Feedrate Planning for a Delta Parallel Kinematics Numerically Controlled Machine using NURBS Toolpaths

Gabriel Karasek[a] and Krystian Erwinski[b]

Institute of Engineering and Technology, Faculty of Physics Astronomy and Informatics, Nicolaus Copernicus University, Torun, Poland

Keywords: Numerical Control, Delta Parallel Kinematics, NURBS, Feedrate Planning.

Abstract: This paper presents a concept of a computationally efficient feedrate planning algorithm for parallel kinematics machine in a linear delta configuration. Non-Uniform Rational B-Spline (NURBS) polynomial curve is used for toolpath definition which provides a smooth trajectory. The feedrate profile is defined as a jerk limited S-Curve which takes into account limitations stemming from the toolpath curvature and the machine kinematics. The article presents the outline of the method, structure of the experimental station including the real-time control system and preliminary experimental results. Further direction of the research is also described. The proposed method can provide a smooth motion trajectory obtained in real-time with small computational requirements.

1 INTRODUCTION

In numerical control feedrate is the end effector velocity tangent to the toolpath. It is the result of combining the velocities of all of the machine's axes. Feedrate planning is the process of determining a feedrate profile that provides a proper value of end effector feedrate while respecting the machine's limitations such as maximum velocity, acceleration and jerk of its axes. These limitations usually stem from limiations of the drive power, drivetrain capabilities or contouring error.

The feedrate planning problem has been studied by many researchers for cartesian machines utlizing optimization algorithms with explicit inclusion of axial constraints (Mercy et al., 2018; Lu et al., 2018), tracking error constraints (Zhang et al., 2019) or contour error constraints (Chen and Sun, 2019). Alternatively simplified approaches were proposed using the feedrate limit function which combines axial limitations of velocity, acceleration and jerk into a single feedrate limit (Ni et al., 2018b; Ni et al., 2018a; Erwinski et al., 2022). In the second case the feedrate profile is usually the widely used jerk-limited s-curve profile. These feedrate planning methods usually consider cartesian machines sometimes with additional rotary axes in a 5-axis configuration (Li et al., 2021). Furthermore many of the aforementioned methods utilize the Non-Uniform B-Spline polynomial curve as a state-of-the-art way do define the toolpath which can provide a smooth trajectory compared to typical G-Code.

The feedrate planning problem for NURBS toolpaths is more complex for machines with non-cartesian kinematics which include parallel kinematics machines. This problem was not investigated previously as much for these types of machines but is currently an emerging research topic (Su et al., 2018; Li et al., 2021). Investigation of feedrate planning with inclusion of joint/drive limitations is the main focus of the authors' current research. The delta parallel configuration with linear actuators is the authors' main focus as such machines can achieve large feedrates and therefore high productivity with more stiffness than delta robots with rotary joints. Due to highly non-linear dependency between joint and end effector velocities, accelerations and jerks the problem is much more complex than for cartesian machines. The inverse kinematics has to be taken into account if axial constraints are to be considered. Additionally the authors' objective is to consider the computational efficiency and implementation aspects which are important if the developed feedrate planning methods are to be used in practice.

In this paper a computationally efficient method

[a] https://orcid.org/0000-0001-8031-8323
[b] https://orcid.org/0000-0001-6899-1785

Karasek, G. and Erwinski, K.
Feedrate Planning for a Delta Parallel Kinematics Numerically Controlled Machine using NURBS Toolpaths.
DOI: 10.5220/0011352900003271
In *Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO 2022)*, pages 743-749
ISBN: 978-989-758-585-2; ISSN: 2184-2809

for feedrate planning is presented for a delta parallel kinematics numerically controlled machine. The jerk limited S-Curve profile is used for feedrate planning and NURBS curves are used as toolpath definition. The feedrate planning and NURBS curve interpolation algorithms are described. The experimental station is described including a description of the algorithms' implementation in a real-time Linux operating system. Preliminary experimental results of the feedrate profile and individual axial velocity, acceleration and jerk profiles and following errors are presented.

