Global Finish Curvature Matched Machining

Jianguo Wang
Brigham Young University - Provo

Follow this and additional works at: https://scholarsarchive.byu.edu/etd

Part of the Mechanical Engineering Commons

BYU ScholarsArchive Citation
https://scholarsarchive.byu.edu/etd/702

This Thesis is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in All Theses and Dissertations by an authorized administrator of BYU ScholarsArchive. For more information, please contact scholarsarchive@byu.edu, ellen_amatangelo@byu.edu.
of a thesis submitted by

Jianguo Wang

This thesis has been read by each member of the following graduate committee and by majority vote has been found satisfactory.

________________________ __________________________________________
Date     C. Greg Jensen, Chair

________________________ __________________________________________
Date     W. Edward Red

________________________ __________________________________________
Date     Spencer P. Magleby
As chair of the candidate’s graduate committee, I have read the thesis of Jianguo Wang in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative material including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

________________________ __________________________________________
Date     C. Greg Jensen
Chair, Graduate Committee

Accepted for the Department

Matthew R. Jones
Graduate Coordinator

Accepted for the College

Alan R. Parkinson
Dean, Ira A. Fulton College of Engineering and Technology
ABSTRACT

GLOBAL FINISH CURVATURE MATCHED MACHINING

Jianguo Wang
Department of Mechanical Engineering
Master of Science

As competition grows among manufacturing companies, greater emphasis has recently been placed on product aesthetics and decreasing the product development time. This is promoting and standardizing widespread use of sculptured surface styling within product design. Therefore, industries are looking for high efficiency machining strategies for sculptured surface machining (SSM). Many researchers have produced various methods in tool path generation for SSM. Five-axis curvature matched machining ($CM^2$) is the most efficient. With the widespread use of 5-axis mill in industries, $CM^2$ is a better solution for improving the machining efficiency for product concept models. $CM^2$ has very good performance for global machining of single patch surface or a quilt of simple sculptured surface patches. But when $CM^2$ is used to generate tool paths for global machining of a large region of complex sculptured surface such as the top or side skins of a vehicle, there will be some limitations, that is, the performance will be influenced greatly in some steep areas where the lead angle of the tool becomes larger to match the
curvature or avoid gouging. Larger lead angles mean smaller effective curvatures at the leading edge of the tool bottom where it contacts the part surfaces. Therefore, the density of CM² tool path is very high in these steep regions.

By setting a smaller upper limit for the lead angle, the density of tool path will not be very high in the steep regions, but there will be some uncut materials. This thesis focuses on how to determine the uncut or rework areas of the previous CM² and how to define the boundary of these regions. Strategies for generating more efficiency CM² tool paths are also discussed. These methods will be tested by applying finish global machining to a one-fourth scale Ford GT model.
ACKNOWLEDGMENTS

I would like to express my thanks to my advisor, Dr. C. Greg Jensen, for his guidance, support, and time spent in help me complete this research. I would like to thank the other members of my graduate committee, Dr. Edward W. Red and Dr. Spencer P. Magleby, for their kind help and encouragement. I would like to express appreciation to all the members of the Direct Machining and Control research group at BYU. I would like to thank Robert Cheatham for his time in helping me prepare the machining example of GT40.

Finally I would like to thank my wife, Huijun, for her inexhaustible support and encouragement. I am grateful to my daughter, Emily Wang, for permitting me to spend much time in writing this journal. I am especially thankful to my parents and parents-in-law for their support and sacrifices.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TABLE OF CONTENTS</td>
<td>VIII</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>XI</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>XIII</td>
</tr>
<tr>
<td>CHAPTER 1: INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>1.2 Thesis Objective</td>
<td>5</td>
</tr>
<tr>
<td>CHAPTER 2: LITERATURE REVIEW</td>
<td>7</td>
</tr>
<tr>
<td>2.1 The Application of Sculptured Surfaces</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Sculptured Surface Machining (SSM)</td>
<td>8</td>
</tr>
<tr>
<td>2.2.1 Sturz milling method</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2 CM²</td>
<td>9</td>
</tr>
<tr>
<td>2.2.3 Principle axis method (PAM)</td>
<td>10</td>
</tr>
<tr>
<td>2.2.4 Modified principle axis method</td>
<td>11</td>
</tr>
<tr>
<td>2.2.5 Machining strip evaluation</td>
<td>12</td>
</tr>
<tr>
<td>2.2.6 Multi-point positioning strategy</td>
<td>12</td>
</tr>
<tr>
<td>2.3 Gouging Avoidance</td>
<td>13</td>
</tr>
<tr>
<td>2.3.1 Local gouging</td>
<td>14</td>
</tr>
<tr>
<td>2.3.2 Global gouging</td>
<td>15</td>
</tr>
<tr>
<td>2.4 Tool path planning</td>
<td>16</td>
</tr>
<tr>
<td>2.4.1 Region-by-region tool path</td>
<td>16</td>
</tr>
<tr>
<td>2.4.2 Global-local tool path</td>
<td>16</td>
</tr>
<tr>
<td>CHAPTER 3: BACKGROUND</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Mathematical Basis of Parametric Curves</td>
<td>19</td>
</tr>
</tbody>
</table>
5.3 Determination of the Effective Boundary Points.............................................. 54
5.4 Smooth the Polylines ........................................................................................ 58
5.5 GUI Design ....................................................................................................... 58

CHAPTER 6: RESULTS.........................................................................................61
6.1 Tool Path Generation for AP-CM² ................................................................... 61
6.2 The Computation of the Boundary of Rework Regions ................................. 62
   6.2.1 Dialog box of settings of rework region computation .............................. 63
   6.2.2 Computation of rework boundary........................................................... 64
   6.2.3 Tool path generation of global-local CM² ............................................. 65
6.3 The Comparison of Different Global Machining Strategies ......................... 66
6.4 Actual Test for Machining One Quarter Scale of GT40 Model ..................... 68

CHAPTER 7: CONCLUSIONS ..............................................................................71
7.1 Future Work.................................................................................................... 72

REFERENCES .....................................................................................................73
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure 1. 1 Ball-end mill surface machining</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 1. 2 Tool axis position in 5-axis machining</td>
<td>3</td>
</tr>
<tr>
<td>Figure 1. 3 Ball-end milling versus CM²</td>
<td>4</td>
</tr>
<tr>
<td>Figure 2. 1 Sturz milling</td>
<td>9</td>
</tr>
<tr>
<td>Figure 2. 2 Multi-point machining</td>
<td>13</td>
</tr>
<tr>
<td>Figure 2. 3 Gouging of 5-axis flat-end milling</td>
<td>15</td>
</tr>
<tr>
<td>Figure 2. 4 Global and global-local tool path</td>
<td>17</td>
</tr>
<tr>
<td>Figure 3. 1 Bezier curves with different degrees</td>
<td>21</td>
</tr>
<tr>
<td>Figure 3. 2 Subdivision of Bezier curve</td>
<td>22</td>
</tr>
<tr>
<td>Figure 3. 3 Mapping of $s$-$t$ domain</td>
<td>28</td>
</tr>
<tr>
<td>Figure 3. 4 The mapping of a planar point to a 3D point</td>
<td>31</td>
</tr>
<tr>
<td>Figure 3. 5 The relationship between a curvature and the principle curvature</td>
<td>32</td>
</tr>
<tr>
<td>Figure 3. 6 The logical view of the CAA V5 component</td>
<td>35</td>
</tr>
<tr>
<td>Figure 3. 7 The physical view of the CAA V5 component</td>
<td>35</td>
</tr>
<tr>
<td>Figure 4. 1 Between contours guide strategy</td>
<td>38</td>
</tr>
<tr>
<td>Figure 4. 2 Parallel contour guide strategy</td>
<td>39</td>
</tr>
<tr>
<td>Figure 4. 3 Spine contour guide strategy</td>
<td>39</td>
</tr>
<tr>
<td>Figure 4. 4 Multi-axis sweeping strategy</td>
<td>40</td>
</tr>
<tr>
<td>Figure 4. 5 Different machining strips for the same effective edge of the tool bottom</td>
<td>41</td>
</tr>
<tr>
<td>Figure 4. 6 The desired machining strip</td>
<td>42</td>
</tr>
<tr>
<td>Figure 4. 7 Uncut regions</td>
<td>44</td>
</tr>
<tr>
<td>Figure 4. 8 Analysis of curvature distribution</td>
<td>46</td>
</tr>
<tr>
<td>Figure 5. 1 Dump tool path</td>
<td>48</td>
</tr>
<tr>
<td>Figure 5. 2 The uncut regions in the side of the machining surfaces</td>
<td>50</td>
</tr>
<tr>
<td>Figure 5. 3 The tool motion direction of a pair of contact points on the boundary</td>
<td>51</td>
</tr>
<tr>
<td>Figure 5. 4 The tool motion direction of a sub trajectory</td>
<td>51</td>
</tr>
<tr>
<td>Figure 5. 5 Uncut regions</td>
<td>53</td>
</tr>
<tr>
<td>Figure 5. 6 The offset boundary of the uncut regions</td>
<td>54</td>
</tr>
<tr>
<td>Figure 5. 7 Schematic of data screening</td>
<td>55</td>
</tr>
<tr>
<td>Figure 5. 8 Major boundary curves</td>
<td>56</td>
</tr>
</tbody>
</table>
Figure 5. 9 Schematic of data screening ........................................................................... 57

Figure 6. 1 Test parts ........................................................................................................ 61
Figure 6. 2 Activate the workbench of surface machining .............................................. 62
Figure 6. 3 The customized tool bar .............................................................................. 63
Figure 6. 4 The dialog box of Settings of Rework Region .............................................. 63
Figure 6. 5 The selection of tool path ........................................................................... 64
Figure 6. 6 The boundary of rework regions ................................................................. 65
Figure 6. 7 Tool path of global CM2 machining with the limit of lead angle ............... 65
Figure 6. 8 Tool path of machining the rework regions ............................................... 66
Figure 6. 9 Tool path of traditional global machining ................................................... 67
Figure 6. 10 Main tool path of global-local CM2 ......................................................... 68
Figure 6. 11 Rework tool path of global-local CM2 ..................................................... 68
Figure 6. 12 Actual machining of the top portion of GT40 ............................................. 69
**LIST OF TABLES**

Table 6.1 Comparison of different global machining ...................................................... 67
CHAPTER 1: INTRODUCTION

Currently, sculptured surfaces are widely used in the aerospace, automobile and die industry. Many commercial tools such as NX, CATIA, Pro/Engineer, etc. provide the platform for the design of sculptured surfaces. Various 3-axis and 5-axis machining methods exist today for the finish machining of sculptured surfaces. Below is a brief review of some of these methods to provide a foundation and frame the problem that will be addressed in this thesis.

