**To link to this article** : DOI:10.1007/s00170-016-9276-y
URL : https://doi.org/10.1007/s00170-016-9276-y

# Simulation of machining errors of Bspline and Cspline

**El Bechir Msaddek**[1] · **Zoubeir Bouaziz**[1] · **Maher Baili**[2] · **Gilles Dessein**[2] · **Mohsen Akrout**[1]

**Abstract** The high-speed milling operation is widely used in industry for the production of aircraft parts, molds, and dies. The machining based on the polynomial programming (especially Bezier or basic spline (Bspline) and cubic spline (Cspline)) brings an interesting gain in cycle time. However, part quality can be degraded by using this type of programming. In this paper, we suggest a simulation methodology for errors caused by the interpolations: Bspline and Cspline in high-speed machining of warped shapes. To do this, we have developed analytical models expressing the basic paths of these interpolations. Then, we have designed a simulation tool based on these models. Experimental verifications have been done to validate our approach.

**Keywords** Polynomial interpolation · Bspline · Cspline · Modeling · Simulation · Errors · High-speed milling

✉ El Bechir Msaddek
elbechir.msaddek@gmail.com

Zoubeir Bouaziz
Zoubeir.Bouaziz@enis.rnu.tn

Maher Baili
Maher.Baili@enit.fr

Gilles Dessein
gilles.dessein@enit.fr

Mohsen Akrout
mohsen.akrout@enis.tn

[1] ENIS, Unit of Applied Fluids Mechanics of Process Engineering and Environment, University of Sfax, Sfax, Tunisia

[2] ENIT-INPT, Laboratory Production Engineering, University of Toulouse, Tarbes, France

## 1 Introduction

A spline curve is a sequence of curve segments which are connected together to form a single continuous curve. For example, a collection of *Bezier* curves, connected endwise, may be called a spline curve. The word “spline” comes from the shipbuilding industry, where it originally meant a thin strip of wood that is used by designers as a flexible curve [1]. Mathematical splines were first mentioned by Isaac Jacob Schoenberg [2] and developed by P. Bezier in relation with a CAD/CAM problematic. According to Siemens, the spline functions are intended, mainly, to provide an interface between CAD systems and the machine controllers. Our study will focus on two types of spline interpolation: Bezier or basic spline (Bspline) and cubic spline (Cspline).

The Bspline may be defined by a polynomial of third degree or more. Programmed positions are not the points of the curve, but only *control points* of the spline. The curve does not pass directly through these points, but that it “tends” to them. A *control polygon* of the spline is created, and it links the control points. According to Siemens, the Bspline is ideal for describing tool paths in freeform surface machining. The Cspline is an interpolation by a cubic polynomial. It passes exactly through the points of the curve. It has a low curvature variation. However, the Cspline has a strong tendency to oscillations between these points. The modification of the Cspline has an overall effect. The modification of one point influences many blocks.

Some studies have stated that spline interpolations are ideal for describing tool paths in freeform surface machining. Pateloup et al. [3] have used the interpolation Bspline for approximating arcs of circle and segments for the pocket machining. Langeron et al. [4] have developed a new format (5xnurbs) for the calculation of 5-axis tool paths using Bspline curves. Yang et al. [5] have solved an optimization problem to find a Bspline quadratic curve which gives a smoother and more accurate polyline tool path. Chen et al. [6] have used the Taylor method in order to calculate the parameters of the curve Bspline and also to lead to a better accuracy. Thus, Soori et al. [7] have calculated the dimensional and geometric errors generated by the interpolations proposed by CAM systems. He et al. [8] have minimized the chord error by smoothing a Bspline curve. However, there are few studies that analyze the influence of spline interpolations on the numerical control (NC) machining [9]. In practice, the machining by these functions presents remarkable errors on the workpiece. This requires the development of a simulation methodology of these errors.

The simulation method is based on the determination of the reference nodes. Since Bspline interpolation assimilates these nodes as control points, it is necessary to determine the points of the Bspline trajectory as they are calculated by the controller and simulate them with MATLAB©. The difference between this Bspline trajectory and the theoretical trajectory gives the machining error caused by this interpolation. Furthermore, the Cspline trajectory calculated by the controller passes through the points generated by CAM. The error simulation of this interpolation also passes through the gap modeling between the Cspline trajectory and the theoretical trajectory.

