JERK LIMITED FEEDRATE PROFILE OPTIMIZATION FOR NURBS TOOLPATHS IN CNC MACHINES WITH H-BOT KINEMATICS

Krystian ERWIŃSKI*, Patryk USTASZEWSKI*

*Faculty of Physics Astronomy and Informatics, Institute of Engineering and Technology, Department of Automatics and Measurement Systems, Nicolaus Copernicus University in Torun, Grudziadzka 5/7, 87-100, Torun, Poland

erwin@umk.pl, patrykust@umk.pl

received 17 July 2025, revised 11 November 2025, accepted 24 November 2025

Abstract: This paper presents an algorithm for generating the feedrate profile for Computerized Numerically Controlled (CNC) machines with non-Cartesian H-Bot kinematics. Feedrate profile defines the relationship between the end-effector or tool velocity tangent to the toolpath and time or the toolpath's parameter. To maximize machining effectiveness the feedrate should be as high as possible. However, the feedrate, acceleration and jerk of the end-effector and individual machine axes should be within their respective limits. The toolpath is defined as a Non-Uniform Rational B-Spline (NURBS) polynomial curve to ensure smooth motion. This makes the problem more difficult for non-Cartesian machines due to non-linear dependencies between kinematic parameters of the end effector and the machine's axes. In this paper Particle Swarm Optimization (PSO) gradient free algorithm is used to determine the optimal shape of the feedrate profile to achieve shortest travel time within the velocity, acceleration and jerk constraints imposed by the machine's axes. Compared to more common approaches the profile is initialized with a near-optimal shape determined by a feedrate limit curve and then optimized to the final shape. The method is experimentally verified on an actual H-Bot gantry plotter. Presented results show that the proposed method effectively generates a feedrate profile within the imposed limits.

Key words: feedrate profile, optimization, NURBS, CNC, H-Bot kinematics

1. INTRODUCTION

Computerized Numerically Controlled CNC machines are the backbone of the manufacturing industry. There are many types of such machines, including milling machines, lathes, laser cutters and industrial 3D printers. Motion of the end effector (milling spindle, laser, print head), which affects the workpiece, is the result of the coordinated motion of individual machine axes. Each axis is driven by one or more electric motors controlled directly by the computerized numerical controller.

Gantry type CNC machines are usually used as milling plotters or cutters using laser, plasma or oscillating tangential knife. The end effector, such as a milling spindle is mounted on a horizontal beam mounted on two supports which move along the length of the workpiece table, called a portal. The beam also mounts the transverse axis which allows for lateral motion of the end effector. These types of machines are usually used for cutting or milling large sheets of metal, plastic, leather. Compared to traditional milling machines the gantry type offers large work space and higher feedrates.

Typically the gantry CNC is constructed as a machine with cartesian kinematics. Due to large work area, the beam is driven by two parallel drivetrains, usually based on a rack and pinion mechanism. The lateral axis also features a rack and pinion or ballscrew drive train depending on its length. The downside of this approach is that the axes have to drive both the mass of the beam, the rack and pinion mechanism as well as the motors with their gearboxes. The increased mass conflicts with the requirement of achieving high speeds and accelerations.

An alternative solution for a gantry CNC machine is the H-Bot gantry kinematic configuration. It has functionally the same applications – mainly cutting of sheet materials using laser, plasma, tangential knife or 2D milling. Both types of machines feature a portal beam riding on two parallel guideways on which a transverse axis is mounted. In the H-Bot machine the axes are actuated by a single toothed belt guided by a system of pulleys in the form of a letter H (hence the name). The belt is driven by two servo motors usually with reduction gearboxes. Contrary to the traditional cartesian gantries these motors are mounted on the machine frame and do not move with the axes. The axes do not require mounting of heavy rack and pinion mechanism and motor cable guides. This significantly decreases the weight driven by the machine motors. This can improve the machine dynamics as higher accelerations and velocities can be achieved with the same power of the motors. Compared to rack and pinion mechanism the toothed belt drive train features much less backlash when properly tensioned. Furthermore the H-Bot gantry requires only two motors, servo drives and gearboxes instead of three which are used in cartesian gantry CNC machines. In practice the H-bot gantry can replace the cartesian gantry especially when used in high speed cutting applications due to its lighter weight, higher dynamic capabilities and lower costs. A schematic comparing the cartesian and H-bot gantries is shown in (Fig. 1).

Feedrate profile defines the feedrate (end-effector velocity tangent to the toolpath) as a function of time or the toolpath parameter. Feedrate profile generation or feedrate planning is performed by taking into account limitations of the machines capabilities such as drive power, maximum drive speed, drivetrain parameters and machine dynamics [1].

DOI 10.2478/ama-2025-0075

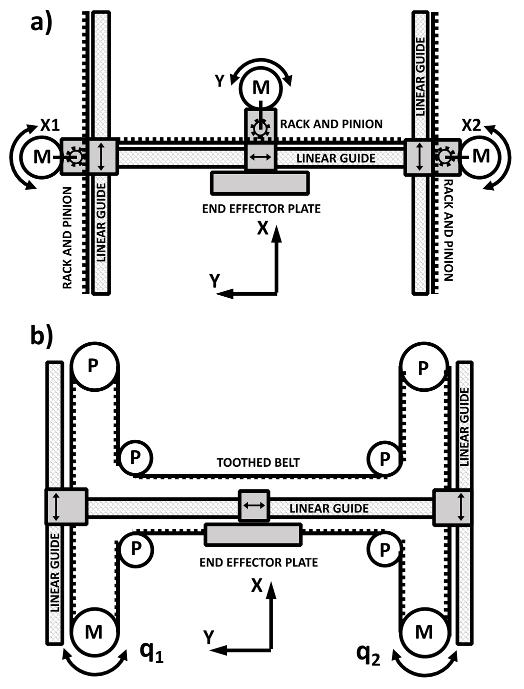

Fig. 1. Schematic of Cartesian gantry (a) and H-Bot gantry (b) CNC machines (M – motor, P-passive pulley, q1,q2 - H-bot motors, X1,X2,Y – cartesian gantry motors, X,Y-Cartesian axes)

In its simplest form, the feedrate profile limits only tangent velocity and acceleration, and the profile shape is a trapezoid [2]. In high-performance machines, the feedrate profile also limits jerk, which is the time derivative of acceleration. The feedrate profile is then represented by an S-Curve profile [3]. Jerk limitation causes the acceleration to change gradually, which in turn limits the rate of change of drive torque.

