Five-Axis Tool Path Optimization Using Rotations and Orientation

Mud-Armeen Munlin
Faculty of Information Science and Technology
Mahanakorn University of Technology
51 Cheum-Sampan Rd., Nong Chok, Bangkok 10530 Thailand
Tel: 662-988-3655 ext 4112 Fax: 662-988-4027
Email: mmunlin@gmail.com

ABSTRACT
This paper presents the algorithms to interactively optimize the tool path of the five-axis milling machine. The algorithms employ the inverse kinematics to derive the corresponding trajectories (rotation and translation movement) of the five-axis machine. The trajectory of the tool path is not unique and depends on the initial set up of the workpiece as well as the combined rotations of the machine rotary axes. The proposed methods interactively select the appropriate workpiece orientation with the combination of rotations. The resulting optimized tool path can be used to simulate the milling process, verify the final cut and estimate the errors of the actual tool path before the real machining. It has been shown that the interactive approach provides an efficient way to the modification of a tool path based on an appropriate set of rotations and the initial workpiece orientation. Tool path optimization is verified by computer simulation software developed by the author as well as by real cutting on MAHO600E five-axis machine.

Keywords: Inverse Kinematics, Five-Axis Machining, Tool Path Optimization, Shortest Path.

1. INTRODUCTION

Milling machines are programmable mechanisms for cutting industrial parts. The machine consists of several moving parts designed to establish the required coordinates and orientations of the tool during the cutting process. The axes of the machine define the number of the degrees of freedom of the cutting device. The movements of the machine parts are guided by a controller which is fed with a so-called NC program comprising commands carrying spatial coordinates of the tool-tip and angles needed to rotate the machine parts to establish the orientation of the tool.

One of the most popular are five-axis machines characterized by three translation and two rotation axes which offer the minimal number of the degrees of freedom required to transport the tool into a spatial position and to establish a required orientation. However, the machines with rotation axes on the table often have to turn around heavy workpieces. Therefore, they must support significant mechanical efforts during machining. As a result, the machines may have low capacities for acceleration. When the machine has to slow down or stop, the speed reduction requires a considerable time for deceleration and re-acceleration. This effect significantly increases machining time, and is amplified in HSM when the rotation axes reach greater speeds. Besides, the rotation axes invoke an inevitable non-linearity of the tool tip trajectory and consequently so-called kinematics errors appearing due to the specific kinematics of the machine.

The main goal of five-axis tool path optimization is minimization the difference between the desired and the actual surface while producing the actual surface for a minimum time. However, mathematical formulations presented in the literature vary in terms of the error criteria and the set of optimized variables. The tool path is 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. [1,2,3]. Furthermore, the error analysis and optimization in the areas of large variations of the rotation angles have not been provided by commercial CAD/CAM systems such as Unigraphics, EdgeCam, Vericut, etc. Besides, only a few research papers deal with the subject.

In [4] the authors analyze the sequence of rotations to minimize the number of the phase reverse steps at discontinuities of the first derivative of the surface (corners etc). A method of avoiding singularities has been presented in [5]. The method certainly has its merits since it allows inserting additional points without any modifications. However, the computation is complex, computationally expensive and does not preserve the original CC points. In [6] the authors proposed an angle switching algorithm to optimize the sequence of the rotation angles without increasing the number of tool positions or changing the tool orientation. The main idea is to minimize the distance traveled by the tool in the angular space at the expense of using multiple solutions of the inverse kinematics equations that is switching the rotation angles at certain position. Considering the entire set of angles requires the shortest path techniques to minimize the total angular distance. In [7] the algorithm was extended to the case when the cost function differentiates between more damaging undercuts and repairable over cuts. It has been noticed in [7] that inserting new points in the resulting path may not be always possible because some optimization is based on replacing large angle variations by smaller ones using different combinations of the angles. Inserting additional points decreases these variations so that in some points the replacement is no longer acceptable. In turn, a simple removal of an angle switch is not suitable either since it may affect the entire tool path.

