An Approach to Feed Rate Determination for High-Speed Machining of 2-D NURBS Profiles

Tianbo Zhao

A Thesis

in

the Department

of

Mechanical & Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

December 2007

Your file Votre référence
ISBN: 978-0-494-40928-2
Our file Notre référence
ISBN: 978-0-494-40928-2

Abstract

An Approach to Feed Rate Determination for High-Speed Machining of 2-D NURBS Profiles

Tianbo Zhao

High-speed machining is a new trend in the manufacturing industry, especially in the mould and aerospace industries. Compared to conventional CNC machining techniques, high-speed machining has two obvious advantages: high feed rates and high part accuracy, which requires some new requirements on machining parameters. One requirement is on the material removal rate (MRR). In non-high speed machining, the MRR is not necessary to be kept constant. However, a constant MRR is a basic requisite in high-speed machining (HSM) processes, because the tool load is very sensitive to the MRR in high-speed machining. Quick tool wear, unexpected breakage, and deflection of tool occur frequently if the MRR is higher. Generally, the mould and aerospace industries have many parts including free surface or complex curves represented by NURBS (Non-Uniform Rational B-Splines). In this work, research is conducted to ensure the MRR be constant in machining 2-D NURBS curve profiles. Another peculiarity of the high-speeding machining is that its toolpath must be smooth without sharp direction changes. Overcut or gouging may occur for momentum of the tool is too big in this machining. If there are many sharp connections in linear toolpath, the feed rate cannot reach to the specified high value, because of frequent starts and stops of the servo-motor, shortening the motor life. In this research, effective NURBS toolpaths are generated to machine complex curve profiles, like NURBS or B-Spline contours.

In this thesis, an intelligent approach to determining the machining parameters for NURBS profiles is established. First, a 2-D geometric model of machining NURBS profiles is given. Second, a cutting force model is established based on the artificial neural-network. Third, appropriate feed rates are gotten from the predicted cutting force of machining. Then new NURBS segments are proposed to represent the previous curve in the specified tolerance. Finally, simulation and NC programs are given to demonstrate the advantages of this research.

Acknowledgements

Firstly, I would like to express my deepest gratitude to my supervisor, Dr Chevy Chen. His illuminating instruction, encouragement, and trust help me finish my research. Without his support, this thesis could not reach its present form. Secondly I would thank my parents, my wife, and all of family members for their support. They often encourage me to overcome the difficulties in my work. Finally I want to thanks for Zheng Hong, Huang guogui, Fu Qiang, and all the members of Dr Chen's Lab. They give me a lot of useful advices and information for my work. Also I thank Concordia University faculty and stuff whose assistance deserve to be appreciated.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

CNC stands for Computer Numerical Control and, specifically, refers to a computerized "controller" that reads code instructions and drives machine tools to fabricate parts [1]. Before computer control was introduced into control in manufacturing, the numerical control (NC) technique was used. The primitive NC appeared in the early days of industry revolution used punch cards to make different patterns in cloth. The industry revolution laid a firm foundation for this modern machining technology. In 1950's, the world's first NC machine tool was made in the Servo-mechanism Laboratory at the Massachusetts Institute of Technology (MIT) in order to make complex-shaped parts, like helicopter rotor blades and missile components for U.S. air force. Then many tools manufacturers began to make various NC machine tools. Up to 1970s, the electronic industry could make small, reliable integrated circuits, and the NC hardware was gradually replaced by computers and PLCs; eventually, NC evolved to CNC.

Computer Numerical Control (CNC) operates machine tools in an accurate way and eliminates possibility of human errors by executing fixed codes in memory. Furthermore, CNC can machine complex shape of contours, pockets, or free-formed surfaces that cannot be manually done with accuracy. Non-stop pursue for higher productivity in

industry drives the modern CNC technique be more powerful. By the 80's of last century, high speed machining (HSM) became more applicable in some industrial sectors, like the mould manufacturing and aeronautical industries. This technology has been successfully used with significant improvements in machine tools, spindles and controllers, and it is becoming a cost-effective manufacturing process to produce parts with high precision and surface quality.

1.1 High-Speed Machining Technique

Salomon C., the father of high-speed machining, found that the machining temperature starts to drop when the cutting speed exceeds a certain value (see Fig.1-1) [2]. His fundamental research showed that a machining process becomes easy if the cutting speed reaches a certain high value. However, until today, there is no unique definition about high-speed machining, because it depends on the workpiece material, machine tool, cutting tool (including inserts), and specific application. Currently, some researchers define high-speed machining as a machining whereby the conventional cutting speeds are exceeded by a factor of 5 to 10.

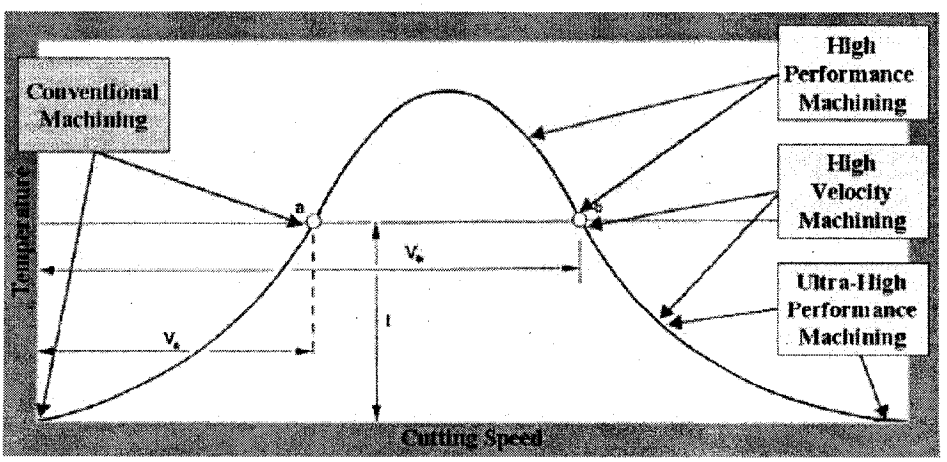


Fig.1-1 Relationship between cutting temperatures and speeds.

With wide use of CNC machines, high-speed machining has demonstrated its superior advantages over other rapid manufacturing techniques. In addition to increased productivity, high-speed machining is capable of generating high-quality surfaces, burr-free edges, and virtually stress-free components, and can be used to produce thin-wall parts, because the cutting forces in high-speed machining are lower. Another significant advantage is minimum heat effects on machined parts. Most heat generated during cutting is removed, reducing thermal deformation and increasing tool life. In many cases, the need for coolant in cutting is eliminated, which, subsequently, reducing pollution and saving money. Due to these advantages, the high-speed machining is widely used in the aerospace, automotive, and machine tool industries for high quality part and high productivity.

However, the high-speed machining usually is conducted on special, expensive machine tools with advanced spindles and controllers, fixtures, tool holders. In addition, it also needs advanced cutting tool materials and coatings.

- High-speed spindle

High-speed spindles are the most fundamental machine tool components for a high speed machining process. A high-speed spindle presents the tradeoff between cutting forces and cutting speeds. There are two limitations on high-speed spindles. The first one is that the spindle motor size is limited. High-speed spindles generally have direct-drive motors, which are fixed inside the spindles. So, the motors cannot be large in size. Another limitation is on the spindle bearing. The bearings of the high-speed spindles have to trade their stiffness for high speeds. Because of these limitations, high-speed

machining employs smaller radial depths of cut.

- Tool holder

The tool holder is the link between the tool and spindle, which affects the concentricity and balance of tool in high-speed rotation. Since high-speed milling generally has smaller radial depths of cut, the cutting load is lighter than that in conventional milling. Under this circumstance, potential variation of cutting loads introduced by tool run-out can be more significant. Cutting tools for high-speed machining have high wear and heat resistance, but low toughness. In high-speed machining, good tool concentricity is essential to achieving acceptable tool life and machining accuracy. In industry, mainly, there are three types of clamps in high-speed machining tool holders (see Fig1-2 ..).

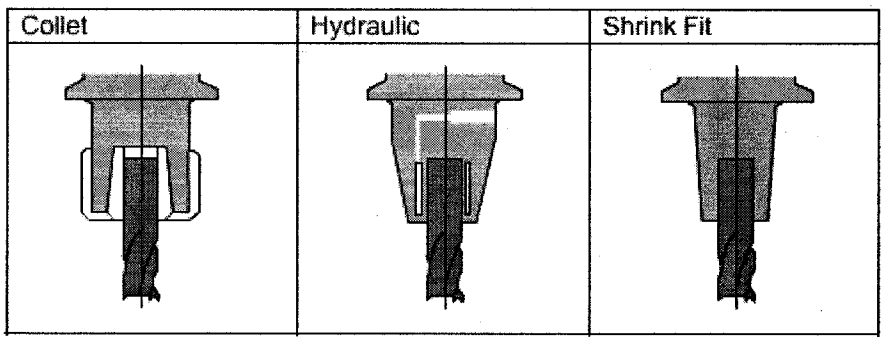


Fig.1-2 Three common type of tool holder in high-speed machining

- Collet: A collet tool holder most likely is the choice of machine shops. It is rigid and inexpensive, but is not easy to use and the accuracy depends on operators' experience. In general, its accuracy can be less than 10 microns.

- Hydraulic: A hydraulic tool holder needs a reservoir of oil to provide clamping pressure around the tool. It is easy to use, and the accuracy is consistent; however, its rigidity is not as good as the collet tool holders, plus it is more expensive.
- Shrink fit: A shrink fit tool holder works in conjunction with a specialized heater. By heating the tool holder, its bore opens up to allow inserting the tool. As the tool holder cools, the bore shrinks around the tool to create a concentric and rigid clamp. This type tool holder has very good concentricity, balance, and rigidity.

- Cutting tools for high-speed machining

High speed steel cutters cannot undertake the high temperature and shock in high-speed machining, so carbide tools are often used. Fine grain-size carbide tools is a good compromise between hardness and toughness, thus they can resist wear and shocks. An alternative tool is the TiAIN coated tool. The TiAIN coating has the ability to retain wear resistance at higher temperatures. In addition, high-speed machining can be conducted by using tools of ceramic, CBN and PCB, which tend to offer higher wear resistance, but less resistance to shock.

1.2 Applications of High-Speed Machining

With the advances in machine and cutting tools, high-speed machining becomes a cost-effective manufacturing process in the mould and aerospace industries to efficiently produce accurate parts.

- Applications in the mould industry

Before the high-speed machining technique was introduced in the mould industry, the manufacture of mould was a very time-consuming process and a stressful job. One reason is that mould materials are very hard to cut. The tolerance of moulds is very tight, sometimes less than 0.0004 inches. In a conventional way, 0.03 inch material is left in finish machining for hand polishing, which might cause days or even weeks depending on the machinist experience. Today, with the support of high-speed machining centers, manual polishing can be shortened to hours, even completely saved. Machined surface finish is largely determined with the heights of the cusps between adjacent toolpaths. "Machining to zero" is the premise of high-speed machining in mould manufacturers, which means there is no or very little hand polishing work left after CNC machining. That is not only shortening the spotting time, but also streamlining the production of manufacture process. For example, a drink bottle mould can be machined in an hour on a high-speed machining center, which saves a lot of machining time.

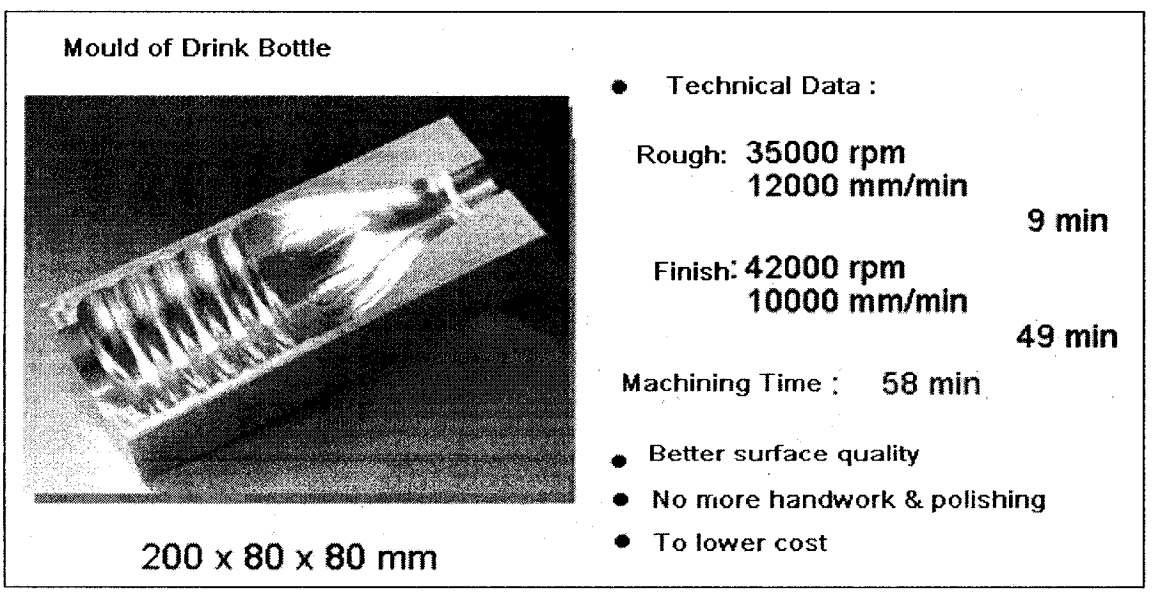


Fig.1-3 Application of HSM in mould industry

- Applications of high-speed machining in the aerospace industry

Many components in the aerospace industry have two distinct features, monolithic and thin-walled. Due to these features, they are almost impossible to be fabricated in conventional CNC machining. A monolithic part refers to a large, intricate aircraft component that is produced from a large solid workpiece. In previous, this type of part was made by assembling several sub-components or by casting. It is evident that the monolithic structures can drastically reduce part counts, assembly costs, and maintenance costs. For instance, a Boeing F/A-18E/F tactical fighter uses 42 percent fewer parts than the earlier models in the same family, resulting 25 percent weight savings [3]. The thin-walled structures, whose thickness can be less than 0.008 inch, have to be cut in high-speed machining with small depth of cut, combining with high feeds.

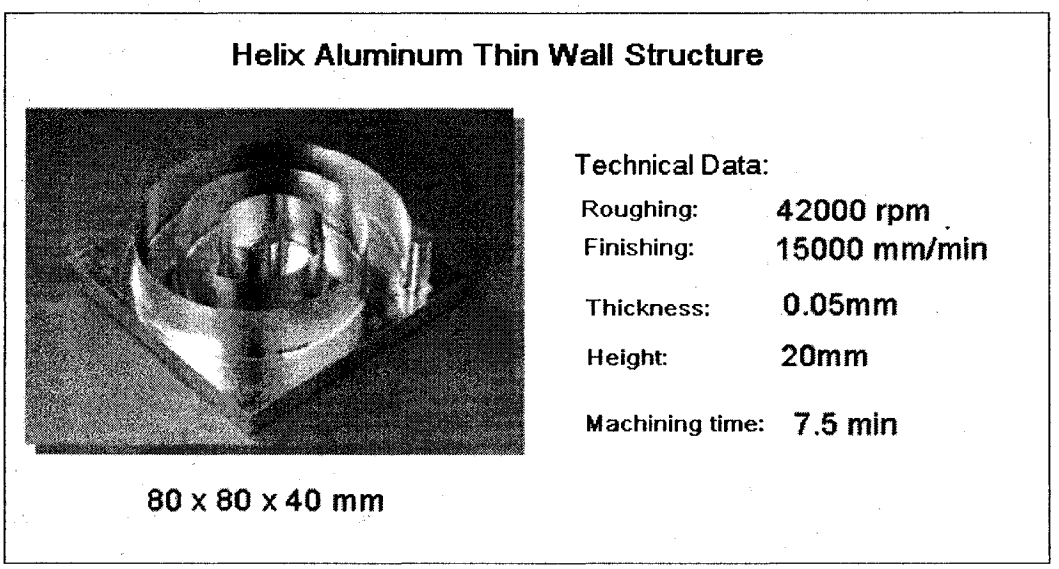


Fig.1-4 Application of HSM in aerospace industry

1.3 Literature Review

In high-speed machining, cutting forces, machining accuracy, and tool life are the major concerns, and high cutting efficiency is always pursued. These concerns are directly related with the machining parameters such as spindle speed, depth of cut, and feed rate, which are specified in NC programs. In normal CNC machining, the parameters are determined according to machinist's handbooks or experience. Even with some databases of high-speed machining provided by industry and research institutes, it is still quite challenge for NC programmers to determine appropriate parameters for high-speed machining. Unfortunately, in-appropriate parameters in high-speed machining surely result in rough surface finish, surface gouge, and severe tool wear (even tool breakage). Therefore, the parameters in high-speed machining should be optimized in order to make high-quality parts and reduce machining costs.

