DEVELOPMENT OF VARIABLE FEEDRATE INTERPOLATOR FOR ERROR MINIMIZATION IN CNC MACHINE CONTROLLER

by

Biddut Bhattacharjee

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science in Engineering

Department of Industrial and Production Engineering (IPE)
Bangladesh University of Engineering and Technology (BUET)

March, 2007

The thesis titled "**Development of Variable Feedrate Interpolator for Error Minimization in CNC Machine Controller**" submitted by **Biddut Bhattacharjee**, Student No. **040408020P**, Session **April 2004**, has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Master of Science in Industrial & Production Engineering on **March 2007**.

BOARD OF EXAMINERS

1. ______________________ Chairman

Dr. Abdullahil Azeem
Assistant Professor
Dept. of Industrial & Production Engineering,
BUET, Dhaka, Bangladesh

2. ______________________ Member (Ex-officio)

Dr. Nikhil Ranjan Dhar
Professor & Head
Dept. of Industrial & Production Engineering,
BUET, Dhaka, Bangladesh

3. ______________________ Member

Dr. A.K.M. Masud
Associate Professor
Dept. of Industrial & Production Engineering,
BUET, Dhaka, Bangladesh

4. ______________________ Member (External)

Dr. Md. Liakot Ali
Assistant Professor
Institute of Information & Communication Technology,
BUET, Dhaka, Bangladesh

CANDIDATE'S DECLARATION

It is hereby declared that this thesis or any part of this has not been submitted elsewhere for the award of any degree or diploma

Biddut Bhattacharjee

This thesis is dedicated to

The students of Industrial & Production Engineering

ABSTRACT

The parametric interpolators of modern CNC machines use Taylor's series approximation to generate successive parameter values which in turn gives the x,y,z coordinates of the tool positions after substituting into the curve equation. In order to achieve greater accuracy higher order derivatives are required which complicates the calculation when the curve is represented by NURBS curve. This method also utilizes the curvature to estimate the chordal error on a given segment which neglects a fraction of the error. In order to avoid calculating higher derivatives and make the calculations easier the classical fourth-order Runge-Kutta method is proposed in this research which only requires the first derivative to be calculated but achieves the accuracy of Taylor's approximation with higher order terms. This thesis also proposes the estimation of chordal error on the average value of the parameters at the end points of a given curve segment, which does not require calculation of curvature at every segment.

Finally a variable feedrate interpolator is proposed which adjusts the feedrate depending on the curve properties to maintain the chordal error within the specified tolerance. To evaluate the feasibility and performance of the proposed method computer simulations are performed on different types of spline curves starting from cubic spline to NURBS Simulation results show the superiority of variable feedrate interpolation in terms of feedrate and chordal error compared to those resulted from using constant feedrate interpolation employing Taylor's approximation.

ACKNOWLEDGEMENTS

First of all I would like to express my sincere gratitude to Almighty whose blessings made me capable of accomplishing this thesis work.

I am indebted to Dr. Abdullahil Azeem for his kind consideration to supervise me. From the very beginning till the end of this research it had been a great privilege and honor for me to work under his inspiring guidance. From providing me access to international journal papers to cheering me up for another try when things appeared to be frustrating, he supported me with everything. I am grateful to him.

I would like to express my gratitude to Dr. Nikhil Ranjan Dhar for giving me permission to start the research and for giving valuable suggestions.

I express my gratitude to Dr. A.K.M Masud and Dr. Md. Liakot Ali for their constructive remarks and for kindly evaluating this research

I like to thank my colleagues of the department, especially A.K.M. Kais Bin Zaman, Md. Moin Uddin, Md Zahid Hossain and Md. Tanvir Irfan for giving me valuable suggestions, providing reference books and MATLAB.

My family members tolerated me a lot during the hard times of coding and writing this report I am grateful to them for being patient.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1
INTRODUCTION

1.1 Background

Free-form or sculptured surfaces are used in a variety of applications in automotive, aerospace, ship building industries. Dies and molds of automotive body, propeller and impeller blades, and aircraft wings etc. are typical examples that involve free-form surface design and machining. Modern CAD/CAM systems provide a designer with tools for defining such free-form surfaces represented by 3-D parametric curves and surfaces. Multi-axis Computer Numerical Control (CNC) machining is mostly used for machining such complex free-form surfaces. When a free-form surface is machined using a CNC machine, the path of the cutting tool can be considered as a 3-D spline curve. Therefore surface machining involves four main phases: i) Optimization of machining process parameters, ii) Tool path planning, iii) Tool path interpolation and iv) Servo control. Phase i) and ii) deals with automatic feature recognition, the selection of cutter and its speed, feedrate and depth of cut etc. and the optimization of tool path. Phase iii) is concerned with the conversion of tool paths obtained from a tool path planning system into time-dependent commands for driving the servo control system of a CNC machine maintaining the parameters resulting from phase i). Phase iv) mainly deals with closed loop control issues such as adaptive control strategies, feedback, stability etc. The traditional approach of machining parts having geometric features represented by parametric curves is shown in Figure 1.1. The designers first develop the model of the part in the CAD system. The part geometry is transferred to the CNC system by means of a part program (G-codes, M-codes) consisting mainly of motion commands. These motion commands are translated in real time by the CNC interpolator into a special form, containing the coordinates of successive tool positions, before being sent to the control loops of machine tool drives. As most CNC interpolators can process only straight line and circular arc motion commands, the parametric curve representing the tool path is discretized into a set of line segments which approximate the original curve to a desired accuracy. This segmentation is accomplished in the CAD/CAM system and the segments are fed into the post-processor for the generation of G-codes. These G-codes are then downloaded into the CNC system for real time command generation of tool positions.

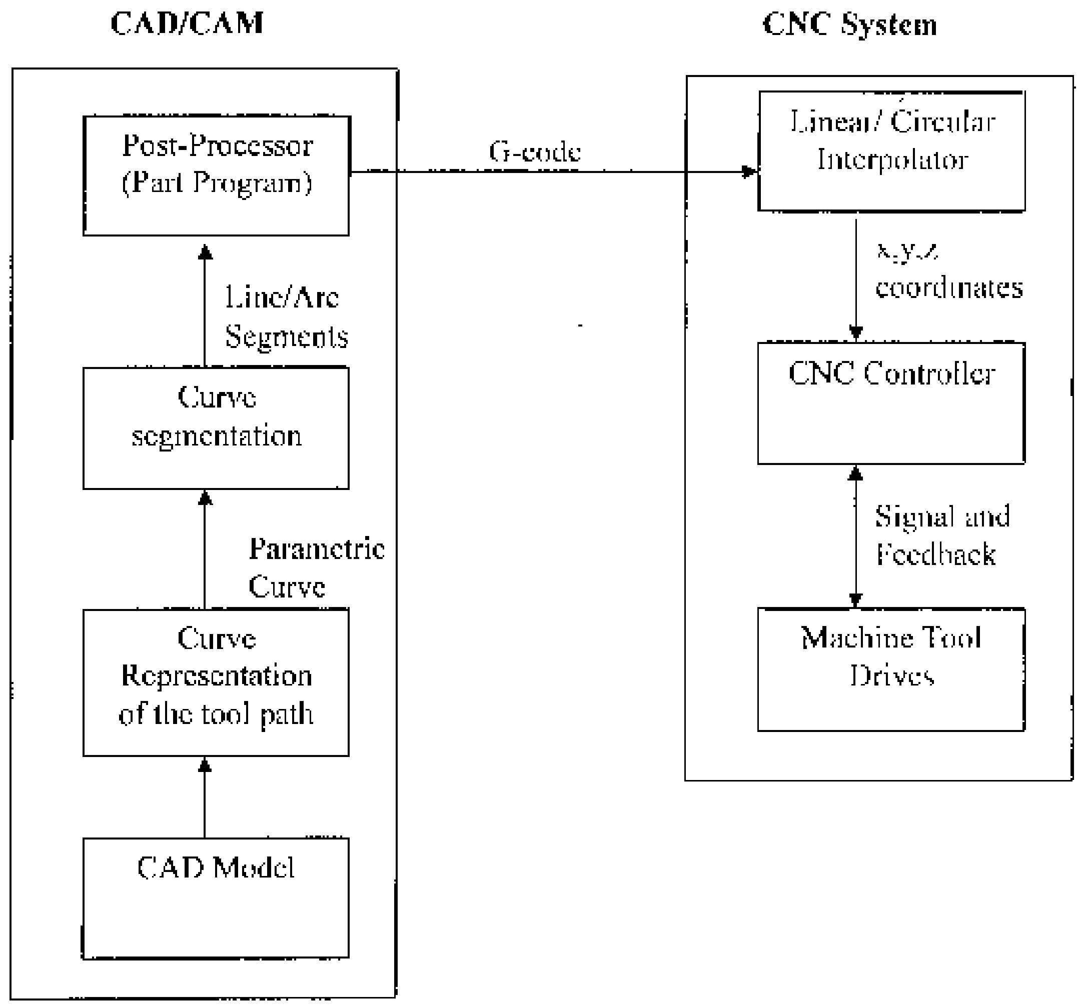

Figure 1.1: Traditional method of machining parametric curves.

To better approximate the original curve and to keep the error within tolerance, the curve has to be divided into a large number of linear and circular arc segments. Such linearized-segmented contours are undesirable in real applications for the following reasons:

- The transmission error between CAD/CAM and CNC machine occurs due to huge number of data.
- A large memory is required for the CNC machine.
- Surface accuracy is deteriorated because of the discontinuity of segmentation
- The speed of tool movement becomes unsmooth because of the linearization of the curve in each segment

- If the number of segments is reduced, the tool path will approximate the curve less accurately and chord error will be increased.

In order to eliminate the problems of traditional approach, modern CNC machines are being developed having interpolators capable of receiving only the parametric curve coefficients and the control points. In recent method, shown in Figure 1.2, the tool path represented by parametric curve is not discretized in the CAD/CAM system; instead the G-codes of the curve containing curve information are transferred to the CNC machine's interpolator which then generates the position commands for the machine tool drives.

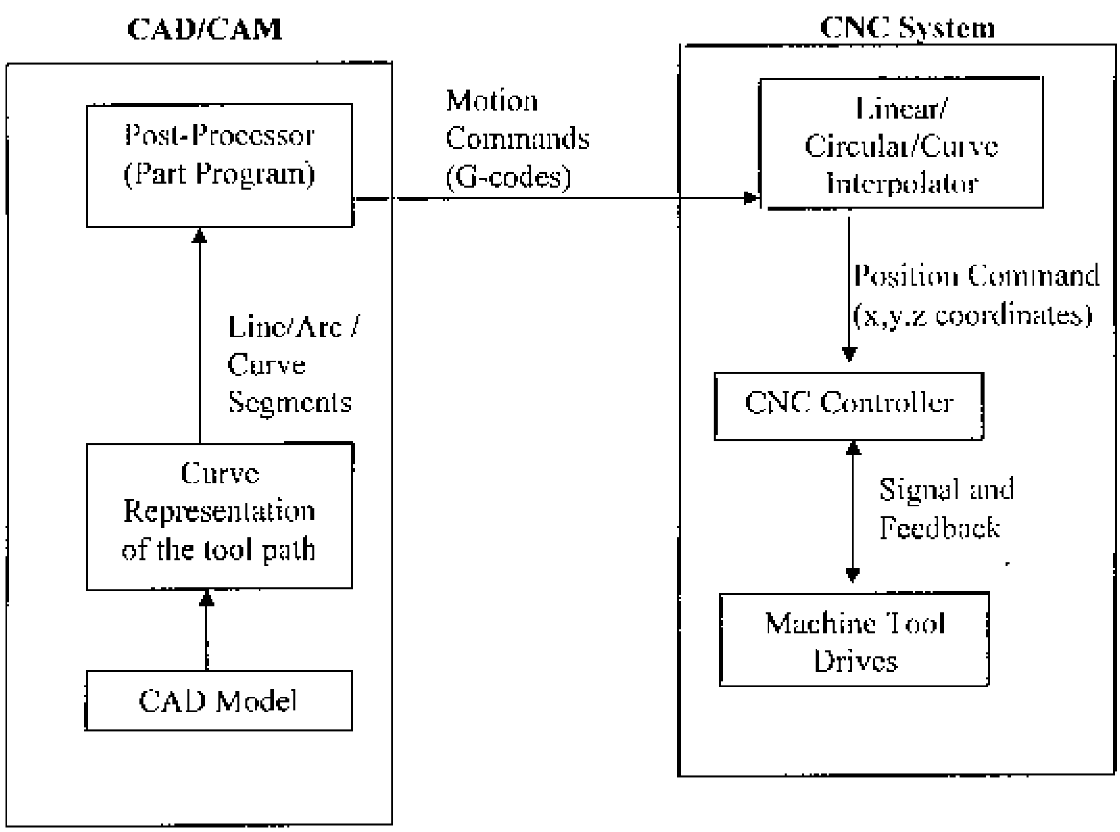

Figure 1.2: Recent method of machining parametric curves

The parametric spline interpolator of modern CNC machine's controller receives only the curve information and generates successive tool positions on the desired curve by increasing the parameter value in an iterative method as follows:

$$u_{i+1} = u_i + \Delta(u_i)$$

Where u_i is the current value of parameter, u_{i+1} is the parameter value after sampling period T of the controller and $\Delta (u_i)$ is the incremental value. The interpolated points (i.e. x, y, z coordinates) on the desired curve are calculated by substituting u_i's into the corresponding mathematical model of the originally designed curve which may be a Bezier curve, B-spline curve, a cubic spline curve or Non-uniform Rational B-Spline (NURBS) curve. The parameter value is updated using either the first-order or the second-order Taylor's series approximation method. When constant feedrate interpolation is used, this approximation results in truncation errors which in turn cause the inaccuracy in the actual feedrate and its fluctuation. The chordal deviation, resulting from the straight line motion of the cutting tool between two adjacent points, directly affects the surface accuracy of the machined part. In machining of free-form surfaces in automotive and aerospace industries, accuracy of the machined part is of outmost importance. The current practice is to estimate the chordal error by approximating the curvature of the corresponding curve segment. This results in error value less than the actual value.

1.2 Objectives

The main objective of this research is to develop a variable feedrate interpolator which changes the feedrate smoothly in order to keep the chordal error within predetermined tolerance. Specifically the objectives can be stated as follows:

- To improve the accuracy of feedrate and its fluctuation with respect to the given feedrate by employing the Fourth-order Runge-Kutta method to generate the successive values of the parameter.
- To compare between chordal error resulting from using the present method and from the proposed method in which chordal error will be calculated at a point on the curve corresponding to the average of the parameter values at the end points of a curve segment.

1.3 Methodology

The methodology followed in this research can be outlined as follows:

- Analysis of the present method of interpolation algorithm employed in CNC controller for generating successive tool positions and identification of its shortcomings.
- Development of mathematical model of the proposed interpolation algorithm by using the Runge-Kutta method..
- Derivation of the mathematical expressions for calculating the chordal error according to the proposed method.
- Development of variable feedrate interpolation algorithm to maintain a predefined allowable chordal deviation.
- Performance evaluation of the proposed interpolation technique by computer simulation based on the proposed mathematical model for 3-D spline curves like a cubic spline curve, a B-spline curve and a Non-uniform Rational B-Spline (NURBS) curve.

1.4 Overview of the Thesis

This thesis consists of eight chapters. The first chapter briefly introduces the subject of computer-aided design and manufacturing, especially the integration of CAD/CAM system and the CNC machine tool. The chapter ends with the overview of interpolation used in modern CNC machines to generate successive tool positions on the parametric curve.

Chapter 2 discusses the state of the art in the field of parametric interpolation for CNC machine tool.

Chapter 3 explains the traditional approach of CNC interpolation using the Taylor's series approximation for machining of parts having free-form curves. The shortcomings of this approach are also pointed out that need to be overcome in order to minimize machining error and maximize surface quality of the machined part.

Chapter 4 explains the proposed method of interpolation employing the classical fourth-order Runge-Kutta method. It also explains a new method for estimating chordal error. Finally the variable feedrate interpolation algorithm is developed in which the feedrate will be adjusted in order to keep the error within specified tolerance.

Chapter 5 presents the theory of B-spline and NURBS curve. The Cox-deBoor algorithms for evaluating points on the curve and the derivatives at a given value of parameter are also explained.

Chapter 6 shows the results of computer simulations. The algorithms for computer program and the curves on which simulation was performed are also presented. The chapter ends with the discussion on the results.

Chapter 7 is the conclusion of this thesis where the whole work is summarized, and the recommendations for further research are mentioned.

CHAPTER 2

LITERATURE REVIEW

Researchers have developed various techniques of parametric interpolation for the CNC machine tool's controller. The main objective of all these research works was to reduce the chordal error and feedrate fluctuation. To machine parts having sculptured surfaces the cutting tool has to traverse along a 3-D spline curve of varying curvature at different segments. Segments of the curve having small radius of curvature are the sources of higher chordal error if the tool traverses with a constant feedrate. Again, the surface quality and machining time depend on the optimum feedrate. Hence, researchers tried to find out ways to resolve this conflict.

Chou and Yang [1] developed a formulation for the command generation of parametric curve machining taking into consideration the dynamics of the 3-axis CNC machine tool and the machining process. The data stored in the CAD system for a part model is usually static in nature and represented by unitless parameters. For the motions and control of machine tool, the data needs to be transformed into a time dependent domain. In their research the geometrical properties of a cutter path, including position, tangent, curvature and torsion are analytically related to the motion dynamics of a machine tool including feedrate, acceleration, jerk and driving force which are needed for command generation and machined surface quality analysis. Based on their work, Huang and Yang [2] developed a generalized variable feedrate parametric interpolator by using the Euler approximation method for generating successive parameter values. Their method yielded satisfactory results compared to those from using linear, circular and cubic spline interpolators as long as the curvature of the curve for machining was small.

Bedi et al. [3] implemented a cubic B-spline interpolation by applying the forward difference algorithm. They implemented the interpolating B-spline curve through the data points lying on the product rather than approximating those points. Bedi et al [4] extended the B-spline curve interpolator to a surface interpolator by utilizing the greater computational capacity of a network of computers called transputers. These transputers when interfaced with the controller of the machine tool can avoid problems such as communication errors, jerky motion, and gouging.

Wang and Yang [5] developed a method of interpolating a set of discrete data points to form a composite quintic spline for application in precision machining. A cubic spline fitted between two consecutive data points by approximating the arc-length with the chordal length is a chordal-length parameterized cubic spline. Their algorithm starts with fitting a set of given data with a cubic spline and then on the basis of this curve a composite quintic spline is generated with nearly arc-length parameterization, 2nd order continuity and no high-order oscillations.

Koren et al. [6] developed a real-time constant feedrate interpolation algorithm for parametric curves based on the first- and second-order Taylor's series approximation method. Shpitalni et al. [7] developed the same interpolator for 2-D implicit curves and 2-D parametric curves. The results in terms of position accuracy and feedrate deviations show significant improvements. Yang and Kong [8] proposed a parametric interpolator for variable feedrate based on the first- and second-order Taylor's expansion. This research is a comparative study of linear and parametric interpolators on several aspects such as memory size, feedrate fluctuation and CPU time.

Kiritsis [9] presented an incremental step algorithm for general 3D and 2D parametric curve interpolation, which, for a given position of the tip of the cutting tool, calculates the next position as closely as possible to the curve by fulfilling two criteria: first, the next step must follow a given direction along the curve; second, its normal distance from the curve must be the minimum one among the corresponding distances of the steps that conform to the first criterion. This algorithm requires many complex calculations which restrict its use for real-time application.

Lo [10] proposed feedback interpolators for both 2-D implicit curves and parametric curves which utilize feedback action to correct feedrate error. Zhang and Greenway [11] implemented a real-time non-uniform rational B-spline (NURBS) interpolator for an articulated robot.

Lo et al. [12] presented a CNC interpolator with a path compensation for the contour error applicable for repeated contour machining processes. Their method generates a compensated path, for which the contour error is opposite to the contour error for the actual path which is monitored and saved by the controller using the feedback devices. and then to utilize the compensated path as the new reference tool path for the subsequent repeated contour machining. The experimental results proved the effectiveness of the algorithm in improving the accuracy of the subsequent repeated machining. A new approach to CNC tool path generation was suggested by Lo [13]. Traditionally the CNC controller generates command with a specified cutter-location (CL) velocity known as the feedrate. But it is the cutter-contact (CC) velocity which is related to the machining efficiency and quality. In this new approach, three different methods of linear, curve and surface interpolator is developed where the CL velocity is derived that results in the motion of the cutting edge with the specified feedrate.

Yeh and Hsu [14] developed a speed-controlled interpolation algorithm by deriving a compensatory parameter which improved the curve speed accuracy. Rong-Shine [15] proposed a real-time surface interpolator for 3-D parametric surface machining on a 3-axis machine tool. This interpolator can read surface g-codes, containing geometric information such as the coefficients of the parametric surface, as well as cutting conditions.