2 NURBS CURVE TOOLPATH DEFINITION AND INTERPOLATION

Non-Uniform Rational B-Splines (NURBS) are polynomial spline curves commonly used in computer graphics but also as definitions of complex shapes for Computerized Numerically Controlled (CNC) machines. They are defined in cartesian space using control points forming a control polygon which roughly defines the curve's shape. The shape can be further influenced by control point weights which pull the the curve closer to the point if the weight is above 1. NURBS are rational curves which means they can be used to define most shapes. An additional advantage is the ability to locally influence the shape of the curve by moving each control point. Furthermore a curve of a defined degree exhibits continuity of it's derivatives if interior knots are not repeated. A cubic curve has G2 geometric continuity which means that it's second derivatives are continuous. This ensures smoothness of the curve as well as the motion trajectory of the end effector moving along this curve. An example of a NURBS curve is shown of Fig.1.

Mathematically NURBS curves are defined as (Piegl and Tiller, 1996):

$$C(u) = \frac{\sum_{i=0}^{n} N_{i,p}(u) w_i \cdot P_i}{\sum_{i=0}^{n} N_{i,p}(u) w_i} \tag{1}$$

where: u - curve parameter value, $N_{i,p}(u)$ - value of an elementary polynomial basis function, P_i - control point coordinates, w_i - control point weight. Each point on the curve is identified by a value of the unitless parameter with a value between 0 and 1 which indicate the beginning and end of the curve respectively. In practice evaluation of the curve is performed using the numerically stable DeBoor's algorithm. This

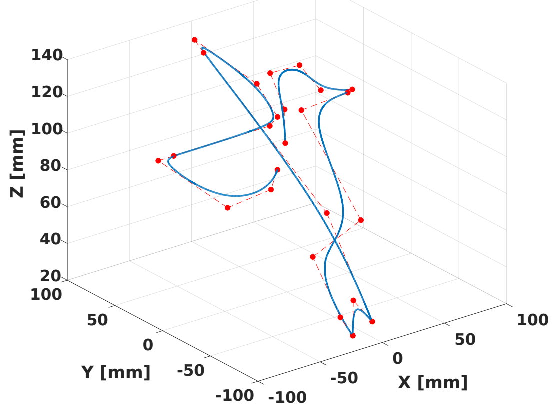

Figure 1: "Bird" NURBS toolpath (blue) with marked control points and polygon (orange).

algorithm is also used to compute curve derivatives with respect to the curve parameters.

Interpolation is the process of evaluating the toolpath in constant time intervals to generate points on the curve. In case of cartesian machines coordinates of the generated points are the position setpoints of each machine axis. In case of parallel kinematics machines the generated coordinates are transformed using inverse kinematics to generate setpoints for each joint. In each interpolation step a new value of the curve parameter u_{i+1} is produced corresponding to the desired arc-length increment and a point on the NURBS curve $C(u_{i+1})$ is produced using DeBoor's algorithm. Arc-length increments Δs between consecutive interpolation points (u_i, u_{i+1}) should be proportional to the desired feedrate F_i in the current interpolation step. This can be expressed as:

$$\Delta s = \int_{u_i}^{u_{i+1}} ||C'(u)|| du = V(t_i) \cdot \Delta t \tag{2}$$

where: $C'(u)$ - first derivative of the curve with respect to the curve paramter, $V(t_i)$ - desired feedrate at current interpolation time step, Δt - interpolation time step. This process is illustrated in Fig.2.

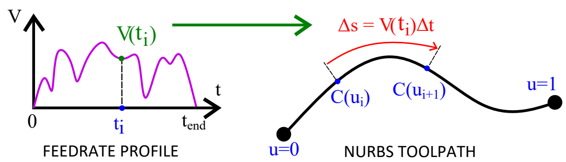

Figure 2: Illustration of NURBS interpolation process based on a polynomial feedrate profile.

The relationship between the curve parameter increment $\Delta u = u_{i+1} - u_i$ and arc-length increment ΔS is non-linear and unique for each curve. This means that in each interpolation step this relationship has to be approximated to satisfy the condition stated in eq.2.

