

**INTERPOLATION ALGORITHMS FOR MACHINING  
PARAMETRIC CURVES**

**A Thesis submitted  
In partial fulfillment of the requirements for the  
Award of the degree of  
MASTER OF ENGINEERING  
IN  
MECHANICAL ENGINEERING  
(CAD/CAM & ROBOTICS)**

***Submitted By:*  
*SACHIN Kr. MANGLA***

**Roll No. 8018114**

***Under the guidance of:***

***Dr. H.B.SHARDA*  
*Dean of Student Affairs &*  
**Prof. Dept of Mechanical Engg**  
**T. I. E. T. Patiala****

***Sh. ASHISH SHARMA*  
*Lecturer*  
**Dept of Mechanical Engg**  
**T. I. E. T. Patiala****



**DEPARTMENT OF MECHANICAL ENGINEERING  
THAPAR INSTITUTE OF ENGINEERING & TECHNOLOGY  
(DEEMED UNIVERSITY)  
PATIALA-147004, INDIA  
2003**

## **CERTIFICATE**

This is to certify that the thesis entitled “**Interpolation algorithms for machining parametric curves**” being submitted by **Mr. Sachin Kr. Mangla** in partial fulfillment of the requirements for the award of the degree of **MASTER OF ENGINEERING (CAD/CAM & ROBOTICS)** at **Thapar Institute of Engineering & Technology (Deemed University), Patiala**, is a bonafide work carried out by him under our supervision and guidance.

**Dr. H. B. SHARDA**

**Dean of Student Affairs and Prof., MED  
Thapar Institute of Engg. & Tech.  
Patiala.**

**Sh. ASHISH SHARMA**

**Lecturer, MED  
Thapar Institute of Engg. & Tech.  
Patiala.**

**Dr. D. S. BAWA**

**Dean of Academic Affairs  
Thapar Institute of Engg. & Tech.  
Patiala.**

**Dr. R. G. TATHGIR**

**Professor & Head, MED  
Thapar Institute of Engg. & Tech.  
Patiala.**

## **ACKNOWLEDGEMENT**

I express my sincere gratitude to my guides, **Dr. H. B. SHARDA**, Dean of Student Affairs and Prof., MED and **Sh. ASHISH SHARMA**, Lecturer, MED, Thapar Institute of Engg. & Tech., Patiala for their valuable guidance, proper advice, painstaking and constant encouragement during the course of my work on this thesis.

I am deeply indebted to my parents for their inspiration and ever encouraging moral support, which enabled me to pursue my studies. I do not find enough words with which I can express my feeling of thanks to my friends for their help, inspiration and moral support which went a long way in successful completion of the present study.

I am also thankful to the entire faculty and staff members of Mechanical Engg. Department for their direct-indirect help and nice cooperation.

**SACHIN Kr MANGLA**

## **ABSTRACT**

The work reported here is concerned with study of the speed controlled interpolation algorithm for machining of parametric curves. The algorithms involving first order and second order Taylor's approximation are also studied & discussed. The speed-controlled interpolation algorithm derives a suitable compensatory parameter for the first-order approximation to obtain desirable motion speed. It is based on the Taylor's approximation. By deriving the compensatory parameter, the algorithm aims at maintaining constant speed and thus improving machining accuracy.

The speed-controlled interpolators have been successfully applied to a NURBS command to achieve better control over feedrate. The results have been studied for different values of the interpolation time ( $T_s$ ) and feedrate for the first order, second order and speed controlled interpolation algorithms. The effect of the weight vector on the speed accuracy have also been discussed.

# CONTENTS

	<b>PAGE NO.</b>
<b>THESIS TITLE PAGE</b>	<b>(i)</b>
<b>CERTIFICATE</b>	<b>(ii)</b>
<b>ACKNOWLEDGEMENT</b>	<b>(iii)</b>
<b>ABSTRACT</b>	<b>(iv)</b>
<b>CHAPTER - 1</b>	
<b>INTRODUCTION</b>	<b>1</b>
1.1 Introduction	2
1.2 CNC ( Computer Numeric Control )	3
1.3 Curves	3
1.4 Types of Curves	4
1.5 Parameters Transmission	5
1.6 Errors in interpolation	6
<b>CHAPTER - 2</b>	
<b>LITERATURE SURVEY</b>	<b>7</b>
<b>CHAPTER 3</b>	
<b>INTERPOLATORS</b>	<b>13</b>
3.1 Introduction	14
3.2 LINEAR & CIRCULAR INTERPOLATORS.	
14	
3.3 Complex Surface Interpolators	15
3.4 REAL TIME CURVE INTERPOLATOR:	
16	
3.5 INTERPOLATORS FOR NONPARAMETRIC CURVES	
18	
3.6 INTERPOLATORS FOR PARAMETRIC CURVES	
19	

3.7 FEEDBACK INTERPOLATORS		
21		
<b>CHAPTER 4</b>		
<b>nurbs Interpolation</b>	<b>22</b>	
4.1 NURBS		23
4.2 Need Of Nurbs		23
4.3 NURBS: DEFINITION		
24		
4.4 IMPORTANT PROPERTIES OF NURBS CURVES		
25		
4.5 Nurbs Interpolation		25
4.6 Processing Nurbs data using linear Interpolation	26	
4.7 NURBS INTERPOLATION IN THE CNC		
27		
4.8 The importance of Interpolation rate		27
<b>CHAPTER 5</b>		
<b>PROPOSED ALGORITHM &amp; METHODOLOGY</b>	<b>28</b>	
5.1 Introduction		29
5.2 Methodology		29
5.3 Parametric curve formulation		30
5.4 The First-order interpolation algorithm		32
5.5 The Second-order interpolation algorithm		33
5.6 The Speed-controlled interpolation algorithm		33
5.6.1 The compensatory parameter		33
5.6.2 Selection of compensatory parameters		36
<b>CHAPTER 6</b>		
<b>RESULTS &amp; DISCUSSIONS</b>		<b>42</b>
6.1 Introduction		43
6.2 Input data		43
6.3 Color convention		44
6.4 First order approximation		44

6.5 Second order approximation	53	
6.6 Speed controlled interpolation	61	
6.7 Effect of the weight vector	70	
<b>CHAPTER 7</b>		
<b>CONCLUSIONS</b>		<b>77</b>
7.1 Conclusions	78	
7.2 Scope for further work	79	
<b>REFERENCES</b>		<b>81</b>
Books		81
Internet links		81
Journals and papers		82

# **Chapter 1**

## **INTRODUCTION**

## 1.1 Introduction:

The importance of machining non linear contours in multi-axis computer numeric control (CNC) systems keeps increasing because one of the most important trends in manufacturing is to machine fewer but more complex parts.

To realize a multi-axis contour machining, an interpolation of the cutter path to motion trajectories of all the axes is required. The interpolation, also referred to, as “command generation” is to generate the reference position commands for the servo control loops in order to coordinate their motion in a multi-axis machining.

The interpolators used by most CNC machines can process only straight lines and circular motion commands. Consequently to machine a curve, current computer aided manufacturing systems approximate the curve by many linear segments and then download these linear programs to the CNC machines. This approach may cause the following problems;

- The segmentation causes discontinuities in the first derivatives along the cutter path and then deteriorates the smoothness of the curves and surfaces.
- The segmentation of a curve to linear segment requires the generation of a huge number of straight-line segments especially when quality surface smoothness is required.
- a large CNC’s memory is required to store the linear segments needed for machining of a complex surface
- A heavy communication between the CAM and the CNC systems is required.

To eliminate the above problems, modern interpolators in CNC systems are required to be capable of generating curves. Conventional interpolators are capable of generating simple 2D curves such as lines, circular arcs and parabolas.

In this chapter the basic terms & terminology used in the thesis are discussed.

## 1.2 CNC (Computer Numeric Control)

CNC stands for Computer Numerical Control and has been around since the early 1970's. Prior to this, it was called NC, for Numerical Control. In the early 1970's computers were introduced to these controls, hence the name change. Today CNC machines have touched almost every form of manufacturing process in one way or another and have replaced some existing manufacturing processes.

Here are a few of the more important benefits offered by CNC equipment.

- The first benefit offered by all forms of CNC machine tools is improved automation.
- The second major benefit of CNC technology is consistent and accurate workpieces.
- A third benefit offered by most forms of CNC machine tools is flexibility.

## 1.3 Curves

Every graphics system has some form of primitive to draw lines. Using these primitives we can draw many complex shapes. However, as these shapes get ever more complex and finely detailed, so does the data needed to describe them accurately. A curve can be described by arrays of coordinated data or by an analytic equation.