A variable feedrate interpolation scheme for planar implicit curves is presented by Xu [16] et al. which relates the geometric properties like tangent, curvature etc to the kinematics of the tool along the curve. They presented the interpolation schemes for the cases of constant feedrate, path-geometry-varying feedrate and time varying feedrate considering initial acceleration and final deceleration. The experimental results proved that machining efficiency can be increased while satisfying a predefined tolerance of the contour error of the curve.

Fleisig et al. [17] presented a new algorithm for the off-line interpolation of data points and real-time axis command generation which results in constant feedrate and reduced angular acceleration. The splines produced by the algorithm are C^2 continuous and independent of machine tool kinematics. Lartigue et al. [18] presented an accurate method to generate a CNC tool path for a smooth free-form surface in terms of planar cubic B-spline curves which can be fed into the curve interpolator. Their method first interpolates the break points generated by computing the offset surface-driving plane intersection curve reflecting the curvature. Then the maximum scallop height along the scallop curve and the maximum pick feed are evaluated so that maximum scallop height coincides with the tolerance. Results verify the method for better machining accuracy and lower memory requirements.

Farouki and Tsai [19] proposed a systematic derivation of the proper Taylor series coefficients for variable feedrate interpolators, using simple notations and recursive formulae suited to real-time computations. Bahr et al. [20] presents a method of CNC interpolation in which the finite forward differencing algorithm for fast evaluation of points on a cubic parametric curve is modified so that the inherent accumulative error in the calculation of one piece of curve will not propagate to the whole composite curve. The results prove that the method is accurate, adaptive to curvature of the curve and reduces jerk.

Zhiming et al. [21] presented a real-time interpolation algorithm for NURBS curves based on the second-order Taylor's approximation which resulted in smaller contour errors and feedrate fluctuations. But their calculation of contour error was based on circular approximation for estimating curvature of a segment which can not give the real value of the error.

An approach for linear and angular feedrate interpolation for planar implicit curves is presented by Xu [22]. In multi-axis CNC machining, the cutting tool combines translation and rotational movements with respect to a workpiece. This approach relates the angular feedrate to the linear feedrate and then the angular feedrate interpolation is developed from the linear feedrate interpolation.

The exact value of arc length for a B-spline or NURBS curve is very difficult to calculate. As a result a feedrate profile with desired accelerations in real-time is also difficult to achieve. Nam et al. [23] proposed a method that results in exact feedrate trajectory through jerk-limited acceleration profiles for parametric curves. The recursive trajectory generation method estimates an admissible path increment and determines the initiation of the final deceleration stage according to the distance left to travel at every sampling time. The simulation results illustrate the generation of smooth feedrate profile.

Tsai et al. [24] developed the NURBS interpolator considering acceleration/ deceleration planning before the interpolation. The experimental results suggest that interpolator with bell shape acceleration/deceleration planning is satisfactory for realization of the acceleration/deceleration. before feedrate interpolation. S. Y. Jeong et al. [25] proposed sampled-data parametric interpolation technique which estimates the parameter using a quintic polynomial obtained from the tabulated parameter and length data. As the arc-length of each segment of the curve is estimated from curvature and chord length, this approach is also not that accurate.

In all the research works the interpolation for parametric curves or surfaces is based on the Taylor's series approximation for generating successive values of the parameter. Considering the accuracy of results first three terms of the series are used, requiring the second derivative of the parametric curve to be calculated at every sampling time. Derivatives of curves like B-spline and NURBS are quite complex and requires several iterations. This research proposes the fourth-order Runge-Kutta (RK) method to find the parameter values requiring only the first derivatives of parametric curve. The RK method is computationally easier to implement due to its recursive nature. This research also proposes a different method of calculating the chordal error avoiding the calculation of curvature at every sampling time. Based on the calculated error the algorithm utilizes a corrective action to find the new parameter value for which the chordal error will be within tolerance.

PRESENT METHOD OF CNC INTERPOLATION

3.1 Taylor's Series Approximation

The general form of a three dimensional parametric curve of nth order can be expressed as

$$\vec{P}(u) = x(u)\hat{i} + y(u)\hat{j} + z(u)\hat{k}\text{, and}$$

$$\begin{aligned} x(u) &= a_{n,x}u^n + a_{n-1,x}u^{n-1} + \dots\dots\dots + a_{1,x}u + a_{0,x} \\ y(u) &= a_{n,y}u^n + a_{n-1,y}u^{n-1} + \dots\dots\dots + a_{1,y}u + a_{0,y} \\ z(u) &= a_{n,z}u^n + a_{n-1,z}u^{n-1} + \dots\dots\dots + a_{1,z}u + a_{0,z} \end{aligned} \qquad \dots\dots\dots\dots (3.1)$$

where, $0 \le u \le u_{max}$.

The coordinates (x,y,z) of points on the curve is obtained by substituting the values of the parameter, u, in the above equations. In Computer Numerically Controlled (CNC) machine tool, the tip of the cutting tool moves from one such point to the next and from there to next, and so on. To find the successive values of the parameter, u, the following iteration is used:

$$u_{i+1} = u_i + \Delta(u_i)$$

Where, u_i is the parameter at time t_i, u_{i+1} is the parameter at time t_{i+1} and $\Delta(u_i)$ is the incremental value. The simplest approach would be to increment u uniformly, i.e. in equal small increments Δu, and calculate the corresponding $x_{i+1} = x(u_{i+1})$, $y_{i+1} = y(u_{i+1})$ and $z_{i+1} = z(u_{i+1})$ at time t_{i+1}. The sampling period, $T = t_{i+1} - t_i$, is constant for a machine's controller. This approach has the following problems:

- Depending on the curvature the chordal length,

  $L_i = \left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2\right]^{1/2}$, between points $(x_{i+1}, y_{i+1}, z_{i+1})$ and (x_i, y_i, z_i) is not equal. But the tool traverses this distance in equal time intervals T. So the feedrate is not constant, rather it varies from one segment to another along the curve.

- The optimal size of the increment Δu is not known. If Δu is too small, the resultant L_is are too small as well and machining slows down. If Δu is too big, the resultant L_is are too big and machining accuracy is not maintained.

That is why real-time parametric interpolator can not be based on uniform segmentation of the curve according to equal increments of the parameter, u. Segmentation of the curve should be such that the chordal lengths, L_i, of each segment are equal so that the feedrate does not vary along the curve. Researchers have developed the method for determining successive values of u so that the chordal lengths L_i (machined at each sampling period T) are constants, resulting in a constant feedrate. The method is outlined below:

The feedrate, V(u) along the curve is defined by

$$V(u) = \frac{ds}{dt} = \left(\frac{ds}{du}\bigg|_{u=u_i}\right)\left(\frac{du}{dt}\bigg|_{t=t_i}\right) \quad \text{.........................(3.2)}$$

$$\text{or, } \frac{du}{dt}\bigg|_{t=t_i} = \frac{V}{\left|\left(\frac{ds}{du}\big|_{u=u_i}\right)\right|} = \frac{V}{\sqrt{\left[\frac{dx(u)}{du}\big|_{u=u_i}\right]^2 + \left[\frac{dy(u)}{du}\big|_{u=u_i}\right]^2 + \left[\frac{dz(u)}{du}\big|_{u=u_i}\right]^2}} \quad \text{..... (3.3)}$$

The solution of the above equation for a constant V results in the required value of the parameter, u, at any time t. As analytical solution of the above equation is difficult and infeasible to implement, a numerical method is used to find the solution. The function u is expanded at the neighborhood in time t_i by using the Taylor's Series Approximation.

$$u(t_{i+1}) = u(t_i) + (t_{i+1} - t_i)\frac{du}{dt}\Big|_{t=t_i} + \frac{(t_{i+1} - t_i)^2}{2}\frac{d^2u}{dt^2}\Big|_{t=t_i} + HigherOrderTerms$$

$$\Rightarrow u_{i+1} = u_i + T\frac{du}{dt}\Big|_{t=t_i} + \frac{T^2}{2}\frac{d^2u}{dt^2}\Big|_{t=t_i} + HigherOrderTerms$$

$$\ldots\ldots\ldots\ldots (3.4)$$

Now,

$$\frac{d^2u}{dt^2}\Big|_{t=t_i} = \frac{d}{dt}\left(\frac{V}{\frac{ds}{du}\Big|_{u=u_i}}\right) = \frac{-V\frac{d}{dt}\left(\frac{ds}{du}\Big|_{u=u_i}\right)}{\left(\frac{ds}{du}\Big|_{u=u_i}\right)^2} = \frac{-V\frac{d}{du}\left(\frac{ds}{du}\Big|_{u=u_i}\right)\frac{du}{dt}}{\left(\frac{ds}{du}\Big|_{u=u_i}\right)^2}$$

$$= \frac{-V\left(\frac{d^2s}{du^2}\Big|_{u=u_i}\right)V}{\left(\frac{ds}{du}\Big|_{u=u_i}\right)^2\frac{ds}{du}\Big|_{u=u_i}} = \frac{-V^2\left(\frac{d^2s}{du^2}\Big|_{u=u_i}\right)}{\left(\frac{ds}{du}\Big|_{u=u_i}\right)^3} \ldots\ldots\ldots\ldots (3.5)$$

And,

$$\frac{d^2s}{du^2}\Big|_{u=u_i} = \frac{d}{du}\left\{\sqrt{\left[\frac{dx(u)}{du}\Big|_{u=u_i}\right]^2 + \left[\frac{dy(u)}{du}\Big|_{u=u_i}\right]^2 + \left[\frac{dz(u)}{du}\Big|_{u=u_i}\right]^2}\right\}$$

$$= \left(\frac{1}{2}\right)\frac{2\left(\frac{dx(u)}{du}\Big|_{u=u_i}\right)\left(\frac{d^2x(u)}{du^2}\Big|_{u=u_i}\right) + 2\left(\frac{dy(u)}{du}\Big|_{u=u_i}\right)\left(\frac{d^2y(u)}{du^2}\Big|_{u=u_i}\right) + 2\left(\frac{dz(u)}{du}\Big|_{u=u_i}\right)\left(\frac{d^2z(u)}{du^2}\Big|_{u=u_i}\right)}{\sqrt{\left[\frac{dx(u)}{du}\Big|_{u=u_i}\right]^2 + \left[\frac{dy(u)}{du}\Big|_{u=u_i}\right]^2 + \left[\frac{dz(u)}{du}\Big|_{u=u_i}\right]^2}}$$

$$= \frac{\left(\frac{dx(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2x(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dy(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2y(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dz(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2z(u)}{du^2}\bigg|_{u=u_i}\right)}{\sqrt{\left[\frac{dx(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dy(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dz(u)}{du}\bigg|_{u=u_i}\right]^2}}$$

.... (3.6)

Putting equation (3.6) into equation (3.5) we get

$$\frac{d^2u}{dt^2}\bigg|_{u=u_i} = -V^2\frac{\left(\left(\frac{dx(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2x(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dy(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2y(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dz(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2z(u)}{du^2}\bigg|_{u=u_i}\right)\right)}{\left(\left[\frac{dx(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dy(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dz(u)}{du}\bigg|_{u=u_i}\right]^2\right)^2}$$

.... (3.7)

Finally, putting equations (3.3) and (3.7) into equation (3.4) and neglecting the Higher order terms we find,

$$u_{i+1} = u_i + \frac{VT}{\sqrt{\left[\frac{dx(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dy(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dz(u)}{du}\bigg|_{u=u_i}\right]^2}}$$

$$-\frac{(VT)^2}{2}\frac{\left(\left(\frac{dx(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2x(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dy(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2y(u)}{du^2}\bigg|_{u=u_i}\right)+\left(\frac{dz(u)}{du}\bigg|_{u=u_i}\right)\left(\frac{d^2z(u)}{du^2}\bigg|_{u=u_i}\right)\right)}{\left(\left[\frac{dx(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dy(u)}{du}\bigg|_{u=u_i}\right]^2+\left[\frac{dz(u)}{du}\bigg|_{u=u_i}\right]^2\right)^2}$$

.... (3.8)

The above equation is the second order Taylor's Series Approximation to find the value of parameter, u_{i+1}, at time t_{i+1}, calculated from the current value, u_i, and the derivatives of the current tool position (x_i, y_i, z_i) with respect to u. This new value, u_{i+1}, is substituted in equation (3.1) to find the next reference tool position (x_{i+1}, y_{i+1},

z_{i+1}). Although this interpolation algorithm is supposed to provide constant feedrate machining, there are variations in actual cutting speed due to the approximation of Taylor's Series. Neglecting the higher order terms (HOT) of the series results in a value of u_{i-1} different from that which would be obtained if the HOT were used. As a result, the chordal distance, L_i, between (x_i, y_i, z_i) and (x_{i+1}, y_{i+1}, z_{i+1}), traveled by the cutting tool in T seconds, is such that the actual feedrate of the tool between this two positions is not exactly V, the specified feedrate. The actual speed, V_i at the ith segment is calculated from the following equation:

$$V_i = \frac{L_i}{T_i} = \left(\frac{1}{T}\right)\left[(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2\right]^{1/2} \quad \text{............} (3.9)$$

3.2 Determination of Chordal Error

As the cutter moves straight between contiguous interpolated points, two position errors may occur as: (a) Radial error; and (b) Chord error, during motion for a parametric curve as shown in Figure 3.1. The radial error is the perpendicular distance between the interpolated points and the parametric curve. Basically, the radial error is caused by the rounding error of computer systems. With the rapid development of microprocessors with higher accuracy in computing, the radial error is no longer a major concern in present applications. The chord error is the maximum distance between the interpolated trajectory CD and the arc AB on the desired curve.

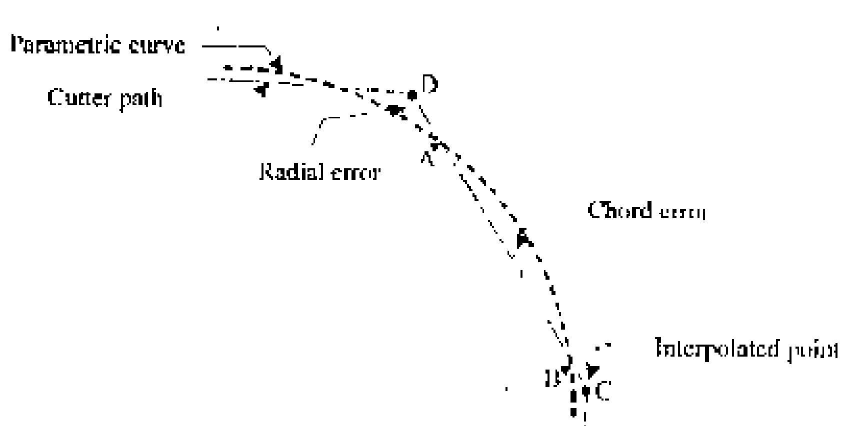

Figure 3.1: The radial and chordal errors.

As the sampling time, T, for a CNC machine's controller is constant, the chordal error depends on the machining feedrate and the radius of curvature of the parametric curve. It can be understood from Figure 3.1 that chordal error is higher if the radius of curvature of arc AB is smaller. So if the tool traverses along the curve with a constant feedrate, V, there will be high machining error on the segments of the curve where the radii of curvature are small. Again for a given curvature if the feedrate is high then the chordal distance between two adjacent tool positions is also high, which in turn results in high chordal error.

To estimate the chordal error, it is assumed that the curve segment between $\bar{P}(u_i)$ and $\bar{P}(u_{i+1})$ is a circular arc with radius ρ_i at parameter u_i, as shown in Figure 3.2.

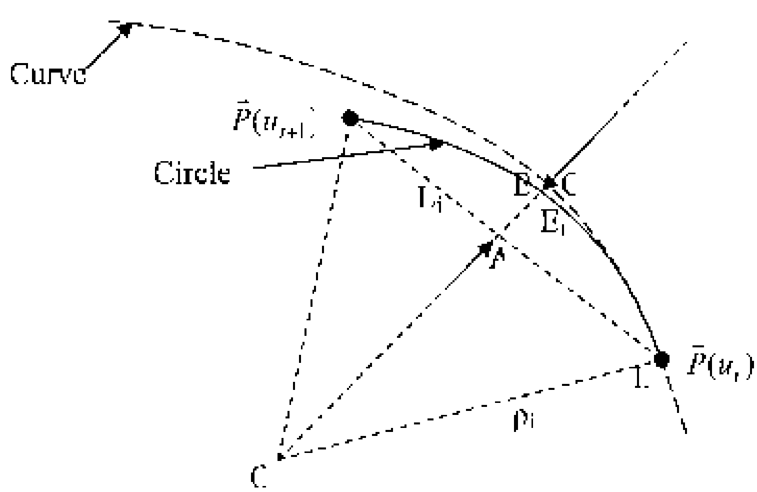

Figure 3.2: Esimation of chordal error.

The radius ρ_i at parameter u_i is found from the curvature, κ_i, at parameter u_i which is calculated by the following equation:

$$\kappa_i = \frac{\left| \vec{P}'(u_i) \times \vec{P}''(u_i) \right|}{\left| \vec{P}'(u_i) \right|^3}$$

$$
= \left[\begin{array}{l} \left(\frac{dx}{du}\Big|_{u=u_i} \frac{d^2y}{du^2}\Big|_{u=u_i} \right) - \left(\frac{dx}{du}\Big|_{u=u_i} \frac{d^2z}{du^2}\Big|_{u=u_i} \right) - \left(\frac{dy}{du}\Big|_{u=u_i} \frac{d^2x}{du^2}\Big|_{u=u_i} \right) \\ + \left(\frac{dy}{du}\Big|_{u=u_i} \frac{d^2z}{du^2}\Big|_{u=u_i} \right) + \left(\frac{dz}{du}\Big|_{u=u_i} \frac{d^2x}{du^2}\Big|_{u=u_i} \right) - \left(\frac{dz}{du}\Big|_{u=u_i} \frac{d^2y}{du^2}\Big|_{u=u_i} \right) \end{array} \right] \Bigg/ \left[\left(\frac{dx}{du}\Big|_{u=u_i} \right)^2 + \left(\frac{dy}{du}\Big|_{u=u_i} \right)^2 + \left(\frac{dz}{du}\Big|_{u=u_i} \right)^2 \right]^{3/2}
$$

.... (3.10)

The chordal distance, L_i, is approximated by $\left| \vec{P}(u_{i+1}) - \vec{P}(u_i) \right|$. So the chordal error,

$$E_i = OB - OA = OB - \sqrt{OD^2 - AD^2}$$

$$\text{or, } E_i = \rho_i - \sqrt{\rho_i^2 - \left(\frac{L_i}{2} \right)^2} = \frac{1}{\kappa_i} - \sqrt{\frac{1}{\kappa_i^2} - \left(\frac{VT}{2} \right)^2} = \frac{1}{\kappa_i} \left[1 - \sqrt{1 - \left(\frac{VT\kappa_i}{2} \right)^2} \right]$$

.... (3.11)

This approach of calculating chordal error neglects BC. Hence the actual error will be greater than that calculated from equation (3.11).

CHAPTER 4

THE PROPOSED METHOD OF CNC INTERPOLATION

The present method of finding successive tool positions utilizes the Taylor's Series Approximation to obtain the next parameter value, u_{i+1}, from the current value, u_i, and substituting this value in the mathematical expression of the parametric curve designed in the CAD/CAM system. It was explained in the previous section that due the omission of higher order terms of the Taylor's Series the actual feedrate between successive tool positions varies from the predetermined feedrate. Hence for constant feedrate interpolation, the feedrate variation is unacceptable. To reduce this variation the fourth-order Runge-Kutta (RK) method is proposed for finding the parameter values.

4.1 Runge-Kutta Method

The Fourth-order RK method is explained here. If we have an ordinary differential equation of the following form

$$\frac{dy}{dx} = f(x, y)$$

The solution [27] of the above according to RK method is:

$$y_{i+1} = y_i + \Phi(x_i, y_i, h)h \text{ . ...} \quad (4.1)$$

where, $h = x_{i+1} + x_i$, and $\Phi(x_i, y_i, h)$ is called the *increment function*, which can be interpreted as a representative slope over the interval. The increment function can be written in general form as

$$\Phi = a_1 K_1 + a_2 K_2 + \dots\dots + a_n K_n$$

Where the a's are constants and the K's are

$$K_1 = f(x_i, y_i)$$
$$K_2 = f(x_i + p_1 h, y_i + q_{11} K_1 h)$$
$$K_3 = f(x_i + p_2 h, y_i + q_{21} K_1 h + q_{22} K_2 h)$$

$$\vdots$$

$$K_n = f(x_i + p_{n-1} h, y_i + q_{n-1,1} K_1 h + q_{n-1,2} K_2 h + \dots\dots + q_{n-1,n-1} K_{n-1} h)$$

where, the p's and q's are constants For fourth order the value of n is equal to 4.

