CAD/CAM INTEGRATION FOR NURBS PATH INTERPOLATION ON PC BASED REAL-TIME NUMERICAL CONTROL

M. Leto, R. Licari[1], E. Lo Valvo[1], M. Piacentini[1]

[1] Dipartimento di Tecnologia Meccanica, Produzione ed Ingegneria Gestionale, University of Palermo, Italy

KEYWORDS: NURBS, CAD/CAM, Numerical Control.

ABSTRACT. As anybody knows, there is a fundamental difference between the newest CAD system and the conventional Numerical Control. As a matter of fact, almost every CAD Software allows programmers to use parametric 2D or 3D curves, whereas standard Numerical Control cannot perform anything more than linear or circular interpolations. NURBS are the most successful among all the proposed parametric curves, since they allow us to give an exact uniform expression to both the analytical and the free form curves. Nevertheless, this typical characteristic nowadays entails that in CAM systems a tool path is rounded down using a small movements and imposing a maximum acceptable gap. This approach leads up to a number of problems of machining quality and wastes machining time. In this paper the authors propose a software able to automatically generate, from a CAD geometric model, instructions for a Numerical Control running on a common Personal Computer.

1 INTRODUCTION

Ever more frequently the modern industry is manufacturing parts presenting curves and complex surfaces, such as moulds and sculptured surfaces. On one hand this need has promptly been met by both the CAD and the CAM systems, through such parametric curves as B-Spline and NURBS (Non Uniform Rational B-Spline), but the same facilities cannot be found in most of the Numerical Control Machines utilized in the manufacturing processes.

By now, when a curve path - being not circumference arcs - has to be processed, the common approach is to calculate a great number of points on the curve and connect them with either straight segments or predefined-tolerance arcs [1]. In this way, the curve will be approximated and the final shape is as much precise as the tolerance becomes smaller – that is as the elements composing it are increasing. But this engenders such undesired effects [2] as: a) a variance in feed rate and speed discontinuity in the step from a segment to the subsequent one; b) a discontinuity in the acceleration, causing vibrations and lowering the working process quality; c) the maximum feed rate cannot be programmed for short segments, so that the higher working speed cannot be reached; d) the subdivision in short segments produces a great quantity of data, overcharging the CNC system.

Recently, a number of different solutions have been proposed in literature in order to solve the above mentioned problems [2-7], but all of them were employed on laboratory CNC tool machines, which cannot be used in a production environment.

In the present work a method based on the arc function - as it is known present in all CNC machine tools - has been developed to make possible to manufacture NURBS curves using the standard hardware and software characteristics of commercial CNC machine tools.
The proposed solution has been implemented on EMC2 [8-9], a Open Source software system for computer control of machine tools. The chosen software has allowed us to add both a new G code for the definition of plain complex curves (e.g.: B-Spline and NURBS) and a new interpolation function able to generate the corresponding path. This system has been used in a number of working processes with newsworthy results.

2 EMC2 (Enhanced Machine Controller)

The EMC software [9] was firstly developed by NIST (National Institute of Standards and Technology). It is an Open Source software, which can be used by everyone. Firstly developed for the Windows NT Real Time version, it was later conveyed on a Linux version with Real Time extensions, both for economic reasons and because of the better stability and efficiency of the Linux operating system. At the beginning this project aimed at providing the USA Small and Medium-Sized Enterprises with a low-price alternative to the traditional Numerical Controls. Its primary function was to work as a Personal Computer Numerical Control for traditional machines which have been converted into automatic.
EMC, nevertheless, has caught the attention of a great amount of people, both programmers and non-programmers, and has grown up in the fertile co-operating environment of the Open Source software. The main difficulty which has been found in installing the Real Time Linux extensions has been overcome by the EMC2 version which supplies the first versions of the program on a CD containing the modified, functioning operating system. Nowadays these versions have evolved in the shape of a live-CD, which is a bootable CD-ROM which contains both the Real Time Operating System and the program [8]. This way they can both be used and tested on a common PC without being installed and without modifying the existing system.
As said before the EMC2 is both an Open-Source and an Open Architecture software. The first feature allows it to be freely distributed with its own source code, so that the user is able to study its internal configuration and modify it according to his needs [10-11]. The second feature regards to the real structure of the source code, which is modular and extensible. This way the user is allowed to add, modify, update, improve or substitute parts of it in a quite easy way, with targeted interventions only on the section he is interested in. The developer is able to modify the source code, implementing specific functions, such as adding new codes matching his peculiar needs. This opportunity has been exploited for the development of the present work.