Curves can be described mathematically by non-parametric or parametric equations. Non-parametric equations can be explicit or implicit. For a non- parametric curve the coordinates  $y$  &  $z$  of a point on the curve are expressed as two separate functions of the third coordinate  $x$  as the independent variable. This curve representation is known as the non-parametric explicit form. If the coordinates,  $x$ ,  $y$ , &  $z$  are related together by two functions a non-parametric implicit form results. For a parametric curve on the other hand a parameter is introduced & the coordinates  $x$ ,  $y$  &  $z$  are expressed as the functions of this parameter.

In parametric form each point on the curve is expressed as a function of a parameter  $u$ . The parameter acts as a local coordinate for points on the curve. The parametric curve is bounded by two parametric values  $u$  minimum &  $u$  maximum. It is convenient to normalize the parametric variable  $u$  to have limits between 0 & 1.

Parametric curves are profiles in different formats like the Bezier curve, B-spline, cubic spline, and NURBS (non-uniform rational B-spline). The general parameter iteration method used is

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

Where  $u_i$  is the present parameter,  $u_{i+1}$  is the next parameter, and  $\Delta u_i$  is the incremental value. The interpolated points are calculated by substituting  $u_i$  into the corresponding mathematical model to recover the originally designed curves.

## 1.4 Types of Curves

There are basically four types of curves. They are piecewise linear approximations, Hermite curves, Bezier curves, and Splines.

### **Hermite curves**

The Hermite curve family is a parametric cubic curve described by 2 endpoints and a tangent vector at each of those endpoints.

### **Bezier curves**

Another family of curves is the Bezier Curve Family. Named after Pierre Bezier, these curves are described by two endpoints and two “control” points. These control points are not on the curve but control the movement of the curve

## B-Splines

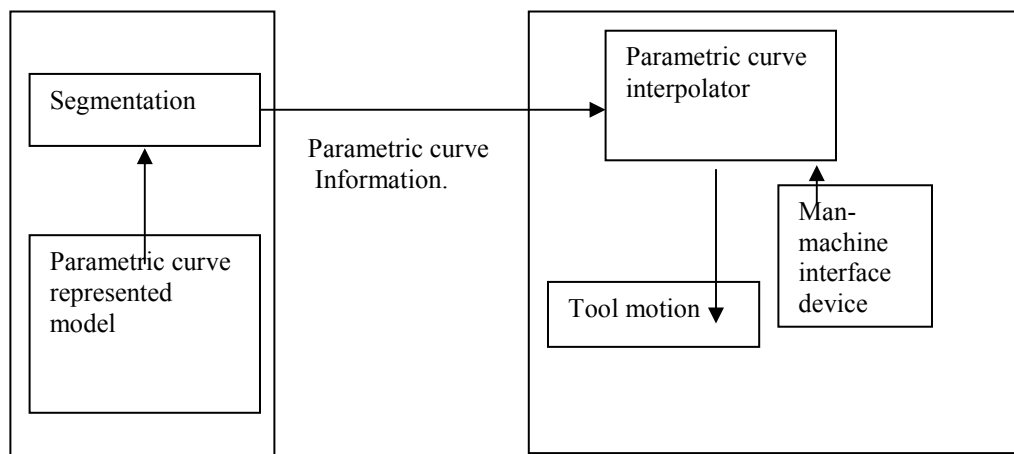
B-Splines consists of curve segments whose polynomial coefficients depend on just a few control points. This is called local control. So moving a control point would only affect a small part of the curve, and the time needed to compute the coefficients is reduced. B-Splines have the same continuity as natural splines, but do not interpolate their control points

## NURBS

A NURB is also known as a non-uniform rational B-spline. It allows the user to reshape a curve by adding a control point (and with it, an additional knot), which makes modifying a curve much easier.

### 1.5 Parameters Transmission

As the generated curves or profiles may be in a parametric form, only parametric curve information is required to be efficiently transferred among CAD/CAM/CNC systems as shown in Fig. 1.

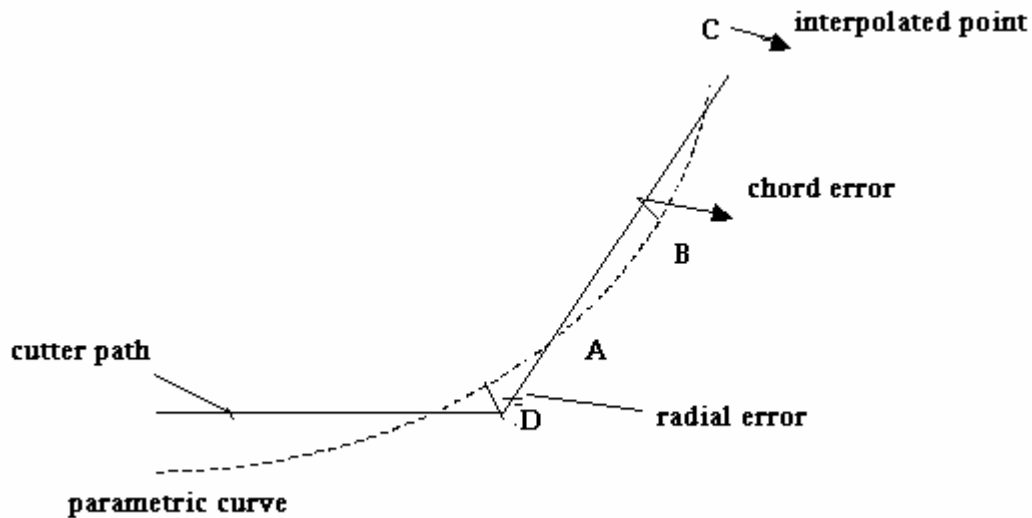


*Fig 1. The machining system with parameters transmission.*

## 1.6 Errors in Interpolation

As the cutter moves straight between contiguous interpolated points, two position errors may occur. They are:

- Radial error.
- Chord error during motion for a parametric curve as shown in Fig. 2.



*Fig 2: The Radial error & the chord error*

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 precision, the radial error is no longer a major concern in present applications. In addition, the chord error is the maximum distance between the secant  $CD$  and the secant arc  $AB$ . A small curvature radius with a fast feedrate command may cause the chord error.

# **Chapter 2**

## **LITERATURE SURVEY**

## LITERATURE SURVEY

Several researchers have developed different interpolation algorithms for parametric curves to improve curve accuracy.

Bedi [1] set  $\Delta u_i$  as a constant to form the uniform interpolation algorithm, which is the simplest method and its chord error and curve speed however are not guaranteed.

According to the analysis of CNC machine kinematics and cutter path geometry, Chou and Yang [2] proposed an improved interpolation algorithm in position, velocity, and acceleration if the CNC machine's kinematic model is known exactly.

Further, Houg and Yang [3] developed the cubic spline parametric curve interpolator by using the Euler approximation method to study the variable federate control problem. Their method yielded satisfactory performances as long as the curvature of the curve for machining was small.

Shpitalni [4] derived the same interpolation algorithm by using Taylor's expansion.

According to him,

$$u_{n+1} = u_n + \frac{(FxT)}{(\sqrt{x_n'^2 + y_n'^2 + z_n'^2})} - \left[ \frac{(F^2 x T^2)(x_n' x_n'' + y_n' y_n'' + z_n' z_n'')}{2(x_n'^2 + y_n'^2 + z_n'^2)^2} \right]$$

where T is the sampling time, F is the feedrate,  $x_n''$ ,  $y_n''$  and  $z_n''$  are the second derivatives of X, Y, Z position with respect to the spatial parameter u at time kT

A drawback of the above interpolation method is that the feedrate error is not calculated off-line or on-line the interpolation, and consequently we do not know whether or not these methods can satisfy the interpolation requirement. A conservative way is to utilize the modified method although an increase in computational time is expected.

Another drawback of the current method is that except the parametric function in additional formulas must be formulated and implemented in the interpolator.

Lo and Chung [5] proposed the contouring error compensation interpolation algorithm which contains real-time contouring error calculations and a simplified parameter iteration method to achieve satisfactory motion accuracy.

The speed-controlled interpolation algorithm by S.-S. Yeh, P.-L. Hsu [6] derives a suitable compensatory parameter for the first-order approximation to achieve better control over feedrate.

Young Joon Ahn, Hong oh kim [7 , 8 ] discussed a approximate implicitization method with piecewise quadratic splines in Bezier form. First the parametric curve was divided into several parts. Then a quadratic rational Bezier curve was adapted to approximate each segment.

