THREE-AXIS TOOL-PATH B-SPLINE FITTING BASED ON PREPROCESSING, LEAST SQUARE APPROXIMATION AND ENERGY MINIMIZATION AND ITS QUALITY EVALUATION

Changya Yan[1], Chenhan Lee[1], Jianzhong Yang[2]
[1]Wuhan Huazhong Numerical Control Co., Ltd, Wuhan, P. R. China
[2]National NC System Engineering Research Center,
Huazhong University of Science and Technology, Wuhan, P. R. China

e-mail: chenhanlee@hotmail.com

Current state-of-art CAM systems generate piecewise linear (G01-based) tool paths that lack G1 and G2 continuity. This paper presents a systematic solution of tool path B-spline fitting for improving the continuity and quality and a tool set to evaluate the quality of either G-Code commands or CNC-interpolated axis motion commands. The core of the solution involves a preprocessor and a cubic B-Spline fitting algorithm. The preprocessor identifies feature points (Hard Break Point Identification) and uses a point smoothing algorithm to improve the quality of the data points by minimizing the length sum subject to shape preserving and tolerance constraints. A B-Spline fitting algorithm based on least square approximation and energy minimization is used to ensure fair (shape-preserving) tool-paths with G2 continuity and stable curvature transition. The evaluation tool set computes quality criterions, such as G2 discontinuity locations, curvature, inflexion, jerks and torsion in the G-code or CNC-interpolated axis motion commands. All above algorithms are integrated into a software package. The validation of the effectiveness and efficiency of above solutions is performed by comparing tool-paths with and without the preprocessing and B-Spline fitting.

Keywords
B-spline fitting, shape preserving, point smoothing, tool path quality evaluation

1. Survey and outline

For a tool path, any discontinuities of the first and second derivative as well as large value of second derivative (curvature) may cause unsmooth path flow. In high-speed CNC machining (HSM), the tool needs to be traversed at high feedrate, so that a smoothed tool path is preferred. G01-based tool paths lack G1 and G2 continuity (for the definitions of parametric continuity and geometric continuity please refer to [Barsky and DeRose 1989]). They need to be smoothed before being put into high-speed machining.

Generally there are two approaches to improve the quality of G01-based tool path: interpolation or approximation, by smooth parametric curves to a given tolerance. Unlike interpolation, approximation (fitting) allows a small deviation from the data points, which makes approximation a better choice for tool path smoothing.

A number of researchers have put effects in fair and shape preserving approximation for fitting a curve to a set of data points [Celniker and Gossard 1991; Veltkamp and Wesselink 1995; Vassilev 1996; Poliakoff et al. 1999; Zhang et al. 2001; Xu et al. 2011]. Curve fitting can be more or less improved by combining energy terms with the least-square error criterion[Vassilev 1996; Park et al. 2000].

A general framework for least-square B-Spline fitting is as follows:

1) Parameter selection: find a set of parameters that can "fixed" the data points at certain values, some literatures also call them "foot points".
2) Knot vector generation. A uniformly spaced knot vector often but not always gives good results.
3) Form a fitting objective function, in which fitting error term is a necessity but weighted energy terms are optional depending on practical applications.
4) Initial a B-Spline construction.
5) Use iterative method to find an optimization solution which brings a minimized value of fitting objective function.

Many works proposed their energy term definitions, among them [Veltkamp and Wesselink 1995] provides a most complete one. The objective of curve smoothing is to find the curve shape with some certain kind of minimal energy, for example, stretching, bending and torsion energy. Then the fair curve fitting turns into a constraint nonlinear optimization problem. [Vassilev 1996] reported fair interpolation and approximation of B-Splines by energy minimization and points insertion, which is suitable for the case of measured (or digitized) data when the points have been originally planned to be evenly distributed. Usually to solve the energy terms, complicated integral calculation is needed, which cannot be solved numerically. Then some simplified energy formulations and approaches are proposed [Moreton and Séquin 1994; Eck and Hadenfeld 1995; Bajaj et al. 1998; G. Farin 2002], in which iterative methods are implemented, but the convergence rate is slow. [Wang et al. 2004; Wang et al. 2006; Liu and Wang 2008] construct similar iterative methods which consider the relationship between shape parameters and location parameters of the curve to improve convergence rate. Among them, [Wang et al. 2006] uses a curvature-based quadratic approximant of squared distances from data points to form the squared distance error term during least square fitting. [Dierckx 1995] and [Kvasov 2000] adopted planar spline curves to accomplish shape preserving approximation. [Costantini and Pelosi 2001; Costantini and Pelosi 2004] approximate an ordered set of spatial points with shape preserving properties, but the degree of the resulting spline are variable, which is not fit for CNC tool path representation.[Mei 2010] uses an adaptive method to select feature points and then implemented a quadratic B-Spline interpolation algorithm. The obtained B-Spline curve has only C1 continuity.[Park and Lee 2007] proposed an approach which adaptively refines a B-Spline curve by selecting dominant points and determining knot placement accordingly, then the B-Spline curve is obtained by least squares minimization. The dominant points properly selected from the given points play an important role in yielding better approximations.

Due to almost limitless distribution forms of data points(there are highly chances of dealing with noisy data points), no methods published can fit any ordered data points to a certain tolerance-band while still keeping the curve shape-preserving and smooth. From previous practices, in addition to an effective curve fitting algorithm, most of the tool path fitting solutions need a pre-processing stage in order to obtain better fitting results.

