Maximum Feedrate Interpolator for Multi-axis CNC Machining with Jerk Constraints

Xavier Beudaert, Sylvain Lavernhe, Christophe Tournier

Maximum Feedrate Interpolator for Multi-axis CNC Machining with Jerk Constraints

X. Beudaert, S. Lavernhe, C. Tournier
LURPA, ENS Cachan, Université Paris Sud 11
61 av du pdt Wilson, 94235 Cachan, France
name@lurpa.ens-cachan.fr

Abstract

A key role of the CNC is to perform the feedrate interpolation which means to generate the setpoints for each machine tool axis. The aim of the VPOp algorithm is to make maximum use of the machine tool respecting both tangential and axis jerk on rotary and linear axes.

The developed algorithm uses an iterative constraints intersection approach. At each sampling period, all the constraints given by each axis are expressed and by intersecting all of them the allowable interval for the next point along the path is computed. Examples and comparisons with an industrial CNC demonstrate the performances of the algorithm.

Keywords:
Feedrate, CNC, Jerk

1 INTRODUCTION

This paper deals with the trajectory generation which is realized by the Computer Numerical Control (CNC). Indeed, starting from a NC program the CNC has to generate the setpoints for the machine's axes. So the input is a reference tool path with a programmed feedrate and the output is a sequence of axis setpoints which have to produce a smooth movement.

For machining operations, the main goal is to follow precisely a given geometric path, for that reason, a decoupled approach is preferred. This approach consists in decoupling the geometry from the temporal evolution law.

The feedrate planning is an old problem which can be solved using different methods and taking into account different constraints. In robotics, Shin and McKay [1] proposed a two pass iterative algorithm based on the constraints intersection principle. The idea is to start from the beginning of the tool path and to go as fast as possible even if some constraints are not respected. Then in the reverse pass, the procedure is repeated with the additional constraint that the feedrate is lower than in the forward pass. Finally, a corrective algorithm is applied to connect both passes with acceleration constraints. This two pass algorithm was reused and improved. Renton and Elbestawi [2] worked on the velocity and acceleration limits determination. Timar et al. [3] used polynomial parametric curves on which they could obtained a closed form solution for the feedrate planning problem with axis acceleration constraints. Dong and Stori [4] tried to prove the optimality of the two pass algorithm. All the previously cited articles took velocity and acceleration constraints of the drives. But it is known that the jerk is an important parameter which should be considered as well. Indeed in many high speed milling operations the jerk is the limiting parameter.

The effect of the jerk limitation was studied in detail by Barre et al. [5]. It is clear that the jerk has to be limited to reduce the frequency content of the trajectory and to avoid exciting the natural modes of the structure. Moreover, several articles are dealing only with the tangential jerk (third derivative with respect to the time of the tool/workpiece movement). This can be interesting for manipulators but to avoid vibrations of a machine tool structure, each axis limit has to be considered. Liu et al. [6] modified the feedrate profile to take into account the jerk and the natural frequencies of the machine tool. This method should be applied carefully not to generate contour error.

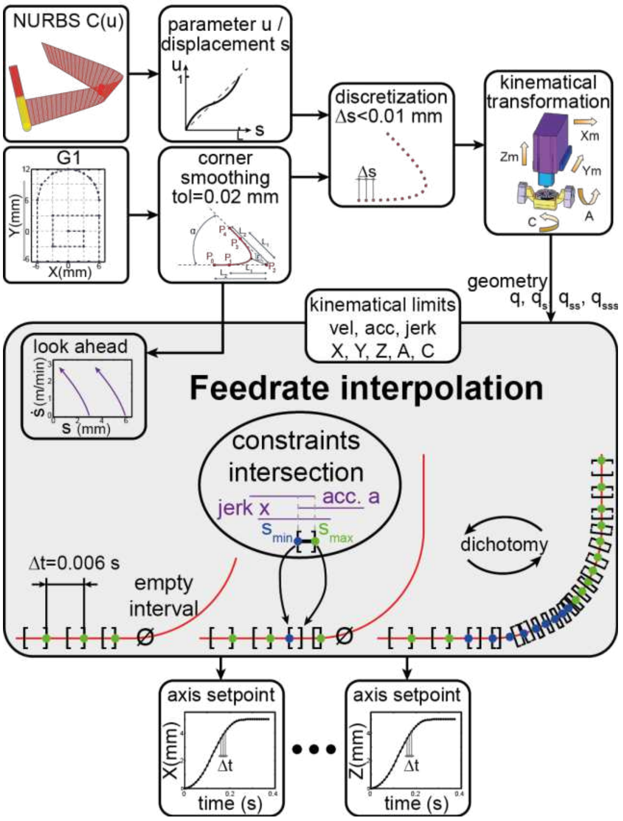


Figure 1: VPOp algorithm

Dong et al. [7] introduced the jerk constraint to the two pass iterative algorithm. At first sight, it does not seem challenging to add another constraint. But in fact, it is really hard to connect the forward and the backward pass respecting the jerk constraints. Indeed a lot of points have to be modified and for complex shapes the switching points can be too close and the whole feedrate profile has to be modified by the corrective algorithm.

