View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Revistas académicas Universidad EAFIT

Low Level Direct Interpolation for Parametric Curves

Oscar ▪ Ruiz
Adriana ▪ Martínez
Elizabeth ▪ Rendón

This article presents an algorithm for the direct interpolation of parametric planar curves $C(u)$ with a CNC machine. It expresses parametric planar curves as sequences of machine tool axes discrete movements of BLU size. Therefore, the curve $C(u)$ is directly approximated by the pulse trains, hence eliminating one source of the machining errors.

Keywords: Direct Numerical Control, Parametric Curve Interpolation.

Ruiz Oscar (Professor). Oruiz@Sigma.Eafit.Edu.Co. Martínez Adriana (Research Assistant). Rendón Elizabeth (Student Assistant). Center For Interdisciplinary Research (Cii) In Cad/Cam/Cg. Eafit University. Medellín, Colombia.

1. INTRODUCTION

In CNC the parametric curve or surface actually machined differs from the ideal mathematical entity. Generation of tool locus for machining of parametric planar curves usually presents two levels of approximation. The first level is implied by the approximation of the parametric curve into a sequence of lower order curves (lines, circular arcs, parabolic segments), which are the usual primitives included in the dictionary of a CNC machine. The second level of approximation lies on the fact that these primitives are actually mathematical abstractions which are expressed as a sequence of elementary, discrete movements of the machine tool axes. These two approximations eventually add up, increasing the machining errors. According to these facts, this investigation presents an algorithm that directly expresses parametric planar curves as sequences of axis movements. This problem can be abstracted as one of approaching a continuous curve in a discrete space. By avoiding intermediate steps, the errors inherent to both approximations are reduced, therefore producing the best possible curve given a CNC precision range (or BLU [(1)]). The objective of this investigation is to explore the expression of parametric planar curves directly into discrete movements of BLU size of the machining tool axis. The net final goal is to introduce a new *G* primitive, namely parametric 2D curve, to enrich the vocabulary of the *G* code in a CNC machine tool.

---

(1) BLU: Basic Lenght Unit, is the axis resolution in a CNC machine tool.

This investigation presents an algorithm that directly expresses parametric planar curves as sequences of axis movements. This problem can be abstracted as one of approaching a continuous curve in a discrete space.

The example presented interpolates bezier curves, although the algorithm is addressed to work with any type of parametric curve. Quantification of the merits of the algorithm is attempted. Section 2 of the article surveys the existing literature on this topic. Section 3 presents the algorithm and its different characteristics. Section 4 discusses the results obtained in the examples. Section 5 draws the general conclusions of the article, and states possible advances in the topic presented.

2. LITERATURE SURVEY

The general problem of approaching a continuous planar curve by a series of discrete positions varies according to whether the curve is expressed in: (i) implicit or (ii) parametric form.

(i) For implicit curves there is an immediate antecedent in the algorithms used to display geometric primitives on discrete spaces. This is the case of Bresenham´s and Midpoint algorithms to draw straight lines and circles on raster graphics devices as sequences of screen pixels (JOY, FOL.91, MOR.85). These algorithms make use of symmetries of the primitive to reduce calculations and accelerate the determination of the pixels to highlight. For example, in a circle, the complete set of (x,y) pixel positions forming the periphery can be inferred from the pixels of one octant. Another optimization of the algorithms regards the statement of formulae that only make use of integer arithmetic. Clearly, for general planar curves, the special conditions that favor Bresenham´s and Midpoint algorithms are not present: in general, there exists neither symmetry nor formulae manipulation for integer arithmetic. In addition to software-based procedures (Bresenham´s and Midpoint), hardware ones are available which perform the interpolation of an implicit planar function by using arrays of DDAs (YOR.76). In summary this scheme obtains the goal function by integrating its derivatives by hardware. A similar software approach, called pattern recognition tracks a continuous planar implicit curve by staircase displacements on a discrete grid (KAP.91).

(ii) In processing line drawings (image consisting of line and /or curve segments which need no to be connected), (FRE.69) one approaches the given mathematical curve using the sequence of the grid discrete points. The curve points on the grid are chosen on the basis of how the line drawing intersect the grid line between two adjacent mesh nodes. Of the two mesh nodes, the one closer to the intersection point is chosen and used for the low level direct interpolation.

The general problem of approaching a continuous planar curve by a series of discrete positions varies according to whether the curve is expressed in: (i) implicit or (ii) parametric form.