According to the Fourth-order RK method equation (12) is written as

$$y_{i+1} = y_i + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4)h,$$

where,

$$K_1 = f(x_i, y_i)$$
$$K_2 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_1 h\right)$$
$$K_3 = f\left(x_i + \frac{1}{2}h, y_i + \frac{1}{2}K_2 h\right)$$
$$K_2 = f(x_i + h, y_i + K_3 h)$$

Now from equation (3.3) we can write.

$$\frac{du}{dt} = \frac{V}{\left|\left(\frac{ds}{du}\right)\right|} = \frac{V}{\sqrt{\left[\frac{dx(u)}{du}\right]^2 + \left[\frac{dy(u)}{du}\right]^2 + \left[\frac{dz(u)}{du}\right]^2}} = f(u)$$

The right side of the above differential equation is a function of u only. So according to the Fourth-order RK method, the solution of the above is

$$u_{i+1} = u_i + \frac{1}{6}\left(K_1 + 2K_2 + 2K_3 + K_4\right)T \quad \dots\dots\dots\dots\dots\dots \text{(4.2)}$$

where,

$$K_1 = f(u_i) = \frac{V}{\sqrt{\left(\left.\frac{dx(u)}{du}\right|_{u=u_i}\right)^2 + \left(\left.\frac{dy(u)}{du}\right|_{u=u_i}\right)^2 + \left(\left.\frac{dz(u)}{du}\right|_{u=u_i}\right)^2}}$$

$$K_2 = f\left(u_i + \frac{K_1 T}{2}\right) = \frac{V}{\sqrt{\left(\left.\frac{dx(u)}{du}\right|_{u=u_i+\frac{K_1 T}{2}}\right)^2 + \left(\left.\frac{dy(u)}{du}\right|_{u=u_i+\frac{K_1 T}{2}}\right)^2 + \left(\left.\frac{dz(u)}{du}\right|_{u=u_i+\frac{K_1 T}{2}}\right)^2}}$$

$$K_3 = \frac{V}{\sqrt{\left(\left.\frac{dx(u)}{du}\right|_{u=u_i+\frac{K_2 T}{2}}\right)^2 + \left(\left.\frac{dy(u)}{du}\right|_{u=u_i+\frac{K_2 T}{2}}\right)^2 + \left(\left.\frac{dz(u)}{du}\right|_{u=u_i+\frac{K_2 T}{2}}\right)^2}}$$

And,

$$K_4 = \frac{V}{\sqrt{\left(\left.\frac{dx(u)}{du}\right|_{u=u_i+K_3 T}\right)^2 + \left(\left.\frac{dy(u)}{du}\right|_{u=u_i+K_3 T}\right)^2 + \left(\left.\frac{dz(u)}{du}\right|_{u=u_i+K_3 T}\right)^2}}$$

The RK method achieves the accuracy of a Taylor Series approach without requiring to calculate the higher derivatives. To achieve high accuracy from the Taylor's Series the higher derivatives are required. Current methods utilize the series up to the 2nd derivative. Computation of the derivatives makes the interpolator slower, which becomes significant in case of curves represented by B-Spline and NURBS. The RK method is also efficient for computer implementation because of the recurrence relationships of the K's. That is why the proposed method of interpolation is faster and generates more accurate parameter values compared to that generated from Taylor's Series Approximation. As a result, the actual feedrate between successive tool positions remains very close to the predetermined feedrate with negligible variations.

4.2 Determination of Chordal Error

The proposed interpolation utilizes a different method for finding the chordal error. The exact value of chordal error is the maximum distance between the straight line connecting two adjacent tool positions and the arc between them as shown in Figure 4.1.

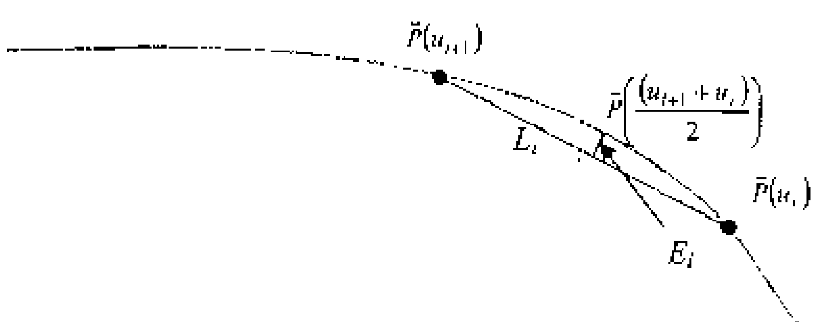

Figure 4.1: Estimation of chordal error on the point of average parameter.

The curve segment between points $\vec{P}(u_i)$ and $\vec{P}(u_{i+1})$ can be approximated by a circular arc. The point on the curve segment corresponding to the average value of the parameters u_i and u_{i+1} can be assumed to be the most distant point from the straight line $\vec{P}(u_{i+1})$ - $\vec{P}(u_i)$. Therefore, the chordal error, E_i, is the perpendicular distance from point $\vec{P}\left(\frac{u_{i+1}+u_i}{2}\right)$ to the line $\vec{P}(u_{i+1})$ - $\vec{P}(u_i)$. Mathematically,

$$E_i = \left|\left[\vec{P}\left(\frac{u_i+u_{i+1}}{2}\right) - \vec{P}(u_i)\right] \times \left[\frac{\vec{P}(u_{i+1}) - \vec{P}(u_i)}{\left|\vec{P}(u_{i+1}) - \vec{P}(u_i)\right|}\right]\right|$$

$$= \frac{1}{L_i}\left|\left(\vec{P}_{mi} - \vec{P}_i\right) \times \left(\vec{P}_{i+1} - \vec{P}_i\right)\right|.$$

$$= \frac{1}{L_i} \left\| \begin{matrix} \hat{i} & \hat{j} & \hat{k} \\ x_{mi} - x_i & y_{mi} - y_i & z_{mi} - z_i \\ x_{i+1} - x_i & y_{i+1} - y_i & z_{i+1} - z_i \end{matrix} \right\|$$

$$= \frac{1}{L_i} \left| \begin{matrix} [(y_{mi} - y_i)(z_{i+1} - z_i) - (z_{mi} - z_i)(y_{i+1} - y_i)]\hat{i} + [(z_{mi} - z_i)(x_{i+1} - x_i) - (x_{mi} - x_i)(z_{i+1} - z_i)]\hat{j} \\ + [(x_{mi} - x_i)(y_{i+1} - y_i) - (y_{mi} - y_i)(x_{i+1} - x_i)]\hat{k} \end{matrix} \right|$$

$$= \frac{1}{L_i} \sqrt{\left\{ \begin{matrix} [(y_{mi} - y_i)(z_{i+1} - z_i) - (z_{mi} - z_i)(y_{i+1} - y_i)]^2 \\ + [(y_{mi} - y_i)(z_{i+1} - z_i) - (z_{mi} - z_i)(y_{i+1} - y_i)]^2 \\ + [(y_{mi} - y_i)(z_{i+1} - z_i) - (z_{mi} - z_i)(y_{i+1} - y_i)]^2 \end{matrix} \right\}}$$

.... (4.3)

4.3 Variable Feedrate Interpolation

The objective of this research is to develop a variable feedrate interpolation scheme for CNC machine tool's controller. As was pointed out earlier that constant feedrate interpolation does not guarantee that the chordal error will be within the specified tolerance. Chordal error may exceed the specified tolerance if the radius of curvature of a curve segment is smaller. So an interpolation algorithm is necessary which generates variable feedrate to keep the chordal error within predetermined tolerance. The algorithm should check the magnitude of chordal error first. then depending on whether it is within tolerance the next tool position will be generated maintaining some relationship which is explained below.

Figure 4.2 shows how the chord length of a circle varies with the distance between the midpoint of that chord and that of the corresponding arc.

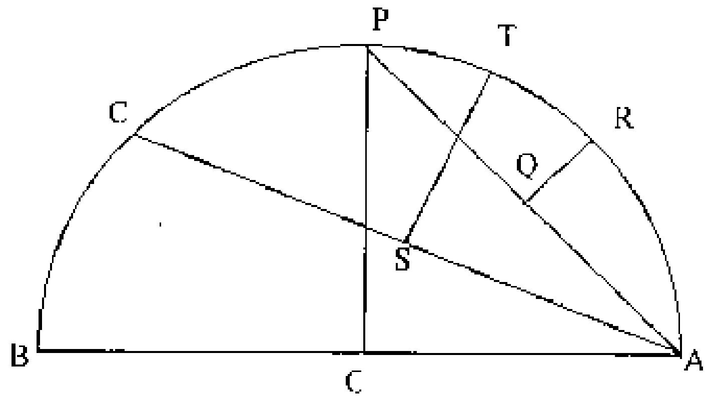

Figure 4.2: Variation of chordal error with chord length.

Let the radius of the circle is r, then

$$QR = OR - OQ = OR - \sqrt{(OA^2 - AQ^2)} = OR - \sqrt{(OA^2 - (AP/2)^2)}$$

$$= r - \sqrt{(r^2 - (AP/2)^2)} \quad \text{.......... (4 4)}$$

Similarly, it can be shown that,

$$ST = r - \sqrt{(r^2 - (AC/2)^2)} \quad \text{........................ (4.5)}$$

Now, if we consider the arc segment of the spline curve between successive tool position as a circular arc then in Figure 4.2, A is the current tool position, P or C is the next tool position, AP or AC is the tool trajectory and QR or ST is the chordal error. From equations (4.4) and (4.5), the general form can be expressed as:

$E_i = \rho_i - \sqrt{(\rho_i^2 - (L_i'/2)^2)}$, where ρ_i is the radius of curvature of the ith segment,
E_i is the chordal error, and
L_i' is the chord length.

or, $L_i' = \sqrt{(8 E_i \rho_i - 4 E_i^2)}$ .... (4.6)

If the current tool position is A and the next position, generated from the interpolation algorithm using the RK method, is such that the calculated chordal error is greater than the tolerance then the next tool position will not be that generated from RK method. To maintain the chordal error within tolerance the next position will be derived by using the relation of equation (4.7), where E_i will be replaced by tolerance. The resulting L_i' is the chord length between the corrected next tool position and the current position. To find the parameter u_{i+1} that will give the corrected next tool position, the algorithm uses $V_i = L_i' / T$.

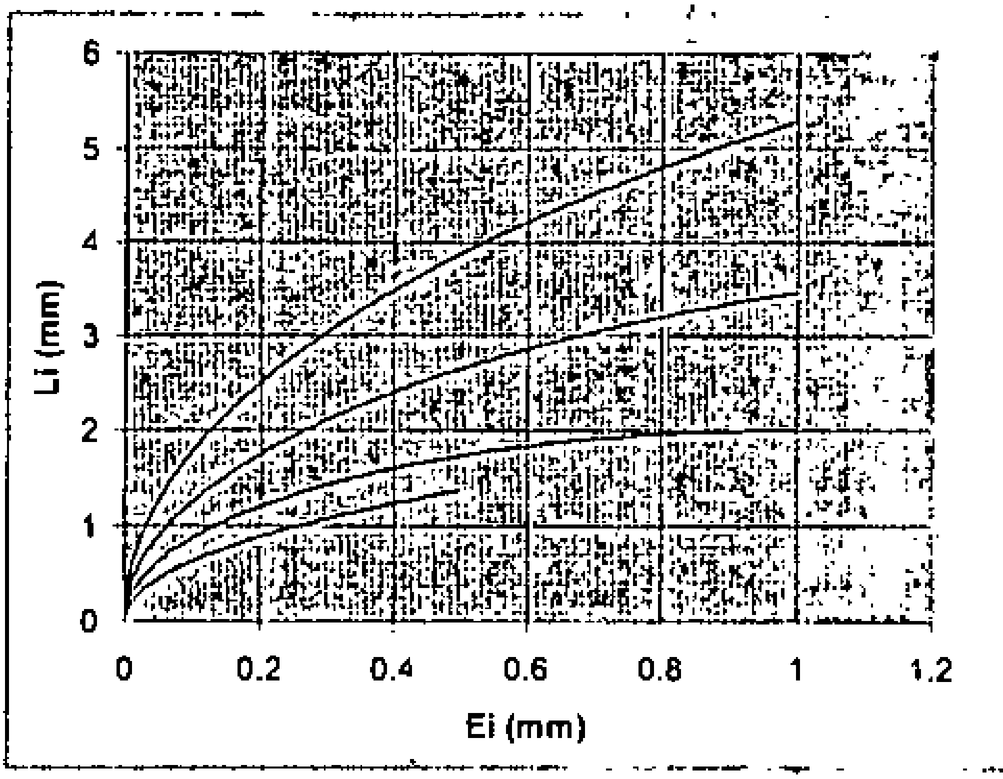

Figure 4.3: Relation between chord length and chord error for different values of radius of curvature.

The equation (3.11) needs the radius of curvature for the corresponding curve segment and requires several computation steps. In order to find a simple relationship between L_i and E_i , L_i is plotted for different values of ρ_i where the range of E_i is $0 \leq E_i \leq \rho_i$.

The chord error resulting from using the Taylor's Series Approximation or Runge-Kutta method hardly exceeds 0.1 mm, but the error tolerance can be much smaller than 0.1 mm. Now Figure 4.3 shows that for a given value of ρ_i, the relationship between L_i and E_i can be fairly approximated by a linear relationship for $0 \leq E_i \leq 0.1$ mm. Therefore, in this range of E_i , we can write

$$L_i / E_i = L_i' / \text{Tolerance}$$
$$\text{or, } L_i' = (L_i / E_i)^* \text{ Tolerance} \quad \text{.......(4.7)}$$

And the feedrate to be used by the RK method to find the next parameter u_{i+1} is

$$V_i = ((L_i / E_i)^* \text{ Tolerance}) / T \quad \text{.......(4.8)}$$

CHAPTER 5

THE B-SPLINE AND THE NURBS CURVE

5.1 B-spline Curve Evaluation

NURBS stands for Non-uniform Rational B-spline. Although B-spline curve is a special form of NURBS curve, the mathematical definition of NURBS curve is based on that of B-spline. So the mathematical definition of B-spline is presented first.

A B-spline curve of order k and having $n+1$ control points is expressed [26] as

$$\vec{P}(u)=\sum_{i=0}^{n}\vec{P}_i N_{i,k}(u) \qquad (5.1)$$

Where. $N_{i,k}(u)$'s are the k-th order basis functions derived from the following recursive formula

$$N_{i,k}(u)=\frac{u-t_i}{t_{i+k-1}-t_i}N_{i,k-1}(u)+\frac{t_{i+k}-u}{t_{i+k}-t_{i+1}}N_{i+1,k-1}(u) \qquad (5.2)$$

$$N_{i,1}(u) = \begin{cases}1, t_i \le u \le t_{i+1}\\ 0, elsewhere\end{cases} \qquad (5.3)$$

And t_i's are the knot values. $(n+k+1)$ knot values from t_0 to t_{n+k} need to be specified to define $(n+1)$ basis functions from $N_{0,k}$ to $N_{n,k}$.

If the B-spline curve has non-periodic knots, the curve passes through the first and the last control points. The non-periodic knots are determined from

$$t_i = \begin{cases} 0 & ,0 \le i < k \\ i-k+1 & ,k \le i \le n \\ n-k+2 & ,n < i \le n+k \end{cases} \qquad (5.4)$$

One of the most important properties of B-spline curve is that it has *local modification property*. A portion of the curve of a B-spline is affected by a finite number of the control points. Alternatively, a control point affects only a finite number of curve segments. Let the curve segment corresponding to the parameter values in the closed interval $[t_l, t_{l+1}]$ be considered. The control points that affect the shape of this segment would be those whose counterpart blending functions of order k are nonzero over the interval $[t_l, t_{l+1}]$. Among the blending functions of order 1, only $N_{l,1}(u)$ is nonzero over that interval. We obtain $N_{l,2}(u)$ by substituting $N_{l,1}(u)$ in the first term and $N_{l-1,2}(u)$ by substituting it in the second term of equation (5.2). In this way propagating the effects of $N_{l,2}(u)$ and $N_{l-1,2}(u)$, respectively, into the blending functions of order 3, and so on the blending functions of order k can be found. Finally, it can be shown that the blending functions $N_{l-k+1,k}(u)$, $N_{l-k+2,k}(u)$............ $N_{l,k}(u)$ have nonzero values over the interval $[t_l, t_{l+1}]$. Therefore the control points affecting the shape of this curve segment would be $\vec{P}_{l-k+1}, \vec{P}_{l-k+2}, \ldots\ldots\ldots\ldots, \vec{P}_l$. That is, only k control points affect the shape of this segment.

Cox-de Boor developed the algorithm for finding the coordinates on a B-spline curve for the parameter u in the range $t_l \le u \le t_{l+1}$, based on the property of B-spline curve explained above. Equation (5.1) can be expressed as follows:

$$\vec{P}(u) = \sum_{i=0}^{n} \vec{P}_i N_{i,k}(u) = \sum_{i=l-k+1}^{l} \vec{P}_i N_{i,k}(u) \quad \ldots\ldots\ldots\ldots\ldots (5.5)$$

Substituting $N_{i,k}(u)$ in equation (5.5) and arranging the summation the following expression can be derived:

$$\vec{P}(u) = \sum_{i=l-k+2}^{l} \left[\frac{u - t_i}{t_{i+k-1} - t_i} \vec{P}_i + \left(1 - \frac{u - t_i}{t_{i+k-1} - t_i}\right) \vec{P}_{i-1} \right] N_{i,k-1}$$

$$= \sum_{i=l-k+2}^{l} \vec{P}_i^{1} N_{i,k-1} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.6)$$

where $\vec{P}_i^{1}$ is defined as

$$\vec{P}_i^{1} = \frac{u-t_i}{t_{i+k-1}-t_i}\vec{P}_i + \left(1-\frac{u-t_i}{t_{i+k-1}-t_i}\right)\vec{P}_{i-1} \quad \text{.........} (5.7)$$

By repeating this procedure an arbitrary r times, we get

$$\vec{P}(u) = \sum_{i=l-k+r+1}^{l} \vec{P}_i^{r} N_{i,k-r}(u) \quad \text{...............} (5.8)$$

where,

$$\vec{P}_i^{r} = \frac{u-t_i}{t_{i+k-r}-t_i}\vec{P}_i^{r-1} + \left(1-\frac{u-t_i}{t_{i+k-r}-t_i}\right)\vec{P}_{i-1}^{r-1} \quad \text{..........} (5.9)$$

Substituting (k-1) for r in equation (5.8) gives

$$\vec{P}(u) = \vec{P}_l^{k-1} N_{l,1}(u) = \vec{P}_l^{k-1}$$

Therefore the coordinates of a point corresponding to a parameter value u in the range $t_l \le u \le t_{l+1}$ are those of $\vec{P}_l^{k-1}$, which is obtained by sectioning the line segments between the associated points recursively, starting from those between the original control points. The process of evaluating $\vec{P}_l^{k-1}$ using equation (5.9) iteratively can be illustrated schematically, as in Figure 5.1.

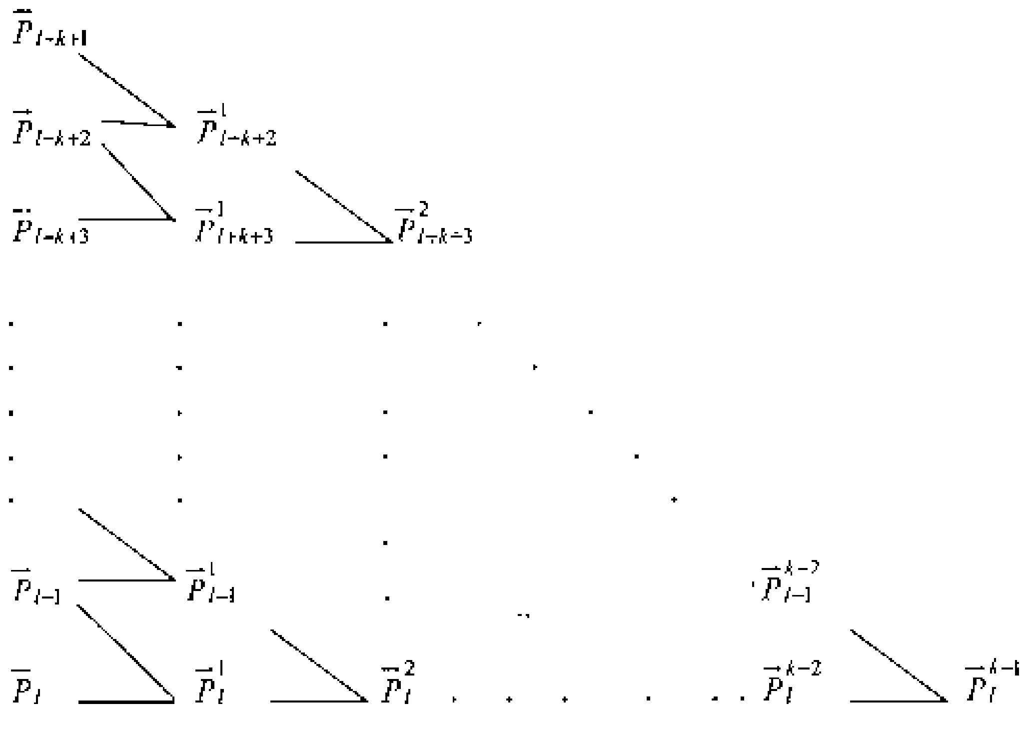

Figure 5.1: Cox-deBoor algorithm for evaluation of B-spline curve.

