An Efficient NURBS Path Generator for a Open Source CNC

ERNESTO LO VALVO, STEFANO DRAGO
Dipartimento di Ingegneria Chimica, Gestionale, Informatica e Meccanica
Università degli Studi di Palermo
Viale delle Scienze, 90128 Palermo
ITALY
ernesto.lovalvo@unipa.it

Abstract: - In this paper a NURBS path generator (on the plane XY) is proposed to be developed for those CNC machine tools and robot which are Open Source or Opening Architecture. The goal is to use simple and efficient techniques to manufacture geometries which are very complex, whose main feature is the presence of free-form surfaces and contours. To reach this goal, the NURBS path generator has been optimized in order to enable the user to select among three different options to generate the tool path, using a specific parameter. The three options are respectively called "*NURBS interpolation with constant increments of u parameter*" (NICU), "*NURBS interpolation with constant displacement increments by linear motion*"(NICL), "*NURBS interpolation with constant displacement increments by circular motion*" (NICC). The NURBS path generator has been implemented for an Open Source Numerical Control (LinuxCNC) in order to evaluate its efficiency, by simulations and experimental tests, in terms of computational complexity, comparison between the trajectory covered and the path programmed, capability to follow speed and path trends.

Key-Words: - NURBS, Path Generator, CNC, Open Source, LinuxCNC, Interpolation

1 Introduction

In modern manufacturing, machining is mainly focused either on the production of mold for mechanical and aerospace components or on that one of biomedical devices; however, at the same time, it can be employed for finishing operation at the end of a production process of any mechanical element: these are the reasons why the development of systems which make it possible high-speed and precise manufacturing has become one of the main interests of researching activities.

For many years commercial and Open Source Computer Aided Design systems have been offering the possibility of creating bidimensional and tridimensional surfaces or curves (both analytical ones and free-form ones). Among techniques which are employed for the drawing and the design of surfaces and curves, the NURBS (Non-Uniform Rational B-Spline) parametric formulation is widely used: in fact the flexibility, the precision and the simplicity of this kind of drafting enables one to work complex geometries and to modify their shape locally.

In the recent years Computer Aided Manufacturing systems and some high performance Computer Numerical Control ones have been employing NURBS parametric formulation to describe and to create tool paths, offering the possibility of using NURBS interpolation.

However in many conventional CNC machine tools there are only linear and circular path interpolators (i.e. linear and circular interpolation functions): therefore this can be considered an evident limit. In fact, when it is necessary to manufacture a complex model, such as, for example, the mold of a car door or the spirals of a scroll compressor, since these profiles are irregular (they are, without a doubt, pre-form surfaces and profiles) the CAM system is obliged to approximate the real curve with a tool path composed by a series of many segments and arcs. However this method doesn't always enable one to create a manufactured surface exactly coinciding with that one of the CAD; in addition to this, it can cause a poor quality surface finishing as a consequence of frequent variations of tool feed rate, so it could be different from that one programmed before.

However, if the CNC, which is present in the machine tool, have an open architecture (as Open Source), it enables anyone has adequate knowledge to study and modify the source code in order to implement a tool path generator which satisfies the desired requirements.

In this paper, it is shown a NURBS path generator (on the plane XY) for manufacturing of complex curves which can be considered suitable for this kind of application. In particular, the developed

ISBN: 978-960-474-402-2

NURBS path generator has been implemented for the LinuxCNC, an Open Source Numerical Control. This has enabled one to evaluate its efficiency in terms of computational complexity, precision of the covered trajectory compared with the path programmed, capability to follow the feed rate trend.
The algorithms, on which the NURBS path generator is based, have been developed considering the interpolation methods of NURBS path shown in literature [1-5].
In particular, it has been considered De-Boor's algorithm for the parametric NURBS representation [6-9].

2 LinuxCNC

LinuxCNC [3,10] is a software which enables one to transform a common Personal Computer into numerical control for machine tools such as milling machines and lathes, taking advantage of Real Time extensions of Linux operating system.
The main feature of LinuxCNC software is to be Open Source and Open Architecture. This enables one to download free (on the website http://www.linuxcnc.org), to study and to modify the source code: therefore, it is possible to implement new functions or to improve available ones adding new codes.
When it is necessary to manufacture a mechanical component, the current LinuxCNC version enables one to choose only one of the following working options: linear interpolation, circular interpolation, NURBS interpolation, standard working cycles for holes and pockets.
If the mechanical component is characterized by free-form surfaces and profiles, using the NURBS interpolation can be certainly considered the right choice.
However the current NURBS interpolation method doesn't always follow these conditions:

- the correspondence between manufactured surface and that one drawn by CAD software;
- the capability of working with a programmed feed rate;
- the presence of elaboration time (required for the creation of tool path) which can be considered adequate for complex NURBS curves.

