NURBS Interpolator with Confined Chord Error and Tangential and Centripetal Acceleration Control

Alessandro Bardine, Stefano Campanelli, Pierfrancesco Foglia, Cosimo Antonio Prete
Dipartimento di Ingegneria dell'Informazione, Università di Pisa
Via Diotisalvi 2, 56126 Pisa, Italy
{alessandro.bardine, stefano.campanelli, foglia, prete}@iet.unipi.it

***Abstract*—NURBS interpolation is highly requested in CNC systems because it allows high speed and high accuracy machining. In this work, an algorithm for NURBS interpolation capable of limiting chord error, centripetal acceleration and tangential acceleration is proposed. The algorithm is composed of two stages that may be executed simultaneously. In the first stage, the algorithm breaks down the curve into segments and, for each segment, calculates the feedrate limit that allows to respect both chord error tolerance and maximum centripetal acceleration limit. The second stage is a speed-controlled interpolator with a tangential acceleration limited feedrate profile generated using information provided by first stage. Software simulations are performed to verify the fulfillment of constraints and performance is compared to those of speed-controlled interpolator and variable feedrate interpolator.**

Keywords-NURBS interpolation; CNC; feedrate; acceleration control; look-ahead

I. INTRODUCTION

In modern manufacturing systems, processing of a workpiece goes through three phases. In the first phase, CAD (Computer Aided Design) software is used to model the shape of the workpiece. In the second phase, CAM (Computer Aided Manufacturing) software is used to analyze the model and generate the program code for CNC (Computer Numerical Control) machines. The third phase is the actual machining of the workpiece in the CNC system.

While modern CAD software supports free-form design of curves, conventional CNC machines only support motion along straight or circular paths, so the CAM has to approximate the original tool path with a sequence of short linear segments. A weakness of this approach is the fact that, at junctions of two consecutive segments, there are corners that cause feedrate fluctuations because they must be machined at low feedrate. In turn, feedrate fluctuations cause mechanical vibrations that degrade machining quality. Besides, the amount of data needed to approximate the curve is very large and increases with tighter approximation tolerance. This results in data storage and transmission issues [1].

Parametric curve interpolators may overcome these limitations. They allow exact tool-path specification, removing sharp corners. Consequently, they allow the achievement of higher feedrate along the curve, reducing the fluctuations caused by deceleration at sharp corners. Non-Uniform Rational B-Splines (NURBS) [2] are particular parametric curves that have become the industry de facto standard for representing and designing shapes because they can represent both free-form and analytical shapes. Furthermore, algorithms for manipulating and computing NURBS are efficient and numerically stable [3].

First approaches to NURBS interpolation utilized uniform increments of the parameter of the curve [4] without offering any control on tool speed. Then, real-time speed-controlled NURBS interpolators for CNC systems were developed [1][5][6] and allowed the machining of NURBS at constant feedrate. Variable feedrate algorithms [7][8] were developed successively. These algorithms adjust the feedrate during interpolation, slowing down when the curvature of the path increases and, thus reducing the chord error (i.e. the distance between the actual curve and the path generated by the interpolator). Existing variable feedrate algorithms lack in acceleration control, so there is not any means of limiting axis solicitations. This may cause exceeding of axis acceleration capabilities and excessive structure vibrations. Interpolators with both confined chord error and acceleration/deceleration control are being developed. Algorithm in [9] detects high curvature zones and builds an adequate feedrate profile, however the sensitive corners detection is done offline. The real-time interpolator proposed in [10] keeps a buffer of interpolated points for look-ahead purposes. When a high curvature zone is detected, part of the buffer must be recalculated. Since execution time of this operation is not constant but depends on variable factors, it is difficult to evaluate the actual processing resources needed for real-time execution of the algorithm. A real-time look-ahead interpolator capable of generating a jerk-limited feedrate profile was presented in [11]. This interpolator implements a look-ahead module that scans the NURBS, searching for points where curvature function of the curve has local minima and maxima, and builds the feedrate profile. That profile is used to command a real-time speed-controlled NURBS interpolator.

In this work, a NURBS interpolator with confined chord error and acceleration/deceleration control is proposed. Unlike the above cited algorithms, where acceleration control, if present, is limited to tangential acceleration, the proposed algorithm allows to limit both centripetal and tangential acceleration, since this is a feature desired by manufacturing companies. The algorithm breaks down the curve into segments, each one with a different feedrate limit based on its curvature. For each segment, a feedrate profile that respects acceleration limits is generated and used for the interpolation.

The presented algorithm is evaluated through simulations to verify that chord error tolerance and acceleration limits are respected. Performances of the proposed algorithm are compared to those of improved versions of two classical algorithms: the speed-controlled interpolator [1] and the variable-feedrate interpolator [7].

II. NURBS INTERPOLATION

In this section, basic concepts on NURBS representation are provided and the classical method of NURBS interpolation is presented. Besides, it is shown how curve geometry and feedrate affect machining quality and basic behavior of acceleration/deceleration control in CNC systems.