5.2 Derivatives of B-spline Curve

The first-order derivative of a B-spline curve is expressed in the form of a B-spline curve equation of 1 degree less than the original curve. The derivative of a B-spline curve, when u is in the interval $t_l \le u \le t_{l+1}$ is

$$\frac{d\vec{P}(u)}{du} = \sum_{i=l-k+2}^{l} \vec{P}_i{}' N_{i,k-1}(u) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.10)$$

$$\text{where, } \vec{P}_i{}' = (k-1)\frac{\vec{P}_i - \vec{P}_{i-1}}{t_{i+k-1} - t_i}$$

Substituting $N_{i,k-1}(u)$ in equation (5.10) and arranging the summation the following expression can be derived:

$$\frac{d\vec{P}(u)}{du} = \sum_{i=l-k+3}^{l}\left[\frac{u-t_i}{t_{i+k-2}-t_i}\left(\vec{P}_i{}'\right)+\left(1-\frac{u-t_i}{t_{i+k-2}-t_i}\right)\left(\vec{P}{}'_{i-1}\right)\right]N_{i,k-2}(u)$$

$$= \sum_{i=l-k+3}^{l}\left(\vec{P}_i{}'\right)^{1} N_{i,k-2}(u) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5.11)$$

$$\text{where, } \left(\vec{P}_i{}'\right)^{1} = \left[\frac{u-t_i}{t_{i+k-2}-t_i}\left(\vec{P}_i{}'\right)+\left(1-\frac{u-t_i}{t_{i+k-2}-t_i}\right)\left(\vec{P}{}'_{i-1}\right)\right]$$

By repeating this procedure an arbitrary r times, we get

$$\frac{d\vec{P}(u)}{du} = \sum_{i=l-k+2+r}^{l} \left(\vec{P}_i'\right)^r N_{i,k-1-r}(u) \qquad \ldots\ldots\ldots\ldots\ldots\ (5.12)$$

where, $$\left(\vec{P}_i'\right)^r = \left[\frac{u-t_i}{t_{i+k-r-1}-t_i}\left(\vec{P}_i'\right)^{r-1} + \left(1-\frac{u-t_i}{t_{i+k-r-1}-t_i}\right)\left(\vec{P}_{i-1}'\right)^{r-1}\right]$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ (5.13)$$

Substituting (k-2) for r in equation (5.12) gives

$$\frac{d\vec{P}(u)}{du} = \left(\vec{P}_l'\right)^{k-2} N_{l,1}(u) = \left(\vec{P}_l'\right)^{k-2}$$

The process of evaluating $\left(\vec{P}_l'\right)^{k-2}$ using equation (5.13) iteratively can be illustrated schematically, as in Figure 5.2.

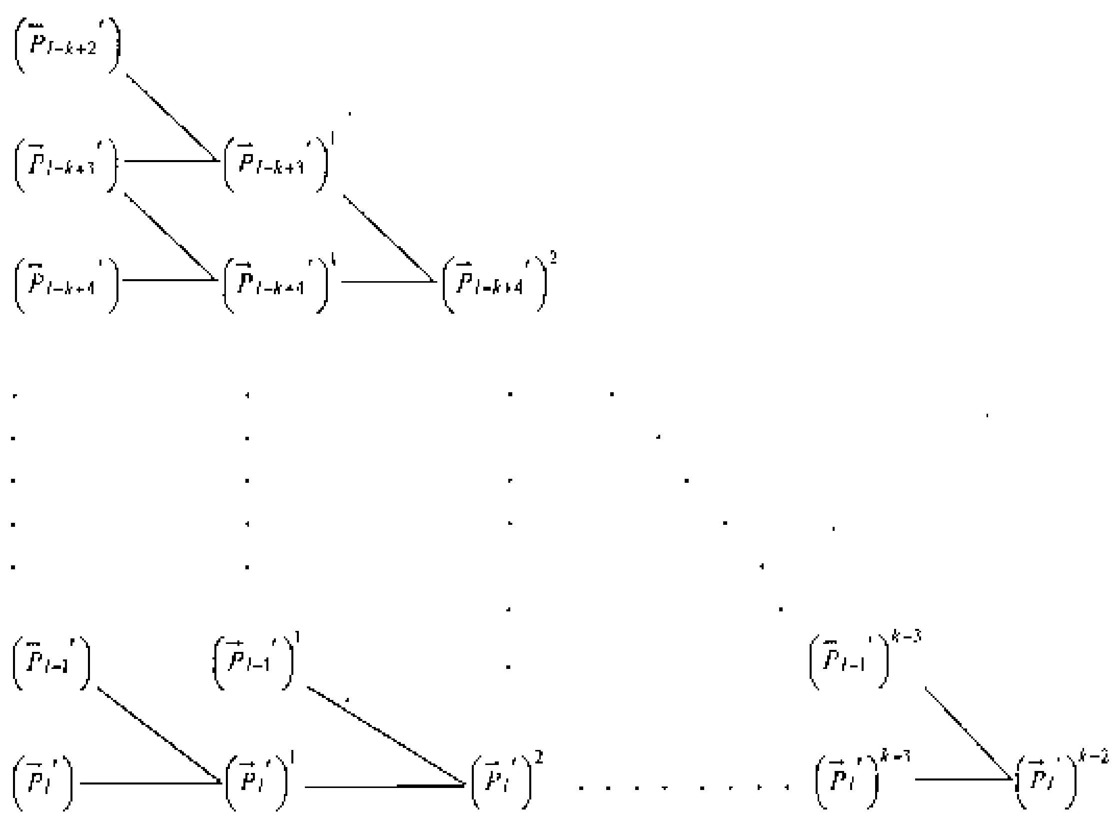

Figure 5.2: Cox-deBoor algorithm for evaluating 1st derivative of B-spline curve.

The second-order derivative of a B-spline curve, when u is in the interval $t_l \le u \le t_{l+1}$ is

$$\frac{d^2\vec{P}(u)}{du^2} = \sum_{i=l-k+3}^{l} \vec{P}_i{}'' N_{i,k-2}(u) \qquad \text{.............................. (5.14)}$$

$$\text{where, } \vec{P}_i'' = (k-2)\frac{\vec{P}_i{}' - \vec{P}_{i-1}{}'}{t_{i+k-2} - t_i} = \frac{(k-2)(k-1)}{t_{i+k-2} - t_i}\left[\frac{\vec{P}_i - \vec{P}_{i-1}}{t_{i+k-1} - t_i} - \frac{\vec{P}_{i-1} - \vec{P}_{i-2}}{t_{i+k-2} - t_{i-1}}\right]$$

Substituting $N_{i,k-2}(u)$ in equation (5.14) and arranging the summation the following expression can be derived:

$$\frac{d^2\vec{P}(u)}{du^2} = \sum_{i=l-k+4}^{l}\left[\frac{u-t_i}{t_{i+k-3} - t_i}\left(\vec{P}_i{}''\right) + \left(1 - \frac{u-t_i}{t_{i+k-3} - t_i}\right)\left(\vec{P}_{i-1}{}''\right)\right] N_{i,k-3}(u)$$

$$= \sum_{i=l-k+4}^{l}\left(\vec{P}_i{}''\right)^1 N_{i,k-3}(u) \qquad \text{.............................. (5.15)}$$

$$\text{where, } \left(\vec{P}_i{}''\right)^1 = \left[\frac{u-t_i}{t_{i+k-3} - t_i}\left(\vec{P}_i{}''\right) + \left(1 - \frac{u-t_i}{t_{i+k-3} - t_i}\right)\left(\vec{P}_{i-1}{}''\right)\right]$$

By repeating this procedure an arbitrary r times, we get

$$\frac{d^2\vec{P}(u)}{du^2} = \sum_{i=l-k+3+r}^{l}\left(\vec{P}_i{}''\right)^r N_{i,k-2-r}(u) \qquad \text{.............................. (5.16)}$$

$$\text{where, } \left(\vec{P}_i{}''\right)^r = \left[\frac{u-t_i}{t_{i+k-r-2} - t_i}\left(\vec{P}_i{}''\right)^{r-1} + \left(1 - \frac{u-t_i}{t_{i+k-r-2} - t_i}\right)\left(\vec{P}_{i-1}{}''\right)^{r-1}\right]$$

$$\text{........................ (5.17)}$$

Substituting (k-3) for r in equation (5.16) gives

$$\frac{d^2\vec{P}(u)}{du^2} = \left(\vec{P}_l{}''\right)^{k-3} N_{l,1}(u) = \left(\vec{P}_l{}''\right)^{k-3}$$

The process of evaluating $\left(\vec{P}_l''\right)^{k-3}$ using equation (5.17) iteratively can be illustrated schematically, as in Figure 5.3.

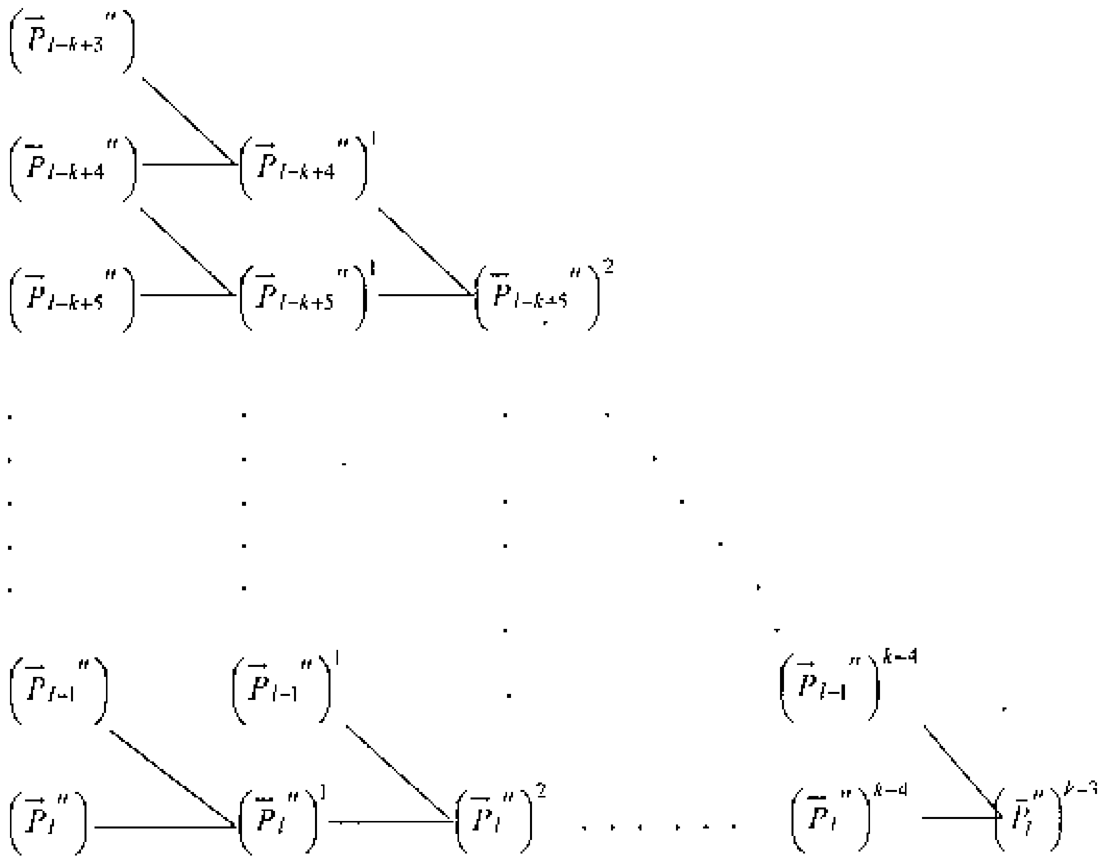

Figure 5.3: Cox-de Boor algorithm for evaluating 2[nd] derivative of B-spline curve.

5.3 NURBS Curve Evaluation

A Non-uniform Rational B-spline (NURBS) curve of order k and having $n+1$ control points is expressed [26] as

$$\vec{P}(u) = \frac{\sum_{i=0}^{n} h_i \vec{P}_i N_{i,k}(u)}{\sum_{i=0}^{n} h_i N_{i,k}(u)} \qquad \text{.................................... (5.18)}$$

Where, $N_{i,k}(u)$'s are the k-th order basis functions derived from equations (5.1) and (5.2) and the h_i's are the weights of each control point.

The coordinates of points on NURBS curve can be evaluated by Cox-de Boor algorithm similarly as for a B-spline curve. When the numerator of equation (5.18) is evaluated the $h_i\vec{P}_i$ have the role of the control points $\vec{P}_i$ in the Cox-de Boor algorithm of finding points on the B-spline curve and for the denominator the h_i play the role of $\vec{P}_i$.

5.4 Derivatives of NURBS Curve

The first-order derivative of a NURBS curve can be obtained by differentiating equation (5.18) with respect to u as follows:

$$\frac{d\vec{P}(u)}{du}=\frac{\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)\frac{d}{du}\left[\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right]-\left(\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right)\frac{d}{du}\left[\sum_{i=0}^{n}h_iN_{i,k}(u)\right]}{\left[\sum_{i=0}^{n}h_iN_{i,k}(u)\right]^2}$$

.... (5.19)

The second-order derivative of a NURBS curve can be found by differentiating equation (5.19) with respect to u as follows:

$$\frac{d^2\vec{P}(u)}{du^2}=\frac{\begin{bmatrix}\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)^3\frac{d^2}{du^2}\left(\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right)-\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)^2\left(\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right)\frac{d^2}{du^2}\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)\\ -2\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)^2\frac{d}{du}\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)\frac{d}{du}\left(\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right)\\ +2\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)\left(\sum_{i=0}^{n}h_i\vec{P}_iN_{i,k}(u)\right)\left[\frac{d}{du}\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)\right]^2\end{bmatrix}}{\left(\sum_{i=0}^{n}h_iN_{i,k}(u)\right)^4}$$

.... (5.20)

In equations (5.19) and (5.20), $\frac{d}{du}\left[\sum_{i=0}^{n} h_i N_{i,k}(u)\right]$ and $\frac{d}{du}\left[\sum_{i=0}^{n} h_i \vec{P}_i N_{i,k}(u)\right]$ are found from the Cox-deBoor algorithm of evaluating 1st derivative of B-spline curve substituting h_i and h_iP_i for P_i respectively.

Similarly, $\frac{d^2}{du^2}\left(\sum_{i=0}^{n} h_i N_{i,k}(u)\right)$ and $\frac{d^2}{du^2}\left(\sum_{i=0}^{n} h_i \vec{P}_i N_{i,k}(u)\right)$ are found from the Cox-deBoor algorithm of evaluating 2nd derivative of B-spline curve substituting h_i and h_iP_i for P_i respectively.

SIMULATION RESULTS

6.1 Algorithms

The proposed variable feedrate interpolation employs the fourth-order Runge-Kutta (RK) method to generate the successive values of the parameter. If the chordal error, calculated at the average value of the current and next parameter, is less than the specified tolerance, the next tool position is found by using the next parameter and the tool traverses at the specified feedrate. If the chordal error exceeds the tolerance, the chord length that would result in error within tolerance is calculated and the resulting feedrate is utilized by the RK method to generate the next corrected parameter value. This algorithm is tested for feasibility and performance by writing the interpolator in MATLAB 6.5 and performing simulations on a personal computer with 2.41 GHz CPU on different types of parametric spline curves, such as: a 3-D cubic spline, a 2-D profile of a cam defined by B-spline curve, a 3-D profile of a fan defined by B-spline curve and a 2-D NURBS curve

The algorithm of interpolation, when constant feedrate and 2nd order Taylor's Series Approximation or 4th order RK method is used for generating successive values of the parameter and coordinates of tool position, is shown by the flowchart in Figure 6.1.

To illustrate the superiority of the proposed interpolator results in terms of chordal error and actual feedrate are compared with those using the present method. The algorithm for calculating actual feedrate and chordal error (by estimating curvature), when using 2nd order Taylor's Series Approximation with constant feedrate, is shown by the flowchart in Figure 6.2.

The algorithm for calculating actual feedrate and chordal error (by using the average parameter value), when using RK method with constant feedrate, is shown by the flowchart in Figure 6.3. Finally the proposed variable feedrate interpolation algorithm is shown by the flowchart in Figure 6.4.

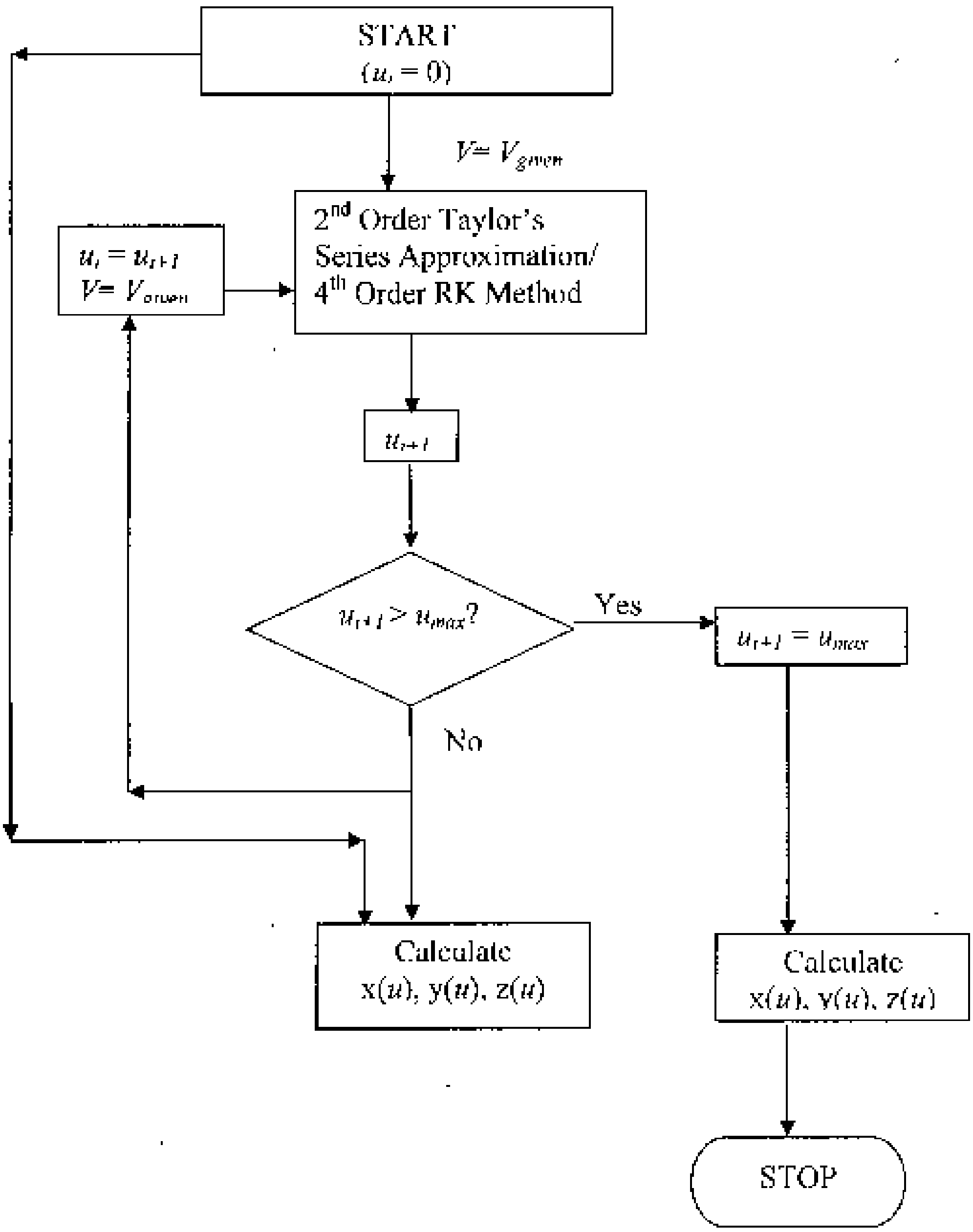

Figure 6.1: Flowchart of the algorithm for generating successive tool positions for constant feedrate Taylor's series approximation or Runge-Kutta method.

