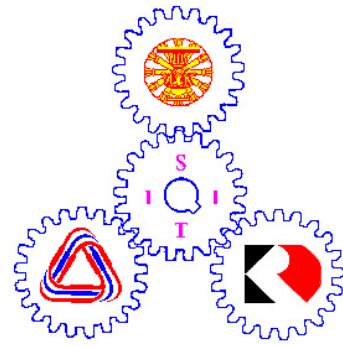




รายงานวิจัยฉบับสมบูรณ์

โครงการอัลกอริทึมใหม่เพื่อหาจุดตัดระหว่างเส้นโค้งและพื้นผิวสำหรับการลดการตัด
เกินของเครื่องแกะสลักแบบ 5 แกน

โดย ผศ. ดร. กัณฑ์ ศรีจันทร์ทองศิริ และคณะ

รายงานวิจัยฉบับสมบูรณ์

โครงการอัลกอริทึมใหม่เพื่อหาจุดตัดระหว่างเส้นโค้งและพื้นผิวสำหรับการลดการตัด
เกินของเครื่องแกะสลักแบบ 5 แกน

คณะผู้วิจัย

- ผศ. ดร. กนต์ ศรีจันทร์ทองศิริ
- ศ. ดร. สแตนนิสลอฟ มาคานอฟ

สังกัด

สถาบันเทคโนโลยีนานาชาติสิรินธร ม. ธรรมศาสตร์
สถาบันเทคโนโลยีนานาชาติสิรินธร ม. ธรรมศาสตร์

สนับสนุนโดยสำนักงานคณะกรรมการการอุดมศึกษา สำนักงานกองทุนสนับสนุนการวิจัย
และสถาบันเทคโนโลยีนานาชาติสิรินธร มหาวิทยาลัยธรรมศาสตร์

(ความเห็นในรายงานนี้เป็นของผู้วิจัย สกอ. และ สกว. ไม่จำเป็นต้องเห็นด้วยเสมอไป)

บทคัดย่อ

รหัสโครงการ: MRG5380231

ชื่อโครงการ: อัลกอริทึมใหม่เพื่อหาจุดตัดระหว่างเส้นโค้งและพื้นผิวสำหรับการลดการ
ตัดเกินของเครื่องแกะสลักแบบ 5 แกน

ชื่อนักวิจัย และสถาบัน: ผศ. ดร. กันต์ ศรีจันทร์ทองศิริ สถาบันเทคโนโลยีนาซาชาติ
สิรินธร ม. ธรรมศาสตร์

อีเมล: gun@siit.tu.ac.th

ระยะเวลาโครงการ: 15 มิถุนายน พ.ศ. 2553 ถึง 14 กุมภาพันธ์ พ.ศ. 2556

บทคัดย่อ:

โครงการนี้เริ่มต้นจากการหาอัลกอริทึมใหม่เพื่อหาจุดตัดระหว่างเส้นโค้งกับพื้นผิว เพื่อนำมาใช้ลดการตัดเกินของเครื่องโมแบบ 5 แกน แต่ภายหลังได้มีการเปลี่ยนแนวคิดเป็นการใช้ความต่างของมุมของการหมุนของจุด CL เพื่อลดค่าความผิดพลาดทางคินาเมติกแทน เนื่องด้วยแนวคิดที่เสนอตอนแรกพบว่าใช้เวลาในการคำนวณนานเกินไป ไม่เหมาะกับการนำมาใช้กับการวางแผนการโมกับเครื่อง 5 แกน

อย่างไรก็ดี แนวคิดใหม่เป็นที่ประสบผลสำเร็จดี เราเสนอการใช้ค่าความต่างของมุมของการหมุนแทนการใช้ค่าความผิดพลาดที่แท้จริงในการวางแผนการโม เนื่องด้วยค่าที่แท้จริงนั้นซับซ้อนและใช้เวลาในการคำนวณสูงมาก ทำให้ขั้นตอนการวางแผนใช้เวลาสูง ในขณะเดียวกัน ค่าความต่างของมุมของการหมุนสามารถคำนวณได้ด้วยความเร็วสูงกว่ามาก และโครงการนี้ได้พิสูจน์ให้เห็นว่าการใช้ค่าความต่างของมุมก็ทำให้ได้เส้นทางการโมที่มีค่าความผิดพลาดมากกว่าการใช้ค่าความผิดพลาดที่แท้จริงเพียงเล็กน้อย แต่ใช้เวลาในการคำนวณรวดเร็วกว่ามาก อัลกอริทึมของเราทำการเลือกตำแหน่งของจุด CL และตำแหน่งของพื้นผิวที่ต้องการสัมผัสกับตำแหน่งของวัสดุที่ให้ค่าความต่างของมุมของการหมุนน้อยที่สุด ซึ่งเราได้ทำการจำลองการตัดด้วยโปรแกรม VERICUT และผลจากการจำลองก็ยืนยันว่าวิธีของเรามีความผิดพลาดน้อยจริง

คำหลัก : เครื่องโม 5 แกน, การวางแผนการโมที่ผิดพลาดน้อยที่สุด, การใช้ค่าความต่างของมุมของการหมุน

Abstract

Project Code : MRG5380231

**Project Title : A new curve/surface intersection algorithm for undercut minimization
in 5-axis machining**

**Investigator : Asst. Prof. Dr. Gun Srijuntongsiri, Sirindhorn International Institute of
Technology, Thammasat University.**

E-mail Address : gun@siit.tu.ac.th

Project Period : 15 June 2010 to 14 February 2013

Abstract:

This project began with finding a new curve/surface intersection algorithm to be used for undercut minimization in 5-axis machining. However, the focus has later been shifted to the idea of using angular variation of the CL points to minimize kinematic error in 5-axis machining instead as we discover that the previous idea results in too long computation time, which is not feasible for 5-axis tool path planning.

In any case, the new idea is successful. We propose to find a tool path that minimizes the variation of the rotation angles instead of attempting to minimize the true kinematic error for 5-axis tool path planning. The reason is that the true kinematic error is computationally expensive and minimizing it directly takes too much computation time. On the other hand, the variation of the rotation angles is comparatively much cheaper to compute. We show that minimizing the angular variation does result in tool paths that have only slightly higher kinematic error than minimizing the true kinematic error directly while taking much shorter time to converge. Our algorithm finds the location of CL points and the orientation of the target surface relative to the work piece that minimizes the angular variation. Finally, we verify our result with virtual cutting in VERICUT. The simulation confirms that our algorithm does yield tool paths with small kinematic errors.

**Keywords : five-axis machining, tool path optimization, rotation angles
minimization.**

Executive summary

Kinematics of a particular five-axis milling machine can drastically change the machining accuracy. A tool path designed for one particular configuration and setup can be totally unacceptable for others. Therefore, the reduction of the kinematic errors due to particular five axis kinematics is an important problem of the CNC machining.

We propose a new numerical algorithm to reduce the kinematic error using minimization of the variation of the rotation angles. The method assigns the tool positions by finding numerically a grid of cutter contact points (CC points) distributed uniformly in the angular space. The second way to reduce the angle variation is the optimal setup of the part surface on the mounting table employed in an iterative loop with the generation of the cutter contact points.