A large obstacle to apply the implicit curve schemes mentioned above is the fact that parametric planar curves are not always convertible to implicit form, and therefore the techniques are not always applicable. This investigation proposes a method to approach a parametric curve by choosing a sequence of vertex on a grid in such a way that the error of approximation is minimized among them. The algorithm correctly approaches bezier as well as spline curves, and its strategy is indifferent to the type of curve used. It should be noticed that although it reminds Touissaint’s work (TOU), it focuses on BLU grid space rather than on screen pixels.

Current CNC machines use the original curve to create new curves based on primitive curves such as circles, lines, ellipses and parabolas. The most modern machines interpolate the original curves using not only primitives but also nurbs. The CNC machine use those interpolated data to create a sequence of pulses to commard the machine's motor(s) (BEA.97, BEA.96).

To achieve this, is necessary to use another interpolation method, this generates a double interpolation and in consequence a double error. The total error could be estimated as the sum of these errors. Using the method described in this paper, only one interpolation is necessary from the original curve to the discrete movements in the machine's axis reducing the error to BLU/SIN45^{o}. (SEE FIG.10).

3. METHODOLOGY

Given a planar parametric curve *C(u)*, and a discretization of the 2D space (a grid of size BLU), the goal is to produce the sequence of grid intersection points that approaches *C(u)*. The proposed algorithm obtains the best approach for *C(u)* given a grid *G*. This is so, because the algorithm chooses in each step the grid intersection that is closest to the curve *C(u)*. Therefore, the error of approximation to the curve is only the inherent to the finite resolution of the machine tool. This is in contrast with other approaches which add the error inherent to the approximation of the curve *C(u)* by other set of primitives (lines, circles and parabolas).

The algorithm used for the BEZIER interpolation basically includes the following functions:

```
function interp_segment (  Curve C, Grid G,
                    List X, List Y,
                    List Pulses_X, List Pulses_Y)

     initialize(C,G,du);
     X=[]
     Y=[]
     Pulses_X=[]
     Pulses_Y=[]
     (xi, yi) = round(C(0),G)
     u = 0
     while (u <= 1.0 ) do
     {Inv: (xi,yi) = last grid intersection found}
                u = u + du
                (xt, yt)=C(u)
                if too_large_jump((xi,yi),G,(xt,yt))
                     u=u-du
                     u=du/2
                elseif hits_grid((xi,yi),G,(xt,yt))
                     (xt,yt) = round((xt,yt),G)
                     Pulses_X = [Pulses_X, sign_with_error(xt-xi, ERROR_DIST)]
                     Pulses_Y = [Pulses_Y, sign_with_error(yt-yi, ERROR_DIST)]
                     xi = xt
                     yi = yt
                     X=[X, xi]
                     Y=[Y, yi]
                fi
     end{while}
end{function}
```

```
function boolean too_large_jump((xi,yi), G, (xt,yt))
    if ((|xt- > G) or (|yt-yi| > G) and
     ( not equal(xt,xi,ERROR_DIST)) and
     ( not equal(yt,yi,ERROR_DIST))
    )
            return( TRUE );
    else
            return( FALSE );
    fi
end{function}
function hits_grid((xi,yi),G,(xt,yt))
    if(      (equal(|xt-xi|,G,ERROR_DIST)) or
             (equal(|yt-yi|,G,ERROR_DIST))
    )
            return( TRUE );
    else
            return( FALSE );
    fi
end{function}
```

OBSERVATIONS

1. The function sign_with_error(v, ERROR_DIST) returns

   0, if equal (v, 0, ERROR_DIST)

   1, if v > 0

   -1, if v < 0

   The sequences so build, *Pulses_X* and *Pulses_Y*, are the inputs for the step motors driving the axes *X* and *Y* of the CNC machine tool.

2. The function initialize (*C,G,du*) assigns a starting value to *du,* based on the length of the curve *C* and the grid size *G*. The starting value clearly depends on the control points of the curve *G*, and is defined to permit several iterations on the parameter *u* within each grid interval.