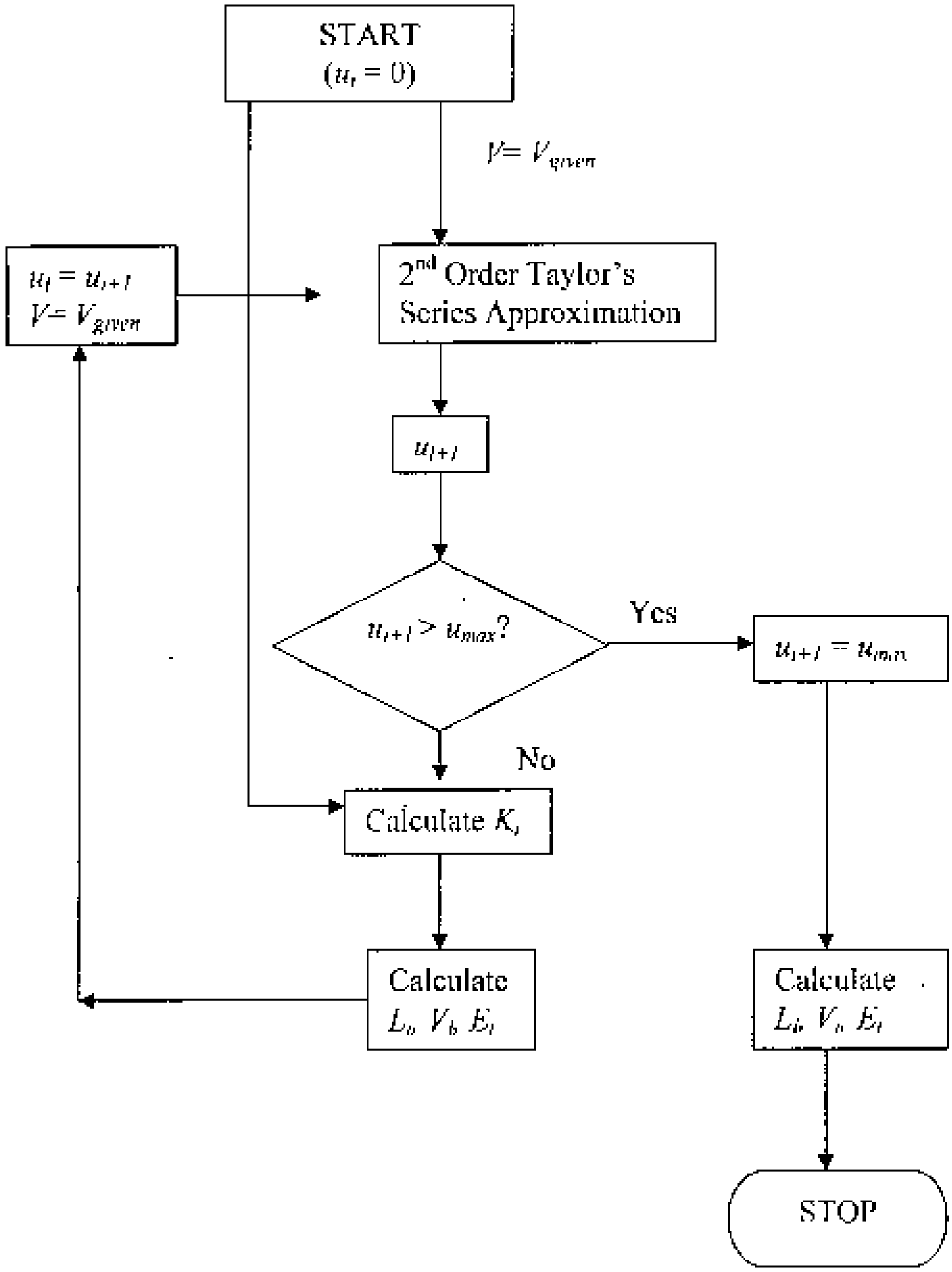

Figure 6.2: Flowchart of the algorithm for calculating actual feedrate and chordal error by using constant feedrate and Taylor's series approximation.

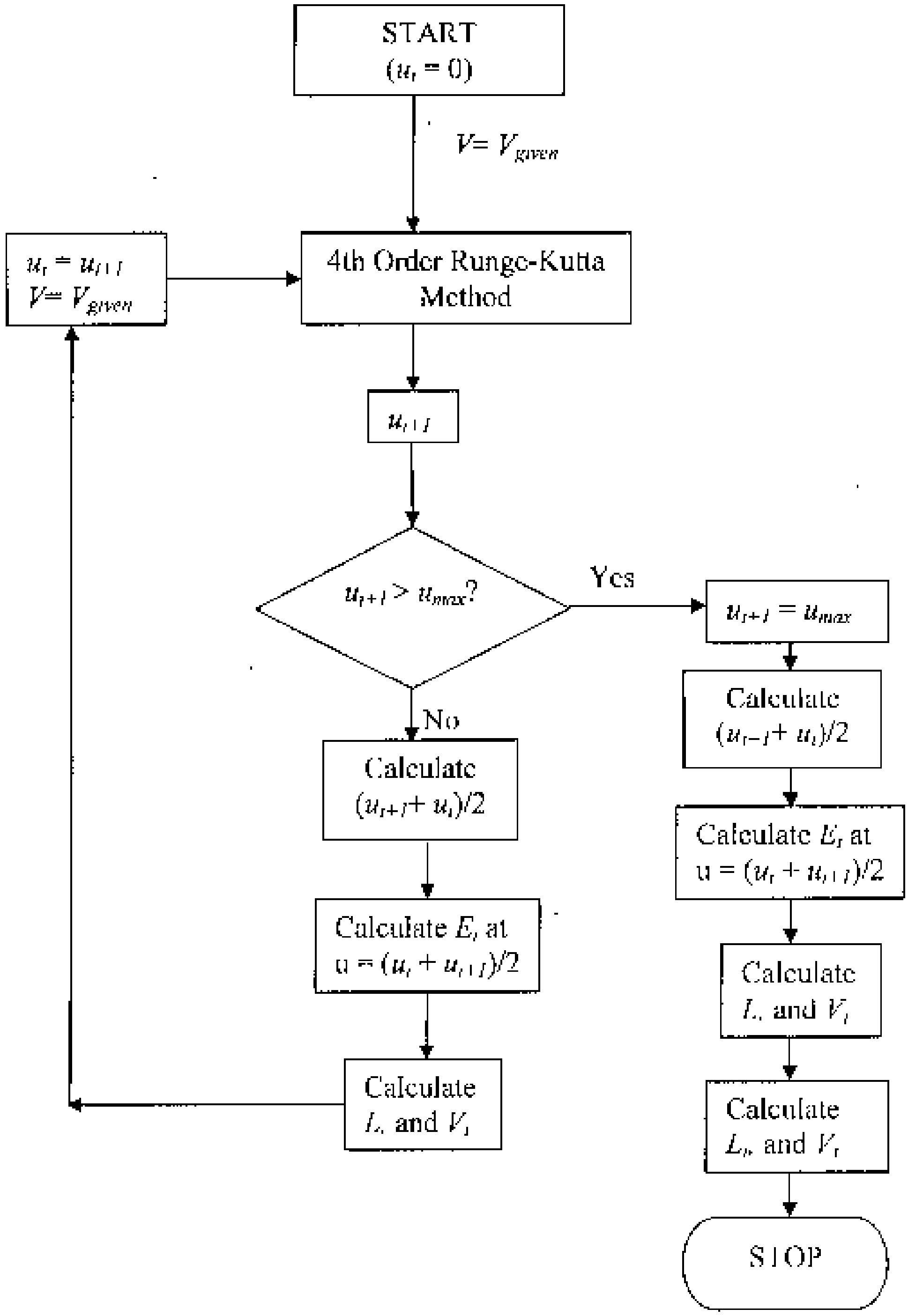

Figure 6.3: Flowchart of the algorithm for calculating actual feedrate and chordal error by using constant feedrate and Runge-Kutta method.

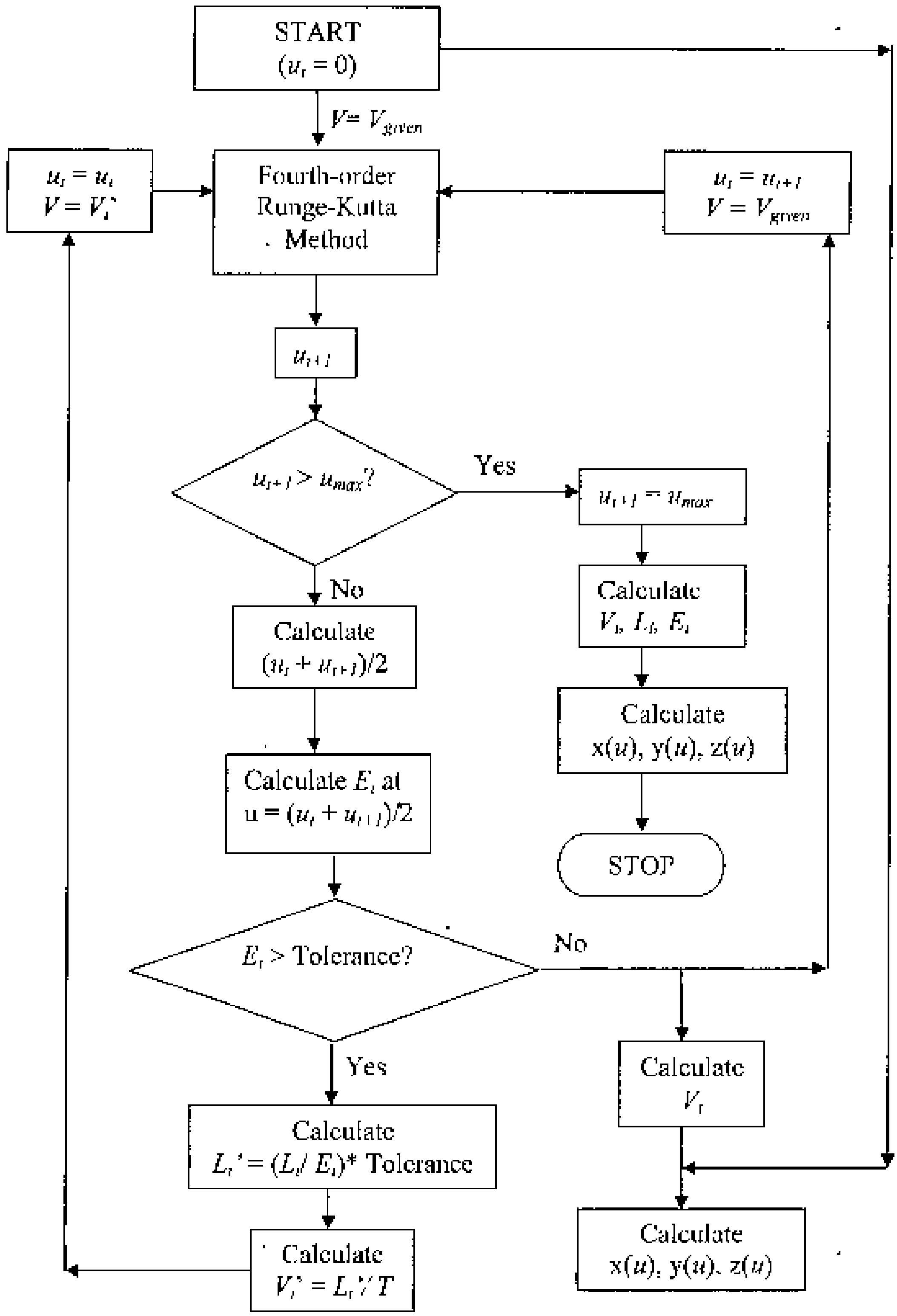

Figure 6.4: Flowchart of the variable feedrate interpolation algorithm using the 4th order Runge-Kutta method.

6.2 Results and Discussion

This section discusses the simulation results of different interpolators in terms of the generated tool positions, chordal error and feedrate for the parametric curves mentioned in the previous section. First, the chordal error and feedrate are compared between the interpolation using the Taylor's approximation and that using the RK method for constant feedrate mode. Then the chordal error, feedrate and tool positions generated from the variable feedrate interpolation are shown.

Figure 6.5(a) shows the cubic spline curve, for which the control points are given in Table 6.1 and the mathematical expression is:

$$\begin{aligned} x &= 50u^3-100u^2+50u \\ y &= 12u^3+35u^2-55u, \qquad\qquad 0 \le u \le 1. \\ z &= -10u^3+12u^2-4u \end{aligned}$$

Table 6.1: Control points of the cubic spline curve.

No.	1	2	3	4
x	0	50	-100	50
y	0	-55	35	12
z	0	-4	12	-10

Simulation results for the cubic spline are shown in Figure 6.5(b). (c), (d), (e) and (f). For constant feedrate interpolation, the Taylor's series approximation results in larger chordal error as compared to that of RK method for the cubic spline curve as shown in Figure 6.5(b). Error increases following the same pattern for both methods The curvature of the curve starts increasing for parameter values greater than 0.2 and maximum curvature corresponds to parameter values between 0.5 and 0.6. The maximum error results from the curve segment where the curvature is maximum. As in Taylor's approximation the error is calculated based on the estimation of the curvature for a given segment and this neglects a small portion of error, the actual error is greater than those shown in Figure 6.5(b). For a given segment the maximum chordal error is very difficult to calculate analytically and would be infeasible for

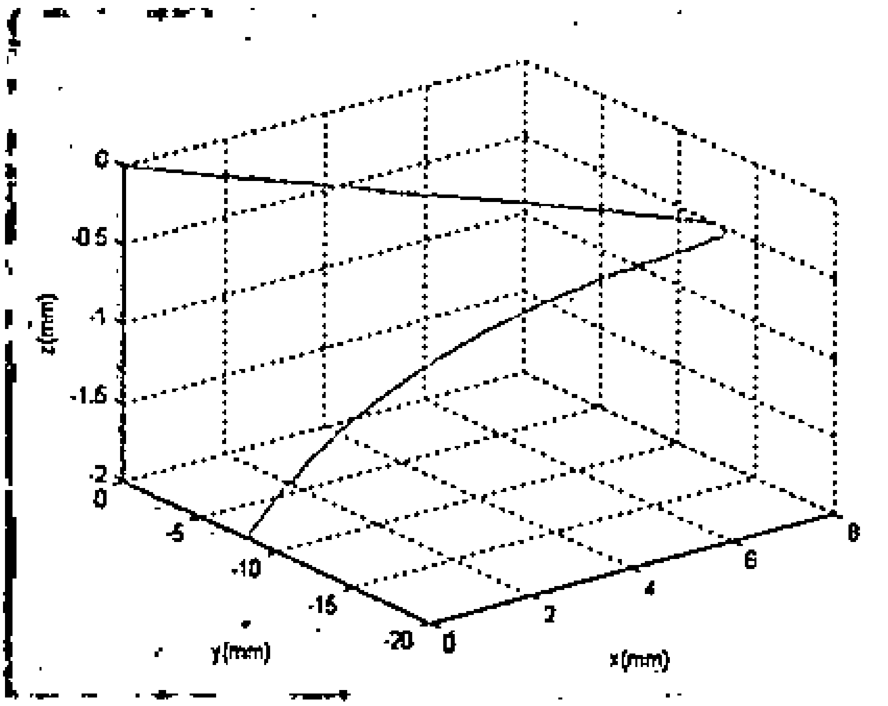

Figure 6.5(a): 3-D view of the cubic spline curve.

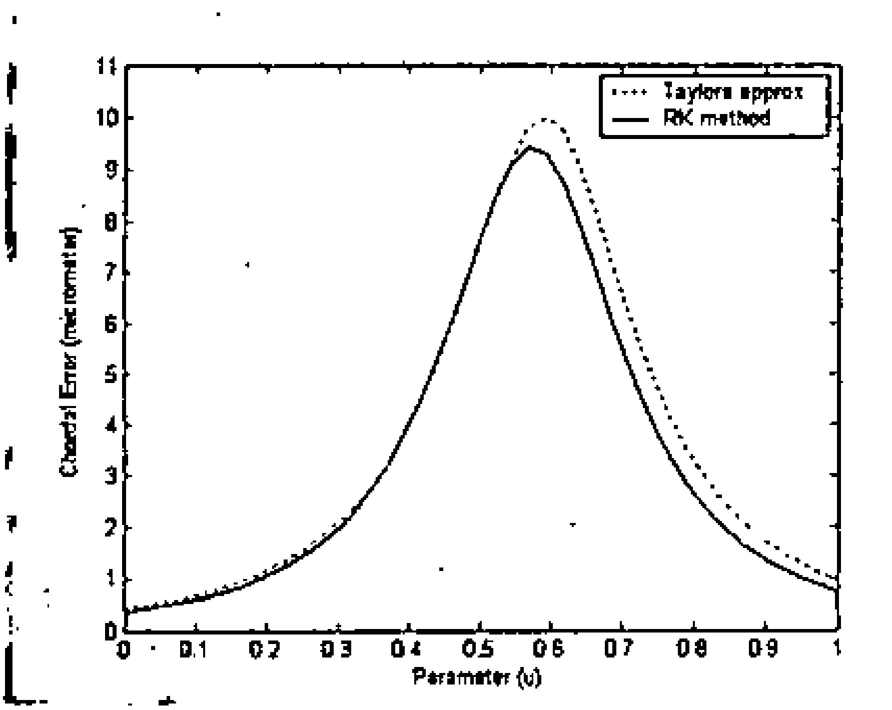

Figure 6.5(b): Chordal error from constant feedrate (200mm/sec) interpolation for the cubic spline.

implementing into the real-time interpolator. That is why to better estimate the error RK method utilizes calculation of error at the average parameter value of a given segment.

The RK method of interpolation also results in better feedrate profile compared to that from the Taylor's approx. for constant feedrate mode as shown in Figure 6.5(c). The fluctuation of feedrate is greater for latter method where the maximum deviation is about 0.23%. For the RK method the maximum feedrate deviation is 0.15%. The reason for this variation in actual feedrate can be explained by remembering that both the methods utilizes an approximation to generate successive parameter values and the tool travels the distance between two adjacent positions in a sampling time, which is constant for a machine's controller. So the chordal length between the generated next position and the current position varies depending on whether the tool is passing through a segment of high curvature or low curvature. As RK method gives better accuracy of generating successive parameters, the variation in chordal length is less than that of the Taylor's approximation method.

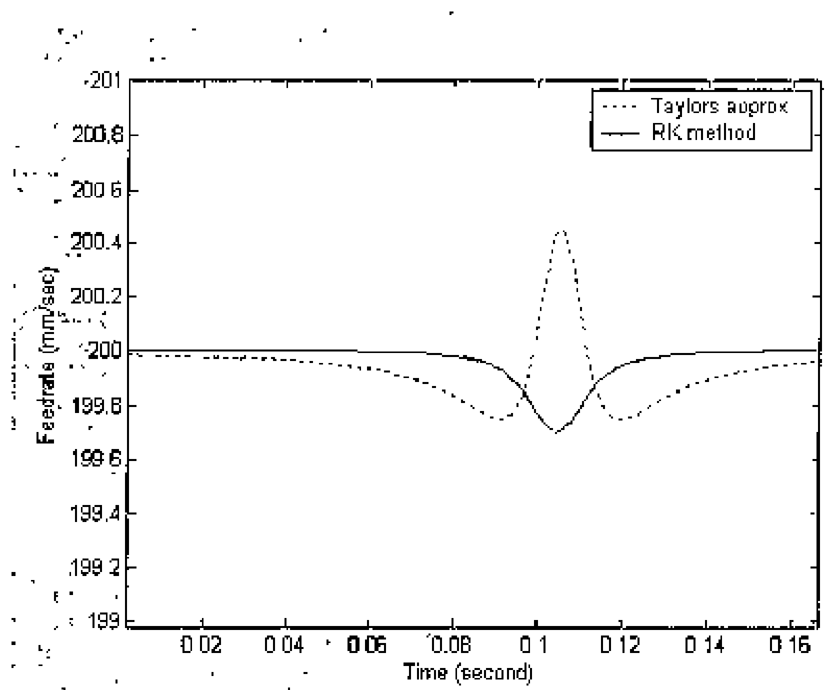

Figure 6.5(c): Actual feedrate from constant feedrate (200mm/sec) interpolation for the cubic spline curve

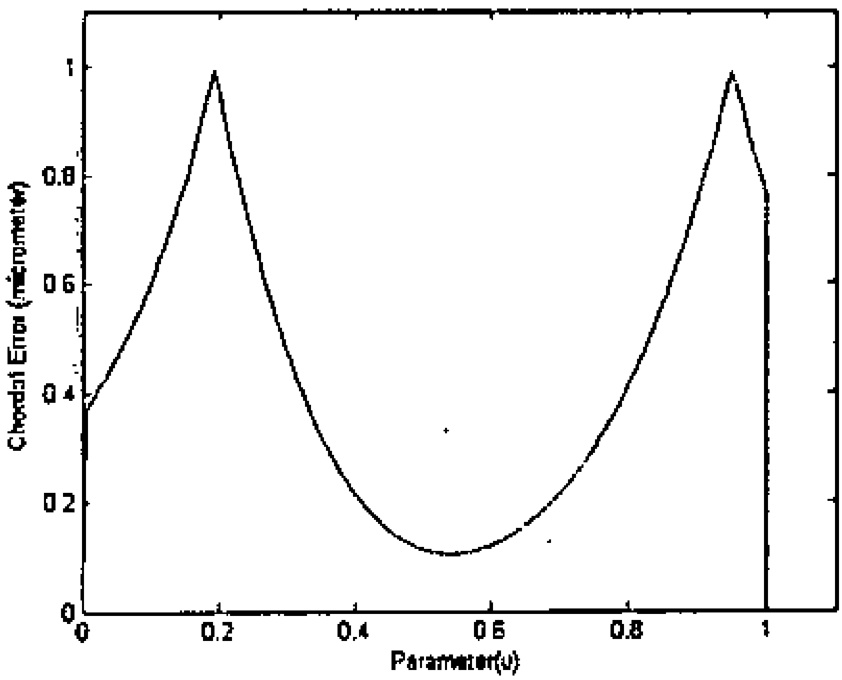

Figure 6.5(d): Chordal error from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the cubic spline curve.

