# PSO based feedrate optimization with contour error constraints for NURBS toolpaths

Krystian Erwinski,
Marcin Paprocki, Andrzej Wawrzak
Faculty of Physics Astronomy and Informatics
Nicolaus Copernicus University, Torun, Poland
erwin@fizyka.umk.pl, zwirek@fizyka.umk.pl,
awawrzak@fizyka.umk.pl

Lech M. Grzesiak
Faculty of Electrical Engineering
Institute of Control and Power Electronics
Warsaw University of Technology
Warsaw, Poland
L.Grzesiak@isep.pw.edu.pl

***Abstract*—Generation of a time-optimal feedrate profile for CNC machines has received significant attention in recent years. Most methods focus on achieving maximum allowable feedrate with constrained axial acceleration and jerk without considering manufacturing precision. Manufacturing precision is often defined as contour error which is the distance between desired and actual toolpaths. This paper presents a method of determining the maximum feedrate for NURBS toolpaths while constraining velocity, acceleration, jerk and contour error. Contour error is predicted during optimization by using an artificial neural-network. Optimization is performed by Particle Swarm Optimization with Augmented Lagrangian constraint handling technique. Results of a time-optimal feedrate profile generated for an example toolpath are presented to illustrate the capabilities of the proposed method.**

## I. Introduction

Computerized Numerical Control works by periodically generating position setpoints for the axis servo drives in order to interpolate the desired toolpath. Position setpoints in each timestep are derived from the feedrate profile and toolpath. Feedrate profile is the function of velocity tangent to the toolpath with respect to time or toolpath parameter. In order to navigate the toolpath as fast as posible the value of feedrate in each time step should be as high as possible. This value is limited by physical limits of velocity, acceleration and jerk imposed by the axis drives. Limits of each axis influence the feedrate limit differently depending on the shape of the toolpath. Computation of the time-optimal feedrate profile allows the machine to work with maximum efficiency and has therefore been an important subject of investigation by many researchers. Because many factors influence the shape of the time-optimal feedrate profile this is a difficult optimization problem.

Most early feedrate scheduling methods consider only feedrate and acceleration constraints. Bobrow et al. [1] and Shin and McKay [2] proposed a bang-bang trajectory generation by switching the acceleration between maximum and minimum limits at specific path points. Dong and Stori [3] use a bidirectional scan algorithm to optimize the feedrate in the s-s phase plane. Later works also consider jerk and jounce constraints. Lai et al. [4] and Fan et al. [5] first determine maximum feasible feedrate at critical points of the toolpath and then use jerk or jounce limited acceleration/deceleration profiles to connect the feedrate profile between these points. Further works also considered acceleration and jerk constraints in each axis not just the feedrate profile. Altinas and Erkorkmaz [6] and Sencer and Altinas [7] use a polynomial feedrate profile and then optimize it's shape using constrained optimization.

Most researchers do not take into account the effect of feedrate on the accuracy of toolpath following. This is usually measured by contour error which is defined as the shortest distance between the desired and actual toolpaths. This paper introduces a novel optimization algorithm for toolpaths defined by Non-Uniform Rational B-Spline (NURBS) curves. The innovative aspect of the presented algorithm is the direct incorporation of contour error constraint into the feedrate optimization process in addition to axial velocity, acceleration and jerk. The feedrate profile is defined as a one dimensional B-Spline. It's control points are iteratively modified during optimization in order to achieve maximum feedrate within the given constraints. A dynamic model of the feed drive in the form of a NARX neural network is used to predict the contour error during optimization. Optimization is performed by Particle Swarm Optimization with constraints handled by the Augmented Lagrangian Method. An example 2-D NURBS toolpath is used to experimentally verify the effectiveness of the proposed method for different values of the contour error constraint.

