Dynamic Bézier Curves: New Findings on Reparameterization by Arc length

Norbert L. *Brodtmann*[1], Daniel Schilberg[2]

[1] CNC + RoBo-mac
33428 Marienfeld – Harsewinkel, Germany

[2] University of Applied Sciences Bochum
44801 Bochum, Germany

ABSTRACT

The Bézier curve is a special curve developed in the 1960s, which is not to be generated "directly" with classical mathematics, therefore it is also called "free-form curve". The Bézier mathematics makes it possible to generate a lot of different curves with simple parameterization. Under dynamic aspects of the CNC process, the Bézier curve developed for static application profiles leads to Shape-dependent fluctuations of the cutting speed in orders of magnitude from 1:3 to 1:10 - and above. In Analysis, the problem of leveling different support point distances - which is causal here -, is studied under the term Parameterization by Arc length. For Bézier curves, however, a generally valid solution has not yet been published due to mathematical preconditions. Based on this background, a generally valid approximation algorithm for the *Reparameterization of Bézier curves according to Arc length* and their Dynamic use on CNC machines, was developed.

Keywords: Reparameterization · leveling dynamic fluctuations · CAD CAM · CNC · Bézier · de Casteljau · step size · support point distance.

FOREWORD AND STATE OF THE ART

Due to the limited space of this paper, only a summary excerpt from more extensive research can be given here, which is limited to Cubic Bézier curves. Knowledge of the structure and mathematics of Bézier curves must be assumed. Details are to be found at: Fetouaki and Skopin (2009), Oberle (2013), Babovsky and Neundorf (2011).

Both terms "step size"- which is more common for Bézier - and "support point distance" in focus of CNC path are used here, however have almost the same meaning.

The "normal" Bézier curve of any class (exponent) is a curve parameterized in the mostly uniform grid "t".

- From a static point of view, a Bézier curve is a locus curve with special possibilities for forming shapes. This was the intention of its development. If shaping is the goal of the Bézier process - as in the case of a car body, for example - the locus curve calculated for t = 0 to t = 1 is the desired result; - "t" is to be evaluated as a dimensionless parameter.
- Dynamically, the Bézier curve has some peculiarities: If "t" is considered as time, then - even for symmetrical Bézier curves and the same time interval - different step sizes of the locus curve will result due to *inhomogeneous parameterization*.

Dynamic Bézier curves

Bézier curves are a standard feature of CAD programs. In CAM (Computer Aided Manufacturing) their use is much less pronounced, as an Internet market research shows. The background is probably that Cartesian xy-support points generated by means of "normal" Bézier curves are mostly unsuitable for further processing on CNC machine tools.

- CNC programs calculate the feed rate of the positioning axes from the distance of the path support points.
- The Bézier locus curve has unequal construction point widths due to inhomogeneous parameterization, which influences the operating speed of CNC machines.

In Analysis, the problem of leveling different support point distances is studied under the term *Parameterization by Arc length*. The preconditions and limits of this are shown in the lecture scripts of FMI and Roth (2013/14). The classical curve discussion requires:

- that the derivation of the function must not become "0" (zero), i.e. a strictly monotonously increasing or decreasing function. Bernstein polynomials of the order no. B_1 and B_2 show in each case a high point, i.e. 1st derivative (velocity) = 0.
- Prerequisite of the parametrizability of curves is furthermore that they must be "smooth", i.e. continuously differentiable as often as desired. This is not the case with polynomials!

The here presented

New Findings on Reparameterization of Bézier-Curves levels the spacing of the path support points and dampens dynamic velocity variation of the original (by "t=const" parameterized) Bézier functions to a range of 10^{-2} to 10^{-3} of their original value.

CUBIC BÉZIER CURVES

The non-parameterized "normal" Bézier curve runs through path sections in the time interval t = 0 to t = 1 in equal time steps, but with different path lengths (Fig. 2 01a),
- which does not affect static applications.

Reparameterization, Numerical Approximation Algorithm

For the *Parameter transformation by arc length* (Fig. 2 01b) it is useful to define the transferred partial step times with $\tau = 0$ to $\tau = 1$.

