

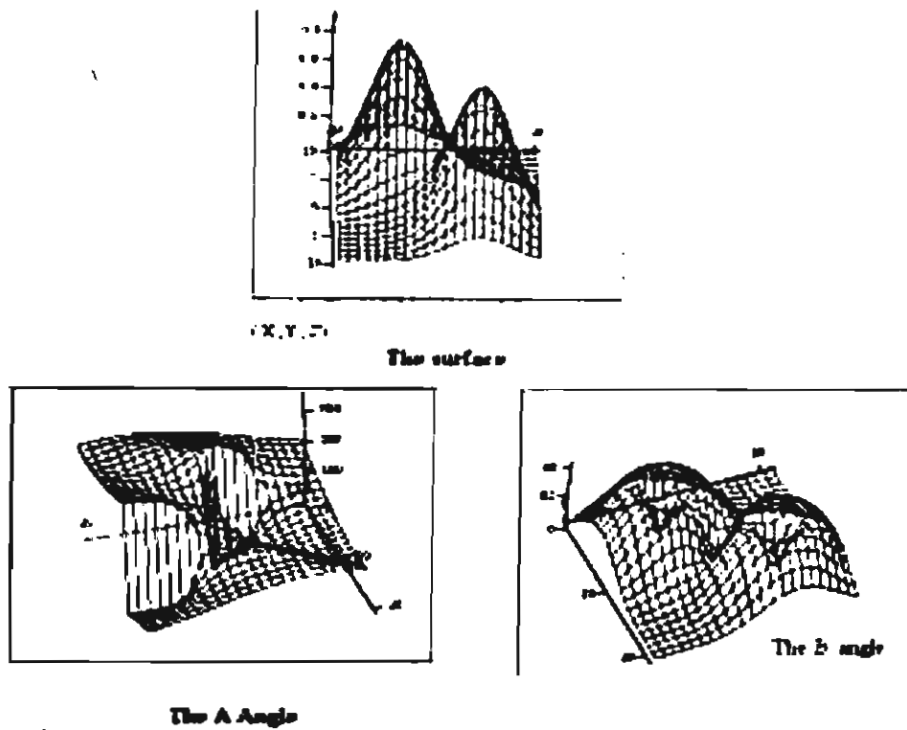


Figure A16. The two bell surface

Workpiece 60x60mm, Tool size 10mm

Surface quality: Excellent.

Overcut: Substantial, Undercut: NO.

Side effect: Substantial.

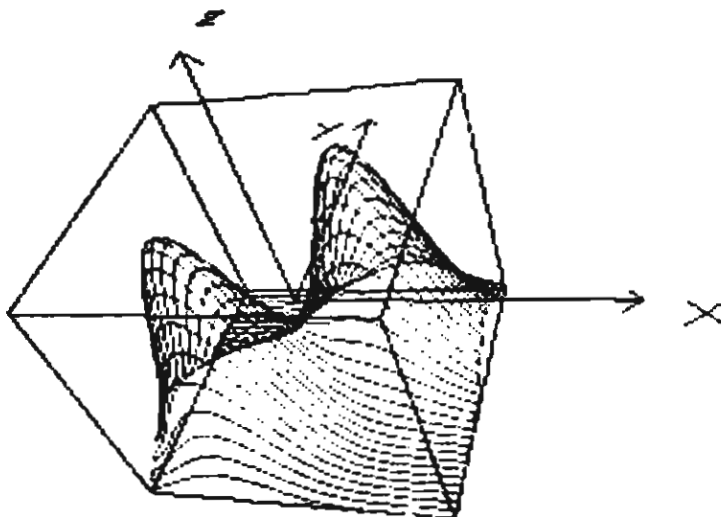


Figure A17. The graphical CL-Points of the two bell surface

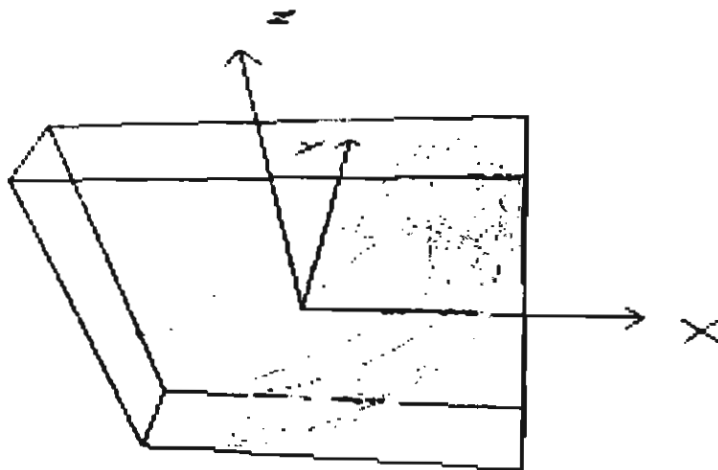


Figure A18. The graphical G-Code of the two bell surface

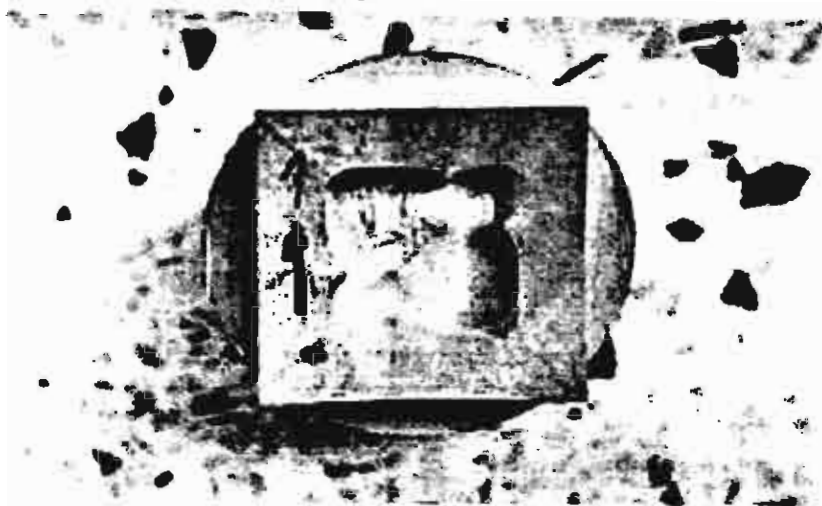


Figure A19. Finished two bell surface

EXPERIMENT 6. Workpiece simulation compared with practical machining and other commercial CAD/CAM systems

This experiment demonstrate the workpiece simulation by incorporating the actual tool trajectories produced by the VMM into the existing CAD/CAM solid modeling system such as Unigraphics.

Current CAD/CAM system normally simulates the workpiece by using the CL points in which they do not represent the actual tools trajectories. Therefore, the result of the simulation does not represent the actual cutting operations and kinematics errors can not be detected and visualized. Our experiment has shown the realistic trajectories simulation as illustrated in Figure A20 where it can be verified by the real cutting operations as shown in Figure A21.

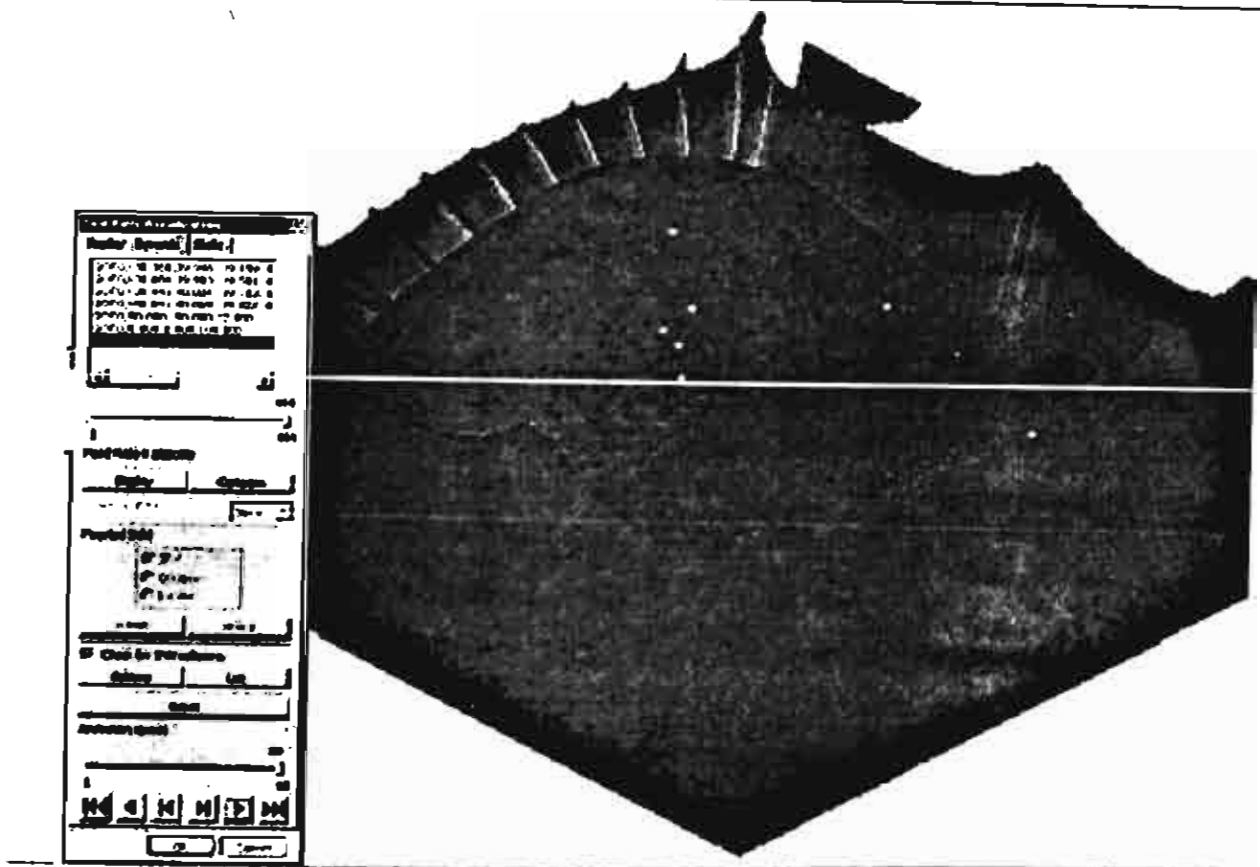


Figure A20. Workpiece simulation integrated with Unigraphics

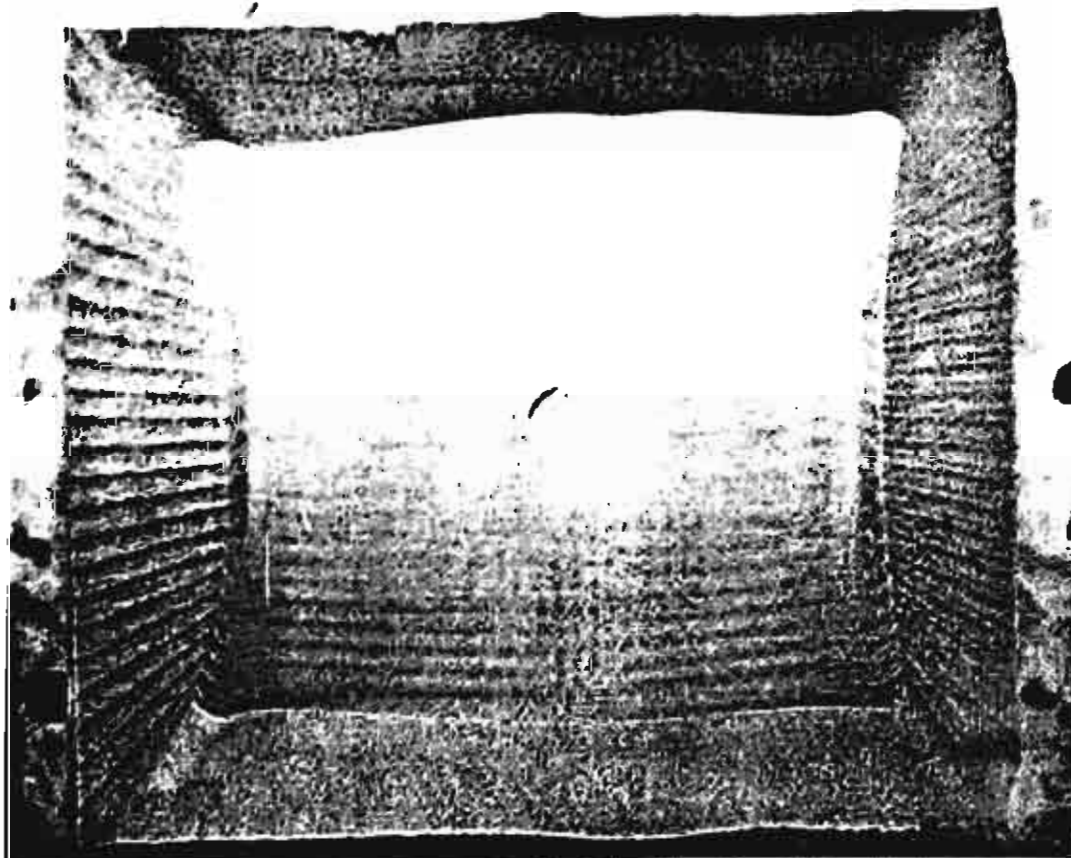


Figure A21. Workpiece verified by the Real Machining

Optimization of Rotations of a Five-Axis Milling Machine Near Stationary Points

M. Munlin¹, S.S. Makhanov¹ and E. L. J. Bohez²

¹Department of Information Technology, Sirindhorn International Institute of Technology, Thailand

²School of Advanced Technologies, Asian Institute of Technology, Thailand

Abstract. We consider a new algorithm designed for five-axis milling to minimize the kinematics error near the stationary points of the machined surface. Given the tool orientations, the algorithm optimizes the required rotations on the set of the solutions of the corresponding inverse kinematics equations. We solve the problem by means of the shortest path scheme based on minimization of the kinematics error.

We present an application of the proposed algorithm to tool-path planning and demonstrate the efficiency of the proposed scheme verified by practical machining.

Keywords: CNC-machines, inverse kinematics, tool-path optimization, the shortest path

1. INTRODUCTION

Optimization of the tool path of a five-axis milling machine presents a considerable challenge. Recent papers have displayed a number of sophisticated methods to optimize the conventional zigzag or spiral pattern (see a survey in [18]). Besides, a variety of methods is available to generate unconventional patterns, for instance: the neural network approach [17], the Voronoi diagram technique [4], the monotone chain method [15], the distance map method [4], space filling curves [18], grid generation methods [12,13], etc.

The optimization criteria and the set of optimized variables vary. The tool path can be optimized with regard to the machining time, accuracy, the length of the tool path, the width of the machining strip, the volume of the removed material, the size of the remaining scallops, etc. In this paper we will optimize the sequence of rotations of the five-axis milling machine which contribute to the inaccuracy of a machined workpiece in the vicinity of stationary points of the desired surface.

The machine is guided by axial commands $\Pi = (W, \mathfrak{R}) \in \mathbb{R}^5$ carrying the three spatial coordinates $W = (x, y, z)$ of the tool-tip in the machine coordinate system and the two rotation angles $\mathfrak{R} = (a, b)$. The tool path $\mathfrak{A} = \{\Pi_0, \Pi_1, \dots, \Pi_m\}$ is a sequence of coordinates in the five-dimensional space. The spatial coordinates of the tool path usually (but not necessarily) lie on the required surface $S \equiv S(u, v)$. Usually, the tool visits the positions Π_p following a structured spatial pattern such as the zigzag or the spiral pattern. However, the path could be also composed from a variety of the unconventional patterns and include tool retractions [18]. A full optimization scheme involves a model of cutting operations, topologies of the prescribed tool path patterns and an optimization procedure. Let p_c be the parameters related to the configuration of the machine (such as coordinates of the centers of rotation, workpiece offset relative to the machine coordinates, etc.) and p_t the parameters related to

the tool (such as the diameter, length, shape, etc) . The model of the cutting operations, being fed with p_c, p_r, S and Π , produces a result of machining, namely, the output surface $T \equiv T(u, v)$. The optimization is usually performed with regard to Π and p_r . The cutting operations could be optimized with regard to the machine configuration p_c as well. However, the "optimal" machine is often a purely theoretical issue [1]. Let $S \equiv S(u, v)$ be the required surface. The general optimization problem is then formulated by

$$\begin{aligned} & \text{minimize } \|\varepsilon\|, \\ & \Pi, p_r \end{aligned} \quad (1)$$

where ε denotes the cost function representing the error given by $\varepsilon(u, v) \equiv |S(u, v) - T(u, v)|$, where $\|\cdot\|$ is an appropriate norm. Optimization (1) is subjected to the following constraints

1) The scallop height constraint. The scallop between the successive tool tracks must not exceed the prescribed tolerance [11,12].

2) The local accessibility constraints. The constraint insures against the removal of an excess material when the tool comes in contact with the desired surface due to the so called curvature interference and the surface interference [7,8,16].

3) The global accessibility constraints. The constraint ensures that the tool does not come in contact with either machine parts (collision detection) or unwanted parts of the desired surface [10].

Given the general context above, we tackle a particular but important problem of optimization of the rotation angles in the vicinity of stationary points of the desired surface. It should be noted that there has been a variety of research focused on the *orientation of the cutting tool*. A fairly comprehensive review on the subject is given, for instance, in [10]. However, the accuracy is also affected by *the way the orientations are being achieved*. In other words, the kinematics error depends not only on the characteristics of the surface versus the tool orientation but on the previous rotations as well. It is not hard to demonstrate that the "history" of rotations becomes in particular important in the vicinity of the stationary points of the desired surface. However, to the best of our knowledge such analysis is not provided by commercial CAD/CAM software such as Unigraphics, EdgeCam, Vericut, etc. Besides, only a few recent research papers deal with the subject. In [5] the authors analyze the sequence of rotations to minimize the number of the phase reverse steps. However, from the viewpoint of the global optimization the phase reverse does not necessarily decrease the accuracy (see our forthcoming Example 1 in section 3). A global angle adjustment procedure proposed in [2] is based on minimization of the sum $\sum_p |\mathcal{R}_p|$

which represents the length of the path in the angular space (a, b) . However, equal increments in the rotation angles do not necessarily mean equal increments in the kinematics error. Therefore, the minimizer of the functional does not necessarily minimize the error. Such is indeed the case of rough cutting when the kinematics error is large. Furthermore, the case of stationary points has not been analyzed. As a matter of fact, we can find no previous academic papers related to rough cutting combined with optimization in the vicinity of the stationary points.