Tool path quality is a very broad topic, some previous work have provided effective but far from complete and systematical solutions. Some commercial NC software provides tools to observe the curvature/derivative curvature plot. [Stefan K¨unzli 2010] uses two methods to evaluate the quality of spline curves: one is the integral

of the derivative of the curvature; the other is the number of sign changes of the derivative of the curvature. [Tapie et al. 2007] provides a tool path quality visualization solution, which use tangential and curvature discontinuities to calculate feed rate decreases. This solution needs not only geometrical models of tool path, but also the technical constraint models of the machine and tool.

In the view of the authors, the following problems still exist:

1) Lack of a robust method to identify and retain the feature points (sharp corners) in the toolpath. If the feature points are not identified correctly, they are fitted by B-spline curves and the intended features are lost after machining.
2) Lack of a robust method to handle highly uneven distributed data points and ensure shape-preserving trajectories, even by applying tension properties and other energy criterions.
3) Lack of a systematical and visual solution to evaluate and interactively improve the tool path quality

This paper aims to provide:

1) A point smoothing algorithm used as an effective preprocessing method to smooth linear data points, help detecting break points (feature points) and alleviating the influence of small defects of path.
2) An effective tool path fitting algorithm, by executing the following process flow: preprocessing->pre-fitting->sampling->B-Spline fitting->blending two B-Splines to make C2 continuous transition, to obtain smoothed, shape preserving and tolerance banded cubic B-Spline tool path.
3) A general model for tool path quality evaluation, and its software solution framework.

We assume that the readers are familiar with the concepts of B-Spline curves and general B-Spline fitting algorithms. And the paper is arranged as follows: In the next section, the preprocessing is introduced, by the order of hard break point identification, subdivision strategy and point smoothing algorithm. Section 3 presents B-Spline fitting process, which includes four steps: Pre-fitting by Shape-preserving C1 B-Splines, reselect sample data points, fitting new sample data points by C2 continuous cubic B-Spline, and then blend these B-Splines by a new cubic B-Spline. Then in section 4, a tool path quality evaluation tool set is presented, which use mathematical descriptions of feature and differential properties of a tool path to evaluate the tool path quality. An application example is given in the next section and then follows the summary and future work plans.

2. PREPROCESSING before fitting

2.1 Overview

Before B-Spline fitting, the original data points need to be processed and then generate new point sequence that is more suitable for further fitting. The data points must first be grouped by "breaks" which should be identified by data analysis. The data points between "breaks" should be further segmented and then their distribution be smoothed.

2.2 Hard Break Points Identification

The number of G01 points in a tool path may be very large that it is impossible to fit them by only one spline. Furthermore, feature points, motion-change locations and long-length lines should be kept unchanged during fitting. The data points to be fitted should be grouped by these "tool path stops", which are called "Hard Break Points (HBP)" in this paper. The aim of Hard Break Points Identification is to group the data points by "HBPs", and BSpline fitting is applied on G01 blocks between two successive HBPs.

Tool path is converted into G-codes by CAM systems with certain tolerance requirement, after that the tolerance information is lost. In most cases, we can only rely on coordinates of data points to analyze feature information of the tool path and identify them as HBP, as shown in Fig. 1. HBPs can be classified as three types:

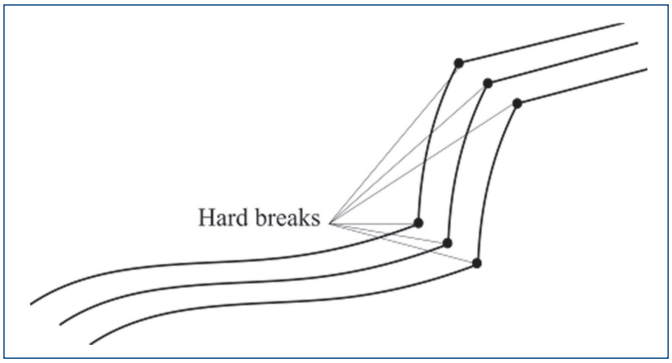

Figure 1. *Hard Break Points on Tool Path*

1) Points representing the ends of a feature part.
2) Points representing a change of motion type.
3) Points belonging to the ends of long lines(representing feature edges) and other analytical lines, such as arcs, conics and helices, etc.

A HBP identification algorithm is listed as below:
Inputs: 3-D coordinates of G01 data point sequence (each point is assigned a unique index)
Out-puts: point indices denoting hard breaks

Step1: line fitting

1) Point reduction: adjacent points of tool path within a certain tolerance range will be reduced to one point.
2) Poly-line simplification: collinear points under a certain tolerance criterion will reduced to a two-point segment, and all middle points on that "segment" will be "omitted".

Step 2: HBP identification

1) Classification of original data points: calculate the distance d of two successive points and the angle θ between two successive segments. d_1 denotes the distance between P_1 and P_{1-1}. θ_i denotes the angle between line segment

$$\overline{P_{i-1}P_i} \quad \text{and} \quad \overline{P_iP_{i+1}}$$

(Fig. 2). According to the distance and angle results, all data points can be classified as following:

a) If the angle value θ exceeds a threshold, assign the corresponding data points a property tag as an absolute HBP(*isAbsHBP*).
b) If the angle value θ is lower than a certain value, assign the corresponding data points an *isParallel* property (successive segments are nearly parallel).
c) If the angle value θ is between isAbsHBP and isParallel, the corresponding data points are identified as HBP candidates *(isHBPCadidate*).
d) Point that belongs to a long length line(*isLongDis*)
e) Point that divides two segments with remarkable different length (*isUneven*)

All above definitions need the support of certain "threshold values", which can be determined by engineering experiences and trial-and-error test.