3 THE BIARCS NURBS INTERPOLATION

NURBS stands for Non-Uniform Rational B-Spline. Non-Rational B-Splines are a special case of Rational B-splines, just as Uniform B-Splines are a special case of Non-Uniform B-Splines. Thus, Non-Uniform Rational B-Splines encompass almost every other possible 3D shape definition.
A NURBS curve is defined as:

$$p(u) = \sum_{i=0}^{n} p_i R_{i,k}(u) \qquad (1)$$

where p_i are the control points and $R_{i,k}(u)$ are defined as:

$$R_{i,k}(u) = \frac{N_{i,k}(u) w_i}{\sum_{j=0}^{n} N_{j,k}(u) w_j} \quad (2)$$

in which w_i are the "weights" and $N_{i,k}(u)$ are B-Spline basis functions, defined as:

$$\begin{cases} N_{i,k}(u) = 1 & t_i \le u \le t_{i+1} \\ N_{i,k}(u) = 1 & otherwise \end{cases} \quad for \quad k = 1$$
$$N_{i,k}(u) = \frac{(u - t_i) N_{i,k-1}(u)}{t_{i+k-1} - t_i} + \frac{(t_i + k - u) N_{i+1,k-1}(u)}{t_{i+k} - t_{i+1}} \quad (3)$$

Parameters t_i are knots and they are calculated in the open uniform form.

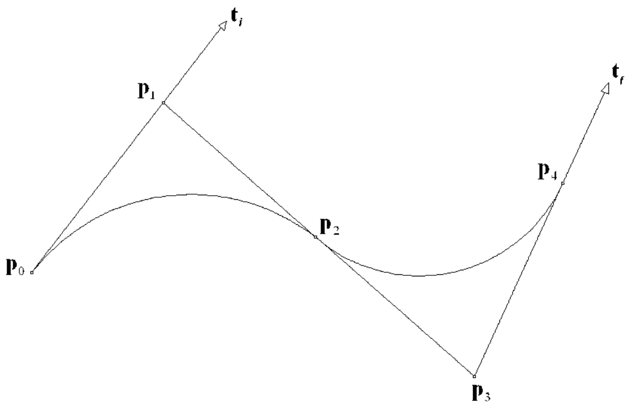


FIGURE 1. Biarc definition

NURBS points are interpolated using biarcs [12]. A biarc consists of two smoothly connected circular arcs that interpolate two end points and two end tangents. It is univocally defined when points p_0, p_1, p_2, p_3, p_4 are known.

Points p_0 and p_4 are starting and ending point of the biarc. Points p_1, p_2 and p_3 are defined as follow:

$$p_1 = p_0 + \alpha t_s \quad (4)$$

$$p_2 = \frac{\beta}{\alpha + \beta} p_1 + \frac{\alpha}{\alpha + \beta} p_3 \quad (5)$$

$$p_3 = p_4 + \beta t_e \quad (6)$$

where α and β are two positive parameters.

Combining (4), (5) and (6) will give the following second order equation in α and β:

$$v \cdot v + 2\beta v \cdot (\alpha t_s + \beta t_e) + 2\alpha\beta \cdot (t_s t_e - 1) = 0 \quad (7)$$

where v is:

$$v = p_0 - p_4 \quad (8)$$

The values of α and β determine the position of point p_2, for this application we will define the ratio r as:

$$r = \frac{\alpha}{\beta} = 1 \quad (9)$$

The equation (7) is defined in $\Re$ and it can be solved if t_s and t_i are known.
The proposed algorithm calculates a number of NURBS points, that depends from the number of defined control points, and interpolates them using biarcs. For every couple of NURBS points (P_i, P_{i+1}) a biarc is calculated considering that:

- P_i and P_{i+1} are the starting and ending points of the biarc defined as p_0 and p_4 before.
- the starting tangent is the ending tangent of the precedent biarc.

