COMPUTER-AIDED DESIGN & APPLICATIONS, 2017
VOL. 14, NO. 4, 377–391
http://dx.doi.org/10.1080/16864360.2016.1257182

Optimal iso-planar cutting direction based on machine kinematic metric: A differential geometry method for freeform surface finishing tool path computation

Chen-Han Lee [a,b] and Changya Yan [b]

[a]Huazhong University of Science and Technology, China; [b]Wuhan Institute of Technology, China

ABSTRACT
While machining width is an important factor of the machining time of freeform surface finishing operations, in reality the kinematic capability of the machine tool is usually the bottleneck of achieving higher feed speed and optimal machining time. The purpose of this paper is to conveniently (and approximately) determine the optimal cut direction considering the speed kinematic capability of the machine tool, without having to compute the actual tool path. We propose a mathematical instrument, called Machine Kinematic Metric (MKM), to easily evaluate infinitesimal machining time on a freeform surface based on machine kinematic consideration. It's a tensor field similar to the metric tensor in differential geometry. MKM is integrated over the part surface to approximate the cut-direction-dependent total machining time, and used to determine the optimal cut direction that minimizes the machining time. To validate the accuracy of the prediction using MKM, we apply the method and compute the machining time at every direction with one degree apart and derive the optimal cut-direction. The computation is performed on two examples: a simple freeform surface and a complex die face model. We then use a commercial CNC emulator software from Huazhong CNC to precisely simulate the machining time in distributed cut directions (five degree apart) for the two models. We find that the optimal cut direction determined from CNC simulation is consistent with the prediction from the proposed method. It validates that the proposed method is a convenient and economical tool to approximately determine the optimal cut direction based on machine speed kinematic capability.

KEYWORDS
Freeform surface machining; Machine kinematic metric; Optimal cutting direction; Machine kinematics; Differential geometry

1. Introduction

Freeform surface machining executed by Computer Numeric Controlled (CNC) machine tools plays an important role in the process of bringing new products to the market. Many production parts, from automotive body panels, consumer electronics, to plastic toys, are made out of dies and moulds in mass-production manufacturing. A typical freeform surfaces is defined as B-spline surface or tessellated model [17], with arbitrary trimmed boundary. Die and mould faces usually consist of freeform surfaces and hence freeform surface machining is critical to die/mould making, especially the finishing work step, and the mass production using die/moulds.

Freeform surface finishing processes, including that of die/mould face, are measured in two aspects: the quality of the parts and the efficiency of the process. The part quality requirements include accuracy (shape precision) and surface quality (roughness). The process efficiency is measured in machining time and tool/machine wear. Given the available equipments (machine tools, cutters, and controllers) in the shop floor, the cutter path (toolpath) used to perform a machining plays the major role in determining the quality and efficiency of the freeform surface finishing process.

Computer Aided Manufacturing (CAM) technology is widely used to obtain quality tool paths (Cutter Location trajectories) that are further post-processed to CNC machining programs (G-Codes). Iso-planar (parallel or pseudo-parallel) tool paths are most widely used for freeform surface finishing, due to the advantage of stable tool paths and predictable results. Typical commercial CAM systems compute iso-planar tool paths based on user-input process parameters, including scallop height, path tolerances, and cut direction (a.k.a. feed direction). Each one of these process parameters affects the above mentioned process quality and efficiency.

Among the process parameters, scallop height (related to machining width) and path tolerances (related to step size) are intuitive and can be determined easily. On the

CONTACT Changya Yan yanchangya@hotmail.com

other hand, the choice of cut direction is highly complicated. It has direct influence on process quality and efficiency (more on this in the following paragraphs) but the effect depends on the part shape and is difficult to analyse and predict. In current state-of-art, the selection of cut direction is mainly based on experiences and time-consuming try-and-improve. Modern CAM systems do not provide a solution to automate the cut direction selection. There is also no existing scientific method to validate if the chosen cut direction is optimal.

The requirements of part quality and process efficiency compete with each other. The common thought process is to first fix the part quality as a constraint then optimize the process efficiency. The reason is that the minimal requirement of quality cannot be violated and hence is a hard constraint. The remainder of this paper is limited to the optimization of process efficiency, namely the machining time, with a predetermined quality constraint.

Many factors affect total machining time and the two most important are machining width and machine kinematic capabilities. Wider machining width (without violating the surface roughness constraint) leads to less number of machining passes and shorter tool path length, which leads to shorter machining time. In addition, the machining time is related to how fast the machine tool axes can move. The instantaneous speed of the machine axes is limited by the kinematic capabilities, that is, the maximal (peak) velocity and acceleration load of each axis.

While machining width with geometric consideration is an important factor of machining time, in reality the kinematic capability of the machine tool is usually the bottleneck of achieving higher feed speed and optimal machining time. The feed speed is handicapped when the feed direction coincides with the slowest axis of the machine tool. In addition, the acceleration limit of the feed axes could also suffocate the feed speed especially when the dynamic property of the machine tool is mediocre.

Acceleration, jerk, and axis speed are the main aspects of the kinematic capability of a machine tool. Among these factors, we focus on the axis speed in our research. The jerk and acceleration limits of the machine are managed by the CNC internal interpolation and speed planning algorithms. Whatever we do in an offline optimization based on acceleration or jerk, the results are modified by the CNC. There are several kinematic reasoning methods from past literature, but we don't know which method is used by the CNC, or even if the commercial CNC follows any one of the published methods. Thus it's almost impossible to validate the offline acceleration or jerk optimization without knowing the internal algorithms of the CNC. We believe that the effective acceleration and jerk optimization must be done with the combination of offline and online (CNC) methods. Due to the amount of work, this paper is limited to offline optimization, considering only the axis speed limits.

The purpose of this paper is to compute the optimal cut direction considering the kinematic speed capability of the machine tool. Our goal also covers getting the answer (optimal cut direction) quickly without having to compute the tool path - computing tool paths is time consuming especially computing multiple tool paths for various cut directions is not feasible in practice. We derive the mathematical framework to formulate the machining time in terms of the axis speed capability and cut direction, and then compute the optimal cut direction that minimizes the machining time.

We must note the optimization of five-axis machining in general is a multi-objective problem, and influenced by great amount of factors (or constraints). These factors come from the workpiece geometry, tool path geometry properties, machine performances, and even costs. This paper aims to provide a new methodology to incorporating machine kinematics performance into tool path optimization, so simplify the formulation process, we only chose machining efficacy as a single objective, and speed limits of machine as constraints. In future work, the multiple objective optimization solution of five-axis machining with more constraints will be addressed.

The next section provides an overview of previous works that are related to the presented study. Sections 3 and 4 establish the main mathematical model, named Machine Kinematic Metric (MKM), for path trajectories and surfaces, respectively. MKM is a differential geometry-based mathematical instrument for computing machining time, like metric tensor to compute distance, hence the naming of "metric." Section 5 presents a numerical method derived from MKM to compute (approximate) the machining time on workpiece surfaces, while section 6 explain how to use this numerical method to compute machining time and derive the optimal cut direction on freeform surfaces. Section 7 presents two test cases being used to perform simulation experiments and validate the reliability and usability of the proposed method. The summary and conclusion are listed in the final section.

2. Related works

The relevant works published previously are organized into the following subsections for easy reference. Freeform surface machining is a general subject and there are vast number of papers in this area. In the first subsection, we made a partial list of the publications that are more closely related to our work. Optimal cut direction based on machining width is not the focus of our work but has similarity in terms of the methodology. We cover the relevant references in the second subsection. Machine kinematic capability optimization is our main focus and the references of previous contributions are listed in the third subsection. The mathematical instrument of fields, including scalar, vector, and tensor fields, is used in some of the previous works in freeform surface machining. The use of fields is essential to our approach and we present the field-based works in the fourth subsection. A summary and discussion of the listed references is given in the last subsection.

2.1. Freeform surface machining

Numerous researchers in the last two decades made significant contributions to the topic of tool path generation for free form surface machining. Traditionally iso-parametric and iso-planar methods have been used for the tool path generation [7]. Afterwards many methods were developed to improve the machining quality and efficiency, such as curved curvature matched machining [3], iso-phote based methods [6], configuration space methods [20],region based tool path generation [5], etc.

Many researchers worked on tool axis/orientation computation in 5-axis machining. The two rotary axes allow the flexibility to avoid collision/gouging and control the cutting condition during machining. However the additional freedom from the rotary axes makes the 5-axis machining much more complicated than 3-axis machining. Interference avoidance and smooth tool orientation transition are the two key subjects in five-axis tool path generation. Researchers reported C-space based tool orientation methods [13], rolling ball method (RBM) [9], arc intersect method (AIM) techniques [10], and tool orientation smoothing algorithms [1].

2.2. Machining width optimization

On top of tool path computation strategies and methods, it is important to improve the efficiency of machining. In general, the approaches to increase the machining efficiency in freeform surface machining could be classified into tool path length reduction and optimal feed speed assignment.

Wider machining strip width leads to less machining passes which also means shorter tool path length. Effective cutting shape (ECS) and effective cutter radius (ECR) were used to increase the machining strip width in tool path optimization [21]. They concluded that the increase of machining strip width is induced by the matching of ECR with the surface normal curvature. Other methods to obtain maximization of machining strip include curvature matched method, adaptive space-filling curve (SFC) method [22], interference-free tool orientation optimization [12], and machining potential field method [3].