A. NURBS representation

The parametric representation of a curve is a function that links a parameter u to a position vector. The parametric representation of a NURBS is defined by a set of weighted control points and a knot vector and by its degree. NURBS are represented by the following equation:

$$\mathbf{C}(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \mathbf{P}_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \quad u \in [u_0, u_m], \tag{1}$$

where $\mathbf{P}_i$ are the $n+1$ control points, w_i the weights, $N_{i,p}(u)$ the p^{th}-degree basis functions and p is the degree of the NURBS [3]. The knot vector $\{u_0, \ldots, u_m\}$ is a non-decreasing sequence of real numbers called knots, where $m = n + p + 1$. The basis function $N_{i,p}(u)$ is recursively defined as

$$\begin{aligned} N_{i,0}(u) &= \begin{cases} 1 & \text{if } u \in [u_i, u_{i+1}) \\ 0 & \text{otherwise} \end{cases} \\ N_{i,p}(u) &= \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \end{aligned} . \tag{2}$$

The first derivative of (1) is

$$\mathbf{C}'(u) = \frac{\sum_{i=0}^{n} N'_{i,p}(u) w_i \mathbf{P}_i - \mathbf{C}(u) \sum_{i=0}^{n} N'_{i,p}(u) w_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i}, \tag{3}$$

where the basis functions derivatives are calculated as

$$N'_{i,p}(u) = \frac{p}{u_{i+p} - u_i} N_{i,p-1}(u) - \frac{p}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) . \tag{4}$$

B. Real-time speed-controlled interpolator

The real-time interpolator is a task that periodically produces reference values for the position control loop of servomotors [12]. At k^{th} interpolation cycle, the reference position is calculated as $\mathbf{C}(u[k])$ using (1). The parameter value $u[k]$ is computed using the Taylor series expansion method [1]. Let $u(t)$ be the function that expresses parameter value over time, then $u[k] = u(kT)$, where T is the period of each interpolation cycle. Using Taylor series expansion, $u[k+1]$ can be expressed as:

$$u[k+1] = u[k] + T\dot{u}[k] + \frac{T^2}{2}\ddot{u}[k] + \cdots . \tag{5}$$

Truncating the series at the first order term[1], $u[k+1]$ can be approximated as:

$$u[k+1] = u[k] + T\dot{u}[k] . \tag{6}$$

Feedrate v can be considered as the derivative of travelled distance s with respect to time t. So we can use the chain rule:

$$v = \frac{\mathrm{d}s}{\mathrm{d}t} = \left(\frac{\mathrm{d}s}{\mathrm{d}u}\right)\left(\frac{\mathrm{d}u}{\mathrm{d}t}\right) \tag{7}$$

to calculate the derivative of $u(t)$ as

$$\frac{\mathrm{d}u}{\mathrm{d}t} = \frac{\mathrm{d}s/\mathrm{d}t}{\mathrm{d}s/\mathrm{d}u} = \frac{v}{\sigma}, \tag{8}$$

where σ is the parametric speed expressed as

$$\sigma = \frac{\mathrm{d}s}{\mathrm{d}u} = \sqrt{x'^2 + y'^2 + z'^2}, \tag{9}$$

where x, y and z are the components of vector $\mathbf{C}(u)$ and their derivatives are considered with respect to u. Substituting (8) and (9) in equation (6), the formula becomes

$$u[k+1] = u[k] + \frac{vT}{\sqrt{x'^2 + y'^2 + z'^2}}, \tag{10}$$

So, using (10) at each interpolation cycle, the interpolator can generate $u[k+1]$ at desired feedrate taking into account the first order derivative of the curve.

---

[1] Truncation error causes feedrate fluctuations. In order to reduce them, higher order Taylor series truncations can be used. For the sake of simplicity, only the first order truncation is considered in following explanation, but formulae up to third order can be found in [13].

C. Geometry-based feedrate variation

Fig. 1 shows radial and chord error. Radial error is the distance between an interpolated point and the actual curve. Since interpolated points are computed using an exact formula, this error depends only on CPU precision. Chord error is the maximum distance between the line segment (chord) that connect two consecutive interpolated points and the correspondent arc on the curve [7].

The choice of tool machining speed must consider the geometry of the path in order to reduce chord error and axis solicitations. When conventional CNC approach is used, path is specified with linear or circular blocks and CAM software determines the maximum feedrate for each block. With NURBS interpolation, however, it is not recommendable to use the same feedrate for an entire NURBS block because geometric properties of a NURBS may highly vary along the curve.

Variable feedrate interpolators have been developed to allow high speed machining with limited chord error [7][8]. Those algorithms adjust feedrate during the interpolation according to the curvature of the curve at the interpolation point. Curvature is a geometric property that measures how much the NURBS deviates from its tangent and is calculated as

$$K = \left| \frac{\sqrt{(y'z'' - z'y'')^2 + (z'x'' - x'z'')^2 + (x'y'' - y'x'')^2}}{\sqrt{(x'^2 + y'^2 + z'^2)^3}} \right| . \quad (11)$$

