Performance of off-line polynomial CNC trajectories within the context of HSM

Sylvain Lavernhe, Xinbin Duan, Christophe Tournier, Claire Lartigue

Proceedings of IMECE'04
2004 ASME Design Engineering Technical Conferences
Anaheim, California, USA, November 13-19, 2004

IMECE2004-60900

PERFORMANCE OF OFF-LINE POLYNOMIAL CNC TRAJECTORIES WITHIN THE CONTEXT OF HSM

Sylvain Lavernhe[1], Xinbin Duan[1], Christophe Tournier[1], Claire Lartigue[1,2]

[1] Laboratoire Universitaire de Recherche en Production Automatisée
ENS de Cachan - Université Paris Sud 11
61 Avenue du Président Wilson, 94235 Cachan Cedex - France

[2] IUT de Cachan - Université Paris Sud 11
9 Avenue de la division Leclerc, 94234 Cachan Cedex – France

name@lurpa.ens-cachan.fr

ABSTRACT

The objective of the article is to present various off-line calculation methods to calculate polynomial tool trajectories, format well adapted to High-Speed Machining. In particular, we are interested in comparing machining performances of various polynomial calculation algorithms such as interpolation, association or inter-approximation with energy minimization. This comparison is achieved using a test part through simulations and machining tests to justify the efficiency of the calculation methods. Measurements of velocity and position during machining highlight differences between the different association methods. Attention is also paid to the visual and geometrical quality of the machined surfaces.

NOMENCLATURE

$\mathbf{A}$	Interpolation matrix
$\mathbf{C}(t)$	B-spline curve
$E^1(\mathbf{X})$	Least-squares error
$\mathbf{K}$	Stiffness matrix
P_i	Control points of the B-spline curve
Q_i	Interpolation points
$\mathbf{v}$	Lagrange multipliers
$\mathbf{X}$	Solution vector storing control points

INTRODUCTION

In order to take advantage of High Speed Machining (HSM), the description of machining tool paths has evolved from the linear format to the polynomial one. As the polynomial format reduces the amount of data to be processed by the Numerical Controller (NC), the programmed feedrate could be reached.

First works focused on real-time polynomial interpolation. The initial calculated tool trajectory is transmitted to the NC unit which performs in real-time the interpolation of the tool path as a Nurbs curve. To preserve continuity of the tool path while maintaining the programmed feedrate, authors suggested different parameterization techniques to generate real-time trajectories with smooth velocity (Altintas and Erkomaz 2003), (Cheng et al. 2002), (Fleisig and Spence 2001).

Henceforth, CAM systems may generate tool trajectories directly calculated in a native polynomial format (Langeron et al. 2004), (Lartigue et al. 2001). Authors thus integrate native polynomial curves in the process of tool trajectory NC treatment involving simplification in the NC tasks. Due to the coherence between models, the machining code interpretation task is removed. Furthermore, tasks of converting the data into control commands for each sampling period and of executing servo control commands on each axis are simplified.

Nevertheless, usual CAM system methods for tool path calculations rely on the linear format. When the trajectory is expressed in the linear format, post-processors may convert this format into a polynomial one. The polynomial trajectory is thus called *off-line polynomial trajectory* in opposition to on-line polynomial trajectory that corresponds to polynomial trajectory calculated in real-time by the NC unit.

In this paper, we focus on off-line calculation methods of tool trajectories. In particular, we are interested in comparing machining performances for various calculation algorithms of polynomial tool trajectories. Indeed, numerous association methods are available to fit a curve to a set of points. Differences are mainly due to the nature of the problem (interpolation or approximation), and the parameterization choice (Park et al. 2000), (Piegl and Tiller 1997), (Pourazady and Xu 2000), (Vassilev 1996). These methods are more dedicated to surface modeling, and are seldom used for tool path calculation that basically relies on an interpolation problem. Furthermore, the parameterization is imposed by the NC unit, and generally corresponds to chord length. We are interested in whether the association method influences or not the NC unit behavior during machining.

For this purpose, we developed a post-processor which can fit various cubic B-Spline curves to sets of points initially generated for linear interpolation. The cubic B-spline format is imposed by the NC unit used. Given that the polynomial format is C^2 continuous, it is not possible to fit exactly the CAD surface near discontinuities. The first step of the post-processor thus consists of an original geometrical analysis of the discrete data to find out break points limiting portions. This provides us with a partition of points in portions that can easily be fitted by cubic B-spline curves. The following association methods are implemented: interpolation, approximation and inter-approximation with energy minimization. The junction of the portions is at least C^0 continuous. At final, the tool trajectory is composed of several curves that fit with precision the chosen part surface and also allows for a high feedrate machining due to its polynomial format.

