Int J Adv Manuf Technol
DOI 10.1007/s00170-006-0699-8

ORIGINAL ARTICLE

Optimal sequencing of rotation angles for five-axis machining

S. S. Makhanov · M. Munlin

Received: 14 December 2005 / Accepted: 13 June 2006

Abstract In this paper, we propose two new algorithms to correct the trajectories of the tool tip of a five-axis milling machine by adjusting the rotation angles in such a way that the kinematics error is reduced. The first algorithm is based on the shortest-path optimization with regards to feasible rotations of the inverse kinematics. The cost function is represented in terms of the total angle variation. We show that such an optimization increases the accuracy of machining and is the most appropriate in the case of a rough cut. The shortest-path procedure applies to either the entire set of trajectories or to only the most inappropriate undercuts inside the workpiece. In the latter case, the algorithm generates an interesting family of solutions characterized by smaller undercuts obtained at the expense of increased overcuts. The second algorithm also exploits the idea of the minimization of the angle variation. It is based on the uniform distribution of the cutter contact points with regards to the rotation angles. The method inserts additional tool positions by numerically finding a grid of points distributed uniformly in the angular space. We prove that the proposed algorithm in the neighborhood of stationary points requires 3–4 times fewer additional points than the conventional scheme. Also, if a maximum angular speed has been exceeded, the controller detects this event and reduces the angular speed. Our correction algorithms minimize the total angle variation, thus, reducing the probability of such an event. Finally, the efficiency of the two algorithms has been verified by a five-axis machine MAHO600E at the CIM Lab of the Asian Institute of Technology of Thailand and HERMLE UWF920H at the CIM Lab of the Kasetsart University of Thailand.

Keywords Five-axis machining · Tool path optimization

1 Introduction

The five-axis machine is guided by axial commands $\varPi \in \mathbb{R}^5$ carrying three spatial coordinates of the tool tip in the machine coordinate system and two rotation angles. The tool path $\mathfrak{A} = \{\varPi_0,\ \varPi_1, \ldots,\ \varPi_m\}$ is a sequence of positions Π_p arranged into a structured spatial pattern, such as a zig-zag or a spiral pattern. The path could also be composed of a variety of unconventional patterns and can include tool retractions. A full optimization scheme involves a model of cutting operations, topologies of the prescribed tool path patterns, and an optimization procedure. Let $\mathfrak{M}$ be a set of parameters related to the configuration of the machine and $\mathfrak{T}$ be a set of parameters related to the tool. Let $S(u,\ v) = (x_S(u,\ v),\ y_S(u,\ v),\ z_S(u,\ v))$ be the required surface. The model of the cutting operations, being fed with $\mathfrak{M}$, $\mathfrak{T}$, S, and $\mathfrak{A}$, produces the result of machining, the output surface T.

The general optimization problem is then formulated by:

$$minimize(C), \quad \varPi,\ \mathfrak{M},\ \mathfrak{T}$$

where C denotes a criteria vector which may include the error $\varepsilon \equiv \|S - T\|$, the length of the path, the negative of the machining strip (strip maximization), the machining time, etc. (for instance, see [3, 6, 9, 10, 17]). The optimization

S. S. Makhanov (✉)
School of Information and Computer Technology, Sirindhorn International Institute of Technology, Thammasat University,
Bangkadi Campus, 131 M. 5 Tiwanon Road, Bangkadi,
Muang Pathum Thani 12000, Thailand
e-mail: makhanov@siit.tu.ac.th

M. Munlin
Tapee College,
8/151 Srivichai Road, Muang District,
Suratthani 84000, Thailand

could be subjected to constraints [8, 18]. The most important constraints are: (1) the scallop height constraint, the scallops between the successive tool tracks must not exceed a prescribed tolerance; (2) the local accessibility constraints, the constraint ensures against the removal of excess material when the tool comes in contact with the desired surface, due to the so-called curvature interference; (3) the global accessibility constraints (see, for instance, [17]).

Given the general context above, the error analysis and optimization with regards to variations of the rotation angles has not been provided by commercial CAD/CAM systems, such as Unigraphics, EdgeCam, Vericut, etc. Only a few research papers deal with the subject. Jung et al. [7] analyze the sequence of rotations to minimize the number of the phase-reverse steps at discontinuities of the first derivative of the surface (corners, etc.). A method of avoiding singularities has been presented by Affouard et al. [2]. However, optimal angle sequencing was not considered. An algorithm based on global optimization with regards to feasible rotations of the machine was proposed by Munlin et al. [15]. However, this paper employs the direct minimization of the kinematics error, which is a computationally expensive procedure. Besides, their paper does not differentiate between the overcut and undercut errors, which is an important issue for rough machining. Finally, it is not clear how to insert additional points if the resulting accuracy is still not sufficient.

In this paper, we deal with the above-mentioned drawbacks. We propose a new modification of the techniques based on minimization with regards to the total angle variation, which does not invoke direct evaluation of the trajectories. We show that such a minimization reduces the kinematics error. Additionally, we introduce a new minimization which applies only to the most inappropriate undercuts. Although this procedure could be computationally expensive as well, it generates an interesting new family of solutions characterized by smaller undercuts obtained at the expense of an increase in overcuts. The second new algorithm also exploits minimizing the total angle variation. It solves a problem of inserting additional points in the areas of large kinematics errors. The additional positions are created by numerically finding a grid of points uniformly distributed in the angular space. We prove experimentally that the proposed method requires 3–4 times fewer additional points than conventional schemes performed near the stationary points.

The efficiency of the two algorithms has been verified by a five-axis machine MAHO600E at the CIM Lab of the Asian Institute of Technology of Thailand and HERMLE UWF920H at the CIM Lab of the Kasetsart University of Thailand. It has been also verified by a virtual milling machine simulator [16] developed by the authors.

Finally, let us introduce an additional benefit. There is always a limit of the angular speed of specific machine parts. As a result, a shorter tool path with many turns may require more time than a longer tool path with fewer turns [1]. For example, in the case of MAHO600E, the maximum angular velocities of the primary and secondary rotational axis are $v_{A,\, max} = 235^\circ/s$ and $v_{B,\, max} = 235^\circ/s$. If the maximum angular speed is exceeded, the controller detects this event and reduces the angular speed, increasing the machining time. Our correction algorithms minimize the total angle variation, thus, reducing the probability of such an event.

2 Kinematics of the five-axis milling machine

Five-axis machines are characterized by nonlinear kinematics due to the two additional degrees of freedom that control tool orientation. In this section, a closed-form representation of the trajectory of the tool tip is derived for general machine kinematics. The kinematics $\mathfrak{K} \equiv \mathfrak{K}(W,\ \Re,\ M)$ of a machine is a transformation from a point M in machine coordinates to a point W in workpiece coordinates. Recall that $\mathfrak{M}$ is a set of optimization parameters and $\Re = (a,\ b)$ is a pair of rotation angles. For simplicity, the transformation will be denoted by $\mathfrak{K}(\mathfrak{M})$when possible. Let $\mathfrak{K}^{-1}(W)$ be the inverse transformation such that $\forall W,\ M,\ \Re,\ \mathfrak{K}^{-1}(\mathfrak{K}(M))\ = M$, and $\mathfrak{K}^{-1}(\mathfrak{K}(W)) = W$. Let $\Pi_p \equiv \left(M_p,\ \Re_p\right),\ \Pi_{p+1} \equiv \left(M_{p+1},\ \Re_{p+1}\right)$ be two successive coordinates of the tool path in $\mathbb{R}^5$. M_p and W_p denote spatial positions of the tool tip in the machine and workpiece coordinate systems, respectively, and $\Re_p$ is the corresponding pair of rotation angles. Now, a closed-form representation of the trajectory of the tool tip between W_p and W_{p+1} will be deduced.

Let us invoke the inverse kinematics to transform part–surface coordinates W_p into machine coordinates $M_p{\equiv}(x_p,\ y_p,\ z_p)$ as follows: $M_p \equiv \mathfrak{K}^{-1}\left(\mathfrak{M},\ \Re_p,\ W_p\right)$. The rotation angles $\Re \equiv \Re(t) = (a(t),\ b(t))$ and the machine coordinates of the tool tip $M{\equiv}M(t)$ are assumed to be changing linearly between the prescribed positions, namely:

$$M(t) = tM_{p+1} + (1-t)M_p,\ \Re(t) = t\Re_{p+1} + (1-t)\Re_p$$

where t is a fictitious time coordinate ($0{\le}t{\le}1$). Transforming machine coordinates M back to workpiece coordinates W for every t yields:

$$\begin{aligned} W_{p,\,p+1}(t) &= \mathfrak{K}(\mathfrak{M},\ \Re(t), M(t)) \\ &= \mathfrak{K}\left(\mathfrak{M},\ t\Re_{p+1} + (1-t)\Re_p, tM_{p+1} + (1-t)M_p\right) \end{aligned} \tag{1}$$

