C-space approach to tool-path generation for die and mould machining

Byoung K Choi*, Dae H Kim and Robert B Jerard†

Presented in the paper is a new approach to 3-axis NC tool-path generation for sculptured surface machining. In the proposed C-space approach, the geometric data describing the design-surface and stock-surface are transformed into C-space elements, and then, all the tool-path generation decisions are made in the configuration space (C-space). The C-space approach provides a number of distinctive features suitable for high speed machining of dies and moulds, including:

(1) gouge-free and collision-free tool-paths;
(2) balanced cutting-loads;
(3) smooth cutter movements; and
(4) verification mechanisms. It is suggested, with some supportive results, that the Z-map model might be a suitable representation scheme for implementing the C-space method. Also discussed are other implementation as well as future research directions.

© 1997 Elsevier Science Ltd.

Keywords: tool-path generation, high speed machining, sculptured surface machining, die-cavity machining, C-space, Z-map model

INTRODUCTION

High Speed Machining (HSM) is now recognized as one of the key manufacturing technologies for higher productivity and as a result, the HSM technology has increasingly been utilized in die-cavity machining where 3-axis NC milling operations are mostly employed. Machine tool technology has made a large step forward in recent years, and now spindle speeds of 40000 rpm and feed rates of 50 m/min are no longer out of reach, which, from the die and mould manufacturing point of view, may already be more than can be effectively utilized. Even though much more conservative cutting conditions are usually employed in practice, the cutting speeds typically used in HSM are about ten times higher than the speeds applied in conventional machining. However, there are two major issues that still limit the application of HSM in the die-making industry: cutting tool material and tool-path generation on sculptured surfaces.

Of the two issues, this paper is concerned with tool-path generation for machining dies and moulds on high speed NC machines. As the conventional tool-path generation methods have inherent limitations in handling the HSM requirements, we propose a new approach to 3-axis NC tool-path generation that has a great potential to meet the challenges of HSM technology.

The paper is arranged as follows: the next section contains some of the basic concepts and definitions that are used throughout the paper. A brief review of the existing tool-path generation methods and a list of key requirements for high speed machining are given in the two sections that follow. The fifth section contains a detailed description of the C-space method itself. Some implementation issues and preliminary results are presented in the sixth section, and concluding remarks and future research directions are given in the final section.

BASIC CONCEPTS AND DEFINITIONS

Presented in this section are basic concepts and definitions that are to be used throughout the paper. In die-cavity machining, a specific pattern of machining operation performed by a cutting tool is called a unit machining operation (UMO) and a number of UMOS are employed to make a die. In this paper, the term design surface will be used to refer to the mathematical surface of a die or mould as specified by the designer, while the term stock surface is used to refer to the surface of the raw-stock or in-process workpiece upon which a UMO is applied. The term machined surface is used to refer to the actual surface generated by a particular UMO. The term machined surface is used to refer to the actual surface generated by a particular UMO. Refer to Figure 1. It should be noted that the machined-surface of a UMO becomes the stock-surface for the next UMO.

Definition 1 (allowance)

At each UMO, the stock-allowance is defined as the volume difference (or thickness) between the stock-surface and the
Tool-path generation: B K Choi et al.

Figure 1 Surfaces and allowances for a unit machining operation

design-surface, while the uncut-allowance is specified as an
intentional uncut (and becomes the distance from the 'ideal'
machined-surface to the design-surface). The cutting-depth
at a UMO is given by the difference between the stock-
allowance and the uncut-allowance as depicted in Figure 1.

Definition 2 (load)

Chip-load is defined as the amount of chip produced during
a unit length of tool-movement, while cutting-load is
defined as the reactive cutting forces being applied to the
tool. The chip-load value may be expressed as the cutter-
engagement-area. Cutting-load is dependent on chip-load as
well as on cutting conditions.

Definition 3 (CL)

As shown in Figure 2a, a point on the surface at which the
cutter makes a tangential point is called the cutter-contact
point (CC-point), while the reference point of the
'on-contact' cutter is called the cutter-location point (CL-
point). A CL-line is defined by the line segment joining to
CL-points, and a CL-path is defined as a sequence of CL-
lines. The CL-surface for a given surface is defined as the
trajectory of the cutter's reference-point when the cutter is
slid over the entire surface (Figure 2b).

Definition 4 (gouge, uncut, collision)

Gouge refers to the over-cutting of the workpiece during the
cutting-mode (G01 mode), while collision refers to
accidental contacts between machining tool elements and
workpiece elements during the traverse mode (G00 mode).
A concave-gouge may occur at a CL-point in a concave
region (Figure 3a), while a convex-concave gouge may
occur on a CL-line in a convex region (Figure 3b). The term uncut
refers to the under-cutting of the workpiece: a
concave-uncut occurs at a concave region when the cutter
is too large to fit into the concave region without a concave-
gouge (Figure 3c); a convex-uncut, which is better known
as a cusp, occurs between adjacent CL-paths.