We present an analysis, systematic numerical experiments and cutting results (ball nose and flat-end cutters) as an evidence of the efficiency and the accuracy increase produced by the proposed method. The algorithm has been verified by a virtual prototype of a five-axis machine MAHO600E of the CIM Lab with the Asian Institute of Technology of Thailand.

Optimization of Five-Axis Machining G-codes in the Angular Space

1 Introduction

Milling machines are programmable mechanisms for cutting complex industrial parts designed in such a way that the cutting tool (cutter) is capable of approaching the desired surface at a given point with a required orientation.

The machine is built of several moving parts designed to establish the required coordinates and orientations of the tool during the cutting process. The movements of these parts are guided by a controller which is fed with the so-called NC program or G-code. The G-code is a set of commands carrying three spatial coordinates of the tool-tip and a pair of rotation angles needed to establish the orientation of the tool. Actually, there are many other formats of the G-codes, however, this format is the most appropriate for machining an arbitrary sculptured surface.

Therefore, this type of G-code is derived from a tool path $\Omega = (\Omega_0, \Omega_1, \dots, \Omega_m)$ which is a sequence of positions $\Omega_p = (M_p, I_p)$, where $M_p = (x_p, y_p, z_p)^T$ are the Cartesian coordinates of the tool tip in the machine coordinate system (cutter location points or CL points) and $I_p = (I_{px}, I_{py}, I_{pz})^T$ the tool orientation vector. The rotation angles $R_p = (a_p, b_p)$ are functions of tool orientations. Therefore, the tool path can also be defined by $\Pi = (\Pi_0, \Pi_1, \dots, \Pi_m)$, where $\Pi_p = (M_p, R_p)$, although the transition from Ω to Π is not unique.

The kinematics of the machine is characterized by:

- Rotation matrices A and B corresponding to the two rotary axes.
- Translations T_{23} and T_{34} , where T_{23} is the coordinate of the center of the A axis in the B -axis coordinate system and T_{34} the coordinate of the center of the B -axis in the spindle coordinate system.
- The length of the tool L treated as an additional translation T_4 ($T_4 = (0, 0, L)^T$ or $(0, 0, -L)^T$ depending on the direction of the tool tip in the spindle coordinate system).

The tool path optimization problem is often formulated as minimization of a criterion vector-function which includes several measures of the machining quality. Typically such quality measures are the difference between the required and the output surface (accuracy), the length of the tool path, the negative of the machining strip (strip maximization), the machining time, etc. The independent variables are $\Omega_0, \Omega_1, \dots, \Omega_m$. In other words, the optimization improves the quality of the output surface and reduces the machining time by assigning the positions and orientation of the tool which constitute the tool

path Ω . In principle, the positions of the tool do not need to follow any rigid geometrical pattern. However, the standard manufacturing tool paths are the zigzag and the spiral path and their modifications such as the multiple zigzag.

The optimization could also be subjected to constraints, the most important of which are the scallop height constraints and the local/global accessibility constraints.

Our paper is focused on minimization of the difference between the actual and the desired trajectories. We will call this difference the kinematic error since, if the tool path is fixed, it depends only on the configuration the particular machine. Note that the three axis kinematics create linear trajectories. Therefore, the kinematic errors are easy to evaluate. As opposed to that, the five-axis machines generate non-linear trajectories which depend on the configuration and the setup. In this paper we analyze the particular error with the understanding that other sources of errors exists such as the chatter (self-excited vibrations), periodic forced vibrations, thermal deformations, tool deflection errors etc. The machine operating conditions such as the material removal rate, wet or dry cutting, clamping conditions, the tool wear and other tool imperfections could be also an important factor. Under certain conditions some of the above mentioned errors could actually exceed the kinematic errors. However, our numerical experiments and the actual machining show that the kinematic errors are also important and could lead to significant inaccuracies (even collisions) in 5 axis CNC machining.

A direct minimization of the kinematic errors requires expensive computational procedures involving inverse kinematics transformations and variable positions of the CL or CC points. In this paper we use the following short-cut. It is well-known that large variations of the rotation angles are the main cause of these errors. For instance, it is usually beneficial to generate a tool-path with minimal number of turns (such as the zigzag) or even make the turns in the air. Therefore, we propose new tool path optimization techniques based on minimization of the total angle variation which does not invoke a direct evaluation of the tool trajectories. The algorithm minimizing the total angle variation inserts additional points in the areas of large angle variations and therefore in the areas of large kinematic errors. The position and orientation of the part on the mounting table is also subject of optimization so that the orientation angles of the part relative to the table are independent variables in the closed form representation of the rotation angles of the tool.

The optimal solution is evaluated by an iterative loop “setup $\rightarrow$ tool path $\rightarrow$ setup $\rightarrow \dots$ ”.

We demonstrate that the method provides a considerably better rate of the decrease of the kinematic error (as a function of the number of the CL points) with the reference to the equi-arc length distribution of the points. The testing is performed in terms of the generic Hausdorff distance between the desired and the actual tool trajectory and between the desired surface and the tool trajectory.

The efficiency of the algorithm has been tested with a virtual prototype of MAHO600E at the CIM Lab of Asian Institute of Technology of Thailand. Since the virtual machine has been verified and compared with the actual machine, the results are equivalent to the actual machining in terms of the kinematic error. Besides, the virtual machine makes it possible to isolate the kinematic effects from the errors due to other sources mentioned above.

Finally, note that there is always a limit of the angular speed of specific machine parts. As a result, a shorter tool path with many turns may require more time than a longer tool path with fewer turns. For example, the maximum angular velocities of the primary and secondary rotational axis of MAHO600E are $\nu_{A,\max} = 235^\circ/\text{sec}$ and $\nu_{B,\max} = 162^\circ/\text{sec}$. If the maximum angular speed is exceeded, the controller reduces the angular speed increasing the machining time. Our optimization algorithm minimizes the total angle variation, thus, reducing the probability of such an event.

2 Kinematics of the Five Axis Milling Machine

This chapter explains the nonlinearity of the tool trajectories and the impact of the rotation angles on the trajectory.

Let $\mathcal{K} \equiv \mathcal{K}\{\text{parameters}\}[\text{arguments}] = \mathcal{K}\{R\}[M]$ be a kinematics transformation from the machine coordinates to the workpiece coordinates. Recall that M denotes the machine coordinates and R the rotation angles. For simplicity we denote the transformation by $\mathcal{K}[M]$ (when possible) keeping in mind the dependence on R .