As more parts with NURBS contours and/or surfaces are employed, proper NURBS toolpaths should be generated and applied to the high-speed machining of these parts. Currently, high-speed machine tools with new controllers can provide NURBS tool motions; however, many commercial CAM software systems cannot generate NURBS toolpaths for high-speed machining of sculptured parts. Instead, some of them are still generating linear paths for this purpose, resulting in large-sized NC files and lower feed rates than specified values. Some commercial CAM software provides trochoidal toolpaths without tool overload at corners and tool full immersions. However, in machining NURBS profile parts, effective depths of cut are various and it is difficult to keep the material removal rates constant by using linear, arc, and trochoidal toolpaths. Although much research has been conducted on NURBS curve fitting and a few software

systems can plan NURBS paths, no work has been proposed to generate NURBS paths suitable for high-speed machining.

1.3.1 ANN Model of Cutting Forces

The aim to keeping the material removal rate in high-speed machining constant actually is to prevent tool-workpiece engagements and cutting forces from rigorous fluctuation. A large cutting force change is a main source of tool chatter, rough surface, and surface gouge and undercut. Hence, before machining, programmers should determine appropriate feed rates and depths of cut through accurate cutting force predictions, in order to have cutting forces remain at the same level. The major challenge in cutting force prediction is to find the quantitative relationship between cutting forces and machining parameters. Over the past two decades, a large number of articles have been published to address cutting force modeling for various manufacture processes, which can be classified into three categories: analytical, empirical, and finite element analysis methods.

The analytical method is based on the mechanics of metal cutting, and the cutting force differs from tool's geometry and dynamitic character of CNC machines. The drawback of analytical models is that predicted cutting force is not accurate due to a series of hypotheses. According to Altintas' estimation [4], the accuracy of analytical models basically is about 70% to 80%. The second method of predicting cutting forces is constructed on extensive empirical machining data. In this method, many groups of machining experiments are conducted with different parameters, and then cutting forces

are calculated by interpolating the experimental data in statistical ways. This method can predict cutting forces with high fidelity, except high experiment costs. Many research works have been conducted to combine the two above-mentioned methods together in order to attain more accurate and practical cutting force models [5, 6]. Artificial neural networks have been adopted in the empirical method. In particular, the multi-layer network with strong capability to simulate non-linear processes is quite suitable for modeling metal cutting processes. The third method, finite element analysis, is a promising way to simulate metal removing, including chip flow and morphology, cutting forces, and residual stresses and cutting temperature fields; some of the factors are beyond the capability of current measurement methods. However, this method is very time-consuming; consequently, it has not been widely used in industry [7].

To accurately predict cutting forces in a machining process is a prerequisite for optimizing machining parameters. Many researchers have proposed different models of cutting force prediction for specified tools and machining conditions. Tlusty [6] conducted experiments to determine cutting force in different geometric profile under same depth and width of cut. Kline [5] proposed a mechanistic model for calculating cutting forces in flat end-milling, and then applied to corner machining for estimating cutter deflections and surface errors. These two works were mainly based on engagement geometries and the empirical relationship of cutting forces and chip geometries. Fussell [8] presented a method to find a function of cutting forces in terms of feed rates on-line during end milling operations. Therefore, the depth of cut and feed rate could be adjusted according cutting forces measured on-line. Theoretically, the way of measuring cutting forces on-line and then adjusting the machining parameters could be a solution, but it is

not practical due to some technical difficulties. Fortunately, by retrieving geometric models of NURBS profile parts from the CAD/CAM system, depths of cut along NURBS toolpaths can be accurately computed based on the geometric models, and cutting forces can be predicted by using a machining database. Within several loops of depth of cut adjustment, cutting forces are ensured remain the same during high-speed machining.

Since machining processes are highly non-linear and case dependent, it is difficult for the traditional identification methods to provide an accurate model. Compared to these methods, the artificial neural networks (ANNs) can identify an implicit relationship between input(s) and output(s) by learning from a set of data that represents a system (or process), and they are more robust and global. With the features of universal approximation, parallel distributed processing, hardware implementation, learning and adaptation, and multivariable systems, ANNs are widely used for system modeling, function optimizing, image processing, and intelligent control. Around 1990s', some researchers began to introduce artificial neural networks to monitor machining processes. Vann and Charles [9] developed a system to represent a machining process by collecting, processing, and interpreting data from sensors. They used ANNs to identify patterns associated with simple machining models. Later, many researchers have established different kinds of neural networks models to describe machining progresses. Mostly, the multi-layer back-forward networks were adopted to compute cutting forces, and the weights of the networks were identified by using the iterative back-propagation (BP) learning process. Szecsi [10] presented a feed forward neural network algorithm to predict cutting forces. Okafor et al [11] evolved the knowledge of a machining process by training these networks on-time. Luo et al [12] proposed a modified BP ANN, which

adjusts its learning rate and adds a dynamic factor in the learning process for on-line modeling of a milling system. Apart from the commonly used feed-forward networks, some researchers tried to use self-organized networks to represent machining processes. Xu [13] used recurrent neural networks to identify cutting forces in end milling operations. This kind of neural network model has a feedback loop where data can be fed back to input, making it suitable for some on-line applications; however, it is less accurate than the feed forward neural networks, because the feedback sometimes makes the network model confused and unstable. El-Mounayri [14] used the radial basis network (RBN) to establish the relationship between machining condition and the cutting forces.

1.3.2 Feed Rate Adjustment and NURBS Toolpath Interpolation

Feed rate adjustment should be carried out according to cutting forces, which are dependent on the parameters such as workpiece material, tool (material and size), depth of cut, and spindle speed, etc. If one of the parameters is changed, the cutting force varies accordingly. To cut NURBS profiles by moving the tool along NURBS toolpaths, the depth of cut always changes, resulting in unexpected cutting force variation. This can be a main resource of low surface quality, and surface gouge is prone to occur. In addition, fluctuating cutting forces can overload the tool, wearing the cutting edges or even break the tool. Adjusting feed rates along a toolpath is an effective way to keep the cutting force constant, and many researchers have tried to optimize feed rates based on different geometric and mechanic models.

Law and Kris [15] conducted mathematical analysis on cutting force and feed rate

adjustment in 2-D corner machining. According to their method, the machining error at corners could be reduced to 60%. Bae [16] proposed an automatic approach to adjusting feed rates in pocketing. His approach is simple and effective, and it establishes a mean cutting force model with cutting engagement geometries in 2-D machining condition. In his approach, exhaustive experiments are carried out to test cutting forces, and then adjust feed rates. His research is very practical in 2-D pocketing of regular features; however, it cannot solve the problem of feed rate adjustment for complex NURBS profile milling. Unfortunately, there is not a simple and effective model of the relationship between the feed rate and other parameters for machining the NURBS or B-spline profiles.

Among the existing methods for milling these profiles, commercial CAM software approximates these curves with short chords and arcs within the specified tolerance, and then the tool cuts along them in a same feed rate. However, this method has two problems. The first one is that the surface quality is not the same under a pre-specified feed rate. The reason for this is that due to curvature change along the NURBS profile, the effective depth of cut along the toolpaths varies, and cutting load is not well-balanced, consequently, the tool may be overloaded or under loaded. Particularly, in high-speed machining, the cutting force is very sensitive to cutting parameters; a little increment of cutting depth can result in tool overload. Another problem is that a large amount of chords and arcs in a NC program reduce the feed rates because of frequent starts and stops of the motor.

Some researchers adopted the NURBS interpolation strategy for optimize feed rates. Cheng presented a real-time control algorithm for implementing variable feed rates in

NURBS interpolators for precise CNC machining [17]. He built a real-time controller to realize variable feed rate interpolation and eliminate path command errors caused by acceleration and deceleration of the tool. His scheme of feed rate adjustment was to satisfy the dynamic characteristics of tool motions based on on-line interpolations, but he did not provide any analysis if the geometric error of a NURBS toolpath was within tolerance. In other hand, additional electronic hardware was needed to realize the function, and the difficulty of integration with CNC control hindered its application in industry. Beside, several researchers have developed real-time parametric interpolators for curve generation by using Bezier and B-spline, which are the main curve interpolation techniques used in CNC controllers.

1.4 Objectives and Outline of This Thesis

The main objective of this thesis work is to optimize feed rates for 2-D high-speed milling of NURBS profile parts. There are two main tasks finished in this work. Firstly, the adjustment of feed rate is worked out, to keep the material remove rate (MRR) constant along the NURBS profile. Secondly, the segment of NURBS toolpath is refitted and connected smoothly for the application for high-speed machining. It includes six chapters, Chapter one introduces CNC machining, high-speed machining, NURBS interpolator techniques, and its applications in the aerospace and mould industries. A literature review is conducted, and the thesis objective is stated. Chapter two renders theoretical preparation for 2-D geometric models for machining NURBS profiles. In Chapter three, a cutting force model is established based on the multi-layer neural network. Training data for the ANNS are obtained from CutData, a popular machining

parameter database. Chapter four presents several off-line NURBS toolpath interpolator methods. Chapter five addresses an integrated program in order to identify the geometric features of NURBS profiles, automatically adjust feed rates based on predicted cutting forces, and output control points and knot vector of NURBS toolpaths. The result is simulated in the CATIA, MATLAB, and MasterCAM systems to demonstrate the advantages of this approach. Finally, the summary and contributions of this work are contained at the end of thesis.

Chapter 2 Geometric Model for 2D NURBS Profile Milling

2.1 Basic Concept of CNC Milling

CNC milling machines can be grouped into 2-, 3-, and 5-axis machines. The 2-axis milling plays an important role in machine shops, since many products are designed for machining on traditional machine tools. Among the machining processes, 2-D contours, pockets, holes and threads can be produced on 2-axis machine tools. Now, in industry, 3-axis milling machines are the mainstream of machine tools. The main difference between the 3- and 2-axis milling is that the z-axis of a 3-axis machine tool can simultaneously move with its x- and y-axes. Complex surfaces can be machined on 3-axis milling machines, so they are widely used in the mould manufacturing industry. With regard to 5-axis milling machines, the cutter or machine table can rotate around its x- and y-axes, while moving in the x, y and z directions. A significant benefit of 5-axis milling machines is that the set-up time is greatly shortened and these machines can make very complex surfaces that cannot be produced on 3-axis milling machines.

Three major types of end-milling tools, flat, ball, and bull-nose end-mills are commonly used in industry. Flat-end mills are the first choice for 2-D machining, which can cut flat planes and straight walls, such as contours, pockets, and faces. Bull-nose

end-mills are a good choice for roughing, and ball end-mills are used for machining surfaces and making half-circle slots. Bull-nose end-mills can also be used for surface machining.

The depth of cut in profile milling refers to the axial and radial depths of cut. The axial depth of cut is the thickness of removed material along the tool axis direction, while the radial depth of cut is the thickness that the tool immerses into the workpiece. In high-speed machining, these two depths of cutting should be properly determined, and, usually, are small so that the tool can endure the enormous cutting forces generated during machining. In CAD systems, complex curves and surfaces are usually designed with the NURBS, which is excellent to present free-form shapes and has been widely adopted in the CAD systems.

2.2 NURBS Definition

There are two types of curve representations: implicit and parametric. NURBS is represented in a parametric form as the following equation.

$$P(u)=\frac{P^{w}(u)}{w(u)}=\frac{\sum_{i=0}^{n} w_i^{(0)}\cdot N_{i,k}(u)\cdot C_i^{(0)}}{\sum_{i=0}^{n} w_i^{(0)}\cdot N_{i,k}(u)} \tag{2.1}$$

where $N_{i,k}(u)$ $(i=0,1,...n)$ is the basis functions, k is the degree, and u is the parameter. The non-periodic knot vector is defined as

$$T^{(0)}=[t_0^{(0)},t_1^{(0)},...,t_{n+k+1}^{(0)}]=[\underbrace{0,...,0}_{k+1},t_{k+1},...,t_n,\underbrace{1,...,1}_{k+1}] \tag{2.2}$$

and $n+1$ control points $C_i^{(0)}=[x_i,y_i]^T$ $(i=0,1,...n)$ with corresponding weights $w_i^{(0)}$. To find the geometric relationship between a NURBS profile and the depths of cut, it is

necessary to derive the first- and second-degree derivatives of NURBS profiles. B-spline curves are a special type of NURBS, and share most of the properties with the NURBS.

- **The 1st derivative of the nominal NURBS curve**

To get the first derivative of a nominal NURBS curve, a set of pseudo control points can be calculated as

$$C_i^{(1)} = \frac{k}{t_{i+k+1}^{(0)} - t_{i+1}^{(0)}} \cdot \left(w_{i+1}^{(0)} \cdot C_{i+1}^{(0)} - w_i^{(0)} \cdot C_i^{(0)} \right). \tag{2.3}$$

the corresponding weights are calculated by the following equation.

$$w_i^{(1)} = \frac{k}{t_{i+k+1}^{(0)} - t_{i+1}^{(0)}} \cdot \left(w_{i+1}^{(0)} - w_i^{(0)} \right) \tag{2.4}$$

So the first derivative of $P^w(u)$ is

$$\frac{dP^w(u)}{du} = \sum_{i=0}^{n-1} N_{i,k-1}(u) \cdot C_i^{(1)} \tag{2.5}$$

The first derivative of w(u) in term of u is $\frac{dw(u)}{du} = \sum_{i=0}^{n-1} N_{i,k-1}(u) \cdot w_i^{(1)}$, and the new knot vector is that,

$$T^{(1)} = [t_0^{(1)}, t_1^{(1)}, \ldots, t_{n+k+1}^{(1)}] = [\underbrace{0,\ldots,0}_{k}, t_{k+1}, \ldots, t_n, \underbrace{1,\ldots,1}_{k}] \tag{2.6}$$

So the first derivative of *P(u)* can be represented as

$$\frac{dP(u)}{du} = \frac{1}{w(u)} \cdot \left[\frac{dP^w(u)}{du} - \frac{dw(u)}{du} \cdot P(u) \right] \tag{2.7}$$

- **The second degree derivative of the nominal NURBS curve**

Another group of pseudo control points for the second derivative of $P^w(u)$ is calculated

by, $C_i^{(2)} = \frac{k-1}{t_{i+k}^{(1)} - t_{i+1}^{(1)}} \cdot \left(C_{i+1}^{(1)} - C_i^{(1)} \right) \quad (i = 0,1,\ldots,n-2)$ (2.8)

The weights are calculated as

$$w_i^{(2)} = \frac{k-1}{t_{i+k}^{(1)} - t_{i+1}^{(1)}} \cdot (w_{i+1}^{(1)} - w_i^{(1)}) \qquad (i = 0,1,...,n-2) \tag{2.9}$$

The second derivative of $P^w(u)$ is

$$\frac{d^2 P^w(u)}{du^2} = \sum_{i=0}^{n-2} N_{i,k-2}(u) \cdot C_i^{(2)}, \tag{2.10}$$

The second derivative of *w(u)* is

$$\frac{d^2 w(u)}{du^2} = \sum_{i=0}^{n-2} N_{i,k-2}(u) \cdot w_i^{(2)}, \tag{2.11}$$

the knot vector of that is,

$$T^{(2)} = [t_0^{(2)}, t_1^{(2)}, ..., t_{n+k+1}^{(2)}] = [\underbrace{0,...,0}_{k}, t_{k+1}, ..., t_n, \underbrace{1,...,1}_{k}]$$

Last the second derivative of *P(u)* is calculated by,

$$\frac{d^2 P(u)}{du^2} = \frac{1}{w(u)} \cdot \left[\frac{d^2 P^w(u)}{du^2} - 2 \cdot \frac{dw(u)}{du} \cdot P(u) - \frac{d^2 w(u)}{du^2} \cdot P(u) \right] \tag{2.12}$$

- **The unit tangent vector of the nominal NURBS curve**

$$T(u) = \begin{pmatrix} T_x(u) \\ T_y(u) \end{pmatrix} = \frac{1}{\sqrt{\left(\frac{dx(u)}{du}\right)^2 + \left(\frac{dy(u)}{du}\right)^2}} \cdot \begin{bmatrix} \frac{dx(u)}{du} \\ \frac{dy(u)}{du} \end{bmatrix} \tag{2.13}$$

- **Cutter location toolpaths and pre-finishing profiles**

Suppose a NURBS profile contour is to be cut from a rectangular workpiece shown in Fig. 2-1, the cutter location (cutter center) toolpath and pre-finishing profile are attained by offsetting the NURBS profile by a specified value.