2.3. Kinematic capability optimization

Several previous works considered machine tool kinematic capabilities in the CAM stage to achieve optimal feed speed. The tool path generated and optimized in CAM with only geometrical consideration could not guarantee a superior machining efficiency because the actual feedrate assigned by the controller must respect the machine tool's physical performance constraints.

Kim [14] provided a time-optimal tool path generation method by considering the speed limits of the machine axes and the surface finish requirements. Beudaert [1] provided a tool path smoothing algorithm based on the maximum reachable feedrate by considering the velocity, acceleration and jerk limits of each drive. The proposed algorithm starts from a given tool path and iteratively smoothes the joint motions in order to raise the real feedrate. Hu and Tang [11] proposed a concept called machine-dependent potential field (MDPF), which characterized the relationship between the material removal rate and the feed direction with considering the kinematic constraints of the machine's axes. Based on the MDPF, the optimal direction is searched which has the maximal MDPF value and a so-called principal MDPF curve can thus be traced out along the optimal direction of each point. With the principle MDPF curve, a region by region tool path generation method is proposed similar to that in [3]. Tool path generation in this way is a compromise between the machining efficiency and the uniformness of the tool path pattern.

2.4. Field based methods

Several authors used scalar and vector fields to generate smooth tool paths. The work in [8] issued a patent to produce spiral tool path for blade milling. The patent uses an imaginary electrical potential field to generate auxiliary lines for constructing a spiral two-dimensional guild path. Makhanov [19] transformed the formulation of a tool path to a curvilinear coordinate system on the surface with the elastic deformation being minimized. It involves solving two scalar fields. Bieterman [2] and Chuang [4] used path scalar fields to represent the tool path pattern to optimize the smoothness of the tool path, he optimization is achieved by solving an Elliptic PDE as a boundary value problem. The formulation of Bieterman and Sandstrom method lacks of metric tensor and is not covariant. The Chuang and Yang method is independent of the surface parameterization, but is limited to 2D planar surfaces only.

Several vector field based methods were used to achieve other optimization requirements besides tool path smoothness. Chiou [3] used a vector field to represent cut directions with locally optimal machining width. The final toolpath is computed from the optimal cut direction field via a greedy method. Kim [14] provide a tool path generation method by considering the speed limits of the motors and the surface finish requirements. The final tool path is generated by fitting a continuous vector field from the greedy solution. Hu and Tang [11] applied a direction field to optimize the feed direction based on material removal rate and the speed and acceleration limits of the machine's axes. The direction field is equivalent to a unitized vector field and the solution is also based on greedy method. Kim [15] established a differential geometry framework of sweeping path problems by using vector field. The solution was obtained using a greedy method instead of formal optimization.

Tensor field was used in Kim [16] to generate iso-cusp-height tool paths. First, Kim [16] defined a Riemannian manifold by assigning a new metric to a part surface. The new metric is constructed from the curvature tensors of a part and a tool surface, and referred as cusp metric. Then, geodesic parallels ware constructed on the new Riemannian manifold. The constant cusp height tool paths were built by a selection from such a family of geodesic parallels.

2.5. Summary of related works

As overviewed in the earlier subsections, Kim [14] used vector field as the mean to compute the tool paths that optimize the machine kinematic performance. It computes a greedy vector field that represents the locally-optimal-kinematic-performance tool path. However, this greedy vector field is not continuous and not suitable to construct the actual tool path. The solution proposed in the paper is to use fitting technique to construct a globally continuous vector field that approximates the greedy vector field. The resulting fitted vector field may not be globally optimal, since the target cost function of the fitting process is based on the fitting error, not the desirable machine kinematic performance.

The above method exposes a limitation of vector field: a vector field represents the locally-optimal solution - a single direction at each location. However, a globally-optimal solution do not always satisfy locally optimization - it has to compromise the local optimization with global continuity and consistency considerations to achieve the global optimal solution. It becomes problematic to construct the objective function when the globally-optimal direction deviates from the locally-optimal direction.

To overcome this difficulty, we apply (rank-2) tensor field instead. To compute the global optimal solution, we need a convenient mathematical tool to construct the objective function in every possible machining direction at every location on the workpiece surface. Vector field can be used to represent the quantity in a single direction but not every direction. Hence, we choose the mathematical tool of tensor, which incorporate the quantitative measure in every direction, instead of a single direction.

Several previously mentioned papers use vector field as well, they have to rely on locally-optimal greedy solution to derive a global solution in an ad-hoc fashion without the formal optimization.

Kim [16] published a different paper that illustrated the power of tensor field. This work applied tensor field to scallop height computation but didn't cover the topic of optimal kinematic performance.

Another example of tensor field usage is in Liu [18] that provided a new region-based tool path generation method by introducing a tensor field to evaluate the machining strip width using ball end mill. This reference also didn't cover the optimal kinematic performance problem.

Summarizing the previous works, we conclude the lack a mathematically rigorous approach to address the problem of computing tool paths with globally-optimal machine kinematic performance. In this paper we aim at deriving such mathematical framework based on differential geometry, especially the use of tensor fields.

3. Machine kinematic metric of a tool path

We first describe the mathematical background of the problem. In order to optimize the finishing tool path pattern on a given freeform surface, we need to find the cut direction that minimizes the inverse cutting velocity square:

$$\rho = \frac{1}{\mathbf{V}^2} = \left(\frac{dt}{d|\mathbf{X}|}\right)^2 \tag{3.1}$$

Keeping the cutting distance $d|\mathbf{X}|$ unitized, the minimization target becomes $(dt)^2$, the infinitesimal cutting time square, which varies with different cut direction. To be more specific, we need to find an instrument that can be used to compute $(dt)^2$ for every given cut direction at every location on the part surface, from which we can then find the cut direction that minimize ρ at every location on the surface. The instrument for computing $(dt)^2$ has to be a field over the part surface and a tensor such that the value is sensitive to the direction, the same reason as metric tensor or shape operator tensor in differential geometry. In later sections we will formulate such instrument, Machine Kinematic Metric, which is a tensor field as expected.

To prepare for the mathematics in later sections, we need to define two coordinate systems: WCS and AMCS. Workpiece Coordinate System (WCS) is attached to the workpiece (part geometry) and commonly used to describe the movement of the cutter (tool) relative to the workpiece, a.k.a. tool path or CL data. Abstract Machine Coordinate System (AMCS) is a coordinate system in the feed-axes space of the machine tool, as shown with a five-axis machine tool example in Fig. 1. It's used to describe the G-code and axis movement. We intentionally label it "abstract" because this space may not have an exact match in the physical space. For instance, a rotational axis of a machine tool does not match any dimension in the physical space.

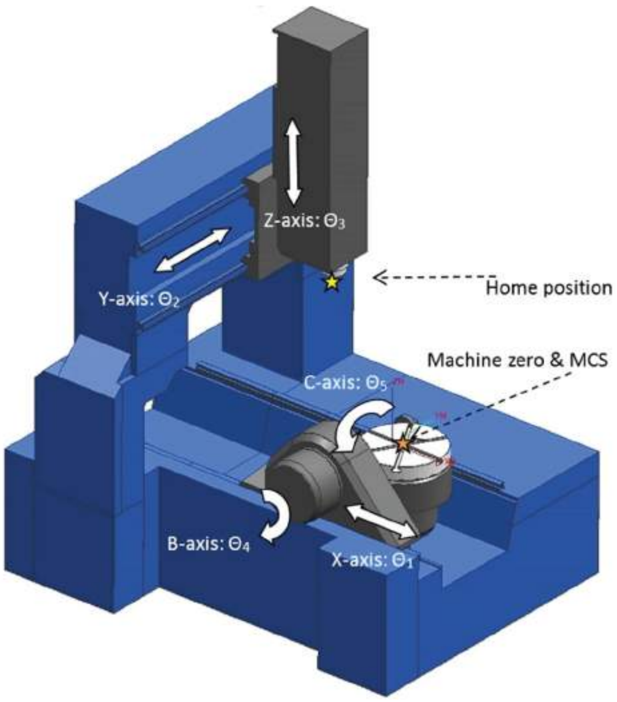


Figure 1. A five-axis machine tool and its axes.

The G-code is derived from the corresponding CL data via an Inverse Kinematic Transformation (IKT). That is, a point in AMCS is computed from a point in WCS via IKT. Inversely, a point in WCS corresponds to a point in AMCS thru Forward Kinematic Transformation (FKT). Fig. 2 illustrates such relationship (it's impossible to visualize a 5-dimensional space so we draw a three-axis frame to denote the AMCS, with the 4th & 5th axes hidden).

The map from WCS to AMCS could be one-to-many for five-axis machines. If we choose a solution that is continuous, the mapping is continuous and differentiable outside of the singularity poles. For three-axis machines, the mapping between WCS and AMCS is strictly continuous and differentiable.

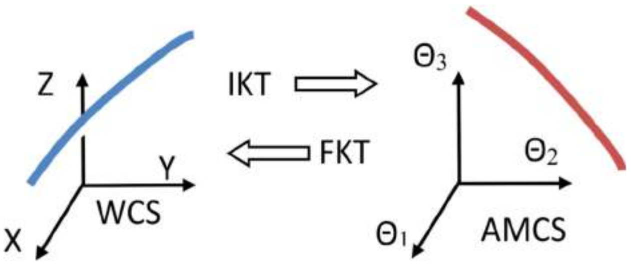


Figure 2. Mapping between WCS (CL data) and AMCS data (G-code).