Let $\mathcal{K}^{-1}[W]$ be an inverse transformation such that $\forall W, M, R, \mathcal{K}^{-1}[\mathcal{K}[M]] = M$ and $\mathcal{K}[\mathcal{K}^{-1}[W]] = W$. Let $\Pi_p \equiv (W_p, R_p)$, $\Pi_{p+1} \equiv (W_{p+1}, R_{p+1})$ be two successive coordinates of the tool path, W_p, W_{p+1} denote two successive spatial positions of the tool path and R_p, 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 to transform the part-surface coordinates into the machine coordinates $M_p \equiv (x_p, y_p, z_p)^T$ and $M_{p+1} \equiv (x_{p+1}, y_{p+1}, z_{p+1})^T$ (see Fig.??). Namely, $M_p \equiv \mathcal{K}^{-1}\{R_p\}[W_p]$. Second, the rotation angles $R \equiv R(t) = (a(t), b(t))$ and the machine coordinates $M \equiv M(t) \equiv (x(t), y(t), z(t))^T$ are assumed to change linearly between the prescribed points $t = s_p$, $t = s_{p+1}$, namely,

$$\begin{aligned} M(s_p, s_{p+1}, t) &= L_{p+1}(t)M_{p+1} + L_p(t)M_p, \\ R(s_p, s_{p+1}, t) &= L_{p+1}(t)R_{p+1} + L_p(t)R_p, \end{aligned}$$

where $L_{p+1}(t) = (t - s_p)/(s_{p+1} - s_p)$, $L_p(t) = (s_{p+1} - t)/(s_{p+1} - s_p)$, and t is the fictitious time coordinate $s_p \leq t \leq s_{p+1}$. Transforming M back to W for every t yields a trajectory of the tool tip in the workpiece coordinates given by

$$W(s_p, s_{p+1}, t) = \mathcal{K}\{R(t)\}[M(t)] = \mathcal{K}\{L_{p+1}(t)R_{p+1} + L_p(t)R_p\}[L_{p+1}(t)M_{p+1} + L_p(t)M_p].$$

In order to represent the tool path in terms of the workpiece coordinates, we eliminate M_p and M_{p+1} by using the inverse transformation $M_p = \mathcal{K}^{-1}\{R_p\}[W_p]$. Substituting M_p and M_{p+1} yields

$$W(s_p, s_{p+1}, t) = \mathcal{K}\{R(s_p, s_{p+1}, t)\}[L_{p+1}(t)\mathcal{K}^{-1}\{R_{p+1}\}[W_{p+1}] + L_p(t)\mathcal{K}^{-1}\{R_p\}[W_p]].$$

Introduce the following coordinate systems: the workpiece coordinate system O_1 , a coordinate system of the first rotary part O_2 , a coordinate system of the second rotary part O_3 and a coordinate system of the spindle O_4 . We shall call the first rotary axis the A -axis and the second rotary axis the B -axis. Consider a particular but popular machine kinematics characterized by two rotary axis

on the table (the so-called 2-0 machine, many variations of Maho by Deckel Gildemeister, Hermle 5 axis machines and many other models.). In this case

$$\begin{aligned}
M &\equiv \mathcal{K}^{-1} \equiv \mathcal{K}^{-1}\{R\}[W] = G_a B[b] (A[a] (W + T_{12}) + T_{23}) + T_{34} - T_4, \\
a &= \begin{cases} \arctan\left(\frac{I_y}{I_x}\right) & \text{if } I_x > 0 \text{ and } I_y > 0, \\ \arctan\left(\frac{I_y}{I_x}\right) + \pi & \text{if } I_x < 0, \\ \arctan\left(\frac{I_y}{I_x}\right) + 2\pi & \text{otherwise,} \end{cases} \\
b &= -\arcsin I_z,
\end{aligned} \tag{1}$$

where

$$G_a = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

and $T_4 = (0, 0, -L)^T$.

Furthermore, (1) is not a unique solution. First of all, a is periodic with the period 2π . Actually, b is also 2π -periodic, however, for this particular configuration b must belong to $[-105^\circ, 105^\circ]$.

Therefore, in order to reduce the angle variations a must be 2π -corrected, that is, if $a_p - a_{p-1} > \pi$ then $(a_p)_{new} = a_p - 2\pi$ and if $a_p - a_{p-1} < -\pi$ then $(a_p)_{new} = a_p + 2\pi$. Note that many modern controllers perform this correction automatically.

Second, if (a, b) satisfies (1), then $(a - \pi, -b - \pi)$ and $(a + \pi, -b - \pi)$ are also the solutions. The fourth solution is given by $(a - 2\pi, b)$ if $a > 0$ or $(a + 2\pi, b)$ otherwise. Further details and a shortest path optimization method with regard to the multiple solutions above are given in [8]. Similar multiple solutions can be derived for other 5 axis configurations.

Note that angle a is undefined at stationary points where $I_x = I_y = 0$. In this case, angle a can be evaluated using interpolation in a neighborhood of the stationary point. Alternatively, the tool path can be modified so that it avoids the stationary positions.

3 kinematic Error and the Angle Variation

In this section we will introduce the kinematic error and the total angle variation. Let $W^D(s_p, s_{p+1}, t) \in S(u, v)$ be a curve between W_p and W_{p+1} extracted from the surface in such a way that it represents the desired tool path between Π_p and Π_{p+1} . We define the error as the deviation between $W^D(s_p, s_{p+1}, t)$ and $W(s_p, s_{p+1}, t)$ given by

$$\epsilon = \sum_p \epsilon_{p,p+1},$$

where $\epsilon_{p,p+1} = \text{dist}(W^D(s_p, s_{p+1}, t), W(s_p, s_{p+1}, t))$ and $\text{dist}()$ is an appropriate distance in the corresponding functional space.

The definitions above can be simplified by replacing the desired trajectories $W^D(s_p, s_{p+1}, t)$ by linear trajectories given by $W^L(s_p, s_{p+1}, t) = W_{p+1}L_{p+1}(t) + W_pL_p(t)$. However, as opposed to the machines coordinates M , the trajectories in the workpiece coordinates are *not* linear. Still, we may use the linear trajectories as a reference, noting that $\epsilon \equiv \text{dist}(W^D, W) \leq \text{dist}(W^D, W^L) +$

$\text{dist}(W^L, W)$, where L is a piecewise linear approximation of S . Hence, when the points are close enough, the error is approximately $\epsilon \approx \epsilon_L = \text{dist}(W^L, W)$. If the orientation of the tool is fixed through the entire cut, then $\epsilon_L = 0$. In other words, the 3-axis mode leads to $\epsilon_L = 0$ since all the trajectories become linear. Does this mean that 3 axis machining is more accurate? Not at all. High accuracy of the five-axis machining comes from direct control of the tool orientations relative to the surface. Many complex parts can be produced only in 5 axis mode.

Mathematically it means that minimization with regard to ϵ_L is subjected to constraints specifying the orientations of the tool.

Finally, the linearization is the simplest option which can be used even when the actual surface is not known, for instance, when the G-code is given but the surface is not specified.

Also note that, the desired trajectory W^D can be extracted from the surface by a variety of ways, for example, using interpolation in the parametric space, the geodesic curves, etc.

Let us now introduce the total weighted angle variation. Consider two positions Ω_p and Ω_{p+1} . We assume the kinematic error is an increasing function of the distance between two positions in the Cartesian and the angular space. Practical experiments show that the angular steps are often more important than the spatial step. In particular, decreasing the angular steps leads to larger decrease in the error than decreasing the spatial step. Therefore, we define *total weighted angle variation*

$$\delta = \sum_p |W^D(s_p, s_{p+1}, t)| (\alpha_a |a_{p+1} - a_p| + \alpha_b |b_{p+1} - b_p|), \quad (2)$$

where $|W^D(s_p, s_{p+1}, t)|$ denotes the arc length of the curve $W^D(s_p, s_{p+1}, t)$ and α_a and α_b are predefined scalars. For MAHO600E, the angles a_p 's and b_p 's are equivalent in the sense that changing a_p 's does not affect the kinematic error more than changing b_p 's, and *vice versa*. Therefore, we use $\alpha_a = \alpha_b = 1$ in our experiments below. Note that this assumption may not hold for other machines.