The technology of 3-axis machining is the most widely used tool path method for sculptured surface machining (SSM). In general, large parts made by sculptured surfaces are machined by 3-axis machines. Ball-end mills are the cutters that are most often used in 3-axis SSM. The ball-end milling process leaves scallops on machined surfaces as shown in Figure 1.1 (a). Scallops are the main factor which influences machining quality. Finish machining generally requires that the scallop height does not exceed 0.05 millimeters. With a given diameter of ball-end mill, the distance between two tool paths is computed in terms of the value of scallop height. For example, in the case of planar ball-end milling (see Figure 1.1 (b)), the value of stepover can be calculated according to the formula below

\[ s = 2\sqrt{Dh - h^2} \]  

(1.1)
where

s: The value of stepover,
D: The diameter of ball mill, and
h: Scallop height.

Therefore, using a ball-end mill with given diameter dimension, produces much higher density of tool paths in convex regions than in concave regions when making the finish cut. This means that the overall productivity and material removal rate of ball-end mill finish surface machining is very low.

(a) Ball-end milling

(b) Step, scallop and diameter for ball-end milling

Figure 1. 1 Ball-end mill surface machining
For 5-axis machining, the strategy of tool axis orientation is chosen according to the part types and tool type. There are many types of cutters such as plain milling cutters, formed milling cutters, face milling cutters, and end milling cutters. However, end milling is often the only choice for SSM. Compared to ball-end milling in 3-axis SSM, 5-axis ball-end milling can improve the machining quality with the ability that the lead and tilt angle of tool axis can be fixed in 5-axis machining to make sure the non-cutting tool tip is not in contact with the surface. The density of tool paths is determined by the parameter of the scallop height. Therefore, the density of tool path for 5-axis ball-end mill finish machining is still high and the machining efficiency is low.

![Diagram of tool axis position in 5-axis machining](image.png)

- **N**: Normal of the machining surface
- **A**: The tool axis direction
- **F**: Feed direction
- **α**: Lead angle of the tool
- **β**: Tilt angle of the tool
- **ω**: Screw angle of the tool

**Figure 1.2 Tool axis position in 5-axis machining**

In 5-axis machining, the tool axis can be controlled by giving lead and tilt or screw angle of the tool axis (shown in Figure 1.2). For Sturz milling method, lead and tilt or screw
angle of the tool will be set to a constant value. For other tool axis strategies, these two angles are variable during the period of machining. The algorithm of curvature matched machining (CM²), first introduced by Dr. Jensen in 1992, serves to cater the curvatures of machining surface by changing the lead angle of the tool (Jensen, C.G., et al, 1992). Figure 1.3 shows the difference between the fixed axis machining and CM². The density of tool path generated by CM² is much smaller, than the one generated by ball-end milling. In addition, the method of CM² can reach the goals of improving surface finish and achieving design surface dimensions more accurately.

![Ball end milling vs Curvature matched machining](image)

Figure 1.3 Ball-end milling versus CM²

1.1 Problem Statement

For the production quality finish machining of sculptured surfaces, it is possible that some tool positions calculated using the algorithm of CM² interfere with the design surface. The algorithm used to generate NC tool paths must have the ability to avoid the interference between the tool and design surface. In addition, the tool position and orientation generated using the algorithm of CM² must be free from collisions between
the spindle and machine tool fixtures. These two types of behaviors are respectively referred to as local and global gouging avoidance.

Gouging avoidance makes CM\(^2\) impossible in some regions. When a large part is machined using CM\(^2\) algorithm, gouging avoidance in some areas causing the lead angle to be extreme and the overall density of tool paths to increase greatly, thus causing the advantages of CM\(^2\) to be significantly weakened. If the upper lead angle limit of the tool axis is restricted, it is anticipated that some material will not be removed and will require reworking these areas with smaller tools or other SSM methods. In a word, an effective solution for Global CM\(^2\) is not currently available with today’s CAD/CAM packages.

1.2 Thesis Objective

The thesis objective is to determine a global CM\(^2\) machining method using improved local and global gouge avoidance, and by restricting the upper lead angle limit. This global CM\(^2\) method will have the added benefit of higher machining efficiencies over what is currently offer by the CAD/CAM vendors as their version CM\(^2\) machining. An automated rework algorithm will also be developed that will more efficiently machine regions that were avoided by the global CM\(^2\). These rework areas will be machined using the most efficient tool path strategy. The main goal of this research is to find a method and algorithm for computing the rework boundary and its offset applying optimal CM\(^2\) methods to these regions.
2.1 The Application of Sculptured Surfaces

Currently, CAD tools have been widely used in the designs of automobiles. The body shapes of automobiles are presented by free-form surface. The most common free-form shape description is parametric surfaces (Farin, G., 1988). Polynomial parametric surfaces are called sculptured surfaces. The terms “sculptured surface”, “curved surfaces”, “free-form surfaces” and sometimes simply “surfaces” are used interchangeably (Choi, B. K., 1991) (Miller, J. R., 1986).

Sculptured surfaces are used in geometric modeling to describe variable shaped surfaces, such as car bodies, airplane wings, turbine blades and so on that cannot be described by simple curved surfaces such as cylinders and cones. Sculptured surfaces are difficult to be machined (Altan, T., et al., 1993). The popular surface representations include Bezier surfaces, Coons surfaces and B-spline surfaces. The theory of these representation methods have been wildly applied to current existing sculptured surface CAD systems (Farin, G., 1988).
2.2 Sculptured Surface Machining (SSM)

In general, parts designed with sculptured surfaces are machined by roughing and 3-axis or 5-axis finishing. A smaller ball-end mill is commonly used in finish machining to obtain a finer finish part. The general method of 3D finishing of a free-form surface is to use a ball mill to trace along the part surface by maintaining an acceptable tolerance (Chang, T. C., Wysk, R. A., and Wang, H. P., 1991). The leading commercial CAD/CAM systems have well developed ball-end machining capacities. Ball-end machining is wildly used in sculptured surface machining.

As the use of sculptured surface modeling grows, engineers will continue seeking better methods for sculptured surface machining. Some tool and die makers have found that, by changing from 3-axis to 5-axis milling, efficiency gains of 10 to 20 times could be achieved (Kim, B.H., et al, 1994) (Vickers, G.W., et al, 1989). If one can control an end mill to fit sculptured surface optimally, it would be possible to accomplish higher efficiency and better surface quality in finish sculptured surfaces (Jensen, C.G., et al, 1992) (Marciniak, K., 1991). Flat-end mills are much more efficient at removing material than ball-end mill of the same size (Loverton, T., et al, 1993). In recent years, many algorithms of tool path generation have been developed based on 5-axis flat-end milling.

In this thesis, flat-end milling was also used for the global vehicle machining. The literature review mainly focuses on the research of 5-axis flat-end milling. The machining strategy in this research was determined based on comparing the following flat-end milling methods.
2.2.1 Sturz milling method

Most commercial systems use Sturz milling method as one of their flat-end 5-axis machining strategies. The tool is inclined at a fixed angle from the surface normal as the cutter moves along the surface of the part (shown in Figure 2.1). The problem for this method is that the local gouging will occur when the cutter is not inclined far enough from the surface and unintended cutting occurs on the lateral and tailing edges of the cutter (Marciniak, K. 1987) (see Figure 2.1). Although some systems give the function of the avoidance of gouging, the material in the complex regions will not be removed. To the author’s knowledge, the solution solving this problem is not available in any of the CAD/CAM systems. Therefore, industry seldom uses this method for machining complex sculptured surfaces.

2.2.2 CM²

C.G. Jensen proposed a method for calculating an optimal lead angle of the tool axis based on the local surface curvature. CM² matches the curvature of the swept profile
of the cutting tool to the normal curvature of the surface in the plane perpendicular to feed direction. The swept profile of the cutter with a zero lead angle is a line or an ellipse with a non-zero lead angle. The swept profile of the cutter with a non-zero lead angle is an ellipse that has a major radius equal to the tool radius. The equation for the ellipse in y-z plane is

\[
\frac{z^2}{r^2} + \frac{y^2}{(r \sin \alpha)^2} = 1
\]

(2.1)

where

- \( r \): The radius of the tool
- \( \alpha \): The lead angle of the tool

Therefore, the minor radius can vary up to the tool radius. The curvature \((c)\) at the cutter contact point is equal to

\[
c = \frac{\sin \alpha}{r}
\]

(2.2)

From the equation above, lead angle of a tool can be determined by the normal curvature of the surface in the plane perpendicular to feed direction.

\[
\alpha = \arcsin(c \cdot r)
\]

(2.3)


### 2.2.3 Principle axis method (PAM)

Instead of inclining the tool in the tool motion direction, the tool is inclined in the direction of minimum curvature on the surface. When the tool contact point moves along
a curve of the smaller principle curvature, the largest machined strip width is obtained
(Marciniak, K. 1987). An optimal lead angle of the tool axis is calculated based on the
minimum curvature \( (k_{\text{min}}) \) of the local surface at the cutter contact point.

\[
\alpha = \arcsin(k_{\text{min}} \cdot r)
\]

(2. 4)

One could consider always machining with the feed along the principal direction
with minimum curvature. However, the principal direction often varies irregularly.
Usually the machining direction is matched to the principal direction as well as possible
(1996) presented similar technique. They used their technique to machine various patches
and investigated the effect of tool path direction on the technique. The main problems
with this method include potential gouging, and the tool paths generated may not be
practical or efficient because the feed direction must always follow the direction of
minimum curvature (Gray, P., et al, 2001)

2.2.4 Modified principle axis method

The direction of minimum curvature is not necessarily the best or most practical
direction of feed as it can vary wildly over complicated surfaces because the curvature
typically changes over the surface. Two problems may arise from following the direction
of minimum curvature: excessive jerk of the machine’s axis and unintuitive, inefficient
feed direction is selected based on the part geometry. Typical choices for machining
passes include isoparametric and offset Cartesian planes. The tilt angle is determined by
the curvature that is calculated in a plane perpendicular to the feed direction. Gouging
can be avoided by adjusting the tilt angle. This modified method addresses the problem of gouging and the restriction on the feed direction of the principle axis method.

