ANGLE CORRECTION FOR FIVE-AXIS MILLING NEAR SINGULARITIES

M. Munlin and S.S. Makhanov
School of Information and Management Technology,
Sirindhorn International Institute of Technology,
Thammasat University, Bangkadi Campus,
131 M. 5 Tiwanon Road, Bangkadi, Muang
Pathum Thani, 12000, Thailand
Tel: 66 2-501-3505 ext. 2003, 2007

Abstract: The inverse kinematics of five-axis milling machines produce large errors near stationary points of the required surface. When the tool travels cross or around the point the rotation angles may jump considerably leading to unexpected deviations from the prescribed trajectories. We propose three new algorithms to repair the trajectories by adjusting the rotation angles in such a way that the kinematics error is minimized. Given the tool orientations and the inverse kinematics of the machine, we first eliminate the jumping angles exceeding π by using the angle adjustment algorithm, leaving the jumps less than π to be further optimized. Next, we propose to apply an angle switching algorithm to compute the feasible rotations and identify an optimized sequence of rotations by the shortest path scheme. Further error reduction is accomplished by the angle insertion algorithm based on a special interpolation to obtain the required rotations near the singularity. We have verified the algorithms by two five-axis milling machines, namely, MAHO600E at the CIM Lab of Asian Institute of Technology and HERMLE UWF902H at the CIM Lab of Kasetsart University.

Keywords: Five-axis machines, inverse kinematics, tool path optimization, kinematics error, angle optimization, the shortest path.

1. Introduction

This paper presents three new angle correction algorithms to minimize the kinematics errors of a five-axis milling near the singularities. The general approach is based on a global approximation of the required surface by a virtual surface constructed from the tool trajectories. The kinematics error is defined as the difference between the required surface and the virtual surface. Due to the two rotations in five-axis milling, the tool tip trajectories are actually curves rather than straight-lines [1,2]. The errors are computed and displayed graphically and numerically through our virtual milling machine simulator (VMM) [3]. The VMM makes it possible to visually estimate the errors of a 3D tool path based on a prescribed set of the cutter location (CL) points as well as a set of the cutter contact (CC) points and the tool inclination angle. Errors, particularly in the vicinity of the large milling errors, i.e. the singular points [4] are minimized by the proposed algorithms invoked in a certain sequence, which depends on the surface and the type of the five-axis machines. The angle correction algorithms include angle adjustment, angle switching, and angle insertion presented along with examples of virtual and practical machining. The algorithms are in particular efficient for rough cuts when the variations of the rotation angles are large. Such sharp variations produce loop-like trajectories of the tool. Moving along such trajectories could destroy the workpiece and even lead to a collision with the machine parts. Such infeasible trajectories must be detected, repaired and optimized in such a way that the kinematics error is minimized.

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, which contribute to the inaccuracy of a workpiece near singularities.

2. Surface Singular Positions (Stationary Points)

Consider a typical configuration of the five-axis milling machine with the rotary axis on the table (Fig.1). The machine is guided by the 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 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. 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 machine.

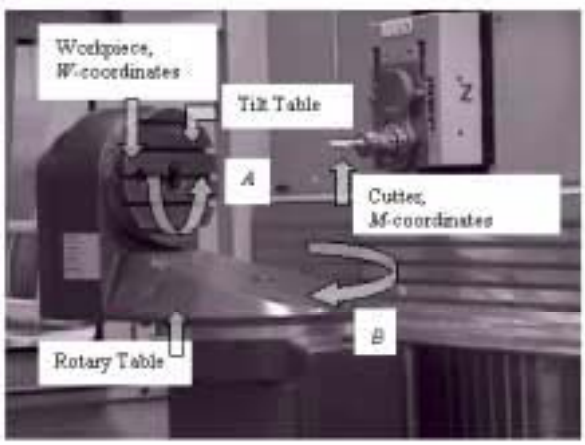

Fig.1 Five-axis milling machine MAHO 600E

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 of the tool path and $\Re_p$, $\Re_{p+1}$ the corresponding rotation angles. In order to calculate the tool trajectory between W_p and W_{p+1} we, first, invoke the inverse kinematics [5] 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 $\Re \equiv \Re(t) = (a(t), b(t))$ 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,

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

where t is the fictitious time coordinate ($0 \le t \le 1$). 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 and with the specified orientation. For instance, the machine configuration depicted in Fig. 1 implies

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

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.