Based on first and second order Taylor's expansion or Runge-Kutta method, several investigators have proposed different kinds of parametric interpolators to generate servo commands for applications in constant feedrate control problems (Koren, 1993 [9] ; Wong [10]). Nevertheless, the capability of performing variable feedrate control is invaluable in high-speed machining applications and deserves more study.

Yang and Kong [11, 12] proposed a parametric interpolator for variable feedrate based on the first- and second- order Taylor's expansion.

Oscar n Ruiz, Adriana n Martínez, Elizabeth n Rendón [13] gave a algorithm for the direct interpolation of parametric planar curves  $C(u)$  with a CNC machine which expresses parametric planar curves as sequences of machine tool axes discrete movements of BLU size. Therefore, the curve  $C(u)$  is directly approximated by the pulse trains, hence eliminating one source of the machining errors. This algorithm directly expresses parametric planar curves as sequences of axis movements. The algorithm is addressed to work with any type of parametric curve.

Bresenham's, DDA & midpoint algorithms are also used to draw straight lines and circles on raster graphics devices as sequences of screen pixels (Book [6]). These algorithms make use of symmetries of the primitive to reduce calculations and accelerate the determination of the pixels to highlight.

In addition to software-based procedures (Bresenham's and Midpoint), hardware ones are available which perform the interpolation of an implicit planar function by using arrays of DDA's

Xunnian Yang, Wang [14], gave a algorithm in which planar point sets are fitted by minimal energy arc splines. The process consists of two steps: computing the optimal tangents for curve interpolation and adjusting the point positions by smoothing the discrete curvatures. To fit the point set with minimal energy arc curve, a simple linear algorithm is given for computing the optimal tangents.

In Biarc approximation of NURBS curves ( Pieg, Tiller [15]), NURBS curve is approximated by a polygon to within a certain tolerance. The polygon is then approximated by biarcs whose end points lie on the curves, & whose end tangents are the tangents of the curve taken at the end points of the biarc. The method is most useful in numerical control to drive the cutter along straight line or circular paths.

A NC system that machines a curved shape at fixed depth of cut experiences time varying cutting forces. Due to the 'curvature effect' the material removal rate is higher

than nominal in concave regions, and lower in convex regions. A curvature dependent feedrate function that automatically compensates for this effect is also formulated, by Farouki and Nicholas, [16]. The curvature compensated feedrate scheme has important potential applications in ensuring part accuracy and in optimizing part programs consistent with a prescribed accuracy.

Consequently, one may formulate CNC interpolators for machining free-form curves that are extremely accurate, efficient, robust, & flexible.

Circular arc interpolation is essential in manufacturing of curve contours. However, the problem of how to determine the parameters of the circular arcs and how to minimize the number of arcs according to desired interpolation accuracy have not been completely solved. A new algorithm of circular arc interpolation for planer curve is also introduced by Cheng and Lee [17]. It is based on the principle of minimax approximation.

Using circular arc interpolation is a better way because the finished profile is smooth. On the other hand, several algorithms have been proposed in which parameters of circular arcs are sequentially determined by a numerical technique so as to restrict the interpolation error less than the desired value. However, these methods are not minimax approximations. [18,19]

For many cases the interpolator cannot satisfy the accuracy required for most CNC systems Therefore, the existing interpolators are not appropriate for the cases with small radii of curvature and large feedrates. So to overcome these difficulties feedback interpolators were introduced by Chih Ching Lo. [20]. He gave a modified interpolator to reduce the feedrate error.

Feedback interpolator for parametric curves is also based on the concept of *feedback*.. The idea of the feedback method is to utilize a very simple algorithm to replace the existing algorithm and then to enhance it by a feedback compensation.

The feedback algorithm is much simpler than the previous one. This algorithm however may result in a large feedrate error. To compensate for the feedrate error, a feedback loop is formed by calculating the current feedrate in real time and utilizing it in the correction of the spatial parameter.

# **Chapter 3**

## **INTERPOLATORS**

### **3.1 Introduction:**

A common requirement of all contouring systems e.g. milling & turning is to generate coordinated movement of the separately driven axes of motion in order to achieve the desired path of the tool.

This involves the generation of the signals prescribing the shape of the produced parts by interpolators. Interpolation is performed by means of a special routine, which generates command signals for each segment of the produced part based upon the initial & final points & the type of curvature of the segment.

Typical interpolators are capable of generating linear, circular & occasionally parabolic paths. Elliptic interpolation is inapplicable in CNC of machine tools but may be useful in other manufacturing systems such as laser beam cutters. Recently interpolators for complex surfaces have been introduced.

### **3.2 Linear & Circular Interpolators.**

The linear interpolator is an iterative algorithm that is typically synchronized with the sampling time  $T$  of the control loop. Typical values of  $T$  are between 0.1 ms to 10 ms. At each iteration, the reference positions to  $X$  &  $Y$  are incremented by  $\Delta X$  &  $\Delta Y$  respectively, where

$$\Delta X = TV_x,$$

$$\Delta Y = TV_y.$$

The increment  $\Delta X$  &  $\Delta Y$  are added up to obtain the reference positions. The calculation of the axial velocities  $V_x$  &  $V_y$  is based on the given incremental distances  $L_x$  &  $L_y$  of the segment & the given desired velocity along the machined line  $V$ .

$$V_x = \frac{V \times L_x}{L}$$

$$V_y = \frac{V \times L_y}{L}$$

where L is the segment length.

Circular interpolation is based on dividing an arc into small angles  $\alpha$ . The minimum  $\alpha$  is the angle that the tool can traverse during one sampling period T (typical Ts are 0.1 ms to 10 ms) and is given by

$$\alpha = V \times \frac{T}{R}$$

In circular interpolator the tool moves between successive iterations, (n-1) & n in short straight lines that their length is  $V \times T$ .

### **3.3 Complex surface interpolators:**

Integrated CAD/ CAM systems interfaced with CNC enable the production of a part surface from geometric models. In this process, the part geometric data have to be transferred from the CAD/ CAM to the CNC.

In the conventional method, the surface is decomposed by the CAD/ CAM system into a sequence of straight lines that approximate the desired part surface with a given tolerance. These straight lines are converted into the cutter location CL file. The CL file actually describes the cutter path in a sequence of tool positions.

Using the post processor, the CL file is finally converted into G-code straight line instructions that are downloaded into the CNC system. This approach is utilized since

most conventional CNC machine tools provide only circular & linear interpolators, & the linear fitting is the simplest approximation.

The drawbacks of this off-line approach are:

- It requires acceleration & deceleration steps at each line of the CL file, thereby producing less smooth curves & substantially increasing the cutting time.
- Tool orientation in 5 axis machining are interpolated inaccurately that subsequently causes position errors & unsmooth surfaces
- The size of the CL file will be very large for complicated parts that cause memory shortage problems & data transmission errors.

### **3.4 Real time curve interpolator:**

Precision & productivity can be improved by the development of real time interpolators for multi axis CNC machine tools. This real time interpolator calculates new commands for the control loops during a short time period. I.e. 1 to 10 ms.

The successive commands are calculated during the execution time of current commands. This approach produces smoother surfaces & requires substantially less machining time compared with the conventional off line approach.

This real time interpolator consist of:

- tool path planning
- trajectory planning
- Inverse kinematics.

The input to the controller is a special G0-code i.e. NC instructions of a curve.

For surface machining the interpolator starts with tool path planning, which generates curves as tool paths, & subsequently executes the trajectory planning portion for the interpolation of each curve.

When curve machining commands are specified, the interpolator starts with trajectory planning, which generates the cutter position X, Y, Z & orientation  $O_x$ ,  $O_y$  &  $O_z$  of each interpolation step, for a constant federate.

For 5 axis machines the interpolator needs an additional portion, the inverse kinematics transformation, to generate the 5 axial reference positions (x, y, z, a, b) which in turn are transmitted to controllers for controlling the machine tool. For 3-axis machines, the inverse kinematics transformation is an identity matrix.

The real time surface interpolator generates the tool position & orientation in each sampling period, & sends the reference commands to the machine controller during cutting. Therefore a real time controller has the following advantages.

- smoother curves & smaller machining time,
- The intermediate acceleration & deceleration steps are eliminated.
- More accurate contour in 5 axis machining (achieved by accurately interpolating the tool positions & orientations.)
- Smaller CNC memory is required.