After this step, we develop a post-processor that converts tool paths of Bspline and Cspline interpolation to NC programs including compatible codes with the machine Huron KX10 3-axis and NCU Siemens 840D. This post-processor is a program containing all information necessary for machining the workpiece. It converts tool path (spline, linear, etc.) to an NC program in ISO language. The execution process consists in determining the error gap based on the mathematical models, selected from the bibliography, of Bspline and Cspline trajectories. Thereafter, we finish by the interpretation of these results and their influences on the machining process.

## 2 Simulation methodology

The simulation process of machining errors is shown in Fig. 1.

We consider in this paper that the theoretical trajectory is the linear interpolation because the simulator of the machine controller interpolates by default the CAD trajectory with the linear interpolation [9, 12]. After the simulation of the test piece machining on Mastercam©, we generate the NCI file containing the point coordinates of the tool path in linear interpolation. However, it is necessary to determine the real points of Bspline trajectory as they are calculated by the controller in order to simulate this trajectory under MATLAB©. The tool path modeling is essential for determining the Bspline trajectory and the machining errors. Thus, we also need to simulate the Cspline trajectory. Then, after determining the simulated trajectories, it remains transforming them

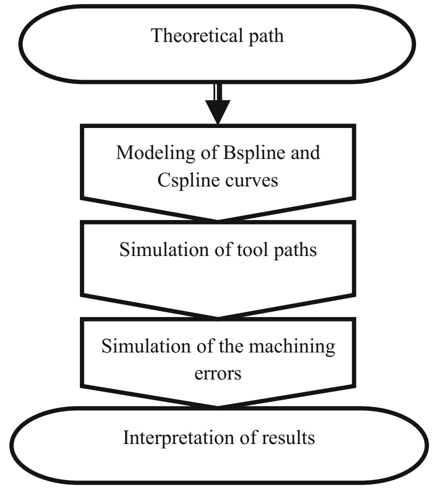



**Fig. 1** Overall simulation approach

into NC files. Subsequently, we analyze the gaps of simulated errors for each interpolation.

## 3 Modeling of Bspline and Cspline interpolations

### 3.1 Bspline interpolation

#### *3.1.1 Basic model*

According to Šulejic [2], a $n$th-degree Bspline curve (of order $n+1$) is defined by

$$C(t) = \sum_{i=0}^{k} N_{i,n}(t)\ P_i\,,\quad \text{with } t \in [a, b] \tag{1}$$

where $P_i$ is $k+1$ control points.

The Bspline curve $C(t)$ is constructed from $k+1$ $n$th-degree basis functions $N_{i\,,\,n}(t)$ defined by recurrence on the inferior degree (Fig. 2).

We consider the vector known as the knot vector be defined $T = (t_0, \ldots, t_m)$ where $T$ is a non-decreasing sequence of real numbers with $t_i \leq t_{i+1}$ and $i = 0\,,\,\ldots\,,m-1$. The $t_i$ is called knots. The $i$th Bspline basis function of

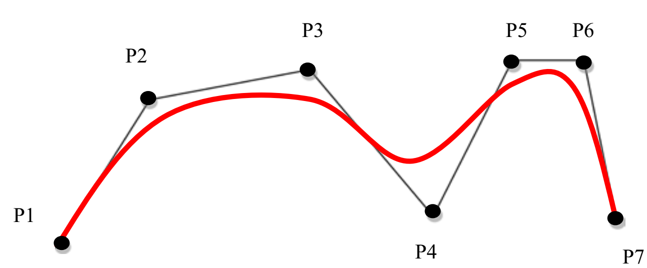



**Fig. 2** Bspline interpolation and approximation of series of points

the $n$th degree is defined by the Cox-de Boor recursion formula

$$N_{i,0}(t) = \begin{cases} 1 & \text{if} \quad t_i \leq t \prec t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$N_{i,n}(t) = \frac{t - t_i}{t_{i+n} - t_i^{\,i}} \cdot N_{i,n-1}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} \cdot N_{i+1,n-1}(t) \tag{3}$$

*3.1.2 De Casteljau algorithm (Bezier curves)*

De Casteljau algorithm is a recursive algorithm which allows presenting the Bezier curves. Thus, the $n$th-degree Bezier curve is constructed from a descriptor polygon of $n+1$ control points.