4 Minimization of the Angle Variation

In this section, we describe our algorithm that minimizes the total weighted angle variation of the tool path by distributing a fixed number of CL points and rotating and translating the target surface relative to the mounting table optimally at the same time. Define the rotated and translated surface

$$\bar{S}(u, v; \mathbf{d}, \mathbf{r}) = R_z(r_3)R_y(r_2)R_x(r_1)S(u, v) + \mathbf{d}, \quad 0 \leq u, v \leq 1,$$

where $\mathbf{d} \in \mathbb{R}^3$, $\mathbf{r} \in \mathbb{R}^3$, and

$$\begin{aligned} R_x(r_1) &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos r_1 & -\sin r_1 \\ 0 & \sin r_1 & \cos r_1 \end{bmatrix}, \\ R_y(r_2) &= \begin{bmatrix} \cos r_2 & 0 & -\sin r_2 \\ 0 & 1 & 0 \\ \sin r_2 & 0 & \cos r_2 \end{bmatrix}, \\ R_z(r_3) &= \begin{bmatrix} \cos r_3 & -\sin r_3 & 0 \\ \sin r_3 & \cos r_3 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned}$$

That is, $\bar{S}(u, v; \mathbf{d}, \mathbf{r})$ is the target surface $S(u, v)$ rotated around x -, y -, and z -axes by r_1 , r_2 , and r_3 radians, respectively, and then translated by $\mathbf{d}$. Let $(W_0, W_1, \dots, W_m)$ be the sequence of given CL points generated from $\bar{S}$ (for example, the sequence of turning points of the zigzag on the rotated and translated surface) and $n_p \geq 0$ be the chosen number of CL points to be inserted between W_p and W_{p+1} . Let $t_{p1}, t_{p2}, \dots, t_{p, n_p}$ denote the location of the inserted points between W_p and W_{p+1} . In other words, our algorithm inserts CL points at $W^D(s_p, s_{p+1}, t_{p1})$, $W^D(s_p, s_{p+1}, t_{p2})$, $\dots$, $W^D(s_p, s_{p+1}, t_{p, n_p})$ between W_p and W_{p+1} (we then calculate the corresponding orientation vectors I_q 's at these points). Therefore, the optimization problem is given by

$$\begin{aligned} \min_{\mathbf{d}, \mathbf{r}, t_{01}, \dots, t_{m-1, n_{m-1}}} & \quad \delta(\mathbf{d}, \mathbf{r}, t_{01}, \dots, t_{m-1, n_{m-1}}) \\ \text{subject to} & \quad s_0 < t_{p1} < t_{p2} < \dots < t_{p, n_p} < s_m \quad (p = 0, \dots, m-1) \\ & \quad M(s_p, s_{p+1}, t_{pi}) \in \hat{M} \quad (p = 0, \dots, m-1; i = 1, \dots, n_p) \quad , \\ & \quad R(s_p, s_{p+1}, t_{pi}) \in \hat{R} \quad (p = 0, \dots, m-1; i = 1, \dots, n_p) \\ & \quad W^D(s_p, s_{p+1}, t_{pi}) \in \hat{W} \quad (p = 0, \dots, m-1; i = 1, \dots, n_p) \end{aligned} \quad (3)$$

where $\delta(\mathbf{d}, \mathbf{r}, t_{01}, \dots, t_{m-1, n_{m-1}})$ is the total weighted angle variation (2) of the tool path generated from $\bar{S}(u, v; \mathbf{d}, \mathbf{r})$ with the inserted CL points at $t_{01}, \dots, t_{m-1, n_{m-1}}$, $M(s_p, s_{p+1}, t_{pi})$ denotes the machine coordinate of $W^D(s_p, s_{p+1}, t_{pi})$, $\hat{M}$ denotes the set of admissible machine coordinates, $\hat{R}$ denotes the set of admissible rotation angles, and $\hat{W}$ denotes the set of admissible workpiece coordinates. Note that $\hat{M}$, $\hat{R}$, and $\hat{W}$ depend on the physical constraints of the target five-axis machine and the workpiece. Some machines may not have these restrictions and in such cases the last three constraints in (3) can be ignored.

5 Numerical Algorithm

Observe that the objective function in (3) is generally non-differentiable and should be minimized by a derivative-free technique such as genetic algorithms [3, 1, 10, 11], pattern search methods [4, 5], or Nelder–Mead method [9, 2]. During our preliminary tests, we found that solving (3) as is takes an impractical amount of computation time. Therefore, our algorithm employs the pattern search method combined with the split-step technique. We alternate between optimizing the surface orientation while keeping the inserted CL points fixed and optimizing the inserted points while keeping the surface orientation fixed

until the process converged. Specifically, we separate (3) into the orientation optimization

$$\begin{aligned} \min_{\mathbf{d}, \mathbf{r}} \quad & \delta(\mathbf{d}, \mathbf{r}, t_{01}, \dots, t_{m-1, n_{m-1}}) \\ \text{subject to} \quad & \begin{aligned} M(s_p, s_{p+1}, t_{pi}) &\in \hat{M} & (p=0, \dots, m-1; i=1, \dots, n_p) \\ R(s_p, s_{p+1}, t_{pi}) &\in \hat{R} & (p=0, \dots, m-1; i=1, \dots, n_p) \\ W^D(s_p, s_{p+1}, t_{pi}) &\in \hat{W} & (p=0, \dots, m-1; i=1, \dots, n_p) \end{aligned} \end{aligned} \quad (4)$$

where $t_{01}, \dots, t_{m-1, n_{m-1}}$ are fixed, and the insertion optimization

$$\begin{aligned} \min_{t_{01}, \dots, t_{m-1, n_{m-1}}} \quad & \delta(\mathbf{d}, \mathbf{r}, t_{01}, \dots, t_{m-1, n_{m-1}}) \\ \text{subject to} \quad & \begin{aligned} s_0 < t_{p1} < t_{p2} < \dots < t_{p, n_p} < s_m & (p=0, \dots, m-1) \\ M(s_p, s_{p+1}, t_{pi}) &\in \hat{M} & (p=0, \dots, m-1; i=1, \dots, n_p) \\ R(s_p, s_{p+1}, t_{pi}) &\in \hat{R} & (p=0, \dots, m-1; i=1, \dots, n_p) \\ W^D(s_p, s_{p+1}, t_{pi}) &\in \hat{W} & (p=0, \dots, m-1; i=1, \dots, n_p) \end{aligned} \end{aligned} \quad (5)$$

where $\mathbf{d}$ and $\mathbf{r}$ are fixed. Our algorithm then proceeds as follows:

- Initialize $\mathbf{d}^{(0)}, \mathbf{r}^{(0)}, t_{01}^{(0)}, \dots, t_{m-1, n_{m-1}}^{(0)}$.
- For $k = 0, 1, \dots$
 - Solve (4) using $t_{ij} = t_{ij}^{(k)}$. Let $(\tilde{\mathbf{d}}, \tilde{\mathbf{r}})$ be the minimum point of this problem.
 - Solve (5) using $\mathbf{d} = \tilde{\mathbf{d}}$ and $\mathbf{r} = \tilde{\mathbf{r}}$. Let $(t_{01}^{(k+1)}, \dots, t_{m-1, n_{m-1}}^{(k+1)})$ be the minimum point of this problem.
- End

In practice, we iterate until the algorithm converges.