Now M_p and M_{p+1} are eliminated by using the inverse transformation $M_p = \mathfrak{K}^{-1}\left(\mathfrak{M},\ \Re_p,\ W_p\right)$, so that the resulting trajectory depends only on the workpiece coordinates and the orientation of the tool. Substituting M_p and M_{p+1} into Eq. 1 yields:

$$\begin{aligned} W_{p,p+1}(t) = \mathfrak{K}(\mathfrak{M},\ & t\mathfrak{R}_{p+1} + (1-t)\mathfrak{R}_p, \\ & t\mathfrak{K}^{-1}\left(\mathfrak{M},\ \mathfrak{R}_{p+1},\ W_{p+1}\right) \\ & + (1-t)\mathfrak{K}^{-1}\left(\mathfrak{M},\ \mathfrak{R}_p,\ W_p\right)) \end{aligned} \tag{2}$$

Equation 2, applicable to an arbitrary machine configuration, allows the evaluation of the kinematics error, defined as the difference between the desired and the actual trajectory. In particular, we consider the kinematics of the five-axis milling machines MAHO600E and HERMLE UWF902H. The inverse kinematics are represented by matrices $A \equiv A[a(t)]$ and $B \equiv B[b(t)]$ associated with the rotations around the primary and the secondary axes, respectively (see Fig. 1), namely:

$$\mathfrak{K}^{-1}\{\mathfrak{R}\}[W] = GB[b](A[a](W + T_{12}) + T_{23}) + T_{34} - T_4$$

where $G = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$ and T_{12}, T_{23}, and T_{34} are respectively the coordinates of the origin of the workpiece in the rotary table coordinates, coordinates of the origin of the rotary table coordinates in the tilt table coordinates, and the origin of the tilt table coordinates in the cutter center coordinates, $T_4 = \left(0\,,0, -T_L\right)$, where T_{L} is the tool length.

3 Kinematics error along the trajectory

In this section, we will introduce the kinematics error and the total angle variation. We will also introduce the overcut and undercut errors.

Let $W^D_{p,p+1}(t) \in S(u, v)$ be a curve between W_p and W_{p+1} extracted from the surface in such a way that it represents the desired tool path between Π_p and Π_{p+1}. We define the error as the deviation between $W^D_{p,p+1} \equiv \left(x^D_{p,p+1},\ y^D_{p,p+1},\ z^D_{p,p+1}\right)$ and $W_{p,p+1} \equiv \left(x_{p,p+1},\ y_{p,p+1},\ z_{p,p+1}\right)$ given by:

$$\begin{aligned} \varepsilon &= \sum_p \left[\int_0^1 \left(W^D_{p,p+1} - W_{p,p+1}\right)^2 dt\right]^{1/2} \\ &= \sum_p \left[\int_0^1 \left(x^D_{p,p+1} - x_{p,p+1}\right)^2 + (y^D_{p,p+1} - y_{p,p+1})^2 \right. \\ & \quad \left. + \left(z^D_{p,p+1} - z_{p,p+1}\right)^2 dt\right]^{1/2} \end{aligned} \tag{3}$$

Total error (Eq. 3) is approximated as follows:

$$\hat{\varepsilon} = \sum_p \left[\begin{array}{l} \dfrac{1}{L_p} \sum\limits_{l_p=1}^{L_p} \left(x^D_{p,p+1,l_p} - x_{p,p+1,l_p}\right)^2 \\ \quad + \left(y^D_{p,p+1,l_p} - y_{p,p+1,l_p}\right)^2 \\ \qquad + \left(z^D_{p,p+1,l_p} - z_{p,p+1,l_p}\right)^2 \end{array}\right]^{1/2} \tag{4}$$

where L_p is the number of sampling points between W_p and W_{p+1}.

Remark The definitions above can be simplified by replacing the desired trajectories $W^D_{p,p+1}$ with linear trajectories given by $W^L_{p,p+1} = tW_{p+1} + (1-t)W_p$, although care should be taken when using this option. As opposed to the machine coordinates *M*, trajectories in the workpiece coordinates are *not* linear. However, we may use the linear trajectories as a reference, noting that $\hat{\varepsilon} \equiv \|W^D - W\| \leq \|W^D - W^L\| + \|W^L - W\|$, where *L* is a piecewise linear approximation of *S*. Hence, when the points are close enough, the error is approximately $\hat{\varepsilon} \approx \varepsilon_L = \|W^L - W\|$. Note that, if the orientation of the tool is fixed through the entire cut, then ε_L=0. In other words, the three-axis mode leads to ε_L=0, since all of the trajectories become linear. Therefore, minimization with regards to ε_L must be subjected to constraints specifying the orientations of the

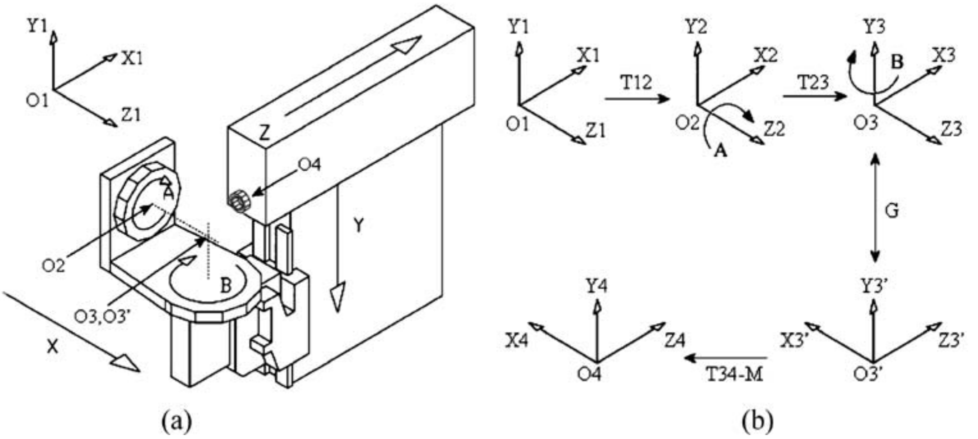

Fig. 1 **a** A TRTRT machine. **b** The corresponding reference coordinate systems

tool. Finally, the linearization is the simplest option which can be used even when the actual surface is not known, for instance, for a given G-code. However, the desired trajectory W_D can be extracted from the surface by a variety of ways, for example, using interpolation in the parametric space, geodesic curves, etc. The results presented in this paper are always valid, irrespective of the method to obtain W_D. Finally, the error induced by linear approximation depends on the maximum distance h between the CC points as $k\left\|W_D''\right\|_C h^2$, where k is a constant (see, for instance, [14]).

Let us now introduce the total angle variation. Consider two positions W_p and W_{p+1}. The corresponding kinematics error is then defined by $\varepsilon_{p,\,p+1} \equiv \left[\int\limits_0^1 \left(W^D_{p,\,p+1} - W_{p,\,p+1}\right)^2 dt\right]^{1/2}$.

Clearly, $\varepsilon_{p,\,p+1} \equiv \varepsilon\left(\Delta a_{p,\,p+1},\ \Delta b_{p,\,p+1},\ \Delta l_{p,\,p+1}\right)$, where $\Delta l_{p,\ p+1}$ is the distance between W_p and W_{p+1} (the spatial step in the workpiece coordinate system).

Furthermore, $\Delta a_{p,\,p+1} \equiv \Delta a_{p,\,p+1}\left(\Delta l_{p,\,p+1}\right) = a_{p+1} - a_p$ and $\Delta b_{p,\,p+1} \equiv \Delta b_{p,\,p+1}\left(\Delta l_{p,\,p+1}\right) = b_{p+1} - b_p$ are the angular steps. Practical experiments show that the angular steps are often more important than the spatial step. In particular, during a rough cut, decreasing the angular steps leads to a larger decrease in the error than a decrease in the spatial step.

The total angle variation between positions W_p and W_{p+1} is given by:

$$c_{p,\,p+1} = \int\limits_0^1 \sqrt{\left(\frac{\partial a_{p,\,p+1}}{\partial t}\right)^2 + \left(\frac{\partial b_{p,\,p+1}}{\partial t}\right)^2}\,dt$$

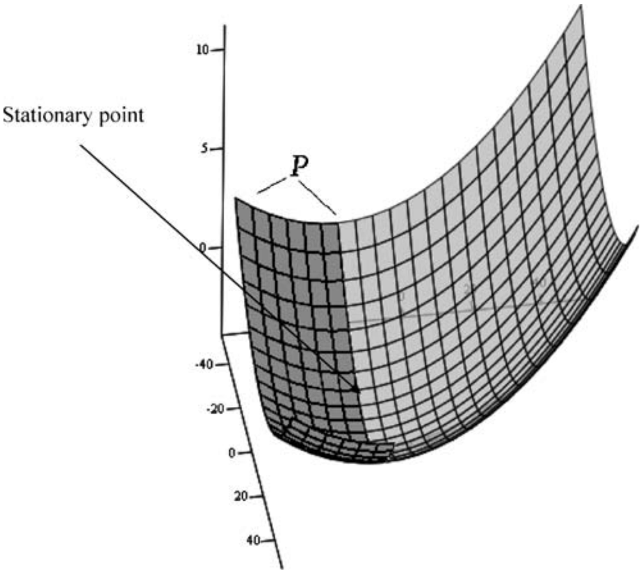

Fig. 2 An experimental part surface S_1. P is the machined area

Since $a_{p,\ p+1}$ and $b_{p,\ p+1}$ are changing linearly, $c_{p,\,p+1} = \sqrt{\Delta a^2_{p,\,p+1} + \Delta b^2_{p,\,p+1}}$. The total angle variation across the entire tool path is then given by:

$$c \equiv \sum_p c_{p,\,p+1} = \sum_p \sqrt{\Delta a^2_{p,\,p+1} + \Delta b^2_{p,\,p+1}}$$

Note that there often exists a neighborhood of the CL point where one of the angles changes faster than another angle. In this case, $\varepsilon_{p,\,p+1} \approx O\left(\Delta a^{m_1}_{p,\,p+1}\right)$ or $\varepsilon_{p,\,p+1} \approx O\left(\Delta b^{m_2}_{p,\,p+1}\right)$, where m_1 and m_2 depend on the surface. In this case, minimization of the total angle variation produces excellent results.