- From the non-parameterized Bézier locus curve, the geometric distances of the xy construction points are first determined at step size $t_{(const)} = 1/n$ (per Pythagoras).
- For each time step ($t_{(const)}$) this results in a individual different length of each step
  - whose sum corresponds on the one hand approximately to the total length of the Bézier curve,
  - on the other hand indicates the "individually normal" distance between supporting points at t = const.
- Time step $$t_{(const)} = \frac{1}{n}$$ [2_01]

  Step size (t = const). $$l_{Secant} = P_n - P_{n-1}$$ [2_02]

  Curve length $$l_{Curve} = \sum_{t=0}^{n} P_n - P_{n-1}$$ [2_03]

- For normalization, each individual step size is divided by the curve segment's total length [2_02] / [2_03]. This normalized value is at the same time the individual deviation factor compared to a levelled partial step length.
- A division of t = const. [2_01] by this deviation factor or reciprocal multiplication leads to: $$\tau_{(transf)} = t_{(const)} \cdot \frac{l_{Curve}}{n \cdot l_{Secant}}$$ [2_04] resp.
- Transferred time increment $$\boxed{\tau_n = \frac{l_{Curve}}{n^2 \cdot l_{Secant}}}$$ [2_05]
- The time step sizes $\sum_{\tau=0}^{n} \tau_n$ do not add up precisely to 1 - depending on the resolution "n" and the curve's shape (Pythagoras rounding error).

- The transferred single time step sizes are therefore to be normalized by the factor resulting from their sum.

Cascading by feedback-loop

If in a feedback loop the results from [2_04] are used again as starting value instead of $t_{(const)}$ in [2_04], then the described algorithm levels fluctuations (curve-dependent) to an order of magnitude of the 1- to 2-digit per mille range.

Based on the transformed time step sizes [2_04] respectively [2_05], by repeated feedback new Bézier xy construction points become computable, leading to an approximate *Reparameterization by arc length* (cf. dots spacing Fig. 2_01a vs. 2_01b).

Reparameterization, Bézier Spline

A cascading of several Bézier curve segments to a spline with smooth connection points requires a gapless connection (P_0 of the following curve = P_3 of the preceding one) as well as tangent equality of both curves in the transition point.

- If the absolute arc length of the segments is equal, acceleration-free transitions result, since each segment is parameterized to equal support point widths in the interval t=0 to t=1. The Bézier circle of 120° segments is a good example of this.
- If the segments are of different arc lengths, the resolution "n" must be chosen proportional to the arc lengths, i.e. interval division for t=0 to t=1, must be adapted.

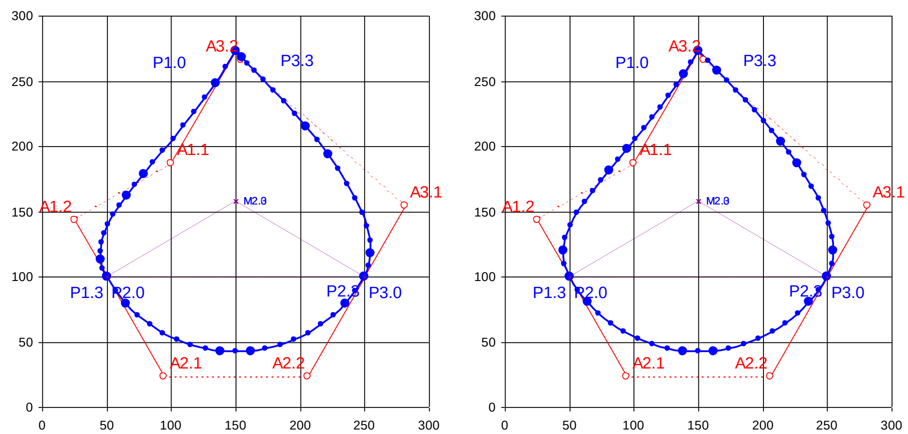

Fig. 2_01a *Orig. inhomog. Bézier step size* 2_01b *Steps levelled by reparameterization*

\- - Curves 1 (left) asymmetric with inflection point

\- Curves 2 (bottom) Circle segment

\- Curves 3 (right) asymmetric without inflection point

The thick dots each mark a step size "double spacing" in the *t*-range 0,0 / 0,5 / 1,0

Curve 1: Step's fluctuation 1: 2.271 levelled fluctuation 1: 1.013

Curve 2: Step's fluctuation 1: 1.135 levelled fluctuation 1: 1.00045

Curve 3: Step's fluctuation 1: 11.108 levelled fluctuation 1: 1.055

Derivative & Velocity

The considerations of derivative and velocity have a general character. The 1st derivative of a locus is commonly considered to be velocity. This is the case, if the locus curve was logged as a path-time diagram; however, it is not generally valid: The 1st derivative brings e.g. for the low point of a parabola the gradient value "0" (zero), which does not have to correspond to the path speed of a vehicle. If this parabola would be passed on a roller coaster, the speed would be highest in the low point!