In the literature, the term 'cutting-load'\(^\text{20}\) is often used to
imply both the chip-load and the cutting-load, but we have
defined them separately (in Definition 2) as they play an
important role in HSM. The term 'part-surface'\(^\text{11}\) is often
used to refer to the ideal machined surface given by the
offset surface of the design-surface (offset by the uncut-
allowance). The terms 'tool-path' and 'cutter-path' are
often used interchangeably, implying a CL-path or a CC-
path. The length of a CL-line is usually called 'step-for-
ward' or 'step-length', and the spacing between adjacent
CL-paths is called 'pick-feed', 'step-over distance of path-
interval'. The terms gouge, uncut and collision in Definition
4 are collectively referred to as cutter-interference.

CONVENTIONAL TOOL-PATH GENERATION
METHODS

As discussed earlier, a specific pattern of machining
operations performed by a given cutting tool is called a
UMO. In this section, existing methods for generating NC-
data for a given UMO are briefly reviewed. In general, NC
tool-paths for a UMO are generated in three steps:

1. tool-path planning;
2. CL-points computation;
3. tool-path linking.

According to how CL-points are obtained the existing
tool-path generation methods may be categorized as either
the CC-point approach or the direct positioning approach.

CC-point approach

In the CC-point approach, tool-paths are obtained in two
steps:

1. tool-paths are planned on the design surface; and
2. the CL-points are computed from the CC-points on the
tool-paths.

Furthermore, according to how the CC-points are planned,
the 'CC-point based' tool-path generation methods are
classified as follows:

1. APT method: Tool-paths are defined along the inter-
section curves between the design-surface and drive
surfaces. This method is the basis of the APT system\(^\text{11}\).
2. Isoparametric method: Tool-paths are often given as
isoparametric curves, separately on each of the para-
meter-domains of the surface patches in the design-surf-
ace\(^\text{21}\).
3. Highest z-value method: Tool-paths are defined on the
xy-domain of the design surface, and then the highest z-
value for each xy-point is computed later\(^\text{1}\).
4. Iso-curvature methods: Tool-paths are planned on the
design surface along the lines of curvature\(^\text{16}\) or along
some other characteristic lines\(^\text{35}\).

Direct positioning approach

In this approach, tool-paths are planned also in two steps:

1. 2D tool-paths are planned on the horizontal guide-
plane; and
2. individual tool positions are lowered until they touch
the part-surface.
The term check-surface\(^2\) is often used to refer to the surface that should not be gouged, and the term ‘checking’ is used for ‘preventing a surface from gouging’. Depending on how the checking is made, we have:

1. **CC-checking method**: The cutter (its CC-point) is checked against the design-surface.
2. **CL-checking method**: The CL-point is checked against the CL-surface.

When the design surface (or its offset surface) is used as a check surface, it is usually represented by a discretized surface model such as the surface-point set model\(^19\), the Z-map model\(^3\), or the polyhedral model\(^10\). The CC-checking method has recently been widely adopted in the major CAM systems specializing in die-cavity machining. The CL-checking method was first introduced as the *inverse offset method*\(^24\).

**REQUIREMENTS FOR HIGH SPEED MACHINING OF DIES**

Earlier we mentioned that, with the current HSM technology, the die-cavities can be machined more than ten times faster. It is analogous to a high speed drive with the legal speed limited suddenly increased by a factor of ten (650 mph or 1000 km/h!). The following requirements for HSM have been identified by the authors based on a close observation of the industrial practices of high speed machining.

1. **Collision avoidance**: Even a minor collision would become fatal in HSM, damaging the machine tool and workpiece.
2. **Chip-load leveling**: The workpiece would be no time to adjust for an abrupt jump in the chip-load, resulting in cutter breakage.
3. **Cutting-load smoothing**: The fluctuations in chip-loads should be estimated a priori so that feedrate can be adjusted adaptively to maintain a smooth cutting-load (to avoid chatter).
4. **Smooth tool-path**: Sharp turns in cutter motion would push the cutter off the course, leaving ‘tool-marks’ on the machined surface and resulting in out-of-tolerance areas.
5. **Verification mechanisms**: Even with the NC-codes prepared with extreme care, it is a must in HSM to verify them before actual machining. It should be emphasized that the above requirements are also required in ‘low-speed’ machining, but in high speed machining, there is little time for human intervention in the event of a problem, even if the operator is watching the machining process.

