Nature inspired optimization of jerk limited feedrate profile for NURBS toolpaths in CNC machines

Krystian Erwinski, Rafal Szczepanski and Tomasz Tarczewski
Departament of Automatics and Measurement Systems, Institute of Engineering and Technology, Nicolaus Copernicus University in Torun, Grudziadzka 5/7, 87-100 Torun, Poland
erwin@umk.pl

Abstract. In this paper a novel approach to solving the feedrate optimization problem in Computerized Numerically Controlled (CNC) machines for Non-Uniform Rational B-Spline (NURBS) toolpaths is presented. Particle Swarm Optimization (PSO) gradient-free algorithm is used to determine the optimal shape of the feedrate profile. The shape of the profile is optimized to achieve shortest travel time within the constraints imposed by the machine’s axes. Compared to more common approaches the profile is initialized with a quasi-optimal shape determined by a feedrate limit curve and then optimized to the final shape. Simulation results are presented that show the performance and computational effectiveness of the proposed method.

1. Introduction
The problem of optimizing CNC machine velocity tangent to a curved path (feedrate optimization) has been a topic of interest to researchers for some time [1]. Its goal is to minimize the travel time along a curved toolpath under various machine constraints. Initial works focused on generating a trapezoid acceleration limited [2] and s-curve jerk limited [3] feedrate profiles. These did not take into account limitations in the machine’s axes. For highly curved paths the axial acceleration and jerk can vary significantly due to their non-linear relationship with the toolpath.

This creates a difficult non-linear optimization problem that requires robust optimization algorithms such as Genetic Algorithms [4] or Particle Swarm Optimization [5]. These methods usually start with multiple random solutions and then explore the solution space to find the best solution based on randomness. These methods are very resistant to local minima but are also slower to converge than traditional gradient based algorithms [6]. In this work, the authors propose using a hybrid approach to enhance the convergence of a nature inspired Particle Swarm Optimization algorithm in the feedrate optimization problem by applying a suitable initial guess.

In modern CNC toolpaths are often expressed as Non-Uniform Rational B-Splines (NURBS) polynomial curves. Position on the curve corresponds to the curve’s parameter which is usually a unitless value between 0 and 1 and is computed using DeBoor’s algorithm [7]. Shape of the curve is defined by control points which form a control polygon around the curve. Moving each control point only influences the curve’s shape locally. The shape is furthermore influenced by weights of each control point which defines the distance between the curve and the control point. The feedrate profile is usually defined as a polynomial function of time or toolpath’s parameter. In this work the feedrate profile is defined as a one-dimensional second order B-Spline curve.
splines are defined the same way as NURBS curve’s by using control points but without weight values. An example planar NURBS curve and B-Spline feedrate profile with control points are shown on Fig.1.

2. Feedrate optimization using constrained PSO
The aim of feedrate optimization is to traverse the NURBS toolpath as fast as possible without violating velocity, acceleration and jerk constraints in the machine’s axes. This is done by modifying the shape of the feedrate profile by changing values of its control points. Because acceleration and jerk depend non-linearly on the feedrate and toolpath shape solving the aforementioned problem requires a robust algorithm that can handle complex non-linear objective and constraint functions.

In this work the Particle Swarm Optimization (PSO) algorithm was chosen. This algorithm is a global optimization technique with proven performance. It is much more robust against being trapped in local minima compared to traditional gradient based optimization. The algorithm can also handle highly non-linear and discontinuous objective functions. Compared to more traditional nature-inspired algorithms such as Genetic Algorithms (GA), PSO is simpler to implement and has faster convergence [8–10].