According to Negulescu [10], a Bezier curve of $n \geq 1$ degree associated with control points $P_0, P_1, \ldots, P_n \in \mathfrak{R}^d$ is called curve $C(t)$ given by the parameterization

$$C(t) = \sum_{i=0}^{n} B_{n,i}(t) \cdot P_i \quad t \in [0, 1] \tag{4}$$

where $B_{n,i}(t)$ is the Bernstein polynomials

$$B_{n,i}(t) = P_n^i \cdot t^i (1-t)^{n-i}, \quad i = 0, \ldots, n \tag{5}$$

where $P_n^i = (\,n \;\; i\,)$ is the binomial coefficients.

The De Casteljau algorithm calculates $C(t)$ recurrently and quickly.

Let $P_i^0 = P_i \forall i = 0, \ldots, n$, then we built $\forall \ t \ \in \ [0, 1]$ recurrently the family of points

$$P_i^j = (1-t) \cdot P_i^{j-1} + t \cdot P_{i+1}^{j-1} \quad \text{for} \quad i = 0, \ldots, n-j, \quad j = 1, \ldots, n \tag{6}$$

*3.1.3 De Boor algorithm (node insertion)*

The De Boor algorithm provides a method for evaluating a Bspline curve. We can find the point of the Bspline curve corresponding to a given parameter value. Any point on a Bspline $C(t)$ has a polar value $C(t, t, \ldots, t)$, and it can be found by inserting the node $(t)$ $n$ times. The De Boor algorithm is based on the use of polar forms. It is easy to understand and is fast and numerically stable for the evaluation or division of a Bspline curve $C(t)$ compared with the value of the specific parameter $t$. This algorithm presents the generalization of the Casteljau algorithm used for Bezier curves.

According to Sederberg and Šulejic [1, 2], the De Boor algorithm is defined by

$$C(t) = \sum_{i=0}^{k+j} N_{i,n-j}(t) \cdot \ P_i^{\,j}, \ \text{ for } \ j = 0, 1, \ldots, n \tag{7}$$

where

$$P_i^j = \left(1 - \alpha_i^j\right) \cdot P_{i-1}^{j-1} + \alpha_i^j \cdot P_i^{j-1} \quad \text{for} \quad j \succ 0 \tag{8}$$

with

$$\alpha_i^j = \frac{t - t_i}{t_{i+n+1} - t_i} \quad \text{and} \quad P_j^0 = P_j \tag{9}$$

The possibility of node insertion, provided by this algorithm, allows the addition of a supplementary node in a Bspline curve without changing the curve shapes. The current basis of the curve is

$$C(t) = \sum_{i=0}^{k} N_{i,n}(t) \cdot \ P_i \ \text{ with } T = (t_0, t_1, \ldots, t_u, t_{u+1}, \ldots) \text{ becomes} \tag{10}$$

$$C(t) = \sum_{i=0}^{k+1} N_{i,n}(t) \cdot \ P_i \ \text{ with } T = \left(t_0, t_1, \ldots, t_u, t, t_{u+1}, \ldots\right) \tag{11}$$

The new De Boor points are

$$\hat{P}_i = (1 - \alpha_i) \cdot P_{i-1} + \alpha_i \cdot P_i \tag{12}$$

where

$$\alpha_i = \begin{cases} 1 & \text{if} \quad i \leq u - k + 1 \\ 0 & \text{if} \quad i \geq u + 1 \\ & \frac{t - t_i}{t_{u+k-1} - t_i} \end{cases} \quad \text{if} \quad u - k + 2 \leq i \leq u \tag{13}$$

## 3.2 Cspline interpolation

The interpolation by high-degree polynomials leads sometimes to large amplitude oscillations. This explains the