The other main method to realize the feedrate planning with jerk constraint is to use a predefined profile as presented in Erkorkmaz and Altintas [8]. The objective of this method is to generate a sequence of predefined profiles which respects the constraints. This solution was also used and improved in many other works, e.g. [9-12]. Olabi et al. [13] applied the predefined profile to an industrial 6-axis machining robot. Lin et al. [14] introduced the control structure and the contour error in their interpolator. But the main problem is that the constraint on the predefined profile limits only the tangential constraints. In practice each axis has its own limitations; furthermore with linear and rotary drives it is impossible to make the link between the tangential jerk and the axis jerk due to the non linear kinematical transformation.

In the literature, few works are realized concerning multi-axis machining with linear and rotary axes. The methods with predefined feedrate profile cannot be applied to this kind of machine as the kinematical transformation between the cartesian space and the joint space is non-linear. Thus the tangential limits have to be really conservative to make sure that the axis kinematical limits are not exceeded.

On 5-axis machining with axis jerk constraints, Lavernhe et al. [15] presented a predictive model of kinematical performance using an iterative time inverse method. This method is suited for G1 tool path but it is difficult to apply it to NURBS tool path. Sencer et al. [16] represented the feedrate profile with a cubic B-Spline which is iteratively modulated to minimize the machining time. This method is suited for NURBS tool path but is difficult to apply to G1 format.

The goal of this paper is to propose a feedrate planning solution for multi-axis structures with jerk constraints on each axis. A decoupled approach is used to separate the geometrical problem to the temporal interpolation problem. Furthermore, the mathematical formulation used here allows treating both linear and rotary axes in the same way. Figure 1 shows the whole procedure. First the geometrical work is carried out. Then the kinematical transformation is used to obtain the joint movements. After that, the temporal interpolation is performed using a constraint intersection principle and a dichotomy which will be detailed. Finally the output is the sampled axis setpoints respecting velocity, acceleration and jerk constraints of each drive.

The plan for the remainder of this paper is as follows. Section 2 details the feedrate planning which aims to obtain the velocity profile which makes best use of the kinematical characteristics of the machine tool. Experiments and simulations are carried out to demonstrate the efficiency of the proposed methods in Section 3. Finally, the paper is concluded in Section 4.

2 FEEDRATE PLANNING ALGORITHM

2.1 Mathematical formalism and drive constraints

Using the formula for the derivative of the composition of two functions (Equation 1), it is possible to express the velocity of the drives $\dot{\mathbf{q}}$ as a function of the geometry $\mathbf{q}_s$ multiplied by a function of the motion $\dot{s}$. Therefore, the motion is decoupled from the geometry. One can note that this formula is valid for linear and rotary axes. The acceleration $\ddot{\mathbf{q}}$ and jerk $\dddot{\mathbf{q}}$ of the drives are obtained identically in Equations 2 and 3.

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \frac{d\mathbf{q}}{ds}\frac{ds}{dt} = \mathbf{q}_s \cdot \dot{s} \tag{1}$$

$$\ddot{\mathbf{q}} = \mathbf{q}_{ss} \cdot \dot{s}^2 + \mathbf{q}_s \cdot \ddot{s} \tag{2}$$

$$\dddot{\mathbf{q}} = \mathbf{q}_{sss} \cdot \dot{s}^3 + 3\mathbf{q}_{ss} \cdot \dot{s}\ddot{s} + \mathbf{q}_s \cdot \dddot{s} \tag{3}$$

$\mathbf{q}_s$, $\mathbf{q}_{ss}$, $\mathbf{q}_{sss}$ are the geometrical derivatives with respect to the displacement s along the tool path. They are known as soon as the geometrical treatment of the tool path is realized.

Because of the physical realization of the drives (motors, driving system, machine tool structure ...) the velocity, acceleration and jerk of each individual drive have to be limited. The jerk limitation is important to reduce the vibration due to the dominating vibratory mode of the axes.