Many commercially available CNC machines use linear or circular segments for toolpath definition. Complex shapes are usually defined as a large number of short linear segments. Due to discontinuities, motion along such paths would result in discontinuous feedrate and acceleration, which in turn could cause vibration and large following errors. To avoid this, the paths are usually smoothed internally by the controller. Alternatively, in state-of-the-art solutions, polynomial paths are used as toolpath definitions. These are usually Non-Uniform Rational B-Splines (NURBS) as they are also used in CAD/CAM software to describe complex shapes. Toolpath definition as polynomial NURBS curves allows dynamic and smooth motion along complex paths.

To achieve high performance, feedrate profile generation with just feedrate, tangent acceleration and jerk limits may not be sufficient. When using NURBS toolpaths, due to their non-linear nature, the acceleration and jerk of the machine's axes may be violated even if tangent limits are kept. The problem of maximizing the feedrate while avoiding axial velocity, acceleration and jerk limit violations for all axes is a difficult problem, especially if the toolpath has highly variable curvature. This difficulty is increased if the machine kinematics is non-Cartesian due to an additional relationship between Cartesian and drive axes.

The solutions can be broadly divided into two categories. The indirect approach usually defines a Feedrate Limit Curve or Feedrate Limit Function, which combines axial velocity limits and linearised axial acceleration and jerk limits into a single feedrate limit. Then the feedrate profile is planned under the Feedrate Limit Curve, usually using S-Curve segments [4][9][19]. This approach is usually computationally efficient but may not achieve maximum performance or may violate axial limits. Some researchers suggest using a higher-order jerk continuous feedrate profile, which should further improve motion smoothness, especially for machines with flexible drive trains [20][21].

The direct approach solves the optimization problem using a constrained optimization algorithm. The feedrate profile is adjusted by the optimization algorithm to maximize the feedrate while not violating the axial velocity, acceleration and jerk limits [10][11][22]. This approach can usually achieve higher feedrates, which implies shorter manufacturing times, but at the cost of higher computational costs. The optimization approach is also much more capable in constraining axial velocity, acceleration and jerk.

In many cases the proposed methods consider jerk-limited feedrate profiles in which the jerk may be discontinuous. For lightweight low-stiffness machines and those which feature elasticity in the drive train, like the belt-driven H-bot, a higher order jerk-continuous feedrate profile may be preferable. In [20] a trigonometric-function-based jerk-continuous feedrate optimization method is proposed. In [23] a quartic feedrate profile is used. In [24] a snap limited linear feedrate optimization is used for an articulated robot. These methods do not consider axis limits or include them indirectly. Direct inclusion of axial limits and optimization of a jerk-continuous feedrate profile is usually a complex optimization problem usually requiring significant computational effort.

In most cases, the feedrate planning problem was investigated for 3-axis or 5-axis Cartesian machines. In this paper, the optimization approach is used to generate a jerk-continuous cubic feedrate profile considering axial limitations for a non-Cartesian H-Bot CNC machine with toolpaths defined as cubic NURBS curves. To improve feedrate generation computation, the feedrate profile is initially generated as an S-Curve and then converted to a cubic B-Spline representation. Optimization of the B-Spline is then performed so that the axial velocity, acceleration and jerk of the H-Bot machine are continuous and their limits are not violated. The Particle Swarm Optimization algorithm is used to optimize the shape of the profile, and the Augmented Lagrangian Method is used to handle axial constraints.

2. NURBS TOOLPATH INTERPOLATION FOR H-BOT

Non-Uniform Rational B-Splines (NURBS) are polynomial splines defined as:

$$\mathbf{C}(\mathbf{u}) = \frac{\sum_{i=1}^{k} \mathrm{N}_{i,n}(u)\omega_i \mathbf{P_i}}{\sum_{i=1}^{k} \mathrm{N}_{i,n}(u)\omega_i} \tag{1}$$

where: $\mathbf{C}(\mathbf{u})$ – point on the NURBS curve, $\mathrm{N}_{i,n}(u)$– basis functions, u – curve parameter, $\mathbf{P_i}$– control point, ω_i – control point weight, k – number of control points

The control points and their weights define the shape of the curve. An example of a NURBS toolpath with marked control points is shown in (Fig. 2). The unitless parameter u defines a unique position on the curve. By increasing the parameter from 0 (start of curve) to 1 (end of curve), the curve is traversed. For defining a toolpath of a CNC machine, a cubic curve is usually used, which has C2 parametric continuity, which means that the first and second derivatives of the curve are continuous. This guarantees that during motion along the curve, the feedrate and acceleration are continuous, and the tangent jerk is finite. In practice, evaluation of the NURBS curve is performed using the numerically stable de Boor's algorithm [6]. This method is also used to obtain the derivatives of the curve with respect to the curve parameter.

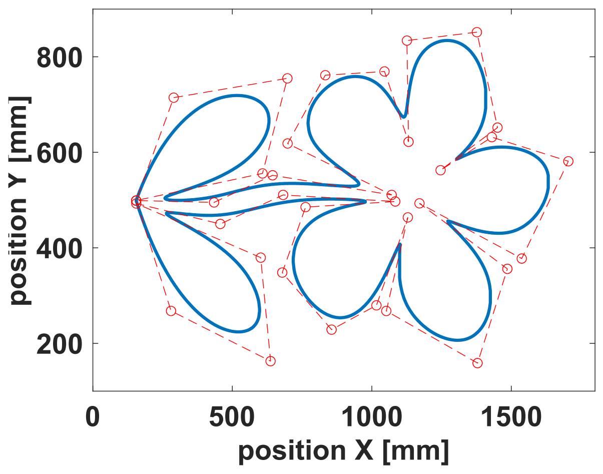

Fig. 2. NURBS toolpath (blue) with control points (red circles) and control polygon (red dashed)

Interpolation of the NURBS toolpath in CNC control systems is the process of determining consecutive positions $\mathbf{C}(\mathbf{u_i})$ on the curve by determining consecutive parameter values in constant time intervals. The increment of the curve parameter Δu_i should correspond to an increment of the curve arc-length Δs_i which in turn corresponds to the current value of feedrate $V_t(u_i)$. This feedrate is determined by the feedrate profile. An illustration of this process is presented in (Fig. 3).

