Smoothing and Time Parametrization of Motion Trajectories for Industrial Machining and Motion Control

Květoslav Belda
Department of Adaptive Systems
The Czech Academy of Sciences, Institute of Information Theory and Automation
Pod Vodárenskou věží 4, 182 08 Prague 8, Czech Republic
belda@utia.cas.cz

Keywords: Path Model, G-Code, Path Smoothing, Time Parametrization, Bézier Curves, Industrial Robotics.

Abstract: The paper deals with path smoothing and time parametrization procedures intended for motion control of industrial machine tools and robots. Path smoothing, considered in this paper, is based on the application of Bézier curves. A possible straightforward solution ensuring compliance with given admissible positional tolerances is introduced. Consequent time parametrization considered here employs arc length and specific construction of acceleration polynomials. It describes the motion along the obtained smoothed curve geometry. It is given by timing the arc length, thus the construction of the feed rate profile. The key parts of the time parametrization comprise: computation of path length; time parametrization with respect to arc length; and decomposition to the individual Cartesian components describing individual curve coordinates. The theoretical results are presented by representative examples in 2D and 3D spaces.

1 INTRODUCTION

Nowadays, a lot of Computer-Aided Design (CAD) tools offer a simple way of construction of various paths and shapes with complicated geometry. This geometry is usually stored as sets of points and modeled by a combination of specific parametric curves. One example of the advanced CAD tools is NX PLM software (Siemens, 2019), which, with the application of industry-oriented modules, offers the option to store path's description in specific G-Code used in Computer Numeric Control (CNC) machine tools (Xavier et al., 2010). This G-Code can be uploaded directly to the real control system of a machine tool or robot to ensure required motion along reference trajectory.

In G-Code, the geometry data are typically restricted to low level segments such as points (dwell, fixed positions), straight lines (abscissas, linear segments) and arcs (circular segments). These restrictions lead to huge numbers of linear segments due to complicated initial curve geometry and several circular segments for pure circular arcs. Transition between such segments are contiguous, but not smooth.

In order to satisfy the given kinematic constrains without having to stop at each point of higher order discontinuity, the path needs to be smoothed in a preprocessing procedure before the actual time parametrization, from which full-featured motion trajectories arise (Zhang et al., 2018). Note that every stop means increase of working time and increasing of energy demands on braking and repeated starting (Othman et al., 2015).

A recent trend in industrial production is to employ increasingly complicated motion trajectories especially in the case of industrial robots, see e.g. Fig. 1. Thus, real-world testing on machine tools becomes increasingly expensive. Therefore, newly generated trajectories are first verified using software simulations so that collisions and unreachable G-code parts can be detected and optimized.

Figure 1: Machining with an industrial robot.

https://orcid.org/0000-0002-1299-7704

Belda, K.
Smoothing and Time Parametrization of Motion Trajectories for Industrial Machining and Motion Control.
In *Proceedings of the 16th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2019) - Volume 2*, pages 229-236
ISBN: 978-989-758-380-3

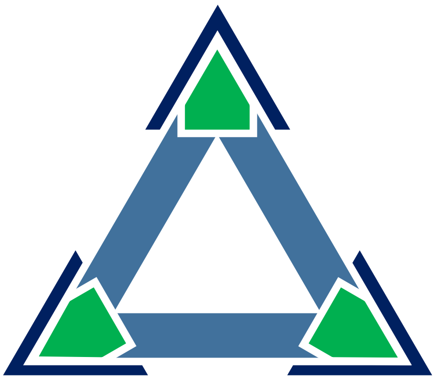

Figure 2: Qualitative triangle: Speed-Accuracy-Surface fin.

The focus of this paper is to introduce straightforward smoothing path algorithm and its incorporation to a specific one-shot off-line time parametrization of the motion trajectories given by individual lines of G-Code. For smoothing of the trajectories, a specific analytical algorithm employing Bézier curves of fifth-degree (quintic Bézier curve) is proposed. The degree of Bézier curve was selected to achieve smooth connection to other trajectory segments (Sencer et al., 2015). The problem can be looked at from a more general point of view, represented by the qualitative triangle: Speed-Accuracy-Surface finish, shown in Fig. 2. It depicts relations among specific qualitative features taken in to account in motion control. All three features cannot naturally be achieved simultaneously in full. Instead, a suitable compromise is sought, which respects the given technological and economic requirements.

Recently, the requirement is to have very short production cycle with reasonable accuracy and quality. Therefore, to solve such antagonistic requirement (dual problem), the technology takes into account definite tolerances i.e. engineering fits, since no element or part can be manufactured completely accurately. The range of permissible dimensions or tolerances is determined with respect to the element or part function. This feature determine useful admissible range in which the motion trajectory should be maintained. The size of the admissible range (tolerance) forms interval for the motion smoothing. This idea of admissible tolerances is considered in the intended solution of smoothing problem, proposed here in our paper.