Equation 4 presents the velocity constraints.

$$-\begin{bmatrix} v_{\max,x} \\ v_{\max,y} \\ v_{\max,z} \\ v_{\max,a} \\ v_{\max,c} \end{bmatrix} \le \begin{bmatrix} X_s^j \\ Y_s^j \\ Z_s^j \\ A_s^j \\ C_s^j \end{bmatrix} \dot{s}_j \le \begin{bmatrix} v_{\max,x} \\ v_{\max,y} \\ v_{\max,z} \\ v_{\max,a} \\ v_{\max,c} \end{bmatrix} \tag{4}$$

All the constraints are set to be symmetrical as it is commonly used in the machine tool characteristics. Then the following set of inequations is obtained respectively for the velocity, acceleration and jerk constraints. The notation | | stands for the absolute value of each scalar term.

$$\left|\dot{q}_j^i\right| \le V_{\max,i}; \left|\ddot{q}_j^i\right| \le A_{\max,i}; \left|\dddot{q}_j^i\right| \le J_{\max,i} \tag{5}$$

The aim of the algorithm is to calculate the next reachable point with a fixed Δt knowing all the characteristics on the previous points. This is done by intersecting all the constraints as proposed in [1]. Using the discretization, each constraint can be reduced to a polynomial inequation. Solving the inequation, an interval over which the constraint is verified is obtained. Finally, as it is shown on Figure 2, the intersection of all these intervals gives the solution interval [$s_{\min}$, $s_{\max}$] into which all the constraints are respected for this step. This interval can be empty, for example when the feedrate is too high at the entrance of a sharp curvature area. In this case, it is necessary to go few steps back and to try to reduce the feedrate as it is shown on Figure 1. It is also possible to have a solution which is a simple interval or a union of distinct intervals [$s_{\min}$, s_2] u [s_3, $s_{\max}$]. In this case, the minimum and maximum allowable solutions $s_{\min}$ and $s_{\max}$ are kept.

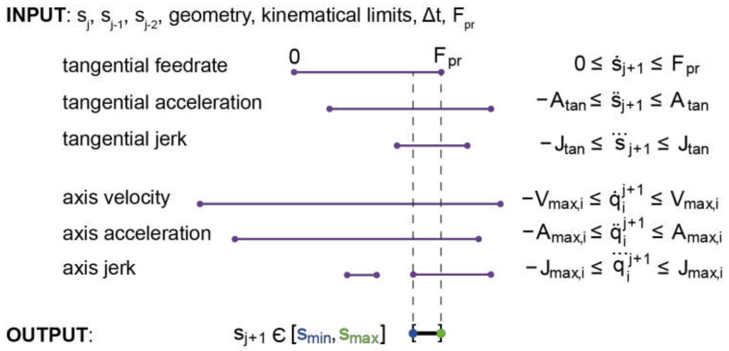

INPUT: s_j, s_{j-1}, s_{j-2}, geometry, kinematical limits, Δt, F_{pr}
tangential feedrate
0
F_{pr}
$0 \leq \dot{s}_{j+1} \leq F_{pr}$
tangential acceleration
$-A_{tan} \leq \ddot{s}_{j+1} \leq A_{tan}$
tangential jerk
$-J_{tan} \leq \dddot{s}_{j+1} \leq J_{tan}$
axis velocity
$-V_{max,i} \leq \dot{q}_i^{j+1} \leq V_{max,i}$
axis acceleration
$-A_{max,i} \leq \ddot{q}_i^{j+1} \leq A_{max,i}$
axis jerk
$-J_{max,i} \leq \dddot{q}_i^{j+1} \leq J_{max,i}$
OUTPUT:
$s_{j+1} \in [s_{min}, s_{max}]$

Figure 2: Constraints intersection

2.2 Detailed algorithm

For a discretized algorithm, two discretizations are conceivable: a geometrical discretization in Δu or in Δs as in [4] for example or a discretization in time Δt. The problem with the discretization in Δs is that eventually you need to send setpoints to the controller with a fixed frequency. So at really low feedrate you need a small Δs increments to be closed to the desired command frequency therefore at high feedrate many useless points will be computed. Moreover, with a fixed Δs the evaluation of the jerk will be really rough. That is why a constant time step has been chosen for the algorithm presented here. Hence, the velocity, acceleration and jerk are computed as follow:

$$\dot{s}_{j+1} = \frac{s_{j+1} - s_j}{\Delta t}; \qquad \dots \qquad ; \dddot{s}_{j+1} = \frac{\ddot{s}_{j+1} - \ddot{s}_j}{\Delta t} \qquad (6)$$

In order to show explicitly how the algorithm is working, it has been applied on a really simple example. The tool path is a straight line and the aim is to reach the programmed feedrate starting from rest. The feedrate will be limited only by the programmed feedrate and the jerk of the axis which are respectively 5 m/min and 5 m/s^3. The Figure 3 shows each calculated point. On Figure 3.a, one can see that the algorithm starts going as fast as possible until a point with no solution (empty intersection) is reached at j=18. Indeed, it is impossible to stay under the programmed feedrate while respecting the acceleration and jerk constraints. That means that the sequence used on the previous points was wrong. To find the switching point where s_{min} has to be chosen instead of s_{max} a dichotomy is used.

