Design of Smooth Ramp Feedrate for Machining Complex NURBS Paths

M. Sekar*
Department of Mechanical Engineering
GMR Institute of Technology
Rajam, India
mailto:sekar@gmail.com

B. Suresha
Department of Mechanical Engineering
National Institute of Engineering
Mysuru, India
sureshab2004@yahoo.co.in

I. Kantharaj
Department of Mechanical Engineering
Karunya University
Coimbatore, India
kantharaj@karunya.edu

Abstract:
NURBS curves are widely adopted in CAD for the purpose of aesthetics and became functional requirements. In machine tool controllers, the position commands of NURBS tool paths are obtained from incremental parametric values. For technical reasons, the tool must be commanded to reach the parameter value corresponding to the end point of the free-form curve; however this is not guaranteed by the Taylor’s expansion. This aspect restricts the implementation of a free-form parametric interpolator for NURBS machining in actual machining practice. In this paper a non-uniform parametric algorithm is proposed to find the ramp increments. In this approach the difference between the computed and the end parameter values is represented as an area of triangle in the time domain with its base equal to the integer values of sampling interval set by the machine controller. Results show that the proposed simulator designed can reach the end parameter value exactly. The novel ramp feedrate method suggested in this paper is also useful for multi-pass cutting in CNC machine.

Keywords: Parametric NURBS, CNC controller, smooth ramp profile, SIMULINK, Real-Time Workshop.

INTRODUCTION

In CNC machines controller outputs the position commands for a parametric NURBS curve using approximation methods. Computer aided manufacturing (CAM) software employ techniques that divides the NURBS into linear G codes segments. However such techniques result in constraints on resources, fluctuating feedrate leading to stability problems and further polishing work to achieve the desired surface finish [1]. In addition the coordinate points computed by the interpolator may not actually represent the required NURBS curve designed in CAD.

The real time NURBS interpolator introduced by Shiptalni et.al has gained wide acceptance among NURBS implementations as the CAD geometry and can directly be employed for interpolation while CNC machining of components designed with NURBS [2]. In particular the parametric incremental interpolators proposed by [3-6] does not need constant increments of chord length. One of the unsolved issues in
parametric curve machining is to reach the end coordinate value irrespective of the given feedrate and control parameters. Computing parameter value from a coordinate is not feasible with NURBS as closed loop mathematic equations are not yet developed. A mismatch between the manufactured product and the NURBS design will be inevitable if the end parameter value corresponding to the final point NURBS is not met. In this article the solution to this problem is solved by representing the difference \((du)\) between the computed parameter and the end parameter as an area of triangle in the time domain. Here \(du\) the area of triangle has the integer multiple of controller sampling intervals.

After the brief overview of NURBS is outlined, the Taylor’s series for real-time parametric interpolation is introduced. The problems pertaining to the reach the end parameter value is then discussed. The solution to the above problem is proposed in the next section followed with details of the development of simulator. Information on development of SIMULINK blocks and implementation in real-time workshop is presented in this article.

FEEDRATE DESIGN FOR NURBS TOOL PATHS

A NURBS curve is a piecewise parametric curve represented by a rational polynomial [7]

\[
C(u) = \sum_{i} N_{i,p}(u) w_i P_{i,n} \quad \text{with} \quad 0 \leq u \leq 1 \quad \text{and} \quad w_i > 0
\]  

where the control polygon consists of \(P_i\) control points, weight vectors with values \(w_i\) and the \(N_{i,p}(u)\) are the \(p\)th degree B-spline basis functions defined on non-uniform knot vector \(U\)

\[
U = \{0, \ldots, 0, u_{p+1}, \ldots, u_{m-p}, 1, \ldots, 1\}
\]

The basis function of the spline with degree \(p\) can be recursively written as

\[
N_{i,p}(u) = \begin{cases} 
1 & \text{if} \quad u_{i+p} \leq u \leq u_{i+1} \\
0 & \text{otherwise} 
\end{cases} 
\]

while the relation between \(m\), \(n\) and \(p\) can be represented as

\[
m = n + p + 1.
\]

The derivative of the basis function is widely used for determining the increments and is given by

\[
N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p}(u) + \frac{p}{u_{i+p} - u_i} N_{i+1,p}(u)
\]