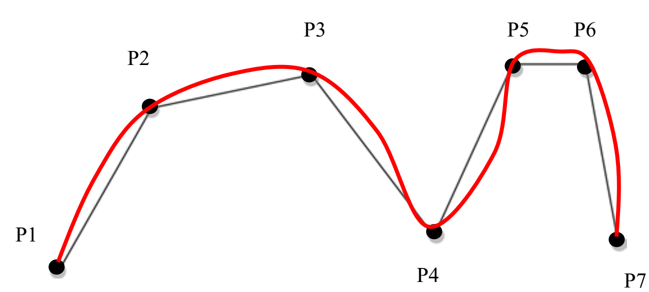
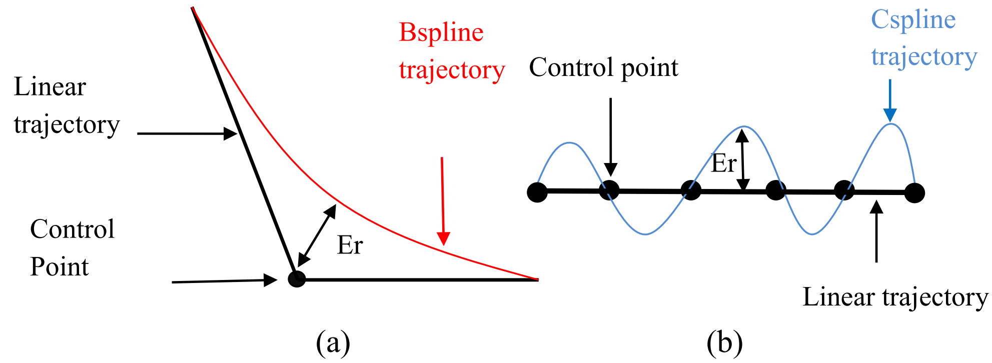



**Fig. 3** Cspline interpolation and approximation of a series of points

Fig. 4 Modeling of the machining errors in Bspline (**a**) and Cspline (**b**)




widespread use of cubic splines which generate interesting regularities of obtained curves. The cubic spline is a series of third-degree polynomials (Fig. 3).

According to Rabut [11], there are several ways to write cubic splines. Here, we present the analytical formula used for simulation. We consider locally the spline function as a third-degree polynomial. For a given $x$, it is first necessary to determine the interval $[x_j \ldots x_{j+1}[$ in which $x$ is located, and then the cubic polynomial is calculated at which the spline is equal to this interval. For this, we write the polynomial as a local form, related to the interval $[x_j \ldots x_{j+1}[$ (limited development in $x_j$)

$$p(x) = p(x_j) + (x - x_j)p'(x_j) + \frac{(x - x_j)^2}{2} p''(x_j) + \frac{(x - x_j)^3}{6} p'''(x_j) \tag{14}$$

with

- $p'(x_j)$, $p''(x_j)$, and $p'''(x_j)$ are respectively the first derivative, the second derivative, and the third derivative of the polynomial $p(x)$ in $x_j$.
- $x$ and $x_j$ are respectively the abscissa of the spline point and the abscissa of the reference point.

## 4 Modeling of machining errors

The controller simulator interpolates the programmed points in CAM by following the algorithm of the polynomial interpolation. Generally, the interpolation generated by the CAM system is the linear interpolation. The points of this interpolation are considered as control points for Bspline interpolation. The Cspline curve passes through these points with sometimes unwanted ripples. Therefore, we deduce the concept of the interpolation error "Er." This error defines the gap between the trajectories: linear and polynomial (Fig. 4).

## 5 Simulation of machining errors (Bspline and Cspline)

### 5.1 Bspline trajectory and machining errors

The Bspline is usually calculated by algorithms that do not require evaluating the basic functions where they are at zero. According to the models developed previously, two algorithms are possible to trace the tool path in Bspline. Since the model used by the machine controller (Siemens 840D) is unknown, we resort to the trajectory simulation by the two algorithms of De Casteljau and De Boor, in order to know the working

