



# A chord error conforming tool path B-spline fitting method for NC machining based on energy minimization and LSPIA

Shanshan He, Daojiang Ou, Changya Yan\*, Chen-Han Lee

National NC System Engineering Research Center, Huazhong University of Science and Technology, 430074 Wuhan, PR China

Received 20 March 2015; received in revised form 20 May 2015; accepted 3 June 2015

Available online 12 June 2015

## Abstract

Piecewise linear (G01-based) tool paths generated by CAM systems lack  $G^1$  and  $G^2$  continuity. The discontinuity causes vibration and unnecessary hesitation during machining. To ensure efficient high-speed machining, a method to improve the continuity of the tool paths is required, such as B-spline fitting that approximates G01 paths with B-spline curves. Conventional B-spline fitting approaches cannot be directly used for tool path B-spline fitting, because they have shortages such as numerical instability, lack of chord error constraint, and lack of assurance of a usable result. Progressive and Iterative Approximation for Least Squares (LSPIA) is an efficient method for data fitting that solves the numerical instability problem. However, it does not consider chord errors and needs more work to ensure ironclad results for commercial applications. In this paper, we use LSPIA method incorporating Energy term (ELSPIA) to avoid the numerical instability, and lower chord errors by using stretching energy term. We implement several algorithm improvements, including (1) an improved technique for initial control point determination over Dominant Point Method, (2) an algorithm that updates foot point parameters as needed, (3) analysis of the degrees of freedom of control points to insert new control points only when needed, (4) chord error refinement using a similar ELSPIA method with the above enhancements. The proposed approach can generate a shape-preserving B-spline curve. Experiments with data analysis and machining tests are presented for verification of quality and efficiency. Comparisons with other known solutions are included to evaluate the worthiness of the proposed solution.

© 2015 Society of CAD/CAM Engineers. Production and hosting by Elsevier. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

**Keywords:** CNC machining; G-code; B-Spline fitting; Progressive iterative approximation; Energy minimization

## 1. Introduction

G-code commands generated by CAM systems play an important role in CNC machining. Among them piecewise linear (G01-based) tool paths are widely used. The lack of  $G^1$  and  $G^2$  continuity of G01-based paths cause unwanted vibrations and slow-downs during machining. To ensure efficient high-speed machining, a method such as B-spline fitting to improve the continuity of tool path is required.

Tool path B-spline curve fitting is often required to fulfill more constraints than conventional B-spline curve fitting in shape and performance [1], such as approximation of the G01 points within a given tolerance, chord error constraint, shape

preservation,  $G^2$  continuity, and minimal number of control points. In addition, computation must be fast and accurate even for large size tool paths to use the solution in actual NC machining.

The conventional B-spline fitting approaches are often solved by minimizing Least Square Fitting (LSF) error or energy functions within a given tolerance [2–8]. LSF involves solving a very large system of linear equations. It is not suitable for industrial solutions because of numerical instability and lack of assurance of a usable result.

Instead of solving linear equation, Progressive and Iterative Approximation (PIA) proposed by Qi et al. [9] and de Boor [10] is a new and effective method for data fitting that eliminates the numerical instability of solving inverse matrices. PIA constructs a series of fitting curves by adjusting control points iteratively [11–15]. Deng et al. [4] reported a method of

\*Corresponding author.

E-mail address: [yanchangya@hotmail.com](mailto:yanchangya@hotmail.com) (C. Yan).

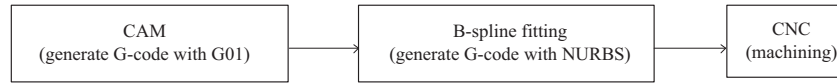


Fig. 1. Flowchart of the relationship between CAM, B-spline fitting, and CNC machining.

Progressive and Iterative Approximation for Least Squares (LSPIA) with the ability to handle point set of large size with less control points than PIA. Nevertheless the LSPIA method cannot be directly used in industrial applications of NC machining due to the following reasons: first, LSPIA does not consider chord error requirement, second, the data points of empirical examples distributed evenly, but actual NC tool paths have more complex and unpredictable shapes and non-uniform data point distribution.

Parameter values of data points and knot vector are critical to the quality of fitting curve. Piegl and Tiller [5, 16] suggested using averaging technique (AVG) and knot placement technique (KTP). Razdan [17] and Li et al. [18] suggested using shape information to determine knot vectors. Park et al. [3] devised a new B-spline curve fitting method based on adaptive curve refinement using dominant points. Their knots were determined by averaging the parameter values of the dominant points.

Tool path B-spline fitting method is often used to improve the machining efficiency within the precision to achieve high-speed and high-precision CNC machining. Yang and Chen [19] proposed a new high precision fitting approach which used roughly fit and fine fit for NURBS generation, and Newton–Raphson method is used to solve the optimization problem, but the authors did not give a method to solve chord error refinement, and Newton–Raphson method cannot ensure a solution that satisfies the accuracy requirement. Syh-ShiuhYeh and Hsin-Chuan Su [20] developed a method for implementing an online non-uniform rational B-spline (NURBS) curve fitting process on CNC machines for improving the quality and efficiency of machining. He fitted the data points using optimal search method and used least square fitting method to solve the optimization problem without considering chord error and numerical instability. If the fitting process fails, he just used stored data points as the motion commands for ensuring the continuous motion of CNC machines. Zhang et al. [21] proposed a method of curve fitting for velocity planning on CNC machines based on quadratic B-splines. The fitting curve was obtained by interpolating feature points. Chasing method was used to compute control points. This method also has shortage of the numerical instability of solving inverse matrices.

The goal of this study is to design and implement an algorithm that is capable of satisfying all of the requirements. Conventional LSF method is not considered for industrial-strength applications because of numerical instability. LSPIA method cannot be directly used for tool path B-spline fitting because it does not consider chord error requirement. Moreover, both the conventional LSF method and LSPIA lack actual machining experiments to validate the usability.

A number of commercial CAD/CAM software systems can generate NURBS tool paths, such as NX, CATIA, Delcam,

PowerMILL, SINUMERIK 840D compressor etc. The NX solution delivers favorable results but from data analysis we found that NX solution has some limitations. It cannot ensure  $G^1$  continuity between two B-splines and determine the feature points of the tool paths automatically – the feature points are determined by a user-provided angle. Our previous work [1] can identify the feature points, which are called “Hard Break Points (HBP)”, automatically. The goal is to ensure  $G^1$  continuity between two B-splines, and G01 and B-spline, except at HBP locations. HBP identify algorithm [1] is not the key discussion point in this paper but is the prerequisite of the tool path B-spline fitting.

Data points in this paper are generated by CAM software, and our fitting results are used for 3-axis actual machining after B-spline fitting. Fig. 1 is a systematic flowchart of the relationship between CAM, B-spline fitting, and CNC machining. Tool path B-spline fitting is an optimize step for tool paths generated by CAM software, and after tool path B-spline fitting, the tool paths have better continuity and smoother than the original G01. It can generate better machining surface and save internal storage.

In this study, we propose to use LSPIA method incorporating an energy term (ELSPIA) to improve the performance and lower the chord errors. We select initial control points which can demonstrate the feature of data points and are uniformly distributed. We design an iterative algorithm such that the foot point parameters are updated strategically; then we analyze the required degrees of freedom of control points to insert new control points effectively; furthermore we apply chord error refinement with ELSPIA method if the chord error requirement is not satisfied.

This paper is organized as follows. In Section 2, we state the requirements and high-level algorithm of tool path B-spline fitting. In Section 3, we introduce the algorithm of LSPIA. In Section 4, we explain our improvement of LSPIA in four aspects. Section 5 is the implementation of data fitting and chord error refinement with ELSPIA. Section 6 presents some numerical validation and Section 7 presents machining experiments. The last Section 8 concludes this paper. Moreover, Appendix supplements some detailed algorithm of this paper.

## 2. Overview of tool path B-spline fitting algorithm

To obtain smooth, shape preserving, and tolerance banded cubic B-Spline tool paths, first the original data points need to be processed to remove noise points, determine HBPs, and identify long segments, then data points of tool path must be grouped by “breaks”, which should be identified by data analysis [1]. Thus the generated new point sequences are more suitable for further fitting.

The conventional B-spline fitting algorithm mainly fitted the data points with data error constraint. But in tool path B-spline fitting, not only data error constraint but also chord error constraint should be specified to minimize the division between fitting curve and polygon formed by the data points. Data errors can be optimized by minimizing the least square fitting error, but chord errors cannot be directly optimized due to the complex computation.

Firstly, let us review the concept of data point fitting. The readers are assumed to be familiar with the concepts of B-spline and optimization theory. A B-spline curve with degree  $p$ , knot vector  $U = \{u_0, u_1, \dots, u_{n+p+1}\}$  is defined as:

$$c(t) = \sum_{i=0}^n B_i^p(t) P_i \quad (1)$$

$\{B_0^p(t), B_1^p(t), \dots, B_n^p(t)\}$  are the normalized B-spline basic functions of degree  $p$  defined on the knot vector  $U$ .  $\{P_0, P_1, \dots, P_n\}$  are the control points.