To assess the performance of each association method, a test surface geometry has been designed. Trajectories are first compared through a geometrical analysis, and are then transmitted to the CNC of the HS milling center. A kinematical analysis of the trajectory follow-up by the NC unit is carried out during the high-speed machining of the part.

The paper is organized as follows. The first section details the steps of the post-processor allowing the calculation of polynomial tool trajectories. The second section focuses on experiments. Following, the third section is dedicated to the analysis of the performance of off-line polynomial trajectories. The paper is ended by some concluding remarks.

OFF-LINE CALCULATION METHODS

This section presents the different steps of the post-processor carried out to calculate tool paths as cubic B-spline curves from an initial trajectory (Fig.1).

Indeed, starting from the CAD model of the part, the trajectory is calculated following usual methods by the CAD/CAM system. The approach, considering offset surface-driving plane intersection methods (Lartigue et al. 2001), is as follows:

- Selection of the surface to be machined
- Definition of the machining strategy (machining direction, CAM parameters)
- Calculation of Cutter Location points (CL points)
- Identification of break points
- Calculation of tool-paths using cubic B-splines

When the linear interpolation is used to generate the iso code (G1 code), as they are the input data of the post processor, CL points must first be extracted from the iso code.

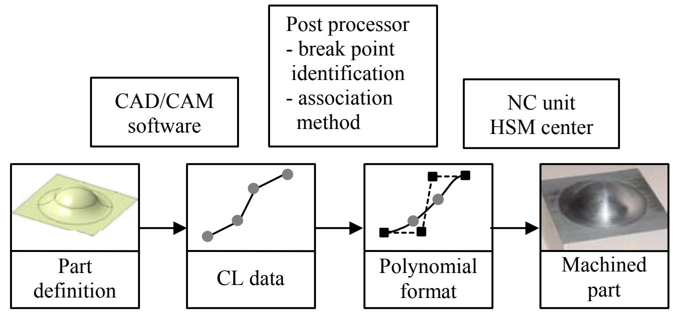

Figure 1. Off line programming scheme

The break point identification step is necessary for two main reasons, linked each other. Firstly we have to preserve the smoothness of the surface which is compound of free-forms presenting geometrical discontinuities. And secondly, if we directly fit a curve to the data, due to C^1 or C^2 discontinuities of the profile, the resulting tool path could consist of small B-spline curve segments, which is not appropriate for High-Speed Machining (HSM). Therefore, we use a partition method relying on the extraction of break points limiting C^2 continuous portions from the set of CL data points.

Resulting from the previous step, the whole set of CL points (representing the tool trajectory) is divided into point subsets, images of C^2 continuous portions which can easily be fitted by cubic B-spline curves. In literature, many association methods are available to fit a curve to a set of points which are more appropriated to surface modeling (Park et al. 2000), (Piegl and Tiller 1997), (Pourazady and Xu 2000), (Vassilev 1996). As the study's objective is to bring out whether the association method influences or not the NC unit behavior during machining, only a few methods have been yet implemented in the post-processor: interpolation, approximation, and inter-approximation with energy minimization.

In next sections, the main stages of the post-processor are detailed.

Break point identification

As information about the design surface is under discrete form (points), the surface topology is unknown. The break point identification method consists of two main stages: data coding and data segmentation (Fig.2).

The coding stage generates a *pseudo-differential* geometrical information: nature of the portion, linear or curvilinear, changes in concavity, changes in tangency, …. It relies on the polygonal approximation of the set of CL points. For each tool-path of the trajectory, the coding algorithm analyses the geometrical configuration of successive CL points (Osty 2001). A code "0", "+", or "-" is affected to each segment built from two successive CL points depending on its length and its orientation as regards the previous one. For instance, for segments the length of which is greater than a given threshold, the corresponding code is "0" (4^{th} portion in figure 2). Consider one segment which turns in the clockwise direction relatively to the previous segment, the corresponding code is "-" (1st portion in figure 2); if turning in the counter clockwise direction, the code is "+" (2nd portion in figure 2). Therefore, the code is an image of the nature of the segment: "0" for a linear portion, "+"or "-" for a curve portion; and its geometrical configuration: "+" for a positive concavity or "-" for a negative one. At this stage, one tool trajectory is a set of CL points to which corresponds a sequence of codes.