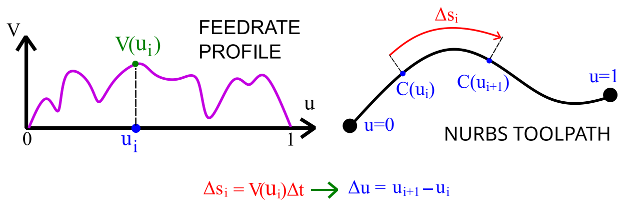

Fig. 3. NURBS toolpath interpolation process

The relationship between the parameter increment and arc length is non-linear and unique for each curve. During each interpolation step, this relationship has to be approximated. There are two main approaches to this problem. The first approach is to use a Taylor series expansion to determine consecutive values of the parameter u. This has the advantage of using a single closed-form formula and low computational complexity. The Taylor series method has the disadvantage of introducing feedrate fluctuations and interpolation errors if the feedrate is large and the curvature of the NURBS toolpath has high variations, such as sharp corners [8].

The second approach is a predictor-corrector method in which the initial parameter increment is verified by performing a numerical integration of the NURBS curve. The increment is then corrected so that the difference between the initial arc-length increment and the desired increment is minimised. This allows for interpolation of complex shapes with higher accuracy and better conformity to the desired feedrate profile [5]. In this paper, a second-order Runge-Kutta predictor-corrector (RK2-PC) method is used [7].

Interpolated Cartesian points on the curve are converted to axial position commands of the H-Bot using the inverse kinematics transformation:

$$\begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = \frac{G}{R}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix} \tag{2}$$

where: R – radius of the drive pulley in mm, G – motor gear ratio, q_1 , q_2 – position of the drive axes, *x,y* – Cartesian position of the end effector.

In order to perform feedrate planning with axial limts, axial velocity, acceleration and jerk have to be determined based on feedrate, tangential acceleration and tangential jerk:

$$\mathbf{V_c} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{C_s} \cdot V_t \tag{3}$$

$$\mathbf{A_c} = \begin{bmatrix} a_x \\ a_y \end{bmatrix} = \mathrm{C_{ss}} \cdot V_t^2 + \mathrm{C_s} \cdot A_t \tag{4}$$

$$\mathbf{J_c} = \begin{bmatrix} j_x \\ j_y \end{bmatrix} = \mathrm{C_{sss}} \cdot V_t^3 + 3\mathrm{C_{ss}} \cdot V_t A_t + \mathrm{C_s J_t} \tag{5}$$

where: $\mathbf{C_s}$, $\mathbf{C_{ss}}$, $\mathbf{C_{sss}}$ are first, second and third derivative of the NURBS curve with respect to the curve arc length,
$\mathbf{V_c}$, $\mathbf{A_c}$, $\mathbf{J_c}$ are vectors of velocity, acceleration and jerk along Cartesian axes. To determine toolpath derivatives with respect to arc-length the following formulas are used:

$$\mathbf{C_s} = \mathbf{C}' u_s \tag{6}$$

$$\mathbf{C_{ss}} = \mathrm{C}''\mathrm{u_s^2} + \mathrm{C}'\mathrm{u_{ss}} \tag{7}$$

$$\mathbf{C_{sss}} = \mathbf{C}'''\mathrm{u_s^3} + 3\mathbf{C}''\mathrm{u_{ss}u_s} + \mathbf{C}'\mathrm{u_{sss}} \tag{8}$$

where: u_s , u_{ss}, u_{sss} – first, second and third derivatives of the toolpath curve parameter with respect to the arc-length.
$\mathbf{C}'$, $\mathbf{C}''$, $\mathbf{C}'''$ – first, second and third derivatives of the NURBS curve with respect to the curve parameter.
Parameter derivatives with respect to arc-length are defined as:

$$u_s = \frac{1}{|C'|} \tag{9}$$

$$u_{ss} = -C' \cdot C'' u_s^4 \tag{10}$$

$$u_{sss} = 4(C' \cdot C'')^2 u_s^7 - \left(C''^2 + C' \cdot C'''\right) u_s^5 \tag{11}$$

Axial velocity, acceleration and jerk can be expressed as:

$$\mathbf{V_q} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \boldsymbol{\mathcal{J}^{-1}} \cdot \mathbf{V_c} \tag{12}$$

$$\mathbf{A_q} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \boldsymbol{\mathcal{J}^{-1}} \cdot \mathbf{A_c} + \boldsymbol{\dot{\mathcal{J}}^{-1}} \cdot \mathbf{V_c} \tag{13}$$

$$\mathbf{J_q} = \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} = \boldsymbol{\mathcal{J}^{-1}} \cdot \mathbf{J_c} + \mathbf{2}\boldsymbol{\dot{\mathcal{J}}^{-1}} \cdot \mathbf{A_c} + \boldsymbol{\ddot{\mathcal{J}}^{-1}} \cdot \mathbf{V_c} \tag{14}$$

where: $\boldsymbol{\mathcal{J}^{-1}}$ is the inverse Jacobian matrix and $\boldsymbol{\dot{\mathcal{J}}^{-1}}, \boldsymbol{\ddot{\mathcal{J}}^{-1}}$ are its first and second time derivatives. The Jacobian is the matrix of partial derivatives of axial positions *q1, q2* with respect to Cartesian positions *x,y*.
The final formulas that of axial velocities, accelerations and jerks are expressed as:

$$\mathbf{V_q} = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{G}{R}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{15}$$

$$\mathbf{A_q} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \frac{G}{R}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a_x \\ a_y \end{bmatrix} \tag{16}$$

$$\mathbf{J_q} = \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} = \frac{G}{R}\begin{bmatrix} -1 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} j_x \\ j_y \end{bmatrix} \tag{17}$$

The formulas described above are used to compute axial kinematic limit violations during feedrate profile optimization.

3. FEEDRATE OPTIMIZATION ALGORITHM

Feedrate optimization is performed in two phases. In the first phase, a jerk-limited S-Curve feedrate profile is generated with maximum allowable feedrate, tangential acceleration and tangential jerk. The initial polynomial profile is then converted to a B-Spline form defined by a set of control points, similar to the NURBS toolpath. Optimization is performed, using the Particle Swarm Optimization (PSO) algorithm, which maximizes the feedrate while taking into account constraints on axial velocity, acceleration and jerk. A block schematic of the feedrate optimization process is presented in (Fig. 4).