There are several methods of NURBS interpolation such as second and third order Taylor series approximation, Predictor-Corrector or Feed Correction Polynomial (Erwinski et al., 2021). In this paper the second order Runge-Kutta Predictor-Corrector (RK2-PC) two-step method proposed in (Jia et al., 2017) is used. In the first step (predictor) a coarse approximation of the desired curve parameter value is produced:

$$\tilde{u}_{i+1} = u_i + \frac{1}{2}(F_1 + F_2)\Delta t \tag{3}$$

$$F_1 = \frac{du}{dt} = \frac{V(t_i)}{||C'(u_i)||} \quad F_2 = \frac{V(t_i)}{||C'(u_i + F_1\Delta t)||} \tag{4}$$

In the second step (corrector) a correction increment Δu_{i+1} is determined which modifies the coarse result $\tilde{u}_{i+1}$ of the first step. Ideally this should eliminate the arc-length error so that the following equation is satisfied:

$$||C(\tilde{u}_{i+1} + \Delta u_{i+1}) - C(u_i)|| = V(t_i) \cdot \Delta t \tag{5}$$

The end position is rewritten using first order Taylor expansion and substituted into equation 5. Then both sides of the obtained formula are squared and after some manipulations a quadratic equation is obtained:

$$a \cdot (\Delta u_{i+1})^2 + b \cdot \Delta u_{i+1} + c = 0 \tag{6}$$

$$a = ||C'(\tilde{u}_{i+1})||^2 \tag{7}$$

$$b = 2 \cdot C'(\tilde{u}_{i+1}) \cdot (C(\tilde{u}_{i+1}) - C(u)) \tag{8}$$

$$c = ||C(\tilde{u}_{i+1}) - C(u)||^2 - V^2(t_i)\Delta t^2 \tag{9}$$

The solution to this equation minimizes the interpolation error. The final corrected NURBS parameter value is thus given by:

$$u_{i+1} = \tilde{u}_{i+1} + \frac{-b + \sqrt{a^2 - 4a \cdot b}}{2a} \tag{10}$$

It can be seen that in the corrector step the RK2-PC method uses feedback from a trial interpolation to determine the actual interpolation error. The most commonly used 2nd and 3rd order Taylor series approximations are open-loop methods. They are susceptible to interpolation error accumulation when large feedrates and toolpath curvatures are involved. They also require computation of higher order NURBS derivatives(Erwinski et al., 2021).

3 DELTA PARALLEL KINEMATICS

Parallel kinematics is an alternative kinematics configuration, where the end effector is connected with the drives through two or more independent joints. Faster feedrate and lower weight of moving system are the main advantages compared to traditional cartesian machines with prismatic joints connected in series. Also, unlike cartesian machines, there is no accumulation of positioning error from each axis. All of axes are moving independently, which means that the maximum position error is equal to the maximum error in each axis. Lower weight of individual axes allows to achieve a lot higher accelerations - up to 300G. (Bouri and Clavel, 2010)

Most popular types of parallel kinematics are rotary delta, mostly used in pick-and-place machines, and linear delta, which most common applications are 3d printers and robots. This paper will focus on linear delta machine kinematics. In this type of machines, linear movements of parallel axes are combined to allow movement of end effector in 3 dimensional space. Main disadvantage of this kinematic configuration is greater number of mathematical equations, that need to be solved in order to determine the required joint setpoints for a desired cartesian position of the end effector (inverse kinematics).

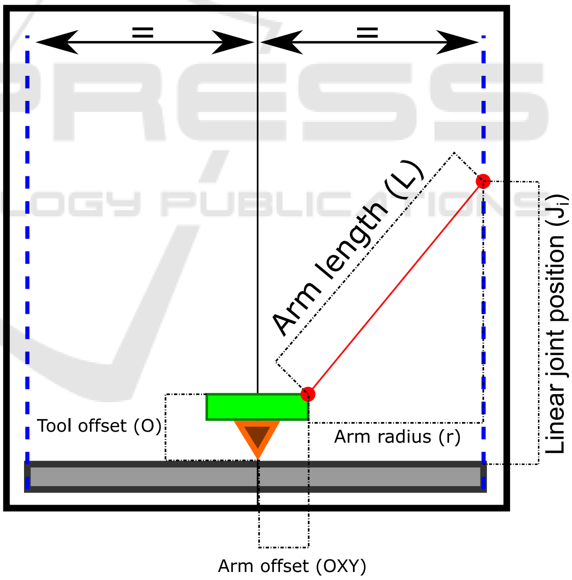

Figure 3: Effector arm of the delta kinematics machine.

