International Journal of Machine Tools & Manufacture 57 (2012) 73–82

Contents lists available at SciVerse ScienceDirect

International Journal of Machine Tools & Manufacture

journal homepage: www.elsevier.com/locate/ijmactool

Feedrate interpolation with axis jerk constraints on 5-axis NURBS and G1 tool path

Xavier Beudaert, Sylvain Lavernhe, Christophe Tournier*

LURPA, ENS Cachan, Université Paris Sud 11, 61 av du pdt Wilson, 94235 Cachan, France

ARTICLE INFO

Article history:
Received 4 February 2012
Accepted 8 February 2012
Available online 16 February 2012

Keywords:
5-Axis machining
Feedrate
Jerk
Drive constraint
Velocity planning
CNC

ABSTRACT

A key role of the CNC is to perform the feedrate interpolation which consists in generating the setpoints sent to each axis of a machine tool based on a NC program. In high speed machining, the feedrate is limited by the velocity, acceleration and jerk of each axis of the machine tool.

The algorithm presented in this paper aims to obtain an optimized feedrate profile which makes best use of the kinematical characteristics of the machine. This minimum time feedrate profile is computed by intersecting all the constraints due to the drives in an iterative algorithm. It is worth noting that both tangential jerk and axis jerk are taken into consideration. The proposed VPOp (Velocity Profile Optimization) method is universal and can be applied to any articulated mechanical structure as it is demonstrated in the examples. Moreover the algorithm has been implemented for various formats: linear interpolation (G1) and NURBS interpolation in 3- and 5-axes. The effectiveness of the algorithm is demonstrated thanks to a comparison with an industrial CNC and can be freely tested using the VPOp software which is available on the Internet http://webserv.lurpa.ens-cachan.fr/geo3d/premium/vpop.

1. Introduction

This paper deals with the interpolation which is realized by the Computer Numerical Control (CNC). Indeed, starting from a NC program the CNC generates 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. As previously shown in [1], there are three main approaches to solve the problem of interpolation: (a) time-parameterization of geometric paths (decoupled approach), (b) combined path-planning and time parameterization (combined approach), (c) reactive and hybrid methods. The reactive and hybrid methods are used for example to drive a mobile robot which must adapt to its unknown environment. This method is not useful in this context as for machining operations the environment is known in advance. The combined approach is mainly used for robotic manipulators as it is well suited for trajectory without strong constraints on the contour error. Indeed, for a point to point movement, it is possible to infer the fastest tool path by planning each joint movement independently. With this method the kinematical characteristics of the joint impose the geometry of the tool path. For machining operations, the main goal is to follow precisely a given geometric path, for that reason, the decoupled approach is preferred. This approach consists in decoupling the geometry from the temporal evolution law. So the first part of the job is to create a "satisfying" geometry and then the feedrate interpolation is performed on that fixed geometry to generate the axis setpoints.

Computer-aided manufacturing (CAM) software generates the tool path first and then the CNC of the machine tool has to create the feedrate profile to follow this given geometry respecting the drive constraints. Industrially, G code is generally employed to describe the tool path. However, this description consists of segments connected with tangential discontinuities. So the discontinuities have to be rounded in order to be able to go through with non-null feedrate. That is why the first task of the numerical controller is to modify the geometry in order to round off the discontinuities. Indeed, the CNC is using the contour tolerance to go through the transitions between G1 blocks with a non-null feedrate. The best solution is to use a native mathematical description such as NURBS (Non-Uniform Rational Basis Splines) but unfortunately it is hardly ever used industrially. The polynomial formalism allows having a continuous path which will be interpolated by the CNC without any geometrical modification as opposed to the G1 description. To overcome this problem, the industrial CNCs offer the possibility to use real time compressors to create a spline with a set of G1 points (see CompCad/CompCurv in [2]).

Feedrate interpolation was first studied in robotics. For example, the problem of transition between segments was addressed in [3]. More recently, a corner rounding method mixing the

* Corresponding author.
E-mail address: tournier@lurpa.ens-cachan.fr (C. Tournier).

0890-6955/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ijmachtools.2012.02.005

Nomenclature

L	the length of the tool path (m)
s	path displacement (m) $s \in [0,L]$
$\dot{s}$	feedrate (m/s)
$\ddot{s}$	tangential acceleration (m/s^2)
$\dddot{s}$	tangential jerk (m/s^3)
F_{pr}	programmed feedrate (m/min)
A_{tan}	maximum tangential acceleration (m/s^2)
J_{tan}	maximum tangential jerk (m/s^3)

The following notation are illustrated on a 5-axis XYZAC structure.

$\mathbf{q} = [X(s)\ Y(s)\ Z(s)\ A(s)\ C(s)]^T$ axes position (m or rad)

$\dot{\mathbf{q}} = [\dot{X}(s)\ \dot{Y}(s)\ \dot{Z}(s)\ \dot{A}(s)\ \dot{C}(s)]^T$ axes velocity (m/s or rad/s)

$\ddot{\mathbf{q}} = [\ddot{X}(s)\ \ddot{Y}(s)\ \ddot{Z}(s)\ \ddot{A}(s)\ \ddot{C}(s)]^T$ axes acceleration (m/s^2 or rad/s^2)

$\dddot{\mathbf{q}} = [\dddot{X}(s)\ \dddot{Y}(s)\ \dddot{Z}(s)\ \dddot{A}(s)\ \dddot{C}(s)]^T$ axes jerk (m/s^3 or rad/s^3)

$\mathbf{q}_s = [X_s(s)\ Y_s(s)\ Z_s(s)\ A_s(s)\ C_s(s)^T$

$\mathbf{q}_{ss} = [X_{ss}(s)\ Y_{ss}(s)\ Z_{ss}(s)\ A_{ss}(s)\ C_{ss}(s)^T$

$\mathbf{q}_{sss} = [X_{sss}(s)\ Y_{sss}(s)\ Z_{sss}(s)\ A_{sss}(s)\ C_{sss}(s)^T$

first, second and third derivatives of the 5-axis positions with respect to the path displacement s

i	$i = 1..5$ for the X,Y,Z,A,C-axes of the machine tool
N	number of discretized calculation points
j	$j = 1..N$ discretized value along the path
s_j	jth discretized point along s

