



**HAL**  
open science

## 5-Axis tool path smoothing based on drive constraints

Xavier Beudaert, Pierre-Yves Pechard, Christophe Tournier

► **To cite this version:**

Xavier Beudaert, Pierre-Yves Pechard, Christophe Tournier. 5-Axis tool path smoothing based on drive constraints. *International Journal of Machine Tools and Manufacture*, 2011, 51 (12), pp.958-965. 10.1016/j.ijmachtools.2011.08.014 . hal-00626187

**HAL Id: hal-00626187**

**<https://hal.science/hal-00626187>**

Submitted on 24 Sep 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# 5-axis Tool Path Smoothing Based on Drive Constraints

Xavier Beudaert<sup>a</sup>, Pierre-Yves Pechard<sup>b</sup>, Christophe Tournier<sup>a,\*</sup>

<sup>a</sup>LURPA, ENS Cachan, Université Paris Sud 11  
61 av du pdt Wilson, 94235 Cachan, France

<sup>b</sup>Missler Software, 7 Rue du Bois Sauvage, 91055 Evry, France

## Abstract

In high speed machining, the real feedrate is often lower than the programmed one. This reduction of the feedrate is mainly due to the physical limits of the drives, and affects machining time as well as the quality of the machined surface. Indeed, if the tool path presents sharp geometrical variations the feedrate has to be decreased to respect the drive constraints in terms of velocity, acceleration and jerk. Thus, the aim of this paper is to smooth 5-axis tool paths in order to maximize the real feedrate and to reduce the machining time.

Velocity, acceleration and jerk limits of each drive allow to compute an evaluation of the maximum reachable feedrate which is then used to localize the areas where the tool path has to be smoothed. So starting from a given tool path, the proposed algorithm iteratively smoothes the joint motions in order to raise the real feedrate. This algorithm has been tested in 5-axis end milling of an airfoil and in flank milling of an impeller for which a N-buffer algorithm is used to control the geometrical deviations. An important reduction of the measured machining time is demonstrated in both examples.

**Keywords:** 5-axis machining, tool path smoothing, drive constraints, machine tool kinematic

## 1. Introduction

Within the context of high speed machining, the tool path and machine motions have to be smooth to achieve the required surface quality. Mathematically, the smoothness is usually defined by a continuous second derivative. But it is important to make a clear distinction between a smooth tool path and a smooth motion. A smooth tool path considers only the geometry which means a second derivative with respect to a geometrical parameter (the displacement for example) whereas a smooth motion deals with the temporal movement (i.e. the second derivative with respect to the time). Indeed, you can have a jerky motion even along a straight line or a really quiet travel along a curvy path.

In the literature, several articles deal with the smoothness of the motion. First works were carried out by

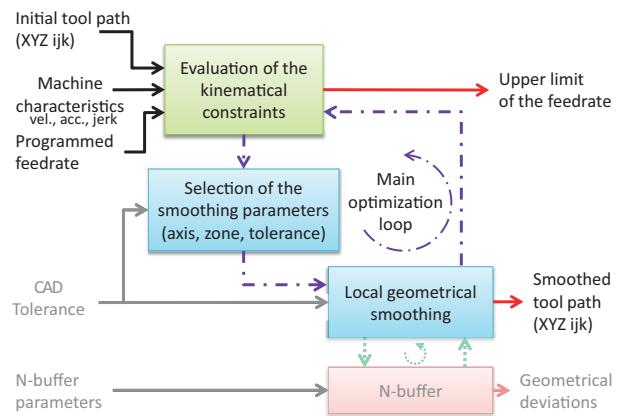


Figure 1: Block diagram of the algorithm.

robotics researchers [1, 2]. Nowadays, it is important to limit the jerk in the trajectory planning to produce a soft motion [3–7]. The aim of these articles is to find a velocity profile which respects all the kinematical constraints of the drives and of the machine tool structure for a given tool path. Industrial numerical controllers also offer the possibility to have a jerk limited motion along the tool path [8].

\*Corresponding author Tel.: 33 1 47 40 29 96; Fax: 33 1 47 40 22 11.

Email address: tournier@lurpa.ens-cachan.fr  
(Christophe Tournier)

**Nomenclature:**

$L$  the length of the tool path

$s$  path displacement ( $mm$ )  $s \in [0, L]$

$\dot{s}$  feedrate ( $mm/s$ )

$\ddot{s}$  tangential acceleration ( $mm/s^2$ )

$\dddot{s}$  tangential jerk ( $mm/s^3$ )

$F_{pr}$  programmed feedrate ( $mm/min$ )

$$\mathbf{q} = [X(s) Y(s) Z(s) A(s) C(s)]^T$$

axes position ( $mm$ ) and ( $degree$ )

$$\dot{\mathbf{q}} = [\dot{X}(s) \dot{Y}(s) \dot{Z}(s) \dot{A}(s) \dot{C}(s)]^T$$

axes velocity ( $mm/s$ ) and ( $rpm$ )

$$\ddot{\mathbf{q}} = [\ddot{X}(s) \ddot{Y}(s) \ddot{Z}(s) \ddot{A}(s) \ddot{C}(s)]^T$$

axes acceleration ( $mm/s^2$ ) and ( $rad/s^2$ )

$$\dddot{\mathbf{q}} = [\dddot{X}(s) \dddot{Y}(s) \dddot{Z}(s) \dddot{A}(s) \dddot{C}(s)]^T$$

axes jerk ( $mm/s^3$ ) and ( $rad/s^3$ )