CAM tool-paths are defined in WCS. A five-axis tool-path in WCS contains the CL trajectory $\boldsymbol{P}(\tau)$ and tool-axis field $\boldsymbol{A}(\tau)$, where τ is a curve parameter but not necessarily the time variable. In practice, a tool-path is often discretized into GOTO points as $\boldsymbol{P_i}$ and $\boldsymbol{A_i}$, but it can be easily parameterized into a G^0 continuous curve (a polyline).

A CAM tool-path in WCS corresponds to an axis-path (G-code program) defined in AMCS, via IKT, as shown in Fig. 3. The AMCS of a five-axis machine corresponds to a five-dimensional abstract vector space: $\Theta = (\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5)$. A five-axis axis-path is a trajectory, or curve, in this abstract vector space: $\Theta(\tau)$.

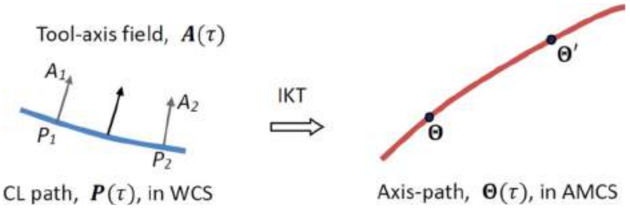

Tool-axis field, $\boldsymbol{A}(\tau)$
A_1
A_2
P_1
P_2
CL path, $\boldsymbol{P}(\tau)$, in WCS
IKT
$\boldsymbol{\Theta}'$
$\boldsymbol{\Theta}$
Axis-path, $\boldsymbol{\Theta}(\tau)$, in AMCS

Figure 3. Axis-path in AMCS.

The goal of time-efficient machining is to minimize the total machining time, which is the integral of the time element over the whole path: $T = \int dt$. It means that minimizing the time element throughout the path will ensure minimal total machining time.

The time element is related to the velocity - higher velocity results in shorter time. However, the maximally achievable velocity is limited by many factors. Here we address the major factor: the maximal axis speed allowed by the machine tool kinematic capability.

Let $\boldsymbol{\Psi} = (\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5)$ represents the maximally allowed speed of the fisve axes (in the abstract axis vector space). For each axis, the minimal time required to move an axis is: $dt = \left|\frac{d\Theta_j}{\Psi_j}\right|$. The minimally allowed time element in the axis-path, $\boldsymbol{\Theta}(\tau)$, is the largest among the minimum time elements of the five axes:

$$\min(dt) = \max_{j=1\ldots5}\left|\frac{d\Theta_j}{\Psi_j}\right| \tag{3.2}$$

In mathematics, the right hand side is known as the Chebyshev distance.

We give an example about how to calculate dt in Eqn.(3.2). Two GOTO points $(\boldsymbol{P}_1, \boldsymbol{A}_1)$ and $(\boldsymbol{P}_2, \boldsymbol{A}_2)$ on a tool path in WCS, shown in Fig. 3, are mapped to AMCS via IKT and obtain two corresponding abstract points Θ and Θ' in AMCS. The five elements of Θ are $(\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5)$, and Θ', $(\Theta'^1, \Theta'_2, \Theta'_3, \Theta'_4, \Theta'_5)$. The minimally allowed time element between $(\boldsymbol{P}_1, \boldsymbol{A}_1)$ and $(\boldsymbol{P}_2, \boldsymbol{A}_2)$ is computed as :

$$\min(dt) = \max\begin{bmatrix}(\Theta'_1 - \Theta_1)/\Psi_1\\(\Theta'_2 - \Theta_2)/\Psi_2\\(\Theta'_3 - \Theta_3)/\Psi_3\\(\Theta'_4 - \Theta_4)/\Psi_4\\(\Theta'_5 - \Theta_5)/\Psi_5\end{bmatrix}$$

Let's introduce a scaling of the machine axes and define the max-speed-scaled axes as:

$$\Omega_j = \frac{\Theta_j}{\Psi_j} \tag{3.3}$$

The minimum time element is then (the wording "min" is dropped in the equation to save typing):

$$dt = |d\boldsymbol{\Omega}|_C = \max_j |d\Omega_j| \tag{3.4}$$

The subscript C in the above equation indicates Chebyshev distance. That is, the minimum time of the path is exactly the Chebyshev distance in the $\boldsymbol{\Omega}$ space. The optimal total machining time is the line integral of dt along the path, computed as:

$$\min T = \int dt = \int |d\boldsymbol{\Omega}|_C \tag{3.5}$$

Define the *machine kinematic metric* (MKM) as the square of the minimum time element (using the square to make it positive definite):

$$M_C = (dt)^2 = |d\boldsymbol{\Omega}|_C{}^2 \tag{3.6}$$

The above MKM allows us to conveniently compute the shortest machining time of a given axis-path. For any given path parameter, τ, the shortest axis-speed-constrained machining time of an axis-path segment can be explicitly computed as:

$$\min T = \int \sqrt{\left(\frac{|d\boldsymbol{\Omega}|_C}{d\tau}\right)^2} d\tau \tag{3.7}$$

Eqn. (3.7) is equivalent to Eqn.(3.5), it just shows that the integral can be computed by any parameter of the path.

4. Machine kinematic metric of a surface

Consider the finishing operation of a part surface defined as S(u, v). Assume that the finishing tool-path is densely distributed over the surface such that it can be treated as a function of the surface parameter. That is, we assume that on every point on the surface there is a corresponding CL point P(u, v) and tool-axis vector A(u, v), both are defined in the WCS. Apparently the CL point and tool-axis vector is defined only on the tool-path, not between path trajectories. However, we assume they are defined between paths via interpolation.

In some situations the tool-axis is defined using a forward leaning angle from the surface normal toward the cut (feed) direction, meaning that the tool-axis at every location depends also on the cut direction. In these cases we assume that the leaning angle is small such that the surface normal is a good approximation of the tool-axis and will be used as $\boldsymbol{A}(u, v)$ to eliminate the dependency on the cut direction.

With IKT, the tool position and axis, $\boldsymbol{P}(u, v)$ and $\boldsymbol{A}(u, v)$, correspond to an axes variable defined in AMCS, $\boldsymbol{\Theta}(u, v)$, where $\boldsymbol{\Theta} = (\Theta_1, \Theta_2, \Theta_3, \Theta_4, \Theta_5)$. Hence, $\boldsymbol{\Theta}(u, v)$ defines a surface in this abstract axes vector space and let's call it the machine *kinematic surface*, as illustrated in Fig. 4. A tool path contact trajectory in the part surface (in WCS) corresponds to a curve (axis-path) in the machine kinematic surface (in the $\boldsymbol{\Theta}$ space).

We are interesting in computing the minimum machining time and the machine kinematic metric introduced in the previous section. The machine kinematic metric is now defined over a surface (the machine kinematic surface), instead of a curve (the axis-path).

As shown in Fig. 5, a parameter curve in (u, v) space corresponds to a tool-path trajectory and a curve-on-surface (surface curve) on the machine kinematic surface. The differential element of the curve parameter $d\tau$ is related to the differential elements of the surface parameters:

$$d\tau = \sum_{\alpha=1,2;\beta=1,2} \frac{d\tau}{du^\alpha du^\beta} du^\alpha du^\beta \tag{4.1}$$

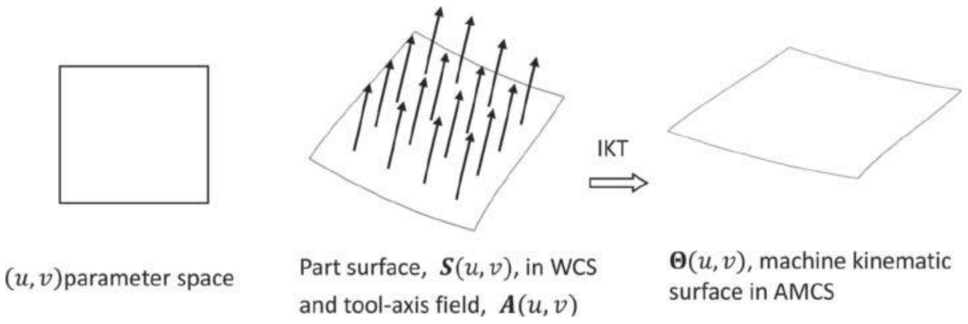

IKT
(u,v)parameter space
Part surface, $\boldsymbol{S}(u,v)$, in WCS
and tool-axis field, $\boldsymbol{A}(u,v)$
$\boldsymbol{\Theta}(u,v)$, machine kinematic
surface in AMCS

Figure 4. Machine kinematic surface.

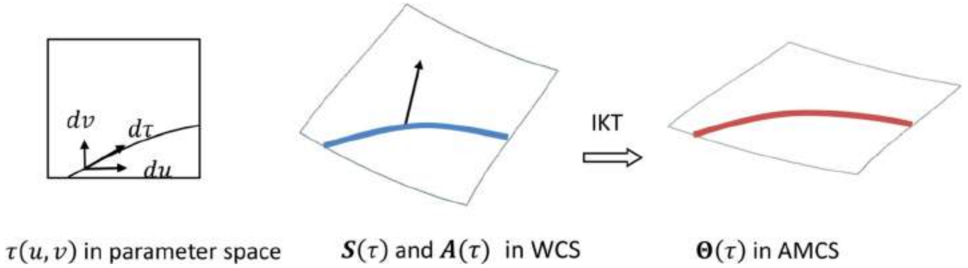

dv
$d\tau$
du
IKT
$\tau(u,v)$ in parameter space
$\boldsymbol{S}(\tau)$ and $\boldsymbol{A}(\tau)$ in WCS
$\boldsymbol{\Theta}(\tau)$ in AMCS

Figure 5. A curve on the machine kinematic surface.