Therefore, we propose a global optimization procedure to minimize the kinematics error with regard to the feasible angles (solutions of the inverse kinematics) performed in the vicinity of the stationary points. The optimization is performed within the postprocessor.

We show that such optimization increases the accuracy of the milling operations and is the most appropriate in the case of a rough cut, when the angular steps are large.

Our procedure is in particular beneficial for high speed milling characterized by small spatial steps. In this case a further decrease of the step size leads to a substantial increase in the machining time due to delays in the servo-update rate.

We present an optimization algorithm based on the iterative shortest path scheme. As opposed to [2] our discrete functional directly represents the total error. Although such minimization could be time consuming, it offers better results. Besides, we show how to reduce the computational load by pre-computing the error produced by the kinematics of a particular machine. Finally, the efficiency of the proposed scheme is verified by our simulation software [14] as well as by practical machining.

2. Optimization of the rotations in the neighborhood of stationary points

Consider a typical configuration of the five-axis milling machine with the rotary axis on the table (Fig.1). Recall that the machine is guided by axial commands carrying the 3 spatial coordinates of the tool-tip in the machine coordinate system M and the two rotation angles.

The CAM software generates a set of successive coordinates (called cutter location points or CL-points) in the workpiece coordinate system W . Typically, the CAM software distributes the CL-points along a set of curves which constitutes the so-called zigzag or spiral pattern. A postprocessing which includes a transformation into the M -system generates a set of the machine axial commands which provide the reference inputs for the servo-controllers of the milling machine. Consider how the axial command translates the centers of rotation and simultaneously rotates the W -coordinates. Let W_p and W_{p+1} be two successive spatial positions belonging to the tool path and $\mathcal{R}_p, \mathcal{R}_{p+1}$ the corresponding rotation angles. In order to calculate the tool trajectory between W_p and W_{p+1} we, first invoke the inverse kinematics [1,8] to transform the part-surface coordinates into the machine coordinates $M_p \equiv (X_p, Y_p, Z_p)$ and $M_{p+1} \equiv (X_{p+1}, Y_{p+1}, Z_{p+1})$. Second, the rotation angles $\mathcal{R} \equiv \mathcal{R}(t) = (a(t), b(t))$ and the machine coordinates $M \equiv M(t) \equiv (X(t), Y(t), Z(t))$ are assumed to change linearly between the prescribed points, namely,

$$M(t) = tM_{p+1} + (1-t)M_p,$$

$$\mathcal{R}(t) = t\mathcal{R}_{p+1} + (1-t)\mathcal{R}_p,$$

where t is the fictitious time coordinate ($0 \leq t \leq 1$). Finally, invoking the transformation from M back to W (for every t) yields $W(t) \equiv (x(t), y(t), z(t))$.

The kinematics are represented by matrix-functions $A \equiv A(a(t))$, $B \equiv B(b(t))$ associated with the rotations around the primary (the rotary table) and the secondary (the tilt table) axes respectively. Although the kinematics are specified by the structure of the machine, the resulting transformation is nothing else than successive rotations and translations designed to transport the tool to the desired point of the workpiece with the specified orientation. For instance, the machine configuration depicted in Fig.1 implies

$$M(t) = B(t)(A(t)(W(t) + R) + T) + C, \quad (2)$$

where, R , T and C 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 (Fig. 1). Although the inverse kinematics depend on the particular combination of the linear and rotational axis the proposed below optimization techniques are general. The techniques are easily modified for any particular combination of the linear and rotational axes [3,8,11].

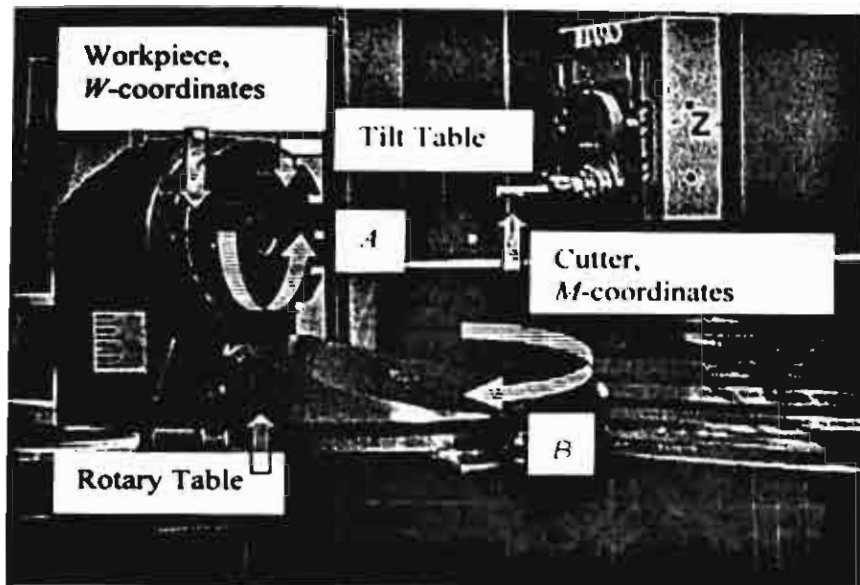
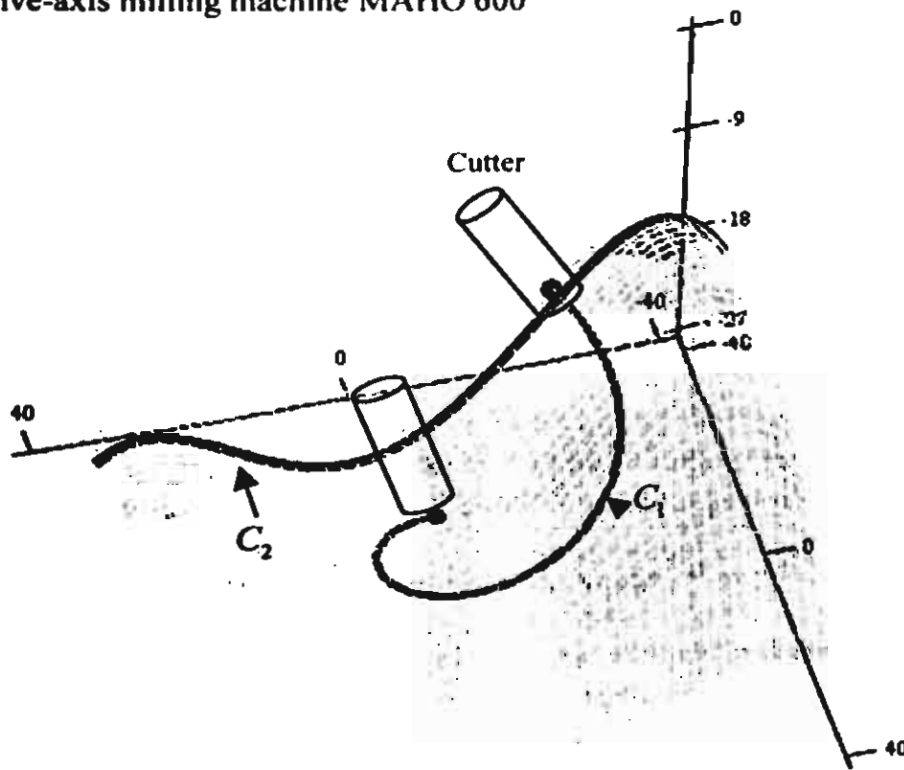


Fig.1 Five-axis milling machine MAHO 600



3.2 Non-linearity of the tool-path in the workpiece coordinates C_1 , the experimental cut C_2 (section 2).

A simple analysis of inverse kinematics (2) reveals that a linear trajectory of the tool tip in the machine coordinates may produce a non-linear trajectory in the workpiece coordinates (see Fig.2 generated by our simulation software [14]). We shall call the deviation from the non-linear trajectory the kinematics error.

Recall that a stationary point of a surface is either the maximum or the minimum point or the saddle point. Mathematically, it means that if (u_s, v_s) is the stationary point then

$$\left. \frac{\partial S}{\partial u} \right|_{(u_s, v_s)} = \left. \frac{\partial S}{\partial v} \right|_{(u_s, v_s)} = 0. \text{ Furthermore, suppose that the orientation of the tool coincides with}$$

the surface normal. Clearly, a stationary point presents a special case when the x and the y components of the normal are equal to zero. It is plain that in this case the rotation angles may jump considerably leading to unexpected deviations from the prescribed trajectory (the kinematics errors).

Note that a fine cut of a smooth surface employing small spatial and angular steps may not demonstrate the detrimental effects near the stationary points. However, a rough cut characterized by large gradients could produce considerable errors.

It is because of the sharp angular jumps that the machine produces the loop-like trajectories of the tool. Moving along such trajectories could destroy the workpiece and even lead to a collision with the machine parts. Fig.3 demonstrates such trajectories in the case of machining a single curve belonging to the surface presented in Fig.2 (the curve denoted by C_2). Fig.3 (a) shows that as opposed to a linearized version of the tool path, the real machining produces a loop-like trajectory induced by the large angular steps. We will show that such trajectories could be repaired (see Fig.3 (c)) by adjusting the rotation angles in such a way that the kinematics error is minimized.

Let us introduce minimization of the kinematics error in the framework of the kinematics of a five-axis milling machine. Without loss of generality, consider the configuration introduced in section 1.

The inverse kinematics of the machine are associated with coordinate systems denoted by $O1, O2, O2', O3$ and $O4$ shown in Fig.4. $O1$ corresponds to the workpiece coordinates, $O2$ to the center of rotation of the tilt table, $O2'$ is $O2$ rotated around y by 90 degrees. Finally, $O3$ and $O4$ correspond to the coordinates with the origin at the center of the rotating table and at the tool tip tool respectively. The machine movements prescribed by a sequence of the machine commands are executed simultaneously allowing for five degrees of freedom of the tool. We recall that the goal of the simultaneous movements of the machine parts is transporting the tool tip into position $W \equiv (x, y, z)$ and establishing orientation $\mathcal{R} \equiv (a, b)$. The angle a is between the x -axis and the projection of the orientation vector onto the x - y plane and the positive direction of the x -axis. The angle b is between the projection of the orientation vector onto the x - z plane and the positive direction of the x axis.

The above makes it possible to represent $\mathcal{R}$ in terms of the components of the orientation vector (i, j, k) . First of all, note that the tool orientation in $O4$ is $(i_4, j_4, k_4) = (0, 0, 1)$. In other words, the rotations must be performed in such a way that the tool becomes collinear with the z -axis. Observe that the resulting coordinate transformation depends on the zero position $W = (0, 0, 0)$, $\mathcal{R} = (0, 0)$ assigned by a special machine command. Without loss of generality, we assign it to the rightmost position of the tilt table shown in Fig. 4.

Rotating the orientation vector by a around z_2 in $O2$ yields

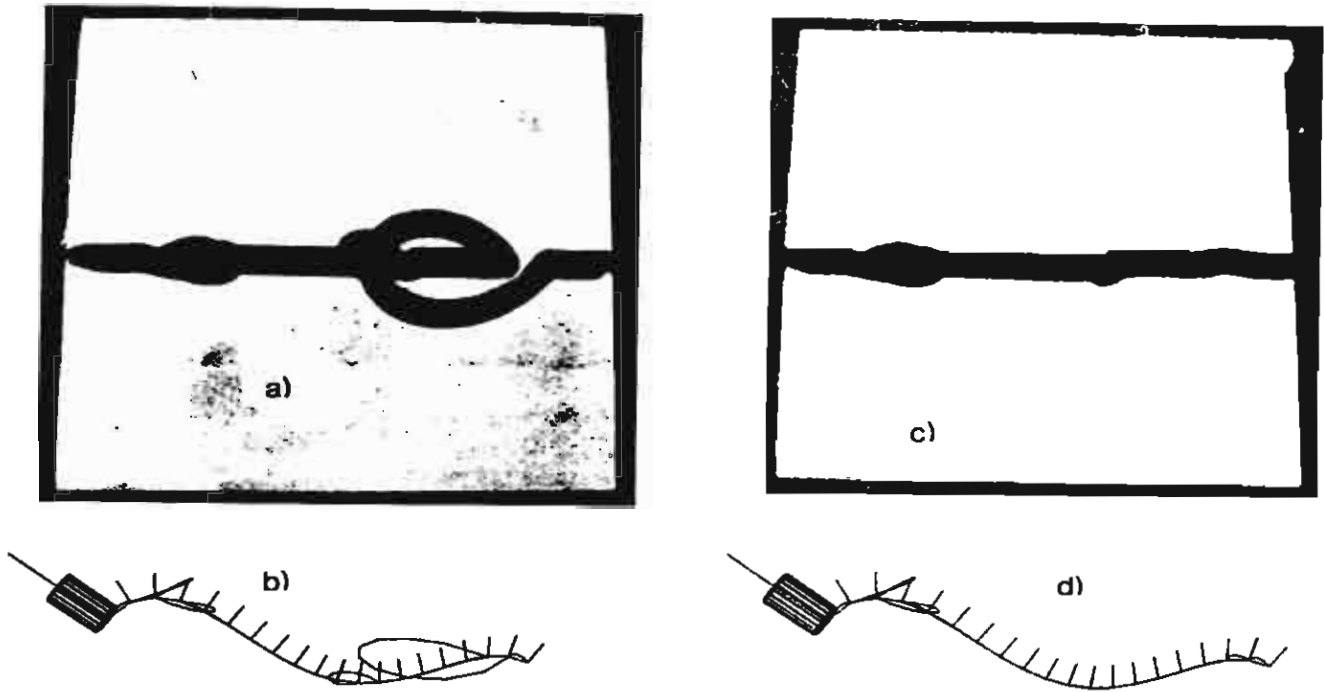


Fig 3 (a) A loop-like trajectory induced by large gradients of the rotation angles damages the workpeice, (b) the trajectory simulated by the virtual milling machine, (c) the “repaired” trajectory, (d) the repaired trajectory simulated by the virtual milling machine

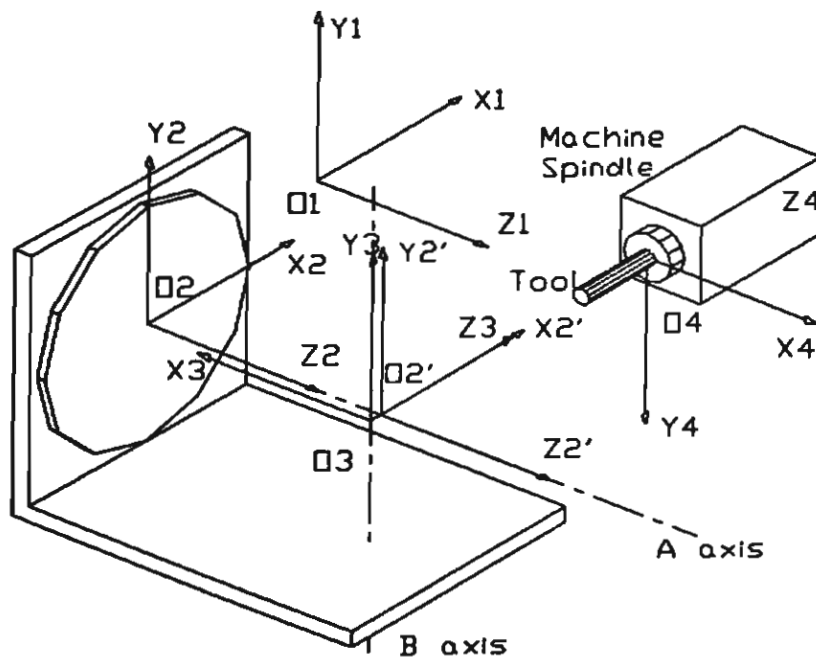


Fig 4 . Reference coordinate systems of the five-axis machine MAHO 600

$$\begin{aligned} i_2 &= i \cos(a) + j \sin(a), \\ j_2 &= -i \sin(a) + j \cos(a), \\ k_2 &= k, \end{aligned}$$

where (i_2, j_2, k_2) denotes the orientation in $O2$.