Therefore, these operating conditions don't enable one to manufacture easily and efficiently very complex geometries characterized by free-form surfaces and profiles.

3 NURBS Path Generation

3.1 Mathematical NURBS formulation

The NURBS (Not Uniform Rational B-Spline) formulation is a parametric representation of curves and surfaces, both analytical ones and free-form ones [5-6].
A NURBS curve is defined when it is possible to establish:

- A set of n+1 control points;
- A weight for each control point which isn't negative;
- The degree of the curve *p*.

By means of the following expression and the previous parameters, it can be defined a vector, called U and formed by m+1 elements: $U = [u_0,...,u_p,u_{p+1},...,u_j,...,u_{m-p-1},u_{m-p},...,u_m]$

$$\begin{cases} u_i = 0 & i < p+1 \\ u_i = i - p + 1 & p+1 \le i \le n \qquad i = 0,......., \ n+p+1 \\ u_i = n - p + 2 & i > n \end{cases}$$

The elements of knot vector are a growing series of real numbers which can be not recursive or not uniform, i.e.:

$$u_0 = ... = u_p < u_{p+1} \le ...u_j \le u_{j+1} \le ...u_{m-p-1} < u_{m-p} = ... = u_m$$

The knot vector is defined not uniform because there are knots which repeat themselves inside it. The aim of the repeated knots is to make NURBS curve pass through the first and last control point; moreover they make the extremities of the curve be tangent to control polygon and, at the same time, they define a curve segment.
The parametric equation of a p degree NURBS curve on the plane x – y is:

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} = \sum_{i=0}^{n} R_{i,p}(u) P_i = x(u)\vec{i} + y(u)\vec{j} \quad u \in [u_0, u_m] \qquad (1)$$

where u is the curve parameter defined in the range $[u_0, u_m]$ (the first element and the last one of knot vector), $R_{i,p}$ (u) is the p-degree rational function, while $N_{i,p}$ (u) is the p-degree B-Spline function calculated by the use of a recursive expression of lower degree functions.

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \qquad (2)$$

The B-Spline zero degree basic function $N_{i,0}$ (u) is a step function which has the following mathematical expression:

$$N_{i,0}(u) = \begin{cases} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases} \qquad (3)$$

The u_i values are elements of U knot vector.
Changing the u value from u_0 to u_m in the equation (1), it is possible to calculate the coordinates of all the points belonging to the curve.
The main advantages of NURBS parametric formulation can be summarized in the possibility of modifying locally the shape of the curve by changing the position of the control points P_i, the p degree, the w_i weights, the U knot vector.

3.2 Operating principle of NURBS path generator

The NURBS path generator has to transform instructions which describe the tool path in NURBS format and are reported in the part-program, into position commands for the Servo Control Loop System (loop control of the servo drive of the axis): this device, in its turn, at regular time intervals, transforms these commands into displacement ones for each servo-motor controlling motion of axes; displacement commands are electrical impulses whose cumulative number causes both the displacement and the rate by which tool moves forward along that axis at those time intervals. The goal is to set the motions of the controlled axes in order to move the tool along the path which, step by step, corresponds to the NURBS programmed one.
Therefore, NURBS interpolation is carried out following a temporal method which is, at the same time, incremental: the consequence is that the NURBS path is created as a series of displacements.
Considering $C(u_k)$ as the command position at the time (T_k), during the temporal increment, the tool is moving forward in order to reach, at the time T_{k+1}, the following command position $C(u_{k+1})$.

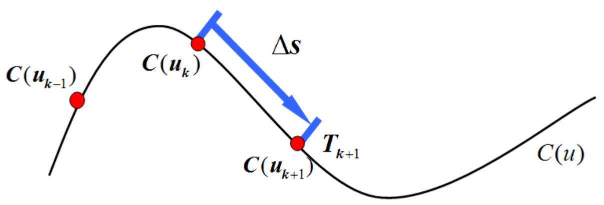


Fig. 1- NURBS interpolation

To set the coordinates of the points to interpolate, it can take advantage of the parametric expression of the curve which has to be created: this is possible by changing the value attributed to u parameter from u_0 to u_m (respectively the first element and the last one of the knot vector)