### 3.5 Interpolators for Nonparametric Curves

A 2D non-parametric curve can be expressed in an implicit form as:

$$G(x, y) = 0$$

Interpolator generates the reference points along the curve in real time under two requirements.

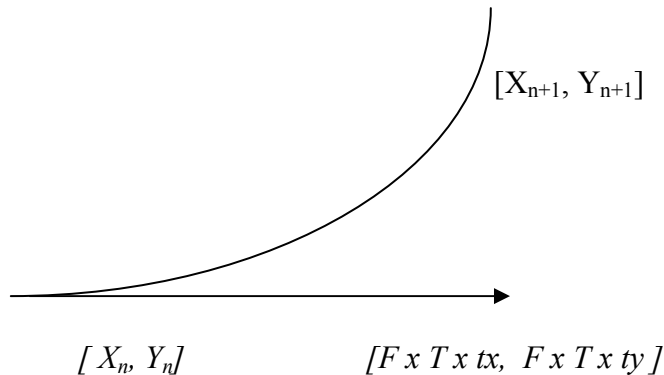
- The first requirement is to maintain the desired federate along the curve.
- The second requirement is to eliminate the contour error, which denotes the deviation of the reference point for the curve.

For a sampling period  $T$  & desired federate  $F$  a simplest interpolation method for 2d curves in implicit forms can be described by the following equations:

$$X_{n+1} = X_n + F \times T \times tx$$

$$Y_{n+1} = Y_n + F \times T \times ty$$

$[X_n, Y_n]$  and  $[X_{n+1}, Y_{n+1}]$  represent the reference positions at time  $nT$  and  $(n+1)T$ , respectively.  $[tx, ty]$  is a unit tangent vector of the curve at  $[X_n, Y_n]$



*Fig 3: Relationship between the interpolated points.*

The above method causes significant contour error, and so to reduce the same the above equations were modified. The modified equations were given by Shpitalni, 1994 [1]. The equations in modified form are:

$$X_{n+1} = X_n + F \times T \times t_x + \frac{F^2 \times T^2 \times n_x}{2 \times \rho}$$

$$Y_{n+1} = Y_n + F \times T \times t_y + \frac{F^2 \times T^2 \times n_y}{2 \times \rho}$$

where  $\rho$  is the radius of curvature and  $(n_x, n_y)$  is the unit normal vector at  $[X_n, Y_n]$ .

The contour error and the federate error increase as the radius of curvature decreases or the federate increases.

### 3.6 Interpolators for Parametric Curves

A 3D parametric curve in terms of parameter  $u$  is represented as:

$$x = x(u)$$

$$y = y(u)$$

$$z = z(u)$$

Usually the  $X, Y, Z$  positions are expressed as polynomial functions of the parameter  $u$ . The current interpolation methods assigns a distribution for the parameter  $u$  so that the reference point will track the parametric curve with the desired federate, which is usually constant in machining. The simplest way to calculate the parameter  $u$  at the sampling instant is utilizing a recursive approach. This is described by the following equation given by : ( Chou & Yang, 1991 [5] ):

$$u_{n+1} = u_n + \frac{(F \times T)}{(\sqrt{(x'_{n^2} + y'_{n^2} + z'_{n^2})}}$$

where T is the sampling period, F is the desired federate and  $x'_n$ ,  $y'_n$  and  $z'_n$  are the first derivatives of the X , Y and Z position w.r.t the parameter u at time kT, i.e,

$$x'_n = \left( \frac{dx}{du} \right) u = u_n,$$

$$y'_n = \left( \frac{dy}{du} \right) u = u_n,$$

$$z'_n = \left( \frac{dz}{du} \right) u = u_n,$$

The reference positions at the sampling instant can be calculated recursively, by substituting the calculated parameter ( $u_{n+1}$ ) into the parametric function, i.e.

$$x_{n+1} = x(u_{n+1})$$

$$y_{n+1} = y(u_{n+1})$$

$$z_{n+1} = z(u_{n+1})$$

where  $x_{n+1}$ ,  $y_{n+1}$  and  $z_{n+1}$  denote the reference positions for X, Y, Z axis at time (k+1) T.

Since the reference points are calculated by substituting  $u_{n+1}$  into the parameter function, the reference points are always located on the parametric curve, and therefore there is no position error caused by this interpolation method. However there is a federate error

For the curves with large radii of curvature the feedrate error is negligible and the above interpolation method is adequate.

For different types of parametric functions we must derive different formulas and implement them in the interpolator. Consequently, the size of the interpolator increases a lot as the types of parametric curves increase.

### **3.7 Feedback Interpolators**

Feedback interpolators are based on the concept of feedback. In it a feedback action is used to compensate for error. To maintain a negligible contour error and a constant feedrate feedback interpolators are required.

# Chapter 4

## NURBS INTERPOLATION

## 4.1 NURBS

A NURB is also known as a non-uniform rational B-spline. It allows the user to reshape a curve by adding a control point (and with it, an additional knot), which makes modifying a curve much easier.

## 4.2 Need of NURBS

One of the advantages of NURB curves is that they offer a way to represent arbitrary shapes while maintaining mathematical exactness and resolution independence. Among their useful properties are the following:

- They can represent virtually any desired shape, from points, straight lines, and polylines to conic sections (circles, ellipses, parabolas, and hyperbolas) to free-form curves with arbitrary shapes.
- They give great control over the shape of a curve. A set of *control points* and *knots*, which guide the curve's shape, can be directly manipulated to control its smoothness and curvature.
- They can represent very complex shapes with remarkably little data

In addition to drawing NURB curves directly as graphical items, we can use them in various other ways that exploit their useful mathematical properties, such as for guiding animation paths or for interpolating or approximating data & also as a tool to design and control the shapes of three-dimensional surfaces, for purposes such as

- surfaces of revolution (rotating a two-dimensional curve around an axis in three-dimensional space)
- extruding (translating a curve along a curved path)

- trimming (cutting away part of a NURB surface, using NURB curves to specify the cut)

### 4.3 NURBS: Definition

Given a set of  $n+1$  control points  $P_0, P_1, \dots, P_n$ , each of which is associated with a non-negative weight  $w_i$  (*i.e.*,  $P_i$  has weight  $w_i \geq 0$ ), and a knot vector  $U = \{ u_0, u_1, \dots, u_m \}$  of  $m+1$  knots, the NURBS curve of degree  $p$  is defined as follows:

$$C(u) = \frac{1}{\sum_{i=0}^n N_{i,p}(u)w_i} \sum_{i=0}^n N_{i,p}(u)w_i P_i$$

where  $N_{i,p}(u)$  is defined as follows:

$$N_{i,p}(u) = \begin{cases} 1 & \text{if } u_i \leq u < u_{i+1} \\ 0 & \text{otherwise} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$

Here are two results available immediately from the above definition

- If all weights are equal to 1, a NURBS curve reduces to a B-spline curve.
- NURBS Curves are Rational. The value multiplied to control point  $P_i$ ,  $N_{i,p}(u)w_i$ , is a polynomial of degree  $p$ . The denominator is the sum of all coefficients and hence is also a polynomial of degree  $p$ . As a result, the coefficient of control point  $P_i$  is the quotient of two degree  $p$  polynomials and function  $C(u)$  is rational.

## 4.4 Important Properties of NURBS Curves

- NURBS curve  $C(u)$  is a piecewise curve with each component a degree  $p$  rational curve
- Equality  $m = n + p + 1$  must be satisfied
- Strong Convex Hull Property: the NURBS curve is contained in the convex hull of its control points. Moreover, if  $u$  is in knot span  $[u_i, u_{i+1})$ , then  $C(u)$  is in the convex hull of control points  $P_{i-p}, P_{i-p+1}, \dots, P_i$
- Local Modification Scheme: changing the position of control point  $P_i$  only affects the curve  $C(u)$  on interval  $[u_i, u_{i+p+1})$
- Partition of Unity - The sum of all non-zero degree  $p$  basis functions on span  $[u_i, u_{i+1})$  is 1:
- If the number of knots is  $m+1$ , the degree of the basis functions is  $p$ , and the number of degree  $p$  basis functions is  $n+1$ , then  $m = n + p + 1$  :
- $C(u)$  is  $C^{p-k}$  continuous at a knot of multiplicity  $k$

## 4.5 NURBS Interpolation

With the NURBS Interpolation Function, the CAM process passes the original CAD "Control Points", "Weights", and "Knots" required to define the NURBS curve directly to the control in the "G Code" format.