This paper is organized as follows. Section 2 describes the tested path model and its G-Code. Section 3 contains the description of the proposed smoothing algorithm based on a specific construction of quintic Bézier curves. Section 4 deals with arc-length time parametrization procedure including the computation of arc-length using Simpson's rule, time parametrization with respect to adaptive arc length and backward decomposition to the individual Cartesian components, i.e. individual curve coordinates. Section 5 demonstrates representative examples of smoothed curves constructed by the proposed algorithm implemented in MATLAB environment.

2 PATH MODEL AND G-CODE

Path model reflects the target application and its technological parameters. In the case of machining, the parameters include maximum feed rate, acceleration and jerk. These parameters are given by the specific construction of the machine tool or robot. Limits on mentioned parameters are accompanied by additional technological limits such as maximum cutting velocity and required accuracy (admissible or prescribed tolerances and character of nominal dimensions).

All these parameters determine resultant motion, the parameters of machine tools determine start-up time (acceleration, time to reach the desired feed rate from zero) and stopping time (deceleration, time to stop from feed rate to zero) whereas inherent running phase depends on prescribed feed rate (given by specific technology and the used cutting tool) and the geometry of the motion path (Msaddek et al., 2014; Luo et al., 2007).

Thus, the path model consists of the above mentioned constraints on the tool motion together with the description of the path geometry. The simplest (and still most common) way of describing the geometry of the path is to use a combination of linear and circular arc segments. Other, more advanced methods for the path description and representation exist, such as B-spline, Akima spline and NURBS, but these elements are not universally supported. A lot of research has been done in the recent years on the topic of construction of and motion planning on these curves (Erkorkmaz et al., 2017; Heng and Erkorkmaz, 2010; Sencer et al., 2015). However, the research is still ongoing and the commercial implementation is not yet common and standardized.

The used G-code in this paper is presented in this section. Its parametric interpretation by linear segments and Bézier curves will be used for the proposed smoothing algorithm. The algorithm outputs will be involved in the method of time parametrization.

Now, let us introduce the G-Code used for testing. The elements in the G-Code are the following: Rapid positioning G00; Linear interpolation G01; (for completeness: Circular interpolations, clockwise/counterclockwise G02/G03; Dwell G04;) End of program M30; Feed rate F $[mm\,min^{-1}]$. The used G-Code example is in the Table 1.

Table 1: G-Code of testing trajectory (*mm*)

N010	*ADIS = 0.005*			*% smth. tol.*	
N020	G01	X100	Y0	Z0	F18000
N030	G01	X100	Y0	Z100	
N040	G01	X100	Y0	Z100	
N050	G01	X100	Y100	Z0	
N060	G01	X0	Y100	Z0	
N070	G01	X0	Y100	Z100	
N080	G01	X0	Y0	Z100	
N090	G00	X0	Y0	Z0	
N100	M30				

(considered starting point: X0 Y0 Z0)

3 SMOOTHING ALGORITHM

This section deals with the algorithm using quintic Bézier curves (curves described using 5^{th} order Bernstein's basis polynomials) (Piegl and Tiller, 1997). The algorithm copes with the smoothing of sharp corners while satisfying the specified tolerance limits. The proposed algorithm relies only on analytic formulas. Two types of parameters u and p are considered here. The parameter u is the parameter given by Bézier curve parametrization. Its relation to arc length cannot be analytically described. Whereas, the parameter p is the parameter corresponding to arc-length. It is directly applicable in abscissa and arc segments, where is in linear proportion to distance and angle respectively.

3.1 Definition of used Bézier Curve

As was mentioned, used Bézier curve is general, quintic curve. It is described by the following set of equations for its geometric points $B(u)$ (1) and appropriate derivatives (2) - (5):

$$\begin{aligned} B(u) &= (1-u)^5 P_1 + 5u(1-u)^4 P_2 \\ &+10u^2(1-u)^3 P_3 + 10u^3(1-u)^2 P_4 \\ &+5u^4(1-u)P_5 + u^5 P_6 \end{aligned} \quad (1)$$

$$\begin{aligned} \frac{dB(u)}{du} &= 5(1-u)^4(P_2-P_1) \\ &+20u(1-u)^3(P_3-P_2) + 30u^2(1-p)^2(P_4-P_3) \\ &+20u^3(1-u)(P_5-P_4) + 5u^4(P_6-P_5) \end{aligned} \quad (2)$$

$$\begin{aligned} \frac{d^2B(u)}{du^2} &= 20(1-u)^3(P_3-2P_2+P_1) \\ &+60u(1-u)^2(P_4-2P_3+P_2) \\ &+60u^2(1-u)(P_5-2P_4+P_3) \\ &+20u^3(P_6-2P_5+P_4) \end{aligned} \quad (3)$$