$\mathbf{V}_{max} = [v_{max,x}\ v_{max,y}\ v_{max,z}\ v_{max,a}\ v_{max,c}]^T$ axes velocity limits (m/s or rad/s)

$\mathbf{A}_{max} = [a_{max,x}\ a_{max,y}\ a_{max,z}\ a_{max,a}\ a_{max,c}]^T$ axes acceleration limits (m/s^2 or rad/s^2)

$\mathbf{J}_{max} = [j_{max,x}\ j_{max,y}\ j_{max,z}\ j_{max,a}\ j_{max,c}]^T$ axes jerk limits (m/s^3 or rad/s^3)

geometry and the time parametrization is proposed in [4] but it requires some assumptions especially for the feedrate at the entry and exit of the discontinuity. A method based on tool path/feedrate modification on the vicinity of the discontinuity was proposed in [5]. As it is mentioned above, it is preferable to decouple the geometry from the time parametrization. In tool path generation, [6] proposed a solution with a cubic B-Spline with six control points to round 3-axis tool paths. A good solution for 3-axis corner rounding is given in the Siemens patent [7] where polynomial splines are used to round the corners.

Once the geometrical treatment of the tool path is carried out, the feedrate planning has to be performed. The aim is to find an optimized feedrate profile which makes best use of the kinematical characteristics of the machine while following the given geometry.

Feedrate planning is a known issue and can be solved using different methods and taking into account different constraints. In robotics, Bobrow et al. [8] and Shin and McKay [9] 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 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 [10] worked on the velocity and acceleration limits determination. Timar et al. [11] used polynomial parametric curves on which they could obtain a closed form solution for the feedrate planning problem with axis acceleration constraints. Dong and Stori [12] 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 well known that jerk is an important parameter which should be considered as well. Indeed in many high speed milling operations jerk is the parameter which limits the feedrate variations.

The effect of jerk limitation on the mechanical structure was studied in detail by Barre et al. [13]. It is clear that jerk has to be limited to reduce the frequency content of the trajectory and to avoid exciting the natural modes of the structure. However, several articles are dealing with the tangential jerk only (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 jerk (third derivative with respect to the time of the axis movement) limit has to be considered too. Liu et al. [14] modified the feedrate profile to take into account the jerk and the natural frequencies of the machine tool. This method should be applied carefully to control the contour error generated.

Dong et al. [15] introduced 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 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 a corrective algorithm. The other main method to realize the feedrate planning with jerk constraints is to use a predefined profile as presented in Erkorkmaz and Altintas [16]. 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. [17–20]. Olabi et al. [21] applied the predefined profile to an industrial 6-axis machining robot. Lin et al. [22] introduced a cascade structure with a feedforward controller and the contour error in their interpolator. But the main problem is that the constraint on the predefined profile limits only the tangential derivatives. In practice each axis has its own limitations; furthermore with linear and rotary drives it is impossible to make the link between tangential jerk and axis jerk due to the non-linear kinematical transformation.

In the literature, few works can be found 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. [23,24] 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. [25] 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 it is difficult to apply it to G1 format.

The goal of this paper is to propose a unified feedrate planning solution for any machine tool structure with both jerk axis constraints and tangential jerk constraint along the tool path. A decoupled approach is used to separate the geometrical problem from the temporal interpolation problem. A one way interpolation method assuring the respect of all the constraints at each time step is used. This time discretization allows avoiding some further reinterpolation problems. The use of a look ahead based on the geometrical analysis of the tool path helps the interpolation in sharp zones. Both linear (G1) and polynomial (NURBS) tool path are handled in this paper. Furthermore, the mathematical formulation used here allows treating both linear and rotary axes in the same way. Hence, the algorithm is suitable for serial and parallel kinematic machine.

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

2. Feedrate planning algorithm

Fig. 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 along this fixed geometry 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 and of the tangential movement.

2.1. Mathematical formalism and drive constraints

Using the formula for the derivative of the composition of two functions (Eq. (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 Eqs. (2) and (3):

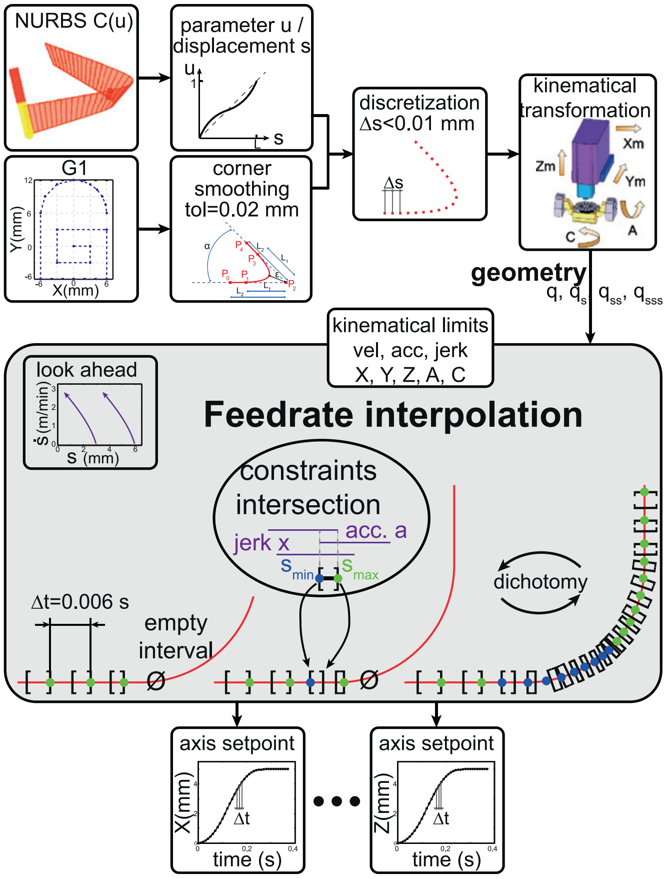

Fig. 1. VPOp algorithm.