In this paper, assume that  $\{Q_j\}_{j=0}^m$  are ordered G01 data points,  $\{t_j\}_{j=0}^m$  are the parameter values of  $\{Q_j\}_{j=0}^m$ . Data error constraint is taken into consideration by minimizing the least square fitting error defined as:

$$E(P) = \sum_{j=0}^m (c(t_j) - Q_j)^2 \quad (2)$$

Wherein,  $P = (P_0, P_1, \dots, P_n)^T$ . The matrix form of Eq. (2) is:

$$E(P) = (AP - Q)^T (AP - Q) \quad (3)$$

Where

$$Q = (Q_0, Q_1, \dots, Q_m)^T,$$

$$A = \begin{bmatrix} B_0^p(t_0) & B_1^p(t_0) & \dots & B_n^p(t_0) \\ B_0^p(t_1) & B_1^p(t_1) & \dots & B_n^p(t_1) \\ \vdots & \vdots & \ddots & \vdots \\ B_0^p(t_m) & B_1^p(t_m) & \dots & B_n^p(t_m) \end{bmatrix}$$

In this study, the initial control points and initial knot vector are determined before the iterative steps of data point fitting. The data point fitting algorithm can be divided into three level cycles: major cycle, medium cycle and minor cycle. After these three cycles, a refinement process is used to reach the chord error requirement. There are many methods to solve the optimization of Eq. (3), but we use LSPIA method in the minor cycle considering performance and industrial application. We will introduce the LSPIA method in Section 3. If there is any data error that cannot satisfy fitting tolerance requirement after LSPIA iteration, we will start foot point parameter updating method in the medium cycle and control point insertion method in the major cycle. At last, chord error refinement method with ELSPIA is used if the chord errors cannot satisfy machining tolerance requirement. The flowchart of tool path B-spline fitting algorithm is shown in Fig. 2.

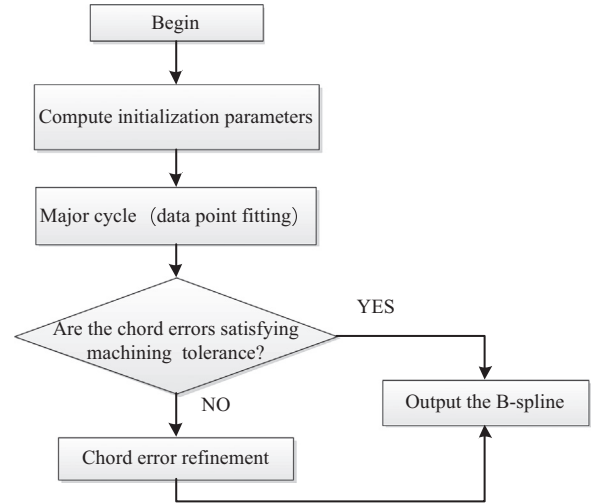


Fig. 2. Flowchart of tool path B-spline fitting.

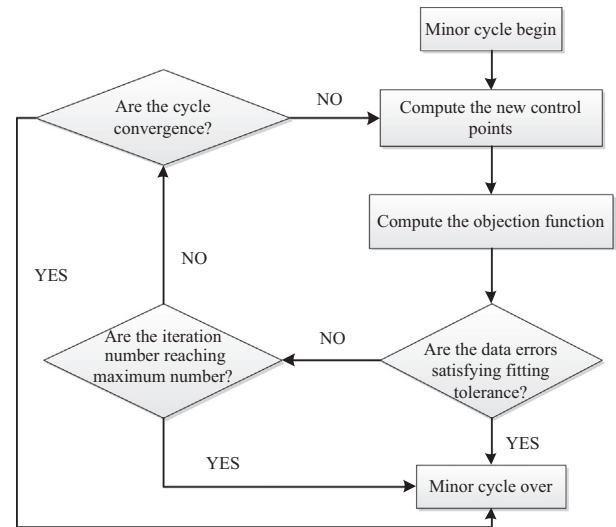


Fig. 3. Flowchart of minor cycle.

### 3. The LSPIA iterative method of data point fitting

In this section, we will introduce the LSPIA iterative method and analyze why it cannot be directly used for tool path B-spline fitting.

Assume that  $\{P_i^0\}_{i=0}^n$  are the initial control points selected from the given data points  $\{Q_j\}_{j=0}^m$ ,  $U = \{u_i\}_{i=0}^{n+p+1}$  are the knot vector. The first B-spline curve is:

$$c^0(t) = \sum_{i=0}^n B_i^p(t) P_i^0 \quad (4)$$

Let  $P^k = (P_0^k, P_1^k, \dots, P_n^k)^T$  be the control point vector in the  $k$ th iteration, and  $\Delta^k = (\Delta_0^k, \Delta_1^k, \dots, \Delta_n^k)^T$  be the adjusting vector of control points in the  $k$ th iteration. LSPIA uses steepest descent method [22] to minimize the fitting error defined by Eq. (3). First the gradient vector of  $E(P)$  in Eq. (3) is computed as:

$$\nabla E(P) = 2A^T(AP - Q) \quad (5)$$

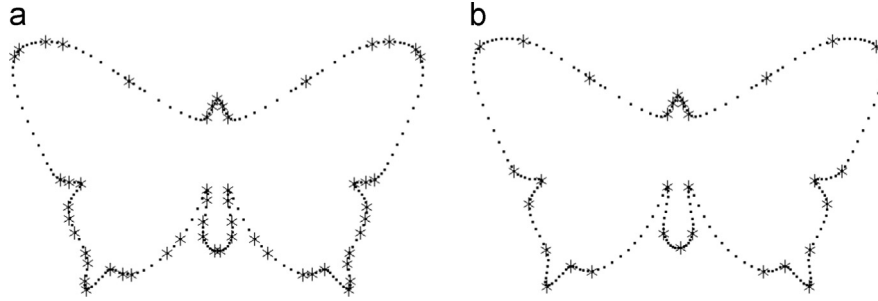


Fig. 4. LCM points and LDCM points: (a) LCM points, (b) LDCM points.

Then we can develop the  $k$ th iterative steps of LSPIA:

$$\begin{cases} \Delta^k = \mu A^T(Q - AP^k) \\ P^{k+1} = P^k + \Delta^k \end{cases} \quad (6)$$

where  $\mu$  is the moving step,  $0 < \mu < \frac{2}{\lambda_0}$ ,  $\lambda_0$  is the largest eigenvalue of matrix  $A^T A$ . The matrix  $A^T A$  is a positive definite and symmetric matrix, The  $(k+1)$ th curve  $c^{k+1}(t)$  is:

$$c^{k+1}(t) = \sum_{i=0}^n B_i^p(t) P_i^{k+1} \quad (7)$$

**Remark 1.1.** For the case of interpolating end points  $Q_0, Q_m$ , we just let  $P_0^0 = Q_0, P_n^0 = Q_m$  and  $P_0^{k+1} = P_0^k, P_n^{k+1} = P_n^k$  in the iteration.

The flowchart of minor cycle is shown in Fig. 3. Compared with the conventional least square fitting method, LSPIA is an efficient and intuitive method for data fitting that eliminates the numerical instability of solving inverse matrices. However, in most cases the data error requirement cannot be satisfied just by the minor cycle. Data errors decrease obviously at first iteration steps but then they slow down after several steps, one of the reasons is that the foot points computed by data point parameters are not precise, and the other one is that the control points lack degree of freedom, meaning that there are not enough controls points. So we need to improve LSPIA method to make the fitting curve to satisfy data error and chord error requirement.

#### 4. The improvement of LSPIA for tool paths B-spline fitting

In this section, we improve LSPIA method in four aspects: (1) initial control points selection by LDCM method and knot vector determination by initial control points; (2) foot point parameters updating method; (3) control points insertion method using the degree of freedom of control points; (4) chord error refinement method using ELSPIA.

##### 4.1. Determination of initial control points and knot vector

In this section, we propose an improved technique for initial control point determination over the Dominant Point Method (DOM) [3] to ensure uniform distribution of control points and convergence of feature locations. Park et al. [3] determined the

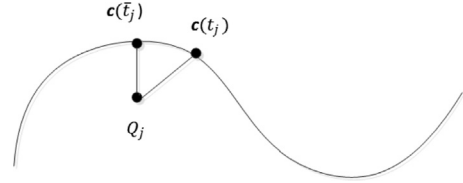


Fig. 5. The approximate foot point and real foot point of  $Q_j$ .

dominant points by Local Curvature Maximum (LCM) points, but LCM points contain noise points. As shown in Fig. 4(a), there are 55 LCM points selected by 245 data points. Although the noise points can be excluded in a way, but the dominant points are gathered at the complex region and none at the flat region. To exclude the noise points, we adopt a Local Difference Curvature Maximum (LDCM) method to select feature points, noted as LDCM points. Assume that the curvature  $k_j$  of  $P_j$  has been computed by the method in Appendix A.  $s$  is a predefined neighborhood value. The difference curvature is defined as:

$$\sigma_j = k_j - \frac{\sum_{i \geq j-s, i \leq j+s, i \neq j} k_i}{2s}, \quad j = s, \dots, m-s, \quad (8)$$

If  $\sigma_j > 0$ ,  $\sigma_j > \sigma_{j-1}$ , and  $\sigma_j > \sigma_{j+1}$ , the point  $P_j$  is considered as a LDCM point. As shown in Fig. 4(b), let  $s=5$ , there are 33 LDCM points selected from 245 data points. To make the initial control points distribute uniformly, we need to add some control points in the region formed by two adjacent LDCM points that have more than  $2s$  data points. The initial control point set ensures that there are at most  $2s$  data points (a data point block) between two initial control points. The end points of each data point block are selected as initial control points to make the fitting curve interpolating the end points of data point blocks instead of all data points.