3.3 The NURBS path generator developed

The proposed NURBS path generator has been developed considering the following basic principles:

- A part program is given to the CNC: it contains the tool path described in the NURBS format of FANUC numerical control. The goal is to take advantage of a CAM software which is able to generate a part program containing the commands which describe the tool path in NURBS format used by numerical control mentioned before [11-12].
- It is possible to generate the tool path using the NURBS parametric formulation (it is employed the De Boor's algorithm to calculate the coordinates of the next positions which have to be implemented) [13-17].
- To generate the next command positions for the Servo Loop Control System (i.e. the next displacement commands) it takes advantage of linear interpolation function or the circular one (there are both linear interpolation functions and circular ones in any CNC). The indirect consequence is that the tool path will be approximated either by a series of linear segments or by a series of bi-arcs [3-4, 18-25].
- To reduce processing time which is necessary to create the tool path, after the acquiring of the command pack in which is programmed the NURBS curve, it happens automatically the subdivision of this one in segments characterized by a number of control points (<3*grade) of the NURBS (the subdivision is recursively carried out using the De Boor's subdivision algorithm). Then each segment of the curve of NURBS curve is singularly elaborated to generate the tool path.
- All calculations needed to define the tool path are carried out in pre-elaboration phase (offline). The goal is to reduce the time computational complexity during the real time work phase which will take place in real time [26-31].

Moreover the NURBS path generator has been developed in order to enable the user to select among three different options to generate the tool path, using a parameter.

- The first option approximates the path programmed with a tool path composed by a series of bi-arcs; so, the coordinates of the points interpolated by such bi-arcs are calculated setting constant increments of u parameter.
- The second one approximates the path programmed with a tool path composed by a series of bi-arcs, but the coordinates of the

ISBN: 978-960-474-402-2

points interpolated by such bi-arcs are calculated setting constant increments of displacement.

- The third one approximates the path programmed with a tool path composed by a series of linear segments but, as in the previous case, the coordinates of the points interpolated by such linear segments are calculated setting constant increments of displacement.

The three options of NURBS interpolation are respectively called:

a) *NURBS interpolation by constant increments of u parameter* - (NICU),
b) *NURBS interpolation by constant displacement increments and linear motion* - (NICL),
c) *NURBS interpolation by constant displacement increments and circular motion* - (NICC).

When the NURBS interpolation is carried out setting constant increments of u parameter, the issue is that the optimal dimension of Δ_u increment is unknown; moreover along the curve, Δ_u increments don't correspond to displacement increments (Δ_s).

Adopting this method, it is possible to find the following anomalies:

- If Δ_u is too small, the elaboration time will be inadequate;
- If Δ_u is too large, there will be an inadequate precision of the generated path in comparison with the programmed path;
- If Δ_s isn't an integer multiple of V*ΔT, the feed rate of the tool will be different from that one programmed. In fact, because of ΔT is constant, if the length of the bi-arc isn't an integer multiple of the factor V*ΔT, the last segment of the bi-arc (whose length is $\Delta s' \neq \Delta s$) will be interpolated with a different rate in comparison with that one programmed ($V' = \frac{\Delta s'}{\Delta T} \neq \frac{\Delta s}{\Delta T} = V$).

On the contrary, if the NURBS interpolation is carried out setting constant displacement increments, the dimension of Δ_u increment is set so that the displacement increment Δs is integer multiple of the product V*ΔT. Therefore the parameter $u_{k+1} = u_k + \Delta u$ is calculated by inverse ratio of length function.

Adopting this strategy, you get the following advantages:

- The feedrate V by which the tool moves from the position to in the time ΔT will coincide to that one programmed;
- Compared with the path programmed, the precision of that one generated will depend on the value of Δs. Given that Δs=V*ΔT, it is possible to get the standard of precision wanted adopting a suitable value of V.

Regarding the inverse ratio of the length function, its analytic expression is unknown and, for this reason, in the range characterized by each element of knot vector, its trend is approximated by a third degree polynomial.

$$u(l) = a + b(l - l_j) + c(l - l_j)^2 + d(l - l_j)^2(l - l_{j+1}) \qquad (4)$$

4 Simulation and Tests

The purpose of simulations and tests has been that one to confirm the ideas which are at the base of NURBS interpolation techniques proposed, to underline possible errors of the algorithm implemented, to make a comparison in order to evaluate the efficiency in terms of computational complexity (elaboration time), precision of the path covered compared with that one programmed and capability of following the feed rate programmed [32-33]. To complete the check, some milling manufacturing was performed by a machine tool, RM-MiniMill.