According to [7], chord error ER may be estimated using

$$ER = \rho - \sqrt{\rho^2 - \left(\frac{vT}{2}\right)^2} , \quad (12)$$

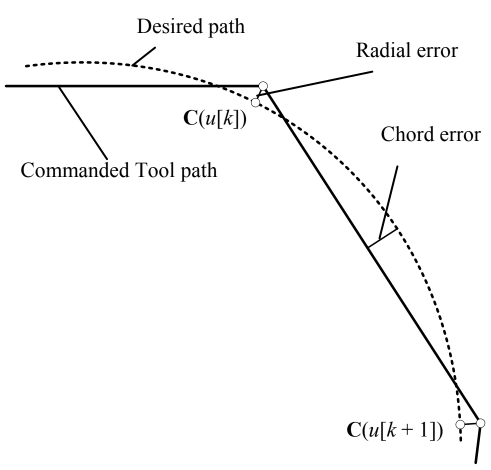

Figure 1. Radial error and chord error.
Solid line is the path generated by the interpolator, while dashed line is the desired path specified by the NURBS.

where ρ is the radius of curvature defined as the reciprocal of curvature.

$$\rho = \frac{1}{K} \quad (13)$$

Equation (12) shows that chord error increases when curvature or feedrate increases so, to limit chord error, the feedrate must be lowered as curvature increases. Maximum feedrate that allows to obtain a chord error smaller than a given tolerance $ER_{\max}$ can be calculated from (12) as

$$v_{chord} = \frac{2}{T}\sqrt{\rho^2 - (\rho - ER_{\max})^2} . \quad (14)$$

Centripetal acceleration is calculated as

$$a_c = \frac{v^2}{\rho} . \quad (15)$$

Like the chord error, the centripetal acceleration also increases as curvature and feedrate increase. It is important that acceleration limits are respected because high acceleration results in mechanical vibrations that degrade machining quality and deviation from desired path. From (15), the maximum feedrate that allows to respect the centripetal acceleration limit $a_{c\max}$ is calculated as

$$v_{acc} = \sqrt{a_{c\max}\rho} . \quad (16)$$

In order to respect both chord error and centripetal acceleration limit, feedrate cannot exceed

$$v_{\max} = \min(v_{acc}, v_{chord}) . \quad (17)$$

D. Acceleration/deceleration control

Acceleration/deceleration control techniques are used to limit tangential acceleration of the tool, preventing abrupt feedrate variations. Two methods of acceleration/deceleration control exists: ADCAI (Acceleration/Deceleration Control After Interpolation) and ADCBI (Acceleration/Deceleration Control Before Interpolation) and the latter is more accurate [12]. Using ADCBI method an acceleration-limited feedrate profile is generated, then the speed-controlled interpolator follows that profile [12]. Different profile shapes can be used to limit acceleration; in this work the trapezoidal profile (Fig. 2) is considered.

In conventional CNC machining, tool path is specified by a sequence of linear and circular blocks. In order to avoid the stop of the tool at the end of each block, look-ahead algorithm was developed. Look-ahead calculates the ending speed of a block analyzing the following blocks.

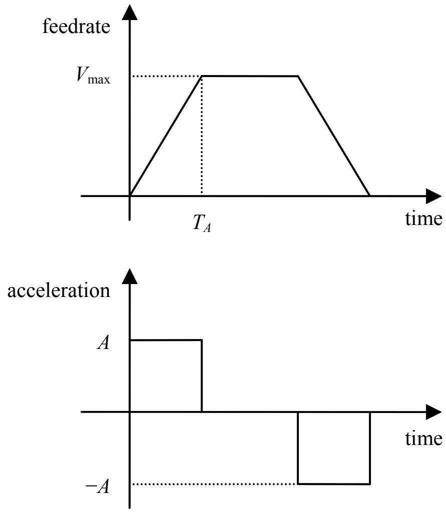

Figure 2. Trapezoidal feedrate profile.
Trapezoidal feedrate profile consists of three phases: the acceleration phase, the constant speed phase and the deceleration phase. During acceleration and deceleration phases, acceleration is constant and its value is respectively A and $-A$. T_A is the duration of acceleration phase and is calculated as $T_A = V_{max} / A$.

Acceleration/deceleration control cannot be implemented in variable feedrate NURBS interpolators presented in [7][8] because they calculate feedrate during the interpolation process, according to geometric properties of interpolation point, while ADCBI requires that the feedrate is known before interpolation. Various interpolators that implement acceleration/deceleration control while respecting geometry constraints have been developed [9][10][11]. The approach used by the proposed algorithm consists in subdividing the NURBS curve in segments whose points have similar geometric properties. For each segment the maximum feedrate is determined before interpolation utilizing ADCBI and look-ahead techniques.

III. Proposed Interpolator

The proposed NURBS interpolator allows to respect both chord error tolerance and centripetal acceleration limit, while implementing acceleration/deceleration control to respect tangential acceleration limit. Proposed software architecture is shown in Fig. 3. The algorithm is divided in two-stages: the segmentation module performs the first stage of the algorithm, while the interpolation module performs the second stage. The two stages exchange information using a buffer and may be executed simultaneously in order to reduce execution time and buffer size requirements.

A. Segmentation module

Segmentation module goes through the NURBS to break it down into curve segments according to curvature values that are encountered. Consecutive points whose curvature values are within the same range are included into the same segment. In this way, high curvature zones of the curve are placed in different segments than low curvature zones. Each segment is inserted in a segment buffer that stores the following information:

- length of the segment;
- value of parameter u corresponding to the end of the segment;
- maximum feedrate, calculated according to (17) assuming the maximum curvature value of the segment (i.e. the upper bound of the corresponding curvature range); using this feedrate, in every point of the segment, both chord error tolerance and centripetal acceleration limit are respected;
- final feedrate of the segment, calculated by look-ahead algorithm analyzing following segments.