$$\begin{aligned} \frac{d^3B(u)}{du^3} &= 60(1-u)^2(P_4-3P_3+3P_2-P_1) \\ &+120u(1-u)(P_5-3P_4+3P_3-P_2) \\ &+60u^2(P_6-3P_5+3P_4-P_3) \end{aligned} \quad (4)$$

$$\begin{aligned} \frac{d^4B(u)}{du^4} &= 120(1-u)(P_5-4P_4+6P_3-4P_2+P_1) \\ &+120u(P_6-4P_5+6P_4-4P_3+P_2) \end{aligned} \quad (5)$$

where $u \in \langle 0,1 \rangle$ is a parameter of Bézier curve; $P_i,\ i = 1,\cdots,6$ are control points of the curve; $B(u) = [x(u), y(u), z(u)]^T$ are the Cartesian coordinates of curve points; $\frac{dB(u)}{du} = [v_x(u), v_y(u), v_z(u)]^T$, $\frac{d^2B(u)}{du^2} = \cdots$ represent the appropriate derivatives with respect to the parameter u.

Note that the equations (1) - (5) are expressed as polynomials in the parameter u with coefficients explicitly given by the combination of the control points.

3.2 Algorithm Principle

The principle of the algorithm is to use Bézier curve segment that smoothens the sharp connection between two intersecting lines. To construct the suitable Bézier segment, all control points can be expressed using only one parameter d, which corresponds to the admissible tolerance ε.

Hence, for prescribed tolerance, a unique analytical solution, depending just only on the tolerance ε and the initial, transition and end points P_s, P_t and P_e (see Fig. 3), can be obtained.

The proposed algorithm procedure includes the following parts:

i) evaluation of Bézier points and derivatives for $u = 0$, $u = 0.5$ and $u = 1$

ii) points and derivatives for intersecting lines

iii) comparison of corresponding derivatives

iv) determining point $B(u)|_{\,u=0.5}$

v) expressing control points via parameter d

vi) computation of the parameter d

These parts represent a sequence of steps for the determination of the parameter d as it is shown in Fig. 3.

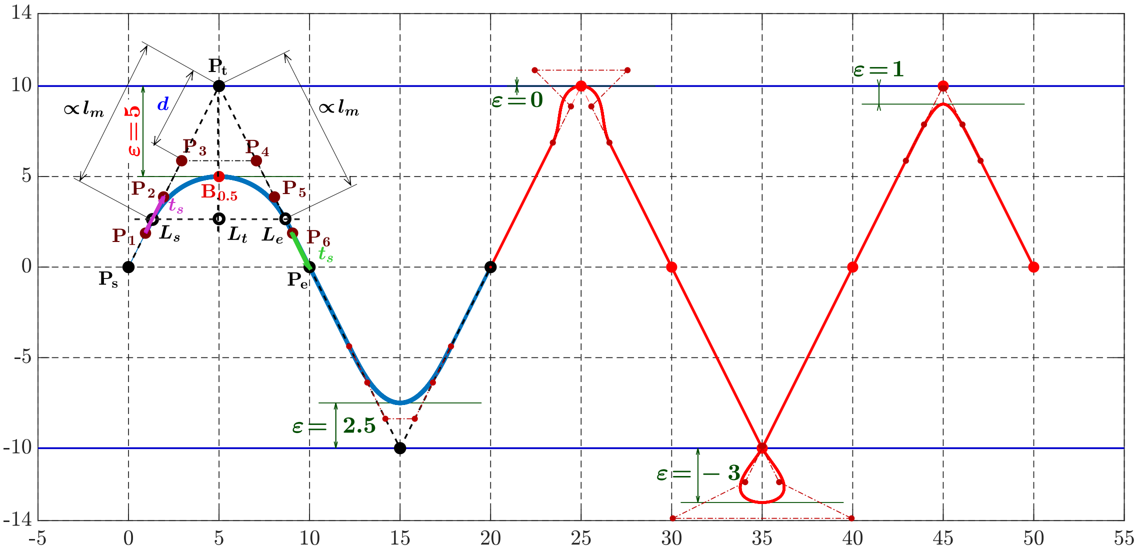

Figure 3: Bézier parameters and examples of the smoothing for different required offsets (tolerances).