In this way all biarcs are tangents each others and a first order curve will result leading to a more stable feederate.
For every biarc the ending tangent t_e is unknown and it is calculated using a numeric method. When t_e is known, the equation (7) can be solved and points p_1, p_2 and p_3 can be found so that the biarc is univocally defined. Finally, centre and radius of the two arcs are calculated so that arc can be executed by EMC2 real time trajectory planner.
It has been said that - even in the newest CAD/CAM systems - if the NURBS are used for the generation of the tool-path, a part-program has to be created, containing instructions for either linear or circular paths. But this affords a number of undesired effects on the feed rate during the manufacturing process. Moreover, such a part-program is also composed of a huge number of lines, so that it is difficult for the user to manage it.
In order to solve - at least in a measure – the aforesaid problems, the Open Source EMC2 Numerical Control source code has been modified, in a way that a specific G-Code can be added for a practical and efficient description of the B-Spline and NURBS curves in the plane, so that an interpolation of the curve can easily be made, allowing at the same time the operator to remarkably reduce the problems related to feed rate.
The two new codes introduced are G5.2 and G5.3 for opening and closing the data block defining a NURBS. In the lines between these two codes the curve control points are defined with both their related "weights" (P) and their parameter (L) which determines the order of the curve (k), and subsequently its degree (k - 1). An example is shown in Figure 2.
The part-program is easy to be read and to perform instant modifications to the tool-path even without the help of a CAD/CAM (which is pretty much impossible with the classical approach). As a matter of fact, with this kind of definition it is even possible to perform B-Spline curves assuming that all weights are equal to 1. Using this curve definition the knots of the NURBS

curve are not defined by the user: they are calculated by the inside algorithm, in the same way as it happens in a great number of graphic applications, where the curve shape can be modified only acting on either control points or weights. This feature makes it easier to write codes than those used in literature [3,6].

```
G0 Z30
G0 X2 Y-1
G0 Z2
G1 Z-5 F100
F600

G5.2 X3.00    Y3.00    P1 L3
     X4.33    Y8.35    P1
     X5.00    Y10.50   P2
     X8.44    Y17.58   P1
     X15.32   Y16.03   P1
     X14.05   Y9.75    P1
     X10.03   Y9.40    P1
     X7.87    Y12.30   P1
     X10.56   Y15.00   P1
     X12.45   Y14.00   P1
     X12.03   Y13.00   P1
G5.3

G0 Z30
G0 X-12.03 Y13.00
G0 Z2
```

NURBS code

FIGURE 2**.** An example of the proposed G-code

4 CAD INTERFACE

For a better and more efficient implementation of the proposed technique, it has been developed a specific routine inside a NURBS-oriented CAD (Rhinoceros 4.0 [13]) able to generate the part-program section related to the curve description. This part-program has been processed by a EMC2 Numerical Control modified by the authors.

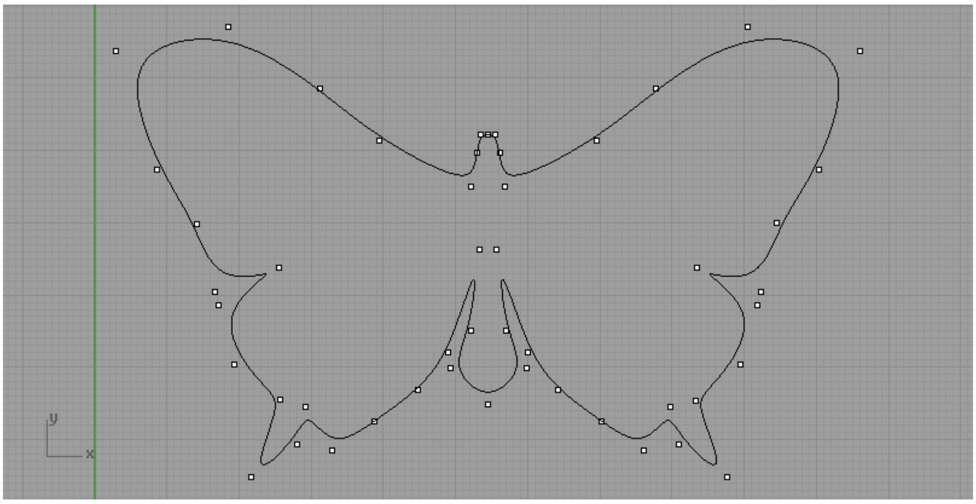
This action was performed, thanks to the programmability features of the adopted CAD, through a module called “RhinoScript” which uses an easy programming language, similar to VBScript. “RhinoScript” allows the operator to access the CAD model curves in the database, and to generate, in a thoroughly automatic way, the ISO codes for machining, according to the previously defined format.