Maximum feedrate, length and final feedrate are used by ADCBI to generate feedrate profile.

Curvature ranges can be calculated in different ways. The proposed algorithm uses the following method. The first range includes all values of curvature such that maximum feedrate calculated using (17) is greater than the commanded feedrate specified by program code. In this way, all segments in first range can be machined at commanded feedrate without exceeding chord error tolerance or centripetal acceleration limit. This range contains curvature values from 0 (straight path that can be machined without feedrate limits) to K_1, that is

$$K_1 = \min\left(\frac{a_c}{v_{comm}^2}, \frac{8ER_{max}}{T^2 v_{comm}^2 + 4ER_{max}}\right), \tag{18}$$

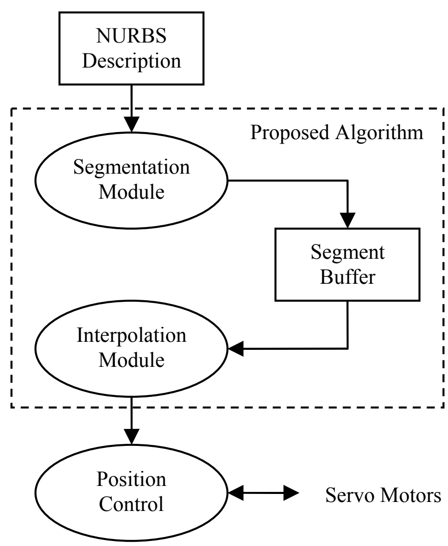

Figure 3. Proposed software architecture.
NURBS description contains control points, weights, knot vector and degree of the curve, besides it contains the commanded feedrate for the curve. Segmentation module analyzes the curve and populates the segment buffer. Interpolator takes segments from the buffer and produces reference values for position control.

where v_{comm} is the commanded feedrate. Equation (18) is obtained from (12) and (15). Upper bounds of following ranges are calculated doubling the upper bound of the preceding range, so the i^{th} range bounds are K_{i-1} (lower bound) and K_i (upper bound), where K_i is

$$K_i = \begin{cases} 0 & \text{if } i = 0 \\ \min\left(\dfrac{a_c}{v_{comm}^2}, \dfrac{8ER_{\max}}{T^2 v_{comm}^2 + 4ER_{\max}}\right) 2^{i-1} & \text{if } i > 0 \end{cases}, \quad (19)$$

Flow-chart in Fig. 4 describes the algorithm used by segmentation module to scan the whole curve and divide it into segments.

Interpolator proposed in [11] also utilizes a segmentation task for constructing feedrate profile. Its segmentation algorithm, however is different from the one presented in this paper and curve is divided where curvature function has local maxima, so segments start with a local maximum that determines initial feedrate, contain a local minimum that determines maximum feedrate and end with another local maximum that determines final feedrate. As usual, equation (17) is used to determine feedrate associated with a curvature value. Feedrate profiles generated this way, however, do not ensure that $v_{\max}$ is respected for all points inside segment.

B. *Interpolation module*

The interpolation module is a speed controlled interpolator similar to the one presented in [1]. However, to decrease feedrate fluctuations caused by truncation error, second order truncated Taylor series expansion [13] is adopted and the following formula is used instead of (10):

$$u[k+1] = u[k] + \frac{v}{\sigma}T + \frac{1}{\sigma}\left(a_t - \frac{x'x'' + y'y'' + z'z''}{\sigma^3}v^2\right)\frac{T^2}{2}. \quad (20)$$

So, when the interpolator generates a new interpolated point, it must consider desired feedrate, desired tangential acceleration a_t and the first and second order derivatives of the curve.

C. *Synchronization between the two modules*

The interpolation module has strict real-time requirements, so it must be implemented with a cyclic task with period T, while the segmentation module can be implemented as a non-cyclic task with no deadlines. It is not necessary that interpolation module waits until the segmentation module finishes the curve processing, so the two modules can run simultaneously using a producer-consumer paradigm in order to reduce buffer size and total execution time. Look-ahead algorithm generates the feedrate profile so that the machine tool stops at the end of the last segment in buffer. This ensures that if segmentation module cannot provide new segments as fast as interpolation module consumes them, the machine stops without exceeding acceleration limits. However such behavior is not desired, so the system must be dimensioned in such a way that buffer never empties until the end of machining.

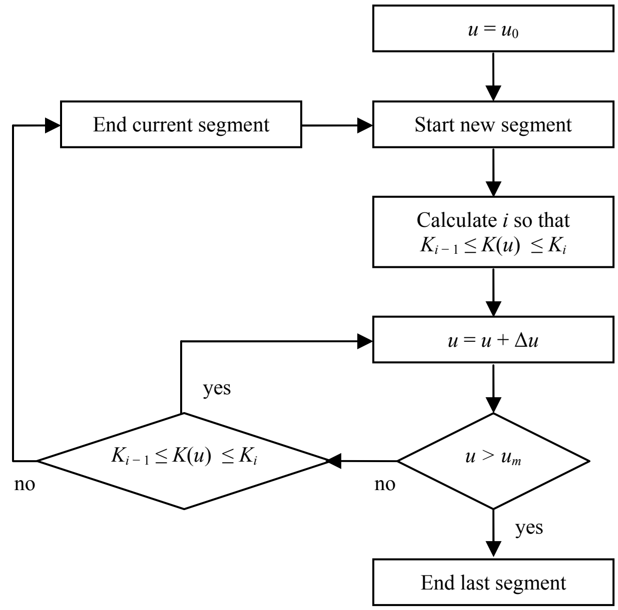

Figure 4. Segmentation module behaviour
This is the method used by segmentation module to break down the curve into segments. The whole curve is analized incrementing each time the parameter u by a Δu step. Δu value is not fixed but depends inversely on $K(u)$ in order to analyze in-depth high curvature zones that are the most throublesome and not wasting computation time on almost straight zones.