A simple analysis of the inverse kinematics equation reveals that a linear trajectory of the tool tip in the machine coordinates may produce a non-linear or circular arc trajectory in the workpiece coordinates (Fig.2). We shall call the deviation from the non-linear trajectory the kinematics error.

Note that a fine cut of a smooth surface employing small spatial and angular steps may not demonstrate the detrimental effects near the singularity 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 may destroy the workpiece and even lead to a collision with the machine parts. Fig.2 demonstrates such trajectories (the curve C_1) in the case of machining a single curve belonging to the surface (curve C_2). We will show that such trajectories could be repaired by adjusting the rotation angles in such a way that the kinematics error is minimized.

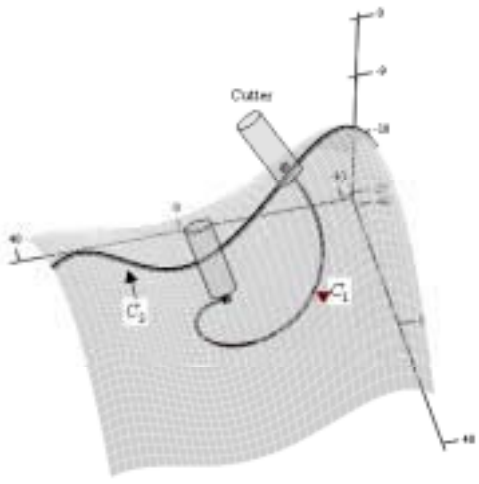

Fig.2 Non-linearity of the tool-path in the workpiece coordinates C_1, the experimental cut C_2.

Furthermore, suppose that the x or y component of the tool vector changes the sign from positive to negative or vice versa. The point where the component is zero may produce a singularity where any rotation is admissible [4,6]. Numerically it means that the rotation angles may jump considerably leading to unexpected deviations from the prescribed trajectory. In [6] the authors made an attempt to spatially avoid such a singular point. However, the computation is complex, expensive and does not preserve the original CC points. In section 3, we propose new simple and less expensive methods to cross such a singularity based on the angle switching and angle insertion principles.

3. Angle Correction Algorithms

The CAD software generates the CC points in the workpiece coordinate system whereas the CAM software generates the G-Code in the machine coordinate system from the prescribed CC points. The G-Code guides the cutting tool of the five-axis machine to travel along a nonlinear trajectory to reach the required CC point. The nonlinear trajectories constitute a trajectory surface, which is slightly different from the actual surface. The error surface is estimated by computing the difference between the actual and trajectory surfaces.

Consider a cell consisting of four CC points (see Fig.3). We apply the inverse kinematics (see section 2) to transform each point into the corresponding machine coordinates. Then, we perform the linear interpolation procedure in the machine coordinates and invoke the inverse transformation from M back to W (for every t) to obtain the tool path $W(t) \equiv (x(t), y(t), z(t))$. Finally, using a bilinear interpolation procedure yields an approximation of the machined surface $T(u,v)$. The error is then calculated by $|S(u,v)-T(u,v)|$, where $S(u,v)=(x(u,v), y(u,v), z(u,v))$ is the actual surface and $T(u,v) = (x_T(u,v), y_T(u,v), z_T(u,v))$, is the trajectory surface as shown in Fig.3. The errors are estimated and visualized graphically by the VMM [3] developed by our research group.