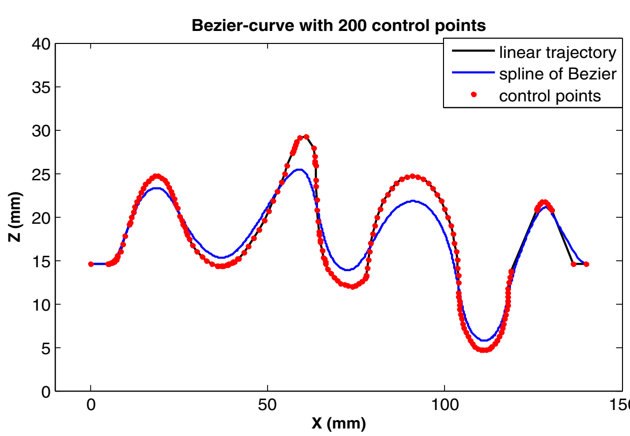



**Fig. 5** Simulation of Bspline interpolation (Bezier spline) based on the De Casteljau algorithm

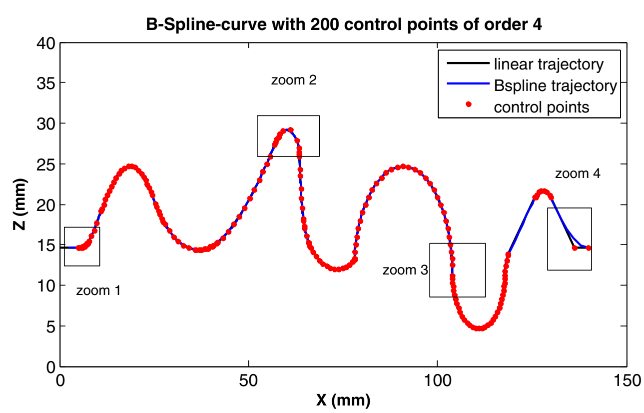
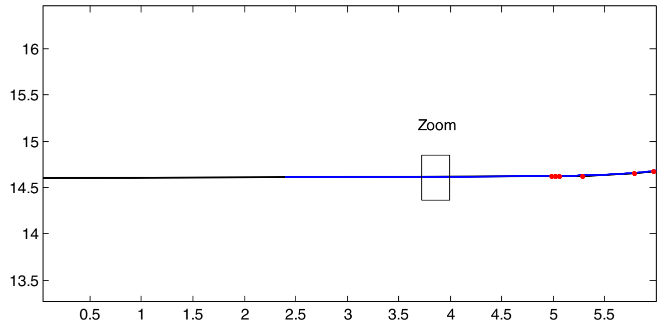
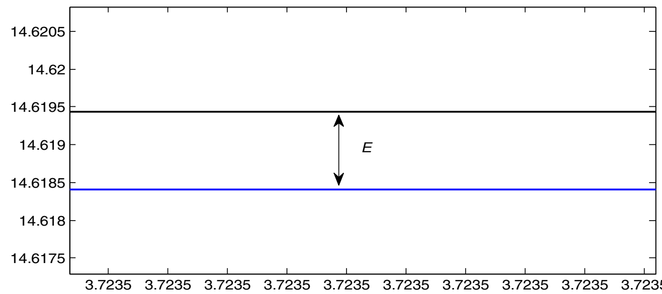
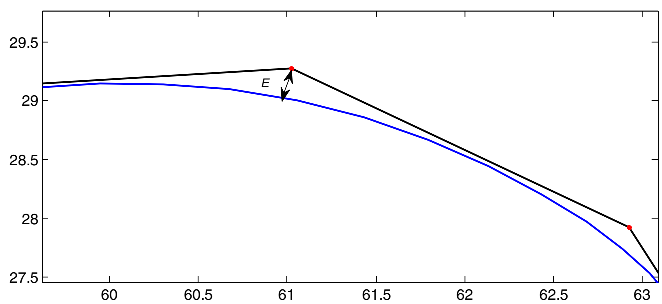
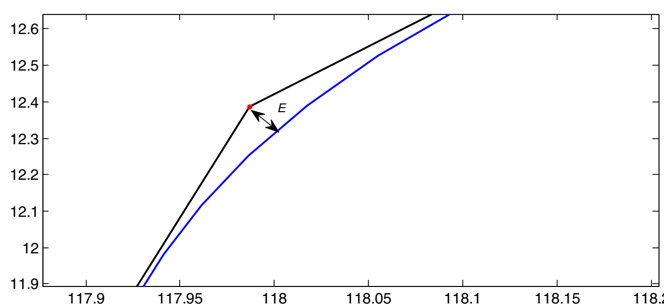
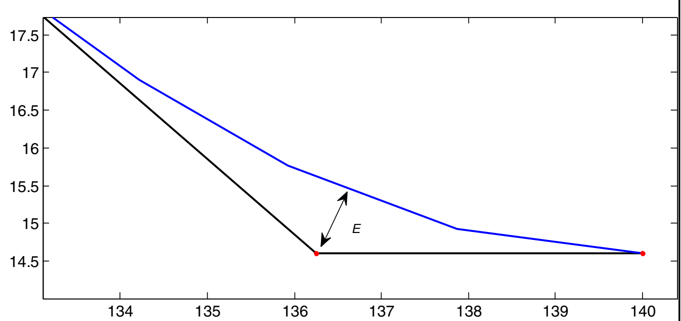



**Fig. 6** Bspline interpolation (basic spline) based on the De Boor algorithm

**Table 1** Zooms of the Bspline trajectory and simulated errors

| Zoom | Schema | Simulated error ($E$, µm) |
| --- | --- | --- |
| 1 |   | 1 |
| 2 |  | 260 |
| 3 |  | 160 |
| 4 |  | 1622 |