Each of the three vertical linear joints is spread at an angle of 120 degrees forming an equilateral triangle with the origin in the center equidistant to each of the vertical joints. As shown in Fig.3, position in X and Y axes is determined by the size of the effector plate and arm angle. Z position is dependant on the height of effector plate above the zero-plane. Usually it is also required to include end effector or tool offset in the Z axis. The inverse kinematics equation for each linear joint is defined by the following formulas:

$$J_i = Z + \sqrt{L^2 - (X - r * C_i)^2 - (Y - r * S_i)^2} + O \tag{11}$$

$$C_i = \cos(\alpha_i), \quad S_i = \sin(\alpha_i) \tag{12}$$

where: J_i - linear joint positions, X, Y, Z - cartesian coordinates of end effector position, r - arm radius (distance between end effector plate and linear joint carriage in the x,y plane), L - arm length, O - vertical tool offset, $i = A, B, C$ - linear joint index, α_i - linear joint angle ($\alpha_A = 0°, \alpha_B = 120°, \alpha_C = 240°$)

4 S-CURVE FEEDRATE PLANNING

The jerk limited feedrate profile is one of most commonly used types of profiling in cnc control systems. A full profile consist of seven phases which are shown on fig.4. Depending on the distance between start and

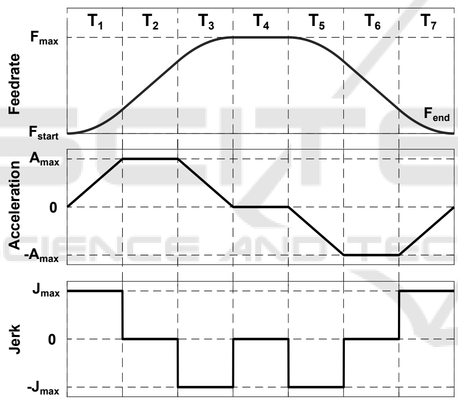

Figure 4: Example feedrate, acceleration and jerk profiles. $T_1 - T_7$ are the durations of each phase of the profile.

end point, start and end velocities (V_{start}, V_{end}), maximum feedrate, acceleration and jerk ($V_{max}, A_{max}, J_{max}$) the number of phases may be reduced. The algorithm to determine the duration of each phase is given below.

The duration of each phase of the S-curve profile is characterized by:

$$T_1 = T_3 = T_5 = T_7 = \frac{A_{max}}{J_{max}} \tag{13}$$

$$T_2 = \frac{V_{max} - V_{start}}{A_{max}} - T_1 \tag{14}$$

$$T_6 = \frac{V_5 - V_6}{A_{max}} \tag{15}$$

If $A_{max}^2 / J_{max} \geq V_{max} - V_{start}$ the total increase in feedrate in phases 1 and 3 is equal or greater than the difference between start and maximum feedrate. Phase 2 is omitted and acceleration is triangular instead of trapezoid. Analogously if $A_{max}^2 / J_{max} \geq V_{max} - V_{end}$ phase 6 is omitted. If phases 2 or 6 are omitted their duration is 0 and:

$$T_1 = T_3 = \sqrt{\frac{V_{max} - V_{start}}{J_{max}}} \tag{16}$$

$$T_5 = T_7 = \sqrt{\frac{V_{max} - V_{end}}{J_{max}}} \tag{17}$$

After determining the duration of phases 1,2,3,5,6,7 using the formulas given above the distance covered with the current profile S_{tot} is computed. If the distance is lower than the curve segment arc-length S_{seg} for which the profile is generated, phase 4 is added. The duration of phase 4 with constant maximum feedrate of V_{max} is $T_4 = V_{max} / (S_{seg} - S_{tot})$. Otherwise if the distance $S_{seg} - S_{tot}$ is negative the maximum feedrate has to be lowered and becomes the peak feedrate V_{peak}. The value of V_{peak} has to be determined numerically. Bisection search is performed between $V_a = V_{max}$ and $V_b = \max(V_{start}, V_{end})$ to find the value of V_{peak} for which the total distance is just below the segment distance. In each iteration the feedrate profile duration and distance are recomputed with V_{peak} instead of V_{max} until $S_{seg} - S_{tot} < V_{peak} \cdot \Delta t$.