The idea is to say that if it is possible to find a sequence which allows reaching $\dot{s} = 0$, it will be possible to find a sequence which respects all the constraints along any geometry (including any potential upcoming sharp corner). To reach $\dot{s} = 0$, it is necessary to decelerate as much as possible so a sequence using always s_{min} is tried.

The first point from where $\dot{s} = 0$ can be reached is find with a dichotomy. On Figure 3.b s_{min} is taken from j=9 and $\dot{s} = 0$ is reached so it is possible to go further. With j=13, $\dot{s} = 0$ is reached again. So on Figure 3.c j=15 is tested, no solution is found. The last iteration of the dichotomy is in j=14, here again $\dot{s} = 0$ cannot be reached. So it is sure that s_{13}=s_{min} has to be chosen to be able to obtain a solution. Once the dichotomy is finished, the algorithm tries again to go as fast as possible taking s_{max} at each step. Finally, Figure 3.d shows all the points which have been calculated to reach the constant feedrate while respecting the jerk constraint. One can see on Figure 3.e that the feedrate profile is really smooth. Figure 3.f demonstrates that the jerk limit is respected. The sequence of jerk presents peaks so it is not the one expected mathematically. But a discretized algorithm will always create this kind of solution as the optimal solution is not contained on the set of the discretized solutions.
There is no attempt to demonstrate the optimality of the solution as it depends on the time step Δt and on the scheme chosen to compute the discrete derivatives. The proposed algorithm is not designed to be optimal but it is robust and it gives a solution which is really close to the mathematical optimal solution. A better solution could be obtained by choosing points in the middle of the [s_{min}, s_{max}] interval at the expense of complexity and computation load.

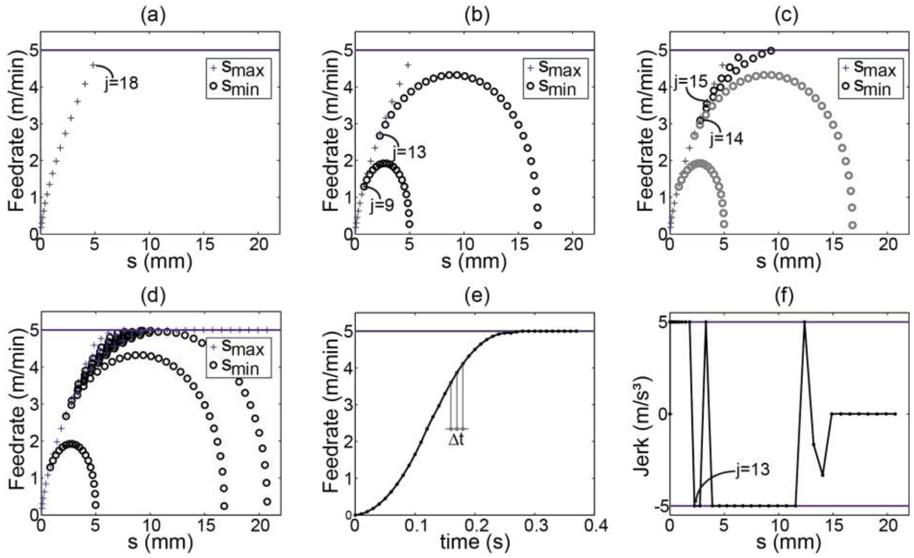


Figure 3: Details of the iterations needed to reach a constant feedrate on a straight line

3 SIMULATION AND EXPERIMENTAL RESULTS

The experiments are carried out on a 5-axis MIKRON UCP 710 machining center (Figure 7 top left) whose kinematical characteristics are given in Table 1. The machine is controlled by a SIEMENS 840D CNC which allows the measurement of the position and velocity of each axis during the movement. The cycle time of the position control loop is 6 ms.

