Tool Orientation Optimization Considering Second Order Kinematical Performance of the Multi-Axis Machine

This paper presents a new tool orientation optimization approach for multi-axis machining considering up to second order kinematical performance of the multi-axis machine. Different from the traditional optimization approach, tool orientations are optimized with the goal of improving the kinematical performance of the machining process, not only increasing the material removal from purely geometrical aspect. The procedure is to first determine a few key orientations on the part surface along the tool path according to the curvature variation. Key orientations are initially optimized to be able to achieve high material removal by comparing the tool swept curve and the actual part surface. Intermediate orientations between key orientations are interpolated smoothly using rigid body interpolation techniques on SO(3). The time-optimal trajectory planning problem with velocity and acceleration constraints of the multi-axis machine is then solved to adjust the initially determined tool orientations to better exploit the multi-axis machine’s motion capacity. Simulation and experiment validate the feasibility and effectiveness of the proposed approach. [DOI: 10.1115/1.4002456]

Keywords: tool orientation, five-axis machining, rigid body interpolation, time optimal, machine’s motion capacity

1 Introduction

Five-axis machining centers are often adopted to manufacture parts with complex shapes or free form surfaces given their advantages such as high material removal, reduction in the number of workpiece setups, and better surface finish [1]. However, the additional freedom coming from the machine’s rotational axes brings much challenge for tool path planning and may induce kinematical and dynamical problems during the machining process. On the tool path planning aspect, tool path curves and tool orientations have to be carefully planned to avoid interference, increase local material removal, or reduce large angular variation. However, on the kinematical aspect, the tool tip trajectory during five-axis machining deviates from the straight line segment connecting adjacent cutter contact points. Nonlinear error is then needed to be estimated and compensated in the postprocessing process [2]. This compensation procedure may dramatically increase the amount of data processed in the controller, leading to frequent velocity fluctuations [3]. The motion limits (i.e., velocity, acceleration, and jerk limits) of the rotational axes also pose strict constraints on the orientation variation along tool paths. Without careful planning of the feed rate during machining planning, the motion limits may be violated and deficiency would be left on the part surface. In high speed machining (HSM) of complex parts, the dynamical stability problem is very crucial because of the high spindle speed and fast feed rate. Due to the complicated motion of the multi-axis machining process, determining cutting depth, spindle speed, and other stability-related factors is a very challenging task.

Various methods have been devised to resolve these problems to increase the efficiency and quality of multi-axis machining [4–16]. Rao et al. [5] presented the principal axis method, which determines the tool path through curvature matching. Chiu and Lee [6] introduced the potential field method, which maximizes the cutting width along the tool path. Makanov and co-workers [7–10] researched on the kinematics error during five-axis machining and proposed various approaches for suppressing this error, e.g., workpiece setup optimization, adaptive grid tool path generation, and rotation planning. Gray et al. [11] proposed a rolling ball method utilizing the graphic card power to fast generate an interference free tool path. Rao and Sarma [12] proposed a local gouging detection and elimination method for five-axis machining of a sculptured surface based on comparing the curvature of a tool swept surface and a part surface. Hosseinkhani et al. [13] proposed a penetration-elimination method for tool positioning with the ability of removing various types of gouging. In high speed rough machining, the trochoidal tool path is introduced [14,15]. This type of tool path is suited to the HSM situation because it more likely maintains a consistent feed rate. However, this type of tool path is mostly used in three-axis milling.

Feed rate planning and various spline interpolation methods are also presented to improve the machining efficiency and quality [1,17–20]. Erkorkmaz and Altintas et al. [17] proposed a quintic spline interpolation method that produces continuous profiles up to the acceleration level. Their method eliminates feed rate fluctuations caused by parametrization errors through recursively adjusting the increment of the spline parameter. Fleisig and Spence [1] extended the spline interpolation method to multi-axis machining in which a near arc-length parametrized quintic spherical Bezier spline is used to control the tool orientation. The time-optimal trajectory planning method, which has been studied extensively in robot manipulator path tracking, is also employed in planning the five-axis machining feed rate where the torque constraints are usually substituted by velocity, acceleration, and jerk constraints [21–23].

There is relatively less research in which geometric parameter (tool path, tool orientation, workpiece setup, etc.) planning and
In this paper, we focus on devising an optimization strategy for tool orientations on a predetermined tool path. The predetermined tool path can be generated using any of current available tool path generation methods, e.g., isoparametrical tool path, isoscallop tool path, and isoplanar tool path. The initially generated tool path is usually feasible but far from being optimal. Our goal is to generate a series of tool orientations along the tool path, which yields efficient multi-axis machining operation, while geometrically large material removal can be reached. To reach this goal, a non-linear programming problem is first solved in search of improved tool positioning that can achieve large material removal. The time-optimal trajectory planning problem considering the machine’s motion capabilities is then iteratively solved to further improve tool orientations to make the machining process more efficient.