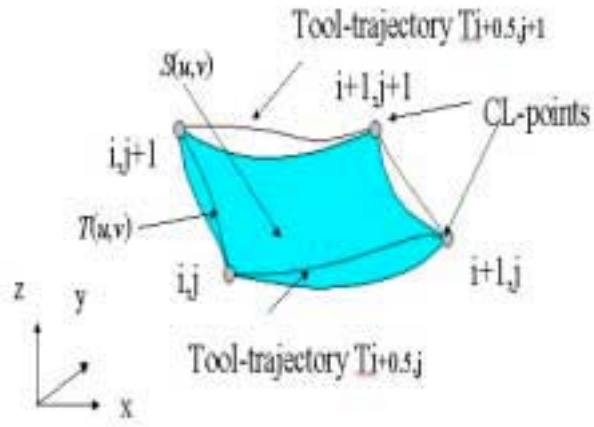

Fig.3 Error surface computation

In order minimize such errors, we propose three new angle correction algorithms namely, “angle adjustment”, “angle switching” and “angle insertion”.

3.1 Angle Adjustment

If a rotation angle jumps through 360^0 e.g. from 5 to 355, the error may increase considerably. Therefore, the angle should be adjusted in order to minimize the difference between the successive values. For instance if the angle changes from 355 to 5 the algorithm must replace 5 by 365, reducing the angular distance from –350 degree to +10 degree.

The new a-angle is given by

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

The angle adjustment eliminates the rotation angles exceeding π, leaving the remaining angles $|a_{i+1} - a_i| \le \pi$. The corrected tool path may still produce considerable kinematics errors and therefore requires the further optimizations. Note that modern five-axis machines such as UWF902H may have such an algorithm integrated into the controller. However, angle adjustment is still useful for the other machines such as MAHO600E.

3.2 Angle Switching

Let (*i,j,k*) be a normal vector. The rotation angles are then given by (see Fig.4).

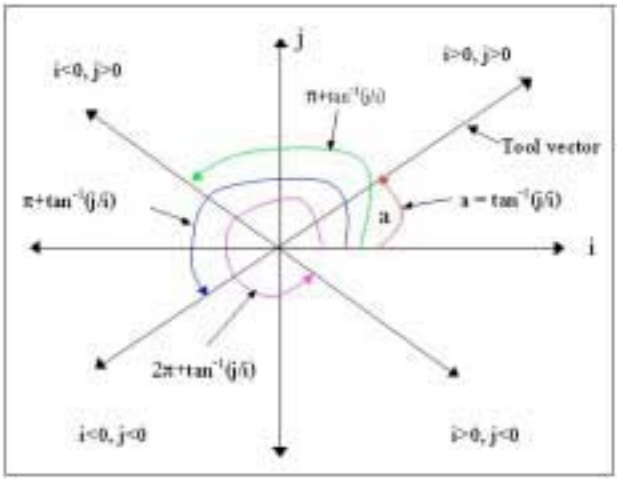

Fig.4 Computation of *a-angle* in each quadrant

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

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

Furthermore, there are four sets of the *a*-angles and the *b*-angles within the range $[0,2\pi]$ that can rotate the tool vector into the required orientation. The set of the *a*-angles is defined by $\{a_{base}, a_{base} - 2\pi, a_{base} - \pi, a_{base} + \pi\}$. After the rotation by a_{base} or $a_{base} - 2\pi$ the tool is positioned in the same quadrant with the original orientation whereas the rotation by $a_{base} - \pi, a_{base} + \pi$ corresponds to the tool being in opposite direction. The set of the feasible *b*-angle $\{b_{base}, -\pi - b_{base}\}$ transforms the tool vector into the required orientation. It is not hard to demonstrate that a_{base} or $a_{base} - 2\pi$ require the rotation $b = b_{base} = -\sin^{-1}(k)$, whereas the rotations $a_{base} - \pi, a_{base} + \pi$ correspond to another solution $b = -b_{base} - \pi$. Therefore, we have four paths represented by four pairs of feasible *(a, b)* given by

$$\Lambda = \begin{cases} a, b \\ a - 2\pi, b \\ a - \pi, -b - \pi \\ a + \pi, -b - \pi. \end{cases}$$

The angle switching selects the shortest path from the feasible sequences $\{p_i, p_{i+1}\}$ in such a way that

$$\Sigma|a_{p+1} - a_p| + \Sigma|b_{p+1} - b_p|$$

is minimized.