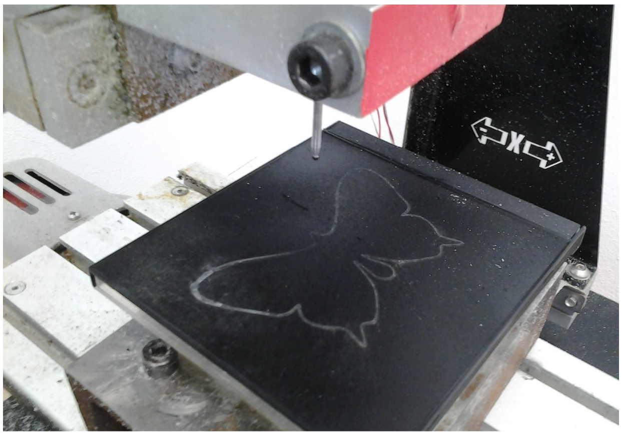


Fig. 2 - Curve Butterfly machining

Milling manufacturing is characterized by incisions, whose shape could be both analytic and arbitrary.

NURBS Order	Control Points	Current LinuxCNC	NICU	NICC	NICL
4	13	0.31	<0.01	0.12	0.1
5	32	1.94	0.01	0.3	0.29
5	50	7.72	0.04	0.65	0.54
4	119	112.2	0.05	0.65	0.63
4	231	884.9	0.11	1.19	1.12
6	589	∞	0.38	15.8	14.0

Table 1 - Processing times [s]

ISBN: 978-960-474-402-2

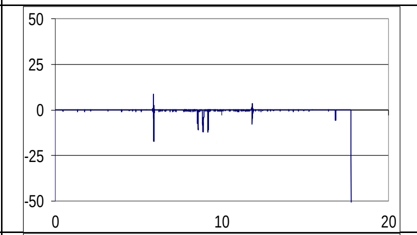
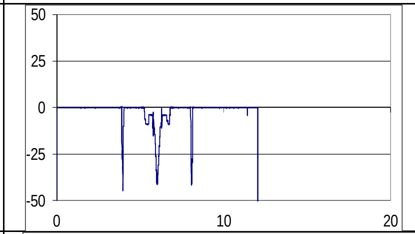
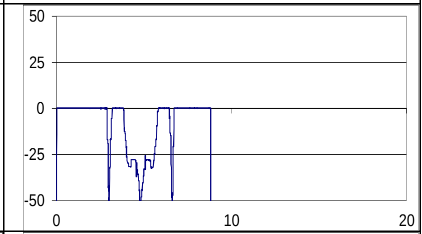
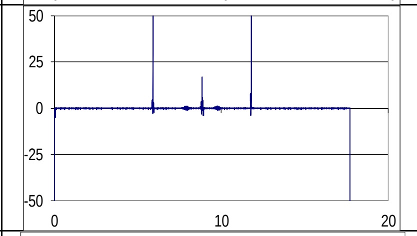
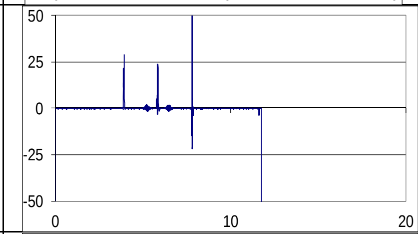
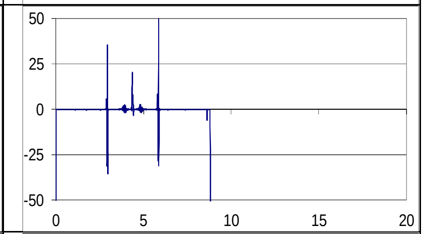
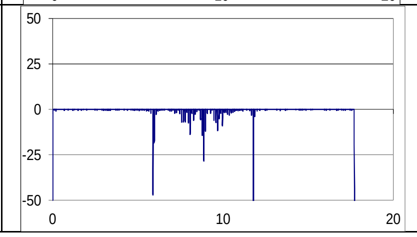
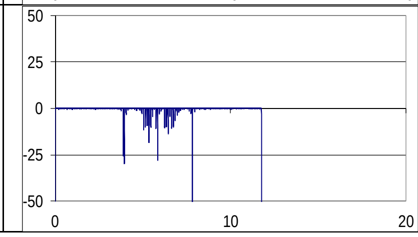
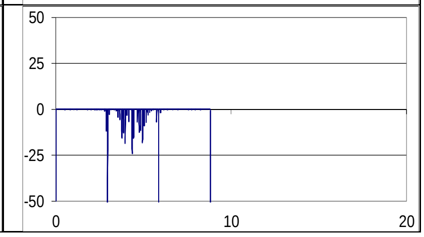
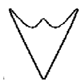
	V=200 mm/min	V=300 mm/min	V=400 mm/min
NICU			
NICC			
NICL			