This paper is organized as follows: Section 2 explains how geometrically optimal tool orientations with large local material removal are obtained. In Sec. 3, drive limits of multi-axis machines during the machining process are considered. This leads to the formulation of a constrained optimization problem, with the objective of reducing total machining time. Section 4 presents the simulation and experimental results of our method. Section 5 demonstrates the effectiveness of our method by making comparisons and explains some implementation issues. Section 6 concludes the paper.

2 Initial Orientation Determination Based on Maximizing Local Material Removal

2.1 Initial Tool Orientations. The initial tool orientations are derived based on purely geometrical concern. Namely, the following functional optimization problem needs to be solved for \((\alpha(s), \beta(s))\):

\[
\min_{\alpha(s), \beta(s)} \left( \text{VOL}_{\text{rem}} - \text{VOL}_{\text{swept}} \right)
\]

such that \(\Phi(\text{VOL}_{\text{swept}}) \geq \delta\) \hspace{1cm} (1)

where \(s\) is the parameter describing the tool path curve \(\gamma(s)\); \(\alpha(s)\) and \(\beta(s)\) are the inclination and tilt angles of the tool relative to the part surface at position \(s\); and \(\text{VOL}_{\text{rem}} - \text{VOL}_{\text{swept}}\) denotes the result volume obtained by subtracting the material that needs to be removed, \(\text{VOL}_{\text{rem}}\), from the volume swept by the tool along the tool path \(\text{VOL}_{\text{swept}}\). \(\Phi(\cdot)\) returns the signed distance from each point in space to the ideal part surface. \(\Phi(\cdot) \geq \delta\) requires that no overcut larger than the specified inner tolerance \(\delta\) happens.

Solving problem (1) is computationally expensive. We introduce the concept of interpolation tool axis (ITA) to transform the original problem to a parameter optimization problem. ITAs are a set of tool axis at sampled positions of the tool path. Once the ITAs are determined, the tool orientations at other positions of the tool path can be obtained by linear, cubic, or any other type of rigid body interpolation method [25]. The orientations of ITAs are determined based on the relative position of the tool swept curve at the sampled positions with the ideal part surface. There are two requirements for this positioning:

- First, the material, as large as possible, is removed at each local region. This requirement is transformed into two different objectives for two main types of milling, respectively: (a) For point milling, the cutting width orthogonal to the feeding direction should be maximized (see Fig. 1); (b) for flank milling, the tool is orientated as close as possible to the part surface, or the maximal deviation between the tool swept curve and the part surface is minimized (see Fig. 2).
- Second, the tool has no collision or local gouges with either the part surface or the environment.

The optimization problem can now be formulated as (for point milling)

\[
\max_{\alpha_i, \beta_i} \sum_i W(l_i, S)
\]

such that \(\Phi(l_i) \geq \delta\) \hspace{1cm} (2)

where \(\{(\alpha_i, \beta_i), i=1, 2, \ldots, m\}\) is a set of parameter pairs specifying ITAs, \(l_i\) is the \(i\)th swept curve, and \(S\) is the ideal part surface. \(W(\cdot)\) is defined as

\[
W(l_i, S) = \left| c'_i - \left[ c'_i \cdot (p_i - p_{b0}) \right] \right|
\]

where \(c'_i\) and \(c_i\) are two intersection points between the \(i\)th swept curve and the material surface. \(p_i - p_{b0}\) approximates the orthogonal direction of tool feed using the next and current cutter location points \(p_i\) and \(p_{b0}\).
Fig. 3 The relationship between the inflection point on the tool path, the position where B axis changes direction, and the cutting mark on the blade surface.

For flank milling, the objective function for initial positioning is

\[
\min \sum_{i} ||b_i - S_0||_2
\]

such that \( \Phi(t_i) = \delta \)  \hspace{1cm} (4)

where \( ||b_i||_2 \) returns the maximal deviation between the tool swept curve and the part ideal surface.

The placement of ITAs is adaptive to the curvature variation along the tool path so that between neighboring ITAs, the differential properties would not change significantly. The detailed procedure for the placement of ITAs will be discussed in Sec. 2.2.