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

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

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

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 displacement  $s$ . Here the discretization step along the displacement is set to 0.01 mm

$j_s$  :  $j^{th}$  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 ( $m/s$ ) and ( $rpm$ )

$$\mathbf{A}_{max} = [a_{max}^x a_{max}^y a_{max}^z a_{max}^a a_{max}^c]^T$$

axes acceleration limits ( $m/s^2$ ) and ( $rot/s^2$ )

$$\mathbf{J}_{max} = [j_{max}^x j_{max}^y j_{max}^z j_{max}^a j_{max}^c]^T$$

axes jerk limits ( $m/s^3$ ) and ( $rot/s^3$ )

On the other hand, few works have been carried out about geometrical smoothing. A corner optimization is proposed in [9–11] but it is applied only in 3-axis machining. In 5-axis, different methods were proposed to smooth the rotary drives of a 5-axis milling machine [12–14]. The fundamental idea is that the slowdowns on the feedrate come from the rotary drives, which appears to be too restrictive. The method proposed in [15] increases the smoothness by minimizing the energy of deformation of the tool path in the context of 5-axis flank milling. This allows a global optimization of the tool path but as it is realized in the Part Coordinate System, machine tool constraints are not taken into account. Industrial numerical controllers also provide solutions to smooth the geometry of the tool path, such as corner rounding functions or tool path compressors [8]. These functions lead to a shorter machining time but the user cannot control the geometrical error generated on the part. Indeed the tolerance is handled axis by axis, which means that the resulting errors on the part in 5-axis milling cannot be controlled.

The main problem of these approaches is that the machine tool characteristics are not considered. Actually, the results of the algorithms will be the same whatever the desired feedrate and the kinematical capacities of the machine tool. However, it is clear that depending on the relative abilities of each drive the solution should change (see section 3 of [16]).

The prediction of the velocity profile generated by

the CNC was used in [17, 18] to improve the machining time by changing the orientation of the tool. Although the complete motion planning is the best way to see where the feedrate is decreasing, it is time consuming and not necessary for the purpose of trajectory smoothing.

Another way to improve the machining time is to use a polynomial description of the tool path with a good parameterization as it is shown in [19–21]. This is used on top of the proposed algorithm to reduce the machining time even more. Actually the native B-Spline format is used for the machining tests presented further.

In this paper, the proposed approach consists in computing the necessary reductions of the feedrate due to velocity, acceleration and jerk constraints of each drive. This information, taken from the field of smooth motion planning, is then used to optimize the geometry and smooth the tool path. This will finally lead to a reduction of the machining time.

The optimization is an iterative process, as it is shown in Fig. 1. An evaluation of the kinematical constraints is first performed, then a local geometrical smoothing is carried out on the selected axis. It is important to note that the tool path is smoothed in the Machine Coordinate System. If needed, a N-buffer technique [22] is used to control the geometrical deviations on the part. In this case, a compromise has to be made between the smoothness and the geometrical tolerance (see subsection 4.2).

The rest of the paper is organized as follows: drive constraints are presented in section 2. These constraints give the axes and positions which have to be smoothed. Using this information the smoothing algorithm is explained in section 3. Two machining tests are carried out in section 4. Results demonstrate the efficiency of the proposed algorithm by measuring the real machining time. Finally the conclusions are summarized in section 5.

## 2. Drive constraints

The aim of this section is to predict the slowdowns of the feedrate. A complete prediction of the velocity along the path would have been time consuming so an approximation of the maximum velocity profile is computed. This approximate maximum limit is given by the velocity, acceleration and jerk constraints of the 5 drives.

First, the mathematical formalism is introduced and the kinematical constraints are exposed and finally the upper limit of the feedrate is presented.

### 2.1. Mathematical formalism

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 which allows the further geometrical optimization. One can note that this formula is valid for linear and rotary axes; it is thus possible to compare the 5 drives of the machine tool which is an important advantage. The acceleration  $\ddot{\mathbf{q}}$  and jerk  $\dddot{\mathbf{q}}$  of the drives are obtained identically in Eq. 2 and 3.

$$\dot{\mathbf{q}} = \frac{d\mathbf{q}}{dt} = \frac{d\mathbf{q}}{ds} \frac{ds}{dt} = \mathbf{q}_s \dot{s} \quad (1)$$

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

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

$\mathbf{q}_s$ ,  $\mathbf{q}_{ss}$ ,  $\mathbf{q}_{sss}$  are the geometrical derivatives with respect to displacement  $s$  along the tool path. They should be known as soon as the path is defined. However, the CNC has some options to round the sharp corners so the executed geometry is modified as well as the amplitude of these parameters. To overcome this problem two solutions are available: whether to have a model of the way the CNC is rounding the corners or to send a native

B-Spline tool path to the machine. With the second solution, the tool path has no sharp corner due to the G1 discontinuities so the geometry is not modified by the CNC.

Thanks to the Eq. 1-3 it is now possible to express the constraints of the drives.

### 2.2. Velocity, acceleration and jerk constraints

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 vibrations due to the dominating vibratory mode of the axes.