Then, the CNC interpolates the original NURBS curve at the CNC's Interpolation Rate. NURBS Interpolation provides the following benefits over "traditional" methods of processing NURBS data:

- More Accurate Parts
- Reduced Cycle Times
- Improved Surface Quality
- Drastically Smaller Part Programs

#### **4.6 Processing NURBS Data Using Linear Interpolation**

In the past, NURBS data has been sent from the CAD system to a CAM process which would post-process the CAD NURBS data into small linear segments and create a "G-Code" Linear Interpolation Part Program for the CNC. This linearized NURBS data presented several problems for the CNC:

- The resulting part program was very long and required a large amount of Part Program storage at the CNC.
- Many CNC's could not process the data fast enough for the feedrate and size of the linear segments.
- Some CNC systems could handle the high block processing rate required but did not have the mass storage required to store the Part Program at the CNC. In this case, the CNC user needed to run the program from an external storage device in a DNC mode sometimes also called "Drip Feed Mode".
- Representing the NURBS curve with small linear segments in the CAM processor led to decreased accuracy. The size of the linear segments was totally dependent on the CNC. The slower the CNC, the longer the segment, and the poorer the accuracy.

## **4.7 NURBS Interpolation in the CNC**

With NURBS interpolation, the initial CAM process creates a part program using the original NURBS data from the CAD system. The CAM process transfers the original CAD "Control Points", "Weights", and "Knots" required to define the NURBS curve directly to the control in "G Code" format. This results in a reduction of program size of 1/10th to 1/100th compared to the size of a comparable linear interpolation part program and significantly improves the fundamental accuracy issues. The CNC interpolates the original NURBS curve at the CNC's interpolation rate.

## **4.8 The Importance of Interpolation Rate**

The Interpolation Rate is the cyclic period of time during which the CNC processor tells all of the axes in the CNC to move a certain number of counts. This happens at each Interpolation Period. During this period, the control is actually moving a linear segment because the velocity is constant during that period. The faster the interpolation rate, the smaller the linear segment, so the more accurate the finished part for a given feedrate. The interpolation rate directly affects the accuracy of the finished part and the machining cycle time.

# **Chapter 5**

## **PROPOSED ALGORITHM & METHODOGY**

## 5.1 Introduction

Modern CNC systems are designed with the function of machining arbitrary parametric curves to save massive data communication between CAD/CAM and CNC systems and improve their machining quality. Although available CNC interpolators for parametric curves generally achieve contouring position accuracy, the specified feedrate, which dominates the quality of the machining processes, is not guaranteed during motion.

This is because of the non uniform mapping between the curves and parameters. Recently, some approximation results concerning motion speed have been reported. In this paper, we derive a compensatory parameter that will help in improving curve speed & accuracy.

Most commercial computer-aided design (CAD) systems are equipped with tools to provide designers the capability of defining two and three-dimensional curves and surfaces. Among the techniques used for representing and designing specific curves and surfaces, the Non-Uniform Rational B-Spline (NURBS) is one that currently attracts a lot of attention.

NURBS offers a common mathematical form for representing and designing both standard analytical shapes and free-form curves (surfaces). By manipulating the values of weights, knot vectors and control points, a wide variety of shapes can be designed using NURBS. In addition, the evaluation of NURBS is reasonably fast and computationally stable.

In these CNC systems, parametric curves are profiles in different formats like the Bezier curve, B-spline, cubic spline, and NURBS (non-uniform rational B-spline). The general parameter iteration method used is

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

where  $u_i$  is the present parameter,  $u_{i+1}$  is the next parameter, and  $\Delta u_i$  is the incremental value. The interpolated points are calculated by substituting  $u_i$  into the corresponding mathematical model to recover the originally designed curves.

### 5.3 Mathematical formulation

NURBS curve of degree  $p$  defined by control points  $P_0, P_1, \dots, P_n$ , knot vector  $U = \{ u_0, u_1, \dots, u_m \}$ , and weights  $w_0, w_1, \dots, w_n$ . is defined as:

$$C(u) = \frac{1}{\sum_{i=0}^n N_{i,p}(u)w_i} \sum_{i=0}^n N_{i,p}(u)w_i P_i$$

Suppose  $C(u)$  is the parametric curve representation function and the time function  $u$  is the curve parameter such that

$$u(t_i) = u_i \quad \text{and} \quad u(t_{i+1}) = u_{i+1}$$

According to Taylor's expansion if a function  $F(x)$  & its first  $(n-1)$  derivatives are continuous in  $[a, a+h]$  and function exists for every value of  $x$  in  $[a, a+h]$ , then

$$F(a+h) = F(a) + hf'(a) + h^2 f''(a) + \dots + h^n f^n(a)$$

By using the above Taylor's expansion, the approximation up to the second derivative for the parameter  $u$  is given as

$$u_{i+1} = u_i + \left. \frac{du}{dt} \right|_{t=u_i} (t_{i+1} - t_i) + \frac{1}{2} \left. \frac{d^2u}{dt^2} \right|_{t=u_i} (t_{i+1} - t_i)^2 + \text{Higher order Terms} \quad (1)$$

The curve speed  $V(u_i)$  can be represented as

$$V(u_i) = \left\| \frac{dC(u)}{dt} \right\|_{u=u_i} = \left\| \frac{dC(u)}{du} \right\|_{u=u_i} \frac{du}{dt} \Big|_{t=t_i}$$

From the above expression the first derivative of  $u$  with  $t$  is obtained.

$$\frac{du}{dt} \Big|_{t=t_i} = \frac{V(u_i)}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \quad (2)$$

By taking the derivative of the previous equation the second derivative of  $u$  with  $t$  is

$$\frac{d^2u}{dt^2} \Big|_{t=t_i} = \frac{-1}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}^2} \left[ V(u_i) \frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{dt} \right] \quad (3)$$

where

$$\begin{aligned} \frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{dt} &= \frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{du} \cdot \frac{du}{dt} \Big|_{t=t_i} \\ &= \frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{du} \cdot \frac{V(u_i)}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \end{aligned} \quad (4)$$

Now substituting Eq. (4) into Eq. (3), the second derivative of  $u$  is rewritten in this form.

$$\left. \frac{d^2 u}{dt^2} \right|_{t=t_i} = - \frac{V^2(u_i)}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}^3} \left[ \frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{du} \right] \quad (5)$$

Where

$$\frac{d \left\| \frac{dC(u)}{du} \right\|_{u=u_i}}{du} = \frac{\frac{dC(u)}{du} \cdot \frac{d^2 C(u)}{du^2}}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \quad (6)$$

By substituting Eq. (6) into Eq. (5), the second derivative of u becomes

$$\left. \frac{d^2 u}{dt^2} \right|_{t=t_i} = - \frac{V^2(u_i) \cdot \left( \frac{dC(u)}{du} \cdot \frac{d^2 C(u)}{du^2} \right)}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}^4} \quad (7)$$

Interpolation time Ts is given as

$$t_{i+1} - t_i = T_s$$

## 5.4 The First order interpolation algorithm

The first order approximation interpolation algorithm is obtained by substituting Eqs. (2) into Eq. (1). By neglecting the higher order term, the first order interpolation algorithm can be written as

$$u_{i+1} = u_i + \frac{V(u_i) \cdot T_s}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} \quad (8)$$

where  $V(u_i)$  is the feedrate command which is taken as 200 mm/ sec.

## 5.5 The Second order interpolation algorithm

The second order approximation interpolation algorithm is obtained by substituting Eqs. (7) into Eq. (1). By neglecting the higher order term, the first order interpolation algorithm can be written as

$$u_{i+1} = u_i + \frac{V(u_i).T_s}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}} - \frac{V^2(u_i).T_s^2 \left( \frac{dC(u)}{du} \cdot \frac{d^2C(u)}{du^2} \right) \Big|_{u=u_i}}{2 \cdot \left\| \frac{dC(u)}{du} \right\|_{u=u_i}^4} \quad (9)$$

where  $V(u_i)$  is the feedrate command which is taken as 200 mm/ sec.

## 5.6 The speed-controlled interpolation algorithm

### 5.6.1 The compensatory parameter

The first-order and the second-order interpolation algorithms as given by Eq. (8) and Eq. (9) respectively are approximated results obtained by neglecting the higher order terms in Eq. (1). Therefore, the approximation error for those methods is unavoidable and an interpolation algorithm concerning the curve speed is given here. The present interpolation algorithm is based on the first-order approximation interpolation algorithm. To the first order algorithm a compensatory value  $\varepsilon(u_i)$  is added to compensate for the higher order terms as

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

where  $\varepsilon(u_i)$  is the compensatory value and

$$u_{i+1}' = u_i + \frac{V(u_i).T_s}{\left\| \frac{dC(u)}{du} \right\|_{u=u_i}}$$

To precisely calculate the compensatory value  $\varepsilon(u_i)$ . Taylor's expansion for the curve and its speed must be concerned. There are two approximation techniques used in deriving the interpolation algorithm.