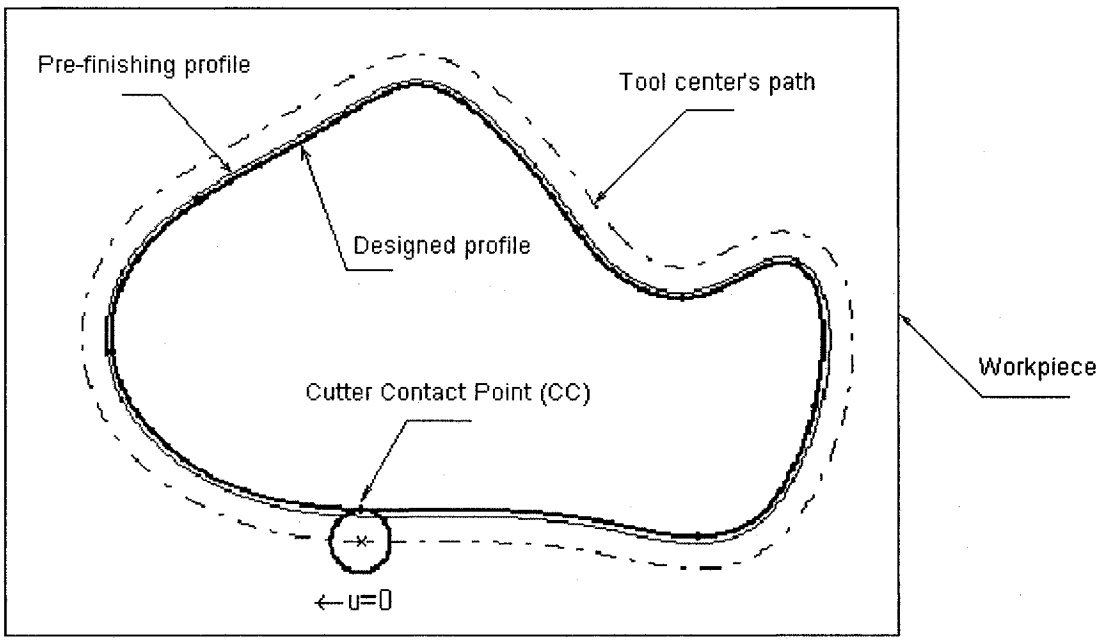


Fig.2-1 NURBS contour and toolpath

There are 4 cases to derivate equations of the pre-finishing and cutter location path, depending on the offset direction and interpolating along the curve direction.

Case 1: u varies clockwise from 0 to 1, and the offset direction is outward. Suppose that r is the tool radius and δ is the constant distance between the semi-finished and finished profiles.

$$P_{cl}(u) = P(u) + r \cdot \begin{pmatrix} -T_y(u) \\ T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) - r \cdot T_y(u) \\ y(u) + r \cdot T_x(u) \end{pmatrix} \tag{2.14}$$

$$P_{semi}(u) = P(u) + \delta \cdot \begin{pmatrix} -T_y(u) \\ T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) - \delta \cdot T_y(u) \\ y(u) + \delta \cdot T_x(u) \end{pmatrix} \tag{2.15}$$

Case 2: u varies clockwise from 0 to 1, and the offset direction is inward.

$$P_{cl}(u) = P(u) + r \cdot \begin{pmatrix} T_y(u) \\ -T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) + r \cdot T_y(u) \\ y(u) - r \cdot T_x(u) \end{pmatrix} \tag{2.16}$$

$$P_{semi}(u) = P(u) + \delta \cdot \begin{pmatrix} T_y(u) \\ -T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) + \delta \cdot T_y(u) \\ y(u) - \delta \cdot T_x(u) \end{pmatrix} \tag{2.17}$$

Case 3: u varies counter-clockwise from 0 to 1, and the offset direction is outward.

$$P_{cl}(u) = P(u) + r \cdot \begin{pmatrix} T_y(u) \\ -T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) + r \cdot T_y(u) \\ y(u) - r \cdot T_x(u) \end{pmatrix} \tag{2.18}$$

$$P_{semi}(u) = P(u) + \delta \cdot \begin{pmatrix} T_y(u) \\ -T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) + \delta \cdot T_y(u) \\ y(u) - \delta \cdot T_x(u) \end{pmatrix} \tag{2.19}$$

Case 4: u varies counter-clockwise from 0 to 1, and the offset direction is inward.

$$P_{cl}(u) = P(u) + r \cdot \begin{pmatrix} -T_y(u) \\ T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) - r \cdot T_y(u) \\ y(u) + r \cdot T_x(u) \end{pmatrix} \tag{2.20}$$

$$P_{semi}(u) = P(u) + \delta \cdot \begin{pmatrix} -T_y(u) \\ T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) - \delta \cdot T_y(u) \\ y(u) + \delta \cdot T_x(u) \end{pmatrix} \tag{2.21}$$

- **The first and second degree derivative of cutter location path**

$$P_{cl}(u) = P(u) + r \cdot \begin{pmatrix} -T_y(u) \\ T_x(u) \end{pmatrix} = \begin{pmatrix} x(u) - r \cdot T_y(u) \\ y(u) + r \cdot T_x(u) \end{pmatrix} \tag{2.22}$$

$$T_x = \frac{\frac{dx}{du}}{\sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2}} \qquad T_y = \frac{\frac{dy}{du}}{\sqrt{\left(\frac{dx}{du}\right)^2 + \left(\frac{dy}{du}\right)^2}} \tag{2.23}$$

The first derivative of the cutter location path is

$$\frac{dx_{cl}(u)}{du} = \frac{dx(u)}{du} - r \cdot \frac{dT_y(u)}{du} \tag{2.24}$$

$$\frac{dy_{cl}(u)}{du} = \frac{dy_{cl}(u)}{du} + r \cdot \frac{dT_x(u)}{du} \tag{2.25}$$

The second derivative of the cutter location path is

$$\frac{d^2x_{cl}(u)}{du^2} = \frac{d^2x(u)}{du^2} - r \cdot \frac{d^2T_y}{du^2} \tag{2.26}$$

$$\frac{d^2y_{cl}(u)}{du^2} = \frac{d^2y}{du^2} + r \cdot \frac{d^2T_x}{du^2} \tag{2.27}$$

2.3 Chip Load Model in NURBS Profile Milling

Generally, after the rough and semi-finish machining, a thin layer of material should be left for finish machining, and then tool moves along the profile with high spindle speeds and lower feed rates. This layer for finish machining can be assumed as a profile offset. Different from regular feature milling, where the effective depth of cut is constant, the effective depth of cut in NURBS profile milling changes at different positions. If the variation of DOC is too large, the surface finish could be worse, so it should be avoidable in high-speed machining.

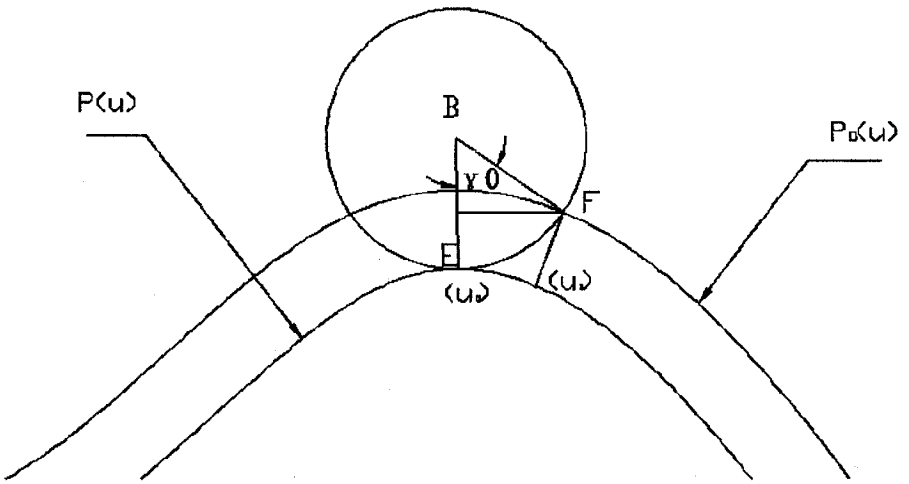


Fig.2-2 Relationship between engagement angle and effective depth of cut

Suppose a flat-end mill cuts along a piece of NURBS profile shown in Fig. 2-2. The tool radius is r, and the nominal radial depth of cutting is δ_0. Because of the NURBS shape, the effective radial depth of cut is not δ_0 anymore, which can be expressed as

$$\delta_e = r - r \times \cos\gamma_0 \tag{2.28}$$

and the pre-finish profile is an constant offset of the NURBS profile, and it could be expressed as

$$P_O(u) = P(u) + \delta_0 \begin{bmatrix} -T_y(u) \\ T_x(u) \end{bmatrix} \tag{2.29}$$

where $T_y(u)$ and $T_x(u)$ are the components of the unit tangent vector of the NURBS profile. To calculate tool-workpiece engagement angles, substituting Eq. (2.28) into Eq. (2.29), the following equation of the angle is obtained.

$$\left\| P_O(u_1) - P_B(u_0) \right\| = r \tag{2.30}$$

$$\gamma_0 = \cos^{-1}\left[1 - \frac{\|P_O(u_1) - P(u_0)\|}{2r^2}\right] \tag{2.31}$$

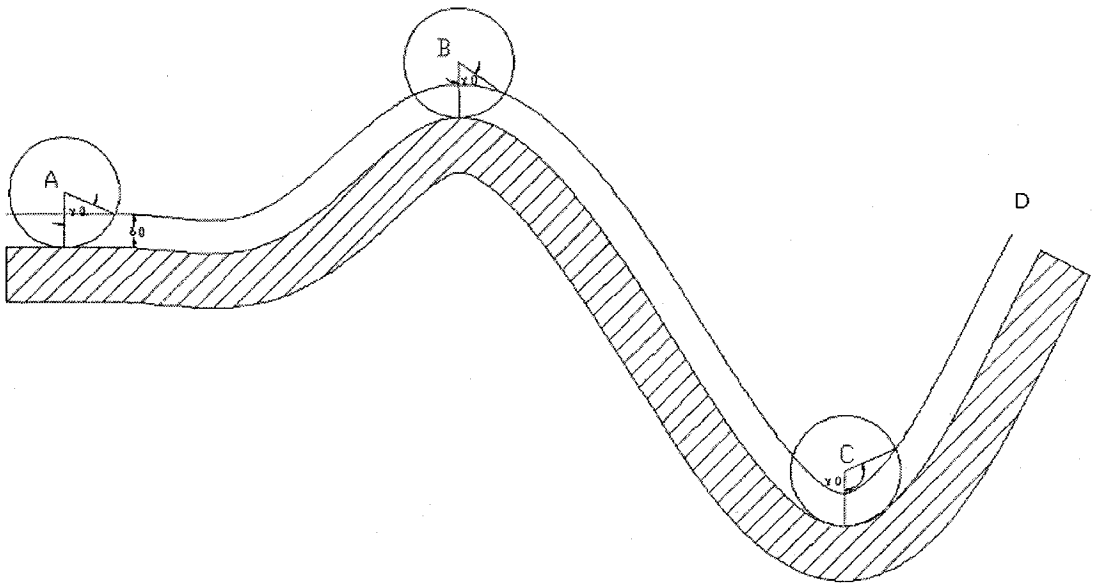


Fig.2-3 engagement angle along NURBS profile

The effective depth of cut can be gotten from Eq. (2.28) in accordance to the engagement angle. The effective radial depth of cut, feed rate, and axial depth of cut are the main parameters to determine resultant cutting forces. A practical approach to accommodating chip load variation in Fig. 2-4 shows the relationship between the engagement angles and the NURBS profile to be machined. From the simulating result, we can find that the maximum engagement angle is about 5 times to the minimum engagement angle, which will cause enormous difference at cutting force at the condition of same depth of cut. To keep the cutting force not fluctuate too much, the effective way is to bring down feed rate at concave profile, and bring up feed rate at convex profile.

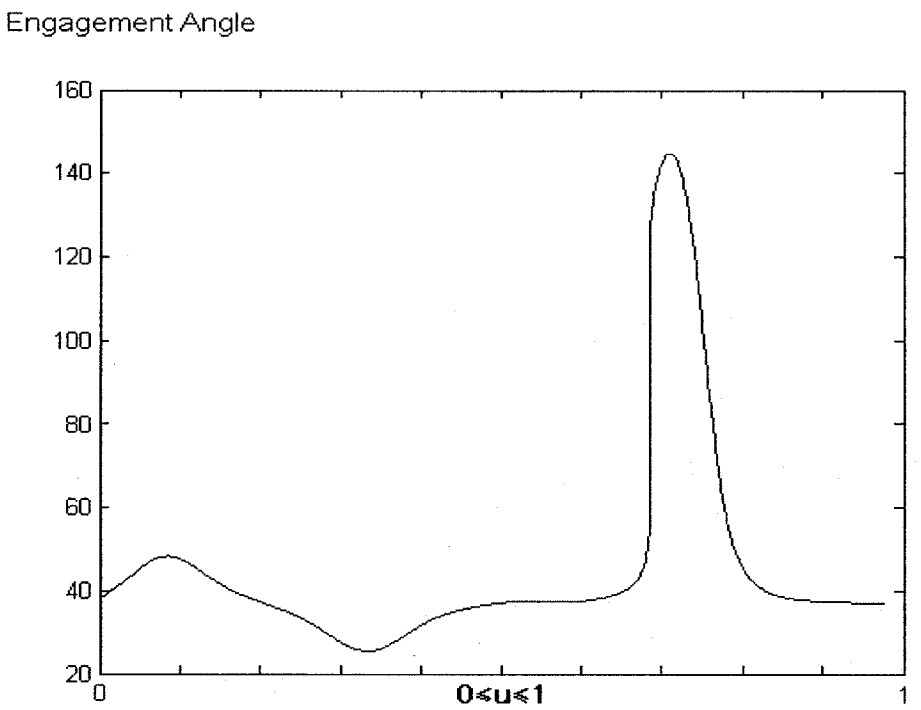


Fig.2-4 Quantitative relation between engagement angle and parameter

Chapter 3 Back-Propagation Multi-Layer Neural Network for Cutting Forces Prediction

In the literature review, some concepts and methods of predicting cutting forces in machining processes are introduced. Metal cutting process typically is non-linear with multi-variables, thus it is impossible to acquire an accurate model through the traditional mechanic analysis method. Because of their strong capability to deal with non-linear process, artificial neural networks are considered as an effective tool to predict cutting forces. There are three benefits by using artificial neural networks. The first one is that ANNs provide an accurate way to describe the highly non-linear metal cutting process. Basically, the prediction accuracy can be up to 98% by the artificial neural networks. The second is that this model is easy to modify by changing the network structure and different types of networks can be combined together for complex problems. The third benefit lies in the prospect of the artificial intelligence technology. Some kinds of neural networks have the capability of self-study and self-organizing; hence, researchers can work out more intelligent CNC machines for real-time applications.

3.1 Basics of Artificial Neural Networks

There is no unique definition of neural networks. According to Haykin, a neural network is a massively parallel-distributed processor that has a natural propensity for storing experiential knowledge and making it available for use [18]. Neural network is a network of many simple processors (neurons), which are connected by communication channels carrying numerical data and encoded by various means. Most neural networks have training rules, whereby the weights of connections are adjusted on the basis of data. In other words, neural networks can learn from examples and demonstrate strong capability for generalization of training data.

3.1.1 Neurons and Network Structures

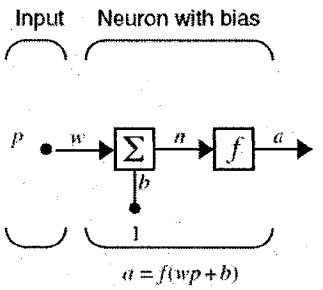


Fig.3-1 Structure of neuron

Neurons consist of a set of weighted input connections, which include a bias input, a state function, a non-linear transfer function, and output (see Fig. 3-1). An input connection refers to an input value that is received from a previous neuron. The bias is not connected to the other neurons in the network and is assumed to have an input value of 1 for the summation function. Weights, which represent strength or importance of an

input, connect to a neuron. Each neuron input, including the bias, has an associated weight. State function is a simple summation function. The output of the state function becomes the input for the transfer function. Transfer function, a nonlinear mathematical function, is used to convert data to a specific scale. There are two basic types of transfer functions: continuous and discrete. Commonly used continuous functions are ramp, Sigmoid, arc tangent and hyperbolic tangent (see Fig. 3-2). Continuous functions are sometimes called squashing functions. Discrete transfer functions are called activation functions, and commonly-used discrete functions are Step and Threshold.