The velocity, acceleration and jerk of each axis are computed based on the measurement of the position setpoints. It allows to get rid of the noise generated by the mechanical transmission which can be seen on the linear encoders. To make a fair comparison, the feedrate planning is realized with $\Delta t = 6$ ms and the derivatives are computed in the same manner using the derivative scheme of the Equation 6.

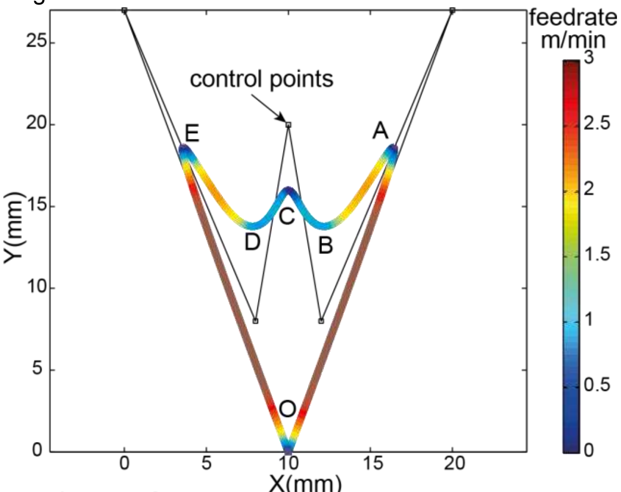
The computation time with an Intel Core Duo P8700 2.53 GHz will be given for information. The algorithm is designed to be used off-line but the computation time is satisfactory considering that the algorithm is developed in Matlab environment.

	X	Y	Z	A	C
V_{max} (m/min - rpm)	30	30	30	15	20
A_{max} (m/s^2 - rev/s^2)	2.5	3	2.1	0.83	0.83
J_{max} (m/s^3 - rev/s^3)	5	5	50	5	100

Table 1: Machine tool drive limits.

3.1 2-axis native B-Spline machining

The best solution is to use directly a native polynomial format instead of discretizing and reinterpolating a set of points. The NURBS format is well adapted for that purpose and CAM software offer the possibility to generate NURBS programs. This format facilitates the feedrate interpolation process as there is no need to modify the geometry that is why many research papers start from that format. Here the only thing to do is to find the relation between the path parameter u and the arc length s of the curve.



degree: 3

control points: $\begin{bmatrix} 10 & 20 & 12 & 10 & 8 & 0 & 10 \\ 0 & 27 & 8 & 20 & 8 & 27 & 0 \end{bmatrix}$

knot vector: [0 0 0 0 0.25 0.5 0.75 1 1 1 1]

Figure 4: Trident - tool path

The example of the third degree trident curve is inspired by the trident used in [14]. It has been modified because the original quadratic B-Spline was not C² continuous at the connexion between the arcs. This example is interesting because it contains linear portions where the programmed feedrate can be reached and sharp corners which are difficult to handle. Moreover, as the curve is symmetric, the feedrate profile should also by symmetric.

The Figure 4 shows the tool path colored as a function of the feedrate. Obviously, as it can be seen, the feedrate is directly linked to the curvature of the tool path. This is clear on a 2D tool path with two equivalent axes, but it will be shown further that it is different in multi-axis machining with linear and rotary drives. The feedrate profile is shown in Figure 5. Here one can see some differences with measured results on the machine which is not near optimal. Then for the X and Y axis, the velocity, acceleration and jerk are computed with the formula given in Equation 6. The results are presented on Figure 7, for this example the limiting parameters are the jerk of the axes and the programmed feedrate. Furthermore, one can see that the jerk limit is exceeded by the Siemens algorithm. But as this algorithm is a black box, it is only possible to guess at the reasons of this behavior. It can be verified that the jerk generated by the VPOp algorithm is within the authorized 120% jerk overrun limit. The measured machining time on the Mikron milling machine is 2.4 s, the planned machining time is 2.3 s and the computation time on a PC is 6.2 s.