FIGURE 1

Curve and Interpolation. BLU size=0.02 mm

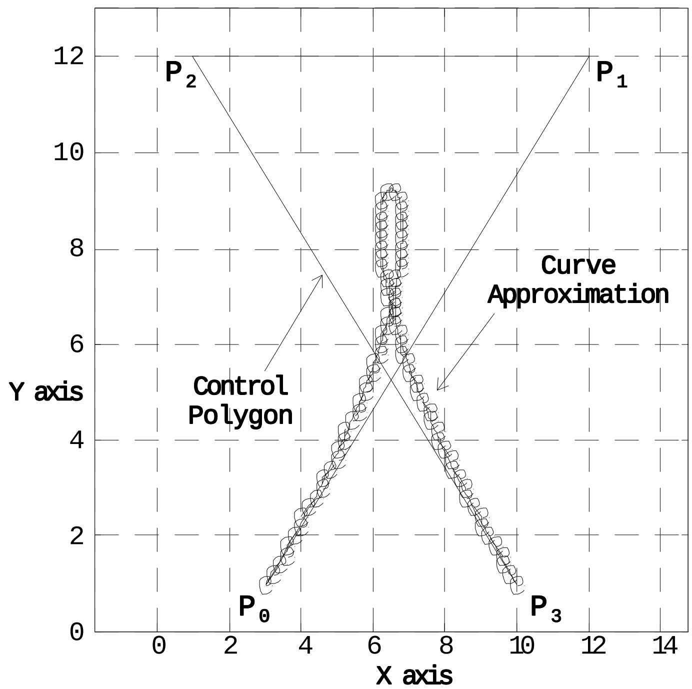


FIGURE 2

SEQUENCE OF PULSES X AXIS MOTOR FOR CURVE IN FIGURE 1

Movement of X axis motor

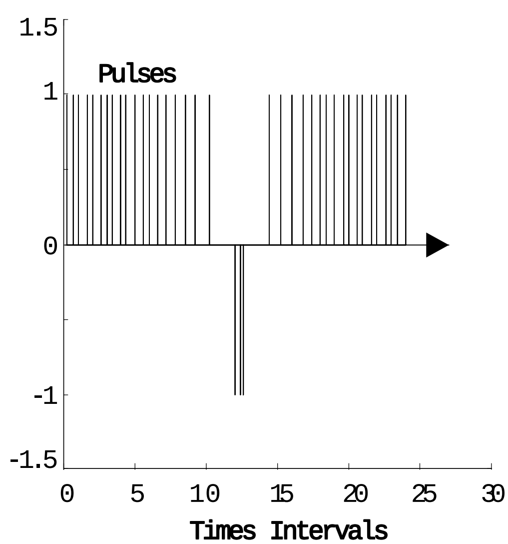
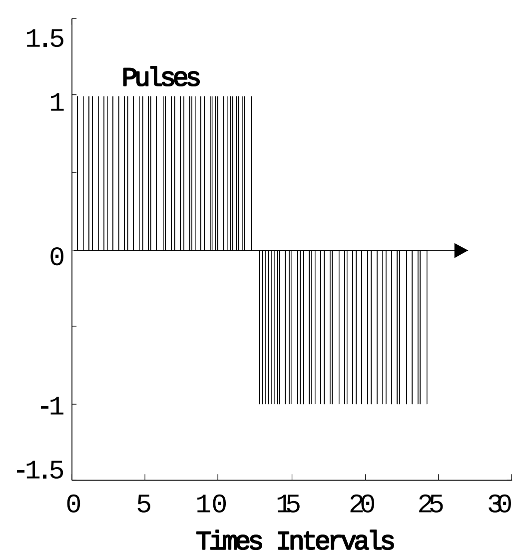


FIGURE 3

SEQUENCE OF PULSES Y AXIS MOTOR FOR CURVE IN FIGURE 1

Movement of Y axis motor



3. The function *too_large_jump*((*xi,yi*), *G*, (*xt,yt*)) examines if *xt* (or *yt*) differs from *xi* (or *yi*) by a distance strictly larger than *G* (line 12). In that case the parameter *du* is decreased, given the fact that it represents a step too large in the parametric space.

4. The function *hits_grid*((*xi,yi*),*G*,(*xt,yt*)) determines whether *xt* (or *yt*) lies away from *xi* (or *yi*) approximately by a distance *G*.

5. The line (*xt,yt) = round((xt,yt),G*) chooses the grid intersection closest to the point *C(u)*. This function is called when *C(u)* lies on a horizontal grid line (*y=yt*) or a vertical one (*x=xt*).

The main strategy of the algorithm is to detect the places in which the curve *C(u)* crosses the vertical (*x=k*g*) or horizontal (*y=k*g*) grid lines (see line 15). If it is detected that the curve crosses say a horizontal grid line (*yt=k*g*, for some integer *k*), the other component (*xt*) of the point is attracted to the closest vertical grid line. A converse situation is produced if the line crosses a vertical grid line.