The explanation of individual parts is as follows:

i) evaluation of Bézier points and derivatives for $u = 0$, $u = 0.5$ and $u = 1$:

$$B(u)|_{u=0} = P_1 \tag{6}$$

$$B(u)|_{u=0.5} = \frac{1}{32}P_1 + \frac{5}{32}P_2 + \frac{10}{32}P_3 + \frac{10}{32}P_4 + \frac{5}{32}P_5 + \frac{1}{32}P_6 \tag{7}$$

$$B(u)|_{u=1} = P_6 \tag{8}$$

$$\left.\frac{dB(u)}{du}\right|_{u=0} = 5(P_2 - P_1) \tag{9}$$

$$\left.\frac{dB(u)}{du}\right|_{u=1} = 5(P_6 - P_5) \tag{10}$$

$$\left.\frac{d^2B(u)}{du^2}\right|_{u=0} = 20(P_3 - 2P_2 + P_1) \tag{11}$$

$$\left.\frac{d^2B(u)}{du^2}\right|_{u=1} = 20(P_6 - 2P_5 + P_4) \tag{12}$$

ii) points and derivatives for intersecting lines:
The initial description, used in this part, follows from parametric equations of lines, see (13) and (16) and Fig. 3:

$$L_s(p) = P_s + p(P_t - P_s) \tag{13}$$

$$\frac{dL_s(p)}{dp} = P_t - P_s \tag{14}$$

$$\frac{d^2L_s(p)}{dp^2} = 0 \tag{15}$$

$$L_e(p) = P_t + p(P_e - P_t) \tag{16}$$

$$\frac{dL_e(p)}{dp} = P_e - P_t \tag{17}$$

$$\frac{d^2L_e(p)}{dp^2} = 0 \tag{18}$$

Note that the parameter p, appearing in (13) and (16), is a natural geometric parameter corresponding to arc length of the segment. This parameter will be used as the reference parameter in time parameterisations. Since the parameter of Bézier curve u is not linearly related to the arc-length, it will be also transformed to the mentioned parameter p. However, both parameters are considered as uniform parameters, i.e. $p \in \langle 0,\, 1\rangle$ as well as $u \in \langle 0,\, 1\rangle$ by definition;

iii) comparison of corresponding derivatives
(for smooth joining of abscissas and Bézier curve):

$$1^{st}\text{ derivative:}\quad 5(P_2 - P_1) = P_t - P_s \tag{19}$$

$$\Rightarrow\quad P_2 = \frac{1}{5}(P_t - P_s) + P_1 \tag{20}$$

$$2^{nd}\text{ derivative:}\quad P_3 - 2P_2 + P_1 = 0 \tag{21}$$

$$\Rightarrow\quad P_3 = \frac{2}{5}(P_t - P_s) + P_1 \tag{22}$$

$$1^{st}\text{ derivative:}\quad 5(P_6 - P_5) = P_e - P_t \tag{23}$$

$$\Rightarrow\quad P_5 = \frac{1}{5}(P_t - P_e) + P_6 \tag{24}$$

$$2^{nd}\text{ derivative:}\quad P_6 - 2P_5 + P_4 = 0 \tag{25}$$

$$\Rightarrow\quad P_3 = \frac{2}{5}(P_t - P_e) + P_6 \tag{26}$$

iv) determining point $B(u)|_{u=0.5}$:

$$\left.\begin{aligned} l_s &= \sqrt{(x_t-x_s)^2+(y_t-y_s)^2+(z_t-z_s)^2} \\ l_e &= \sqrt{(x_t-x_e)^2+(y_t-y_e)^2+(z_t-z_e)^2} \end{aligned}\right\}$$
$$l_m = \min(l_s, l_e) \tag{27}$$

$$\left.\begin{aligned} L_s &= P_t + (P_s - P_t)\tfrac{(0.5\,l_m)}{l_s} \\ L_e &= P_t + (P_e - P_t)\tfrac{(0.5\,l_m)}{l_e} \end{aligned}\right\}$$
$$L_t = \tfrac{1}{2}(L_s + L_e) \tag{28}$$

$$l_t = \sqrt{(x_{L_t}-x_{P_t})^2+(y_{L_t}-y_{P_t})^2+(z_{L_t}-z_{P_t})^2} \tag{29}$$

$$B(0.5) = P_t + \varepsilon\,\frac{L_t - P_t}{l_t} \tag{30}$$

v) expressing control points via parameter d:
Unknown control points of Bézier curve can be expressed straightforwardly by specific distance d from transition point (intersection point of two contiguous lines). Specifically, d is the distance from transition point P_t to control points P_3 and P_4.

$$t_s = \frac{P_t - P_s}{l_s}, \quad (l_s = \|t_s\|) \tag{31}$$

$$t_e = \frac{P_t - P_e}{l_e}, \quad (l_e = \|t_e\|) \tag{32}$$

$$P_1 = P_t - \frac{2}{5}l_m t_s - d\,t_s \tag{33}$$

$$P_2 = P_t - \frac{1}{5}l_m t_s - d\,t_s \tag{34}$$

$$P_3 = P_t - d\,t_s \tag{35}$$

$$P_4 = P_t - d\,t_e \tag{36}$$

$$P_5 = P_t - \frac{1}{5}l_m t_e - d\,t_e \tag{37}$$

$$P_6 = P_t - \frac{2}{5}l_m t_e - d\,t_e \tag{38}$$

vi) computation of the parameter d:
The computation of the searched parameter d is determined by eqs. (30) and (7) with eqs. (33) – (38).