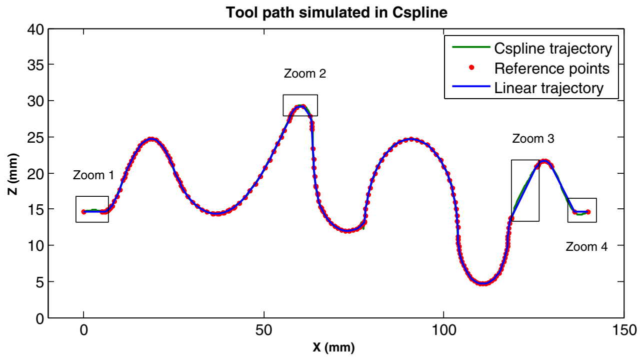



**Fig. 7** Simulation of the Cspline interpolation based on the analytical model

algorithm that can be treated subsequently. The points of the tool path generated by CAM (200 points) are considered as control points in the NCU for the Bspline interpolation.

Figure 5 shows the Bspline trajectory of Bezier simulated by the De Casteljau algorithm with the linear interpolation and the control points.

Figure 6 shows the Bspline trajectory simulated by the De Boor algorithm with the linear interpolation and the control points (200 points).

According to the experimental verification on the machine Huron KX10 [12] and comparing the simulated trajectory by the machine and those simulated in MATLAB© previously, we consider that the work of the NCU Siemens 840D is based on the De Boor algorithm for polynomial interpolation of Bspline type. In the following, we will continue by this algorithm in order to simulate the generated errors during machining with this type of interpolation.

Table 1 presents the different zooms of some critical areas (areas with maximum errors) and the values of simulated errors.

The simulation method based on the De Boor algorithm for the Bspline (basic spline) interpolation has shown good efficiency. The simulated error can reach more than 1.6 mm in critical areas.

### 5.2 Cspline trajectory and machining errors

According to the model developed previously, the trajectory of the machining tool can be plotted in Cspline passing by the reference points generated by CAM. The interpolation model in Cspline, used by the machine controller (Siemens 840D), is also unknown. First, we simulate the Cspline trajectory by the controller simulator (experimental verification). Then, this trajectory is simulated using the analytical model developed in MATLAB©. Afterward, a comparison is made to determine the actual model used by the NCU simulator.

Figure 7 shows the Cspline interpolation with linear trajectory and control points (red dots).

According to the experimental verification made on the machine Huron KX10 [12], we conclude that the work of the NCU Siemens 840D is based on the model described by Rabut for Cspline interpolation. In the following, we will continue with this model for the simulation of errors generated during machining with this type of interpolation.

Table 2 presents the different zooms of some critical areas and the values of the simulated errors.

The simulated errors can reach more than 600 μm in critical areas. The finer discretization in Cspline causes the increasing of point's number. So, the error value becomes smaller but the machine undergoes more fluctuations.

## 6 Interpretation

The maximum simulated errors are about 600 μm for the Cspline and about 1600 μm for the Bspline. Indeed, there is a big difference between the errors generated by the two interpolations. For more precision, during the machining of warped shapes by smooth interpolations such as interpolations based on spline, it is essential to know the order of the error magnitude at the passage of concave and convex forms.

During machining of planar surfaces, the errors generated by the Bspline interpolation are lower than those generated by the Cspline interpolation. The defect of the surface machining (zoom 1) is almost nil in Bspline, while it reaches 230 μm in Cspline (large fluctuations). Thus, the Cspline trajectory is favorable to the change of direction during machining, because it passes through the control points. On the contrary, for Bspline trajectory, the error exceeds 1.6 mm when crossing the discontinuity in tangency (zoom 4). Therefore, the slowdown of the machine diminishes at the passage of discontinuities. This keeps the feed rate near to the set speed. For convex shapes of the warped shape (as shown in zoom 2), the Bspline interpolation reduces the workpiece matter, while the Cspline interpolation increases it. The opposite is true for the concave shapes. The errors generated on these forms are very close.