Assume that  $\{P_i^0\}_{i=0}^n$  are the initial control points selected from  $\{Q_j\}_{j=0}^m$ ,  $\{t_j\}_{j=0}^m$  are the parameter values of  $\{Q_j\}_{j=0}^m$ . Mostly the parameter values are computed by chord length or centripetal methods [2,5]. We use the AVG method of initial control points to compute the knot vectors:

$$u_{p+j} = \frac{1}{p} \sum_{i=j}^{j+p-1} t_{f(i)}, \quad j = 1, \dots, n-p$$

where  $f(i)$  is a function that returns the index of the point  $Q_j$  corresponding to initial control point  $P_i^0$ . In order to make the fitting curve interpolate the end points of data point blocks,

the end knots should always have multiplicity of  $(p + 1)$ . The other knots need to have no multiplicity to obtain curve with  $G^2$  continuity.

#### 4.2. Foot point parameters updating method

In the iterative steps of LSPIA, the data point parameter  $t_j$  is used as the foot point parameter of the data point  $Q_j$ . The data error of  $Q_j$  is measured by  $|c(t_j) - Q_j|$ . Actually,  $|c(t_j) - Q_j|$  is not the real deviation between data point  $Q_j$  and B-spline curve  $c(t)$ . As shown in Fig. 5,  $c(t_j)$  is a approximate foot point of  $Q_j$ , and  $c(\bar{t}_j)$  is the projection point of  $Q_j$  on the B-spline,  $\bar{t}_j$  is the parameter of the projection point. In Section 3, the adjust vector of control point  $P_i$  is defined as:  $\Delta_i = \mu \sum_{j=0}^m B_i^p(t_j)(Q_j - c(t_j))$ , foot point parameters will also affect the fitting result of LSPIA because imprecise foot point parameters result in imprecise adjust vectors. Thus we refresh the foot point parameters to get better fitting result.

The foot point parameters updating method is to compute the projection point parameters of data points and use them as new foot point parameters. The projection point parameter  $\bar{t}_j$  is computed by solving Eq. (9). The solution is obtained by using Newton iteration method.

$$(Q_j - c(t)) \cdot c'(t) = 0 \tag{9}$$

#### 4.3. Control points insertion to fulfill the data error requirement

##### 4.3.1. Finding the control points lacking degrees of freedom

In the iterative steps of LSPIA, for every index  $i$ , the control point  $P_i$  is adjusted by the adjust vector  $\Delta_i = \mu \sum_{j=0}^m B_i^p(t_j)(Q_j - c(t_j))$ , for every index  $j$ , if  $B_i^p(t_j) > 0$ ,  $Q_j$  is considered as a related data point of  $P_i$ . For control point  $P_i$ , if its related data point offset vectors have consistent directions, then we can greatly decreased the related data errors by adjusting  $P_i$  with  $\Delta_i$ . In this case  $P_i$  is considered as having enough degree of freedom. Otherwise, we think  $P_i$  as lacking degree of freedom. Assuming that the related data points of control points  $P_i$  are from  $Q_s$  to  $Q_e$ , we measure the degree of freedom of  $P_i$  by  $\varphi_i$  which is defined in the following expression:

$$\varphi_i = \frac{\sum_{j=s}^e |B_i^p(t_j)(Q_j - c(t_j))|}{|\Delta_i|}$$

For control points  $P_i$ , the greater the  $\varphi_i$ , the more lacking of degree of freedom. To fulfill the data error requirement, we use data errors and degree of freedom of control points to find a knot interval to insert knot.

##### 4.3.2. Finding the insert interval and insert knot

When a knot is inserted into the interval  $[u_i, u_{i+1}]$ , the adjacent control points  $P_{i-2}$  and  $P_{i-1}$  become new control

points  $P_{i-2}$ ,  $P_{i-1}$  and  $P_i$ , the other control points remain unchanged. Therefore, we define the degree of freedom of the knot interval  $[u_i, u_{i+1}]$  by the related data errors and the degree of freedom of  $P_{i-2}$  and  $P_{i-1}$ .

For each interval  $[u_i, u_{i+1}]$ , if there has a data error that exceed fitting tolerance, we compute the degree of freedom of the knot interval  $[u_i, u_{i+1}]$ :  $\gamma_i = \varphi_{i-2} + \varphi_{i-1}$ , else  $\gamma_i = 0$ . The knot interval with maximum  $\gamma_i$  is selected as the insert interval.

#### 4.4. ELSPIA method to fulfill the chord error requirement

The optimization problem of finding the minimization of Eq. (3) only considers data error constraint, dose not consider chord error. In this section, the stretching energy term is used to decrease the chord error. The objective function Eq. (2) can be modified as:

$$E(P) = \sum_{j=0}^m (c(t_j) - Q_j)^2 + \omega \int_0^1 \left[ \frac{dc(t)}{dt} \right]^2 dt \tag{10}$$

The first term in Eq. (10) is the LSF error to constraint data error, and the second term is called stretching energy term to lower the chord error.  $\omega$  is the weight of energy term. The matrix form of Eq. (10) is:

$$E(P) = (AP - Q)^T (AP - Q) + \omega P^T D P \tag{11}$$

where  $D = D_{ij} = \int_0^1 B_i'(t) B_j'(t) dt$ . The matrix  $D$  can be computed by the numerical integration method presented in [21]. Similar to LSPIA, the steepest descent method can be used to minimize the objective function.

ELSPIA method can decrease the chord error, but the computation load is greater than LSPIA. So in actual algorithm implementation, we only use ELSPIA in the chord error refinement step. The minor cycle procedure still uses LSPIA method.

In the chord error refinement section, only several control points should be adjusted. Thus, a weight  $\rho_i$  is given to each control point  $P_i$  (if  $P_i$  needs to be adjusted,  $\rho_i = 1$ , otherwise  $\rho_i = 0$ ). We use the following method to solve the optimize problem. First the gradient vector of  $E(P)$  in Eq. (11) is computed as:

$$\nabla E(P) = 2(A^T (AP - Q) + \omega DP) \tag{12}$$

Let  $P^k = (P_0^k, P_1^k, \dots, P_n^k)^T$  be the control point vector in the  $k$ th iteration, and  $\Delta^k = (\Delta_0^k, \Delta_1^k, \dots, \Delta_n^k)^T$  be the adjusting vector of control points in the  $k$ th iteration.  $H$  is a diagonal matrix formed by the weights of control points.

$$H = \begin{bmatrix} \rho_0 & & \\ & \ddots & \\ & & \rho_n \end{bmatrix}$$

Then we can improve the iterative steps of ELSPIA by using the matrix description:

$$\begin{cases} \Delta^k = \mu(A^T(Q - AP^k) - \omega DP^k) \\ P^{k+1} = P^k + H\Delta^k \end{cases} \tag{13}$$

where  $0 < \mu < \frac{2}{\lambda_0}$ ,  $\lambda_0$  is the largest eigenvalue of  $H(A^T A + \omega D)$ . The  $(k+1)$ th curve  $c^{k+1}t$  is:

$$c^{k+1}(t) = \sum_{i=0}^n B_i^p(t) P_i^{k+1} \quad (14)$$

**Theorem 1.2.** *The ELSPIA iterative method is convergent after several iterative steps.*

For the proof of Theorem 1.2, please refer to the Appendix B.

### 5. The implementation of tool path data point fitting and chord error refinement

#### 5.1. The endpoints constraints

In this paper, the fitted B-spline curve needs to satisfy two kinds of endpoint constraints. First, the B-spline curve must pass the endpoint of data points, and then the tool paths need to be  $G^1$  continuity at the endpoint between two adjacent B-splines or a B-spline and a segment. The endpoint tangent vectors are given before fitting.

In Section 4.1, we know that the end knots are always with multiplicity  $(p+1)$  in order to make the fitted curve interpolating the end points of data point blocks, and Remark 1.1 ensures the endpoint of control points do not move in the iterative steps of LSPIA.

To ensure  $G^1$  continuity at the endpoint of B-spline, given two unit tangent vectors  $V_1$  and  $V_2$ ,  $l_1$  and  $l_2$  are the length of endpoint tangent vectors of the B-spline. The fitted B-spline must satisfy the equations:

$$c'(0) = l_1 V_1, \quad c'(1) = l_2 V_2$$

The endpoint tangent vectors of the B-spline with multiplicity  $(p+1)$  end knots are:

$$c'(0) = p \frac{P_1 - P_0}{u_{p+1} - u_1}, \quad c'(1) = p \frac{P_n - P_{n-1}}{u_{p+n} - u_n}$$

The length of the end tangent vectors is important to the quality of the B-spline at end portion. It may cause loop at the end of B-spline if the tangent length is too long, and the end two control points will be too close if the tangent length is too short. So we determine the length of end tangent vectors by the following approximate calculation:

$$l_1 = \frac{Q_1 - Q_0}{t_1 - t_0}, \quad l_2 = \frac{Q_m - Q_{m-1}}{t_m - t_{m-1}}$$

To ensure  $G^1$  continuity, control points  $P_0, P_1, P_{n-1}$ , and  $P_n$  are determined beforehand and will not move in the iterative steps of data points fitting.