In order to align the z -axis we rotate $O2$ by 90 degrees around y_2 , obtaining

$$i_3 = -k_2, j_3 = j_2, k_3 = -i_2.$$

Next, we rotate $O3$ by b around y_3 , namely,

$$\begin{aligned} i_4 &= -i_3 \cos(b) + k_3 \sin(b), \\ k_4 &= i_3 \sin(b) + k_3 \cos(b), \\ j_4 &= j_3. \end{aligned}$$

Finally, given $(i_4, j_4, k_4) = (0, 0, 1)$ we obtain

$$\begin{aligned} 0 &= -i \sin(a) + j \cos(a), \\ 0 &= -i_3 \cos(b) + k_3 \sin(b), \\ 1 &= i_3 \sin(b) + k_3 \cos(b). \end{aligned} \tag{3}$$

Consider a solution of system (3) given by

$$a_{base} = \begin{cases} \tan^{-1}(\frac{j}{i}), & \text{if } i > 0 \text{ and } j \geq 0, \\ \pi + \tan^{-1}(\frac{j}{i}), & \text{if } i < 0, \\ 2\pi + \tan^{-1}(\frac{j}{i}), & \text{otherwise.} \end{cases}$$

$$b_{base} = -\sin^{-1}(k).$$

Consider a set of feasible solutions given by $\mathfrak{S}_a = \{a_{base}, a_{base} - 2\pi, a_{base} - \pi, a_{base} + \pi\}$ (see Fig.5). Clearly $\mathfrak{S}_a = \{a: |a_{base} - a| \leq 2\pi\}$. Moreover, after the rotations a_{base} or $a_{base} - 2\pi$ the tool is positioned at the same quadrant with the original projection of the tool, whereas the rotations $a_{base} - \pi, a_{base} + \pi$ correspond to the tool being located at a different quadrant shown in Fig. 6. Therefore, a_{base} or $a_{base} - 2\pi$ require the rotation $b_{base} = -\sin^{-1}(k)$ whereas rotations $a_{base} - \pi, a_{base} + \pi$ correspond to $b = -\pi - b_{base}$. Therefore, every tool orientation requires one of the four possible rotations given by

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

Let $W_{p,pH}(t)$, $L_{p,pH}(t)$ be the actual and the linear trajectory of the tool tip between Π_p and Π_{pH} . Consider the following minimization problem

$$\text{minimize } (\varepsilon^{\text{kinematic}}),$$

$$\mathfrak{R}_p \in \Lambda_p$$

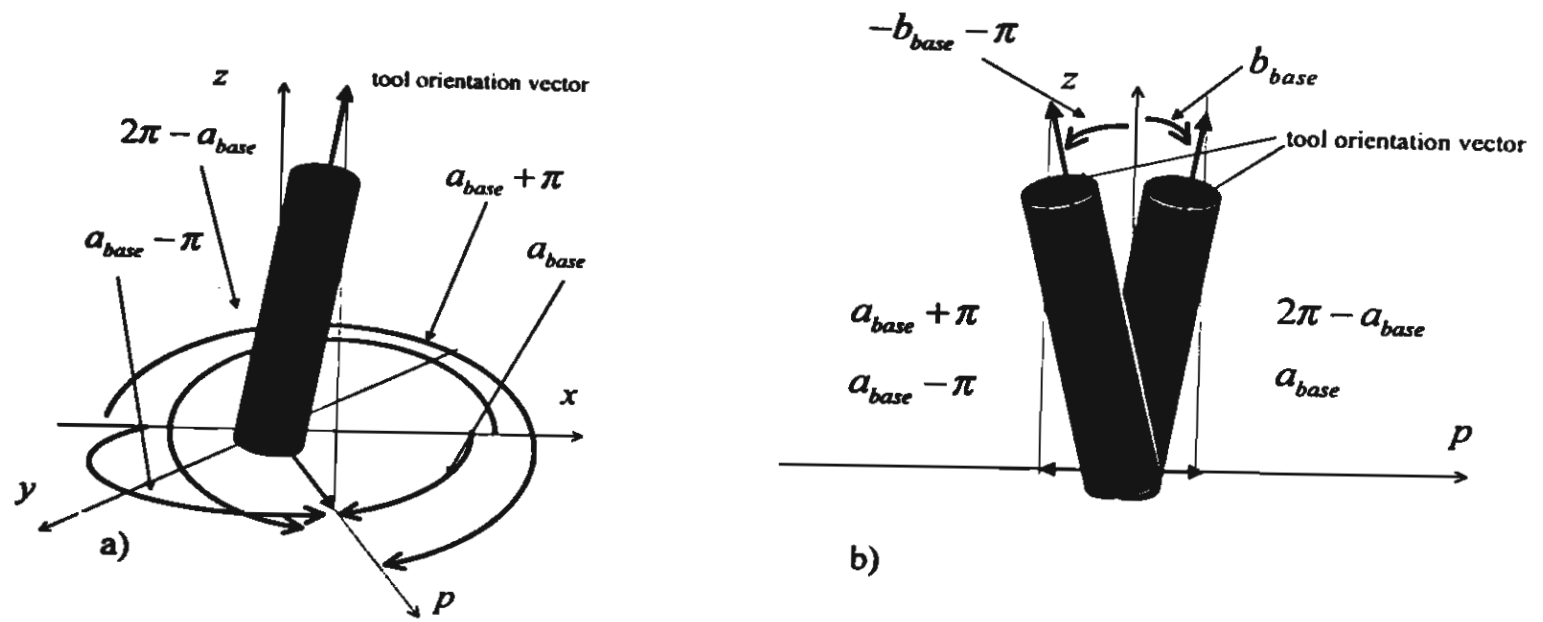


Fig.5 The set of feasible rotations

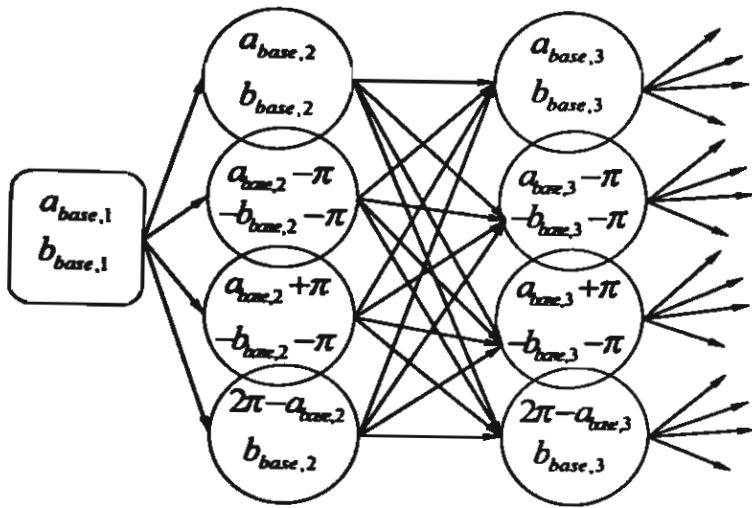


Fig.6 A graph corresponding to the set of feasible rotations

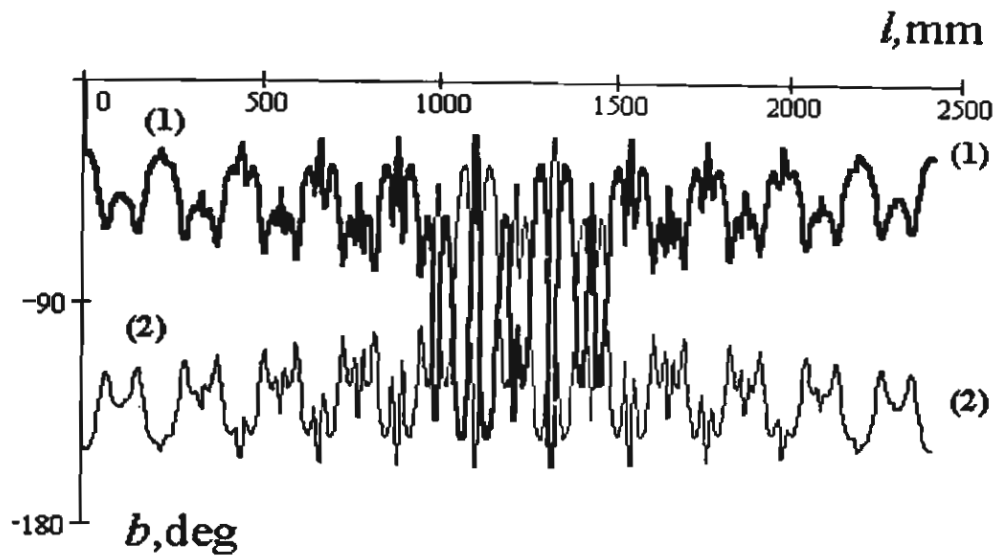


Fig.7 Trajectories corresponding to (1): b_{base} (2): $-\pi - b_{base}$. Bold: the optimized trajectory composed from (1) and (2), l is a coordinate along the tool path.

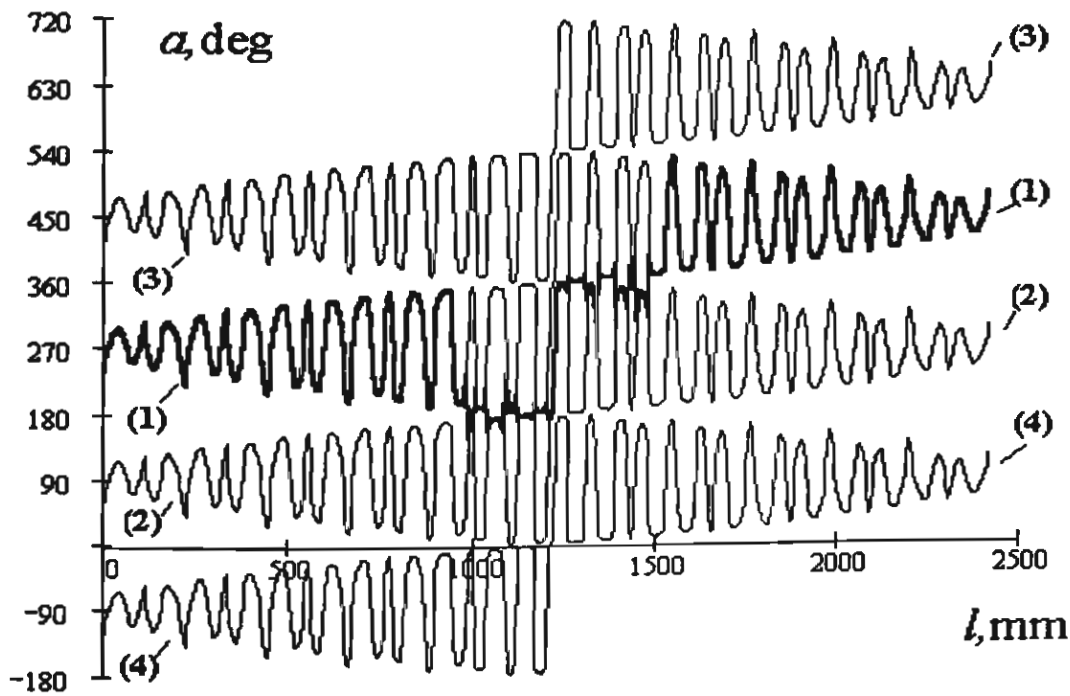


Fig.8 Trajectories corresponding to (1): a_{base} (2): $a_{base} - \pi$ (3): $a_{base} + \pi$ (4): $a_{base} - 2\pi$. Bold- the optimized trajectory composed from the trajectories (1) (2) and (4).

where the discrete functional $\varepsilon^{\text{kinematic}} = \sum_p \varepsilon_{p,p+1}^{\text{kinematic}}$ represents the total kinematics error and

$\varepsilon_{p,p+1}^{\text{kinematic}} = \|W_{p,p+1} - L_{p,p+1}\|_2$ is the kinematics error between Π_p and Π_{p+1} .

It is not hard to demonstrate that the above optimization is nothing else than the shortest path problem. Indeed, each position Π_p is characterized by the 4 graph nodes Λ_p whereas the edge between the nodes represents the kinematics error (see Fig.6). Therefore, such minimization could be performed by the conventional so-called greedy discrete algorithm [19]. We employ the classical Dijkstra's algorithm as applied to the resulting directed acyclic graph. The graph is constructed by means of the adjacency list. We use the priority queue (the binary heap) [19] to keep track of the smallest error along the path until we reach the destination. Thus, we achieve the running time of $O(E \log N)$, where E is the number of edges and N is the number of points in the vicinity of the stationary point.

The above shortest path scheme is visualized by Fig.7-8, where l denotes the distance along the tool path. As a matter of fact, the 4 options are represented by 4 trajectories in the angular space (a,b) . It is plain that the trajectories are close when b is near $-\pi/2$. In this case it is possible to change the trajectory by considering the shortest path producing minimal error.

Fig.9 shows that the optimization may only be required near the stationary points, namely, when b is close to $-\pi/2$ and a is "jumping" from 0 to π or from $\pi/2$ to $3\pi/2$ etc. Consequently, the computational load could be substantially reduced by introducing an "optimization windows" in the neighborhood of the stationary points. The size of the window is evaluated by an iterative approach. First, the user specifies an initial size of the window heuristically by analyzing the tool trajectories and the kinematics errors visualized by means of our virtual five-axis simulator [14]. Next, the procedure determines the source and the destination vertices for each line across the optimization window and performs the shortest path optimization. We increment the window and iterate until the error does not decrease. The procedure is easily generalized to the case of several stationary points invoking multiple optimization windows.

We remark that prior to the proposed optimization we perform the following correction

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

The above correction eliminates jumps exceeding π . Therefore, the shortest path routine deals only with $|a_{i+1} - a_i| \leq \pi$.

Finally, the computational load could be further decreased by pre-computing areas where the kinematics error corresponding to a prescribed average space step and to a particular rotation is always smaller than that generated by the remaining options. Since such calculations are performed only once for a particular machine, the optimization in these areas no longer invokes a procedure to calculate the kinematics error (see our forthcoming Example 1, next section).

3. Examples and practical machining

Example 1. A trajectory passing through a stationary point "across or around the hill"

Consider two successive points $\Pi_1 = (W_1, \mathcal{R}_1)$ and $\Pi_2 = (W_2, \mathcal{R}_2)$. Let the average space step $s = |W_1 - W_2|$ be 1mm and let $\mathcal{R}_1 \equiv (a_1, b_1) = (0, -85^\circ)$ and $\mathcal{R}_2 \in [120, 180] \times [-60, -80]$. It is a typical combination of the angles in the neighborhood of a stationary point (see Fig. 9).

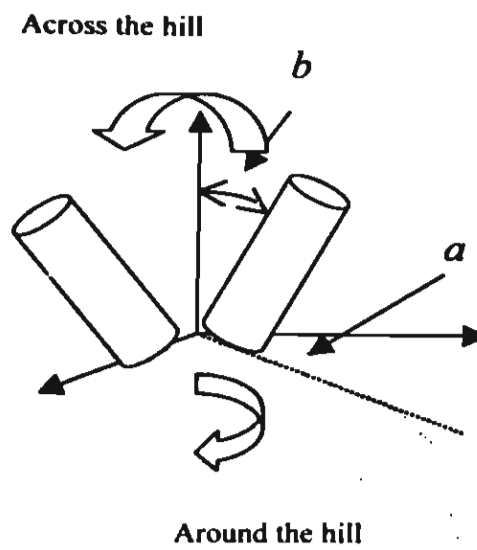


Fig.9 Around or across the hill ?

Consider a linear trajectory T and actual trajectories T_{across} , T_{around} corresponding to $\mathfrak{R}_2 \equiv (a_2, b_2) = (-10^\circ, -95^\circ)$ and $\mathfrak{R}_{2, \text{base}} \equiv (a_{2, \text{base}}, b_{2, \text{base}}) = (170, -85)$ respectively. Note that $a_2 = a_{2, \text{base}} - 180^\circ$, $b_2 = b_{2, \text{base}} - 180^\circ$. Let the distance between the midpoint of the linear trajectory and the center of the a -rotation $r_a = 10$ mm.

The kinematics error is $\varepsilon_{\text{across}}^{\text{kinematic}} = 1.49$ mm and $\varepsilon_{\text{around}}^{\text{kinematic}} = 8.89$ mm respectively. In other words moving “across the hill” by T_{across} is better than T_{around} (“around the hill”).

However, it is not always the case. A small r_a entails an alternative choice. For instance, $r_a = 0.5$ mm produces $\varepsilon_{\text{around}}^{\text{kinematic}} = 0.71$ mm, $\varepsilon_{\text{across}}^{\text{kinematic}} = 1.59$ mm. Therefore, when the cutter location point is close to the center of the a -rotations, the tool should move “around the hill” by rotating the tilt table.

Given an average space step one could evaluate the kinematics error for varying r_a . Such evaluation makes it possible to pre-compute the movements of the tool and substantially reduce the computational load spent on evaluating the error.

Fig.10 shows the kinematics error corresponding to T_{across} and T_{around} as functions of a_2 and b_2 for $r_a = 0.1$ mm, 5 mm and 20 mm respectively. Clearly, $r_a = 0.1$ mm requires T_{around} for any angle, $r_a = 20$ mm entails T_{across} whereas $r_a = 5$ mm requires a further evaluation of the kinematics error for each particular pair (a, b) .

Such pre-computing performed for a set of prescribed spatial steps characterizes an optimization strategy for a particular five-axis machine. For instance, minimization of the kinematics error of the milling machine MAHO 600 (Fig.1) for $s = 1$ mm requires T_{around} if $r_a \in [0, 5]$ mm and T_{across} if $r_a > 10$ mm. However, when $r_a \in (5, 10)$ mm the decision can not be pre-computed and shall be made by evaluating $\varepsilon^{\text{kinematic}}$ explicitly.

Example 2. The shortest path for a concave-convex surface

The example demonstrates the shortest path optimization performed in the case of a surface given

$$\text{by } S(u, v) = \begin{pmatrix} 100u - 50 \\ 100v - 50 \\ -80v(v-1)(3.55u - 14.8u^2 + 21.15u^3 - 9.9u^4) - 28 \end{pmatrix}, \quad u, v \in [0, 1]$$

(see Fig.2).

A non optimized tool path (20 x 20) CL-points characterized by the loop-like trajectories induced by large gradients of the rotation angles near the stationary points is shown in Fig. 11 a). The loops produce a considerable error. However, the optimization makes it possible to substantially increase the accuracy (see Fig. 11 b)).

Consider Table 1, which displays some results of optimization. For instance, optimization of the tool paths consisting of 400 and 800 points leads to an error reduction of about 65% and 20 % respectively.

Fig.12 shows surfaces machined with and without optimization. A non-optimized cut is characterized by a trace of the loop-like trajectory near the stationary point. The trace constitutes a serious violation of the accuracy. However, the optimized cut does not have such a defect.