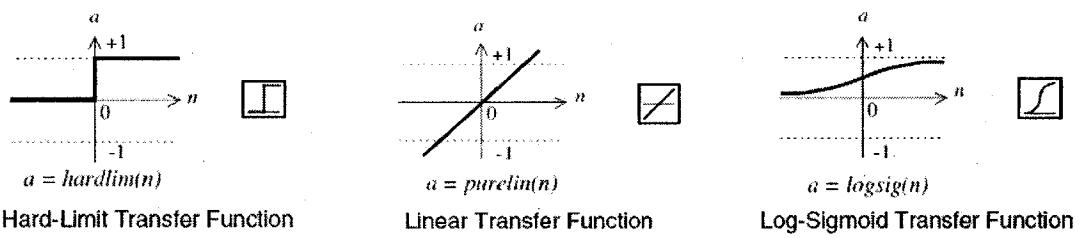


Fig.3-2 Typical transfer functions

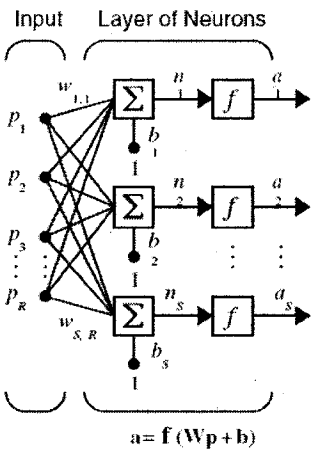


Fig.3-3 Single layer of neural network

If the input is a vector with dimension R and S neurons in a layer of the network (see Fig. 3-3), the output and weights are not scalar anymore. In this network, each element of the input vector p is connected to each neuron input through the weight matrix W. The i^{th} neuron has a summer that gathers its weighted inputs and bias to form its own scalar output n(i). The various n(i) taken together form an S-element net input vector n. Finally, the neuron layer outputs form a column vector a. A layer is not constrained to have the number of its inputs equal to the number of its neurons, which means the row number of weight matrix (W) are not necessary equal to its columns number.

A network can have several layers. Each layer has a weight matrix **W**, a bias vector **b**, and an output vector **a**. A three-layer network is shown below (see Fig. 3-4).

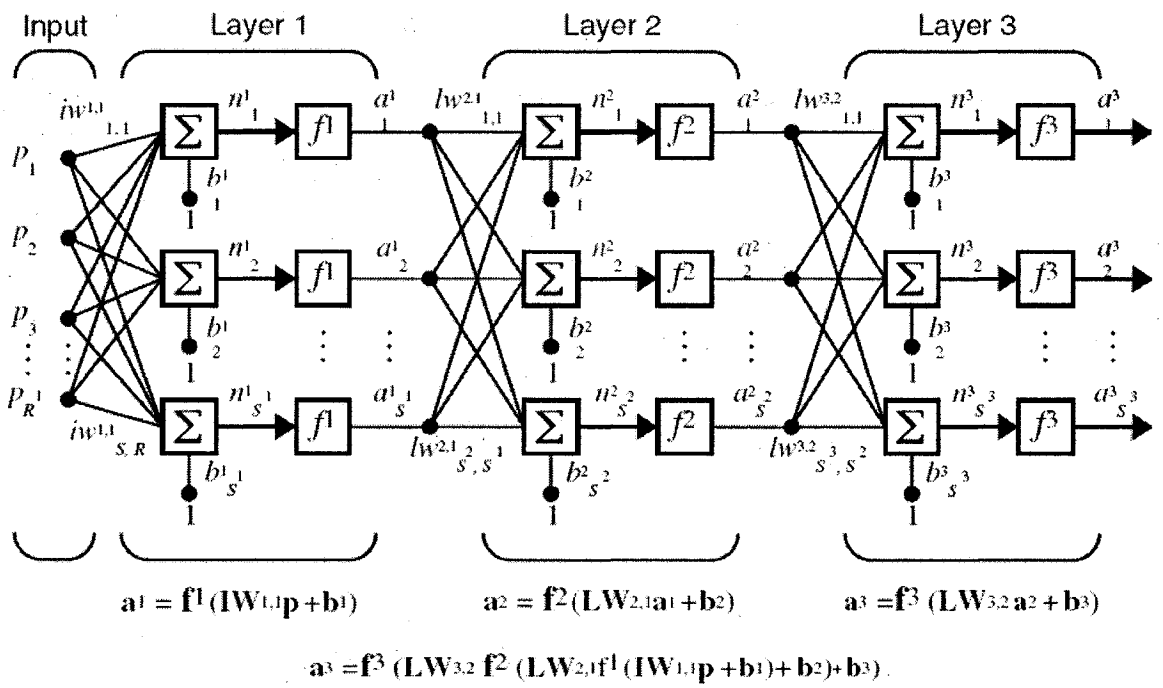


Fig. 3-4 Multi-layer of neural networks

The network shown above has R^1 inputs, S^1 neurons in the first layer, S^2 neurons in the second layer, etc. The outputs of each intermediate layer are the inputs to the following layer. For example, layer 2 can be analyzed as a one-layer network with S^1 inputs, S^2 neurons, and a $S^2 \times S^1$ weight matrix W^2. The layers of a multilayer network play different roles. A layer that produces the network output is called the output layer. All other layers are called hidden layers. Some researchers define the inputs as the first layer, which can bring different expressions to the network structure. Multiple-layer networks are quite powerful. For instance, a network of two layers shown above, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. This kind of two-layer network is extensively used in the feed forward back-propagation networks.

3.1.2 Back-propagation Training for Feed Forward Networks

A feed forward neural network is a network where the signal is passed from an input layer of neurons through a hidden layer to the output layer of neurons. The hidden layer is denoted as "hidden", because it contains neither input nor output data and the output of the hidden layer generally remains unknown to the user. The feed forward network with one hidden layer is one of the most popular neural networks, as we mentioned that this kind of network can simulate almost any non-linear process. Neural networks incorporate either supervised or unsupervised learning into the training. A network trained using supervised learning is presented with a target variable and fits a function that can be used to predict the target variable. This is analogous to the use of such statistical procedures as regression for prediction and classification. A network trained using unsupervised

learning does not have a target variable. The network finds characteristics in the data that can be used to group similar records together. This is analogous to cluster analysis in classical statistics. My thesis research focuses only on the supervised learning feed forward neural networks with one hidden layer.

The procedure for selecting the parameters for a given problem is called training the network. In this section, I will outline a training procedure of back propagation. Back propagation was created by generalizing the Widrow-Hoff learning rule to the multiple-layer networks. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with the specific output vectors. Standard back propagation is a gradient descent algorithm, as is the Widrow-Hoff learning rule, in which the network weights are moved along the negative of the gradient of the performance function. The equations that describe this operation are

$$a^{m+1} = f^{m+1}\left(w^{m+1}a^{m} + b^{m+1}\right), \text{ for } m=0, 1, \ldots, M-1 \tag{3.1}$$

where M is the number of layers in the network. The neurons in the first layer receive external inputs: $a^0 = p$, which provides the starting point for Eq. (3.1). The outputs of the neurons in the last layer are considered the network outputs: $a = a^m$.

The back propagation algorithm for the multi-layer networks is a gradient descent optimization procedure, in which we can minimize a mean square error performance index. The algorithm is provided with a set of examples of proper network behavior:

$$\{p_1,t_1\},\ \{p_2,t_2\},\cdots,\ \{p_Q,t_Q\} \tag{3.2}$$

where p_Q is an input to the network, and t_Q is the corresponding target output. As each input is applied to the network, the network output is compared to the target. The

algorithm should adjust the network parameters in order to minimize the sum-squared error

$$F(x)=\sum_{q=1}^{Q}e_q^2=\sum_{q=1}^{Q}(t_q-a_q)^2 \tag{3.3}$$

where x is a vector containing all network weights and biases. If the network has multiple outputs, Eq. (3.3) can be generalized as

$$F(x)=\sum_{q=1}^{Q}e_q^T e_q=\sum_{q=1}^{Q}(t_q-a_q)^T(t_q-a_q). \tag{3.4}$$

Replacing the sum-squared error by the error on the latest target,

$$\hat{F}(x)=(t(k)-a(k))^T(t(k)-a(k))=e^T(k)e(k) \tag{3.5}$$

The expectation of the squared error has been replaced by the squared error at iteration k. The steepest descent algorithm for the approximated mean square error is

$$w_{i,j}^m(k+1)=w_{i,j}^m(k)-\alpha\frac{\partial\hat{F}}{\partial w_{i,j}^m} \tag{3.6}$$

$$b_i^m(k+1)=b_i^m(k)-\alpha\frac{\partial\hat{F}}{\partial b_i^m} \tag{3.7}$$

where α is the learning rate.

For a single layer network, the partial derivatives of Eqs. (3.5) and (3.6) can be computed conveniently, since the error can be written as an explicit linear function of the network weights. For a multi-layer network, the error is not an explicit function of the weights in hidden layers. Because the error is an indirect function of the weights in the hidden layers, we will use the chain rule of calculus to calculate the derivatives in Eqs. (3.5) and (3.6):

$$\frac{\partial \hat{F}}{\partial w_{i,j}^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial w_{i,j}^m} \tag{3.8}$$

$$\frac{\partial \hat{F}}{\partial b_i^m} = \frac{\partial \hat{F}}{\partial n_i^m} \times \frac{\partial n_i^m}{\partial b_i^m} \tag{3.9}$$

The second terms of these equations can be easily computed, since the net input to layer *m* is an explicit function of the weights and bias in that layer:

$$n_i^m = \sum_{j=1}^{s^{m-1}} w_{i,j}^m a_j^{m-1} + b_i^m \tag{3.10}$$

Therefore,

$$\frac{\partial n_i^m}{\partial w_{i,j}^m} = a_j^{m-1}, \frac{\partial n_i^m}{\partial b_i^m} = 1. \tag{3.11}$$

If defined $s_i^m = \frac{\partial \hat{F}}{\partial n_i^m}$.

Now, we can express the approximated steepest descent algorithm as

$$w_{i,j}^m(k+1) = w_{i,j}^m(k) - \alpha s_i^m a_j^{m-1} \tag{3.12}$$

$$b_i^m(k+1) = b_i^m(k) - \alpha s_i^m \tag{3.13}$$

These can be expressed in a matrix form as

$$W^m(k+1) = W^m(k) - \alpha s^m (a^{m-1})^T \tag{3.14}$$

$$b^m(k+1) = b^m(k) - \alpha s^m \tag{3.15}$$

3.2 Machining Parameters Database

Since the metal cutting process is too complicated to be described with theoretical equations, the empirical method is being considered as an effective, practical way to model the machining process. There are more than 20 parameters that can affect the resultant cutting forces in the milling process; however, they can be simplified to 4 or 5 parameters that are dominant in the whole process. Using an artificial neural network to model metal cutting forces is considered as an effective way to describe the highly non-linear metal cutting process, and many researchers have acquired valuable results by establishing various network models.

As the first step to establish a neural network, a typical range of machining parameters and experimental data are need to be identified to as training and testing data for the neural network. To build this database, generally, an experiment is designed under a given condition, with several inputs and outputs, such as feed rate, spindle speed, depth of cut, and the outputs can be the resultant force or 3 components of the resultant cutting force. Under these specified experimental conditions, cutting forces are measured and recorded by a Kistler dynamometer. Another method is to get experimental data from authoritative engineering databases. Here, based on a popular database, CutData, we set up our experiment conditions in this database, and then transform the result to what we need for training and testing the neural network. In my experiments with this database, the fixed condition, including the workpiece material of Aluminum 6061, 2 flutes, and 10 mm diameter of the carbide tool, is selected, and the spindle speed is set as 20,000 rpm.

- Radial depth of cut (ROC)

ROC represents the immersion of the tool in the workpiece along the radial (Y-axis) direction. It can vary from 0% to 100% of the tool diameter. In this experiment, ROC is designed to change from 10% to 100% of the diameter. Six groups of values are recorded in the experiment.

r1: 10% of the tool diameter

r2: 20% of the tool diameter

r3: 30% of the tool diameter

r4: 40% of the tool diameter

r5: 50% of the tool diameter

r6: 100% of the tool diameter

- Feed rate

Four groups of values are specified in the experiment.

f1: 3000 mm/min

f2: 6000 mm/min

f3: 9000 mm/min

f4: 12000 mm/min

- Axial depth of cut (AOC)

AOC signifies the immersion of the tool in the axial (Z-axis) direction. It is an important factor to affect the machining errors, tool deflections. It is set from 1 to 5 mm. Under this condition, 120 groups of experiments are conducted, and the experiment data (shown in Table 1) are recorded for training the artificial neural networks. Furthermore, some other experiment data are needed to test the accuracy of the neural network parameters. Testing data should not be over the range of training data, and they should

be randomly distributed in the range to ensure the validation quality. The testing data are recorded in Table 2.

Table 1. Training data of network (spindle speed 20,000 rpm)

Axial Depth of Cut is 1 mm					
Cutting Force (N)		Feed rate (mm/min)			
		3000	6000	9000	12000
δ_e	0.6	4.356	8.762	13.143	17.53
	1.2	9.801	17.524	26.286	35.048
	1.8	13.144	26.287	39.426	52.578
	2.4	17.543	35.048	52.589	70.077
	3.0	21.893	43.809	65.711	87.558
	3.6	26.306	52.589	78.795	105.113
Axial Depth of Cut is 2 mm					
Cutting Force (N)		Feed rate (mm/min)			
		3000	6000	9000	12000
δ_e	0.1	8.724	17.537	26.299	35.06
	0.2	17.524	35.064	52.573	70.096
	0.3	26.274	52.653	78.844	105.129
	0.4	35.049	70.096	105.142	140.188
	0.5	43.809	87.496	131.493	175.252
	1.0	52.657	105.156	157.686	210.281
Axial Depth of Cut is 3 mm					
Cutting Force (N)		Feed rate (mm/min)			
		3000	6000	9000	12000
δ_e	0.1	13.145	26.279	39.429	52.543
	0.2	26.27	52.523	78.833	105.162
	0.3	39.315	78.839	118.261	157.745
	0.4	52.525	105.164	157.803	210.303
	0.5	65.717	131.42	197.145	263.013
	1.0	78.874	157.721	236.522	315.49
Axial Depth of Cut is 4 mm					
Cutting Force (N)		Feed rate (mm/min)			
		3000	6000	9000	12000
δ_e	0.1	17.524	35.048	52.572	70.096
	0.2	35.051	70.10	105.135	140.192
	0.3	52.576	105.156	157.726	210.29
	0.4	70.098	140.192	210.288	280.231
	0.5	87.62	175.236	262.853	350.55
	1.0	105.14	210.312	314.827	420.614
Axial Depth of Cut is 5 mm					
Cutting force (N)		Feed rate (mm/min)			
		3000	6000	9000	12000
δ_e	0.1	21.893	43.811	65.705	87.551
	0.2	49.007	87.621	131.441	175.146
	0.3	65.715	131.432	197.114	262.9
	0.4	87.619	175.145	262.861	350.492
	0.5	109.465	219.014	328.641	438.129
	1.0	131.444	262.869	394.221	525.685

Table2. Testing data for neural network (spindle speed 20,000 rpm)

Axial depth (mm)	Radial depth (mm)	Feed rate (mm/min)	Cutting force (N)
2.7426	1.4189	5593.7	52.987
2.9867	2.5991	11869	224.249
2.1594	2.6066	5368.9	73.521
3.1713	1.0139	9782.7	67.087
4.0151	0.94255	8936.8	82.411
4.7273	2.8219	8419.1	273.294
4.5797	3.5502	8444.4	334.161
3.0099	1.704	8935.5	111.532
1.2678	3.2501	4650.3	46.756
4.3673	1.6238	8728.9	150.761
2.567	2.8963	4532.8	81.969
3.6633	2.9839	7856.4	209.106
4.8068	1.4181	9173	152.156
2.767	2.5852	10891	189.612
1.4599	2.0491	3116	22.645
4.2898	1.6589	5793.6	100.369
1.6788	1.6031	10012	65.599
1.7833	1.3187	9108.5	52.146
2.4915	1.0891	3668.9	24.236
2.1722	3.0666	3636	58.923
2.8123	2.7341	3107.4	58.156
3.7728	2.1462	9328.8	183.863
3.9556	2.9383	8242.4	233.178
4.0748	3.5175	7582.9	264.547
3.35	0.79887	3668.6	23.893
2.7426	1.4189	5593.7	52.987
2.9867	2.5991	11869	224.249
2.1594	2.6066	5368.9	73.521

3.3 Neural Network System for Predicting Cutting Forces

3.3.1 Structure of the Artificial Neural Networks

In my thesis work, a 3-layer feed-forward neural network is established to simulate cutting forces in the milling process. A resultant cutting force is the output of the neural network, while the feed rate, radial depth of cut, and axial depth of cut are three inputs of this network. Since there is only one output in this network, there should be only a neutron in the output layer. The neutron numbers of other two layers are determined by comparing the fitness of training and testing results. Exhaustive experiments are conducted till finding an appropriate network structure, where both training errors and testing errors are within the specified tolerance. The transfer function is Sigmoid, which can have any value between plus and minus infinity, and squashes the output into the range 0 to 1. This transfer function is commonly used in the back propagation networks, in part because it is differentiable.