Sequence of codes = « **- - - + + + + - + + 0** »

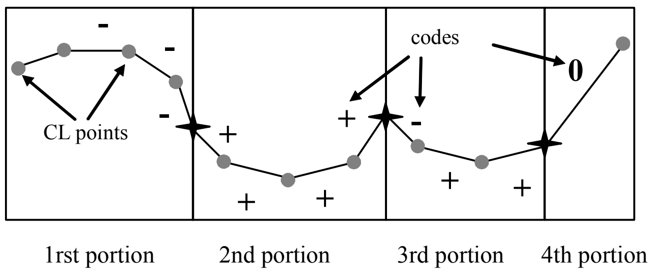

Figure 2. Break point identification

The next stage is data segmentation to identify break points which is performed through the analysis of the sequence of codes. Break points represent geometrical singularities of the shapes of the part such as C^1 or C^2 discontinuities between portions. In the coding sequence, break points correspond to changes in the code value. In figure 2, three break points can be identified (star symbols) defining the four portions. The last portion the code of which is "0" is a linear portion; others are curvilinear. Hence, identified break points define the beginning and/or the end of a portion of the tool path which is considered as C^2 continuous. Table 1 details for each break point the nature of the corresponding discontinuity.

Break point		**Nature of the singularity**
...0	0...	intersection of two straight lines
...0	+---... -+++...	intersection straight line – curve *(tangency discontinuity)*
...0	---... +++...	intersection straight line - curve *(curvature discontinuity)*
...---	+++...	curvature discontinuity between two curves
...+++ ...---	-+++... +---...	tangency discontinuity between two curves

Table 1. Break point and geometrical discontinuity

The method here detailed is simple, efficient and well adapted to segmentation of tool trajectories into C^2 continuous portions. At final, at each C^2 continuous portion of the tool path, a cubic B-spline curve is fitted according to various association methods.

Methods of association

As previously exposed, among the numerous association methods available in literature, only the following methods have been implemented in the post-processor:

- interpolation
- approximation
- inter-approximation with energy minimization

Interpolation is the most popular method used for tool path generation for it is simple. However, it requires a very precise identification of break points (Langeron et al. 2004), (Lartigue et al. 2001), (Valette et al. 2000).

Polynomial approximation is seldom used in tool path generation. Nevertheless, it deserves to be studied, considering that it should decrease the number of B-spline curves while being less sensible to a precise identification of break points (Valette et al. 2000). As small wavelets may appear near junction points, the last method is implemented to preserve the smoothness of the tool path.

Note that, we assume that the reader is familiar with the concepts of B-spline curves. For all cases, polynomial curves are cubic B-spline curves defined according to Eq. (1):

$$\mathbf{C}(t) = \sum_{i=0}^{n} \mathrm{N}_{i3}(t)\,\mathbf{P}_i \qquad t \in [t_0, t_{n+4}] \qquad (1)$$

$\mathbf{C}(t)$ is built from a non uniform knot vector t_i, a set of (n+1) control points $\mathbf{P}_i$, N_{i3} are the blending functions commonly used (for which the degree is equal to 3) (Piegl and Tiller 1997).

Interpolation

From the initial set of CL points, only a series of tool position points is interpolated. Let $\mathbf{Q}_i$ be the interpolation points and μ_i, the associated parameters (i= 0, …., m).

The problem consists of the calculation of the curve $\mathbf{C}$, defined by the control points $\mathbf{P}_j$ and the knot values t_i, that verifies:

$$\mathbf{C}(\mu_i) = \sum_{j=p-3}^{p} \mathrm{N}_{j3}(\mu_i)\,\mathbf{P}_j = \mathbf{Q}_i\,,\ t_p \le \mu_i \le t_{p+1}\,,\ i = 0,.,m \quad (2)$$

The problem is a simple interpolation scheme which is linear and easy to solve. The resolution is carried out in five steps (Langeron et al. 2004):

1) Choice of the interpolation points from the initial tool trajectory portion. Indeed, to simplify calculations, the whole set of CL points is not used as interpolation points. Interpolation points are equally selected from the initial data, including the first and the last point of the C^2 continuous portion.
2) Parameterization calculation. To be coherent with the NC unit used, the cumulative chord length has been considered:

$$\mu_0 = 0\,;\ \mu_m = L = \sum_{i=2}^{m} \left|\mathbf{Q}_i - \mathbf{Q}_{i-1}\right|$$

$$\mu_i = \mu_{i-1} + \left|\mathbf{Q}_i - \mathbf{Q}_{i-1}\right| \quad i = 2,\ldots, m-1$$

