methods (see Figure 5.1). In the conventional CNC machining with the G1 linear interpolation, the polygon of the CC points (called CC polygon) that is not on the part surface is used to approximate the theoretical, smooth path of cutter contact points (called CC path) that is on the part surface. Second, based on the CC points, the corresponding CLs are computed and their polygon is used as the path of cutter locations (called CL path), which is illustrated in Figure 5.2. Then, simulation for the tool cutting the surface along the CL path is conducted to detect possible tool-and-part gouging and interference, and the CLs causing the defects are deleted from the CL path. Finally, the CL path is fed into the CNC controller of the machine

tool to be used. This is the main steps of the conventional tool-path generation method.

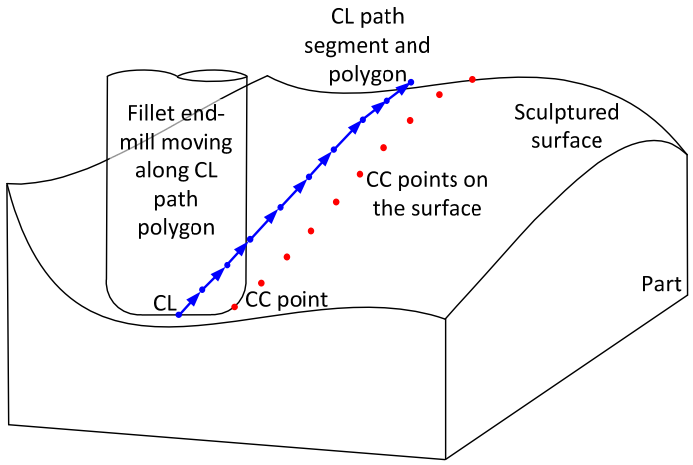


Figure 5.2 Illustration of a conventional linear (G1) CL path represented with a polygon.

Suppose a whole NURBS CL path is calculated using a conventional way and represented with one equation. If a segment of the path can cause gouging and/or interference, the path has to be decomposed to remove the invalid segment. This would be complicated and require a lot of un-necessary effort. To address the problems in this work, prior to calculating arc-length parameterized NURBS tool-paths, gouging and interference detection is conducted with an optimization model. To ensure the path in accordance with the theoretical CL path and its parameterization unity, an optimization model for error globally bounded is established. By using this method, the NURBS path accuracy can be increased and the number of control points can be reduced.

5.3 Theoretical Cutter Location Paths and Their Points Arc-Length Calculation

5.3.1 Theoretical cutter contact paths on sculptured surfaces

As a critical model of free-form surface in CAD/CAM software, NURBS surface type is often used to design sculptured parts, and its simple form, B-spline surface, is defined [60] as

$$\mathbf{S}(u,v)=\begin{bmatrix} x_{\mathrm{S}}(u,v) \\ y_{\mathrm{S}}(u,v) \\ z_{\mathrm{S}}(u,v) \end{bmatrix} := \sum_{i=0}^{\mathrm{M}_u}\sum_{j=0}^{\mathrm{M}_v}\left[N_{i,\mathrm{K}}(u)\cdot N_{j,\mathrm{L}}(v)\cdot \mathbf{P}_{i,j} \right];\ u\in[u_{\min},u_{\max}], v\in[v_{\min},v_{\max}] \quad (5.1)$$

where $\mathbf{P}_{i,j}$ are control points, u and v are parameters, and $N_{i,\mathrm{K}}(u)$ and $N_{j,\mathrm{L}}(v)$ are blending functions with orders of K and L, respectively. The first derivatives of the surface in terms of u and v are $\frac{\partial \mathbf{S}(u,v)}{\partial u}=\left[\frac{\partial x_{\mathrm{S}}(u,v)}{\partial u} \quad \frac{\partial y_{\mathrm{S}}(u,v)}{\partial u} \quad \frac{\partial z_{\mathrm{S}}(u,v)}{\partial u}\right]^{\mathrm{T}}$ and $\frac{\partial \mathbf{S}(u,v)}{\partial v}=\left[\frac{\partial x_{\mathrm{S}}(u,v)}{\partial v} \quad \frac{\partial y_{\mathrm{S}}(u,v)}{\partial v} \quad \frac{\partial z_{\mathrm{S}}(u,v)}{\partial v}\right]^{\mathrm{T}}$, respectively. According to the differential geometry theory, the components of the first fundamental matrix of the surface are defined as $\mathrm{E}:=\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathrm{T}}\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]$, $\mathrm{F}:=\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathrm{T}}\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]$, and $\mathrm{G}:=\left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]^{\mathrm{T}}\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]$. Thus, the first derivatives of E, F and G in terms of u are

$$\frac{\partial \mathrm{E}}{\partial u}=2\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathrm{T}}\cdot\left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u^2}\right] \quad (5.2)$$

$$\frac{\partial \mathsf{F}}{\partial u} = \left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u^2}\right] + \left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u \cdot \partial v}\right] \tag{5.3}$$

and

$$\frac{\partial \mathsf{G}}{\partial u} = 2 \cdot \left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u \cdot \partial v}\right] \tag{5.4}$$

respectively. The first derivatives of E, F and G in terms of *v* are

$$\frac{\partial \mathsf{E}}{\partial v} = 2 \cdot \left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u \cdot \partial v}\right] \tag{5.5}$$

$$\frac{\partial \mathsf{F}}{\partial v} = \left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u \cdot \partial v}\right] + \left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial v^2}\right] \tag{5.6}$$

and

$$\frac{\partial \mathsf{G}}{\partial v} = 2 \cdot \left[\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]^{\mathsf{T}} \cdot \left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial v^2}\right] \tag{5.7}$$

respectively.

By applying a tool-path planning strategy, number of M_{CC} CC points, $\mathbf{CC}_1$, $\mathbf{CC}_2$, ..., and $\mathbf{CC}_{\mathsf{M}_{\mathsf{CC}}}$, of the surface can be calculated for each tool-path as

$$\mathbf{CC}_1 := \mathbf{S}(u_1, v_1),\ \mathbf{CC}_2 := \mathbf{S}(u_2, v_2),\ \text{...,}\ \text{and}\ \mathbf{S}_{\mathsf{M}_{\mathsf{CC}}} := \mathbf{S}\left(u_{\mathsf{M}_{\mathsf{CC}}}, v_{\mathsf{M}_{\mathsf{CC}}}\right).$$

A polygon is constructed by sequentially passing the CC points, which is often called their CC path in practice. By definition, the theoretical CC path, $\mathbf{CC}(t)$, is a smooth curve with a parameter *t*, passing through the CC points and exactly lying on the surface (see Figure 5.3(a)). Currently, the CC path is used to approximate the theoretical CC path for CNC machining with the G1 interpolation. However, the theoretical CC path is indispensible in computing NURBS tool-paths.

Given the CC points, the theoretical CC path $\mathbf{CC}(t)$ can be easily found. First, a polynomial with a parameter *t* is fit to the CC points in the parametric space *u*-*v* of the surface. (see Figure 5.3(b)) The coordinates of the CC points $\overline{\mathbf{CC}}_1$, $\overline{\mathbf{CC}}_2$, …, $\overline{\mathbf{CC}}_{M_{CC}}$ are $[u_1, v_1]$, $[u_2, v_2]$, …, $\left[u_{M_{CC}}, v_{M_{CC}}\right]$ in the parametric space, respectively. By using the least squares method, a polynomial with a parametric *t*, e.g., a cubic B-spline curve, can be found as

$$\overline{\mathbf{CC}}(t) := \begin{bmatrix} u(t) \\ v(t) \end{bmatrix}, \quad t \in [t_{\min}, t_{\max}] \tag{5.8}$$

The first derivative of this curve is $\frac{d\overline{\mathbf{CC}}(t)}{dt} = \left[\frac{du(t)}{dt}, \frac{dv(t)}{dt}\right]^{\mathrm{T}}$. By substituting Eq. (5.8) into Eq. (5.1), the theoretical CC path can be formulated as

$$\mathbf{CC}(t) = \begin{bmatrix} x_{CC}(t) \\ y_{CC}(t) \\ z_{CC}(t) \end{bmatrix} = \begin{bmatrix} x_S(u(t), v(t)) \\ y_S(u(t), v(t)) \\ z_S(u(t), v(t)) \end{bmatrix}, \quad t \in [t_{\min}, t_{\max}] \tag{5.9}$$

With change of parameter *t*, all CC points of the theoretical CC path can be calculated. The CC path is necessary to find the theoretical CL path.

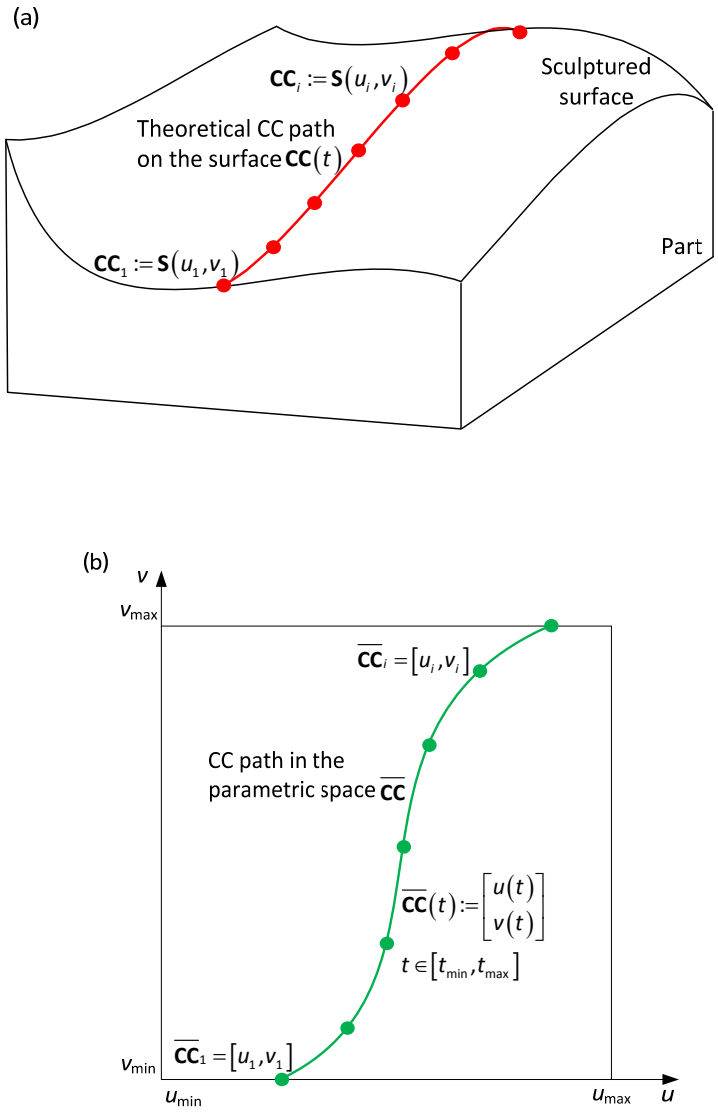


Figure 5.3 (a) A group of CC points determined with a machining strategy and a theoretical CC path on the sculptured surface and (b) the points in the parametric space corresponding to the CC points and a curve fit to these points.

5.3.2 Formula of the theoretical cutter location paths for fillet end-mills

Since fillet (or bull-nose) end-mill is in a generic shape of end-mills, the theoretical CL path and its arc-length equations are derived here for this type of

cutter, which can also be applied to ball and flat end-mills directly. According to Eq. (5.9) of the theoretical CC path $\mathbf{CC}(t)$, its corresponding theoretical CL path for a fillet end-mill is formulated, and some equations are prepared for estimating arc-length of this CL path afterward. In this work, the end-mill's radius is denoted as R_T and its fillet (or corner) radius as R_F. Based on the sculptured surface equation, Eq. (5.1), its unit normal vector is represented as

$$\mathbf{n}(u,v)=\begin{bmatrix} x_n(u,v) \\ y_n(u,v) \\ z_n(u,v) \end{bmatrix}=\left(E\cdot G-F^2\right)^{-\frac{1}{2}}\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}\right] \tag{5.10}$$

Then, the equations of the first derivatives of the unit normal vector in terms of u and v are

$$\begin{aligned}\frac{\partial \mathbf{n}(u,v)}{\partial u}=&\left(E\cdot G-F^2\right)^{-\frac{1}{2}}\cdot\left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u^2}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}+\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial^2 \mathbf{S}(u,v)}{\partial u\cdot\partial v}\right]-\\ &\frac{\left(E\cdot G-F^2\right)^{-\frac{3}{2}}}{2}\cdot\left(G\cdot\frac{\partial E}{\partial u}+E\cdot\frac{\partial G}{\partial u}-2\cdot F\cdot\frac{\partial F}{\partial u}\right)\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]\end{aligned} \tag{5.11}$$

and

$$\begin{aligned}\frac{\partial \mathbf{n}(u,v)}{\partial v}=&\left(E\cdot G-F^2\right)^{-\frac{1}{2}}\cdot\left[\frac{\partial^2 \mathbf{S}(u,v)}{\partial u\partial v}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}+\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial^2 \mathbf{S}(u,v)}{\partial v^2}\right]-\\ &\frac{\left(E\cdot G-F^2\right)^{-\frac{3}{2}}}{2}\cdot\left(G\cdot\frac{\partial E}{\partial v}+E\cdot\frac{\partial G}{\partial v}-2\cdot F\cdot\frac{\partial F}{\partial v}\right)\cdot\left[\frac{\partial \mathbf{S}(u,v)}{\partial u}\times\frac{\partial \mathbf{S}(u,v)}{\partial v}\right]\end{aligned} \tag{5.12}$$

respectively. It is evident that the unit vector of the tool axis $\mathbf{a}_1$ is $\mathbf{a}_1 := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$. Besides, a constant matrix $\mathbf{a}_2$ is set as

$$\mathbf{a}_2 := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and a vector $\mathbf{b}(u,v)$ used to compute the CL point is defined as

$$\mathbf{b}(u,v) = \begin{bmatrix} x_b(u,v) \\ y_b(u,v) \\ z_b(u,v) \end{bmatrix} := \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \mathbf{n}\right)^{-\frac{1}{2}} \left(\mathbf{a}_2 \cdot \mathbf{n}\right) \tag{5.13}$$

The first derivatives of $\mathbf{b}(u,v)$ in terms of u and v can be formulated as

$$\frac{\partial \mathbf{b}(u,v)}{\partial u} = \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \mathbf{n}\right)^{-\frac{1}{2}} \left(\mathbf{a}_2 \cdot \frac{\partial \mathbf{n}}{\partial u}\right) - \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \mathbf{n}\right)^{-\frac{3}{2}} \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \frac{\partial \mathbf{n}}{\partial u}\right) \left(\mathbf{a}_2 \cdot \mathbf{n}\right) \tag{5.14}$$

and

$$\frac{\partial \mathbf{b}(u,v)}{\partial v} = \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \mathbf{n}\right)^{-\frac{1}{2}} \left(\mathbf{a}_2 \cdot \frac{\partial \mathbf{n}}{\partial v}\right) - \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \mathbf{n}\right)^{-\frac{3}{2}} \left(\mathbf{n}^T \cdot \mathbf{a}_2 \cdot \frac{\partial \mathbf{n}}{\partial v}\right) \left(\mathbf{a}_2 \cdot \mathbf{n}\right) \tag{5.15}$$

respectively. Therefore, the equation of calculating CLs for the fillet end-mill is derived as

$$\mathbf{CL}(u,v) = \begin{bmatrix} x_{CL}(u,v) \\ y_{CL}(u,v) \\ z_{CL}(u,v) \end{bmatrix} := \mathbf{CC}(u,v) + R_F \cdot \left[\mathbf{n}(u,v) - \mathbf{a}_1\right] + \left(R_T - R_F\right) \cdot \mathbf{b}(u,v) \tag{5.16}$$

Meanwhile, the equations of the first derivatives of $\mathbf{CL}(u,v)$ in terms of u and v are found as

$$\frac{\partial \mathbf{CL}(u,v)}{\partial u} = \frac{\partial \mathbf{S}(u,v)}{\partial u} + \mathrm{R_F} \cdot \frac{\partial \mathbf{n}(u,v)}{\partial u} + (\mathrm{R_T} - \mathrm{R_F}) \cdot \frac{\partial \mathbf{b}(u,v)}{\partial u} \tag{5.17}$$

and

$$\frac{\partial \mathbf{CL}(u,v)}{\partial v} = \frac{\partial \mathbf{S}(u,v)}{\partial v} + \mathrm{R_F} \cdot \frac{\partial \mathbf{n}(u,v)}{\partial v} + (\mathrm{R_T} - \mathrm{R_F}) \cdot \frac{\partial \mathbf{b}(u,v)}{\partial v} \tag{5.18}$$

respectively.

According to Eqs. (5.8) and (5.16), the formula of the theoretical CL path is attained as

$$\mathbf{CL}(t) := \begin{bmatrix} x_{\mathrm{CL}}(u(t),v(t)) \\ y_{\mathrm{CL}}(u(t),v(t)) \\ z_{\mathrm{CL}}(u(t),v(t)) \end{bmatrix},\ t \in [t_{\min}, t_{\max}] \tag{5.19}$$

Eqs. (5.11), (5.12), (5.14), (5.15), (5.17) and (5.18) will be used to calculate the arc-length of any point on the theoretical CL path in the next section. This path is a critical reference for generating its NURBS representation because it can better control the NURBS accuracy and smoothness. Currently, people select a group of discrete points and fit a NURBS path to these points. In this way, the NURBS path around the sample points are ensured; however, the path between these points could wiggle around these points, resulting an un-smooth path.

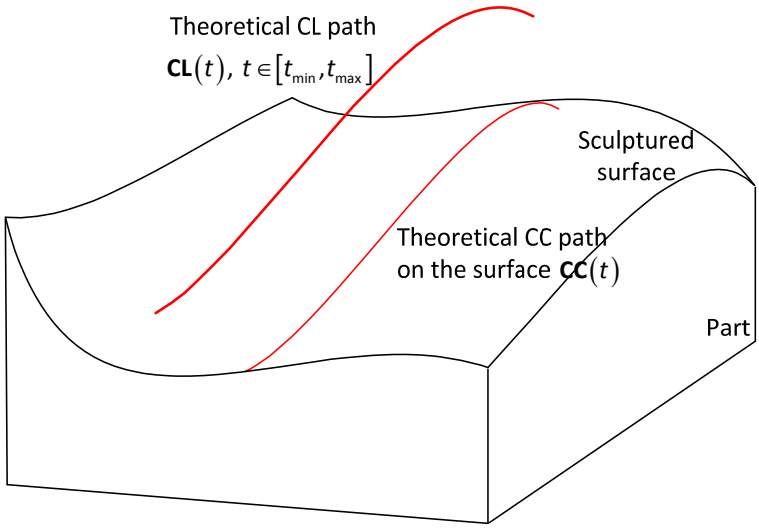


Figure 5.4 A theoretical CL path determined with a theoretical CC path on the sculptured surface.

5.3.3 Cutter locations sampling and their arc-lengths calculation

In this work, an initial NURBS CL path is generated using the conventional way, which is to sample a group of CLs of the theoretical CL path, and then fit a NURBS curve to these CLs using the least squares method. By using Eq. (5.19), number of M_{CL} CLs can be sampled along the theoretical CL path, $\mathbf{CL}_1 := \mathbf{CL}(u(t_1), v(t_1))$, $\mathbf{CL}_2 := \mathbf{CL}(u(t_2), v(t_2))$, ..., and $\mathbf{CL}_{M_{CL}} := \mathbf{CL}(u(t_{M_{CL}}), v(t_{M_{CL}}))$ (see Figure 5.5). In order to generate an arc-length parameterized path, it is necessary to calculate the true arc-lengths of the samples. Thus, the equations of the first derivatives of the theoretical CL path in Eq. (5.19) in terms of *u* and *v* have to be found as follows.

$$\frac{d\mathbf{CL}(t)}{dt} = \frac{\partial \mathbf{CL}(u(t), v(t))}{\partial u} \cdot \frac{du(t)}{dt} + \frac{\partial \mathbf{CL}(u(t), v(t))}{\partial v} \cdot \frac{dv(t)}{dt} \tag{5.20}$$

where $\frac{\partial \mathbf{CL}(u,v)}{\partial u}$ and $\frac{\partial \mathbf{CL}(u,v)}{\partial v}$ have been found in Eqs. (5.17) and (5.18); $\frac{du(t)}{dt}$ and $\frac{dv(t)}{dt}$ using Eq. (5.8). Hence, the arc-length of a CL point with parameter t can be calculated as

$$l(t) = \int_{t_{\min}}^{t} \left| \frac{d\mathbf{CL}(t)}{dt} \right| \cdot dt,\ t \in [t_{\min}, t_{\max}] \tag{5.21}$$

To calculate true arc-lengths of the sample CLs, the adaptive Simpson quadrature method is applied to Eq. (5.21). After the sample CLs and their arc-lengths are computed, gouging and interference detection should be conducted at these CLs in order to eliminate defective CLs and re-group sample CLs, before an initial NURBS CL path is fit to each group of valid sample CLs.