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

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

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

$\mathbf{q}_s$, $\mathbf{q}_{ss}$, $\mathbf{q}_{sss}$ 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, etc.) 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. For example, for a 5-axis machine tool, the velocity constraints for each discretized point j along the tool path are presented in the equation

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

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

$$|\dot{q}_i^j| \le V_{max,i}, \quad |\ddot{q}_i^j| \le A_{max,i}, \quad |\dddot{q}_i^j| \le J_{max,i} \tag{5}$$

It is possible to obtain in a closed form an approximation of the upper limit of the feedrate. This limit is a good evaluation of the feedrate which can be used to optimize the tool path as in [26] but it is not sufficient to control a machine tool. Indeed, the need is to find a velocity profile which respects all the kinematical constraints without any approximation.

2.2. Discretization and principle of the algorithm

For a discretized algorithm, two discretizations are conceivable: a geometrical discretization in Δu (constant curve parameter step) or in Δs (constant path displacement step) as in [12,15] for example or a discretization in time Δt (constant time step). The problem with the discretization in Δs is that eventually the setpoints have to be send to the controller with a fixed frequency so a reinterpolation is needed. At really low feedrate a small Δs increment is needed 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. The quintic spline interpolation presented in [16] will also have this problem to reinterpolate the points computed in Δs so it cannot ensure that the jerk limitation will not be exceeded. That is why a constant time step has been chosen for the VPOp algorithm presented here. Hence, the velocity, acceleration and jerk are computed as

$$\dot{s}_{j+1} = \frac{s_{j+1}-s_j}{\Delta t}, \quad \ddot{s}_{j+1} = \frac{\dot{s}_{j+1}-\dot{s}_j}{\Delta t}, \quad \dddot{s}_{j+1} = \frac{\ddot{s}_{j+1}-\ddot{s}_j}{\Delta t} \tag{6}$$

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 [8,9]. Using the discretization, each constraint can be reduced to a polynomial inequations (7)–(9). The unknown parameter is s_{j+1}, and f are functions of the geometrical derivatives. Solving the inequation, an interval over which the constraint is verified is obtained. Finally, as shown in Fig. 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. It is possible to have a solution which is a simple interval or a union of distinct intervals $[s_{min},s_2] \cup [s_3,s_{max}]$. In this case, the minimum and maximum allowable solutions s_{min} and s_{max} are kept. This interval can also 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 to reduce the feedrate:

$$-V_{max,i} \le q_{s,i}^{j+1} \cdot \frac{s_{j+1}-s_j}{\Delta t} \le V_{max,i} \tag{7}$$

$$-A_{max,i} \le s_{j+1}^2 \cdot f_{A2} + s_{j+1} \cdot f_{A1} + f_{A0} \le A_{max,i} \tag{8}$$

$$-J_{max,i} \le s_{j+1}^3 \cdot f_{J3} + s_{j+1}^2 \cdot f_{J2} + s_{j+1} \cdot f_{J1} + f_{J0} \le J_{max,i} \tag{9}$$

The functions f_{A0}, f_{A1}, f_{A2} used in Eq. (8) are detailed below Eqs. (10)–(12). Those equations are obtained using Eqs. (1)–(3) and (6). The functions f_{J0}, f_{J1}, f_{J2}, f_{J3} can be obtained in the same way. Those functions depend on the positions s_j, s_{j-1}, s_{j-2} calculated in the previous iterations and on the geometrical derivative at the point $j+1$ which will be approximated as explained in Section 2.6:

$$f_{A2} = \frac{q_{ss,i}^{j+1}}{\Delta t^2} \tag{10}$$

$$f_{A1} = \frac{1}{\Delta t^2} \cdot (q_{s,i}^{j+1} - q_{ss,i}^{j+1} \cdot 2s_j) \tag{11}$$

$$f_{A0} = \frac{1}{\Delta t^2} \cdot (q_{ss,i}^{j+1} \cdot s_j^2 + q_{s,i}^{j+1}(s_{j-1} - 2s_j)) \tag{12}$$

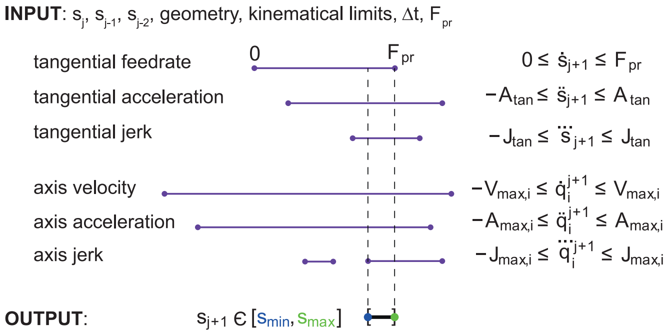
INPUT: s_j, s_{j-1}, s_{j-2}, geometry, kinematical limits, Δt, F_{pr}
tangential feedrate
0
F_{pr}
$0 \le \dot{s}_{j+1} \le F_{pr}$
tangential acceleration
$-A_{tan} \le \ddot{s}_{j+1} \le A_{tan}$
tangential jerk
$-J_{tan} \le \dddot{s}_{j+1} \le J_{tan}$
axis velocity
$-V_{max,i} \le \dot{q}_i^{j+1}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

Fig. 2. Constraints intersection principle.

2.3. Detailed explanation of the algorithm

The iteration principle to find the correct sequence of s_{min}, s_{max} is briefly described in Fig. 1. 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. Fig. 3 shows each calculated point. In Fig. 3a, one can see that the algorithm starts going as fast as possible until a point with no solution (empty intersection) is reached after $j=18$. 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 allows reaching the programmed feedrate without exceeding it. 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 found with a dichotomy. In Fig. 3b 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