IV. SIMULATION RESULTS

In this section the testing environment is presented, then simulations results are commented and performances of the proposed algorithm are compared to two classical interpolators.

A. *Test parameters*

The NURBS curve shown in Fig. 5 has been chosen for testing because its curvature varies considerably along the curve, so benefits of variable feedrate are emphasized. Curve parameters are:

- degree $p = 2$;
- control points ($\mathbf{P}_i$): $\begin{bmatrix}15\\15\\15\end{bmatrix}, \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\30\\30\end{bmatrix}, \begin{bmatrix}15\\15\\15\end{bmatrix}, \begin{bmatrix}30\\0\\0\end{bmatrix}, \begin{bmatrix}30\\30\\30\end{bmatrix}, \begin{bmatrix}15\\15\\15\end{bmatrix}$ (mm);
- weights (w_i): {1, 100, 100, 1, 100, 100, 1};
- knots (u_i): {0, 0, 0, 0.25, 0.5, 0.5, 0.75, 1, 1, 1}.

The commanded feedrate for test curve is 200 mm/s.

Interpolator parameters utilized are listed in Table I. Acceleration limits must be chosen depending on the servo motors characteristics and load, interpolator period is chosen depending on hardware computational power and chord error is chosen depending on desired machining precision. Values in Table I represent a typical scenario.

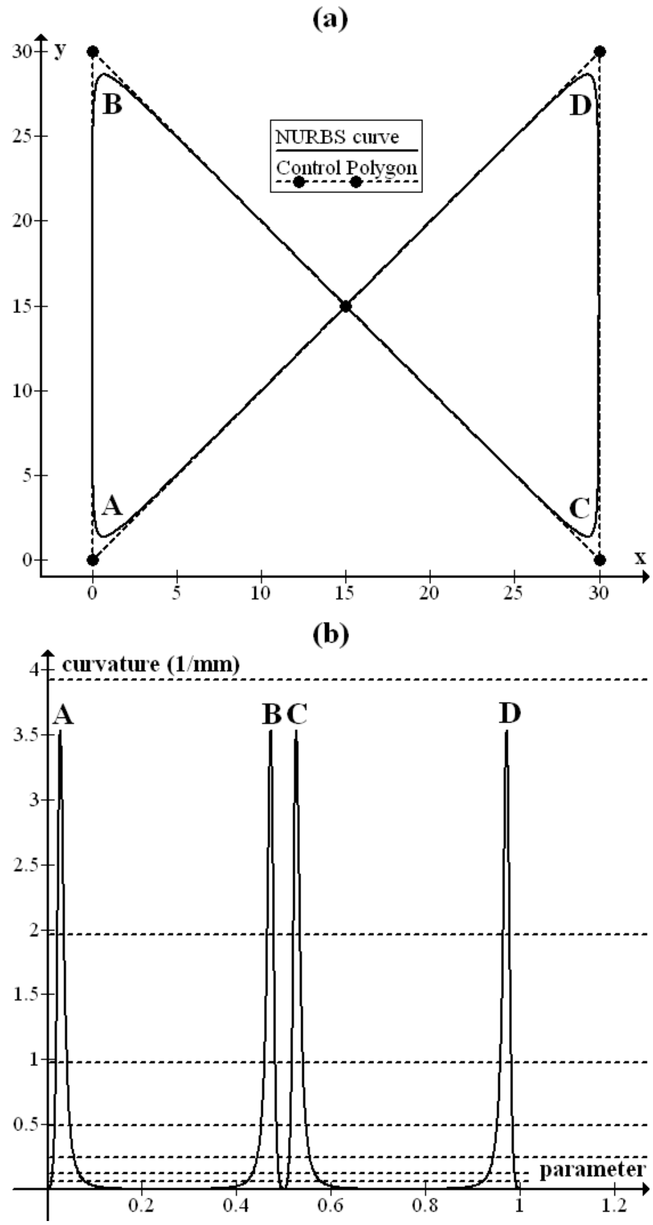

Figure 5. The test NURBS curve.
(a) NURBS curve shape: solid line represents the curve, while dashed line is the control polygon (i. e. the polyline that joins all control points)
(b) Curvature: solid line represents the curvature function, while dashed lines are the bounds of curvature ranges when commanded feedrate is 200 mm/s
A, B, C and D mark high curvature zones.