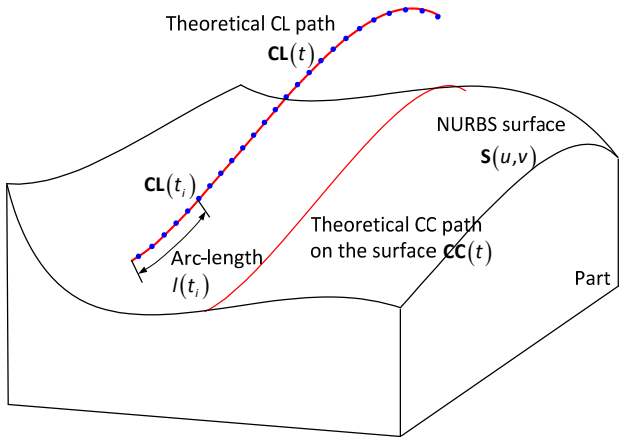


Figure 5.5 Sampling a group of CLs of the theoretical CL path and calculating their arc-lengths.

5.4 Gouging-and-Interference Detection for Valid Sample Cutter Locations

In sculptured surfaces machining, the cutter is prone to gouge and interfere with the surfaces due to their complex geometric shape. Figures 5.6(a) and 5.6(b) show a tool gouging and interfering with a part, respectively. Thus, it is important to conduct gouging and interference check in high accuracy. Suppose the NURBS path is planned in the same way as G1 CL path planning: (1) generating a whole path, and (2) delete the path segments causing gouging after machining simulation. In the NURBS machining of sculptured surfaces, if the cutter overcuts the surfaces while moving along the NURBS path in simulation, the NURBS path has to be decomposed at the CL points between gouging and non-gouging segments, and the segments with gouging or interference have to be cut off and removed. This is troublesome and time consuming. An example is rendered in Figure 5.7. To address this problem, a new approach different from the conventional way is proposed. In this work, a number of CC points of the theoretical CC path are sampled and their CL points are calculated. Then, gouging and interference detection at these CL points are conducted; the defected CL points are eliminated, and the valid consecutive CL points are grouped. Finally, each group of CL points is fit with a NURBS path, which is free of gouging and interference.

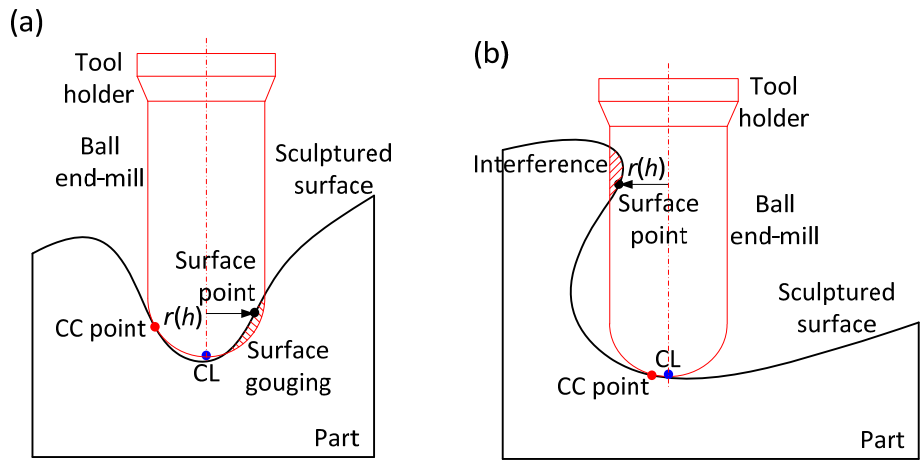


Figure 5.6 (a) The tool is locally gouging the part surface while cutting at a CC point and (b) the tool is globally interfering with the part.

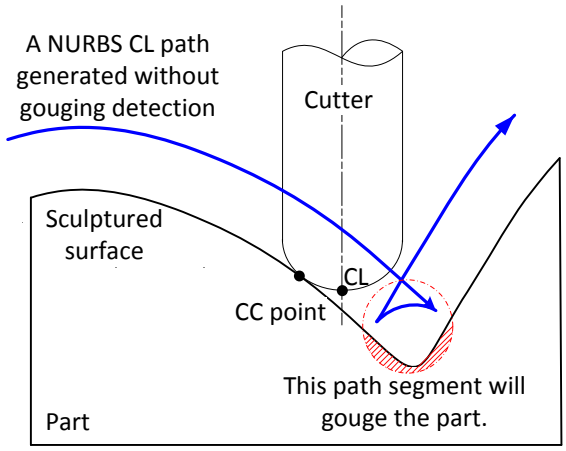


Figure 5.7 A NURBS CL path with a segment that will cause gouging on the sculptured surface.

5.4.1 A general optimization model of gouging and interference detection

To detect the cutter gouging and interference with the part's surfaces including the sculptured surface and its surroundings, computer graphics methods are the dominate tool in current CAD/CAM software. Although they are continuously improved by taking advantage of the latest computer graphics techniques, their methodology is mainly based on simple, arduous geometric operations, such as Boolean operations and polygon intersection, without much

intelligence. With ever increasing demand for higher part precision in industry, a perceivable, common weakness of them is that it often takes a long time for them to conduct the detection; they could cause memory overflow, freezing the computer. To find an effective solution, a new, general optimization model of gouging-and-interference detection is established in this section, which is more precise, efficient and completed.

Specifically, in this completed model, the geometry of the whole tooling system and the part geometry are considered; the tooling system includes the APT cutter's cutting edges, cutter shank and the tool holder, meanwhile, the part includes the sculptured surface and its surroundings. The APT cutter is with the generic shape of flat, ball and fillet end-mills. Thus, the profile of the tooling system is constructed with six segments, A-B (the bottom cutting edge), B-C (the fillet), C-D (the side cutting edge), D-E (the side cutting edge and the shank), E-F (the tool holder) and F-G (the tool holder). In Figure 5.8, the profile with the parameters as α, β, γ, R_F, R_T, R_H, L_T, L_S and L_H are defined in the cutter coordinate system $r-O-h$. Ball, flat and fillet end-mills are special cases of the APT cutter. For ball end-mills, $\alpha=\beta=0$ and $R_F=R_T$; for flat end-mills, $\alpha=\beta=0$ and $R_F=0$; and for fillet end-mills, $\alpha=\beta=0$ and $0<R_F<R_T$. For a general APT cutter, $\alpha\neq\beta\neq 0$; thus, AB and CD can be found as

$$AB=\frac{1}{\cos(\alpha+\beta)}\cdot\left[R_F\cdot\sin(\alpha+\beta)+R_T\cdot\cos\beta-R_F-L_T\cdot\sin\beta\right]$$

and

$$CD = \frac{1}{\cos(\alpha+\beta)} \cdot \left[R_F \cdot \sin(\alpha+\beta) - R_T \cdot \sin\alpha - R_F + L_T \cdot \cos\alpha \right]$$

respectively. The heights of the turning points of the profile can be found accordingly. $H_B = AB \cdot \sin\alpha$, $H_C = L_T - L_S - CD \cdot \cos\beta$, $H_D = L_T - L_S$, $H_E = L_T$, $H_F = H_E + \frac{R_H - R_T}{\tan\gamma}$ and $H_G = L_T + L_H$. The relationship between the height and the radius of the profile point is represented as

$$r(h) = \begin{cases} 0, & (h < 0) \\ \dfrac{h}{\tan\alpha}, & (0 \le h < H_B) \\ AB \cdot \cos\alpha - R_F \cdot \sin\alpha + \sqrt{R_F^2 - (R_F \cos\alpha - h + AB \cdot \sin\alpha)^2}, & (H_B \le h < H_C) \\ R_T - (L_T - L_S - h) \cdot \tan\beta, & (H_C \le h < H_D) \\ R_T, & (H_D \le h < H_E) \\ (h - L_T) \cdot \tan\gamma + R_T, & (H_E \le h < H_F) \\ R_H, & (H_F \le h \le H_G) \\ 0, & (H_G < h) \end{cases} \quad (5.22)$$

This relationship is used in gouging-and-interference detection to check whether or not a point under inspection is within the tooling system while machining. Compared to the existent methods only considering the cutting section of a tool without the tool shank and holder, this tooling system is completed in geometry.

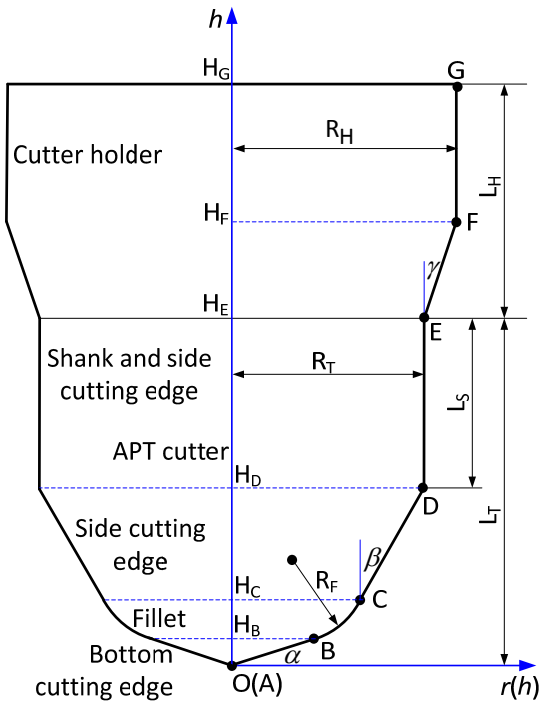


Figure 5.8 The geometric model of the tooling system including the APT cutter and the tool holder.

Geometrically, when the cutter gouges or collides with the part surfaces at a cutter location, at least one point on the part is within the revolving surface of the profile modeled above (see Figure 5.8). In the other words, the distance between this point and the cutter axis is less than the profile radius at the height of the point. Figure 5.6 demonstrates the geometric relationship between the cutter and the part in gouging and interference. Thus, the optimization model for gouging and interference detection is constructed to minimize the objective function of the distances between the cutter axis and the part points minus their corresponding cutter profile radii. Suppose the tip A of the APT cutter is at a CL in 3-axis milling and the coordinate of the CL point is $\left[x_{CL}, y_{CL}, z_{CL}\right]^T$. For any point $\left[x, y, z\right]^T$ on the part surfaces $\left\{\mathbf{S}, \mathbf{S}_1, \cdots \mathbf{S}_i, \cdots\right\}$, the objective function is represented as

$$\text{Minimize } f = \left[x - x_{CL}\right]^2 + \left[y - y_{CL}\right]^2 - \left[r\left(z - z_{CL}\right)\right]^2 \tag{5.23}$$

and

$$\text{Subject to} \left[x, y, z\right]^T \in \left\{\mathbf{S}, \mathbf{S}_1, \cdots \mathbf{S}_i, \cdots\right\}$$

If the minimum of f is greater than zero, the surface is not within the volume of the tooling system, and the tool will not gouge and interfere with the part. However, if the minimum is negative, the cutter will overcut the part and damage the part. Since this is a global optimization problem, solving it using the conventional method takes a long time. In this work, the hybrid optimization method is employed, and the result is very satisfactory.

5.4.2 Hybrid optimization method

To solve global optimization problems, genetic algorithm (GA), simulate annealing (SA), ant colony, differential evolution and particle swarm optimization (PSO) are often used. Among them, the PSO method is more popular for its simple implementation and quick convergence within good accuracy. However, it can take a long time for the PSO to pinpoint the global optimum, if the required accuracy is very high. On the contrary, the Newton gradient method, a local optimization solver, is able to quickly find the local optimum solution, but it is unable to find the global solution. To increase computing efficiency, Chen and Liu [19] proposed the hybrid optimization method that adopts the PSO in the rough search stage and the Newton gradient method in the fine search stage, in order to take the advantages of the

global and local optimization methods. In this work, this method is further improved by using the discrete PSO method.

To improve the hybrid optimization method, in the rough search stage, the continuous problem domain is discretized, and the discrete PSO method is then applied. Compared to searching in the continuous domain, the solver can find the pseudo optimum in the discrete domain more efficiently. With the pseudo optimum as the initial point, the Newton gradient method can accurately pinpoint in a fraction of second. Figure 5.9 shows an example of using the improved hybrid optimization method. In this example, a NURBS surface with several local maximums is used. The new and the PSO methods are applied to find the global maximum value of the surface, and the results are listed Table 5.1. It is evident that the improved hybrid method is much more efficient in solving this problem with the same accuracy, compared to the PSO method. Therefore, the improved hybrid method is employed to the optimization model of gouging-and-interference detection.

Table5.1 Comparison between the improved hybrid PSO optimization and the PSO method

	PSO	Improved hybrid optimization method			
	Test 1	Test 2		Test 3	
		Discrete PSO method	Gradient method	Discrete PSO method	Gradient method
Accuracy	10^{-4}	10^{-2}	10^{-4}	10^{-1}	10^{-4}
Iteration #	109	77	2	34	4
Time (s)	5.2513	3.7044	0.1927	1.6509	0.3991
Total Time (s)	5.2513	3.8972		2.0501	
Improvement		25.78%		60.96%	

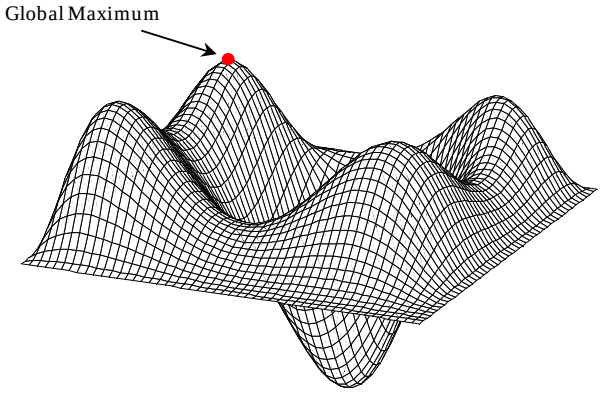


Figure 5.9 Estimation of the global minimum for a NURBS surface with several local minimums.

For a theoretical CL path, a group of CLs are selected. By applying the hybrid optimization method to the gouging-and-interference model at these CLs, the CLs are inspected for validity. Meanwhile, the critical CL between a valid and an invalid CL is calculated as well. Then, the invalid CLs are deleted, and the valid CL points are re-grouped for generating piecewise NURBS tool paths in the next section.

5.5 NURBS CL Path with the Arc-Length Parameterization

For a group of gouging-free CL points, by using the least squares method [1], a NURBS curve can be found as

$$\overline{\mathbf{CL}}(l)=\begin{bmatrix} x_{\overline{\mathrm{CL}}}(l) \\ y_{\overline{\mathrm{CL}}}(l) \\ z_{\overline{\mathrm{CL}}}(l) \end{bmatrix}=\sum_{i=0}^{n_l}\left[N_{i,k}(l)\cdot\mathbf{P}_i\right],\ l\in\left[l_{\min},l_{\max}\right] \tag{5.24}$$

where parameter *l* is the arc length. In the fitting process, the deviation between the NURBS CL path and the theoretical CL path should be less than a prescribed tolerance t_{path} for high accuracy. As a feature of arc-length parameterized NURBS paths, the first derivative of the path in terms of the arc-length parameter should be unity. Hence, the difference between the magnitude of the first derivative and one should be reduced within a prescribed tolerance t_{para} in the iterative fitting for high accuracy. Ideally, it is necessary to ensure that the paths deviation and the first derivative non-unity at every point of the paths are less than the tolerances, which means the errors are globally bounded. However, it is not practical. An alternative way of generating the CL path with error globally bounded in this work is to find the maximum error with the global optimization method and reduce the error within the tolerance.

5.5.1 Fitting with parameterization error globally bounded

In theory, a genuine arc-length parameterized NURBS CL path is characterized with the unity of the first derivative in terms of the arc-length parameter, which is represented as

$$\left\| \frac{d}{dl} \overline{\mathbf{CL}}(l) \right\| = 1 \tag{5.25}$$

However, the NURBS path cannot meet the characteristic equation at the beginning of fitting the CL points. The deviation between the first derivative and one is called the parameterization error, which is formulated as

$$\varepsilon_{\text{para}} = \left\| \frac{d}{dl} \overline{\mathbf{CL}}(l) \right\| - 1 \tag{5.26}$$

In the iterative fitting process, the NURBS path is modified by changing the number of control points or/and the base function order in order to reduce the parameterization error, until it is less than a prescribed tolerance t_{para}. So, this criterion is represented as

$$1 - \mathrm{t}_{\text{para}} \le \left\| \frac{d}{dl} \overline{\mathbf{CL}}(l) \right\| \le 1 + \mathrm{t}_{\text{para}} \tag{5.27}$$

After an arc-length parameterized CL path is generated, all points of the path should be checked against the criterion. If all points are qualified, this CL path is truly parameterized with the arc-length. Otherwise, the path is improved in a new round of fitting. However, it is not practical to check all points. For this purpose, the hybrid

optimization method is used, which is effective to find the maximum parameterization error along the path. Thus, the maximum error should be less than the tolerance t_{para}.

5.5.2 Path error bounded fitting

For an arc-length parameterized NURBS CL path $\overline{\mathbf{CL}}(l)$ generated above, the deviation of the path $\overline{\mathbf{CL}}(l)$ compared to the theoretical CL path $\mathbf{CL}(t)$ refers to the path error, and it is important that the path error is within a prescribed tolerance t_{path}. Mathematically, the path error is defined here. At a point $\left[x_{CL}(t_0), y_{CL}(t_0), z_{CL}(t_0)\right]^T$ of the theoretical NURBS path $\mathbf{CL}(t)$, the tangent vector of the path is $\left[dx_{CL}(t_0)/dt, dy_{CL}(t_0)/dt, dz_{CL}(t_0)/dt\right]^T$. The equation of a plane perpendicular to the tangent vector and passing through the point is

$$\begin{bmatrix} x & y & z \end{bmatrix} \cdot \begin{bmatrix} \dfrac{dx_{CL}(t_0)}{dt} \\ \dfrac{dy_{CL}(t_0)}{dt} \\ \dfrac{dz_{CL}(t_0)}{dt} \end{bmatrix} = \begin{bmatrix} x_{CL}(t_0) & y_{CL}(t_0) & z_{CL}(t_0) \end{bmatrix} \cdot \begin{bmatrix} \dfrac{dx_{CL}(t_0)}{dt} \\ \dfrac{dy_{CL}(t_0)}{dt} \\ \dfrac{dz_{CL}(t_0)}{dt} \end{bmatrix} \tag{5.28}$$

where the coordinates $\left[x, y, z\right]^T$ refers to any point of the plane. The point $\left[x_{\overline{CL}}(l_0), y_{\overline{CL}}(l_0), z_{\overline{CL}}(l_0)\right]^T$ on path $\overline{\mathbf{CL}}(l)$ is the intersection between the path and the plane, so the point should meet Eq. (5.28). The distance between points $\left[x_{\overline{CL}}(l_0), y_{\overline{CL}}(l_0), z_{\overline{CL}}(l_0)\right]^T$ and $\left[x_{CL}(t_0), y_{CL}(t_0), z_{CL}(t_0)\right]^T$ is the error of path $\overline{\mathbf{CL}}(l)$ with

reference to path $\mathbf{CL}(t)$ (see Figure 5.10). Upon all points of the theoretical path $\mathbf{CL}(t)$, the maximum error of path $\overline{\mathbf{CL}}(l)$ can be found in the following optimization model.

$$\text{Maximize } f(t,l)=\sqrt{\left[x_{CL}(t)-x_{\overline{CL}}(l)\right]^2+\left[y_{CL}(t)-y_{\overline{CL}}(l)\right]^2+\left[z_{CL}(t)-z_{\overline{CL}}(l)\right]^2} \tag{5.29}$$

Subject to: $t\in[t_{min},t_{max}]$,

$l\in[l_{min},l_{max}]$, and

$$\left[x_{\overline{CL}}(l)\quad y_{\overline{CL}}(l)\quad z_{\overline{CL}}(l)\right]\cdot\begin{bmatrix}\frac{dx_{CL}(t)}{dt}\\ \frac{dy_{CL}(t)}{dt}\\ \frac{dz_{CL}(t)}{dt}\end{bmatrix}=\left[x_{CL}(t)\quad y_{CL}(t)\quad z_{CL}(t)\right]\cdot\begin{bmatrix}\frac{dx_{CL}(t)}{dt}\\ \frac{dy_{CL}(t)}{dt}\\ \frac{dz_{CL}(t)}{dt}\end{bmatrix}$$

As a criterion of the error-bounded fitting, the maximum path error $f_{max}(t^*,l^*)$, the solution of this optimization model, should be less than the tolerance t_{path}. So the criterion is represented as

$$f_{max}(t^*,l^*)<t_{path} \tag{5.30}$$

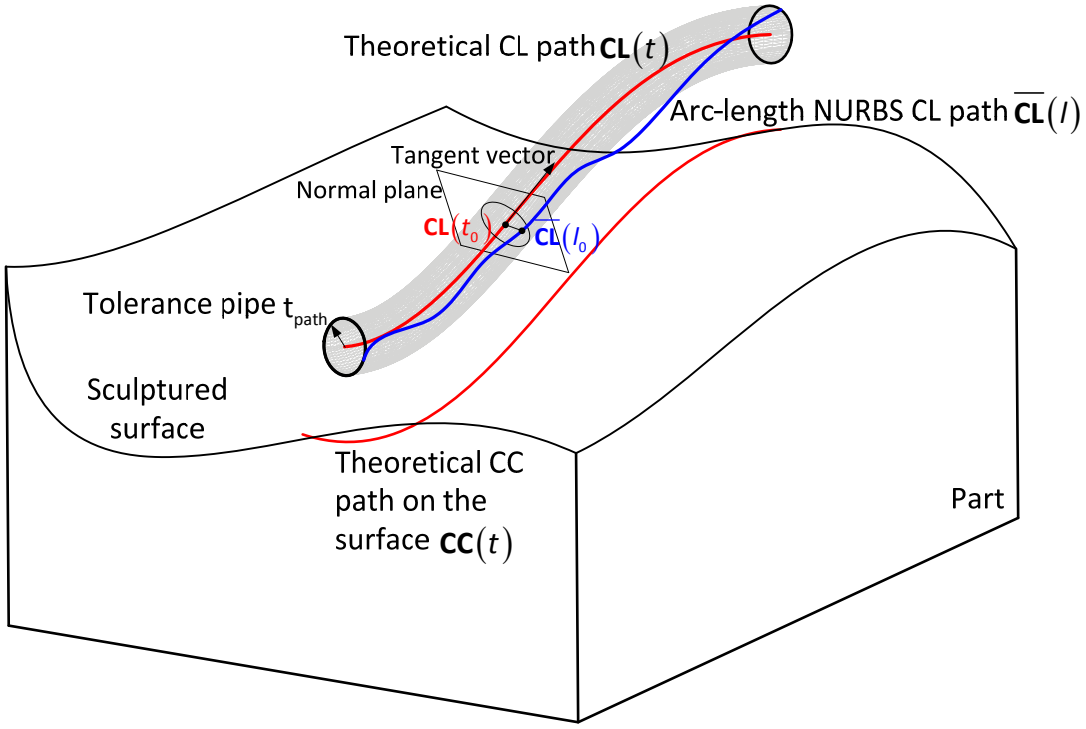


Figure 5.10 The definition of the NURBS CL path error in terms of the theoretical CL path.