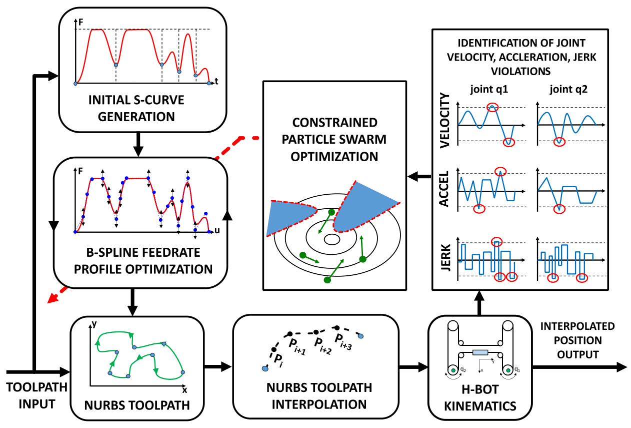
F
t
INITIAL S-CURVE GENERATION
F
u
B-SPLINE FEEDRATE PROFILE OPTIMIZATION
CONSTRAINED PARTICLE SWARM OPTIMIZATION
IDENTIFICATION OF JOINT VELOCITY, ACCLERATION, JERK VIOLATIONS
joint q1
joint q2
VELOCITY
ACCEL
JERK
TOOLPATH INPUT
y
x
NURBS TOOLPATH
P_i
P_{i+1}
P_{i+2}
P_{i+3}
NURBS TOOLPATH INTERPOLATION
H-BOT KINEMATICS
INTERPOLATED POSITION OUTPUT

Fig. 4. Block schematic of the feedrate optimization algorithm

3.1. S-Curve initial guess

S-curve feedrate profile, presented in (Fig. 5) consists of 7 second-order polynomial segments with trapezoidal acceleration profile and discontinuous jerk profile.

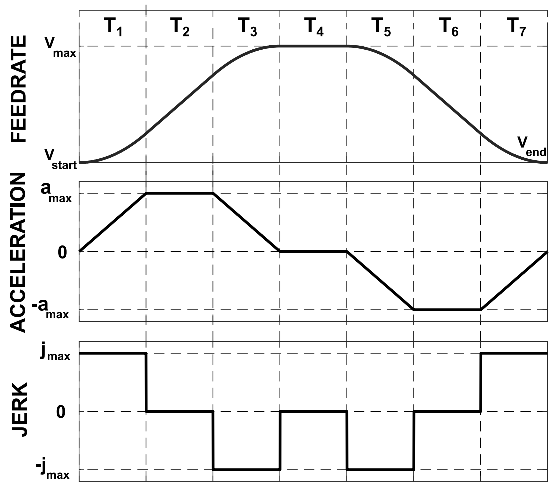

Fig. 5. S-curve jerk limited feedrate profile (T_1-T_7 – durations of each profile segment)

The profile can be mathematically defined as a function of time:

$$V(t) = \begin{cases} J_{max}t^2 + V_{start} & 0 \geq t \geq \tau_1 \\ A_1 t + V_1 & \tau_1 > t \geq \tau_2 \\ -\frac{1}{2}J_{max}t^2 + A_2 t + V_2 & \tau_2 > t \geq \tau_3 \\ V_{max} & \tau_3 \geq t > \tau_4 \\ -\frac{1}{2}J_{max}t^2 + V_4 & \tau_4 > t \geq \tau_5 \\ A_5 t + V_5 & \tau_5 > t \geq \tau_6 \\ \frac{1}{2}J_{max}t^2 + A_6 t + V_6 & \tau_6 > t \geq \tau_7 \end{cases} \tag{18}$$

where: V_{max}, A_{max}, J_{max} – maximum feedrate, tangential acceleration and jerk, $V_{1\text{-}6}$ – feedrate at the end of each phase, V_{start} – initial feedrate, $A_{1\text{-}6}$ – acceleration at the end of each phase, $\tau_{1\text{-}7}$ – time at the end of each phase

To plan an S-curve feedrate profile on a certain distance S_{seg}, the distance traversed during each phase is first computed assuming maximum feedrate, acceleration and jerk. The distance in the constant feedrate phase is adjusted so that the sum of distances of each phase matches the desired displacement.

$$\begin{cases} S_1 = \frac{1}{6}J_{max}T_1^3 + V_{start}T_1 \\ S_2 = \frac{1}{2}A_1 T_2^2 + F_1 T_2 \\ S_3 = -\frac{1}{6}J_{max}T_3^3 + \frac{1}{2}A_2 T_3^2 + F_2 T_3 \\ S_5 = -\frac{1}{6}J_{max}T_5^3 + V_{max}T_5 \\ S_6 = -\frac{1}{2}A_5 T_6^2 + V_5 T_6 \\ S_7 = \frac{1}{6}J_{max}T_7^3 - \frac{1}{2}A_6 T_7^2 + V_6 T_7 \\ S_4 = S_{seg} - (S_1 + S_2 + S_3 + S_5 + S_6 + S_7) \end{cases} \tag{19}$$

where: $S_{1\text{-}7}$ – displacement during each phase, $T_{1\text{-}7}$ – duration of each phase, which is computed as:

$$\begin{cases} T_1 = T_3 = T_5 = T_7 = \frac{A_{max}}{J_{max}} \\ T_2 = \frac{V_{max} - V_{start}}{A_{max}} - T_1 \\ T_6 = \frac{V_5 - V_6}{A_{max}} \end{cases} \tag{20}$$

To plan the S-Curve feedrate profile for a NURBS toolpath, the critical points of the NURBS curve have to be identified. These points mark fragments of the path with curvature maxima. At these points, the feedrate profile should achieve a minimum to improve interpolation and tracking accuracy. To select the critical points, a chord error feedrate limit function is defined for each sampling point[12]:

$$v_{chord}(u_i) = max\left(\frac{2}{\Delta t}\sqrt{\rho^2(u_i) - (\rho(u_i) - \epsilon_{max})^2}, V_{max}\right) \tag{21}$$

$$\kappa(u_i) = \frac{1}{\rho(u_i)} = \frac{||C'(u_i) \times C''(u_i)||}{||C'(u_i)||^3} \tag{22}$$

where: $\kappa(u_i)$, $\rho(u_i)$ – curvature and curvature radius for parameter u_i at sample point i,, ϵ_{max} – maximum allowable chord error, V_{max} – maximum allowable feedrate, i – index of current feedrate sampling point.

Chord error is the distance between a line connecting consecutive interpolation points and the actual curve approximated locally by a circle with a radius equal to the local curvature radius of the curve. Chord error is the measure of interpolation accuracy and

$v_{chord}(u_i)$ determines the maximum feedrate at each sampling point between start and end of the curve, for which the chord error limit is not violated. Minima of the chord error feedrate limit function determine sections of the NURBS toolpath between which S-Curve feedrate profiles are generated. Values of the v_{chord} function determine initial and final velocities of the S-curve segment.