Eq. 4 presents the velocity constraints.

$$- \begin{bmatrix} v_{max}^x \\ v_{max}^y \\ v_{max}^z \\ v_{max}^a \\ v_{max}^c \end{bmatrix} \leq \begin{bmatrix} jX_s \\ jY_s \\ jZ_s \\ jA_s \\ jC_s \end{bmatrix} j\dot{s} \leq \begin{bmatrix} v_{max}^x \\ v_{max}^y \\ v_{max}^z \\ v_{max}^a \\ v_{max}^c \end{bmatrix} \quad (4)$$

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.

$$|j\dot{q}^i| \leq V_{max}^i ; |j\ddot{q}^i| \leq A_{max}^i ; |j\ddot{q}^i| \leq J_{max}^i \quad (5)$$

The aim is now to find the maximum value of the feedrate  $\dot{s}$  allowed by these constraints. To obtain an exact solution, the Eq. 5 should be solved recursively because of the link between  $\dot{s}$ ,  $\ddot{s}$  and  $\ddot{s}$ . However a good approximation can be given in a closed form as it is explained below.

### 2.3. Approximation of the maximal feedrate

The first inequation is easy to use as it gives immediately the highest feedrate allowed by the velocity of the axes.

$$j\dot{s} \leq \min_{i=1..5} \left( \frac{V_{max}^i}{|j\dot{q}_s^i|} \right) \quad (6)$$

But in the other inequations, there is the link  $\ddot{s} = \frac{d}{dt}(\dot{s})$ . As an approximate upper limit of the feedrate is only required, it is possible to use the limit when  $\ddot{s} = 0$ . This will be exact at sharp corners as the feedrate is decreasing and then increasing. Furthermore these are the most interesting areas for us. Thus a limit given by the accelerations of the axes is obtained in a close form:

$${}_j\dot{s} \leq \min_{i=1..5} \left( \sqrt[3]{\frac{A_{max}^i}{|{}_j q_{ss}^i|}} \right) \quad (7)$$

For the jerk, the same kind of problem has to be faced. Taking  $\ddot{s} = 0$  and  $\dddot{s} = 0$ , a limit of the feedrate given by the jerks is obtained.

$${}_j\dot{s} \leq \min_{i=1..5} \left( \sqrt[3]{\frac{J_{max}^i}{|{}_j q_{sss}^i|}} \right) \quad (8)$$

Of course, the feedrate is limited by the programmed feedrate  $F_{pr}$ . Discretizing the tool path, it is possible to obtain an approximate limit of the real feedrate for each position along the path with the Eq. 9. It is interesting to point out that this evaluation is really fast as there is no iteration.

$${}_j\dot{s} \leq \min_{i=1..5} \left( F_{pr}, \frac{V_{max}^i}{|{}_j q_s^i|}, \sqrt[3]{\frac{A_{max}^i}{|{}_j q_{ss}^i|}}, \sqrt[3]{\frac{J_{max}^i}{|{}_j q_{sss}^i|}} \right) \quad (9)$$

It is important to notice that this is an approximation of the maximum reachable feedrate. That means that the real feedrate can cross this limit while respecting all the constraints. With the approximations which is made, the link between two successive points is lost. But of course as the acceleration is limited, it will not be possible to follow exactly the proposed limit because some time is required to accelerate and to decelerate along the tool path. Practically, this limit gives a really good indication about the real feedrate reached by the machine tool as it will be shown in the examples below.

Finally, taking into account all the constraints of the drives, the areas where the real feedrate will decrease can be predicted. Moreover, the cause of the slowdowns is known so it is possible to smooth the corresponding axes in these areas.

### 3. Smoothing algorithm

The aim of the smoothing algorithm is to raise the upper limit of the feedrate by reducing the magnitude of the axes geometrical derivatives  $\mathbf{q}_s$ ,  $\mathbf{q}_{ss}$ ,  $\mathbf{q}_{sss}$ . For each tool position along the path, it is possible to know which axis is limiting the feedrate with Eq. 9. The tool path smoothing is done iteratively thanks to a local joint movement smoothing. As it is said before, the axis motion is smoothed locally around the discontinuity so it is important to make sure that the junctions between the initial axis movement and the smoothed zone will be at least  $C^2$  in order to avoid the slowdowns due to the junction discontinuities. Moreover, the geometry should be

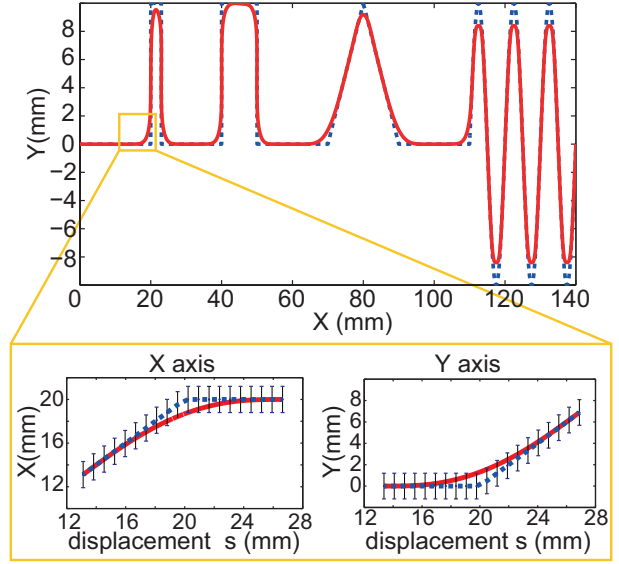


Figure 2: 2D example tool path.

controlled so the smoothed spline should respect a given tolerance. And finally, the spline has to be smooth that is to say that the second derivative should be minimal.