In summary, the key items related to the requirements for the HSM of dies and moulds, are: gouging, collision, chip-load, cutting-load, smooth tool-path, and verification. None of the conventional tool-path generation methods seem to be suitable for the HSM application. Most of all, the drawback of the CC-point approach is that it is prone to gouging, and the direct positioning approach is good only for preventing concave-gouging (but nothing much more for HSM). Thus, as a supplement to the tool-path generation systems, various methods of cutting simulation and verification\(^18,25\) have been utilized in die-cavity machining.

**C-SPACE APPROACH TO TOOL-PATH GENERATION FOR HIGH SPEED MACHINING OF DIES AND MOULDS**

As discussed earlier, the conventional tool-path generation methods are not suitable for high speed die-machining applications even with the help of simulation and verification mechanisms. The fundamental limitation of the conventional methods may come from the fact that:

1. tool-paths are generated utilizing only ‘local’ information; and
2. the geometric information from which the tool-paths are generated are insufficient for handling the high speed machining (HSM) requirements.

Thus, we need a tool-path generation method which is based on a ‘global’ decision-making logic utilizing sufficient information. This requirement seems to be met by the C-space approach to be presented in this section. First, a descriptive definition of C-space is given as follows:

**Definition 5 (C-space)**

A set \(Q\) is called a configuration space (C-space) for a system if every element of \(Q\) corresponds to a valid configuration of the system and each configuration of the system can be identified with a unique element of \(Q\) (For example, refer to p.25 in Reference\(^26\)).

The configuration of a 3-axis NC machine system is specified by a 3D position vector (denoting the positions of the \(x\), \(y\), and \(z\)-axis), while its C-space is given by the volume \(V_{3D}\) in a Cartesian space within the reach of the cutting tool. Thus, the first requirement for any surface machining is
that its CL-surface be properly contained within the C-space of the NC machine. In practice, this may be an important criterion for selecting machines and cutters, but this aspect will not be considered in this paper (assuming that the machine is big enough for the job).

The C-space approach to tool-path generation

The concept of C-space has long been applied to a range of spatial planning problems in connection with the automatic planning of manipulator transfer movements. Even though the term 'C-space' has rarely been used in die-cavity machining, the C-space approach is not completely new in the field of tool-path generation: It is an outgrowth of the 'soft master-model method' which in turn is an extension of the CL-checking method.

The C-space approach to spatial planning for manipulator transfer movements may be summarized as follows:

1. Find a safe C-space of a moving object 'A' in the presence of a set of obstacles 'B'.
2. Find safe (and good) paths for A within the safe C-space.

Here, the safe C-space denotes a set of configurations of the moving object satisfying the condition that no overlap between the object A and the obstacle B', namely $A \cap B' = \emptyset$.

The same C-space idea is easily applied to the (3-axis) tool-path planning problem if the cutting tool is treated as the moving object and the workpiece plays the role of the obstacle. In tool-path planning, however, we need to have two types of safe C-space:

1. Free C-space: safe C-space denoting the 'free-of-collision' space.
2. Machining C-space: safe C-space denoting the 'machining' space.

Now let's consider the two CL-surfaces, one for the design surface and the other for the stock surface, as depicted in Figure 4:

1. $S_D = $ design CL-surface (CL-surface for the design surface).
2. $S_S = $ stock CL-surface (CL-surface for the stock surface).

Then, as shown in Figure 4, the two CL-surfaces would divide the entire C-space of the NC machine into the three disjoint C-spaces as follows:

1. $V_F = $ free C-space (space above $S_S$ exclusive);
2. $V_M = $ machining C-space (space between $S_S$ and $S_D$ inclusive);
3. $V_G = $ gouging C-space (space below $S_D$ exclusive).

For a given die-machining UMO we are given:

1. the design surface of the die;
2. its stock surface;
3. its uncut-allowance;
4. cutter geometry.

We want to determine the following C-space elements within the C-space of NC-machine ($V_{NC}$):

C-space Elements: $\{S_S, S_D, V_F, V_M, V_G\}$

such that the volume-type C-space-elements satisfy

$V_{NC} = V_F \cup V_M \cup V_G$ with $\emptyset = V_F \cap V_M \cap V_G$.

In summary, the overall procedure for obtaining the C-space elements is as follows:

1. The stock CL-surface ($S_S$) is obtained from the stock surface.
2. The design CL-surface ($S_D$) is obtained from the design surface taking into account the uncut-allowance.
3. The C-space volume elements ($V_F, V_M, V_G$) are obtained from the CL-surfaces.

As will be discussed shortly, the above C-space elements contain all the geometric information necessary for generating tool-paths for the UMO. Finally, the C-space approach to tool-path generation may be summarized as follows:

2. Generate tool-path from the C-space elements.

The question is then how do we implement the C-space method, which is the subject of the next section. However, before going into implementation details, the remainder of this section will be devoted to characterizing the C-space method and comparing it with the traditional methods.