In the above equation, we use du^{α} to indicate one of (u, v). Use the same vector, $\Psi = (\Psi_1, \Psi_2, \Psi_3, \Psi_4, \Psi_5)$, to represent the maximally allowed speed of the five axes and the same scaling, $\Omega_{\mathrm{j}}\sim = \frac{\Theta_{\mathrm{j}}}{\Psi_{\mathrm{j}}}$, such that the maximally allowed speed of each axis is exactly one. Hence, we have a scaled machine kinematic surface, $\Omega\,(\mathrm{u},\mathrm{v})$.

The optimal total machining time of a surface is the surface integral of dt over the surface, computed as: $\min T = \int\limits_S dt = \int\limits_S |d\boldsymbol{\Omega}|_C$.

Using the same reasoning, we derive the same conclusion that the squared optimal time element is the Chebyshev distance in the $\boldsymbol{\Omega}$ space (the subscript C indicates Chebyshev distance):

$$M_C = (dt)^2 = |d\boldsymbol{\Omega}|_C{}^2 \tag{4.2}$$

The above is the machine kinematic metric (MKM) over the machine kinematic surface, and looks the same as the MKM in Eqn. (3.6).

We can use simple differential geometry technique to compute the optimal machining time of any curve in the machine kinematic surface. The key is to utilize the bilinear property of the rank-2 tensor:

$$M_C = \sum_{\alpha,\beta} \frac{|d\boldsymbol{\Omega}|_C{}^2}{du^{\alpha}du^{\beta}} du^{\alpha}du^{\beta} \tag{4.3}$$

Now recall that a tool-path trajectory corresponds to a curve in the machine kinematic surface. If this curve is parameterized by τ, the squared optimal machining time of the curve (path trajectory) will be:

$$\min T = \int \left[\sum_{\alpha,\beta} \sqrt{\frac{|d\boldsymbol{\Omega}|_C{}^2}{du^{\alpha}du^{\beta}} \frac{du^{\alpha}}{d\tau} \frac{du^{\beta}}{d\tau}} \right] d\tau \tag{4.4}$$

The above equation is consistent with Eqn. (3.7). It also shows that the surface integral can be computed by any parameters of the surface.

Given the MKM defined in Eqn. (4.3), we can compute the velocity (i.e., the feedrate) in any direction of $\boldsymbol{du}$ on the surface. The physical length of an infinitesimal segment in the WCS is related to the metric tensor of $\boldsymbol{S}(u,v) :\to (dl)^2 = \Sigma_{\alpha,\beta} G_{\alpha,\beta} du^{\alpha}du^{\beta}$. Hence, the maximally-allowed velocity square (MAVS) is:

$$\rho(\boldsymbol{du}) = v^2 = \frac{(dl)^2}{(dt)^2} = \frac{\sum_{\alpha,\beta} G_{\alpha\beta} du^{\alpha}du^{\beta}}{\sum_{\alpha,\beta} \frac{|d\boldsymbol{\Omega}|_C{}^2}{du^{\alpha}du^{\beta}} du^{\alpha}du^{\beta}} \tag{4.5}$$

The above equation enables us to compute the maximally allowed velocity (feedrate) at any location and in any direction on the surface.

5. First-order approximation of optimal machining time

Eqn. (4.4) allows us to compute the optimal machining time using the formulation of MKM when the path trajectory is given. The goal of this section is to compute the optimal machining time without the path trajectory. We will need to formulate the computation based on the part surface and machine tool kinematic (speed) capability, with certain approximation to simplify the computation.

At this point we don't know the exact number and location of the lifts (retracts) in the tool path and the traversal trajectories, and can't compute the non-cutting (movement of the cutter in the air) portion of the machining time. For the time being we will confine the discussion within the cutting portion of the total machining time (and ignore the non-cutting time).

To compute the total machining time of the whole part surface, we need the machining time element of a small surface area, dT. The time element, dt, established earlier is the machining time of a path trajectory element, while the machining time of an area element is the multiplication of the trajectory-element-time, dt, and the number of stepovers, dn (number of trajectories in the area), that is:

$$dT = dt\,dn \tag{5.1}$$

The integral over the surface is based on the area integral, dA, so we need to figure out the connection between dT and dA. Since the tool path is distributed over the part surface, the area element (shown in Fig. 6), dA, equals to path length element, dl, times the path width element, dw, which is number of stepovers, dn, times the stepover distance, S_p. We then have the following connection:

$$\begin{aligned}\min \mathrm{T} &= \int dT = \int \frac{dt\,dn}{dl\,dw}\,dl\cdot dw\\ &= \int \frac{dt}{dl\,S_p}\,dA = \int \frac{1}{\sqrt{\rho}S_p}\,dA\end{aligned} \tag{5.2}$$

The MAVS, ρ, can be derived from Eqn. (4.5). Keep in mind that MAVS depends on the feed direction. Hence, to compute the above machining time integral we need to know the path direction at every location of the surface.

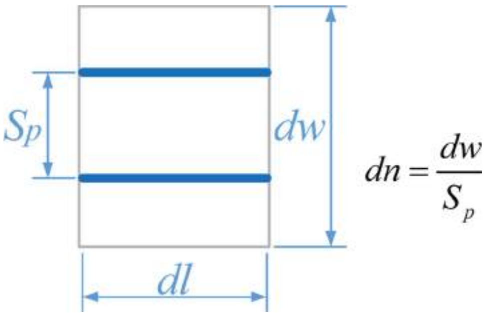


Figure 6. An area element on part surface.

To better understand the meaning of Eqn. (5.1) & Eqn. (5.2), let's work out a simplified example of rectangular part surface, as shown in Fig. 7. In the example we aim at achieving an approximated result to simplify the computation.

For AMCS, we use the scaled $\boldsymbol{\Omega}$ space defined in Eqn. (3.3) for the machine kinematic surface. Consider a common toolpath pattern that is based on angled iso-parametric lines, as shown in the left side of Fig. 7. The path passes, parameterized by γ , are straight lines in the surface parameter space with angle θ from the u-axis. The spacing between the passes (stepover) is constant $\Delta\delta$ in the parameter space. The path parameter is related to the surface parameter as:

$$\begin{aligned}du &= \cos\theta\,d\gamma\\ dv &= \sin\theta\,d\gamma\end{aligned} \tag{5.3}$$

The value of scaled machine kinematic surface at the four corners are $\boldsymbol{\Omega}_{00}$, $\boldsymbol{\Omega}_{01}$, $\boldsymbol{\Omega}_{10}$, and $\boldsymbol{\Omega}_{11}$. Assume the machine kinematic surface is small enough that we can approximate it using a bilinear interpolation. That is:

$$\begin{aligned}\boldsymbol{\Omega}(u,v) = {}&\boldsymbol{\Omega}_{00}(1-u)(1-v) + \boldsymbol{\Omega}_{10}u(1-v)\\ &+ \boldsymbol{\Omega}_{01}(1-u)v + \boldsymbol{\Omega}_{11}uv\end{aligned} \tag{5.4}$$

The derivatives of the scaled machine kinematic surface are:

$$\begin{aligned}\frac{d\boldsymbol{\Omega}}{du} &= (\boldsymbol{\Omega}_{10} - \boldsymbol{\Omega}_{00})(1-v) + (\boldsymbol{\Omega}_{11} - \boldsymbol{\Omega}_{01})v\\ \frac{d\boldsymbol{\Omega}}{dv} &= (\boldsymbol{\Omega}_{01} - \boldsymbol{\Omega}_{00})(1-u) + (\boldsymbol{\Omega}_{11} - \boldsymbol{\Omega}_{10})u\end{aligned} \tag{5.5}$$

We shall further simplify the problem by approximating the above with the mean value at $u =$ 0.5 and $v =$ 0.5:

$$\begin{aligned}\frac{d\boldsymbol{\Omega}}{du} &\cong \frac{1}{2}(\boldsymbol{\Omega}_{10} - \boldsymbol{\Omega}_{00} + \boldsymbol{\Omega}_{11} - \boldsymbol{\Omega}_{01}) = \Delta\boldsymbol{\Omega}_0\\ \frac{d\boldsymbol{\Omega}}{dv} &\cong \frac{1}{2}(\boldsymbol{\Omega}_{01} - \boldsymbol{\Omega}_{00} + \boldsymbol{\Omega}_{11} - \boldsymbol{\Omega}_{10}) = \Delta\boldsymbol{\Omega}_1\end{aligned} \tag{5.6}$$

The above approximation is justified if the scaled machine kinematic surface patch is quite small, meaning that the value of $\boldsymbol{\Omega}$ varies very little.