Finally, the case of a rough cut often requires us to differentiate between the undercut and the overcut error. By

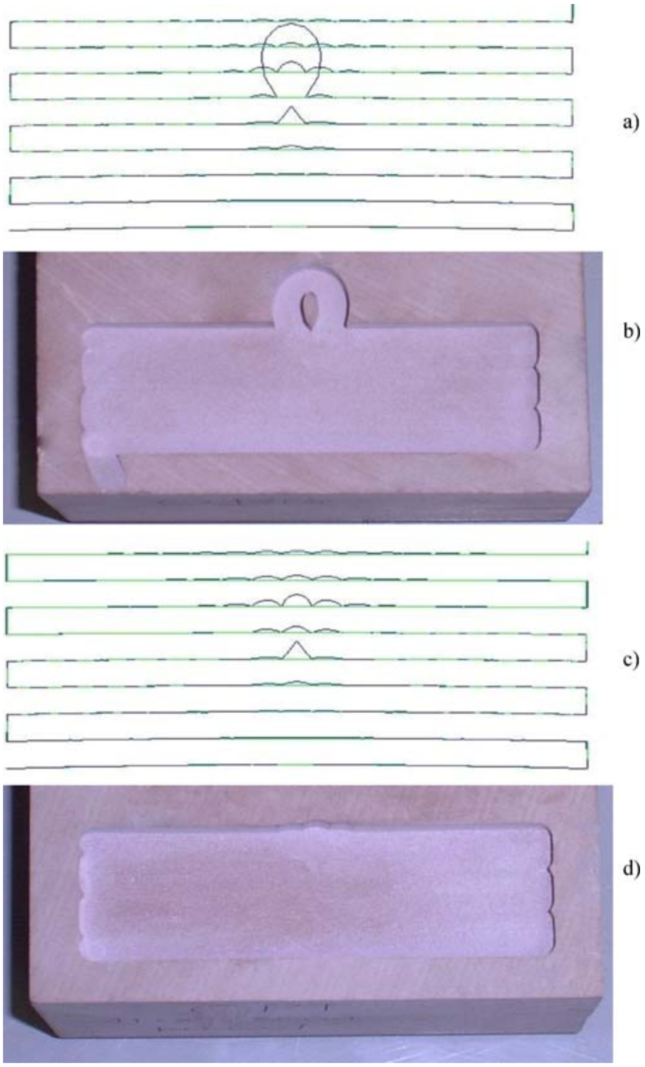

Fig. 3 **a** Conventional tool path simulated by the virtual machine. **b** Surface S_1 machined by HERMLE UWF902H. **c** Optimized tool path, simulated by the virtual milling machine. **d** Optimized surface machined by HERMLE UWF902H

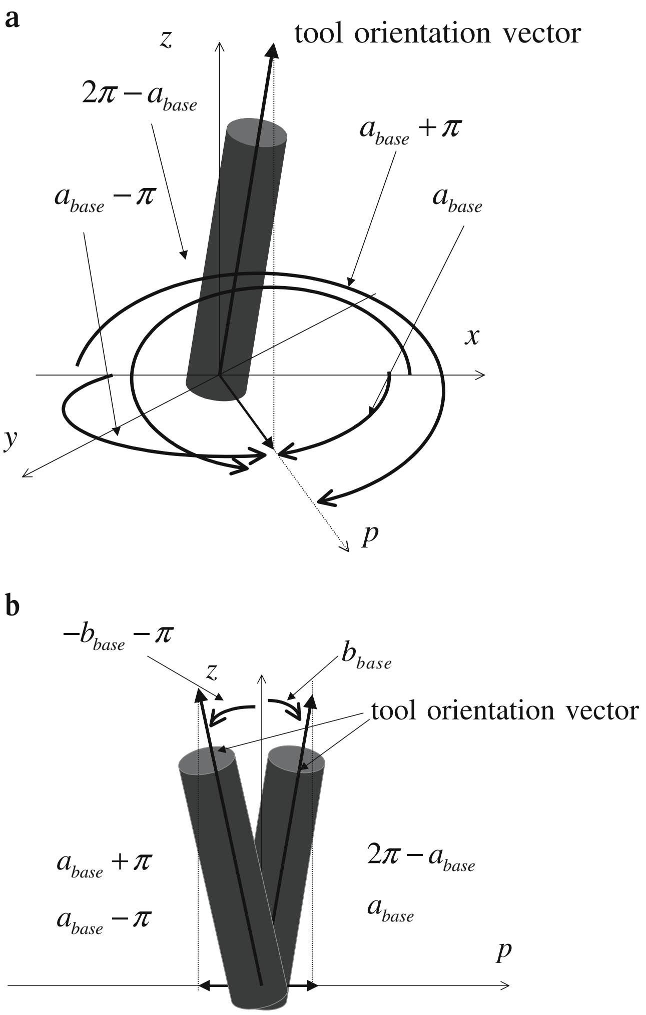

Fig. 4 **a** Four possible rotations for the *a* angle. **b** Two possible rotations for the *b* angle

undercut, we understand it as a part of the trajectory resulting in an excessive amount of the removed material. The overcut is understood as removing less material than is required.

We define the undercut and overcut error as follows:

$$\varepsilon_U = \sum_p \int_0^1 \begin{cases} \left(W^D_{p,p+1} - W_{p,p+1}\right)^2 & \text{if } z^D_{p,p+1}(t) > z_{p,p+1}(t) \\ 0 & \text{otherwise} \end{cases} dt$$
$$\varepsilon_O = \sum_p \int_0^1 \begin{cases} \left(W^D_{p,p+1} - W_{p,p+1}\right)^2 & \text{if } z^D_{p,p+1}(t) \le z_{p,p+1}(t) \\ 0 & \text{otherwise} \end{cases} dt \tag{5}$$

The corresponding undercut and overcut angle variation are then defined by:

$$c_U = \sum_p c_{U,p,p+1}$$
$$c_O = \sum_p c_{O,p,p+1}$$

where:

$$c_{U,p,p+1} = \sqrt{\Delta a^2_{p,p+1} + \Delta b^2_{p,p+1}}\,\Delta_{U,p,p+1}$$
$$c_{O,p,p+1} = \sqrt{\Delta a^2_{p,p+1} + \Delta b^2_{p,p+1}}\,\Delta_{O,p,p+1}$$
$$\Delta^U_{p,p+1} = \int_0^1 \begin{cases} t & \text{if } z^D_{p,p+1}(t) > z_{p,p+1}(t) \\ 0 & \text{otherwise} \end{cases} dt \tag{6}$$
$$\Delta^O_{p,p+1} = \int_0^1 \begin{cases} t & \text{if } z^D_{p,p+1}(t) \le z_{p,p+1}(t) \\ 0 & \text{otherwise} \end{cases} dt$$

Clearly, $\varepsilon = \varepsilon_O + \varepsilon_U$ and $c = c_O + c_U$. Since undercut is not repairable, reducing the undercut error is often much more important than reducing the overcuts or the total error.

4 Optimization problem and a solution method

In this section, we will give an introductory example of the optimization method. Next, we will introduce a general optimization problem and a solution method.