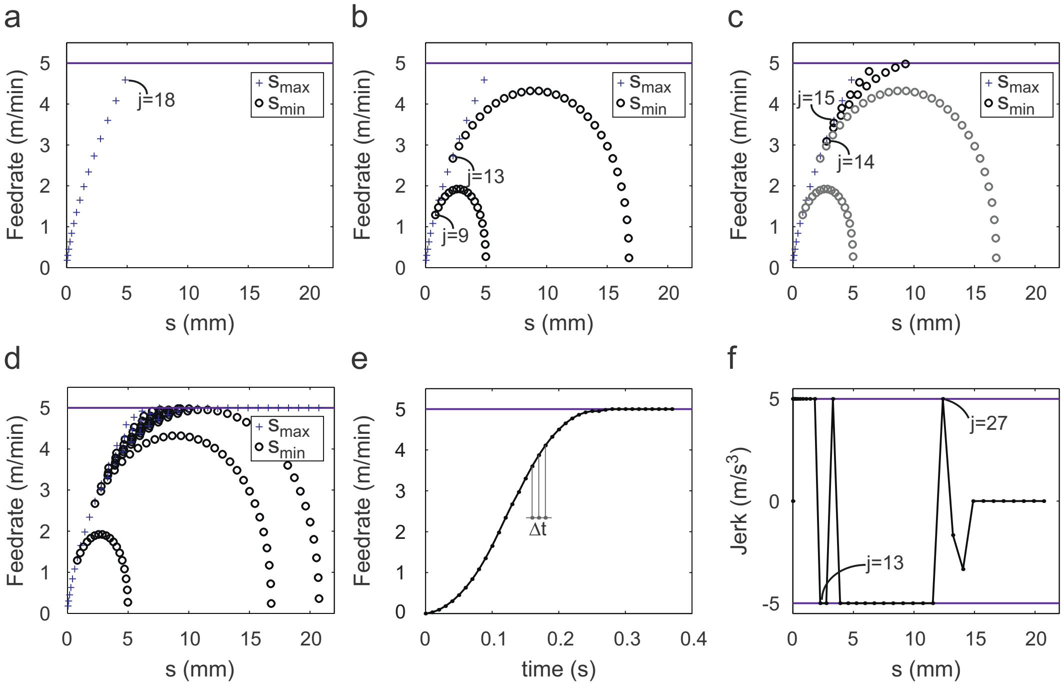

Fig. 3. Details of the iterations needed to reach a constant feedrate on a straight line (with jerk constraint).

in Fig. 3c $j=15$ 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, Fig. 3d shows all the points which have been calculated to reach the constant feedrate while respecting the jerk constraint. One can see in Fig. 3e that the feedrate profile is really smooth. Fig. 3f demonstrates that the jerk limit is respected, after the point $j=27$ the limiting parameter is the programmed feedrate. 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. 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.

The same kind of problems will be encountered when a sharp corner is reached with a high feedrate as in the first trial of Fig. 1. So using the same technique of dichotomy, the first point from where it is possible to reach $\dot{s}=0$ has to be found. If it is possible to find a sequence to reach $\dot{s}=0$, it will be possible to find a sequence which respects all the constraints on any geometry (including any potential upcoming sharp corner). Here, it is assumed that if it is possible to reach $\dot{s}=0$, it will always be possible to go further. In reality, the algorithm sometimes needs to go back even more because the deceleration is too high to be able to stop at $\dot{s}=0$ with $\ddot{s}=0$.

2.4. Algorithm

The detailed algorithm is written below:

```
Prepare the geometry of the tool path
Apply the look ahead for each G1 transition and at the end of the tool path
while the end of the tool path is not reached do
    Compute the intersection of the intervals
    if intersection is empty then
        while the end of the dichotomy is not reached do
            Use a dichotomy to find the point $j$ where $s_j=s_{min}$ has to be chosen. Indeed as there is no solution, the first point from where it is possible to reach $\dot{s}=0$ taking $s_{min}$ for each step has to be found. To speed up the algorithm, this research is realized with a dichotomy.
        end while
    else
        $s_{j+1}=s_{max}$
    end if
end while
```

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.

2.5. Look ahead

With a G1 tool path, it is known in advance that the feedrate will be minimum at the discontinuities. So a look ahead strategy is used in order to reduce the number of iterations. The look ahead consists in calculating the necessary feedrate reduction to approach a sharp corner. Starting from the middle of each transition with zero acceleration and jerk, this look ahead uses the algorithm previously described but applied in the backward direction. As opposed to a two pass algorithm, there is no need to connect this backward limit to the previously calculated points because it is considered as another limit which is added to the constraints in the intersection algorithm (see Fig. 8). This principle of look ahead is also used at the end of the tool path to decelerate to rest.

With B-Spline tool path, there is no indication about the potential sharp corners so this look ahead method cannot be applied in advance. Moreover, in 5-axis, the curvature does not mean much. Indeed, it is necessary to take into account the inverse kinematical transformation and the kinematical characteristics of each axis. So the look ahead is used once the algorithm has detected the problematic sharp corner.

2.6. Approximation of the geometrical derivatives

In Eqs. (7)–(9), the geometrical derivatives in $j+1$ are used. So these terms have to be approximated as they depend on s_{j+1} which is the unknown. As the evolution along the tool path is smooth, it is possible to have a good approximation of s_{j+1} with Eq. (13). Nevertheless because of the error made on the value of the geometrical derivatives, the jerk constraints can be slightly exceeded in the high curvature area. Indeed, due to the numerical derivation (Eq. (6)) a really tiny variation in the position (< 1 μm) can lead to an important variation of jerk.

$$q_s^{j+1}=q_s(s_{j+1})\simeq q_s(s_j+\Delta t(\dot{s}_j+\Delta t\ddot{s}_j)) \tag{13}$$

To overcome this problem of jerk overruns, the magnitude of the jerk is checked at each iteration with the time discretized formula equation (14). If the jerk is too high, the value of q_i^{j+1} is corrected by inverting this equation with J_i^{j+1} equal to the jerk limit. With this verification, it is possible to ensure that the jerk of the axes will never exceed a given value parametrized by an authorized jerk overrun usually set to 120%:

$$J_i^{j+1}=\frac{q_i^{j+1}-3q_i^j+3q_i^{j-1}-q_i^{j-2}}{\Delta t^3} \tag{14}$$

3. Simulation and experimental results

The experiments are carried out on a 5-axis MIKRON UCP 710 machining center (Fig. 9 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 setpoint sent to the controller during the movement. The minimum cycle time of the position control loop is 6 ms. This is enough for the comparison carried out here even if last generation CNC can reach a cycle time of 1 ms.

The velocity, acceleration and jerk of each axis are computed based on the measurement of the position setpoints. It allows getting rid of the noise generated by the mechanical transmission which can be seen on the linear encoders. The cycle time which can be used in VPOp is not limited, but to make a fair comparison the feedrate planning is realized with $\Delta t=6$ ms. Moreover the derivatives are computed in the same manner using the derivative scheme of Eq. (6).

Table 1
Machine tool drive limits.

	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

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

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.