PSO is a swarm algorithm inspired by behavior of fish and birds [11]. Multiple solutions are generated and then updated until they converge to the neighborhood of a minimum. These solutions are called particles. Each particle is described with position in the solution space and velocity. In each algorithm iteration fitness values for each particle is computed and for each particle the position of its best solution is remembered (personal best or pbest). The overall best solution found by the swarm is also stored (global best or gbest). The velocity and position values of each particle are updated according to the following equations[12].

\[
v^*_i,j = \omega \cdot v_{i,j} + \phi_1 \cdot rand_{(0,1)} \cdot (p_{i,j} - x_{i,j}) + \phi_2 \cdot rand_{(0,1)} \cdot (g_i - x_{i,j}) \\
x^*_{i,j} = x_{i,j} + v^*_{i,j}
\]

where: \(v^*_i,j, v_{i,j}\) - new and previous \(i\)-th velocity coordinate of \(j\)-th particle, \(x^*_i,j, x_{i,j}\) - new and previous \(i\)-th position coordinate of \(j\)-th particle, \(rand_{(0,1)}\) - pseudorandom number between 0 and 1, \(\phi_1, \phi_2, \omega\) - acceleration and inertia coefficients, \(p_{i,j}\) - \(i\)-th coordinate of personal best position of \(j\)-th particle, \(g_i\) - \(i\)-th coordinate of global best position.

Coefficients \(\phi_1, \phi_2, \omega\) control the rate of convergence of the swarm and balance between exploration (ability to find new solutions) and exploitation (ability to refine existing solutions).
These can vary widely for different problems but in case of feedrate optimization it was found that default parameters of $\phi_1 = 0.4962$ and $\omega = 0.7298$ work well for all tested cases so these values were used. The algorithm is terminated when maximum number of iterations is reached which is generally problem based and has to be determined empirically to guarantee convergence of the swarm.

Standard version of the PSO algorithm can only constrain values of the optimization parameters by limiting the positions of the particles. In order to incorporate arbitrary non-linear constraints, necessary for the feedrate optimization problem, an extension of the basic algorithm is required. The most popular constraints handling methods are: Deb’s rules and penalty method. The comparison of the above-mentioned constraint handling methods are given in [13]. It was shown that the penalty method provides better convergence and repeatability of obtained solution. It also allows to approach the solution through infeasible solutions which increases convergence speed. Therefore, the penalty methods is applied to the feedrate optimization problem. For each constraint a penalty is added to the objective functions and it’s minimization does not require any modification of the optimization method. In this work an extension of the basic penalty method is used, called Augmented Lagrangian method. The objective function is replaced by the Augmented Lagrangian $L(x)$ defined as [14]:

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

(3)

where: $f(x)$ - objective function value for solution vector $x$, $\rho$ - penalty factor, $c_i(x)$ - value of constraint $i$ for solution $x$, $\lambda_i$ - Lagrange multiplier for constraint $i$, $M$ - number of constraints. This replaces a constrained optimization problem with an equivalent unconstrained optimization problem if correct values of $\rho$ and $\lambda_i$ are used. Optimization is performed in two nested loops. In the inner loop unconstrained minimization of the Augmented Lagrangian objective function is performed using PSO for a predefined number of iterations. In the outer loop the Augmented Lagrangian objective function parameter update is performed after minimization in the inner loop. Compared to a standard quadratic penalty method this two-level approach has the advantage of optimally selecting the penalty factors so that the unconstrained problem is in fact equivalent to the initial constrained problem. The authors use penalty factor and Lagrange multiplier update rules provided in [14].

For the feedrate optimization problem the fitness function is defined as:

$$f(P_i) = \frac{1}{N} \sum_{i=1}^{N} \left( 1 - \frac{P_i}{F_{\text{max}}} \right)^2$$

(4)

where: $F_{\text{max}}$ - maximum feedrate, $P_i$ - value of the $i$-th control point of the B-Spline feedrate profile, $N$ - number of control points. For a two axis machine the constraints are defined as:

$$c_i = \left\{ \frac{|v_{ax}(u_i)|}{V_{x_{\text{max}}}} - 1, \frac{|v_{ay}(u_i)|}{V_{y_{\text{max}}}} - 1, \frac{|a_{ax}(u_i)|}{A_{x_{\text{max}}}} - 1, \frac{|a_{ay}(u_i)|}{A_{y_{\text{max}}}} - 1, \frac{|j_{ax}(u_i)|}{J_{x_{\text{max}}}} - 1, \frac{|j_{ay}(u_i)|}{J_{y_{\text{max}}}} - 1 \right\}$$

(5)

where: $v_{ax}$, $j_{ax}$ - values of axial velocity, acceleration and jerk, $V_{x_{\text{max}}}, V_{y_{\text{max}}}$ - maximum values of axial velocity, acceleration and jerk. Both the objective function and constraints are normalized to their specific maximum value and offset by 1 so that violated constraints are positive and constraints without violation are negative.