When the tool passes through an area near a stationary point, the rotation angles may jump considerably, leading to

Table 1 Kinematics error for the optimized and non-optimized tool path. Surface S_1 MAHO600E

Grid size	No optimization max error/ max angle var (mm/°)	Optimization max error/ max angle var (mm/°)	Max error/angle var decrease (%)	Path length non-opt/opt(mm)
10×20	23.862/168.779	12.426/78.072	47.925/53.743	2,825.67/2,255.54
15×20	19.300/162.186	8.517/101.524	55.870/34.403	2,500.3/2,123.02
20×20	20.228/160.08	7.558/84.927	62.636/46.947	2,367.57/2,101.15
30×20	16.253/141.89	7.162/88.23	55.934/37.818	2,183.63/2,038.27
40×20	8.711/103.885	6.878/88.399	21.042/14.907	2,069.32/2,020.11
100×20	7.395/90.898	7.103/89.102	3.949/1.976	1,916.11/1,911.17
130×20	3.999/68.416	3.999/68.416	0.000/0.000	1,876.49/1,876.49

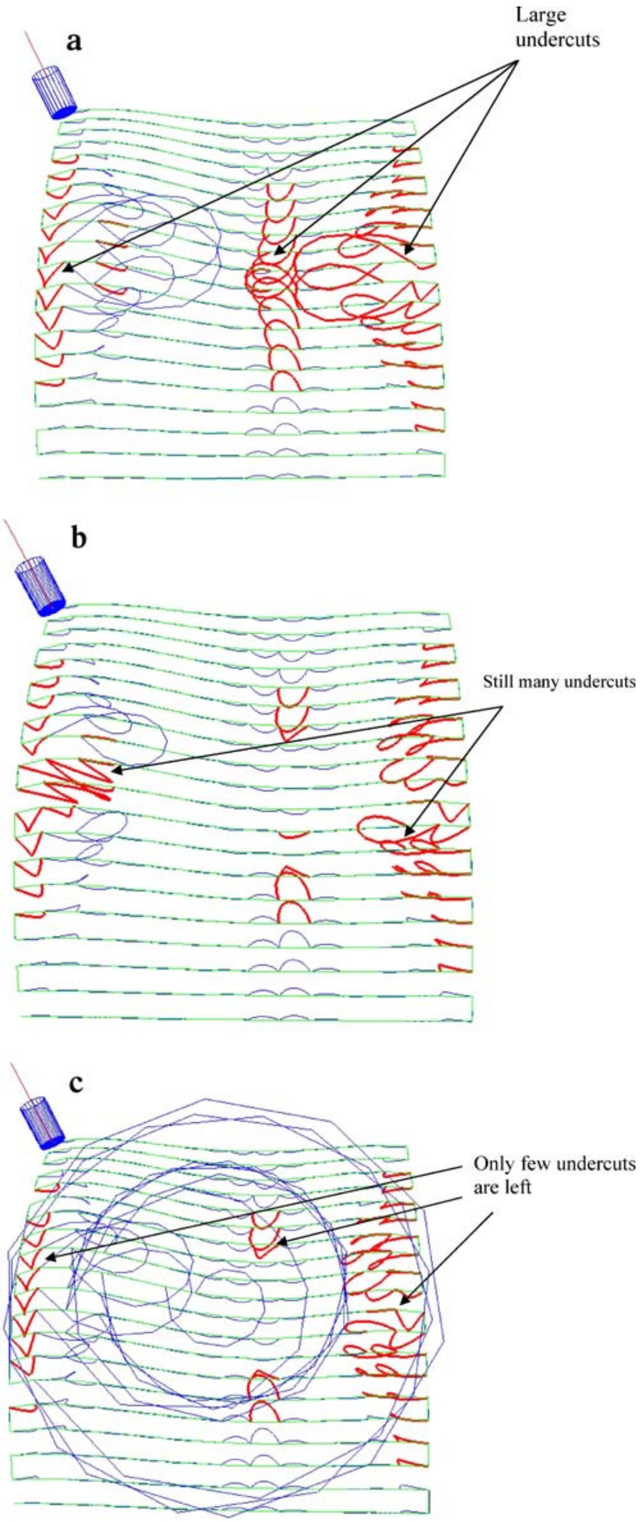

Fig. 5 **a** Conventional tool path for S_2 on HERMLE UWF902H. Large undercuts are indicated by *thick curves*. **b** Tool path optimized with regards to the total error. There are still many trajectories with large undercuts. **c** Tool path optimized with regards to the undercut error. The large undercuts have been replaced by large but harmless overcuts

unexpected deviations from the prescribed trajectory and large kinematics errors. The sharp angular jumps create loop-like trajectories of the tool. Our introductory example presents a cut performed for a part surface S_1 as depicted in Fig. 2 on HERMLE UWF902H. Figure 3a demonstrates a large loop in the case of machining a curve belonging to the surface located close to the stationary point. It also shows that linearization of the tool path is not always applicable and could even be dangerous, since moving the tool along the loop-like trajectories could destroy the workpiece and even lead to a collision with the machine parts. Figure 3b shows that changing the sequence of the rotations leads to a much less pronounced loop and, consequently, to a significant reduction of the kinematics error.

Now, using our conjecture that minimizing the total angle variation reduces the kinematics error, we formulate the following minimization problem:

$$\underset{\Lambda}{\text{minimize}}(w_O c_O + w_U c_U) \tag{7}$$

where Λ is a set of possible rotations and w_{O} and w_{U} are the weighting coefficients representing the importance of the overcuts and undercuts. For instance, if $w_{\text{O}}=w_{\text{U}}=1$, then the total angle variation will be minimized. If $w_{\text{O}}=0$ and $w_{\text{U}}=1$, only the variation along the undercuts will be minimized. If $w_{\text{O}}=0.1$ and $w_{\text{U}}=1.0$, then the variation is minimized with regards to both the undercut and overcut, but the importance of the undercut is 10 times larger than that of the undercut etc. The weights may also depend on the variation itself. For example, one may establish the following rule "if $c<c'$, then $w_{\text{O}}=0$ and $w_{\text{U}}=1$, else $w_{\text{O}}=w_{\text{U}}=1$", where c' is a prescribed threshold. In other words, if the total variation is not too large, we minimize only the undercuts; however, if the variation is more than a prescribed c', then the minimization applies to the total variation.

So how to evaluate set Λ? Consider the case of a five-axis milling machine with the rotary axis on the table, such as MAHO600E and HERMLE UWF902H. If the tool is aligned along the surface normal, then the rotation angles are evaluated by:

$$a_{base} = \begin{cases} tan^{-1}\left(\frac{j}{i}\right) & i > 0 \text{ and } j \geq 0 \\ \pi + tan^{-1}\left(\frac{j}{i}\right) & i < 0 \\ 2\pi + tan^{-1}\left(\frac{j}{i}\right) & otherwise \end{cases} \tag{8}$$
$$b_{base} = -sin^{-1}(k)$$

where (i, j, k) is the surface normal. It is not hard to demonstrate that the inverse kinematics admits four solutions (Fig. 4a,b), given by:

$$\Lambda = \begin{cases} a_{base},\ b_{base} \\ a_{base} - 2\pi,\ b_{base} \\ a_{base} - \pi,\ -b_{base} - \pi \\ a_{base} + \pi,\ -b_{base} - \pi \end{cases} \tag{9}$$

Solutions similar to Eqs. 8 and 9 can be established for other types of the machine kinematics, such as "one axis on the tool and one on the table" and "both axis on the tool."

It should be noted that we consider the case when the tool vector coincides with the surface normal only for the sake of simplicity. Usually, the tool must be inclined in order to avoid curvature interference and global gouging. However, there is no loss of generality, since appropriate adjustments of the above equations for the inclined tool can be easily made. Finally, the number of solutions depends on the number of the rotational degrees of freedom d as 2^d. In other words, for a six-axis machine, we will have eight solutions and so forth. It is not hard to demonstrate that the optimization is the shortest-path problem. The corresponding graph is constructed in such a way that each position Π_p is characterized by four graph nodes Λ_p, where the edge between the nodes represents the kinematics error or the angle variation (see Fig. 4). Therefore, such a minimization could be performed by a conventional so-called greedy discrete algorithm.

We apply a classical shortest-path algorithm to the resulting directed acyclic graph as follows. The nodes represent the tool positions and the arcs represent the error or the angle variation along the trajectory. Next, we detect the overcuts and undercuts. Every trajectory $W_{p,p+1}(t) = \left(x_{p,p+1}(t),\ y_{p,p+1}(t),\ z_{p,p+1}(t)\right)$ is subdivided by a certain number of points t_k. At every point $W_{p,p+1,k} = \left(x_{p,p+1,k},\ y_{p,p+1,k},\ z_{p,p+1,k}\right)$, we find the corresponding parametric coordinates $(u_k,\ v_k)$ by numerically solving the following system:

$$\begin{cases} x_{p,p+1,k} = x_S(u,\ v) \\ y_{p,p+1,k} = y_S(u,\ v) \end{cases} \tag{10}$$

Note that, for explicit surfaces, the above equations are linear and, therefore, a numerical solution is not required.

Now, given the solution $(u_k,\ v_k)$ to the system Eq. 10, we calculate $z_S(u_k,\ v_k)$ for every k. Next, we calculate the angle variations $c_{U,\ p,\ p+1}$ and $c_{O,\ p,\ p+1}$ for every p and for every graph node. Finally, we use the weights w_O and w_U and apply Dijkstra's shortest-path algorithm (see, for instance, [21]).

5 Numerical examples and cutting experiments

We have shown that a loop due to a large variation in the rotation angles can be almost entirely eliminated (Fig. 3a,b) for a surface having a single stationary point. Table 1 shows the performance of the method for the same surface machined on MAHOO600E. The optimization was performed without discriminating between ε_U and ε_O. Consider Table 1. Clearly, minimizing the total angle variation leads to minimizing the total kinematics error. On the contrary, an increase of the number of CL points does not necessarily lead to a decrease of the error (see lines 40×20 and 100×20), since the grids are not nested (the points of the coarse grid do not necessarily belong to the fine grid).