Note that we can also determine the best path by finding the minimum kinematics errors directly. Let $W_{p,p+1}(t)$, $L_{p,p+1}(t)$ be the actual and the linear trajectory of the tool tip between CC points *p* and *p*+1. Consider the following minimization

$$\underset{\Re_p \in \Lambda_p}{\text{minimize}} (\varepsilon^{\text{kinematic}}),$$

where the discrete functional $\varepsilon^{\text{kinematic}} = \sum_p \varepsilon^{\text{kinematic}}_{p,p+1}$ represents the total kinematics error and where $\varepsilon^{\text{kinematic}}_{p,p+1} = \| W_{p,p+1} - L_{p,p+1} \|_2$ is the kinematics error. Such minimization is performed by the so-called greedy discrete algorithm [7]. Namely, 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 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 at the vicinity of the stationary point.

The following steps describe the angle switching algorithm.

1. Locate the singular position by checking the large errors and the sign of the tool vector to determine the source (*s*) and the destination (*t*) point.
2. Construct the error graph to represent four feasible paths from *s* to *t* using the adjacency list. The graph nodes represent the vertices and the arcs represent the error between two adjacent nodes as illustrated in Fig.5.
3. Apply the Dijkstra's shortest path algorithm to compute the smallest error from *s* to *t,* from the graph constructed by step 2.
4. Update all CC points from *s* to *t* using the shortest path from step 3.

The errors displayed by the VMM (Fig.6) enable the user to locate the "problem areas" where the loops exist or the errors are greater than the tolerance. Fig.7 demonstrates the trajectories repaired by the angle switching algorithm. The practical machining for such an error curve and its repaired trajectory is illustrated in Fig8.