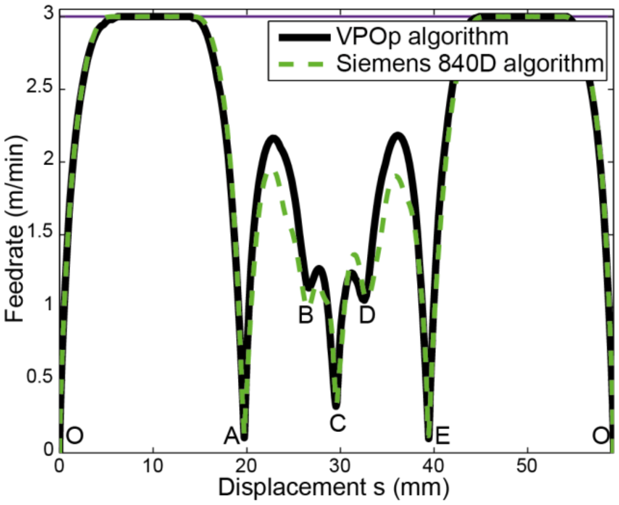


Figure 5: Trident - Feedrate profile

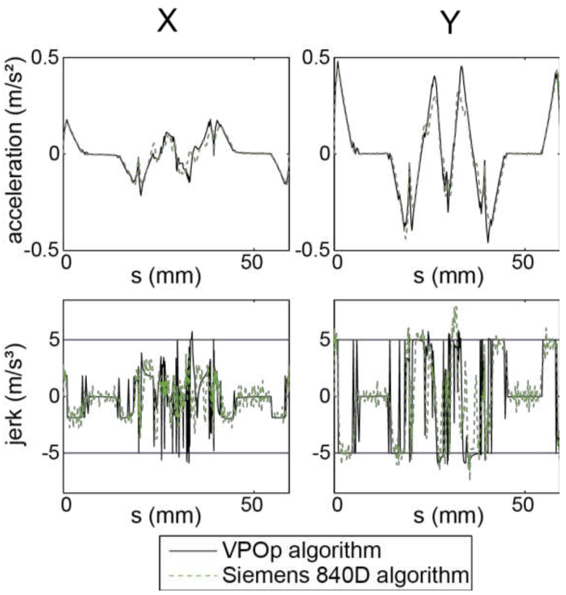
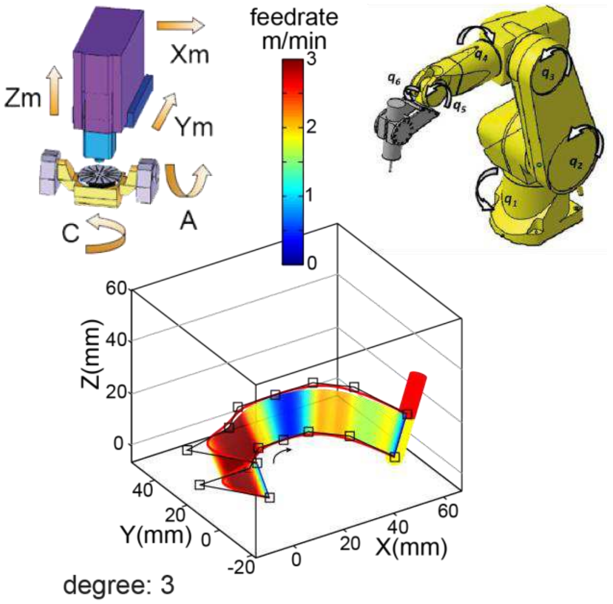


Figure 6: Trident - Constraints verification

3.2 5-axis native B-Spline machining with a 5-axis milling machine and a robot arm

To demonstrate the generality of the proposed algorithms, the feedrate planning is applied to control the 5-axis Mikron machine and a robot arm with six degree of freedom. As the application is a milling operation, the last joint of the robot is redundant with the others so its movement is arbitrarily locked. The kinematical characteristics correspond to those of the RX170B machining robot of Stäubli [13] (Figure 7 top right). The native B-Spline tool path described on Figure 7 represents a flank milling operation on an open pocket. To define a 5-axis tool path two curves with the same parametrization are needed: a bottom curve to define the tool tip trajectory and a top curve to define the tool orientation along the curve [17]. The parametrization of the B-Spline does not matter because the tool movement is discretized according to a near arc length evaluation. Here the only thing to do is to find the relation between the path parameter u and the arc length s of the curve.
The feedrate displayed on Figure 7 thanks to the colorbar is simulated using the characteristics of the robot. As it can be seen, the feedrate variations do not correspond to the curvature any more.