It should be noted that the optimization may make sense only for the so-called rough cutting or for cuts characterized by sharp gradients. For instance, Table 1 presents the case of a rough cut when the number of the required cutter location points is not very large. Increasing the number of points along the cutting direction (see Table 1) shows that small angular steps make the

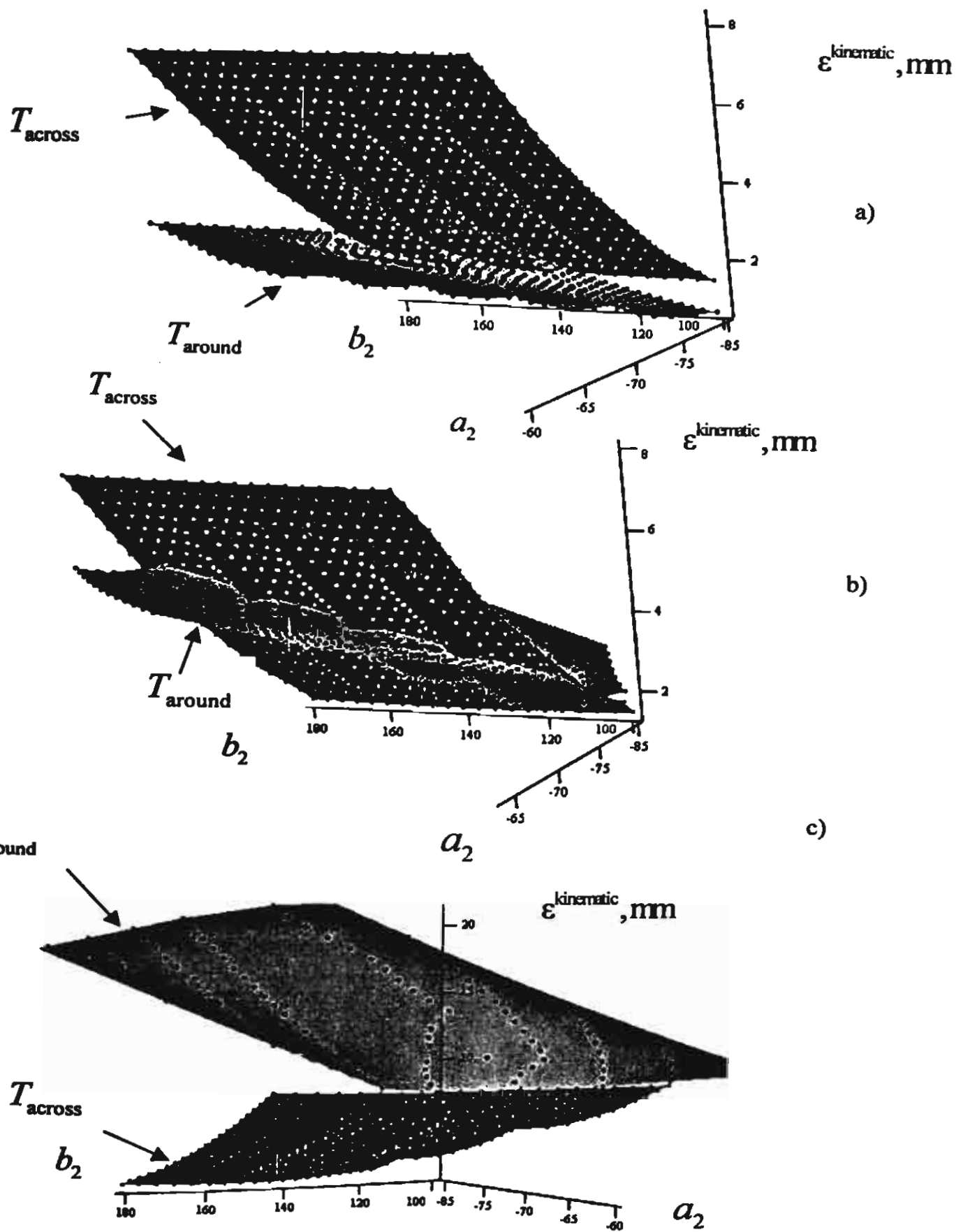


Fig. 10 The kinematic error as a function of a_2 and b_2 . The rotation radius $r_a = 0.1 \text{ mm}$ (a), 5 mm (b) and 20 mm (c).

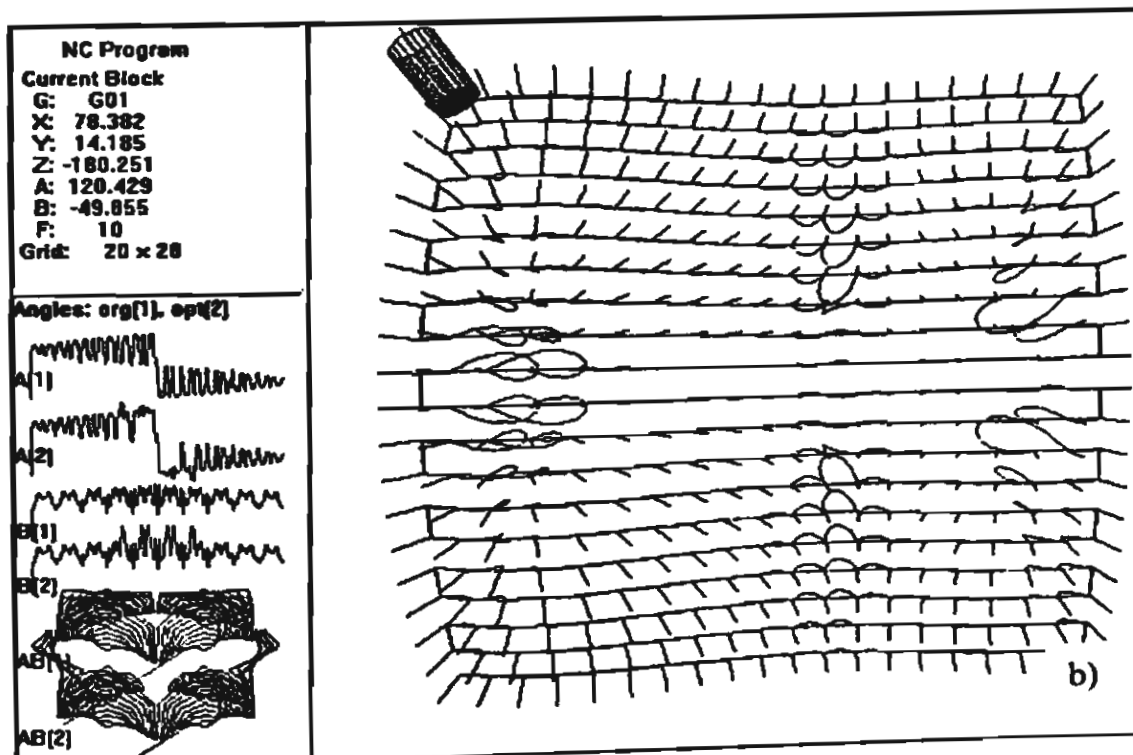
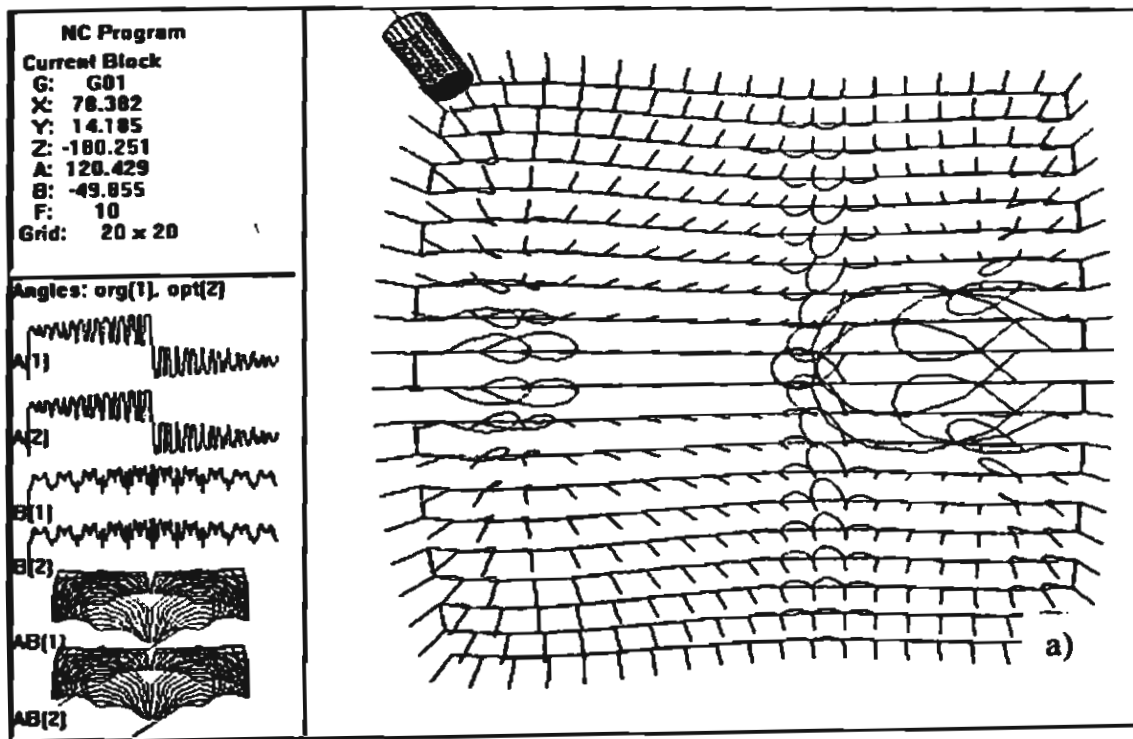


Fig. 11 The virtual milling machine. Tool trajectories before the optimization (a) and after the optimization (b).

optimization superfluous (see the path 130 x 20). When the angular step is small, switching between the feasible trajectories (see Fig.7-8) increases the step and therefore amplifies the error. Finally, it should be noted that we consider only smooth surfaces. However, in the case of sharp corners of the required surface (e.g. when one or both of the first derivatives are discontinuous) inserting additional CL points does not decrease the jump in the rotation angles. In this case the proposed method must be combined with either smoothing the angles or the surface itself near the singular points.

Example 3. Optimization in the case of tool inclination

In practice, five-axis machining requires that the tool is slightly inclined with regard to the surface normal. The inclination improves the quality of the surface and makes it possible to eliminate the so-called surface and curvature interference. If the surface is convex then a surface interference is eliminated by shifting the tool in the direction opposite to the motion vector by the tool radius (Fig.13). In the convex case the tool may be or may not be inclined. It depends on the technological characteristics of the cutting process. Usually a small lead angle of about $5-10^\circ$ is recommended [9]. However, a concave surface requires an inclination to eliminate the undercut (see Fig. 14). In this case the region characterized by large kinematics errors is shifted with regard to the stationary point in the direction of the tool movement. However, if the curvature of the machined surface is small then the inclination angle required to avoid the undercut is also small. Therefore, the optimization is required only in the neighborhood of the stationary points. However, in the case of large inclinations the algorithm must deal with regions where k (the z-component of the orientation vector) is close to 1. Table 2 and 3 show the error before and after optimization performed for the surface introduced by Example 2 when the tool is inclined by 5° and 15° lead angle respectively. Although the error decrease is no longer monotone it is of the same order. The tables clearly show that our optimization scheme is applicable to the case of an inclined tool as well. Finally, it is clear that with the kinematics error that large, errors due to curvature interference, local gouging as well as the thermal errors are negligible.

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

Number of the CL-points	No optimization error (mm) avg/max	Optimization error (mm) avg/max	The max error decrease (%)	Path length Non-opt/opt(mm)
20 x 20	0.245/15.838	0.185/5.541	65.01	2508.88/2241.74
30 x 20	0.125/12.841	0.103/5.405	57.91	2330.73/2180.67
40 x 20	0.079/6.997	0.073/5.555	20.61	2217.96/2168.54
50 x 20	0.060/9.287	0.053/4.105	55.80	2191.15/2108.12
60 x 20	0.049/10.144	0.043/4.341	57.21	2166.09/2078.01
70 x 20	0.042/8.673	0.038/5.449	31.17	2131.56/2070.28
100 x 20	0.032/5.969	0.031/5.725	4.09	2068.25/2063.25
130 x 20	0.028/3.254	0.028/3.254	0.00	2029.41/2029.41

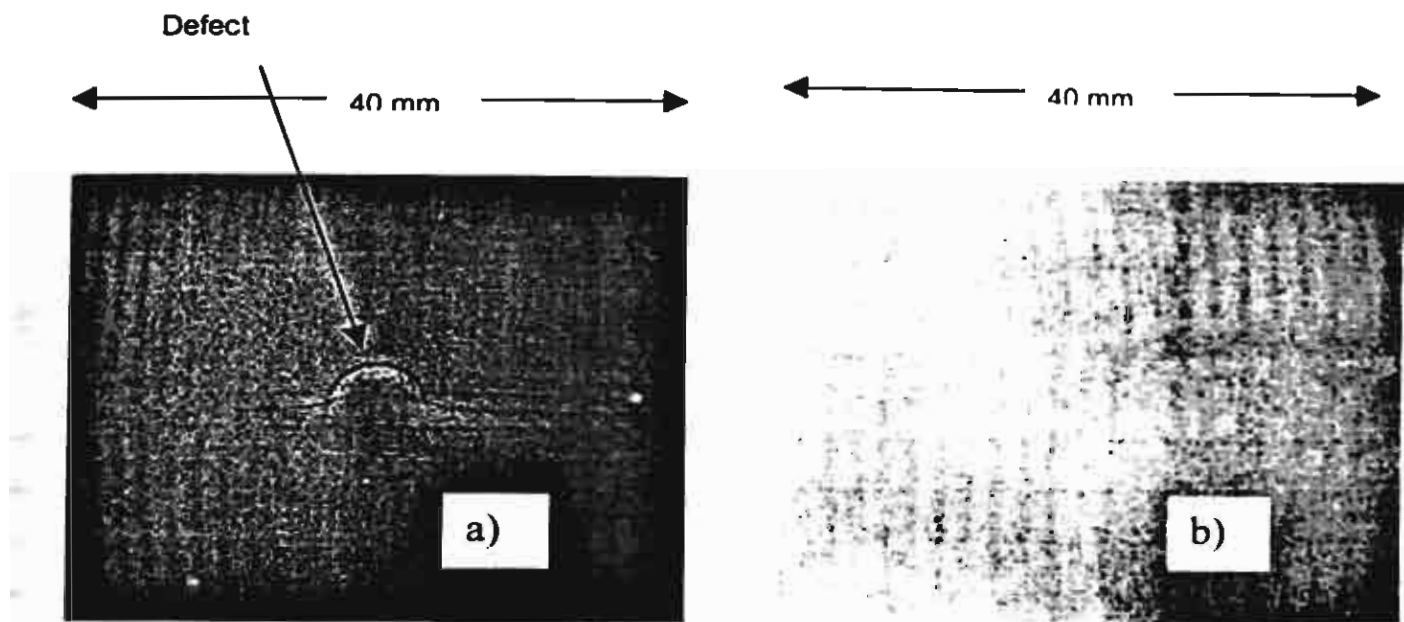


Fig.12 Vicinity of the stationary point of the experimental surface (edges enhanced) without optimization (a) and with the optimization (b).

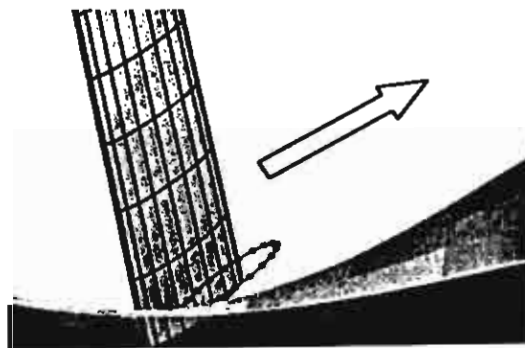


Fig. 13. Avoiding an undercut by shifting the tool along the cutting direction

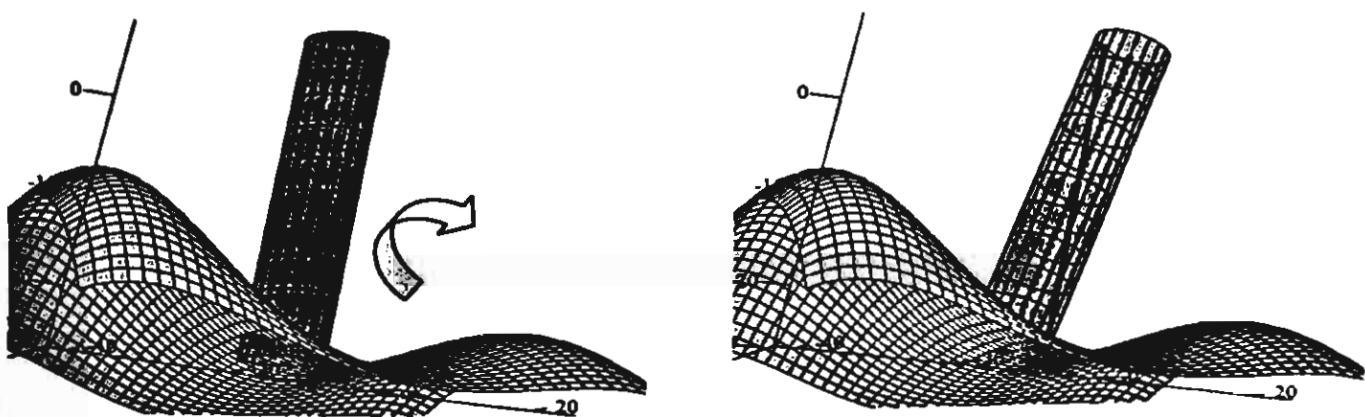


Fig. 14. Avoiding rundercut by changing the tool inclination

Table 2 Kinematics errors for the optimized and non-optimized tool path with the inclination angle 5° .