Characteristics of the C-space approach

Now we shall present more detailed characteristics as well as basic properties of the C-space approach in connection with HSM. We will describe those characteristics with 'finishing or semi-finishing' type endmilling operations in mind. To begin with, we present some propositions related to the property of the CL-surface.

Proposition 1

There are always exists a unique CL-surface for a 3-axis endmill.
Table 1 Comparison of characteristics among different tool-path generation methods

<table>
<thead>
<tr>
<th></th>
<th>Isoparametric method</th>
<th>CC-checking method</th>
<th>C-space method</th>
<th>Cutting simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concave-gouge</td>
<td>poor</td>
<td>prevent</td>
<td>prevent</td>
<td>detect</td>
</tr>
<tr>
<td>Convex-gouge</td>
<td>poor</td>
<td>fair</td>
<td>prevent</td>
<td>detect</td>
</tr>
<tr>
<td>Rapid-collision</td>
<td>poor</td>
<td>fair</td>
<td>prevent</td>
<td>detect</td>
</tr>
<tr>
<td>Concave-uncut</td>
<td>poor</td>
<td>N/A</td>
<td>detect</td>
<td>detect</td>
</tr>
<tr>
<td>Path smoothness</td>
<td>poor</td>
<td>fair</td>
<td>good</td>
<td>detect</td>
</tr>
<tr>
<td>Chip-load balancing</td>
<td>N/A</td>
<td>N/A</td>
<td>balance</td>
<td>detect</td>
</tr>
<tr>
<td>Cutting-loads</td>
<td>N/A</td>
<td>N/A</td>
<td>fair estimate</td>
<td>good estimate</td>
</tr>
<tr>
<td>Feature extraction</td>
<td>compatible</td>
<td>N/A</td>
<td>compatible</td>
<td>N/A</td>
</tr>
<tr>
<td>Compound-surface</td>
<td>one patch</td>
<td>OK</td>
<td>OK</td>
<td>OK</td>
</tr>
</tbody>
</table>

N/A: not applicable or cannot be handled.

Proposition 2

The CL-surface of the surface S for a 3-axis endmill is equivalent to the inverse tool-offset (ITO) surface of S (See Figure 5).

Proposition 3

If the ball-endmill CL-surface is $G^1$-continuous, there will be no concave-uncuts when the part is machined with the NC-data obtained from the CL-surface.

Proposition 1 asserts that the C-space approach will always produce a unique and valid CL-surface, while Proposition 2 provides a robust method for generating the CL-surface. (In contrast, the conventional CC-point approach cannot always generate valid CL-points because the CL-point may not be well defined for certain CC-points.) The ITO-surface is defined as the envelope of the 'cutter swept volumes' obtained by sweeping the 'inverse' cutter over the entire surface with its reference point keep on the surface (i.e. the inverse offset method for generating the ITO-surface). Proposition 3 is useful for clean-up cut planning: If the CL-surface is $G^1$-continuous, no clean-up cut is necessary; Otherwise, clean-up cutting operations may be needed along the sharp edges of the CL-surface. The following is a generalization of Proposition 3.

Proposition 4

In ball-endmilling, the chip-loads can be estimated from the differential properties of the design CL-surface ($S_D$) when the cutting-depth is uniform: Chip-loads are high in concave regions and low in convex regions.

Unlike the previous propositions, Proposition 4 needs further investigation. Nevertheless, we postulate in the proposition that sufficient geometric information for computing the chip-loads is imbedded in the CL-surface (when the cutting-depth is uniform) and that a method for computing the chip-loads could be developed.

Recall that the key words for high speed machining of dies and moulds are gouging, collision, chip-load, cutting-load, smooth tool-path, and verification. Given in the following are brief descriptions of how the (HSM) requirements might be handled, at least partially with the C-space method:

1. **Gouge-free machining**: There will be no gouging if all the cutting-moves (G01 NC-blocks) are kept out of the gouging C-space $V_G$.
2. **Uncut handling**: The concave-uncut can be detected and clean-up cutting operations may be planned along the sharp edges of the design CL-surface $S_D$ (c.f. Proposition 3).
3. **Collision avoidance**: The 'cutter' collisions can be avoided if all the traverse-moves (G00 NC-blocks) are confined with the free C-space $V_F$.
4. **Chip-load leveling**: When $S_D$ has sharp-edges, the chip-load leveling may be achieved via pencil-trace cutting operations along the sharp edges (c.f. Proposition 4).
5. **Cutting-load smoothing**: Cutting-loads may be smoothed out by adjusting feedrates (increased at convex regions and decreased at concave regions) according to the radii of curvature of $S_D$ (c.f. Proposition 4).
6. **Smooth tool-path**: It is always possible to generate smooth tool-paths by tracing out smooth curves on the CL-surface $S_D$.
7. **Verification mechanisms**: The existing NC-data can be verified in the C-space with respect to gouging, collision, uncut, cutter-loads, or air-cut.
8. **Machining features**: In general, extracting machining features for die-cavity machining would be easier with the C-space representation.