The first one is the Taylor's expansion of the parameter  $u$  with respect to the time  $t$  to obtain the first or second order approximation interpolation algorithms, as in Eqs. (8) and (9).

The second one is the Taylor's expansion of curve  $C$  with respect to the parameter  $u$ .

If we represent the curve as

$$C(u) = \begin{bmatrix} C_x(u) \\ C_y(u) \end{bmatrix}$$

in the parametric form for the X and Y axes, then the interpolated points  $C_x(u_{i+1})$  and  $C_y(u_{i+1})$  are approximated as follows:

$$C_x(u_{i+1}) = C_x(u'_{i+1}) + \frac{dC_x(u'_{i+1})}{du} \cdot \varepsilon(u_i) \quad (10)$$

$$C_y(u_{i+1}) = C_y(u'_{i+1}) + \frac{dC_y(u'_{i+1})}{du} \cdot \varepsilon(u_i) \quad (11)$$

By comparing Eqs. (1), (8) and (9) with Eqs. (10) and (11), the higher-order term of the Taylor's expansion in the first approximation is estimated as the compensatory value  $\varepsilon(u_i)$  by applying the second approximation. Eqs. (10) and (11) implies that the difference between position

$$\begin{bmatrix} C_x(u_{i+1}) \\ C_y(u_{i+1}) \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} C_x(u'_{i+1}) \\ C_y(u'_{i+1}) \end{bmatrix}$$

is determined by the slope of curve with the adjustment gain  $\varepsilon(u_i)$ .

Though  $V(u_i)$  is the instantaneous velocity at  $u = u_i$  in the interpolation algorithm, it is usually assigned as the constant feedrate command  $F$  in real interpolation applications.

To accurately achieve a linear motion from

$$\begin{bmatrix} C_x(u_i) \\ C_y(u_i) \end{bmatrix} \quad \text{to} \quad \begin{bmatrix} C_x(u_{i+1}) \\ C_y(u_{i+1}) \end{bmatrix}$$

with the desired speed  $V(u_i)$

$$\frac{\sqrt{[C_x(u_{i+1}) - C_x(u_i)]^2 + [C_y(u_{i+1}) - C_y(u_i)]^2}}{T_s} = V(u_i) \quad (12)$$

the following quadratic equation for the compensatory parameter is provided

$$U\varepsilon^2 + Z\varepsilon + W = 0$$

where,

$$\begin{aligned} U &= X'(u'_{i+1})^2 + Y'(u'_{i+1})^2 \\ Z &= 2[DX.X'(u'_{i+1}) + DY.Y'(u'_{i+1})] \\ W &= DX^2 + DY^2 - (V(u_i).T_s)^2 \end{aligned}$$

$$DX = C_x(u'_{i+1}) - C_x(u_i)$$

$$DY = C_y(u'_{i+1}) - C_y(u_i)$$

$$X'(u'_{i+1}) = \frac{dC_x(u'_{i+1})}{du}$$

$$Y'(u'_{i+1}) = \frac{dC_y(u'_{i+1})}{du}$$

The compensatory value  $\varepsilon(u_i)$  can be directly obtained as

$$\varepsilon_{1,2}(u_i) = \frac{-Z \pm \sqrt{Z^2 - 4UW}}{2U}$$

$$= \frac{-[DX.X'(u'_{i+1}) + DY.Y'(u'_{i+1})] \pm \sqrt{[X'(u'_{i+1})^2 + Y'(u'_{i+1})^2](V(u_i)T_s)^2 - [DY.X'(u'_{i+1}) + DX.Y'(u'_{i+1})]^2}}{X'(u'_{i+1})^2 + Y'(u'_{i+1})^2} \quad (13)$$

### 5.6.2 Selection of compensatory parameters

Eq. (13) will give two roots for the quadratic equation,

$$U\varepsilon^2 + Z\varepsilon + W = 0$$

So for the selection of compensatory parameter characteristics of roots have to be discussed in real applications & accordingly selection has to be made.

Let us define two vectors, Differential vector C & Difference vector D as

$$D = \begin{bmatrix} DX \\ DY \end{bmatrix}$$

$$C = \begin{bmatrix} X'(u'_{i+1}) \\ Y'(u'_{i+1}) \end{bmatrix}$$

&  $\theta$  angle between the two. Their relationship with other parameters is as shown.

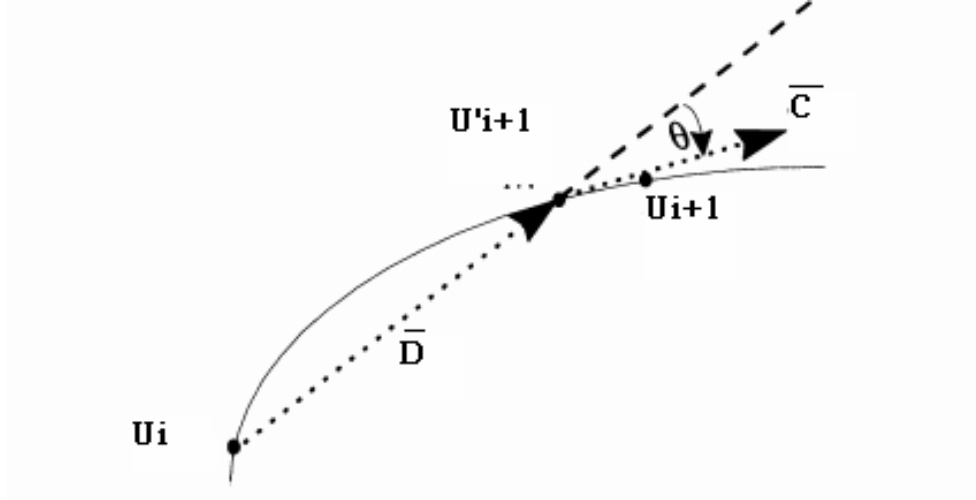


Fig 4: Geometrical representation of parameters.

Then eq. 13 can be re written as

$$\varepsilon_{1,2}(u_i) = \frac{-[\vec{D} \cdot \vec{C}] \pm \sqrt{\|\vec{C}\|^2 \cdot (V(u_i)T_s)^2 - |\vec{C} \times \vec{D}|^2}}{\|\vec{C}\|^2} \quad (14)$$

Calculating the term inside the root we have

$$\begin{aligned} & \|\vec{C}\|^2 \cdot (V(u_i)T_s)^2 - |\vec{C} \times \vec{D}|^2 \\ &= \|\vec{C}\|^2 \cdot T_s^2 \left[ V^2(u_i) - \frac{|\vec{C} \times \vec{D}|^2}{\|\vec{C}\|^2 \cdot T_s^2} \right] \\ &= \|\vec{C}\|^2 \cdot T_s^2 \left[ V^2(u_i) - \left| \frac{C}{\|\vec{C}\|} \times \frac{\vec{D}}{T_s} \right|^2 \right] \end{aligned}$$

$$= \left\| \vec{C} \right\|^2 T_s^2 \left[ V^2(u_i) - \left\| \frac{\vec{D}}{T_s} \right\|^2 \sin^2 \theta \right]$$

As both the terms i.e.

$$\left\| \vec{C} \right\|^2 > 0$$

and

$$T_s^2 > 0,$$

So

$$\left\| \vec{C} \right\|^2 (V(u_i) T_s)^2 - \left| \vec{C} \times \vec{D} \right|^2$$

has the same sign as

$$\left[ V^2(u_i) - \left\| \frac{\vec{D}}{T_s} \right\|^2 \sin^2 \theta \right]$$

The solution of eq 13 can be categorized as:

- $\varepsilon(u_i)$ 's are two different real numbers if

$$\left[ V^2(u_i) > \left\| \frac{\vec{D}}{T_s} \right\|^2 \sin^2 \theta \right]$$

- $\varepsilon(u_i)$ 's are the same real numbers if

$$\left[ V^2(u_i) = \left\| \frac{\vec{D}}{T_s} \right\|^2 \sin^2 \theta \right]$$

- $\varepsilon(u_i)$ 's are a pair of complex conjugate numbers if

$$\left[ V^2(u_i) < \left\| \frac{\vec{D}}{T_s} \right\|^2 \sin^2 \theta \right]$$

Sign of the above expression is controlled by  $\theta$ . Conditions of same real roots & conjugate roots are as shown.