Dimension of the problem is equal to the number of optimized feedrate profile control points. Large complex toolpaths will also have complex feedrate profiles with many control points. In
order to limit the optimization problem dimensionality, a moving window approach is applied. Only a fragment of the profile is optimized at any given time. After optimization is finished the optimization window is shifted by a fixed amount of points and optimization is performed for the next set of feedrate control points. In each window each constraint is evaluated at fixed locations corresponding to equally spaced values of parameter $u$. Locations of these points are set so that they are spaced on average every 2mm.

3. Generation of initial guess

In this paper in order to present the PSO algorithm with a reasonable initial solution a jerk limited S-Curve profile is first generated [3, 15]. Axial constraints are not considered so axial velocities, accelerations and jerk can vary significantly even for constant feedrate profiles. To indirectly limit the axial values a Feedrate Limit Funcion (FLF) is introduced [16]:

$$FLF = \min \left( \frac{||C'(u)||F_{max}}{|C'_m(u)|}, \sqrt{\frac{||C''(u)||^2A_{m_{max}}}{|C''_m(u)|}}, \frac{3}{3}\sqrt{\frac{||C'''(u)||^3J_{m_{max}}}{|C'''_m(u)|}} \right)$$

where: $F_{max}$ - maximum feedrate, $C'_m(u), C''_m(u), C'''_m(u)$ - NURBS curve 1st, 2nd and 3rd derivative for axis $m$, $A_{m_{max}}, J_{m_{max}}$ - maximum acceleration and jerk for axis $m$.

The FLF linearizes axial constraints aggregating them into one feedrate constraint. Limits for velocity, acceleration and jerk are computed for each axis and the lowest feedrate value is chosen for each $u$. In order to generate the initial S-curve profile for a given NURBS toolpath the FLF is first scanned and it’s minima are located. Then the arc-length between these points is computed and then a S-curve profile is fitted between them. Duration of the acceleration and deceleration phases are computed so that the distance covered by the acceleration-deceleration profile is equal to the arc-length between the FLF minima. The initial S-curve feedrate profile is generated that limits feedrate, tangent acceleration and jerk and passes under the feedrate limit imposed by the FLF.

The initial S-curve, parametrized with time has to be converted to a B-Spline representation parametrized by the toolpath parameter $u$. A set of feedrate data points on the S-curve are picked and stored with their corresponding parameter value $u$. A global B-Spline interpolation of these data points is performed and the obtained curve is a close approximation of the initial S-curve profile. The advantage of using a B-Spline representation of the feedrate profile is it’s guaranteed continuity of a given degree and the property of only local influence of each parameter on the curve shape. The B-Spline representation also has good numerical properties when interpolating data.

The generated initial feedrate profile is used to initialize the PSO algorithm. One particle is initialized with the feedrate profile’s control points and it’s position becomes the initial global best. Other particles are initialized by adding a random uniform disturbance to the initial guess position. This disturbance is set to be 10% of the possible value range of the optimization space. Each of these initial particle positions is set to be the initial personal best of this particle. Feedrate optimization is then performed as is described in the previous sections.

4. Feedrate optimization algorithm simulation results

To test the performance of the algorithm the example NURBS toolpath from Fig.1 was used. The algorithm was tested with maximum axial velocity, acceleration and jerk set to 2500mm/s, 25000mm/s$^2$ and 500000mm/s$^3$ respectively for each axis and a 250µs interpolation period - parameters typical for high performance laser cutters with linear motor drives. For these parameters an S-curve initial guess profile was generated and a B-Spline curve with 100 control points was interpolated to obtain an initial guess feedrate profile. The initial feedrate profile was optimized using the algorithm described in the previous sections. Optimization was performed
Figure 2: Feedrate initial (blue) and optimized (red) feedrate and axial profiles with Feedrate Limit Function (green)