A very useful optimization can be performed when the undercuts and overcuts are differentiated. Consider machining surface S_2 given by:

$$S_2(u,\ v) = \begin{pmatrix} 100u - 50 \\ 100v - 50 \\ -80v(v-1)(3.55u - 14.8u^2 + 21.15u^3 - 9.9u^4) - 28 \end{pmatrix}$$

Table 2 Undercut error optimization. S_2 on HERMLE UWF902H

Grid size	No optimization max error (mm)/ max angle var (°)		Undercut optimization max error (mm)/max angle var (°)		Path length non-opt/ opt (mm)	Angular variation non-opt/ opt (°)
	Undercut	Overcut	Undercut	Overcut		
10×20	19.11/139.78	23.40/169.95	13.33/109.90	56.16/356.22	2,929.02/3,641.59	7,763.87/9,317.35
15×20	9.54/32.96	19.28/162.13	9.52/32.76	60.93/364.84	2,676.55/3,985.66	7,921.93/10,131.55
20×20	8.07/88.17	20.21/160.01	4.41/23.83	60.93/364.84	2,486.07/5,021.87	7,972.646/11,477.04
30×20	6.28/149.23	16.24/141.87	2.40/45.76	51.55/372.37	2,253.76/3,845.65	8,018.01/10,119.48
40×20	5.41/139.58	8.81/104.05	1.60/14.187	13.24/130.02	2,117.19/2,166.59	8,035.95/7,993.36
100×20	0.00/4.00	7.40 /90.83	0.00/4.00	7.13/89.27	1,924.58/1,919.35	8,055.03/8,032.72
130×20	0.00/3.09	4.00/68.34	0.00/3.09	4.00/68.34	1,880.72/1,880.72	8,056.50/8,056.50

by HERMLE UWF902H. Let w_O=0 and w_U=1; that is, the minimization is performed only with regards to the undercuts. Figure 5a displays the conventional tool path with undercuts (the thick curves). Figure 5b shows how minimization of the total error eliminates the loops located at the right side of the workpiece. However, minimization of only the undercut error produces an entirely different path. The undercut loops at the boundary of the workpiece have been replaced by large but harmless overcuts (see Fig. 5c). Table 2 shows the behavior of the undercut and overcut error, along with the angle variation as the number of points in the cutting direction increases. The decrease in the undercut error has been achieved at the expense of the increase of the overcuts and the length of the path.

Let us demonstrate the techniques by optimizing the undercuts on MAHO600E. The conventional tool path and the tool path optimized by means of w_O=1 and w_U=1 are displayed in Fig. 6a,b, whereas a tool path optimized with w_O=0 and w_U=1 are displayed in Fig. 6c. The corresponding machined surfaces are displayed in Fig. 7a–c. Clearly, optimization with regards to only undercuts

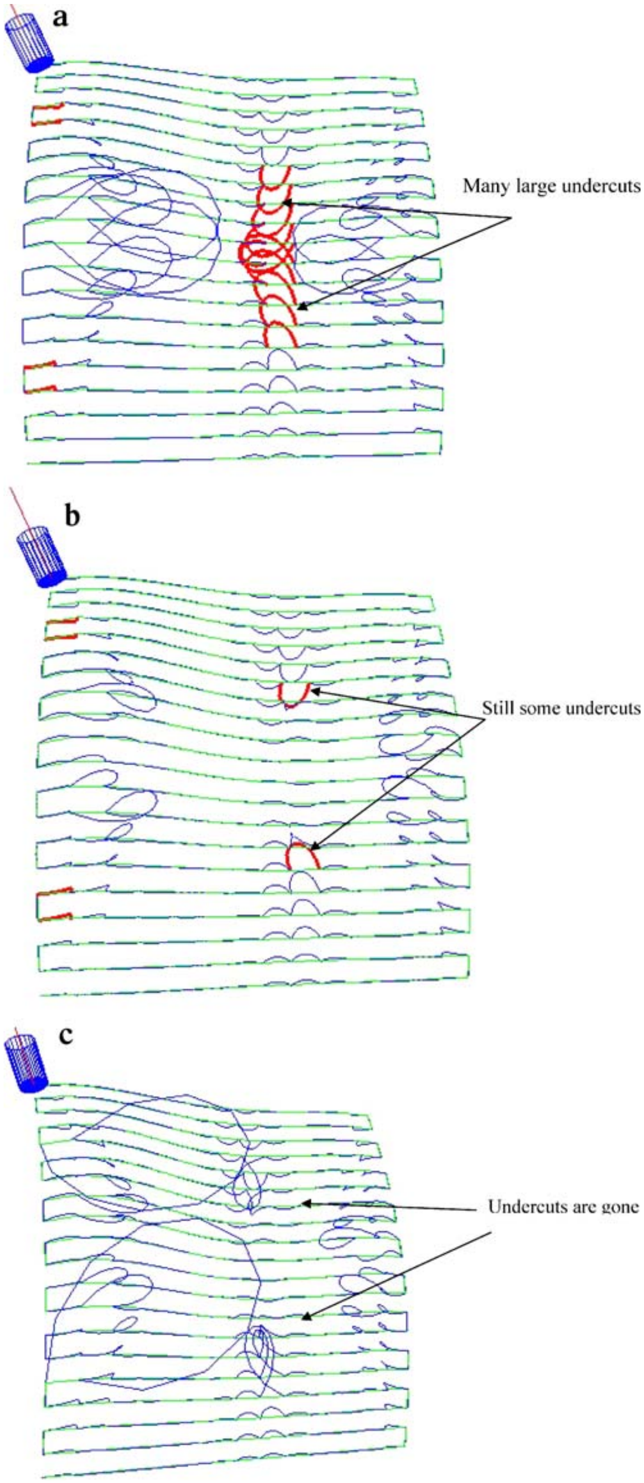

Fig. 6 **a** Conventional tool path for S_2 on MAHO600E. **b** Optimization with regards to the total error. **c** Optimization with regards to the undercut error, S_2, MAHO600E

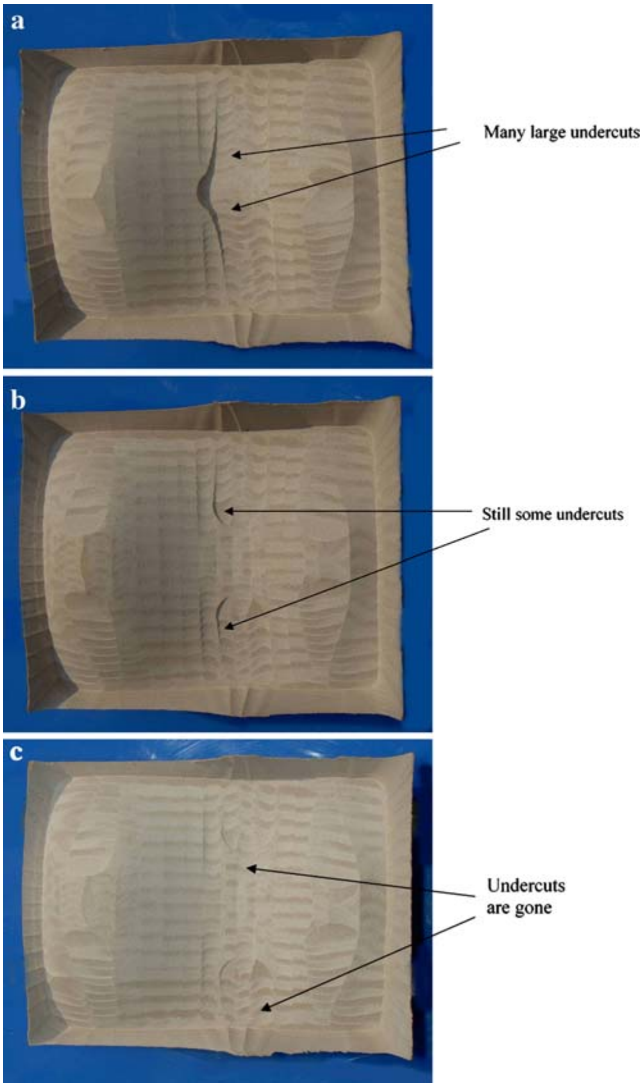

Fig. 7 **a** Without optimization, S_2, MAHO600E (corresponds to Fig. 6a). **b** Optimization with regards to the total error, S_2, MAHO600E (corresponds to Fig. 6b). **c** Optimization with regards to the undercut error, S_2, MAHO600E (corresponds to Fig. 6c)

makes it possible to cut a rough but reasonable surface which could be further refined, whereas cutting without the undercut optimization may destroy the workpiece. The behavior of the kinematics error versus the number of points and the maximum angle variation is displayed in Table 3. Comparing HERMLE UWF902H and MAHO600E (Table 2 and Table 3) shows that the accuracy of the two cuts without optimization is comparable, whereas the optimized cut on MAHO600E outperforms HERMLE UWF902H.

It should be noted that the method applies to the case when the surface is known only at some discrete set of points. In this case, the minimization can be performed with regards to either the angle variation or with regards to $\hat{\varepsilon}$ (see Eq. 2), defined by means of the difference between the actual and the linearized tool path. In particular, the method can be applied to optimize a CNC program, that is, a code containing a sequence of coordinates of the tool in the five-dimensional space. Figure 8a,b illustrates the techniques applied to a multipatch surface designed to produce an industrial impeller. The length of the tool path has been reduced by 18.164% or by almost 5,000 mm.