5 EXPERIMENTAL RESULTS

In order to verify the validity of this procedure, some examples present in literature have been taken into consideration. In particular we have selected two Butterfly shapes [5-6].

In Figure 3 and 4 are shown the four steps needed to perform the test:

a) draw the shape using Rhinoceros 4.0
b) generate the part-program using RhinoScript
c) preview of the path on EMC2
d) machined part



a)

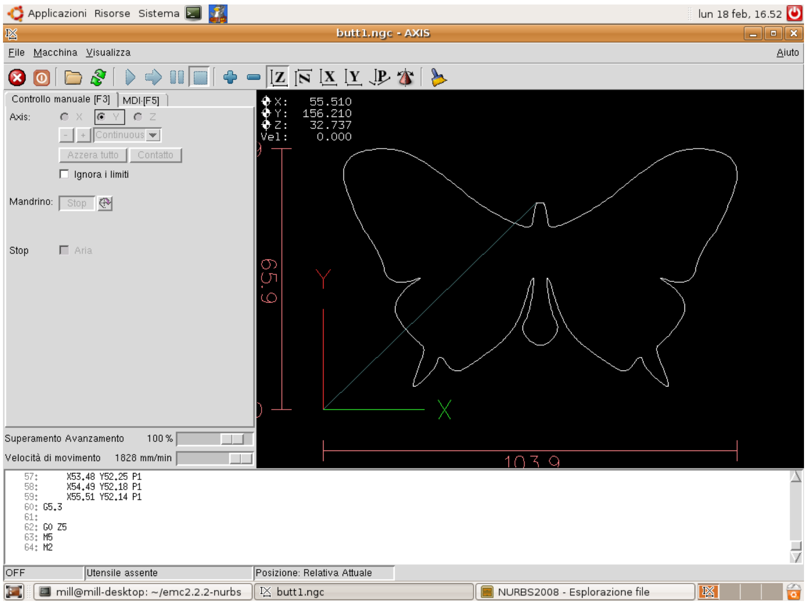
```
G0 Z10
G0 X53.48 Y52.14
M3 S1000
G0 Z1
G1 Z-1 F100
F600

G5.2 X54.49 Y52.14 P1 L3
     X55.51 Y52.14 P1
     X56.08 Y49.62 P1
     X56.78 Y44.97 P1.2
     X69.58 Y51.36 P1
     X77.79 Y58.57 P1
     X90.53 Y67.08 P1
     X105.97 Y63.80 P1
     X100.40 Y47.33 P1
     X94.57 Y39.91 P1
     X92.37 Y30.48 P1
     X83.44 Y33.76 P2
     X91.89 Y28.51 P1
     X89.44 Y20.39 P1
     X83.22 Y15.45 P5
     X87.62 Y4.83 P3
     X80.94 Y9.27 P1
     X79.83 Y14.54 P1
     X76.07 Y8.52 P1
     X70.18 Y12.55 P1
     X64.17 Y16.86 P1
     X59.99 Y22.12 P1
     X55.68 Y36.36 P1
     X56.92 Y25.00 P1
     X59.77 Y19.83 P1
     X54.49 Y14.94 P1
     X49.22 Y19.83 P1
     X52.06 Y24.99 P1
     X53.31 Y36.36 P1
     X49.22 Y22.12 P1
     X44.81 Y16.86 P1
     X38.80 Y12.55 P1
     X32.91 Y8.52 P1
     X29.15 Y14.54 P1
     X28.04 Y9.27 P1
     X21.63 Y4.83 P3
     X25.77 Y15.45 P5
     X19.36 Y20.39 P1
     X17.10 Y28.51 P1
     X25.54 Y33.75 P2
     X16.60 Y30.50 P1
     X14.20 Y39.80 P1
     X8.67 Y47.41 P1
     X3.00 Y63.79 P1
     X18.47 Y67.08 P1
     X31.20 Y58.57 P1
     X39.41 Y51.36 P1
     X52.20 Y44.97 P1.2
     X52.90 Y49.61 P1
     X53.48 Y52.14 P1
     X54.49 Y52.14 P1
     X55.51 Y52.14 P1
G5.3

G0 Z5
M5
M2
```