## II. Feedrate profile optimization for NURBS toolpaths

NURBS are polynomial splines that allow complicated toolpaths to be represented with relatively few data points compared to series of line segments usually used in CNC machining. Furthermore NURBS toolpaths can have high continuity which enables smooth traversal without sharp changes to feedrate. In order to interpolate a NURBS toolpath a relationship between increment in arc-length and the increment in curve parameter has to be found. This relationship is non-linear and strongly dependant on the curve shape. Therefore a Taylor approximation is usually used in each interpolation step to determine the curve parameter increment based on current feedrate. A third order formula is used in this work [8]:

$$u_{i+1} = u_i + \dot{u}_i T + \frac{\ddot{u}_i}{2} T^2 + \frac{\dddot{u}_i}{6} T^3 \tag{1}$$

where $T$ is the interpolation period. Derivatives of the NURBS toolpath parameter $u$ with respect to time $(\dot{u}, \ddot{u}, \dddot{u})$ are computed using following equations:

$$\begin{aligned} \dot{u} &= \frac{V}{\sigma} \\ \ddot{u} &= \frac{A}{\sigma} - \frac{\sigma' V}{\sigma^2} \dot{u} \\ \dddot{u} &= \frac{J}{\sigma} - \frac{3\sigma' V}{\sigma^2} \ddot{u} - \frac{\sigma'' V}{\sigma^2} \dot{u}^2 \end{aligned} \tag{2}$$

where: $\sigma, \sigma', \sigma''$ are the parametric velocity and its derivatives with respect to the curve parameter and $V, A, J$ are feedrate, acceleration and jerk.

The feedrate profile is defined as a second order one dimensional B-Spline curve. The x dimension is the toolpath parameter which is also the parameter of the feedrate profile curve. The y dimension is the feedrate value. Feedrate optimization is the process of adapting the b-spline curves shape so that its value is as high as possible in every point and all constraints are satisfied. Shape of the b-spline can be changed by modyfining its control points. Values of the control points are the optimization variables used by the optimization algorithm. Because control points influence the curve only locally one fragment of the feedrate profile can be optimized while the others are not influenced. This locality property can clearly be seen in fig. 1. Optimizing the whole feedrate profile at once

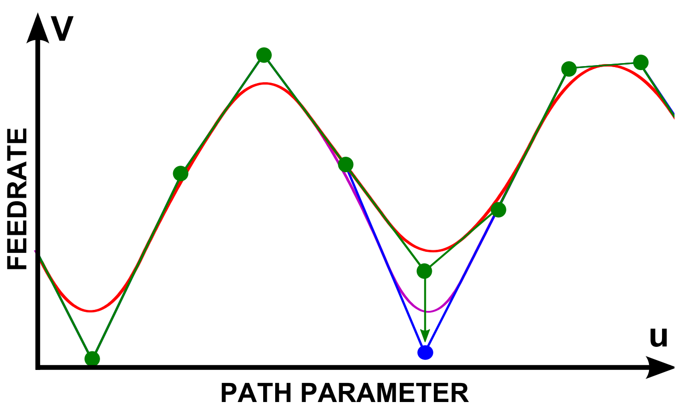


Fig. 1. B-Spline feedrate profile

would lead to a optimization problem with huge number of variables and constraints. Instead a moving window approach is used and only a fragment of the profile with constant number of control points and constraints is optimized in one optimization run. The window is then shifted and a next set of points are optimized until the end of the toolpath is reached. Constraints are evaluated at evenly spaced parameter locations instead of being checked at every interpolation step. Because of this the number of constraints is constant while optimizing each window. Ten constraint evaluation points are placed between each control point. Choosing the curve parameter as the x variable instead of time also facilitates optimization because the feedrate profile endpoint remains constant instead of being changed with every change of feedrate.

In order to achieve minimum machining time the value of every feedrate profile control point in each optimization window should be as high as possible. This problem can be formulated in the form of a objective function to be minimized:

$$F_{obj} = \frac{1}{N} \sum_{i=1}^{N} \left(1 - \frac{v_i}{V_{max}}\right)^2 \tag{3}$$

where $v_i$ are the feedrate control point values, $V_{max}$ is the maximum feedrate and $N$ is the number of control points in each window. This optimization function is subject to constraints of velocity, acceleration and jerk in each axis as well as contour error. The objective function and constraints are normalized with respect to their maximum values so that their different orders of magnitude do not influence the optimization process.

$$\begin{aligned} c_v(u_i) &= \frac{v(u_i)}{V_{max}} - 1, \qquad c_a(u_i) = \frac{a(u_i)}{A_{max}} - 1 \\ c_j(u_i) &= \frac{j(u_i)}{J_{max}} - 1, \qquad c_{ce}(u_i) = \frac{\epsilon_c(u_i)}{E_{c_{max}}} - 1 \end{aligned} \tag{4}$$

Values of axial velocity, acceleration and jerk are computed analytically from the feedrate profile and NURBS toolpath for each constraint evaluation point. Computation of contour error requires a predictive model of the machines feed drives This algorithm uses off-line trained NARX neural networks to predict following errors of each axis in response to axial velocity commands. Predicted following errors are used to compute the contour error based on the toolpath geometry (fig. 2). Contour error computation for NURBS toolpath is performed using the formula presented in [9].