2) HBP refinement process based on the angle of tangential vector of successive segments computed by three-point-arc interpolation. This step is specifically applied on points with *isHBPCadidate* property, which are potential HBPs. A three-

point-arc method [Yau and Wang 2007] is introduced. Neighboring points' *isLongDis/isUneven/isParallel* property will be important supplements during the calculation.

a) If an *isHBPCadidate* point's both neighboring segments have *isLongDis* or *isUneven* property, make this *isHBPCadidate* point a concrete HBP.

b) Else initialize the three-point-arc method to build two arcs passing through triplet adjacent points P_{i-2},P_{i-1},P_i and P_i,P_{i+1},P_{i+2} respectively. As shown in Fig. 2. Two arcs produce two tangential vector on Point P_i: V_a and V_b.
Take the angle between V_a and V_b as the angle value θ between G01 block

$$\overline{P_{i-1}P_i} \quad \text{and} \quad \overline{P_iP_{i+1}}$$

c) Calculate a compound parameter for every data point P_{ci} according to Eq. (1).

$$P_{ci} = f(\theta, d_i, d_{i+1}) = 2^B \theta^2 \tag{1}$$

Where $B=sign(\max(d_i/d_{i+1}, d_{i+1}/d_i)-2)$.

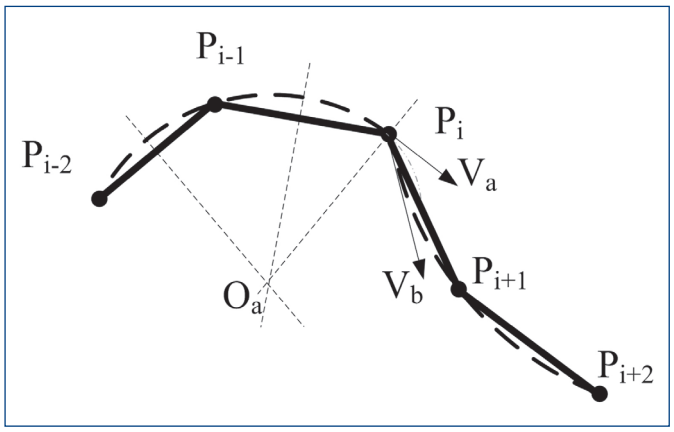

Figure 2. *Compute angle θ by three-point-arc interpolation*

This compound parameter is based on θ and the length values of successive segments around point P_i. If one of the lengths is two times longer or shorter than the other, the possibility that these triplet points belong to an arc segment or a flattened free curve is lower, so the compound parameter calculation will have a higher proportional factor.

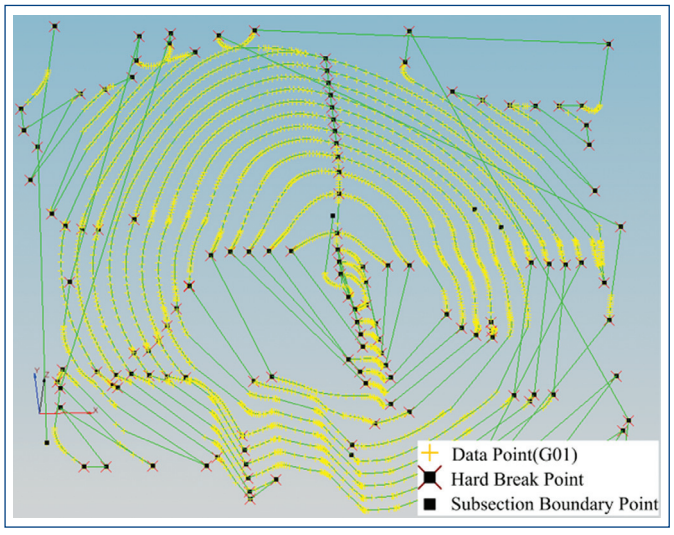

Figure 3. *HBPs identification and subsectioning of a sample tool path*

d) Calculate a threshold value:

$$P_{th} = \sqrt{P_c^{Mean} P_c^{Max}} \tag{1}$$

Where $P_c^{Mean} = \frac{1}{n}\sum_{i=1}^{n} P_{ci}$ and $P_c^{Max} = MAX(P_{ci})$

3) Based on the above information, we can identify all HBPs, which come through three sources:
   a) All data points with *isAbsHBP* property.
   b) All data points which belongs to a line segment with *isLongDis* property.
   c) Any data points with *isHBPCadidate* property and if its compound parameter P_{ci} satisfies $P_{ci} > P_{th}$.

Fig. 3 shows a tool path with all Hard Breaks Points being identified.

2.3 Subsection Division

We need to break a long tool path between two HBPs with large number of data points into several small subsections. The main reasons are:

1) After HBP identification, the number of input points for one B-Spline fitting could still be between a few to a few thousands.
2) The efficiency of point smoothing and B-Spline approximation algorithm is sensitive to the number of data points.
3) It is difficult for the global B-Spline approximation algorithm to handle large number of data points.

A subsection will be a unit for point smoothing and B-Spline fitting. Afterwards the"break"needs to be blended by another G2 Continuity B-Spline to keep the long tool path still be G2 everywhere. These breaks are called "soft breaks".

To control the subsection division strategy, two conditions need to be satisfied:

1) The number of data points in one subsection needs to be in a certain range (for example, between 10 to 30).
2) Because higher curvature regions need more local control during fitting, the transition property at soft breaks should be as smooth as possible.