2.2 Placement of ITAs. A simple strategy for the placement of ITAs is to select several places evenly distributed on the tool path. The optimization process presented in the later section will always be trying! to reduce machining time by adjusting the initial tool orientation distribution. However, it is found by researchers that tool path across places possessing large curvature variation usually yields large machining errors and may induce kinematics problems [26]. Problems associated with large curvature variation are also identified in several experiments when conducting this research. Figure 3 shows a cutting mark appearing in one of our experiments near an inflection point on the tool path. A further analysis of the motion of the machine tool identifies that the B axis of the machine tool reverses its direction near this inflection point. It is reported in some research that such cutting mark may be caused by the actuator reversal [27]. Even if this cutting mark is not solely caused by the actuator reversal, a changing rotating direction would also bring complex kinematical issues, as explained in Ref. [28]. and would possibly increase machining time because of acceleration and deceleration processes. Therefore, it is reasonable for us to place ITAs according to curvature and curvature variation of the tool path to avoid large axis motion variation. Another reason for placing ITAs according to curvature analysis is based on interference avoidance concerns. In the traditional “relative to part surface” tool orientation determination method, the tool is required to maintain a constant orientation in the local coordinate frame (spanned by the feed direction, normal vector at the cutter contact point, and their cross product). When the tool passes a region with a large curvature without control, the tool shaft may swing too suddenly, causing collisions or gouging.

ITAs should also be placed where the neighboring environment changes, like a new neighboring feature appears along the tool path. This is for avoiding collision between the tool and the neighboring feature. For example, when machining a rotor blade, neighboring blades appear when the cutter enters/exits in/from the suction or pressure side. To avoid collision between the tool shaft and the neighboring blades or the surface being machined, ITAs should be inserted at some where the tool enters the space between two blades. A detailed procedure is listed below for the placement of ITAs.

1. Define a threshold for curvature variation and estimate curvature and curvature variation at each cutter contact point.
2. Based on the predefined threshold, partition the tool path into multiple parts and insert ITAs at starting and ending positions of each part of tool path where the curvature variation is large.
3. Determine positions along the tool path where new neighboring features appear, or possible collision may occur and insert ITAs.

The first and second steps are simple and fast processed because estimating curvature is a necessary step when performing discretization of the tool path. As the third step depends on the typical part being machined, it may not be able to be processed automatically by the tool path optimization program.

It is usually necessary to make some simplification to increase computational efficiency. For a part with complicated features, which needs many sweeping paths to cover the whole surface (i.e., a zigzag tool path pattern is employed), we select several tool paths, which contain complicated moves and determine the position and orientation of ITAs on these paths according to the procedure presented previously. A rigid body interpolation technique is then used for generating ITAs on other tool paths. Namely, there are two interpolations needed: first, interpolating ITAs between preselected tool paths to obtain ITAs on all tool paths and then interpolating tool orientation along each single tool path to obtain all tool orientations at cutter contact points (see Fig. 4).

For spiral shaped or helical shaped tool path patterns, we may select several cycles from the total tool path, which contain complicated moves and use the same two-step interpolation procedure for generating all the tool orientations.

2.3 Interference Avoidance. In a previous discussion, the orientations of the tool other than ITAs are obtained through interpolation without considering possible interference between the interpolated tool and the part surface. To deal with this problem, once the interpolated tool interferes with the part surface, the interfering tool orientation \((\alpha_{\text{interfering}}, \beta_{\text{interfering}})\) is adjusted to its nearest tool orientation \((\alpha_{\text{sub}}, \beta_{\text{sub}})\), which is interference free. Namely, \((\alpha_{\text{sub}}, \beta_{\text{sub}})\) is the solution of

\[
\arg \min ||[\alpha_{\text{interfering}}, \beta_{\text{interfering}}] - [\alpha, \beta]||_2
\]

such that \([\alpha, \beta] \in \Omega_{\text{free}}\)

where \(\Omega_{\text{free}}\) represents the collection of all tool orientations free of interference.

There are two steps for solving problem (5). The first step is to obtain \(\Omega_{\text{free}}\). Identifying \(\Omega_{\text{free}}\) is actually to determine the accessibility cone of the tool at one cutter contact point, which can be quickly done using computer’s GPU power, as shown in Ref. [29]. Calculating problem (5) would then be to extract one tool orientation from a collection of candidate tool orientations within the accessibility cone possessing the smallest Euclidean distance \(||[\alpha_{\text{interfering}}, \beta_{\text{interfering}}] - [\alpha, \beta]||_2\).
tained purely geometrically. In this section, the geometrically optimized tool orientations are further improved according to some index relating to the machine’s kinematical performance. Figure 5 shows the relationship between the tool orientation and corresponding machine kinematical configuration. The following steps are performed.