Therefore, the author proposes two new optimization algorithms. The first algorithm iteratively switches the rotation angles when the additional points are inserted. This algorithm is based on the equi-distribution with regard to the rotation angles combined with the angle...
switching scheme. The second algorithm solves an important problem of finding an optimal position and orientation of the workpiece which minimizes the kinematics error. Such analysis is not provided by commercial CAD/CAM software. The proposed algorithm is compared with the results obtained by traditional positioning of the workpiece. Namely, the minimal \( z \) is at a certain distance above the mounting table (collision avoidance). The center of the base of the part (which is the square 100 by 100 mm) is positioned at the center of rotation of the mounting table. The \( x \)-axis and \( y \)-axis of the workpiece coordinate are parallel to the surface of the mounting table and the \( z \)-axis is perpendicular to the table. The efficiency of the algorithm has been verified by a virtual machining [8] as well as by real cutting on five-axis machine MAHO600E.

### 2. ITERATIVE ANGLE SWITCHING ALGORITHM

When it is necessary to insert additional points, there are many methods for such insertions. However, from the viewpoint of the kinematics error one of the most efficient methods near stationary points is angular insertion proposed in [7]. The method “injects” the points into large loops by equidistributing them with regard to the rotation angle having the largest variation. This complies with the idea of functional that the angle variation affects the kinematics error the most.

Our introductory example presents a saddle surface \( S \) given by

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

Consider the optimized saddle surface \( S(u,v) \) in Fig.1, where the tool path obtained by the angle switching algorithm [6]. The largest loop is on the right side of the surface. The rotation angles (degree) before applying the algorithm are \( a_i = 319 \), \( b_i = -79 \), \( a_{i+1} = 224 \), \( b_{i+1} = -80 \). The shortest path optimization produces \( a_i = 319 \), \( b_i = -79 \), \( a_{i+1,new} = a_{i+1} + 180 = 404 \), \( b_{i+1,new} = b_{i+1} - 180 = -100 \). Inserting a point in the middle of the loop yields \( a_{mid} = 267 \), \( b_{mid} = -82 \). Taking into that we have to perform the same modification as with \( a_i \), \( b_i \), we have the new pair of angles given by \( a_{mid} = a_{mid} + 180 = 447 \), \( b_{mid} = b_{mid} - 180 = -98 \), which produces a larger loop (Fig.2). Moreover \( a_{mid} \notin [a_i,a_{i+1}] \) anymore. Therefore, a single additional point may destroy the integrity of a particular shortest path. The iterative angle switching algorithm is proposed to iteratively switch the angles when the additional points are inserted as follow:

1. Run the angle switching (AS) algorithm [6].
2. If the kinematics error is within the prescribed tolerance, quit, otherwise, find the trajectory with the kinematics error exceeding the tolerance.
3. Mark this trajectory.
4. Return to the original tool path (ORG) and using the angle insertion (AI) algorithm [7] to insert a point inside the selected trajectory even though in the original path they do not produce large kinematics error.
5. Repeat step 1.

A combination of the angle switching (step 1) and the angle insertion (step 4) leads to even better optimizations. It is clear that the algorithm converges because in the worst case additional points are inserted into every interval. However, in this case switching of the angles is no longer possible to accomplish. Therefore, we are interested in solutions when benefits of the angle switching are combined with error reduction produced by inserting some relatively small number of points. In other words, the question whether it is practical that a few points have been inserted to reduce the error and some switching is still present in the path. We also are interested whether the above mentioned angle insertion techniques still provides some benefits as the basic insertion method in this algorithm.

Fig.3 show the surface \( S(u,v) \) subjected to the above mentioned procedures after the maximum error has been reduced to a certain prescribed value using several optimization methods. It has been proven experimentally that the proposed method requires 47.3 % less additional
points than conventional schemes performed near the stationary points. The efficiency of the algorithm has been verified by a five-axis machine MAHO600E at the CIM Lab of Asian Institute of Technology of Thailand (Fig.4). It has been also verified by solid modeling in Unigraphics V18 [9] as well as through the virtual milling machine simulator [8] developed by the author.