#### 5.2. Foot point parameters updating in the medium cycle

The foot point parameters are firstly computed by the chord length parameters of data points, and then are updated by the parameters of projection points. In this study, first the parameters are updated right after the first B-spline is obtained, and then the

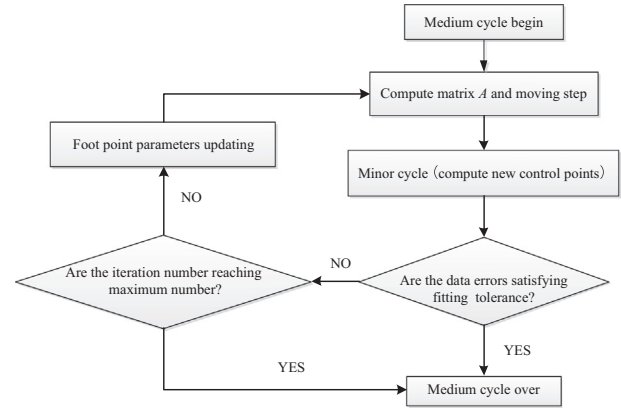


Fig. 6. Flowchart of medium cycle.

parameters are updated every ten iterations of the minor cycle. The flowchart of the medium cycle is shown in Fig. 6.

Fig. 7(a) and (b) plot the fitted curves of a data set before and after foot point parameters updating respectively. The machining tolerance is 0.03 mm and the fitting tolerance is 0.015 mm. During fitting, the minor cycle iterates 20 times without control points insertion. It can be seen that the fitted curve using foot point parameters updating method has better properties in shape-preserving and chord error.

#### 5.3. Control points insertion in the major cycle

In Section 4.3, we introduce the control point insertion method in the major cycle. The flowchart of the major cycle is shown in Fig. 8. After we select an interval that lacks degree of freedom and has at least two data point parameters inside, a knot can be inserted in this interval by selecting the middle point of the interval or the data point parameter with maximum data error in the interval. In our solution the middle point strategy is adopted. After control point insertion, both the data errors and chord errors can be decreased.

#### 5.4. Determination of the control points that can be adjusted in the chord error refinement step

In tool path B-spline fitting, the most important and difficult problem is to satisfy the chord error requirement. To lower chord errors, the fitting tolerance in the data fitting steps can be set smaller than the machining tolerance. We check the chord errors by using the method introduced in Appendix. Chord error refinement aims to decrease the chord error that exceeds the allowed machining tolerance after fitting. The flowchart of chord error refinement is shown in Fig. 9. There is no need to adjust all control points in chord error refinement. So we need a strategy to determine which control points need to be selected to apply adjusting.

Chord error refinement is first to find a knot interval that contains at least one point where the chord error exceeds the machining tolerance, secondly insert a knot in the interval, determine weights of control points, and then use the ELSPIA method to compute the control points.

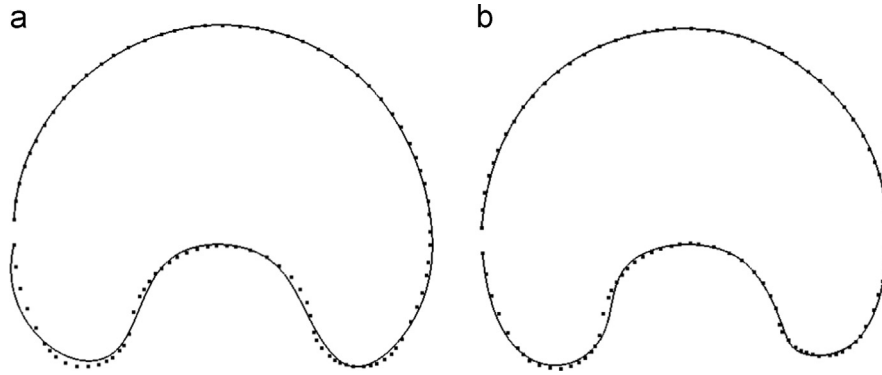


Fig. 7. The fitted curve before and after updating of foot points parameters: (a) before foot point parameters updating and (b) after foot point parameters updating.

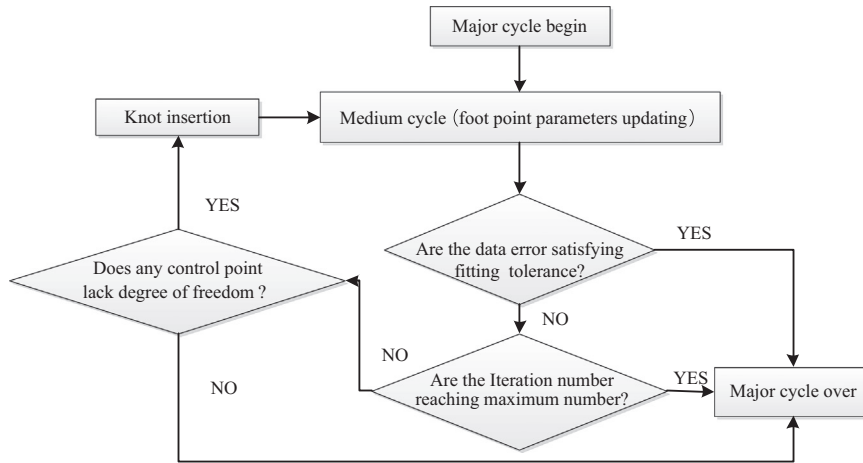


Fig. 8. Flowchart of major cycle.

Assuming that the interval  $[u_i, u_{i+1}]$  contains at least one point where the chord error exceeds the machining tolerance, the adjusting control points are  $P'_{i-2}$ ,  $P'_{i-1}$ , and  $P'_i$  (new control points after knot insertion), the weights of  $P'_{i-2}$ ,  $P'_{i-1}$ , and  $P'_i$  need to be set to 1. The control points with weights equaling to 1 will be adjusted in the following chord error refinement process.

### 5.5. Weight selection

There are two weights in this section to be determined. The first is moving step  $\mu$ . An estimated  $\mu$  can be determined by Gershgorin disc theorem [19]. For the details of the determination, please refer to the Appendix. The other is the weight of energy term. As we know, the adjusting vector of control points in the  $k$ th iteration is:

$$\Delta^k = \mu(A^T(Q - AP^k) - \omega DP^k) = \mu(A^T Q - A^T AP^k - \omega DP^k)$$

To ensure that  $A^T AP^k$  and  $DP^k$  have the same order of magnitude, the weight  $\omega$  can be estimated by the matrixes  $A^T A$  and  $D$ .

$$\omega \approx \frac{\text{trace}(A^T A)}{\text{trace}(D)}$$

Using the same data points of Fig. 7, Table 1 compares the fitting result before and after chord error refinement. In the fitting, the machining tolerance is 0.03 mm and fitting tolerance is

0.015 mm. It can be seen that the fitted curve can satisfy the data error and chord error requirement after chord error refinement.

In our actual testing of various shapes of G-code, at least 99% of the fitted points satisfied the chord error refinement if given a reasonable machining tolerance, but if there were still some segments that exceeded the allowed machining tolerance, we had to cut off the exceeding spline into G01 segments to ensure the continuous motion of the CNC machining. The cutting sections were ensured to have  $G^1$  continuity between G01 and B-spline, but  $G^0$  continuity between G01 and G01. It was not a perfect strategy and B-spline blending method will be used in the future work.

## 6. Numerical validation

### 6.1. Fitting result

A software framework is built to implement the tool path B-spline curve fitting algorithm. The tool paths in this study are generated by NX 9.0 with a given machining tolerance, and the data sets are cut off from the actual tool paths. Four data sets are fitted by cubic B-spline curves by using the proposed method. The data points and fitted curves are plotting in Fig. 10, and the fitting information is listing in Table 2. It can be seen that the number of control points is less than 50% of data point quantity, and the maximum data errors and maximum chord errors both satisfy the machining tolerance requirement.

The whole G-code can be fitted by our algorithm at the same time. Fig. 11 plots the G-codes before and after B-spline fitting. The G-code in Fig. 11(a) and (b) are the portion of the G-code at the lower right corner of Fig. 11(a) and (b), respectively. The G01

points are colored with light green and the B-splines are colored with yellow. Fig. 11(a) is the original G-code formed by G01, and Fig. 11(b) is the G-code formed by G01 and B-splines.

In Fig. 11(b), the dark green sections are long segments, which do not be fitted for they are feature of the parts. We can see that the G-code after fitting is smoother than the original G01 points. The original G01 code only has  $G^0$  continuity, but the G-code after B-spline fitting have  $G^1$  continuity between G01 and B-spline, and between two B-splines. Moreover, the cubic B-spline without multiply knots has  $G^2$  continuity.

6.2. Comparisons with the LSPIA method

Fig. 12 compares the fitting result of LSPIA and ELSPIA. In Fig. 12(a), the fitting curves of both LSPIA and ELSPIA are plotted, and the details are plotting in Fig. 12(b) and (c), respectively. It can be seen that the fitted curve of ELSPIA is closer to the polyline formed by data points, so the chord error of ELSPIA is smaller than LSPIA, which demonstrates that the stretching energy term works and stretches the fitting curve to decrease the chord error.

6.3. Comparisons with the NX fitting result

In this section, we compare our fitting results with the fitting results of CAM software NX. The fitted G-codes are plotted in Figs. 13–15 and the fitting information is listed in Table 3.