$$\left.\begin{aligned} B(0.5) &= P_t + \varepsilon\,\frac{L_t - P_t}{l_t} \end{aligned}\right\}$$
$$\left.\begin{aligned} 32B(0.5) &= P_1 + 5P_2 + 10P_3 + 10P_4 + 5P_5 + P_6 \\ &= 32P_t - 16d(t_s+t_e) - \frac{7}{5}l_m(t_s+t_e) \end{aligned}\right\}$$

$$\Rightarrow \quad d = 2\,\frac{(P_t - B(0.5))}{.(t_s+t_e)} - \frac{7}{80}\,l_m \tag{39}$$

4 TIME PARAMETRIZATION

The time parametrization with respect to arc length includes just computation of path length, time parametrization with respect to arc length (timing of geometric parameter $p(t)$) and decomposition to the individual curve coordinates. The following individual sections will show related theoretical background.

4.1 Computation of the Path Length

Computation of the path length is defined as:

$$\ell = s(u)|_{u=1} = \int_0^{u=1} ds = \int_0^{u=1} \sqrt{\dot{x}^2+\dot{y}^2+\dot{z}^2}\,du \tag{40}$$

The integral (40) can be evaluated analytically only for simple segments such as lines or arcs. However, for Bézier curve it is necessary to use some approximative numerical method. Here, it is suitable to consider Simpson's rule: The given interval of parameter u is divided into an even number of subintervals by equidistant points, similarly as in the case of the trapezoidal rule, and in any interval $[u_{2i}, u_{2i+2}]$, $i = 0, 1, \cdots, \frac{m}{2}-1$, of the length $2h$ and the Newton-Cotes formula is used, then the resulting formula is

$$\begin{aligned} \int_a^b f(u)du \;&= \tfrac{1}{3}h[f(u_0)+4f(u_1)+2f(u_2) \\ &+\; 4f(u_3)+\cdots+4f(u_{m-3}) \\ &+\; 2f(u_{m-2})+4f(u_{m-1})+f(u_m)] + E(f(u)) \end{aligned} \tag{41}$$

where $f(u) = \sqrt{\dot{x}(u)^2+\dot{y}(u)^2+\dot{z}(u)^2}$, $h = \frac{b-a}{m}$, $a = 0$ and $b = 1$; and $E(f(u)) = -\frac{1}{180}h^4\,\frac{d^4f}{du^4}\,(o)$ is a numerical error (Rektorys, 1994).

4.2 Timing of Geometric Parameter

Time parametrization based on arc length can be realized according to selection of acceleration polynomial a of 1^{st}, 3^{rd} or 5^{th} order. For purpose of this paper, let us consider acceleration polynomial a of 1^{st} order. Other possibilities are described e.g. in (Heng and Erkorkmaz, 2010; Huang and Zhu, 2016).

For the selection of the acceleration polynomial of a of 1^{st} order, there exist two main acceleration shapes: triangular and trapezoidal, both with three special cases: limit case without involved central zero acceleration determining constant feed rate; start-up and stop phases. The mentioned two main shapes are illustrated in Fig. 4 (Belda and Novotný, 2012).