Based on Eqn. (4.3) and chain rule, we can compute the time element along the path (with Chebyshev distances):

$$(dt)^2 = |d\boldsymbol{\Omega}|_C{}^2 = \left[\left|\frac{d\boldsymbol{\Omega}}{du}\frac{du}{d\gamma} + \frac{d\boldsymbol{\Omega}}{dv}\frac{dv}{d\gamma}\right|_C d\gamma\right]^2 \tag{5.7}$$

From Eqn. (5.3) & (5.6) , we then have:

$$dt = |\Delta\boldsymbol{\Omega}_0\cos\theta + \Delta\boldsymbol{\Omega}_1\sin\theta|_C d\gamma \tag{5.8}$$

We can derive the surface machining time, combining the above result and Eqn. (5.1): (with $dn = d\delta/\Delta\delta$):

$$\int dT = \int dtdn = \int \frac{|\Delta\boldsymbol{\Omega}_0\cos\theta + \Delta\boldsymbol{\Omega}_1\sin\theta|_C}{\Delta\delta}d\gamma\,d\delta \tag{5.9}$$

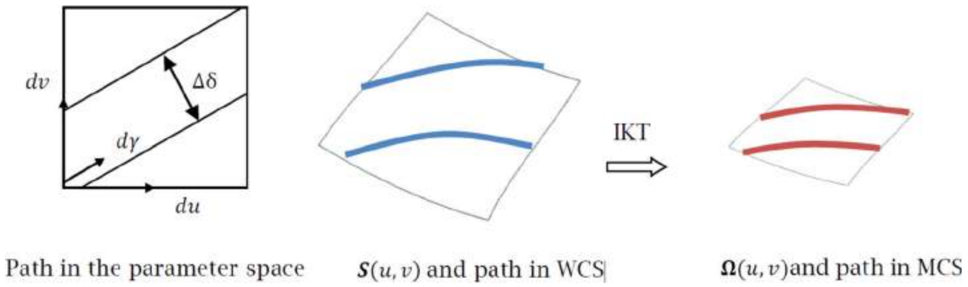


Figure 7. A simple example of rectangular part surface, and the corresponding rectangular machine kinematic surface.

If the above assumption holds true that the machine kinematic surface is small enough for linear approximation, the integrand is a constant and can be taken out of the integral, leaving a trivial integration of the square parameter box that is exactly one. Hence, we have:

$$\int dT = \frac{|\Delta\mathbf{\Omega}_0 \cos\theta + \Delta\mathbf{\Omega}_1 \sin\theta|_C}{\Delta\delta} \tag{5.10}$$

The above equation indicates that the surface machining time (to be precise, it means the optimal machining time allowed by the axis-speed constraint) solely depends on the cut angle, θ, in the parameter space. We can use it to determine the optimal cut angle that would lead to the least machining time under the axis-speed constraint.

6. Application in 3-axis surface finishing

We now consider the case of three-axis finishing of sculptured surfaces, such as die faces. Let's define the tool-axis orientation the Z-axis of the AMCS. It's common to finish die faces with tool paths that are straight when projected to the X-Y plane of the AMCS. This type of X-Y parallel tool paths are easy to compute and the results are predictable.

For such X-Y parallel tool paths, once the stepover distance in the X-Y plane is determined (based on scallop height requirement) the main variable is the direction of the path represented by an angle measured from the X-axis, known as cut angle. The task on hand is to compute the optimal cut angle in order to minimize the machining time that can be computed via Eqn. (5.1) & Eqn. (5.2)

Assume the machine tool is capable of superior acceleration capability such that the speed limit of the axes is the major factor that determines the machining time. A die face is usually quite large and consists of multiple surfaces, so we can't apply Eqn. (5.10) to the whole die face. Instead, we can subdivide the whole die face into many small elements such that each element is small enough for the approximation in Eqn. (5.10) to be applicable.

The subdivision scheme is inspired by Z-map. That is, we build a rectangular grid in the X-Y plane. Each grid cell is a small square in the X-Y plane, and its projection onto the die face corresponds to a small surface patch and the sum of them forms the complete die face. Hence, the surface integral in Eqn. (5.2) can be decomposed into the sum of surface integrals over these surface patches.

Assume each patch is small enough such that we can apply the approximation scheme described in section 4. The situation here is similar to section 4, except that the parameters are the (x,y) coordinates (of the X-Y plane) instead of (u,v), while the cut angle θ is measured from the X-direction. The counterpart of Eqn. (5.6) is:

$$\begin{aligned} \frac{d\mathbf{\Omega}}{dx} &\cong \frac{1}{2}(\mathbf{\Omega}_{10} - \mathbf{\Omega}_{00} + \mathbf{\Omega}_{11} - \mathbf{\Omega}_{01}) = \Delta\mathbf{\Omega}_{\mathrm{x}} \\ \frac{d\mathbf{\Omega}}{dy} &\cong \frac{1}{2}(\mathbf{\Omega}_{01} - \mathbf{\Omega}_{00} + \mathbf{\Omega}_{11} - \mathbf{\Omega}_{11}) = \Delta\mathbf{\Omega}_{\mathrm{y}} \end{aligned} \tag{6.1}$$

The time element along the path is:

$$dt = |\Delta\mathbf{\Omega}_{\mathrm{X}} \cos\theta + \Delta\mathbf{\Omega}_{\mathrm{Y}} \sin\theta|_C d\gamma \tag{6.2}$$

Similar to Eqn. (5.10), the machining time integral of a patch is:

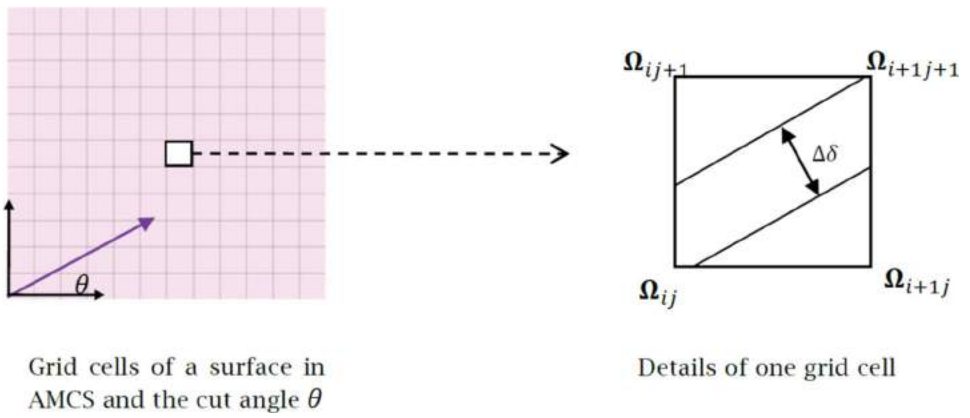
$$\int dT = \frac{|\Delta\mathbf{\Omega}_{\mathrm{X}} \cos\theta + \Delta\mathbf{\Omega}_{\mathrm{Y}} \sin\theta|_C}{\Delta\delta} \Delta A \tag{6.3}$$

The symbol ΔA in the above equation is the area of the grid cell in the X-Y plane of AMCS, and $\Delta\delta$ now represents the step-over distance measured in the X-Y plane of AMCS.

Use (i,j) to label the patch cells, the numerical approximation of the integral corresponds to a summation of all patches, using the previous equation:

$$\begin{aligned} T(\theta) &= \sum_{\mathrm{ij}} |\Delta_{\mathrm{ij}}\mathbf{\Omega}_{\mathrm{X}} \cos\theta + \Delta_{\mathrm{ij}}\mathbf{\Omega}_{\mathrm{Y}} \sin\theta|_C \frac{\Delta A}{\Delta\delta} \\ \Delta_{\mathrm{ij}}\mathbf{\Omega}_{\mathrm{X}} &= \frac{1}{2}(\mathbf{\Omega}_{i+1j} - \mathbf{\Omega}_{ij} + \mathbf{\Omega}_{i+1j+1} - \mathbf{\Omega}_{ij+1}) \\ \Delta_{\mathrm{ij}}\mathbf{\Omega}_{\mathrm{Y}} &= \frac{1}{2}(\mathbf{\Omega}_{ij+1} - \mathbf{\Omega}_{ij} + \mathbf{\Omega}_{i+1j+1} - \mathbf{\Omega}_{i+1j}) \end{aligned} \tag{6.4}$$

The illustration in Fig. 8 helps understand the meaning of grid cells and related variables in the above equation.


θ
$\boldsymbol{\Omega}_{ij+1}$
$\boldsymbol{\Omega}_{i+1j+1}$
Δδ
$\boldsymbol{\Omega}_{ij}$
$\boldsymbol{\Omega}_{i+1j}$

Grid cells of a surface in AMCS and the cut angle θ

Details of one grid cell

Figure 8. Illustration of a Z-map and corresponding grid cells of a surface in AMCS, and the variables for the computation of T(θ).

The task is then to find the cut angle θ that minimizes $T(\theta)$. To show how this works, let's consider a very simple example of a X-Y planar surface. Assume the size of each grid cell is 1.0 mm and there are $M \times N$ cells in the surface. There is no movement in the Z-axis so it can be ignored; and the feed speed upper bounds are set to Ψ_X and Ψ_Y. In the max-speed-scaled $\boldsymbol{\Omega}$ space, the differences of $\boldsymbol{\Omega}$ in a grid cell are $\left(\frac{1}{\Psi_\mathrm{X}}, \frac{1}{\Psi_\mathrm{Y}}\right)$. Thus, the last equation yields: $\Delta_{\mathrm{ij}}\boldsymbol{\Omega}_\mathrm{X} = \left(\frac{1}{\Psi_\mathrm{X}}, 0\right)$, $\Delta_{\mathrm{ij}}\boldsymbol{\Omega}_\mathrm{Y} = \left(0, \frac{1}{\Psi_\mathrm{Y}}\right)$; and the optimal machining time is $T(\theta) = \frac{MN}{\delta}\max\left(\left|\frac{\cos\theta}{\Psi_\mathrm{X}}\right|, \left|\frac{\sin\theta}{\Psi_\mathrm{Y}}\right|\right)$, where Chebyshev distance (max operator) is applied to the $\boldsymbol{\Omega}$ vectors. The plot of $T(\theta)$ with ($\Psi_\mathrm{X} = 2000$ and $\Psi_\mathrm{Y} = 1000$) is in Fig. 9.