It can be seen that the initial and final feedrate profiles are very similar. Main difference can be seen in the axial plots of acceleration and jerk. The initial profile exhibited some constraint violations of acceleration and jerk around the 0.5 and 0.8 parameter values. These correspond to the curvature extremes of the Bird toolpath which are the "beak" and "tail" regions. The optimization algorithm rectified these violations and the final axial profiles are all within constraints. Optimization was performed several times and each time results were very similar to those presented. For comparison a random initial profile was also generated and optimized using the same algorithm with identical parameters. Final profile obtained from random initialization was very similar but exhibited some jerk constraint violations and the trajectory took longer to execute (4.75s vs. 4.32s). The value of the Augmented Lagrangian objective function for both initialization types in each window was shown in Fig.3. Optimization convergence rate of the randomly initialized profile is slower than the one with a good initial guess. The objective function starts from a higher value and converges to higher values compared to the S-curve initialized case. The objective function value (in fact the value of the Augmented Lagrangian function) changes significantly every 100 iterations due to update of the penalty factors. This is not the case if optimization is initialized with the S-curve initial guess due to much lower constraint violation. It can be seen that for window 2 the objective function value was increased due to constraint violations (increase in penalty factor). Also for window 4 it can be seen that the objective value still decreases after 2000 iteration. The S-curve initial guess indeed improves the convergence rate and quality of the feedrate optimization.
5. Conclusion

The paper presents a feedrate optimization method based on nature inspired Particle Swarm Optimization algorithm for CNC machine tools. Using polynomial NURBS toolpaths and applying axial constraints on the feedrate optimization makes this a difficult non-linear constrained optimization problem. The algorithm is enhanced with the Augmented Lagrangian constraint handling technique in order to solve the feedrate optimization problem with axial velocity, acceleration and jerk constraints. The authors proposed generating an S-curve feedrate profiles between minima of a Feedrate Limit Function. The S-curve is then converted to a B-Spline profile and used as an initial guess for optimization. It is shown that generating an initial guess based on S-curve jerk limited profiling can significantly improve convergence rate of the algorithm decreasing computation time of this difficult problem. Further work will focus on generalization of the algorithm for machine types with non-trivial kinematics such as serial manipulators, parallel machines (delta, h-bot).

References

[1] Mansour S Z and Seethaler R 2017 J. Manuf. Sci. Eng. 139
[2] Timar S D and Farouki R T 2007 Robot. Comput. Integr. Manuf. 23 563–579
[3] Du X, Huang J and Zhu L M 2015 J. Comput. Des. Eng. 2 206–217
[4] Xu G, Chen J, Zhou H, Yang J, Hu P and Dai W 2019 Int. J. Adv. Manuf. Technol. 101 715–731
[5] Szczepanski R, Erwinski K and Paprocki M 2017 Proceedings of the 22nd International Conference on Methods and Models in Automation and Robotics (MMAR) 431–436
[6] Sencer B, Altintas Y and Croft E 2008 Int. J. Mach. Tools Manuf. 48 733–745
[7] Piegl L A and Tiller W 1997 The NURBS book (2nd ed.) (New York, NY, USA: Springer-Verlag New York, Inc.) ISBN 3-540-61545-8
[8] Majak J, Pohlak M, Eerme M and Velsker T 2012 Mechanics 18 453–460
[9] Kaganski S, Majak J, Karjust K and Toompalu S 2017 Procedia CIRP 63 283–288
[10] Elbes M, Alzubi S, Kanan T, Al-Fuqaha A and Hawashin B 2019 Evolutionary Intelligence 1–17
[11] Kennedy J, Kennedy J and Eberhart R 2001 Swarm intelligence (Morgan Kaufmann)
[12] Clerc M and Kennedy J 2002 IEEE Trans. Evol. Comput. 6 58–73
[13] Szczepanski R, Tarczewski T, Erwinski K and Grzesiak L M 2018 Proceedings of the International Conference in Informatics in Control (ICINCO 2018) 279–286
[14] Birgin E and Martinez J 2008 Optim. Methods Softw. 23 177–195
[15] Ni H, Yuan J, Ji S, Zhang C and Hu T 2018 IEEE Access 6 66403–66417
[16] Sun Y, Chen M, Jia J, Lee Y S and Guo D 2019 Sci. China Technol. Sci. 62 1067–1081