Assessment of the C-space method

Having identified the main characteristics of the C-space method, it would be appropriate to assess the advantages (or disadvantages) of the C-space method compared to the traditional tool-path generation methods. Also of interest is the role of cutting simulation in the C-space method.

Summarized in Table 1 is how well the above HSM requirement items might be handled by the most commonly used traditional methods. The tool-path generation methods considered are the isoparametric method and the CC-checking method together with the C-space method, and the requirement items considered are:

1. gouging and collision handling;
2. uncut handling;
3. tool-path smoothing;
4. loads balancing;
5. feature extraction;
6. compound surface machining.

Also shown in the table is the role of cutting simulation. The following comparative observations could be made from the contents of the table, even though the exactness of the individual assessments may need a further clarification.

1. The isoparametric method may not be suitable for die and mould machining because it cannot handle multi-patched surfaces.
The CC-checking method, which is most widely used in major CAD/CAM systems, prevents concave gouging, but is not good enough to handle other HSM requirements.

Cutting simulation can be used to compensate for some of the weaknesses of the CC-checking method, but not the feature extraction requirement.

Cutting simulation is effective in detecting errors, but it can not prevent nor correct them by itself.

The main advantage of the C-space method is to prevent most of the errors.

For a better smoothing of cutting-loads, the C-space method may be supplemented with cutting simulation, which usually is a very time-consuming process.

At the beginning of this section, it has been argued that the limitation of the conventional methods comes from the fact that:

1. Tool-paths are generated utilizing only ‘local’ information.
2. The geometric information from which the tool-paths are generated is insufficient for handling the HSM requirements.

Conversely, the power of the C-space method comes from the fact that it is based on a ‘global’ decision-making logic utilizing sufficient information.

In order to clarify these arguments, the basic nature of tool-path generation has to be understood. First of all, it
should be understood that a tool-path generation problem would require comprehensive geometric information involving:

(1) Design-surface geometry.
(2) Cutting-tool geometry.
(3) Stock-surface geometry.

But, none of the traditional methods make full use of the above three. For example, only the local geometry of the design-surface (i.e. position and normal vectors at the CC-point) is being utilized by the isoparametric method. Similarly, the CC-checking method relies only on the height values of the design-surface right below the cutting tool.

On the other hand, the C-space method is categorized as a 'global' method because it makes use of all the three sources of geometric information. Namely, the C-space method is a global method because:

(1) both the design-surface geometry and the cutting-tool geometry are imbedded in design CL-surface $S_D$, and
(2) the stock-surface geometry is preserved in stock CL-surface $S_S$.

IMPLEMENTATION ISSUES AND EXAMPLES

The potential benefits of the C-space approach can only be realized through a reliable and efficient implementation scheme. For a successful implementation, we need to have:

(1) A compact scheme for representing the C-space elements.
(2) An efficient method for computing the C-space elements.
(3) Algorithms for generating tool-paths in C-space.

In this section, the above implementation issues will be examined for the design CL-surface.

The design CL-surface ($S_D$) is the most important C-space element because most of the geometric information needed for tool-path generation is contained in it. In looking for a suitable representation scheme for $S_D$, the following items have to be considered:

(1) The CL-surface ($S_D$) can be represented in nonparametric surface form: $z = f(x,y)$.
(2) The CL-surface can be obtained by computing the inverse tool-offset (ITO) surface.
(3) It is required to trace the sharp edges of $S_D$ (for chip-loading leveling).
(4) It is required to compute curvatures of $S_D$ (for cutting-load smoothing).
(5) It is required to extract some machining features from $S_D$ (for tool-path planning).
(6) It is required to intersect $S_D$ with planes (for tool-path generation).

Representation and computation of C-space elements

It turns out that the above requirements are quite effectively handled by the Z-map model which is a special form of discrete non-parametric surface model. It is a 2D-array
of real numbers in which the z-values of the surface sampled at the regular 'gridpoints' are stored. With the Z-map representation, the ITO-surface may be easily obtained from the inverse offset method \textsuperscript{14} by employing one of the cutting simulation algorithms \textsuperscript{17,9}.