It's not hard to prove that $T(\theta)$ is minimal when $|\cos\theta/\Psi_\mathrm{X}| = |\sin\theta/\Psi_\mathrm{Y}|$, which implies: $\tan\theta = \Psi_\mathrm{Y}/\Psi_\mathrm{X}$. The above result serves as a greedy solution of the optimal cutting time and can be used as theoretical comparison and the initial guess for experimental search.

7. Simulation and experiments

To evaluate the effectiveness of the proposed method, we carry out the surface finishing experiments on two test cases: one simple freeform surface model and a complex multi-surface die face model. For each test case, we compute the machining (surface finishing) time as a function of cut angle using the MKM method, and compare it to the experiment using a (real) commercial CNC simulator. We describe the two experiments and their results in the following sections.

7.1. Case 1: A simple freeform surface

Consider a simple freeform surface illustrated in Fig. 10. The machining strategy is a 3-axis zig-zag tool path and the process parameters and machine tool kinematic capabilities are listed in Tab. 1. The programmed feedrate is higher than the resultant velocity limit of all linear axes. The acceleration capability is assumed to be superior and

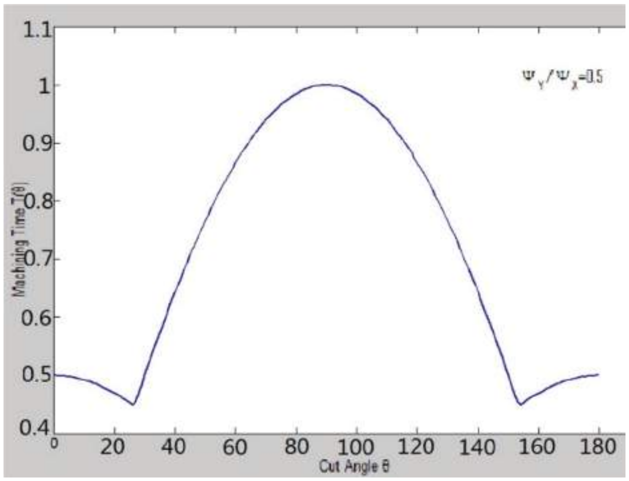


Figure 9. Illustration of T(θ) for a X-Y planar surface.

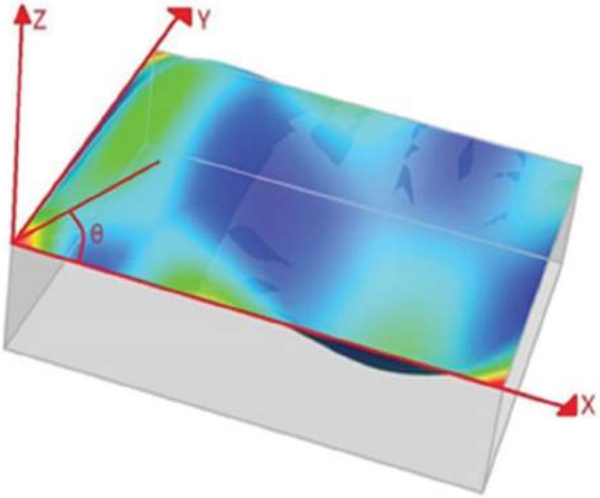


Figure 10. Freeform surface model and the cut angle θ.

Table 1. Process parameters of example 1.

Cutter	Ball end mill,Φ10 mm
Blank Dimensions (X× Y ×Z)	110mm × 100 mm × 35 mm
Step-over of Tool Path ($\Delta\beta$)	1mm
Velocity Limit of X axis (Ψ_X)	1430 mm/min
Velocity Limit of Y axis (Ψ_Y)	715 mm/min;
Velocity Limit of Z axis (Ψ_Z)	300 mm/min
Programmed Feedrate	10000 mm/min
Acceleration Limits of X,Y,Z	$1.04 \times 10^3 \text{mm/s}^2$
Jerk Limits of X,Y,Z	125mm/s^3

hence not a factor. Once the parameters listed in Tab. 1 are determined, the only variable to influence the machining time is the cut angle θ measured from X-axis (as shown in Fig. 10).

Theoretical Computation of T(θ)

The steps to compute T(θ) and the optimized cut angle θ are as follows:

- Build a rectangular gird in the X-Y plane. Each grid cell is a 1 mm ×1 mm square. Then all cells are projected to the freeform surface to form a discretized surface net that approximates the original surface $S(u, v)$, shown in the left portion of Fig. 11.
- Map the surface $S(u, v)$ into the AMCS thru IKT, and apply the maximally allowed speed scaling of the three linear axes, to obtain the abstract kinematic surface $\Omega(u, v)$ in Cartesian coordinate system, as shown in the right portion of Fig. 11.
- For each of the cut angle $\theta = 0, 1^\circ, 2^\circ, \ldots 179^\circ$, compute the numerical approximation of the integral according to Eqn. (6.4), which denotes the predicted machine time of the zig-zag tool path with cut angle θ.

The total lengths of the tool paths according to cut angle θ are listed in Fig. 12. It shows that the minimum length is around $\theta = 90^\circ$.

The theoretical prediction curve of T(θ) is plotted in Fig. 14. The cut angle that minimizes T(θ) , theoretically, is around $\theta = 58^\circ$.

Simulation Experiment and Discussion

We then use a commercial CNC control simulator to simulate the machining processes of the series of zig-zag tool paths. The simulator, shown in Fig. 13, is provided by Wuhan Huazhong Numerical Control Co., Ltd, and used to simulate the machining time by using the same command interpolation and motion planning algorithms that are used in HNC-8 series controller products. The simulated machining time equals to the actual machining time executed by the HNC-8 Controller at a machine tool without manual interference.

We use UG NX to generate 36 zig-zag tool paths, one for every 5° difference of cut angle. Each tool path is post-processed and sent to the above HNC-8 Simulator, which is set to the axis speed limits listed in Table 1. The

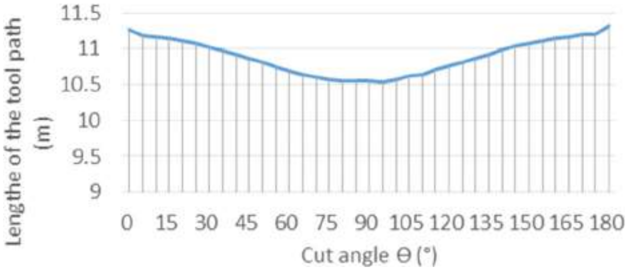


Figure 12. The total length of the toolpath (G01) according to cut angle θ.

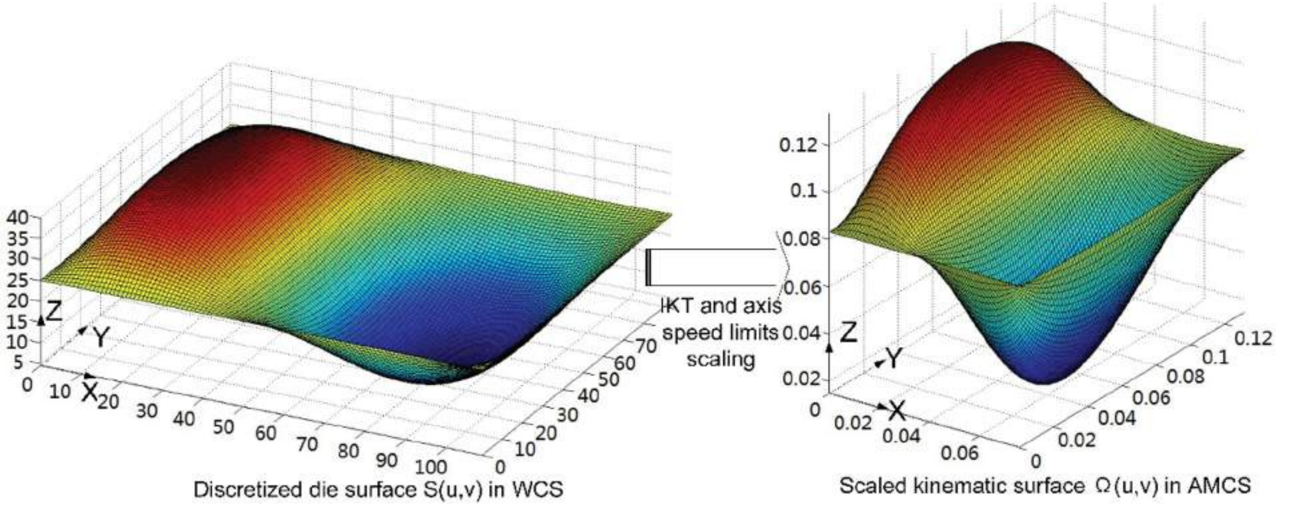
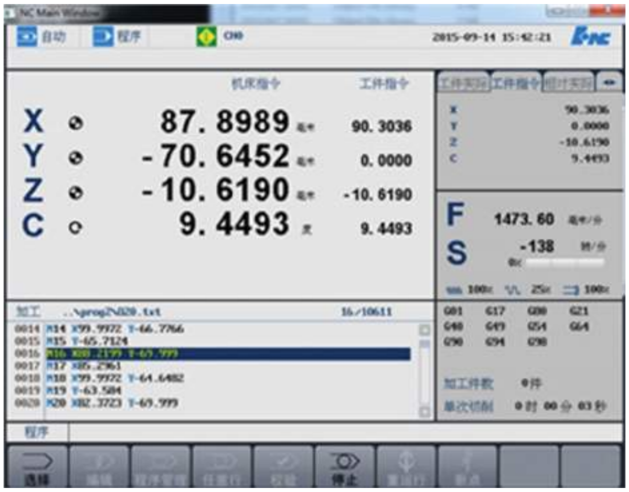


Figure 11. Freeform surface and the corresponding scaled kinematic surface.