Here a quasi curvature value calculated as Eq. is used to evaluate one data point's curvature on the tool path. The subsectioning process helps to further group subsequences of the data points which have sufficiently simple shape to admit accurate fitting.

Fig. 3 also shows the G01 blocks sequence between two HBPs is further divided into several subsections at data point with lower curvature transition (lower compound parameter value).

2.4 Point Smoothing

Many G01 based tool paths have a tendency to be noisy and fluctuating. A point smoothing algorithm is proposed as a preprocessing process to improve the smoothness of the noisy data points between any successive HBPs. For better illustration, an example with noisy and fluctuating G01 blocks is shown in Fig. 4.

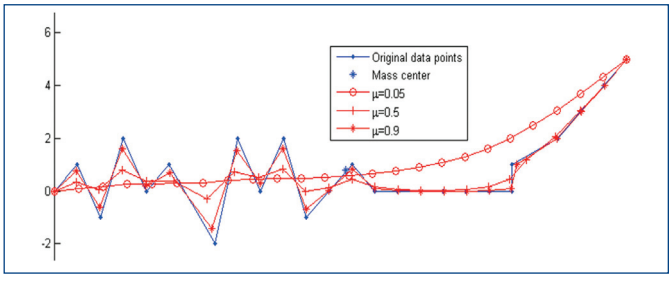

Figure 4. *Point smoothing of noisy and fluctuating point sequence*

This problem can be modeled as an equality constraint nonlinear optimization problem. Given a poly-line constructed by ordered spatial points, find same number of new points to form a new poly-line, which is the approximation of the original one and satisfies the following conditions:

1) The new ploy-line has minimum length (stretching energy is minimized).
2) The new ploy-line is shape preserving to the original one.

Given n+1 data points p_0, p_1, ..., p_n, find a point set containing n+1 points, X={x_0, x_i, ... x_n}, which makes the following objective function minimized:

$$\min(f(X)) : f(X) = \sum_{i=0}^{n} \left| x_{i+1} - x_i \right|^2 + \mu \left| x_i - p_i \right|^2$$
$$s.t. \quad \sum_{i=0}^{n} \left| x_i - P_m \right|^2 = C = \sum_{i=0}^{n} \left| P_i - P_m \right|^2 \qquad (3)$$

Where P_m denotes the center of mass of all data points,

$$P_m = \frac{1}{n+1} \sum_{i=0}^{n} P_i .$$

An SQP method[Spellucci 1998; Spellucci 1998] is adopted to solve the constraint optimization problem.

A weighting factor μ is added to allow the freedom to control the importance of stretching and shape preserving requirements. As shown in Fig. 4, with a lower μ value, the optimized new poly-line is more stretched, and each data point is dragged further away from its original counterpart. It can be easily deduced that when μ=0, the result will be a straight line segment which is the shortest way between two 3D points. With a higher μ value, the resulting poly-line will be closer to the original one. Especially, when μ=1, the result will be as same as the original one, which has the best shape preservation property but no stretching energy is optimized to improve the smoothness of the original tool path.

A higher μ value (for example μ=0.8) should be used when the algorithm is applied to smooth noisy data points of tool path with tolerance requirements. The algorithm is also applicable to smooth noisy data points with less critical shape preserving requirement, in which case a lower μ value (μ=0.5) is preferred.

The point smoothing algorithm is used in this paper as a preprocessing step to improve the smoothness of the original data point sequence, which will be a better fitting target for the B-Spline fitting algorithm in section 4. Fig. 5 reveals its effectiveness for better B-Spline fitting applied on noisy and fluctuating data points.

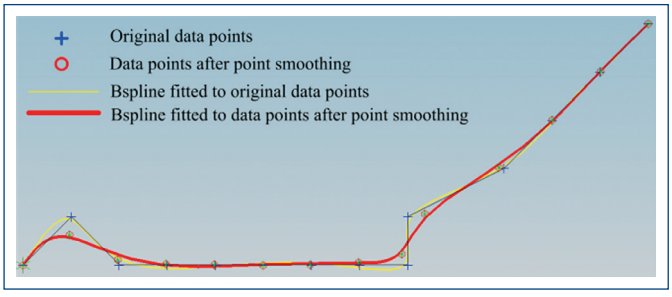

Figure 5. *B-Spline Fitting(by Open Cascade variational smoothing algorithm) After Point Smoothing*

3. B-spline fitting

Overview

After pre-processing, we reduced the tool path fitting problem into B-Spline approximations of data points on subsections. First a shape preserving B-Spline with G1 continuity will be built, based on which a sampling processing will reselect data points with optimized distribution forms. A cubic B-Spline will fit these sampled data points. Afterwards a B-Spline blending algorithm will be executed to guarantee G2 continuity between subsections.

In this paper we adopted a B-Spline fitting algorithm based on least square approximation and weighted energy minimization to fit the discrete G01 data points, which is provided by the Open CASCADE software development platform. Two types of B-Spline Approximation algorithms are used:

1) Approximate a B-Spline Curve passing through an array of points. The resulting B-Spline will satisfy the specified degree, continuity and tolerance requirements.
2) Approximate a B-Spline Curve passing through an array of points using variational smoothing algorithm, which tries to minimize additional energy criterium: Weight1*CurveLength + Weight2*Curvature + Weight3*Torsion.