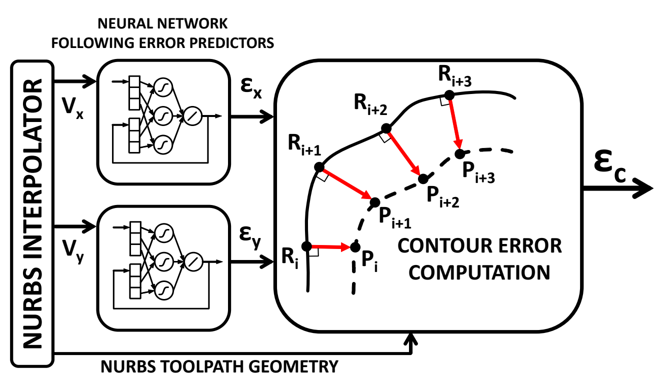


Fig. 2. Block schematic of the neural contour error predictor

The optimization process iteratively modifies the shape of the feedrate profile until all constraints are satisfied and no more improvement in feedrate can be achieved. This optimization process is shown on fig. 3.

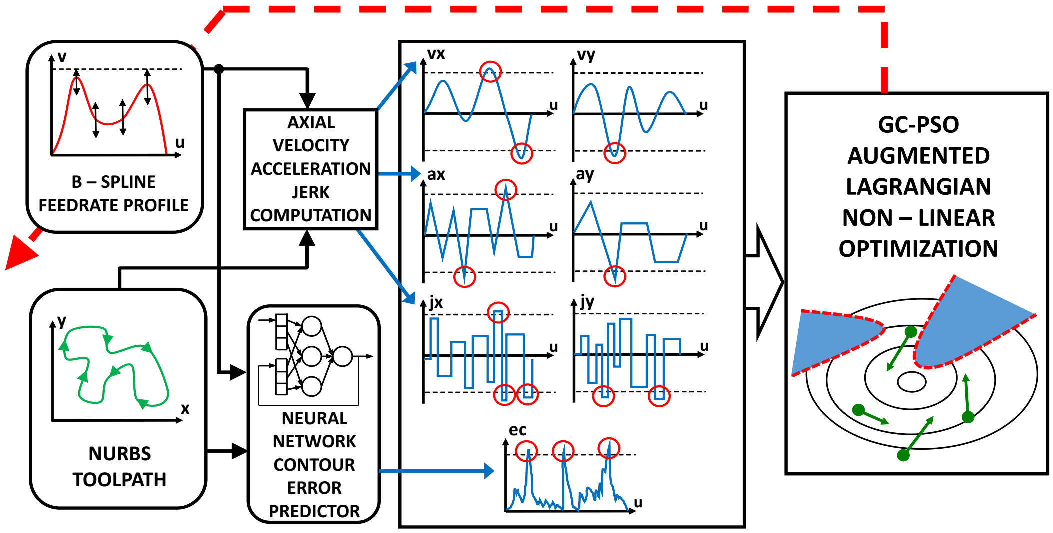


Fig. 3. Block schematic of the feedrate optimization algorithm

## III. Constrained PSO Optimization

Feedrate optimization with contour error constraints is a difficult non-linear problem prone to premature stoppage in local minima. In order to avoid local minima the well proven Particle Swarm Optimization algorithm was used [10]. PSO is a non-derivative multi-agent optimization method that works by moving virtual particles through the solution space. PSO uses the following equations to modify position and velocity vectors of each particle:

$$\begin{aligned} x_{i,j}^{new} &= x_{i,j} + v_{i,j}^{new} \\ v_{i,j}^{new} &= \omega \cdot v_{i,j} + \phi_1 \cdot rand_{(0,1)}^{U} \cdot (p_{i,j} - x_{i,j}) + \\ &\quad \phi_2 \cdot rand_{(0,1)}^{U} \cdot (g_i - x_{i,j}) \end{aligned} \tag{5}$$

$\omega, \phi_1, \phi_2$ are algorithm parameters respectively; $rand_{(0,1)}^{U}$ are pseudo-random numbers between 0 and 1 with uniform distribution; $g_i$ is the best position found so far (global best or gbest); $p_{i,j}$ is the best position found so far by each particle (personal best or pbest); $x_{i,j}$ and $v_{i,j}$ are the postion and velocity of each particle.