2015-09-14 15:42:21

		机床指令		工件指令
X	87.8989	毫米	90.3036
Y	-70.6452	毫米	0.0000
Z	-10.6190	毫米	-10.6190
C	9.4493	度	9.4493

工件实际 工件指令 相对实际

X 90.3036
Y 0.0000
Z -10.6190
C 9.4493

F 1473.60 毫米/分
S -138 转/分

加工 ..\prog2\020.txt 16/10611

```
0014 N14 X99.9972 Y-66.7766
0015 N15 Y-65.7124
0016 N16 X88.2199 Y-65.999
0017 N17 X85.2961
0018 N18 X99.9972 Y-64.6482
0019 N19 Y-63.584
0020 N20 X82.3723 Y-65.999
```

G01 G17 G80 G21
G40 G49 G54 G64
G90 G94 G98

加工件数 0件
本次切削 0时 00分 03秒

程序

Figure 13. The UI of HNC-8 simulator.

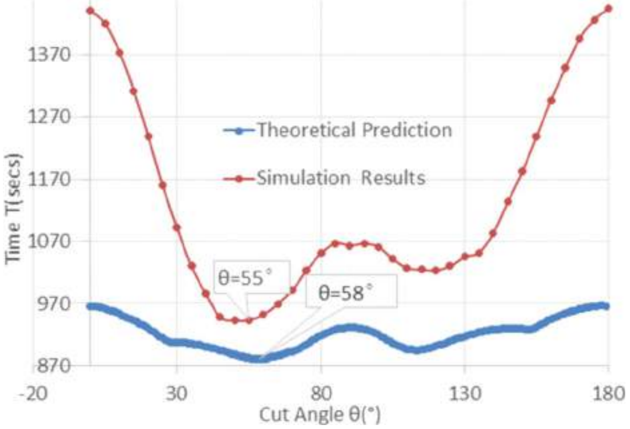


Figure 14. Comparison of T(θ) between theoretical prediction of our method and HNC-8 Simulator in Case 1.

execution time for every tool path is recorded and also plotted in Fig. 14, against the theoretical prediction curve. It shows that the optimal machining time occurs around $\theta = 55°$.

From the comparison between the simulation results and the theoretical prediction, it can be seen that the general shape of the machining-time curves are consistent. The maximum machine time occurs at cut angle $\theta = 0$ in both curves and the minimum machine time occurs around $\theta = 55°$in simulation and $\theta = 58°$ in theoretical computation. It also can be seen that the values of simulated machining time are always larger than the theoretical values. The main reason is that in the theoretical computation of T(θ) we do not consider the time spent on non-cutting (air) moves and the slowdown due to axes acceleration constraints.

7.2 Case 2: A complex multi-surface die face

In this example, we consider a complex multi-surface die face illustrated in Fig. 15. The machine strategy is 3-axis zig-zag tool path and the process parameters and machine tool kinematic capabilities are listed in Tab 2. The programmed feedrate is higher than the resultant velocity limit of all linear axes. Again, the acceleration capability is ignored. The definition of cut angle θ is shown in Fig. 15.

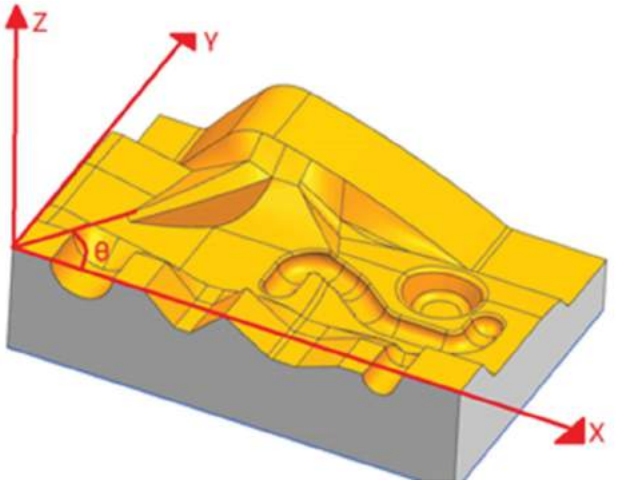


Figure 15. A complex multi-surface die model and the cut angle θ.

Table 2. Process Parameters of example 2.

Cutter	ball end mill , Φ10 mm
Blank Dimensions (X×Y×Z)	110 mm×100 mm×35 mm
Step-over of Tool Path ($\Delta\beta$)	1 mm
Velocity Limit of X axis (Ψ_X)	1430 mm/min
Velocity Limit of Y axis (Ψ_Y)	715 mm/min;
Velocity Limit of Z axis (Ψ_Z)	150 mm/min
Programmed Feedrate	10000 mm/min
Acceleration Limits of X,Y,Z	$1.04 \times 10^3 mm/s^2$
Jerk Limits of X,Y,Z	$125mm/s^3$

Theoretical Computation of T(θ)

Similar to the previous example, we build a rectangular gird in the X-Y plane with cell size of (1 mm×1 mm) square. The discretized surface net that approximates the original surface, $S(u, v)$, shown in the left portion of Fig. 16. The mapped abstract kinematic surface $\Omega(u, v)$ in AMCS thru IKT with axis speed scaling is shown in the right portion of Fig. 16.

The range of cut angle θ is [0,180]. We set a cut angle sequence ($\theta = 0,1°, \ldots ,179°$), then compute the machining time T(θ_i) of each tool path with cut angle θ_i according to Eqn. (6.4). The theoretical prediction curve in Fig. 17 shows the computed machining time of the sequence of tool paths. The minimal machining time occurs at $\theta = 79°$.

Simulation Experiment and Discussion

UG NX is used to generate 36 zig-zag tool paths, one for every 5° difference of cut angle. Each tool path is

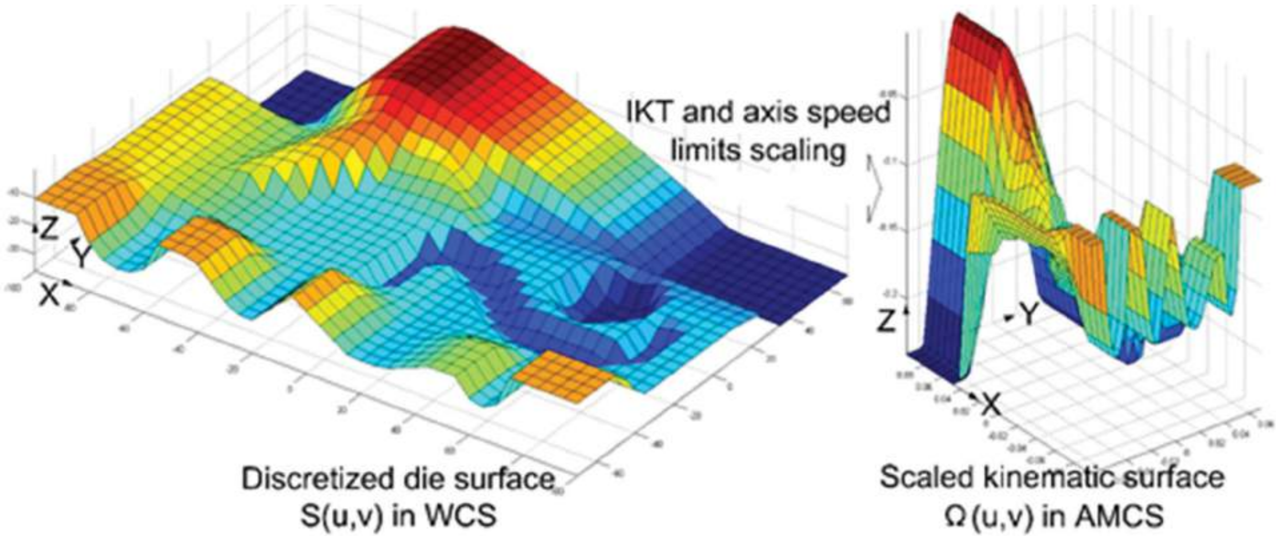


Figure 16. Discretized multi-surface die face and the corresponding scaled kinematic surface.

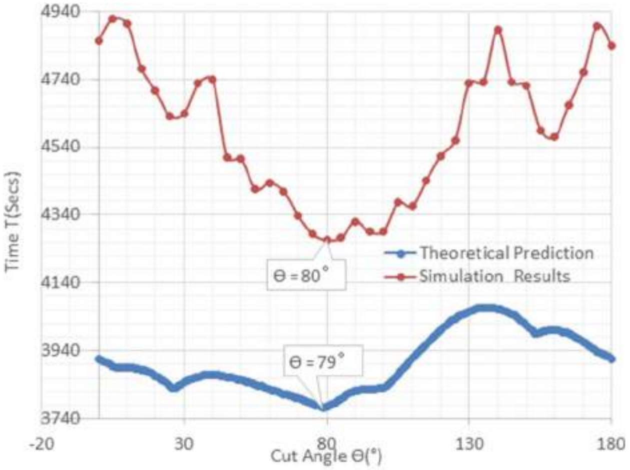


Figure 17. Comparison of T(θ) between theoretical prediction of our method and HNC-8 Simulator in Case 2.

post-processed and sent to the above HNC-8 Simulator, which is set to the axis speed limits listed in Tab 2. The execution time for every tool path is recorded and plotted in Fig. 17. It shows that the optimal machining time occurs around $\theta = 80^\circ$.