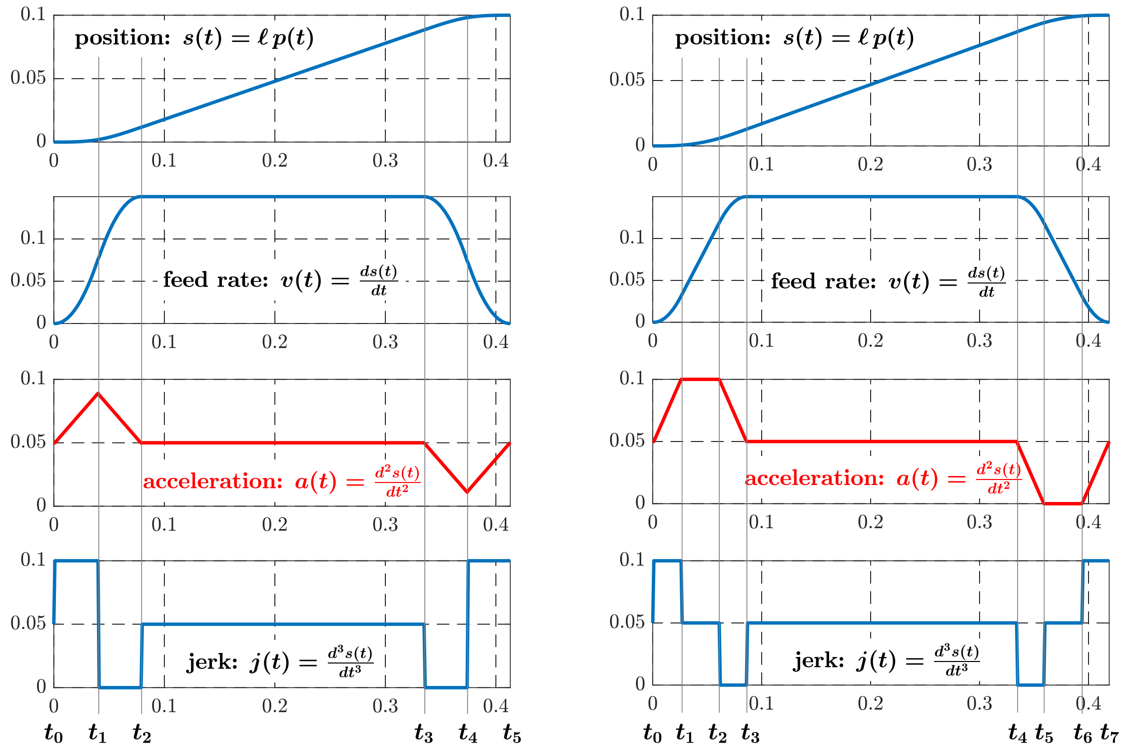

Figure 4: Kinematic quantities considering acceleration $a(t)$ of 1^{st} order with triangular (left) or trapezoidal (right) profile.

Time parametrization follows from the profiles (Fig. 4) and it is determined for all indicated time intervals, in which all related limiting kinematic quantities (j_{max}, a_{max}, v_{max}) are considered. The intervals for triangular acceleration profile are:
$\langle t_0, t_1\rangle, \langle t_1, t_2\rangle, \langle t_2, t_3\rangle, \langle t_3, t_4\rangle$ and $\langle t_4, t_5\rangle$;
and for trapezoidal acceleration profile are:
$\langle t_0, t_1\rangle, \langle t_1, t_2\rangle, \langle t_2, t_3\rangle, \langle t_3, t_4\rangle, \langle t_4, t_5\rangle, \langle t_5, t_6\rangle$ and $\langle t_6, t_7\rangle$.
In general, the following consecutive integration is solved just within defined time intervals:

$$j(t) = a_0, \quad a_0 = k\, j_{max}, \quad k \in \{-1, 0, 1\} \tag{42}$$

$$a(t) = \int j(t)\, dt = \quad a_0\, t \; + \quad a_1 \tag{43}$$

$$v(t) = \int a(t)\, dt = \frac{1}{2} a_0\, t^2 + \quad a_1\, t \; + a_2 \tag{44}$$

$$s(t) = \int v(t)\, dt = \frac{1}{6} a_0\, t^3 + \frac{1}{2} a_1\, t^2 + a_2\, t + a_3 \tag{45}$$

$p(t) = \frac{s(t)}{\ell}$, $\dot{p}(t) = \frac{v(t)}{\ell}$, $\ddot{p}(t) = \frac{a(t)}{\ell}$ and $\dddot{p}(t) = \frac{j(t)}{\ell}$.

The result of the integration are coefficients in individual resulting equations. Then, the geometric parameter including its appropriate derivatives can be applied in the decomposition of the motion trajectory to individual coordinate components (axes) (Belda et al., 2007).

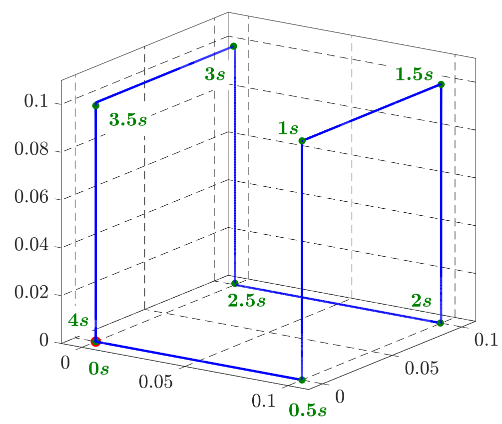

Figure 5: Testing trajectory (axes in $[m]$).

5 REPRESENTATIVE EXAMPLES

5.1 Runs with tolerances $\approx 10^{-6} m$

As was mentioned, the G-Code, used for comparative runs, is already introduced in the Table 1. Its graphical representation is in Fig. 5. The time labels show approximative time and direction of the motion.

Table 2: Testing parameters.

parameter	symbol	value
admissible tolerance	ε	$5 \cdot 10^{-6}\,m$
max. feed rate	v_{max}	$0.3\,m\,s^{-1}$
max. acceleration	a_{max}	$2\,m\,s^{-2}$
max. jerk	j_{max}	$200\,m\,s^{-3}$
sampling period	T_s	$10^{-3}\,s$

The used parameters of the runs is in the Table 2. The runs are shown in the following Fig. 6 - Fig. 7.