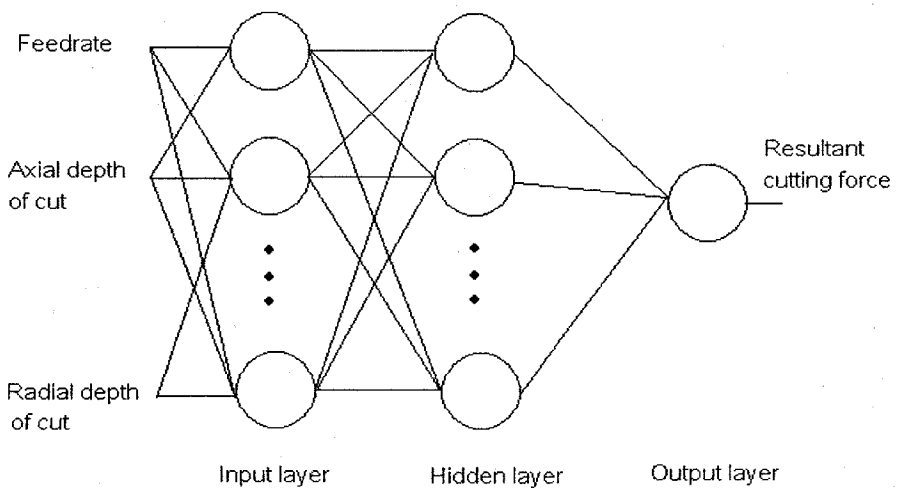


Fig.3-5 Structure of the ANN in cutting force prediction.

3.3.2 Back-Propagation Training

The back-propagation training algorithm is an iterative gradient algorithm designed to minimize the mean square error between the actual output of the hidden layers and the desired output. Before the ANN can be trained and the mapping learnt, it is important to process the experimental data into patterns. Training/testing pattern vectors are formed. Each pattern is formed with an input condition vector P_i.

$$P_i = \begin{pmatrix} Feedrate \\ AxialCutDepth \\ RadialCutDepth \end{pmatrix}$$

The corresponding target vector is a resultant cutting force $T_i = [CuttingForce]$

Since the training data are only representatives of the whole parameter space, it is necessary to normalize the experiment data to represent the whole parameter space. The normalizing equation is

$$P = (P_R - P_{\min})\frac{P_{N\max} - P_{N\min}}{P_{\max} - P_{\min}} + P_{N\min} \tag{3.16}$$

Here P_R is the real value of the variable before normalization. P_{min} and P_{max} are the minimum and maximum values of the variable P. They are normalized to values P_{Nmin}, P_{Nmax} such that $0<P_{N\,min}<P_{N\,max}<1$. For example, feed rate is varied from 3000 to 12000 mm/min in our experiments. Thus, P_{min}=3000 and P_{max} =12000. Then we choose P_{Nmin} to be 0.1 and P_{Nmax} to be 1. In this way, the range 3000 to 12000 is now mapped to 0.1 to 1. So when P_R=6000 mm/min, P= 0.4.

The simplest implementation of back-propagation updates the network weights and biases in the direction in which the performance function decreases most rapidly. As we

introduces at Section 3.1, a momentum factor is built into this evaluation to avoid the problem of being caught in a local optima.

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) + \alpha \cdot \frac{\partial \hat{F}}{\partial w_{i,j}^{m}} + \eta \cdot w_{i,j}^{m}(k) \tag{3.17}$$

$$b_{i}^{m}(k+1) = b_{i}^{m}(k) + \alpha \cdot \frac{\partial \hat{F}}{\partial b_{i}^{m}} + \eta \cdot b_{i}^{m}(k) \tag{3.18}$$

where α is the learning rate and η is the momentum coefficient.

A variation of the standard back-propagation technique for adjusting the network weights is the Levenberg–Marquardt technique, which is used in my thesis to train the network. This algorithm is similar to the quasi-Newton method, an approach of second-order training speed without much additional computational costs. It can be written as

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - H(n_{ki}, t_{ki}) g_{ki} \tag{3.19}$$

where H is the Hessian matrix of the mean square error (performance function), g_{ki} is the gradient with respect to current weights and biases. In the Levenberg–Marquardt technique, the Hessian matrix (second derivatives) **H** can be approximated as $H = J^T J$ (3.19), and the gradient may be computed as

$$g = J^T e \tag{3.20}$$

where **J** is the Jacobian matrix containing the first derivatives of the network errors with respect to network weights and e is the error for the *i*th pattern.

$$w_{i,j}^{m}(k+1) = w_{i,j}^{m}(k) - \left[J^T J + \mu \cdot I \right]^{-1} \cdot J^T \cdot e \tag{3.21}$$

If the scalar μ is zero, the Levenberg–Marquardt method is Newton's method using the approximate Hessian matrix. If μ is large, this changes to be a gradient descent with

a small step size. Since Newton's method is faster and more accurate near an error minimum, the aim is to shift towards Newton's method as quickly as possible.

The results of training and testing conducted on the 120 experimental data sets are presented. The errors in simulation are small (<0.0005) for both training and validating recall. Fig. 3-6 shows the simulation of cutting forces under feed rate is 6000 mm/min, and Fig. 3-7 shows the cutting forces under the condition that the axial depth of cut set to 3 mm.

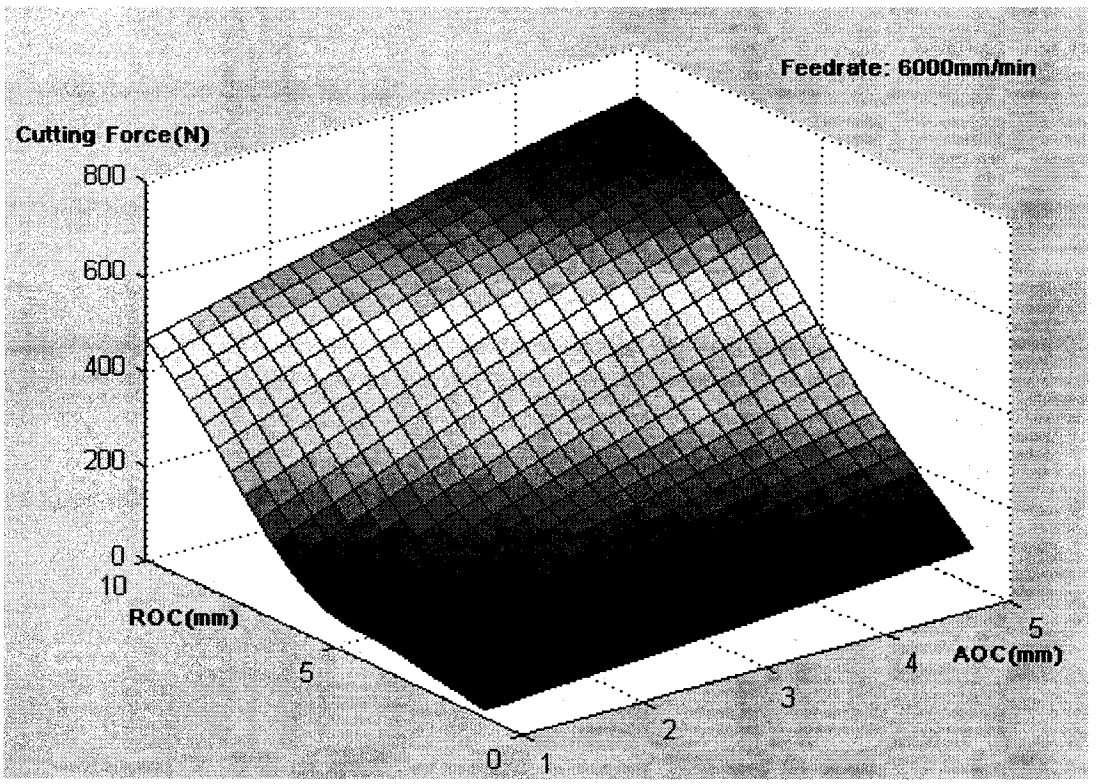


Fig. 3-6 Cutting force simulation at constant feed rate

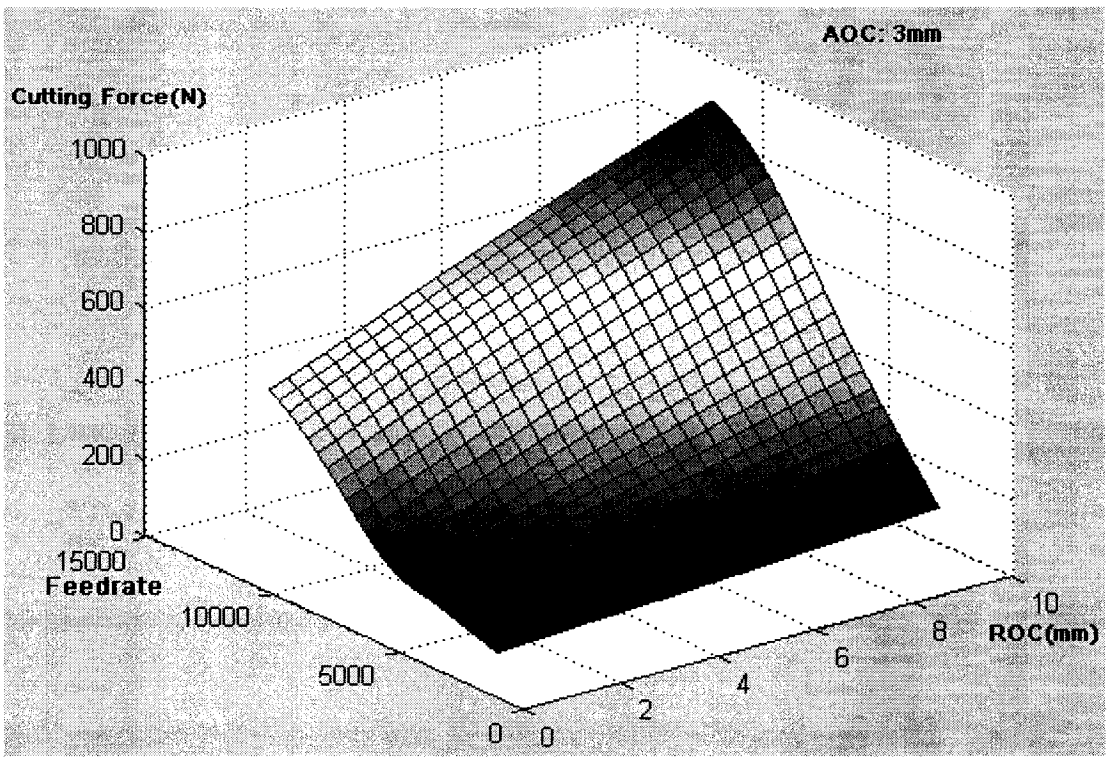


Fig.3-7 cutting force simulation at constant axial depth of cut

Chapter 4 Off-Line Fitting of NURBS Toolpaths

If cutting forces could not be accurately predicted, NC programmers usually define the machining parameters under the worst condition to ensure not to break the tool or damage the surfaces of parts; even though this could decrease the production efficiency. The multi-layer neural network model gives us a way to accurately predict the resultant cutting force, so the machining parameters, such as feed rate, could be calculated from this neural network model to satisfy the machining condition that the resultant cutting force does not greatly fluctuate. The adaptation of feed rate helps increase the productivity and reduce the machining error, especially in high-speed machining. The previous NURBS toolpath could be dissected into several parts according to the feed rate. To ensure the smoothness of motion, each part is approximated by a new NURBS equation, and then the curve parameters are input into the CNC controller to interpolate the tool motion fitting for the NURBS toolpaths.

There are two ways to fit a given set of points into NURBS curve. The first method is interpolation and second is approximation. In the case of interpolation, curve is passed through the given set of points. In the case of approximation the curve constructed may not necessarily satisfy the given data but only approximates it. If one tries to fit NURBS curve using interpolation, the resultant number of control points will be equal to the given

data points. Therefore interpolation is not a desirable way to fit for large number of data points. As to approximation method, the continuity between two neighbor approximating curve segments is requisite for high-speed machining. In this thesis an improved approximation method with end derivative constraint is used to fit points.

4.1 Global Curve Interpolation

Suppose to interpolate a set of points $\{Q_1, Q_2, ...Q_n\}$ with p-th degree nonrational B-spline curve. If we specify a value u_k to each control points Q_k, and assign an appropriate knot vector $U = \{u_1, u_2, ...u_n\}$, we can set up a $n \times n$ system of linear equations.

$$Q_k = C(u_k) = \sum_{k=1}^{n} N_{k,p}(u_k) P_k \tag{4.1}$$

Correspondingly, get the solutions for the coordinates of P_i. In this interpolation process, the most important is to choose appropriate knot vector and u_k, since these parameters can affect the shape of curves. Generally, three common methods are used to choose u_k.

- Equally spaced:

$$u_0 = 0,\ u_n = 1, \text{ and } u_k = \frac{k}{n}, \quad k = 1, ...n-1 \tag{4.2}$$

this method is not recommended, because the points are seldom distributed evenly, and this method could produce erratic shape.

- Chord length:

Suppose d to be the total chord length

$$d = \sum_{k=1}^{n} |Q_k - Q_{k-1}|, \text{ then } u_1 = 0 \quad u_n = 1 \tag{4.3}$$

$$u_k = u_{k-1} + \frac{|Q_k - Q_{k-1}|}{d} \qquad k = 2,...,n \tag{4.4}$$

This method is widely used, since it is generally adequate and approximates a uniform parameterization.

- Centripetal method:

Suppose

$$d = \sum_{k=1}^{n} \sqrt{|Q_k - Q_{k-1}|} \tag{4.5}$$

Then $u_1 = 1 \qquad u_n = 1$

$$u_k = u_{k-1} + \frac{\sqrt{|Q_k - Q_{k-1}|}}{d} \qquad k = 2,...n \tag{4.6}$$

this method can give a better result then chord length method, especially in the case of sharp turns. The knot can be calculated by this equation,

$$u_0 = ...u_p = 0 \tag{4.7}$$

$$u_{j+p} = \frac{1}{p} \sum_{i=j}^{j+p-1} u_i \tag{4.8}$$

With this equation the knots reflect the distribution of u_k. Using the equation with equation to compute the u_k leads to a system, it can be solved by Gaussian elimination method.

4.2 Least Squares Curve Approximation

Approximation is a useful curve fitting technique; however, it is more difficult than interpolation. In interpolation, the number of control points is equal to that of

interpolating points, and there is no need to check fitting errors. In approximation, we do not know the number of control in advance, thus, the approximation method is generally iterative in order to find appropriate control points and fit the curve with a desired accuracy.

There are a couple of methods to solve the approximation problem. For example, a non-linear optimization can solve the problems that control points, knot vector, parameters u_k, and even weights are unknown. However, this method can be very expensive. Generally, the approximation can be simplified as linear problems if the fixed number of control points is the only unknown, and the linear least squares techniques are good enough to solve this kind of problem.