The S-curve segments are planned for each segment with maximum predefined feedrate, tangential acceleration and jerk so that the distance covered by each profile segment (S_{seg}) is equal to the arc-length distance between critical points. If the distance between the critical points is too short to fit the full s-curve, the maximum feedrate in that segment is iteratively decreased until the profile displacement fits the arc-length distance between the critical points.

3.2. Feedrate profile optimization using PSO

Feedrate optimization is performed on a feedrate profile in the form of a B-Spline curve. The B-spline is similar to the NURBS curve but without weights. The B-Spline is defined as follows:

$$\mathbf{C(u)} = \sum_{\mathrm{i=1}}^{\mathrm{k}} \mathrm{N_{i,n}(u)}\, \mathbf{P_i} \tag{23}$$

The S-curve polynomial feedrate profile, generated in the previous step, is converted to a one-dimensional cubic B-Spline representation. The feedrate profile is reparametrized from a function of time to a function of the NURBS toolpath parameter.
This procedure is performed in the following steps:

- For each segment delimited by the critical points of the NURBS curve the initially planned S-curve feedrate profile is sampled and the with equal time intervals. Ten samples per segment were used with the number being determined by trial and error to balance accuracy and computational complexity.
- The toolpath is interpolated and for each sampled feedrate profile value the corresponding value of NURBS curve parameter u is also sampled.
- Using the curve parameter and feedrate samples recorded in the previous step, a cubic B-spline curve is fitted in that segment using least-squares. The mean squared error between sampled feedrate points and value of the B-Spline evaluated for corresponding parameter values are minimised [6]. The endpoints of the fitted segment are equal to the critical points at the start and end of the segment.
- The above steps are repeated for each segment and the final feedrate profile consists of a joined B-spline parametrized with the NURBS toolpath parameter.
- The reparametrized B-Spline profile is returned as a series of control points

The optimization of cubic B-Spline feedrate profile is performed by adjusting the control point values, which are treated as optimization variables.

The objective function, can be formulated as:

$$\mathbf{f}(\boldsymbol{P}) = \frac{1}{N}\sum_{\mathrm{i=1}}^{\mathrm{N}}\left(1 - \frac{\boldsymbol{P_i}}{V_{max}}\right) \tag{24}$$

where: N – number of optimized control points, P_i – control point, i – control point index, V_{max} – maximum allowable feedrate.
The objective function maximizes the normalized sum of control points so that the feedrate is as high as possible.

The optimization problem is subject to constraints of maximum axial velocity, acceleration and jerk:

$$c_{v(1,2)} = \frac{|v_{(1,2)}(u_i)|}{v_{max}} - 1 \tag{25}$$

$$c_{a(1,2)} = \frac{|a_{(1,2)}(u_i)|}{a_{max}} - 1 \tag{26}$$

$$c_{j(1,2)} = \frac{|j_{(1,2)}(u_i)|}{j_{max}} - 1 \tag{27}$$

Where: $c_{v(1,2)}, c_{a(1,2)}, c_{j(1,2)}$ - normalized constraint violation values for velocity, acceleration and jerk in each axis (1 or 2). These constraints are computed based on axial velocity, acceleration and jerk computed using equations (16-18). $v_{max}, a_{max}, j_{max}$ – maximum allowable value of axial velocity, acceleration and jerk.

Minimisation of the objective function is performed using the widely proven Particle Swarm Optimization algorithm (PSO)[14]. PSO is a swarm intelligence, nature-inspired optimization method in which virtual particles (candidate solutions) move through the search space and converge on the minimum of the objective function, which is indicated by the "global best" position (last position with the best fitness so far). In each iteration, the velocities and positions of each particle (solutions) are updated according to the following equations [13]:

$$v_{i,j}^{*} = \omega \cdot v_{i,j} + \phi_1 \cdot rand_{(0,1)} \cdot (p_{i,j} - x_{i,j}) + \phi_2 \cdot rand_{(0,1)} \cdot (g_i - x_{i,j}) \tag{28}$$

$$x_{i,j}^{*} = x_{i,j} + v_{i,j}^{*} \tag{29}$$

Where: $x_{i,j}$ - position of particle j in step i, $v_{i,j}$ - position of particle j in step i, $p_{i,j}$ – personal best position of particle j in step i, g_i- global best position at iteration i, $\omega, \phi_1, \phi_2,$ - parameters of the algorithm (inertia weight, cognitive coefficient, social coefficient), $rand_{(0,1)}$ - random number between 0 and 1 drawn from a uniform distribution.

The standard Particle Swarm Optimization method does not handle arbitrary constraints. Only the optimization variables are limited. To handle axial velocity, acceleration and jerk constraints, such defined by equations (23-25), the method has to be extended with some constraint handling technique. There are several constraint handling techniques such as Deb's method, Penalty method and Augmented Lagrangian Method [18].

The Augmented Lagrangian Method, used in this paper, adds a penalty factor to the objective function which is dependent on the amount of constraint violation. This augmented objective function creates the Augmented Lagrangian which is defined as [15]:

$$L(\boldsymbol{P}) = f(\boldsymbol{P}) + \frac{\rho}{2}\sum_{i=1}^{M}\left(\max\left\{0, c_i(\boldsymbol{P}) + \frac{\lambda_i}{\rho}\right\}\right)^2 \tag{30}$$

where: $f(\boldsymbol{P})$- objective function value for current solution vector $\boldsymbol{P}$, ρ - penalty factor, $c_i(\boldsymbol{P})$ - value of constraint *i* for current solution vector $\boldsymbol{P}$, λ_i - Lagrange multiplier for constraint *i*, *M* - number of constraints.

The Augmented Lagrangian is minimised by the Particle Swarm Optimization several times. After each minimisation, the Lagrange multipliers and penalty factor are updated to find their optimal value. This leads to the solution of the unconstrained problem that minimises the augmented Lagrangian to be equivalent to the solution of the original constrained problem [17].