From the comparison between the simulation results and the theoretical prediction (Fig. 17.), it can be seen that the general shape of the machining-time curves are consistent. The minimum machine time occurs around $\theta = 80^\circ$ in simulation and $\theta = 79^\circ$ in theoretical computation. The time optimal tool path with cut angle $\theta = 80^\circ$ is shown in Fig. 18. It also can be seen that the HNC-8 simulated machining time is always larger than the theoretical values. The reasons were discussed in the previous example.

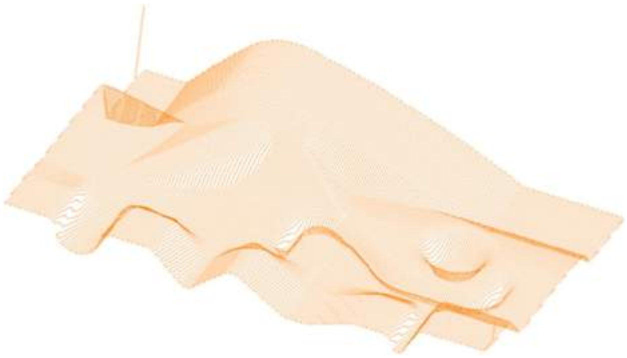


Figure 18. Tool path generated by UG NX with cut angle = 80°.

8. Summary and conclusion

We derived a mathematical model, named Machine Kinematic Metric (MKM) to encode the speed kinematic capabilities of a machine tool. MKM can be used conveniently to analyse the workpiece and evaluate the machining time to finish the workpiece face. We also presented a numerical method of using MKM to compute the (approximate) machining time for every cut direction.

To validate the reliability and usability of the proposed method, we perform simulation experiments on two test cases. For each test case, we perform numerical computation using MKM to compute the machining time on each cut direction and derive the optimal cut direction (with the minimal machining time). We then carry out the experiment for each test case with HNC-8 controller and prove that the experimental results are consistent with the numerical predictions.

The experiments confirm that the proposed MKM method is reliable and provides a convenient and economical mean to estimate machining time and determine the optimal cut direction at the tool path programming stage. The numerical computation is approximate and the model is simplified such that there is a small deviation between the actual optimal cut direction and the computed one. However, the experiments showed that the deviation is within several degrees and reasonable for practical purpose.

Another benefit is that the experiments show cut direction causes significant difference in machining time. Conventional CAM softwares tend to leave the determination of the cut angle to end users, and the end users make the decision depending on experiences instead of machine tool's kinematic performances. The experiments themselves have their value for pointing out the determination of cut angle is another source to optimize the tool path.

While the work presented in this paper established and validated the MKM method, there are several remaining challenges. Due to the extensive amount of work required, these tasks will be accomplished and presented in future reports. The following is a partial list of the future tasks:

- In addition to the axis velocity constraints, include axis acceleration constraints to cover both aspects of machine kinematic capabilities.
- Incorporate Scallop Height Metric with MKM, in order to achieve the optimization with dual objectives: machine kinematic capabilities and machining width.
- Besides zig-zag tool paths, extend the numerical computation to more complex path patterns, including spiral and boundary conforming patterns.

Acknowledgement

Wuhan Huazhong Numerical Control Co., Ltd provided the HNC-8 emulator that allows us to carry out the experiments. Dr. Liu, Xu (NUAA) and Mr. Guo, Xiao (HUST) helped checked the numerical computation in Section 7 with MATLAB. Mr. Guo also helped with the HNC-8 experiments. The authors gratefully acknowledge the support of the National Science and Technology Major Project of the Ministry of Science and Technology of China (2013ZX04007041) and the National Natural Science Foundation of China (51575386). We also want to thank the financial support of Wuhan Institute of Technology.

ORCID

Chen-Han Lee http://orcid.org/0000-0002-7017-0078
Changya Yan http://orcid.org/0000-0002-1614-0193

References

[1] Beudaert, X.; Pechard, P.Y.; Tournier, C.: 5-Axis tool path smoothing based on drive constraints, International Journal of Machine Tools and Manufacture, 2011,

[2] Bieterman, M.B.; Sandstrom, D.R.: A curvilinear toolpath method for pocket machining, Journal of manufacturing science and engineering, 125, 2003, 709. http://dx.doi.org/10.1115/1.1596579

[3] Chiou, C.J.; Lee, Y.S.: A machining potential field approach to tool path generation for multi-axis sculptured surface machining, Computer-Aided Design, 34(5), 2002, 357–371. http://dx.doi.org/10.1016/S0010-4485(01)00102-6

[4] Chuang, J.J.; Yang, D.C.H.: A laplace-based spiral contouring method for general pocket machining, The International Journal of Advanced Manufacturing Technology, 34(7), 2007, 714–723. http://dx.doi.org/10.1007/s00170-006-0648-6

[5] Ding, S.; Mannan, M.; Poo, A.N.; Yang, D.; Han, Z.: Adaptive iso-planar tool path generation for machining of free-form surfaces, Computer-Aided Design, 35(2), 2003, 141–153. http://dx.doi.org/10.1016/S0010-4485(02)00048-9

[6] Eigensatz, M.; Pauly, M.: Positional, metric, and curvature control for constraint-based surface deformation, Computer Graphics Forum (Proc. Eurographics 2009), 28(2), 2009, 551–558. http://10.1111/j.1467-8659.2009.01395.x

[7] Elber, G.; Cohen, E.: Toolpath generation for freeform surface models, Computer-aided Design, 26(6), 1994, 490–496. http://dx.doi.org/10.1145/164360.164500

[8] Fenkl, H.; Probst, M.; Slepcevic, P.: Milling method, in, ALSTOM(Switzerland)Ltd, Baden(CH), 2002,

[9] Gray, P.; Bedi, S.; Ismail, F.: Rolling ball method for 5-axis surface machining, Computer-Aided Design, 35(4), 2003, 347–357. http://dx.doi.org/10.1016/S0010-4485(02)00056-8

[10] Gray, P.J.; Bedi, S.; Ismail, F.: Arc-intersect method for 5-axis tool positioning, Computer-Aided Design, 37(7), 2005, 663–674. http://dx.doi.org/10.1016/j.cad.2004.08.006

[11] Hu, P.; Tang, K.: Five-axis tool path generation based on machine-dependent potential field, International Journal of Computer Integrated Manufacturing, 2015,

[12] Ju, T.; Losasso, F.;Schaefer, S.;Warren, J.: Dual contouring of hermite data, ACM Trans. Graph., 21(3), 2002, 339–346. http://dx.doi.org/10.1145/566654.566586

[13] Jun, C.-S.; Cha, K.; Lee, Y.-S.: Optimizing tool orientations for 5-axis machining by configuration-space search method, Computer-Aided Design, 35(6), 2003, 549–566.

[14] Kim, T.; Sarma, S.E.: Toolpath generation along directions of maximum kinematic performance; a first cut at machine-optimal paths, Computer-Aided Design, 34(6), 2002, 453–468. http://dx.doi.org/10.1016/S0010-4485(01)00116-6

[15] Kim, T.; Sarma, S.E.: Optimal sweeping paths on a 2-manifold: a new class of optimization problems defined by path structures, Robotics and Automation, IEEE Transactions on, 19(4), 2003, 613–636. http://dx.doi.org/10.1109/TRA.2003.814497

[16] Kim, T.: Constant cusp height tool paths as geodesic parallels on an abstract Riemannian manifold, Computer-Aided

Design, 39(6), 2007, 477–489. http://dx.doi.org/10.1016/j.cad.2007.01.003

[17] Lasemi, A.; Xue, D.; Gu, P.: Recent development in CNC machining of freeform surfaces: A state-of-the-art review, Computer-Aided Design, 42(7), 2010, 641–654. http://dx.doi.org/10.1016/j.cad.2010.04.002

[18] Liu, X.; Li, Y.; Ma, S.; Lee, C.-h.: A tool path generation method for freeform surface machining by introducing the tensor property of machining strip width, Computer-Aided Design, 66, 2015, 1–13. http://dx.doi.org/10.1016/j.cad.2015.03.003

[19] Makhanov, S.S.; Ivanenko, S.A.: Grid generation as applied to optimize cutting operations of the five-axis milling machine, Applied numerical mathematics, 46(3), 2003, 331–351. http://dx.doi.org/10.1016/S0168-9274(03)00039-4

[20] Morishige, K.; Kase, K.; Takeuchi, Y.: Tool path generation using C-space for 5-axis control machining, Journal of manufacturing science and engineering, 121(1), 1999, 144–149. http://dx.doi.org/10.1115/1.2830567

[21] Redonnet, J.-M.; Djebali, S.; Segonds, S.; Senatore, J.; Rubio, W.: Study of the effective cutter radius for end milling of free-form surfaces using a torus milling cutter, Computer-Aided Design, 45(6), 2013, 951–962. http://dx.doi.org/10.1016/j.cad.2013.03.002

[22] Yau, H.-T.; Tsou, L.-S.: Efficient NC Simulation for Multi-Axis Solid Machining With a Universal APT Cutter, Journal of Computing and Information Science in Engineering, 9(2), 2009, 021001–021001. http://dx.doi.org/10.1115/1.3130231