3) Calculation of the knot vector, considering a non-uniform B-spline curve:

$$\begin{cases} nt = n+4\textbf{,}\ \textit{length of the knot vector} \\ t_0 = t_1 = t_2 = t_3 = 0 \\ t_{nt-3} = t_{nt-2} = t_{nt-1} = t_{nt} = L \\ t_i = (\mu_{i-3} + \mu_{i-2} + \mu_{i-1})/3 \quad i = 4\textbf{,......,}\ n \end{cases}$$

4) Expression of the interpolation matrix and resolution:

$$\begin{bmatrix} N_{03}(\mu_0) & \dots & N_{n3}(\mu_0) \\ \vdots & \ddots & \vdots \\ N_{03}(\mu_m) & \dots & N_{n3}(\mu_m) \end{bmatrix} \times \begin{pmatrix} \mathbf{P}_0 \\ \vdots \\ \mathbf{P}_n \end{pmatrix} = \begin{pmatrix} \mathbf{Q}_0 \\ \vdots \\ \mathbf{Q}_m \end{pmatrix} \Rightarrow \mathbf{A.X} = \mathbf{Q}$$

With **A**, a (m+1)(n+1) matrix system,
X, a solution vector storing control points**,**
Q, a vector storing input tool-position points

5) Verification of the respect of the machining tolerance. Indeed, as the curve represents a tool trajectory which initially respects the machining tolerance, this condition must still be respected with the polynomial trajectory. Therefore, when the machining tolerance is not ensured, additional interpolation points must be chosen from the initial set of CL points.

This method has been largely used and tested for tool path calculation as B-spline curves (Langeron et al. 2004), (Lartigue et al. 2001), (Valette et al. 2000).

Approximation

Seldom tested for tool path calculation, the approximation method exposed here is a very general constrained least-squares curve fitting method developed by Park et al. (2000). Consider that steps 1 to 4 are similar to that exposed for the interpolation problem, and let $E^{l}(\mathbf{X})$, be the least-squares error, defined by Eq. (3):

$$E^{l}(\mathbf{X}) = \sum_{i=0}^{m} \left\| \mathbf{P}_i - \mathbf{C}(\mu_i) \right\|^2 \qquad (3)$$

If no constraints are given, the approximation problem simply consists in minimizing the least-squares error, which yields to (in a matrix form):

$$\mathbf{min}\left\{E^{l}(\mathbf{X}) = (\mathbf{A.X}-\mathbf{Q})^T(\mathbf{A.X}-\mathbf{Q})\right\} \Rightarrow \mathbf{A}^{\mathrm{T}}\mathbf{A.X} = \mathbf{A}^{\mathrm{T}}\mathbf{Q} \qquad (4)$$

When constraints such as positions, tangents are given as linear equations: $\mathbf{C}_k\mathbf{X} = \mathbf{D}_k$ (k=0,….., r), a solution vector is determined by solving the previous problem subject to constraints:

$$\begin{cases} \mathbf{min}\left\{E^{l}(\mathbf{X})\right\} \quad \textit{subject to}\ \mathbf{C.X} = \mathbf{D} \\ \Rightarrow \begin{bmatrix} \mathbf{A}^{\mathrm{T}}\mathbf{A.X} & \mathbf{C}^{\mathrm{T}} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \times \begin{bmatrix} \mathbf{X} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \mathbf{A}^{\mathrm{T}}\mathbf{Q} \\ \mathbf{D} \end{bmatrix} \end{cases} \qquad (5)$$

Where **v** is a vector storing lagrange multipliers.

For tool path calculation, each portion must be linked to the following one respecting that the end point of the portion is the starting point of the following one. Therefore, constraints of position are always given.

Tangency continuity between portions should ensure velocity continuity at the junction points. Few tests have also been carried out considering tangency continuity at the junction points, but results are not discussed here for they are not significant.

Inter-approximation with energy minimization

Inter-approximation corresponds to the problem defined in Eq. (5), for which the constraints are constraints of positions for the first and the last point of each portion. In some cases, imposing constraint of positions may conduct to a tool path presenting some waves near the junction points (Valette et al. 2000). To solve this problem, we adopt the *inter-approximation with energy minimization* method suggested by Park et al. (2000). Authors introduce the quadratic functional $E^{l+e}(\mathbf{X})$ as a sum of the energy functional of the curve **C***(t)* as defined in (Pourazady and Xu 2000) and the least-squares error term $E^{l}(\mathbf{X})$ defined by Eq. (3):

$$E^{l+e}(\mathrm{X}) = \int_{t} (\alpha\left\|\dot{\mathbf{C}}(t)\right\|^2 + \beta\left\|\ddot{\mathbf{C}}(t)\right\|^2)\,dt + \gamma\sum_{i=0}^{m}\left\|\mathbf{P}_i - \mathbf{C}(\mu_i)\right\|^2 \qquad (6)$$

Where α, β and γ are called stretching, bending and fitting coefficient, respectively.