In this work the Guaranteed Convergence PSO (GC-PSO) variant was used [11] which is resistant to premature convergence to a non-minimum point (swarm stagnation). GC-PSO uses a different equation to update velocity of the gbest particle. This equation randomly modifies the velocity of the particle that found the last gbest position:

$$v_{i,gbest}^{new} = \omega \cdot v_{i,gbest} + R \cdot \left(1 - 2 \cdot rand_{(0,1)}^{U}\right) \tag{6}$$

The modified equation forces the gbest particle to do a random search in a sphere with radius $R$ around the gbest position. This guarantees that the final gbest position is in fact a minimum of the objective function [12].

One important advantage of PSO is that it does not require a good feasible initial guess as the initial particle positions are generated randomly. It is also derivative free and includes randomness and is therefore resilient against premature convergence to a local minimum. The algorithm is also easy to implement and has low computational requirements despite being very effective at minimizing non-linear objective functions. Most variants of PSO are good at finding the neighbourhood of the minimum of the objective function but usually take many iterations to refine the solution due to random nature of the algorithm. Therefore a secondary local optimization algorithm BOBYQA developed by M. Powell [13] was used to refine the solution found by GC-PSO.

Most PSO variants including GC-PSO are only box constrained algorithms. In order to benefit from the advantages of PSO and simultaneously handle non-linear constraints GC-PSO was combined with the Augmented Lagrangian Method [14]. This method works by adding a penalty to the objective function for each violated constraint $c_i(x)$ and modifying the penalty factors $\lambda$ and $\rho$ after each successful optimization. The augmented objective function (Augmented Lagrangian) has the following form [15]:

$$L(\mathbf{x}) = F_{obj}(\mathbf{x}) + \frac{\rho}{2} \sum_{i=1}^{M} \left( \max\left\{0, c_i(\mathbf{x}) + \frac{\lambda_i}{\rho}\right\}\right)^2 \tag{7}$$

Unconstrained minimization of the Augmented Lagrangian with proper penalty factors is equivalent to minimizing the objective function without violating the constraints. The penalty factors are set automatically during optimization. This however requires repeated minimization of the augmented objective function after each update of the penalty factors using the previously obtained solution as an initial guess. During testing it was found that solutions obtained during successive runs are close to each other in the optimization space. It was therefore decided to only use GC-PSO in the first run to reliably find a good initial solution and only use BOBYQA local optimization during successive runs.

TABLE I
PSO PARAMETERS USED FOR FEEDRATE OPTIMIZATION

| | |
|---|---|
| swarm size | 25 |
| $\omega$ | 0.7298 |
| $\phi_1, \phi_2$ | 1.4962 |
| initial $R$ | 1.0 |

The algorithm was implemented and tested to find the best parameters for the feedrate optimization process. During trials it was found that default parameters proposed by Clerc [16] provide the best reliability, speed of convergence and solution quality (see table I). For the Augmented Lagrangian Method and BOBYQA algorithm default parameters proposed by their authors were used. The flow diagram of the feedrate optimization algorithm was shown on figure 4.