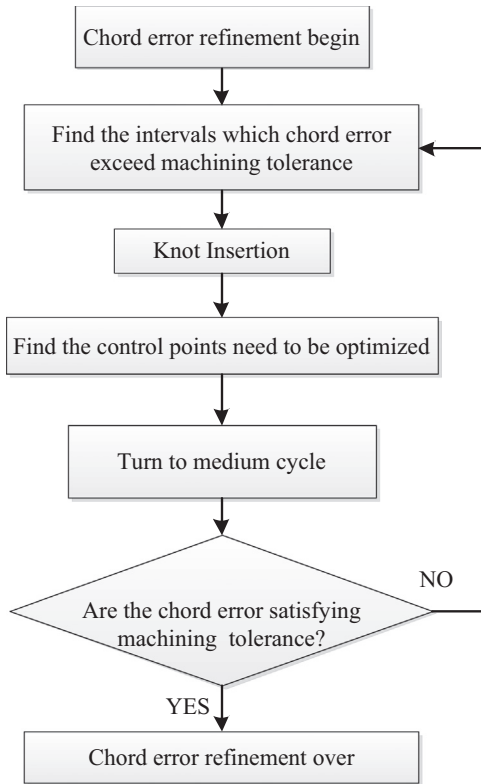


Fig. 9. Flowchart of chord error refinement.

Table 1 Fitting results before and after chord error refinement.

Fitting information	Before refinement	After refinement
Maximum data error [mm]	0.02028	0.01208
Maximum chord error [mm]	0.07492	0.02841

Table 2 Fitting information.

Fitting information	Examples			
	a	b	c	d
Machining tolerance [mm]	0.005	0.01	0.004	0.002
Fitting tolerance [mm]	0.0025	0.005	0.002	0.001
Number of data points	129	162	100	506
Number of control points	27	76	48	120
Maximum data error [mm]	0.0024	0.0058	0.0019	0.00099
Maximum chord error [mm]	0.0037	0.0098	0.0035	0.00125

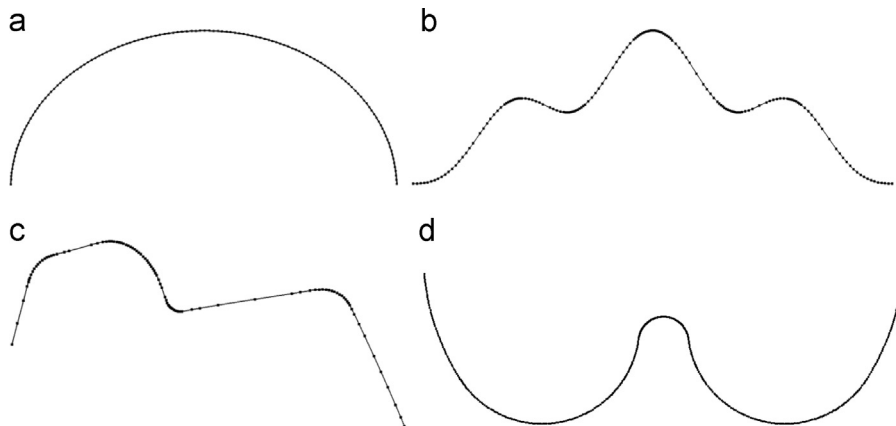


Fig. 10. The fitted curves: (a) example a, (b) example b, (c) example c and (d) example d.



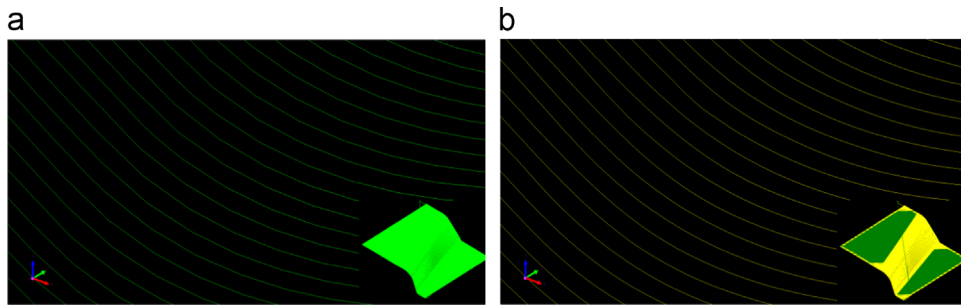


Fig. 11. G-code before and after B-spline fitting: (a) original G-code and (b) G-code after fitting.

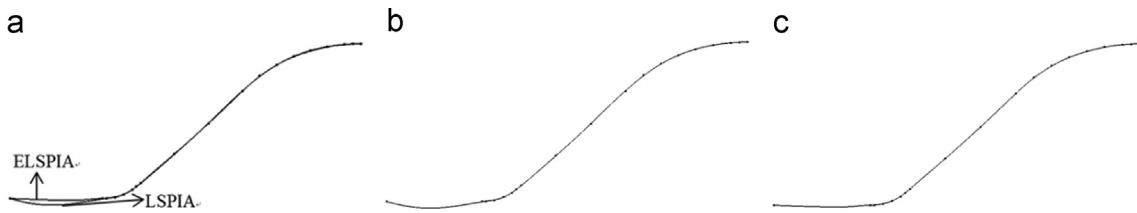


Fig. 12. Comparison between LSPIA and ELSPiA: (a) Fitted curves of LSPIA and ELSPiA, (b) Fitted curve of LSPIA and (c) Fitted curve of ELSPiA.

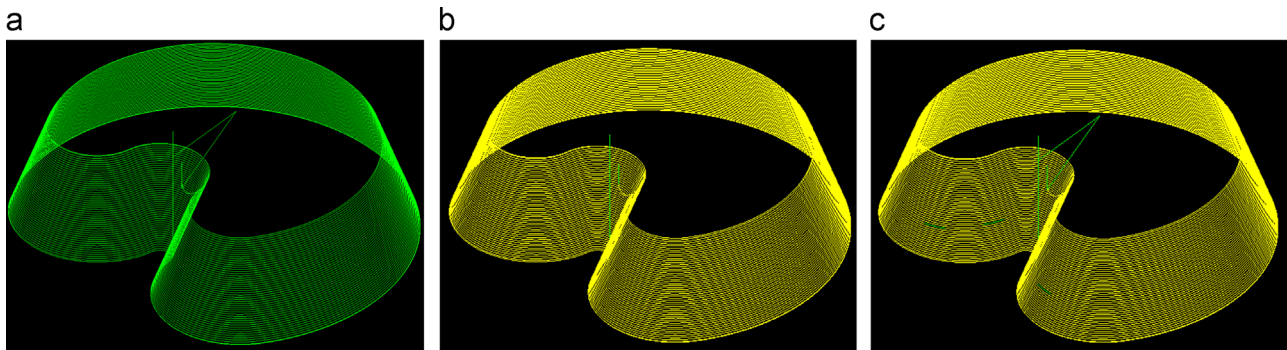


Fig. 13. B-spline fitting of workpiece "C-shape block": (a) original G-code, (b) NX fitting and (c) our method fitting.

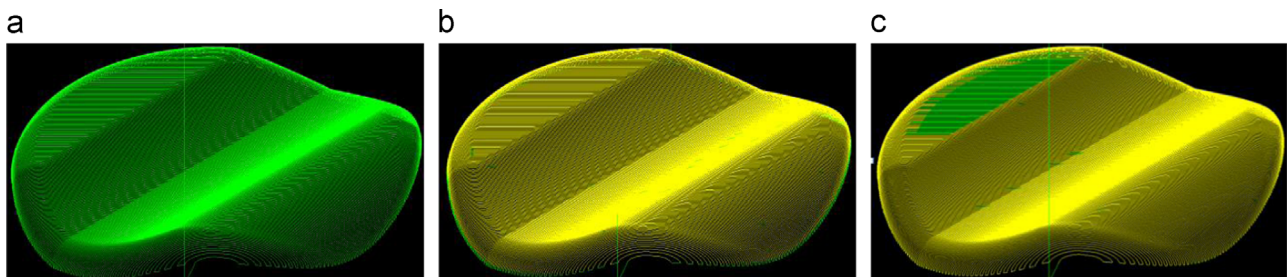


Fig. 14. B-spline fitting of workpiece "Bird's nest": (a) original G-code, (b) NX fitting and (c) our method fitting.

Figs. 13–15 demonstrates three workpieces' toolpaths fitted by NX and our method. In our method, the HBPs and long segment are identified in the preprocessing and the long segments remain unchanged in the fitting process, but NX fits the G-code without considering long segment. We can find the difference in Figs. 14 and 15.

Table 3 lists the fitting information of the three workpiece in Figs. 13–15. It can be seen that the G-codes fitted by our method have less points of  $G^1$  discontinuity and  $G^2$  discontinuity than NX, which demonstrates that our method can improve the continuity of tool paths. There are still several  $G^1$  discontinuity points because our method did not confine  $G^1$

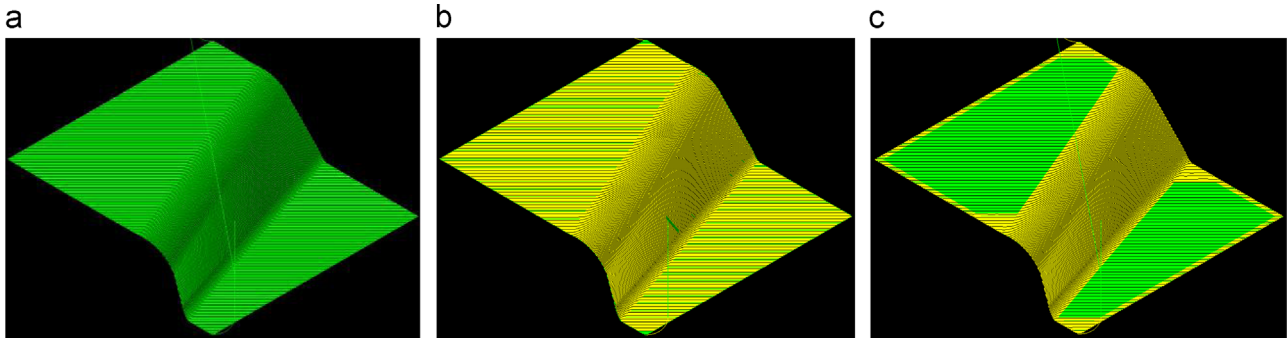


Fig. 15. B-spline fitting of workpiece “Variant curvature”: (a) original G-code, (b) NX fitting and (c) our method fitting.