#### 3.1. Explanations on a simple example

To illustrate this smoothing algorithm, a simple 2D example will be first used even if the algorithm is not designed to be applied in such an easy case. The tolerance used is really large for the purpose of this explanation (Fig. 2). For each sharp corner, the second geometrical derivative is infinite  $\mathbf{q}_{ss} \rightarrow \infty$  so according to Eq. (2) the feedrate has to be zero to respect the inequations (5). That means that each axis instruction has to be modified as it is shown in the bottom of Fig. 2. The piecewise polynomial spline on the bottom of Fig. 2 minimizes its second derivative while respecting all the constraints mentioned above. The method is implemented in Matlab by using a function developed by D'Errico [23].

As one can see on Fig. 2, the smoothing algorithm is applied locally around each axis discontinuity. The combination of the smoothing of the X and Y axes leads to a smooth tool path in the Part Coordinate System. The demonstration of how a tolerance on the part can be handled and how the smoothing parameters are chosen is carried out further.

#### 3.2. Selection of the smoothing parameters

Three parameters have to be defined to apply the smoothing algorithm: the tolerance on the axis move-

ment, the axis which has to be smoothed and the zone where it has to be smoothed.

The maximum reachable feedrate obtained with the Eq. 9 is used to determine which axis will be smoothed for the current iteration. The initial idea is to take the position of the minimum reachable feedrate and to smooth the tool path around this area. But sometimes, after few iterations, it will be impossible to have a smoother tool path in some areas because the tool path will be as smooth as possible according to the allowed part tolerance. So to choose the area which will be smoothed, some information given by the N-buffer about the tolerance left are needed too. Of course, the smoothing algorithm can be applied several times in the same area.

In end milling with a ball end mill, it is possible to smooth the tool path without introducing any geometrical deviation on the part as it will be shown further. But in flank milling, the smoothing algorithm will generate some geometrical deviations which have to respect the given part tolerance. Due to the non-linear kinematic transformation between the Machine Coordinate System and the Part Coordinate System, the effect of the smoothing tolerance on the part tolerance is hardly predictable. So this axis tolerance is first chosen with a heuristic guess and then a N-buffer technique is used for each iteration in order to ensure that the CAD tolerance is respected. Thus in flank milling the choice of the axis tolerance for the smoothing algorithm is handled thanks to a second optimization loop (see Fig. 1).

### 3.3. Detailed algorithm

The following simplified algorithm explains the proposed method to smooth a given tool path. For each iteration, the upper feedrate limit is computed in order to select the smoothing parameters as explained above.

In flank milling, the CAD tolerance has to be respected. So for an iteration of the main optimization loop, the axis tolerance is gradually reduced if needed. If the smoothed axis is a rotary axis, the tool center position is kept constant because a small rotation can lead to an important error on the part due to the lever arm. The repositioning of the tool center is realized thanks to the movement of the 4 other axes of the machine tool as explained in the following subsection.

In end milling with a ball end cutter, the repositioning of the tool center allows to smooth the tool path by changing only the orientation of the tool and not the position. So the tool path is smoothed without introducing any geometrical error.

```

While improvement is possible
  Find the lowest point of the upper limit of the feedrate
  Select the smoothing parameters
  If flank milling
    While part is out of tolerance
      Smooth the selected axis
      If A or C is smoothed
        Reposition the tool center
      End
      N-buffer
      Reduce the tolerance
    End
  Else if end milling
    Smooth the selected axis
    Reposition the tool center
  End
End

```

### 3.4. Tool center repositioning

In 5-axis ball end milling, it is possible to smooth the tool path without generating any geometrical error by replacing the tool center to its original position. Indeed, once one axis is smoothed, you can use the other axes to keep the center of the tool at the same position. The result is that the orientation of the tool is modified but not its position. One should notice that here the orientation of the tool is considered to be completely free, neglecting the problem of cutting speed.

The inverse kinematic transformation of the machine is given in Eq. 10 where:

- $[X_{pr} Y_{pr} Z_{pr}]$  defines the programmed coordinates of the tool center,
- $[X Y Z A C]$  defines the 5-axis coordinates of the machine tool,
- $[P_x P_y P_z]$  defines the work offset,
- $b_y b_z$  are the distances between the rotary axes of the Mikron UCP710 machine tool,
- $[m_x m_y m_z]$  characterized the machine zero,
- $j_z$  is the tool length offset.