2.2.5 Machining strip evaluation

By calculating an optimal tool lead angle, the minimum curvature of the effective edge of the tool bottom is matched to the local surface curvature at the cutter contact point in the plane perpendicular to the feed direction. If the tool is too large for the local radius of curvature, the tool is rotated around the surface normal at the contact point to avoid gouging. A quadratic equation can be used to approximate the surface around the cutter contact point in the principal curvature coordinate system (Marciniak, K., 1991). Lee, Y.S. (1998a, 1998b) and Lee, Y.S., Ji, H. (1997a, 1997b) use this approximation to determine if the area is convex, concave, hyperbolic or parabolic. Such quadratic surface is offset by a distance equal to the scallop height. The width of machining strip is determined by intersecting the surface offset by the scallop height with the effective edge. For actual machining, such approximation will lead to local gouging in some steep concave regions. Anyway, the principle of machining strip evaluation is helpful for determining the diameter of the flat-end mill used in this research.

2.2.6 Multi-point positioning strategy

Multi-point machining (MPM) method was proposed by Warkentin, A., et al (1996, 2000). When concave regions are machined, orienting the tool to maximize the number of contact points between the cutter and the surface as it traverses the part greatly improves the machining efficiency. The method of single contact point machining produces the deviation profiles with “V” shape. MPM produces the deviation profiles
with “W” shape (see Figure 2.2). MPM is out of the ordinary and promising. The problem with this method is that it is difficult and complex to implement, also, many of the parameters involved remain to be studied (Gray, P., et al, 2001).

There are several strategies of 5-axis sculptured surface machining discussed above, but it seems that only Sturz method, CM$^2$ and modified principle axis method may be used in practice. Obviously, Sturz method is not a good solution for machining a quilt of complex sculptured surfaces. Modified principle axis method is the same as CM$^2$ in essence. Currently, both tool path planning and gouging avoidance of CM$^2$ have been developed and is available in such commercial packages as CATIA. However, in applying this commercial CM$^2$ algorithm to the entire surfaces of a Ford GT model many global problems arose. In this thesis, CM$^2$ is to be selected as the strategy for developing a 5-axis global vehicle machining application, which avoids gouging and produces more efficient tool paths.

2.3 Gouging Avoidance

Avoidance of interference between the tool or spindle and design surfaces or fixtures is an important issue for the generation of tool path. The algorithm of tool path
creation should insure that the tool does not intersect the design surface. This is referred to as local gouging avoidance. In addition, there is no collision at any tool position and orientation between spindle and design surfaces or fixtures. This is referred to as global gouging avoidance. For both local and global gouging avoidance, there are two aspects that have to be considered. One is gouging detection. The other is gouging correct.

Although both local gouging and global gouging are used to detect machining interference, they are entirely different. Whether using advanced 5-axis tool placement or simple 3-axis machining method, the approach to detecting global interference is the same (Petrizze, D., 1997). Nowadays, most commercial systems have the ability to avoid global gouging. A few of them provide the solutions to local gouging avoidance in the tool path generation of advanced 5-axis machining such as CM².

2.3.1 Local gouging

Gouging happens when material is removed in such a way as to violate the tolerance bounds of design surface. There are various types of gouging that can occur in machining. Many different cutter types, surface profiles, and other machining conditions may cause gouging. The literature just discusses the situation of 5-axis flat-end milling.

In 5-axis flat-end milling, cutter interference can occur because of complex cutter orientation. Although the flat-end milling has the advantage of higher efficiency of removing material than traditional ball-end mill machining, there are times when the trailing edge and sides of the flat-end mill undercut the design surface (Figure 2.3). These types of machining errors require material addition and reworking of the machined surface and generally present a more difficult finishing problem than does leaving excess
material or scallops (Jensen, et al., 2000). To avoid gouging, the lead angle of the tool has to correct.

![Figure 2. 3 Gouging of 5-axis flat-end milling](image)

**2.3.2 Global gouging**

Global gouging occurs when there are collisions between the spindle and design surface or the fixtures used in machining operation. Actually many commercial systems provide the function of NC tool path verification for NC programmers. Based on supplied NC tool path, some verifiers are capable of displaying and animating the entire machining process including all machine tool components being used (CAMAX, 1996). With the help of these verifiers, NC programmer can find global gouging in time and also eliminate it through adjusting the tool path planning.
2.4 Tool path planning

Many researchers work on finding the relationship between the part surface geometry and the tool path machining efficiency. Wang, H., et al. (1987) supposed that the optimal tool paths are normally parallel to the longest boundary. Marciniak, K. (1987, 1991) and Kruth, J. P., et al (1994) concluded that, when the tool path matches the principal curvature direction of the machining surface, the largest width between two adjacent tool paths will be obtained. Sarma, R. (1996) pointed out that the adjacent tool paths are not parallel to each other for sculptured surface machining.

2.4.1 Region-by-region tool path

In the methods above, tool paths are optimized locally. Therefore, the strategy of region-by-region tool path is commonly used. The tool paths are generated in every machining region independently and each region is machined separately. The machining of entire surfaces is completed when all the regions are machined (Ding, S., et al., 2003). High efficiency machining will be achieved in each region using this strategy. But every region should be consisted of by simple surfaces or a single patch.

2.4.2 Global-local tool path

Ding, S., et al. (2003) proposed the strategy of global-local tool path for 3-axis sculptured surface machining with ball-end mill. The entire surfaces are first machined and then, the machining of local regions required higher density of tool path for cutting will be completed (see Figure 2.4 (b)). Compared to global-local path method, global path machining (see Figure 2.4 (a)) has redundant tool path outside of local regions in Figure 2.4 (b). Therefore, the strategy of global-local path planning is desired for
improving machining efficiency. As to date, this tool path planning method is just limited in the field of 3-axis machining and the boundaries of local regions are calculated based on single patch sculptured surface. It is still a method of local tool planning in a sense.

Considering gouging avoidance and reducing the redundant tool path, global optimization of tool path distribution is still far from reality. For ball-end mill global machining, to avoid local gouging that may occur in sculptured surface machining, one approach is to use a cutter that has a smaller radius than the smallest radius of curvature on the sculptured surfaces. Obviously the density of tool paths is very high. The machining efficiency will be decreased greatly. This approach is not recommendable. Another approach is to use a large cutter and skip the region of local and global gouging. The uncut material in these regions is removed by planning the tool path with a smaller cutter. The detection and correction of the local interference is a difficult problem (Chen, T., et al, 2002).

Actually flat-end mill 5-axis global machining faces the same situation. To date, no literature is found that addresses how to optimize the tool paths for global flat-end mill
machining of a big part with complex sculptured surfaces such as the top or side portion of a car. With the help of the algorithm of CM², this thesis will work on seeking the approaches for optimizing the CM² tool path distribution for the global finish machining of the parts with complex sculptured surfaces. A new method of global-local CM² will be proposed. In detail, the main global CM² tool path should be generated first to globally machine parts with a small upper limit of lead angle and to skip the regions where there are local or global gouging, and the steep concave areas where the curvature of effective edge of the bottom of the flat mill at the contact points are too small to match the curvature of the part surface. Then the local tool path should be created to machine these uncut regions. In this research, the finish machining of the global top portion of a quarter scale GT model will be conducted.
The background chapter is intended to provide the necessary mathematics and programming principles and foundation to allow the reader to understand the proposed method, its implementation and the results. The reader is encouraged to read the sections below that are unfamiliar.

3.1 Mathematical Basis of Parametric Curves

In general, a curve can be define by a mapping \( P(t) \) of the interval \([a, b]\) into Cartesian space. If \( t \) is interpreted as time, \( P(t) \) represents the movement of point \( P \) and the curve is the trajectory of the point. In orthogonal Cartesian coordinates, a curve be expressed parametrically as the equation below

\[
P(t) = (x(t), y(t), z(t)) , \quad t \in [a, b]
\]  

If set \( u = a + (b - a)t \), it is possible to normalize \( u \in [a, b] \) to \( t \in [0,1] \). For example, when three basis functions \( x(t) = \cos t, y(t) = \sin t, z(t) = 0 \), the equation below

\[
P(t) = (x(t), y(t), z(t)) = (\cos t, \sin t, 0) , \quad t \in [0,1]
\]  

represents the circular arc.
3.1.1 Curve segments

In general, curve segments are expressed as the following form in the Cartesian coordinate system

\[ P(t) = \sum_{i=0}^{n} P_i B_n^i(t), \quad t \in [0,1] \]  

(3.3)

\( P_i, i = 1, 2, \ldots, n, \) is interpreted as the coordinates of point \( P_i. \) \( B_n^i(t) \) is the basis function corresponding to point \( P_i. \)

When \( B_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i}, \) \( B_n^i(t) \) is called Bernstein basis function.

\[ P(t) = \sum_{i=0}^{n} P_i B_n^i(t) = P_i \binom{n}{i} t^i (1-t)^{n-i}, \quad t \in [0,1] \]  

(3.4)

The curve expressed by the equation (3.4) is called Bezier segment.

3.1.2 Bezier curve

For Bezier curve

\[ B_n^i(t) = \binom{n}{i} t^i (1-t)^{n-i} \]  

(3.5)

It can be proved that \( P(t) = \sum_{i=0}^{n} B_n^i(t) = 1 \)

Bezier curve has the following properties:

- Coordinate system independence
  This means that Bezier curve will not change if the coordinate system is changed.
- Convex hull property
Bezier curve always lie within the convex hull of the control point

- **Symmetry**
  Bezier curve does not change if the control points are ordered in reverse sequence.

- **Variation diminishing property**
  If given straight line intersects the curve in $x$ number of points and the control polygon in $y$ number of points, it will always hold that $x = y - 2p$, $p$ is zero or positive integer.