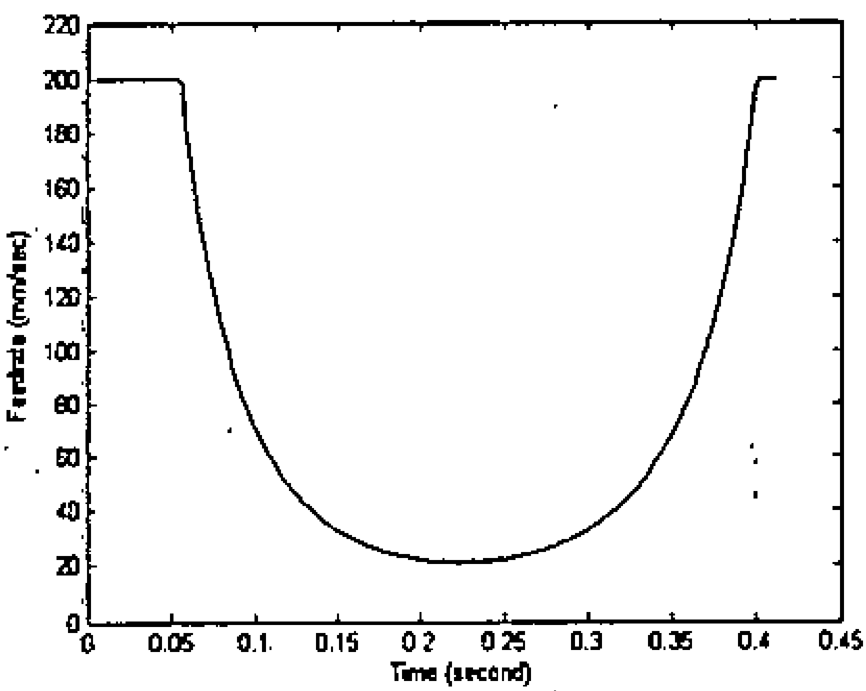

Figure 6.5(e): Actual feedrate from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the cubic spline curve.

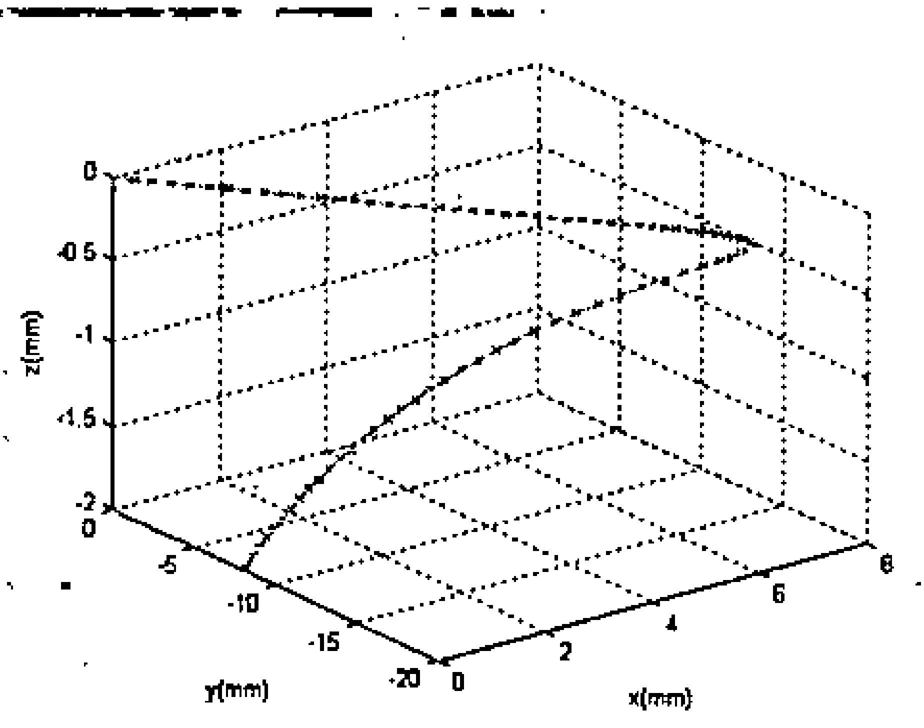

Figure 6.5(f): Tool positions generated from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the cubic spline curve.

Figure 6.5(d), (e) and (f) shows the results of proposed variable feedrate interpolation for the cubic spline curve. The chordal error is within the tolerance of 1μm. There are two positions where there is close to the tolerance, the maximum value is 0.991 μm. The reason for this is the algorithm of variable feedrate interpolation. The algorithm generates the next position command if the chordal error is less than 1μm, otherwise it recalculates the next position. The portion of the curve having low curvature results in error close to the tolerance.

The feedrate profile resulting from the variable feedrate interpolation is as expected. The portion of the curve having low curvature results error less than 1μm and so the tool moves with the specified feedrate of 200 mm/sec. For the segments of the curve resulting error greater than tolerance, the interpolator adjusts the feedrate so that chordal lengths are very small and the tool traverses these small distances at constant time (i.e. sampling time). The variation in feedrate is also smooth resulting in smooth acceleration and deceleration of the tool.

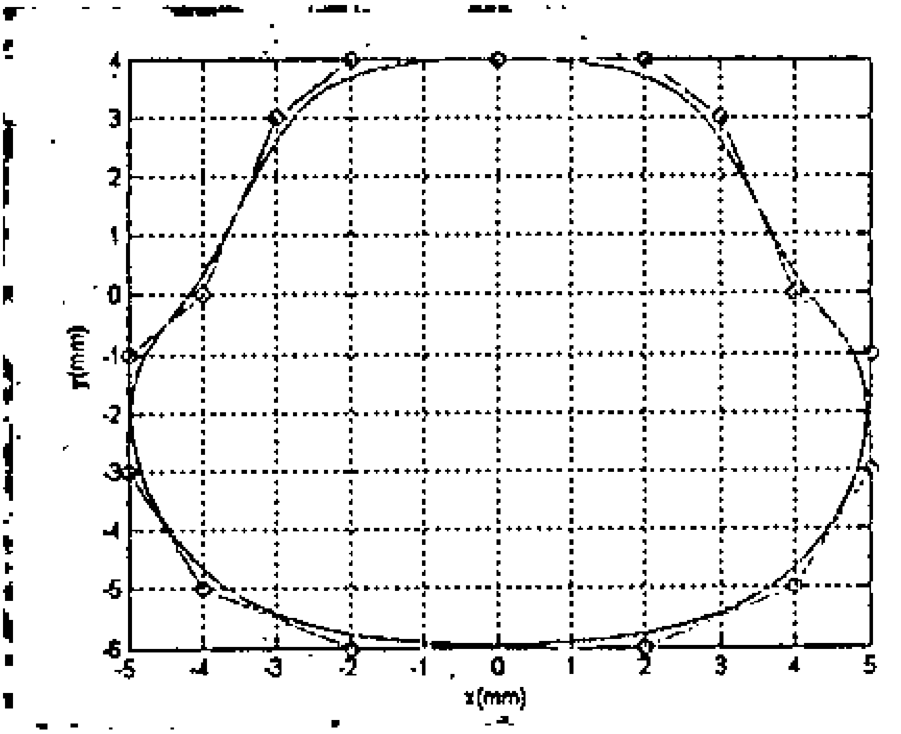

Figure 6.6(a): The cam profile.

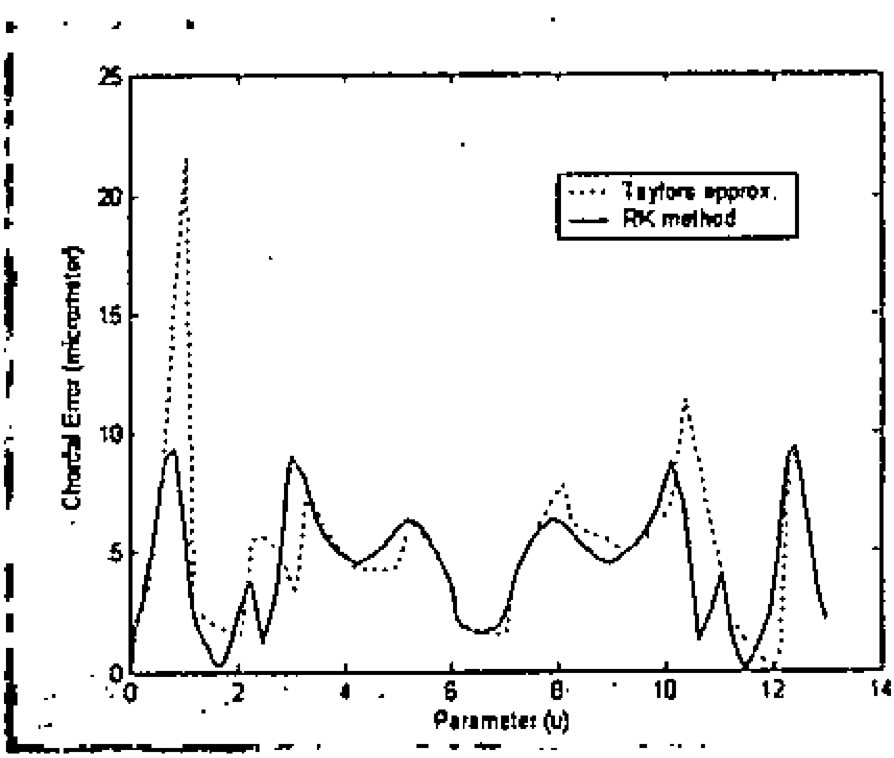

Figure 6.6(b): Chordal error from constant feedrate (200mm/sec) interpolation for the cam (B-spline curve).

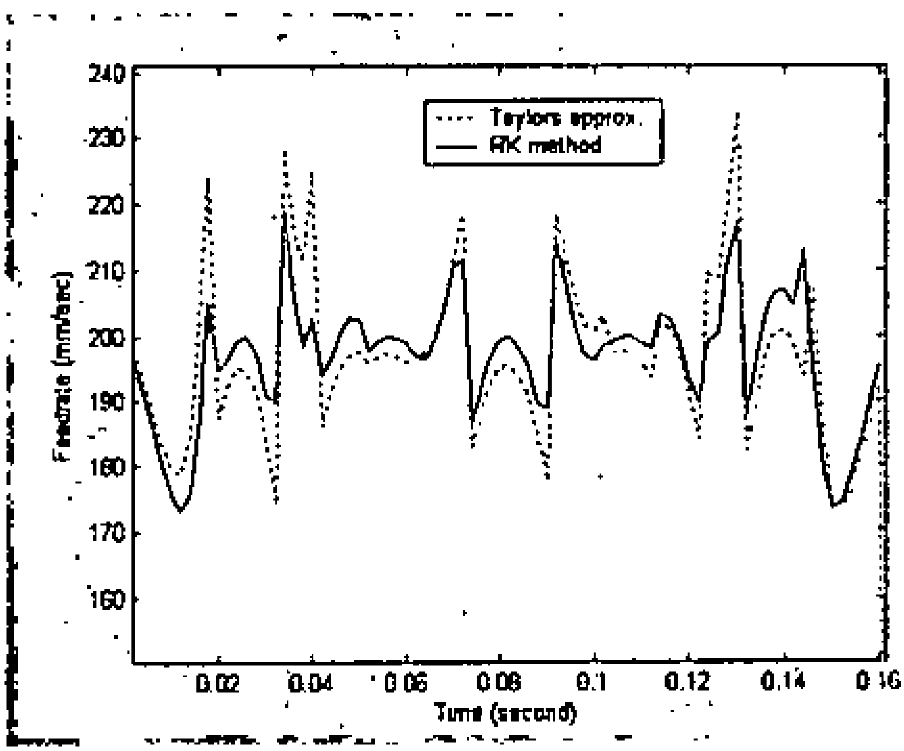

Figure 6.6(c): Actual feedrate from constant feedrate (200mm/sec) interpolation for the cam (B-spline curve).

Figure 6.6(a) shows the cam profile described by a B-spline curve of order 4 having the knot vector: t = [0,0,0,0,1,2,3,4,5,6,7,8,9,10,11,12,13,13,13,13] and control points as shown in Table 6.2.

Table 6.2: Control points of the cam

No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
x	0	2	3	4	5	5	4	2	-2	-4	-5	-5	-4	-3	-2	0
y	4	4	3	0	-1	-3	-5	-6	-6	-5	-3	-1	0	3	4	4
z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6.6 (b) and (c) shows the simulation results of constant feedrate interpolation for the cam profile. As can be seen in Figure 6.6(b), the average value of chordal error resulting from Taylor's approximation is greater than that from RK method. The maximum chordal error resulting from the former method and is 21.6 μm and from the latter method is 9.4 μm. The deviation of actual feedrate from the specified value of 200 mm/sec is greater from Taylor's approximation than that from the RK method

as shown in Figure 6.6(c). The maximum deviation is 61% for the former method and 13% for the latter. As feedrate affects the machining efficiency, surface quality and the dynamics of the machine tool and RK method generates less fluctuation in feedrate, it is better in those respects compared to the Taylor's approximation. So this explains why using the RK method for parametric interpolation not only provides computational efficiency but also provides more accurate feedrate profile.

The chordal errors resulting from the variable feedrate interpolation for the cam profile are shown in Figure 6.6(d). The maximum error is 0.99 μm which is within the tolerance. As the cam profile has mainly four sections of high curvature, the figure shows four regions of high error although within the tolerance. In all other sections of the curve error is of much lower value as those sections have higher radii of curvature.

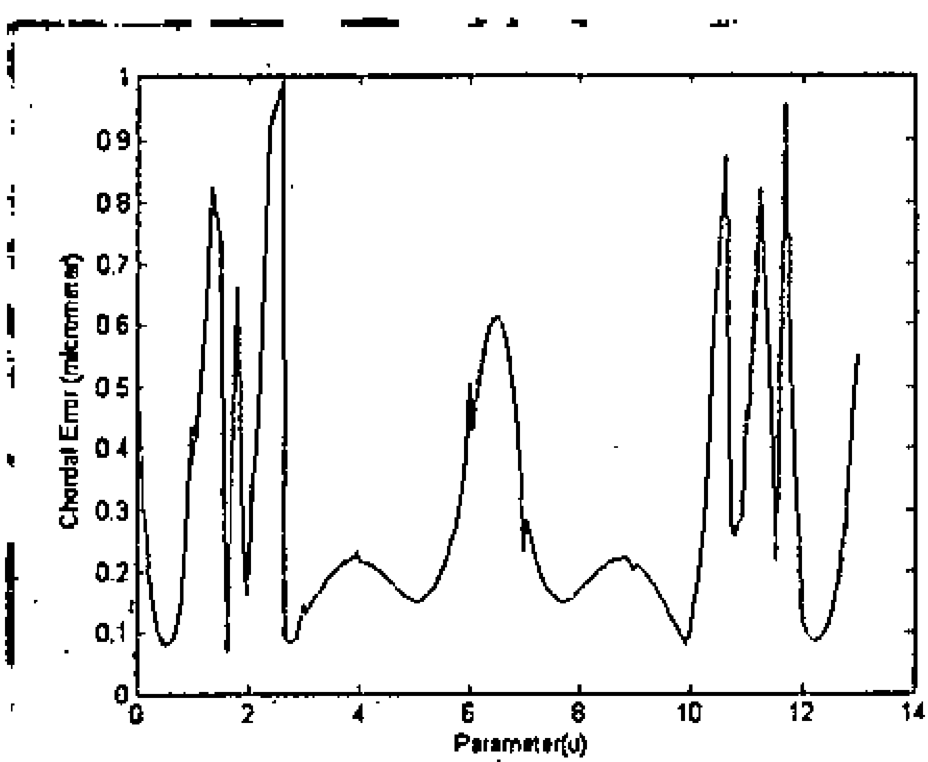

Figure 6.6(d): Chordal error from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the cam (B-spline curve).

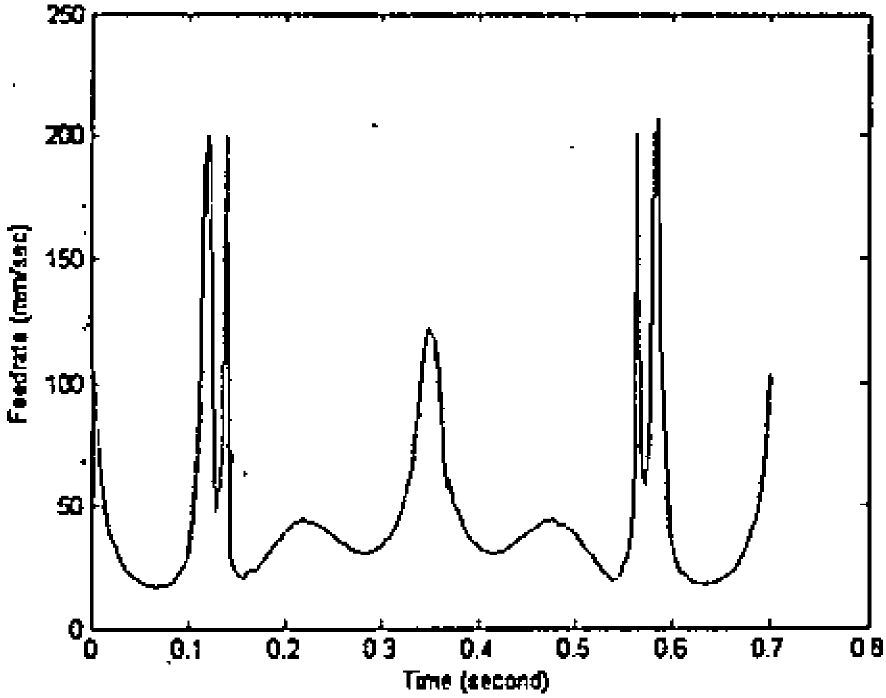

Figure 6.6(e): Actual feedrate from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the cam (B-spline curve).

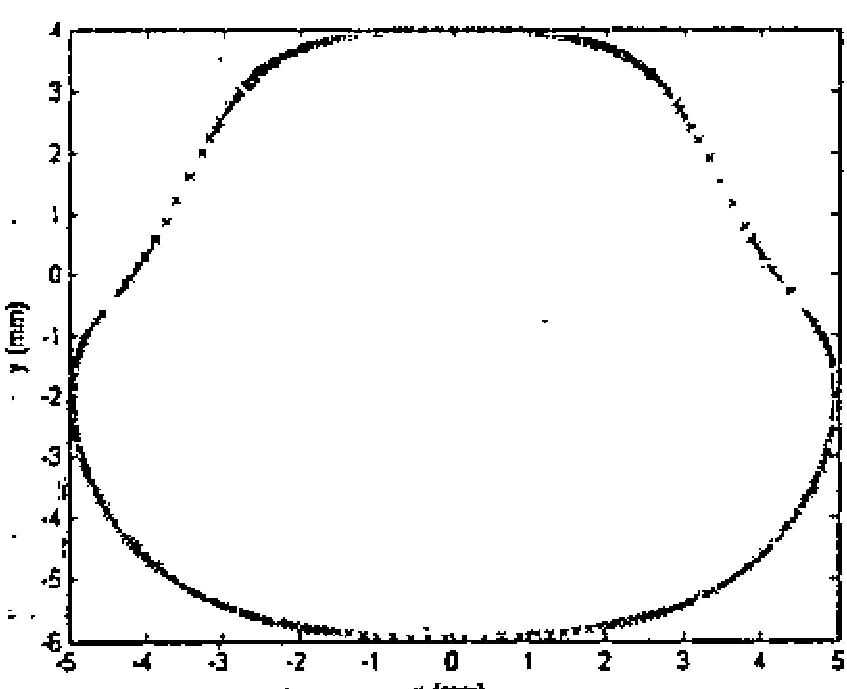

Figure 6.6(f): Tool positions generated from the variable feedrate interpolation for the cam profile (B-spline curve).

Actual feedrate resulting from the variable feedrate interpolation for the cam profile is shown in Figure 6.6(c). Depending on the curvature at different segments of the curve the interpolator reduces the feedrate so that chordal error remains less than the tolerance. The maximum and minimum values of actual feedrate are 207 m/sec and 17 mm/sec respectively. As can be found from the Figure 6.6(a) the curve has low curvature at the top and bottom of the plot. The actual feedrate increases at these segments. The surface quality will be as desired because the feedrate does not exceed the specified value by much. The change of feedrate is smooth which would result in smooth acceleration/deceleration of the tool.