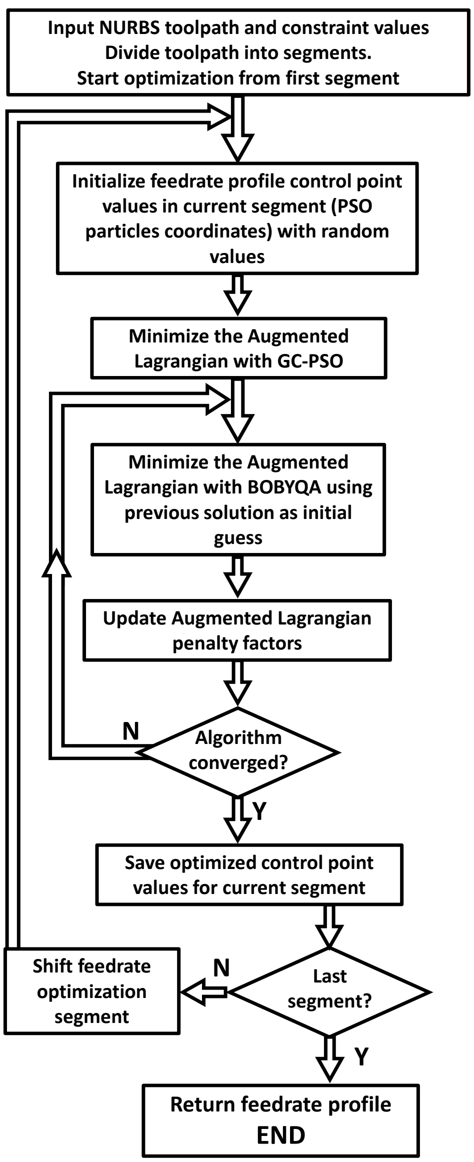


Fig. 4. Flow diagram of the feedrate optimization algorithm

## IV. Results

To test the proposed feedrate optimization method an open CNC control system was used. It consists of a PC based CNC controller with Linux RTAI real-time operating system and LinuxCNC control software. The PC based controller directly communicates with the axial servo drives of a 2-axis CNC router via Ethernet Powerlink fieldbus whose stack has been implemented in Linux RTAI and integrated into LinuxCNC [17]. The optimization algorithm was implemented as a user space program that optimized the given NURBS toolpath off-line. A software real-time module was added to the LinuxCNC software that interpreted and interpolated NURBS toolpaths with the pre-optimized B-Spline feedrate profiles. The laboratory setup used to test the algorithm is shown on figure 5.

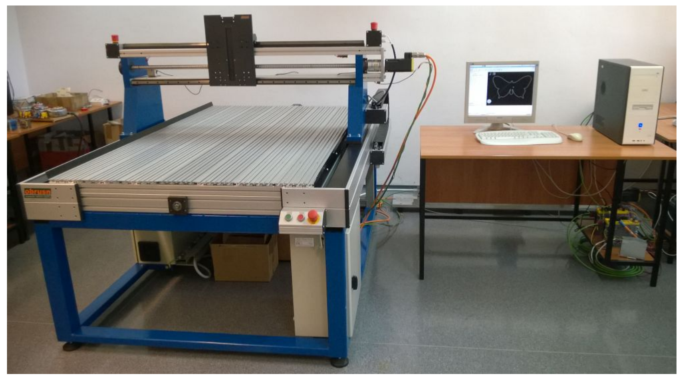

Fig. 5. Biaxial CNC router used for testing

Neural networks modelling the machine's feed drives were taught off-line and implemented into the algorithm to predict the contour error. A test NURBS toolpath (fig. 6) was input into the program and an off-line feedrate optimization process was performed using the Augmented Lagrangian GC-PSO optimization algorithm. Velocity, Acceleration and Jerk axial constraints where set to 300 $mm/s$, 2500 $mm/s^2$ and 50000 $mm/s^3$ respectively. In order to limit the contour error to 0.05mm the maximum contour error constraint was set to 0.45mm in order to compensate for modelling errors. Feedrate control points were evenly placed 7.5mm from each other. The optimization window was 40 points wide with 10 points overlap between segments. The optimized feedrate profile is shown on fig. 7. It can be seen that the optimization algorithm minimized the feedrate in fragments of the toolpath that have the highest curvature and maximized it in fragments where the curvature is small.

Velocity, acceleration and jerk profiles in x and y axes are shown on figures 8, 9 and 10 respectively. The presented results clearly show that the feedrate optimization algorithm successfully limits the axial velocity, acceleration and jerk to the desired values imposed by capabilities of the machine's feed drives.

Contour error predicted by the neural network contour error predictor in response to the optimized feedrate profile was