When the NURBS toolpath exhibits areas of significant curvature the axial velocities, accelerations and jerks may vary significantly even if a constant feedrate is used. This is especially true in case of parallel kinematics machines. In order to decrease this effect a feedrate limit function can be utilized. Such a function combines axial velocity, acceleration and jerk limits into a single feedrate limit. Furthermore chord error limits can also be included. The chord error is the distance between an arc between two consecutive interpolation points and a line between those points. This approximates the interpolation error due to discrete interpolation of the curve. The feedrate limit function (FLF) can be expressed as (Sun et al., 2019; Zhao et al., 2013):

$$V_{chr}^{lim} = \frac{2}{\Delta t} \sqrt{\rho^2(u) - (\rho(u) - \varepsilon_{max})^2} \tag{18}$$

$$V_{acc}^{lim} = \sqrt{\frac{A_{max}}{\kappa(u)}} \tag{19}$$

$$V_{acc}^{lim} = \sqrt[3]{\frac{J_{max}}{\kappa^2(u)}} \tag{20}$$

$$\kappa(u) = \frac{1}{\rho(u)} = \frac{||C'(u) \times C''(u)||}{||C'(u)||^3} \tag{21}$$

$$FLF = \min\left(V_{max}, V_{chr}^{lim}, V_{acc}^{lim}, V_{acc}^{lim}\right) \tag{22}$$

where: κ - curvature, ρ - curvature radius, ε_{max} - maximum allowable chord error.

The FLF is used to determine critical points of the NURBS curve toolpath for which the feedrate has to be reduced taking into account the maximum allowable chord error, feedrate, centripetal acceleration and jerk. The NURBS curve is evaluated at constant intervals of the u parameter and the value of the FLF is computed at these points. The array of FLF value is scanned do determine coarse local minima of the FLF. A local minimum is a point for which its two neighbouring points have a higher value of the FLF. These coarse points are then refined using Brent's method to find a precise value of the FLF minimum and its corresponding value of parameter u. The process is performed for the whole curve and an array of curvature critical points is generated. The S-Curve feedrate profile is then planned between these critical points with the FLF values serving as initial and final feedrates. Cartesian axis limits are also applied by iteratively repeating the interpolation process with the s-curve segment and verifying if axial constraints are violated. If that is the case maximum acceleration and/or jerk maximum values are decreased and the profile is generated again. This is repeated until all axial constraints are satisfied. A detailed description of the algorithm used for a cartesian machine is given in (Erwinski et al., 2022). The algorithm will be further improved to include additional constraints, especially linear joint velocity, acceleration, constraints. This will require inclusion of the inverse kinematics in the feedrate planning process.

5 EXPERIMENTAL STATION

In order to perform real-time control and gather experimental data a PC-based control system for linear delta 3D printer was developed. A desktop PC based controller was used (Intel i3-6100T@3.2GHz, 8GB RAM) with Linux operating system (DEBIAN 11) with the kernel patched using the RT-Preempt real-time patch. This patch enables full kernel preemption which enables the real-time control processes to run at the highest priority. LinuxCNC open-source machine controller software, was installed to utilize its real-time APIs (HAL and RTAPI). This allowed to include custom feedrate planning and NURBS interpolation algorithms and run them in the real-time system. In order to interface the drives EtherCAT real-time communication bus stack was used based on the IGH EtherCAT master stack. Nanotec C5E servo-stepper drives were used with Nema17 stepper motors equipped with incremental encoders (4000 pulse/rev). The drives and motors worked in closed loop vector control mode. All devices were connected via Ethercat bus. CiA 402 state machine was used. Adequate software driver, that allowed for a cycle time of 250μs, was developed. Aforementioned driver also allowed to use HAL API with Nanotec drives. The block schematic of the experimental station is shown in Fig.5.