Table 1: Comparison of the proposed iterative angle switching scheme with other methods

<table>
<thead>
<tr>
<th>N</th>
<th>Error AS</th>
<th>Error AS$\dagger$ PI</th>
<th>Error AS$\dagger$ ORG$\dagger$ PI$\dagger$ AS</th>
<th>Error AS$\dagger$ ORG$\dagger$ AI$\dagger$ AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7.451</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>1</td>
<td>N/A</td>
<td>9.447</td>
<td>2.173</td>
<td>2.144</td>
</tr>
<tr>
<td>3</td>
<td>N/A</td>
<td>5.826</td>
<td>0.571</td>
<td>0.543</td>
</tr>
<tr>
<td>7</td>
<td>N/A</td>
<td>3.009</td>
<td>0.144</td>
<td>0.136</td>
</tr>
<tr>
<td>15</td>
<td>N/A</td>
<td>1.51</td>
<td>0.35</td>
<td>0.33</td>
</tr>
</tbody>
</table>

3. OPTIMIZATION OF THE WORKPIECE SETUP

Given the five-axis machine configuration in Fig.5 and Fig.6, the following introductory example shows that the optimization could be surprisingly efficient. Let $W_p$, $W_{p+1}$ be two successive spatial positions of the tool path and $I_p$, $I_{p+1}$ the corresponding tool orientations. The actual tool trajectory between $W_p$ and $W_{p+1}$ depends on the initial workpiece position $T_{12}$ and the orientation $r_a$, $r_b$ with regard to the mounting table. Fig. 7 shows the workpiece setup performed by first rotating the workpiece around the $Z_1$-axis by $r_a$ and around the $Y_1$-axis by $r_b$. The workpiece is offset from the center of the first rotary axis coordinate $O_2$ by $T_{12}$ (see Fig.6). In the new workpiece coordinates we have

$$
W'_p = R_b [r_b] R_a [r_a] W_p + T_{12}, \\
I'_p = R_b [r_b] R_a [r_a] I_p,
$$

(2)

where $R_a, R_b$ are the corresponding matrices of rotation.

Table 1 compares the accuracy achieved by the iterative angle switching with other methods when the number of inserted points is fixed. N represents number of inserted points and PI stands for the equi-spaced point insertion. The advantage of the proposed method is transparent. For instance, when 3 points are inserted the method improves the accuracy by factor of 10 which is undoubtedly a very impressive gain. However, the iterative method does not produce much different results between the equi-spaced point insertion and equi-angular point insertion. This is because the method presents an appropriate combination of angle switching and inserting, whereas PI and AI use only a single technique.
Let \( K = K \{ \text{parameters} \} \{ \text{arguments} \} = K \{ R \} \{ M \} \) be a kinematics transformation from the machine coordinates to the workpiece coordinates. In order to simplify the notations we will omit the parameters when appropriate.

Let \( K^{-1} \equiv K^{-1} \{ R \} \{ W \} \) be the inverse transformation such that \( \forall W, M, R \),

\[
K^{-1} [K[M]] = M \quad \text{and} \quad K [K^{-1} [W]] = W.
\]

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 in \( \mathbb{R}^3 \). \( W_p \) and \( W_{p+1} \) denote two successive spatial positions of the tool path and \( R_p = (a_p, b_p) \), \( R_{p+1} = (\alpha_p, \beta_p) \) 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 \( M_p \equiv (X_p, Y_p, Z_p) \) and \( M_{p+1} \equiv (X_{p+1}, Y_{p+1}, Z_{p+1}) \).