TABLE I. PARAMETERS OF INTERPOLATOR

Parameter	Symbol	Value
Maximum centripetal acceleration	$a_{c\max}$	2450 mm/s^2
Maximum tangential acceleration	$a_{t\max}$	2450 mm/s^2
Chord error tolerance	$ER_{\max}$	1 µm
Interpolator period	T	0.5 ms

B. Performance comparison with speed-controlled and adaptive feedrate interpolators.

Performances of the proposed interpolator are compared with a speed-controlled interpolator and a variable feedrate interpolator. The speed-controlled interpolator is based on the one presented in [1] and follows a trapezoidal feedrate profile for the whole NURBS. The variable feedrate interpolator is an improvement of the speed-controlled one and uses (14) to reduce feedrate in order to respect chord error tolerance. In order to do a fair comparison, all the algorithms use second order Taylor series truncations even though they were originally proposed using the first order truncation. Besides, the used segmentation scheme is compared to the segmentation scheme presented in [11]. The interpolator using that scheme will be referred as "max-segmented".

Fig. 6 shows the actual feedrate of the interpolators. Speed-controlled interpolator suffers from feedrate fluctuations caused by Taylor series truncation error that can be noticed at high curvature zones A, B, C and D (Fig. 6-a). Lower feedrate causes a reduction of truncation error, so these fluctuations are not present in the proposed algorithm. Machining time of the proposed algorithm is greater than the others because of deceleration at zones A, B, C and D (Fig. 6-d). The max-segmented interpolator also decelerates on sensitive zones while maintaining a machine time smaller than the proposed algorithm (Fig. 6-c). However, it cannot respect centripetal acceleration constraints, as shown in Fig. 7 where max-segmented algorithm (Fig. 7-a) exceeds the maximum feedrate that allow centripetal acceleration to be within the limits, while proposed algorithm (Fig. 7-b) always respects it.

Fig. 8 shows the chord error results. Variable feedrate, max-segmented and proposed algorithm respect the given tolerance, while speed controller exceeds the tolerance value at zones A, B, C and D (Fig. 8-a). In this case, since centripetal acceleration constraints require a lower feedrate than chord error constraints, the proposed algorithm performs much better than variable feedrate interpolator.

Fig. 9 shows tangential acceleration results. Proposed interpolator and max-segmented interpolator are able to maintain tangential acceleration within the maximum specified value. Since variable feedrate algorithm changes feedrate abruptly, its performance is far worse than the others.

Table II summarizes simulation results.

V. CONCLUSIONS

In this work, a two stage NURBS interpolator capable of confining chord error, centripetal acceleration and tangential acceleration is presented. The algorithm builds a feedrate profile that allows high machining speed in low-curvature zones while reducing speed when curvature increases, in order to respect chord error and centripetal acceleration constraints. The transition between different feedrate values is not abrupt, since the feedrate profile is tangential acceleration limited.

Simulation results have demonstrated that constraints are fulfilled and there are not noticeable feedrate fluctuations because sensitive zones are machined at low feedrate, allowing more precise interpolation.

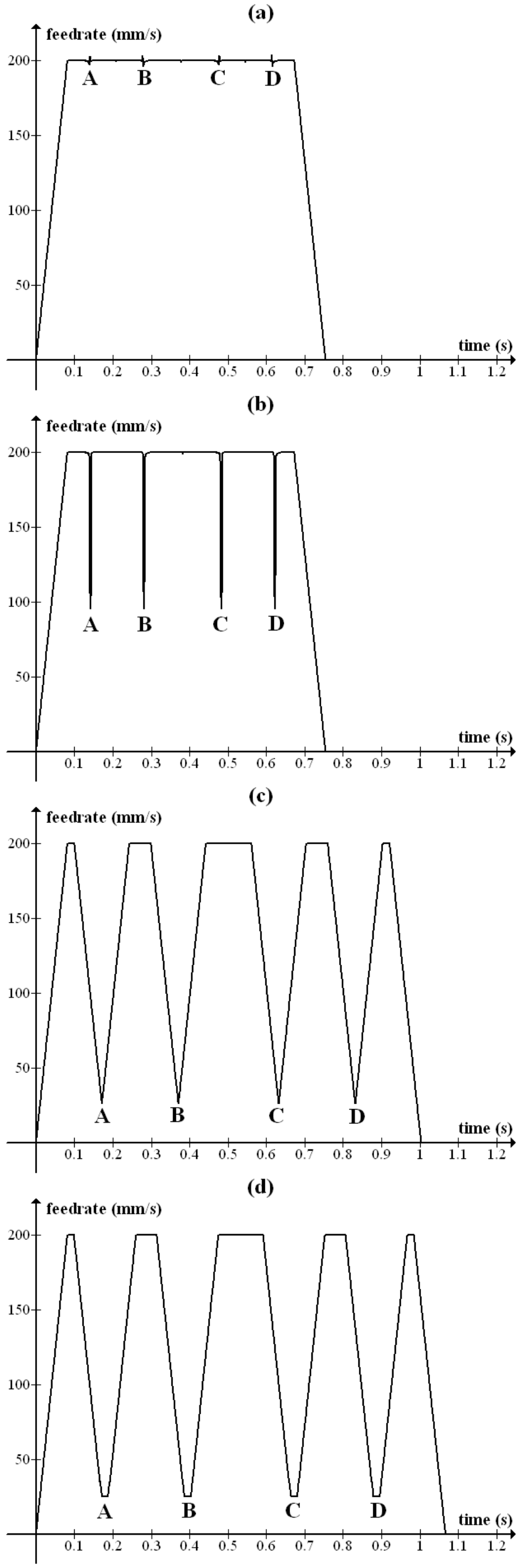

Figure 6. Feedrate results.
(a)Speed-controlled interpolator (b) Variable feerate interpolator
(c) Max-segmented (d) Proposed interpolator
A, B, C and D mark high curvature zones.