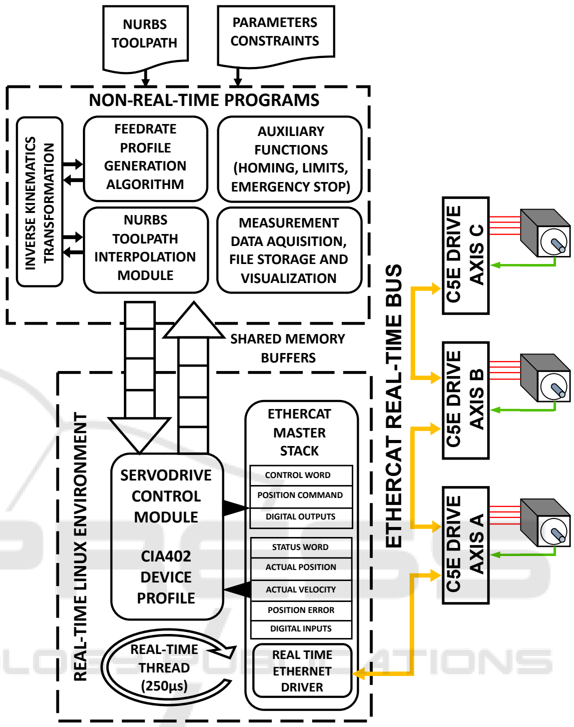

Figure 5: Block schematic of the experimental station's control software.

Machine used for testing purposes was all-aluminum KOSSEL linear delta 3D printer. The Kossel is a modular system for linear delta machines, that focuses on upgradeability and easy manipulation of workspace dimensions. This particular machine used GT2 toothed belts driven directly by the servo-stepper drives with linear guides (HIWIN MGN12) for increased stiffness. The mechanical parameters of the machine, are:

- arm length (L): 195mm
- arm radius (r): 65mm
- arm offset (Oxy): 30mm

The picture of the experimental station is shown in Fig.6.

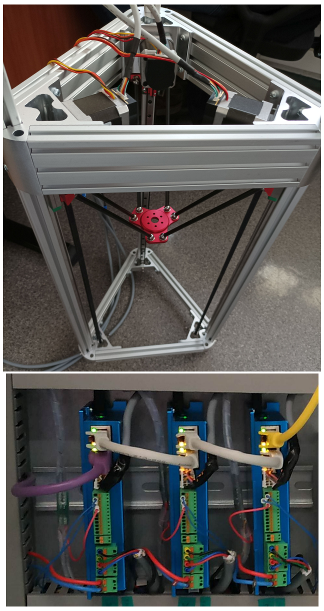

Figure 6: Picture of the Kossel machine used for experimental testing (top) with Nanotec C5E servo-stepper drives (bottom).

6 EXPERIMENTAL RESULTS

The feedrate profile was generated according to the feedrate planning algorithm described in Section 4 for the NURBS toolpath presented in Fig.1. Kinematic limits were set to $V_{max} = 100mm/s, A_{max} = 1000mm/s^2, J_{max} = 20000mm/s^3$. Maximum chord error was set to $\varepsilon = 0.001mm$. Figure 7 presents the generated feedrate profile along with. cartesian axis velocity, acceleration and jerk. The thick blue line on the first plot indicates the feedrate profile. The dashed lines indicate chord error, centripetal acceleration and jerk components of the Feedrate Limit Function (orange, green and violet respectively). The pink dots indicate the identified critical points between which the s-curve feedrate profiles were planned. The subsequent subplots show velocity, acceleration and jerk in each cartesian axis (blue - X, orange - Y, green - Z). The dashed black lines indicate the positive and negative limit of each value.

Fig.8 presents position, velocity, acceleration and jerk of the linear joints. Fig.9 presents actual following error of the linear joints.