The machine tool has 5 degrees of freedom and only 3 are required to reposition the tool center. For each iteration of the smoothing algorithm, one axis is smoothed so there is 4 axes left to keep the tool center position constant. As the kinematical characteristics of the rotary axes are lower than for the linear axes, the choice to preserve them from rapid compensation movements is made when it is possible. So if a rotary axis is smoothed, the other rotary axis is set to be constant and  $[X Y Z]$  are computed thanks to the Eq. 10 with the original  $[X_{pr} Y_{pr} Z_{pr}]$ .

$$\begin{aligned}
X &= \cos(C)(X_{pr} + P_x) + \sin(C)(Y_{pr} + P_y) + m_x \\
Y &= \cos(A)[\sin(C)(X_{pr} + P_x) + \cos(C)(Y_{pr} + P_y) + b_y] + \sin(A)[Z_{pr} + P_z + b_z] + m_y \\
Z &= \sin(A)[\sin(C)(X_{pr} + P_x) + \cos(C)(Y_{pr} + P_y) + b_y] + \cos(A)[Z_{pr} + P_z + b_z] + m_z
\end{aligned} \tag{10}$$

Table 1: Machine tool drive limits

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

If a linear axis is smoothed, the A-axis is set to be constant and the problem leads to compute C-axis with an equation of the form:

$$cst_1 \sin(C) + cst_2 \cos(C) = cst_3 \tag{11}$$

Thus the position of the other axes can be computed to keep the tool center position constant.

This algorithm will now be applied in end and in flank milling to demonstrate the efficiency of this 5-axis tool path smoothing algorithm based on drive constraints.

#### 4. Applications

The proposed smoothing algorithm is applied on two different industrial parts. Only finishing operations are considered here but the algorithm could easily be applied to a roughing operation. The experiments are carried out on a 5-axis MIKRON UCP 710 machining center whose kinematical characteristics are given in Table 1. Air cutting tests are conducted to compare machining time and effective feedrates. 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 programmed feedrate is set to  $F_{pr} = 5000 \text{ mm/min}$  for both examples.

##### 4.1. 5-axis end milling of an airfoil like surface

The first example is dealing with the 5-axis end milling of an airfoil (Fig. 3). The initial tool path is created using a multi-axis helix machining operation of the Advanced Machining mode of CATIA V5. The parameters used are: a scallop height of 0.01 mm, a fixed leading and tilt angle of 0 and 5 degree respectively and

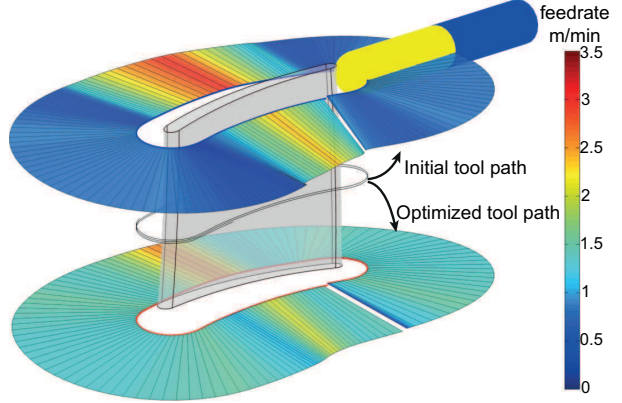


Figure 3: Airfoil tool path coloured according to the real feedrate.

a  $\phi 10 \text{ mm}$  ball end mill. As a ball end cutter is used, it is possible to change the orientation of the tool without creating any geometrical deviation on the final part. So the algorithm described in Fig. 1 is used, and once the local geometrical smoothing is applied to an axis, the tool center position is reset to its original location thanks to the other axes movements. Thus the tool path is iteratively optimized in order to raise the upper limit of the feedrate computed in section 2.

For the experiments, a portion (Fig. 3) of the tool path is extracted and sent in a native B-Spline format to the CNC controller. The feedrate limits given by the velocity, acceleration and jerk of each axis are shown on Fig. 4. The characteristics of the initial tool path are given on the top row. As it can be seen from this example, the main limitation of the feedrate is due to the velocity of the C-axis. Actually, the machining strategy leads to an important use of the C-axis since the tool has to revolve around the part. On Fig. 4.a it can be observed that the measured feedrate is actually limited by the C-axis velocity. So the optimization algorithm is changing the C-axis motion in order to raise this limitation. The results of the optimization are shown on the bottom row of Fig. 4. The major difference is that the limit given by the velocity of the C-axis is increased and as the tool path is smoother, the acceleration and jerk limits are raised as well. Once again, one can see on Fig. 4.d that the measured feedrate match really well the upper limit. The Fig. 3 shows the measured feedrate along a revolution around the airfoil. On top of the figure,