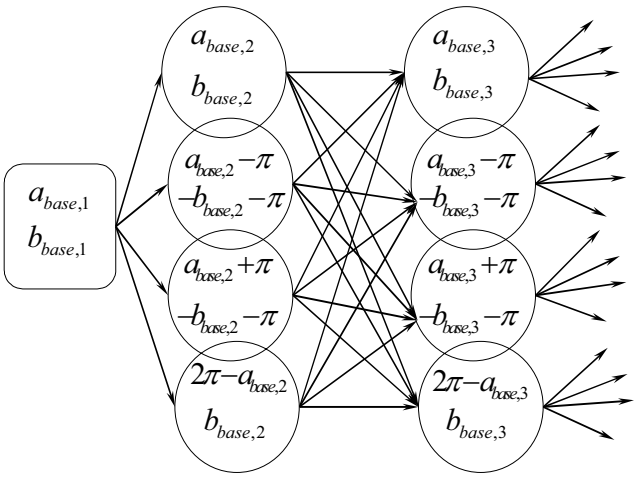

Fig.5 Graph represents four feasible paths from s to t

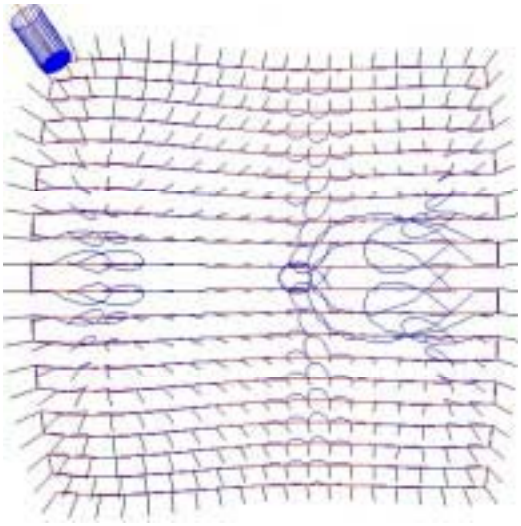

Fig.6 Original saddle surface with large errors

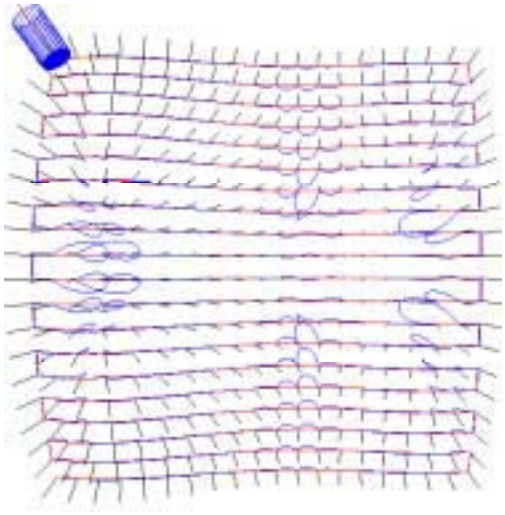

Fig.7 The repaired trajectory using angle switching

a

b

c

d

Fig.8 (a) A loop-like trajectory induced by large rotation angles damages the workpiece, (b) the trajectory simulated by the VMM, (c) the actual “repaired” trajectory, (d) the simulated repaired trajectory.

3.3 Angle Insertion (Uniform Angular Grid)

The above algorithms may still leave errors, which can be further eliminated only by inserting additional CC points. In particular when the tool path crosses the stationary point in the direction parallel (or nearly so) to the secondary rotary axes. The workpiece often must be rotated by an angle close to $\pi/2$. Therefore, further optimization is required to reduce such angular jumps. If the trajectory lies far from singularities a conventional approach to insert spatially uniform grid of the CC points between the two given positions solves the problem. However, such interpolation does not remove the jumps near the singularities. In order to solve this problem, a special interpolation called the angle insertion is proposed to partition the large angular interval into a grid having equal angular increments. Unfortunately, we can not merely split the angular interval in the machine coordinates. The resulting angles do not correspond to the actual CC points and orientations. In other words the trajectory does not follow the surface. The correct set of the CC points and the tool orientations can only be calculated from the surface equation. Therefore, we propose the following angle insertion procedure.

1. Construct a uniform grid $a_i = a' + i\Delta a$ (where Δa is the angular step) near the singularity

2. For each a_i find the CC point and the orientation that produce a_i by the bisection method as follows:

2.1 Compute a midpoint (x_m, y_m).

2.2 Invert the parametric equations $S_x(u,v) = x_m$, $S_y(u,v) = y_m$ to find the corresponding u_m, v_m and consequently z_m.

2.3 Calculate the orientation vector and obtain the corresponding a_m.

2.4 Compare a_m with the target a_i

The bisection procedure above runs until it converges within the prescribed accuracy.

Table 1 and Fig. 9 illustrate the procedure. The angular interval has been partitioned into the almost equal sub intervals. Moreover, the loops at the singularity have been significantly reduced.

Note that usually only one of the two rotation angles changes sharply near the singularity. Therefore, the bisection should be applied in only one dimension. However, theoretically there exists a possibility when the both angles must be bisected. In this case one should use either a two dimensional version of the above procedure or to bisect a linear combination of the angles.

The singular point can be located by finding the largest loop in which the tool vector changes the sign of the *x* or the *y* component.

Table 1 displays the kinematics errors and the rotation angles before and after applying the proposed angle insertion to the test surface 1 given by

$$S(u,v) = \begin{pmatrix} 100u - 50 \\ 100v - 50 \\ 30((u-0.5)^2 - (v-0.5)^2) - 6 \end{pmatrix}$$

The largest loop turns to 8 small loops (Fig.9) in which each loop is 60 times smaller or 98% error reduction.