Assume that a group of points, $Q_1,...Q_m$, are given, and we want to use n control points (m>n), and a p^{th} degree non-rational curve to fit this group of points,

$$C(u) = \sum_{i=1}^{n} N_{i,p}(u) P_i \,,\; u \in [0,1]$$

Satisfying that $Q_1 = C(0)$, $Q_m = C(1)$. We can represent the objective function by the least square of difference between the fitting curve and data points,

$$\sum_{k=2}^{m-1} |Q_k - C(u_k)|^2 \tag{4.9}$$

which is the minimum with respect to n variables P_i, and u_k can be calculated by Eq.4.10

Let

$$R_k = Q_k - N_{1,p}(u_k) Q_1 - N_{n,p}(u_k) Q_m \qquad k = 2,...m-1 \tag{4.10}$$

Set

$$|Q_k - C(u_k)|^2 = \sum_{k=2}^{m-1} \left| R_k - \sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right|^2$$

$$f = \sum_{k=2}^{m-1} = \sum_{k=2}^{m-1} \left(R_k - \sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right) \left(R_k - \sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right) \tag{4.11}$$

$$= \sum_{k=2}^{m-1} \left[R_k \cdot R_k - 2 \sum_{i=2}^{n-1} N_{i,p}(u_k)(R_k \cdot P_i) + \left(\sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right) \cdot \sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right]$$

where f is a scalar-valued function of n-2 variables $P_2,...P_{n-1}$. Then, we set the derivatives of f with n-2 points to zero. The l^{th} derivative is

$$\frac{\partial f}{\partial P_l} = \sum_{k=2}^{m-1} \left(-2N_{l,p}(u_k) R_k + 2N_{l,p}(u_k) \sum_{i=2}^{n-1} N_{i,p}(u_k) P_i \right) = 0 \tag{4.12}$$

which can be represented as

$$-\sum_{k=2}^{m-1} N_{l,p}(u_k) R_k + \sum_{k=2}^{m-1} \sum_{i=2}^{n-1} N_{l,p}(u_k) N_{i,p}(u_k) P_i = 0 \tag{4.13}$$

then

$$\sum_{i=2}^{n-1} \left(\sum_{k=1}^{m-1} N_{l,p}(u_k) N_{i,p}(u_k) \right) P_i = \sum_{k=2}^{m-1} N_{l,p}(u_k) R_k \tag{4.14}$$

This equation is one linear equation in the unknowns points, $P_2,...,P_{n-1}$. If $l = 2,...,n-1$, it yields n-2 equations in n-2 unknowns points.

$$(N^T N) P = R \tag{4.15}$$

$$N = \begin{bmatrix} N_{2,p}(u_k) & \cdots & N_{n-1,p}(u_k) \\ \vdots & \ddots & \vdots \\ N_{2,p}(u_{m-1}) & \cdots & N_{n-1,p}(u_{m-1}) \end{bmatrix} \tag{4.16}$$

$$R = \begin{bmatrix} N_{2,p}(u_1) R_2 + \cdots + N_{2,p}(u_{m-1}) R_{m-1} \\ \vdots \\ N_{n-1,p}(u_1) R_2 + \cdots + N_{n-1,p}(u_{m-1}) R_{m-1} \end{bmatrix} \tag{4.17}$$

The knot vector and parameter u_k are computed by Eq. (4.4). If the fitting error is larger than specified value, we can increase the number of control point, then repeat the algorithm to find new control points.

4.3 Global Approximation with End Derivatives Specified

In engineering application, a large number of data points can be approximated by several complex curves, because different part of profile may not have same machining parameters. The machine tool moves along the complex curves to cut material. The connections between these complex curves should be smooth and continuous in 1st derivative at least, otherwise gouging or undercutting may occur because of inertia of tool's motion. It means that the complex curves are not only to be approximated in a specified accuracy, but also in specified end derivatives. Some NC controllers begin to support spline interpolation that enables the programmer to input a series of control points and fit a smooth curve. By using this function, machine dynamic response is improved and leads to smoother tool movements and improved machining accuracy. Until now, there are no machines can support any form of complex curves. Most of them support kinds of specified form of curves. For example, the HEIDENHAIN controller supports two types of interpolation, the Bezier and cubic splines. In this thesis, we suppose to approximate points by cubic splines with the constraints of 2nd derivatives continuity.

Given a set of points, $Q_r, r = 0, \cdots, m$, and we want to approximate it with b-spline curve, $C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i$. We define $p = 3, n = 4$, to represent cubic b-spline in according to the format in HEIDENHAIN control. $D_s^{(1)}, D_s^{(2)}$, are 1st and 2nd derivatives at the start point; $D_e^{(1)}, D_e^{(2)}$ are 1st and 2nd derivatives at the end point.

Assume the parametrisation $t_{0,\ldots} t_m$ as well as the knot vector $U = (u_0, \ldots u_{n+p+1})$ has been computed. So,

$$Q_0 = C(t_0) = P_0 \tag{4.18}$$

$$Q_m = C(t_m) = P_n \tag{4.19}$$

$$C^{(i)}(t_0) = D_s^{(i)}, i = 1,2 \tag{4.20}$$

$$C^{(j)}(t_m) = D_e^{(j)} \quad j = 1,2 \tag{4.21}$$

According to Piegl's algorithm, the problem can be solved by two steps, the first is to compute the end control points based on the derivatives, and the second is to compute the rest of control points with least square minimization.

The derivatives of curve is represented by $C^{(j)}(u) = \sum_{i=0}^{n} N_{i,p}^{(j)}(u) P_i$, where

$$N_{i,p}^{(j)}(u) = p\left(\frac{N_{i,p-1}^{(j-1)}(u)}{u_{i+p} - u_i} - \frac{N_{i+1,p-1}^{(j-1)}(u)}{u_{i+p+1} - u_{i+1}} \right) \tag{4.22}$$

Therefore the algorithm to compute the end control points is as follows:

$P_0 = Q_0$

$$P_i = \frac{1}{N_{i,p}^{(i)}(t_0)} \left[D_s^{(i)} - \sum_{h=0}^{i-1} N_{h,p}^{(i)}(t_0) P_h \right], \quad i = 1,2 \tag{4.23}$$

$$P_n = Q_m$$

$$P_{n-j} = \frac{1}{N^{(j)}_{n-j,p}(t_0)}\left[D^j_e - \sum_{h=0}^{j-1} N^{(j)}_{n-h,p}(t_m) P_{n-h} \right], \qquad j = 1,2 \quad (4.24)$$

The rest of control points $P_{k+1}, \ldots P_{n-k-1}$ are computed by least-squares minimization.

$$f(P_{k+1}, P_{n-k-1}) = \sum_{r=1}^{m-1} \left| Q_r - C(t_r) \right|^2 \quad (4.25)$$

To solve the minimization problem, we write

$$Q_r - C(t_r) = R_r - \sum_{i=k+1}^{n-k-1} N_{i,p}(t_r) P_i ,$$

where

$$R_r = Q_r - \sum_{i=0}^{k} N_{i,p}(t_r) P_i - \sum_{j=0}^{k} N_{n-j,p}(t_r) P_{n-j} \quad (4.26)$$

So,

$$f(P_{k+1}, P_{n-k-1}) = \sum_{r=1}^{m-1} \left| R_r - \sum_{i=k+1}^{n-k-1} N_{i,p}(t_r) P_i \right|^2 \quad (4.27)$$

By the standard least-squares derivation, the equation to be solved is

$$\sum_{i=k+1}^{n-k-1} \left(\sum_{r=1}^{m-1} N_{i,p}(t_r) N_{j,p}(t_r) \right) P_i = \sum_{r=1}^{m-1} N_{j,p}(t_r) R_r , \quad j = k+1, \ldots n-k-1 \quad (4.28)$$

The equation can be represented by matrix form,

$$(N^T N) P = R$$

$$P = (N^T N)^{-1} R \quad (4.29)$$

where

$$N = \begin{bmatrix} N_{k+1}(t_1) & \cdots & N_{n-k-1}(t_1) \\ \vdots & \ddots & \vdots \\ N_{k+1}(t_{m-1}) & \cdots & N_{n-k-1}(t_{m-1}) \end{bmatrix} \tag{4.30}$$

$$R = \begin{bmatrix} N_{k+1}(t_1)R_1 + \cdots N_{k+1}(t_{m-1})R_{m-1} \\ \\ N_{n-k-1}(t_1)R_1 + \cdots N_{n-k-11}(t_{m-1})R_{m-1} \end{bmatrix} \tag{4.31}$$

Chapter 5 Applications of NURBS Toolpath in HSM

5.1 Conventional Methods in CAM System

Today, computer-aided manufacturing (CAM) systems have always been an effective way to provide toolpaths and NC codes for the whole manufacture process; however, they are short of optimization ways to solve problems of some special machining jobs, such as machining complex curve profiles or pockets. For example, an engineer designs appropriate shapes for a part in the CATIA system, and then transfers the file of this part to a manufacturing engineer or a NC programmer.

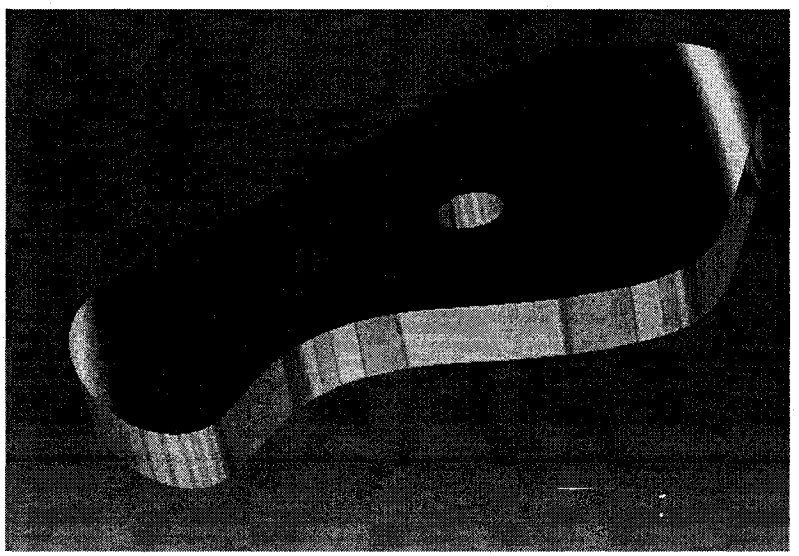


Fig. 5-1 Design of a part in the CATIA system

Generally, this manufacture engineer imports the CATIA file into a CAM system, and then toolpaths and NC codes are generated according the manufacturing procedure.

(1) Import the CATIA file into a CAM system (Fig. 5-2)

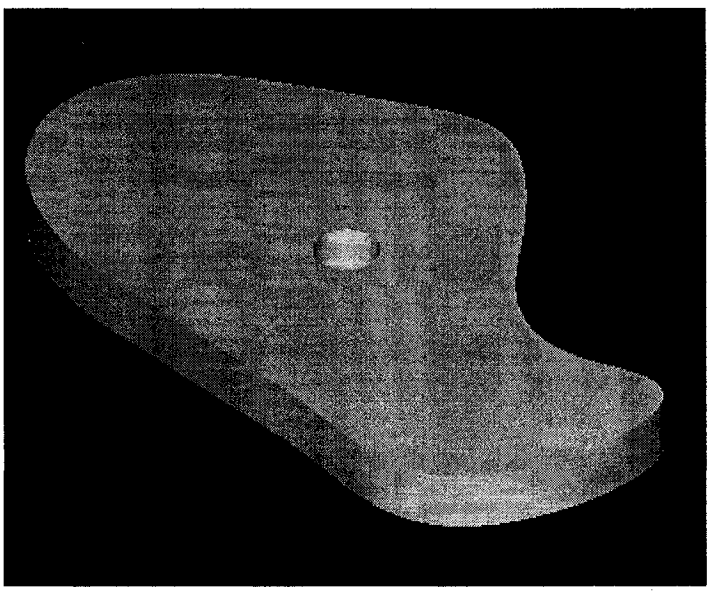


Fig. 5-2 Imported solid model in the MasterCAM system

(2) Set up a stock material

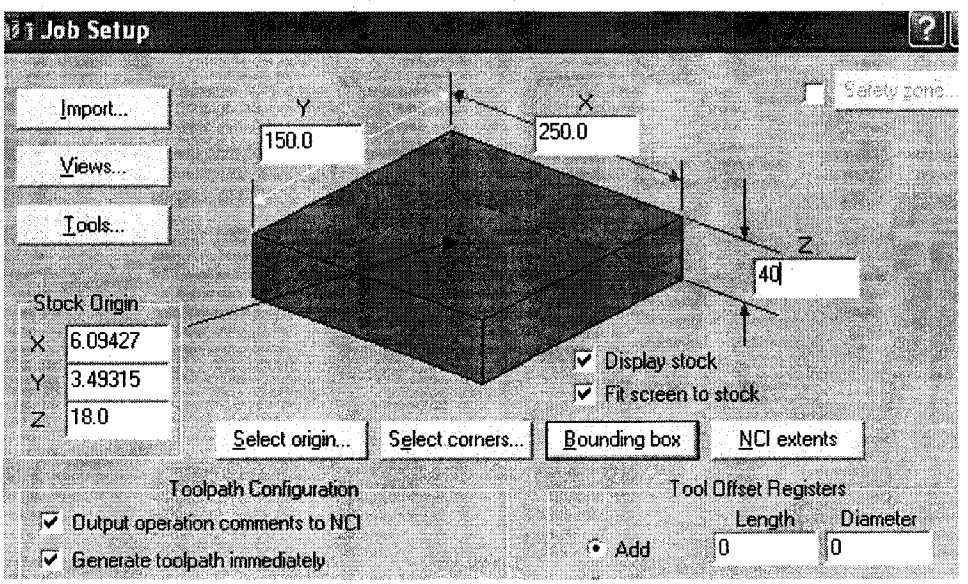


Fig. 5-3 MasterCAM menu for the job set-up

Consider the piece of material is 250 mm long, 150 mm wide, and 40 mm deep (Fig.5-3).

(3) Machining stage

To machine this part, there are two stages. The fist stage is rough machining, and the second stage is finish machining. In rough machining stage, a larger size of tool is chosen to ensure high machining efficiency. In this case, a 20 mm diameter flat end-mill is used for rough machining. In finish machining stage, the accuracy and surface quality is the most important thing, so a smaller tool is used for high accuracy and better surface quality. Here, a 8 mm flat end-mill is selected for finishing. The CAM will generate toolpaths for rough and finish machining, as shown in Fig. 5-4, and the simulation result is also shown in Fig. 5-5.

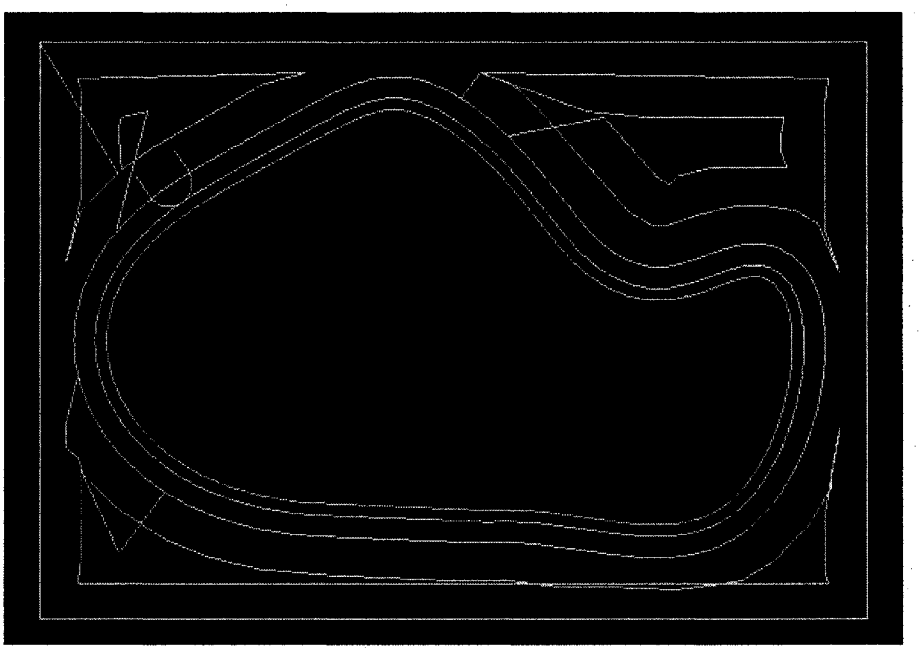


Fig. 5-4 Toolpaths for roughing and finishing of the part

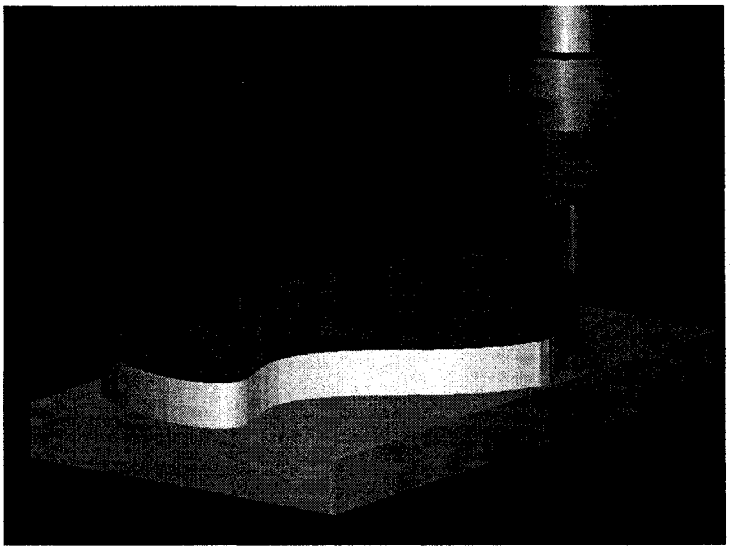


Fig. 5-5 Simulation of the cutting process

Without an additional means, the CAM system will use many small segments and arcs to approximate the complex curve, which may result rough quality surface in some areas and a larger NC program (Fig. 5-6).