It should be noted that the locations of inserted CL points between W_p and W_{p+1} do not affect the weighted angle variation values of the tool path from W_0 up to W_p and from W_{p+1} up to W_m besides the possibility that R_{p+1} may be 2π -corrected differently depending on the location of the last inserted point t_{p, n_p} (or, depending on how 2π -correction is implemented, it can be R_p instead that is 2π -corrected differently). Therefore, the insertion step can be performed between one pair of the original CL points (W_p, W_{p+1}) at a time starting from one end of the tool path to improve computation time at the cost of minor reduction in accuracy. That is, first finding the optimal $t_{01}, \dots, t_{0n_0}$ that minimizes the weighted angle variation between W_0 and W_1 , then finding the optimal $t_{11}, \dots, t_{1n_1}$ that minimizes the weighted angle variation between W_1 and W_2 , and so on.

Finally, we note that the techniques to avoid gouging for a flat-end cutter by Lee and Ji [6], Lo [7], or Makhanov and Anotaipaihoon [8] can be incorporated into our algorithm. This is performed by inclining the tool while keeping the CC points at the chosen inserted points $W^D(s_p, s_{p+1}, t_{pi})$'s and using the resulted CL points to evaluate δ during the optimization. By this technique, our algorithm finds the tool path that avoids gouging with the minimum total weighted angle variation.

6 Numerical Examples and Cutting Experiments

In this section we demonstrate that the total weighted angle variation does indeed reduce the kinematic error by a significant amount. To do so, we estimate the kinematic error with the Hausdorff distance between the actual tool path trajectory $\{W(s_p, s_{p+1}, t) : p = 0, \dots, m-1\}$ and the desired trajectory $\{W^D(s_p, s_{p+1}, t) : p = 0, \dots, m-1\}$. Recall that the *Hausdorff distance* between two sets X and Y is defined by

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \inf_{y \in Y} \|x - y\|, \sup_{y \in Y} \inf_{x \in X} \|x - y\| \right\}.$$

In our experiments, we use the 2-norm for all computation of the Hausdorff distances.

The first surface for our experiments is given by

$$S(u, v) = \left\{ \begin{bmatrix} 100u - 50 \\ 100v - 50 \\ -400v(v-1)(3.55u - 14.8u^2 + 21.15u^3 - 9.9u^4) - 140 \end{bmatrix} : 0 \leq u, v \leq 1 \right\}. \quad (6)$$

We shall refer to the surface in (6) as the “double bells.” The surface is also shown in Figure 1. For the first experiment, we compare the following three types of tool paths: (i) an unoptimized tool path (with equally-spaced inserted points between the two endpoints of every track and $\mathbf{d} = \mathbf{r} = 0$), (ii) a tool path generated with insertion optimization only, and (iii) a tool path generated with both orientation and insertion optimization. Table 1 shows the total weighted angle variations and the estimated kinematic error (i.e., the Hausdorff distance between the actual and the desired trajectories) of the three types of tool paths with ten tracks and varying number of inserted points (n_p) per track. Figures 2, 3, and 4 show the three tool paths and their actual trajectories for $n_p = 8$ case, respectively. Blue crosses in these figures represent the CL points along the tool paths. Figure 5 rotates Figure 4 to the same orientation as the other two for easier comparison. Note that we reduce the resolution of the plotted surface in these figures so that the tool paths can be seen more easily. We see that our optimization method, which minimizes the total weighted angle variations, does decrease the estimated kinematic error for all of the test cases. We also see that fully-optimized tool paths have smaller kinematic error compared to insertion-optimization-only ones for all of the cases. This shows that orientation optimization is indeed useful. Note that smaller δ does not always imply smaller d_H (such as between the fully-optimized tool path of $n_p = 4$ case and the fully-optimized tool path of $n_p = 8$ one). Such occasional discrepancy is to be expected as the total weighted angle variation is only an approximation of the actual kinematic error.

Consider now examples of the virtual machining by prototyping MAHO600E in VERICUT. This virtual machine has been tested and compared with the actual machine. Therefore, the results of the virtual machining are equivalent to the actual machining in terms of the kinematic error. Besides, the virtual machine allows to differentiate between the kinematic effects and the errors originated from other sources such as tool deflection, chatters, and thermal deformations. Figure 6 shows the simulated result from VERICUT of the double-bells surface using the unoptimized tool path. Figure 7 shows the result using the

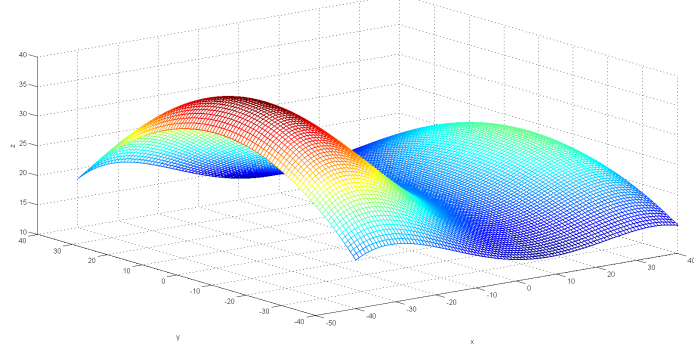


Figure 1: The first target surface for the experiments (the “double bells”).

Table 1: The total weighted angle variations (δ) and the Hausdorff distance from the desired tool paths (d_H) of the three types of tool paths for the double-bell surface. All tool paths have ten tracks.

n_p	Unoptimized		Insertion only		Fully optimized	
	δ	d_H	δ	d_H	δ	d_H
2	1248.7641	9.8456	239.8314	6.9870	110.598	6.6368
4	790.7744	6.9576	230.1996	4.0306	82.566	3.2421
6	584.4124	5.4528	185.806	3.3030	80.2953	2.2186
8	468.6081	4.1537	191.7998	2.9517	110.7122	2.3537
10	389.423	3.3335	185.1702	2.5509	97.2301	1.7139

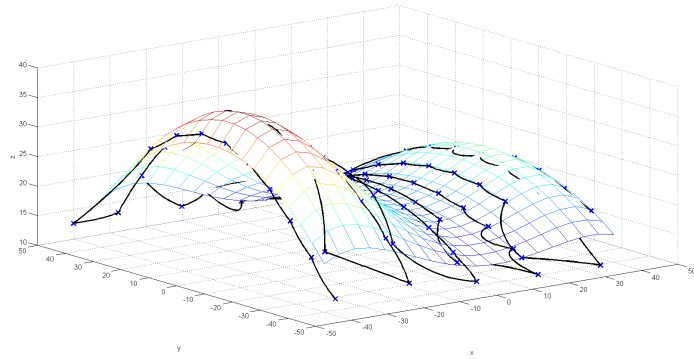


Figure 2: The unoptimized tool path (with equally-spaced CL points) for the double-bells surface with a flat-end tool head having radius of 8 mm. The tool path has 10 tracks with 10 CL points per track ($n_p = 8$). Blue crosses represent the CL points.