Number of the CL-points	No optimization error (mm) avg/max	Optimization error, (mm) avg/max	error decrease (%)	Path length Non-opt/opt (mm)
20 x 20	0.236/13.986	0.180/5.199	62.83	2485.82/2224.49
30 x 20	0.128/7.871	0.098/4.299	45.38	2368.56/2160.61
40 x 20	0.084/7.981	0.065/3.652	54.24	2288.54/2117.87
50 x 20	0.060/5.655	0.053/3.629	35.83	2192.93/2113.41
60 x 20	0.049/6.117	0.043/2.678	56.22	2165.54/2084.19
70 x 20	0.041/5.235	0.037/2.276	56.52	2135.49/2070.56
100 x 20	0.031/3.062	0.031/3.062	0.00	2070.54/2070.54
130 x 20	0.028/1.886	0.028/1.886	0.00	2034.90/2034.90

**Table 3 Kinematics errors for the optimized and non-optimized tool path.
The inclination angle 15° .**

Number of the CL-points	No optimization error (mm) avg/max	Optimization error (mm) avg/max	error decrease (%)	Path length non-opt/opt (mm)
20 x 20	0.249/11.174	0.194/4.569	59.11	2555.35/2286.40
30 x 20	0.129/6.293	0.121/4.374	30.49	2326.21/2265.82
40 x 20	0.091/8.515	0.084/4.471	47.49	2293.42/2218.14
50 x 20	0.071/7.906	0.065/3.430	56.62	2256.84/2179.24
60 x 20	0.057/5.402	0.056/2.840	47.43	2199.05/2174.77
70 x 20	0.050/6.537	0.048/3.383	48.29	2182.04/2149.31
100 x 20	0.039/5.051	0.038/3.247	35.72	2136.00/2123.82
130 x 20	0.034/3.941	0.034/3.941	0.00	2106.55/2106.55

4. Conclusions

A tool path thought as a trajectory in the five-dimensional space is the subject of optimization on the set of solutions of the inverse kinematics equations with regard to the required rotations. The optimization is formulated in terms of a discrete functional representing the total kinematics error along the tool-path. It is sufficient to minimize the functional only in the neighborhood of the stationary points by means of the shortest path scheme. As opposed to many optimization schemes the procedure does not require additional CL points. This is in particular important for high speed milling when an increase in the number of the points leads to a substantial increase in the machining time. Finally, the procedure could be coupled with other optimization schemes which

insert additional points or distribute the existing points in a desirable fashion. In particular, such optimization constitutes an efficient tool in the case of rough machining in the five-axis mode. The numerical experiments verified by practical machining demonstrate the accuracy increase ranging from 5 to 65 % in the case of rough cutting.

Acknowledgements

This research is sponsored by the postdoctoral scholarship awarded to Dr. M. Munlin by the Thailand Research Fund (TRF). We also wish to thank the anonymous referees and the Editor-in-Chief of the journal Prof. John Woodwark for the valuable comments.

References

1. E.L.J. Bohez, Compensating for systematic errors in five-axis NC machining, *Computer-Aided Design*, 34(2002), 391-403.
2. Cha-Soo Jun, Kyungduck Cha and Yuan-Shin- Lee, Optimizing tool orientations for 5-axis machining by configuration-space search method, *Computer Aided Design* (in press).
3. Chen-Hua She, A postprocessor based on the kinematics model for general five-axis machine tools, *Journal of Manufacturing Processes*, 2(2) (2002), 67-77.
4. J.Jeong, K.Kim, Generation of tool paths for machining free-form pockets with islands using distance maps, *Advanced Manufacturing Technology*, 15(1999), 311-316.
5. Y.H. Jung, D.W. Lee, J.S. Kim and H.S. Mok, NC post-processor for 5 axis milling machine of table-rotating/tilting type, *Materials Processing Technology*, 130-131 (2002) 641-646
6. Y.-S. Lee, Admissible tool orientation control of gouging avoidance for five-axis complex surface machining, *Computer-Aided Design*, 29(1997) 507-521.
7. K. Kim, J. Jeong , Finding feasible tool-approach directions for sculptured surface manufacture, *IIE Transactions*, 28(1) (1996) 829-836.
8. Y. Koren, Five-axis interpolators, *Annals of CIRP*, 44(1) (1995) 379-382.
9. J.-P. Kruth, P. Klewais, optimization and dynamic adaptation of the cutter inclination during five-axis milling of sculptured surfaces, *Annals of CIRP*, 43(1) 443-448.
10. B. Lawers, P. Dejonghe and J.P. Kruth, Optimal and collision free tool posture in five-axis machining through the tight integration of toll path generation and machine simulation, *Computer-Aided Design*, in press.
11. R. Lin, Y. Koren, Real time interpolator for machining ruled surfaces, *ASME Annual Meeting Proceedings*, 1994, DSC-V.55-2, pp.951-960.
12. S. S. Makhanov, An application of the grid generation techniques to optimize a tool-path of industrial milling robots, *Journal of Computational Mathematics and Mathematical Physics*, 39 (1999), 1524-1535.
13. S. S. Makhanov, D. Batanov, E. Bohez, K. Sonthipaumpoon, W. Anotaipaboon and M. Tabucanon, On the tool-path optimization of a milling robot, *Computers and Industrial Engineering*, 43(2002), 455-472.
14. M. Munlin, Virtual 5-Axis Milling Machine: Error Estimator, *Thammasat International Journal of Science and Technology*, 7(1), (2002), 40-49.
15. S.C. Park, B.K. Choi, Tool-path planning for directional parallel area milling, *Computer-Aided Design*, 32(2000) 17-25.
16. A.Rao, R. Sarma, On local gouging in five-axis sculptured surface machining using flat end tools. *CAD* 32(2000) 409-420.
17. S.-H. Suh, Y.-S. Shin, Neural network modeling for tool path planning of the rough cut in complex pocket milling, *Journal of Manufacturing Systems*, 15(5) (1996) 295-324.
18. R. Sarma, An assessment of geometric methods in trajectory synthesis for shape creating manufacturing operations, *Journal of Manufacturing Systems*, 19(2000), 59-72.
19. M. A. Weiss, Data structures and algorithm analysis in C, Addison Wesley, 1997.

New Numerical Algorithms to Optimize Cutting Operations of a 5 Axis Milling Machine

Stanislav S. Makhanov¹, Mud-Armeen Munlin¹ and Sergey A. Ivanenko²

¹ Department of Information Technology,
Sirindhorn International Institute of Technology,
Tammassat University,
Rangsit Center,
Pathum Thani, 12121, Thailand,
makhanov@siit.tu.ac.th

² Computing Center of the Russian Academy of Science,
Vavilov Str.40,
Moscow GSP-1, 117967,
Russia, *sivan@extl.ccas.ru*

Abstract. Optimization of cutting operations is an active area of research in the CNC-based manufacturing. The limited capabilities of the CAD/CAM systems require development of a new software and new numerical methods verified by practical machining. We formulate the problem of tool-path optimization in terms of interpolation of the required part surface in the curvilinear coordinate system associated with the cutter location points. Next, we present some introductory examples to demonstrate that the concept of adaptive curvilinear grid contains almost all the basic ingredients of tool-path planning, such as: adaptation to regions of large milling errors, conventional zigzag/spiral patterns and constraints related to the scallop height. Consequently, we propose a new grid based optimization procedure characterized by adaptation to a control function which depends on the rotations required to correctly position the tool. We also consider a particular but important optimization of the rotation angles near the stationary points based on the shortest path scheme. Finally, we present an application of the algorithms to tool-path planning and demonstrate the efficiency of the proposed scheme by methodological examples verified by practical machining.

Keywords: Grid generation, CNC-machines, Optimization of a tool-path, the shortest path.

1. INTRODUCTION

We represent a zigzag tool path of a five axis milling machine as a coordinate in a curvilinear coordinate system adapted to regions of the large milling errors. Such a curvilinear coordinate system is constructed numerically in the framework of adaptive grid refinement.

The problem of tool path optimization is then formulated in terms of interpolation of the required surface in the curvilinear coordinate system associated with the cutter location points.

In order to construct the required grid we introduce a variational grid generator based on minimization of the Dirichlet-type functional [3,8,9,31] subjected to constraints related to the prescribed scallop height of the machined part. The corresponding variational problem is then solved numerically by a penalty-type iterative algorithm.

We also consider a particular but important optimization of the rotation angles at the vicinity of stationary points based on the shortest path scheme. Finally, we present an application of the algorithm to tool-path planning of industrial milling robots and demonstrate the efficiency of the proposed scheme.

2. Geometric-kinematics errors in a 5-axis milling machine and tool path optimization

Innovations in the field of mechanical engineering and CAD/CAM have enhanced the involvement of milling robots in various manufacturing processes. Nowadays, computer guided milling machines are employed to produce free-shape surfaces in mass manufacturing industries (e.g., automobile, airplane, ship-building). Unfortunately, several physical phenomena, such as: machine kinematics, thermal effects, static and dynamic loading, common-cause failures often affect the quality of the desired surface. However, the particular effect of machine kinematics-geometric errors seems to be the most significant [4,7,11,15,28,29].

It should be noted that the paper does not focus on the geometric error of the *structural elements* of the milling machine, such as positioning errors of the machine axes, spindle errors, thermally induced geometric errors, etc. We will confine ourselves to the influence of the specific geometric-kinematic factor, namely the particular tool path that guides the cutting tool. In other words, we investigate the extent to which the machining tool path evaluated by the CAD/CAM computer contributes to the inaccuracy of a machined workpiece.

Consider a typical configuration of the 5-axis milling machine with the rotary axis on the table (Fig.1). The machine is guided by axial commands carrying the 3 spatial coordinates of the tool-tip in the machine coordinate system M and the two rotation angles. The supporting CAM software generates a successive set of coordinates (called cutter location points or CL-points) in the workpiece coordinate system W . Typically, the CAM software distributes the CL-points along a set of curves which constitutes the so-called zigzag or spiral pattern(Fig.2).An appropriate transformation into the M -system generates a set of the machine axial commands which provides the reference inputs for the servo-controllers of the milling robot.

Consider how the axial command translates the centers of rotation and simultaneously rotates the W -coordinates. Let $W_p \equiv (x_p, y_p, z_p)$ and $W_{p+1} \equiv (x_{p+1}, y_{p+1}, z_{p+1})$ be two successive spatial positions of the tool path and $R_p \equiv (a_p, b_p)$, $R_{p+1} \equiv (a_{p+1}, b_{p+1})$ be the corresponding rotation angles.

In order to calculate the tool trajectory between W_p and W_{p+1} we, first, invoke the inverse kinematics [15] to transform the part-surface coordinates into the machine coordinates $M_p \equiv (X_p, Y_p, Z_p)$ and $M_{p+1} \equiv (X_{p+1}, Y_{p+1}, Z_{p+1})$. Second, the rotation angles $R \equiv R(t) \equiv (a(t), b(t))$ and the machine coordinates $M \equiv M(t) \equiv (X(t), Y(t), Z(t))$ are assumed to change linearly between the prescribed points, namely,

$$M(t) = tM_{p+1} + (1-t)M_p,$$

$$R(t) = tR_{p+1} + (1-t)R_p,$$

where t is the fictitious time coordinate ($0 \leq t \leq 1$). Finally, invoking the transformation from M back to W (for every t) yields $W(t) \equiv (x(t), y(t), z(t))$.

The kinematics are represented by the matrix-functions $A \equiv A(a(t))$, $B \equiv B(b(t))$ associated with the rotations around the primary (the rotary table) and the secondary (the tilt table) axes respectively. They are specified by the structure of the machine. The machine configuration depicted in Fig. 1 implies

$$M(t) = B(t)(A(t)(W(t) + R) + T) + C,$$

where, R , T and C 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 (Fig.1). Fig.3 shows the effect of non-linearity. The non-linear trajectory between the two cutter location points in Fig. 3 is produced by our software (the virtual milling machine[23]). Although this is an actual trajectory the plot is purely illustrative. A fine industrial machining does allow for such sharp variations of the rotation angles. In

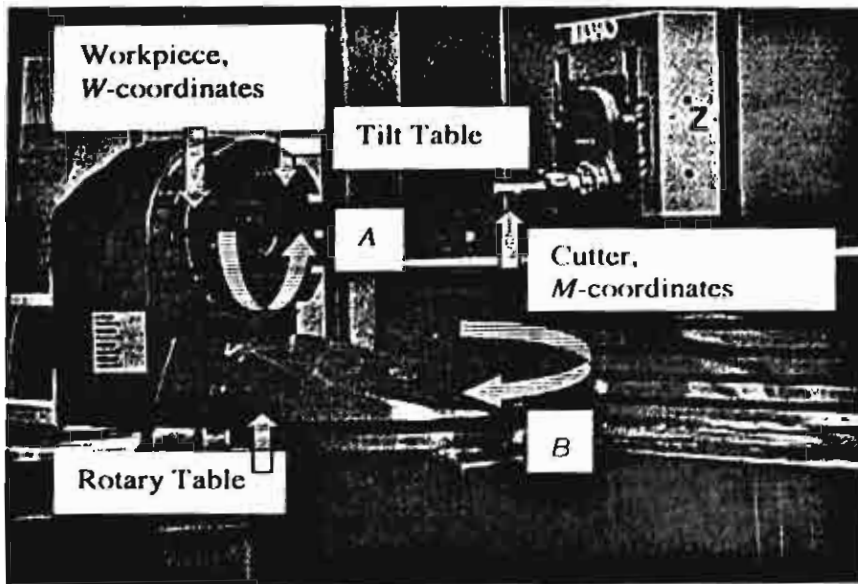


Fig.1 Five-axis milling

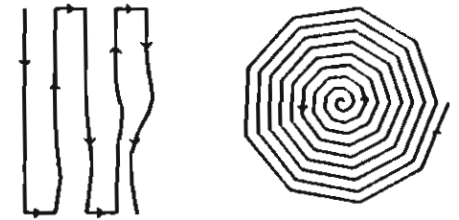


Fig.2. Zigzag and spiral tool-path

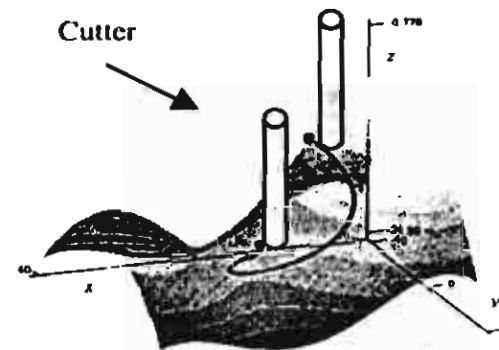


Fig.3 Simulated non-linear tool-path , machine coordinates

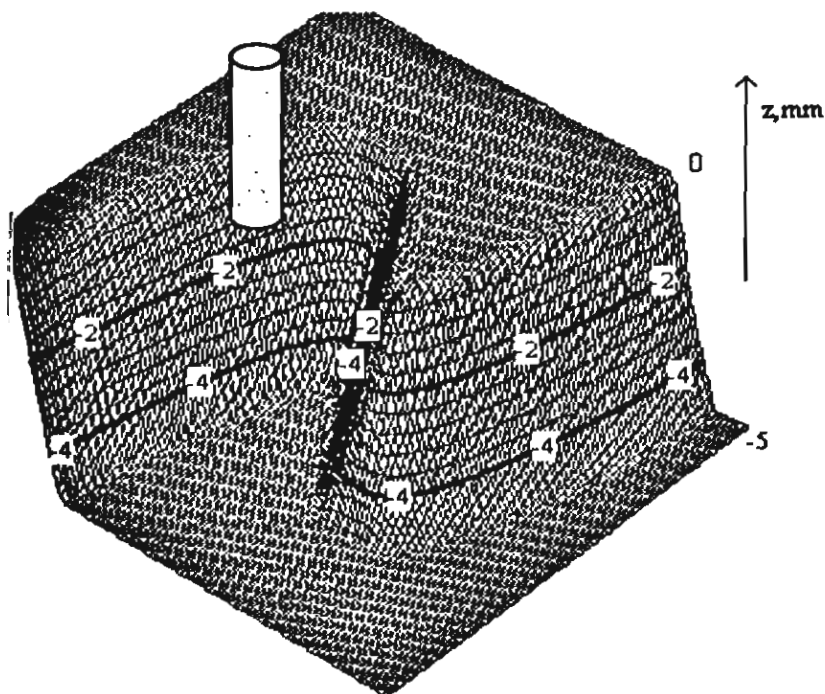


Fig. 4 The required surface.

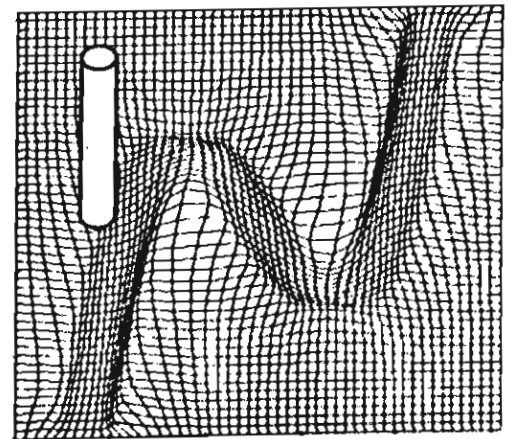


Fig.5 Curvilinear grid associated with the tool path.

order to ensure a prescribed tolerance, the CAM software will estimate the local errors and will incorporate additional cuts (if applicable) into a single output-block. However, such a strategy invokes a substantial increase of the CL-points and consequently a substantial increase of the machining time.

2. Preliminary examples of the grid based tool path optimization

Grid generation techniques are surprisingly well-adapted to tool-path optimizations. The kinematics equations[15] imply that the deviation from the linear trajectory increases with the variation of the rotation angles. In turn, the rotation angles depend on the curvature of the required surface. Therefore, the grid adapted to the regions of large gradients of the rotation angles or the large curvature may produce a better surface. Consider a surface having sharp variations along a sinus shaped curve (Fig.4). The corresponding adapted grid is depicted in Fig.5. The tool moves along the curvilinear coordinate enhancing the quality of the required surface(Fig.6). Furthermore, the grid generation techniques are applicable to generate a tool path in the case of complex boundary. Consider a complex shaped domain depicted in Fig. 7. First of all, note that such regions are not likely to often appear in the practice of conventional manufacturing. However, the authors of [6] address this domain as an example of complex pocket milling which may not be solved by means of a regular zigzag pattern. However, Fig.7 shows that the grid generation technique enables us to simultaneously generate appropriate zigzag and spiral tool paths. The grid is well adapted to the internal and external boundary. Besides, the flexibility of the grid generation approach allows adaptation to the regions where high quality milling is required.

3. Grid generation and tool path planning

This section introduces general formulations of the tool path optimization and presents a discussion regarding the assumptions made to fit the problem into the frame work of the variational gridding.

First of all, observe, that the general pattern is not necessarily a structured grid. Popular manufacturing options include space filling curves[29], evolving curves[17], distance maps[6], etc. It should be also noted that for some manufactured parts the trajectories are allowed to intersect. Therefore, the pattern may not always fit in the framework of grid generation.