Once the grid intersection is recorded, the pulses required to drive the machine from the previous intersection to the current one are determined (See 17,18). Each entry of the pulse train may take one of tree values: {no pulse, pulse forward, pulse backward} (0,1,-1). With this convention, the curve *C(u)* is represented by two sequences:

for X axis: 1,1,0,0,0,0,-1,-1,-1....

for Y axis: 0,0,-1,-1,-1,0,-1,0,0,0,....

4. RESULTS

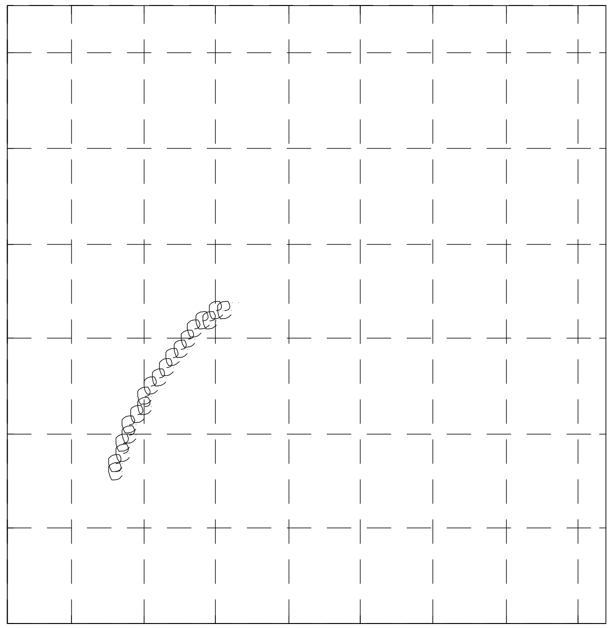
The algorithm presented above was tested with three Bezier curves. Figures 1, 4 and 7 present the mathematic versions, as well as the approximations resulting of the algorithm explained. Figures 2, 3, 5, 6, 8, and 9 present the corresponding pulse trains to machine the curves with a CNC machine of 0.02 mm BLU. The maximum error is found when the curve approaches a grid vertex while the algorithm chooses its diagonal one as closest to the curve. This could happen because the intersection of the curve with grid boundaries (See Figure 10) would produce grid intersection (*xk, yk*) in the *k* iteration.

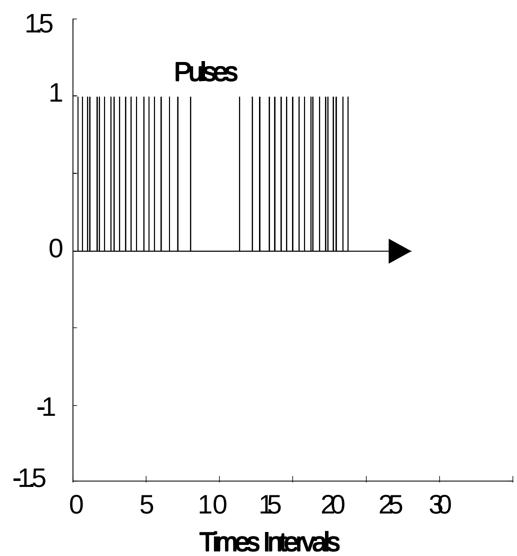
5. CONCLUSIONS

The present article has presented an algorithm for the direct interpolation of parametric curves with a CNC machine. The exact nature of the curves is not relevant to the algorithm given the fact that its mathematical form does not explicitly appear in it. The algorithm presented is the best approximation that a parametric curve may have with a grid of given BLU value. This is so because it eliminates the approximation of the *C(u)* curve with given primitives (lines, circles, etc.) which have to be, in turn, approached by the pulse trains driving the axes of the CNC machine. In this algorithm the curve *C(u)* is directly approximated by the pulse trains, therefore eliminating one source of the machining errors.