Presented in the following is an example of CL-surface implementation based on an 'edge-extended' Z-map model which we call EZ-map model. As shown in Figure \textit{6}, a fixed number of z-values are adaptively sampled, if necessary, from the 'grid-edges' that are located in the near-vertical (or sharp-corner) regions. As a result, the grid-solution at the near-vertical regions can be selectively increased from the 'z-map grid interval' to the 'e-map grid interval'.

Shown in Figure \textit{7} is a CAD model for the design surface of a stamping die for the fuel tank of a passenger car, and its 'master' Z-map model is shown in Figure \textit{8}. In the example case, the EZ-map model shown in Figure \textit{8} is defined as follows:

1. Size of the Z-map domain: 1100 mm by 730 mm (in x- and y-directions).
2. z-map grid interval = 1.0 mm.
3. e-map grid interval = 0.1 mm.
4. Memory size = 7.07 Mbytes.

The procedure for generating a Z-map model from a CAD is called z-map sampling or (virtual) digitizing. Running on a SGI workstation (Indigo2 XZ), it takes about 50 min to digitize the surface and store the data (over 1 million points) in the EZ-map.

Shown in Figure \textit{9} is the design CL-surface for a ball-endmill of 30 mm in diameter. It takes about 34 min to generate it from the master model of Figure \textit{8} and to store in the same EZ-map form (using the algorithm of Reference \textsuperscript{8}). Concave type sharp-edges (or pencil curves) in the CL-surface are displayed in Figure \textit{10}, which takes about 4 min to compute (using the method of Reference \textsuperscript{17}). Finally, a color display of 'maximum curvatures' of the CL-surface is shown in Figure \textit{11}. The radius of maximum curvature at a point is approximated by the radius of the smallest circle among the ones fitted along different surface-directions. Convex regions are denoted by red colors and convex regions are denoted by blue colors. The intensity (or darkness) of the colors is related to the value of radius of curvature. The green areas denote flat or saddle regions. It took about 30 seconds to obtain the curvature display.

Even though the above illustration does not represent a full implementation of the C-space method, it does demonstrate that the C-space method can be implemented using the EZ-map representation. That is, the two CL-surfaces (S\textsubscript{c} and S\textsubscript{d}) are represented by respective EZ-maps, and the volume-type C-space elements (V\textsubscript{p}, V\textsubscript{M}, and V\textsubscript{D}) are defined by the CL-surface. (The precision of the EZ-map was enough for practical purposes since the stamping die was successfully machined with the NC-codes that had been generated from the CL-surface by using the CL-checking method).

Since the Z-map model is recommended as a candidate for C-space representation scheme, it might be useful to elaborate on the basic concept behind the Z-map method. Sometimes, the term 'G-buffer method' \textsuperscript{30} is also used. A Z-map is nothing more than a 2D array of numbers in which the only the z-values of a (compound) surface are stored. It may be the most compact form of storing (virtually)...
digitized data and yet it has a number of advantages. The \( x, y \) coordinates of the \( Z \)-map element \( z[i,j] \) are computed as:

\[
\begin{align*}
x &= x_0 + \Delta_x i \\
y &= y_0 + \Delta_y j
\end{align*}
\]

where \( x_0 \) and \( y_0 \) denote the coordinates of the 'bottom-left' corner point of the \( Z \)-map domain, and \( \Delta_x \) and \( \Delta_y \) denote the grid-intervals. The \( Z \)-map surface is easily evaluated by using a non-parametric interpolation method. Some details on the subject may be found in Reference. Even with its inherent limitations in coping with vertical walls and sharp edges on the surface, the \( Z \)-map representation has been used in machining stamping dies. For example, Ford and Chrysler are currently using \( Z \)-map based CAM systems in machining their dies.

For a 1000 by 1000 die, a standard \( Z \)-map with 1.0 mm grid-interval (GI) would require 4 Mbytes of memory because 4 bytes are needed for a real number. If the \( z \)-map resolution is increased ten times (GI = 0.1), the \( Z \)-map would require 400 Mbytes of memory! However, the EZ-map may provide the same precision (with 0.1 mm of e-map GI) with about 8 Mbytes.

**Tool-path generation using C-space elements**

The overall procedure for generating dependable NC toolpaths for die and mould machining usually consists of:

1. The process planning stage to obtain a set of UMOP plans.
2. Technical planning stage to select a tool-path topology and milling strategy.
3. CL-path generation by determining the step-length and path interval.
4. Consolidation stage to remove cutter-interference and determine cutting conditions.
5. NC verification stage to verify and confirm the resulting NC tool-paths.