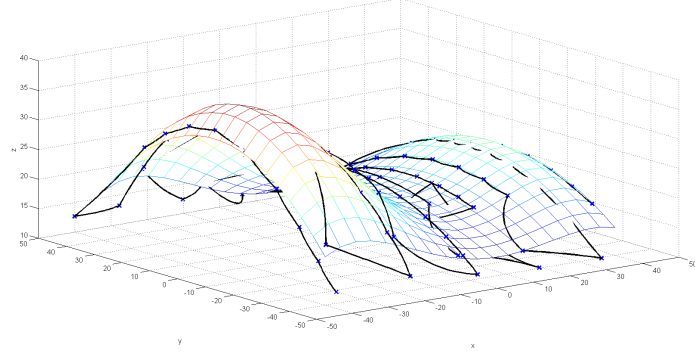


Figure 3: The tool path generated with insertion optimization only for the double-bells surface with a flat-end tool head having radius of 8 mm. The tool path has 10 tracks with 10 CL points per track ($n_p = 8$). Blue crosses represent the CL points.

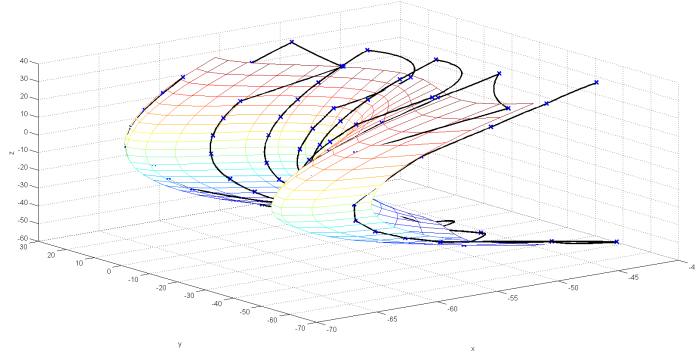


Figure 4: The tool path generated with both orientation and insertion optimization for the double-bells surface with a flat-end tool head having radius of 8 mm. The tool path has 10 tracks with 10 CL points per track ($n_p = 8$). Blue crosses represent the CL points.

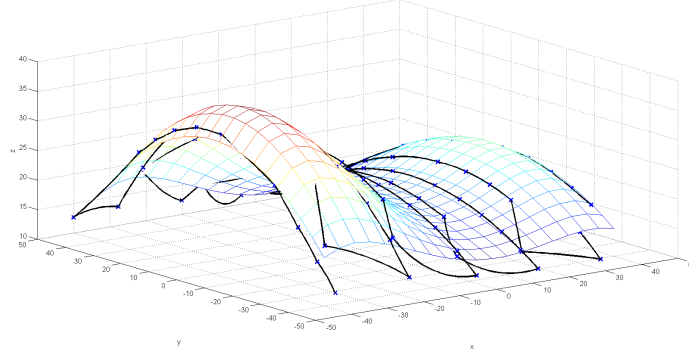


Figure 5: The tool path generated with both orientation and insertion optimization for the double-bells surface with a flat-end tool head having radius of 8 mm after having been rotated back to the original orientation. The tool path has 10 tracks with 10 CL points per track ($n_p = 8$). Blue crosses represent the CL points.

tool path generated with insertion optimization only. Figure 8 shows the result using the tool path generated with both orientation and insertion optimization. All three cuts are done with a flat-end tool head with radius of 8 mm. The three tool paths all have 20 tracks and 20 CL points per track and are shown in Figures 9, 10, and 11, respectively. Note that Figure 11 has been rotated to the original orientation for better comparison with the other two tool paths. We see that our optimization reduces the kinematic error for the same number of CL points. Also observe that the cut from the fully-optimized tool path has smaller kinematic error compared to the cut from the insertion-optimization-only one.

Finally, we perform virtual cutting of a second target surface in VERICUT. The surface is a Bézier surface of order (4, 4) and is shown in Figure 12 below. We shall refer to this surface as the “valley.” This surface has higher curvature than the double-bells and thus requires larger number of CL points to cut with reasonable accuracy.

Figure 13 shows the simulated result from VERICUT of the valley surface using the unoptimized tool path. Figure 14 shows the result using the tool path generated with both orientation and insertion optimization. Both cuts are done with a flat-end tool head with radius of 3.5 mm. The two tool paths have 50 tracks and 30 CL points per track and are shown in Figures 15 and 16, respectively. Figure 16 has been rotated to the original orientation. As before, we see that our optimization reduces the error in the simulated cut.

7 Conclusions

We propose a method to minimize kinematic error in five-axis machining by optimally inserting CL points between given key CL points in the tool path and rotating and translating the target surface relative to the workpiece. Instead of using the true kinematic error as the objective function in the optimization, we propose to minimize the total weighted angle variation as it is much simpler to

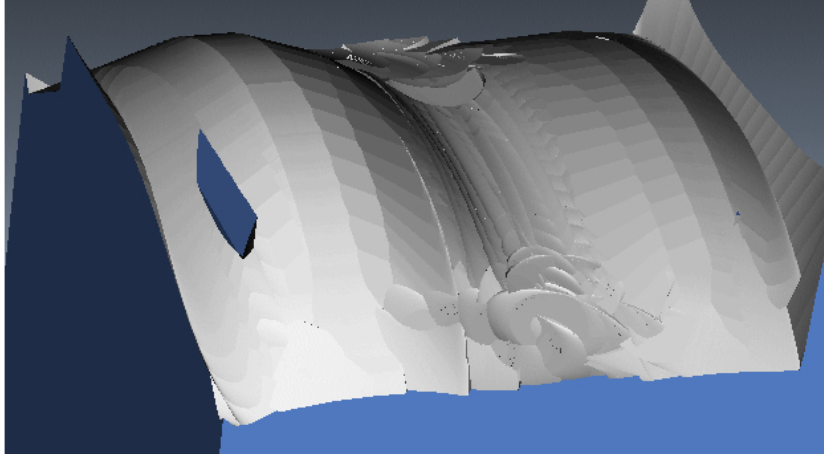


Figure 6: The simulated result from VERICUT of the double-bells surface using the unoptimized tool path (with equally-spaced CL points) with a flat-end tool head having radius of 8 mm. The tool path has 20 tracks with 20 CL points per track ($n_p = 18$).