3.2 Pre-fitting

Tool path with large number of data points between two successive HBPs is split into one or more subsections. Point smoothing algorithm is firstly applied on each subsection, then least-square fitting with weighted energy terms is used to fit these data points within certain tolerance band. Energy terms in the objection function could be any combination of stretching/bending/torsion energy, each of them contributes improvement on one aspect of the resulting curves: *Curve is more stretched /curve is smoother without high curvature/ Curve is trying to be kept in plane.*

Because of the uneven distribution of data points, it's hard to fit all data points with one B-Spline curve with C2 continuity, and at the same time satisfying tolerance band requirements and better smooth quality. In this pre-fitting stage, we can lower our expectation to obtain a C1 B-Spline curve with optimized energy distribution , that is to say, the pre-fitting resulting curve(denoted by B-Spline0) will be C1 continuous, but have good shape preserving property.

3.3 Sampling and Building a longer Cubic B-Spline

In order to generate a smooth B-Spline curve with less deviation from the data point set and fewer control points, we need fewer data points on flatten regions, and more points on complex, high curvature regions(such as corners and transitions). On the B-Spline0, select the sample points based on the area of chord deviation. Then use the resulting sample points of one subsection to build a longer B-Spline.

OCC API function class ***AppDef_BSplineCompute*** allows us to build a 3D B-Spline curve which approximates a set of points. The client program has to define the lowest and highest degree of the curve (in this case both should be cubic), its continuity requirement (C2) and a tolerance value for it. The resulting B-Spline will be cubic and has C2 continuity. Because the boundary points between subsections are chosen at low curvature locations, the tangent vector of the boundary point on each of successive subsections will be nearly the same (C1 continuity is naturally guaranteed). In fact this requirement could even be integrated into Cubic B-Spline approximation algorithms as tangential point constraints on the first and last data points. This property could be an added benefit for next stage's B-Spline blending algorithm.

3.4 B-Spline Blending algorithm

Now each B-Spline curve obtained by fitting the sampled data points of subsection has G2 continuity, but at the joining point between every successive subsections has only G0 continuity(in the worst case). We need a blending algorithm to connect these two B-Spline curves by a blending cubic B-Spline with G2 continuity on the joining point. Then the tool path between HBPs will be a piecewise B-Spline curve with G2 continuity everywhere.

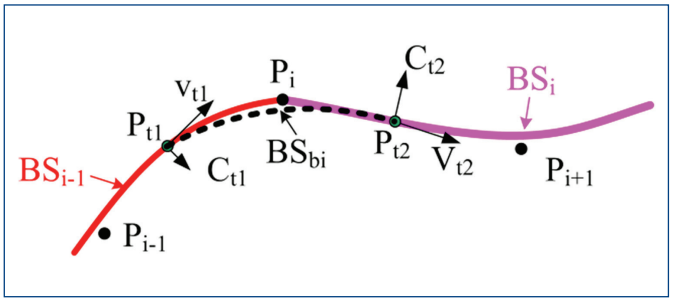

Figure 6. *Schematic view of B-Spline blending*

Fig. 6 illustrates the schematic view of B-Spline blending algorithm. P_i is a joining point of BS_{i-1}(B-Spline1 of subsection(i-1)) and BS_i(B-Spline1 of subsection(i)). P_{i-1} is P_i's closest data point on subsection (i-1) and P_{i+1} is P_i's closest data point on subsection (i). The blending B-Spline BS_{bi} will start at P_{t1}, which lies on BS_{i-1} and has G2 continuity between BS_{i-1} and BS_{bi}, and end at P_{t2}, which lies on BS_i and has G2 continuity between BS_{bi} and BS_i. Keep in mind that according to the subsectioning algorithm, both the original tool path and the fitted B-Spline1 should have lower curvature values at P_i.

The algorithm steps are listed as follows:

1) Determining P_{t1} on BS_{i-1} and P_{t2} on BS_i.
   a) Query the curve property of the trimmed part of BS_{i-1} between P_{i-1} and P_i. If there are inflexion points, marked the closest inflexion point to P_i as reference point P_{r1}. If there is no inflexion point, make P_{i-1} as P_{r1}.
   b) Same method as a) to determine P_{r1} on BS_i.
   c) Find the midpoint of P_{r1} and P_i (by curve length), mark it as P_{t1}, with parametert1.
   d) Same method as c) to determine P_{t2} and t2.
   e) Adjust t1 and t2; Make sure that P_{t1}, P_i, and P_{t2} are evenly distributed.
2) Build a Blending Cubic B-Spline to fit P_{t1}, P_i, and P_{t2}, with certain tangential and curvature constraints on P_{t1} and P_{t2}.
   a) Calculate the tangent $V_{t1}(V_{t2})$ and curvature vector (second derivatives) $C_{t1}(C_{t2})$on the location $P_{t1}(P_{t2})$ on $BS_{i-1}(BS_i)$. And make $V_{t1}(V_{t2})$ and $C_{t1}(C_{t2})$ unit vectors.
   c) Use OCC API ***AppDef_B-SplineCompute*** to fitted P_{t1}, P_i, and P_{t2} by a cubic B-Spline BS_{bi}, with $V_{t1}(V_{t2})$ and $C_{t1}(C_{t2})$ being assigned at $P_{t1}(P_{t2})$ as point constraints.
3) Verify the resulting BS_{bi} is in the tolerance band of original tool path. If not, go to step 1):e), reselect t1 and t2 and repeat the process to form a shorter BS_{bi}.
4) Stop until the tolerance requirement is satisfied.