b)

FIGURE 3. a) Rhinoceros model b) NC code obtained with RhinoScript

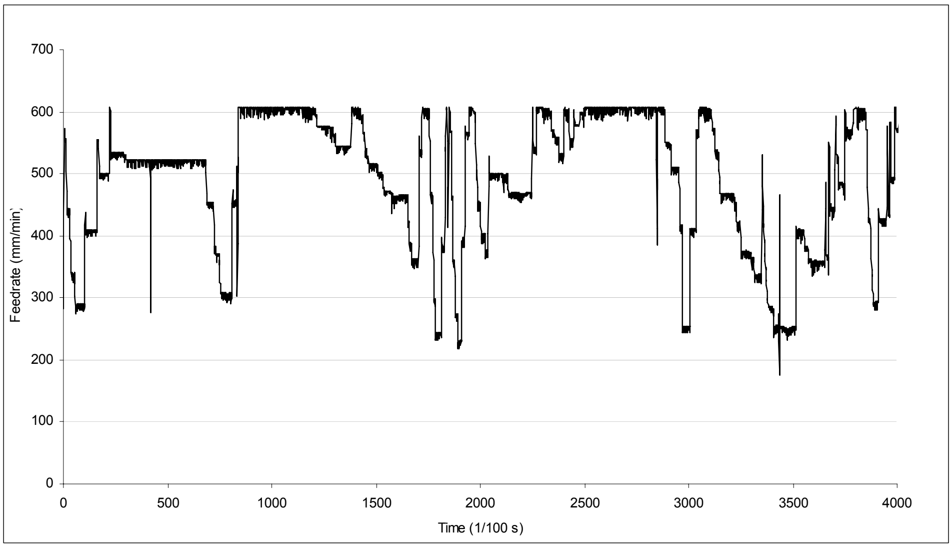


a)

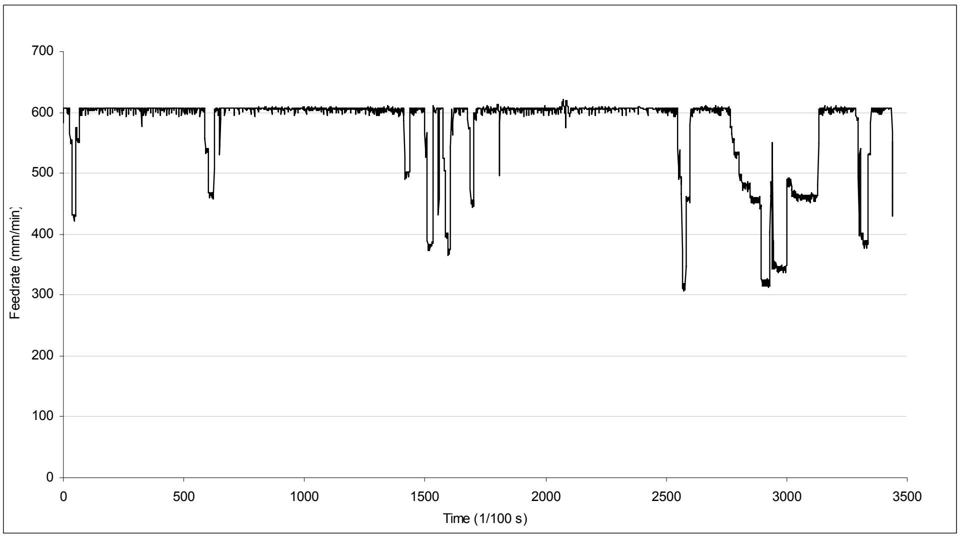
b)