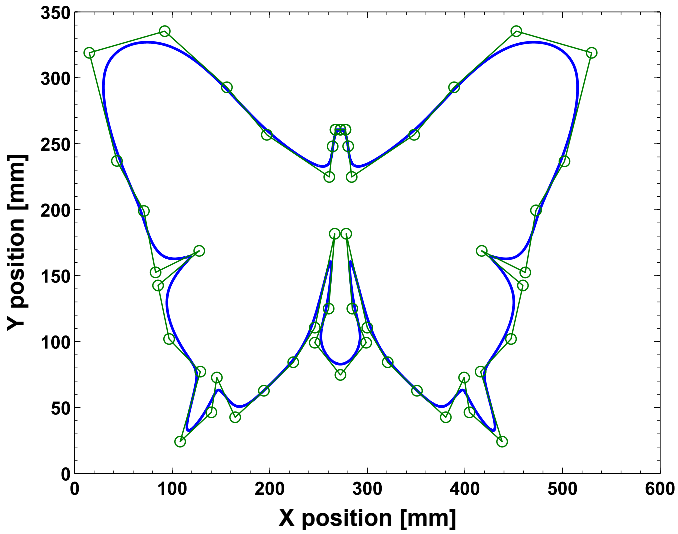


Fig. 6. Test NURBS toolpath

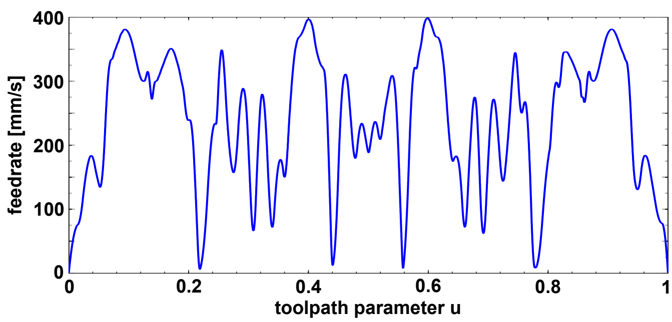


Fig. 7. Optimized feedrate profile (curve parameter)