Table 2 - Feedrate error (in %) for Trident curve 

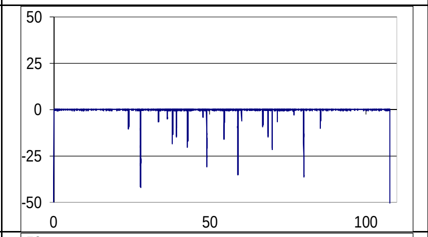
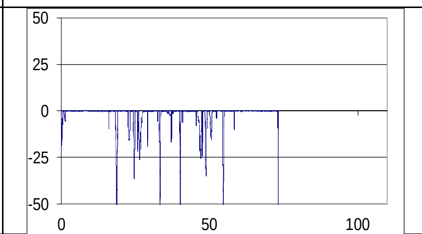
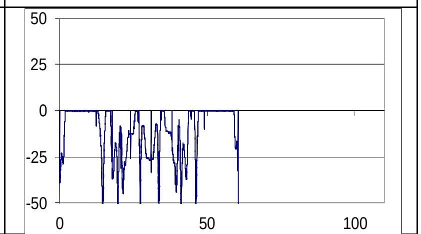
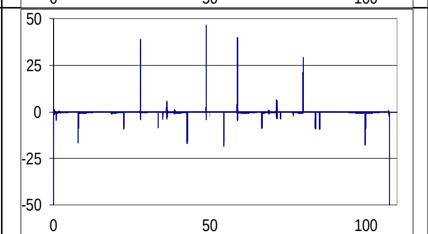
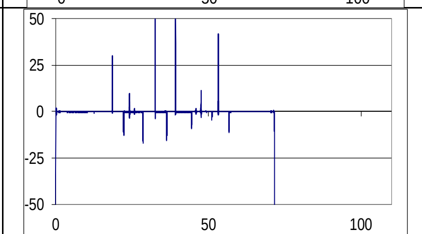
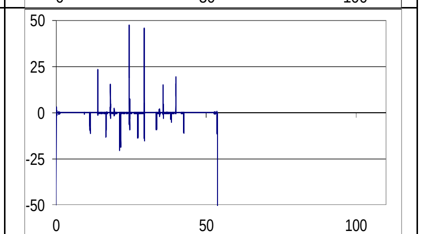
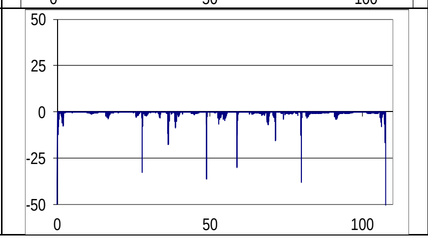
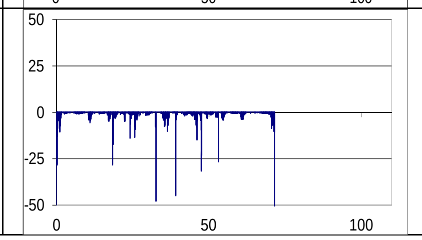
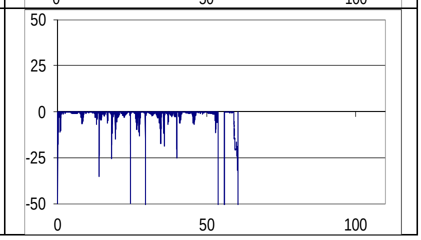
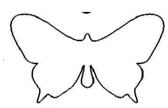
	V=200 mm/min	V=300 mm/min	V=400 mm/min
NICU			
NICC			
NICL			

Table 3 - Feedrate error (in %) for Butterfly curve 

In Tab. 1 the processing times needed to generate the tool path are shown, using a Personal Computer with a Mobile AMD Sempron Processor 3400+ and 2 GB RAM.