FIGURE 4

Curve and Interpolation. BLU size=0.02 mm





Low Level Direct Interpolation for Parametric Curves

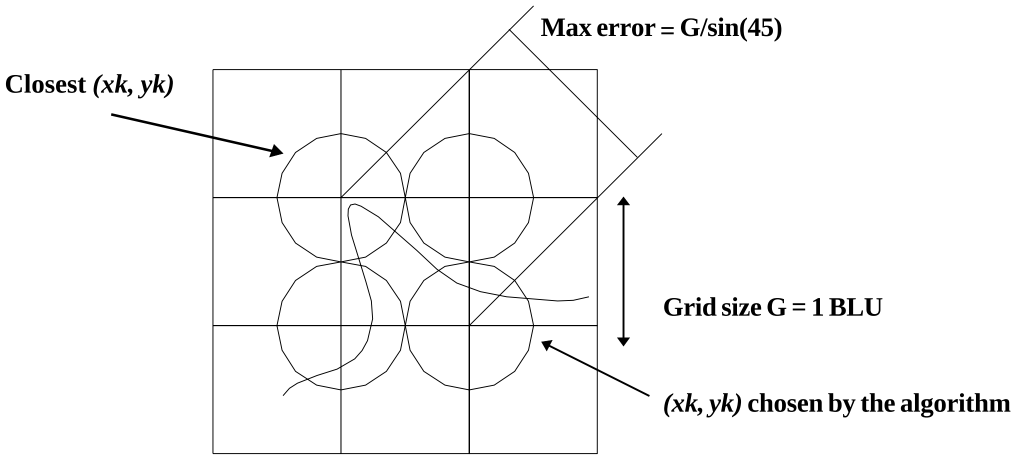
Oscar Ruiz
Adriana Martínez
Elizabeth Rendón

This article presents an algorithm for the direct interpolation of parametric planar curves *C(u)* with a CNC machine. It expresses parametric planar curves as sequences of machine tool axes discrete movements of BLU size. Therefore, the curve *C(u)* is directly approximated by the pulse trains, hence eliminating one source of the machining errors.

Keywords: Direct Numerical Control, Parametric Curve Interpolation.

Ruiz Oscar (Professor). Oruiz@Sigma.Eafit.Edu.Co. Martínez Adriana (Research Assistant). Rendón Elizabeth (Student Assistant). Center For Interdisciplinary Research (Cii) In Cad/Cam/Cg. Eafit University. Medellín, Colombia.

FIGURE 10
ESTIMATION OF MAXIMAL DEVIATION BETWEEN CURVE *C(u)* AND CHOSEN INTERSECTION (x_i, y_i)



REFERENCES

[WON.95] Wong Y.S., Loh, H.T, Teng, C.S. (1995). An optimal PieceWise Curve-Fitting Approach for CNC Machining, Transactions of NAMRI/SME, Vol 23, pp. 157-162.

[FRE.74] Freeman, H. (1974). Computer processing of line-drawing images, Computing Surveys, Vol. 6, No. 1, pp. 58-97.

[FRE.69] Freeman, H., Glass, J.M. (1969). On the quantization of line-drawing data, IEEE Transactions on System Science and Cyber-netics, Vol. SSC-5, pp. 70-79.

[TOU] Toussaint, G.T. Processing line drawings, pp. 48-62.

[KAP.91] Kapoor, S. (1991).Internal notes. University of Illinois, Department of Mechanical and Industrial Engineering.

[YOR.76] Yoram, K. (1976). Computer Control of Manufacturing Systems", Mc Graw-Hill, pp.1-25.

[JOY] Joy, K.I. Bresenham´s Algorithm, On-line Computer Graphics Notes, Department of Computer Science. University of California, Davis. pp.1-15.

[FOL.91] Foley, J., Van Dam, A., Feiner, S., Hughes, J. (1991). Computer Graphics: Principles and Practice", Addison Wesley, pp. 73-78.

[HAN.96] Wenz, H.J. (1996). Interpolation of Curve Data by Blended Generalized Circles, Computer Aided Geometric Design, Vol. 13, Issue 8, pp. 673-680.

[FAR.95] Farin, G., Jung, D. (1995). Linear Precision of Rational Bezier Curves, Computer Aided Geometric Design, Vol. 12, Issue 4, pp. 431-433.

[MOR.85] Mortenson, M. (1985).Geometric Modeling. New York, John Wiley and Sons.

[BER.71] Bergren, C. A. (1971). Simple Algorithm for Circular Interpolation, Control Engineering, pp. 57-59.

[BEA.97] Beard, T. (1997). Interpolating Curves, Modern Machine Shop, Vol. 70, No. 5, pp. 60-67.

[BEA.96] Beard, T. (1996). Machining in Circles, Modern Machine Shop, Vol. 69, No. 2, pp. 52-61.