If **K** denotes the stiffness matrix of the curve **C***(t)*, **K** is the weighted sum of its stretching and bending terms:

$$\mathbf{K} = \alpha\int_{t} \dot{\mathbf{N}}\dot{\mathbf{N}}^T\,dt + \beta\int_{t} \ddot{\mathbf{N}}\ddot{\mathbf{N}}^T\,dt \qquad (7)$$

Coefficients of the **K** matrix are determined using a numerical method such as the Gaussian quadrature.

Therefore, the general minimization problem yields to:

$$\begin{cases} \mathbf{min}\left\{E^{l+e}(\mathbf{X})\right\} \quad \textit{subject to}\ \mathbf{C.X} = \mathbf{D} \\ \Rightarrow \begin{bmatrix} \mathbf{K} + \gamma(\mathbf{A}^{\mathrm{T}}\mathbf{A}) & \mathbf{C}^{\mathrm{T}} \\ \mathbf{C} & \mathbf{0} \end{bmatrix} \times \begin{bmatrix} \mathbf{X} \\ \mathbf{v} \end{bmatrix} = \begin{bmatrix} \gamma(\mathbf{A}^{\mathrm{T}}\mathbf{Q}) \\ \mathbf{D} \end{bmatrix} \end{cases} \qquad (8)$$

For more details concerning the expression of the minimization problem, the reader is invited to refer to Park et al. (2000).

All association methods exposed above have been implemented in the post-processor to generate off-line polynomial CNC trajectories. Performance and efficiency of each one for machining of free-form surfaces have been assessed through the machining of a test part.

EXPERIMENTS

Test part

To perform algorithm comparisons, a test part (Fig.3) is designed which is made up of a portion of sphere (R = 18mm) connected to a plane by the intermediary of a fillet (r = 10 mm). Thus, the whole part surface presents curvature discontinuities (C^2 discontinuities) between the three sub-surfaces. The basic dimensions of the part are 50 x 50 mm.

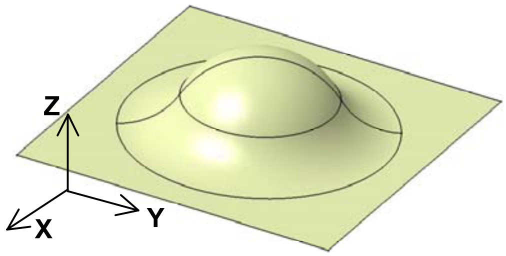

Figure 3. The test part

Generation of the entire tool path

First, the tool path is generated using commercial CAM software in the linear format and considering the following parameters:

- machining is carried out according to YOZ parallel planes;
- chordal deviation (machining tolerance) and scallop height are set up to 10 µm;
- the tool is a ball end cutter tool, the radius of which is equal to 5 mm.

Then, the output APT file is generated by the built-in post processor. As the scallop height is set up to 10 µm, this leads to 87 paths in the transversal direction (from X = -50 mm to X = 50 mm).

From the APT file, CL points are extracted using the post-processor before the calculation of polynomial tool paths. Capabilities of the CAM software to produce polynomial tool paths by interpolation are also used. Hence, polynomial tool trajectories generated using the proposed post-processor can be compared to the one generated by the CAM software. At final, the comparative analysis can be conducted for linear trajectories, polynomial trajectories issued from interpolation (post-processor and CAM software trajectories), from approximation and from inter-approximation with energy minimization.