It can be seen that the algorithm can limit the cartesian velocity, acceleration and jerk to their limits despite using the non-linear NURBS toolpath and an S-Curve feedrate profile. The actual joint limits are not taken into account at this moment. Further improvement of the feedrate generation will lead to inclusion of linear joint limits by applying

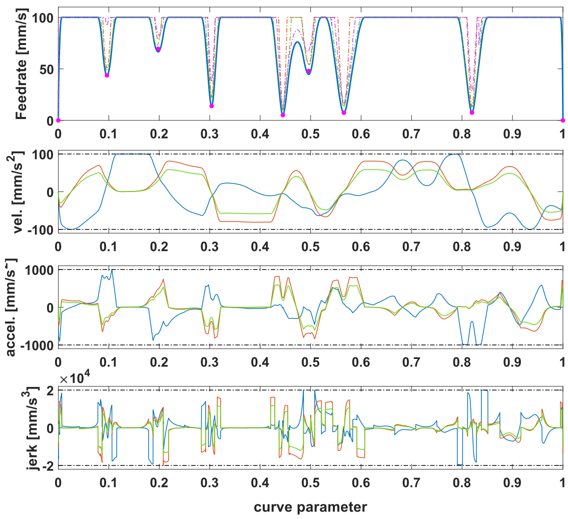

Figure 7: Feedrate profile with cartesian axis velocity, acceleration and jerk.

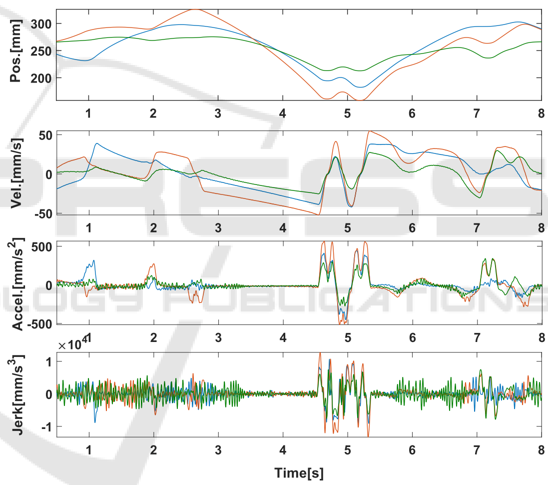

Figure 8: Actual linear joint position, velocity, acceleration and jerk recorded on the Delta Kossel machine.

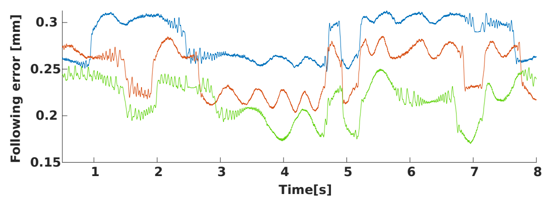

Figure 9: Actual following error of the linear joints recorded on the Delta Kossel machine.

the inverse kinematics transformation and optimizing the feedrate profile parameters to meet the limits.Furthermore additional constraints such as following and contouring error will be added. The experimental station will also be used for research of positioning error and torque ripple compensation methods using advanced controllers (Tarczewski et al., 2014),(Tarczewski and Grzesiak, 2009).

In order to verify the computational efficiency of the algorithm's implementation in the real-time system the feedrate generation process was repeated several hundred times and the total computation time was measured each time. The average computation time of the whole profile was 99.6ms with maximum and minimum time equal to 104ms and 97.6ms respectively. The total execution time of the toolpath from Fig.1 with the feedrate profile from Fig.7 was equal to 8.27s. This means that the proposed method has high computational effectiveness. The computation time of feedrate generation algorithm is much shorter than the toolpath execution time. This means that the system with the algorithm has on-line real-time capabilities. Even with the inclusion of additional constraints such as linear joint constraints the system should still retain its real-time capabilities.

7 CONCLUSIONS

This paper presents a method for jerk limited feedrate planning for Non-Uniform Rational B-Spline (NURBS) toolpaths in a parallel kinematics machine in linear delta configuration. The algorithm uses jerk-limited feedrate planning with limitation of of cartesian axes velocity, acceleration and jerk. The presented experimental results show that the algorithm can effectively constrain cartesian axis constraints. Further improvements of the algorithm will constrain velocity, acceleration and jerk in the linear joints. Furthermore the computation times show that the algorithm is computationally effective and is viable for real-time implementation. Future research will include improvement of the algorithm and its extension with additional constraints.