Similarly, the hybrid optimization method is used to solve the above optimization problem for the maximum path error. In the fitting iteration, when the path $\overline{\mathbf{CL}}(l)$ meets the criteria of the path and the parameterization errors, the path is a qualified NURBS CL path with the arc length parameter. The path can be fed into the controller of a CNC machine tool to cut the sculptured surface.

5.6 The Approach Procedure

With the innovative key techniques for generating NURBS CL paths, the new approach can be easily implemented and its procedure is clearly shown in the following flow chart in Figure.5 11.

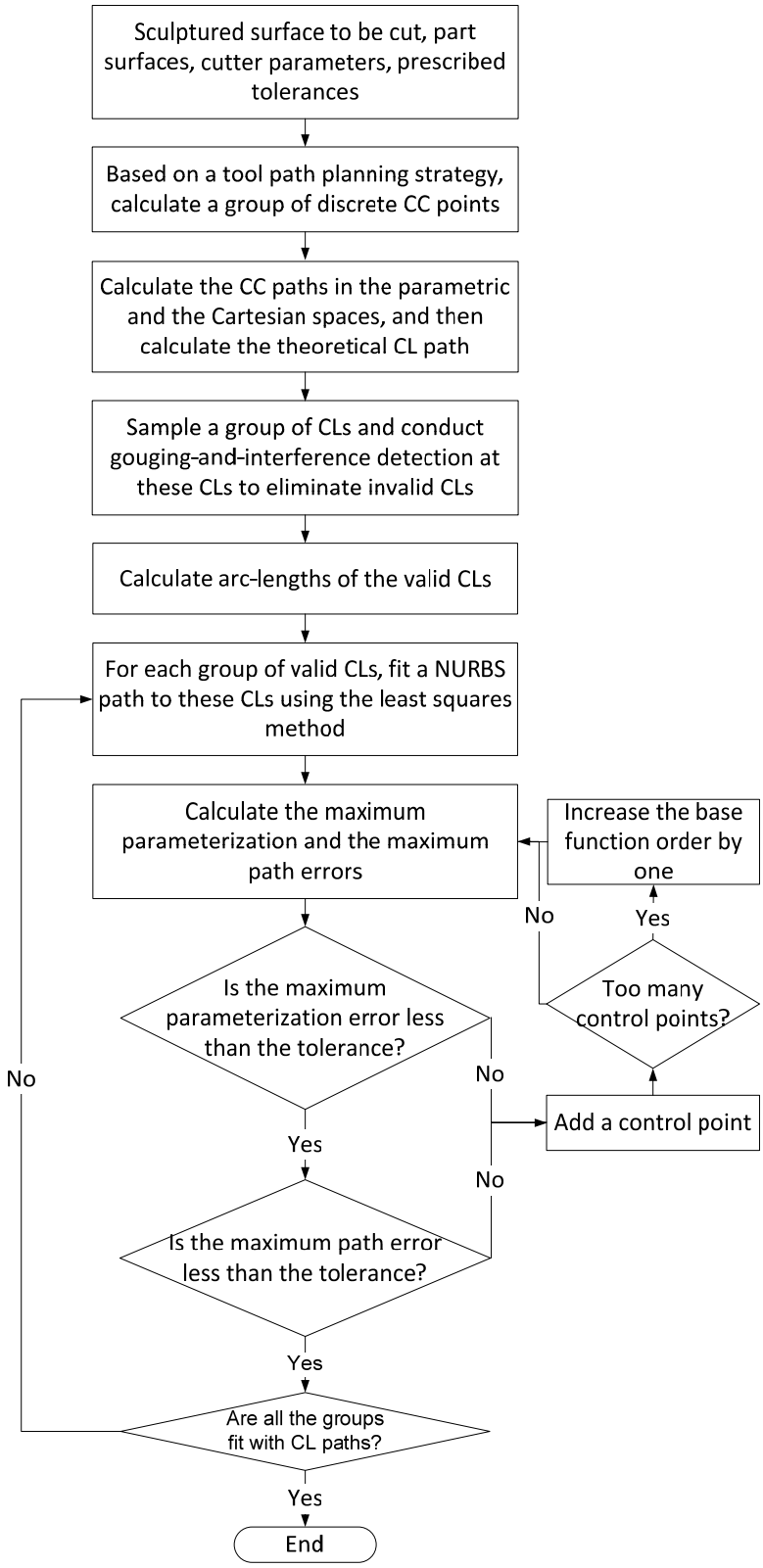


Figure 5.11 The flow chart of the approach procedure.

5.7 Applications

To demonstrate its effectiveness in generating accurate, gouging-and-interference free, arc-length parameterized NURBS CL paths and its advantages over major CAD/CAM software, CATIA V5, the new approach is applied to two sculptured surfaces to generate the paths, and then CATIA is used to generate its NURBS CL paths for finish machining of the surfaces. The parts of the two examples are machined on an OKUMA 5-axis CNC milling center to verify the surface quality cut with our and CATIA NURBS paths. In the first example, a sculptured surface to be machined is a NURBS surface with four by four control points shown in Figure 5.12. A ball end-mill of half an inch diameter is used in finish machining. The surface tolerance is 0.01 inches. In this approach, the tolerances of the path and the parameterization errors are prescribed as 0.01 inches and 0.01, respectively. Using this new approach, 30 NURBS CL paths are generated for finish machining and are plotted in Figure 5.13. These paths are smooth and free of gouging and interference. Their path and parameterization errors are within the tolerances.

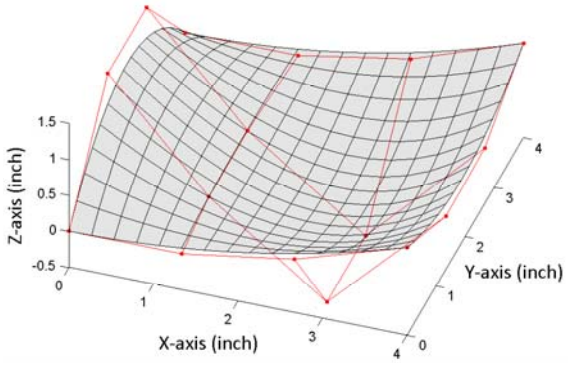


Figure 5.12 A sculptured surface with four by four control points.

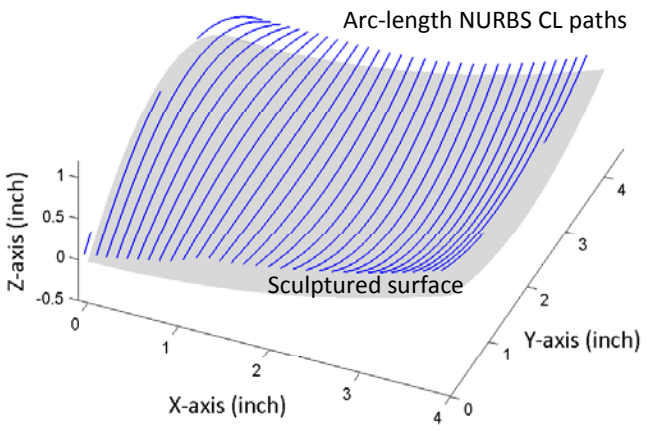
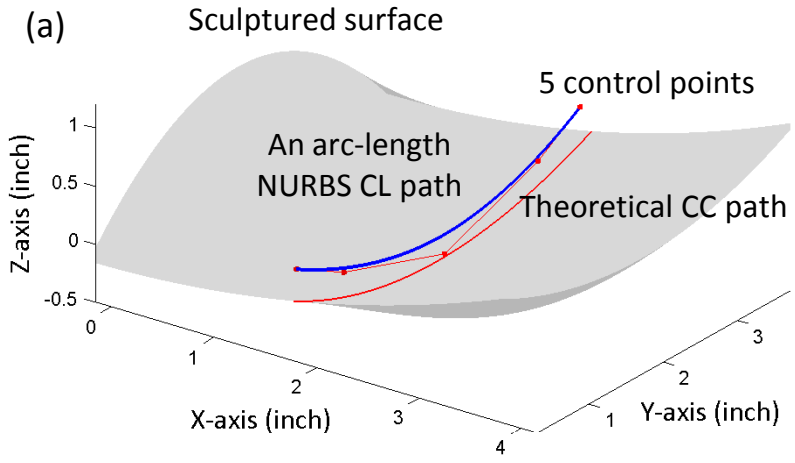


Figure 5.13 The arc-length NURBS CL paths generated with the proposed method.

Among the arc-length NURBS CL paths generated above, a path in the middle is selected and plotted in Figure 5.14(a). It is evident that the path is smooth. The curves of its path and parameterization errors are plotted in Figures 5.14(b) and 5.14(c), respectively. It is clear that all the path errors and the parameterization errors are less than the prescribed tolerances. Thus, this path, as a typical example of the other paths, is accurate.



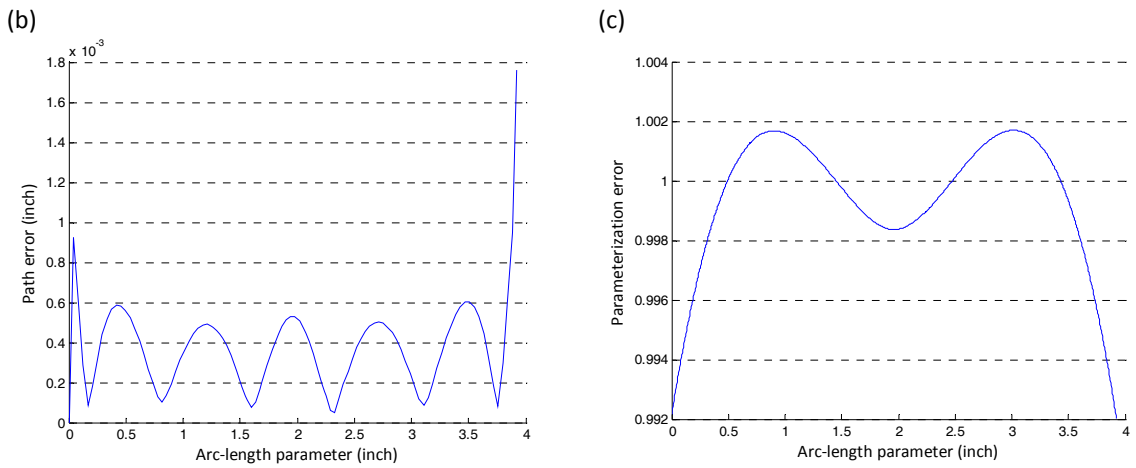


Figure 5.14(a) An arc-length NURBS CL path with five control points, (b) the path error plot of the CL path, and (c) the parameterization error plot of the CL path.

To compare our approach to the NURBS path generation function of CATIA, CATIA is used to plan NURBS CL paths for machining the same sculptured surface. The same number of NURBS paths is generated as the path number of our approach, and the CATIA paths are plotted in Figure 5.15(a). However, it is clear that the CATIA paths are not smooth. Figure 5.15(b) shows a zoom-in of the paths, which are not smooth. Moreover, the number of the total control points of the CATIA NURBS paths is 1074, and that of our paths is 225. By using the NURBS tool paths, the surface cut with our paths is much smoother that the surface cut with the CATIA paths. The photos of the machined parts are provided in Figure 5.16.

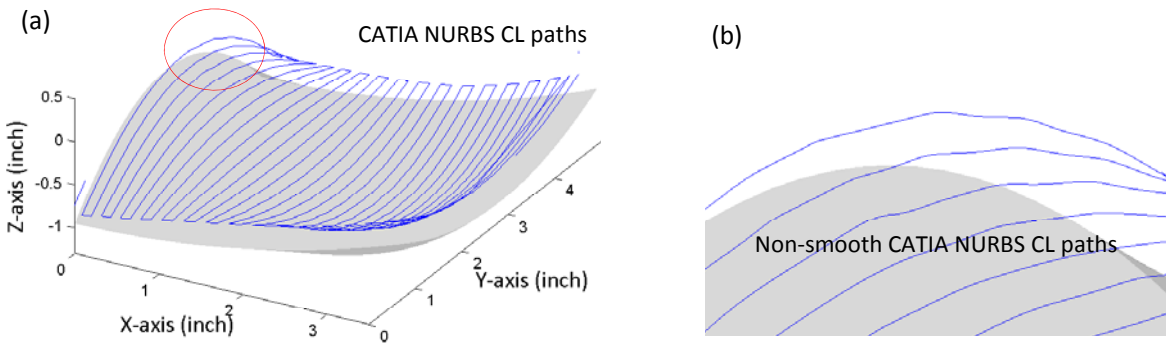


Figure 5.15(a) The NURBS CL paths generated by CATIA, and (b) the non-smooth NURBS CL paths.

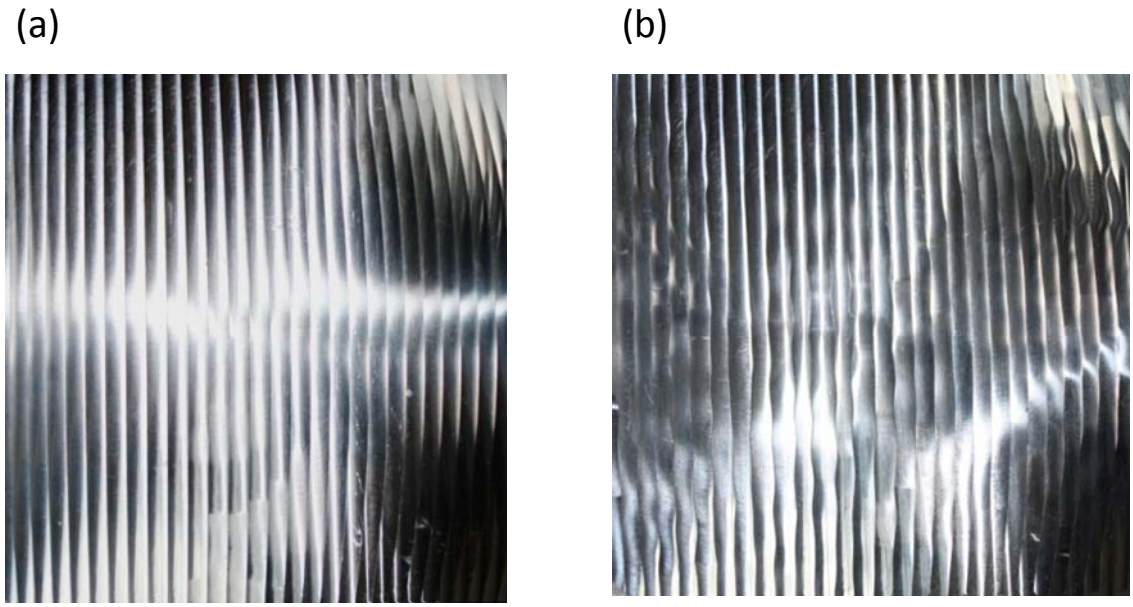


Figure 5.16(a) The sculptured surface cut with our NURBS CL paths, and (b) the sculptured surface cut with the CATIA NURBS CL paths.

In the second example, a sculptured surface is adopted and shown in Figure 5.17. To cut the surface, a ball end-mill with half an inch diameter is used and the tolerance is 0.025 inch. In our approach, since the cutter cannot access the bottom of the surface, the NURBS CL paths are truncated to avoid gouging and interference, which are plotted in Figure 5.18. The part is cut with the tool-paths and the surface

bottom is then cut with a ball end-mill with a quarter inch diameter. The machined part is rendered in Figure 5.19.

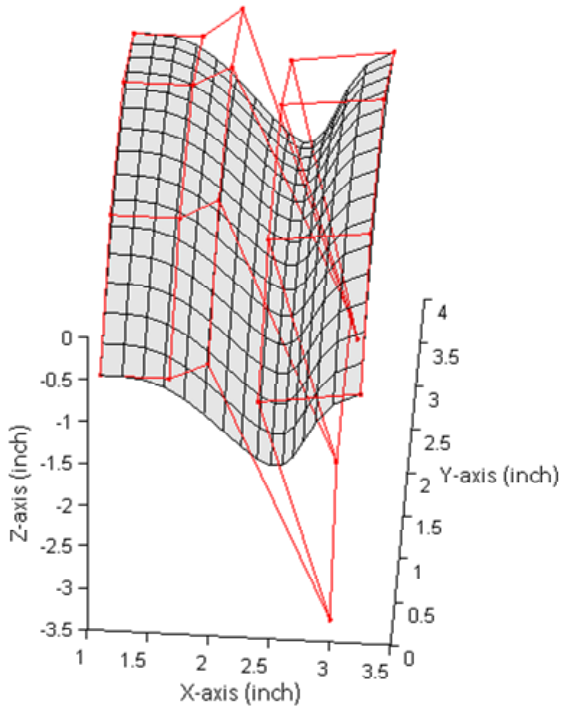


Figure 5.17 A sculptured surface with four by six control points.

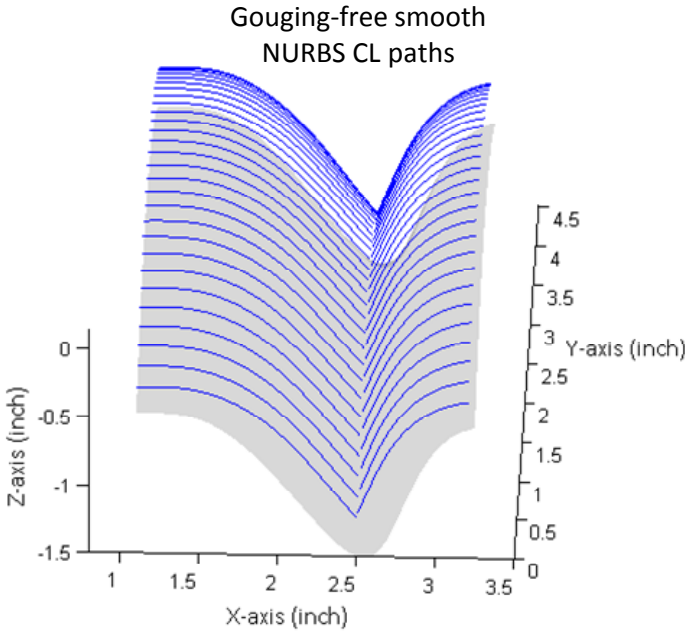


Figure 5.18 Arc-length NURBS CL paths generated with our approach.