The NURBS interpolator has been evaluated with 2 test curves widely adopted in literature, called Trident [31] and Butterfly [26], with part program reported in Fig. 3 and Fig.4 respectively.

```
G21 G40 G54 G90 G94
G0 Z50
X0Y0
G96 S200 M3
G0  Z50
    X10 Y0
    Z1
G1 Z-1 F100
G6.2 X10 Y0 R1 K0 P4 F100
    X20 Y27.000 R1 K0
    X12 Y8.000 R1 K0
    X10.000 Y20.000 R1 K0
    X8.000 Y8.000 R1 K0.25
    X0.000 Y27.000 R1 K0.5
    X10.000 Y0.000 R1 K0.75
K1
K1
K1
K1

G1 X10.000 Y0.000 Z1
G0 X10.000 Y0.000 Z50
M2
```

Fig. 3 - Part program for Trident curve

The Tab. 2 and 3 report the trends of rate errors as percent value $\left(Err\% = \frac{V_{com} - V}{V_{com}} \right)$ with the increasing in feedrate commanded and with the change in the path programmed.

The result of the comparison underlines that whereas the NURBS interpolation method of the current LinuxCNC software version gets into crisis with curves having more than 100 control points, on the contrary all three NURBS interpolation methods proposed take elaboration times which are more than acceptable, even with NURBS curves having more than 500 control points.

In particular the NURBS interpolation method with constant increments of the u parameter is the choice considered more acceptable in all that applications when reducing elaboration times is the main need.

Moreover, if the NURBS interpolation is carried out setting constant displacement increments, the average error of feedrate is practically negligible both with the increasing in the feedrate commanded and with the change in the tool path programmed.

```
G21 G40 G54 G90 G94
G96 M3 S 200
G0 Z50
X0Y0

G0 Z50 (movimento per avvicinarsi alla po
(Curva acquisita N°=1)
 X54.493 Y52.139
 Z1
 G1 Z-1 F100

G6.2 X54.493 Y52.139 R1 K0 P5 F100
    X55.507 Y52.139 R1 K0
    X56.082 Y49.615 R1 K0
    X56.780 Y44.971 R1.2 K0
    X69.575 Y51.358 R1 K0
    X77.786 Y58.573 R1 K1
    X90.526 Y67.081 R1 K2
    X105.973 Y63.801 R1 K3
    X100.400 Y47.326 R1 K4
    X94.567 Y39.913 R1 K5
    X92.369 Y30.485 R1 K6
    X83.440 Y33.757 R2 K7
    X91.892 Y28.509 R1 K8
    X89.444 Y20.393 R1 K9
    X83.218 Y15.446 R5 K10
    X87.621 Y4.830 R3 K11
    X80.945 Y9.267 R1 K12
    X79.834 Y14.535 R1 K13
    X76.074 Y8.522 R1 K14
    X70.183 Y12.550 R1 K15
    X64.171 Y16.865 R1 K16
    X59.993 Y22.122 R1 K17
    X55.680 Y36.359 R1 K18
    X56.925 Y24.995 R1 K19
    X59.765 Y19.828 R1 K20
    X54.493 Y14.940 R1 K21
    X49.220 Y19.828 R1 K22
    X52.060 Y24.994 R1 K23
    X53.305 Y36.359 R1 K24
    X48.992 Y22.122 R1 K25
    X44.814 Y16.865 R1 K26
    X38.802 Y12.551 R1 K27
    X32.911 Y8.521 R1 K28
    X29.152 Y14.535 R1 K29
    X28.040 Y9.267 R1 K30
    X21.364 Y4.830 R3 K31
    X25.768 Y15.447 R5 K32
    X19.539 Y20.391 R1 K33
    X17.097 Y28.512 R1 K34
    X25.537 Y33.750 R2 K35
    X16.602 Y30.496 R1 K36
    X14.199 Y39.803 R1 K37
    X8.668 Y47.408 R1 K38
    X3.000 Y63.794 R1 K39
    X18.465 Y67.084 R1 K40
    X31.197 Y58.572 R1 K41
    X39.411 Y51.358 R1 K42
    X52.204 Y44.971 R1 K43
    X52.904 Y49.614 R1 K44
    X53.478 Y52.139 R1 K45
    X54.492 Y52.139 R1 K46
K47
K47
K47
K47
K47
G1 X54.492 Y52.139 Z1
G0 X54.492 Y52.139 Z50
M5
M2
```

Fig. 4 - Part program for Butterfly curve

On the contrary, if the interpolation is carried out setting constant increments of u parameter, the average error of feedrate can become not negligible

ISBN: 978-960-474-402-2

for high values of feedrate and, above all, if the tool path programmed is characterized by continuous and rapid changes of direction. This means an inadequate finishing of manufacturing surface.