Finally, before the above-mentioned optimization, we perform following correction:

$$a_{i+1,\,new} = \begin{cases} a_{i+1} - 2\pi & \text{if } a_{i+1} - a_i > \pi \\ a_{i+1} + 2\pi & \text{if } a_{i+1} - a_i < -\pi \\ a_{i+1} & otherwise \end{cases}$$

The above correction eliminates jumps exceeding π, so that the shortest-path routine deals only with $|a_{i+1} - a_i| \leq \pi$. Corrections of this type is not always required, since it has been incorporated in some modern controllers, such as the controller of HERMLE UWF902H.

6 Uniform angular grids

This section will present our algorithm based on the so-called equi-distribution principle (see [12]) in the angular space. The algorithm inserts additional points in such a way that the resulting total angle variation is minimized. The algorithm does not require evaluation of the actual trajectories.

Many algorithms to calculate the so-called feed rate (the step between the consecutive CL points) have been proposed. The simplest approach is to linearize the surface along the tool path, calculate the error, and insert additional points until the error is within the required tolerance (see, for example, [11]). Another approach is to interpolate the desired trajectory by a spline or a NURBS approach and establish the required feed rate using properties of the curve. The feed rate may depend on the space coordinates of the curve, as well as on the angular changes; see, for example, [22]. However, the above methods do not include the actual machine kinematics. A combination of the actual kinematics of the machine with an interpolating method is made by Lo [9, 10]. However, this paper does not consider a global minimization and the particular impact of the rotations.

A general framework for applying grid generation to tool-path optimization was proposed by Makhanov et al. [12] and developed by Makhanov and Ivanenko [13]. It was shown that grid generation based on the Dirichlet function, invoked in an iterative loop with a suitable preprocessing, substantially decreases the kinematics error. Grid generation is applicable if $\varepsilon \to 0$ as the area of the grid cell A tends to zero. If it is the case, we introduce a function representing the so-called equi-distribution principle subjected to the constraints mentioned in the introduction. The equi-distribution principle requires that $A\varepsilon \approx$const across the entire grid. However, it requires the calculation of the actual error.

In this paper, we consider a rough cut when the tool rotation angles jump as described above. In this case, the adaptive grid may require very small spatial cells. As a result, the algorithm may create twisted and degenerated cells which may not converge or may show a very slow convergence. Besides, constructing a structured grid requires an equal number of tool positions along each track of the tool path. Therefore, it is often appropriate to

Table 3 Undercut error optimization, S_2 on MAHO600E

Grid size	No optimization max error (mm)/max angle var (°)		Optimization max error (mm)/max angle var (°)		Path length non-opt/opt (mm)	Angular variation non-opt/opt (°)
	Undercut	Overcut	Undercut	Overcut		
10×20	8.41/169.66	23.86/170.03	1.84/28.26	38.46/157.94	2,825.67/3,575.11	7,768.03/7,867.60
15×20	7.23/164.57	19.30/162.20	1.98/40.47	32.92/159.15	2,500.3/2,417.3	7,926.18/7,339.96
20×20	6.78/159.42	20.23/160.83	0.96/26.98	7.56/87.37	2,367.57/2,101.14	7,976.88/7,148.41
30×20	6.25/149.31	16.25/141.94	0.00/12.57	7.16/89.19	2,183.63/2,032.2	8,022.24/7,848.15
40×20	5.39/139.65	8.71/104.12	0.00/9.64	6.88/88.56	2,069.32/2,020.11	8,040.19/7,788.45
100×20	0.00/4.00	7.40/90.90	0.00/4.00	7.10/89.20	1,916.11/1,911.17	8,059.30/8,036.42
130×20	0.00/3.09	4.00/68.42	0.00/3.09	4.00/68.41	1,876.49/1,876.49	8,060.76/8,080.76

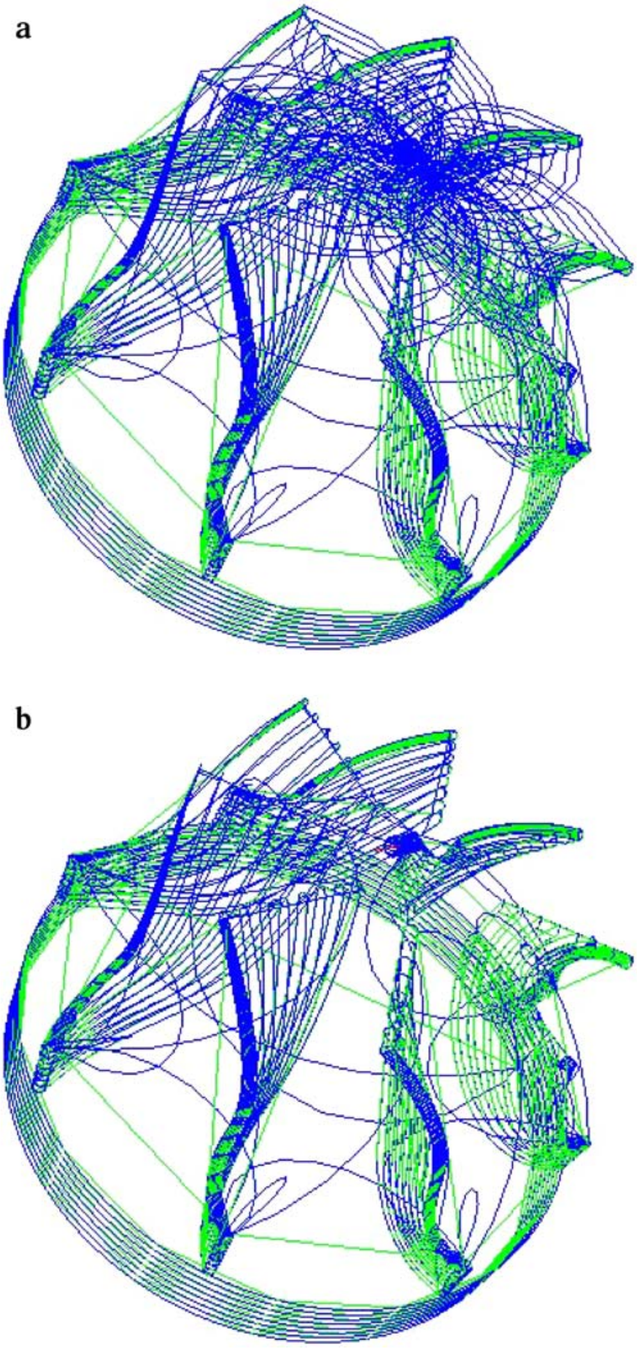

Fig. 8 **a** Conventional tool path for impeller surface. The path length is 30,707 mm. **b** Optimized tool path for the impeller. The path length is 25,129.4 mm

construct a one-dimensional grid along each track of the tool independently.

Suppose we have constructed a "basic" tool path represented by a rectangular or curvilinear grid. This grid makes it possible to perform an initial evaluation of the angle variation. Now we are allowed to insert additional points to decrease the error. How should these points be inserted? We will construct local grids with equal angular increments around points characterized by large angle variations. We shall call such grids the uniform angular grids. Recall that, near the stationary points, there often exists a neighborhood such that one of the rotations is not performed or one of the angles changes faster than another angle. In this case, it is sufficient to construct a grid uniform only with regards to the "fast" angle.

The algorithm consists of the following steps. First of all, it detects points characterized by sharp angular variations. Next, for every point, the algorithm determines the positions between which the uniform angular grid should be constructed. Usually, the interval includes a few tool positions right before and after the singularity which creates the kinematics loop. The space between every pair of the selected points W_p and W_{p+1} must be subdivided in such a way that $\Delta a_{p,\ p+1,\ i,\ i+1}$≈const or $\Delta b_{p,\ p+1,\ i,\ i+1}$≈const for every subinterval $[i,\ i+1]$.

We have already noted that the stationary point is cut by either changing sharply the first or the second rotation angle. Irrespective of whether the singular point is a CC point or not, the tool will follow a path "across or around the hill," depending on the choice of the rotation angles (see a relevant discussion in Munlin et al. [15]). Consequently, the algorithm selects the appropriate angle (say, angle a) and subdivides the angular interval $\Delta a_{p,\ p+1}$ into equal subintervals $\Delta a_{p,\ p+1,\ i,\ i+1}$, such that $a_{p,p+1,i} = a_p + \Delta_{p,p+1} i$, where $\Delta_{p,\ p+1}$ is the corresponding step. The corresponding spatial positions S_i are found from the surface equation as follows. First of all, we specify a parametric spatial trajectory T=$T(s)$ on the surface through the stationary point. Next, for each i, we numerically solve the equation $a(T(s)) = a_{p,p+1,i}$, where $a(T(s))$ denotes the rotation angle on $T(s)$ calculated from the surface normal. The above equation is solved numerically by the standard bisection method.

7 Uniform angular grids: numerical and machining experiments

Consider the machining surface S_2 by MAHO600E. This time, we are allowed to add additional points in order to decrease the error. First of all, consider the surface machined using the iso-parametric tool path represented by a local uniform grid (see Fig. 9a). The large error loops appearing due to sharp angle variations can be eliminated by inserting the additional points in these areas using the uniform angular distribution of the points, as shown in Fig. 9b. The substantial reduction of the error (50 times!) has been achieved by inserting only eight points inside each loop where the error is larger than a required tolerance.