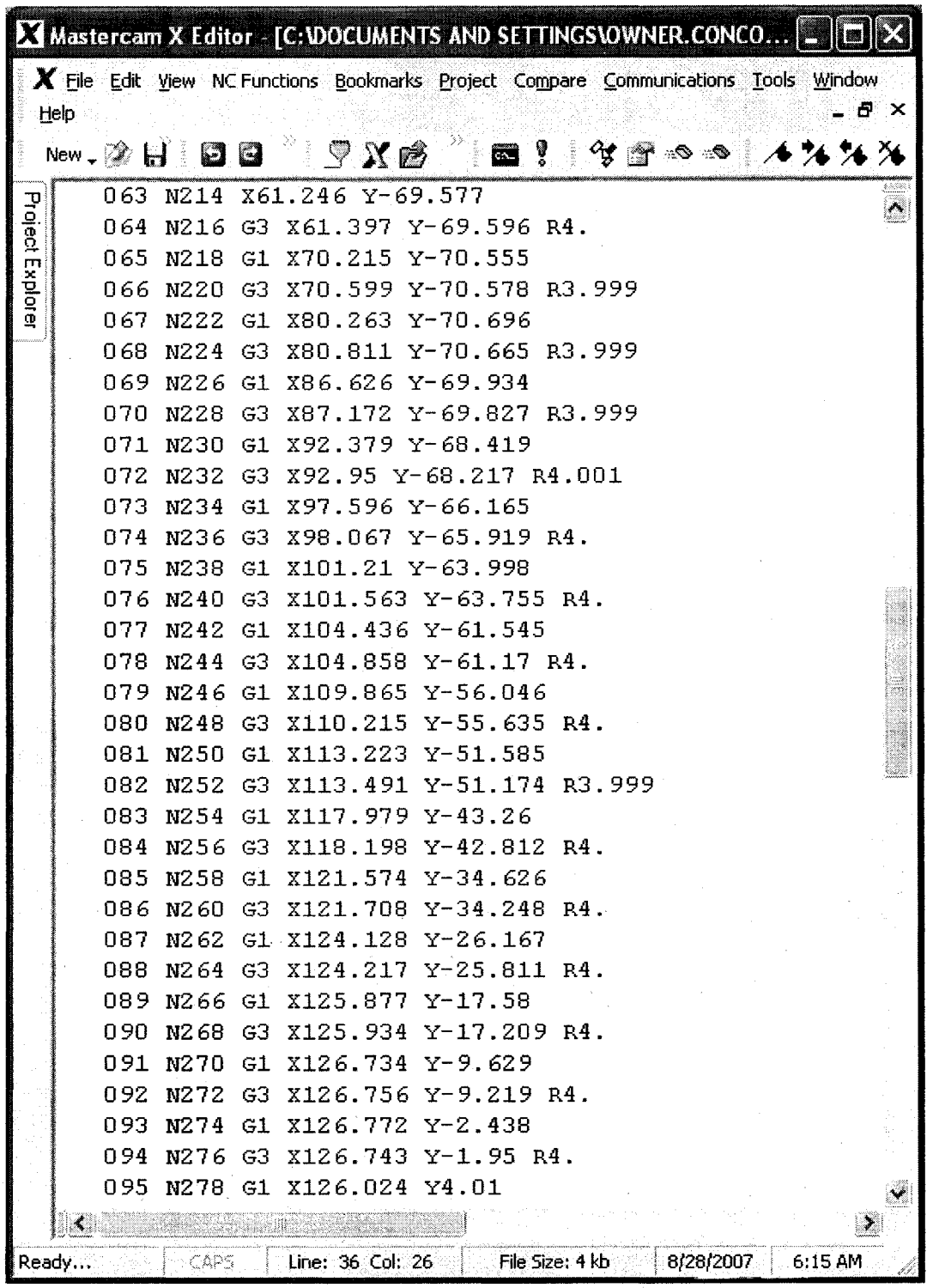


Fig. 5-6 NC codes of the part generated with the MasterCAM system

5.2 Optimized Feed rate Determination

Machining a complex curve profile with a constant feed rate may result in the difference in surface quality and accuracy, because the cutting force is variable during the process of machining. To get high quality and accuracy in a complex curve profile machining, a cutting force model need to be accurately established. As discussed before, the multi-layer neural network is a good method to represent cutting forces. Engagement angles and feed rates are the input parameters to determine cutting forces. According to the geometric model of a complex curve profile, engagement angles are calculated. The procedure to calculate engagement angles as follows.

(1) Import the NURBS profile from the CATIA into MATLAB. The numerical value of the NURBS profile, toolpath, and the first and second derivatives of the toolpaths are calculated.

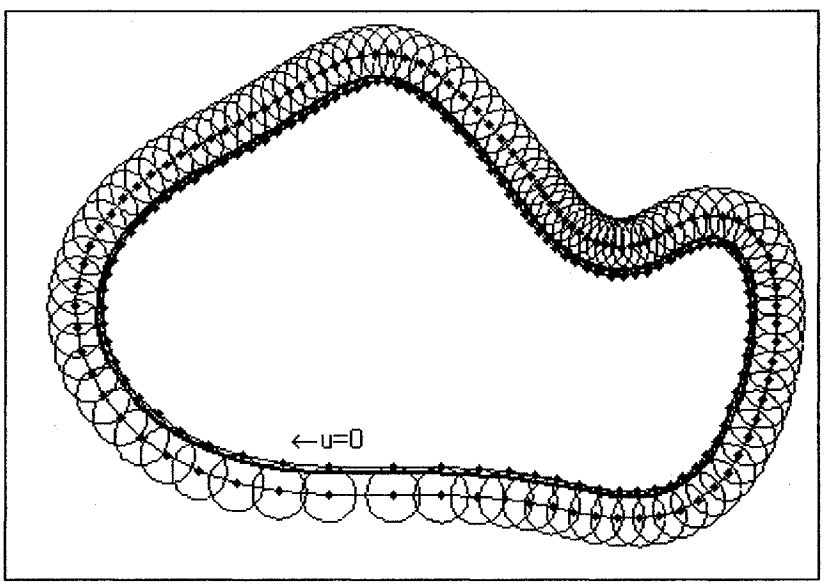


Fig. 5-7 NURBS profile and its corresponding toolpath

(2) Calculate the engagement angles along the complex curve profile (DOC=2 mm).

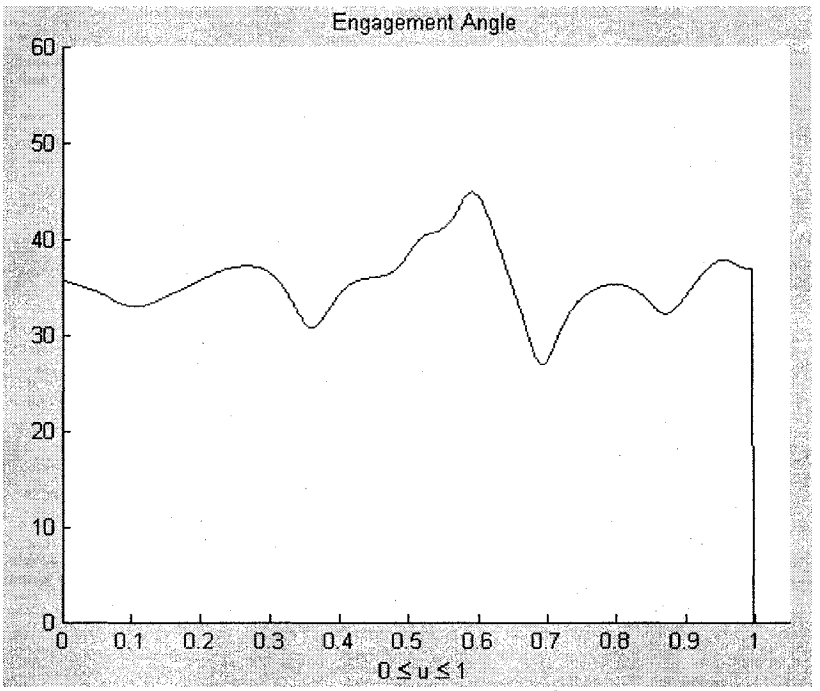


Fig. 5-8 Cutter engagement angles along the NURBS contour

The maximum engagement angle is about 2 times to the minimum engagement angle, which may bring the difference in surface quality under the same feed rate, especially, in the condition of high-speed machining. So it is necessary to establish the relationship between cutting forces and feed rates, then appropriate feed rates are calculated from the expected cutting force.

(3) A multi-layer neural network model for predicting cutting forces

A multi-layer feed forward neural network is a useful tool to describe non-linear process to predict cutting forces. There are three input parameters for the networks, feed rate, radial depth of cut, and axial depth of cut; and cutting force as the output of network.

The training data and validation data are acquired from the CUTDATA database. Through exhaustive experiments, a network with the structure 3-4-1 is finally selected, since both training errors and validation errors are small and fitting well.

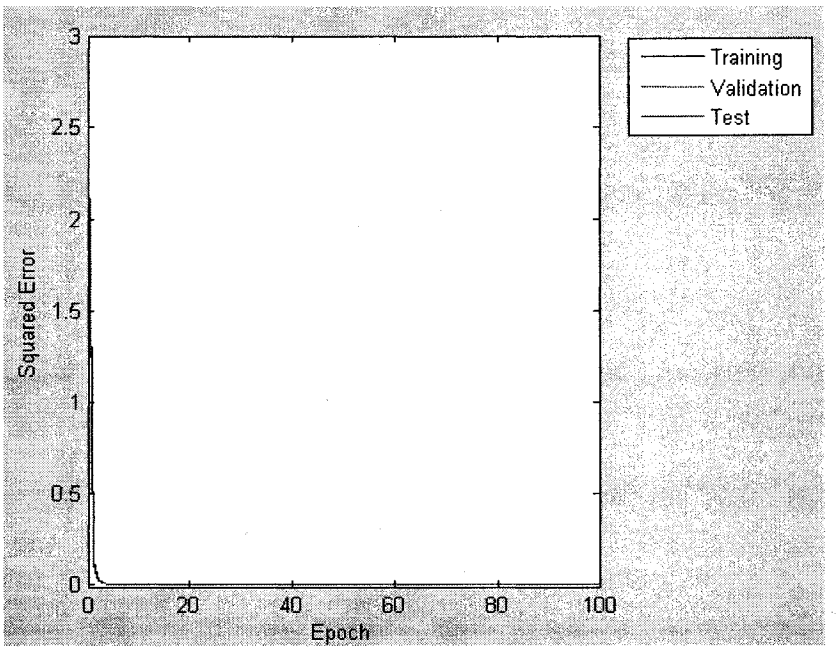


Fig. 5-9 Results in training and validating the neural network

The parameters of this network are listed below. The input layer includes

$$\text{Weights matrix}=\begin{bmatrix} 0.0113 & -0.1826 & -0.0001 \\ -0.0443 & 0.2114 & 0 \\ -0.0112 & 0.2888 & 0 \end{bmatrix}, \text{ biases vector}=\begin{bmatrix} 0.6619 \\ .0201 \\ .0061 \end{bmatrix}$$

The second layer includes

$$\text{weights matrix}=\begin{bmatrix} 1.0075 & 2.3985 & -2.053 \\ -.2845 & -1.5384 & -1.0943 \\ -1.7786 & 2.0842 & -.7903 \\ 2.1356 & -.9117 & -1.3747 \end{bmatrix},$$

$$\text{biases vector}=\begin{bmatrix} 1.9433 \\ -1.7581 \\ -2.2483 \\ 2.7658 \end{bmatrix}$$

The output layer includes

$$\text{weights matrix}=[-2.2241 \quad -.2149 \quad -2.9518 \quad -.9839]$$

$$\text{Biases vector}=[.0124]$$

Suppose in the condition that the axial depth of cut is 3 mm, the cutting forces along the profile is

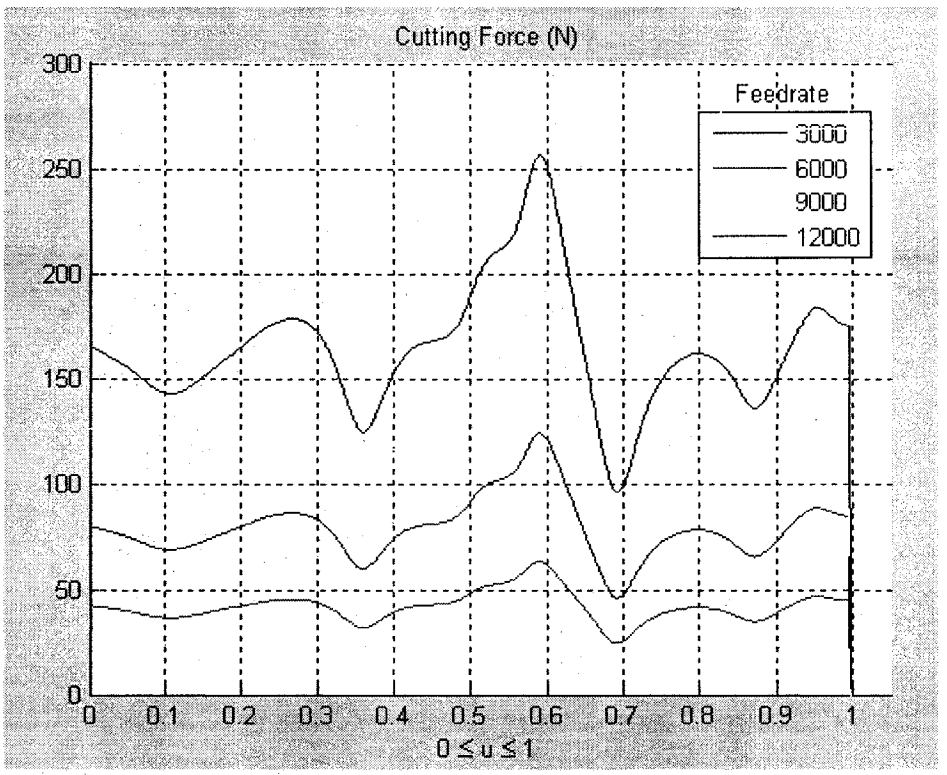


Fig. 5-10 Cutting forces simulation at different feed rates

(4) feed rate recalculating

The cutting force along the NURBS profile changes greatly, which the maximum cutting force is about 2.5 times of minimum value in the same feed rate. The difference

of cutting forces can result bad surface quality and large machining errors where the cutting force is larger. We can reduce the feed rate to get a better quality; however, it still cannot solve the problem. To lower the feed rate could only decrease the machining error, but the roughness of the whole surface could still be different because of the variable cutting force. In addition, reducing feed rates will make machining time longer and cannot take the advantage of the machine capability. A better way to overcome this problem is to break the whole profile into many small pieces, and recalculating feed rates to make the cutting force at a constant value. Generally, the tool manufacturers will recommend the appropriate cutting speeds and feed rates for the tool, hence, we can calculate the cutting force based on the data. In our case, it is appropriate to choose a carbide tool to machine aluminum alloy material, which specify the cutting force between 110 N/mm^2 and 130 N/ mm^2. The recalculated cutting force and feed rate are shown below,