- Basically applies:

> The speed at which a path curve is traversed,
> - is not dependent on the shape of the path curve or its derivation,
> - but on the vehicle or its drive.

A parameter transformation does not affect the graph of the locus curve or its derivative; it shifts the xy construction points of the graph "on the graph". The inverse makes clear:

If equal support point distances are traversed at equal time intervals, the velocity between the support points is the same.

Bézier circle

From a CNC point of view, the circle approximation, in particular the reparameterization according to arc length is rather academic in nature.

The problem of segment cascading is nevertheless explained from a dynamic point of view using the example of the Bézier circle, since this can be reproduced without any problems.

The graph of a circle can be approximated very well with cubic Bézier curves.

For a circle cascaded from 3 segments of 120° each, the peripheral standard deviation of the graph compared to a circle based on sin/cos is in the order of 10^{-3}. If the supporting points are displayed separately according to $y = f_1(t)$… as well as $x = f_1(t)$… each as an independent path-time diagram, the cascading of composite Bézier curves shows, as expected, no kinks or jumps, the segments themselves are still continuously differentiable.

Bézier circle, 1st derivation

The 1st derivation (speed) reveals - although the tangent condition is fulfilled - a similar problem for a Bézier circle as for a Bézier straight line: While the "sin/cos circle" is traversed on a CNC machine with the same angular and thus cutting speed, in a Bézier circle not only different speeds occur within the individual segments; the segment transitions are - as a result - by no means jerk-free:

The speed increases from the center point to the end point (acceleration positive), but decreases again after the transition to the new segment (acceleration negative). The transition is abrupt! Since the acceleration reversal takes place in the connection point without temporal transition, the acceleration goes towards infinity!

The derivative of a Bézier curve can optionally be represented for x and y separately in 2 independent graphs x = f(t) as well as y = f(t) or from the xy pair also as a Bézier curve. "Derivations of a Bézier curve are themselves Bézier curves" (Lichtmannegger, 2010).

Bézier circle, 1st derivation

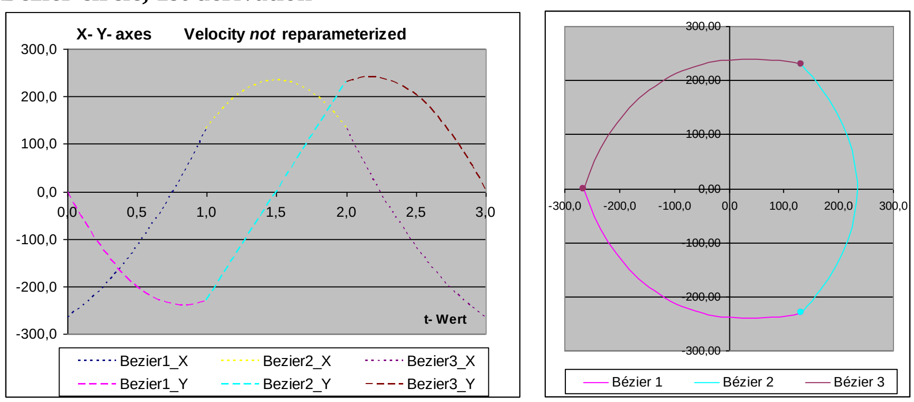

Fig. 2_02a *xy-axes velocity versus time* 2_02b *Bézier (vector length)*
images identical: -- > t = const.
as well as -- > t <> const.

With the Bézier circle (Fig. 2_02b) the 1st derivative reminds of the "disk of the rotary piston Wankel engine", whose center lies in the zero point of the coordinate system. If the distance between the zero point of the coordinate system and the bend point of the curve is normalized to the unit vector, then for every other vector position its magnitude corresponds to the relative angular velocity.

Bézier circle, vector length versus time

As an "*integral versus the velocity amount*", the length of each section is calculated from the velocity multiplied by its time base. A time base that is the same throughout thus results in time-proportional line segments. If we normalize the initial and final velocity in the base points to "1" and 100%, respectively, the minimum velocity in the center of the circle segment is about 88%, (cf. Fig. 2_03a). In order to minimize these speed differences, it is therefore obvious to adjust the step size of the t-values, cf. reparameterization, 2.1.