The example of the third degree trident curve is inspired by the trident used in [22]. It has been modified because the original quadratic B-Spline was not C^2 continuous at the connection between the arcs. The control points are given in Appendix A. 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.

Fig. 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 Fig. 5. Here one can see some differences with measured results on the machine which is not near optimal. Then for the X- and Y-axes, the velocity, acceleration and jerk are computed with the formula given in Eq. (6). The results are presented in Fig. 6, for this example the limiting parameters are the jerk of the axes and the programmed feedrate. Furthermore, on the bottom plots, 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 120% authorized jerk overrun limit. The measured machining time on the Mikron milling machine is 2.4 s, the planned machining time is 2.3 s. For information, the computation time on a PC is 6.2 s.

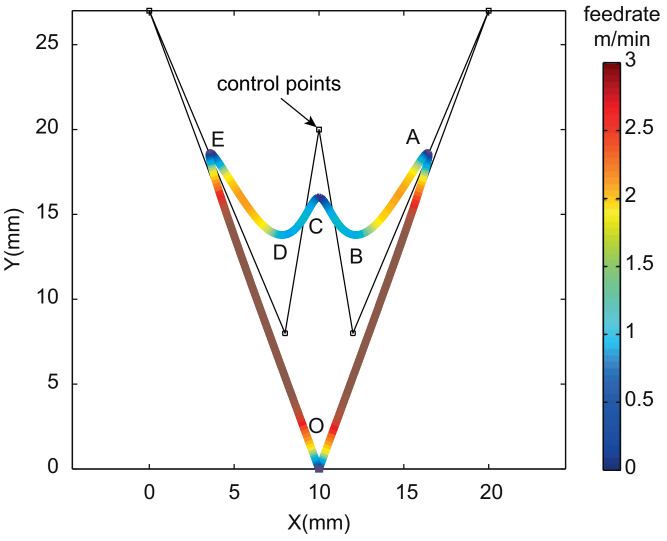

Fig. 4. Trident—tool path.

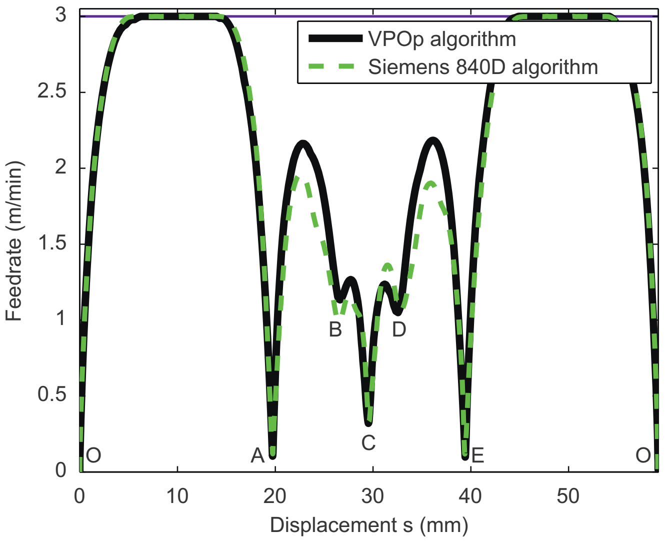

Fig. 5. Trident—feedrate.

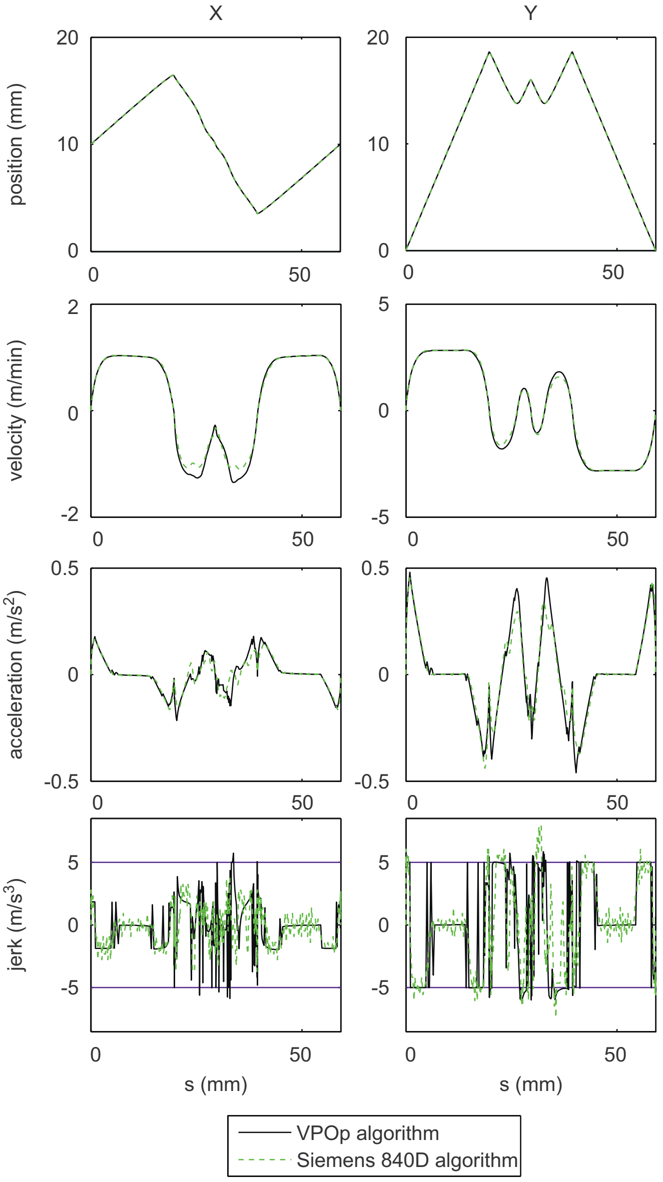

Fig. 6. Trident—constraints verification.