- **Endpoint interpolation**
  At the both end of the control points, $B^i_n(t)$ meets the following conditions:
  
  \[
  B^0_n(0) = 1, \quad B^i_n(0) = 0, \quad i = 1, \ldots, n \\
  B^n_n(1) = 1, \quad B^i_n(1) = 0, \quad i = 1, \ldots, n - 1
  \]

\[\text{Figure 3. 1 Bezier curves with different degrees}\]
For \( n = 1 \)  \[ P(t) = (1-t)P_0 + tP_1 \]

For \( n = 2 \)  \[ P(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \]

For \( n = 3 \)  \[ P(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2 (1-t)P_2 + t^3 P_3 \]

For \( n = 4 \)  \[ P(t) = (1-t)^4 P_0 + 4t(1-t)^3 P_1 + 6t^2 (1-t)^2 P_2 + 4t^3 (1-t)P_3 + t^4 P_4 \]

Figure 3.1 shows three Bezier curve examples.

### 3.1.3 Increasing the degree of Bezier curve

Bezier segment \( P_0, P_1, \ldots, P_n \) of degree \( n \) can be presented as a segment \( Q_0, Q_1, \ldots, Q_{n+1} \) of degree \( n+1 \).

\[
Q_i = a_i P_{i-1} + (1-a_i) P_i \quad a_i = i/(n+1), \quad i = 0, \ldots, n+1, \quad (3.6)
\]

### 3.1.4 Subdivision of Bezier curve

The subdivision algorithm is the most fundamental algorithm for Bezier curve. A Bezier curve is defined over the parameter interval \([0, 1]\). A Bezier curve can be

Figure 3.2 Subdivision of Bezier curve
subdivided into two new Bezier curve. One of them is defined over \([0, r]\). The other is defined over \([r, 1]\). These two curves are equivalent to the original Bezier curve. The control points of the new Bezier curve can be calculated in terms of the formula below:

\[
P_i^j = (1 - r)P_i^{j-1} + rP_{i+1}^{j-1}; \quad j = 1, \ldots, n; \quad i = 0, \ldots, n-j
\]  

(3.7)

Figure 3.2 shows the examples of Bezier curve subdivision.

### 3.1.5 Multi-segment curves

The degree of Bezier curve should be as small as possible. For \(n = 1\), Bezier curve is a line. For \(n = 2\), three control points determine that Bezier curve is a planar curve. To get a three-dimensional Bezier curve, the degree should be equal to or larger than three. There are two different ways to define curves with a complicated shape. One is to increase the degree of the curve. The other is to join a sequence of smaller-degree curve segment together smoothly. Obviously, it is hard for the first one to control the local shape. A piecewise collection of Bezier curves, connected end to end, is called a spline curve.

### 3.1.6 B-spline curves

A B-spline curve is a special spline curve defined by a sequence of degree \(n\) Bezier curves that are joined with \(C^{n-1}\) continuity. The use of B-Spline curves in geometric design was first proposed by Gordon, W.J., Riesenfeld, R.F. (1974) and Riesenfeld (1973).

The parameter interval of the individual Bezier curves that make up a B-spline can be specified by a knot vector that is a list of parameter values shown below.

\([u_1, u_2, u_3, \ldots]\)
The knot vector is the way to state the definition of the basis. In particular, it manages the continuity between the different segments of the basis functions. The knot vector is a set of non-decreasing parameter values.

The control points of each Bezier curve segment can be labeled using polar values. For degree \( n \) Bezier curves over the parameter interval \([t_1, t_2]\), the control points can be labeled below.

\[
(p_1, p_2, \ldots, p_n)
\]

When \( j \leq n-i \), \( p_j = t_1 \)

When \( j > n-i \), \( p_j = t_2 \)

For a cubic B-spline, the control points of the Bezier segment over parameter interval \([t_1, t_2]\) can be labeled using polar value below

\[
(t_1, t_1, t_1), (t_1, t_1, t_2), (t_1, t_2, t_2), (t_2, t_2, t_2)
\]

The knot vector for this Bezier segment is expressed as

\[
[t_1, t_1, t_1, t_2, t_2, t_2]
\]

Using affine and symmetric properties of polar values, it is very easy to insert a knot in the knot vector of a B-spline and decompose a B-spline into Bezier curves.

The equation of a degree \( n \) B-Spline curve is

\[
P(t) = \sum_{i=0}^{n} B_{k}^{i}(t)P_i \quad \text{for} \quad a \leq t \leq b
\]  

(3.8)

where \( P_i, i=0, ..., n \) are the control points. The degree \( k \) B-Spline basis functions \( B_{k}^{i}(t) \) in interval \( t_i \leq t < t_{i+1} \) are given by the recursive equation

\[
B_{k}^{i}(t) = \begin{cases} 
1 & t_i \leq t < t_{i+1} \\
0 & otherwise 
\end{cases}
\]  

(3.9)
\[ B_i^k(t) = \frac{t - t_i}{t_{i+k} - t_i} B_{k-1}^i(t) + \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{k-1}^{i+1}(t) \] (3.10)

With the following convention

\[ \frac{t - t_i}{t_{i+k} - t_i} B_{k-1}^i(t) = 0 \quad \text{if} \ t_{i+k} = t_i \]

\[ \frac{t_{i+k+1} - t}{t_{i+k+1} - t_{i+1}} B_{k-1}^{i+1}(t) = 0 \quad \text{if} \ t_{i+k+1} = t_{i+1} \]

The relation between the number of knots \((m+1)\), the degree \((k)\) of \(B_i^k(t)\) and the number of control points \((n+1)\) is given as follows

\[ m = (n+1) + k \] (3.11)

Knot values are non-decreasing, so a knot vector can have knots with the same value. In this case, the knot is called multiple, and its multiplicity is the number of repetitions of the same value. Multiple knots diminish the continuity between adjacent Bezier segments. For a B-spline of degree \(k\), the continuity across a knot of multiplicity \(p\) is commonly \(k-p\). There are as many segments as knots of different values plus one. If the increment is always 1, the knot vector is called uniform. The most common B-spline curves are those of degree three.

### 3.1.7 NURBS curves

A B-Spline curve whose knot vector is not evenly spaced is called a non-uniform B-spline. A NURBS (Non Uniform Rational B-Spline) curve can be viewed as the projection of a non-rational B-Spline curve defined in four-dimensional homogeneous coordinate space back into a three-dimensional physical space (Rogers, D. F., 2001). NURBS curves have been widely used in geometric design.
The NURBS defines a curve as a piecewise rational polynomial function of a parameter $t$. A NURBS curve is defined by control points $P_i$, $i=0, \ldots, n$, whose influence is weighted by rational polynomial functions $R^i_k(t)$, $i=0, \ldots, n$, (dependent on the parameter) and weights $w_i$, $i=0, \ldots, n$, (independent on the parameter). The rational polynomial functions $R^i_k(t)$ are defined by the B-Spline basis, set of piecewise polynomial functions $B^i_k(t)$, $i=0, \ldots, n$, of same degree $k$. The degree of the NURBS curve is the degree of the polynomial functions.

$$P(t) = \frac{\sum_{i=0}^{n} w_i B^i_k(t) P_i}{\sum_{j=0}^{n} w_j B^j_k(t)} = \sum_{i=0}^{n} \left( \frac{w_i B^i_k(t)}{\sum_{j=0}^{n} w_j B^j_k(t)} \right) P_i = \sum_{i=0}^{n} R^i_k(t) P_i \quad \text{(3.12)}$$

The definition of the basis $B^i_k(t)$ is uniquely determined by a knot vector, containing the parameters of the limits of pieces of the basis polynomial functions. They represent an interval for the parameter values to calculate a segment of shape. The first and last knots correspond to the first and last control point.

In general, the control points are not the points of the NURBS curve. By convention, the first and last control points are the beginning and end point of the curve respectively, except for the periodic NURBS curves. These control points can be seen as an attracting region for the curve, which influence is weighted.

NURBS curve has the following properties:

- The NURBS provides a unified mathematical model for representing
  - analytic shapes (such as conics, that cannot be handled by the Bezier model, by uniform B-Splines or non uniform B-Splines)
• free form entities, used to design car bodies for example.

• Their model easily manages the continuity between the arcs, and their algorithm is fast and numerically stable.

• They are invariant under common geometric transformations such as translations and rotations.

For a more detailed information of the parametric curves above, see (Sederberg, 2003), (Farin, G., 1999) and (Dassault Systemes, 2004a).

3.2 Mathematical Basis of Parametric Surfaces

In general, the mapping of $s$-$t$ domain on the plane into space is the representation of a sculptured surface. Surface points can be expressed parametrically by mapping $S(s, t)$ where $S$ is a point in space and $(s, t)$ is a point in $s$-$t$ domain on the plane (Figure 3.3). The parametric equation of $S(s, t)$ can be written as

$$S(s, t) = (x(s, t), y(s, t), z(s, t)) \quad (3.13)$$

The parameter $s, t$ can be used as curvilinear coordinates of a point on surface. A sculptured surface may be just a surface patch if it is defined on a rectangular bounded domain. If the domain of a surface is trimmed by one or many curves, the surface is then called a trimmed surface. An assembly of patches or trimmed surfaces with prescribed inter-patch continuity condition is called a “piecewise surface” (Pi, J., 1996). A complex sculptured surface part is always an assembly of trimmed surfaces. The continuity between two adjacent patches or trimmed surface may be the most commonly $C^0$, $C^1$ or $C^2$ in the direction of $s$ or $t$. 
3.2.1 Ruled surfaces

A ruled surface is a surface that is created by moving a line in space. To define such a surface, a base curve $B(s)$ and a director curve $D(t)$ can be used. The surface consists of points $S(s, t)$. The equation of $S(s, t)$ is

$$ S(s, t) = B(s) + tD(s) \quad (3.14) $$

A ruled surface can also be generated by sweeping a straight line along two curves of $B(s)$ and $P(s)$. In such a case, the equation of $S(s, t)$ is

$$ S(s, t) = tB(s) + (1-t)P(s) \quad (3.15) $$

3.2.2 Bezier patches

The equation of a Bezier patch is

$$ S(s, t) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_n^i(s)B_m^j(t)P_{i,j} \quad (3.16) $$

where the basis functions of $B_n^i(s)$ and $B_m^j(t)$ are the Bernstein polynomials.
\[ B_n^i(t) = \binom{n}{i} s^i (1-s)^{n-i} \quad B_m^j(t) = \binom{m}{j} t^j (1-t)^{m-j} \]

\[ P_{i,j} \] is the \( i,j \)th control point. There are \( n+1 \) and \( m+1 \) control points in the \( i \) and \( j \) directions respectively. Bezier patches have the following properties:

- In general, the patches just pass through the control points of the corners of the control point grid.
- The patches are contained in the convex hull of the control points.