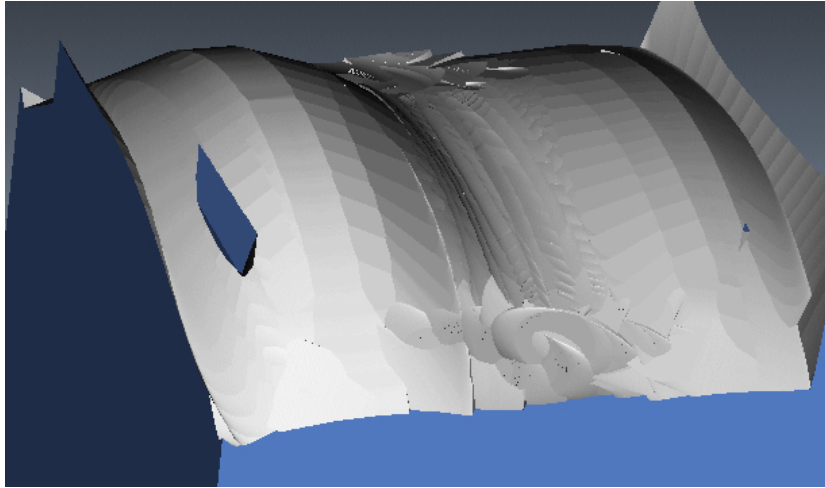


Figure 7: The simulated result from VERICUT of the double-bells surface using the tool path generated with insertion optimization only with a flat-end tool head having radius of 8 mm. The tool path has 20 tracks with 20 CL points per track ($n_p = 18$).

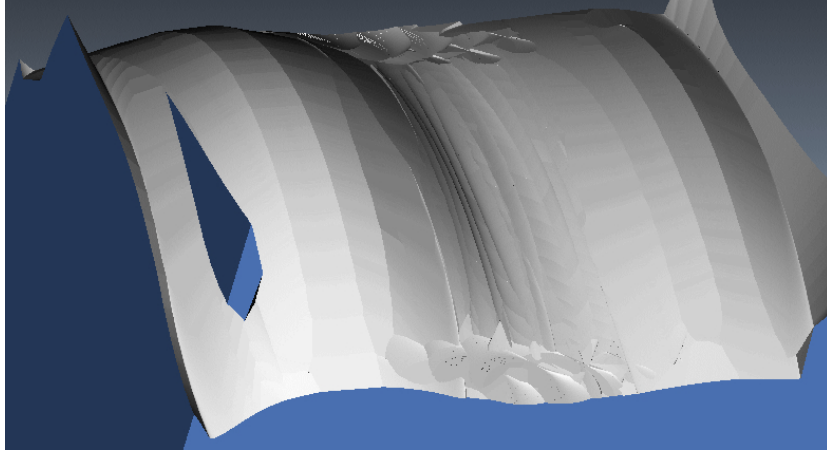


Figure 8: The simulated result from VERICUT of the double-bells surface using the tool path generated with both orientation and insertion optimization with a flat-end tool head having radius of 8 mm. The tool path has 20 tracks with 20 CL points per track ($n_p = 18$).

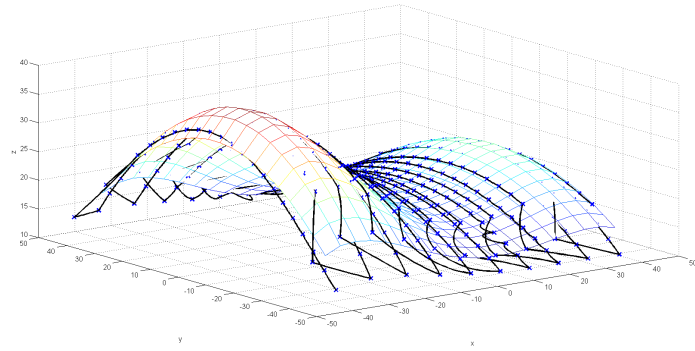


Figure 9: The unoptimized tool path (with equally-spaced CL points) for the double-bells surface with a flat-end tool head having radius of 8 mm. The tool path has 20 tracks with 20 CL points per track. Blue crosses represent the CL points.

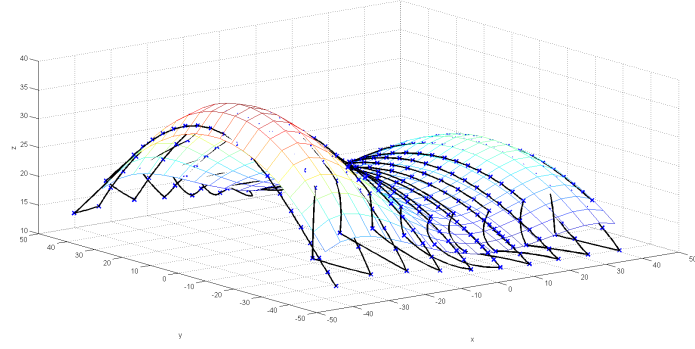


Figure 10: The tool path generated with insertion optimization only for the double-bells surface with a flat-end tool head having radius of 8 mm. The tool path has 20 tracks with 20 CL points per track. Blue crosses represent the CL points.

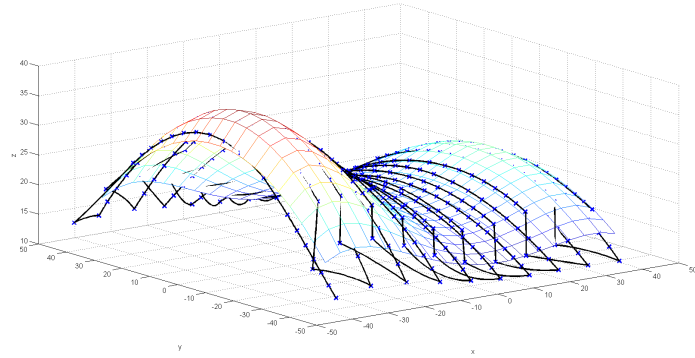


Figure 11: The tool path generated with both orientation and insertion optimization for the double-bells surface with a flat-end tool head having radius of 8 mm after having been rotated back to the original orientation. The tool path has 20 tracks with 20 CL points per track. Blue crosses represent the CL points.

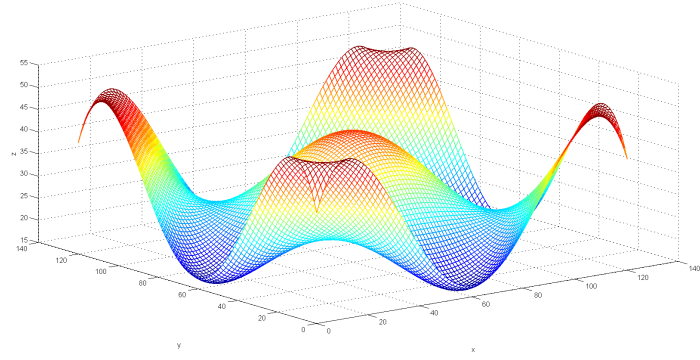


Figure 12: The second target surface for the virtual cutting.