The 3-D fan profile, shown in Figure 6.7(a), represented by a B-spline curve of order 4 has the knot vector,

t = [0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 34, 34, 34],

And the control points as shown in Table 6.3.

Table 6.3: Control points of the fan

No.	1	2	3	4	5	6	7	8	9	10	11	12
x	0	40	60	40	20	20	40	60	100	100	100	80
y	100	100	80	60	40	20	0	20	20	0	-40	-60
z	0	5	10	5	5	0	-5	-10	-5	0	5	10
No.	13	14	15	16	17	18	19	20	21	22	23	24
x	60	40	20	0	20	20	0	-40	-60	-40	-20	-20
y	-40	-20	-20	-40	-60	-100	-100	-100	-80	-60	-40	-20
z	5	5	0	-5	-10	-5	0	5	10	5	5	0
No.	25	26	27	28	29	30	31	32	33	34	35	36
x	-40	-60	-100	-100	-100	-80	-60	-40	-20	0	-20	-20
y	0	-20	-20	0	40	60	40	20	20	40	60	100
z	-5	-10	-5	0	5	10	5	5	0	-5	-10	-5
No.	37											
x	0											
y	100											
z	0											

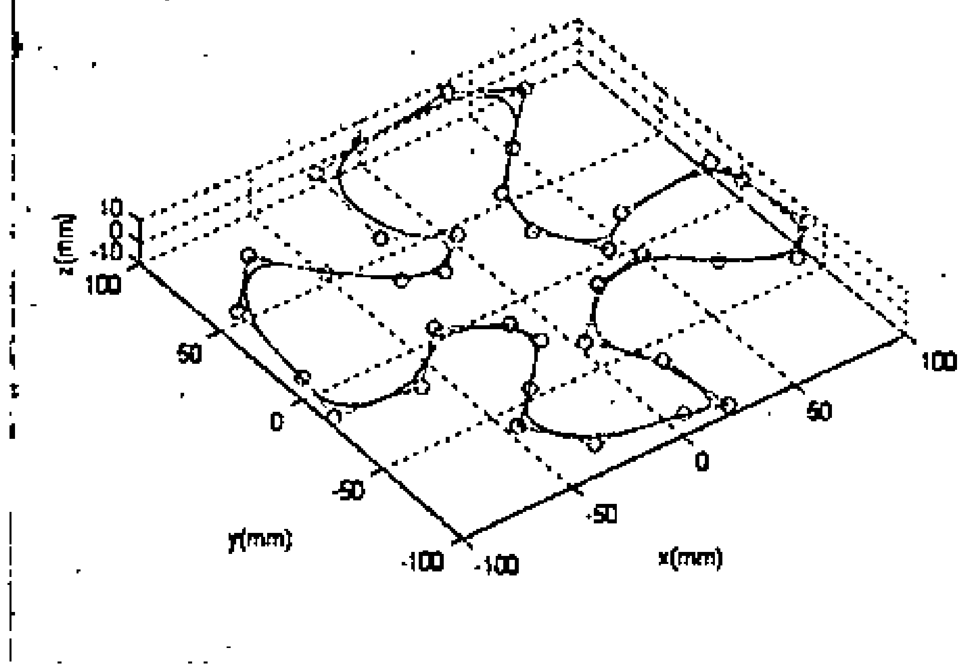

Figure 6.7(a): 3-D view of the fan profile.

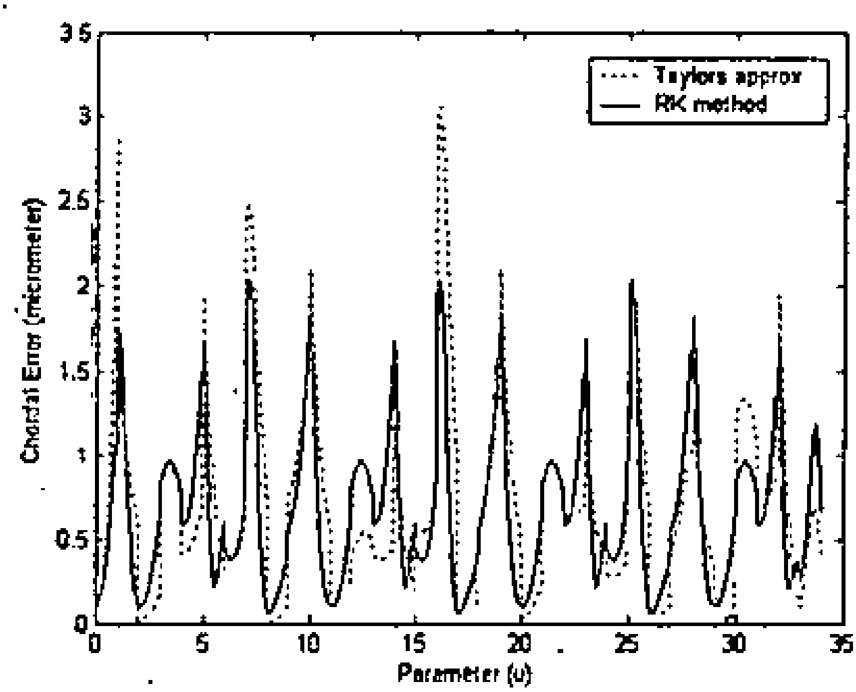

Figure 6.7(b): Chordal error from constant feedrate (200mm/sec) interpolation for the fan (B-spline curve).

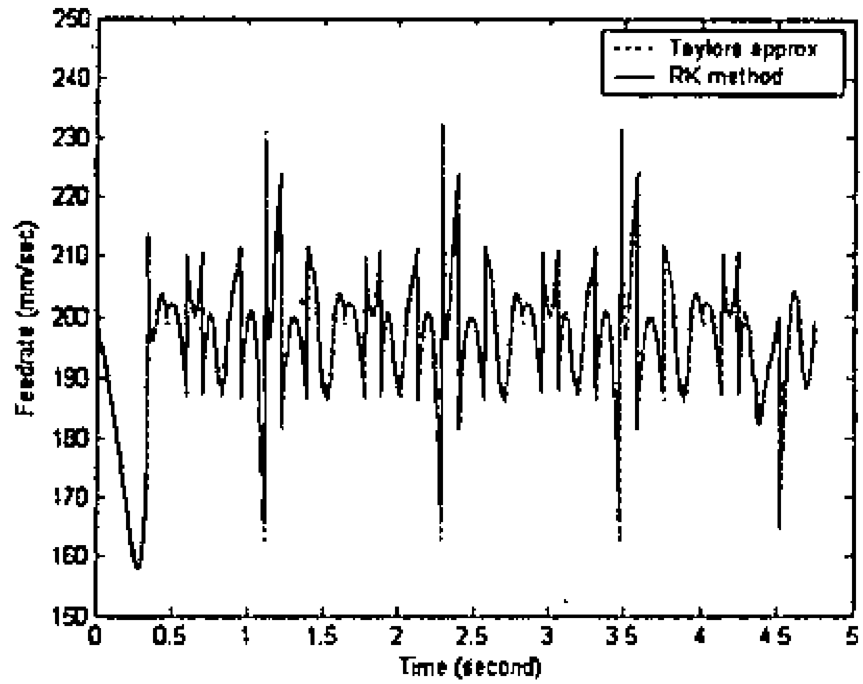

Figure 6.7(c): Actual feedrate from constant feedrate (200mm/sec) interpolation for the fan (B-spline curve)

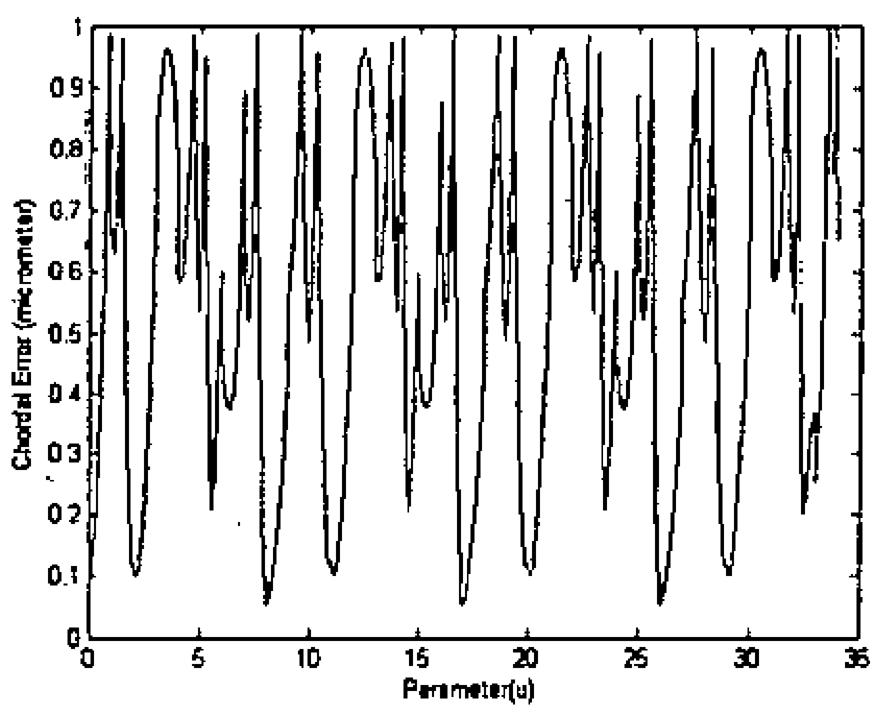

Figure 6.7(d): Chordal error from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the fan (B-spline curve).

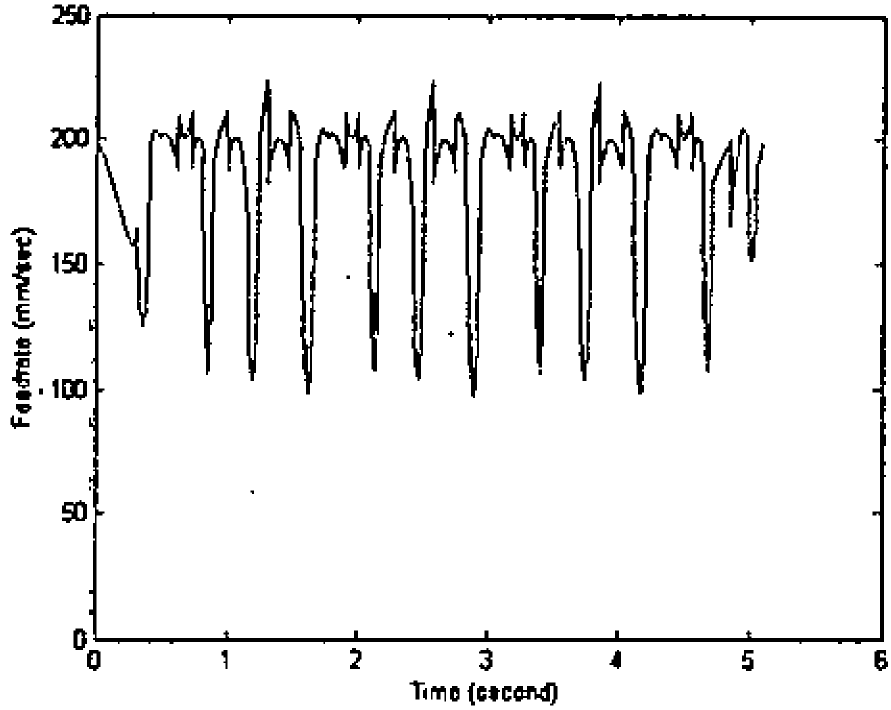

Figure 6.7(e): Actual feedrate from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the fan (B-spline curve).

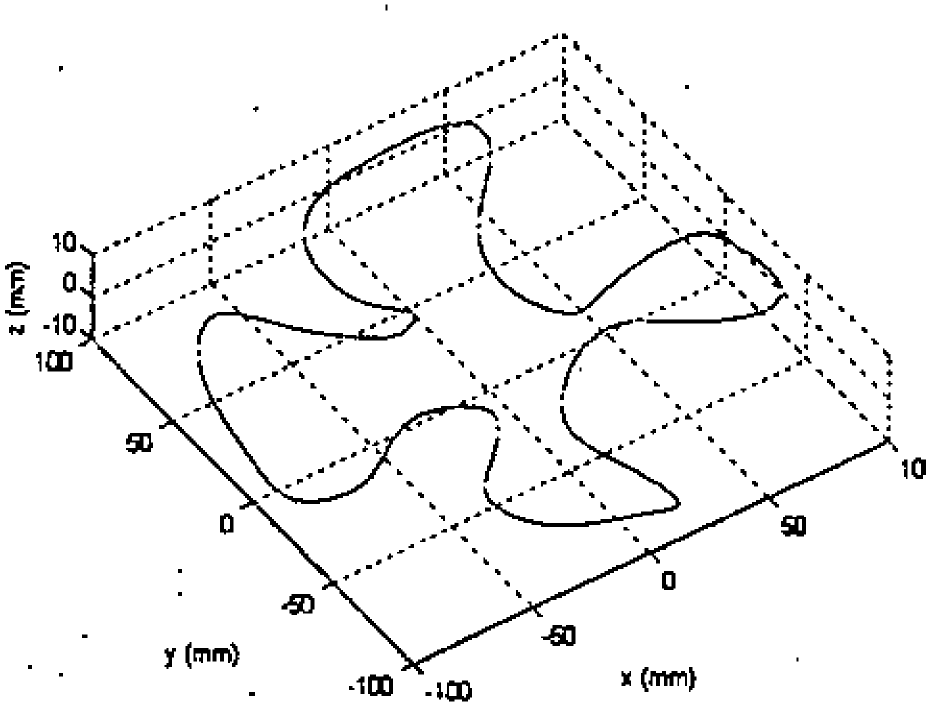

Figure 6.7(f): Tool positions generated from the variable feedrate interpolation for the fan (B-spline curve).

For constant feedrate interpolation, the simulation results of chordal error and feedrate are shown in Figure 6.7 (b) and (c) for the 3-D fan profile. The chordal error resulting from using the Taylor's approximation exceeds that from the RK method several times. The maximum errors for the two methods are 3 μm and 2 μm respectively. The results for both the methods show that there are twelve peaks of the error plot. This is due to the fact that the fan has four blades and each has three segments where the radii of curvature are smaller. When the tool passes through these segments the chordal error becomes greater. So the peaks of the error plot correspond to these high curvature segments of the fan profile. Figure 6.7 (c) suggests that there is no significant difference between the feedrate profiles resulting from two methods. For both the methods maximum feedrate deviation is around 21%. The fluctuation of feedrate in a fraction of a second results in high acceleration/deceleration and jerk, which in turn loads the servo control of the machine.

The results of variable feedrate interpolation for the fan are shown in Figure 6.7 (d), (e) and (f). The results are satisfactory in that chordal error is confined within the tolerance. Whenever the tool comes out of a segment of high curvature the error increases and if this magnitude is within the limit the algorithm permits the tool to travel to the next position. The results of actual feedrate are also satisfactory, because feedrate does not exceed the specified value to a greater extent. The maximum feedrate obtained is around 224 mm/sec. So the machined surface quality can be achieved as desired. The variations in feedrate are significant only at the segments of high curvature. The Figure 6.7 (e) shows twelve valleys corresponding to those segments of the fan profile having smaller radii of curvature. As error increases when the tool travels through those segments. the algorithm decreases the feedrate according to the relation established in Chapter 4.

The 2-D NURBS curve, shown in Figure 6.8 (a), of order 3 has the control points shown in Table 6.4, the weight vector h = [1, 25, 25, 1, 25, 25, 1] and the knot vector t = [0, 0, 0. 0.25, 0.5, 0.5, 0 75, 1, 1, 1].

Table 6.4: Control points of the 2-D NURBS curve.

No	1	2	3	4	5	6	7
x	0	-150	-150	0	150	150	0
y	0	-150	150	0	-150	150	0
z	0	0	0	0	0	0	0

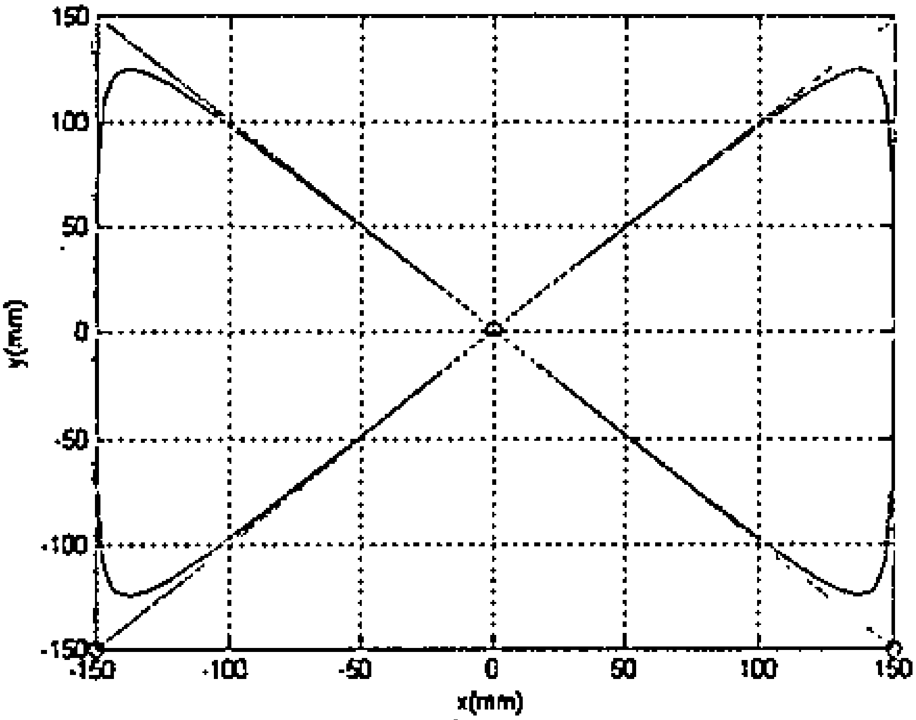

Figure 6.8 (a): The 2-D NURBS curve

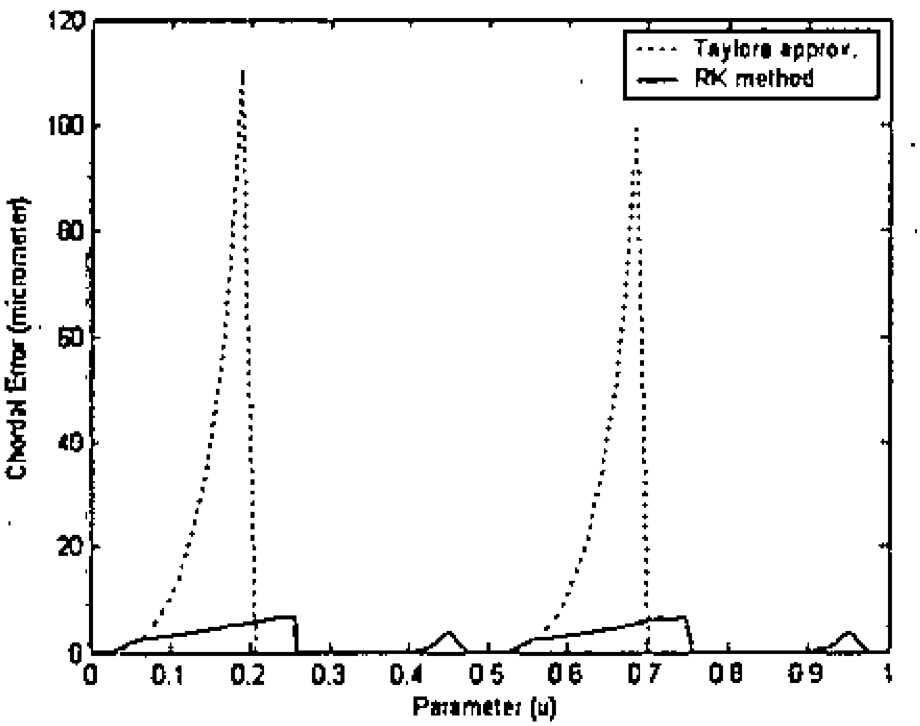

Figure 6.8 (b): Chordal error from constant feedrate (200mm/sec) interpolation for the 2-D NURBS curve