The update rules of the penalty factor and Lagrange multipliers are defined as [15]:

$$\lambda_i^{*} = min\{\max\{0, \lambda_i + \rho c_i(\boldsymbol{P})\}, 10^{20}\} \tag{31}$$

DOI 10.2478/ama-2025-0075

$$\rho_i^* = \begin{cases} 10\rho & ICM^* > 0.5 * ICM \\ \rho & ICM^* \leq 0.5 * ICM \end{cases} \tag{32}$$

$$ICM^* = \max\left\{ICM, \left|\max\left\{c_i(\boldsymbol{P}), -\frac{\lambda_i}{\rho}\right\}\right|\right\} \tag{33}$$

The Infeasibility Compatibility Measure (ICM) is the measure of constraint violation. The idea behind the update method is to increase the penalty factor and Lagrange multipliers if the ICM increases which should lead to the elimination of the constraint violation in subsequent runs of the optimization. The result of each augmented Lagrangian minimisation is used as an initial guess for the next cycle.

Optimization of the B-Spline feedrate profile is performed in a moving window approach. A constant number of control points is optimized during each run, equal to 50. Each window has an overlap o 10 control points with the previous window. The first and last control point in the window is not modified. In each optimization window the constraints are evaluated at equally spaced constraint evaluation points (CEP) with the distance set to 0.5mm.

Initialization of the Particle Swarm Optimization algorithm is usually done by randomizing the swarm in the whole search space. In this approach the pre-generated feedrate profile is used to initialize one particle to exactly match the initial solutions. Other particle positions are randomized around that initial solution. This narrows down the search space and improves convergence speed.

The algorithm parameters suggested in [13], commonly used in literature and in this paper are: $\omega = 0{,}7298$, $\phi_1 = \phi_2 = 1{,}4926$. Other important parameters include the number of particles and termination conditions. The termination conditions are usually the number of iterations or function evaluations. The number of particles was chosen to be 25 and the termination criteria for a single PSO optimization is chosen as 200 iterations. The number of maximum augmented Lagrangian optimizations is chosen as 25.

4. EXPERIMENTAL VERIFICATION OF THE FEEDRATE GENERATION ALGORITHM

4.1. Experimental setup

The experimental setup, used to experimentally verify the proposed feedrate optimization algorithm, consists of an H-Bot gantry numerically controlled machine, shown in (Fig. 6), with a PC-based numerical controller. The CNC machine is driven by Delta ASDA-A3-E 750 W servo drives with permanent magnet servo motors. The HTD toothed belt is actuated by pulleys driven by the servo motors with 1:5 reduction gearboxes. The drives are controlled by the PC-based controller using the EtherCAT communication bus.

The PC based controller is implemented on an industrial PC with the Linux operating system with RT-Preempt real-time kernel. An open-source IGH EtherCAT Master stack is implemented as a real-time module to communicate with the servo drives. The communication is conducted according to the CAN in Automation (CiA) 402 standard drive profile. The real-time thread cycle that governs all real-time models, including the communication cycle, was set to 250µs.

The feedrate optimization algorithm is implemented as a non-real-time user program. The generated optimized feedrate profile is then sent to the real-time NURBS interpolator. The interpolator generates the consecutive position commands to the drives in real-time during machine operation. The data such as actual position, speed, current etc. is collected from the drives and linear encoders in real-time and saved to file through a FIFO buffer. The data can be later analysed in tools such as Matlab. The experimental setup is described in more detail in [16]. A block schematic of the experimental setup's control system is presented in (Fig. 7).

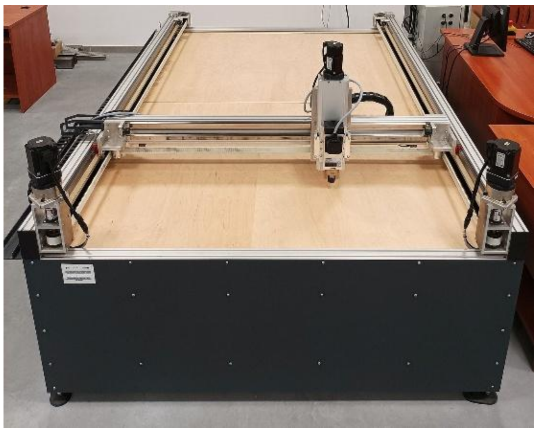
Fig. 6. Picture of the experimental setup with the H-Bot CNC machine

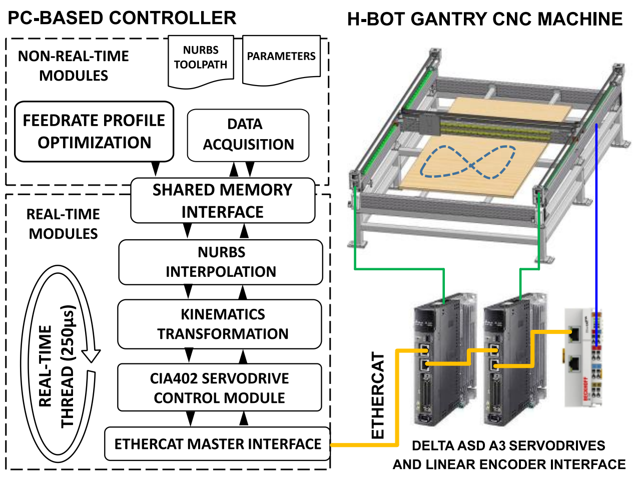

Fig. 7. Block schematic of the H-Bot control system

4.2. Experimental results

In order to experimentally verify the proposed method an example NURBS toolpath ("flower") was used (Fig. 2) which has a total length of 7001.5 mm. First an initial guess s-curve was generated with limits of feedrate, tangent acceleration and tangent jerk were set to: 1250 mm/s, 12000 mm/s2 and 120000 mm/s3 respectively. The generated initial S-curve profile is shown in (Fig. 8).

When generating the initial profile, the axial velocity, acceleration and jerk limits are not taken into account directly, but can be limited indirectly by limiting feedrate, tangent acceleration and jerk. The maximum feedrate was set so that the axial velocity limits are within the nominal motor velocity (314 rad/s or 3000 rev/min). Expected axial acceleration and jerk limits are 1700 rad/s2 and 51500 rad/s3. The acceleration limit serves to limit drive current to its nominal value of 5A. Jerk limit serves to limit vibration and is set experimentally. Acceleration and jerk limits are occasionally violated, which could result in motor overloads, following error spikes or excessive vibration. The generated trajectory run time is 7.821s.

The generated initial feedrate profile was converted to a cubic B-Spline and input to the feedrate optimization algorithm. The obtained profile had 402 control points. The optimization algorithm was then run on the B-spline profile. The trajectory execution time is 7.667s while the average optimization time from 10 separate runs is 76.1s. The resulting trajectory was executed on the H-Bot CNC machine using the "flower" toolpath. The toolpath with indicated feedrate is shown in (Fig. 9). The optimal feedrate profile with axial velocity, acceleration and jerk for both axes is shown in (Fig.10). Measured drive current and following errors for both axes are shown in (Fig.11).