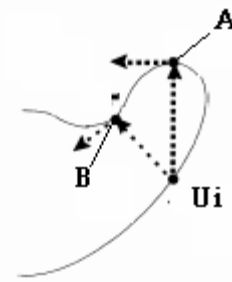


Fig 5: Conditions of same real roots & conjugate roots

We can easily interpret that multiple real roots & the complex conjugate roots exist at the points where the curvature is very large.

The curvature should be small to achieve precise interpolation. So here we deal with only the real roots. So according to eq 14

$$\begin{aligned} \varepsilon_{1,2}(u_i) &= \frac{- \left[ \frac{\vec{D} \cdot \vec{C}}{\|C\|} \right] \pm \sqrt{(V(u_i)T_s)^2 - \left| \frac{\vec{C}}{\|C\|} \times \vec{D} \right|^2}}{\|C\|} \\ &= \frac{- (\|D\| \cos \theta) \pm \sqrt{(V(u_i)T_s)^2 - \|D\|^2 \sin^2 \theta}}{\|C\|} \end{aligned}$$

$$= \frac{-(\|D\| \cos \theta) \pm \sqrt{(V(u_i)T_s)^2 - \|D\|^2 + \|D\|^2 \cos^2 \theta}}{\|C\|}$$

Suppose

$$(V(u_i)T_s)^2 - \|D\|^2 = \mu$$

By applying Taylor's expansion , roots of the quadratic equation are approximated as

$$\varepsilon_1 = \frac{\mu}{2\|C\|\|D\|\cos \theta}$$

$$\varepsilon_2 = \frac{2\|D\|\cos \theta}{\|C\|}$$

The quantity  $\mu$  is very small, so the first root is near zero and the other root is negative and relatively large.

So to achieve reliable compensation & forward motion during the interpolation process, the small compensatory parameter is preferable as

$$\varepsilon(u_i) = \frac{-[DX.X'(u_{i+1}) + DY.Y'(u_{i+1})] + \sqrt{[X'(u_{i+1})^2 + Y'(u_{i+1})^2](V(u_i)T_s)^2 - [DY.X'(u_{i+1}) + DX.Y'(u_{i+1})]^2}}{X'(u_{i+1})^2 + Y'(u_{i+1})^2}$$

This value of the compensatory parameter will be added each time while calculating the next point.

The roots condition

$$\left\| \vec{C} \right\|^2 \cdot (V(u_i)T_s)^2 - \left| \vec{C} \quad X \quad \vec{D} \right|^2$$

has to be examined before calculating the compensatory value each time. For undesirable conditions the compensatory value is set to zero to avoid the complex conjugate roots.

# **Chapter 6**

## **RESULTS & DISCUSSIONS**

## 6.1 Introduction

In this chapter the results of the program are taken applying various conditions as per the requirements. In this thesis work the interpolation algorithm is written in Turbo C 3.0 and is executed on a personal computer with different processing speeds.

Also the interpolation algorithm is applied to a planar NURBS curve with a constant speed feedrate command.

## 6.2 Input Data

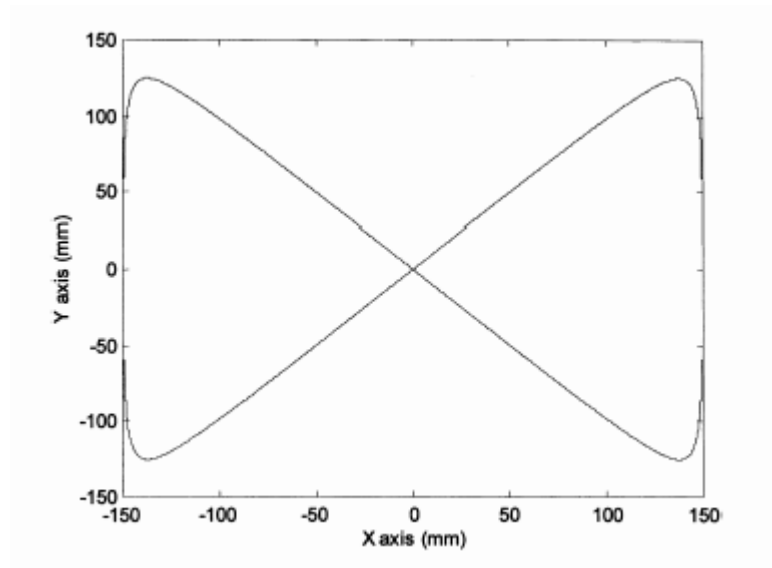
The control points, weight vector, & knot vector of NURBS are assigned as follows

- The ordinal control points are

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} -150 \\ -150 \end{bmatrix}, \begin{bmatrix} -150 \\ 150 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 150 \\ -150 \end{bmatrix}, \begin{bmatrix} 150 \\ 150 \end{bmatrix} \text{ and } \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{ (mm).}$$

- The weight vector is  $W = [1, 25, 25, 1, 25, 25, 1]$ .
- The knot vector is  $U = [0, 0, 0, 0.25, 0.5, 0.5, 0.75, 1, 1, 1]$
- The interpolating processes are as follows
  - The sampling time in interpolation is  $T_s = 0.002$  sec and
  - The feedrate command is  $F=200$  mm / sec =12 m / min.

The NURBS curve generated with the help of above data when no algorithm is applied is shown in Fig. 6



*Fig 6: The example of NURBS*

### **6.3 Color convention**

In the thesis the following color convention is used to make the figures more illustrative. The encircled points are the ordinal points or the control points. The part in dark gray color is the part where the velocity exceeds the given range as specified in the literature, for different interpolation algorithms. The part in light gray means the velocity for that part is within the prescribed limits

Results & discussions corresponding to the above data when the first order, second order & the speed controlled interpolation algorithm are applied..

### **6.4 First order approximation**

When the first order approximation algorithm is applied to the above NURBS curve it takes the shape as shown in Fig. 7. The interpolated time is taken as  $T_s = 0.002$  sec & feedrate 200 mm per sec. The velocity for this feedrate and interpolation time is specified in the range 194-204 mm per second.

From the above plots it is clear that second order approximation achieves the largest speed deviation within [ -0.2, 0.2 ] mm per second. The results agree with those given in the literature.

## **6.6 Speed controlled interpolation**

The speed-controlled interpolation algorithm derives a suitable compensatory parameter for the first-order approximation to obtain desirable motion speed. It is based on the Taylor's approximation.

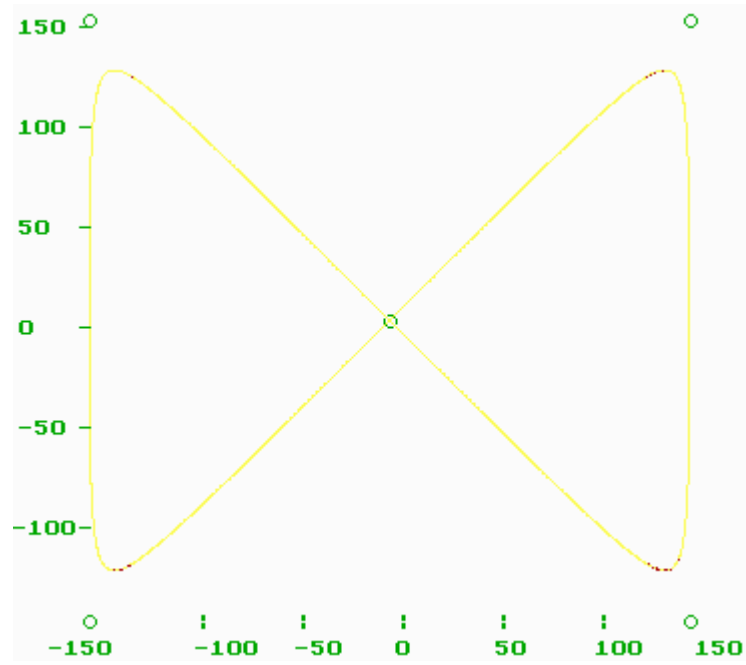
When the speed controlled interpolation approximation algorithm is applied to the above NURBS curve it takes the shape as shown in Fig. 13.

The interpolated time is taken as  $T_s=0.002$  sec & feedrate 200 mm per sec.

The velocity for this feedrate and interpolation time is specified in the range 199.995 - 200.005 mm per second in the literature but because of latest version of software the accuracy achieved here is in the range of  $200 \pm 5e - 12$ .