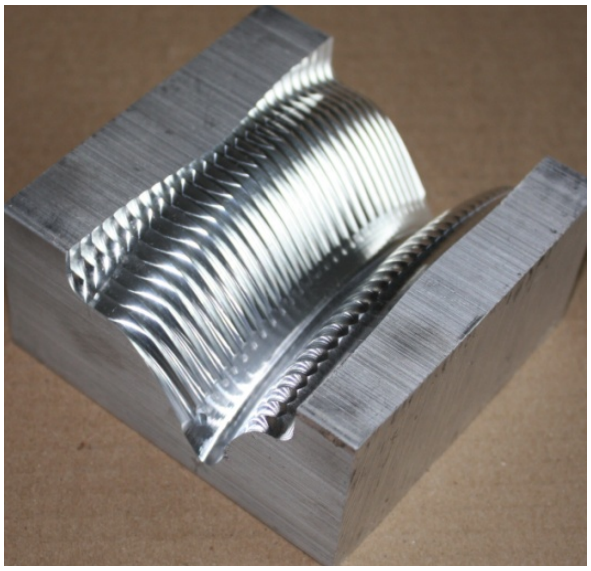


Figure 5.19 The sculptured surface cut with our NURBS CL paths.

For comparison, CATIA is also applied to the surface in the second example. After CATIA generates the NURBS CL paths and conducts gouging and interference detection, paths are attained and plotted in Figure 5.20. Unfortunately, the paths are not smooth and still cause gouging at the bottom of the surface. The machined part verified the surface quality is bad (see Figure 5.21).

In the two examples, it is confirmed that the arc-length NURBS CL paths generated with our approach are accurate, smooth, and gouging-free. This approach can be implemented in CATIA for sculptured surface machining.

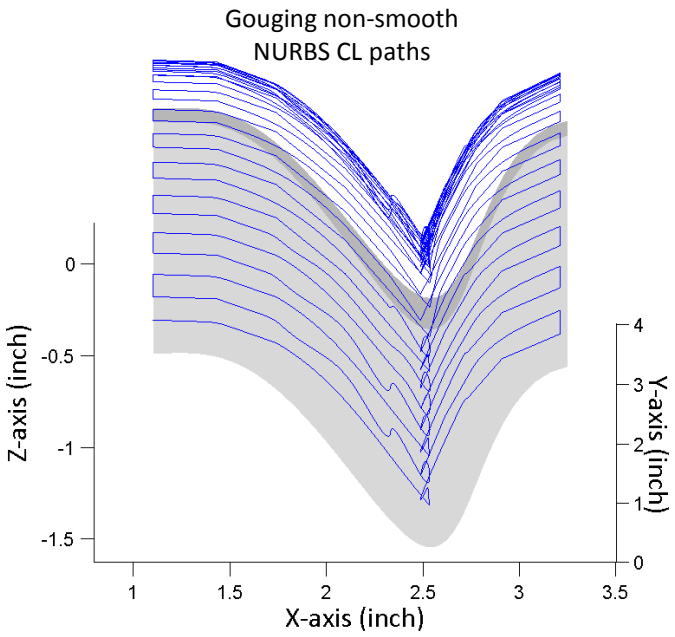


Figure 5.20 The NURBS CL paths generated with CATIA.

Figure 5.21 The sculptured surface cut with the CATIA NURBS CL paths.

5.8 Summary

This work proposes a new approach to generating accurate, gouging-free, smooth NURBS CL paths with the arc-length parameter. The main contributions of this work include (1) generating arc-length parameterized NURBS CL paths with the

path and parameterization errors globally bounded using a new optimization model, (2) conducting gouging and interference detection to eliminate invalid sample CLs before fitting a NURBS CL path to them, which is different from the conventional method, and (3) the close-form equations are derived to calculate the true arc-length for points on the theoretical CL paths. The approach can be easily implemented into current CAD/CAM software to benefit the manufacturing industry.

Chapter 6 Piecewise NURBS Tool-Paths with the Arc-Length Parameter and their Application on High Feed, Accuracy CNC Milling of 2-D Curved Profiles

6.1 Introduction

Current research on NURBS (non-uniform rational B-spline) machining includes two types: (1) NURBS tool paths generation prior to machining and (2) real-time NURBS interpolation for NURBS tool trajectory in CNC machining. The first type is only focused on interpolating a group of cutter locations with a NURBS path in geometry, subject to a constraint that the path errors should be less than a prescribed tolerance; and the second type simply assumes a general NURBS path as input and calculates cutter locations of the path in real-time of machining, in order to ensure that tool trajectory errors are less than the tolerance and tool kinematics is under control. The first type work actually is the input of the second type; unfortunately, they are isolated without interaction. The consequence includes that (a) general non-linear NURBS paths of the first type are difficult for algorithms of

the second type to work on, and (b) if the feed is high, the interpolation is challenged to keep the trajectory errors within tolerance and maintain good tool kinematics along highly curved paths. To root out the problems in this work, a new type of tool path □ piecewise NURBS tool paths with the arc length parameter □ is first proposed for high feed and accuracy NURBS machining, and an innovative approach is established to generate this type of tool path off-line by taking the NURBS interpolation into consideration. First, an accurate NURBS path with the arc length parameter is generated, and then the tool trajectory errors are predicted. If the maximum tool trajectory error is larger than the tolerance in the prescribed high feed rate, the path is decomposed at the breaking points. For qualified curved segments, their feed rates remain the same; and, for disqualified curved segments, their feed rates are reduced. The piecewise NURBS tool paths with the arc length parameter are generated prior to machining, and they are a genuine solution to high feed and accuracy NURBS machining. This work can greatly promote the NURBS machining in industrial practice.

In the phase of NC tool path planning, based on a reference path – a general NURBS cutter location path, it is re-parameterized with the arc length. This path is then fed to the NURBS interpolator for CNC machining. Since it takes into account the interpolation mechanism, this path can reduce the cutter trajectory and feed rate errors. To generate this type of tool path, an innovative approach is proposed with two features. One is to sample points on the reference path and calculate their arc lengths by decomposing the reference tool path into Bezier curve segments. The other is to fit a NURBS path by bounding the parameterization error.

6.2 Piecewise NURBS NC Tool Paths with Arc Length Parameter and Different feeds

In CNC milling, the real-time cutter trajectory is mainly determined by a pre-generated path of cutter location (also called a reference tool path in the work), a prescribed feed rate and the interpolation algorithm used in the CNC controller. After the G-codes of the reference tool path are fed into the controller, it instantaneously extracts cutter locations of the path to execute finite tool motions, eventually composing the actual tool trajectory. For linear and circular reference paths, their corresponding interpolators can easily generate highly accurate cutter trajectories, compared to the reference paths, with a consistent feed as prescribed. Thus, these reference paths are often planned without taking controllers into consideration. Recently, some CNC controllers such as Siemens 840D and Fanuc 32i are empowered with the NURBS interpolation in order for tools cutting free-form shape along curved trajectory. However, if the prescribed feed rate is high, it is quite challenging for these controllers to keep the chord errors of the tool trajectory within the tolerance and the feed rate constant as the prescribed. To root out this difficult problem, it is necessary to understand the real-time NURBS interpolation mechanism and improve reference NURBS tool paths prior to machining; therefore, piecewise NURBS NC tool paths with the arc length parameter for high feed, accurate milling is originally proposed in this work.

As background, the real-time NURBS interpolation mechanism is introduced here and the unsolved technical problems are stressed. It is common in industry

practise that general NURBS cutter location paths $\mathbf{CL}(u)=\left[x(u),y(u),z(u)\right]^{\mathrm{T}}$, in which the parameter u is unit free, are generated as reference tool paths and fed into CNC controllers. Usually, this type of tool path is represented as

$$\mathbf{CL}(u)=\frac{\sum_{i=0}^{n}\left(w_i\cdot N_{i,m}(u)\cdot\mathbf{P}_i\right)}{\sum_{i=0}^{n}\left(w_i\cdot N_{i,m}(u)\right)}=\sum_{i=0}^{n}\left(R_{i,m}(u)\cdot\mathbf{P}_i\right), \tag{6.1}$$

where u is a general parameter ranging between a and b, $\mathbf{P}_i\ (i=0,\ldots,n)$ are the control points, $w_i\ (i=0,\ldots,n)$ are the corresponding weights, and $N_{i,m}(u)\ (i=0,\ldots n)$ are the base functions with degree of m. The knot vector is $\mathbf{U}=\left[\,u_0,\cdots u_m,u_{m+1},\cdots,u_n,u_{n+1},\cdots u_{n+m+1}\right]$, where $u_0=u_1=\cdots=u_m=a$ and $u_{n+1}=\cdots=u_{n+m+1}=b$. At present, the general NURBS tool path is the only type of NURBS tool path that can be calculated using existent methods or commercial CAD/CAM software. After this type of path is fed into controllers together with a specified feed rate v, it is expected that the cutter truly moves along this reference path in the same velocity of v. Then, the NURBS interpolator calculates, in real time, instantaneous cutter locations along the reference path, and the tool directly feeds among the locations in finite motions. Thus, the cutter trajectory is composed of the chords connecting the locations, and the trajectory error ε_t is the deviation between the chords and the reference tool path. The objective of the NURBS interpolator is to ensure the trajectory and feed rate errors are within the specified tolerances. The feed rate is defined as

$$v = \left\| \frac{d}{du}\left(\mathbf{CL}(u)\right) \right\| \cdot \frac{du}{dt}$$

$$\frac{du}{dt} = \frac{v}{\left\|\mathbf{CL}'(u)\right\|} \tag{6.2}$$

Because the solution of Eq. (6.2) is difficult in the general case, approximation techniques are used based on Taylor's expansion. For small curvatures, the first order approximation is normally acceptable:

$$u_{i+1} = u_i + \Delta \mathrm{T} \cdot \left.\frac{du}{dt}\right|_{u=u_i} = u_i + \frac{v \cdot \Delta \mathrm{T}}{\left\|\mathbf{CL}'(u)\right\|} \tag{6.3}$$

If the tool path has high curvature values then second order approximation is suitable

$$u_{i+1} = u_i + \frac{v}{\left\|\mathbf{CL}'(u_i)\right\|} \cdot \Delta \mathrm{T} - \frac{1}{2} \cdot \frac{v^2}{\left\|\mathbf{CL}''(u_i)\right\|^4} \cdot \Delta \mathrm{T}^2 \tag{6.4}$$

After the parameter u_{i+1} of the next cutter location is calculated, its 3-D coordinate $\mathbf{CL}(u_{i+1})$ is computed, then the cutter moves straight from points $\mathbf{CL}(u_i)$ to $\mathbf{CL}(u_{i+1})$. For some general NURBS tool paths, the relationship between the parameter u of cutter location and its arc length is highly non-linear; hence, the value of the second derivative $\left\|\mathbf{CL}''(u)\right\|$ can be much larger than zero. The chord length $\left|\mathbf{CL}(u_i)\mathbf{CL}(u_{i+1})\right|$ varies greatly at different locations; so the instantaneous feed rate $v_i = \left|\mathbf{CL}(u_i)\mathbf{CL}(u_{i+1})\right| / \Delta \mathrm{T}$ fluctuates significantly and the feed rate error ε_v is high

(see Figure 6.1). Moverover, the arc length $\overset{\frown}{\mathbf{CL}(u_i)\mathbf{CL}(u_{i+1})}$ of the NURBS path between cutter locations $\mathbf{CL}(u_i)$ and $\mathbf{CL}(u_{i+1})$ can be quite different from the chord length. Consequently, the trajectory error ε_t could be larger than the tolerance of the part, scraping it in this cut. Meanwhile, the cutter feed rate error ε_v is large, causing high acceleration and deceleration in tool motion.

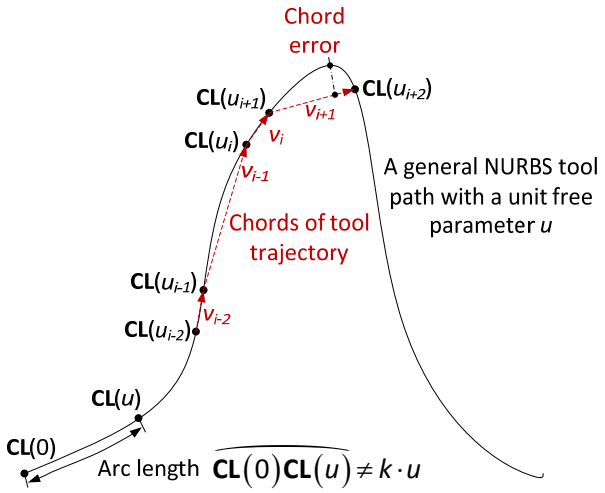


Figure 6.1 An exaggerated, illustrative diagram for real-time NURBS interpolation of a general, non-linear NURBS tool path with a unit free parameter u.

To address the aforementioned problems, many researchers have improved the conventional representation of the non-linear relationship between path segments and their chords using different orders of Taylor's expansion [67, 68], the spline interpolation [42-44, 47, 48] and the length-to-parameter function [76, 48, 78]. Unfortunately, these methods cannot root out the problems because of the inappropriate tool path, NURBS tool paths $\mathbf{CL}(u)$ with a unit-free parameter. In this work, first, a NURBS cutter location path $\mathbf{CL}(s)$ with the arc length parameter s is

proposed. This new path can be calculated by re-parameterizing the general NURBS tool path, and it is an effective solution to the problems of the existent NURBS interpolators.

To better understand the NURBS cutter location path with the arc length parameter in theory, it is necessary to introduce its main properties and advantages here. This type of path is denoted as $\mathbf{CL}(s)=\left[X(s),Y(s),Z(s)\right]^{\mathrm{T}}$, in which the parameter *s* is the arc length. A parameter value corresponds to a cutter location on the path and is the arc length between the start of the path and the cutter location. Since

$$ds=\left\|d\left(\mathbf{CL}(s)\right)\right\| \tag{6.5}$$

the characteristic equation of the path is

$$\left\|\mathbf{CL}'(s)\right\|=\sqrt{\left(\frac{dX}{ds}\right)^2+\left(\frac{dY}{ds}\right)^2+\left(\frac{dZ}{ds}\right)^2}=1 \text{ and } \left\|\mathbf{CL}''(u)\right\|=0 \tag{6.6}$$

Therefore, the governing equation for the NURBS interpolation can simply be

$$s_{i+1}=s_i+v\cdot\Delta\mathrm{T} \tag{6.7}$$

where the feed *v* is not very high. The two parameters, s_i and s_{i+1}, correspond to two cutter locations on the path, $\mathbf{CL}(s_i)$ and $\mathbf{CL}(s_{i+1})$, respectively. Ideally, the chord length $\left|\mathbf{CL}(s_i)\mathbf{CL}(s_{i+1})\right|$ meets the following equation,

$$\left|\mathbf{CL}(s_i)\mathbf{CL}(s_{i+1})\right| = v \cdot \Delta \mathrm{T} \tag{6.8}$$

The arc length between points $\mathbf{CL}(s_i)$ and $\mathbf{CL}(s_{i+1})$ is

$$\left|\overset{\frown}{\mathbf{CL}(s_i)\mathbf{CL}(s_{i+1})}\right| = s_{i+1} - s_i \tag{6.9}$$

Since

$$\left|\mathbf{CL}(s_i)\mathbf{CL}(s_{i+1})\right| \approx \left|\overset{\frown}{\mathbf{CL}(s_i)\mathbf{CL}(s_{i+1})}\right| \tag{6.10}$$

the chords of the finite tool motions can accurately represent the reference NURBS tool path. Thus, the tool trajectory closely matches the reference path (see Figure 6.2(a)). Meanwhile, the chord lengths of the tool motions are the same, and the cutter can remain the same feed rate *v* along the path. However, the NURBS path with the arc length parameter is quite effective only when the feed rate is small and normal; if the feed is very high, Eq. (6.10) cannot hold at locations with large curvatures, and the chord errors could be way beyond the tolerance (see Figure 6.2(b)). Therefore, a general, non-linear NURBS tool path with a unit free parameter is a major source of error; and for a NURBS tool path with the arc length parameter, high feed rate is another main source of error in NURBS machining.

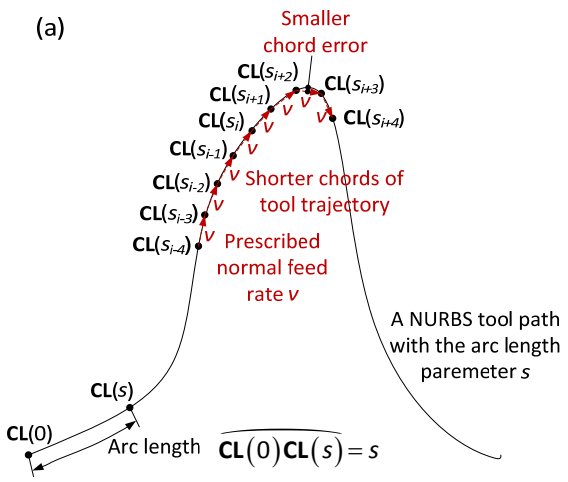


Figure 6.2 An exaggerated, illustrative diagram for real-time NURBS interpolation of a NURBS tool path with the arc length parameter; (a) in a prescribed normal feed rate

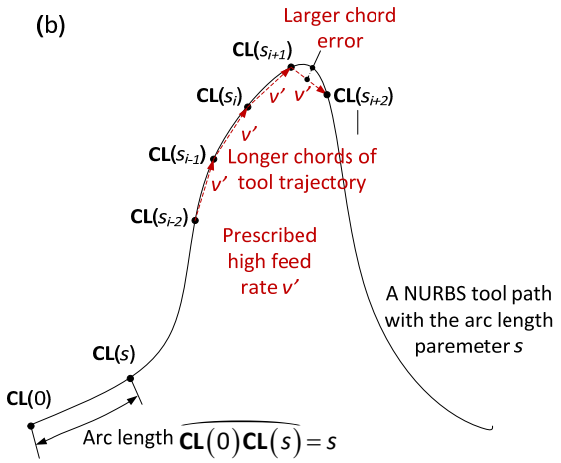


Figure 6.2 (b) prescribed high feed rate.

To increase the tool trajectory accuracy and kinematics in NURBS machining, although many NURBS interpolation algorithms have been proposed to solve the aforementioned problems, it is easier and better to improve NURBS tool paths prior to machining. In this work, the type of NURBS tool path with the arc length parameter is adopted to eliminate the first source of error, and when the feed rate is high, multiple NURBS tool paths with different feed rates are used to root out the second source of error. First, to generate a NURBS tool path with the arc length

parameter, based on a reference NURBS cutter location path with a unit-free parameter computed with commercial CAD/CAM software, it is re-parameterized with the arc length parameter. The main features of this work include (1) a new efficient approach to sampling cutter locations and calculating their arc lengths, and (2) fitting a new NURBS tool path with the arc length parameter by having its geometric and parameterization deviations bounded. This work is mainly focused on 2-D NURBS tool path generation with the arc length parameter.

6.3 A New Approach to Cutter Locations Sampling and their Arc Lengths Calculation

Given a NURBS cutter location path $\mathbf{CL}(u)$ with a unit free parameter (also called a general NURBS path), this work essentially is to re-represent it with the arc length parameter as $\mathbf{CL}(s)$. In the re-parameterization, an important work is to select a group of cutter locations along $\mathbf{CL}(u)$, according to a guide of less samples for less curved segment and more samples for highly curved segment, and compute their arc lengths. Existent methods of re-parameterizing a general NURBS curve are similarly to extract cutter locations sequentially along the curve and calculate their arc lengths using numerical methods, such as the quadrature method [76]. However, these methods are not effective to sample cutter locations according to the reference path curvature and are not robust in computing arc lengths. In this work, an original, efficient approach to cutter locations sampling and their arc lengths calculation is proposed to address the aforesaid problems. Its kernel technique is to

decompose the general NURBS path into short Bezier curve segments continuously until the control polygon of every Bezier curve segment can accurately represent the curve segment; as a result, the sample cutter locations are the end control points of each Bezier segment and their arc lengths can be easily calculated using the control polygons.