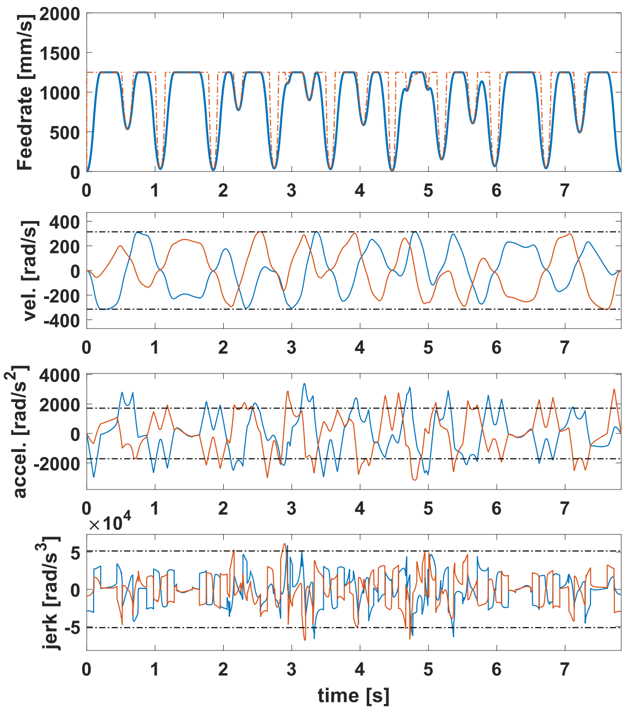

Fig. 8. Initial S-curve Feedrate profile with chord error feedrate limit (dashed line) and axial velocity, acceleration and jerk profiles for both axes - "flower" NURBS toolpath

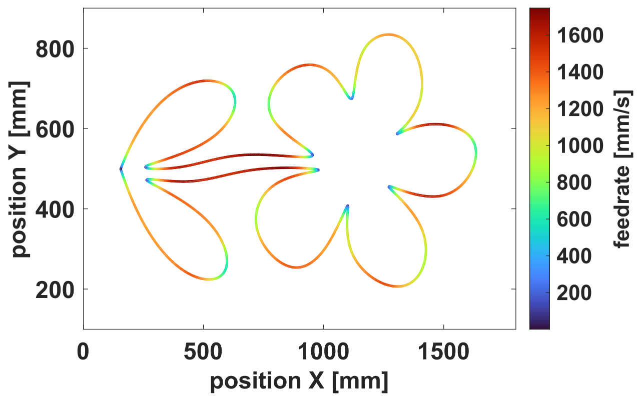

Fig. 9. "Flower" NURBS toolpath. The curve color indicates the feedrate in mm/s

It can be seen that the generated profile does not violate axial constraints despite the feedrate profile reaching higher values than the initial 1250mm/s. The resulting trajectory also has a shorter runtime than the initial trajectory. The axial acceleration and jerk could be limited indirectly by limiting tangent acceleration and jerk, but this would result in sub-optimal execution times and lower productivity. Replacing the standard S-Curve with a cubic B-Spline feedrate profile caused the jerk to be continuous, which improves motion smoothness and reduces potential vibration. The drive current is within nominal values and the following error is bounded.

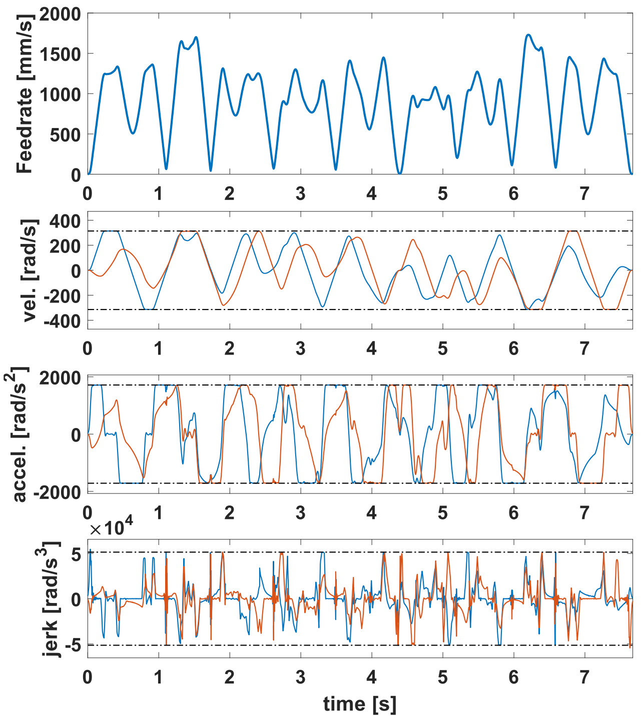

Fig. 10. Feedrate profile and axial velocity, acceleration and jerk profiles for both axes - "flower" NURBS toolpath

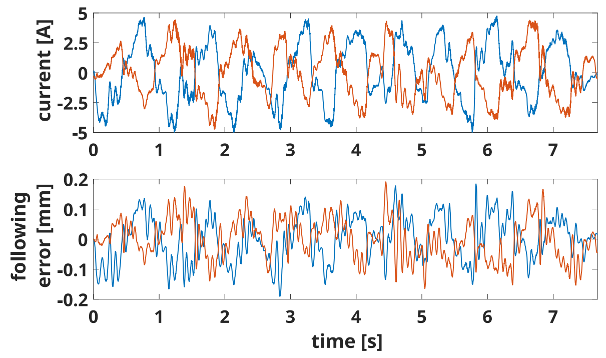

Fig. 11. Drive current and following error in X and Y axes.

5. CONCLUSION

This paper presents a feedrate optimization algorithm for a CNC machine with non-Cartesian H-Bot kinematics and Non-Uniform Rational B-Spline (NURBS) toolpaths. The feedrate profile is optimized using the Particle Swarm Optimization (PSO) algorithm. To speed up convergence and improve reliability, the feedrate profile is first planned as an S-Curve and then converted to a B-Spline curve. The B-Spline profile is then adjusted by the PSO algorithm with the Augmented Lagrangian constraint handling technique to maximize feedrate.

Experimental results show that the velocity, acceleration and jerk in the machine axes are within their prescribed limits. Compared to the initial profile, the trajectory execution time was reduced, and no constraint violations were observed. Axial velocity, acceleration and jerk limitation also limit the drive current to nominal values and can serve to indirectly limit the following error.