Figure 8 presents the results of the algorithm and the comparison with the measurement made on the Mikron machine. The derivatives are computed with Equation 6. So it proves that the algorithm respects the jerk constraint of each drive. One can see that the feedrate is mainly limited by the programmed feedrate, velocity and acceleration of the C-axis and jerk of X, Y and A axes. The tangential feedrate plot shows that the feedrate calculated by the VPOp algorithm is higher than the one measured on the Siemens 840D. But most of the time the results obtained by VPOp are similar to the one measured and this difference is probably due to the real time calculation constraint imposed to the industrial CNC. The measured machining time on the Mikron milling machine is 3.7 s, the planned machining time is 2.8 s and the computation time on a PC is 10 s.



degree: 3

bottom control points: $\begin{bmatrix} 5 & -10 & 10 & 20 & 30 & 40 & 50 & 55 \\ 0 & 20 & 20 & 30 & 30 & 30 & 20 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$

top control points: $\begin{bmatrix} 0 & -15 & 5 & 15 & 30 & 45 & 55 & 60 \\ 0 & 20 & 25 & 35 & 35 & 35 & 25 & 0 \\ 15 & 15 & 15 & 15 & 15 & 15 & 15 & 15 \end{bmatrix}$

knot vector: [0 0 0 0 0.2 0.4 0.6 0.8 1 1 1 1]

Figure 7: Open pocket - 5-axis machine and robot arm.

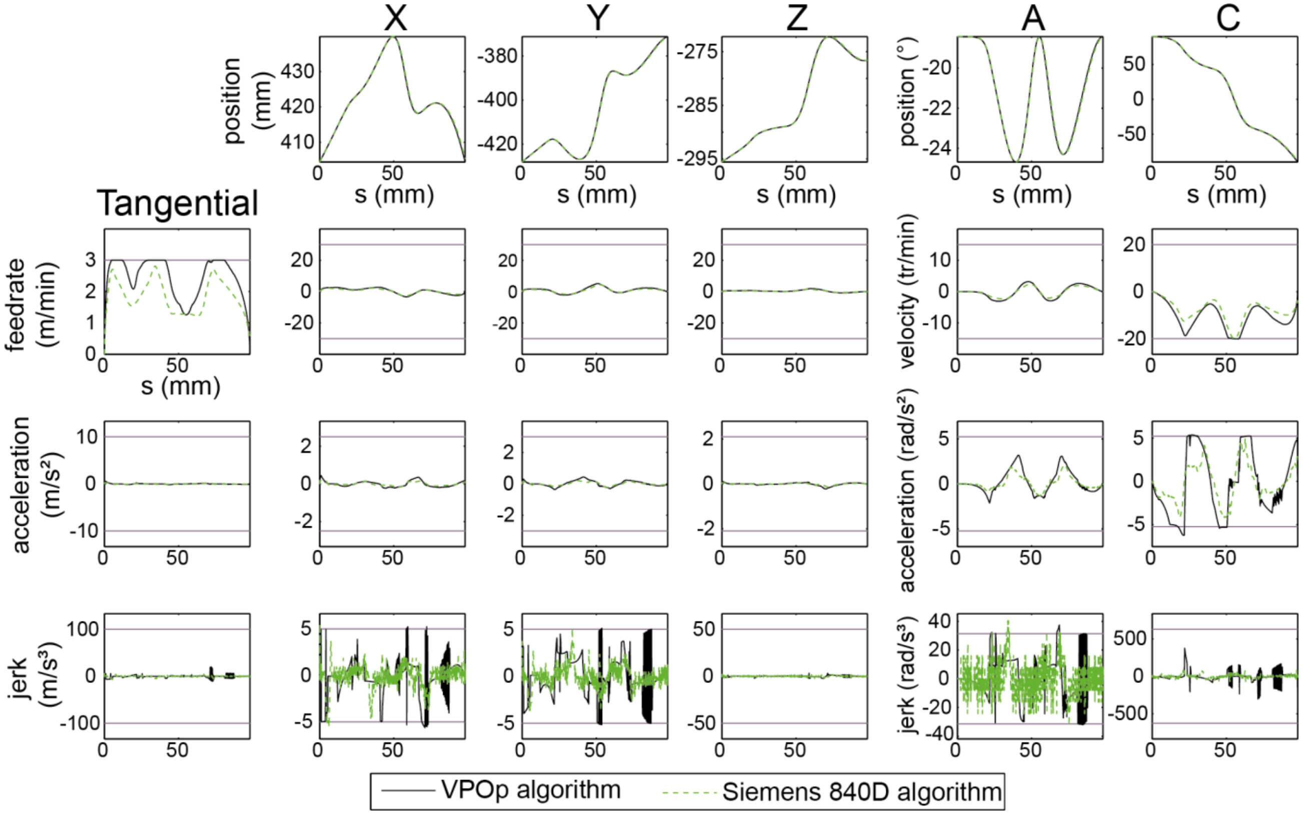


Figure 8: Open pocket - constraints verification on the 5-axis machine.

4 CONCLUSION

High speed milling requires smooth trajectory generation algorithms able to limit the jerk of each linear or rotary axis of the machine tool.