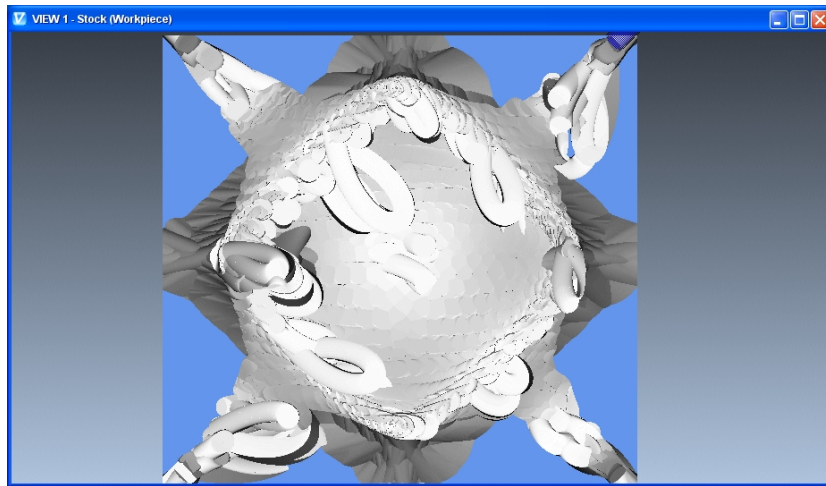


Figure 13: The simulated result from VERICUT of the valley surface using the unoptimized tool path with a flat-end tool head having radius of 3.5 mm. The tool path has 50 tracks with 30 CL points per track ($n_p = 28$).

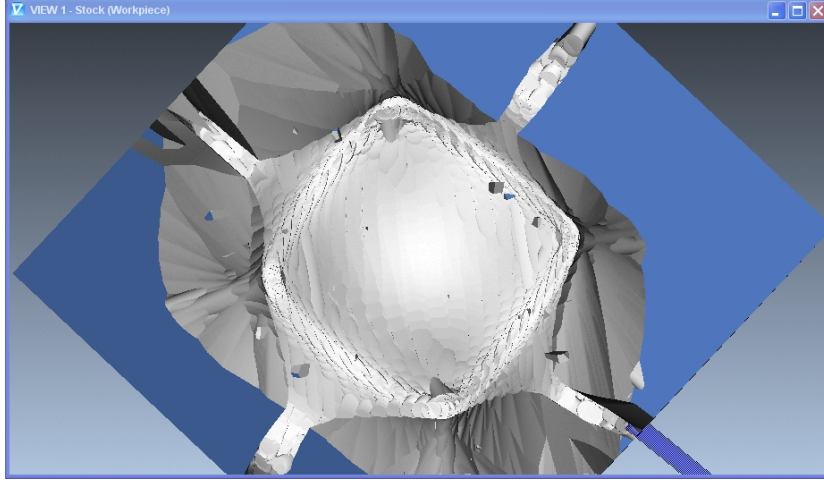


Figure 14: The simulated result from VERICUT of the valley surface using the tool path generated with both orientation and insertion optimization with a flat-end tool head having radius of 3.5 mm. The tool path has 50 tracks with 30 CL points per track ($n_p = 28$).

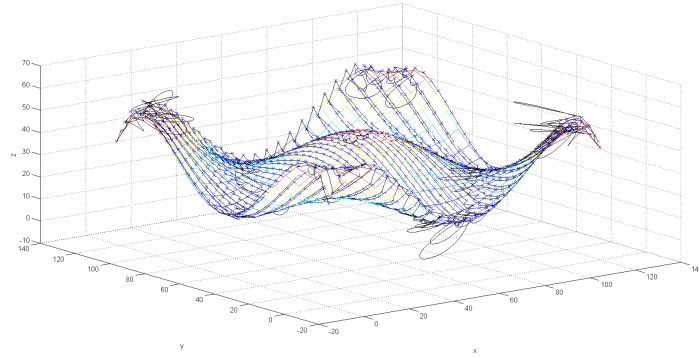


Figure 15: The unoptimized tool path (with equally-spaced CL points) for the valley surface with a flat-end tool head having radius of 3.5 mm. The tool path has 50 tracks with 30 CL points per track. Blue crosses represent the CL points.

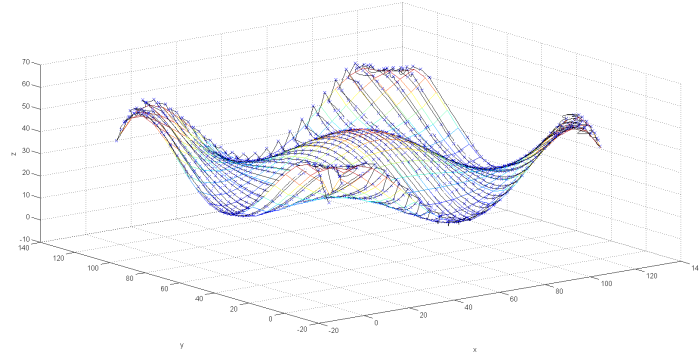


Figure 16: The tool path generated with both orientation and insertion optimization for the valley surface with a flat-end tool head having radius of 3.5 mm after having been rotated back to the original orientation. The tool path has 50 tracks with 30 CL points per track. Blue crosses represent the CL points.

compute and approximates the true kinematic error quite well. Our experimental results show that the weighted angle variation minimization does minimize the kinematic error, too, as we expect. We also see that doing both insertion and orientation optimization yields more accurate tool paths than doing only insertion optimization. Finally, we verify our results with VERICUT simulation of a MAHO600E machine and see that the optimized tool paths yield more accurate surfaces in the simulation, too.

8 Acknowledgments

This work is supported by the grant number MRG5380231 from Thailand Research Fund, Office of the Higher Education Commission of Thailand, and Sirindhorn International Institute of Technology, Thammasat University, Thailand. The authors also would like to thank Samart Moodleah for his help with VERICUT software.

References

- [1] T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.
- [2] E. K. P. Chong and S. H. Żak. *An Introduction to Optimization (Wiley-Interscience Series in Discrete Mathematics and Optimization)*. Wiley-Interscience, 3 edition, February 2008.
- [3] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

- [4] R. Hooke and T. A. Jeeves. “Direct search” solution of numerical and statistical problems. *J. ACM*, 8(2):212–229, April 1961.
- [5] T. G. Kolda, R. M. Lewis, and V. Torczon. Optimization by direct search: New perspectives on some classical and modern methods. *SIAM Review*, 45(3):385–482, September 2003.
- [6] Y.-S. Lee and H. Ji. Surface interrogation and machining strip evaluation for 5-axis cnc die and mold machining. *International Journal of Production Research*, 35(1):225–252, 1997.
- [7] C. C. Lo. Efficient cutter-path planning for five-axis surface machining with a flat-end cutter. *Computer-Aided Design*, 31(9):557–566, 1999.
- [8] S. S. Makhanov and W. Anotaipaiboon. *Advanced Numerical Methods to Optimize Cutting Operations of Five Axis Milling Machines*. Springer Publishing Company, Incorporated, 1st edition, 2007.
- [9] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1964.
- [10] C. R. Reeves and J. E. Rowe. *Genetic Algorithms—Principles and Perspectives, A Guide to GA Theory*. Kluwer Academic Publishers, London, UK, 2003.
- [11] E. Talbi. *Metaheuristics: From Design to Implementation*. Wiley Publishing, 2009.

Output จากโครงการวิจัยที่ได้รับทุนจาก สกว.

1. ได้มีการเขียนบทความวิจัยจากผลลัพธ์ที่ได้จากโครงการนี้ เพื่อส่งวารสารวิชาการนานาชาติเพื่อพิจารณาตีพิมพ์ แต่ยังไม่ได้รับการตอบรับ