6.3.1 Decomposition of a general NURBS path into Bezier curve segments

To decompose a general NURBS path $\mathbf{CL}(u)$ into a number of short Bezier curve segments, a well-established method is adopted and is briefly introduced here. A general NURBS path is defined as $\mathbf{CL}(u)=\sum_{i=0}^{n}\left(R_{i,m}(u)\cdot\mathbf{P}_i^0\right)$ with the knot vector $\mathbf{U}^0=\left[u_0,\cdots,u_m,u_{m+1}\cdots u_k,u_{k+1},\cdots u_n,u_{n+1},\cdots u_{n+m+1}\right]$. By inserting a knot $\bar{u}$ $\left(u_m<u_k\le\bar{u}<u_{k+1}<u_{n+1}\right)$ into the knot vector, a new knot vector $\mathbf{U}^1$ is generated as

$$\mathbf{U}^1=\left[u_0^1\cdots u_{n+m+2}^1\right]=\left[u_0,\cdots,u_m,u_{m+1},\cdots u_k,\bar{u},u_{k+1},\cdots,u_n,u_{n+1},\cdots u_{n+m+1}\right] \tag{6.11}$$

Then, this tool path can be represented, using the new knot vector and a new set of control points, as

$$\mathbf{CL}(u)=\sum_{i=0}^{n+1}\left[R_{i,m}^1(u)\cdot\mathbf{P}_i^1\right] \tag{6.12}$$

where $R_{i,m}^1(u)$ are defined with the knot vector $\mathbf{U}^1$. The new control points are calculated with the following equation:

$$\mathbf{P}_i^1 = (1-\alpha_i^1)\cdot\mathbf{P}_{i-1}^0 + \alpha_i^1\cdot\mathbf{P}_i^0 \tag{6.13}$$

and

$$\alpha_i^1 = \begin{cases} 1 & 0 \le i \le k-m \\ \dfrac{\overline{u}-u_i}{u_{i+m}-u_i} & k-m+1 \le i \le k \\ 0 & k+1 \le i \le n+1 \end{cases} \tag{6.14}$$

where $\mathbf{P}_{-1}^0$ and $\mathbf{P}_{n+1}^0$ are not defined and their terms are ignored in Eq. (6.13). To simply decompose the general NURBS path into two NURBS curve segments, the inserted knot $\overline{u}$ is equal to an existing knot u_k and has to be inserted several times until the multiplicity of the knot u_k is equal to m. If the original multiplicity of the knot u_k is q, the insertion has to repeat by $m-q$ times. The control points in the r^{th} insertion can be calculated as

$$\mathbf{P}_i^r = \left(1-\alpha_i^r\right)\cdot\mathbf{P}_{i-1}^{r-1} + \alpha_i^r\cdot\mathbf{P}_i^{r-1} \tag{6.15}$$

and

$$\alpha_i^r = \begin{cases} 1 & 0 \le i \le k-m+r-1 \\ \dfrac{\overline{u}-u_i}{u_{i+m-r+1}-u_i} & k-m+r \le i \le k-q \\ 0 & k-q+1 \le i \le n+r \end{cases} \tag{6.16}$$

where $1 \le r \le m-q$. For control points $\mathbf{P}_{i-1}^{r-1}$ and $\mathbf{P}_i^{r-1}$ not defined, their terms are ignored in Eq. (6.15). Thus, the general NURBS path is decomposed into two

segments; the control points of one segment are $\mathbf{P}_0^{m-q}$, $\mathbf{P}_1^{m-q}$,...., and $\mathbf{P}_k^{m-q}$, and the control points of the second segment are $\mathbf{P}_k^{m-q}$, $\mathbf{P}_{k+1}^{m-q}$, ..., $\mathbf{P}_{n+m-q}^{m-q}$. The first and last control points of the segments are on the general NURBS path. This is a basic method of decomposing a NURBS tool path (including a Bezier curve) into two shorter segments.

After increasing the multiplicity of each distinct knot to the degree m, the NURBS tool path is decomposed into shorter Bezier curve segments, and the segment number is the same as the number of non-zero knot intervals in the knot vector. By using this method, a general NURBS tool path can be decomposed into a group of Bezier curve segments (see Figure 6.3).

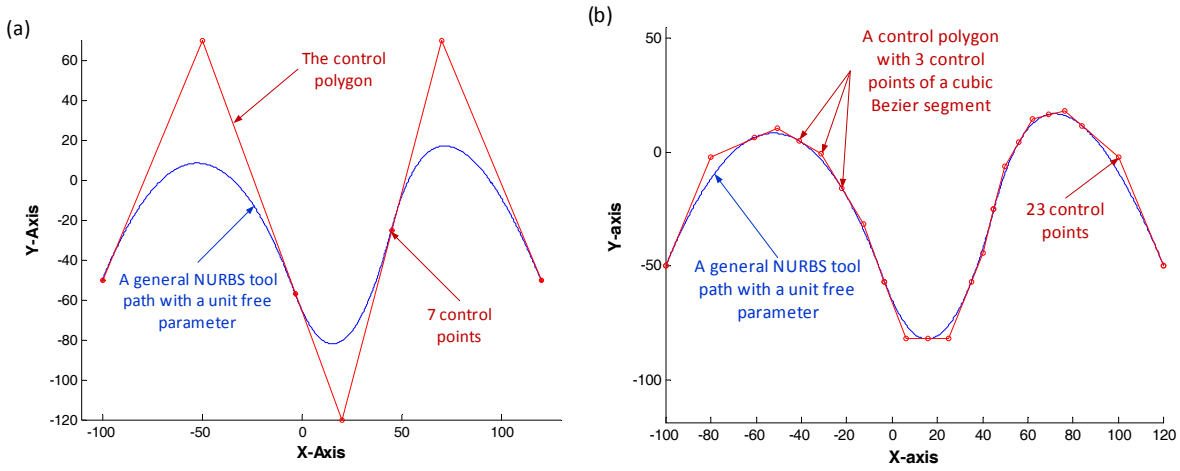


Figure 6.3 (a) A general NURBS tool path with a unit free parameter and its control polygon and (b) the Bezier curve segments and their control polygons after the general NURBS path is decomposed.

6.3.2 Criterion of cutter location sampling

The main advantage of this approach is efficient, accurate computation of the arc lengths of sample cutter locations and path curvature dependent sampling. After the general NURBS path is decomposed into shorter Bezier curve segments, they are repetitively decomposed in the same way; thus, their control polygons represent the path better with less deviation. Until the deviation is less than a prescribed, small tolerance ε_a, the arc length of the path can be simply calculated using its control polygon, which is more accurate and efficient than using the quadrature method. The approach is mathematically approved with the following lemma.

Lemma: For a Bezier curve $\mathbf{CL}(u)$ $\left(\mathbf{CL}(u)=\sum_{i=0}^{m} B_{i,m}(u)\cdot\mathbf{P}_i\right)$ with control points $\mathbf{P}_0$, $\mathbf{P}_1$, ..., $\mathbf{P}_m$, the first and last control points $\mathbf{P}_0$ and $\mathbf{P}_m$ are on the Bezier curve (see Figure 6.4). The control polygon is $\mathbf{P}_0\mathbf{P}_1\ldots\mathbf{P}_m$ and its edge length L_p is $\left|\mathbf{P}_0\mathbf{P}_1\ldots\mathbf{P}_m\right|$ $\left(L_p=\left|\mathbf{P}_0\mathbf{P}_1\right|+\left|\mathbf{P}_1\mathbf{P}_2\right|+\cdots+\left|\mathbf{P}_{m-1}\mathbf{P}_m\right|\right)$. The chord of the curve is $\mathbf{P}_0\mathbf{P}_m$ and its length L_c is $\left|\mathbf{P}_0\mathbf{P}_m\right|$. A small tolerance ε_a of the closeness between L_p and L_c is specified as a very small positive number. If

$$\left|\mathbf{P}_0\mathbf{P}_1\cdots\mathbf{P}_m\right|-\left|\mathbf{P}_0\mathbf{P}_m\right|<\varepsilon_a \tag{6.17}$$

the Bezier curve arc length L_a is

$$L_a\cong\frac{\left|\mathbf{P}_0\mathbf{P}_m\right|+\left|\mathbf{P}_0\mathbf{P}_i\mathbf{P}_{i+1}\mathbf{P}_m\right|}{2} \tag{6.18}$$

and the maximum deviation between the Bezier curve and the chord is at the same level of ε_a.

[Proof]

According to the convex hull property of Bezier curve, the curve is within the region confined by the convex hull of the control polygon. Generally, the convex hull is composed of the first and the last control points, and some or all of the control points between them. Without losing generality, the convex hull is assumed to be composed of four control points such as $\mathbf{P}_0$, $\mathbf{P}_i$, $\mathbf{P}_{i+1}$, $\mathbf{P}_m$, and the length of the convex hull is

$$\left|\mathbf{P}_0\mathbf{P}_i\mathbf{P}_{i+1}\mathbf{P}_m\mathbf{P}_0\right| = \left|\mathbf{P}_0\mathbf{P}_i\right| + \left|\mathbf{P}_i\mathbf{P}_{i+1}\right| + \left|\mathbf{P}_{i+1}\mathbf{P}_m\right| + \left|\mathbf{P}_m\mathbf{P}_0\right| \tag{6.19}$$

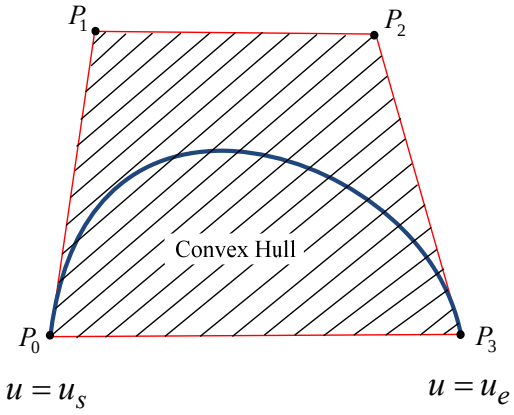


Figure 6.4 A Bezier segment with its control points, control polygon, chord and convex hull (hatched region).

According to a theorem of polygon that the length of an edge is less than the sum of the lengths of the other edges (see Figure 6.4), we can have

$$\left|\mathbf{P}_0\mathbf{P}_i\right| < \left|\mathbf{P}_0\mathbf{P}_1\right| + \left|\mathbf{P}_1\mathbf{P}_2\right| + \cdots + \left|\mathbf{P}_{i-1}\mathbf{P}_i\right| \tag{6.20}$$

and

$$\left|\mathbf{P}_{i+1}\mathbf{P}_m\right| < \left|\mathbf{P}_{i+1}\mathbf{P}_{i+2}\right| + \cdots + \left|\mathbf{P}_{m-1}\mathbf{P}_m\right| \tag{6.21}$$

Thus, the length of the control polygon is less than the length of the convex hull.

If,

$$\left|\mathbf{P}_0\mathbf{P}_1 \cdots \mathbf{P}_m\right| - \left|\mathbf{P}_0\mathbf{P}_m\right| < \varepsilon_a \tag{6.22}$$

This in-equation means the control polygon is very close to the chord. Since the Bezier curve is within its convex hull, the length of the curve L_a is

$$\left|\mathbf{P}_0\mathbf{P}_i\mathbf{P}_{i+1}\mathbf{P}_m\right| > L_a > \left|\mathbf{P}_0\mathbf{P}_m\right| \tag{6.23}$$

and

$$L_a - \left|\mathbf{P}_0\mathbf{P}_m\right| < \varepsilon_a \text{ and } \left|\mathbf{P}_0\mathbf{P}_i\mathbf{P}_{i+1}\mathbf{P}_m\right| - L_a < \varepsilon_a \tag{6.24}$$

Therefore

$$L_a \cong \frac{\left|\mathbf{P}_0\mathbf{P}_m\right| + \left|\mathbf{P}_0\mathbf{P}_i\mathbf{P}_{i+1}\mathbf{P}_m\right|}{2} \tag{6.25}$$

Project the control polygon onto the chord, and the projected length is equal to the chord length.

$$\left|\mathbf{P}_0\mathbf{P}_i\right|\cdot\cos\alpha_{0,i}+\left|\mathbf{P}_i\mathbf{P}_{i+1}\right|\cdot\cos\alpha_{i,i+1}+\left|\mathbf{P}_{i+1}\mathbf{P}_m\right|\cdot\cos\alpha_{i+1,m}=\left|\mathbf{P}_0\mathbf{P}_m\right| \tag{6.26}$$

where $\alpha_{0,i}$ is the angle between $\mathbf{P}_0\mathbf{P}_i$ and $\mathbf{P}_0\mathbf{P}_m$, $\alpha_{i,i+1}$ is the angle between $\mathbf{P}_i\mathbf{P}_{i+1}$ and $\mathbf{P}_0\mathbf{P}_m$, $\alpha_{i+1,m}$ is the angle between $\mathbf{P}_{i+1}\mathbf{P}_m$ and $\mathbf{P}_0\mathbf{P}_m$. Substituting this into Eq. (6.22), we get

$$\left|\mathbf{P}_0\mathbf{P}_i\right|\cdot\left(1-\cos\alpha_{0,i}\right)+\left|\mathbf{P}_i\mathbf{P}_{i+1}\right|\cdot\left(1-\cos\alpha_{i,i+1}\right)+\left|\mathbf{P}_{i+1}\mathbf{P}_m\right|\cdot\left(1-\cos\alpha_{i+1,m}\right)<\varepsilon_a \tag{6.27}$$

Thus, the in-equations $1-\cos\alpha_{0,i}\ll\varepsilon_a$, $1-\cos\alpha_{i,i+1}\ll\varepsilon_a$, $1-\alpha_{i+1,m}\ll\varepsilon_a$ hold, which means the angles, $\alpha_{0,i}$, $\alpha_{i,i+1}$, and $\alpha_{i+1,m}$, are very small. It can be approximate that

$$1-\cos\alpha_{0,i}\cong\sin\alpha_{0,i},\ 1-\cos\alpha_{i,i+1}\cong\sin\alpha_{i,i+1}\ \text{and}\ 1-\cos\alpha_{i+1,m}\cong\sin\alpha_{i+1,m} \tag{6.28}$$

Substituting this equation into Eq. (6.27), we get

$$\left|\mathbf{P}_0\mathbf{P}_i\right|\cdot\sin\alpha_{0,i}+\left|\mathbf{P}_i\mathbf{P}_{i+1}\right|\cdot\sin\alpha_{i,i+1}+\left|\mathbf{P}_{i+1}\mathbf{P}_m\right|\cdot\sin\alpha_{i+1,m}<\varepsilon_a \tag{6.29}$$

It is reasonable to assume that $\mathbf{P}_{i+1}$ is the farthest control point away from the chord, and the greatest distance between the control points and the chord is the distance $D_{\max}$ between $\mathbf{P}_{i+1}$ and the chord, which is $\left|\mathbf{P}_0\mathbf{P}_i\right|\cdot\sin\alpha_{0,i}+\left|\mathbf{P}_i\mathbf{P}_{i+1}\right|\cdot\sin\alpha_{i,i+1}$. Hence,

$$D_{\max}<\varepsilon_a \tag{6.30}$$

Since the Bezier curve is bounded by the convex hull, the largest deviation between the curve and the chord is less than the distance $D_{\max}$. Therefore, the maximum deviation between the Bezier curve and the chord is at the level of ε_a. **[End]**

Based on the lemma, the criterion for cutter locations sampling is established here. If the difference between the lengths of the control polygon and the chord of a Bezier curve segment is within the tolerance ε_a, the end control points are two sample points and the arc length of the segment is approximated with the length of control polygon. If the criterion is not met, the Bezier curve segment is decomposed into two shorter sub-Bezier segments by inserting a knot into the current knot vector *m* times. This process continues until all the Bezier curve segments meet the criterion of sampling points. Since curved path segments have to be represented with more Bezier curves polygons, the method can sample more cutter locations on the large curvature portion and fewer locations on the small curvature portion.

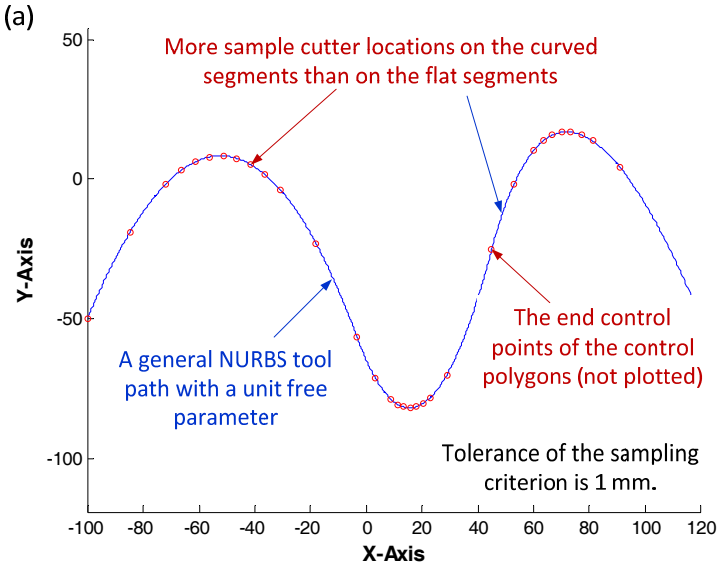


Figure 6.5 A general NURBS tool path with a unit free parameter is decomposed for sample cutter locations based on prescribed tolerances: (a) 1 mm,

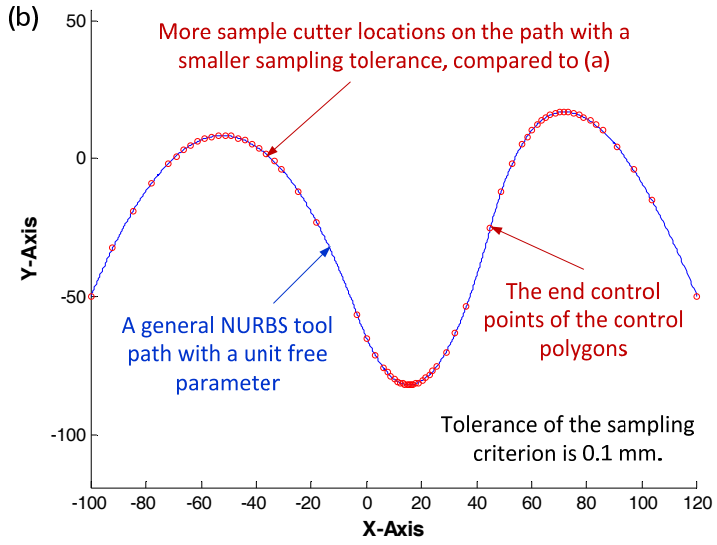


Figure 6.5 (b) 0.1 mm

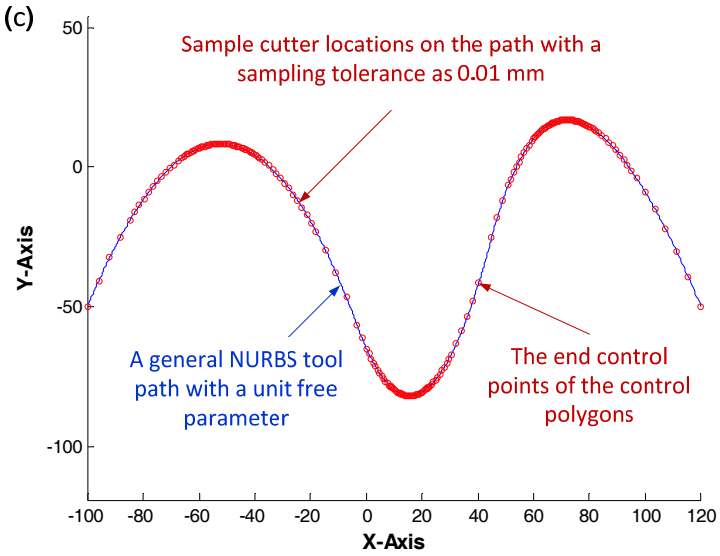


Figure 6.5 (c) 0.01 mm

6.3.3 Problems of the adaptive quadrature method

Existent curve re-parametrization approaches often adopt the adaptive quadrature method to calculate the arc lengths of sample points on a given curve. This method works on many curves; however, for some NURBS (including Bezier) curves, it is defective. Our new approach is a valid solution. Given a NURBS curve $\mathbf{CL}(u)=\left[x(u),y(u),z(u)\right]^{\mathrm{T}}$ with a knot vector $\mathbf{U}=\left[u_0,\cdots,u_m,u_{m+1}\cdots u_k,u_{k+1},\cdots u_n,u_{n+1},\cdots u_{n+m+1}\right]$ and $u\in[a, b]$. Let $L(u)$ represents the arc length of a point with parameter u on $\mathbf{CL}(u)$, and its formulae is

$$L(u)=\int_a^u\left\|\mathbf{CL}'(u)\right\|\cdot du=\int_a^u\sqrt{x'(u)^2+y'(u)^2+z'(u)^2}\cdot du \tag{6.31}$$

Since a simple form of Eq. (6.31) cannot be derived, a numerical method, the adaptive quadrature method, is often used to calculate the arc length, and

mathematically, the quadrature method is based on Simpson's rule. For NURBS curves with dramatic change of curve derivatives and short knot spans, the quadrature method could ignore that curve portion within the short knot span and with high value of $\|\mathbf{CL}'(u)\|$, causing an arc length error.

To demonstrate the problem, a quadratic NURBS curve is given and plotted in Figure 6.6. The control points are $\mathbf{P}_1$ (100,0), $\mathbf{P}_2$ (150,0), $\mathbf{P}_3$ (150,0), $\mathbf{P}_4$ (190,0), $\mathbf{P}_5$ (190,50), $\mathbf{P}_6$ (150,50), $\mathbf{P}_7$ (150,50), $\mathbf{P}_8$ (150,50), and $\mathbf{P}_9$ (100,50); their knot vector $[u_0, u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8, u_9, u_{10}, u_{11}]$ is [0, 0, 0, 200, 200.005, 200.01, 200.015, 200.02, 200.025, 400, 400, 400]; and their weights vector is [1, 1, 1, 1, 1, 1, 1, 1, 1]. Using the quadrature method, the arc length of this curve is 100.000 *mm*, while our new approach finds the arc length is 213.222 *mm*, which is the true arc length. The quadrature method is wrong; the reason is the presence of a short knot span $[u_3, u_8]$. When the quadrature method bisects the initial parameter interval $[u_0, u_{11}]$ into two sub-intervals, the midpoints of the sub-intervals do not fall into $[u_3, u_8]$, but the determination condition is met. Consequently, the curve length in this short parametric interval is ignored. Fortunately, our new approach can find the correct result, since only the addition and subtraction are required.