The above five-stage procedure for tool-path generation (TPG) is basically the same for both the traditional TPG methods and the C-space method. However, the third stage, the CL-path generation stage, may be somewhat different from among different TPG-methods. In the following, a CL-path generation algorithm for a \( Z \)-map CL-surface will be presented for a finish UMO. Let's assume that the following are given:

1. \( r \): design \( CL \)-surface (\( Z \)-map) a given ball-endmill of radius \( \rho \) as in Figure 9.
2. \( C_j \): a set of pencil-curves as shown in Figure 10.
3. \( \omega, \lambda_{\text{max}} \): cutting-width (pick-feed distance in 3D) and maximum step-length.
4. \( \varepsilon \): machining-tolerance.
5. Tool-path topology and milling option = xy-parallel and oneway ball-endmilling.
6. Tool start position \((x_0,y_0)\): feed-forward direction \((+x)\), pick-feed direction \((+y)\), etc.
7. \( z = h(x,y) \): \( Z \)-map height evaluation function.
8. \( \alpha = g(x,y,v) \): \( Z \)-map gradient (\( \alpha \): inclination angle) in the direction of \( v \).
9. \( R = c(x,y,v) \): \( Z \)-map curvature (\( R \): radius of normal curvature) in the \( v \) direction.

Then, the procedure for generating finishing tool-path may be described as follows:

**TPG-algorithm (xy-parallel oneway-finishing):**

1. Generate relief pencil-cutting tool-paths from the pencil-curves \( \{C_j\} \).
2. Initialize: \( x = x_0; y = y_0; \Delta y = \omega \).

---

**Figure 10** Concave-type sharp-edges in the CL-surface
(3) Repeat \(/ / m a r c h i n g a l o n g t h e c u r r e n t p a s s / /\)
(3.1) Output a CL-point \([x, y, z = f(x, y)]\).
(3.2) \(R_t = c(x, y, + x)\): radius of curvature in the feed-forward direction.
(3.3) \(\lambda = \min \{\lambda_{\max}, 2[2\pi R_t]^{1/2}\}\): step-length (from circulate approximation).
(3.4) \(\alpha_t = g(x, y, + x)\): inclination-angle in the feed-forward direction.
(3.5) \(\alpha_p = g(x, y, + y)\): inclination-angle in the pick-feed direction.
(3.6) \(x = x + \omega \cos(\alpha_t)\): step in +x direction.
(3.7) \(\Delta y = \min \{\Delta y, \omega \cos(\alpha_p)\}\): update the pick-feed value.

\} until the end of current pass.

(4) Start the next pass: \(y = y + \Delta y, x = x_0, \Delta y = \omega\).
(5) If not completed go to Step 3, else stop.

As discussed earlier in this section, the differential properties (e.g. radius of normal curvature) of the CL-surface can be approximated from the Z-map points (e.g. by fitting a circle). But, a more general method for evaluating the height, gradient, and curvature of a Z-map (or EZ-map) is as follows: Each row and column of Z-map points is interpolated as a composite cubic curve and the surface of each Z-map cell is represented as a bicubic Coons patch\(^7\). Then, the functions \(z = h(x, y), \alpha = g(x, y, v), R = c(x, y, v)\) are easily evaluated from the bicubic Coons patch.

As an illustrative example, we present the result of applying the above TPG-algorithm to the finish machining of the fuel-tank stamping-die of Figure 7. The following values are used in the current illustrative example:

1. \(\rho = 15\) mm (ball-endmill).
2. \(\{C_i\}\) a set of pencil-curves shown in Figure 10.
3. \(\omega = 6\) mm (a large value is specified for display purposes; \(\omega = 0.7\) mm in practice).
4. UMO = xy-parallel area-cut [zigzag, BEM30Φ].
5. \(\delta = 5\) mm (cutting-depth = stock-allowance).

Note that the zigzag milling-strategy is used in this example (instead of oneway). Depicted in Figure 12 is the NC tool-path for a relief pencil-cutting (the dotted lines denote rapid tool-moves), and the result of this pencil-cutting is shown in Figure 13. The main finish-cut tool paths are displayed in Figure 14, where only a quarter of the die-surface is considered (to enhance the display).

If the cusp-height \(\eta\) is given, instead of the cutting-width \(\omega\), a formula for obtaining \(\omega\) from \(\eta\) may be derived from the well-known cutting-width formula\(^7\):

\[
\omega_{CC} = \frac{R_c}{(R_c + \rho)(R_c + \eta)^{3/2}} \sqrt{(4R_c + \rho)^2(R_c + \eta)^2 - \left[R_c^2 + 2R_c\rho + (R_c + \eta)^2\right]^2}
\]

where \(R_c\) is the radius of normal curvature in the pick-feed direction and \(\omega_{CC}\) is the cutting-width, all defined on the CC-surface (i.e. part-surface). If the following substitutions are made in the above formula

\[
\omega = \omega_{CC}(R_c + \rho)R_c \quad \text{and} \quad R = R_c + \rho
\]