### 3.2.3 B-spline patches

A B-spline surface that is \( n \)th degree for \( s \) and \( m \)th degree for \( t \) is defined as

\[ S(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_k^i(s)B_p^j(t)P_{i,j} \quad s \in [0,1], \ t \in [0,1] \quad (3.17) \]

Where the blending functions of \( B_k^i(s) \) and \( B_p^j(t) \) should be calculated using the formulas of (3.9) and (3.10).

### 3.2.4 NURBS patches

The parametric definition of the patch is

\[ S(s,t) = \frac{\sum_{i=0}^{n} \sum_{j=0}^{m} w_{i,j} B_k^i(s)B_p^j(t)P_{i,j}}{\sum_{j=0}^{n} \sum_{j=0}^{m} w_{i,j} B_k^i(s)B_p^j(t)} \quad (3.18) \]

Similar to the NURBS curves, the NURBS patches deal with the continuity using the multiplicity of the knots, but now, two knot vectors are needed, one for each direction of the surface.
For additional information of the parametric surfaces above, see (Sederberg, 2003), (Farin, G., 1999) and (Marciniak, K., 1991).

### 3.3 Differential Properties of Sculptured Surfaces

To compute the cutter location for both ball-end milling and 5-axis flat-end milling, the differential forms for sculptured surfaces machined are required. The normal vectors, tangent planes and curvatures can be obtained based on the differential properties of the sculptured surfaces.

#### 3.3.1 Differential properties

The vectors tangent to the surface at \( S(s, t) \) in the direction of \( s \) and \( t \) are

\[
R_s = \frac{\partial S(s, t)}{\partial s}
\]

and

\[
R_t = \frac{\partial S(s, t)}{\partial t}
\]

Then the normal vector of a surface (see Figure 3.4) can be calculated by

\[
N(s, t) = \left\langle \frac{\partial S(s, t)}{\partial s} \otimes \frac{\partial S(s, t)}{\partial t} \right\rangle = \frac{R_s \otimes R_t}{[R_s \otimes R_t]}
\]

Using \( N(s, t) \), the cutter location for ball-end milling can be calculated.

The first fundamental form

\[
|G| = \left| \frac{\partial S(s, t)}{\partial s} \otimes \frac{\partial S(s, t)}{\partial t} \right|^2 = |R_s \otimes R_t|^2
\]
The second fundamental form

\[
|B| = \left| \begin{array}{ccc}
\frac{\partial^2 S(s,t)}{\partial s^2} \bullet N(s,t) & \frac{\partial^2 S(s,t)}{\partial s \partial t} \bullet N(s,t) \\
\frac{\partial^2 S(s,t)}{\partial t \partial s} \bullet N(s,t) & \frac{\partial^2 S(s,t)}{\partial t^2} \bullet N(s,t)
\end{array} \right|
\]  

(3.21)

3.3.2 Principle curvatures

Gaussian curvature

\[
k_g = \frac{|B|}{|G|}
\]  

(3.22)

Mean curvature

\[
k_m = \left\{ \left[ \frac{\partial S(s,t)}{\partial s} \right]^2 \left( \frac{\partial^2 S(s,t)}{\partial s^2} \bullet N(s,t) \right) + \left( \frac{\partial S(s,t)}{\partial s} \right)^2 \left( \frac{\partial^2 S(s,t)}{\partial s^2} \bullet N(s,t) \right) \right\} - \left[ \frac{\partial S(s,t)}{\partial t} \right] \left( \frac{\partial^2 S(s,t)}{\partial t \partial s} \bullet N(s,t) \right) + \left[ \frac{\partial S(s,t)}{\partial t} \right] \left( \frac{\partial^2 S(s,t)}{\partial s \partial t} \bullet N(s,t) \right) \right\} \frac{1}{2|G|}
\]  

(3.23)
Principle curvatures

\[ k_1 = k_m - \sqrt{k_m^2 - k_g} \quad k_2 = k_m + \sqrt{k_m^2 - k_g} \]  

(3.24)

Then, \( k_g \) and \( k_m \) can be expressed as

\[ k_g = k_1 \cdot k_2 \]

and

\[ k_m = (k_1 + k_2)/2 \]

For \( k_1 \) and \( k_2 \), there is the principle below at a point of \( S(s, t) \)

\[ k_1 \cdot k_2 = 0 \text{ or } k_1 \perp k_2 \]

\( F \) is a vector in the plane tangent to \( S(s, t) \) at the point shown in Figure 3.5. The curvature in the plane perpendicular to the vector \( F \) can be expressed as

\[ c = k_1 \cos^2 \theta + k_2 \sin^2 \theta \]  

(3.25)

\[ k_1 \text{ and } k_2 \text{ are the principle curvature of the part surface at the contact point} \]

\[ \text{Figure 3.5 The relationship between a curvature and the principle curvature} \]
3.4 Introduction of Component-Based Application

The computation of rework regions for CM² in this thesis is developed using C++ based on Component-Based Application. The research in this thesis benefits from the nature of object-oriented programming (OOP) and the application based on components. The introduction of both will assist readers to quickly understand the programming structure in this research. There are many books that discuss OOP in depth. To know more about Component Application Architecture (CAA), readers may refer to Dassault Systemes (2004a).

3.4.1 Components and interfaces

Many CAA services are offered using CAA components hidden behind a curtain of interfaces. Said another way, Components are the cores that create an application. A component is a piece of code that can be executed and may finish some tasks. Components can be used via the interfaces exposed by them. They cannot be modified. In such cases, the implementation details of components are hidden, but a component can be replaced by another one exposing the same interfaces and executing the same job. These properties of CAA components ensure upward compatibility to their client applications.

With the help of OOP technology, objects are made of data and functions, and components are made of objects. Objects provide the features required by components:

- Objects can expose their interfaces and hide their implementations.
- Objects have polymorphism and can be exchanged at run time with other objects that match the same interface.
• New versions of objects with new capabilities can be provided while keeping client applications running without the need of rebuilding them.

Based on the concept of CAA, Dassault Systemes released their next generation application development platform named CAA V5. From the point of a logical view of the CAA V5 component (see Figure 3.6), a CAA V5 component can be seen as an object exposing interfaces. On the other hand, the component supplier has a detailed physical view of the CAA V5 component (see Figure 3.7). Said another way, the programmer who creates this component knows which C++ classes are used in it, how these C++ classes are taken together to create the component, and which interface is implemented by which class. In a word, a CAA V5 component is an object that can be reusable, that is provided in a binary form, and that can be instantiated and used by client applications. Therefore, via the interfaces exposed by components, the components may be applied to represent a real object, such as a mathematical surface or a solid feature, or a software object, such as a dialog box or an end user command, or finally a set of services, such as a model checker or a stress analyzer. A CAA V5 component may be created using a class or several classes that are assembled to provide the component type and the component behaviors.

An interface of a CAA V5 component represents either the type or the behavior, or a part of the behavior, of the implementing component. The interface includes all the operations to be performed by the implementing component. An interface pointer to a given component can be applied to request this component to perform some of these operations. Also another pointer to another interface that it implements may be asked to return. Hence, a pointer can be “skipped” from one to another according to the requirement of the component. An interface can also be seen as a chain that links the
Figure 3.6 The logical view of the CAA V5 component

Figure 3.7 The physical view of the CAA V5 component
interface designer, the component developer, and the client application programmer. This chain is made of the interface name and specifications, such as the interface task, the operations and their purposes, and the compliance of the component to these specifications. This should not change with the time except the fact that the specifications are allowed.

With the help of interfaces, components may be used using a standard protocol and changed without influence on the client applications, and their implementation details may be hidden. In addition, with the assistance of this standard protocol, the actual component location is hidden and the client application is independent of the creation and the accessibility of the component.

CATIA V5 is developed based on CAA V5 by Dassault Systemes. Therefore, the programs that are developed in this research will be written in CAA V5 so that they can be easily integrated with CATIA V5. Another nice feature about developing in CAA is the notion of reuse and upward compatibility on newer versions of CATIA V5 without the need of rebuilding the dll.
CHAPTER 4: METHOD

As discussed previously the machining of large surface regions with complex shapes are divided in many different machining regions according to their different shape characteristics. The programming of these regions is generally finished region by region. Sometimes local optimization of these region by region tool paths is feasible which improves the overall efficiency. The drawback of this method is that the programming is not automated and is very time-consuming. Although computing the tool paths for this traditional region by region machining is less time consuming, in order to finish the machining of the complex shape, the density of tool path for global machining is very high, making the overall machining time much longer than it needs to be. Programmers are in a dilemma wanting to use global machining strategy, but without the long programming and machining times. The research of this thesis focuses on how to optimize the tool path for global machining. The following presents the principle of the method of global-local CM².

Internal to the CATIA system are algorithm very similar to the CM² algorithms developed by Dr. Jensen. It is the implementation of these algorithms within the CATIA system that cause the tool path density to be much higher in steep regions than it needs to be. The method presented here will develop, implement and test new algorithms that apply a global optimized CM² machining strategy within the CATIA environment.
4.1 Tool Path Planning

CATIA currently supplies two strategies or approaches for planning 5-axis global tool paths, multi-axis contour driven strategy and multi-axis sweeping strategy.

4.1.1 Multi-axis contour driven strategy

Within the multi-axis contour driven strategy, CATIA provides three options to choose from; between contours guide, parallel contours guide, and spine contour guide.

- *Between contours* guide strategy

![Figure 4.1 Between contours guide strategy](image)

The tool paths are defined by the top and bottom guide curves while respecting user-defined geometry limitations and machining strategy parameters (Dassault Systemes, 2004b) (see Figure 4.1). Both guide curves should be smooth. With this method, the machining quality is very good at the boundary between machining region and its’ adjacent areas. The boundary of a complex patchwork of sculptured surfaces is often irregular. This strategy is not feasible for global CM² vehicle machining.
• *Parallel contours* guide strategy

![Figure 4. 2 Parallel contour guide strategy](image)

For this strategy, the tool sweeps out an area by following contours parallel to a reference contour that is called guide curve (Dassault Systemes, 2004 (2)) (see Figure 4.2). The machining region should be fan-shaped except that the guide curve is a line. In general, this strategy will be used when guide curve is the same as or parallel to one boundary of the machining region.