This paper presents a comprehensive solution which decouples the geometrical treatment of the programmed tool path and the feedrate interpolation. Indeed, if the tool path geometry is not at least C² continuous the jerk limitation cannot be respected. In case of NURBS tool path description, the geometry is already continuous so the algorithm just needs to find the relation between the arc length and the parameter. Once a correct geometry is obtained, the feedrate interpolation is performed on the axis movements. This iterative algorithm computes the intersection of the constraints given by each axis at each step. Thus a jerk limited velocity profile can be obtained for any kind of machine as soon as the kinematical transformation is known. Several examples demonstrate that the algorithm is efficient and that the jerk of each axis is really respected.

Further developments could use this feedrate interpolator in a 5-axis Open CNC machine tool and allow additional optimizations thanks to a precise control of the real machining feedrate. All the work presented here is implemented with Matlab and the VPOp software can be downloaded from

http://webserv.lurpa.ens-cachan.fr/geo3d/premium/vpop.

5 REFERENCES

[1] Shin, McKay, 1985, Minimum time control of robotic manipulators with geometric path constraints, IEEE Transactions on Automatic Control, 30/6:531-541.

[2] Renton, D., Elbestawi, M. A., 2000, High speed servo control of multi-axis machine tools, International Journal of Machine Tools and Manufacture, 40/4: 539-559.

[3] Timar, S. D., Farouki, R. T., Smith, T. S., Boyadjieff, C. L., 2005, Algorithms for time-optimal control of CNC machines along curved tool paths, Robotics and Computer-Integrated Manufacturing 21/1:37-53.

[4] Dong, J., Stori, J. A., 2006, A generalized time-optimal bidirectional scan algorithm for constrained feed-rate optimization, Journal of Dynamic Systems, Measurement, and Control 128:379-390.

[5] Barre, P.-J., Béarée, R., Borne, P., Dumetz, E., 2005, Influence of a jerk controlled movement law on the vibratory behaviour of high-dynamics systems, Journal of Intelligent and Robotic Systems 42:275-293.

[6] Liu, X., Ahmad, F., Yamazaki, K., Mori, M., 2005, Adaptive interpolation scheme for nurbs curves with the integration of machining dynamics, International Journal of Machine Tools and Manufacture 45/4-5:433-444.

[7] Dong, J., Ferreira, P., Stori, J., 2007, Feed-rate optimization with jerk constraints for generating minimum-time trajectories, International Journal of Machine Tools and Manufacture, 47/12-13:1941-1955.

[8] Erkorkmaz, K., Altintas, Y., 2001, High speed CNC system design part I: jerk limited trajectory generation and quintic spline interpolation, International Journal of Machine Tools and Manufacture, 41/9:1323-1345.

[9] Nam, S.-H., Yang, M.-Y., 2004, A study on a generalized parametric interpolator with real-time jerk-limited acceleration, Computer-Aided Design, 36:27-36.

[10] Lai, J.-Y., Lin, K.-Y., Tseng, S.-J., Ueng, W.-D., 2008, On the development of a parametric interpolator with confined chord error, feedrate, acceleration and jerk, The International Journal of Advanced Manufacturing Technology 37/1-2:104-121.

[11] Heng, M., Erkorkmaz, K., 2010, Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability, International Journal of Machine Tools and Manufacture, 50/3:281-293.

[12] Lee, A.-C., Lin, M.-T., Pan, Y.-R., Lin, W.-Y., 2011, The feedrate scheduling of NURBS interpolator for CNC machine tools, Computer-Aided Design, 43/6:612-628.

[13] Olabi, A., Béarée, R., Gibaru, O., Damak, M., 2010, Feedrate planning for machining with industrial six-axis robots, Control Engineering Practice, 18/5:471-482.

[14] Lin, M.-T., Tsai, M.-S., Yau, H.-T., 2007, Development of a dynamics-based NURBS interpolator with real-time look-ahead algorithm, International Journal of Machine Tools and Manufacture 47/15:2246-2262.

[15] Lavernhe, S., Tournier, C., Lartigue, C., 2008, Kinematical performance prediction in multi-axis machining for process planning optimization, The International Journal of Advanced Manufacturing Technology, 37/5-6:534-544.

[16] Sencer, B., Altintas, Y., Croft, E., 2008, Feed optimization for five-axis CNC machine tools with drive constraints, International Journal of Machine Tools and Manufacture, 48/7-8:733-745.

[17] Langeron, J. M., Duc, E., Lartigue, C., Bourdet, P., 2004, A new format for 5-axis tool path computation, using bspline curves, Computer-Aided Design 36/12:1219-1229.