It may seem that inserting a uniformly spaced set of points may lead to the same or similar result. However, it is

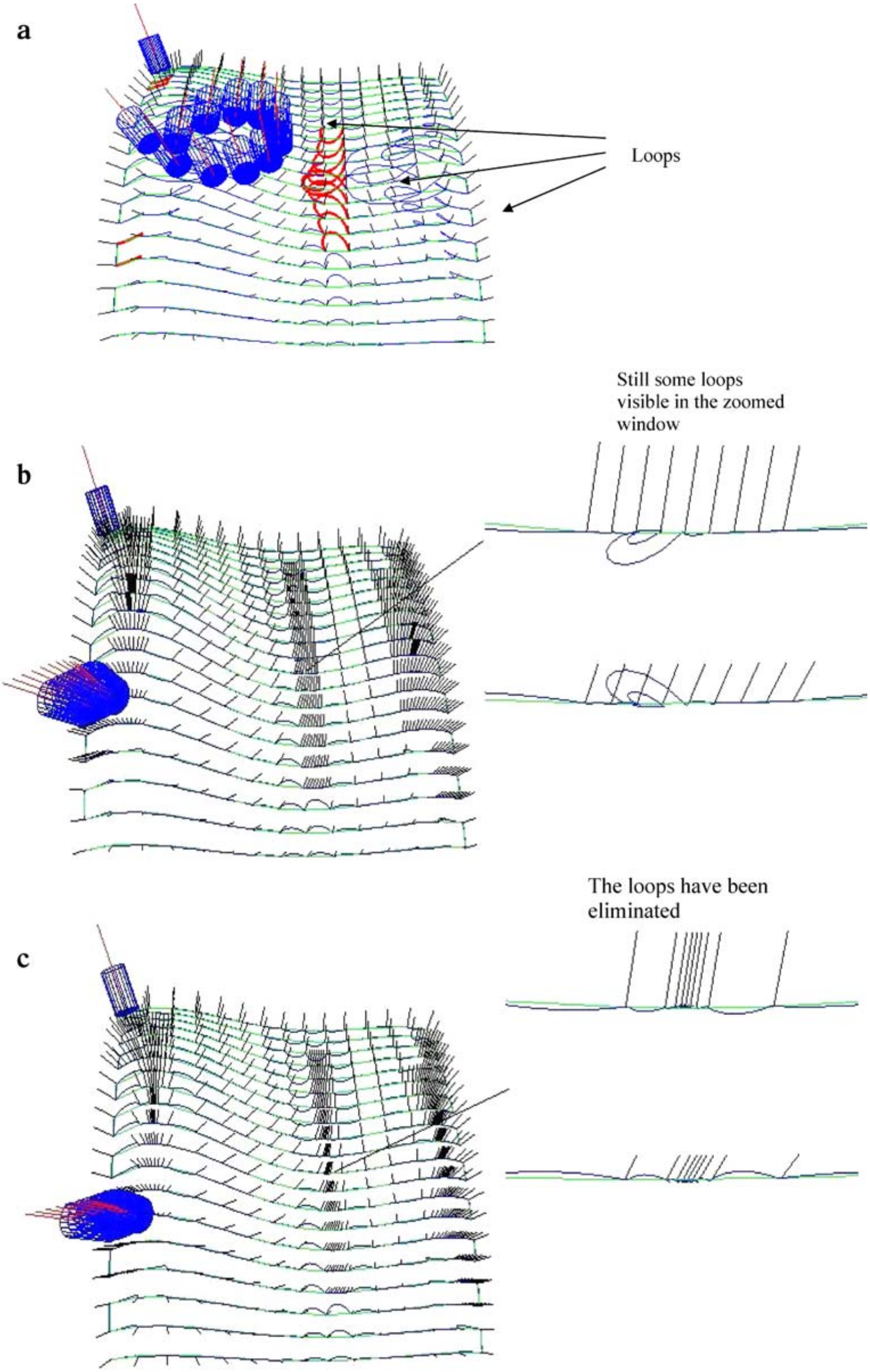
a
Loops
b
Still some loops visible in the zoomed window
c
The loops have been eliminated

Fig. 9 **a** Tool path and tool orientations, S_2, MAHO600E, conventional points insertion. **b** Tool path and tool orientations, S_2, MAHO600E, conventional points insertion. **c** Tool path and tool orientations, S_2, MAHO600E, angular grid insertion

not the case. Clearly, the uniform distribution of the points does not have a significant impact on the error. The error has been reduced only by 4 times by inserting the eight additional points. Table 4 displays the kinematics errors and the rotation angles before and after applying the proposed angle insertion. Clearly, the uniform angular grid allows us to substantially decrease the error, whereas the conventional spatial grid is not efficient.

Clearly, when a certain number of additional points have been inserted, the error decrease is approximately the same as for the space or angular insertion. Therefore, the method is applicable only for the rough cuts characterized by sharp

Table 4 Error versus number of inserted points; the basic grid size is 15×20

Number of inserted points	Max error (mm) conventional/angular grid	Path length (mm) conventional/angular grid	Angular variation (°) conventional/uniform angular grid
0	19.300/19.300	5.714/5.714	162.202/162.202
8	4.241/0.416	0.714/0.168	68.660/20.631
16	1.707/0.138	0.357/0.084	43.016/10.629
32	0.490/0.092	0.178/0.042	20.647/5.352
64	0.158/0.089	0.089/0.021	10.955/2.680
128	0.099/0.089	0.044/0.010	5.378/1.340

variations of the rotation angles. Figure 9b,c explains the advantages of the angular grids. Clearly, the conventional uniform grid does not remove the kinematics loops, although the amplitude has been decreased. As opposed to that, the angular grid changes the behavior of the trajectories by entirely eliminating the loops. Figure 10a,b shows the differences between the machined workpiece.

It should be noted that our cutting experiments have been performed with the flat-ended cutter selected due to its popularity for fast, rough milling when the sharp angular variations are the most expected. However, the proposed techniques are also applicable to the ball-nose cutter being often used for finish machining. As a matter of fact, the proposed algorithms apply to the general APT cutter (automatically programmed tool) without major modifications. The equations of the APT cutter include the most popular shapes, such as the flat-end shaped cutter, toroidal cutter, the ball-nose cutter, etc. [3, 4].

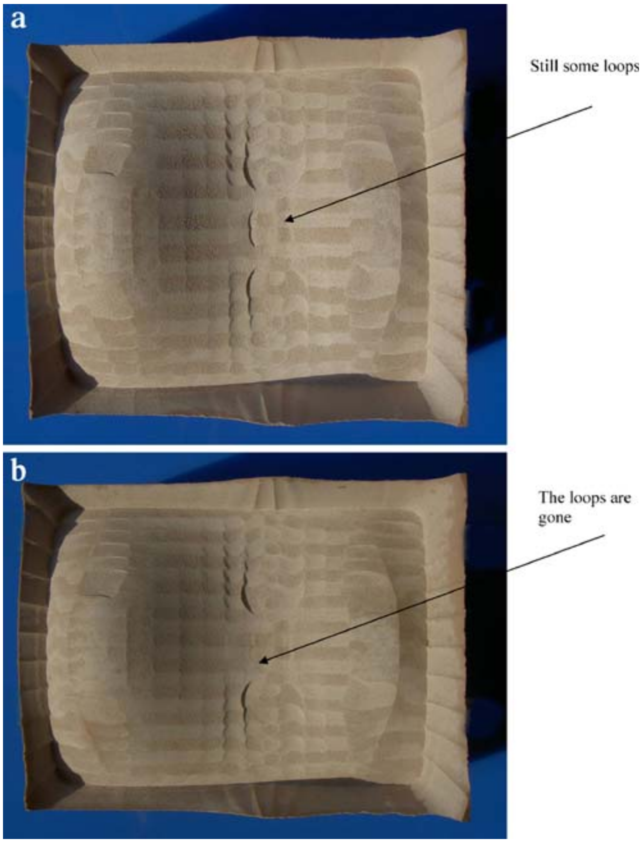

Fig. 10 **a** Spatial grid for S_2, MAHO600E (corresponds to Fig. 9b). **b** Angular grid, S_2, MAHO600E (corresponds to Fig. 9c)

Combination of the two proposed methods might lead to even better optimizations. However, usually, the angular grid should be constructed before the shortest-path procedure or in an iterative loop presented below. Even when the angular grid is applicable after the shortest-path routine, the implementation is not as straightforward as it may seem, since constructing the angular grid after the shortest-path optimization requires us to keep track of changes associated with every point. Furthermore, applying the angular grid after Algorithm 1 is often not possible because inserting even one point may create an entirely different graph for the shortest-path optimization. Let us illustrate this case by an example. Consider surface S_2 with the tool path obtained after optimal sequencing by Algorithm 1 (see Fig. 11). The largest loop between point 141 and 142 which can not be treated by Algorithm 1 is on the right side of the surface. The angles before applying Algorithm 1 are $a_{141} = 319^\circ$, $b_{141} = -79^\circ$, $a_{142} = 224^\circ$, and $b_{142} = -80^\circ$. The shortest-path optimization produces $a_{141} = 319^\circ$, $b_{141} = -79^\circ$, $a_{142,\text{ new}} = a_{142} + 180 = 404^\circ$, and $b_{142,\text{ new}} = -b_{142} - 90 = -100^\circ$. Inserting a point in the middle of the loop yields $a_{\text{mid}} = 267^\circ$ and $b_{\text{mid}} = -82^\circ$. Taking into account the "history," we modify the pair of angles as follows: $a_{\text{mid}} = a_{\text{mid}} + 180^\circ = 447^\circ$ and $b_{\text{mid}} = -b_{\text{mid}} - 90 = -98^\circ$, which leads to a larger loop as depicted in Fig. 12. Note that