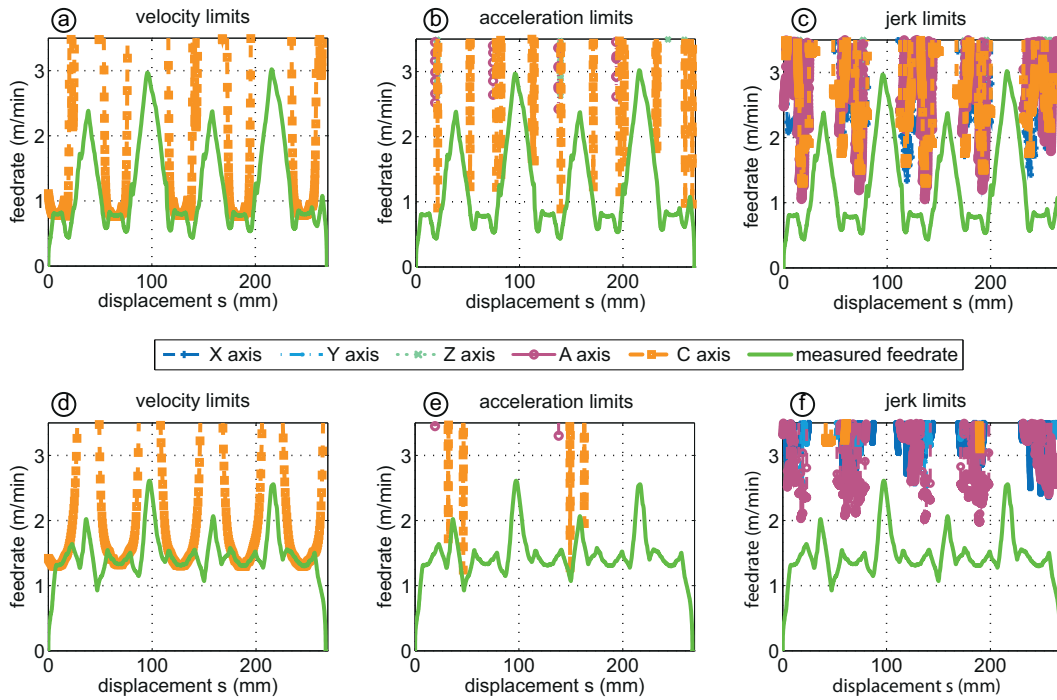


Figure 4: Feedrate limitations given by the velocity, acceleration and jerk for the airfoil. Top row: Initial B-Spline tool path, bottom row: Optimized B-Spline tool path.

the feedrate decreases a lot on the leading and trailing edge of the airfoil. Whereas for the optimized tool path (on the bottom), the measured feedrate is a lot higher in these locations due to a change of the tool orientation. This is clearly explained in Fig. 5. Close to the leading and trailing edge, the tool axis of the optimized path (in solid red line) has an important leading angle in order to anticipate the movement.

Finally, the measured machining time for the initial and optimized tool path is equal to 17.2 s and 11.9 s respectively. For this application, a reduction of 30 percent of the machining time is obtained.

These performances are achieved because in this example, the C-axis of the machine tool is used a lot in the initial tool path whereas its kinematical capacities are quite low. For every 5-axis point milling tool path, it will be possible to smooth the axis which limits the feedrate and to compensate the movement thanks to the other axes which have better kinematical capacities as it is shown in Section 3.4. Thus the proposed method reduces the machining time without creating any geometrical deviation but an even faster solution could have been generated if the tolerance specified on the part had been used.

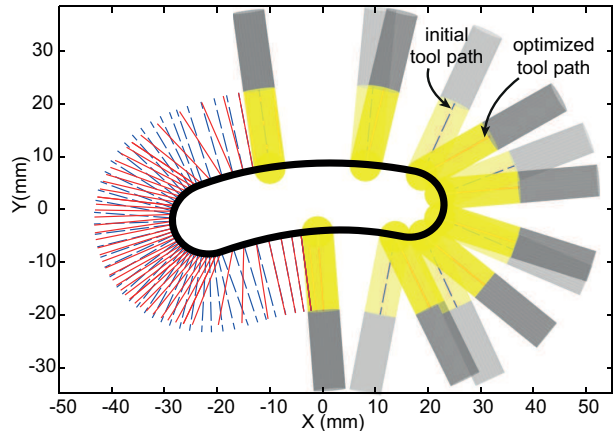


Figure 5: Airfoil initial (dash blue lines) and optimized (solid red lines) tool paths.



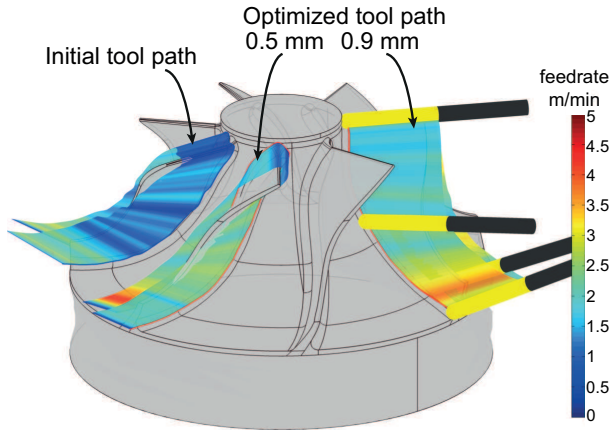


Figure 6: Impeller tool path coloured according to the real feedrate.

#### 4.2. 5-axis flank milling of an impeller

The second example is taken from Pechard et al. [15]. The blades are machined with a unique trajectory for both sides (Fig. 6). The initial tool path is obtained minimizing the geometrical deviations as explained in [15]. Indeed as the surface of the blade is a non-developable ruled surface, the overcut and undercut cannot be avoided. As it is a flank milling operation, the orientation of the tool has to be controlled accurately. A N-buffer technique is introduced in the optimization loop to control the geometrical errors on the part. Indeed during the local smoothing it is impossible to have a relation between the tolerance on each axis and the resulting effect on the part. To overcome this problem, the direct kinematical transformation is realized and the geometry is checked for each iteration.

