Discrete modeling of sculptured surface machining for robust automatic feedrate selection

Jeffrey Gordon Hemmett
University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/dissertation

Recommended Citation
https://scholars.unh.edu/dissertation/19

This Dissertation is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI®

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
NOTE TO USERS

This reproduction is the best copy available.

UMI®
DISCRETE MODELING OF SCULPTURED SURFACE MACHINING
FOR ROBUST AUTOMATIC FEEDRATE SELECTION

by

Jeffrey G. Hemmett

B.S.M.E. University of New Hampshire, 1992
M.S.M.E. University of New Hampshire, 1994

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Engineering: Systems Design

May, 2001
This dissertation has been examined and approved.

Dissertation Director, Dr. Barry K. Fussell
Associate Professor of Mechanical Engineering,

Dr. Robert B. Jerard
Professor of Mechanical Engineering,

Dr. May-Win Thein
Assistant Professor of Mechanical Engineering,

Dr. Igor I. Tsukrov
Assistant Professor of Mechanical Engineering,

Dr. Robert L. Drysdale III,
Professor of Computer Science,
Dartmouth College

5/2/2001
Date
DEDICATION

In memory of my mother,
Who taught me that the sky is the limit,

And to my Father,
Who taught me how to get there.
ACKNOWLEDGEMENTS

First and foremost I would like to acknowledge the love and support of Hester Jaarsma. She stood by me through all of the transitions in life that returning to school brought about and somehow during this time decided that she would be willing to become my wife. Without her devotion, or the yogurts and carrots she kept stocked in the refrigerator so that I would always have something to eat at the lab, you would not be reading this right now. My family also deserves a great deal of credit, and so I would like to thank my father, Monica, Ted, Beth, and Paul. I would also like to thank my 'new' family, the Jaarsmas. They were all always there with a supporting word when we spoke, a nice respite when I visited, and understood when I was too busy to contact them at all. We have all been through a lot of changes over the past several years, and it is comforting to know that reliable support is always there when you need it.

I would like to thank Dr. Barry Fussell for the opportunity he provided to work on this project, for although at times I would have liked to curse him for it, this experience has only increased my abilities and my knowledge of myself. I also owe a great deal of credit to both Barry and Dr. Robert Jerard for your ability to offer help and direction when needed, and also knowing when to say nothing. I would also like to thank the rest of my committee, Dr. May-Win Thein, Dr. Igor Tsukrov, and Dr. Scott Drysdale, for taking time out of their busy schedules to review this work and helping me in this endeavor. I also owe a great deal to the other members of the Design and Manufacturing lab, both for being silly and for being serious, generally at the appropriate times. I would also like to acknowledge the role of National Science Foundation, who funded the majority of this research.

Another source of support, quite literally, was Turbocam Inc., who provided the majority of my summer funding, and also provided an environment that inspired creative and functional solutions. I would like to thank Marian Noronha for signing my checks but not saying what I had to do to earn them. And to Mike O'Donnel and Craig Clause, thank you for your help, good humor, and Foodees pizza.

Finally, I would like to thank Don and Cindy Marsh, as our off-road jeeping adventures and the subsequent repairs could keep my mind off of this research for hours at a time. I also feel I should mention Mr. Charlie Papazan, whose simple credo was a great help in getting through these trying times.
# TABLE OF CONTENTS

DEDICATION .......................................................................................................................... iv

ACKNOWLEDGEMENTS ......................................................................................................... v

TABLE OF CONTENTS .............................................................................................................. vi

LIST OF TABLES ...................................................................................................................... xiii

LIST OF FIGURES ................................................................................................................... xv

LIST OF SYMBOLS .................................................................................................................. xxx

ABSTRACT ............................................................................................................................... xxxvii

## CHAPTER PAGE

1  INTRODUCTION .................................................................................................................. 1

1.1 Introduction ...................................................................................................................... 1

1.2 NC Milling Overview ...................................................................................................... 2

1.3 Milling Efficiency .......................................................................................................... 11

1.4 Automatic Adaptive Feedrate Estimation ................................................................... 14

1.5 Milling Force Estimation and Automatic Adaptive Feedrate Estimation ..................... 16

1.6 Integrated Modeling Approach to Automatic Adaptive Feedrate Estimation .............. 17

1.7 Summary of Advancements .......................................................................................... 20

1.8 Thesis Overview ........................................................................................................... 21

2  MECHANISTIC MODELING FOR MILLING FORCE ESTIMATION AND FEEDRATE SELECTION .................................................................................................................. 23

2.1 Chapter Introduction .................................................................................................... 23