A. Real time Interpolator for CNC machines

NURBS interpolation can result in the uniform of non-uniform increment to the parameter values. The values obtained are substituted in Eqn.1 to determine the corresponding command position. In contrast to the conventional CNCs, NURBS position commands are obtained by considering both parametric domain \((u)\) and time domain \((t)\). However there exists no mathematical functional relationship between the \(u\) and \(t\) domains. Hence the real time interpolator use numerical integration methods to find the next immediate parameter for every sampling period. For a NURBS curve represented by Eqns. 1-6, the incremental change of parameter value in time domain is obtained from the command velocity \(V(u)\) as follows:
\[ V(u) = \left[ \frac{dC(u)}{dt} \right] \]  

Differentiating the parameter \( u \) with respect to time

\[ \frac{du}{dt} = \left[ \frac{dC(u)}{du} \right] \]  

Applying Taylor’s series for every sampling interval \( T_s \), \( t=kT_s \) with \( k \) as the number of samples the next incremental value to parameter can be computed as

\[ u_{i+1} = u_i + \Delta T \hat{u} \Bigg|_{u_i} + \frac{(\Delta T)^2}{2} \hat{u}'' \Bigg|_{u_i} + \frac{(\Delta T)^3}{6} \hat{u}''' \Bigg|_{u_i} + \ldots \]  

\[ \hat{u} = \frac{V}{\sigma} \]  

\[ \hat{u}' = \frac{\sigma' \dot{V} - \sigma \dot{V}' \hat{u}}{\sigma^2} = \frac{\dot{V}}{\sigma} - \frac{\sigma' \dot{V} \hat{u}}{\sigma^2} \]  

\[ \hat{u}'' = \frac{\ddot{V}}{\sigma} - \frac{3\sigma' \dot{V} \hat{u}}{\sigma^2} - \frac{\sigma' \dot{V} \hat{u}^2}{\sigma^2} \]  

On revisiting the Taylor’s equation for NURBS interpolation and the recursive equations to compute the derivatives of parameter in time domain, the first and second order derivatives of instantaneous parameters are necessary [8].

### B. Investigation of Parametric Interpolator

In real-time CNC controller design, the X and Y positions of a parametric curve do not use the HOTs for computation because of the recursive polynomial functions of NURBS. A continuous feedrate within the limits of acceleration and jerk is preferred for better machine dynamics [9]. Smooth feedrate profile results in a coordinated motion with reduced tracking errors due to rapid and accurate execution of low-jerk trajectory. Though there are many practical issues in the implementation of a real-time parametric interpolator, this paper discusses one of the key issues regarding interpolating exactly until the end parameter value of the given curve.

It is a standard practice to normalize the knot vector from 0-1. In CNC controller the initial parameter value equal to 0 is substituted in equation 1 to find the coordinates. To generate the coordinate position for the next sampling time, \( u_{i+1} \) is required which can be computed by implementing Eqn. 9 in the CNC controller. The process is iterated with \( k \) times until the parameter reaches the maximum value of \( u=1 \).

Since the Taylor’s interpolator computes increments to the parameter non-linearly, the end parameter value i.e. \( u=1 \) may not reached at all times. Considering the geometry of the NURBS curve, large distances may be left to cover even for small incremental change in parameter values. This results in a machining process where the final coordinates are left un-machined leaving an undercut or overcut in the components. A reduction in surface accuracy, dimensional deviation and assembly problems are inherent in this method.

### C. Proposed Solution for Parametric Interpolator with smooth ramps

In the proposed non-uniform parametric increment algorithm, the change (\( du \)) between the computed \( u_{\text{end Taylor}} \) and \( u=1 \), is represented as an area of triangle in the time domain (Fig. 1). The objective is to find cumulative sum of area at each sampling interval. The number of sampling intervals represents the width \( (B) \) of the triangle and can be provided as an input value. The instantaneous value of ‘\( h \)’ can be found from
similar triangles, where ‘\( H \)' is the height of the triangle. Referring to Fig.1, the difference between the final parameter \( u=1 \) and the starting parameter is mentioned as \( du \). In time domain \( du \) is expressed as the area of triangle with the base equal to \( n \) multiples of controller sampling time. A triangular profile is chosen because the increments in \( u \) can be increased and decreased smoothly in the parametric domain.