First, the machining time using the current tool path and tool orientations obtained from Sec. 2 working under maximum motion capacity of multi-axis machine is estimated. We refer to this result as minimal machining time \( T_{\text{MMT}} \). The tool orientations are then adjusted continuously until the least MMT is achieved. This result is referred to as least minimal machining time \( T_{\text{LMMT}} \). The result tool orientations corresponding to the LMMT are considered as having maximally exploited the motion capacity of the multi-axis machine. Namely, we seek solution for the following optimization problem:

\[
\begin{align*}
\min_{\alpha(s), \beta(s)} & \quad T_{\text{least}} \\
\text{such that} & \quad |\dot{q}| \leq q_{\max} \\
& \quad |\ddot{q}| \leq q_{\max}
\end{align*}
\]

where \( \dot{q}(s) \) and \( \ddot{q}(s) \) are velocity and acceleration vectors of the multi-axis machines, \( q_{\max} \) and \( q_{\max} \) are maximally allowed velocity and acceleration of each axis, \( T_{\text{least}} \) is the MMT required to finish the operation given tool orientation \( (\alpha(s), \beta(s)) \). Similarly, to reduce the computational cost, only tool orientations (i.e., ITAs) at several sampled positions are optimized, and tool orientations at other places are interpolated using rigid body interpolation techniques.

### 3.1 Estimating Minimal Machining Time

The optimization problem (6) requires estimating the minimal time to finish a machining operation, i.e., \( T_{\text{least}} \) given \( (\alpha(s), \beta(s)) \) at each iteration step. Given the tool path and tool orientations, estimating the machining time under maximal motion capacity of multi-axis machines requires solving the time-optimal trajectory problem, which has been addressed in a number of references in the robotics community. A method proposed in Ref. [30] is adopted to resolve the machining time estimation problem, which can yield a global optimal solution.

---

**Fig. 4** Determining ITAs on zigzag and spiral shaped tool path pattern

**Fig. 5** Relationship between the tool orientation and corresponding machine kinematical configuration
When tool orientations are provided during solving Eq. (6) at each iteration step, \( T_{\text{least}} \) can be obtained by solving
\[
\min \int_0^1 \frac{1}{s} ds
\]
such that \( |q'(s)s| \leq \dot{q}_{\text{max}} \)
\[
|q'(s)s + q''(s)s^2| \leq \ddot{q}_{\text{max}}
\]
(7)
\[
\dot{s}(s) = 0
\]
\[
\dot{s}(0) = 0
\]
\[
\dot{s}(1) = 0
\]
where we assume the path coordinate \( s \in [0,1] \), and at start and end positions the velocity of the tool is zero.

Using variable substitution \( a(s) = \dot{s}, \quad b(s) = s^2 \) as in Ref. [30]. Note that \( b'(s)s = b(s) + 2\dot{s}s = 2\dot{s}s = 2\dot{a}(s) \Rightarrow b'(s) = 2\dot{a}(s) \), Eq. (7) can be rewritten as
\[
\min \int_0^1 \frac{1}{b(s)} ds
\]
such that \( q'(s)b(s) \leq \dot{q}_{\text{max}} \)
\[
|q'(s)a(s) + q''(s)b(s)| \leq \ddot{q}_{\text{max}}
\]
(8)
\[
b(s) = 0
\]
\[
b(0) = 0
\]
\[
b(1) = 0
\]
\[
b'(s) = 2\dot{a}(s)
\]
where after variable substitution, the velocity constraint is expressed using its equivalent squared form to maintain the convex property [31]. The convexity of problem (8) ensures that the global solution can be obtained.

3.2 Achieving High Material Removal Versus High Feeding Speed. Achieving maximal material removal at local regions and achieving the highest feeding speed may compromise each other. Namely, when tool orientations are modified to achieve high feeding speed, the local material removal may be reduced or vice versa. To deal with this situation, the geometrically optimal tool orientations can be used to define an allowed interval within which the optimization problem (6) is solved. The interval width could be large if the user emphasizes more on kinematical performance or small if the user cares more about material removal. One feasible strategy is proposed here.

1. Use any existing tool path generation method to generate tool path. Tool orientations at sampled positions along the tool path are denoted as \( (\alpha_i, \beta_i), \quad i = 0, 1, \ldots, k \), where \( k \) is the total number of sampled positions.
2. Determine geometrically optimal tool orientations \( (\alpha_i, \beta_i), \quad i = 0, 1, \ldots, k \), by solving problem (2) or problem (4). The original tool path is now improved to be able to achieve a larger local material removal.
3. Use the results in steps (1) and (2) to define an allowed varying interval for tool orientations for further kinematical performance improvement. Namely, we add constraints for problem (6):

4 Numerical Simulation and Experiment
The tool orientation optimization method is implemented to machine a blade surface on a Mikron HSM600U five-axis machining center equipped with a Heidenhain iTNC530 control unit. The velocity limits of the machine are 40 m/min for X, Y, and Z axes and 360 rpm for B and C axes. The acceleration limits are 10 m/s² for X, Y, and Z axes and 30 rad/s² and 130 rad/s² for B and C axes. The model of the blade surface is shown in Fig. 6. The tool path is generated helically around the blade surface from top to bottom (see Fig. 7).