Table 2 compares the performance of each angle correction algorithm on the test surface 2 given by

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

The most impressive results for the test surface 2 (Fig.6 and Fig.7) are obtained for MAHO600E in the case of a rough cut on 20 by 20 grid. The error has been reduced in more than 20 times after the angle switching and in more than 40 times after the angle insertion. It tells us that the error could be reduced by more than 80 times when then two methods are combined. Moreover the angle insertion works almost 10 times better than a standard CC point insertion. In the case of HERMLE UWF902H the angle switching creates another loop. Although it is 3 times smaller, it is a non-removable loop created by crossing the singular point parallel to the secondary rotary axis. The standard CC point insertion also does not have a significant impact on the error reduction (4 times smaller). It leaves two big loops, one before and another one after the singular point with error 6.104 mm each. However, the angle insertion produces an impressive result by reducing the error in almost 60 times. Finally, since the loops become smaller, the path length of the entire surface is also shorter as illustrated in Table 2.

4. Conclusions

We have developed a 3D interactive G-Code generation (post processing) and a tool path simulation software capable of generating, simulating and optimizing the tool trajectory for the five-axis milling machine. The software has been verified by surfaces having the saddle type regions, multiple extreme, steep regions with a high curvature etc. In particular, such optimization constitutes an efficient tool path in the case of rough machining in the five-axis mode. The numerical experiments demonstrate the accuracy increase ranging from 61 to 98 % depending on the surface and the configuration of the five-axis machine. The software produces impressive results and therefore can be used by the CAD/CAM industries to estimate the errors, to verify the tool path graphically and to produce and optimize the NC programs. The future work includes analysis and correction of the infeasible trajectories such as the overcut trajectories and the infeasible angles.

5. Acknowledgements

This research is sponsored by the National Electronics and Computer Technology Center (NECTEC), National Science and Technology Development Agency (NSTDA) of Thailand.

6. References

[1] E. Bohez, “Compensating for systematic errors in five-axis NC machining”, *Computer-Aided Design*, 34(2002), 391-403.

[2] S.A. Ivanenko, S.S. Makhanov and M. Munlin, “New numerical algorithms to optimize cutting operations of a five-axis milling machine*”, Applied Numerical Mathematics*, V49, Issues 3-4, June 2004, Pages 395-413.

[3] M. Munlin, “Virtual Five-Axis Milling Machine: Error Estimator”, *Thammasat International Journal of Science and Technology*, 7(1), (2002), 40-49.

[4] E. Bohez, S. S. Makhanov and K. Sonthipaumpoon, “Adaptive nonlinear tool path optimization for five-axis machining”, *International Journal of Production Research*, V38 (17), (2000), 4329-4343.

[5] Koren, Y, “Five-axis interpolators”, *Annals of CIRP*, 44(1) (1995) 379-382.

[6] Affouard, E. Duc, C. Lartigue, J.M. Langeron, and P. Bourdet, “Avoiding 5-axis singularities using tool path deformation”, *International Journal of Machine Tools and Manufacture*, V44, Issue 4, March 2004, Pages 415-425.

[7] M. A. Weiss, “Data structures and algorithm analysis in C”, *Addison Wesley*, 1997.