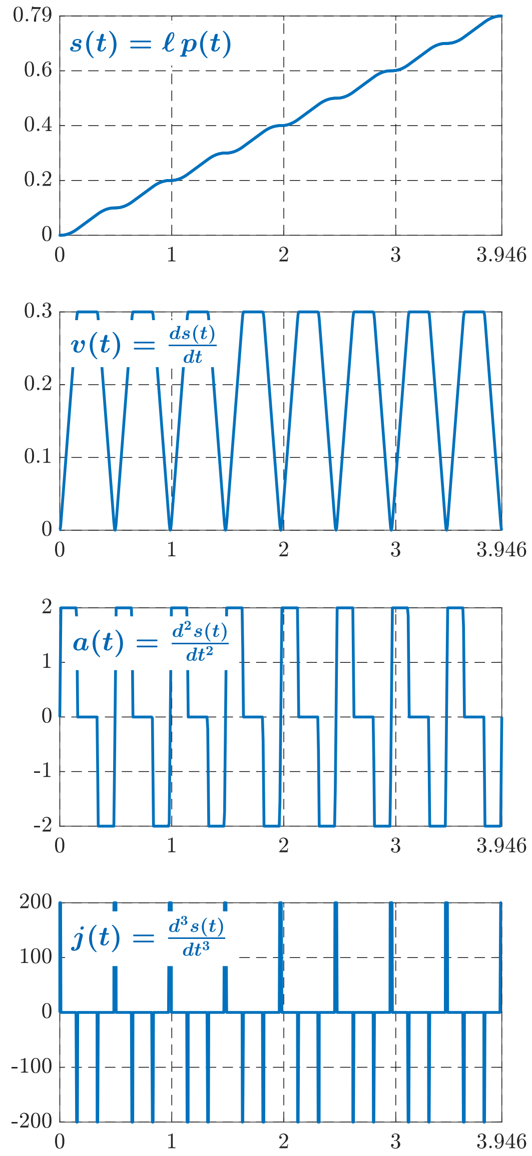

Figure 6: Time behaviours [s]: $s(t)$, $v(t)$, $a(t)$, $j(t)$.

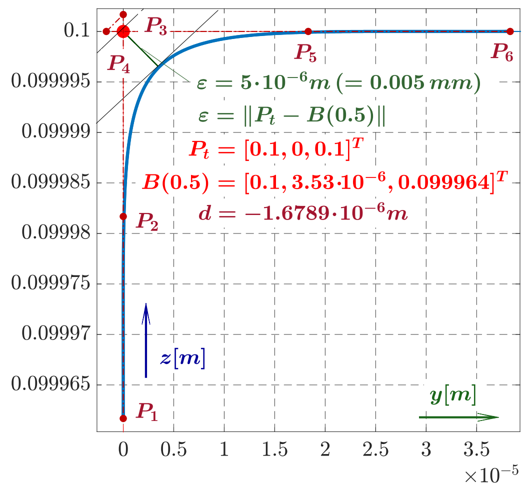

Figure 7: Detail of smoothing with proposed algorithm.

Fig. 6 shows results of proposed smoothing algorithm with time parametrization. Since the tolerance $\varepsilon = 0.005\,mm$ is too small, it is necessary to explore the details in Fig. 7, which shows one Bézier segment of the testing geometric path (segment at front upper-right vertex $P_t = [0.1, 0, 0.1]^T$). Total length saving is $0.04\,mm$ per 7 corners only. However, in sum in industrial machining, the saving is noticeable.

5.2 Example with tolerances $\approx 10^{-3}m$

Values of d-parameter for set of Bézier curves in general 3D position (Fig. 8) are in the Table 3.

Table 3: List of d-parameters.

tolerance ε	parameter d
$\varepsilon_1 = 2.0\,mm$	$d_1 = 1.9223\,mm$
$\varepsilon_2 = 1.5\,mm$	$d_2 = 1.2579\,mm$
$\varepsilon_3 = 0.0\,mm$	$d_3 = -0.8750\,mm$
$\varepsilon_4 = 1.0\,mm$	$d_4 = 0.3972\,mm$
$\varepsilon_5 = -2.0\,mm$	$d_5 = -3.8915\,mm$
$\varepsilon_6 = 3.0\,mm$	$d_6 = 3.5864\,mm$
$\varepsilon_7 = -3.0\,mm$	$d_7 = -4.6811\,mm$
$\varepsilon_8 = 0.5\,mm$	$d_8 = -0.2406\,mm$

6 CONCLUSION

This paper discusses the problem of smoothing connection of linear segments by Bézier curves. The results demonstrate achieving of prescribed tolerances in the context of machine tools ($\approx 10^{-6}m$) as well as bigger motion ($\approx 10^{-3}m$) for robotic case by derived algorithm using MATLAB environment.