Table 2 presents the results. The initial tool path has a maximum deviation of 0.4 mm and the aim was to smooth the tool path as much as possible with an allowed tolerance of 0.5 mm and 0.9 mm. For each tool path, two tests have been carried out with G1 and native B-Spline description of the tool path. First of all, one can see that the tolerance specified is respected. Then the machining time is drastically reduced whatever the format of the tool path. For the tolerance of 0.9 mm, the geometrical deviation is increased a lot whereas the reduction of the machining time is small. Of course, using a native B-Spline the machining time is reduced because the tool path is smoother than with G1 discontinuities.

Fig. 7 shows the limits given by the jerk on each axis. For this example the velocity and acceleration limits are not plotted because the jerk is the limiting parameter all along the path. The graphics correspond respectively to the initial (left) and optimized with a tolerance of 0.5 mm (right) tool paths. The results shown here are ob-

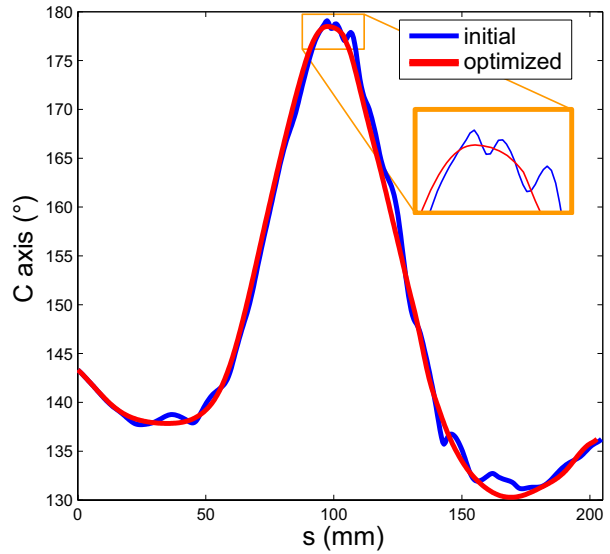


Figure 8: C-axis initial and optimized (tol=0.5 mm) instructions.

tained using the native B-Spline format in order to avoid the discontinuity problems. It can be noticed that the correlation between the approximate upper limit and the measured feedrate is striking. Moreover, the X and Y axes limits are not far from the A-axis limit so it shows that the linear axes have to be taken into account too to smooth 5-axis tool paths.

The optimization smoothed the tool path and the computed feedrate limits are increased. One can notice that the programmed feedrate is never reached. On the right hand plot, the feedrate between 0 and 100 mm does not reach the upper limit. Actually, it could have been higher while respecting all the constraints. So with an optimized velocity planning algorithm, even more time could have been saved.

Fig. 8 presents the result obtained on the C-axis. It is clear that the oscillations are filtered; so the geometrical derivatives  $C_s, C_{ss}, C_{sss}$  are decreased and the tool path is smoother.

This example shows that the developed algorithm allows to smooth a 5-axis flank milling tool path while respecting a given tolerance on the part.

Here again, the measured machining time is reduced by more than 30 percent depending on the format and the given tolerance. This performance is achieved because the overcut and undercut are not evenly distributed along the path in the initial trajectory generation algorithm. Thus the proposed algorithm takes advantage of the whole allowed tolerance to distort the tool path in order to smooth it.

Table 2: Results of the optimization for the impeller.

Impeller	Undercut (mm)	Overcut (mm)	Measured machining time (s)
Initial tool path	0.13	-0.39	G1:23.5 BS:9.3
Optimized tool path (tol=0.5mm)	0.40	-0.49	G1:13.7 BS:6.4
Optimized tool path (tol=0.9mm)	0.89	-0.66	G1:11.3 BS:6.0

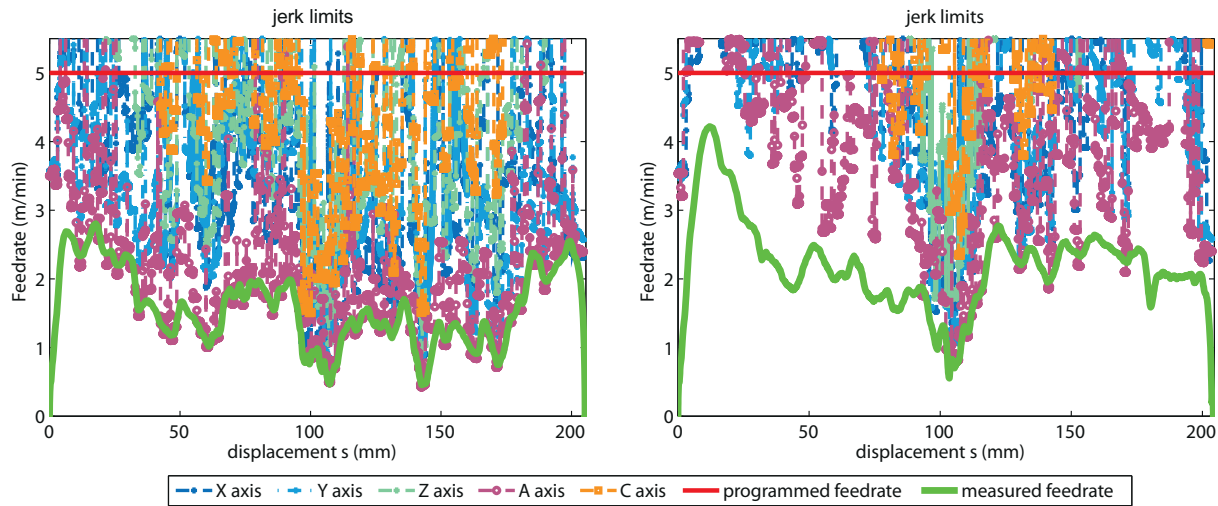


Figure 7: Feedrate limitations given by the jerk for the impeller. Left: Initial B-Spline tool path, right: Optimized B-Spline tool path (tol=0.5 mm).