• *Spine contour*

![Figure 4. 3 Spine contour guide strategy](image)
For this strategy, the tool sweeps across a contour in perpendicular planes (Dassault Systemes, 2004b) (see Figure 4.3). Using this strategy, the tool path will be overlapped or very unbalanced for global machining a big part.

4.1.2 Multi-axis sweeping strategy

For this strategy, the tool paths are parallel to the plane that is defined by the machining direction ($F$) and the view direction ($V$) (Dassault Systemes, 2004b) (see Figure 4.4).

Figure 4.4 Multi-axis sweeping strategy

Comparing to multi-axis contour driven strategy, multi-axis sweeping strategy is more feasible for globally machining to top portion of a vehicle.

4.2 Limits of Improving Machining Efficiency

Figure 4.5 shows that the same effective edge has a different machining strip width when it is used for machining different areas. Figure 4.5 (c) is the case of CM². So it has the largest strip width, that is, it has the best machining performance. The
Machining strip width is essentially determined by the diameter of the tool, scallop height, and the lead angle or local curvature of part surface at the cutter contact point. In convex region, it is also influenced by the geometry shape. Both Figure 4.5 (a) and (b) have smaller machining strip width.

![Diagram of machining strips](chart)

The direction of the tool motion is perpendicular to the paper

**Figure 4.5 Different machining strips for the same effective edge of the tool bottom**

For CM² application, Figure 4.6 (a) and (b) are the best result for machining the regions that are planar or convex in the direction of the stepover of tool path. Even so, the strip width is very small when the machining region is steep convex in the direction of stepover. For the machining region that is planar in the direction of stepover, the larger the diameter of tool, the bigger is the strip width. CM² used in concave regions can achieve the largest machining strip. When there is a gouging avoidance in CM², the situation of Figure 4.5 (a) and (b) will happen and the strip width in Figure 4.5 (c) will also be decreased.

From the points mentioned above, the factors that restrict the improvement of CM² efficiency are:

- lead angle or gouging avoidance in the planar region or in the convex area with steep shape in the direction perpendicular to the tool motion
• steep convex shape in the stepover direction

• gouging avoidance in concave region

![Diagram of machining](image)

The direction of the tool motion is perpendicular to the paper

Figure 4. 6 The desired machining strip

4.3 Determining the Tool’s Diameter

In order to find how much the diameter of the appropriate tool is, the following two factors have to be considered.

a) The curvature distribution of part surfaces in the concave region for global-local CM$^2$

\[ r \leq \sin \alpha_M / k_{max} \]  \hspace{1cm} (4.1)

where

\[ k_{max} \] : the maximum curvature in the concave machining area

\[ \alpha_M \] : the upper limit of lead angle
b) The geometry shape in the stepover direction if the shape is convex

\[ r > \frac{w_{\text{convex}}}{2} \]  \hspace{1cm} (4.2)

where

\( w_{\text{convex}} \): Machining strip in the convex regions

In general, increasing the diameter of the tool causes the increase of machining strip width. When the diameter of the mill is larger than a specific value, increasing the diameter of the tool is invalid for expanding the machining strip width for machining the convex region. In addition, more uncut material may be left in the concave region with bigger curvature when the part is machined by a tool with too large diameter. For actual machining, the tool’s diameter is often determined by referring to formula (4.1).

When \( r > \sin \alpha_{sl} / k_{\text{max}} \), there will be uncut material left, and the larger \( r \) is, the more uncut material is left. The determination of \( r \) is based on the balance of cut and uncut regions. Adaptive balance can achieve the higher machining efficiency.

### 4.4 Elimination of the Limits

The geometry shape in the stepover direction should be as flat as possible if the shape is convex. In practice, it is difficult to judge what the best direction is. Most often, the direction with the longest segment of tool path is selected by programmers.

For global-local CM\(^2\), the lead angle should be set to be small enough so that the curvature in the planes perpendicular to the tool motion becomes the limits of lowering the density of tool path. This may result that there is uncut material left in the region where a bigger lead angle should be given for gouging avoidance or curvature matching. If the method of determining the rework regions is obtained, the strategy of improving
machining efficiency by limiting lead angle is feasible. Said another way, the global-local CM$^2$ is first implemented by the main global CM$^2$ with smaller lead angle, and then by machining the uncut regions using the adaptive methods, such as ball-end milling or CM$^2$.

4.5 Determination of Rework Regions

By comparing the upper limit of lead angle ($\alpha_M$) to the one ($\alpha$) with which the effective edge of the tool bottom matches the curvature of the part surfaces at the cutter contact point, the uncut regions can be determined. All the regions where $\alpha_M$ is smaller than $\alpha$ are uncut (see Figure 4.7).

\[ \alpha_M = \arcsin(k_M \cdot r) \quad (4.3) \]

$k_M$ is the maximum curvature which effective edge of the tool bottom can match for the actual machining because of the limit of the lead angle. It is determined by

\[ k_M = \frac{C}{\alpha} \]

where $C$ is the effective edge of the tool bottom and $\alpha$ is the lead angle.
analyzing the curvature distribution in the region that is concave in the direction of stepover, and in the planes that are perpendicular to the direction of stepover. $\alpha$ can be calculated using equation (2.3). Here $c$ is all the possible curvature of part surfaces in the planes perpendicular to the tool motion at contact points. $c$ can be calculated using equation (3.25). After substitute $c$ into equation (2.3), the lead angle can be calculated if gouging avoidance is not considered

$$\alpha = \arcsin\left(\left(k_1 \cos^2 \theta + k_2 \sin^2 \theta\right) \cdot r\right) \tag{4.4}$$

With the help of CAD packages, principle curvatures, $k_1$ and $k_2$, may be easily obtained. $c$ is calculated using equation (3.25). Therefore, $k_M$ can be determined by referring the value of $c$ in the machining region. The value of $k_M$ should just be larger than $c$ in the regions which are machined by the main global CM$^2$, or less than $c$ in the small uncut regions. Said another way, $c$ is larger than $k_M$ in uncut regions. Therefore, the larger $k_M$, the smaller are uncut regions. Figure 4.8 shows that, using CATIA V5, $k_1$ can be directly obtained from the analysis result of -0.0341 and $k_2$ is equal to -1/58.291.

Now that the value of $k_M$ is available, the radius of the flat-end mill used for finishing the main global CM$^2$ may be determined with the equation below.

$$r = \frac{\sin \alpha_M}{k_M} \tag{4.5}$$

Equation (4.5) also shows that the larger $k_M$ is, the smaller the tool radius. Therefore, the smaller the tool radius, the smaller and fewer the resulting uncut regions, however, Lee, Y.S. (1998a, 1998b) and formula (4.2) suggest, the smaller the tool radius, the higher the density of global CM$^2$ tool path. In addition, if the tool radius is too large, the number of uncut regions will be large, thus causing extensive and time consuming tool path
planning. The best tool radius should be determined using optimization. The determination of the ideal tool radius is recommended as a future research topic. In this thesis, the selection of tool radius is based on the number of uncut regions, where the number does not exceed 2.

![Analysis of curvature distribution](image)

Figure 4.8 Analysis of curvature distribution

The tool path of CM^2 created by CATIA V5 does not contain the parameters of \( \alpha \) and \( c \), but contains the normal vector \( (N) \) of part surfaces and the tool axis vector \( (A) \) at each contact point (see Figure 4.7). Therefore, \( \alpha \) can be calculated in terms of the equation below

\[
\alpha = \arccos\left(\frac{A \cdot N}{|A||N|}\right)
\]

(4.6)
The method described in chapter four was fully implemented for the
determination of rework regions for the global-local CM2. To compute the boundary of
these rework regions, tool path of assumed previous CM2 (AP-CM2) should be first
generated using the method of CM2 in CATIA V5 R13. The lead angle limit for this tool
path should be larger than the one for the main tool path of CM2 of the global-local CM2
machining. It is necessary to dump the information for the computation of rework regions
from AP-CM2 tool path. This chapter includes

• the design of adaptive data structure
• the method of data screening
• the strategies of the determination of the effective boundary points
• the generation of smooth boundary curves
• the design of GUI(Graphic User Interfaces)

5.1 Data Structure and Tool Path Dumping

In this thesis, the boundary points are extracted from tool path of AP-CM2. They
should sit on or be as close as possible to the machining surfaces. The contact points and
other data pertaining to contact points, such as the normal of machining surfaces and the
vector of tool axis at each contact point, are all necessary. To easily locate the individual
contact point an index should also be included. The *struct* type of *MyTotalToolPath* is defined in the program as the following.

```c
struct MyTotalToolPath
{
    double LocalAxis[3];
    double ContactPoint[3];
    double ContactNormal[3];
    int ToolPathNb;
    int Start;
    int End;
};
```

The variable of *PathPtIndex* is declared as the type of *MyTotalToolPath* for tool path dumping.

**Figure 5.1 Dump tool path**
ToolPathNb can be used for determining which trajectory a contact point belongs to. Start and End are used to determining the number of the first and the last contact point of a trajectory. These items are very important for the computation of the rework region.

The selection agent of _ActAcq is declared as a path element agent of CATPathElementAgent. This is a data member of class CAAMaiDumpToolPathCommand. By activating the icon of computing the rework region and selecting the tool path for dumping, CAAMaiDumpToolPathCommand.dll can be performed. The value of _ActAcq is finally transferred to the external function of CAAMaiDumpMultipleMotion in which the selected tool path is dumped to MyTotalToolPath (see Figure 5.1).

5.2 Data Screening

Each point (P_i or P_o) (see Figure 4.7) sitting on the boundary of uncut regions meets the condition below

- The lead angle at it is less than $\alpha_M$.
- The lead angle at the contact point next to $P_i$ is larger than $\alpha_M$ (see Figure 4.7).
- The lead angle at the contact point previous to $P_o$ is larger than $\alpha_M$ (see Figure 4.7).

If the uncut region exists in the side of the machining surfaces (see Figure 5.2), in the case of Figure 5.2 (a), $P_o$ will be on the boundary of the machining surfaces, and in the case of Figure 5.2 (b), $P_i$ may be on the boundary of the machining surfaces.