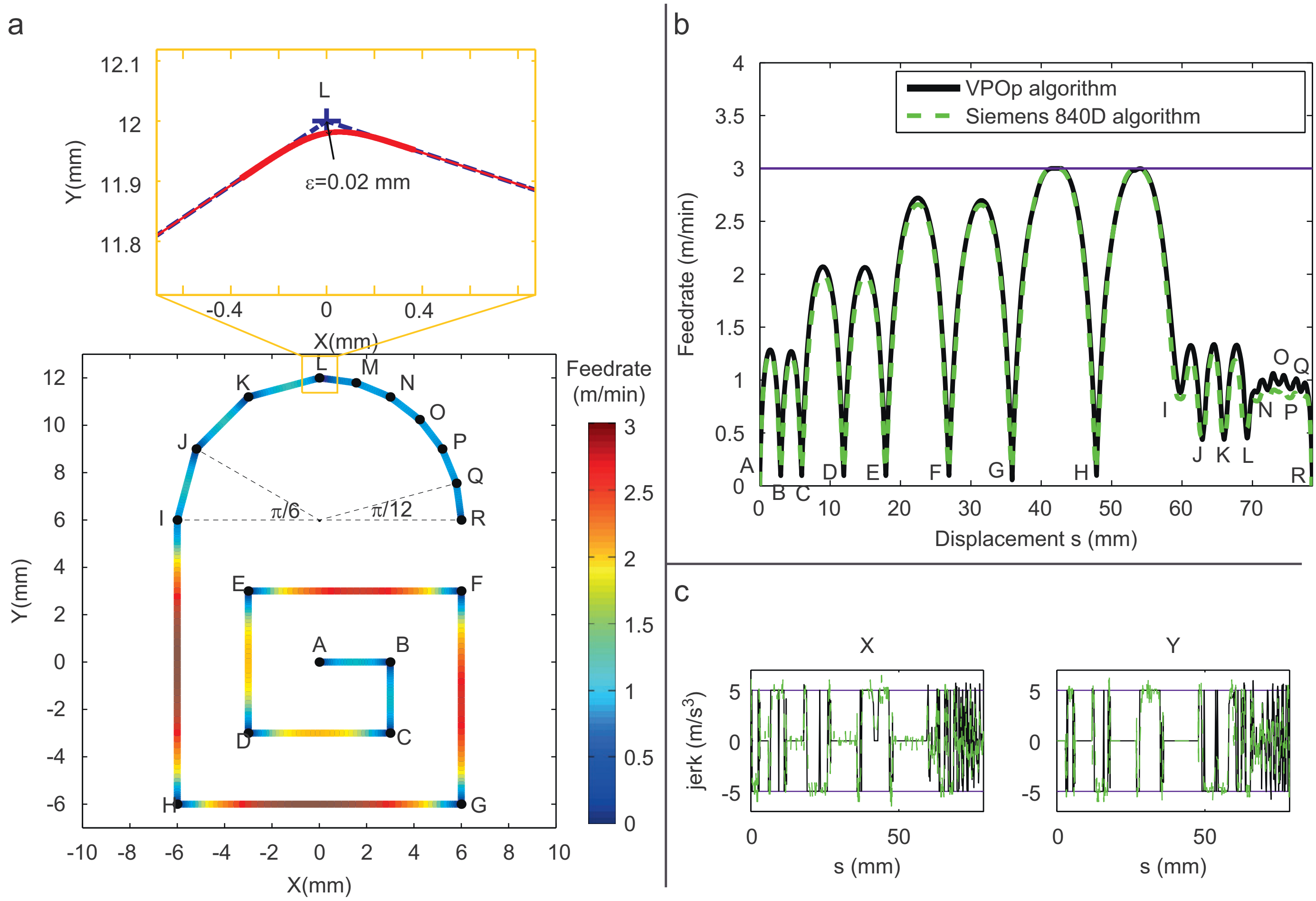

Fig. 7. (a) Lock tool path, (b) feedrate profile, (c) jerk constraints verification.

3.2. 2-Axis G1 machining

The geometrical treatment and the feedrate planning of a G1 tool path are presented on the lock tool path in Fig. 7a. This example contains short segments upon which the programmed feedrate cannot be reached and a discretized arc of circle which emphasized the importance of the discretization. So it contains the typical characteristics of a contouring strategy tool path. On top of Fig. 7, one can see that the contour tolerance is set to $\varepsilon = 0.02$ mm. To have a C^2 continuous transition, a cubic B-Spline with five control points is used. With really short segments, it will be impossible to use the whole tolerance.

Fig. 8 shows the look ahead limits in red for the first two discontinuities (points B and C). With this look ahead technique, it is possible to have a symmetric velocity profile around the discontinuities. Fig. 7b allows comparing the results given by the Siemens 840D CNC and by the Velocity Planning Optimization (VPOp) algorithm presented here. The solutions are really close as they are both near optimal. One should notice that the behavior is really similar even for the discretized circle (points I to R). So this algorithm can also be used to predict the real feedrate.

The oscillations of the feedrate (points I to R) are due to the geometry of the tool path. Those oscillations could be harmful for the quality of the machined part. For points J,K,L the length of the segments allows a higher feedrate but sharp corners require a low feedrate. For points M to R the segment length is shorter but thanks to the corner rounding algorithm the tool path is smoother. Hence a lower feedrate fluctuation is obtained. It is important to notice that the discretization used for G1 tool path has an important effect on the feedrate.

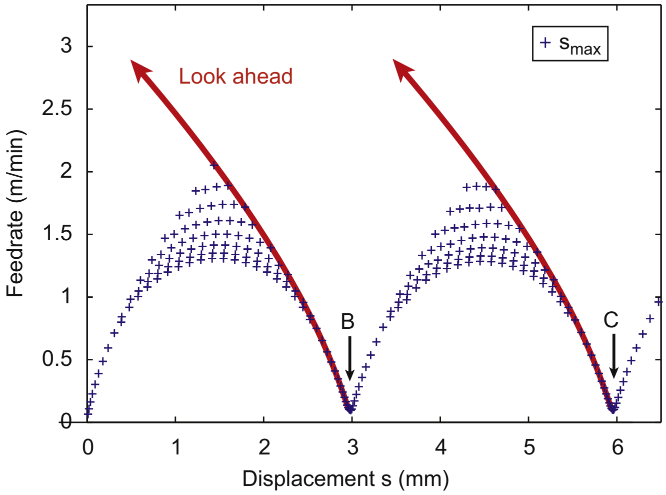

Fig. 8. Look ahead for the lock.