![Fig. 1. Remaining parameter \( u \) represented in time domain.](image1)

The area of the triangle in Fig. 2 represents the \( du \). At each time interval \( b \) of total time \( B \), area can be obtained based on similar triangles \( 0-b-h \) and \( 0-B/2-H \). The parameter value corresponding to area at \( b \) can be obtained as the sum of starting parameter \( (1-du) \) and the area \( a_b \).

![Fig. 2. Remaining \( u \) during first half of travel.](image2)

\[ u_s = (1 - du) + a_s \text{ where } a_s = 0.5bh \]  

(13)

Using similar triangles, we get,

\[ u = (1 - du) + \frac{b^2}{B^2} \frac{2du}{B} \text{ (if } b \leq B/2) \]  

(14)

The increase in \( b \) results in smooth increase of \( u \) value. Similarly for the other condition \( (b > B/2) \) the following formula can easily be derived.

\[ u = 1 - \frac{b^2}{B^2} \frac{2du}{B} \text{ (if } b > B/2) \]  

(15)

D. Implementation in SIMULINK and Real-Time Workshop

MATLAB provides a power tool namely the Real-Time Workshop which extends the capabilities of both SIMULINK and MATLAB functionalities to a target hardware [10]. Generally the target hardware is not compatible to MATLAB \( m \) files and the proposed CNC architecture requires special blocks which are not
available in the SIMULINK. Therefore customized codes are written in C to develop the interpolator block set. In this paper details regarding the real-time interpolator block are discussed.

The algorithm consisting of Eqns. 1-15 are implemented as level 2 SIMULINK S function. The Real-Time Workshop of MATLAB has the target link compiler to convert the MATLAB code into executable. With the automatic code generation tailored for a variety of target platforms, these codes can easily be deployed with RTW. RTW compatible SIMULINK blocks support platform independency and promote rapid development and deployment of systems. In addition the integration with MATLAB provides sharing of data to the workspace so as to enable plotting and carry out other analysis in MATLAB command prompt.

Fig. 3 shows the interpolator SIMULINK block implemented for the purpose of simulation and real-time execution in a suitable hardware target. The code generation report from the RTW has a list of various files generated to aid the real-time implementation of the SIMULINK block in a hardware platform. The files include source files, header files, data type definition files and object codes. The simulator consists of scopes and functions to export the computed values to MATLAB workspace for easy plotting. To enable diagnosis of the algorithm in real-time, global variables compatible to the hardware platform are defined. The corresponding header file of the target hardware is also essential for the successful uploading of the object codes into the controller memory.

E. Testing the block in SIMULINK

For testing of the interpolator block and the outputs, a circle is modeled as a parametric NURBS curve. Following figures show the outputs obtained from the interpolator SIMULINK block. In real-time implementation the coordinate data is sent to X, Y stage loop controllers to generate the required profile. The data corresponding to the outputs are also exported to workspace for the purpose of analysis. Fig. 4 shows the plot constructed using the data from the workspace and the Figs 5. a-c show the scope graphs taken from SIMULINK model.
Fig. 4. Parametric circle profile generated from workspace data.

Fig. 5 a, b, c. Outputs from SIMULINK Interpolator block.  
(Coordinates x, y, and parameter u plotted against time, Coordinates are in micro meters)
RESULTS AND DISCUSSION

Fig. 6 shows the sample NURBS parametric curve used to test the proposed algorithm with RTW generic target. The control points, weight vectors and knot vectors are listed in Table 1. The given curve has a degree of 2 so that second Taylor’s approximation can be used in CNC interpolation except for the smooth ramping at the end. The important simulation parameters for the controller are shown in Table 2. A sampling time of 10 milli seconds has been selected for the present simulation purpose. If the curve involves higher degree number of recursive iterations will increase and a small sampling time may not be enough to compute the coordinate points in real time. The developed code has features to change the sampling time after testing the entire code in a real-time system.

| X(μm) | Y(μm) | W | U   |
|-------|-------|----|-----|
| 10000 | 0     | 1  | 0,0 |
| 20000 | 20000 | 1  | 0.2 |
| 12000 | 8000  | 1  | 0.4 |
| 10000 | 20000 | 1  | 0.6 |
| 8000  | 8000  | 1  | 0.8 |
| 0     | 20000 | 1  | 1   |
| 10000 | 0     | 1  | 1,1 |