Table 3  
Fitting information of NX and our method.

G-code name	Machining Tolerance [mm]	G01 number	Control points number	$G^1$ discontinuity number	$G^2$ discontinuity number	Maximum chord error [mm]	Average chord error [mm]
C-shape block	0.01	9472	NX 2879	89	99	0.010105	0.008757
			Our 3275	2	33	0.009981	0.004997
Bird's nest	0.001	43,660	NX 16550	5514	5682	0.001540	0.000675
			Our 27290	1082	1223	0.001011	0.000483
Variant curvature	0.01	11,512	NX 5843	2403	2603	0.010608	0.006646
			Our 6213	773	1297	0.010097	0.004567

continuity in the HBPs location. The maximum chord errors of NX and our result listed in Table 3 can reach chord error requirement. The average chord errors of our method are smaller than NX's of the three examples, which can make a contribution to obtain better machining results, but it may be one of the reasons that we have more control points than NX.

## 7. Machining experiments

### 7.1. Machining results

In this section, the proposed scheme is implemented in a Drilling and milling machining center (HNC(Huazhong Numerical Control)-818A/MD CNC system) Tool TC500R made by Shenyang Machine, as shown in Fig. 16. The configuration information of TC500R is listing in Table 4

In the machining, some machining parameters such as feed rate and spindle speed need to be set beforehand, we list the machining parameters of the “C-shape block”, “Bird's nest”, and “Variant curvature” in Table 5. The work-blank of the three adopted workpiece is cube with dimension of 40mm × 40mm × 40mm, and the material of the work-blank is 6061 aluminum alloy. The cutter used in the machining is a ball end mill made of cemented carbide.

The machining effects are shown in Figs. 17–19. Each workpiece is machined three times using: original G01, NURBS fitted by NX, and NURBS fitted by our method. To obtain more precise machining time measurement, we utilized



Fig. 16. Machining device: Drilling and milling machining center (HNC-818A/MD controllers) TC500R.

the cycle time of the servo drive to compute the actual machining time excluding the air (non-cutting) time of the cutter. As shown in Table 6, there is not much difference among the actual machining time of the three workpiece. The

Table 5  
Parameters for the actual machining.

G-code name	Feed rate [mm/min]	Spindle speed [r/min]	In-process stock from rough machining [mm]	Cut pattern	Step over
C-shape block	1000	6000	0.25	Helical	Number: 50
Bird's nest	1000	6000	0.1	Zig zag	Maximum Scallop Height: 0.0008 mm
Variant curvature	1000	6000	0.3	Zig zag	Maximum Scallop Height: 0.001 mm

Table 4  
Configuration information of TC500R.

Axis	X-axis	Y-axis	Z-axis	Spindle
Motor type	GK6063-6AF31-JE	GK6063-6AF31-JE	GK6063-6AF31-JB	SVM-90L-A2/10000
Rated torque [NM]	11	11	11	23.66
Rated speed [PRM]	3000	3000	3000	1500
Phase current [A]	13.5	13.5	13.5	14.4
Rated Power [KW]	3.5	3.5	3.5	3.7
Servo driver type	HSV-180UD-50	HSV-180UD-50	HSV-180UD-50	HSV-180US-50

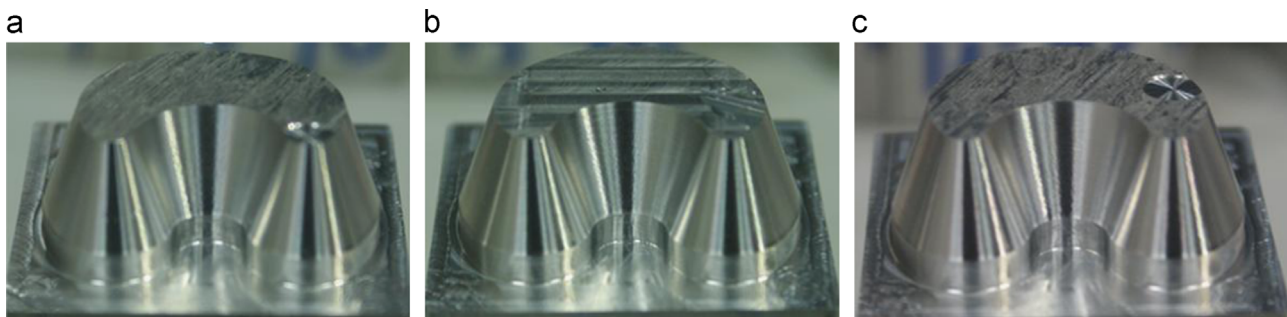


Fig. 17. Machining picture of workpiece “C-shape block”: (a) original G-code, (b) NX fitting and (c) our method fitting.

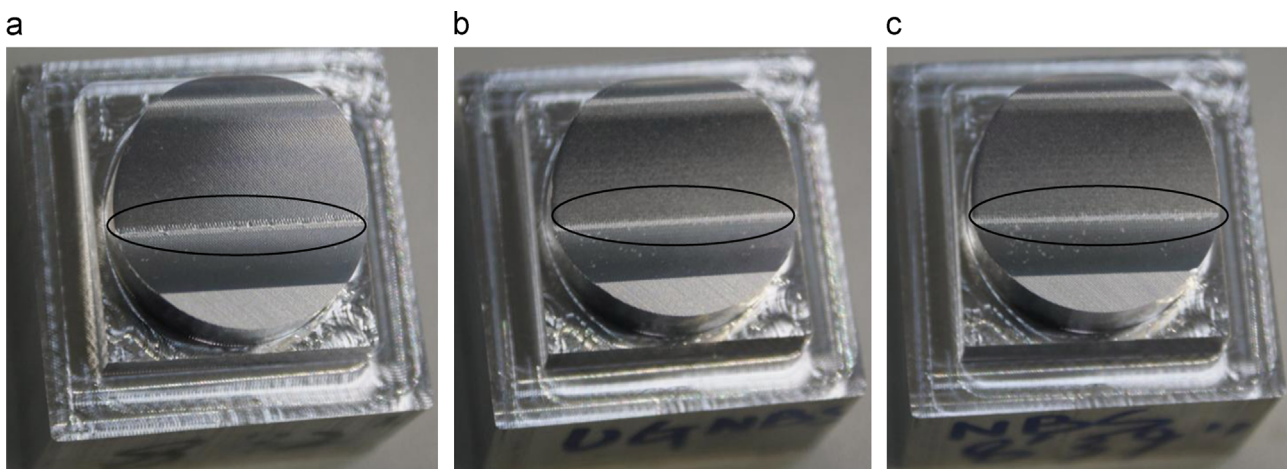


Fig. 18. Machining picture of workpiece “Bird's nest”: (a) original G-code, (b) NX fitting and (c) our method fitting.

result is not conclusive because only one commercial machine tool and CNC system is used for the tests but it reflects that there are rooms for improvements of both NX and our method.

### 7.2. Roughness testing

To evaluate machining quality, we measured the surface roughness of the machining workpiece. The device we used is

a Comprehensive Measurement System for Surface Profile TAYLOR HOBSON PGI830. As shown in Fig. 20. The measurement range of the device is 200mm in X-axis and 8 mm in Z-axis, the measurement accuracy is with resolution 0.8 nm/8 mm. In the testing, the sample length is set to 2 mm in the surface and 5 mm in the plane. We take the index Ra to measure surface roughness. The testing locations are marked in Fig. 21.

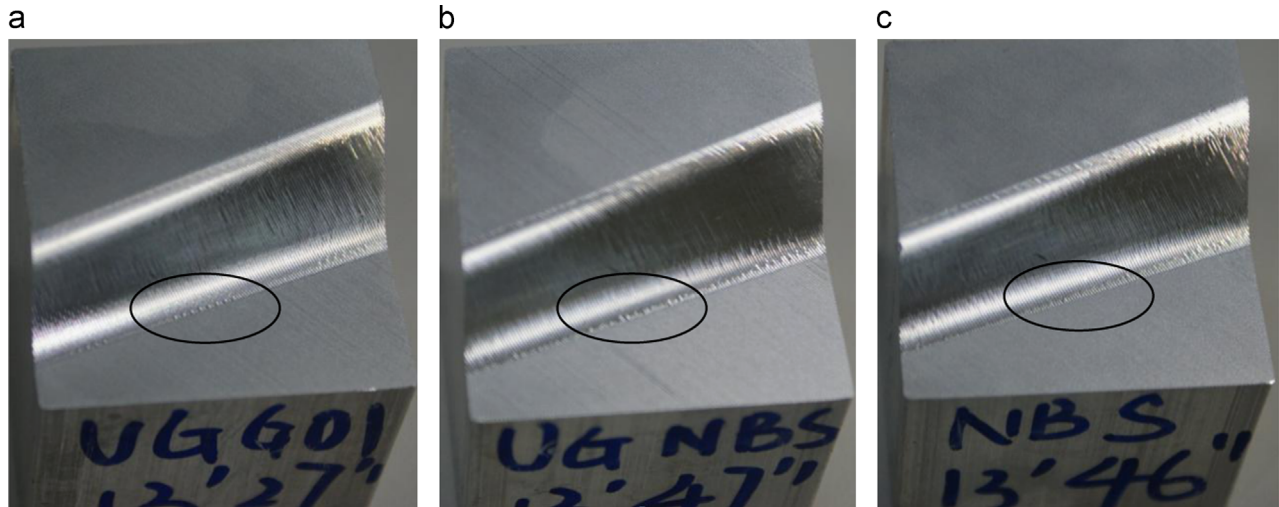


Fig. 19. Machining picture of workpiece “Variant curvature”: (a) original G-code, (b) NX fitting and (c) our method fitting.