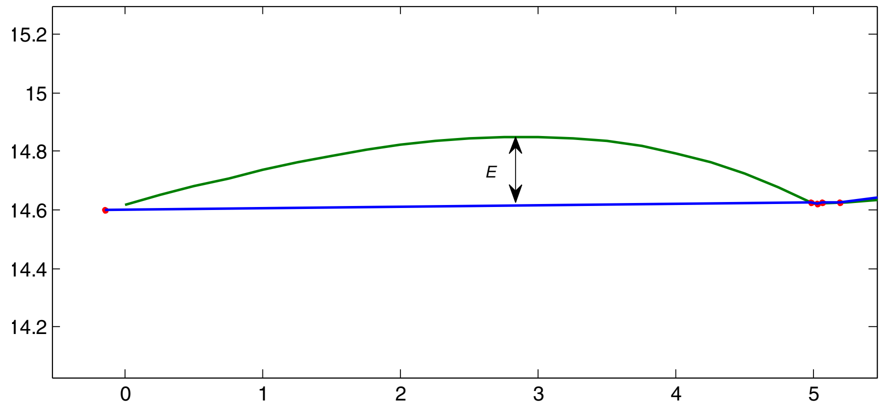
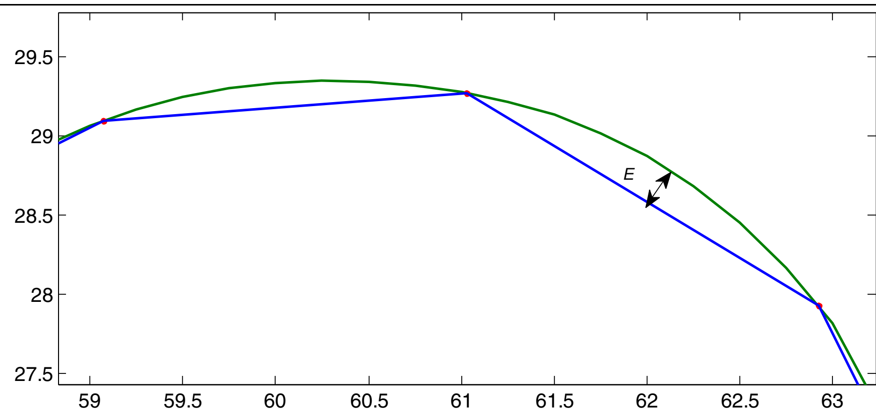
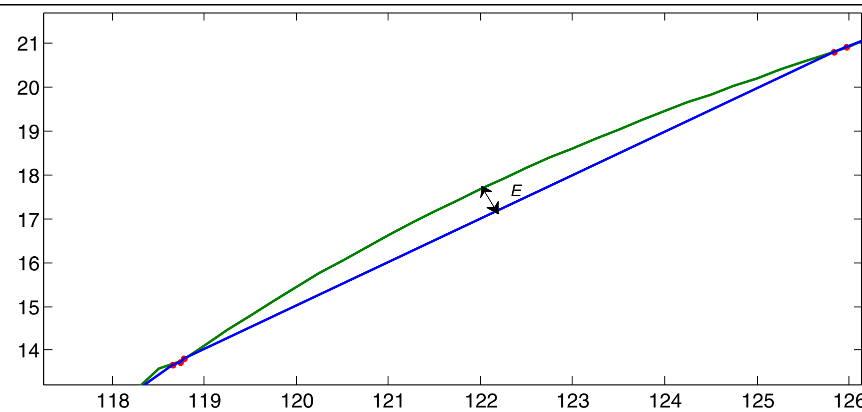
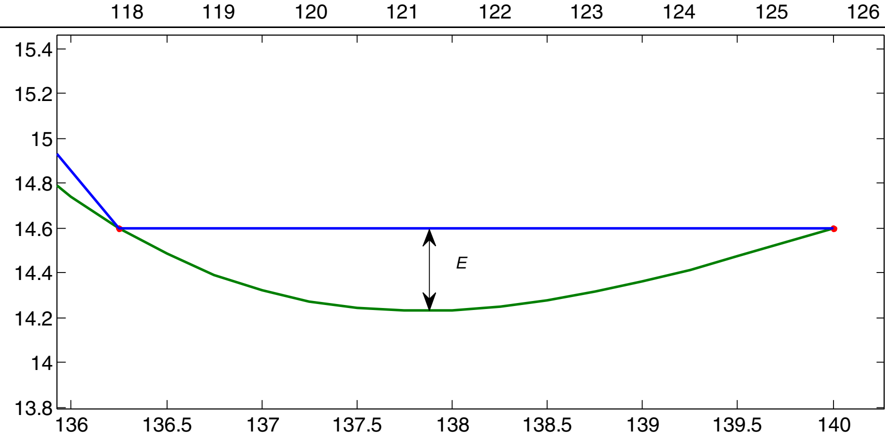
Figure 8 shows the comparison histogram of the errors generated by the two interpolations: Bspline and Cspline of four areas of the workpiece.