## 5. Conclusion

In this paper, a new approach is proposed to address the problem of tool path smoothing. The main difference with the other approaches of the literature is that the velocity, acceleration and jerk of each axis of the machine tool are considered. Indeed, the smoothing algorithm is applied to the movement of the joints in the Machine Coordinate System after an evaluation of the maximum reachable feedrate. This straightforward evaluation of the maximum feedrate allows to localize the critical areas where the tool path has to be smoothed. A local smoothing of the joint geometrical evolution is iteratively applied in order to improve the smoothness at these critical points. The illustrated experimental results show a significant reduction of the machining time. It has been shown that despite the fact that the smoothing algorithm is carried out in the Machine Coordinate System, it is possible to handle a geometrical tolerance on the part with a N-buffer simulation. In 5-axis flank milling, the optimization takes advantage of the tolerance all along the path to improve the feedrate and re-

duce machining time. In 5-axis point milling with a ball end mill, it is possible to save machining time without creating any geometrical deviation.

## References

- [1] K. Shin, N. McKay, Minimum time control of robotic manipulators with geometric path constraints, *IEEE Transactions on Automatic Control* 30 (6) (1985) 531-541.
- [2] 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.
- [3] 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.
- [4] 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.
- [5] 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, *The International Journal of Advanced Manufacturing Technology* 37 (1-2) (2008) 104-121.
- [6] M. Heng, K. Erkorkmaz, Design of a nurbs interpolator with minimal feed fluctuation and continuous feed modulation capa-

- bility, *International Journal of Machine Tools and Manufacture* 50 (3) (2010) 281-293.
- [7] 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.
- [8] Siemens, Sinumerik - 5-axis machining, 2009.
- [9] R. Paul, *Robot manipulators: mathematics, programming, and control: the computer control of robot manipulators*, The MIT Press, 1981.
- [10] V. Pateloup, E. Duc, P. Ray, Corner optimization for pocket machining, *International Journal of Machine Tools and Manufacture* 44 (12-13) (2004) 1343-1353.
- [11] X. Pessoles, Y. Landon, W. Rubio, Kinematic modeling of a 3-axis NC machine tool in linear and circular interpolation, *The International Journal of Advanced Manufacturing Technology* 47 (5-8) (2010) 639-655.
- [12] C.-S. Jun, K. Cha, Y.-S. Lee, Optimizing tool orientations for 5-axis machining by configuration-space search method, *Computer-Aided Design* 35 (6) (2003) 549-566.
- [13] M.-C. Ho, Y.-R. Hwang, C.-H. Hu, Five-axis tool orientation smoothing using quaternion interpolation algorithm, *International Journal of Machine Tools and Manufacture* 43 (12) (2003) 1259-1267.
- [14] C. Castagnetti, E. Duc, P. Ray, The domain of admissible orientation concept: A new method for five-axis tool path optimisation, *Computer-Aided Design* 40 (9) (2008) 938-950.
- [15] P.-Y. Pechard, C. Tournier, C. Lartigue, J.-P. Lugarini, Geometrical deviations versus smoothness in 5-axis high-speed flank milling, *International Journal of Machine Tools and Manufacture* 49 (6) (2009) 454-461.
- [16] Y. Altintas, C. Brecher, M. Weck, S. Witt, Virtual machine tool, *CIRP Annals - Manufacturing Technology* 54 (2) (2005) 115-138.
- [17] S. Lavernhe, C. Tournier, C. Lartigue, Optimization of 5-axis high speed machining using a surface based approach, *Computer-Aided Design*, 4(10-11) (2008) 1015-1023.
- [18] 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.
- [19] R. V. Fleisig, A. D. Spence, A constant feed and reduced angular acceleration interpolation algorithm for multi-axis machining, *Computer-Aided Design* 33 (1) (2001) 1-15.
- [20] C. Lartigue, C. Tournier, M. Ritou, D. Dumur, High-performance NC for hsm by means of polynomial trajectories, *CIRP Annals - Manufacturing Technology* 53 (1) (2004) 317-320.
- [21] J. M. Langeron, E. Duc, C. Lartigue, P. Bourdet, A new format for 5-axis tool path computation using bspline curves, *Computer-Aided Design* 36 (12) (2004) 1219-1229.
- [22] R. B. Jerard, R. L. Drysdale, K. Hauck, B. Schaudt, J. Magewick, Methods for detecting errors in numerically controlled machining of sculptured surfaces, *IEEE Computer Graphics and Applications* 9 (1) (1989) 26-39.
- [23] J. D'Errico, Slm shape language modeling, matlab central file exchange (2005).