4. Evaluation techniques

When machining complex features by HSM, many factors will influence its efficiency and accuracy: tool path geometrical characteristics and technical constraints of the machine, such as machine tool, numerical controlled unit (NCU) and cutting tool limits. In this paper, we only consider geometrical models of the tool path and evaluate the tool path quality by analyzing its feature and differential properties:

Features:

1) Shape corner point
2) C2 discontinuity point
3) Inflexion point

Differential Properties:

1) Curvature
2) Jerk
3) Torsion

The density of the data points of a tool path produced by any CAM software could be highly uneven and fluctuating due to built-in triangulation discretization process. We use finite difference method to calculate these quality indicators and illustrate them by plot and characteristic tags in a 3D graphic view. Before any calculation, a point reduction process is necessary for reducing successive points that are clustered too closely to a single point.

These quality indicators are computed as follows.

1) Shape corner points, which is equivalent to HBPs.
2) Curvature

   As shown in Fig. 7, $\bar{P}_i$, is the coordinate vector of data point P_i. We regard the line sequence as a curve C(s), *s* is its arc parameter. Here we adopt chord length as an approximant of *s*.

   For a data point P_i on C(s), its curve parameter can be compute as:

$$s_i = \sum_{i=1}^{n} \left\| \bar{P}_i - \bar{P}_{i-1} \right\|$$

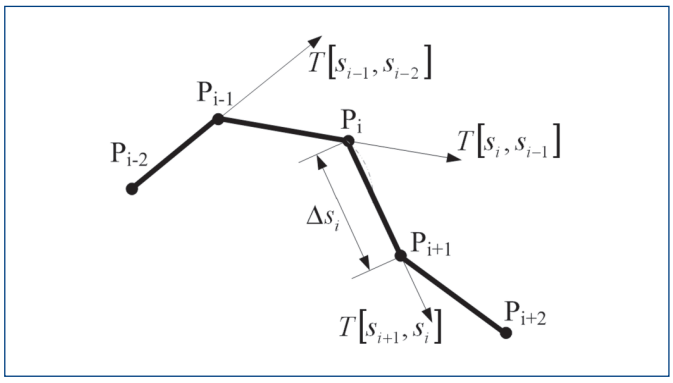

Figure 7. *Schematic view of a G01 based tool path*

   The first –order finite difference is computed as

$$T[s_i, s_{i-1}] = \frac{\bar{P}_i - \bar{P}_{i-1}}{s_i - s_{i-1}}$$

   The curvature at P_i can be computed as

$$k_i = \left| T[s_{i+1}, s_i, s_{i-1}] \right| = \left| \frac{T[s_{i+1}, s_i] - T[s_i, s_{i-1}]}{s_{i+1} - s_{i-1}} \right|$$

3) Jerk

   First we compute:

$$T[s_{i+2}, s_{i+1}, s_i, s_{i-1}] = \frac{T[s_{i+2}, s_{i+1}, s_i] - T[s_{i+1}, s_i, s_{i-1}]}{s_{i+2} - s_{i-1}}$$

$$T[s_{i+1}, s_i, s_{i-1}, s_{i-2}] = \frac{T[s_{i+1}, s_i, s_{i-1}] - T[s_i, s_{i-1}, s_{i-2}]}{s_{i+1} - s_{i-2}}$$

   Jerk at P_i can be computed as:

$$J_i = \frac{1}{2} \left| \left(T[s_{i+1}, s_i, s_{i-1}, s_{i-2}] + T[s_{i+2}, s_{i+1}, s_i, s_{i-1}] \right) \right|$$

4) C2 discontinuity point

   A discontinuity point is the location on the tool path where the curvature changes abruptly. *The rate of change of the* curvature with respect to *arc length is* used to help analyzing C2 discontinuity.

$$F(s_i) = \frac{dk}{ds} = \frac{1}{2} \left(\frac{k_{i+1} - k_i}{\Delta s_i} + \frac{k_i - k_{i-1}}{\Delta s_{i-1}} \right)$$

To calculate F (s_i) at P_i, we need five points around P_i to avoid local instability. For long-length line segment, it is necessary to add new points to adjust the data point distribution. The data points with its F (s_i) greater than the mean value of all F (s_i) are regarded as C2 discontinuity candidates.

For a C2 discontinuity candidate point P_i, if F (s_i) is a locally large value in its neighboring domain, P_i will be regarded as a C2 discontinuity point. Here we take 8 data points around P_i to double check.

$$F(s_i)_m = \sum_{j=i-4}^{i+4} F(s_i)$$

If ($F(S_i)$>K•F($Si)_m$), then P_i is identified as a C2 discontinuity point, where *K* is a proportional factor, usually k=3~5.

5) Inflexion

For analytic curves, an inflexion point is the point at which the curvature (or the second derivative) changes sign, or the binormal vector changes direction, where the binormal vector is defined to be the cross product of the unit tangent and unit normal vector. For tool path denoted by G01 blocks, finite difference is an effective method to calculate tangent and normal vectors.

Tangent vector: the first-order difference.

Normal vector: the second-order difference.

Due to the uneven and fluctuating distribution of G01 data points, the direction of the tool path may sway randomly. The calculation based on finite difference may detect some "fake inflexions", which is called "mathematical inflexions" here. These mathematical inflexions must undergo a double check process to identify the real one and get rid of the fake inflexions. We call the real one "Engineering inflexion". After all mathematical inflexions are figured out, we use some certain rule to determine a distance threshold T_d. T_d is used to define a neighboring domain around each mathematical inflexion point. In the neighboring domain we calculate the mean curvature of every data points on both sides of the mathematical inflexion point, denoted by C_{m1} and C_{m2}. The mathematical inflexion can be identified as an "engineering inflexion point" if C_{m1} and C_{m2} satisfy

$$C_{m1}C_{m2} < 0 \quad \text{and} \quad |C_{m1}| > T_c \ , |C_{m2}| > T_c$$

Where T_c is a curvature threshold. Both T_d and T_c can be chosen based on the dimension of the bounding box of the final part and machining tolerance.