4.1 Determination of ITA. A blade surface consists of its suction part, pressure part, and two edge parts. Figure 8 illustrates one cycle of the tool path around the blade surface and the curvature distribution using a comb plot. It can be seen that the curvature varies largely near two edges, and there are inflection points at the pressure part of the tool path curve. For the blade surface example, the curvature does not vary much in the direction orthogonal to the tool path curve. Positions where ITAs are placed are marked using black round circles. ITAs placed at the two corners of the blade surface are fixed considering that interference between the tool holder and the blade surface may occur at these regions. The ITAs are placed similarly at two other cycles. Considering the periodical characteristics of the tool path, ITAs between the selected cycles are obtained using rigid body interpolation techniques.
4.2 Simulation and Experiment Results. Problem (1) and then problem (6) are solved to determine the optimal tool orientation series in machining the rotor blade. Due to the periodic characteristics of the machining operation (helically engaged tool path), we only take 15,000 tool path segments in our simulation to save the computational cost. Figures 9–11 show, when working under its maximal motion capacity along the trajectory after optimization of tool orientations, the B/C axes’ position, velocity, and acceleration profiles of the five-axis machines. Note that the B and C axes of the five-axis machine are strictly connected to the variation in the tool orientation. Figure 12 shows the final tool orientations obtained. Only a portion of all tool orientations is displayed in order to present a clear visual effect. Figure 13 shows the machining result using tool orientations obtained by the method proposed in this paper.

5 Discussion

5.1 Comparison of Machining Efficiency. Comparisons are made between the method proposed in this paper with the traditional “relative to drive” tool orientation determination method. Figure 14 shows the comparison result, which estimates machining time by solving problem (8). The seven sets of tool orientations used for comparison with the proposed method approximately cover a feasible tool orientation range that can be obtained using the relative to drive method. It is found theoretically that at least 35% of machining time can be saved using optimized tool orientations obtained by the proposed method in this paper.

The actual machining time for three sets of experiments is estimated from a video recorded during experiments. As the tool helically engages from the top to the bottom of the blade surface, the machining times used to finish ten cycles before optimization are approximately 55 s and 59 s, respectively, while the time cost after optimization is approximately 43 s. Such increase in efficiency can be understood by analyzing the G code used for machining (see Fig. 15). Before optimization, the B axis changes its direction twice within one cycle of the tool path, while such actuator reversals disappear after tool orientations are optimized. The elimination of direction changes would thus reduce the acceleration and deceleration process and shorten the total machining time. Besides, the motion range of the B axis is significantly reduced (from −98 deg to −80 deg to −95 deg to −88 deg) after optimization. During experiment, it is also observed that there is no sudden change in the tilt and rotary table, which happens when using the unoptimized tool orientations.

5.2 Influence of the Placement of ITAs. The placements of ITAs have an influence on the machining performance both on quality and efficiency. In one of our experiment, an interpolation tool axis at the edge of the blade surface is omitted. The machining result shows vibration waves, as seen in Fig. 16. The poor surface may be due to two reasons: (a) variation in the tool axis around the blade edge is too large so that the tooth of the cutter may have scratched on the blade surface; (b) when the tool passes...
the inflection point, the B axis of the machine tool experiences actuator reversals, which induce unexpected vibration. The efficiency is also influenced. When the tool passes through the corner with a large variation, the rotary and tilt table of the five-axis machine undergoes large motion steps, which may exceed the motion capacity of the machine tool. In our experiment, a sudden movement of the machine tool is observed, and the feed rate has to be set to a lower value to ensure safety.

5.3 Computational Efficiency. Currently, the algorithm of obtaining MMT by solving Eq. (8) is stable and efficient due to its convexity. But there is a lack of efficient strategy to update the tool orientation after estimating the MMT to find a solution of problem (6). The simulation based method is used now in this research. The allowed varying intervals of tool orientations are discretized, and a finite set of ITA combinations is thus obtained.

Fig. 11 (a) B/C acceleration profile. (b) B/C acceleration profile: a locally enlarged view.