Consider the required surface $S \equiv S(u, v)$ and a tool path, where $\Pi_p = (W_p, \mathcal{R}_p)$. Assume that the tool visits each position Π_p in a “grid like” fashion. Introduce a set of points $\{(u, v)_{i,j}\}$ in the parametric domain such that $\Pi = \{S(u_{i,j}, v_{i,j}), \mathcal{R}(u_{i,j}, v_{i,j})\}$, where i and j are running in such a way that Π constitutes a discrete version of a parametric curve in R^5 whereas $\{(u, v)_{i,j}\}$ a discrete analogy of a structured curvilinear grid.

Note that one could also employ unstructured grids. However the corresponding unstructured tool path requires to minimize the number of the tool retractions [24]. Therefore, such optimization presents a considerable challenge.

The model of the cutting operations has an input given by p_c, p_t, S, Π and an output given by T , where p_c denotes parameters related to the configuration of the machine, p_t the parameters related to the shape of the tool(see [25,29]). The model may or may not involve the kinematics of the machine. For instance, the majority of existing solid modeling codes are based on the linearization which does not employ the actual machine kinematics[23]. However such models do not allow for an advanced analysis of the geometric error. The optimization could be also performed with regard to p_c and p_t . However, more often than not the “optimal machine” is only a theoretical issue whereas the optimal size and shape of the tool could substantially enhance the quality of practical industrial machining.

Suppose that the configuration of the machine is given. The optimization is then performed with regard to the error $w(u, v) = \|S(u, v) - T(u, v)\|$



Fig.6 The machined part

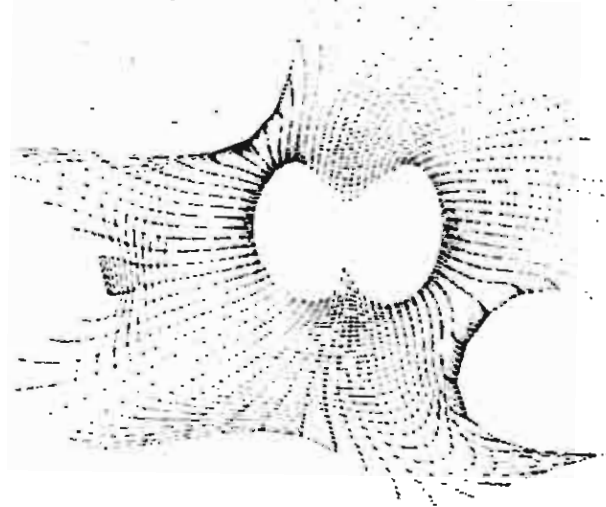
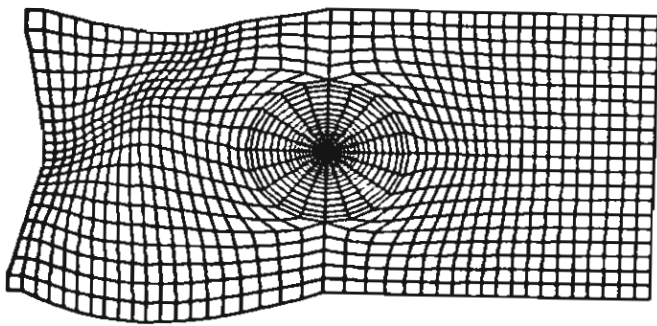
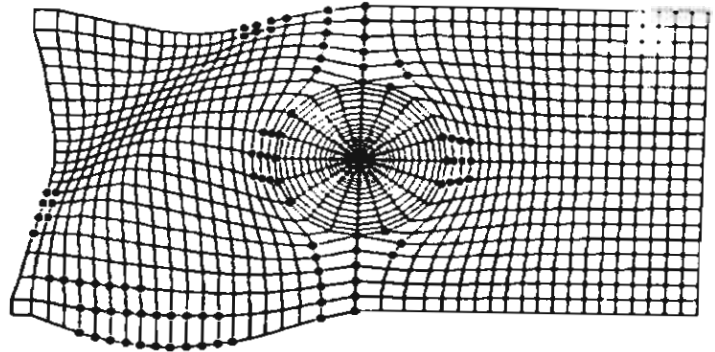


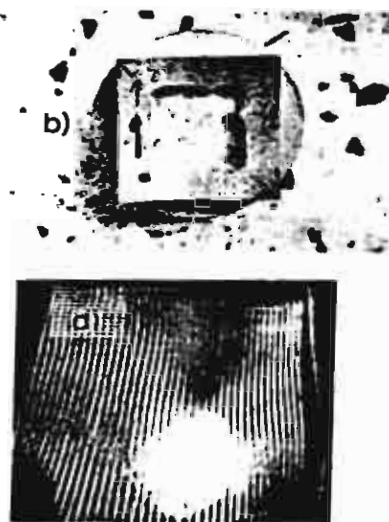
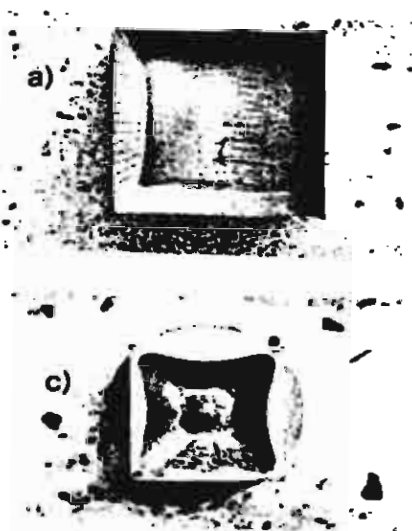
Fig.7 Complex pocket milling



**Fig.8 Composed spiral/zigzag pattern.
Constraint minimization**



**Fig.9 Composed path.
Unconstrained minimization**



**Fig. 10. Experimental machined surfaces. a), b) concave convex surface wood
c) parabolic surface, wood d) concave convex Bezier surface, steel.**

$$\text{minimize } (w), \\ \Pi, p_i$$

where $\| \cdot \|$ is an appropriate norm and where Π belongs to a set of the feasible "grid like" paths, satisfying non-linear constraints $D_k(p_i, S, \Pi) \geq 0$. A typical set of the constraints involves

- 1) The scallop height constraint (see our forthcoming section 4 where this constraint is discussed at length, see also [13,25,29])
- 2) Local accessibility constraints. The constraint insures against the removal of an excess material when the tool comes in contact with the desired surface due to the so called curvature interference and the surface interference[25]. The mathematical formulations are based on finding the intersection of the tool shape and the required surface and the inequality $r_2 \leq r_1$, where r_1 is the radius of the curvature of the surface at the contact point (in the direction orthogonal to the tool movement) and r_2 the radius of the curvature of the so-called swept section of the tool (see various formulation in [16,25,29])
- 3) Global accessibility constraints. The constraint ensures that the tool does not come in contact with either the machine parts (collision detection) or the unwanted parts of the desired surface. The constraints are often constructed by means of the so-called visibility cones (see [32]).

An extended set of constraints could include the tool acceleration constraint [24] and the tool retraction constraint [24].

Recent research papers have displayed a number of sophisticated methods to optimize a zigzag or spiral pattern (Fig.3) combined with techniques dealing with the geometric complexity of the workpiece (see for instance, [1,2,6,7,24]). Besides, there exists a variety of off-line methods to generate a suitable non-uniform tool-path, for instance: the neural network modeling approach [27] and the Voronoi diagram technique [6]. However, a robust algorithm to generate such complicated patterns is still an open problem.

As opposed to the above schemes the curvilinear grids make it possible to easily generate and optimize simple and robust zigzag or spiral patterns due to an obvious analogy of the conventional patterns with the curvilinear grids. For instance, the zigzag pattern corresponds to a curvilinear Cartesian grid whereas the spiral pattern to a polar grid. Therefore, the grid based tool path planning is suitable to be embedded into the conventional CAM-applications.

Furthermore, the grid refinement enables us to introduce the most important components of the tool-path planning, such as: adaptation to regions of large cutting errors, a *global* measure of the smoothness of the tool-path as well as engineering requirements related to scallops between the consecutive tracks. As opposed to the conventional methods dealing with the individual errors between the CL-points, the grid generation technology makes it possible to treat the problem in terms of the global optimization and to adapt all the CL-points simultaneously. Since such an adaptation is designed to find the global minima, it substantially improves the accuracy of the machined surface. Besides, the tool path optimization for the complex pocket milling (one of the most difficult machining problems (see, for instance, [7,20,24])) is solved by constructing appropriate grids adapted to the internal and the external boundaries.

Finally, nowadays powerful grid generators are able to generate grids for complex geometries independent of their type. On the face of it, constructing the tool-paths seems a simple matter. However, application of the general purpose grid generators to the tool path planning is not as straightforward as it may seem. First of all, the tool path must satisfy constraints which depend on the characteristics of the machined surface and the machining tool. Therefore, the problem must be treated as constraint optimization. Second, advanced tool path optimization involves elimination of curvature interference, surface interference, global and local accessibility, adjustment of the rotation angles, etc. Therefore, the grid generation must be invoked in the iterative loop involving a suitable preprocessing. It should be noted that an interaction between the workpiece and the tool is a complicated technological process which depends on the material, rotation of the tool, positioning errors of the

machine axes, spindle errors, thermally induced geometric errors, etc. A suitable adaptation of the grid generation methods to the variety of the errors is still an open problem.

The grid generation principles are applicable if $w \rightarrow 0$ as the area of the grid cell tends to zero. We minimize a functional F representing the equidistribution principle subjected to the above constraints. The optimization problem is then given by

minimize F

subject to the constraints $D_k \geq 0$, $k = 1, \dots, N_c$, where N_c the number of the constraints.

The functional may adapt the path either to the error or to an error estimate represented in terms of the characteristics of the required surface such as the rotation angles or the curvature.

Therefore, the grid generation allows for an optimization with regard to the space coordinates of the tool path. However, the general tool path planning requires optimization with regard to $\mathcal{R}$ as well. Furthermore, the relationship between the required tool orientation and the rotation angles $\mathcal{R}$ is not unique. Not only does this relationship depend on the configuration of the machine but it also is not unique for a particular machine. Consequently, even if the tool orientation is fixed the optimization must be performed with regard to all feasible rotation angles. Our tool path generator allows to treat the jumps of the rotation angles as well as the rotations by π and 2π by means of a special angle adjustment procedure designed for an arbitrary tool path pattern (see our forthcoming section on the optimization of the rotation angles). Note, that the initial tool orientation along the normal vector may also be replaced by the orientation based on a technological viewpoint or heuristic experimentally verified policies (see some feasible engineering solutions in [16]).

4. Adaptation to the control function

This section introduces tool path optimization based on the Dirichlet functional. The control function could be presented in terms of the milling error or in terms characteristics of the required surface. We demonstrate the techniques in the important case when the control function is represented in terms of the rotation angles.

The optimization is based on an interpolating surface comprising (by means of the inverse kinematics) the tool trajectories. Introduce a set of CL-points $\{(u, v)_{i,j}\}$, being a discrete analogy of a mapping from the "computational region" $\Delta = \{0 \leq \xi \leq 1, 0 \leq \eta \leq 1\}$ onto the physical region defined in the parametric coordinate system (u, v) . In other words, the set of the CL-points $(u, v)_{i,j}$ is a structured curvilinear grid. Next, consider trajectories between the points $(\xi_i, \eta_j), (\xi_{i+1}, \eta_j)$ and $(\xi_i, \eta_{j+1}), (\xi_{i+1}, \eta_{j+1})$, respectively denoted by $T_{i+0.5,j}(t)$ and $T_{i+0.5,j+1}(t)$. An obvious change of variables $t = (\xi - \xi_i) / (\xi_{i+1} - \xi_i)$ combined with the standard blending interpolation technique produces a subsurface $T_{i+0.5,j+0.5}(\xi, \eta)$ spanned onto the grid-cell $\{(u, v)_{i,j}, (u, v)_{i+1,j}, (u, v)_{i,j+1}, (u, v)_{i+1,j+1}\}$. The subscript ξ indicates the movement of the tool along the curves $\xi = \text{const}$. The non-linear interpolating surface $T(\xi, \eta)$ is then composed from the subsurfaces $T_{i+0.5,j+0.5}(\xi, \eta)$. Note that the obvious iterative loop "error $\Leftrightarrow$ tool-path" leads to computationally costly numerical procedures. On the other hand a suitable control function derived from the surface properties which and does not depend on the tool path is still an open problem. However, it is often the case that w is proportional to the derivatives of the rotation angles. Therefore, the tool path adapted to the control function $|a| + |b|$ may produce a better surface. The required grid is then constructed by minimizing the following Dirichlet type functional [9,10].

$$F \equiv \iint_{\Delta} \frac{(u_{\xi}^2 + u_{\eta}^2)(1 + K_u^2) + (v_{\xi}^2 + v_{\eta}^2)(1 + K_v^2) + 2K_u K_v (u_{\xi} v_{\eta} + u_{\eta} v_{\xi})}{J \sqrt{1 + K_u^2 + K_v^2}} d\xi d\eta.$$

$J = J(\xi, \eta)$ denotes the Jacobian of the mapping, $K \equiv \lambda_K k$, $k \equiv k(u, v) = |a(u, v)| + |b(u, v)|$, λ_K the weight coefficient, the subscripts ξ, η, u and v denote the partial derivatives. The functional will adapt

the CL points to the regions where $grad(k)$ is large. Obviously, k must be regularized near $a=0$ and $b=0$. Consider an approximation of F introduced in [10]. Denote the approximation by $\tilde{F}$.

The optimization procedure is endowed with constraints related to the maximum allowable scallop height denoted by h_{max} which characterizes the error between the consecutive tool-tracks. The constraint is given by $d \geq \rho$, where d is the maximum allowable distance and ρ the distance between the cutter location points. The explicit form $d=d(h_{max})$ depends on the shape of the tool and the surface curvature(see details in [4, 15,16]). We approximate the constraint by

$$D_{i,j+1/2} \equiv (d_{i,j+1/2})^2 - (\rho_{i,j+1/2})^2 \geq 0,$$

where $\rho_{i,j+1/2} = |S(u,v)_{i,j+1} - S(u,v)_{i,j}| = \sqrt{(x_{i,j+1} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2 + (z_{i,j+1} - z_{i,j})^2}$.

In order to solve the constraint minimization problem, we define the penalty function $\tilde{I} = \sum_{i,j} \lambda_{i,j} p(D_{i,j})$, and the grid-function $I \equiv \tilde{F} + \lambda_p \tilde{I}$, where $\lambda_{i,j}$ are the penalty coefficients,

λ_p the weight coefficient, $p(D)$, $D \in (-\infty, 0)$, is a convex decreasing function with $p(D) \rightarrow \infty$ if $D \rightarrow -\infty$. The relevant finite-difference scheme given by

$$\Omega_u \equiv \frac{\partial I}{\partial u_{i,j}} = 0, \quad \Omega_v \equiv \frac{\partial I}{\partial v_{i,j}} = 0$$

is quite lengthy and therefore omitted.

We solve the corresponding algebraic system by the quasi Newtonian iterations given by

$$\begin{aligned} u_{i,j}^{n+1} &= u_{i,j}^n - [\tau(\Omega_u \Omega_{vv} - \Omega_v \Omega_{uv})(\Omega_{uu} \Omega_{vv} - \Omega_{vu} \Omega_{uv})^{-1}]_{i,j} \\ v_{i,j}^{n+1} &= v_{i,j}^n - [\tau(\Omega_v \Omega_{uu} - \Omega_u \Omega_{vu})(\Omega_{uu} \Omega_{vv} - \Omega_{vu} \Omega_{uv})^{-1}]_{i,j}, \end{aligned}$$

where n is the iteration index, τ the iteration parameter, which is chosen so that the grid consists of the convex quadrilaterals. In order to construct the convex grid we control positiveness of the discrete Jacobian (see further details in [10]). The penalty coefficients are computed by an iterative procedure [20] given by

$$\lambda_{i,j}^{l+1} = \begin{cases} \lambda_{i,j}^l + \delta \lambda_{i,j}^l, & \text{if } D_{i,j} < 0, \\ \lambda_{i,j}^l, & \text{otherwise,} \end{cases}$$

where $\delta \lambda_{i,j}^l$ denotes the corresponding increment, l the number of the penalty iteration.

The initial mesh is generated by means of a marching method. Next, we compute the control function K at each node and evaluate derivatives of Ω . We repeat the iteration step to convergence updating the Lagrange coefficients $\lambda_{i,j}^l$. In order to improve the stability of the algorithm, we use linear smoothing.

Example. A spiral tool-path embedded into a zigzag tool-path.

This example demonstrates advanced techniques of producing a tool-path composed from segments corresponding to different types of motion. The tool-path on the workpiece 60×30 (Fig.8) is adapted to the curvilinear boundaries and to the zones of large milling errors located inside the circular region and at the left part of the workpiece. The constraint $h_{max}=0.01$ imposed on the scallop height is quite significant. Fig. 9 displays a grid (147 iterations) constructed by an unconstrained minimization. The grid is unacceptable since the maximum scallop height is about 50 times more than the prescribed value. The points where the constraints are not satisfied and points positioned too close to each other are indicated in the figure by circles. The composed tool-path (175 iterations) in Fig.8 is generated with the penalty function $p(D)=[\min(D,0)]^2$, $\delta \lambda=1$ and $\lambda_p=0.25$. The tool-path is well adapted to the regions adjacent to the large milling errors as well as to the constraints. The application demonstrates that our method allows to satisfy the prescribed constraints by modifying the space steps in the irrelevant regions(the regions where the error is small). Moreover, the number of the required steps

does not increase significantly. Therefore, the penalty functions technique constitutes an essential supplementary measure to improve the properties of the tool-path.