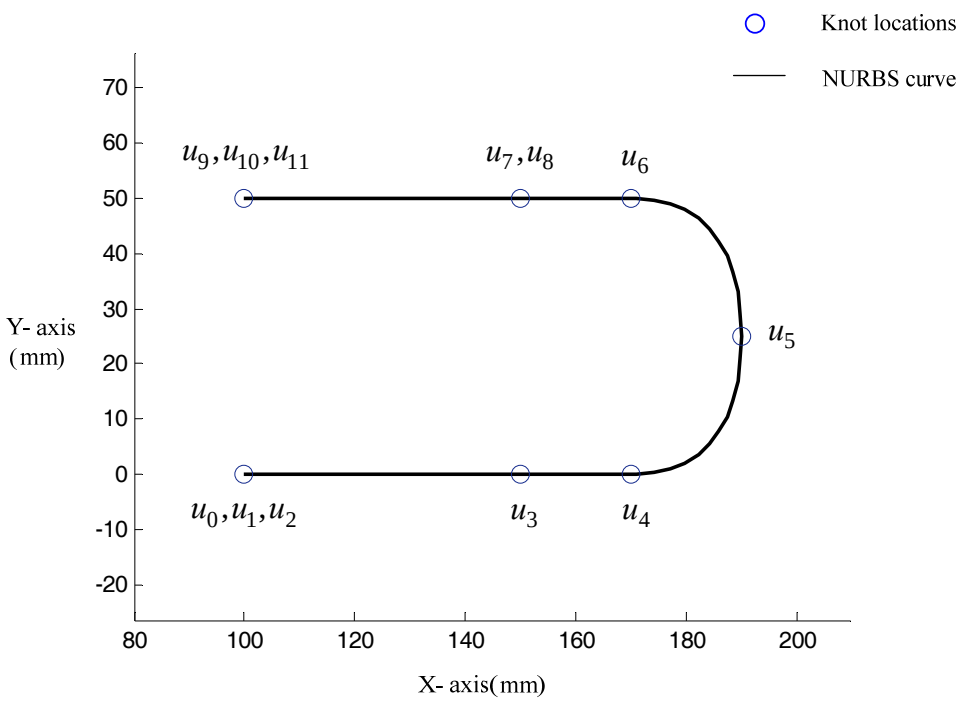


Figure 6.6 A quadratic NURBS curve and its knot locations.

6.4 NURBS Tool Path Generation with Error Bounded Fitting

Based on a group of sample cutter locations found above, a NURBS path with the arc length parameter is fit to them using the least square method. To ensure high quality of the path, the path and parameterization errors, which are defined in the following, are bounded in an iterative process. At the beginning, the control points and the order of blending functions are specified. After using the least square method, a NURBS tool path is attained. Then its path and parameterization errors are checked against the prescribed tolerances. If they are not within the tolerances, more control points will be used or the order of the blending functions will be increased. After several iterations, a NURBS tool path with the arc length parameter will be calculated, whose error is small with regard to the general NURBS tool path

and whose first derivative is almost equal to one. Here, the path and parameterization errors are defined.

6.4.1 Path error definition

Suppose a general NURBS path $\mathbf{CL}(u)=\left[x(u),\, y(u)\right]^{\mathrm{T}}$ is given, and the NURBS path with the arc length parameter $\mathbf{CL}(s)=\left[X(s),\, Y(s)\right]^{\mathrm{T}}$ is found. The normal $\mathbf{N}(u)$ of the general NURBS path at any point can be represented as

$$\mathbf{N}(u)=\mathbf{CL}(u)+d(u)\cdot\mathbf{n}(u)$$

where the unit normal $\mathbf{n}(u)$ can be found as

$$\mathbf{n}(u)=\frac{1}{\sqrt{x'(u)^2+y'(u)^2}}\cdot\begin{bmatrix}-y'(u)\\ x'(u)\end{bmatrix} \text{ or } \mathbf{n}(u)=\frac{1}{\sqrt{x'(u)^2+y'(u)^2}}\cdot\begin{bmatrix}y'(u)\\ -x'(u)\end{bmatrix}$$

The error of the NURBS path with the arc length parameter $\mathbf{CL}(s)$ with respect to $\mathbf{CL}(u)$ is the length $d(u)$ at the intersection between the normal $\mathbf{N}(u)$ and path $\mathbf{CL}(s)$ (see Figure 6.7). Assume the specified tolerance is t_{path}, if the maximum path error $\max\left(d(u)\right)$ at parameter $u_{\max}$ is less than t_{path}, the path is accurate, its parameterization error is then checked. If the maximum error is greater than t_{path}, the number of control points is increased by one and a new fitting process starts until the path error is less than t_{path}.

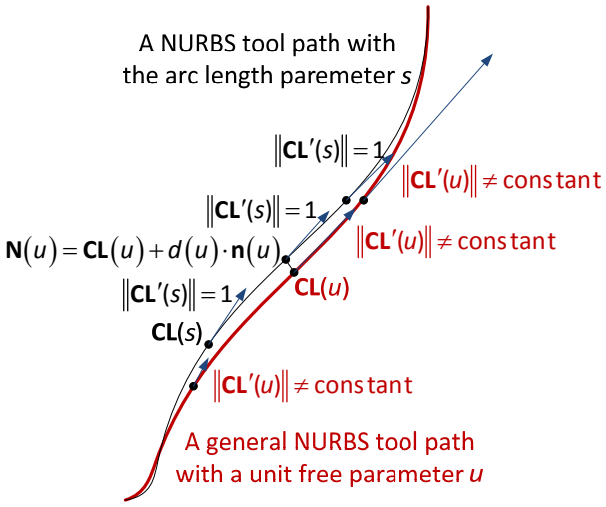


Figure 6.7 An illustration diagram of a general NURBS tool path with a unit free parameter and a NURBS tool path with the arc length parameter.

6.4.2 Parameterization error definition

For a genuine NURBS tool path with the arc length parameter $\mathbf{CL}(s)$, the parameter of a cutter location is equal to the arc length of the path from the beginning to this location. Theoretically, the first derivative of this path $\|\mathbf{CL}'(s)\|$ is equal to one (see Figure 6.7). However, re-parameterization of a general NURBS path using the arc length parameter cannot be carried out using closed-form equations; and the first derivative is not exactly equal to one. The fluctuation of the first derivative about one is defined as the parameterization error, which is represented as

$$\varepsilon(s) = \|\mathbf{CL}'(s)\| - 1$$

In the iterative process of generating this path, the NURBS path re-parameterization is gradually improved. After a NURBS path is attained, its first derivative is

calculated and compared with one. The prescribed tolerance for the parameterization error is t_{para}. If the maximum parameterization error $\max(\varepsilon(s))$ is less than t_{para}, the NURBS path is qualified; otherwise, the number of control points or the order of the blending functions will be increased by one.

The general NURBS curve in Figure 6.3(a) is taken as an example to re-parameterized with arc length. The corresponding path and parameterization error are shown in Figures 6.8 and 6.9.

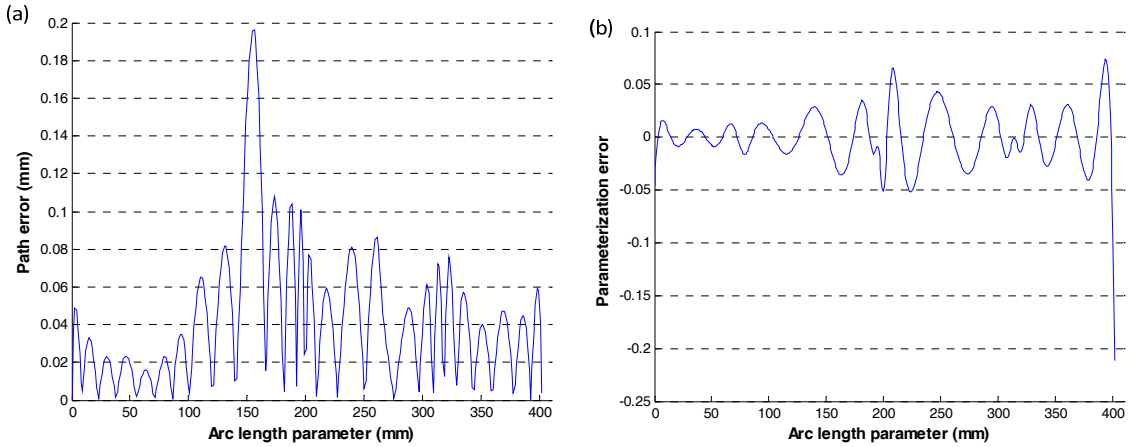


Figure 6.8 For the example in Fig. 3, (a) the path error plot and (b) the parameterization error plot of a NURBS path with the arc length parameter and 27 control points.

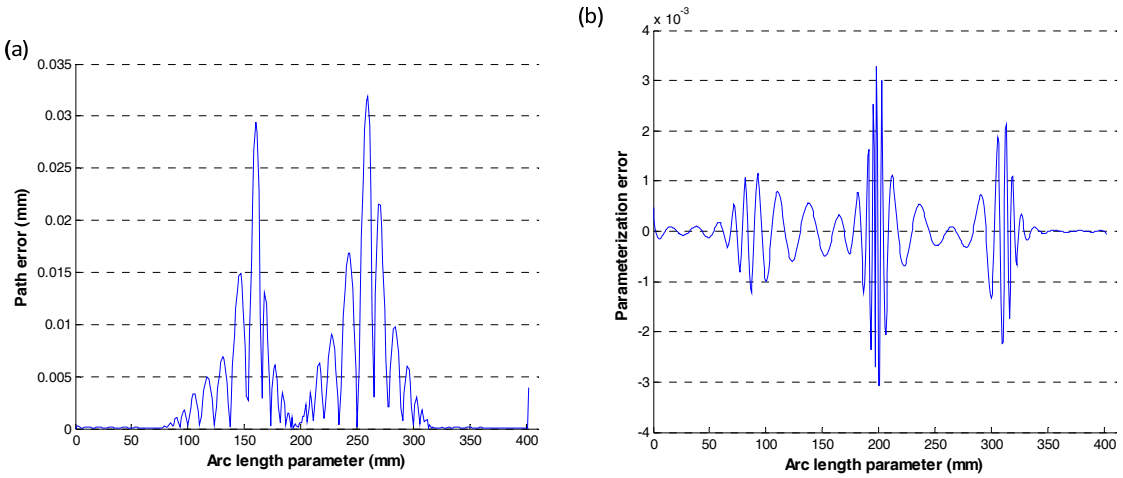


Figure 6.9 For the example in Fig. 3, (a) the path error plot and (b) the parameterization error plot of a NURBS path with the arc length parameter and 50 control points.

6.5 Piecewise NURBS Tool Paths Generation

A NURBS tool path with the arc length parameter is crucial to the NURBS interpolation for accurate tool trajectory and consistent tool velocity, if the prescribed feed rate is not high and the path is not dramatically curved. However, the high feed free-form profile end-milling does not meet the preposition. Even with a NURBS path with the arc length parameter, the tool trajectory error could be much larger than the prescribed tolerance (see Figure 6.10), if the feed rate is high. To solve this problem, several NURBS interpolation algorithms try to reduce the instantaneous tool velocity to ensure the trajectory error, acceleration/deceleration, and jerk are under the limits. Technically, these solutions are passive and are difficult to eliminate the problem. Our work proposes an effective solution that is to represent the highly curved path segments with piecewise NURBS tool paths (see Figure 6.11(a)) and apply reduced feed rates to the piecewise paths.

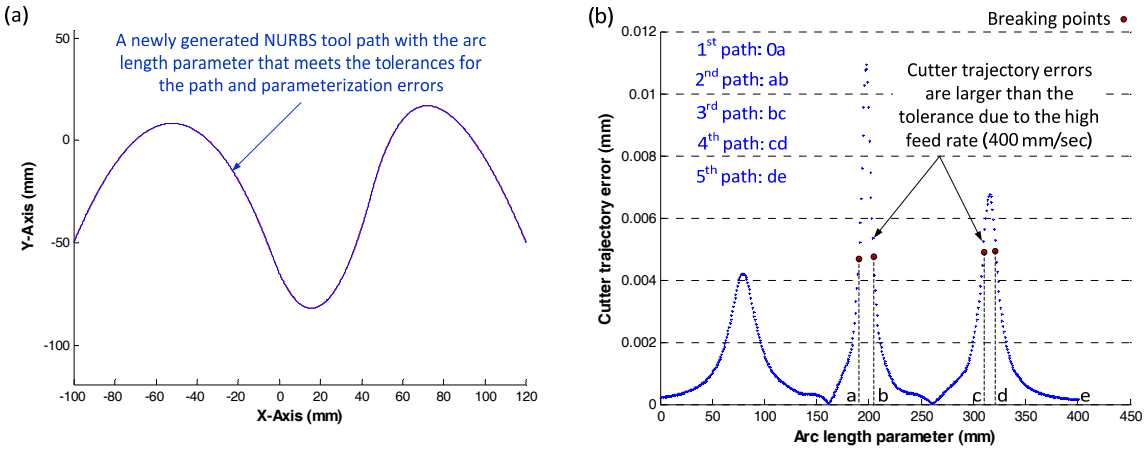


Figure 6.10 For the example in Fig. 3, (a) a NURBS path with the arc length parameter and (b) the cutter trajectory error plot

To decompose a whole NURBS tool path with the arc length parameter into pieces, the popular NURBS interpolation algorithm, which is represented in Eq. (6.4), is adopted in this work. Applying the high feed rate to the algorithm, the tool trajectory errors can be predicted, which are compared against the prescribed tolerance. If no error is larger than the tolerance, this NURBS path is qualified to feed to the CNC controllers. Otherwise, the whole path is decomposed into the segments with larger errors and the qualified segments. For the qualified segments, the aforementioned method is applied to find piecewise NURBS tool paths with the arc length parameter. For the disqualified segments, the feed rate is gradually reduced and the tool trajectory errors are predicted accordingly. Until the errors are less than the tolerance (see Figure 6.11(b)), the segments are presented with piecewise NURBS paths with the newly found feed rates. Finally, the piecewise NURBS tool paths with different feed rates can be fed to the CNC controllers. The tool trajectory can be easily generated with high accuracy, and the tool can feed with good kinematics.

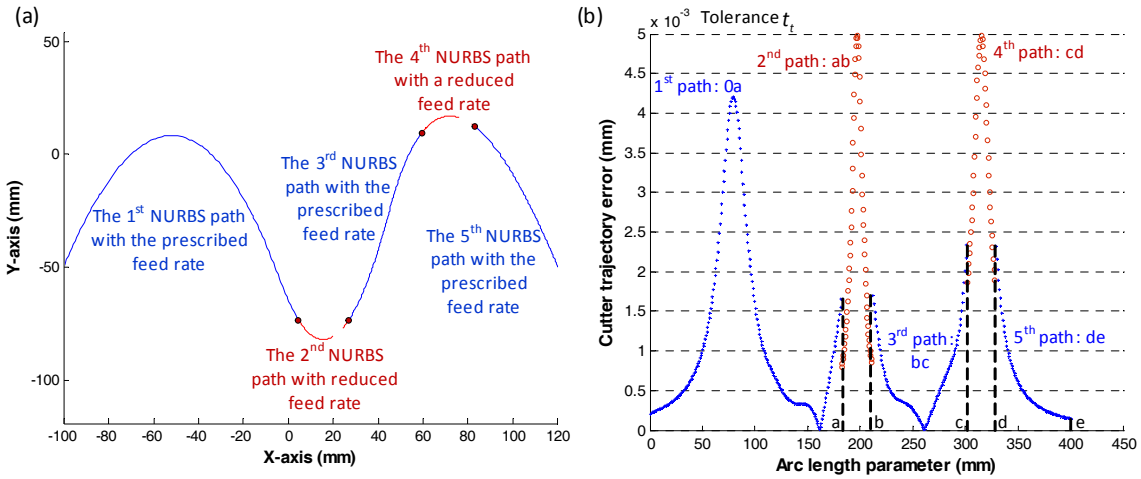


Figure 6.11 For the example in Fig. 3, (a) piecewise NURBS paths with the arc length parameter and (b) the cutter trajectory error plot

6.6 Procedure of Planning the New NURBS Paths

NURBS tool paths with the arc length parameter for machining sculptured part surfaces are planned in process planning. Fortunately, it can root out the problems of the existing NURBS interpolation methods for accurate NURBS cutter trajectory. Thus, it is necessary to calculate this type of tool path in the phase of NC path generation. The main procedure of planning the tool path includes three steps.

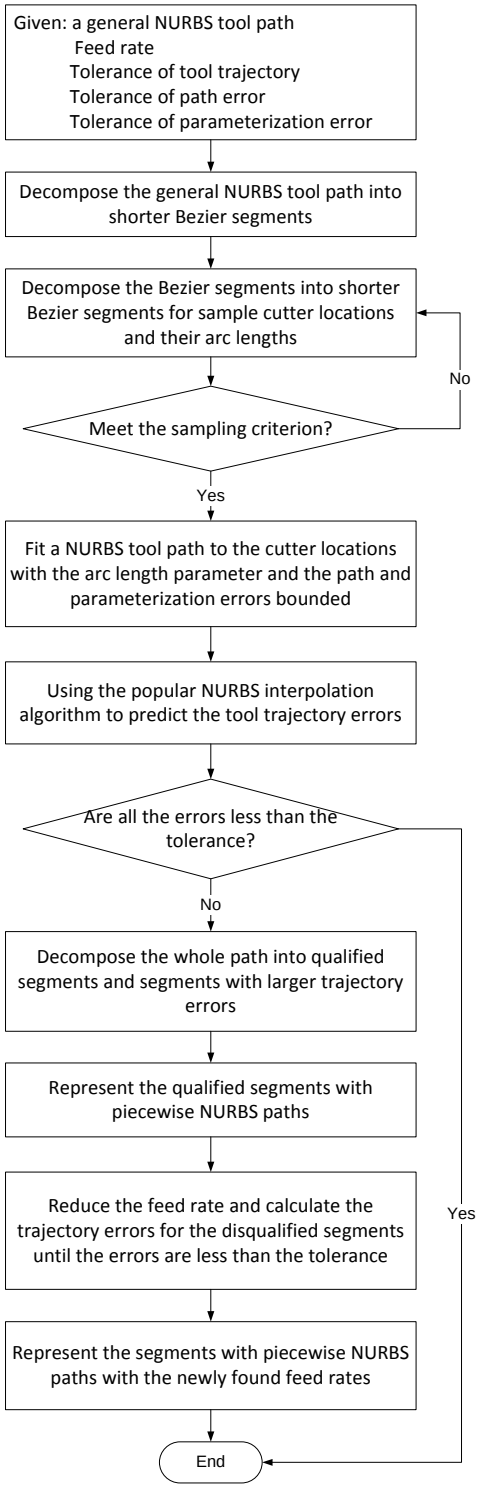


Figure 6.12 The flowchart of the procedure of this work.

6.7 Applications

To demonstrate the advantage of piecewise NURBS tool paths with the arc length parameter in CNC machining and the new approach for this type of path, a 2-D NURBS profile is adopted, and a general NURBS tool path with a unit free parameter for CNC NURBS machining of this profile is generated with the CATIA CAD/CAM software, which is shown in Figure 6.13. The 16 control points of this general NURBS path are listed in Table 6.1. For high feed machining, the feed rate is prescribed as 400 mm/sec; and for high accuracy, the tolerances are specified in Table 6.2.

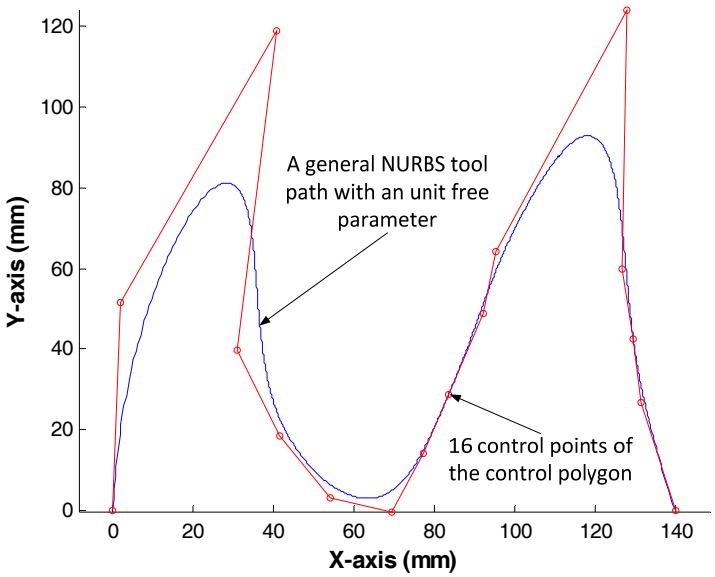


Figure 6.13 A general NURBS tool path with a unit free parameter.