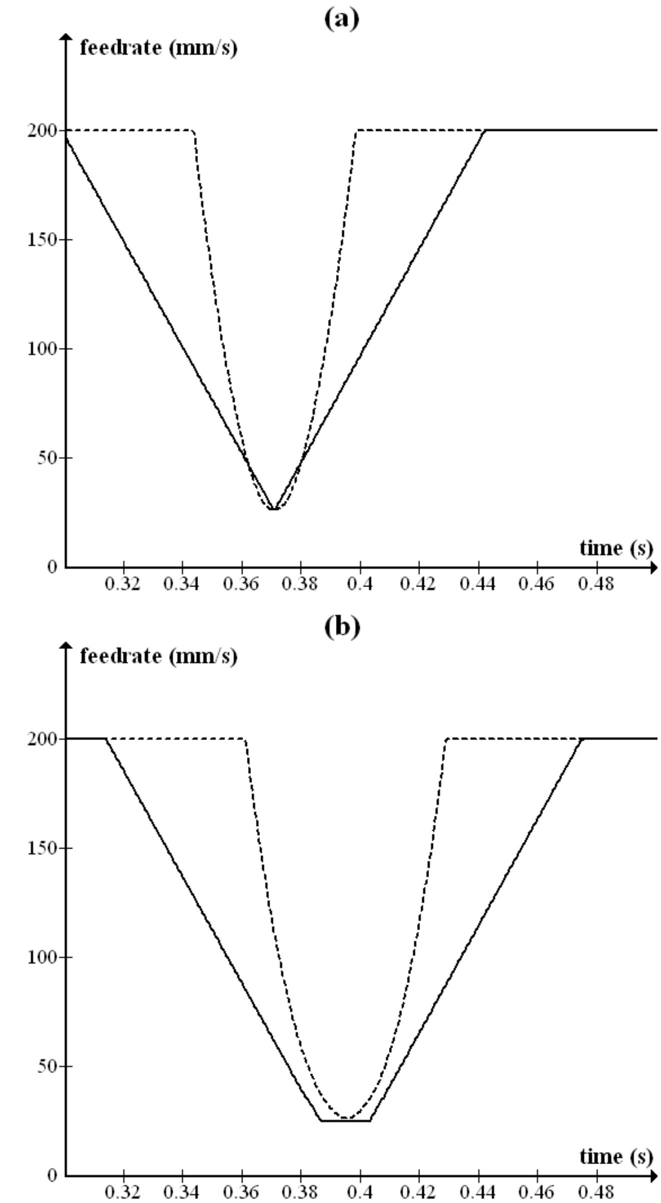

Figure 7. Detailed view of feedrate results of the two segmentation schemes.
(a) Max-segmented (b) Proposed interpolator
Feedrate results are shown between 0.3 s and 0.5 s.
Solid line represents feedrate results and dashed line represents maximum feedrate that allows to respect centripetal acceleration limit.
The proposed segmentation scheme always respects the maximum feedrate unlike the max-segmented scheme.

TABLE II. SIMULATIONS RESULTS FOR DIFFERENT ALGORITHMS

Interpolation algorithm	**Chord error (µm)**		**Maximum tangential acceleration (mm/s^2)**	**Machining time (s)**
	MAX	***RMS***		
Speed-controlled	4.46	0.42	9,313.8	0.755
Variable feedrate	1	0.21	72,037	0.755
Max-segmented	0.11	0.03	2,455.5	1.004
Proposed	0.07	0.02	2,450.3	1.0675

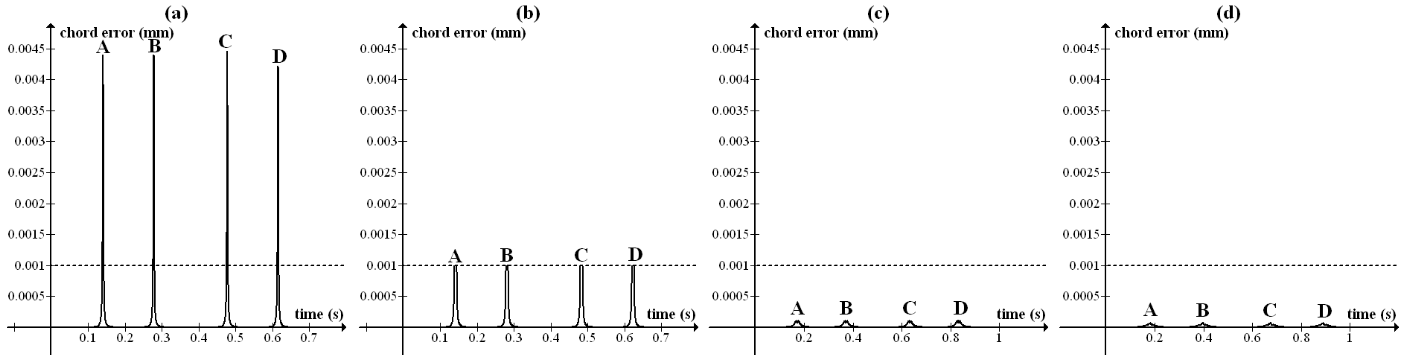

Figure 8. Chord error results.
(a)Speed-controlled interpolator (b) Variable feerate interpolator (c) Max-segmented (d) Proposed interpolator
Dashed line represents chord error tolerance.
A, B, C and D mark high curvature zones.