5. Practical machining

Some of our experimental surfaces are presented in Fig. 10. We verified our techniques by concave (a) and convex (b) Bezier surfaces as well as by parabolic surfaces (c) and concave-convex Bezier surfaces (d). We used workpieces made from wood (a)-(c) and steel (d). Real machining presented in Table 1 demonstrates a significant average increase in the accuracy of milling. h_c denotes the average step for the rectangular pattern, δ_A the accuracy increase, R_c , R_A denote the roughness of the workpiece produced by the conventional and the adaptive method, symbol * indicates that real machining was not performed. Observe that the constraint minimization techniques could produce a grid which actually does not decrease the error, even increase it with regard to the rectangular tool-path generated without regard to the constraints. Indeed, our formulation represents the accuracy of milling by the two criterions w and $h_{\max}$ rather than solely by w . Therefore, the constraints related to $h_{\max}$ may substantially affect the solution. Besides, the solution to the minimization problem is not unique or may not exist. In practice an initial grid typically lies outside the feasible region and there is no prior knowledge whether the set of grids satisfying the prescribed constraints comprises at least one element. Consequently, the convergence is analyzed by means of numerical experiments. Note, that the adaptive approach allows nor a simple estimate of the number of required tracks neither an estimate of the number of the CL-points belonging to one track. Therefore, we developed a realistic rule applicable to the majority of practical situations. If a number of the CL-points not satisfying the scallop constraints is more than 2/3 of the total number of points than the adaptation is not possible or requires a very large number of iterations. In other words, the computational efficiency of the algorithm depends on the number of points admissible for re-distribution. If this number is small than, first of all, the error can not be substantially reduced, secondly, the optimal grid may not exist at all. Even if such a grid exists, the iterations may fail to converge. Table 2, illustrates computational complexity of the algorithm for a zigzag tool-path having 40 x 40 CL-points constructed for the Bezier surface (steel, Fig.10 (d)). ϵ_A denotes the absolute error, the accuracy increase characterized by the ratio $\epsilon_R=100 \epsilon_A/\epsilon_C$, where ϵ_C the error on the rectangular pattern, τ_c the computational time on a PC (Pentium 3), I' the number of the required iterations, N_B the number of the "bad points"(the points where the constraint $h_{\max}=0.01$ is violated) belonging to the initial rectangular pattern, the symbol # indicates divergence. Observe that $h_{\max}=0.015$ mm leads to a substantial accuracy increase of about 40 % with regard to the rectangular tool-path, whereas $h_{\max}=0.013$ mm yields only a 22% increase. $h_{\max}=0.001$ mm and $h_{\max}=0.012$ mm actually increase the error. It should be noted that $h_{\max}=0.015$ mm seems to be the most suitable since the scallop height and the error are approximately in the same magnitude. The calculations reveal that $h_{\max}$ and ϵ varying in the same magnitude often lead to the minimum number of iterations. It means that appropriate scallop heights could be taken as fractions of ϵ corresponding to the conventional tool-path.

Table 1 : Accuracy and roughness of the surface

h_c , mm	δ_A , % / mm	R_c μm	R_A μm
3.60	34 / 0.2600	*	*
1.80	41 / 0.0930	*	*
1.20	34 / 0.0580	34.8	17.3
0.90	36 / 0.0410	14.3	6.6
0.60	32 / 0.0260	5.9	4.3
0.45	40 / 0.0180	2.6	2.1

Table 2: Convergence of the algorithm

h_{max} , mm	τ_c	I'	ε	ε_R	N_B
0.0001	#	#	#	#	1132
0.0010	32	1710	0.094	200	500
0.0012	11	550	0.047	105	92
0.0130	9	421	0.032	78	61
0.0150	4	250	0.024	59	25
0.0170	3	200	0.020	55	0

6. Optimization of the rotation angles in the neighborhood of a stationary point

The inverse kinematics of the five-axis machine involves five coordinate systems denoted by O1, O2, O2', O3 and O4 shown in Fig.11 see also Fig.1. O1 corresponds to the workpiece coordinates, O2 to the center of rotation of the tilt table, O2' is O2 rotated around y by 90 degrees. Finally O3 and O4 correspond to the coordinates with the origin at the center of the rotating table and at the tip of the tool

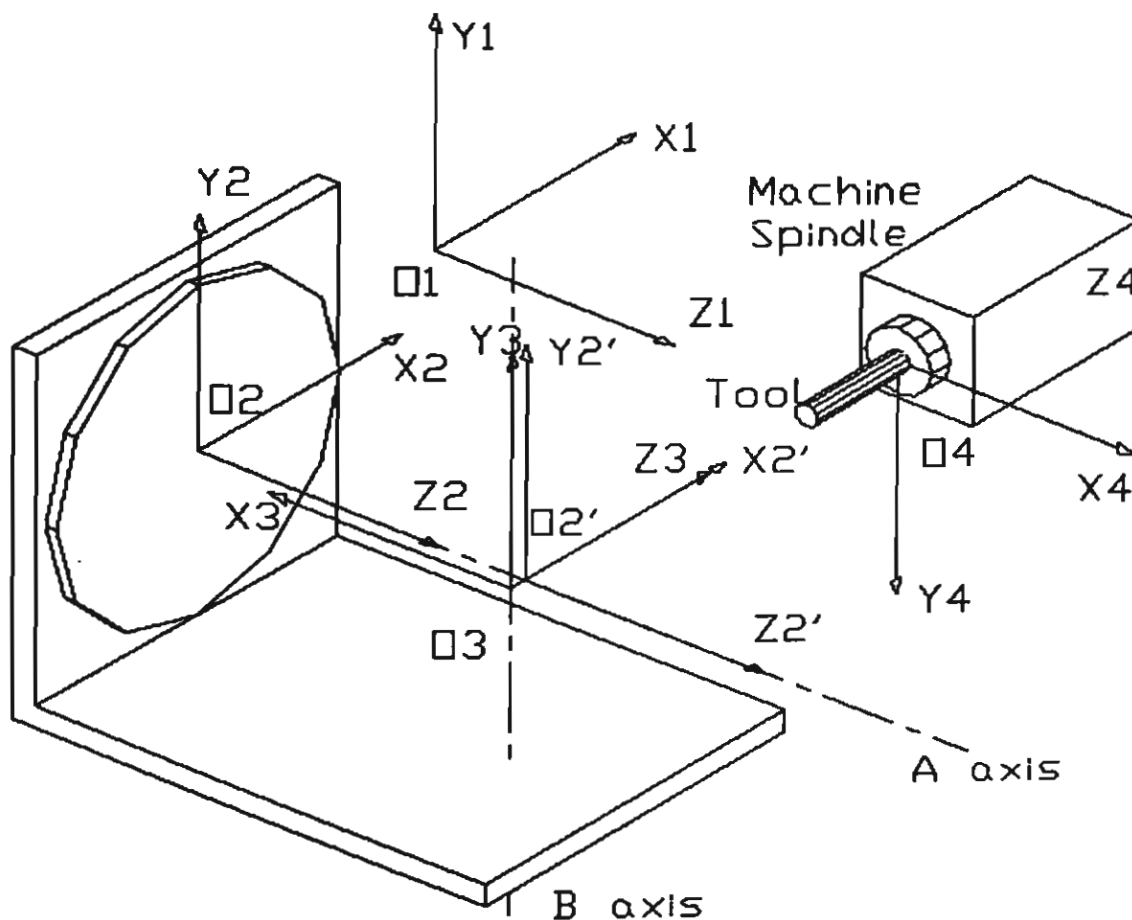


Fig. 11. Five-axis milling machine. Reference coordinate systems

respectively. The milling machine is a mechanism, which performs 1) Rotating O2 around z_2 2) Translating O3 along x_3 , 3) Translating O3 along y_3 4) Rotating O3 around y_3 and 5) Translating O4 along z_4 . The five types of movement allow for five degrees of freedom of the machine tool. The goal of the simultaneous movement of the machine is transporting the tool tip to the position $W \equiv (x, y, z)$

with the orientation (i, j, k) performed by simultaneous rotation of the machine parts specified by $\mathfrak{R}=(a,b)$. Here a is the angle between the x -axis and the projection of the orientation vector onto the x - y plane and the positive direction of the x -axis, whereas b is the angle between the projection of the orientation vector onto the x - z plane and the positive direction of the x axis. Note, that we consider $b \in [0, -\pi]$. Suppose that the required orientation of the tool is (i, j, k) . After the prescribed rotations the tool orientation in O4 is $(i_4, j_4, k_4) = (0, 0, 1)$ (the tool is collinear with the z -axis). Note that the coordinate transformation depends on the zero position $= (0, 0, 0, 0, 0)$ which is assigned by a special machine command. We demonstrate our optimization techniques for the zero position defined by the rotating table at the rightmost position shown in Fig.1. Consider the rotations. First, the coordinates are rotated by a in O2. Therefore, the orientation vector in O2 becomes

$$\begin{aligned} i_2 &= i \cos(a) + j \sin(a), \\ j_2 &= -i \sin(a) + j \cos(a), \\ k_2 &= k. \end{aligned}$$

In order to align the z -axis we rotate by O2 by 90 degrees around y_2

$$i_3 = -k_2 \quad j_3 = j_2 \quad k_3 = -i_2.$$

Next, we rotate O3 by b around y_3 , namely,

$$\begin{aligned} i_4 &= -i_3 \cos(b) + k_3 \sin(b), \\ k_4 &= i_3 \sin(b) + k_3 \cos(b), \\ j_4 &= j_3 \end{aligned}$$

Finally, $(i_4, j_4, k_4) = (0, 0, 1)$. The above equations yield

$$\begin{aligned} 0 &= -i \sin(a) + j \cos(a), \\ 0 &= -i_3 \cos(b) + k_3 \sin(b), \\ 1 &= i_3 \sin(b) + k_3 \cos(b). \end{aligned}$$

Consider the a -angle. Denote the solutions within the range $[0, 2\pi]$ by a_{base} and define a set of feasible solutions $\mathfrak{S}_a = \{a : |a_{base} - a| \leq 2\pi\}$.

Clearly,

$$a_{base} = \begin{cases} \tan^{-1}(\frac{i}{j}), & \text{if } i \geq 0 \text{ and } j \geq 0, \\ \pi + \tan^{-1}(\frac{i}{j}), & \text{if } (i < 0 \text{ and } j > 0) \text{ or } (i < 0 \text{ and } j < 0), \\ 2\pi + \tan^{-1}(\frac{i}{j}), & \text{otherwise.} \end{cases}$$

Consequently, $\mathfrak{S}_a = \{a_{base}, a_{base} - 2\pi, a_{base} - \pi, a_{base} + \pi\}$ (see Fig 12). Note that, after rotation a_{base} or $a_{base} - 2\pi$ the tool is positioned at the same quadrant with the original projection of the tool whereas rotations $a_{base} - \pi, a_{base} + \pi$ correspond to the tool being located at the different quadrant (Fig.12). Clearly, a_{base} or $a_{base} - 2\pi$ require rotation $b_{base} = -\sin^{-1}(k)$ whereas rotations $a_{base} - \pi, a_{base} + \pi$ correspond to $b = -\pi - b_{base}$. Therefore, each position $\Pi_p = (W_p, \mathfrak{R}_p)$ generates four possible "offspring" positions represented by pairs of the feasible angles given by

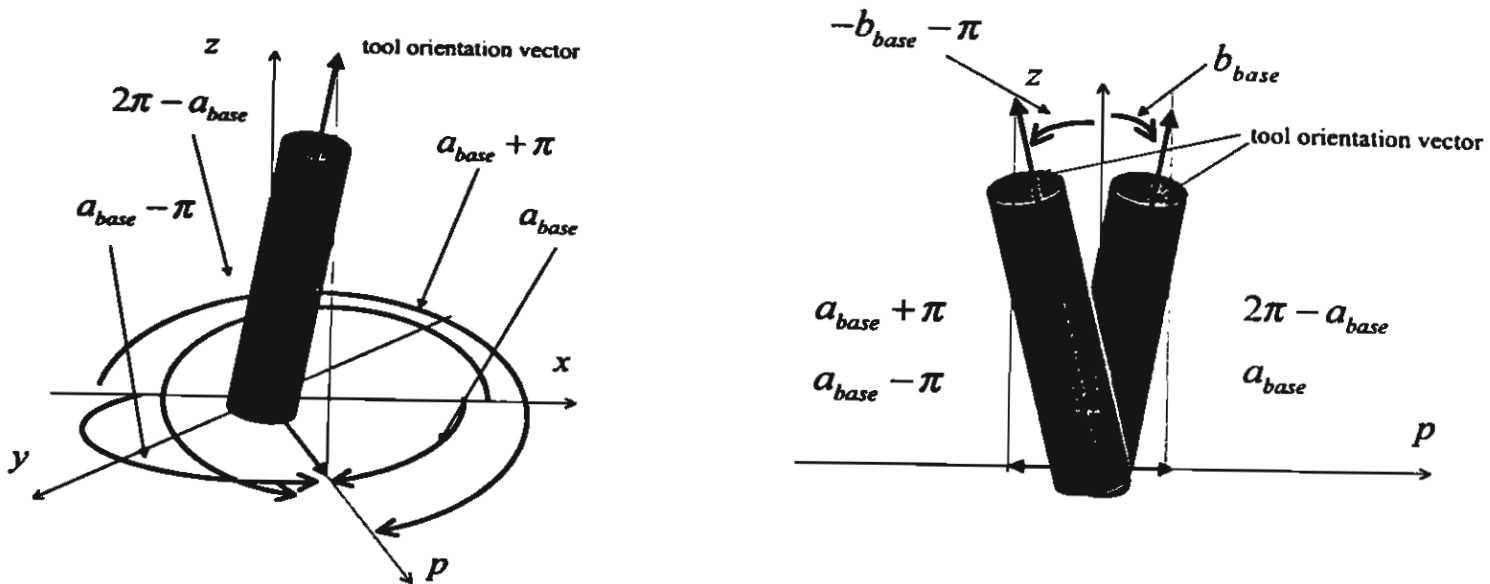


Fig.12 The set of feasible rotations

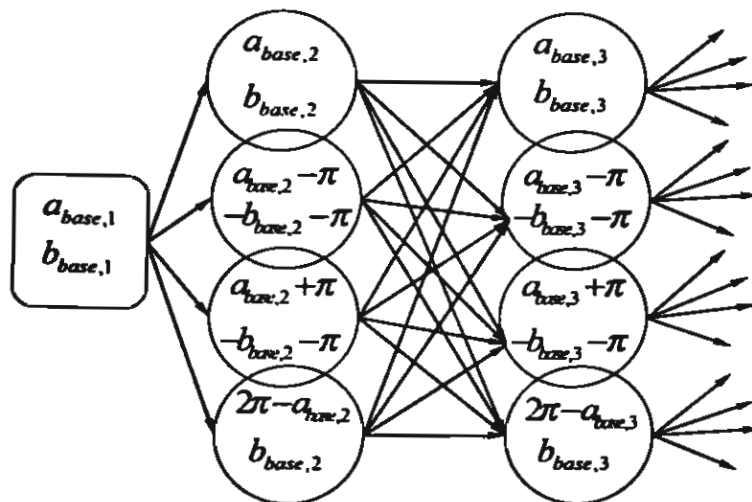


Fig.13 Optimization. The graph corresponding to the set of feasible rotations

$$\Lambda_p = \begin{cases} a_{p+1}, b_{p+1} \\ a_{p+1} - 2\pi, b_{p+1} \\ a_{p+1} - \pi, -b_{p+1} - \pi \\ a_{p+1} + \pi, -b_{p+1} - \pi. \end{cases}$$

Since the milling error depends on the derivatives the rotation angles we define the following cost functional $c \equiv c(a, b) = \int_{\Pi} |\text{grad}(\mathcal{R})| ds = \int_{\Pi} \sqrt{\left(\frac{da}{dl}\right)^2 + \left(\frac{db}{dl}\right)^2} ds$, where $l = l(s)$ is the arc length and $s \in [0, 1]$ is the parametric coordinate along the tool path Π .

We consider the following minimization problem

$$\underset{\Lambda_p}{\text{minimize}}(\tilde{c})$$

where $\tilde{c} \equiv \frac{1}{N} \sum_p \tilde{c}_p$, $\tilde{c}_p = \frac{(a_{p+1} - a_p)^2 + (b_{p+1} - b_p)^2}{(x_{p+1} - x_p)^2 + (y_{p+1} - y_p)^2 + (z_{p+1} - z_p)^2}$ and N is the total number of the

grid points. Clearly, $\tilde{c}$ is a discrete version of the continuous functional c .

The minimization problem is solved by a discrete algorithm based on the shortest path strategy. We construct a graph with nodes representing the feasible angles and arcs representing the cost $\tilde{c}_p$ (see Fig. 13). First, we define optimization windows at the vicinity of the stationary points and determine the source and the destination vertices denoted by s and t . The windows could be specified by a trial and error approach by means of our virtual five-axis simulator [23] which allows to graphically visualize manufacturing of industrial parts and to display the geometric error (see Fig.14, Fig 15 and Fig.16). Next, we construct a graph to represent the feasible paths from s to t . Finally, we apply the Dijkstra's shortest path algorithm [33] to compute the smallest distance from s to t . Since a decrease in the cost function leads to the error decrease, the optimization enhances the quality of the machined surface near the stationary points.