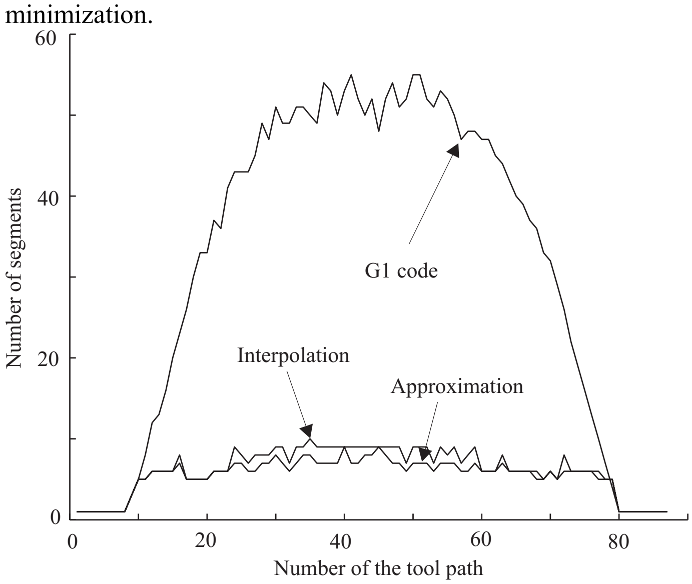

Figure 4. Number of segments for each tool path

The first study concerns the comparison of the number of segments of each tool trajectory defined in the transversal direction from path number 1 to path number 87 (Fig.4). A segment corresponds to an elementary portion of trajectory associated to the format: a line segment for the linear format, and an arc for the polynomial one. The difference between the linear format (G1 code) and the polynomial formats is significant showing that amount of information is decreased when using polynomial formats. However, polynomial interpolation and polynomial approximation present a similar behaviour. The fact that file sizes transmitted to the NC unit are largely reduced hints for segment length to be longer than when using the linear format. The kinematical behaviour should thus be improved.

Figure 5 presents the geometry of the curves describing a given tool path for each association method. The chosen tool path is that passing through the top of the part surface (top of the sphere, X = 0 mm, 44^{th} tool path). As break points are identified at the previous stage, the number of B-spline curves is the same whatever the association method used. Nevertheless, figure 5 shows that interpolation leads to one additional arc point located on the fillet. Although tool trajectories are geometrically similar, their mathematical description varies which can lead to different kinematical behaviors during machining. This phenomenon can be observed for all the transversal tool paths.

This result was foreseeable taking into account the density of points generated with the linear format. Due to the expected quality, the chordal deviation must be confined which does not leave sufficient degrees of freedom to obtain curves with different behaviors.

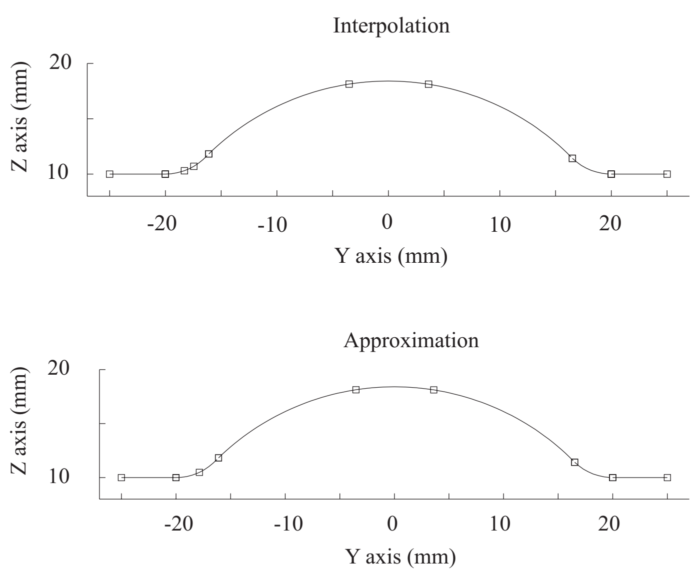

Figure 5. Elementary portions of trajectory for two association methods

PERFORMANCE OF POLYNOMIAL TRAJECTORIES DURING MACHINING

The calculated curve is transmitted to the NC unit in the native polynomial format. The machining of the part is carried out on a machining centre Mikron UCP 710 equipped with an industrial numerical controller. To respect the context of HSM, the programmed feedrate is equal to 6 m/min.

Visual comparison

The various parts obtained using the various trajectories do not show major differences in aspect as it could be expected after the previous geometrical analysis of the trajectories generated by the three algorithms. Indeed, trajectories are extremely close, and machining confirms this similarity (Fig. 6).

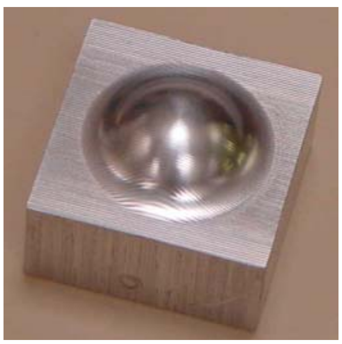

Figure 6. Machined part

Kinematical analysis

During machining, we took readings of position, speed and acceleration in several areas of the part and in particular on the tool path passing through the top of the sphere (Fig. 7). Graphs present the evolution of the feedrate (mm/min) according to the position of the tool for the different association methods.