Substituting LocalAxis and ContactNormal into equation (4.6), the lead angle can be calculated using the equation below.
\[
\alpha = \arccos \left( \frac{\text{LocalAxis} \bullet \text{ContactNormal}}{\|\text{LocalAxis}\| \|\text{ContactNormal}\|} \right)
\] (5.1)

In addition, both \(P_i\) and \(P_o\) are expressed by the coordinate of \(\text{BoundaryPoint}_X\), \(\text{BoundaryPoint}_Y\) and \(\text{BoundaryPoint}_Z\). In order to compute multiple uncut regions, \(P_i\) or \(P_o\) should be given index numbers (\(\text{BoundaryPoint}\)) and a tool path index (\(\text{ToolPathIndex}\)). In general, the distribution of points of \(P_i\) and \(P_o\) is irregular. The boundary of rework regions cannot be generated by simply connecting the adjacent points of \(P_i\) or \(P_o\). A rework region should be surrounded by smooth curves. All the points of \(P_i\) and \(P_o\) are on the boundary or outside of the actual uncut region.

### 5.2.1 Determination of the tool motion direction

To help optimize the boundary of rework regions in the future, each contact point on the boundary will be attached with a direction variable of \(\text{MotionDirection}\). The value of this variable may be “Forward” or “Backward.” It is determined by the intersection angle (\(\varphi_b\)) between the direction of the cross product of the view direction (\(V\)) and the
reference direction \(R\), and the one of the cross product of the view direction and vector \(M_b\) (see Figure 5.3). In general, vector \(R\) is the same as the global machining direction. Both \(R\) and \(V\) are set by programmers. \(V\) can be obtained using the class of ViewDirection and \(R\) can be obtained via GUI. \(M_b\) is the vector from point \(P_i\) to point \(P_o\) (see Figure 5.3).

![Figure 5.3](image)

**Figure 5.3 The tool motion direction of a pair of contact points on the boundary**

![Figure 5.4](image)

**Figure 5.4 The tool motion direction of a sub trajectory**

\[
\varphi_b = \cos^{-1}\left(\frac{(\mathbf{V} \otimes \mathbf{R}) \cdot (\mathbf{V} \otimes \mathbf{M}_b)}{||\mathbf{V} \otimes \mathbf{R}|| |\mathbf{V} \times \mathbf{M}_b|} \right)
\]

(5.2)

\(\varphi_b\) can be calculated in terms of equation (5.2). If \(\varphi_b\) is less than 90 degrees, the tool motion direction at the point of \(P_i\) and \(P_o\) is “Forward.” Otherwise, it is “Backward.”

In addition, each sub trajectory is also assigned a direction variable of PathDirect. The value of this variable is determined by the intersection angle (\(\varphi\)) between the direction of the cross product of the view direction (\(V\)) and the reference direction (\(R\),
and the one of the cross product of the view direction and vector M (see Figure 5.4). M is the vector from the point of “Start point” to the point of “End point” of a sub trajectory (see Figure 5.4). $\varphi$ can be calculated in terms of equation (5.3). If $\varphi$ is less than 90 degrees, the tool motion direction of a sub trajectory is “Forward.” Otherwise, it is “Backward.”

$$\varphi = \cos^{-1}\left( \frac{(V \otimes R) \cdot (V \otimes M)}{|V \otimes R| |V \otimes M|} \right)$$

(5.3)

5.2.2 Pairing point $P_i$ and $P_o$

$P_i$ and $P_o$ always show up as pairs (see Figure 4.7). It is related to the direction of tool motion that a contact point on the boundary of the uncut regions is which of $P_i$ and $P_o$. For an ideal case, there should always be a point $P_i$ and $P_o$ on a sub trajectory that goes across an uncut region. If the tool motion direction at a contact point on the boundary of the uncut region points into this region, this contact point is $P_i$. If the tool motion direction at a contact point on the boundary of the uncut region points out of this region, this contact point is $P_o$.

In the actual uncut region of a sub trajectory, there is often more than one pair of $P_i$ and $P_o$. Figure 5.5 shows that there are a pair of $P_i^1$ and $P_o^1$, and a pair of $P_i^2$ and $P_o^2$. Program is capable of detecting if they belong to two different rework regions. Obviously, there are abundant points of $P_i$ and $P_o$ in Figure 5.5. They should be deleted.

A pair of $P_i$ and $P_o$ has not only the same tool motion direction, but also the same tool path index. All the pairs of $P_i$ and $P_o$ where the tool motion direction is different from the one of sub trajectory, said another way, where both $\varphi_b$ and $\varphi$ are not less or larger
than 90 degrees at the same time, will be considered as irregular points. These points will be deleted and not be used to create the boundary curves of the rework regions.

The ideal pairs of $P_i$ and $P_o$ should have the same tool motion direction as the sub trajectory which they are sitting on. Figure 5.7 shows the schematic of pairing point $P_i$ and $P_o$.

5.2.3 Offset of the boundary of uncut regions

An adaptive outward offset of the boundary of rework regions ensures that there will be no uncut material left after the machining of rework regions. The points on the offset of the boundary should be as close as possible to the machining surface. They may be found on the polylines that are created by the contact points. The length of the polyline from the original paired points ($P_i$ and $P_o$) on the boundary to the point $P_b$ on the offset boundary is equal to the value of offset (see Figure 5.6 and equation (5.4)). $P_b$ can be calculated using equation (5.5) and (5.6).

$$
|P_iP_{i-1}| + \ldots + |P_{i-k}P_b| = |P_oP_{o+1}| + \ldots + |P_{o+k}P_b| = \text{offset}
$$

(5.4)
In the case of Figure 5.6 (a),

\[
P_b = P_{i-k} + (P_{i-k-1} - P_{i-k}) \frac{\text{offset}}{||P_{i-k-1} - P_{i-k}||} \left( |PP_{i-1}| + ... + |P_{i-k+1} - P_{i-k}| \right)
\] (5.5)

In the case of Figure 5.6 (b),

\[
P_b = P_{o+k} + (P_{o+k+1} - P_{o+k}) \frac{\text{offset}}{||P_{o+k+1} - P_{o+k}||} \left( |PP_{o+1}| + ... + |P_{o+k-1} - P_{o+k}| \right)
\] (5.6)

![Figure 5.6 The offset boundary of the uncut regions](image)

Figure 5.7 show the schematic of the data screening. Then how to connect these paired points of \(P_i\) and \(P_o\) into the smooth boundary curves of rework regions should be figured out.

### 5.3 Determination of the Effective Boundary Points

Although the paired points of \(P_i\) and \(P_o\) sit around the rework region, the boundary curve created by connecting them directly is wavy. Figure 5.8 shows that just connecting the points of \(P_o\) is a good solution. A rework region is encircled by two major curves. If the points of \(P_i\) at both ends of two major curves are included, the boundary curves with such a connection will be better. If the end point of one major curve is \(P_i\), the end point of the other curve should be \(P_o\), and vice versa (see Figure 5.8).
Figure 5.7 Schematic of data screening
Another issue is, beginning from one end point, how to find the subsequent points of $P_o$. How can the broken polylines of major boundary curves shown in Figure 5.8 be created automatically? How can the major boundary curves be generated in the case of multiple rework regions?

![Figure 5.8 Major boundary curves](image)

$P_{i,1}^l$ is the first point of the first polylines. $P_{o,2}^l$ is the first point of the second polylines. If the tool motion at the paired points of $P_{i,1}^l$ and $P_{o,2}^l$ is “Forward” (or “Backward”), the subsequent points of the first polylines except the last point should be points of $P_o$ with the tool motion direction of “Backward” (or “Forward”) and the following points of the second polylines except the last point should be points of $P_o$ with the tool motion direction of “Forward” (or “Backward”). If the tool motion at the last paired points of $P_{i,1}^n$ and $P_{o,2}^n$ for the first rework region is “Forward” (or “Backward”), $P_{i,1}^n$ (or $P_{o,2}^n$) is the last point of the first polylines and $P_{o,2}^n$ (or $P_{i,1}^n$) is the last point of the second polylines. Figure 5.9 shows the schematic for determining the effective points that consist of the polylines.
Push all paired points of \( P_i \) and \( P_o \) in the container

\[ j=1, k=1, \text{get } P_{i,1}^j = P_i^j \]
\[ CtrlPt_1[1] = P_i^j \]

\[ j < \text{Total paired number?} \]

Yes

\[ j=j+1, \text{get } P_i^j \text{ and } P_o^j \]

No

\[ P_i^j \rightarrow \text{MotionDirection} \]
\[ = !P_o^k \rightarrow \text{MotionDirection?} \]

Yes

No

\[ \left| P_i^{j-1}P_o^j \right| < \text{RegionDist} \]
\[ \text{or } \left| P_o^{j-1}P_o^j \right| < \text{RegionDist?} \]

Yes

No

\[ CtrlPt_1_{\text{tmp}} = P_i^j, \quad CtrlPt_1[k] = P_o^j \]
\[ P_{o,1}^j = P_o^j, \quad J_1 = P_i^j \rightarrow \text{ToolPathIndex} \]
\[ CtrlPt_1_K = k \]

No

\[ j > 1? \]

Yes

\[ k=k+1 \]

No

\[ j=1, k=1, \text{get } P_{o,2}^j = P_o^j \]
\[ CtrlPt_2[1] = P_o^j \]

\[ j < \text{Total paired number?} \]

Yes

\[ j=j+1, \text{get } P_i^j \text{ and } P_o^j \]

No

\[ P_i^j \rightarrow \text{MotionDirection} \]
\[ = P_o^{j-2} \rightarrow \text{MotionDirection?} \]

Yes

No

\[ \left| P_i^jP_o^{j-1} \right| < \text{RegionDist} \]
\[ \text{or } \left| P_o^{j-2}P_o^j \right| < \text{RegionDist?} \]

Yes

No

\[ CtrlPt_2_{\text{tmp}} = P_i^j, CtrlPt_2[k] = P_o^j \]
\[ P_{o,2}^j = P_i^j, J_2 = P_o^j \rightarrow \text{ToolPathIndex} \]
\[ CtrlPt_2_K = k \]

No

\[ j > 1? \]

Yes

\[ k=k+1 \]

No

\[ J_1 > J_2? \]

Yes

No

\[ CtrlPt_2[CtrlPt_1_K+1] = CtrlPt_2_{\text{tmp}} \]
\[ CtrlPt_2[CtrlPt_2_K+1] = CtrlPt_1_{\text{tmp}} \]

Create the first major polylines with the points of \( CtrlPt_1 \)
Create the second major polylines with the points of \( CtrlPt_2 \)

End

Loop

No

Is the container empty?