Table 2: Simulation parameters

- Degree of the curve: 2
- No. of derivatives: 2
- Sampling time: 0.01 sec
- Feedrate limit: 25000 mm/sec
- Acceleration limit: 5000 μm/sec²
- Jerk limit: 5000 μm/sec³
- Chord error: 0.5μm

With the initial feedrate of \( V=1.5 \times 10^4 \) μm/sec the Taylor’s interpolator reaches the parameter value 0.9984. Non-uniform increments based on the remaining \( u (du=0.0016) \) are computed using the ramping algorithm. The general interpolator has no inherent feedrate planning and hence cannot be directly implemented in the controller since the controller cannot start with the command feedrate from the rest position. It is assumed that the solution for this problem exists and the feedrate schedules are available for the curve as shown in Fig. 7.
Fig. 6. Parametric NURBS curve

Fig. 7. Ramping algorithm in time and parameter domains

Table 3: Comparison of NURBS curve data

| u     | X(µm) | Y(µm) |
|-------|-------|-------|
| 0.9984 | 0     | 1     |
| 1     | 10000 | 0     |
| Δu=0.0016 | Δx=151 | Δy=302 |

It can be noticed if the algorithm is not used, the coordinate point corresponding to $u=1$ may never be reached and a distance of $\Delta x$ and $\Delta y$ will be left unvisited by the machine tool (Table. 3). This may result in poor surface finish and may even lead to broken tool when high axial depth of cuts is employed. Moreover, a sharp profile will not be machined accurately and the desired curvature may get altered. Though this can be overcome with a shorter sampling time, reduction of sampling time involves complexity in real time calculations of NURBS coordinate. In addition, the CNC controller also must allocate time to compute control loops of the individual motors. It also requires time slice to process the feedback data in real-time. Hence the proposed method is practically suitable for real time implementations.
CONCLUSIONS

A novel algorithm for the smooth reaching of the end parametric value in real-time interpolation of a NURBS parametric curve is proposed. The proposed algorithm is integrated to a CNC architecture using SIMULINK and simulation results are shown in this paper. Results show that with the use of proposed algorithm the parameter value can be reached exactly. In addition, the design of CNC interpolator as SIMULINK block with RTW permits rapid development and deployment of the CNC architecture to a variety of hardware platforms. The developed interpolator SIMULINK blocks can be used in both simulation and real-time environments.

REFERENCES

[1] M. Sekar, J. Srinivas, N. Kang and S.H. Yang, “An Integrated Approach to the Improvement of Stability Lobes,” Int. J. of Precision Engineering and Manufacturing, Vol. 9, pp: 83-85, 2008.
[2] M. Shiptalni, Y. Koren, and C. C. Lo, “Real-time curve interpolators.”, Computer Aided Design, Vol. 26, No. 11, pp. 832-838, 1994.
[3] M. Y. Cheng, M. C. Tsai and J. C. Kuo, “Real-time NURBS command generators for CNC servo controllers.” International Journal of Machine Tool and Manufacture, Vol. 42, No. 7, pp. 801-813, 2002.
[4] S. S. Yeh and P. L. Hsu, “Adaptive-feedrate interpolation for parametric curves with a confined chord error.” Computer-Aided Design, Vol. 34, No. 3, pp. 229-237, 2002.
[5] M. Sekar, S. H. Kweon, D. M. Lee, S. H Yang, “Parameteric NURBS curve Interpolators : A review.” International Journal of Precision Engineering and Manufacturing, Vol. 9, No. 2, pp. 84-92, 2008.
[6] Yong, T., Narayanasami, R. “A parametric interpolator with confined chord error, acceleration and deceleration for NC machining”, Computer-Aided Design, Vol.35, pp.1249–1259, 2003.
[7] L. Piegl, and W. Tiller,The NURBS book, Springer, 1997.
[8] Kevin M. Nittler and Rida T. Farouki, “Efficient high-speed cornering motions based on continuously-variable feedrates”, International Journal of Advanced Manufacturing Technology, Vol.37, No. 5 pp. 545–552, 2016.
[9] Zhen-yuan Ji, De-ning Song, Jian-wei Ma, Guo-qing Hu and Wei-wei Su, “A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints”, International Journal of Machine Tools & Manufacture, Vol. 116, pp. 1-17, 2017.
[10] MATLAB and SIMULINK manuals, 2008.