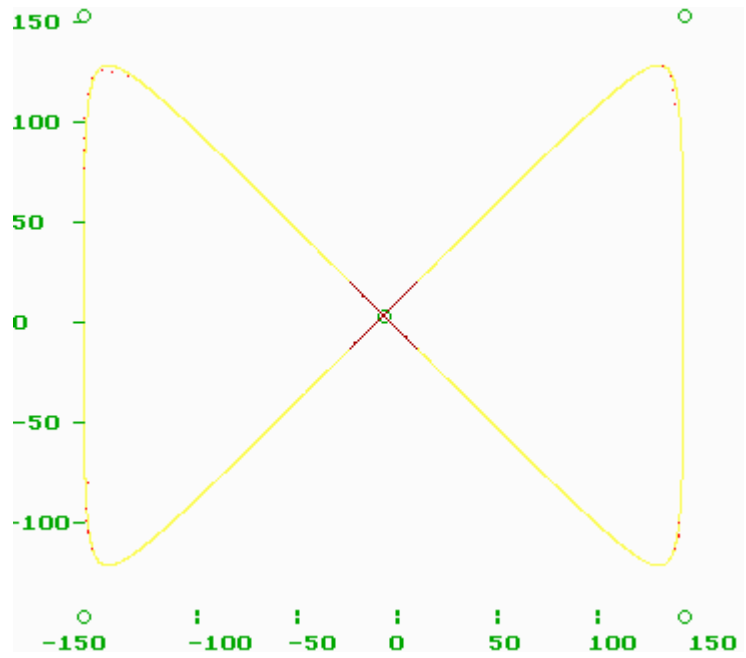
From fig 13 – 16 we can interpret that errors in the desired velocity have increased with increase in the interpolation time or federate & has decreased with the decreasing federate. The deviations in the velocity for different federates is almost constant

The results for different sampling times and federates are summarized at the end. The variations in different parameters i.e. u, x coordinate, y coordinate & velocity corresponding to the interpolation time ( $T_s$ ) 0 .002 seconds and federate 200 mm per second for the Second order approximation are also tabulated for the first 100 interpolating steps



*Fig 13: The example of NURBS at  $T_s=0.002$  sec and feedrate 200 mm/sec*

Impact of different interpolation time & federates is shown in the subsequent figures.



*Fig 14: The example of NURBS at  $T_s=0.003$  sec and feedrate 200 mm/sec*

# **Chapter 7**

## **CONCLUSIONS**

## 7.1 Conclusions

On the basis of the results obtained by simulation of the algorithms for different interpolating time & feedrates the following conclusions can be drawn:

- All interpolation algorithms i.e. first order approximation, second order approximation & speed controlled interpolation result in high contouring accuracy. First order approximation achieves the largest speed deviation within [ -5, 5 ] mm per second. Second order approximation achieves the largest speed deviation within [ -0.2, 0.2 ] mm per second & Speed controlled interpolation achieves it within [ + 5e-12, - 5e-12 ] mm per second. Thus taking into account the control resolution of speed on various axes, the speed controlled algorithm gives the best desirable results.
- Deviations from the desired velocity increases with increase in the interpolation time and vice versa. This increase is too much for the first order & the second order approximations while for the speed controlled interpolation the deviation is much less.
- Deviations from the desired velocity increases with increase in the feedrate and vice versa. This increase is again high for the first order & the second order approximations. In speed controlled interpolation the number of deviations remain in the range of 1 to 2 percent.
- Abrupt deviations in velocity of the curve are because of uneven weights specified at the control points. By applying the uniform weights the deviations in the velocity are greatly reduced. More uneven the weights more are the deviations in the cutting speed & vice versa.

## **7.2 Scope For Further Work**

The present work is carried out with constant feedrate command. This could be further extended by taking into account variable feedrates. The variable feedrate is a more realistic approach as the acceleration and deceleration profiles may be specified to achieve better machining quality.

## REFERENCES

### Books:

1. Koren, Y., Computer Control of Manufacturing Systems, McGraw Hill, New York, New York, 1983.
2. Koren, Y., Robotics for Engineers, McGraw Hill, New York, New York, 1985.
3. Koren, Y., Numerical Control of Machine Tools, Khanna Publishers, New Delhi, India, 1984.
4. Piegl L, Tiller W. The NURBS book. Berlin: Springer, 1997.
5. James D. Foley et al., "Introduction to Computer Graphics", 1994, Addison-Wesley.
6. Computer Graphics: Principles and Practice, by Foley, Van Dam, Feiner, and Hughes.
7. CAD / CAM Theory & practice , by Ibrahim Zeid, McGraw Hill 1998.
8. Computer Graphics, by Donald Hearn & M. Pauline Baker, Prentice-Hall.
9. David F. Rogers, Rae A. Earnshaw (editors), "State of the Art in Computer Graphics - Visualization and Modeling", 1991, New York, Springer-Verlag, pp. 225 - 269

### Journals & Papers:

1. Bedi S, Ali I, Quan N. Advanced techniques for CNC machines. Trans ASME J Engg Industry 1993 vol 115 pages 329–336.
2. Chou JJ, Yang DCH. Command generation for three-axis CNC machining. Trans ASME J Engg Industry 1991; vol 113: pages 305–310.
3. Huang JT, Yang DCH. A generalized interpolator for command generation of parametric curves in computer controlled machines. Japan/USA Symposium on Flexible Automation 1992; volume 1: pages 393–399.
4. Shpitalni M, Koren Y, Lo CC. Realtime curve interpolators. Computer-Aided Design 1994 vol 26 pages 832–838.
5. Lo CC, Chung CY. Curve generation and control for biaxial machine tools. J. CSME 1997; vol 18: pages 175–182.
6. The speed-controlled interpolator for machining parametric curves, S.-S. Yeh, P.-L. Hsu / Computer-Aided Design 31 (1999) pages 349–357
7. Kim DI. Study on interpolation algorithms of CNC machine tools. IEEE Industry Applications Conference 1995; vol 3: pages 1930–1937.

8. Young Joon Ahn, Hong oh kim, Approximate implicitization of planar parametric curves using quadratic splines, *Computer-Aided Design* 1990 vol 26 pages 83–88.
9. Koren, Y., Lo, C-C., and Shpitalni, M., *CNC Interpolators: Algorithms and Analysis*, ASME Manufacturing Sciences and Engineering, PED-Vol. 64, 1993, pp. 83-92.
10. Lim, F. S., Wong, Y. S., and Rahman, M., Circular Interpolators for Numerical Control: A Comparison of the Modified DDA Techniques and an LSI Interpolator, *Computers in Industry*, Vol. 18, 1992, pp. 41-52.
11. Yang DCH, Kong T. Parametric interpolator versus linear interpolator for precision CNC machining. *Computer-Aided Design* 1994; vol 26(3): pages 225–233.
12. Chou JJ, Yang DCH. On the generation of coordinated motion of five-axis CNC / CMM machines. *Trans ASME J Engg Industry* 1992; vol 114: pages 15–22.
13. Low Level Direct Interpolation for Parametric Curves; Oscar n Ruiz, Adriana n Martínez, Elizabeth n Rendón.
14. Planar point set fairing and fitting by arc splines, Xunnian Yang, Wang, *journal CAD*, vol 33, 2001, pages 35 -43.
15. Biarc approximation of NURBS curves, Les A Piegi, Tiller, *journal CAD*, vol 34, 2002, pages 807-814.
16. Farouki RT, Manjunathaiah J, Nicholas D, Yuan G-F, Jee S. Variable feedrate CNC interpolators for constant material removal rates along Pythagorean-hodograph curves. *Computer Aided Design* 1998; vol 30: pages 631 ± 40.
17. Qiu, H., Cheng, K., and Li, Y., Optimal Circular Arc Interpolation for NC Tool Path Generation in Curve Contour Manufacturing, *Computer Aided Design*, Vol. 29, No. 11, 1997, pp. 751-760.
18. Koren, Y., and Masory, O., Reference Pulse Circular Interpolators for CNC Systems, *ASME Journal of Engineering for Industry*, Vol. 108, 1981, pp. 131-136.
19. Masory, O., and Koren, Y., Reference Word Circular Interpolators for CNC Systems, *ASME Journal of Engineering for Industry*, Vol. 104, No. 4, 1992, pp. 400-405.
20. Lo, C-C. Feedback Interpolators for CNC Machine Tools, *ASME Journal of Manufacturing Science and Engineering*, Vol. 119, 1997, pp. 587-592.