5 Conclusion

By tests performed, you can underline advantages and disadvantages of the three NURBS interpolation methods proposed. The final result is that it doesn't exist a NURBS interpolation method which is better than other ones, but the choice about which one has to be adopted will depend by the following aspects:

- standard of precision by which you want to create the tool path;
- feedrate by which you want to perform the manufacturing;
- computing capacity of the PC.

On the basis of the results achieved, you can conclude that if the computing capabilities of the PC employed are limited, it is convenient to adopt a NURBS interpolation method with constant increments of u parameter; on the contrary, if you want to have high precision of the path covered in comparison with that one programmed and you want to manufacture using middle feedrate which is near to the value programmed, it will be convenient to adopt one of the two methods which carries out the interpolation setting constant increments of displacement. In particular, using feedrates upper than 300 mm/min, the NURBS interpolation with constant increments by circular motion seems to be more efficacious (above all if you want to avoid the segmentation of the tool path); instead, using feedrate lower than 300 mm/min, it is convenient the NURBS interpolation with constant increments of displacement by linear motion.

For this reasons, the choice about which NURBS interpolation method has to be employed to generate the tool path depends by the user who will choose that one which will satisfy his needs.

References:

[1] Lei, W.T., Sung, M.P., Lin, L.Y., Huang, J.J., Fast Real-Time NURBS Path Interpolation for CNC Machine Tools, *International Journal of Machine Tools and Manufacture*, Vol. 47, Iss. 10, 2007, pp. 1530-1541.

[2] Lei, W.T., Wang, S.B., Robust Real-Time NURBS Path Interpolators, *International Journal of Machine Tools and Manufacture*, Vol. 49, Iss. 7-8, 2009, pp 625-633.

[3] Leto, M., Licari, R., Lo Valvo E., Piacentini M., CAD/CAM Integration for NURBS path interpolation on PC based real.time numerical control, *Proceedings of AMST 2008 Conference*, 2008, pp. 223-233.

[4] Piegl, L., Tiller, W., Biarc Approximation of NURBS Curves, *Computer-Aided Design*, Vol. 34, Iss. 11, 2002, pp. 807-814.

[5] Hong-Tzong, Y., Jun-Bin, W., Chien-Yu, H., Chih-Hua, Y., PC-based Controller with Real-time Look-ahead NURBS Interpolator, V, *Computer-Aided Design*, Vol. 4/1-4, (2007), pp. 331-340.

[6] Piegl, L., Tiller, W., *The NURBS Book*, Springer Verlag, 1997.

[7] Bahr, B., Xiao, X., Krishnan, K., A Real-time Scheme of Cubic Parametric Curve Interpolations for CNC Systems, *Computers in Industry*, Vol. 45, Iss. 3, 2001, pp. 309-317.

[8] Cheng. M.-Y., Tsai, M.C., Kuo, J.C., Real-time NURBS Command Generators for CNC Servo Controllers", *Journal of Machine Tools and Manufacture*, Vol. 42, Iss. 7, 2002, pp. 801-813.

[9] Liu, Q.J., Yue, J.H., Wang, Y.F., Dong, J.C., Wang, T.Y., A Parametric Curve Interpolation Algorithm for High Speed Machining, *Key Engineering Materials*, Vol. 458, Iss. 35, 2010, pp. 35-41.

[10] LinuxCNC, www.linuxcnc.org

[11] Jung, Y. H., Lee, D.W., Kim, J. S., Mok, H. S., NC Post-Processor for 5-axis Milling Machine of Table-rotation/tilting type, *Journal of Materials Processing Technology*, Vol. 130-131, 2002, pp. 641-646.

[12] Koren, Y., Lin, R.S., Five-Axis Surface Interpolators, *Annals of CIRP*, Vol. 44, No. 1, 1995, pp. 379-382.

[13] Liu, X., Ahmad, F., Yamazaki, K., Mori, M., Adaptive Interpolation Scheme for NURBS Curves With the Integration of Machining Dynamics, *International Journal of Machine Tools and Manufacture*, Vol. 45, Iss. 4-5, 2004, pp. 433-444.

[14] Wang, F.C., Yang, D.C.H., Nearly Arc-Length Parameterized Quintic-Spline Interpolation for Precision Machine, *Computer-Aided Design*, Vol. 25, No. 5, 1993, pp. 281-288.

[15] Erkorkmaz, K., Altintas, Y., Quintic Spline Interpolation with Minimal Feed Fluctuations, *Journal of Manufacturing Science and Engineering*, Vol. 127, No. 2, 2005, pp. 339-349.