The tool path generated by the CAM software presents a significant deceleration when the tool is going down after the top of the sphere (0<Y<15) contrary to the tool paths generated with other algorithms. The difference of feedrate can reach 1500 mm/min, almost 30% of the programmed one. From the productivity point of view, this variation is not dramatic here for part dimensions and distances to cover are small enough.

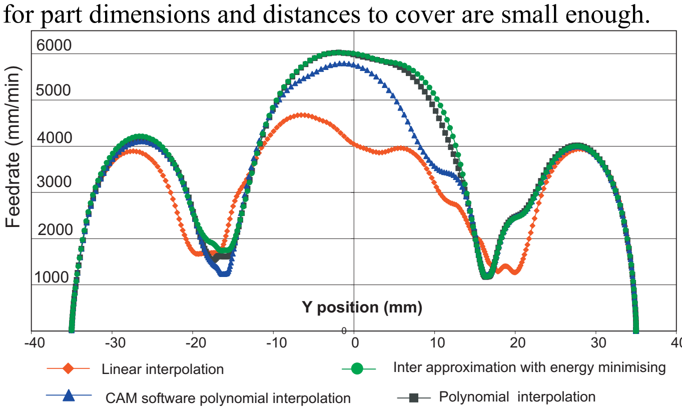

Figure 7. Feedrate evolution for the different methods

In the same way, the cutting pressures for the considered finishing operation are very weak. This variation would have a larger influence if it were a profiling operation in flank milling. Indeed, the tool deviation is proportional to the cutting pressures, therefore to the effective feedrate.

The analysis of the NC files brings out that although the number of B-spline curves is the same for all the methods, on the considered portion, the B-spline generated using the CAM software presents one more pole than the others. This deceleration could have been due to the action of gravity. However, if the tool is driven in the opposite direction, the deceleration is always on the portion of curve with the additional pole. Nevertheless, this type of slowdown also appears during the machining of the fillet (15<Y<25) for the three polynomial trajectories, whereas they exactly have the same number of poles.

In order to understand this phenomenon, we built a cubic B-spline curve, defined with 4, 5 or 6 poles (Fig. 8). The resulting nodal sequences are thus the following ones:

{0 0 0 0 1 1 1 1}
{0 0 0 0 .5 1 1 1 1}
{0 0 0 0 .25 .5 1 1 1 1}

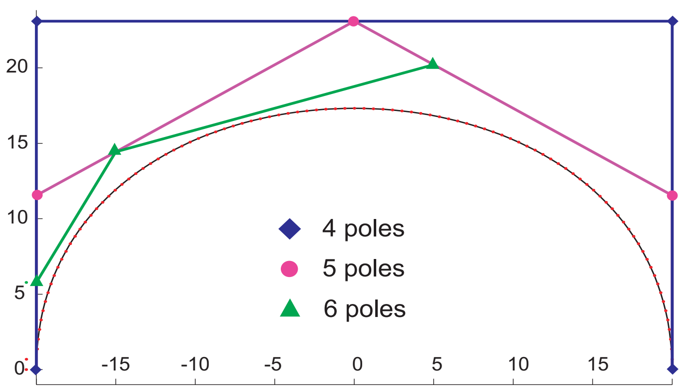

Figure 8. Various control polygons of the test curve

The evolution of the feedrate for this curve is observed in figure 9. Initial speed is not null since the curve is connected in tangency to segments on the right- and left-hand sides. Each stage of deceleration is similar to that previously observed (Fig. 7).