we obtain an expression for finding the ‘CL-surface’ cutting-width \(\omega\) (from \(\eta\)) as follows:

\[
\omega = \frac{\sqrt{(4R^2(R - \rho + \eta)^2 - [R^2 - \rho^2 + (R - \rho + \eta)^2]^2}}}{R - \rho + \eta}
\]

where \(R = c(x, y, + y)\) is the radius of normal curvature of the CL-surface in the pick-feed direction, \(\rho\) is the ball-endmill radius (or effective cutter radius), and \(\eta\) is the maximum allowable cusp-height. In the above expressions, the radius of curvature is regarded as a positive number at a convex region and a negative number at a concave region.

As a final note, it should be noted that, as long as the
Figure 12  Relief pencil cutting tool-paths

Figure 13  Result of the relief pencil-cutting (cutting simulation)
CL-surface is valid, the above C-space based TPG-algorithm guarantees a gouge-free tool-path, which was not possible with the traditional TPG-methods. For a detailed discussion on the subject of ‘traditional’ gouge-avoidance, the reader is referred to Choi and Jun⁴. In addition, the relief pencil-cutting in Step 1 of the above TPG procedure helps balance chip-loads as well as cutting-loads. Furthermore, it would be possible to use the curvature values, along the feed-forward and pick-feed directions, in balancing cutting-loads (as pointed out in Proposition 4), but more research is needed. At any rate, for a more reliable result in balancing cutting-loads, one may have to rely on cutting simulation.

CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

The main objective of the paper has been to present a new approach to tool-path generation for high speed machining (HSM) of dies and moulds because the existing approaches are not suitable for handling the HSM requirements. It is postulated that the proposed C-space approach could (partially) meet all the HSM requirements. The main characteristics of the C-space approach include:

1. gouge-free and collision-free machining;
2. chip-load leveling (via relief cutting);
3. cutting-load smoothing (by adjusting feeds);
4. generation of smooth tool-paths;
5. verification mechanisms;
6. tool-path planning based on machining features.

The hard part of the C-space approach is finding an efficient implementation scheme. An extended Z-map called EZ-map is introduced as a candidate for representing the C-space elements (Sₜ, Sₚ, Vₕ, Vₘ, and Vₜ), and an illustrative example for a stamping-die of an automotive part is presented. At least for the similar type of dies, the EZ-map seems to have passed a preliminary test, in terms of its memory requirement and computation time. However, the proposed C-space approach is limited to 3-axis machining (up to now no consideration is given for 5-axis machining), and it has yet to go through a rigorous testing as well as theoretical analyses. As for the future research directions, one may think of the theoretical aspects of the C-space approach, implementation issues, and applications.

First of all, theoretical research is needed in such topics as:

1. mathematical foundation for the C-space approach to multi-axis NC machining;
2. fundamental properties (features) of the C-space method;
3. basic theorems (including proofs for the propositions);
4. bounding of errors introduced at discretization¹.

The research topics related to the implementation issues include:

1. discrete representation schemes for the C-space elements;
2. basic algorithms for the discrete models;
3. computational geometric issues.

Finally, there are a number of application areas that deserve further research effort:

1. adaptive feed control based on the C-space information;
2. C-space based feature-extraction for die-cavity machining;

668
(3) C-space based CAPP for die-cavity machining;
(4) optimization of tool-paths and cutting conditions utilizing the C-space information, etc.

ACKNOWLEDGEMENT

For this research, the first author was supported in part by the Korea Research Foundation.

REFERENCES

28. Private communication with Jeffrey G. Hemmett.
29. Private communication with J.W. Park.

Byoung K Choi is a Professor of Manufacturing Systems Engineering in the Department of Industrial Engineering at KAIST since he joined KAIST in early 1985. He received a BS from Seoul National University, a MS from KAIST, and a PhD from Purdue University, all in industrial Engineering. His research interests are in the area of sculptured surface machining, die-cavity machining, CAPP, system modeling and simulation, and virtual manufacturing.

Dae H Kim is a deputy manager of the Die-making division of Daewoo Electronics Ltd while currently studying at KAIST as a PhD Student. He has been working with Daewoo as a die design and manufacturing engineer since he got an MS from KAIST in 1984. His primary professional interests are in sculptured surface NC machining, CAPP for die-cavity machining, and CIM for die manufacturing.

Robert B Jerard is a Professor in the Mechanical Engineering Department at the University of New Hampshire. His research interests are in the area of generation, simulation and verification of machining programs for sculptured surface machining. He received his BS degree from the University of Vermont, MS from MIT and PhD from the University of Utah. He is a member of ASME, SME, SAE and the IEEE Computer Society.