Table6.1 Control points of a general NURBS tool path with a unit free parameter

Control points	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
x (mm)	0	1.7878	40.5466	30.7910	41.3353	54.1981	69.3975	77.2690
y (mm)	0	51.5414	118.6363	39.9483	18.7282	3.0307	-0.2338	14.0278
Control points	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
x (mm)	83.7591	92.1069	95.3723	127.8367	126.6988	129.4184	131.6819	140.0000
y (mm)	28.7536	49.0031	64.1980	123.8235	59.9702	42.4562	26.7862	0

Table6.2 Prescribed tolerances used in the proposed approach

Tolerances	**Sampling points (T_j)**	**Path error (T_f)**	**Parameterization error (T_p)**	**Tool trajectory error (T_t)**
Prescribed values	0.0001 mm	0.005 mm	0.005	0.005 mm

First, the general path is continuously decomposed, until the cutter locations of the path meet the sampling criterion given in Section 6.3.2, resulting 7374 sample cutter locations (see Figure 6.14). By using the least square method and subject to the tolerances of the path and parameterization errors defined in Section 6.4, a NURBS tool path with the arc length parameter is generated and plotted in Figure 6.15. The path error curve is plotted in Figure 6.16(a), and the maximum path error is close to 1.6×10^{-3} mm, which is less than the tolerance 5.0×10^{-3} mm. The parameterization error is plotted in Figure 6.16(b), and the maximum path error is less than the tolerance 5.0×10^{-3} mm. Thus, the NURBS path with the arc length parameter geometrically is a qualified arc length parameter path.

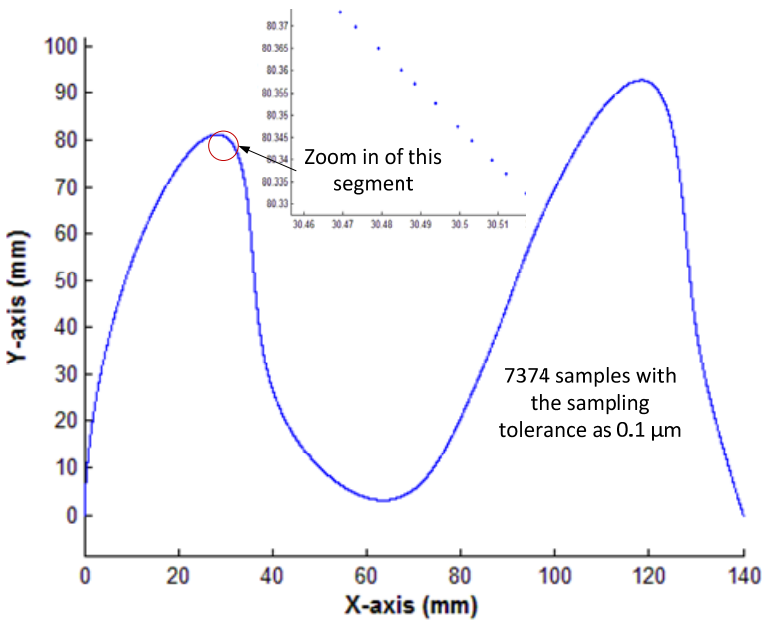


Figure 6.14 By decomposing the NURBS path, sample cutter locations are computed and plotted.

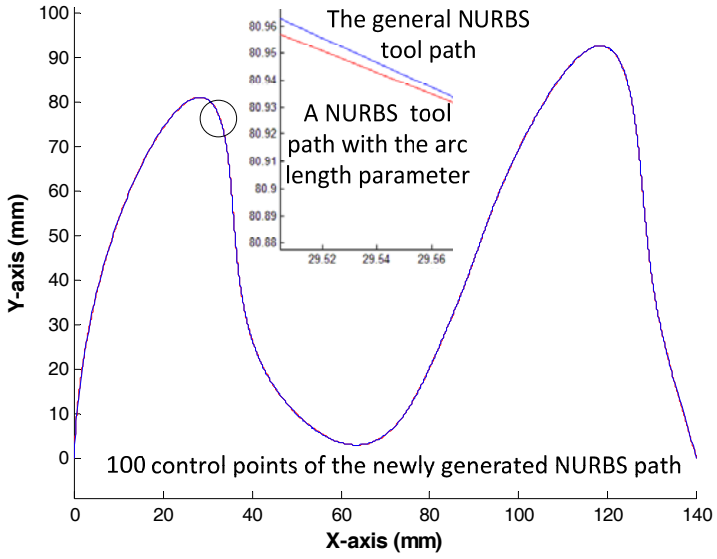


Figure 6.15 A NURBS tool path with the arc length parameter is generated and is plotted with the general NURBS path.

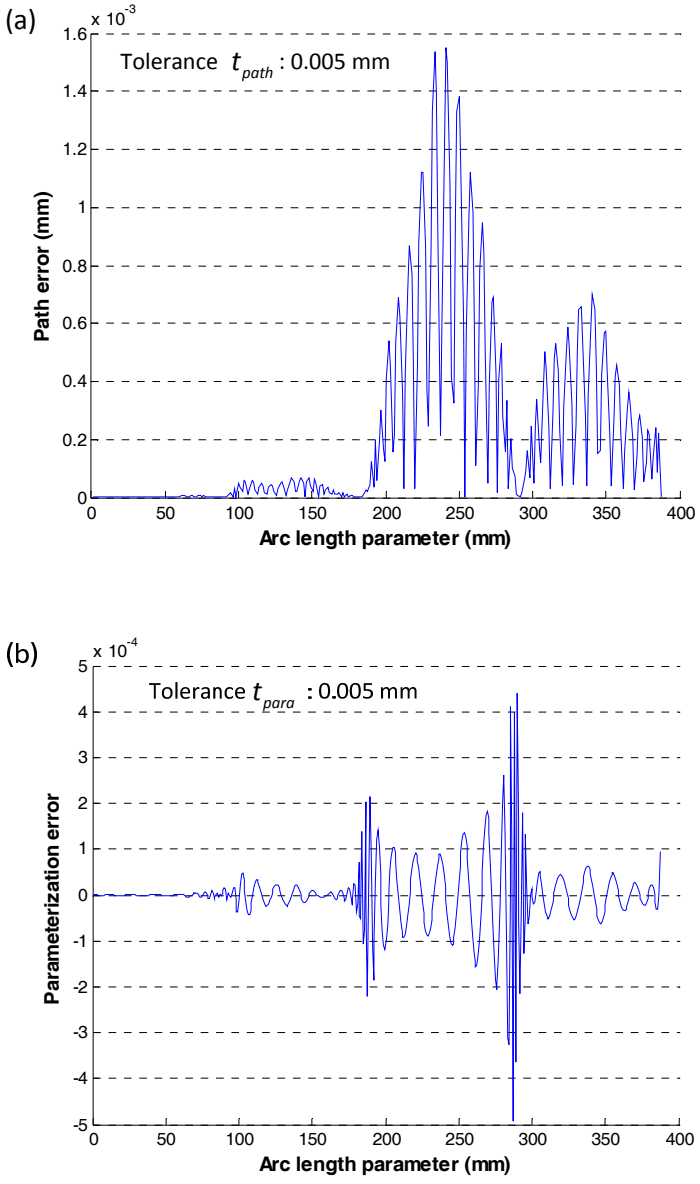


Figure 6.16 (a) The path error curve and (b) the parameterization error curve of the NURBS tool path with the arc length parameter.

Second, by applying the popular NURBS interpolation algorithm to the new NURBS path, the cutter trajectory and feed rate error curves are attained and plotted in Figures 6.17(a) and 6.17(b), respectively. However, the trajectory errors in three curved path segments are beyond the tolerance 5.0×10^{-3} mm. Likewise, the feed rate significantly fluctuates in these segments. With such a high feed rate (400

mm/sec), the cutter trajectory error could reach 0.025 mm; even a good NURBS path with the arc length parameter is used. If this path is fed into the CNC controller, it is difficult for the controller to reduce the feed rate in real time to ensure the trajectory accuracy and the tool kinematics.

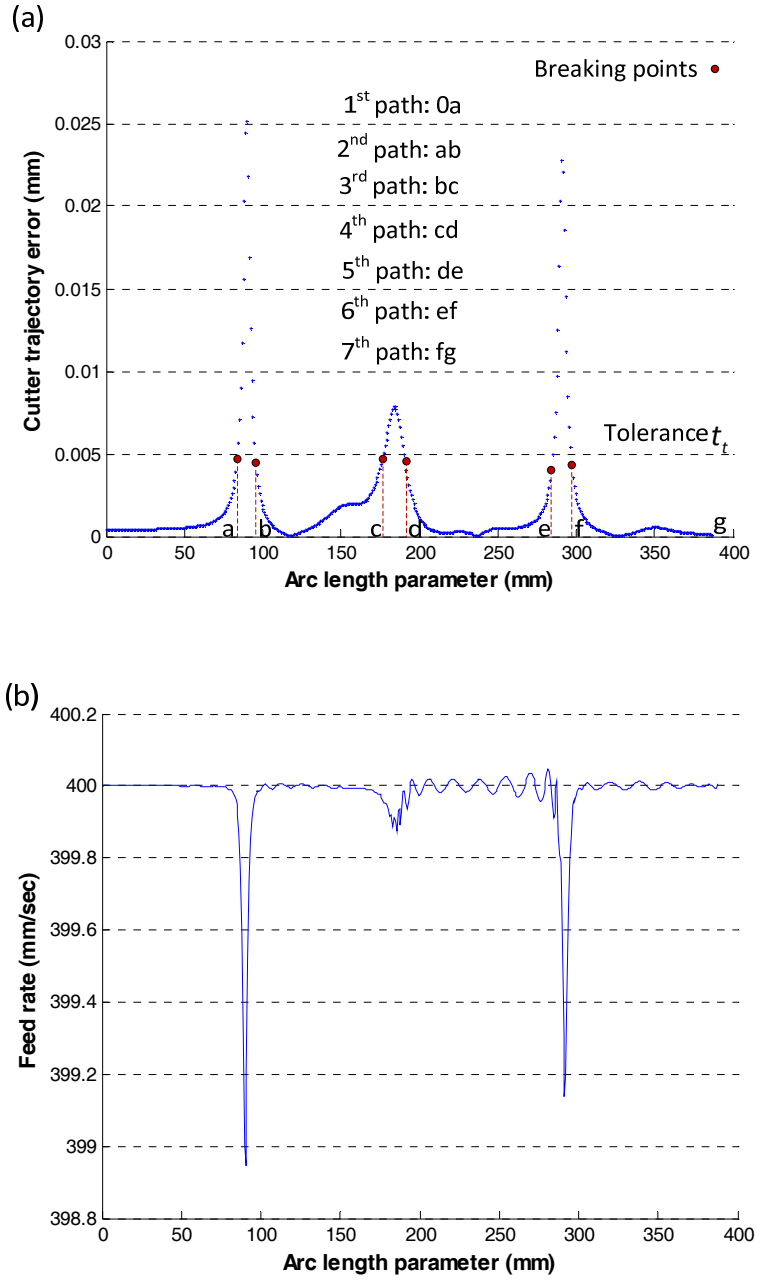


Figure 6.17 (a) The cutter trajectory error curve and (b) the feed rate error curve of the NURBS path with the arc length parameter.

Then, the NURBS tool path with the arc length parameter is decomposed into pieces, and reduced feed rates are applied to the path segments with large trajectory errors. In the cutter trajectory error plot in Figure 6.17(a), the points close to and less than the tolerance are used to divide the NURBS path into seven NURBS tool paths with the arc length parameter, and these points are called breaking points (see Figure 6.18). The control points of the seven NURBS paths are 15, 31, 18, 24, 17, 30 and 15. Since the trajectory errors of the 2^{nd}, 4^{th} and 6^{th} NURBS paths are larger than the tolerance (5.0×10^{3} mm) in the high feed rate (400 mm/sec), their feed rates are reduced so that their trajectory errors are within the tolerance. The proper feed rates for the 2^{nd}, 4^{th} and 6^{th} NURBS paths are 176, 318 and 186 mm/sec, respectively, and the cutter trajectory error and the feed rate curves of the piecewise NURBS tool paths are plotted in Figures 6.19 and 6.20 respectively. With these tool paths, the existent NURBS interpolation function can easily generate high accurate cutter trajectory with good tool kinematics.

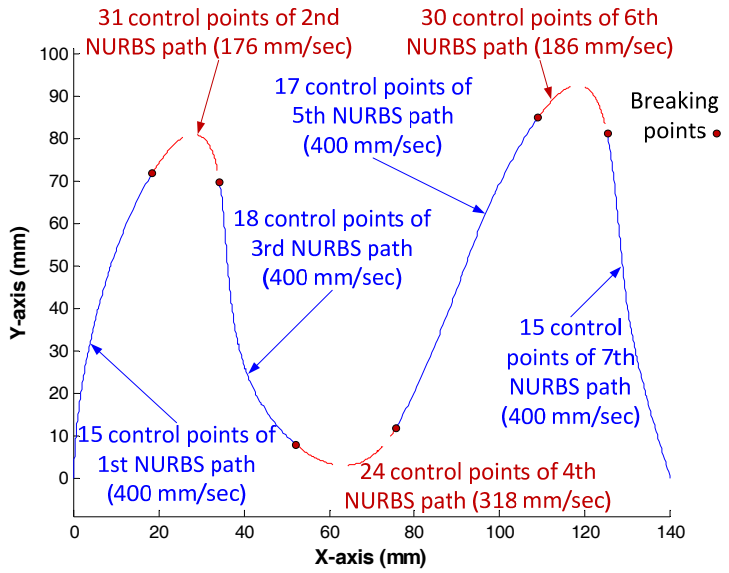


Figure 6.18 The piecewise NURBS tool paths with the arc length parameter.

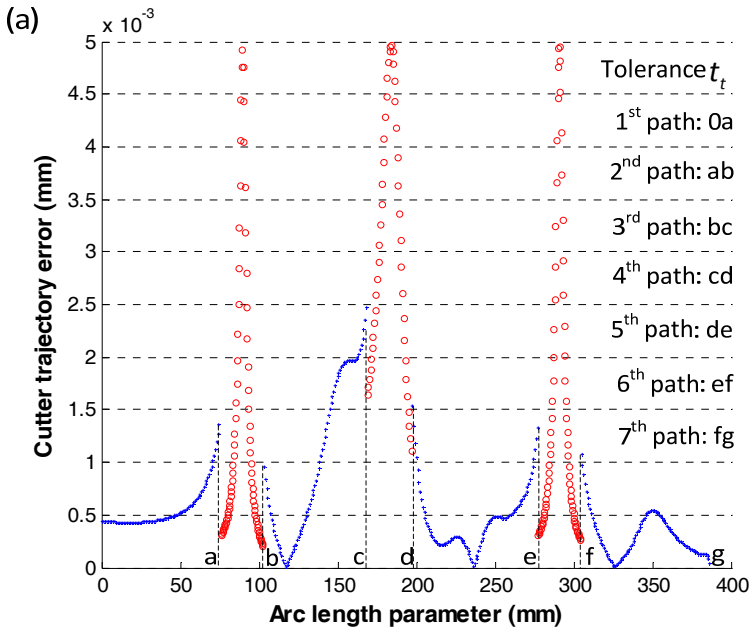


Figure 6.19 The cutter trajectory error curve.

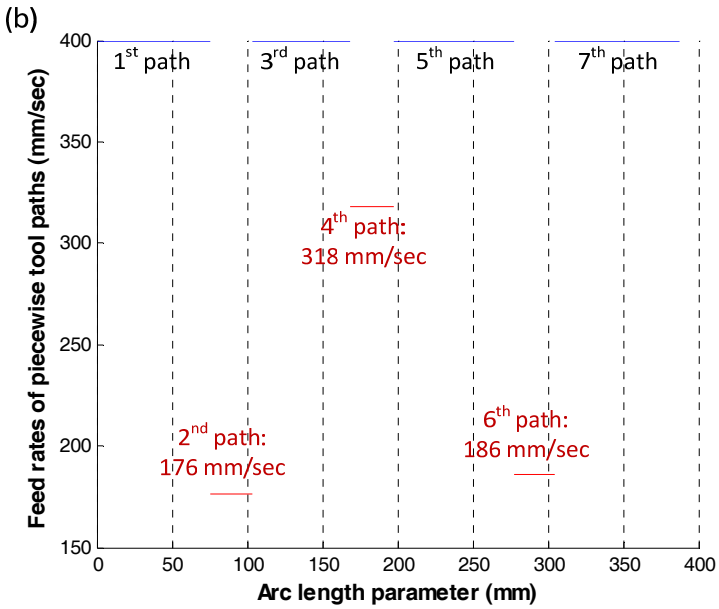


Figure 6.20 The feed rate curve of the piecewise NURBS tool paths with the arc length parameter.

Finally, the piecewise NURBS tool paths are fed into a CNC machine tool. For the validation of the proposed approach we machined the profile shown in Figure 6.13 using two different CL paths; (1) single arc length parameterized NURBS CL path shown in Figure 6.15 with one feed rate 400mm/sec, and (2) piecewise arc length parameterized NURBS CL path shown in Figure 6.18 with different feed rates (see Figure 6.20). The required feed rate is very high for this test which needs high spindle revolutions also. Since, the available machine for this test (OKUMA 5-axis CNC milling center) has maximum 8000 rpm; we selected plastic board made of Polyethylene as work piece material, and sharp pointed metallic tool to cut the plastic along the CL path with high feed rates. This combination of plastic work piece and sharp pointed metallic tool allows machining the profile with desired high feed rates while using low spindle rpm. The machining results are shown in Figure 6.21.

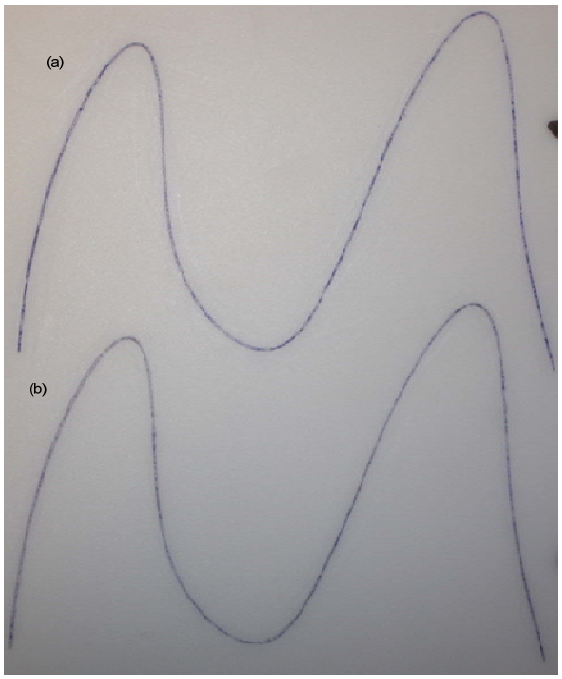


Figure 6.21 A 2-D profile machined using (a) single arc length parameterized NURBS CL path with one feed rate and (b) piecewise arc length parameterized NURBS CL path with different feed rates.

6.8 Summary

This work first proposes a new type of tool path □ piecewise NURBS cutter location paths with the arc length parameter □ and an original approach to generating this type of tool path based on a general NURBS cutter location path with a unit-free parameter. This tool path generated prior to machining is an effective solution to real-time NURBS machining of high trajectory accuracy and good tool kinematics by using the existent NURBS interpolation. The significant contribution of this work is that it eliminates two main sources of large cutter trajectory errors for the real-time NURBS interpolation, which are general NURBS paths with a unit free parameter and high feed rates. In the newly proposed approach, the NURBS decomposition technique is originally applied on cutter location sampling, which

has the following advantages. (1) Cutter locations are automatically sampled and dependent on the input path shape, which means more samples on curved path segments and fewer on straight segments. (2) The arc lengths of the sample cutter locations are simply calculated with several additions without using complex integral calculus. (3) It is very convenient to divide a NURBS tool path into several NURBS pieces for different feed rates. This approach can be quickly implemented into current commercial CAD/CAM software to promote NURBS machining in the manufacturing industry.

Chapter 7 Conclusions and Future Work

In this research, new methods have been developed for gouging free piecewise NURBS tool path generation in pocket and 3-axis milling process and re-parameterization of the NURBS CL path with arc length. The major contributions of this research are summarized as follows:

- A new, general optimization model of gouging and interference detection is established. In this model the geometry of the whole tooling system and the part geometry are considered. For complex part surfaces, gouging detection is a global optimization problem; therefore an improved hybrid, optimization method is introduced which uses discrete PSO method in the rough search stage and the Newton gradient method in the fine search stage in order to take the advantages of the global and local optimization methods. The proposed model is valid for both simple and complex compound sculptured surface milling while using different cutter shapes mostly used in the industry.
- An accurate approach to approximate B-spline offsets with curves in the same form is developed, which is appropriate for CNC pocketing. The research contribution is the upper bound function of the offset error constructed in this work, which can be used to easily estimate the

maximum offset error in order to globally control the offset error. The approximate offsets in B-spline form using the proposed algorithm have fewer control points, less function degree, and all their offset errors are within the specified tolerance.