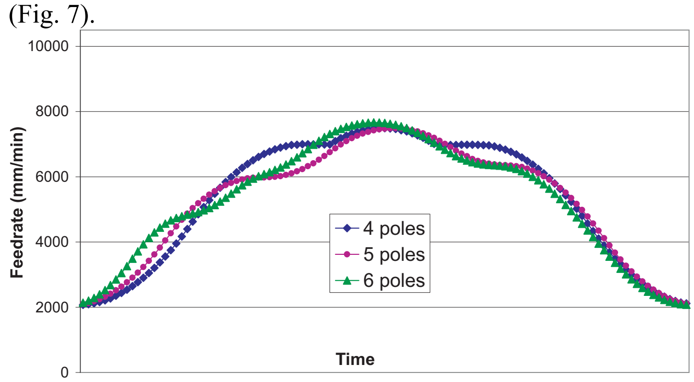

Figure 9. Feedrate evolution for each control polygon

We expected that the NC unit reads as many lines as necessary to describe the first arc of the cubic B-spline (four lines in this case), and reads the following lines during machining to generate next arcs. In this configuration, slowdowns should take place when the NC reads the next line and computes the new arc. Following this analysis, no slowdowns or steps should appear when machining the 4-poles B-spline curve (compound with one arc), only one step (respectively two) should appear for the 5-poles curve (respectively 6-poles curve). But, the analysis of feedrate evolutions in figure 9 leads to deny this behaviour: the number of slowdowns and their positions do not correspond to mathematical arcs. Indeed, it looks like the numerical controller generates as many arcs as there are segments in the polygon built from the poles, i.e. lines in the program. Example for the machining program of the curve with 6 poles:

```
G1 Y-20  Z0       (start point)
BSPLINE SD=3
X0 Y-20  Z5.7735       PL=0    first step
X0 Y-15  Z14.4338      PL=0.25 second step
X0 Y5    Z20.2073      PL=0.25 third step
X0 Y20   Z11.547       PL=0.5  fourth step
X0 Y20   Z0            PL=0    (end point)
```

Slowdowns take place only if the machine is in acceleration or a deceleration stage. Experiments show that decelerations are strongly linked to the value of the NC parameters: maximal jerk and acceleration for each axis. If the programmed feedrate is reached before the following line of the ISO code, the deceleration disappears. When distances to be traversed are so small that the programmed feedrate could not be reached, it is the number of poles of each B-spline curve that conditions possible decelerations. A curve which minimizes the number of poles permits to reach higher feedrate in a shorter time and to maintain a higher federate during machining time.

CONCLUSIONS

We have developed and implemented a post-processor able to transform tool paths described in linear interpolation into cubic B-spline trajectories. Our purpose is to analyze the influence of different methods of curve association (interpolation, approximation, minimization of energy, etc.) on the machined surface quality and on the kinematical behavior of the machine tool. In order to minimize the chordal deviation and to seek a better surface quality, the number of cutter location points must be densified, which strongly constraints the phase of curve association to reveal notable geometrical differences between curves. On the other hand, the nodal sequences and the polygons of control of the B-spline curve are sufficiently different to reveal different kinematical behaviours.

On the numerical controller used, the polygon of control plays a significant role during the phases of acceleration and deceleration. The lack of information about the NC behaviour enables us to only bring out qualitative rules. The number of poles must be minimal if the acceleration parameter is too weak to reach the programmed feedrate.

REFERENCES

Altintas, Y., Erkomaz, K., 2003, "Feedrate optimization for Spline optimization in High Speed Machine tool", Annals of the CIRP, Vol. 51(1), pp. 297-302

Cheng, M.-Y., Tsai, M.-C., Kuo, J.-C., 2002, "Real-time Nurbs command generators for CNC servo-controllers", International Journal of Machine Tools and Manufacture, Vol. 42(7), pp. 801-813.

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

Langeron, J.-M., Duc, E., Lartigue, C., Bourdet, P., 2004, "A new format for 5-axis tool path computation using B-spline curves", Computer-Aided Design, To be published

Lartigue, C., Thiebaut, F., Maekawa, T., 2001, "CNC tool path in terms of B-spline curves", Computer-Aided Design, vol. 33, pp. 307-319

Osty, G., 2001, “Extraction de particularités sur données discrètes issues de numérisation 3D. Partitionnement de grands nuages de points”, PhD Thesis, ENS de Cachan (France)

Park, H., Kim, K., Lee, S.-C., 2000, "A method for approximate nurbs compatibility based on multiple curve refitting", Computer-Aided Design, Vol. 32, pp. 237-252

Piegl, L., Tiller, W., 1997, "The Nurbs Book", Springer-Verlag Editor

Pourazady, M., Xu, X., 2000, "Direct manipulations of B-spline and Nurbs curves", Advances in Engineering Software, Vol. 31, pp. 107-118

Valette L., Duc E., Lartigue C., 2000, “A method for computation and assessment of NC tool-paths in free-form curve format”, 2nd International Seminar on Improving Machine Tool Performance, La Baule (France), CDRom paper 18 pages.

Vassilev, T.I., 1996, "Fair Interpolation and approximation of B-splines by energy minimization and points insertion", Computer-Aided design, Vol. 28(9), pp. 753-760