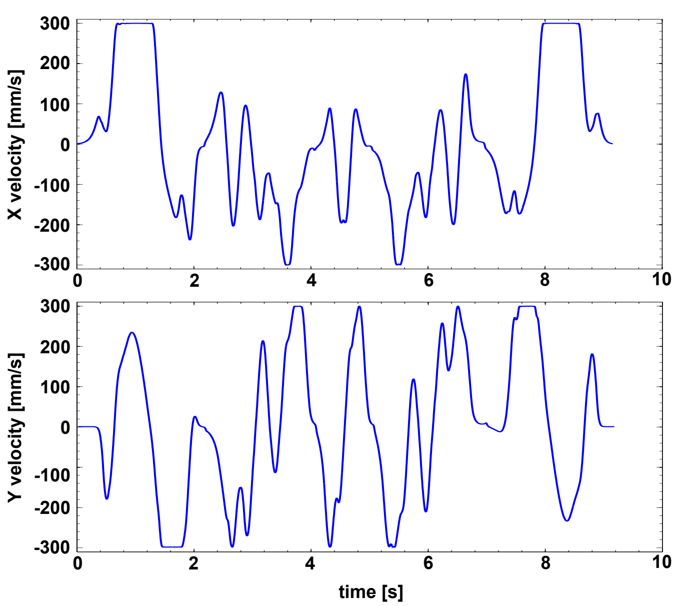


Fig. 8. Axial velocity of the optimized feedrate profile

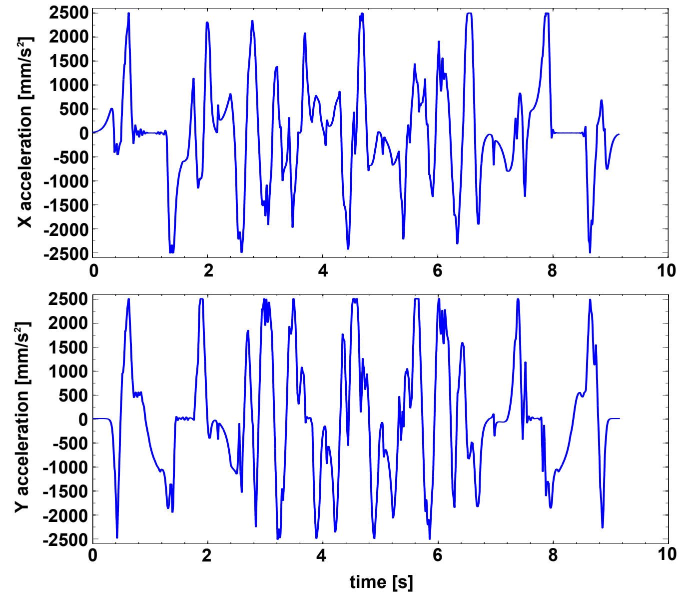


Fig. 9. Axial acceleration of the optimized feedrate profile

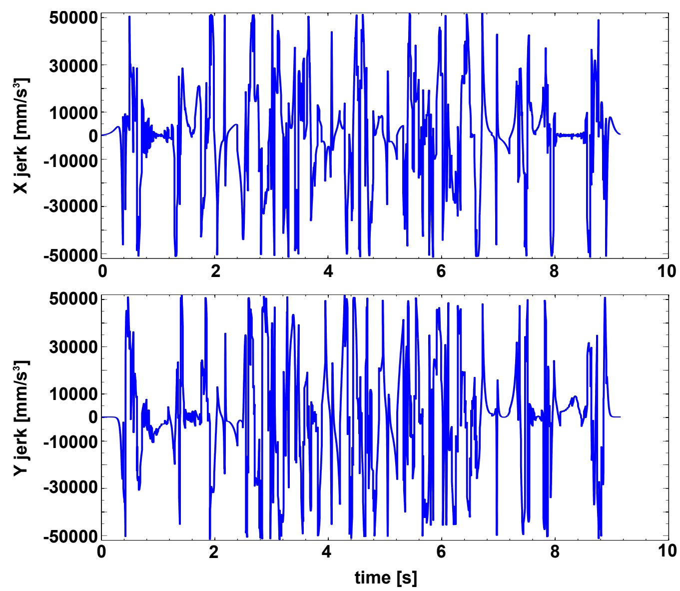


Fig. 10. Axial jerk of the optimized feedrate profile

shown on figure 11a. It can be seen that the contour error is limited to the desired tolerance at every point of the trajectory. This proves that the algorithm is effective providing the model of the machine's feed drive is accurate. A perfect prediction model cannot be created for a physical feed drive due to many factors that contribute to its following error. The machines feed drives also have limited repeatability due to imperfections in their construction and random disturbances. This discrepancy between the contour error predicted by the neural network predictor and the actual contour error measured on the CNC machine used for testing was experimentally verified to be less than 0.005mm. In order to guarantee that the contour error

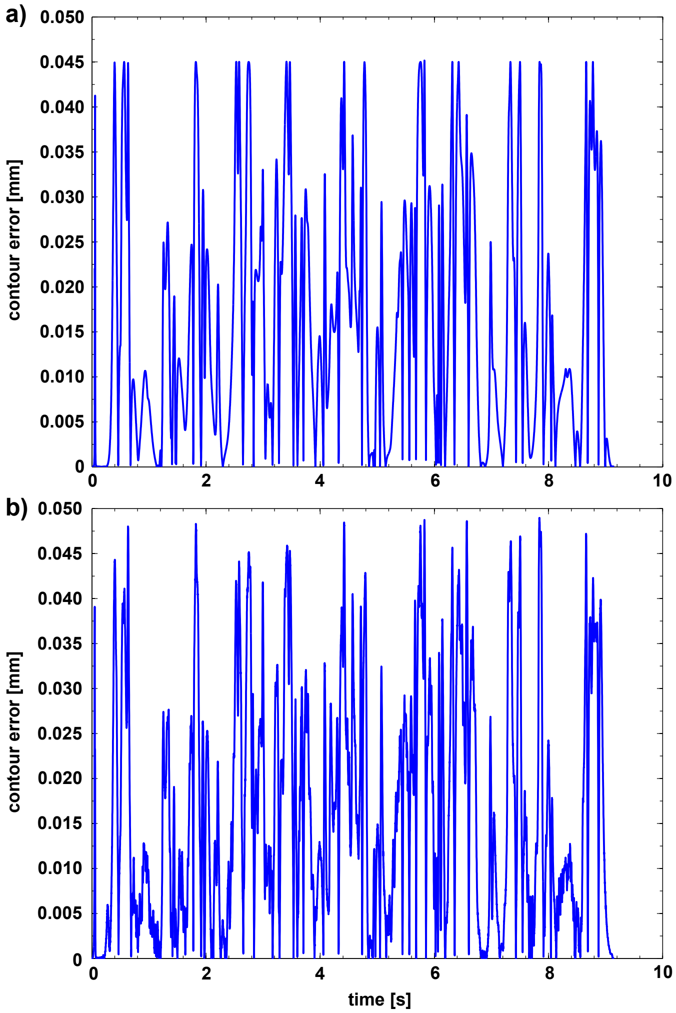


Fig. 11. Predicted (a) and actual (b) contour error of the optimized feedrate profile

is less than 0.05mm the contour error constraint had to be lowered to 0.045mm. The actual contour error shown in figure 11b is violating the contour error constraint slightly due to model inaccuracy but is less than 0.05mm which is the desired contour error tolerance.