6) Torsion

In Mathematics, Torsion is the rate of change of a space curve's osculating plane (denoted by the binormal vector)

$$\tau = -\frac{d\vec{B} \cdot \vec{N}}{ds}$$

For G01 blocks, we calculate the torsion of data point P_i based on 5 points around P_i. Let $\vec{B}_i$, denote the binormal vector at P_i , torsion τ_i is computed as

$$\tau_i = \left| \frac{1}{2} \cdot \left(\frac{\vec{B}_i - \vec{B}_{i-1}}{s_i - s_{i-1}} + \frac{\vec{B}_{i+1} - \vec{B}_i}{s_{i+1} - s_i} \right) \right|$$

5. Application examples

A software framework is built for implementing above mentioned algorithms, which includes a tool path quality evaluation module and a Cubic B-Spline fitting tool. Fig. 8 shows all HBPs of a tool path constructed by G01 blocks. Fig. 9 shows its color plot of curvature produced by the quality evaluation module, where it shows unsmooth changes of curvature and inflexions. Fig. 10 shows the quality evaluation results of the same tool path after B-Spline fitting. All G01 blocks are preprocessed and all feature points are indentified as HBPs. Then the tool path between HBPs are smoothed by Cubic B-Spline fitting. The color plot of curvature of the resulting tool path shows a great quality improvement (Fig. 10).

6 Summary

Many existing B-Spline fitting algorithms with shape preserving properties suffer from noisy G01 data points. This paper provides a systematical solution for quality improvement of such tool paths which are common in G-Code commands or CNC-interpolated axis motion commands. The solution includes an algorithm framework based on preprocessing and least-square B-Spline fitting with energy minimization, and a tool set to evaluate the quality of tool paths in any forms. All above solutions are integrated into a software package. The distribution adjustment of data points is

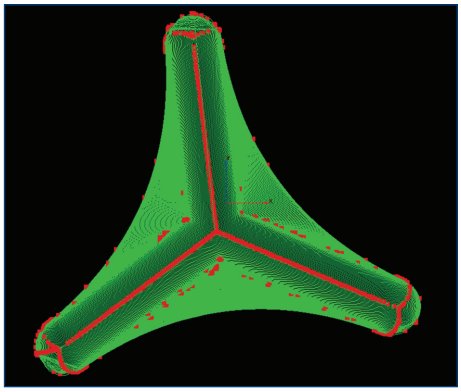

Figure 8. *HBPs (Feature Points) of a tool path*

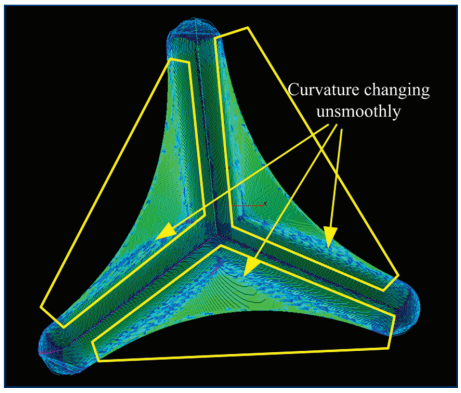

Figure 9. *Curvature Plot of a tool path without B-Spline Fitting*

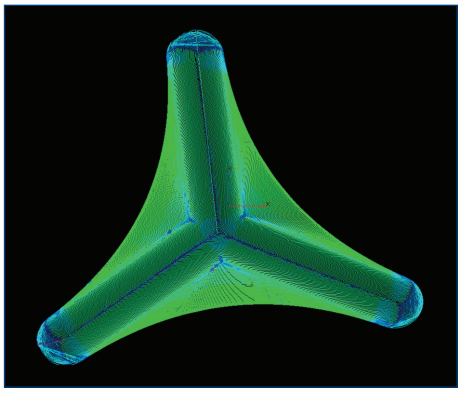

Figure 10. *Tool path improved by B-Spline Fitting algorithm proposed in this work*

fundamental for smoothing and shape preserving B-Spline fitting, so a point smoothing algorithm is introduced as a preprocessing method, which is modeled on mass distribution property preserving as well as stretching length minimization. An effective preprocessing consists of HBP identification, a point smoothing algorithm, a pre-fitting process by shape preserving and C1 B-Spline approximation. The cubic B-Spline fitting is then based on reselected sample data points which have an optimized distribution and always leading to better smoothed and shape preserving B-Splines. Then a tool set to evaluate the quality of tool path is presented, by introducing the mathematical solution for calculating feature properties (corner point recognition, inflexion and C2 dis-continuity points) and differential properties(curvature, jerk and torsion) of tool paths. The tool set is used to evaluate a tool path before and after tool path fitting proposed in this paper. Results show great improvement in qualities indicators, like curvature transition and jerk-free properties.

It should be noticed that the surface and form accuracies are not precisely guaranteed. The main reason comes from two aspects: 1) The exact surface of final part is approximated by discrete linear G01 commands, which causes geometric information losing; 2) G01-based G-codes files themselves carry no tolerance information. An effective method to meet the machining accuracy requirement is to firstly compute a guess value of tolerance (intol & outtol) based on analysis and statistics of G01 commands, then use a conservative tolerance value(for example, a half of the guess value) in the proposed B-Spline fitting algorithm.