Table 6  
Actual machining time of the three examples.

G-code name	Actual machining time [s]		
	G01	NX fitting	Our method fitting
C-shape block	374	374	374
Bird's nest	491	489	494
Variant curvature	794	800	792

We test every workpiece in three locations, as marked in Fig. 21. For each location, we test the roughness in two directions: follow and crossing, meaning it is following or crossing the cutting direction, respectively. We measure the roughness in two directions because both affect the surface quality, although tool path fitting focuses on only the quality of the follow direction. In order to remove the effects of errors, we test every location ten times and compute the mean value as the roughness of the location.

Summarizing the 9 groups of data in Table 7 regarding the crossing roughness, our method is better than G01 in 6 cases, and 1 case better than the NURBS of NX. As for the follow roughness, our method is better than G01 in 5 cases, and 6 cases better than the NURBS of NX. Overall, the differences are not significant enough to draw the conclusion.

## 8. Conclusion

This paper presents a new approach to approximate G01 paths with cubic B-spline curves. It can improve the continuity of tool paths and ensure efficient high-speed machining. We use a progressive and iterative approximation incorporating energy term to avoid the computational cost of solving linear equations and lower the chord error by using stretching energy term. We take three measures to satisfy the requirements for

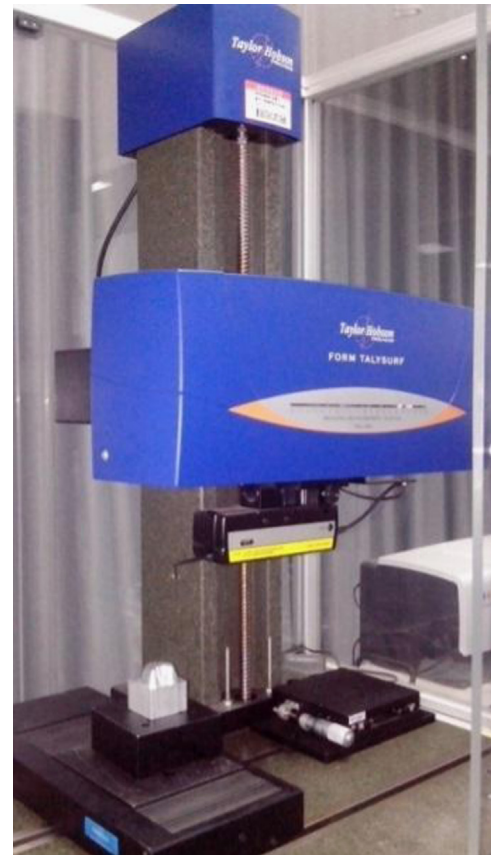


Fig. 20. Roughness test device: Comprehensive Measurement System for Surface Profile.

tool path fitting: ELSPIA, foot point parameters updating and control points' refinement.

In the experiment, we compared our fitting results with the original G01 tool paths and the fitting results of NX. Our method has less points of  $G^1$  and  $G^2$  discontinuity and smaller average chord error than NX's, but our method generates more control points than NX. Roughness testing

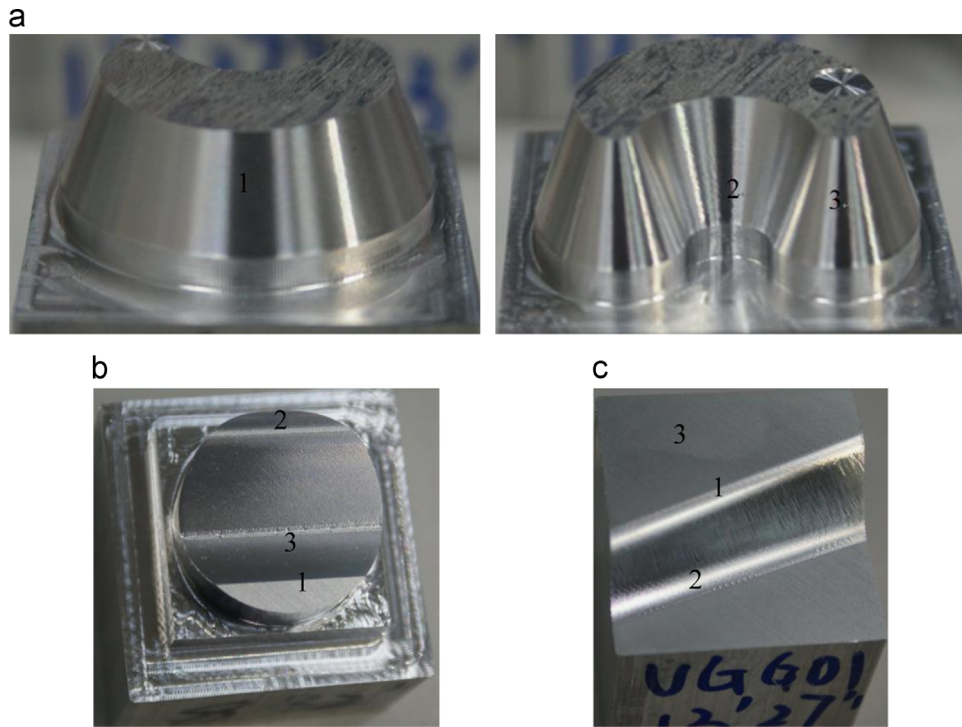


Fig. 21. Roughness testing location: (a) “C-shape block”, (b) “Bird’s nest” and (c) “Variant curvature”.

Table 7  
Roughness tests of the three examples.

G-code name	Crossing roughness Ra [μm]								
	Location 1			Location 2			Location 3		
	G01	NX	OUR	G01	NX	OUR	G01	NX	OUR
C-shape block	0.4084	0.416	0.3924	0.439	0.3988	0.4093	0.4157	0.3977	0.3997
Bird’s nest	0.633	0.6384	0.6659	0.5796	0.4729	0.7469	0.6021	0.6492	0.6594
Variant curvature	0.8778	0.7861	0.788	0.8194	0.7595	0.8155	0.562	0.5389	0.5439
G-code name	Follow roughness Ra[μm]								
	Location 1			Location 2			Location 3		
	G01	NX	OUR	G01	NX	OUR	G01	NX	OUR
C-shape block	0.2399	0.2824	0.2654	0.2096	0.2399	0.2432	0.301	0.3745	0.3092
Bird’s nest	0.776	0.7268	0.7589	0.4756	0.4361	0.4635	0.4064	0.4174	0.3986
Variant curvature	0.4654	0.7065	0.6261	0.3118	0.2762	0.2362	0.43	0.4898	0.4185

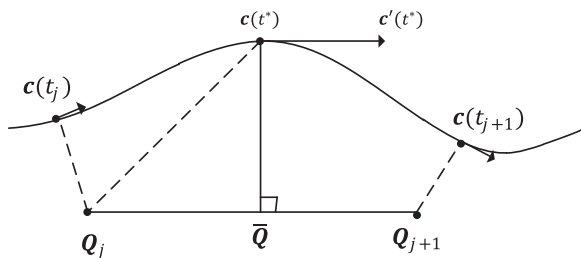


Fig. 22. Check chord error.

indicates that our method has better quality in some case, but worse quality than G01 or NX in some locations. Experiment reflects that the algorithm in this paper is robust and suitable

for general industrial applications but still have some improvement space.

Machining time and surface roughness are the ultimate measuring stick of the worthiness of tool path fitting algorithms. Our results showed slight improvement over the G01 and NX NURBS tool paths, but the results are not statistically significant to be conclusive. More work is required to enhance/modify the fitting algorithm to further improve the path continuity and surface quality. We believe the algorithm presented in the paper is robust and ready for real-world applications. It also provides a good start for further improvement.

Currently we are working on an algorithm to address the crossing roughness in order to achieve the overall surface quality. In addition, we plan to extend the tool path B-spline

fitting method to 5-axis machining. The crossing optimization of 5-axis machining is also under consideration.

### Acknowledgments

The authors gratefully acknowledge National Science and Technology Major Project of the Ministry of Science and Technology of China (2013ZX04007041). The authors would also like to thank the support from colleagues Haiqing Jiang, Zhengming Hu, Minmin Wang and Yanfen Huang.