Namely, \( M_p \equiv K^{-1} \{ R \} \{ W_p \} \). Second, the rotation angles \( \Re = \Re(t) = (a(t), b(t)) \) and the machine coordinates \( M = M(t) = (X(t), Y(t), Z(t)) \) are assumed to change linearly between the prescribed points, namely,

\[
M(t) = tM_{p+1} + (1-t)M_p, \quad \Re(t) = t\Re_{p+1} + (1-t)\Re_p, \quad \text{where} \quad t \text{ is the fictitious time coordinate } (0 \leq t \leq 1).
\]

Finally, transforming \( M \) back to \( W \) for every \( t \) yields a trajectory of the tool tip in the workpiece coordinates given by

\[
W_{p,p+1}(t) = K \{ R \} \{ M(t) \} = K \{ t\Re_{p+1} + (1-t)\Re_p \} \{ tM_{p+1} + (1-t)M_p \}.
\]

In order to represent the tool path in terms of the workpiece coordinates we eliminate \( M_p, M_{p+1} \) by using the inverse transformation \( W_p = K^{-1} \{ R \} \{ W_p \} \).

Substituting, \( M_p, M_{p+1} \) yields

\[
W_{p,p+1}(t) = K \{ t\Re_{p+1} + (1-t)\Re_p \} \{ tK^{-1} \{ R \} \{ W_{p+1}(t) \} + (1-t)K^{-1} \{ R \} \{ W_p(0) \} \}.
\]

The above formulae applicable to an arbitrary machine configuration allows for evaluating the kinematics error defined as the difference between the desired and the actual trajectory.

In particular, we consider kinematics of the five-axis milling machine MAHO600E depicted in Fig.5. The inverse kinematics are represented by matrices \( A = A(a(t)), B = B(b(t)) \) associated with the rotations around the primary and the secondary axes respectively, namely,

\[
K^{-1} \{ R \} \{ W \} = [h(t)] [W + T_{12} + T_{23} + T_{34} - T_L].
\]

where \( T_{12}, T_{23}, T_{34} \) are respectively the coordinates of the origin of the workpiece in the rotary table coordinates, coordinates of the origin of the rotary table coordinates in the tilt table coordinates and the origin of the tilt table coordinates in the cutter center coordinates, \( T_4 = (0, O, -T_L) \), where \( T_L \) is the tool length.

The set of the optimized parameters \( \mathbf{M} \) comprises of

1) Two rotation matrices \( \mathbf{M}_R = \{ A, B \} \) corresponding to the two rotary axes.
2) A workpiece setup \( \mathbf{M}_S = \{ T_{12}, r_a, r_b \} \).
3) Two translations associated with the position of the workpiece and the design of the five-axis machine \( \mathbf{M}_T = \{ T_{23}, T_{34} \} \).
4) The length of the tool \( L \). Since we consider the tool aligned with the \( z \)-axis, \( L \) is treated as an additional translation \( T_4 \) (\( T_4 \) is either \( (0, 0, L) \) or \( (0, 0, -L) \) depending on the direction of the tool tip in the spindle coordinate system). Furthermore, we include \( T_4 \) into \( \mathbf{M}_T \) so that \( \mathbf{M}_T = \{ T_{23}, T_{34}, T_4 \} \).

Now given fixed (but arbitrary) \( \mathbf{M}_R \) consider the following problem

\[
\text{minimize } \varepsilon, \quad \mathbf{M}_T \cdot \mathbf{M}_S.
\]
\[
\varepsilon = \sum_{P} \int_{0}^{1} (W_{p,p+1}^D - W_{p,p+1})^2 \, dt
\]

\[
= \sum_{P} \int_{0}^{1} (x_{p,p+1}^D - x_{p,p+1})^2 + (y_{p,p+1}^D - y_{p,p+1})^2 + (z_{p,p+1}^D - z_{p,p+1})^2 \, dt
\]

where \( \varepsilon \) is the total kinematics error. The optimization depends on the type of the five-axis kinematics categorized by relative positions of the rotary axis. We present numerical examples of optimization verified by means of computer simulation software [8] as well as by real machining.