Future works include further development of the software package, and experimental verification of the effectiveness of the systematical solution presented in this paper by NC machining practices conducted by HNC-8 controllers.

Acknowledgments

The authors gratefully acknowledge the support of the National Natural Science Foundation of China (50975111 and 50835004) and National Science and Technology Major Project of the Ministry of Science and Technology of China (2010ZX04001-201). The authors would also like to thank the support from colleagues Ming Zhang, Kun Yin and Daojiang Ou.

References

[Bajaj 1998] Bajaj, C. L., Chen, J., et al. Energy formulations of A-splines. Computer Aided Geometric Design,1998, 15(5): 1-21.

[Barsky 1989] Barsky, B. A. and DeRose, T. D. Geometric continuity of parametric curves: three equivalent characterizations. Computer Graphics and Applications, IEEE,1989, 9(6): 60-69.

[Celniker 1991] Celniker, G. and Gossard, D. Deformable curve and surface finite-elements for free-form shape design. SIGGRAPH Comput. Graph.,1991, 25(4): 257-266.

[Costantini 2001] Costantini, P. and Pelosi, F. Shape-preserving approximation by space curves. Numerical Algorithms, 2001, 27(3): 237-264.

[Costantini 2004] Costantini, P. and Pelosi, F. Shape-preserving approximation of spatial data.
Advances in Computational Mathematics, 2004, 20(1): 25-51.

[Dierckx 1995] Dierckx, P. Curve and surface fitting with splines, Oxford University Press, USA,1995

[Eck 1995] Eck, M. and Hadenfeld, J. Local energy fairing of B-spline curves.1995.

[Farin 2002] G. Farin, e. HANDBOOK OF COMPUTER AIDED GEOMETRIC DESIGN, 2002

[Kvasov 2000] Kvasov, B. I. Methods of shape-preserving spline approximation, World Scientific Pub Co Inc, 2000

[Künzli 2010] Stefan Künzli, T. L. Generating smooth C2 curves in R2 and R3 for irregular point data, SA, SMR.

[Liu 2008] Liu, Y. and Wang, W. A Revisit to Least Squares Orthogonal Distance Fitting of Parametric Curves and Surfaces Advances in Geometric Modeling and Processing. F. Chen and B. Jüttler, Springer Berlin/Heidelberg,2008, 4975: 384-397.

[Mei 2010] Mei, Z. Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. China Science, 2010.

[Moreton 1994] Moreton, H. P. and Séquin, C. H. Minimum variation curves and surfaces for computer-aided geometric design. Designing Fair Curves and Surfaces,1994: 123-159.

[Park 2000] Park, H., Kim, K., et al. A method for approximate NURBS curve compatibility based on multiple curve refitting. Computer-Aided Design,2000, 32(4): 237-252.

[Park 2007] Park, H. and Lee, J. H. B-spline curve fitting based on adaptive curve refinement using dominant points.
Computer-Aided Design,2007, 39(6): 439-451.

[Poliakoff 1999] Poliakoff, J. F., Yew Kee, W., et al. An automated curve fairing algorithm for cubic B-spline curves. Journal of Computational and Applied Mathematics,1999, 102(1): 73-85.

[Spellucci 1998] Spellucci, P. A new technique for inconsistent QP problems in the SQP method. Mathematical Methods of Operations Research, 1998, 47(3): 355-400.

[Spellucci 1998] Spellucci, P. An SQP method for general nonlinear programs using only equality constrained subproblems. Mathematical programming,1998, 82(3): 413-448.

[Tapie 2007] Tapie, L., Mawussi, K. B., et al. Machining strategy choice: performance viewer. Advances in Integrated Design and Manufacturing in Mechanical Engineering II,2007: 343-356.

[Vassilev 1996] Vassilev, T. I. Fair interpolation and approximation of B-splines by energy minimization and points insertion. Computer-Aided Design,1996, 28(9): 753-760.

[Veltkamp 1995] Veltkamp, R. C. and Wesselink, W. Modeling 3D Curves of Minimal Energy. Computer Graphics Forum, 1995, 14(3): 97-110.

[Wang 2004] Wang, W., Pottmann, H., et al. Fitting B-Spline curves to point clouds by squared distance minimization. 2004.

[Wang 2006] Wang, W., Pottmann, H., et al. Fitting B-spline curves to point clouds by curvature-based squared distance minimization. ACM Trans. Graph., 2006, 25(2): 214-238.

[Xu 2011] Xu, G., Wang, G., et al. Geometric construction of energy-minimizing Béezier curves. SCIENCE CHINA Information Sciences, 2011, 54(7): 1395-1406.

[Yau 2007] Yau, H. T. and Wang, J. B. Fast Bezier interpolator with real-time lookahead function for high-accuracy machining. International Journal of Machine Tools and Manufacture, 2007, 47(10): 1518-1529.

[Zhang 2001] Zhang, C., Zhang, P., et al. Fairing spline curves and surfaces by minimizing energy. Computer-Aided Design,2001, 33(13): 913-923.

Contacts

Wuhan Huazhong Numerical Control Co.,Ltd, Wuhan, P. R. China
University of Science and Techological Park Road
Wuhan, Hubei, China, 430223, tel.: 86-27-87 180 031
National NC System Engineering Research Center,
Huazhong University of Science and Technology, Wuhan, P. R. China
tel.: +86...15 071 310 900
e-mail: chenhanlee@hotmail.com