Bézier Circle, Reparameterization

An intermediate step of the reparameterization is shown in the following figure:

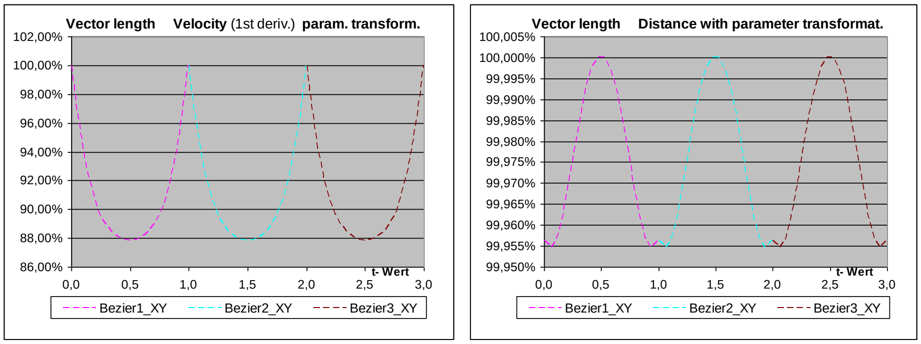

Fig. 2_03a *Speed versus time* 2_03b *Step size versus time*

-- > *t = parameterization to constant velocity*

- The graph Fig. 2_03a shows the example of a circle of 3 segments each 120° with n = 20 Bézier steps corresponding to Fig 2_02b in a different diagram format:
  Regardless of using reparameterized time steps (τ) - or unparameterized time step $t_{(const)}$ - the same graph results for the velocity course of the circle (min = approx. 88% max).
- Fig. 2_03b shows the leveling of the step size to min = approx. 99.95% max.
  - Cascading (cf. 2.1 / 2.1.1) improves the fluctuation (min/max) to 0.05%!

It is surprising at first, that the algorithm transforms the distance of the grid points (Fig. 2_03b), but has no influence on the fluctuation width of the velocity (Fig. 2_03a), although the graphs are connected via their time steps!

Here two explanations.

The Bézier locus curve is determined from the Bernstein polynomials, the construction points for any "t" lie on the graph determined from "t". This is also true for the t-values transformed for leveling the orbital velocity (time steps "τ"). "The derivation of a Bézier curve is itself a Bézier curve" (Lichtmannegger, 2010).

- which is determined from derived Bernstein polynomials, therefore changed time steps still produce the same velocity curve.
- Fig. 2_03a shows the velocity evaluation of the 1st derivative (Differen*tial* quotient). As expected, the Differ*ence* quotient, which evaluates the secant slope (construction point width divided by time step), produces a graph with the same image. In the sense of the result comparability of both graphs/procedures it is purposeful to assign a xy-point position from interpolation of neighboring tangent points to the secant slope. Fig. 2_03a shows for both methods an almost equal graph.
  The calculation approach "Differ*ence* quotient"
  makes clear, that a corrected distance of the support point is based on a proportional change of the time base "t" or "τ" - the velocity therefore does not change significantly.

The approximation algorithm according to 2.1 *parameterizes* each Bézier circle segment *according* to or *proportional to the arc length*. For this purpose, the (reparameterized) construction points calculated with transferred time step size "τ" (having now the same support point width) are assigned to the *original, equal time intervals* (t = const.).