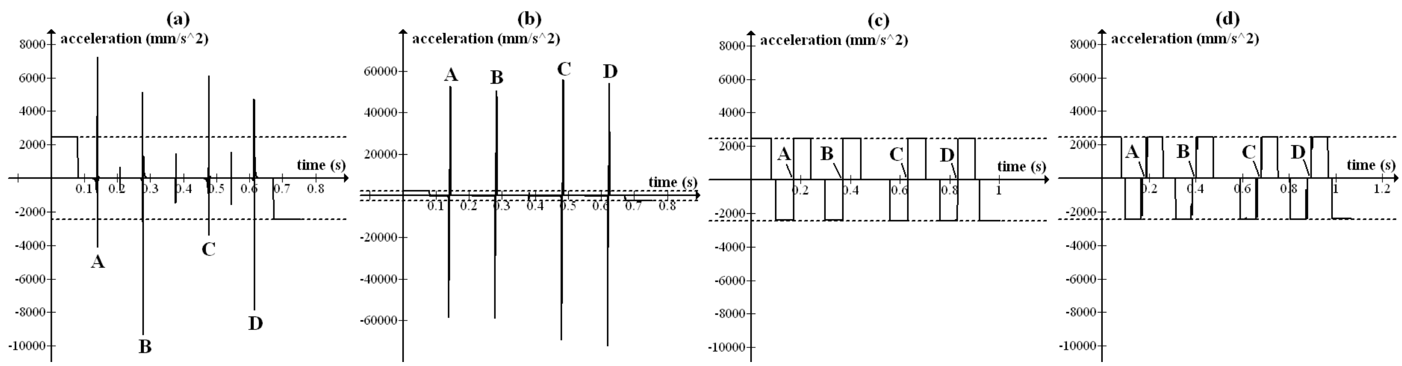

Figure 9. Tangential acceleration results.
(a)Speed-controlled interpolator (b) Variable feerate interpolator (c) Max-segmented (d) Proposed interpolator
Dashed line represents tangential acceleration limit.
Note that the variable feedrate interpolator graph has a different scale from other graphs.
A, B, C and D mark high curvature zones

ACKNOWLEDGMENT

This work is supported by ISAC srl, Italy as a part of the project "NURBS interpolation for ISAC Numerical Control". In particular, the authors would like to thank Andrea De Nardis and Paolo De Nardis for their support.

REFERENCES

[1] Q. G. Zhang and R. B. Greeenway, "Development and implementation of a NURBS curve motion interpolator" in Robotics and Computer-Integrated Manufacturing, vol. 14, pp. 27–36, 1998.

[2] L. Piegl, "On NURBS: a Survey" in IEEE Computer Graphics and Applications, vol. 11, pp. 55–71, 1991.

[3] L. Piegl and W. Tiller, The Nurbs Book, 2nd ed. Springer, 1997.

[4] S. Bedi, I. Ali, and N. Quan, "Advanced Interpolation Techniques for N.C. Machines" in ASME Journal of Engineering for Industry, vol. 115, no. 3, pp. 329–336, 1993.

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

[6] C. W. Cheng, M. C. Tsai and J. Maciejowski, "Accurate Feedrate Control of CNC Machine Tools along NURBS Curves" in Proceedings of the 43rd IEEE Conference on Decision and Control, pp. 3053–3058, 2004.

[7] S. S. Yeh and P. L. Hsu, "Adaptive feedrate interpolation for parametric curves with a confined chord error" in Computer-Aided Design, vol. 34, pp. 229–237, 2002.

[8] X. Zhiming, C. Jincheng and F. Zhengjin, "Performance evaluation of a real-time interpolation algorithm for NURBS curves" in The International Journal of Advanced Manufacturing Technology, vol. 20, pp. 270–276, 2002.

[9] T. Yong and R. Narayanaswami, "A parametric interpolator with confined chord errors, acceleration and deceleration for NC machining" in Computer-Aided Design, vol. 35, pp. 1249–1259, 2003.

[10] D. Du, Y. Liu, C. Yan and C. Li, "An accurate adaptive parametric curve interpolator for NURBS curve interpolation" in The International Journal of Advanced Manufacturing Technology, vol. 32, pp. 999–1008, 2007.

[11] M. T. Lin, M. S. Tsai and H. T. Yau, "Development of Real-time Look-Ahead Algorithm for NURBS Interpolator with Consideration of Servo Dynamics" in Proceedings of the 46th IEEE Conference on Decision and Control, pp. 1862–1867, 2007.

[12] S. H. Suh, S. K. Kang, D. H. Chung and I. Stroud, Theory and Design of CNC Systems, Springer, 2008.

[13] R. T. Farouki and Y. F. Tsai, "Exact Taylor series coefficients for variable-feed rate CNC curve interpolators" in Computer-Aided Design, vol. 33, pp. 155–165, 2001.