Let us now analyze how the orientation of the workpiece changes the kinematics equations. Suppose the workpiece has been rotated clockwise by \( r_a \) around the x-axis and by \( r_b \) around the y-axis (Fig.7). The five-axis kinematics is affected by the rotations in a different way which depends on the type of the machine. Let \( r_a \), \( r_b \) be the rotation angles defining the orientation of the workpiece and \( R_a \), \( R_b \) the corresponding rotation matrices. Given the machine configuration in Fig.6, then the kinematics of the machine is given by

\[
M = B[b'](A[a')(W' + T_{12}) + T_{23}) + T_{41}.
\]

where

\[
W' = R_b[r_a]R_a[r_b]W
\]

\[
= \begin{bmatrix}
\cos(r_a) \cos(r_b) x + \cos(r_a) \sin(r_b) y - \sin(r_b) z \\
- \sin(r_a) x + \cos(r_a) y \\
\sin(r_a) \cos(r_b) x + \sin(r_a) \sin(r_b) y + \cos(r_a) z
\end{bmatrix}
\]

The conversion between tool orientation \( I = (I_x, I_y, I_z) \) and two rotary angles \( a \) and \( b \) is given by

\[
I_x = \cos(a) \cos(b),
I_y = \sin(a) \cos(b),
I_z = -\sin(b).
\]

After the rotations the tool orientation becomes

\[
I' = R_b[r_a]R_a[r_b]I.
\]

After some algebraic manipulations we have

\[
I' = \begin{bmatrix}
\cos(a - r_a) \cos(b) \cos(r_b) + \sin(b) \sin(r_b) \\
\sin(a - r_a) \cos(b) \\
\cos(a - r_a) \cos(b) \sin(r_b) - \sin(b) \cos(r_b)
\end{bmatrix}.
\]

Let us now find such \( r_a \), \( r_b \) and \( T_{12} \) that minimize the difference between a desired tool path \( W_{p,p+1}^D(t) \) between \( W_p \) and \( W_{p+1} \) and the actual tool path. Fig.8 and Fig.9 show the actual tool trajectories before and after the optimization. The optimal workpiece orientations have reduced the error of the entire surface up to 89% and at the largest loop up to 96% as shown in Fig.10, Fig.11 and Table 2.

![Fig.8: Workpiece orientation \( r_a = 0, r_b = 0 \)](image)

![Fig.9: Workpiece orientation \( r_a = 3.612, r_b = 1.673 \)](image)

The performance of the optimization is presented by Table 2. The results reveal that the average error and the largest error can be reduced as much as 89% and 96% respectively. We can interactively specify the combination of \( r_a \) and \( r_b \). The simulation has revealed that the appropriate combination of \( r_a \) and \( r_b \) can be chosen so that the kinematics error is minimized. Wrongly chosen the orientation may increase the errors as shown in negative percentage.
4. CONCLUSIONS

Minimization of the angle variation for rough cuts presented by iterative angle switching leads to a substantial increasing of accuracy. Our preliminary results show a 20% improvement over the pure angular insertion scheme and about 50% improvement with the reference to spatially equi-distributed points. Besides, for a rough cut the proposed scheme may increase the accuracy by factor 10. However, for fine cuts these advantages could be considerably reduced and the proposed scheme may even become unnecessary. The algorithms produce impressive results and therefore, can be used by the CAD/CAM industries to estimate the errors, to verify the tool path graphically and to generate and optimize the NC programs.

The workpiece orientation setup presents a new optimization of five-axis machining through positioning of the blank workpiece in the 3D working space. The proposed optimization algorithm is appropriate since it requires nothing else but changing the initial setup which is a very simple, zero cost operation and can be done interactively or automatically through the computer simulation. The optimal workpiece orientations have proven extremely effective in many cases displaying the accuracy increase up to 96%.

5. REFERENCES