[16] Ho, M.C., Hwang, Y.R., Hu, C.H., Five-Axis Tool Orientation Smoothing Using Quaternion Interpolation Algorithm, *International Journal of Machine Tools and Manufacture*, Vol. 43, Iss. 12, 2003, pp. 1259-1267.

[17] Yeh, S.S., Hsu, P.L., The Speed-Controlled Interpolator for Machining Parametric Curves, *Computer-Aided Design*, Vol. 31, Iss. 5, 1999, pp. 349-357.

[18] Koren, Y., Lo, C.C., Shpitalni, M., CNC interpolators: Algorithms and Analysis, *ASME Production Engineering Division, Proceedings of the 1993 ASME Winter Annual Meeting*, Vol. 64, 1993, pp. 83-92.

[19] Koren, Y. Lin, R. S., Real-Time Five Axis Interpolator for Machining Ruled Surfaces, *Proceedings of the 1994 International Mechanical Engineering Congress and Exposition*, Vol. 55, 1994, pp. 951-959.

[20] Lai, Y.L., Tool-Path Generation of Planar NURBS Curve, *Robotics and Computer-Integrated Manufacturing*, Vol. 26, Iss. 5, 2010, pp. 471-482.

[21] Yeh, S.S., Hsu, P.L., Adaptive-Feedrate Interpolation for Parametric Curves with a Confined Chord Error, *Computer-Aided Design*, Vol. 34, No. 3, 2001, pp. 229-237.

[22] Zhang, Q.G., Greenway, R.B., Development and Implementation of a NURBS Curve Motion Interpolation, *Robotics and Computer-Integrated Manufacturing*, Vol. 14, No. 1, 1998, pp. 27-36.

[23] Lin, H., Wang, G., Dong, C., Constructing Iterative Non-Uniform B-Spline Curve and Surface to Fit Data Points, *Science in China Series F: Information Sciences*, Vol. 47, No. 3, 2003, pp. 315-331.

[24] Heng, M., Erkorkmaz, K., Design of a NURBS Interpolator with Minimal Feed Fluctuations and Continuous Feed Modulation Capability, *International Journal of Machine Tools and Manufacture*, Vol. 50, Iss. 3, 2010, pp. 281-293.

[25] Lo, C.C., A New Approach to CNC Tool Path Generation, *Computer-Aided Design*, Vol. 30, Iss. 8, 1998, pp. 649-655.

[26] Lei, W.T., Sung, M.P., Lin, L.Y., Huang, J.J., Fast real-time NURBS path interpolation for CNC machine tools, *International Journal of Machine Tools and Manufacture*, Vol. 47/10, 2007, pp. 2246-2262.

[27] Park, J., Nam, S., Yang, M., Development of a Real-Time Trajectory Generator for NURBS Interpolation Based On The Two-Stage Interpolation Method, *Int J Adv Manuf Technol*.,Vol. 26, No. 4, 2005, pp. 359-365.

[28] Zhiming, X., Jincheng, C., Zhengjin, F., Performance Evaluation of a Real-Time Interpolation Algorithm for NURBS Curves, *International Journal of Advanced Manufacturing Technology*, Vol. 20, No. 4, 2002, pp. 270-276.

[29] Tsai, M.C., Cheng, C.W., A Real-Time Predictor-Corrector Interpolator for CNC Machining, *Journal of Manufacturing Science and Engineering*, Vol. 125, Iss. 3, 2003, pp. 449-460.

[30] Cheng, C.W., Tsai, M.C., Real-Time Variable Feedrate NURBS Curve Interpolator for CNC Machining, *Journal of Advanced Manufacturing Technology*, Vol. 23, No. 11-12, 2004, pp. 865-873.

[31] Lin, M.T., Tsai, M.S., Yau, H.T., Development of a dynamic-based NURBS Interpolator with Real-Time Look-Ahead Algorithm", *International Journal of Machine Tools and Manufacture*, Vol. 47, Iss. 15, 2007, pp. 2246-2262.

[32] Erkorkmaz, K., Heng, M., A Heuristic Feedrate Optimization Strategy for NURBS Toolpaths, *CIRP Annuals - Manufacturing Technology*, Vol. 57, Iss. 1, 2008, pp. 407-410.

[33] Tsai, Y.F., Farouki, R.T., Feldman, B., Performance Analysis of CNC Interpolators for Time-Dependent Feedrates along PH Curves, *Computer Aided Geometric Design*, Vol. 18, No. 3, 2001, pp. 245-265.

ISBN: 978-960-474-402-2