FIGURE 4. a) EMC2 screenshot b) machined part

More specifically, as for the example reported in [6], a comparison was made between a commercial CAM-generated part program and the new procedure-generated one.
The obtained code was composed of around 1000 code lines, whilst the one obtained by the proposed method is made of less than 100 lines.



a)

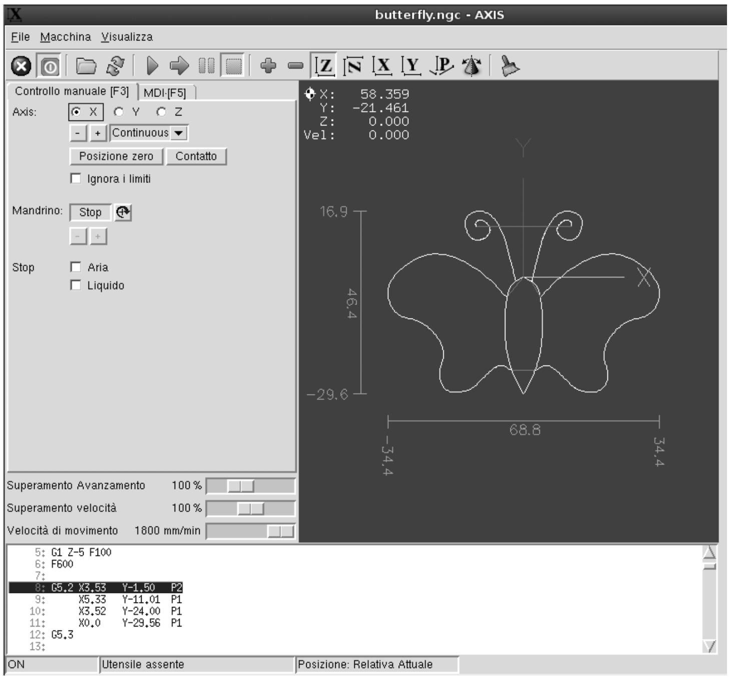


b)

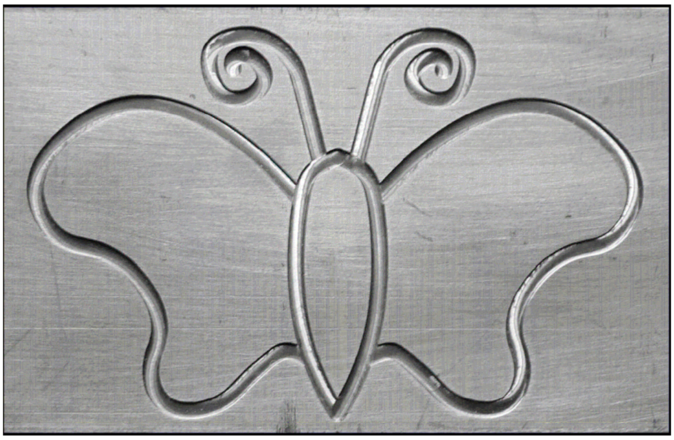
FIGURE 5**.** Feed rate trend: a) without NURBS interpolation b) using NURBS interpolation

In Figure 5 is reported the feed rate trend without NURBS interpolation and using NURBS interpolation.

Moreover, it is immediately clear that the feed rate trend, except for some brief droop caused by momentary variations of the real profile curve, is far more constant in the second than in the first case, where the variations are wide and frequent.



a)



b)

FIGURE 6**.** a) EMC2 interface b) corresponding machining test

It has also to be pinpointed that, with a predefined feed rate of 600 mm/min, in the first case the average feed rate was 493 mm/min, while in the second case it was 575 mm/min. The machining time was of 40 seconds in the first case, while in the second it was more than 33 seconds, with a difference of around 20% .