Fig. 12 Tool orientation distribution after optimization (only part of all orientations is plotted for visualization concerns)

Fig. 13 The machining result of rotor blade

Fig. 14 Comparison between the method proposed in this paper and the relative to drive method

Fig. 15 The motion of B axis before and after optimization
Each combination is simulated independently. Such searching procedure can be iteratively performed using the last obtained best tool orientations to define a smaller searching zone. Figure 17 illustrates this idea. The simulation based method is acceptable for practical usage considering that the problem can be offline calculated and parallel computing techniques can be employed. For the current experiment and simulation, to obtain a reduction 35% of machining time, the computational cost for problem (6) is approximately 6 h on a laptop with an Intel Core2 1.83 GHz CPU, 1 Gbyte random access memory (RAM). MATLAB is used for programming.

5.4 Influence of Tool Orientation Adjustment on Scallop Height Constraint. Considering machining a precise part such as a rotor blade as in our experiment, a ball end mill is usually employed. In such a case, the adjustment of tool inclination and tilt angle will not break the scallop height constraint, which can be seen from the side step calculation equation (where the normal curvature of the tool is constant) [24].

$$w = \sqrt{8h_{\text{scallop}}/\kappa_{\text{ball}}} = \kappa_t (n \times t)$$

where \(w\) is the side step, \(\kappa_{\text{ball}}\) is the curvature of the ball end mill, \(\kappa_t (n \times t)\) is the normal curvature of the surface in the direction orthogonal to the feeding direction, and \(h_{\text{scallop}}\) is the prescribed scallop height.

It would be also possible to include the scallop height constraint in the optimization process. This could be realized using two ways. Based on the analysis of Refs. [32,33], the local cutting width can be easily estimated using an approximation method with an analytical equation. Even in the worst case for precision concern, the local cutting width can be calculated using a swept curve, as explained in Sec. 2. Therefore, during the optimization process, the cutting width \(w\) can be checked to see if it is less than the cutting width of the original tool orientation \(w_o\) obtained using any existing tool path generation method (which satisfies the scallop height). In case \(w_o \geq w\), the tool orientation will be adjusted to the nearest orientation, which ensures that the cutting width is larger or equal to the original cutting width. Noting that the initial geometrical optimization of tool orientations has ensured a larger cutting width at the local region, which means that the scallop height has been reduced compared with the original tool path obtained using any existing tool path generation method (which satisfies the scallop height constraint), the possibility of breaking the scallop height constraint within a predefined tool orientation adjustment region, as explained in Sec. 3, is usually small. Namely, it is possible to define a small region for tool orientation variation to respect the scallop height constraint.

Another way is that we just insert additional tool paths to ensure the scallop height constraint when adjusting the tool orientation; the machining time for the additional tool paths will also be included in calculating the MMT during optimization. The time consumed by machining the additional paths will thus penalize the solution.

6 Conclusion

This paper proposes a tool orientation optimization approach considering up to second order kinematical performance of the multi-axis machine. The tool is first positioned to achieve high material removal in local regions. In the second phase, an iterative search procedure is performed to search for tool orientations, which can better exploit the machine’s motion capacity. The time-optimal trajectory planning problem is solved at each iterative step in the second phase considering velocity and acceleration limits of the multi-axis machines. The time-optimal problem is reformulated, and a global optimal solution can be achieved. Simulation and experiment demonstrate the feasibility and effectiveness of the proposed method. The benefits of the proposed method are as follows.

- The result tool orientations ensure that the motion capacity of the multi-axis machines is better exploited.
- The feed rate obtained after solving the optimization problem (7) can be used to modify the G code to achieve higher machining efficiency.

Although the local material removal might be reduced during the second phase of tool orientation optimization, the varying interval specified can be used to control the compromise between the two conflicting goals. Moreover, the feeding speed usually plays a much more important role in high speed machining of parts such as blade surfaces during the finishing stage. Therefore, the proposed method still yields a highly efficient tool path.

The tool path considered in the simulation and experiment is smooth and spirals down from the top to the bottom of the blade surface. If a nonsmooth tool path appears, it is also possible to estimate the velocity and acceleration vectors using finer discretization of CC curves when calculating the MMT to approximate the local velocity vector. Besides, it is often suggested using
smooth moves instead of sharp turns in the tool path because a nonsmooth transition would possibly induce vibration of the machine tool. Such strategy is currently implemented in most CAM software (see Fig. 18 for an example).

Acknowledgment

This work is supported by the 973 Research Foundation of China (Grant No. 2005CB724100, and No. 2011CB706800) and the National Natural Science Foundation of China (Grant No. 50835004).