The measured machining time on the Mikron milling machine is 4.5 s, the planned machining time is 4.2 s. For information, the computation time on a PC is 5.8 s. Finally, jerks are represented in Fig. 7c. The velocity and acceleration are not represented because the only limiting parameters are the jerks of the *X*- and *Y*-axes. It can be verified that the axis jerk constraints are within the authorized jerk overrun limits.

3.3. 5-Axis native B-Spline machining with a 5-axis milling machine and a robot arm

To demonstrate the generality of the proposed algorithm, 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 [21] (Fig. 9 top right). The native B-Spline tool path described in Fig. 9 represents a flank milling operation on an open pocket. The control points are given in Appendix B. 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 [27]. The parametrization of the B-Spline does not matter because the tool

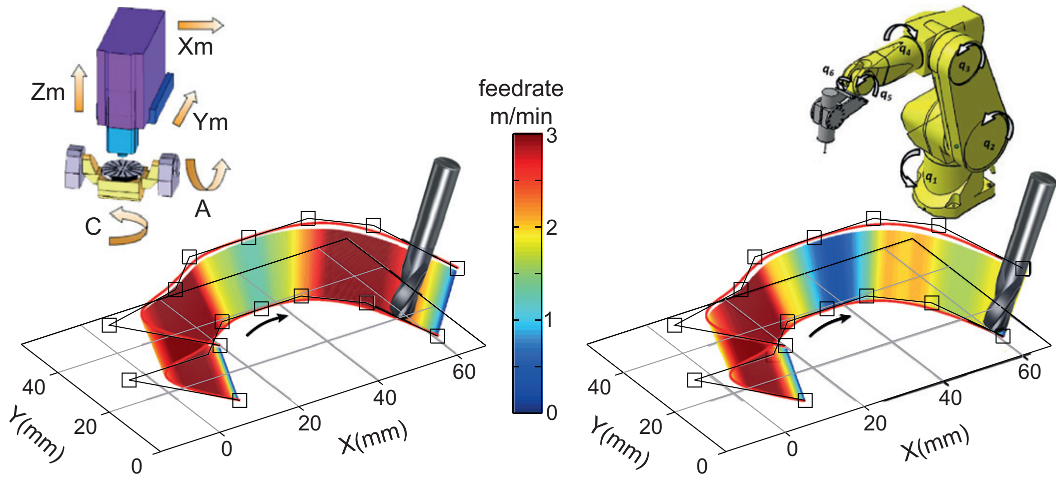

Fig. 9. Open pocket with 5-axis machine and robot arm.

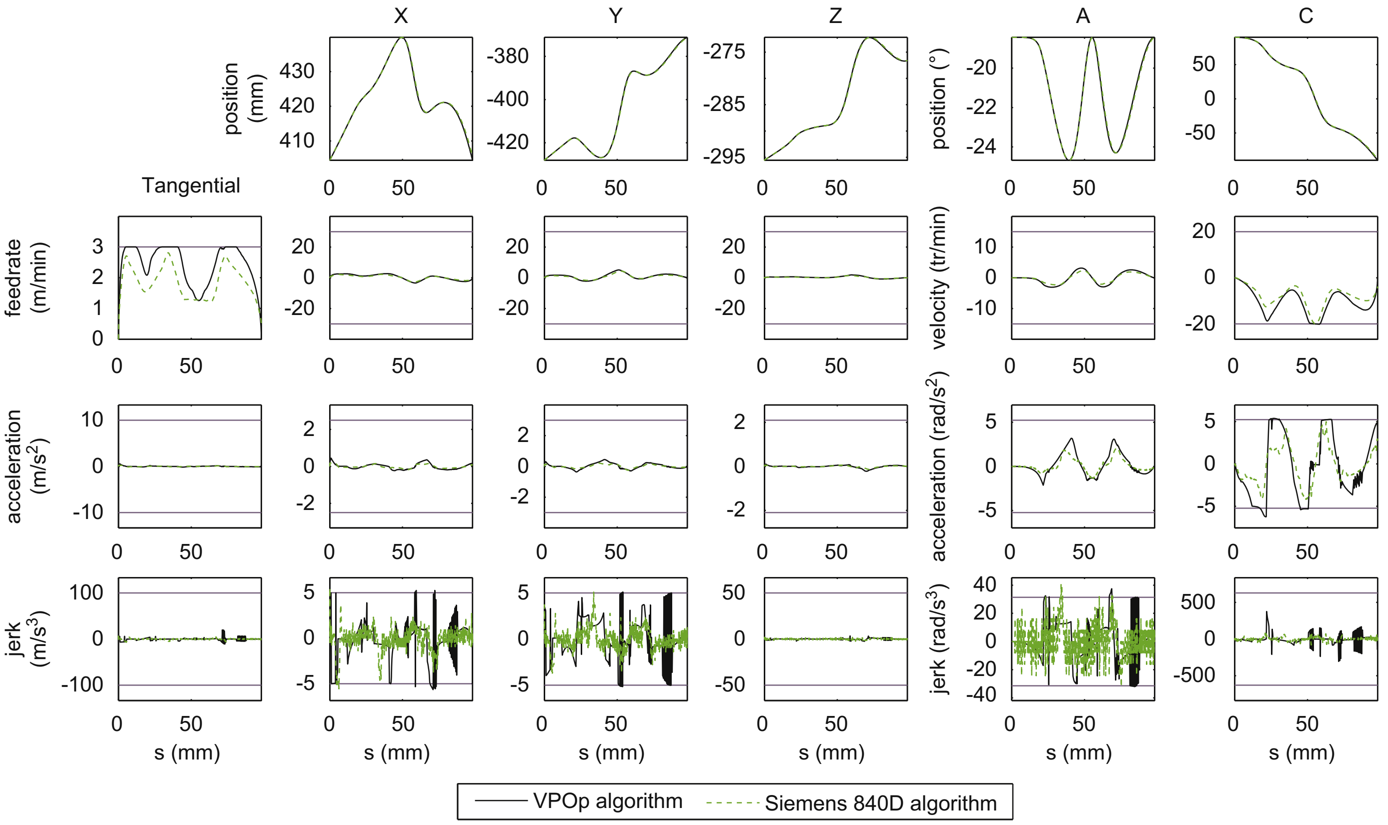

Fig. 10. Open pocket—constraints verification on the 5-axis machine.