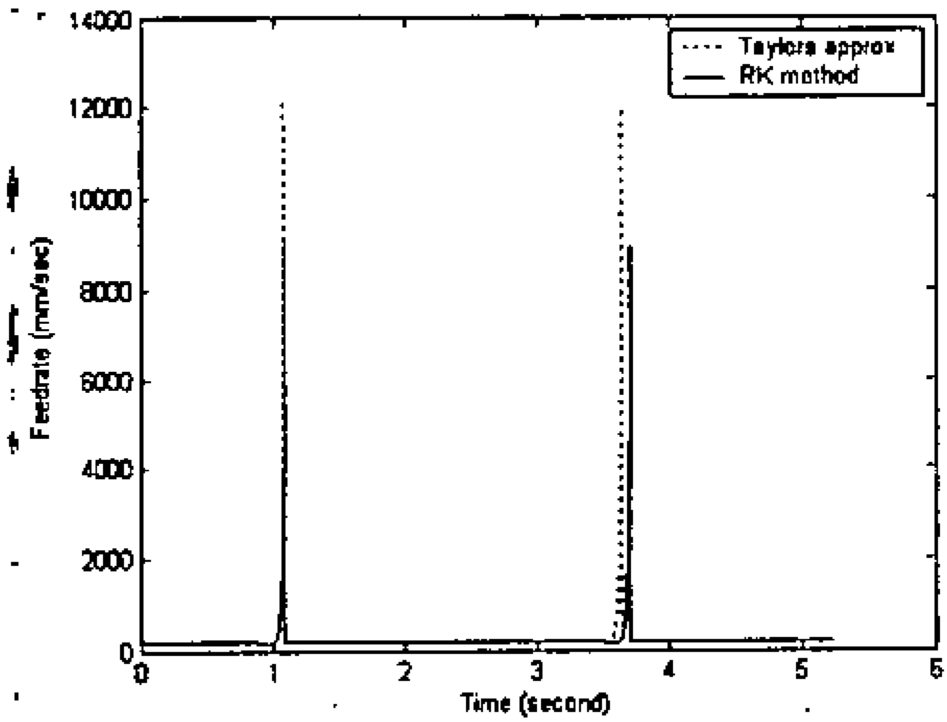

Figure 6.8(c): Actual feedrate from constant feedrate (200mm/sec) interpolation for the 2-D NURBS curve

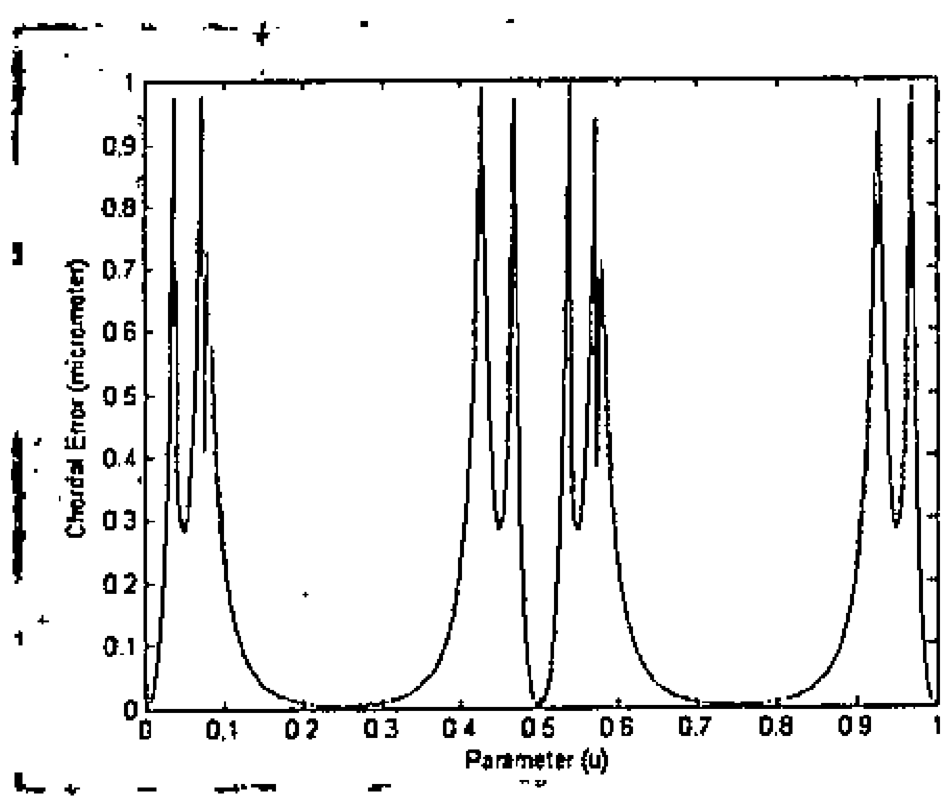

Figure 6.8(d): Chordal error from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the 2-D NURBS curve

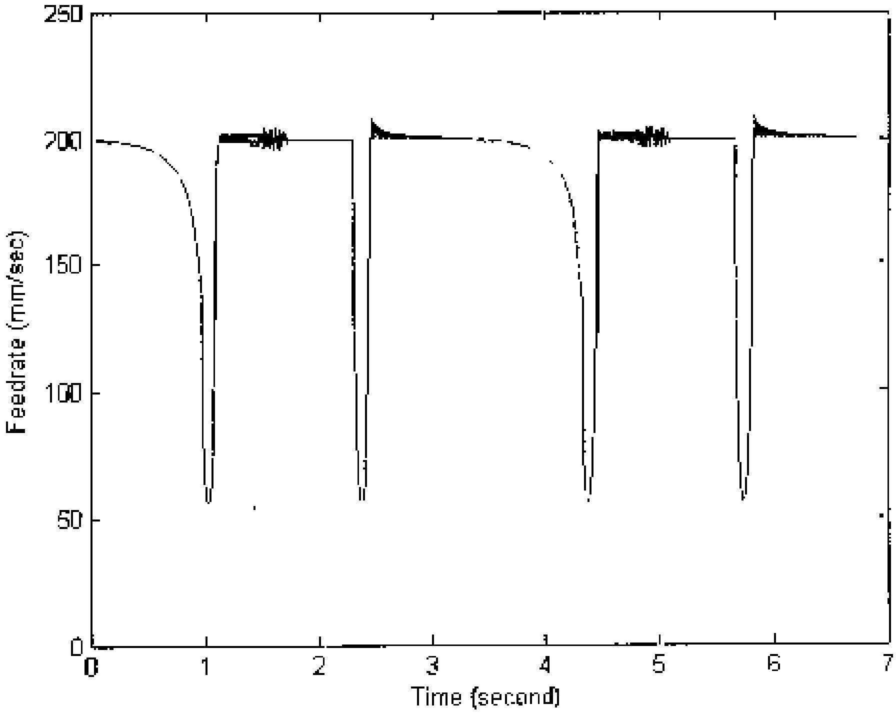

Figure 6.8(e): Actual feedrate from variable feedrate interpolation with specified feedrate 200mm/sec, tolerance 1μm, for the 2-D NURBS curve

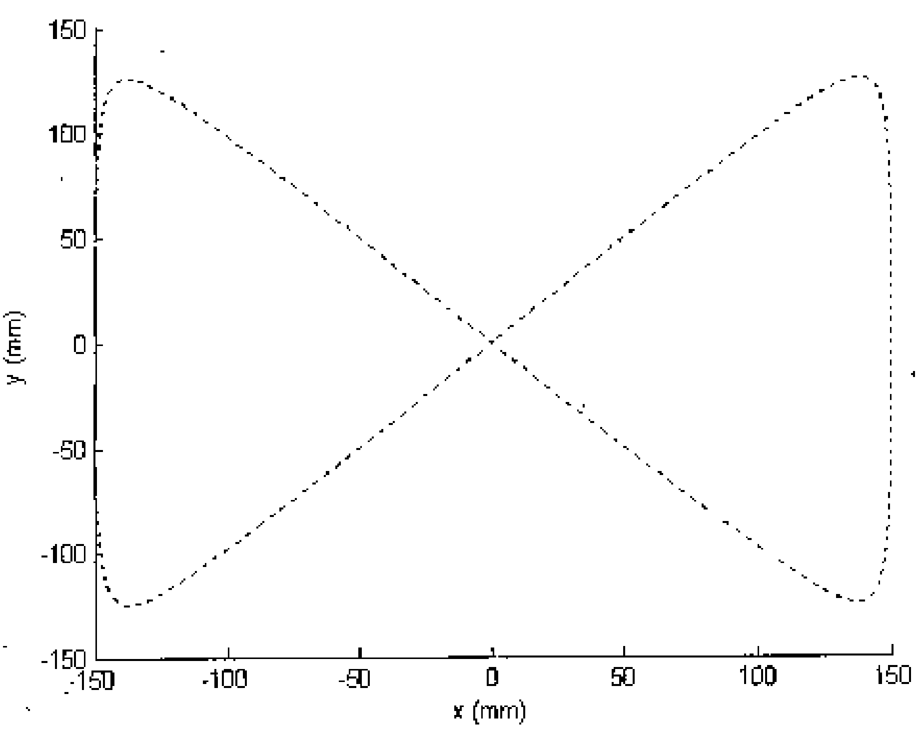

Figure 6.8(f): Tool positions generated from the variable feedrate interpolation for the 2-D NURBS curve

For the 2-D NURBS curve, simulation results of constant feedrate interpolation are shown in Figures 6.8 (b) and (c). The suitability of RK method is best explained by the plot of error in Figure 6.8(b) for the 2-D NURBS curve which has four segments of high curvature and two segments of almost straight line. At the segments of higher curvature the Taylor's approx. results much greater chordal error than the RK method. The maximum chordal errors from these methods are 99 μm and 6.5 μm respectively. The maximum values correspond to those when the tool enters into a segment of smaller radius of curvature from a segment almost straight.

There are two peak values (12174 mm/sec for Taylor's approximation, 8993 mm/sec for RK method) of federate, as shown in Figure 6.8 (c), for both the methods resulting when the tool leaves the curved portions and enters the straight portions. The reason for these high values of feedrate is that the tool traverses from one point to the next in constant time period, i.e. the sampling time of the controller of the CNC machine. In the straight portion of the curve, a change in the value of parameter results in a much distant point on the curve, which the tool travels in a sampling time resulting in a higher feedrate. These two changes would result in tremendous load on the servo control of the machine and would cause the machine to vibrate violently. Although resulting in fluctuation of feedrate the RK method results in a smaller value of feedrate.

The results of variable feedrate interpolation for the 2-D NURBS curve are shown in Figures 6.8 (d), (e) and (f). The chordal error remains well below the tolerance. There are four regions where the error increases and corresponds to those segments of the curve having high curvature. When the tool travels along the two segments which are almost straight the chordal error remains very close to zero. The actual feedrate resulting from the simulation of variable feedrate interpolation remains very close to the specified value of 200 mm/sec for most of the time. These are the times corresponding to when the tool travels along the straighter portions of the curve. In the four segments where the curvature is high, the interpolator reduces feedrate to confine error within the limit. From Figure 6.8 (e) it can be found that the feedrate slightly exceeds the specified value at some segments which is due to the fact that from a segment of higher curvature when the interpolator generates the next parameter for the segment of relatively lower curvature or approximately straight segments, the chordal distance becomes slightly greater than that would result in specified feedrate. It is impossible to map from the chordal distance to the parameter increment that could be incorporated into the interpolation algorithm resulting in feedrate strictly within the specified value.

Results of simulation for all the curves (with given feedrate 200 mm/sec and chordal error tolerance 1μm.) are summarized in Table 6.6. Based on the simulation results, it can be fairly concluded that the proposed variable feedrate interpolation employing the Runge-Kutta method performs satisfactorily resulting in chordal error within tolerance and feedrate profile with less fluctuations about the specified value. Applying the algorithm on various types of spline curves commonly used to define free-form surface validates the effectiveness of the proposed method of CNC interpolation.

Table 6.5: Simulation results of interpolation algorithms on different

Curves \ Interpolation algorithm			Constant feedrate		Variable feedrate (4^{th} order Runge-Kutta method)
			2^{nd} order Taylor's approx	4^{th} order Runge-Kutta method	
3-D Cubic spline	Chordal error (μm)	Max.	9.9916	9.4105	0.9911
		RMS[a]	3.6048	3.3256	0.4275
	Actual feedrate (mm/sec)	Max.	200.4505	199.9995	199.9995
		Min.	199.7492	199.7047	21.1883
		RMSFE[b]	0.1427	0.0810	136.6245
Cam profile (2-D B-spline)	Chordal error (μm)	Max.	21.5807	9.3980	0.9920
		RMS	5.6585	4.8497	0.2668
	Actual feedrate (mm/sec)	Max.	233.9522	219.0104	207.0273
		Min.	78.8487	173.1842	16.8091
		RMSFE	12.8428	10.0976	158.7886
Fan profile (3-D B-spline)	Chordal error (μm)	Max.	3.0579	2.0447	0.9999
		RMS	0.8280	0.7568	0.5682
	Actual feedrate (mm/sec)	Max.	231.4522	232.3512	223.8730
		Min.	158.3405	157.8427	97.2110
		RMSFE	11.2670	11.0587	34.8698
2-D NURBS	Chordal error (μm)	Max	99.1841	6.5224	0.9926
		RMS	4.3782	0.7885	0.1933
	Actual feedrate (mm/sec)	Max.	12174	8993.7	208.7425
		Min.	76.9376	134.4962	56.2202
		RMSFE	591.1387	449.3853	36.0122

[a] RMS: Root mean square of chordal error
[b] RMSFE: Root mean square of feedrate error

CHAPTER 7
CONCLUSIONS

Industries like aircraft, shipbuilding and automobile uses free-form surfaces for various applications. Automobile body, impellers, aircraft wings are typical examples which requires the dies and molds with cavities of free-form contour. The machining of these dies and molds is quite complex as there is usually very tight tolerance on error. Obviously, CNC machine tools are used to machine such parts. Traditionally, free-form curve is designed on the CAD/CAM system and the curve is divided into a large number of linear and circular segments as the controllers of CNC machines do not have the capability of interpolating curves. The post-processor then generates the M-codes, G-codes etc which are downloaded into the CNC system. This method has some inherent problems of error during data communication, memory constraints etc.

To eliminate those problems and make the CNC machine more versatile researchers have developed parametric interpolators for CNC machine's controller. This interpolators only needs the parametric curve information, based on which successive tool position commands are generated and necessary signals are sent to the servo-control loops of respective axes of the machine tool.

Parametric interpolation utilizes the Taylor's series approximation for generating successive values of the parameter to be substituted into the curve equation for finding the coordinates of the successive tool positions. Taylor's approximation method involves the calculation of derivatives of the parametric curve with respect to the parameter. To achieve greater accuracy higher order derivatives are required, which is considerably complex if the parametric curve is a B-spline or a NURBS curve. Traditional method calculates the chordal error on a given segment by approximating the curvature of that segment which overlooks a portion of the error.

The chordal error is greater where the curve has a smaller radius of curvature. Usually an optimum value of feedrate is specified for any machining operation. Traditional interpolators use this specified feedrate for constant feedrate machining. In constant feedrate interpolation the magnitude of chordal error is greater when the tool traverses a segment of high curvature. This error usually exceeds the tolerance typically used in those industries where accuracy is of prime concern.

In this research a variable feedrate interpolation algorithm is suggested which would reduce the feedrate depending on chordal error. In this method error is calculated at the average value of the parameters of the end points of the given curve segment. Thus the calculation of curvature at each segment is avoided. The interpolator employs the fourth-order Runge-Kutta method to generate successive values of the parameter. The RK method only requires the first derivatives of parametric curve but achieves the accuracy of Taylor's approximation with higher order derivatives. Thus the proposed method simplifies the interpolation process. The algorithm works as follows: the RK method generates the next value of parameter which is used to calculate the coordinates of the next tool position only if the chordal error is within tolerance. If the error exceeds the tolerance the interpolator calculates the value of feedrate that would result in a next parameter for which error will be within the tolerance.

The proposed interpolation is compared with the traditional method by performing simulations on a personal computer and using different types of curves like cubic spline, B-spline and NURBS. The results show that the proposed method of interpolation using the RK method generates the parameter values for which the feedrate profile is smoother than that from using the Taylor's approximation in constant feedrate mode. The variable feedrate interpolation results in chordal error within the tolerance and the feedrate remains constant at the specified value as long as error is within limit. The feedrate is decreased when necessary to maintain the error within tolerance.

Hence the proposed variable feedrate interpolation results in much smoother fluctuation of feedrate while maintaining the chordal error within tolerance.

REFERENCE

[1] Chou, J. J. and Yang, D. C. H., "Command generation for three-axis CNC machining", *Transactions of ASME, Journal of Engineering for Industry*, Vol. 113(1), pp. 305-310, 1991

[2] Huang, J. T. and Yang, D. C. H. "A generalized interpolator for command generation of parametric curves in computer controlled machines." *Japan/USA Symposium on Flexible Automation*, Vol. 1, pp.393–399, 1992.

[3] S. Bedi and N. Quan, "Spline interpolation technique for NC machines", *Computers in Industry*, Vol. 18, pp. 307-313, 1992.

[4] S. Bedi, I. Ali and N. Quan, "Advanced interpolation techniques for NC machines", *Transactions of ASME, Journal of Engineering for Industry*, Vol. 115, pp. 329-336, 1993.

[5] F. C. Wang and D.C.H. Yang, "Nearly arc-length parameterized quintic-spline interpolation for precision machining", *Computer-Aided Design*, Vol. 25(5), pp. 281–288, 1993.

[6] Koren Y., Lo C. C. and Shpitalni M., "CNC interpolators: algorithm and analysis", *ASME Journal of Manufacturing Science and Engineering*, Vol. 64, pp. 83-92, 1993.

[7] Shpitalni , M., Koren Y. and Lo, C. C., "Real-time curve interpolators", *Computer-Aided Design*, Vol. 26(11), pp. 832–838, 1994.

[8] Yang, D. C. H. and Kong, T., "Parametric interpolator versus linear interpolator for precision CNC machining", *Computer-Aided Design*, Vol. 26(3), pp. 225–234, 1994.

[9] Dimitris Kiritsis, "High precision interpolation algorithm for 3D parametric curve generation" *Computer-Aided Design*, Vol. 26(11), pp. 850–856, 1994.

[10] Lo C. C., "Feedback interpolators for CNC machine tools", *ASME Journal of Manufacturing Science and Engineering*, Vol. 119, pp. 587-592, 1997.

[11] Zhang, Q. G., Greenway, R.B., "Development and implementation of a NURBS curve motion interpolator", *Robotics and Computer-integrated manufacturing*, Vol. 14, pp. 27-36, 1998.

[12] Lo C. C. and C. Y. Hsiao, "CNC machine tool interpolator with path compensation for repeated contour machining". *Computer-Aided Design*, Vol. 30(1), pp. 55–62, 1998.

[13] Lo C. C., "A new approach to CNC tool path generation". *Computer-Aided Design*, Vol. 30(8), pp. 649–655, 1998.

[14] Yeh, S. S. and Hsu, P. L. "The speed-controlled interpolator for machining parametric curves". *Computer-Aided Design*, Vol. 31, pp. 349–357, 1999

[15] Lin, R. S., "Real-time surface interpolator for 3-D parametric surface machining on 3-axis machine tools". International Journal of Machine Tools & Manufacture, Vol. 40, pp. 1513-1526, 2000.

[16] H. Y. Xu, H. Y. Tam, Z. Zhou and P. W. Tse. "Variable feedrate CNC interpolation for planar implicit curves" *International Journal of Advanced Manufacturing Technology*, Vol. 18, pp. 794–800, 2001.

[17] R. V. Fleisig and A. D. Spence, "A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining" *Computer-Aided Design*, Vol. 33, pp. 1-15, 2001.

[18] C. Lartigue, F. Thiebaut and T. Maekawa, "CNC tool path in terms of B-spline curves". *Computer-Aided Design*, Vol. 33, pp. 307–319, 2001.

[19] Farouki, R. T. and Tsai, Y. F., "Exact Taylor series coefficients for variable-feedrate CNC curve interpolators". *Computer-Aided Design*, Vol. 33 pp. 155–165, 2001.

[20] B. Bahr, X. Xiao and K. Krishnan, "A real-time scheme of cubic parametric curve interpolations for CNC systems". *Computers in Industry*, Vol. 45 pp. 309–317, 2001.

[21] Zhiming, X., Jincheng, C., Zhengjin, F., "Performance evaluation of a real-time interpolation algorithm for NURBS curves". *International Journal of Advanced Manufacturing Technology*, Vol. 20, pp. 270–6, 2002

[22] H. Y. Xu, "Linear and angular feedrate interpolation for planar implicit curves". *Computer-Aided Design*, Vol. 35, pp. 301–317, 2003.

[23] S. H. Nam and M. Y. Yang, "A study on a generalized parametric interpolator with real-time jerk-limited acceleration". *Computer-Aided Design*, Vol. 36, pp. 27–36, 2004.

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

[25] Jeong, S. Y., Choi, Y. J. and Park, P. G., "Parametric interpolation using sampled data" *Computer-Aided Design*. Vol. 38. pp. 39–47. 2006

[26] Kunwoo Lee, "Principles of CAD/CAM/CAE Systems". Addison-Wesley. 1999.

[27] S. C. Chapra and R. P. Canale, "Numerical Methods for Engineers". 4^{th} Edition. Tata Megraw Hill Publications.