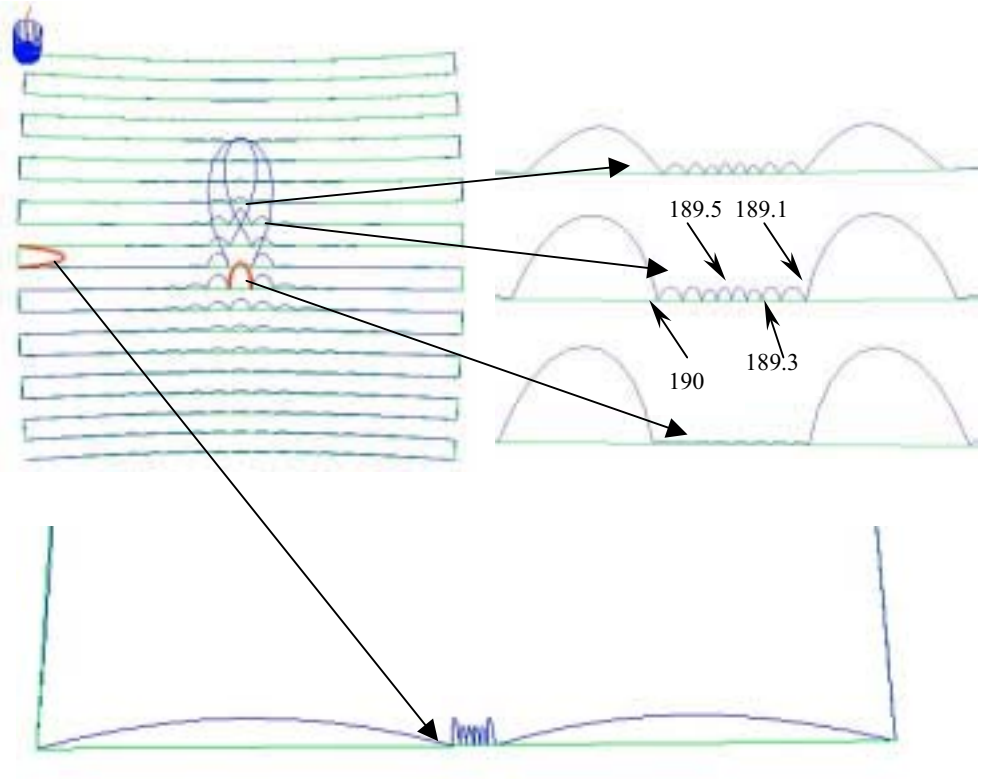

Fig.9 The original trajectory (left) and the repaired trajectory (right and bottom) using angle insertion

Table 1. Kinematics error and angular interval for the test surface 1

Test Surface1 Grid 20 by 20	G-Code Block	A-Angle (degree)	C-Angle (degree)	Kinematics errors (mm)	Error Decrease (%)	The C-angular interval (degree)
Original block 189 and 190	189	1.278807	135.018008	7.352	N/A	89.96398
	190	1.278807	224.981984	N/A	N/A	N/A
The angle insertion between the original block 189 and 190	189.1	1.278807	135.018008	0.125	98.299	11.331202
	189.2	1.086171	146.349210	0.128	98.258	11.550172
	189.3	0.975995	157.899382	0.107	98.544	10.623209
	189.4	0.922652	168.522591	0.125	98.299	11.477405
	189.5	0.904027	179.999996	0.125	98.299	11.477406
	189.6	0.922652	191.477402	0.107	98.544	10.623208
	189.7	0.975995	202.100610	0.128	98.258	11.550159
	189.8	1.086171	213.650769	0.125	98.299	11.331215
	190	1.278807	224.981984	N/A	N/A	N/A

Table 2. Performance of the proposed methods for the test surface 2

Test Surface2 Grid 20 by 20	Machine					
	MAHO600E at AIT			HERMLE UWF902H at KU		
	Max Errors (mm)	Error Decrease (%)	Path Length (mm)	Max Errors (mm)	Error Decrease (%)	Path Length (mm)
Angle Adjustment	6.388	N/A	2367.57	23.199	N/A	2697.97
CC points insertion	1.209	81.07	2317.06	6.104	73.68	2582.49
Angle insertion	0.150	97.65	2306.51	0.411	98.22	2497.03
Angle switching	0.305	95.22	2301.03	8.974	61.31	2606.26