In order to prove the real validity of the analysed method, not only a simulation has been performed, but also a real machining of the profile with a small numerical control three axes milling machine commanded by EMC2 through the parallel port of a common PC running a Linux Ubuntu Operating System with Real Time extensions [10]. The Figure 6 shows the machining result.

6 CONCLUSION

In this work EMC2, a NC Open Source software, installed on Real Time Linux, has been customised for the machining of complex curves on the plane, such as B-Spline and NURBS with a three-axes milling machine. Thanks to its features, it has been possible to not only to configure the software in the optimum way for the machine management, but also modify the source code by adding a new function for both the definition of curves and the performing of the desired machining operation. These results have been easily obtained thanks to the direct use of the information from a NURBS oriented CAD.

It is hence possible to identify the following advantages in the proposed method:

a) a better shape and manageability of the part-program containing such complex curves as NURBS and B-Spline;
b) a higher reading easiness, which makes it possible to immediately modify the tool-path;
c) an easier implementation of the interpolation method using biarcs, which does not request complex calculations, exploiting the circular interpolation functions which are already present in every NC;
d) a better movement fluidity, with a lower machining time than in the case of classical interpolation method, thanks to the possibility to reach a higher feed rate;
e) direct use of a user-friendly interface easy to be implemented within a common CAD software.

In the end, this method was validated by experimental tests demonstrating its actual functionality.

7 ACKNOWLEDGEMENTS

Authors would like to thank to Mr. Jeff Epler for his support and suggestions.
This work has been performed with the financial support of M.I.U.R. of Italy.

REFERENCES

1. Zelinsky, P., (1999), Understanding NURBS Interpolation, http://www.mmsonline.com/ articles/079901.html
2. Lin, M.T., Tsai, M.S., and Yau, H.T., (2007), Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm, International Journal of Machine Tools and Manufacture, Vol 47/15, 2246-2262.
3. 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, 801-813.
4. Cheng, C.W., Tseng, W.P., (2006), Design and implementation of a real-time NURBS surface interpolator, The International Journal of Advanced Manufacturing Technology, Vol 30/1-2, 98-104.
5. Yau, H.T., Lin, M.T., Tsai, M.S., (2006), Real-time NURBS interpolation using FPGA for high speed motion control, CAD Computer Aided Design, Vol 38/10, 1123-1133.

6. Lei, W.T., Sung, M.P., Lin, L.Y., Huang, J.J., (2007), Fast real-time NURBS path interpolation for CNC machine tools, International Journal of Machine Tools and Manufacture, Vol 47/10, 2246-2262.
7. Hong-Tzong, Y., Jun-Bin, W., Chien-Yu, H., Chih-Hua, Y., (2007), PC-based Controller with Real-time Look-ahead NURBS Interpolator, Computer-Aided Design & Applications, Vol 4/1-4, 331-340.
8. Enhanced Machine Controller – EMC, http://www.linuxcnc.org
9. Proctor, F.M., Shackleford, W., Yang, C., Barbera, T., Fitzgerald, M.L., Frampton, N. Bradford, K., and Koogle, D, (1995), Simulation and Implementation of an Open Architecture Controller. Modeling, Simulation, and Control Technologies for Manufacturing, Proc. of the SPIE 2596.
10. Licari, R., Vanoli, R, Lo Valvo, E., (2006), Internet based development of an open source low-cost CNC, Proc. of the 17th International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Mechatronics & Robotics", 423-424.
11. Licari, R., Vanoli, R, Piacentini, M., (2006), A more simple G-code programming within an Open Source CNC system, Proc. of the 17th International DAAAM Symposium "Intelligent Manufacturing & Automation: Focus on Mechatronics & Robotics", 221-222.
12. Park, H., (2004), Optimal Single Biarc Fitting and its Applications, Computer Aided Design and Application, Vol 1, No 1-4, 187-195.
13. Rhinoceros 4.0, http://www.rhino3d.com/