Nomenclature

\( \alpha(s), \beta(s) = \) new optimization variables after variable substitution
\( c_1, c_2 = \) two intersection points between the \( t \)th swept curve and the material surface
\( g(t) = \) tool configuration at interpolated position \( t \)
\( h_{\text{scallo}}p = \) scallop height
\( k_{(n \times t)} = \) normal curvature of the part surface in the direction orthogonal to the feeding direction
\( \kappa_{\text{ball}} = \) curvature of the ball end mill
\( l_1 = \) \( t \)th swept curve
\( N_1 = \) tool surface normal in the tool frame
\( n_1, n_2 \in \mathbb{R}^3 = \) tool orientation vectors at \( p_0 \) and \( p_1 \)
\( p_0, p_1 = \) adjacent cutter location coordinates
\( q, q, q = \) axis position, velocity, and acceleration of the five-axis machine
\( q_{\max}, q_{\max} = \) velocity and acceleration limits of the motion axes of the five-axis machine
\( R_0, R_1, R = 3 \times 3 \) matrix representing the tool orientation
\( R, r = \) big and small radii of the fillet end mill
\( S = \) ideal part surface
\( s = \) tool path curve parameter
\( T(\cdot) = \) parametrical equation of tool surface
\( T_{\text{least}} = \) MMT required to finish the machining
\( V = \) motion screw of the tool
\( \text{VOL}_{\text{mn}} = \) material needed to be removed
\( \text{VOL}_{\text{sweep}} = \) volume swept by the tool along the tool path
\( v_i = \) velocity of a point on the tool surface in the tool frame
\( W(\cdot) = \) function returns the cutting width
\( w = \) cutting width
\( w_0 = \) original cutting width using any existing tool path generation method
\( \alpha, \beta = \) orientation angles of the tool relative to the part surface
\( \alpha(s), \beta(s) = \) tool orientations along the tool path
\( \alpha_i, \beta_i = \) tool orientations at the \( t \)th sampled position
\( \alpha^0_i, \beta^0_i = \) initial tool orientation along the tool path at the \( t \)th sampled position
\( \alpha^g_i, \beta^g_i = \) geometrically optimal tool orientations at the \( t \)th sampled position
\( \xi_1, \xi_2 = \) two orthogonal vectors used to constitute the local moving frame of the tool
\( \delta = \) machining inner tolerance
\( \theta(\theta_1, \theta_2) = \) tool surface parameter vector
\( \Omega_{\text{free}} = \) the collection of all tool orientations free of interference
\( \Phi = \) signed distance function
\( \gamma(s) = \) tool path curve represented using a single parameter \( s \)

Appendix: Tool Swept Curve Calculation at Sampled Positions

A swept curve, which is the boundary of the solid swept by the moving tool, can be obtained using the tangential condition \([34,35] \).

\[
N_i \cdot v_i = 0 \tag{A1}
\]

where \( N_i \) is the normal at points of the tool surface expressed in the tool frame and \( v_i \) is the velocity of points on the tool surface in the tool frame.

We assume that at every sampled cutter contact point (CC) and its neighboring point (CC') the relative orientation between the tool and the part surface is almost identical because the sampled positions are adaptive to the curvature variation along the tool path and two adjacent cutter contact points are usually very close to each other.

Let \( p_0, p_1 \in \mathbb{R}^3 \) be the adjacent cutter location point coordinates (CL coordinates), where \( p_0 \) corresponds to a selected sampled position, and \( n_0, n_1 \in \mathbb{R}^3 \) be the corresponding tool orientation vectors at \( p_0 \) and \( p_1 \). Subscripts 0 and 1 represent two adjacent cutter locations in a later discussion.

\( p_0 \) and \( n_0 \) are known once the tool path curve is generated, and tool orientation at the sampled position is provided during solving problem (2) or problem (4). \( p_1 \) and \( n_1 \) are then derived based on the assumption of identical relative orientations. When the tool moves from \( p_0 \) to \( p_1 \), the CL coordinates are linearly interpolated,

\[
p(t) = (1-t)p_0 + tp_1 \tag{A2}
\]

where 0 is interpolation parameter.

Calculation of the tool swept curve also requires intermediate tool orientations between two cutter location points. Define \( R = [\xi_1, \xi_2, n] \in \text{SO}(3) \), where \( \xi_2 = n \times (p_1 - p_0) \) and then \( \xi_1 = \xi_2 \times n \).

At each position 0, the tool orientation can be obtained through solving \([25] \)

\[
R(t) = R_0 \exp(t \log(R_1^{-1} R_0)) \tag{A3}
\]

The tool orientation vector is just the third column of the \( R(t) \).