### Appendix A. Compute discrete curvature

In this paper, given the data points  $\{\mathbf{Q}_j\}_{j=0}^m$ , for every  $\mathbf{Q}_j$ , the discrete curvature  $k_j$  is computed by second difference quotients. The computational steps are:

Step1: compute the chord length:  $s_j = \sum_{j=1}^m |\mathbf{Q}_j - \mathbf{Q}_{j-1}|$ ;

Step2: compute the first difference quotients:  $T[s_j, s_{j-1}] = \frac{\mathbf{Q}_j - \mathbf{Q}_{j-1}}{s_j - s_{j-1}}$ ;

Step3: compute the second difference quotients:  $T[s_{j+1}, s_j, s_{j-1}] = \frac{T[s_{j+1}, s_j] - T[s_j, s_{j-1}]}{s_{j+1} - s_{j-1}}$ ;

Step4: the discrete curvature is :  $k_j = 2T[s_{j+1}, s_j, s_{j-1}]$ ;

### Appendix B. Proof of Theorem 1.2

**Proof.** Letting  $\mathbf{I}$  be the  $n+1$  rank identity matrix and  $\mathbf{V} = \mathbf{A}^T \mathbf{A} + \omega \mathbf{D}$ .

$$\begin{aligned} \frac{1}{2} \nabla E(\mathbf{P}^{k+1}) &= \mathbf{V} \mathbf{P}^{k+1} - \mathbf{A}^T \mathbf{Q} \\ &= \mathbf{V}(\mathbf{P}^{k+1} - \mathbf{V}^{-1} \mathbf{A}^T \mathbf{Q}) \\ &= \mathbf{V}[\mathbf{P}^k + \mu \mathbf{H}(\mathbf{A}^T \mathbf{Q} - \mathbf{V} \mathbf{P}^k) - \mathbf{V}^{-1} \mathbf{A}^T \mathbf{Q}] \\ &= \mathbf{V}(\mathbf{I} - \mu \mathbf{H} \mathbf{V})(\mathbf{P}^k - \mathbf{V}^{-1} \mathbf{A}^T \mathbf{Q}) \\ &= \dots \\ &= \mathbf{V}(\mathbf{I} - \mu \mathbf{H} \mathbf{V})(\mathbf{P}^0 - \mathbf{V}^{-1} \mathbf{A}^T \mathbf{Q}) \end{aligned}$$

Note that when  $0 < \mu < \frac{2}{\lambda_0}$ ,  $0 < \rho(\mathbf{I} - \mu \mathbf{H} \mathbf{V}) < 1$ , where  $\rho(\mathbf{I} - \mu \mathbf{H} \mathbf{V})$  is the spectral radius of  $(\mathbf{I} - \mu \mathbf{H} \mathbf{V})$ . Therefore,

$$\lim_{k \rightarrow \infty} (\mathbf{I} - \mu \mathbf{H} \mathbf{V})^k = (\mathbf{0})_{n+1}$$

Where  $(\mathbf{0})_{n+1}$  is an  $n+1$  rank zero matrix.  $\frac{1}{2} \nabla E(\mathbf{P}^\infty) = 0$ . So the ELSPIA method is convergent after several iterative steps.

### Appendix C. Weight selection

For Eq. (13), the iterative sequence is convergent when  $0 < \mu < \frac{2}{\lambda_0}$ .  $\lambda_0$  is the largest eigenvalue of  $\mathbf{H}(\mathbf{A}^T \mathbf{A} + \omega \mathbf{D})$ . We can give a practical best weight by estimating the eigenvalues of  $\mathbf{H}(\mathbf{A}^T \mathbf{A} + \omega \mathbf{D})$  with Gershgorindisc theorem [19]. The practical best weight is:  $\mu = \frac{2}{C}$ ,  $C = \max_i \{c_i\}$ , ( $i = 0, 1, \dots, n$ ),  $c_i$  is the

sum of the absolute value of the  $i$ th row elements of  $\mathbf{H}(\mathbf{A}^T \mathbf{A} + \omega \mathbf{D})$ .

### Appendix D. Check chord error

In this study, we check chord errors by segment processing. As shown in Fig. 22, we can find the corresponding curve segment  $\mathbf{c}(t)$  ( $t \in [t_j, t_{j+1}]$ ) for segment  $\mathbf{Q}_j \mathbf{Q}_{j+1}$ , then the maximum distance is measured by the distance of common vertical segment. The purpose is to find a parameter  $t^*$ , s.t.  $\mathbf{c}(t^*) \bar{\mathbf{Q}} \cdot \mathbf{c}'(t^*) = 0$ .  $\bar{\mathbf{Q}}$  is the pedal from the point  $\mathbf{c}(t^*)$  to do vertical line for the segment  $\mathbf{Q}_j \mathbf{Q}_{j+1}$ .  $\mathbf{c}'(t^*)$  is the tangent vector at the point  $\mathbf{c}(t^*)$ .

To solve the equation  $\mathbf{c}(t) \bar{\mathbf{Q}} \cdot \mathbf{c}'(t) = 0$ , replace  $t$  with  $x$  in order not to make confusion with data point parameters  $t_j$ , we translate it into solving the minimize problem of the following function:

$$f(x) = \mathbf{Q}_j \mathbf{c}(x) \times \mathbf{Q}_j \mathbf{Q}_{j+1} \times \mathbf{Q}_j \mathbf{Q}_{j+1} \cdot \mathbf{c}'(x)$$

We use the Newton iteration method to solve the problem, set an initial value  $x_0$ , the iteration step is:

$$\begin{aligned} x_{i+1} &= x_i - \frac{f(x_i)}{f'(x_i)} \\ &= x_i - \frac{\mathbf{Q}_j \mathbf{c}(x) \times \mathbf{Q}_j \mathbf{Q}_{j+1} \times \mathbf{Q}_j \mathbf{Q}_{j+1} \cdot \mathbf{c}'(x)}{\mathbf{Q}_j \mathbf{c}(x) \times \mathbf{Q}_j \mathbf{Q}_{j+1} \times \mathbf{Q}_j \mathbf{Q}_{j+1} \cdot \mathbf{c}''(x) + \mathbf{c}'(x) \times \mathbf{Q}_j \mathbf{Q}_{j+1} \times \mathbf{Q}_j \mathbf{Q}_{j+1} \cdot \mathbf{c}'(x)} \end{aligned}$$

### References

- [1] Yan CY, Lee CH, Yang JZ. Three-axis tool-path b-Spline fitting based on preprocessing, least square approximation and energy minimization and its quality evaluation. *Modern Machinery (MM) Science Journal* 2012.
- [2] Park H. An error-bounded approximate method for representing planar curves in B-splines. *Computer Aided Geometric Design* 2004;479–97.
- [3] Park H, Lee J. B-spline curve fitting based on adaptive curve refinement using dominant points. *Computer-Aided Design* 2007;39:6.
- [4] Deng C, Lin H. Progressive and iterative approximation for least squares B-spline curve and surface fitting. *Computer-Aided Design* 2014;47(1) 32–44.
- [5] Piegl L, Tiller W. *The NURBS book*. London, UK: Springer-Verlag; 1995.
- [6] Wesselink W, Veltkamp RC. Interactive design of constrained variational curves. *Computer Aided Geometric Design* 1995;12:533–46.
- [7] Vassilev TI. Fair interpolation and approximation of B-splines by energy minimization and points insertion. *Computer Aided Design* 1996;28(9) 753–60.
- [8] Park H, Kim K, Lee S-C. A method for approximate NURBS curve compatibility based on multiple curves refitting. *Computer-Aided Design* 2000;32:237–52.
- [9] Qi D, Tian Z, Zhang Y, Feng J. The method of numeric polish in curve fitting. *Acta Mathematica Sonica* 1975:173–8418 1975:173–84.
- [10] De Boor C How does Agee's smoothing method work? In: Proceedings of the 1979 Army Numerical Analysis and Computers Conference; 1979; Army Research Office; pp. 299–302.
- [11] Lin H, Wang G, Dong C. Constructing iterative non-uniform B-spline curve and surface to fit data points. *Science in China Series: Information Sciences* 2004;47(3)315–31.
- [12] Lin H, Bao H, Wang G. Totally positive bases and progressive iteration approximation. *Computers and Mathematics with Applications* 2005: 575–86.
- [13] Lin H, Zhang Z. An extended iterative format for the progressive-iteration approximation. *Computers & Graphics* 2011;35(5)967–75.

- [14] Shi L, Wang R. An iterative algorithm of nurbs interpolation and approximation. *Journal of Mathematical Research and Exposition* 2006;**26**:735–43.
- [15] Lu L. Weighted progressive iteration approximation and convergence analysis. *Computer Aided Geometric Design* 2010;**27**:129–37.
- [16] Piegl L, Tiller W. Surface approximation to scanned. *The Visual Computer* 2000;**16**(7):386–95.
- [17] Razdan A. *Knot placement for B-spline curve approximation*. Arizona State University; 1999.
- [18] Li W, Xu S, Zhao G, Goh LP. Adaptive knot placement in B-spline curve approximation. *Computer Aided Geometric Design* 2005;**37**(8):791–7.
- [19] Yang XJ, Chen ZC A new high precision fitting approach for NURBS tool paths generation. In: International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. September 24–28; 2005; Long Beach, California USA; pp. 255–262.
- [20] Yeh S, Su H. Implementation of on line NURBS curve fitting process on CNC machines. *The International of Advanced Manufacturing Technology* 2009;**40**(5/6):531–40.
- [21] Zhang M, Yan W, Yuan, CM, et al. Curve fitting and optimal interpolation on CNC machines based on quadratic B-splines. *Science China Information Sciences* 2011;**54**(7):1407–18.
- [22] Burden RL, Faires JD. *Numerical analysis*. Boston: PWS-Kent Pub; 1989.