Bézier circle, speed levelled by reparameterization

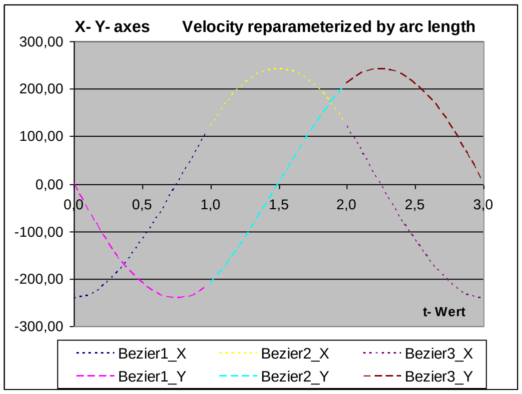

Fig. 2_05a *xy-axes velocity versus time*

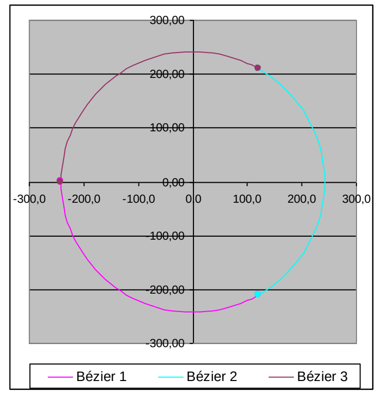

Fig. 2_05b *Bézier (vector length)*

- In the example of the cascaded Bézier circle, the 1st derivative shows (almost) uniform velocity for all 3 segments. The xy-coordinates reparameterized to the same step size lead to *almost smooth* transitions at the same time step "t" (Fig. 2_05a / 2_05b).
- The 1st derivative of the *unparameterized* locus curve (Fig. 2_02b) showed a graph similar to the "disk of the rotary piston Wankel engine ". The reparameterization transforms this into an almost sin/cos like graph (Fig. 2_05).

The algorithm published at 2.1 improves the fluctuation of the velocity (min/max) to 0.05%!

SUMMARY

Thanks to the ingenious mathematicians Pierre Étienne Bézier (Renault) and Paul de Casteljau (Citroën), curves and shapes became calculable in the 1960s, the "elegance" of which caused the greatest sensation. These curves were developed for static shaping, they are standard for CAD systems, but not directly applicable for dynamic processes.

With the knowledge published above, it is now possible, with simple mathematics, to calculate very good approximations for a reparameterization of Bézier curves according to arc length and to reduce shape-dependent variations in cutting speed to a range of 10^{-2} to 10^{-3} of their original value; please compare Fig.. 2_01a / 2_01b.

ACKNOWLEDGEMENT

I would like to thank Dr. rer. nat. Thomas Harbich for our intensive exchange of ideas, his suggestions and valuable advice - not only in the context of this paper - but especially in the focus of catenary routing as well as parameterization and its classical derivation.

REFERENCES

Babovsky, Hans und Neundorf, Werner (2011) Numerische Approximation von Funktionen: Bézier–Kurven pp 16 -34 / Der Algorithmus von de Casteljau pp 38 – 39 [online], Technische Universität Ilmenau, Institut für Mathematik Prof. Dr. H. Babovsky, PD Dr. W. Neundorf, Available from: https://www.db-thueringen.de/receive/dbt_mods_00017667

Fetouaki, Hamid and Skopin Emma (2009), Bézier-Kurven pp1 - 5 / Der Algorithmus von de-Casteljau pp 6 -16 / Die Bernsteinform einer Bézier-Kurve pp 17 - 35 / Ableitungen von Bézier-Kurven pp 36 – 49 [online] Universität Kassel / FB Mathematik/Informatik, Prof. Dr. Andreas Meister, Available from: http://pool-serv1.mathematik.uni-kassel.de/~fetouaki/seminar/presentation_handout.pdf

FMI, Fakultät für Mathematik und Informatik, Nach Bogenlänge parametrisierte Kurve pp 1 – 20 [online] Friedrich-Schiller-Universität Jena. Available from: https://users.fmi.uni-jena.de/~matveev/Lehre/KlassDiffGeo/vorlesung2.pdf

Lichtmannegger, Andreas (2010), Die Ableitungen einer Bézierkurve sind selbst wieder Bezierkurven [online] Technische Universität München, Zentrum Mathematik, Prof. Dr. Jürgen Richter-Gebert. Available from: https://www-m10.ma.tum.de/bin/view/MatheVital/BezierKurven/WebHome.html https://www-m10.ma.tum.de/bin/view/MatheVital/BezierKurven/AbleitungenAllgemein.html

Oberle, Hans Joachim (2013/14) Bézier–Kurven pp 125 -128 / Der Algorithmus von de Casteljau pp 128 – 130 [online] Universität Hamburg / FB Mathematik Prof. Dr. H. J. Oberle, i.R. Available from: https://www.math.uni-hamburg.de/home/oberle/approx-13/Skript/approx11.pdf

Roth, Jürgen, Differentialgeometrie, 2. Kurventheorie, (Nach Bogenlänge parametrisierte Kurven) pp 2.8 – 2.114 [online], Universität Koblenz, Institut für Mathematik, Prof. Dr. Jürgen Roth. Available from: https://www.juergen-roth.de/skripte/differentialgeometrie/differentialgeometrie_2_kurventheorie.pdf