- This work proposes a new approach to generating accurate, gouging-free, smooth NURBS CL paths for three axis milling. The main contributions of this work include (1) generating arc-length parameterized NURBS CL paths with the path and parameterization errors globally bounded using a new optimization model, (2) conducting gouging and interference detection to eliminate invalid sample CLs before fitting a NURBS CL path to them, which is different from the conventional method, and (3) the close-form equations are derived to calculate the true arc-length for points on the theoretical CL paths.
- A new strategy is proposed to generate piecewise NURBS cutter location paths with the arc length parameter if the NURBS CL path has already been generated with unit free parameter. To estimate the arc length parameter accurately within the tolerance, the technique of NURBS decomposition into Bezier segments is used. The parameterization and path errors are kept within the tolerances. To control the cutter trajectory errors and for smooth feed rate profiles during high speed machining, a new method is introduced to segment the NURBS CL path into pieces with different feed rates.

Computer implementation an experimental tests show the effectiveness of the developed methodologies. The techniques presented in this dissertation can be used to generate NURBS CL paths for the CNC milling process.

For future research, following topics are suggested to expand the present research work:

- Conducting experimental verifications of 3-axis milling using round end mill cutter;
- Extending the upper bound function of the offset error to flat and round end mill cutters in 3-axis milling ;
- Generating NURBS tool-paths for 5-axis surface milling
- Developing a small CAM module based on the proposed algorithms for NURBS machining.

Bibliography

[1] Piegl, L. & Tiller, W., 1997, "The NURBS book", 2nd edition, Springer.

[2] Choi, B.K. & Jerard, R.B., 1998, "Sculptured surface machining: theory and applications", Kluwer.

[3] Koren, Y., 1976, "Interpolation for a computer numerical control system", *IEEE Transactions on Computers,* Vol. C-25, No. 1, pp. 32-37.

[4] Cheng, C. & Tseng, W., 2006, "Design and implementation of a real-time NURBS surface interpolator", *International Journal of Advanced Manufacturing Technology,* Vol. 30, No. 1-2, pp. 98-104.

[5] Makhanov, S.S. & Anotaipaiboon, W., 2007, "Advanced Numerical Methods to Optimize Cutting Operations of Five Axis Milling Machines (Springer Series in Advanced Manufacturing)", Springer.

[6] Yeh, S. & Su, H., 2009, "Implementation of online NURBS curve fitting process on CNC machines", *International Journal of Advanced Manufacturing Technology,* Vol. 40, No. 5-6, pp. 531-540.

[7] Cheng, M., Tsai, M. & Kuo, J., 2002, "Real-time NURBS command generators for CNC servo controllers", *International Journal of Machine Tools and Manufacture,* Vol. 42, No. 7, pp. 801-813.

[8] Piegl, L., 1991, "On NURBS: A survey", *IEEE Computer Graphics and Applications,* Vol. 11, No. 1, pp. 55-71.

[9] *NURBS Interpolation*. Available: http://www.moldmakermag.com/cgi-bin/frame_it_cover.pl?cover8.html [2010, 11/21/2010].

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

[11] Chen, W., Zeng J., Li, L. & Ding Q., 2003, "An approach to gouging avoidance for sculptured surface machining", *Journal of Materials Processing Technology*, Vol. 138 pp. 458-60.

[12] Ouyang, D., Van Nest, B.A. & Feng, H., 2005, "Determining gouge-free ball-end mills for 3D surface machining from point cloud data", *14th International Conference on Flexible Automation and Intelligent Manufacturing*, pp. 338-45.

[13] Zhou, L. & Lin, Y., 2001, "An effective global gouge detection in tool-path planning for freeform surface machining", *International Journal of Advanced Manufacturing Technology,* Vol. 18, No. 7, pp. 461-473.

[14] Oliver, J.H., Wysocki, D.A. & Goodman, E.D., 1993, "Gouge detection algorithms for sculptured surface NC generation", *Journal of Engineering for Industry,* Vol. 115, No. 1, pp. 139-144.

[15] George, K.K. & Ramesh Babu, N., 1995, "On the effective tool path planning algorithms for sculptured surface manufacture", *Computers and Industrial Engineering,* Vol. 28, No. 4, pp. 823-838.

[16] Hatana, A. & Grieve, R.J., 2001, "Pre-processing approach for cutter interference removal", *International Journal of Production Research,* Vol. 39, No. 3, pp. 435-460.

[17] Ding X.M., Fuh, J.Y.H. & Lee, K.S., 2001, "Interference detection for 3-axis mold machining", *Computer Aided Design,* Vol. 33, No. 8, pp. 561-569.

[18] Yang D.C.H. & Han, Z., 1999, "Interference detection and optimal tool selection in 3-axis NC machining of free-form surfaces", *Computer Aided Design,* Vol. 31, No. 5, pp. 303-315.

[19] Chen, Z.C., & Liu, G., 2009, "An intelligent approach to multiple cutters of maximum sizes for three-axis milling of sculptured surface parts", *Journal of Manufacturing Science and Engineering, Transactions of the ASME,* Vol. 131, No. 1, pp. 0145011-0145015.

[20] Kennedy, J. & Eberhart, R., 1995, "Particle swarm optimization", *Neural Networks, 1995. Proceedings, IEEE International Conference on*, pp. 1942.

[21] Panduro, M.A., Brizuela C. A., Balderas L. I. & Acosta D. A., 2009, "A comparison of genetic algorithms, PSO and the differential evolution method for the design of scan able circular antenna arrays", *Progress In Electromagnetics Research,* Vol. 13, , pp. 171-186.

[22] Kim, S. & Yang, M., 2005, "Triangular mesh offset for generalized cutter", *Computer Aided Design,* Vol. 37, No. 10, pp. 999-1014.

[23] Hwang, J.S. & Chang, T., 1998, "Three-axis machining of compound surfaces using flat and filleted end mills", *Computer-Aided Design,* Vol. 30, No. 8, pp. 641-647.

[24] Cobb, E.S., 1984, "Design of Sculptured Surfaces using the B-Spline Representation", Ph.D. thesis.

[25] Coquillart, S., 1987, "Computing Offsets of B-Spline curves", *Computer Aided Design,* Vol. 19, No. 6, pp. 305-309.

[26] Tiller, W. & Hanson, E.G., 1984, "Offsets of Two-Dimensional Profiles", *IEEE Computer Graphics and Applications,* Vol. 4, No. 9, pp. 36-46.

[27] Klass, R., 1983, "An offset spline approximation for plane cubic splines", *Computer Aided Design,* Vol. 15, No. 5, pp. 297-299.

[28] Farouki, R.T. & Sakkalis, I., 1990, "Pythagorean hodographs", *IBM Journal of Research and Development,* Vol. 34, No. 5, pp. 736-752.

[29] Pham, B., 1988, "Offset approximation of uniform B-splines", *Computer Aided Design,* Vol. 20, No. 8, pp. 471-474.

[30] Hoschek, J., 1988, "Spline Approximation of Offset Curves", *Computer Aided Geometric Design,* Vol. 5, No. 1, pp. 33-40.

[31] Hoschek, J. & Wissel, N., 1988, "Optimal approximate conversion of spline curves and spline approximation of offset curves", *Computer Aided Design,* Vol. 20, No. 8, pp. 475-483.

[32] Elber, G. & Cohen, E., 1991, "Error bounded variable distance offset operator for free form curves and surfaces", *International Journal of Computational Geometry & Applications,* Vol. 1, No. 1, pp. 67-78.

[33] Elber, G. & Cohen, E., 1992, "Offset approximation improvement by control point perturbation", *Mathematical methods in computer aided geometric design II,* pp. 229-237.

[34] Elber, G., Lee I.K. & Kim M.S., 1997, "Comparing offset curve approximation methods", *IEEE Computer Graphics and Applications,* Vol. 17, No. 3, pp. 62-71.

[35] Lee I.K., Kim M.S. & Elber, G., 1996, "Planar curve offset based on circle approximation", *Computer Aided Design,* Vol. 28, No. 8, pp. 617-30.

[36] Lee I.K., Kim M.S. & Elber, G., 1997, "New approximation methods for planar offset and convolution curves", *Proceedings of International Conference on Theory and Practice of Geometric Modeling*, Berlin, Germany, pp. 83-101.

[37] Lee I.K., Kim M.S. & Elber, G., 1998, "Polynomial/rational approximation of Minkowski sum boundary curves", *Graphical Models and Image Processing,* Vol. 60, No. 2, pp. 136-65.

[38] Maekawa, T. 1999, "An overview of offset curves and surfaces", *Computer Aided Design,* Vol. 31, No. 3, pp. 165-73.

[39] Piegl, L. & Tiller, W., 1999, "Computing offsets of NURBS curves and surfaces", *Computer Aided Design,* Vol. 31, No. 2, pp. 147-56.

[40] Young, J.A., Yeon, s.K. & Shin, Y., 2004, "Approximation of circular arcs and offset curves by Bezier curves of high degree", *Journal of Computational and Applied Mathematics,* Vol. 167, No. 2, pp. 405-16.

[41] Zhao, H. & Wang, G., 2007, "Error analysis of re parameterization based approaches for curve offsetting", *Computer-Aided Design,* Vol. 39, No. 2, pp. 142-148.

[42] Wang, F. & Yang, D.C.H., 1993, "Nearly arc-length parameterized quintic-spline interpolation for precision machining", *Computer-Aided Design,* Vol. 25, No. 5, pp. 281-288.

[43] Yang, D.C.H. & Wang, F., 1992, "Quintic spline interpolator for motion command generation of computer-controlled machines", *ASME, Design Engineering Division (Publication) DE*, Vol. 46, pp. 537-544.

[44] Wang, F., & Wright, P.K., 1998, "Open architecture controllers for machine tools, Part 2: a real time quintic spline interpolator", *Journal of Manufacturing Science and Engineering, Transactions of the ASME,* Vol. 120, No. 2, pp. 425-432.

[45] Emami, M.M., & Arezoo, B., 2010, "A look-ahead command generator with control over trajectory and chord error for NURBS curve with unknown arc length", *Computer Aided Design,* Vol. 42, No. 7, pp. 625-632.

[46] Heng, M. & Erkorkmaz, K., 2010, "Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability", *International Journal of Machine Tools and Manufacture,* Vol. 50, No. 3, pp. 281-293.

[47] Wang, F., Wright, P.K., Barsky, B.A. & Yang, D.C.H., 1999, "Approximately arc-length parameterized C3 quintic interpolatory splines", *Journal of Mechanical Design, Transactions Of the ASME,* Vol. 121, No. 3, pp. 430-439.

[48] Erkorkmaz, K. & Altintas, Y., 2005, "Quintic spline interpolation with minimal feed fluctuation", *Transactions of the ASME Journal of Manufacturing Science and Engineering,* Vol. 127, No. 2, pp. 339-49.

[49] Zhu, T., Wong, Y.S. & Mannan, M.A., 2005, "A composite B-spline method for cutter path generation on free-form surfaces", *International Journal of Computer Applications in Technology,* Vol. 24, No. 2, pp. 83-8.

[50] Lartigue, C., Thiebaut, F. & Maekawa, T., 2001, "CNC tool path in terms of B-spline curves", *Computer Aided Design,* Vol. 33, No. 4, pp. 307-19.

[51] Yau, H.T. & Kuo M.J., 2001, "NURBS machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces", *International Journal of Production Research,* Vol. 39, No. 1, pp. 21-41.

[52] Langeron, J.M., Due, E., Lartigue, C. & Bourdet, P., 2004, "A new format for 5-axis tool path computation, using B-Spline curves", *Computer Aided Design,* Vol. 36, No. 12, pp. 1219-29.

[53] Yang, X. & Chen, Z.C. 2005, "A new high precision fitting approach for NURBS tool paths generation", *DETC2005: ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, September 24, 2005 - September 28* American Society of Mechanical Engineers, Long Beach, CA, United states, pp. 255.

[54] Chen, Z.C. & Yang, X., 2007, "A practical approach to generating accurate NURBS tool paths for CNC machining of sculptured surface parts", *Materials Science Forum,* Vol. 532-533, , pp. 500-3.

[55] Bey, M., Boudjouad, S. & Tafat-Bouzid, N., 2007, "Tool-path generation for free-form surfaces with B-Spline curves", *Strojniski Vestnik,* Vol. 53, No. 11, pp. 733-41.

[56] Lin, K.Y., Ueng, W.D. & Lai, J.Y., 2008, "CNC codes conversion from linear and circular paths to NURBS curves", *International Journal of Advanced Manufacturing Technology,* Vol. 39, No. 7-8, pp. 760-73.

[57] Li, W., Liu, Y., Yamazaki, K., Fujisima, M., & Mori, M., 2008, "The design of a NURBS pre-interpolator for five-axis machining", *International Journal of Advanced Manufacturing Technology,* Vol. 36, No. 9-10, pp. 927-935.

[58] Shih, J. & Frank Chuang, S., 2008, "One-sided offset approximation of freeform curves for interference-free NURBS machining", *Computer Aided Design,* Vol. 40, No. 9, pp. 931-937.

[59] Lai, Y.L., 2010, "Tool-path generation of planar NURBS curves", *Robotics and Computer-Integrated Manufacturing,* Vol. 26, No. 5, pp. 471-82.

[60] Rogers, D.F. & Fog, N.G., 1989, "Constrained B-spline curve and surface fitting", *Computer Aided Design,* Vol. 21, No. 10, pp. 641-8.

[61] Ma, W. & Kruth, J.P., 1995, "Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces", *Computer Aided Design,* Vol. 27, No. 9, pp. 663-675.

[62] Park, H., Kim, K. & Lee, S., 2000, "A method for approximate NURBS curve compatibility based on multiple curve refitting", *Computer Aided Design,* Vol. 32, No. 4, pp. 237-52.

[63] Borges, C.F. & Pastva, T., 2002, "Total least squares fitting of Bezier and B-spline curves to ordered data", *Computer-Aided Geometric Design,* Vol. 19, No. 4, pp. 275-89.

[64] Park, H., & Lee, J.H., 2007, "B-spline curve fitting based on adaptive curve refinement using dominant points", *Computer Aided Design,* Vol. 39, No. 6, pp. 439-51.

[65] Brujic, D., Ainsworth, I. & Ristic, M., 2010, "Fast and accurate NURBS fitting for reverse engineering", *International Journal of Advanced Manufacturing Technology*, pp. 1-10.

[66] Bedi, S., Ali, I. & Quan, N., 1993, "Advanced interpolation techniques for NC machines", *Transactions of the ASME Journal of Engineering for Industry,* Vol. 115, No. 3, pp. 329-36.

[67] Shpitalni, M., Koren, Y. & Lo, C.C., 1994, "Real time curve interpolators", *Computer Aided Design,* Vol. 26, No. 11, pp. 832-8.

[68] Yang, D.C.H. & Kong, T., 1994, "Parametric interpolator versus linear interpolator for precision CNC machining", *Computer Aided Design,* Vol. 26, No. 3, pp. 225-34.

[69] Yeh, S.S. & Hsu, P.L., 2002, "Adaptive feed-rate interpolation for parametric curves with a confined chord error", *Computer Aided Design,* Vol. 34, No. 3, pp. 229-37.

[70] X. Zhiming, C. Jincheng, & F. Zhengjin, 2002, "Performance evaluation of a real-time interpolation algorithm for NURBS curves", *International Journal of Advanced Manufacturing Technology,* Vol. 20, No. 4, pp. 270-6.

[71] Lo C.C. 1997, "Feedback interpolators for CNC machine tools", *Transactions of the ASME Journal of Manufacturing Science and Engineering,* Vol. 119, No. 4, pp. 587-92.

[72] Cheng, M., Tsai, M. & Kuo, J., 2002, "Real-time NURBS command generators for CNC servo controllers", *International Journal of Machine Tools & Manufacture,* Vol. 42, No. 7, pp. 801-13.

[73] Zhou, K., Wang, G., Jin, H. & Tan, Z., 2008, "NURBS interpolation based on exponential smoothing forecasting", *International Journal of Advanced Manufacturing Technology,* Vol. 39, No. 11-12, pp. 1190-1196.

[74] Yeh, S. & Hsu, P., 1999, "The speed-controlled interpolator for machining parametric curves", *Computer Aided Design,* Vol. 31, No. 5, pp. 349-57.

[75] Park, P., Soon, Y.J. & Yun, J.C., 2006, "Parametric interpolation using sampled data", *Computer Aided Design,* Vol. 38, No. 1, pp. 39-47.

[76] Lei, W.T. & Wang, S.B., 2009, "Robust real-time NURBS path interpolators", *International Journal of Machine Tools & Manufacture,* Vol. 49, No. 7-8, pp. 625-33.

[77] Erkorkmaz, K. & Altintas, Y., 2001, "High speed CNC system design. Part I: Jerk limited trajectory generation and quintic spline interpolation", *International Journal of Machine Tools and Manufacture,* Vol. 41, No. 9, pp. 1323-1345.

[78] Lei, W.T., Sung, M.P., Lin, L.Y. & Huang, J.J., 2007, "Fast real-time NURBS path interpolation for CNC machine tools", *International Journal of Machine Tools & Manufacture,* Vol. 47, No. 10, pp. 1530-41.

[79] Sharpe, R.J. & Thorne, R.W., 1982, "Numerical method for extracting an arc length parameterization from parametric curves", *Computer-Aided Design,* Vol. 14, No. 2, pp. 79-81.

[80] Farouki, R.T., 1997, "Optimal parameterizations", *Computer Aided Geometric Design,* Vol. 14, No. 2, pp. 153-168.

[81] Joseph (Yossi) Gil, Daniel Keren, 1997, "New approach to the arc length parameterization problem", *13th Spring Conference on Computer Graphics*, pp. 27-34.

[82] Hernández-Mederos, V. & Estrada-Sarlabous, J., 2003, "Sampling points on regular parametric curves with control of their distribution", *Computer Aided Geometric Design,* Vol. 20, No. 6, pp. 363-382.

VITA

Surname: Khan Given Name: Maqsood Ahmed

EDUCATION

Ph.D.	Department of Mechanical and Industrial Engineering Concordia University, Montreal, Quebec, Canada	2007-2011
M.Sc.	Department of Mechanical Engineering NED University of Eng. and Tech., Karachi, Pakistan	1999-2001
B.E.	Department of Mechanical Engineering NED University of Eng. and Tech., Karachi, Pakistan	1994-1998

AWARDS AND TRAINING

Scholarship for Ph.D., NED University of Eng. and Tech.	2007-2010
One month training for CNC Wire EDM, Taiwan	2005
Two months training for CNC machines, Switzerland	2001

PUBLICATIONS

- Maqsood A. Khan, Z. C. Chen, "Piecewise NURBS Tool Path with the Arc Length Parameter and their Application on High Feed, Accuracy CNC Milling of 2-D Curved Profiles", *Submitted to Journal of Manufacturing Science and Engineering, Transactions of ASME, 2010*.
- Z. C. Chen, Maqsood A. Khan, "A New Approach to Generating Arc Length Parameterized NURBS Tool Paths for Efficient 3-Axis Machining of Smooth, Accurate Sculptured Surfaces", Submitted to *Computer-Aided Design*, 2010.
- Maqsood A. Khan, Z. C. Chen, "Generating Planar Offset Curves with Globally Bounded Offset Error for B-Spline NC Tool Paths", *Submitted to International Journal of Production Research, 2010.*
- Maqsood A. Khan, Z. C. Chen, "An Integrated Approach to Generating CL-paths in B-Spline Form with Global Error Control for 3-Axis Sculptured Surface Machining", *Submitted to International Journal of Machine Tools and Manufacture, 2010.*
- Maqsood A. Khan, Z. C. Chen, "Gouging Detection and Removal for 3-Axis Machining Using Improved Hybrid Optimization Method", *Submitted to International Journal of Advanced Manufacturing Technology, 2010.*

- Maqsood A. Khan, Z. C. Chen, "Arc-length parameterized NURBS path interpolation for high speed and precision machining", *Proceeding of the 19th International Symposium on Air Breathing Engines,* September 13-19, 2009, Montreal, Quebec, Canada

- Maqsood A. Khan, Z. C. Chen, "An Effective Approach to Approximating 2-D Free-Form Curve Offsets for B-Spline NC Tool Paths with Offset Error Globally Bounded", *Proceedings of the Canadian Society for Mechanical Engineering Forum 2010*,June 7-9, 2010, Victoria, British Columbia, Canada

- Maqsood A. Khan, Z. C. Chen, "An Integrated Approach to Generating Accurate Cutter Location Path with Approximate Arc Length Parameter", *Proceedings of the 9th International Conference on Frontiers of Design and Manufacturing*, July 17-19, 2010, Changsha, China

- Maqsood A. Khan, Z. C. Chen, "A new Approach to Generating Accurate NURBS Cutter Location paths with Arc Length Parameter", *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 15-18, 2010, Montreal, Quebec, Canada

- Maqsood A. Khan, Z. C. Chen, "An Effective Approach to Approximating 2-D Free-Form Curve Offsets for B-Spline NC Tool Paths with Offset Error Globally Bounded", *Proceedings of the ASME 2010 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, August 15-18, 2010, Montreal, Quebec, Canada