Example. A mold for a phone set. Fig. 15, Fig.16 and Fig.17 show the numerical and the graphical output of the simulator in the case of an experimental surface given by

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

This surface represents a simplified version of a mold for producing a phone set. Fig.16 and 17 illustrate the optimization. The non optimized tool path is characterized by loop-like trajectories induced by large gradients of the rotation angles near the two stationary points (the minimum and the maximum point). The loops produce a considerable error (see Table 3). Although the optimized path also contains the loops, they are much smaller. The corresponding accuracy increase is shown in Table 3. For instance, the optimization of the tool path consisting of 400 points leads to the 85 % (!) error reduction. It should be noted that the optimization makes sense only for the so called rough cutting, that is, the cutting characterized by low accuracy and a small number of the CL points. We prove this argument by increasing the number of points along the direction of the tool movement. Table 3 demonstrates that in the case of a rough cut the optimization technique makes it possible to substantially reduce the error. However, increasing the number of points along the cutting direction (see Table 3 the bottom line) shows that small angular steps make the optimization superfluous. It is plain that when the angular step is small, switching between the feasible trajectories increases the

Optimization windows
at the vicinity
of stationary points

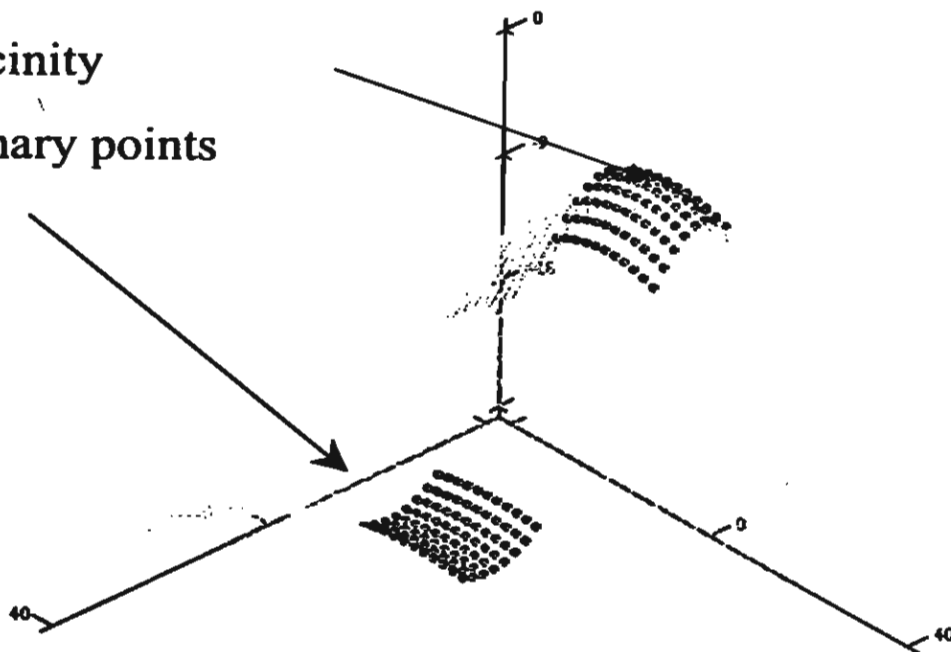


Fig. 14. Optimization windows

Data Sheets							
GCode	CL	NewGC	NewCL	IGC	NewIGC		
N	Z	A	B	F	Dist	Error	
270	-197.751	357.774	-60.582	F100	0.007	0.351	
271	-211.194	355.056	-74.530	F100	0.025	0.577	
272	-220.734	333.165	-86.854	F100	0.806	12.838	
273	-226.046	191.275	-82.870	F100	0.003	0.170	
274	-228.460	185.551	-76.327	F100	0.001	0.111	
275	-229.347	184.026	-72.450	F100	0.000	0.088	
276	-229.568	183.398	-70.797	F100	0.000	0.080	
277	-229.568	183.168	-71.004	F100	0.000	0.078	
278	-229.528	183.246	-72.826	F100	0.000	0.078	
279	-229.466	183.750	-76.086	F100	0.001	0.081	
280	-229.291	185.295	-80.613	F100	0.003	0.101	
281	-228.854	192.679	-86.143	F100	0.893	3.649	

P(u,v) :=	100u-50	Total Points	600
	100v-50	Max Dist	0.89298E
	((3.55u-14.8u ² +21.15u ³ -9.9u ⁴	Max Errors	12.8379
		ID	281
		ID	272

OK	Cancel	Apply
----	--------	-------

Fig. 15. Numerical output of the virtual milling machine

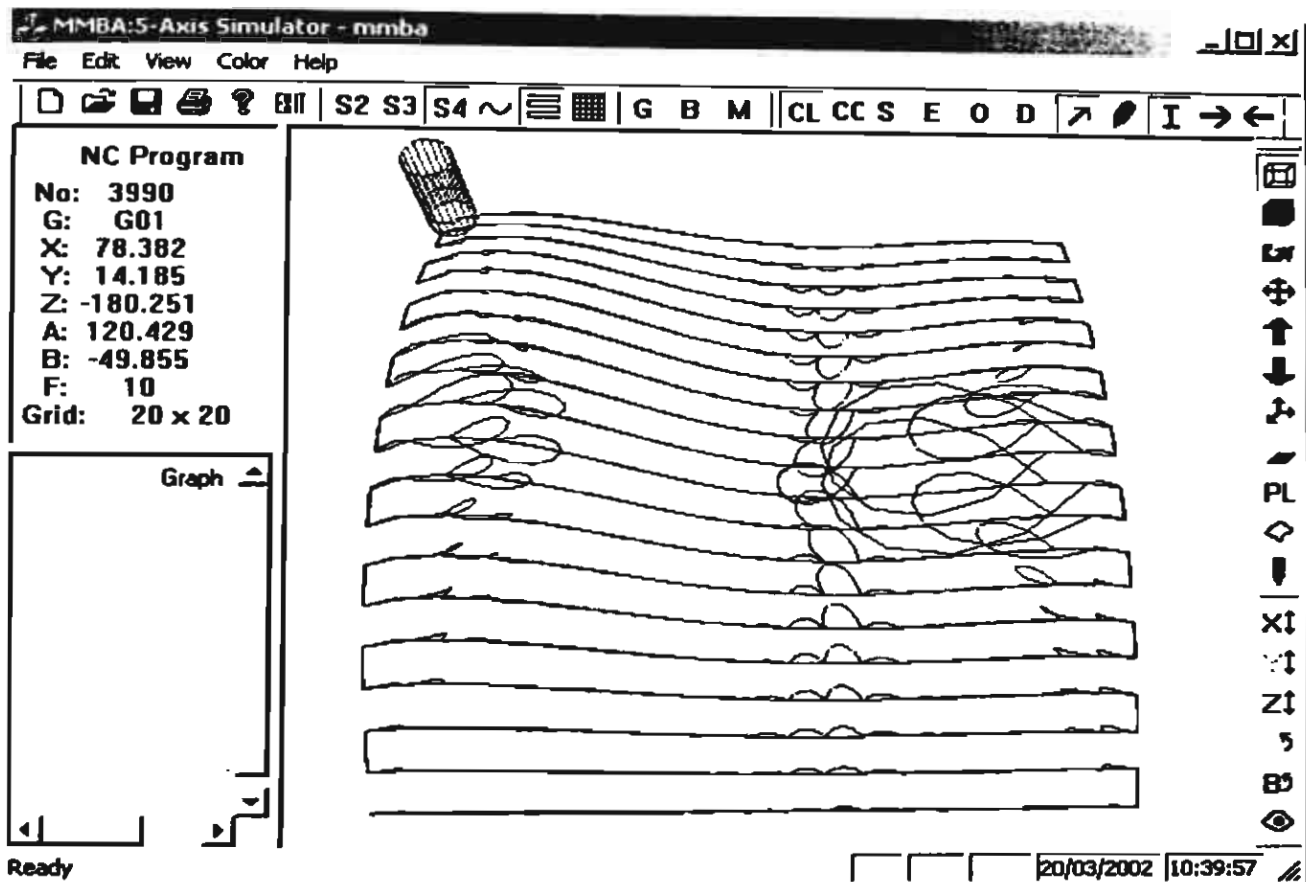


Fig. 16. Large error (loops) for the non optimized rough cut.
The Virtual 5-axis simulator.

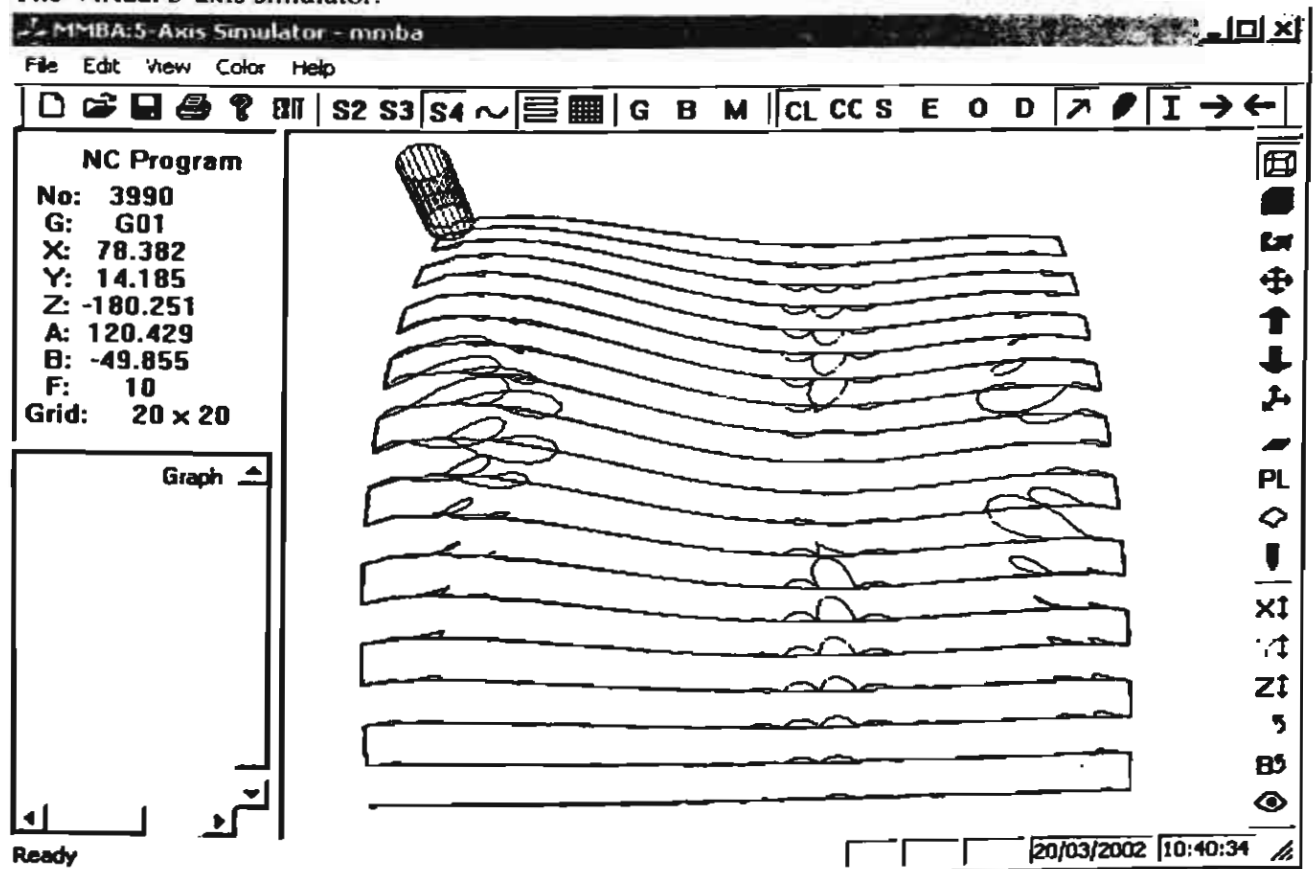


Fig. 17. Optimized rough cut. The Virtual 5-axis simulator

gradient and therefore amplifies the error. In other words if the steps are small the best strategy is the conventional scheme, that is, to stay on the initially chosen path.

Table 3. The shortest path optimization

Tool path array	Without optimization		The shortest path optimization		Accuracy increase δ_A , %
	cost $\tilde{c}$	error (mm)	cost $\tilde{c}$	error (mm)	
20 x 20	0.067	2.922	0.009	0.446	84.74
30 x 20	0.103	1.166	0.010	0.185	84.13
40 x 20	0.109	0.653	0.050	0.163	75.04
50 x 20	0.129	0.532	0.035	0.135	74.62
60 x 20	0.077	0.153	0.021	0.124	18.95
70 x 20	0.080	0.135	0.031	0.115	14.81
100 x 20	0.087	0.108	0.072	0.101	6.48
130 x 20	0.088	0.096	0.088	0.095	1.04
150 x 20	0.090	0.092	0.090	0.092	0

Conclusions

Grid generation offers a new concept to build a system of mathematical models for tool-path planning of the industrial milling robots. The approximation of the required surface by means of the associated surfaces combined with the Dirichlet type variational approach proves to be significant in increasing the accuracy. Although theoretical estimates of convergence are not still available, the proposed algorithm establishes the convergence and stability for the majority of practical cases, whereas the number of required iterations is only slightly different from that of unconstrained gridding.

The proposed grid based strategy can be enhanced by the shortest path optimization near the stationary points.

Acknowledgements

This research is sponsored by National Electronics and Computer Technology Center (NECTEC) and National Science and Technology Development Agency (NSTDA). We also acknowledge the postdoctoral scholarship awarded to Dr. M. Munlin by the Thailand Research Fund (TRF).

References

- [1] R. Ahmadi, H. Matsuo, The line segmentation problem, *Operation Research*, 100(1) (1991), 42-55.
- [2] T. Altan, B. Lilly, J.P. Kruth, W. Konig, H.K. Tonshoff, C.A. Van Luttervelt, A.B. Khairi, Advanced techniques for die and mold manufacturing, *Annals of CIPR*, 42(2)(1993), 707-716.
- [3] J.U. Brackbill, J.S. Saltzman, Adaptive Zoning for Singular Problems in Two Dimensions, *Journal of Computational Physics*, 46(3) (1982) 342-368.
- [4] J. Chou, D. Yang, on the generation of coordinate motion of five-axis CNC/CMM machines, *Journal of Engineering for Industry*, 114 (1992), 15-22.
- [5] G. Farin, *Curves and Surfaces for Computer Aided Geometric Design. A Practical Guide*, Academic Press, N.Y., 1997.
- [6] J. Jeong, K. Kim, Generation of tool paths for machining free-form pockets with islands using distance maps, *Advanced Manufacturing Technology*, 15(1999), 311-316.
- [7] Y. Huang, J.H. Oliver, Integrated simulation, error assessment, and tool path correction for five-axis NC milling, *Journal of Manufacturing*, 14(5) (1995), 331-344.

- [8] S.A. Ivanenko, Construction of curvilinear meshes for computing flows in basins, in : "Modern Problems of Computational Aerodynamics", CRC Press, London, 1992, 211-250.
- [9] S.A. Ivanenko, Adaptive grids and grids on surfaces, *Comp. Maths Math. Phys.* 33(9) (1993), 1179-1194.
- [10] S. A. Ivanenko, G. V. Muratova, Adaptive grid shallow water modeling, *Applied Numerical Mathematics*, 32(4) (2000) 447-482.
- [11] E.A. Gani, B. Lauwer, P. Klewais, J. Detand, An investigation of the surface characteristics in five-axis milling with thoroid cutters, in *Pacific Conference on Manufacturing, Proceedings*, 1994, 53-60.
- [12] Y.-S. Lee, Admissible tool orientation control of gouging avoidance for 5-axis complex surface machining, *CAD*, 29(1997) 507-521.
- [13] J.C. Lin, C.C. Tai, Accuracy optimization for mould surface profile milling, *Int. J. Adv. Manuf. Technol.*, 15(1999) 15-25 (add for scallop)
- [14] K. Kim, J. Jeong, Finding feasible tool-approach directions for sculptured surface manufacture, *IIE Transactions*, 28(1) (1996) 829-836.
- [15] Y. Koren, Five-axis interpolators, *Annals of CIRP*, 44(1) (1995) 379-382.
- [16] J.-P. Kruth, P. Klewais, optimization and dynamic adaptation of the cutter inclination during five-axis milling of sculptured surfaces, *Annals of CIRP*, 43(1) 443-448.
- [17] P. Kulkarni, D. Dutta, Deposition strategies and resulting part stiffness in layered manufacturing, *ASME Journal of Mfg Science and Engg.* 121(1) (1999) 93-103.
- [18] R. Lin, Y. Koren, Real time interpolator for machining ruled surfaces, *ASME Annual Meeting Proceedings*, 1994, DSC-V.55-2, pp.951-960.
- [19] V.D. Liseikin, The construction of structured adaptive grids-a review, *Journal of Computational Mathematics and Mathematical Physics*, 36(1) (1996) 1-32.
- [20] S. S. Makhanov, Harmonic mapping as applied to optimize the tool path of a 5-axis milling machine, *Proc. 16 International Conference on Production Research*, Prague 29 July-3 August, 2001.
- [21] S.S. Makhanov, E.J. Vanderperre, K. Sonthipaumpoon, On generation of curvilinear grids, *South-East Asian Bulletin of Mathematics*, 23 (1999) 231-241.
- [22] S. S. Makhanov, An application of the grid generation techniques to optimize a tool-path of industrial milling robots, *Comp. Maths Math. Phys.*, 39 (1999), 1524-1535.
- [23] M.-A. Munlin, S.S. Makhanov, Constraint-based simulation of a 5-axis milling machine, *International Conference on Production Research, Proceedings*, Bangkok, Thailand, August 2-4, 2000
- [24] S.C. Park, B.K. Choi, Tool-path planning for directional parallel area milling, *CAD*, 32(2000) 17-25.
- [25] A. Rao, R. Sarma, On local gouging in five-axis sculptured surface machining using flat end tools. *CAD* 32(2000) 409-420.
- [26] A.K. Srivastava, S. Veldhuis, A. Elbestawit, Modeling geometric and thermal errors in a five-axis CNC machine tool, *Int. J. Mach. Tools and Manufact.*, 35(9), (1995) 1321-1337.
- [27] S.-H. Suh, Y.-S. Shin, Neural network modeling for tool path planning of the rough cut in complex pocket milling, *Journal of Manufacturing Systems*, 15(5) (1996) 295-324.
- [28] R. Sarma, D. Dutta, Tool path generation for NC grinding, *Int. J. Mach. Tools Manuf.*, 38 (1998) 177-185.
- [29] R. Sarma, An assessment of geometric methods in trajectory synthesis for shape creating manufacturing operations, *Journal of Manufacturing Systems*, 19(2000), 59-72.
- [30] J.F. Thompson, B.K. Soni, N.P. Wetherhill Eds, *Handbook of Grid Generation*, CRC Pub, 1999
- [31] A.M. Winslow, Numerical solution of quasilinear Poisson equation on non-uniform triangle mesh, *Journal of Computational Physics*, 1(2) 1967, 149-172.
- [32] T.C. Woo, V.F. von Turkovich, Visibility map and its application to numerical control, *Annals of the CIRP*, 39(1) (1990) 451-454.
- [33] J. R. Evans and E. Mineka, *Optimization algorithms for networks and graphs*, Marcel Dekker Inc., 1992.