Yes

End

Figure 5.9 Schematic of data screening
5.4 Smooth the Polylines

Instead of polylines, B-spline major curves can be created. These B-splines interpolate the points of \( \text{CtrlPt}_1 \) and \( \text{CtrlPt}_2 \). The continuity of these curves may be defined as \( C^2 \). The curves are B-splines of the degree 3. In terms of equation (3.11), the number of knots \((m+1)\) can be calculated as the following.

\[
m+1 = n + 5
\]

\( n+1 \) is the total number of control points. For the first major curve, \( n+1 \) is equal to \( \text{CtrlPt}_1\_K \). For the second major curve, \( n+1 \) is equal to \( \text{CtrlPt}_2\_K \). The knot vector may be expressed as

\[
[0,0,0,t_1,t_2,\ldots,t_i,\ldots, t_{n-2},t_{n-2},t_{n-2},t_{n-2}]
\]

\( t_i \) can be calculated according to the length of polylines from a control point to the start point of \((0,0,0)\). \( w_i \) is set to be 1. Then using equation (3.9) and (3.10), base function \( B^i(t) \) can be calculated. Substituting the points of \( \text{CtrlPt}_1 \) and \( \text{CtrlPt}_2 \) into equation (3.12), a matrix equation may be built. By solving this equation, the control points are available. Therefore, the major curves can be approximated by B-spline curves.

5.5 GUI Design

In order to launch the program developed in this research, an add-in belonging to the workbench of “Surface Machining” should first be created. This add-in includes the tool bar that consists of two commands. One of commands is used for parameter setting. The other is used for boundary computation of the rework regions. The add-in is declared in the class of \( \text{CAAESmiSurfaceMachiningAddin} \) under the module of
CAASmiUserOperationCommand. The command of boundary computation links the module of CAAMaiDumpToolPathCommand. The command of parameter setting links the module of ReworkBdOffsetUI.
CHAPTER 6: RESULTS

The purpose of this thesis is to implement the boundary computation of rework regions for global-local CM². The experimental results of global-local CM² and the computation of rework regions are presented in this chapter.

6.1 Tool Path Generation for AP-CM²

In order to compute the rework regions, the tool path of AP-CM² with large lead angle should be created first. In this example, there are two parts that are used for the test of global-local CM². They are opened in CATIA (see Figure 6.1). For part A, the tool path of AP-CM² was generated with the lead angle of 45 degree, the scallop height of
0.1mm and the machining tolerance of 0.01mm. The direction of the tool path is show in Figure 6.1(a). For part B, the tool path planning strategy the tool path of AP-CM² was created with the lead angle of 8 degree, the scallop height of 0.2mm and the machining tolerance of 0.3mm. The direction of the tool path is show in Figure 6.1(b).

6.2 The Computation of the Boundary of Rework Regions

Launch CATIA and open the CATProcess file. Then the workbench of Surface Machining should be activated (see Figure 6.2). To actuate the application of rework region computation, a tool bar used for rework region computation and parameter setting is added in the workbench of Surface Machining. This customized tool bar interacts between end users and CATIA system.
The customized tool bar includes the following two icons (see Figure 6.3):

- Parameter setting icon: allows end users to set the parameters used for computing the boundary of rework regions.
- Computation icon: allows end users to compute the boundary of the rework regions.

6.2.1 Dialog box of settings of rework region computation

When the icon of parameter settings is activated, the dialog box of Settings of Rework Region will be opened (see Figure 6.4). In the box of Lead angle of previous CM², the lead angle for actual machining, $\alpha_m$, should be entered. The value of offset should be entered in the box of Offset of rework boundary and the machining direction...
vector for the global-local CM² should be typed in the box of *Machining Direction*. The parameters set above will be used in the computation of rework boundary.

### 6.2.2 Computation of rework boundary

To compute the boundary of rework regions, the icon should be activated, the item of *tool path* with “computed” should be selected (see Figure 6.5). Then, based on the tool path generated in the section of 6.1 and the parameters set above, the boundary of rework regions will be computed. Figure 6.6 (a) is the rework boundary for the global-local CM² of the part in Figure 6.1 (a). Figure 6.6 (b) is the rework boundary for the global-local CM² of the part in Figure 6.1 (b).

![Figure 6.5 The selection of tool path](image-url)
6.2.3 Tool path generation of global-local CM²

Now that the boundary of rework region has been created, the tool path of global CM² machining with the limit of lead angle may be generated with the lead angle of $\alpha_{sL}$ (see Figure 6.7). For part A, the rework region can be machined using CM² with a
smaller flat mill (see Figure 6.8 (a)). For part B, the rework region can be machined with a ball mill (see Figure 6.8 (b)). For both parts, the machining direction is the same as the one of AP-CM².

(a) Tool path of CM² for machining the rework region for part A  
(b) Tool path of ball-end milling for machining the rework region for part B

Figure 6. 8 Tool path of machining the rework regions

6.3 The Comparison of Different Global Machining Strategies

In order to compare global-local CM² to other traditional global machining strategies, the tool path of global ball-end milling and traditional global flat-end mill CM² of part B is generated using CATIA V5 (see Figure 6.9). All of the specifications used in these global machinings are listed in Table 6.1. Also this table clearly shows that global-local CM² has the best global machining performance.
(a) Tool path of 3-axis global ball-end milling for machining part B

(b) Tool path of traditional global flat-end CM² for machining part B

Figure 6.9 Tool path of traditional global machining

Table 6.1 Comparison of different global machining

<table>
<thead>
<tr>
<th>Tool (Diameter): (mm)</th>
<th>Global Ball mill: 25.4</th>
<th>Flat mill: 25.4</th>
<th>Flat mill: 25.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rework Ball mill: 6.35 Ball mill: 6.35</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machining tolerance: (mm)</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>Scallop height: (mm)</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>Feed: (mm/minute)</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>Spindle speed: (RPM)</td>
<td>3000</td>
<td>3000</td>
<td>3000</td>
</tr>
<tr>
<td>Machining time: (minute)</td>
<td>Global Neglected 549 256 76</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rework Neglected 25</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total machining time: (minute)</td>
<td>&gt; 549</td>
<td>256</td>
<td>101</td>
</tr>
</tbody>
</table>
6.4 Actual Test for Machining One Quarter Scale of GT40 Model

In this research, an actual machining of the top portion of one quarter scale of GT40 model was conducted to test the performance of global-local CM². Figure 6.10 shows the main tool path of global-local CM². Figure 6.11 shows the rework tool path of global-local CM². The actual machining is shown in Figure 6.12. The total time for machining the top portion is 94 minutes.

**Machining Tolerance:** 0.03 mm  
**Scallop height:** 0.05 mm  
**Lead Angle:** 7 degrees  
**Flat mill diameter:** 6 mm  
**Feed:** 1000 mm/minute  
**Machining time:** 133 minutes

![Figure 6.10 Main tool path of global-local CM²](image)

![Figure 6.11 Rework tool path of global-local CM²](image)
Main CM² global local machining

**Machining Tolerance:** 0.03 mm
**Scallop height:** 0.05 mm
**Lead Angle range:** 7 degrees
**Flat mill diameter:** 6 mm
**Feed:** 150 inches/minute
**Machining time:** 90 minutes

Rework machining

**Machining Tolerance:** 0.03 mm
**Scallop height:** 0.05 mm
**Ball mill diameter:** 3 mm
**Feed:** 200 inches/minute
**Machining time:** 4 minutes

Figure 6. 12 Actual machining of the top portion of GT40
CHAPTER 7: CONCLUSIONS

This research has shown that the rework regions can be determined for CM^2, and that global-local CM^2 is a feasible strategy for improving the efficiency of machining a quilt of sculptured surfaces. The concept and principle for global-local CM^2 were presented. The methodology about how to determine the parameters used in computation was introduced. The strategies for determining the boundary points of rework regions and obtaining the offset of the rework boundary were developed. The tool for dumping AP-CM^2 tool path and GUI for setting the parameters for global-local CM^2 was coded using CAA V5. The comparison of different global machining was given in this thesis. Finally, a test of global-local CM^2 for cutting one quarter scale GT40 model was conducted.

The main advantages of global-local CM^2 are listed below.

- Programming time for CM^2 tool path planning has been significantly reduced when machining a quilt-work of sculptured surfaces.
- Determining the uncut regions for global CM^2 vehicle is now completely automated.
- Machining efficiency is significantly improved.

The main problems of global-local CM^2 developed in this thesis are listed below.

- The tool path of AP-CM^2 is sometimes irregular in the uncut regions. This is due to the fact that a single algorithm is attempting to address all rework
regions, regardless of the shape or location. When implemented in a 
commercial package this will need to be more robust.

- The current algorithm has difficulties dealing with the irregular rework 
  regions where the tool path of AP-CM² goes through twice or more.
- Using NURBS for the rework boundary is not possible in CAA V5 
  because CATIA has not released the NURBS APIs.

7.1 Future Work

To improve this research two things are needed. First is direct access to the 
CATIA CM² algorithms. This would allow better control over the placement of CM² tool 
paths within the uncut regions. All irregular tool paths in the reworked areas could be 
eliminated. It would also allow the new algorithm for screening and pairing \( P_i \) and \( P_o \) to 
be fully integrated within the CATIA machining module. With access to the internal CM² 
it would be possible to fix the second problem which is to improve the handling of AP-
CM² tool paths that go through the reworked region multiple times. The algorithms for 
computing this type of rework regions need additional work and develop but only after 
CATIA has released their CM² API.

As discussed in the Method chapter this research can be extended and enhanced 
by an investigation of the optimal cutter parameters. What should the ideal radius or 
corner radius be? Should the cutter have a taper angle? What should its length be? All of 
these parameters in some way affect the density (i.e., efficiency) of the calculated CM² 
tool path. It is recommended that a future study examine global finish machining using an 
optimized tool.
REFERENCES