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 in Fig. 9 thanks to the colorbar is simulated both for the 5-axis Mikron machine and for the robot. As it can be seen, the feedrate variations do not correspond to the curvature any more. In multi-axis it depends on the movement of each joint and on its kinematical characteristics.

Fig. 10 presents the results of the algorithm and the comparison with the measurement made on the Mikron machine. The derivatives are computed with Eq. (6). 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. For information, the computation time on a PC is 10 s.

The geometrical treatment of the programmed tool path is carried out in the part coordinate system so the machine structure has no effect on it. Once the geometry is obtained, the feedrate interpolation is performed on the axis movements. Thus a jerk limited velocity profile can be obtained for redundant or parallel structures without any problem as soon as the kinematical transformation is given.

All the algorithms presented here have been implemented in the VPOp software which is available on the Internet http://webserv.lurpa.ens-cachan.fr/geo3d/premium/vpop. Interested readers will find all the details about the examples used here and will be able to run the algorithms on their personal computer.

4. Conclusion

High Speed Machining involves high velocities and accelerations which can be harmful both for the machine and for the surface quality of the workpiece. To solve this problem it is necessary to control each axis kinematical parameters (velocity, acceleration and especially jerk) as well as the velocity, acceleration and jerk of the tool-workpiece movement. The control of these parameters is often carried out at the expense of productivity, without taking advantage of machining center capabilities.

To overcome this issue, this paper presents a unified and efficient solution to minimize the machining time by making best use of the kinematical performances of the machine tool (velocity, acceleration and jerk along the tool path and for each axis).

Our method is based on a decoupled approach which separates the problem of geometrical treatment of the programmed tool path and of feedrate interpolation. In the first stage, the local rounding of the geometry is achieved according to well-known strategies. The novelty in solving the global problem lies in the treatment performed for the feedrate interpolation considering the previously defined geometry. Indeed, the VPOp method uses the principle of constraints intersection to freely add various limitations and especially the jerk of each axis. The one way algorithm with a look ahead and a dichotomy allows us to avoid the connection problems of a two pass (forward and backward) algorithm. This iterative algorithm computes the intersection of the constraints given by each rotary or linear axis at each time step. This time discretization, which do not need a reinterpolation, ensures that the constraints are really respected.

Several examples in 3 to 5-axis demonstrate that the algorithm is efficient and that the jerk of each axis is respected. The results are compared with the measurements made on the 5-axis milling machine equipped with a Siemens 840D CNC. Finally, an example with a robot arm demonstrates that the proposed method could be widely used. Hence, the VPOp method improves the previously proposed interpolation techniques and can be applied to different machine tool structures and different tool path descriptions (linear and polynomial).

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. The VPOp software can be downloaded from http://webserv.lurpa.ens-cachan.fr/geo3d/premium/vpop.

Acknowledgments

This work is supported by the French Ministry of Research and the Digiteo foundation (2010-47D).

Appendix A. Trident curve

B-Spline degree: 3
Control points:

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

Knot vector:

$[0\ 0\ 0\ 0\ 0.25\ 0.5\ 0.75\ 1\ 1\ 1\ 1]$

Appendix B. Open pocket curve

B-Spline 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:

$[0\ 0\ 0\ 0\ 0.2\ 0.4\ 0.6\ 0.8\ 1\ 1\ 1\ 1]$

References

[1] G. Pardo-Castellote, R.H. Cannon, Proximate time-optimal algorithm for online path parameterization and modification, in: IEEE International Conference on Robotics and Automation, vol. 2 (2), 1996, pp. 1539–1546.

[2] Siemens, Sinumerik—5 axis machining, 2009.

[3] R. Paul, Manipulator Cartesian path control, IEEE Transactions on Systems, Man, and Cybernetics 9 (11) (1979) 702–711.

[4] X. Pessoles, Y. Landon, W. Rubio, Kinematic modelling of a 3-axis NC machine tool in linear and circular interpolation, International Journal of Advanced Manufacturing Technology 47 (5–8) (2010) 639–655.

[5] C. Ernesto, R. Farouki, High-speed cornering by CNC machines under prescribed bounds on axis accelerations and toolpath contour error, International Journal of Advanced Manufacturing Technology 58 (1) (2012) 327–338.

[6] V. Pateloup, E. Duc, P. Ray, B-Spline approximation of circle arc and straight line for pocket machining, Computer-Aided Design 42 (9) (2010) 817–827.

[7] S.J. Yutkowitz, W. Chester, Apparatus and Method for Smooth Cornering in a Motion Control System, United States, Siemens Energy & Automation, Inc, Alpharetta, GA, 2005 (US Patent 6922606).

[8] J. Bobrow, S. Dubowsky, J. Gibson, Time-optimal control of robotic manipulators along specified paths, International Journal of Robotics Research 4 (3) (1985) 3–17.

[9] K. Shin, N. McKay, Minimum time control of robotic manipulators with geometric path constraints, IEEE Transactions on Automatic Control 30 (6) (1985) 531–541.

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

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

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

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

[14] X. Liu, F. Ahmad, K. Yamazaki, M. Mori, Adaptive interpolation scheme for NURBS curves with the integration of machining dynamics, International Journal of Machine Tools and Manufacture 45 (4–5) (2005) 433–444.

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

[16] K. Erkorkmaz, Y. Altintas, High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation, International Journal of Machine Tools and Manufacture 41 (9) (2001) 1323–1345.

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

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

[19] M. Heng, K. Erkorkmaz, Design of a NURBS interpolator with minimal feed fluctuation and continuous feed modulation capability, International Journal of Machine Tools and Manufacture 50 (3) (2010) 281–293.

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

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

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

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

[24] S. Lavernhe, C. Tournier, C. Lartigue, Optimization of 5-axis high-speed machining using a surface based approach, Computer-Aided Design 40 (10–11) (2008) 1015–1023.

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

[26] X. Beudaert, P.-Y. Pechard, C. Tournier, 5-Axis tool path smoothing based on drive constraints, International Journal of Machine Tools and Manufacture 51 (12) (2011) 958–965.

[27] J.M. Langeron, E. Duc, C. Lartigue, P. Bourdet, A new format for 5-axis tool path computation, using B-Spline curves, Computer-Aided Design 36 (12) (2004) 1219–1229.