Future research will focus on directly handling following and contour error constraints and extension of the proposed method to other non-Cartesian kinematics.

REFERENCES

1. Suh SH, Kang SK, Chung DH, Stroud I. Theory and Design of CNC Systems. Springer Series in Advanced Manufacturing. London: Springer London; 2008. https://doi.org/10.1007/978-1-84800-336-1
2. Timar SD, Farouki RT. Time-optimal traversal of curved paths by Cartesian CNC machines under both constant and speed-dependent axis acceleration bounds. Robotics and Computer-Integrated Manufacturing. 2007;23(5):563–79. https://doi.org/10.1016/j.rcim.2006.07.002
3. Rong ZX, Xie L, Cong XL, Dao SD. Adaptive parametric interpolation scheme with limited acceleration and jerk values for NC machining. International journal, advanced manufacturing technology. 2006;36(3-4):343–54. https://doi.org/10.1007/s00170-006-0834-6
4. Erwinski K, Wawrzak A, Paprocki M. Real-time jerk limited feedrate profiling and interpolation for linear motor multiaxis machines using NURBS toolpaths. IEEE Transactions on Industrial Informatics. 2022; 18(11):7560-7571. https://doi.org/10.1109/TII.2022.3147806
5. Erwinski K, Paprocki M, Karasek G. Comparison of NURBS trajectory interpolation algorithms for high-speed motion control systems. 2021 IEEE 19th International Power Electronics and Motion Control Conference (PEMC); 2021. https://doi.org/10.1109/PEMC48073.2021.9432561
6. Piegl L, Tiller W. The NURBS book. Berlin: Springer; 1997.
7. Zhang J, Song DN, Ma J, Hu G, Sui W. A NURBS interpolator with constant speed at feedrate-sensitive regions under drive and contour-error constraints. International Journal of Machine Tools & Manufacture. 2017;116:1–17. https://doi.org/10.1016/j.ijmachtools.2016.12.007
8. Ni H, Zhang C, Chen C, Hu T, Liu Y. A parametric interpolation method based on prediction and iterative compensation. International Journal of Advanced Robotic Systems. 2019;16(1): 172988141982818-172988141982818. https://doi.org/10.1177/1729881419828188
9. Zhao H, Zhu L, Ding H. A real-time look-ahead interpolation methodology with curvature-continuous B-spline transition scheme for CNC machining of short line segments. International Journal of Machine Tools and Manufacture. 2013;65:88–98. https://doi.org/10.1016/j.ijmachtools.2012.10.005
10. Lu TC, Chen S, Eileen Chih-Ying Yang. Near Time-Optimal S-Curve Velocity Planning for Multiple Line Segments Under Axis Constraints. IEEE Transactions on Industrial Electronics. 2018;65(12):9582–92. https://doi.org/10.1109/TIE.2018.2818669
11. Sun Y, Chen M, Jia J, Lee YS, Guo D. Jerk-limited feedrate scheduling and optimization for five-axis machining using new piecewise linear programming approach. Science China Technological sciences/Science China Technological Sciences. 2019;62(7):1067–81. https://doi.org/10.1007/s11431-018-9404-9
12. Annoni M, Bardine A, Campanelli S, Foglia P, Prete CA. A real-time configurable NURBS interpolator with bounded acceleration, jerk and chord error. Computer-Aided Design. 2012;44(6):509–21. https://doi.org/10.1016/j.cad.2012.01.009
13. Clerc M, Kennedy J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation. 2002;6(1):58–73. https://doi.org/10.1109/4235.985692
14. Eberhart RC, Yuhui Shi, Kennedy JF. Swarm intelligence. Morgan Kaufmann; 2001.
15. Birgin EG, Martínez JM. Improving ultimate convergence of an augmented Lagrangian method. Optimization methods & software. 2008;23(2):177–95. https://doi.org/10.1080/10556780701577730
16. Paprocki M, Erwinski K, Zivanovic S. CNC machine laboratory stand with H-Bot parallel kinematics using the EtherCAT bus. 43rd JUPITER conference, 39th symposium NC-Robots-FTS : Proceedings, 2022, 3.39-3.46. https://machinery.mas.bg.ac.rs/handle/123456789/4637
17. Sedlaczek K, Eberhard P. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering. Structural and Multidisciplinary Optimization. 2006;32(4):277–86. https://doi.org/10.1007/s00158-006-0032-z
18. Szczepanski R, Tomasz Tarczewski, Krystian Erwinski, Grzesiak LM. Comparison of Constraint-handling Techniques Used in Artificial Bee Colony Algorithm for Auto-Tuning of State Feedback Speed Controller for PMSM. Proceedings of the 15th International Conference on Informatics in Control, Automation and Robotics; 2018.
19. Sang Y, Yao C, Yiqian Lv, He G. An improved feedrate scheduling method for NURBS interpolation in five-axis machining. Precision Engineering. 2020;64:70–90. https://doi.org/10.1016/j.precisioneng.2020.03.012
20. Nie M, Zou L, Zhu T. Jerk-Continuous Feedrate Optimization Method for NURBS Interpolation. IEEE Access. 2023;11:25664–81. https://doi.org/10.1109/ACCESS.2023.3248081
21. Fang Y, Qi J, Hu J, Wang W, Peng Y. An approach for jerk-continuous trajectory generation of robotic manipulators with kinematical constraints. Mechanism and Machine Theory. 2020;153:103957. https://doi.org/10.1016/j.precisioneng.2020.03.012
22. Erwinski K, Szczepanski R, Tarczewski T. Nature inspired optimization of jerk limited feedrate profile for NURBS toolpaths in CNC machines. IOP Conference Series: Materials Science and Engineering. 2021;1140(1):012031. https://iopscience.iop.org/article/10.1088/1757-899X/1140/1/012031
23. Ren X, Fan J, Pan R. A novel and efficient jerk-smooth feedrate scheduling algorithm for NURBS interpolation. The International Journal of Advanced Manufacturing Technology. 2023;130(3): 1221-1239. https://doi.org/10.1007/s00170-023-12732-z
24. Chen M, Sun Y, Xu J, Liu J. Snap-bounded and time-optimal feedrate scheduling for robotic milling of complex surface parts with analytical solution. IEEE Transactions on Industrial Informatics. 2025;21(5):3880-3889. https://doi.org/10.1109/TII.2025.3528566

Krystian Erwinski: https://orcid.org/0000-0001-6899-1785

Patryk Ustaszewski: https://orcid.org/0009-0008-2336-6220