## V. Conclusion

This paper presents a CNC feedrate optimization algorithm for NURBS toolpaths. The proposed method handles velocity, acceleration and jerk constraints imposed by the machines feed drives. It also introduces contour error constraints to directly handle machine dynamics in the feedrate optimization process. In order to predict contouring performance during off-line optimization a neural network contour error predictor was used. Optimization was performed by Guaranteed Convergence Particle Swarm Optimization algorithm which is robust to premature convergence to local minima inherent in any difficult non-linear optimization problem. BOBYQA local optimization procedure was used as a secondary algorithm to refine solutions found by GC-PSO. To handle non-linear constraints the Augmented Lagrangian Method was combined with GC-PSO. Tests performed using a biaxial CNC router show that the proposed method generates a time-optimal feedrate profile that successfully limits axial velocity, acceleration and jerk to the desired maximum values imposed by the feed drives' limits. The method also limits contour error providing modelling inaccuracies are taken into account. Further work will concentrate on improving the contour error predictor and parallelization of the algorithm to enable on-line operation.

## References

[1] J. E. Bobrow, S. Dubowsky, and J. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.

[2] K. G. Shin and N. D. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *Automatic Control, IEEE Transactions on*, vol. 30, no. 6, pp. 531–541, 1985.

[3] J. Dong and J. Stori, "A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization," *Journal of dynamic systems, measurement, and control*, vol. 128, no. 2, pp. 379–390, 2006.

[4] J.-Y. Lai, K.-Y. Lin, S.-J. Tseng, and W.-D. Ueng, "On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk," *The International Journal of Advanced Manufacturing Technology*, vol. 37, no. 1-2, pp. 104–121, 2008.

[5] W. Fan, X.-S. Gao, W. Yan, and C.-M. Yuan, "Interpolation of parametric cnc machining path under confined jounce," *The International Journal of Advanced Manufacturing Technology*, vol. 62, no. 5-8, pp. 719–739, 2012.

[6] Y. Altintas and K. Erkorkmaz, "Feedrate optimization for spline interpolation in high speed machine tools," *CIRP Annals-Manufacturing Technology*, vol. 52, no. 1, pp. 297–302, 2003.

[7] B. Sencer, Y. Altintas, and E. Croft, "Feed optimization for five-axis cnc machine tools with drive constraints," *International Journal of Machine Tools and Manufacture*, vol. 48, no. 7, pp. 733–745, 2008.

[8] R. Farouki and Y. Tsai, "Exact taylor series coefficients for variable-feedrate cnc curve interpolators," *Computer-Aided Design*, vol. 33, no. 2, pp. 155–165, 2001.

[9] L. Zhu, H. Zhao, and H. Ding, "Real-time contouring error estimation for multi-axis motion systems using the second-order approximation," *International Journal of Machine Tools and Manufacture*, vol. 68, pp. 75–80, 2013.

[10] R. C. Eberhart, J. Kennedy *et al.*, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, vol. 1. New York, NY, 1995, pp. 39–43.

[11] F. van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimizer," in *Proceedings of the IEEE international conference on systems, man and cybernetics*, vol. 3, 2002, pp. 94–99.

[12] F. Van Den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, University of Pretoria, 2006.

[13] M. J. Powell, "The bobyqa algorithm for bound constrained optimization without derivatives," *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, 2009.

[14] K. Sedlaczek and P. Eberhard, "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*, vol. 32, no. 4, pp. 277–286, 2006.

[15] E. G. Birgin and J. M. Martínez, "Improving ultimate convergence of an augmented lagrangian method," *Optimization Methods and Software*, vol. 23, no. 2, pp. 177–195, 2008.

[16] M. Clerc, *Particle swarm optimization*. John Wiley & Sons, 2010, vol. 93.

[17] K. Erwinski, M. Paprocki, L. M. Grzesiak, K. Karwowski, and A. Wawrzak, "Application of ethernet powerlink for communication in a linux rtai open cnc system," *Industrial Electronics, IEEE Transactions on*, vol. 60, no. 2, pp. 628–636, 2013.