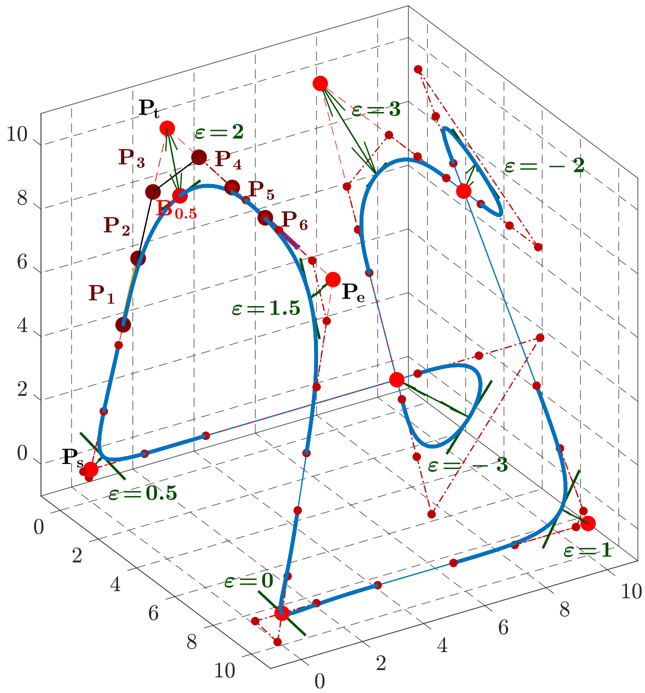

Figure 8: Example of general trajectory smoothing with tolerances $\approx 10^{-3}m$.

Future work will be focussed on on-line solution of smoothing and time parametrization that will be able to consider higher order polynomials.

REFERENCES

Belda, K., Böhm, J., and Píša, P. (2007). Concepts of model-based control and trajectory planning for parallel robots. In Klaus, S., editor, *Proc of 13th IASTED Int. Conf. on Robotics and Applications*, pages 15–20. Acta Press.

Belda, K. and Novotný, P. (2012). Path simulator for machine tools and robots. In *Proc. of the 17th Int. Conf. on Methods and Models in Automation and Robotics*, pages 373–378. West Pomeranian University of Technology.

Erkorkmaz, K., Chen, Q.-G., Zhao, M.-Y., Beudaert, X., and Gao, X.-S. (2017). Linear programming and windowing based feedrate optimization for spline toolpaths. *CIRP Annals*, 66(1):393 – 396.

Heng, M. and Erkorkmaz, K. (2010). Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability. *Int. J. Machine Tools & Manufacture*, 50:281–293.

Huang, J. and Zhu, L. (2016). Feedrate scheduling for interpolation of parametric tool path using the sine series representation of jerk profile. *Proc. of the Inst. of Mech. Engineers, Part B: J. Engineering Manufacture*, 231.

Luo, F.-Y., Zhou, Y.-F., and Yin, J. (2007). A universal velocity profile generation approach for high-speed machining of small line segments with look-ahead. *The Int. J. of Advanced Manufacturing Technology*, 35(5):505–518.

Msaddek, E. B., Bouaziz, Z., Baili, M., and Dessein, G. (2014). Influence of interpolation type in high-speed machining (hsm). *The Int. J. of Advanced Manufacturing Technology*, 72(1):289–302.

Othman, A., Belda, K., and Burget, P. (2015). Physical modelling of energy consumption of industrial articulated robots. In *Proc. 15th Int. Conf. on Control, Automation and Systems, ICCAS 2015*, pages 784–789. Institute of Control, Robotics and Systems, ICROS.

Piegl, L. and Tiller, W. (1997). *The NURBS Book*. Springer.

Rektorys, K. (1994). *Survey of Applicable Mathematics*. Springer.

Sencer, B., Ishizaki, K., and Shamoto, E. (2015). A curvature optimal sharp corner smoothing algorithm for high-speed feed motion generation of NC systems along linear tool paths. *Int. J. Adv. Manuf. Technol.*, 76:1977–1992.

Siemens (2019). NX docomentation. online.

Xavier, P., Yann, L., and Walter, R. (2010). Kinematic modelling of a 3-axis NC machine tool in linear and circular interpolation. *CoRR*, pages 1004–2354.

Zhang, Y., , Zhao, M., Peiqing, Y., , Jiang, J., and Zhang, H. (2018). Optimal curvature-smooth transition and efficient feedrate optimization method with axis kinematic limitations for linear toolpath. *The Int. J. of Advanced Manufacturing Technology*, 99(1):169–179.