Denote \( g(t) = [R(0), p(t)] \), which represents tool configuration at 0. The screw of the tool is \( V = g(t)^{-1} g(t) \). Denote the tool surface parametrical equation as \( T(\theta)(\theta_1, \theta_2) = \) (the parameter vector of the tool surface), the normal of which is \( N(\theta_1, \theta_2) = d\theta_1 \times d\theta_2 \times d\theta_2 \). Therefore, the velocity of any point on the tool surface under the tool coordinate frame can be calculated as

\[
v_i = V[T \ 1]^T \tag{A4}
\]

Substituting Eqs. (A2) and (A3) into Eq. (A4) and then using the swept curve definition (Eq. (A1)), we may obtain the solution of swept curves for the tool undergoing linear varying motion between two adjacent cutter location points 0,

\[
F(t, 0) = (CT(\theta) + (\exp(tC))R_1^{-1}(p_1 - p_0)N(\theta)) \tag{A5}
\]

where \( C = \log(R_1^{-1} R_0) \in se(3) \) is skew symmetric. Let
Taking flat, ball, and fillet end mills as examples, we derive the closed form solution of $F=0$ (see Fig. 19). The result is listed in Table 1. Note that for ball and fillet end mills, only the solution of the ball part and the torus part is provided because the cylindrical part solution can be similarly derived as for the flat end mill.

### Table 1 Closed form solution of swept curves for three types of milling cutters

<table>
<thead>
<tr>
<th>Type</th>
<th>Parametric equation</th>
<th>Velocity $\mathbf{v}_i$</th>
<th>Normal $\mathbf{N}_i$</th>
<th>$F=0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flat end mill</td>
<td>$\begin{bmatrix} R \cos \theta_1 \ R \sin \theta_1 \ \theta_2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} -\omega_3 R \sin \theta_1 + \omega_2 \theta_2 + \omega_1 \ \omega_3 R \cos \theta_1 - \omega_1 \theta_2 + \omega_2 \ -\omega_2 R \cos \theta_1 + \omega_1 \theta_2 + \omega_3 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \cos \theta_1 \ \sin \theta_1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \theta_1 = \tan^{-1} \frac{\omega_3 + \omega_2 \theta_2}{\omega_2 - \omega_1 \theta_2} \ \theta_2 = \tan^{-1} \frac{\omega_3}{\omega_1} \end{bmatrix}$</td>
</tr>
<tr>
<td>Ball end mill</td>
<td>$\begin{bmatrix} R \sin \theta_1 \cos \theta_2 \ R \sin \theta_1 \sin \theta_2 \ -R \cos \theta_1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} -\omega_3 R \sin \theta_1 \sin \theta_2 - \omega_2 R \cos \theta_1 + \omega_1 \theta_2 + \omega_2 \ \omega_3 R \cos \theta_1 \sin \theta_2 + \omega_1 \theta_2 + \omega_2 \ -\omega_2 R \sin \theta_1 \cos \theta_2 + \omega_1 \theta_2 + \omega_3 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \sin \theta_1 \cos \theta_2 \ \sin \theta_1 \sin \theta_2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \theta_1 = \tan^{-1} \frac{\omega_3}{\omega_1 \cos \theta_2 + \omega_2 \sin \theta_2} \ \theta_2 = \tan^{-1} \frac{\omega_3}{\omega_1 \cos \theta_2 + \omega_2 \sin \theta_2} \end{bmatrix}$</td>
</tr>
<tr>
<td>Fillet end mill</td>
<td>$\begin{bmatrix} (R + r \sin \theta_1) \cos \theta_2 \ (R + r \sin \theta_1) \sin \theta_2 \ -r \cos \theta_1 \end{bmatrix}$</td>
<td>$\begin{bmatrix} -\omega_3 (R + r \sin \theta_1) \sin \theta_2 - \omega_2 r \cos \theta_1 + \omega_1 \theta_2 + \omega_2 \ \omega_3 (R + r \sin \theta_1) \cos \theta_2 + \omega_1 \theta_2 + \omega_2 \ -\omega_2 (R + r \sin \theta_1) \cos \theta_2 + \omega_1 (R + r \sin \theta_1) \sin \theta_2 + \omega_3 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \sin \theta_1 \cos \theta_2 \ \sin \theta_1 \sin \theta_2 \end{bmatrix}$</td>
<td>$\begin{bmatrix} \theta_1 = \tan^{-1} \frac{\omega_3}{\omega_1 \cos \theta_2 + \omega_2 \sin \theta_2} \ \theta_2 = \tan^{-1} \frac{\omega_3}{\omega_1 \cos \theta_2 + \omega_2 \sin \theta_2} \end{bmatrix}$</td>
</tr>
</tbody>
</table>

References