REFERENCES

Bouri, M. and Clavel, R. (2010). The linear delta: Developments and applications. In *ISR 2010 (41st International Symposium on Robotics) and ROBOTIK 2010 (6th German Conference on Robotics)*, pages 1–8. VDE.

Chen, M. and Sun, Y. (2019). Contour error–bounded parametric interpolator with minimum feedrate fluctuation for five-axis cnc machine tools. *Int. J. Adv. Manuf. Technol.*, 103(1-4):567–584.

Erwinski, K., Paprocki, M., and Karasek, G. (2021). Comparison of nurbs trajectory interpolation algorithms for high-speed motion control systems. In *2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC)*, pages 527–533. IEEE.

Erwinski, K., Wawrzak, A., and Paprocki, M. (2022). Real-time jerk limited feedrate profiling and interpolation for linear motor multi-axis machines using nurbs toolpaths. *IEEE Transactions on Industrial Informatics (early access, published online 1.02.2022)*.

Jia, Z.-Y., Song, D.-N., Ma, J.-W., Hu, G.-Q., and Su, W.-W. (2017). A nurbs interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. *Int. J. Mach. Tools Manuf.*, 116:1–17.

Li, G., Liu, H., Yue, W., and Xiao, J. (2021). Feedrate scheduling of a five-axis hybrid robot for milling considering drive constraints. *Int. J. Adv. Manuf. Technol.*, 112(11):3117–3136.

Lu, T.-C., Chen, S.-L., and Yang, E. C.-Y. (2018). Near time-optimal s-curve velocity planning for multiple line segments under axis constraints. *IEEE Trans. Ind. Electron.*, 65(12):9582–9592.

Mercy, T., Jacquod, N., Herzog, R., and Pipeleers, G. (2018). Spline-based trajectory generation for cnc machines. *IEEE Trans. Ind. Electron.*, 66(8):6098–6107.

Ni, H., Yuan, J., Ji, S., Zhang, C., and Hu, T. (2018a). Feedrate scheduling of nurbs interpolation based on a novel jerk-continuous acc/dec algorithm. *IEEE Access*, 6:66403–66417.

Ni, H., Zhang, C., Ji, S., Hu, T., Chen, Q., Liu, Y., and Wang, G. (2018b). A bidirectional adaptive feedrate scheduling method of nurbs interpolation based on s-shaped acc/dec algorithm. *IEEE Access*, 6:63794–63812.

Piegl, L. and Tiller, W. (1996). *The NURBS book*. Springer Science & Business Media.

Su, T., Cheng, L., Wang, Y., Liang, X., Zheng, J., and Zhang, H. (2018). Time-optimal trajectory planning for delta robot based on quintic pythagorean-hodograph curves. *IEEE Access*, 6:28530–28539.

Sun, Y., Chen, M., Jia, J., Lee, Y.-S., and Guo, D. (2019). Jerk-limited feedrate scheduling and optimization for five-axis machining using new piecewise linear programming approach. *Sci. China Technol. Sci.*, 62(7):1067–1081.

Tarczewski, T., Grzesiak, L., et al. (2014). Torque ripple minimization for pmsm using voltage matching circuit and neural network based adaptive state feedback control. In *2014 16th European Conference on Power Electronics and Applications*, pages 1–10. IEEE.

Tarczewski, T. and Grzesiak, L. M. (2009). High precision permanent magnet synchronous servo-drive with lqr position controller. *Electrical Review*, 85(8):42–47.

Zhang, Y., Ye, P., Zhao, M., and Zhang, H. (2019). Dynamic feedrate optimization for parametric toolpath with data-based tracking error prediction. *Mech. Syst. Signal Process.*, 120:221–233.

Zhao, H., Zhu, L., and Ding, H. (2013). A real-time look-ahead interpolation methodology with curvature-continuous b-spline transition scheme for cnc machining of short line segments. *Int. J. Mach. Tools Manuf.*, 65:88–98.