## 7 Conclusion

After modeling and simulating Bspline and Cspline interpolations, calculated by the controller, our choice was directed towards the method of simulation nodes and associated trajectories for evaluating the machining errors. We have appreciated the benefits of this method regarding the accuracy of the recorded values.

After the modeling of the Bspline and Cspline trajectories, we have developed a postprocessor of NC file generation for error simulation. The quantification of machining errors to functions "spline" as Bspline and Cspline allowed us to achieve

**Table 2** Zooms of Cspline trajectory and simulated errors

| Zoom | Schema | Simulated error (µm) |
|---|---|---|
| 1 |  | 230 |
| 2 |  | 313 |
| 3 |  | 600 |
| 4 |  | 376 |

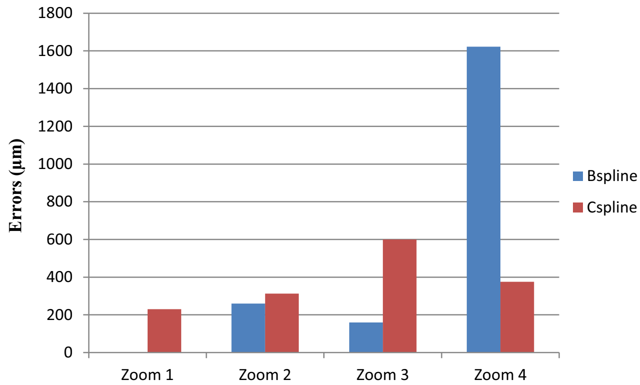



**Fig. 8** Histogram comparing the simulated machining errors of interpolations: Bspline and Cspline

relevant results about the accuracy of machining warped shapes by using these types of polynomial interpolations.

The Bspline interpolation generates fewer errors on the convex and concave shapes of the warped shape, but it generates large errors when crossing the discontinuities in tangency. In contrast for the interpolation Cspline, the passage of the trajectory by the reference points increases the precision in changes of direction, but it causes the deceleration of the machine.

## References


1. Thomas W. Sederberg, "An introduction to B-spline curves" 2005.
2. Šulejic M (2011) Geometric modeling: B-spline and NURBS curves. University of Salzburg, Department of Computer Science, Austria
3. Pateloup V, Duc E, Ray P (2010) Bspline approximation of circle arc and straight line for pocket machining. Comput Aided Des 42: 817–827
4. Langeron JM, Duc E, Lartigue C, Bourdet P (2004) A new format for 5-axis tool path computation, using Bspline curves. Comput Aided Des 36:1219–1229
5. Yang Z, Shen L-Y, Yuan C-M, Gao X-S (2015) Curve fitting and optimal interpolation for CNC machining under confined error using quadratic B-splines. Comput Aided Des 66:62–72
6. Chen M, Zhao W-S, Xi X-C (2015) Augmented Taylor's expansion method for B-spline curve interpolation for CNC machine tools. Int J Mach Tools Manuf 94:109–119
7. Soori M, Arezoo B, Habibi M (2013) Dimensional and geometrical errors of three-axis CNC milling machines in a virtual machining system. Comput Aided Des 45:1306–1313
8. He S, Daojiang O, Yan C, Lee C-H (2015) A chord error conforming tool path B-spline fitting method for NC machining based on energy minimization and LSPIA. J Comput Design Eng 2:218–232
9. Msaddek EB, Bouaziz Z, Baili M, Dessein G (2014) Influence of interpolation type in high speed machining (HSM). Int J Adv Manuf Technol 72(1–4):289–302
10. Negulescu C (2007-2008) "Interpolation and approximation: Bezier curves". Aggregation of mathematics. University of Provence, year, CMI
11. Rabut C (2009-2010) "Modeling, part one: natural cubic splines". Department of Science and Techniques for Engineers, INSA, Toulouse
12. E. B. Msaddek, "Optimizing trajectories of milling tools at high speed by taking into account the constraints associated with the MT-NC couple", Doctoral thesis of the National School of Engineers of Sfax, December 24, 2014.