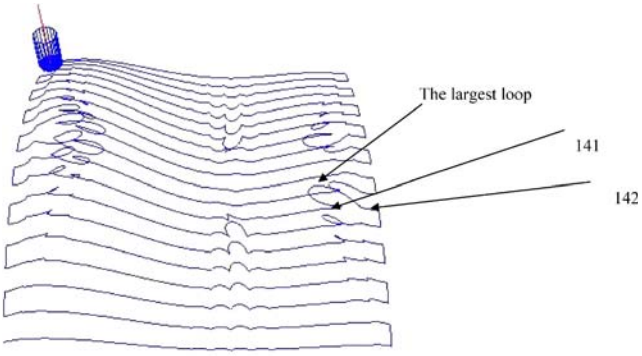

Fig. 11 Tool path for surface S_2 on MAHO600E after the optimal sequencing

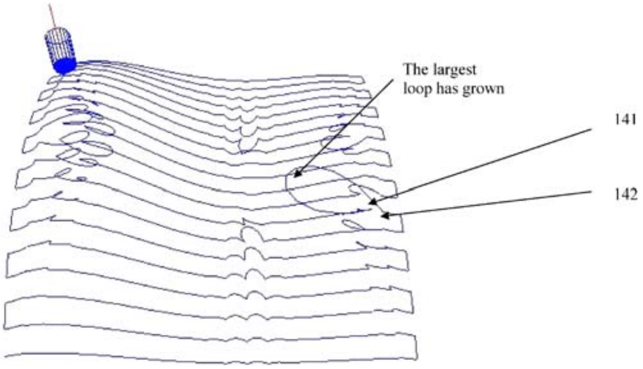

Fig. 12 Tool path for surface S_2 on MAHO600E after the optimal sequencing and inserting one point into the largest loop

$a_{mid} \notin [a_{141}, a_{142}]$any more. In other words, the additional point has destroyed the particular shortest path. Since the remaining part of the shortest path depends on these angles, the entire path has been destroyed and the optimization should be performed again.

The following procedure works very well when combining the two methods. First, we apply Algorithm 1. Second, we select the largest loops appearing after applying Algorithm 1. Next, we go back to the original angles and apply Algorithm 2, which inserts angular grids into the selected loops. Finally, we again apply Algorithm 1 to the modified tool path. If the required tolerance has not been achieved (new loops have appeared), we select the largest loops again and go back to inserting points into the original tool path.

Finally, apart from the machine kinematics, the accuracy of machining is often affected by the tool accelerations and decelerations, due to frequent changes of the tool path directions. Since the phenomenon often occurs in the areas of sharp variations of the rotation angles, let us discuss the proposed method with the reference to the acceleration errors.

In order to reduce the acceleration error, the entire tool path is usually treated as an interpolating curve parameterized with regards to the chord length between two consecutive reference points [5, 19, 20].

Generating the tool positions by incrementing the chord length leads to feed-rate instabilities due to the difference between the chord and the arc lengths. The instabilities induce undesirable accelerations and jerk fluctuations. Therefore, if the fit curve is parameterized with respect to the arc length, these accelerations will be eliminated. However, the above parameterization is still an open problem, even in 3D, although there exists a convenient nearly arc-length spline parameterization introduced by Wang and Yang [19] and Wang and Wright [20].

Our proposed procedures apply to the regions where the angular variations are the major source of errors. However, the procedures do not treat the accelerations explicitly. Therefore, when such accelerations lead to large inaccuracies, the proposed algorithms should be combined with the above-mentioned spline-interpolating methods. However, observe that reducing the *angular* variation leads to equal increments in the angular space. Therefore, it reduces the *angular accelerations* appearing in the five-axis mode due to sharp variations of the tool orientations.

Ideally, in five-axis machining, the tool path must be regarded as a curve in the five-dimensional space and parameterized with regards to its arc length in 5D. Such parameterizations should be associated with several sources of error, such as the kinematics error, errors due to accelerations, etc. Unfortunately, such parameterizations still constitute an open problem.

8 Conclusions

Minimization of the total angle variation for rough cuts leads to a substantial accuracy increase ranging from 10% to 80 %. The optimization can be formulated in terms of the total error, as well as in terms of the undercut and overcut errors. Optimization of the undercut error ensures against the removal of the excess material during the rough cut. Further improvement of the accuracy can be achieved by constructing the uniform grid in the angular space around the CC points characterized by large angle variations. The methods are most efficient in the case of the rough cut characterized by large angle variations, which produce considerable errors. Concatenation of the two methods is not as straightforward as it may seem. The shortest-path optimization should be applied in an iterative loop with the angular grids.

Acknowledgments This work is supported by the National Science and Technology Development Agency of Thailand (NSTDA) and by the Thailand Research Fund (TRF).

References

1. Arkin EM, Bender MA, Demaine ED, Fekete SP, Mitchell JS, Sethia S (2001) Optimal covering tours with turn costs. In: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), Washington, DC, January 2001, pp 138–147
2. Affouard A, Duc E, Lartigue C, Langeron J-M, Bourdet P (2004) Avoiding 5-axis singularities using tool path deformation. Int J Mach Tools Manuf 44(4):415–425
3. Bohez E, Makhanov SS, Sonthipermpoon K (2000) Adaptive nonlinear tool path optimization for 5-axis machining. Int J Prod Res 38(17):4329–4345

4. Bohez ELJ, Minh NTH, Kiatsrithana B, Natasukon P, Ruei-Yun H, Son LT (2003) The stencil buffer sweep plane algorithm for 5-axis CNC tool path verification. Computer-Aided Design 35 (12):1129–1142
5. Erkorkmaz K, Altintas Y (2001) High speed CNC system design. Part I: jerk limited trajectory generation and quintic spline interpolation. Int J Mach Tools Manuf 41(9):1323–1345
6. Lauwers B, Dejonghe P, Kruth JP (2003) Optimal and collision free tool posture in five-axis machining through the tight integration of tool path generation and machine simulation. Computer-Aided Design 35(5):421–432
7. Jung YH, Lee DW, Kim JS, Mok HS (2002) NC post-processor for 5-axis milling machine of table-rotating/tilting type. Mater Process Technol 130(131):641–646
8. Lee Y-S (1997) Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining. Computer-Aided Design 29(7):507–521
9. Lo C-C (1999) Efficient cutter-path planning for five-axis surface machining with a flat-end cutter. Computer-Aided Design 31 (9):557–566
10. Lo C-C (1999) Real-time generation and control of cutter path for 5-axis CNC machining. Int J Mach Tools Manuf 39(3):471–488
11. Li F, Wang XC, Ghosh SK, Kong DZ, Lai TQ, Wu XT (1995) Tool-path generation for machining sculptured surface. J Mater Process Technol 48(1):811–816
12. Makhanov SS, Batanov D, Bohez E, Sonthipaumpoon K, Anotaipaiboon W, Tabucanon M (2002) On the tool-path optimization of a milling robot. Comput Ind Eng 43(3):455–472
13. Makhanov SS, Ivanenko SA (2003) Grid generation as applied to optimize cutting operations of the five-axis milling machine. Appl Numer Math 46(3–4):353–377
14. Marchuk GI (1982) Methods of numerical mathematics, 2nd edn. Springer, Berlin Heidelberg New York
15. Munlin M, Makhanov SS, Bohez ELJ (2004) Optimization of rotations of a five-axis milling machine near stationary points. Computer-Aided Design 36(12):1117–1128
16. Munlin M, Makhanov SS (2001) A software for simulation of inverse kinematics of a 5-axis milling machine. In: Proceedings of the 16th International Conference on Production Research (ICPR-16), Prague, Czech Republic, July/August 2001
17. Sarma R (2000) An assessment of geometric methods in trajectory synthesis for shape-creating manufacturing operations. J Manufact Syst 19(1):59–72
18. Yoon J-H (2003) Tool tip gouging avoidance and optimal tool positioning for 5-axis sculptured surface machining. Int J Prod Res 41(10):2125–2142
19. Wang F-C, Yang DCH (1993) Nearly arc-length parameterized quintic-spline interpolation for precision machining. Computer-Aided Design 25(5):281–288
20. Wang F-C, Wright PK (1998) Open architecture controllers for machine tools. Part 2. A real time quintic spline interpolator. J Manuf Sci Eng 120(2):425–432
21. Weiss MA (1997) Data structures and algorithm analysis. Addison-Wesley, Reading, Massachusetts
22. Xu H-Y (2003) Linear and angular interpolation for planar implicit curves. Computer-Aided Design 35(3):301–317