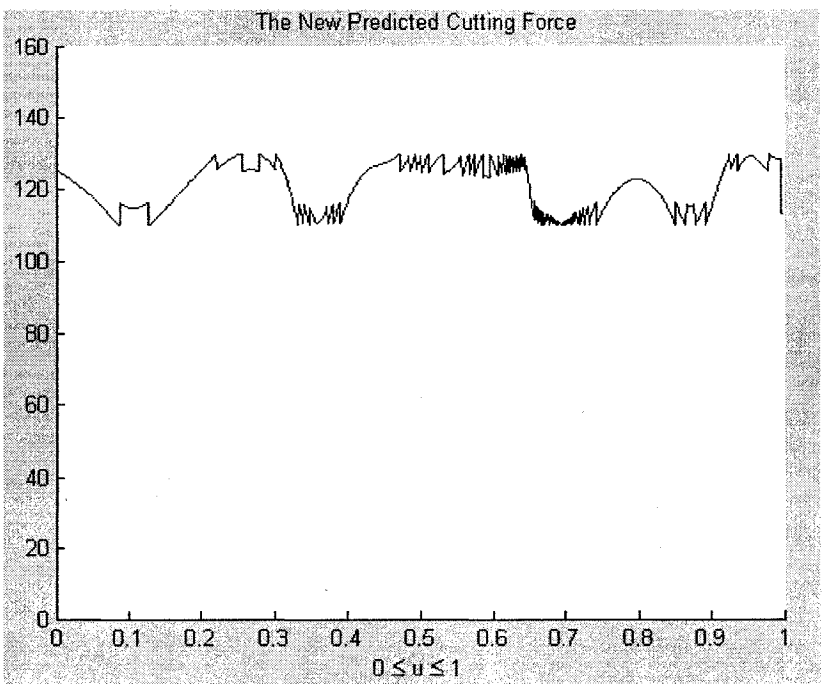


Fig. 5-11 Resultant cutting forces

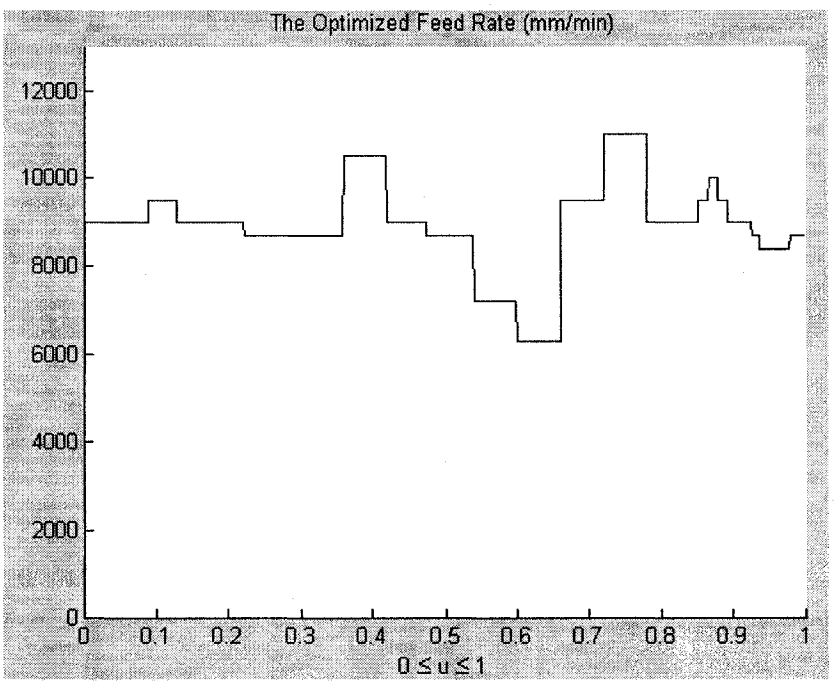
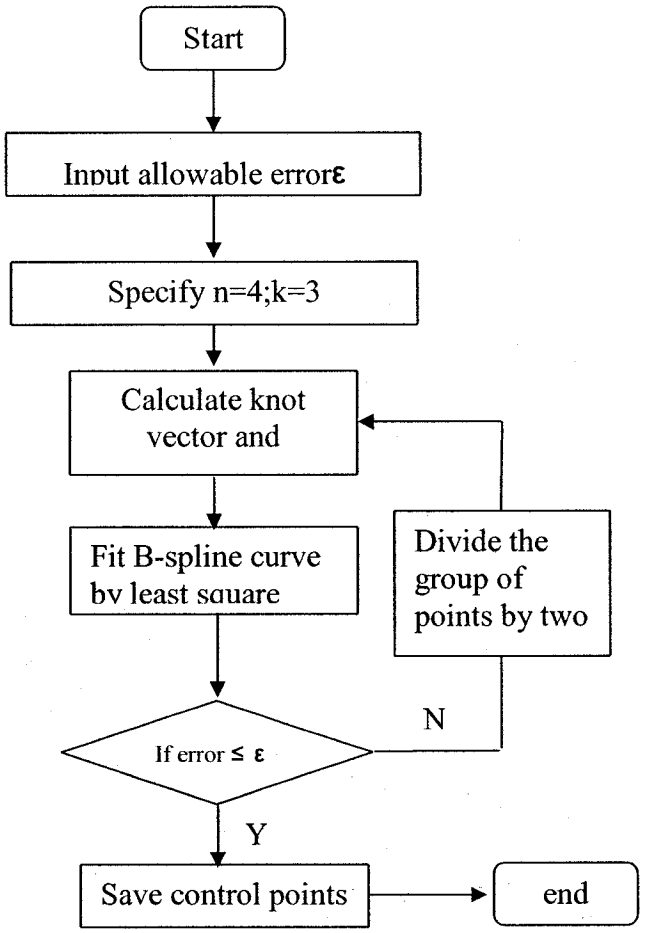


Fig. 5-12 Variable feed rates along the NURBS profile

5.3 Cubic Spline Curve Approximation

After recalculating the feed rate, the whole curve of toolpath is broken into 19 pieces by the value of feed rates. The single piece will be approximated to a new cubic spline under same feed rate, and then the new cubic spline will be fed into the CNC controller. The approximate errors and continuity between neighbors curve segments should be determined in this phase. In this case, suppose to use cubic spline curves to approximate each segment of toolpath. The algorithm of the process is below,



We set the error as 0.078 mm (about 0.002 inch), which is quite precise in machining, and the fitting result is shown below. The red curve is the fitting result, while the blue points are the fitting target.

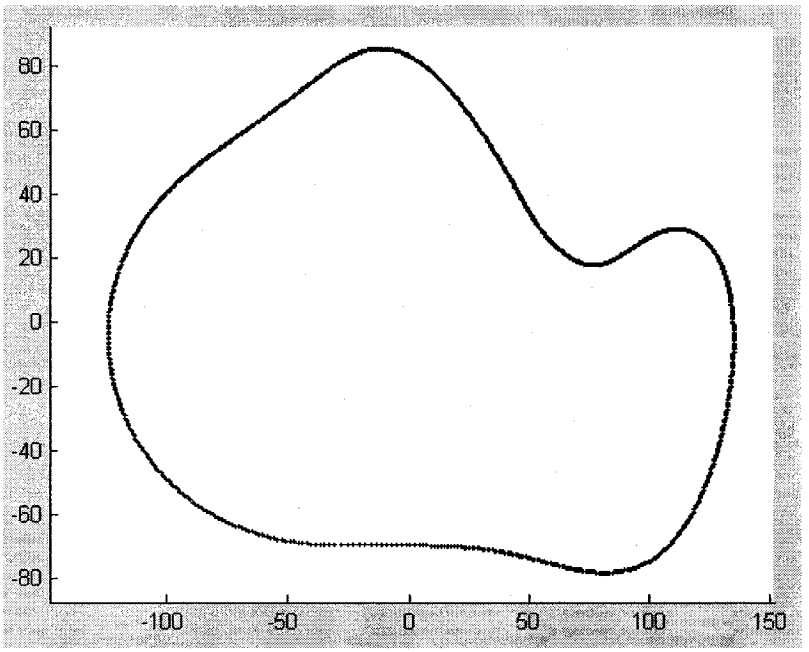


Fig. 5-13 Fitting toolpaths with different feed rates

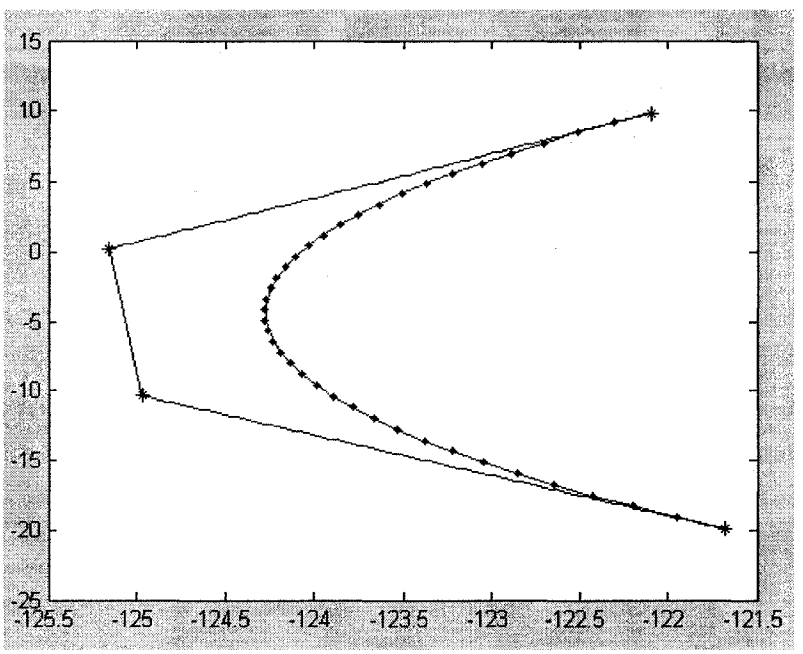


Fig. 5-14 A segment of the fitting NURBS toolpath

Through the algorithm of approximation with constraints of end derivatives, the group of points can be fitted well. Even at the piece of sharper corners, the fitting error can still be limited in the specified tolerance. The fitting result is shown in Table 3,

Table3. Parameters of fitting B-splines

1		2		3		4		5	
-30.786	-69.771	-82.068	-59.751	-115.997	-31.735	-101.900	38.471	-82.996	51.278
-48.291	-69.774	-95.578	-53.679	-131.932	-9.178	-96.063	43.382	-59.667	61.542
-66.022	-66.962	-107.968	-44.334	-123.191	22.716	-89.620	47.517	-39.332	79.354
-82.068	-59.751	-115.997	-31.735	-101.900	38.471	-82.996	51.278	-14.328	85.139
6		7		8		9		10	
-14.328	85.139	18.548	70.896	38.583	49.356	58.668	25.160	77.032	17.483
-1.750	86.279	25.892	64.372	45.288	41.314	63.861	20.960	84.900	17.634
9.550	78.886	32.462	57.025	50.541	31.932	70.175	17.425	91.523	22.441
18.548	70.896	38.583	49.356	58.668	25.160	77.032	17.483	98.463	25.521
11		12		13		14		15	
98.463	25.521	128.553	19.515	134.290	-15.357	126.879	-43.602	118.731	-58.884
108.301	30.975	135.323	9.334	133.085	-25.044	124.678	-48.947	116.349	-62.315
122.280	29.706	135.611	-3.602	130.560	-34.564	122.029	-54.128	113.615	-65.519
128.553	19.515	134.290	-15.357	126.879	-43.602	118.731	-58.884	110.472	-68.279
16		17		18		19		20	
110.472	-68.279	97.392	-75.993	85.973	-78.345	75.687	-78.264	65.274	-76.870
106.653	-71.647	93.716	-77.294	82.550	-78.613	72.195	-77.977	34.729	-69.393
102.200	-74.305	89.857	-78.043	79.107	-78.544	68.725	-77.472	3.163	-69.450
97.392	-75.993	85.973	-78.345	75.687	-78.264	65.274	-76.870	-28.085	-69.771

According to the results of fitting, the NC program is simplified to 20 blocks. In HEIDENHAIN system, the format of spline with three vertex points is represented as,

G6 X61=..Y61=..Z61=.. X62=..Y62=..Z62=..X.. Y.. Z..

So the final program is

Fig. 5-15 Optimized NC codes for the NURBS toolpath

Chapter 6 Conclusions and Future Works

6.1 Conclusions

This work proposed an intelligent approach of feed rate and toolpath determination for 2D NURBS profile of high-speed machining. There are two main contributions done in this work. One is the adjustment of feed rate along the NURBS profile in high speed machining, which is proven to be an effective way to keep material remove rate (MRR) constant in NURBS profile machining. Another contribution is to refitting NURBS toolpath that means the toolpath can be represented by NURBS formula directly, instead of chords and arc segments. At the same time, continuity of neighbor toolpath is satisfied by a given condition, to ensure the requirement of high-speed machining.

The cutting force is predicted by the multi-layer neural networks system, and the training and validation data are from CutData, a popular engineering database for machining. By identifying the geometric features of the NURBS profile, the cutting force is determined at any parts of the curve profile, and then the feed rate is determined by the value of cutting force.

6.2 Future Works

In this thesis, 2D chip load model is established to determine the feed rate along the NURBS contour machining. And the new NURBS toolpath is re-fitted in the format of HEIDENHAIN control. Logically, there are two problems left to be improved. The first one is to establish a 3D chip load model for multi-axis machining. This is a very practical problem in surface finishing with using the flowline toolpath. The flowline toolpath can be represented by 3D NURBS format, and feed rate also need to be modified according to the chip load. Another problem is that more generic toolpath fitting method is needed. Many type of control begin to support the NURBS toolpath interpolation; however they use different parameters to realize their NURBS toolpath.

Chapter 7 Bibliography

[1] Stephen, F., Arthur, G., and Peter, S., Computer Numerical Control Simplified, P.P. ISBN: 0831131330.

[2] Longbottom, J.M., and Lanham, J.D., 2006 "A review of research related to Salomon's hypothesis on cutting speeds," *International Journal of Machine Tools & Manufacture,* Vol. 46:1740-1747.

[3] Zelinski, P., 1999, ''Boeing's One Part Harmony,'' *MMS Online*, August 1999, http://www.mmsonline.com/articles/089903.html

[4] Altintas, Y., "Manufacturing Automation", P.P. ISBN: 0521659736.

[5] Kline, W.A., DeVor, R.E., and Lindberg, J.R., 1982, "The prediction of cutting forces in end milling with application to cornering cuts," *International Journal of Machine Tool Design and Research,* Vol. 22:7-22.

[6] Tlusty, J., and MacNeil, P., 1975, "Dynamics of cutting forces in end milling," *Annals of the CIRP*, Vol. 24:21-25.

[7] Ozel, T., and Altan, T., 2000, "Process simulation using finite element method-prediction of cutting forces," *International Journal of Machine Tools and Manufacture*, Vol. 40: 713-738.

[8] Fussell, B.K., and Srinivasan, K., 1989, "On-line identification of end milling process parameters Journal of Engineering for Industry," *Transactions ASME,* Vol. 111, No. 4: 322-330.

[9] Vann, C.S., and Cutkosky, M.R., 1990, "Generating process histories for feedback from cam to cad: Production Engineering Division," *Advances in Integrated Product Design and Manufacturing*, Vol. 47: 227-234.

[10] Szecsi, T., 1999, "Cutting force modeling using artificial neural networks," *Journal of Materials Processing Technology*, Vol. 92-93, pp. 344-349.

[11] Okafor, A., and Chukwujekwu; A.O., 1995, "Predicting quality characteristics of end-milled parts based on multi-sensor integration using neural networks: individual effects of learning parameters and rules," *Journal of Intelligent Manufacturing*, Vol. 6, No. 6, pp. 389-400.

[12] Luo, T., Lu, W., Krishnamurthy, K., and McMillin, B., 1998, "Neural network approach for force and contour error control in multi-dimensional end milling operations," *International Journal of Machine Tools & Manufacture*, Vol. 38, No. 10-11, pp. 1343-1359.

[13] Suh, S.H., and Shin, Y.S., 1996, "Neural network modeling for tool path planning of the rough cut in complex pocket milling," *Journal of Manufacturing Systems*, Vol. 15, No. 5, pp. 295-304.

[14] Xu, Q., Krishnamurthy, K., McMillin, B., and Lu, W., 1994, "Identification of cutting force in end milling operations using recurrent neural networks," *IEEE International Conference on Neural Networks,* Vol. 6, pp. 3828-33.

[15] Hazim, E.M., 1996, "Prediction of forces in ball-end milling using RBF neural networks," *Society of Manufacturing Engineers*, Vol. 8:145-162.

[16] Law, K.M.Y., and Geddam, A., 2003, "Error compensation in the end milling of pockets: A methodology," *Journal of Materials Processing Technology,* Vol. 139, pp.21-27.

[17] Bae, S.H., Ko, K., Kim, B.H., and Choi, B.K., 2003, "Automatic feed rate adjustment for pocket machining," *Computer-Aided Design,* Vol. 35:495-500.

[18] Cheng, C.W., and Tsai, M.C., 2004, "Real-time variable feed rate NURBS curve interpolator for CNC machining," *International Journal of Advanced Manufacturing Technology,* Vol. 23, pp. 865–873.

[19] Kloypayan, J. and Lee, Y.S., 2002, "Material engagement analysis of different endmills for adaptive feedrate control in milling processes", *Computers in Industry*, Vol. 47:55-76.

[20] Haykin, S., Neural Networks: A Comprehensive Foundation, P.P. ISBN: 0780334949.

[21] Wang, H.C., Jang, P., and Stori., J.A., 2005, "A Metric-Based Approach to Two-Dimensional (2D) Tool-Path Optimization for High-Speed Machining", *Journal of Manufacturing Science and Engineering*, Vol. 127:33-48.

[22] Yau, H.T., and Kuo, M.J., 2001, "NURBS machining and feed rate adjustment for high-speed cutting of complex sculptured surfaces", *International Journal of Production Research*, Vol. 39:21-44.

[23] Han, U.L., and Cho, D.W., 2003, "An intelligent feedrate scheduling based on virtual machining," *International Journal of Advanced Manufacturing Technology*, Vol. 22:873-882.