2.2 Volumetric Based Force Models .................................................................................. 24

2.3 Instantaneous Mechanistic Modeling ........................................................................... 26

2.4 Coordinate System Definition for Discrete Mechanistic Modeling ............................. 30
2.5 Cutter Rotation Simulation ................................................................. 31
2.6 The Contact Patch .............................................................................. 32
2.7 Chip Thickness Calculations ............................................................... 33
  2.7.1 Chip Thickness ............................................................................. 33
  2.7.2 The Classical Chip Thickness Model ............................................. 33
  2.7.3 The Generalized Chip Thickness Model ....................................... 35
  2.7.4 Feed-Per-Tooth ........................................................................... 36
  2.7.5 Trochoidal Chip Thickness Models .............................................. 38
  2.7.6 Cutter Runout ............................................................................. 39
2.8 Discrete Mechanistic Modeling Methods ........................................... 42
  2.8.1 The Classical Discrete Mechanistic Model .................................... 42
  2.8.2 The Generalized Discrete Mechanistic Model ............................. 44

3 MECHANISTIC MODEL AND FEEDRATE SELECTION IMPLEMENTATION ................. 49
3.1 Chapter Introduction ........................................................................... 49
3.2 The Discrete Cutting Tool Representation .......................................... 50
  3.2.1 Discrete Cutter Description .......................................................... 50
  3.2.2 Axial Discretization and Disc Thickness Values .......................... 51
3.3 Definition of the Axial Disc Parameters ............................................. 56
  3.3.1 Axial Disc Definition ................................................................. 56
  3.3.2 Axial Thickness ......................................................................... 56
  3.3.3 Axial Location ........................................................................... 57
  3.3.4 Disc Radius ............................................................................... 58
  3.3.5 Cutter Profile Normal ............................................................... 58
  3.3.6 Cutting Flute Representation ..................................................... 59
3.4 Simulation of the Bottom of a Flat End Cutter .................................. 62
3.5 The Discrete Cutter Model and Applied Mechanistic Modeling ........... 65
3.6 Automatic Feedrate Selection via Iterative Mechanistic Modeling ......... 66
3.7 Feed Inhibitors .............................................................. 69
3.8 Cutter Rotation Simulation ............................................. 72
  3.8.1 Cutter Rotation Simulation and Automatic Feedrate Selection 72
  3.8.2 Calculation of the Cutter Rotation Simulation Array .......... 72
  3.8.3 Cutter Rotation Simulation Optimization ......................... 73

4 GEOMETRIC MODELING OF IN-PROCESS STOCK FOR
MILLING FORCE ESTIMATION ........................................... 77
  4.1 Chapter Introduction .................................................... 77
  4.2 In-Process Stock Geometric Model Requirement ................. 77
  4.3 In-Process Stock Geometric Model Selection .................... 78
    4.3.1 Solid Modeling Approach ....................................... 78
    4.3.2 Discrete Geometry Representation ................................ 80
  4.4 Definition of the In-Process Stock Geometric Model ............. 83
    4.4.1 The Z-buffer Model ............................................... 83
    4.4.2 Individual Z-buffer Elements ................................... 85
    4.4.3 Updating the Z-buffer Model and Intersected Segment Definition 87
    4.4.4 Estimation of Volumetric Material Removal .................... 91
  4.5 Geometric Model Verification ....................................... 92
  4.6 Custom Z-buffer Model Definition ................................... 96
  4.7 Z-buffer Model File Storage .......................................... 101
  4.8 Computer Memory Issues ............................................. 103

5 TOOLPATH ENVELOPE MODELING AND INTERSECTION CALCULATIONS .... 107
  5.1 Chapter Introduction .................................................. 107
  5.2 Tolpath Envelope Modeling Techniques ............................ 107
    5.2.1 Solid Modeling of the Swept Toolpath Envelope ............. 107
    5.2.2 CSG Solid Modeling of the Toolpath Envelope ............... 109
7 CNC MACHINE MODELING AND RELATIVE VELOCITY CONTROL

FOR 5-AXIS CNC MILLING

7.1 Chapter Introduction
7.2 Machine Coordinate to Workpiece Coordinate Transformations
7.3 The Relative Velocity Vector Vnet
7.4 Calculation of the Relative Velocity Vector
7.4.1 General Description
7.4.2 Linear Axis Contributions
7.4.3 Rotational Axis Contributions
7.5 Kinematic Modeling for Calculation of Relative Velocities
7.5.1 General Definition
7.5.2 Application Example: The Table-on-Table 5-axis Configuration
7.6 The Controller Model
7.6.1 Individual Axis Velocity Calculation
7.6.2 G-code Data Acquisition
7.7 Relative Velocity as a Function of Axial Location
7.8 Standalone Relative Velocity Control
7.8.1 Software implementation of Relative Velocity Control
7.8.2 Results of the Standalone Model

8 MECHANISTIC MATERIAL CONSTANT CALCULATION

8.1 Chapter Introduction
8.2 The Mechanistic Material Constants
8.3 Data Acquisition for Mechanistic Constant Calculation

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.4</td>
<td>Mechanistic Constant Calculation</td>
<td>196</td>
</tr>
<tr>
<td>8.4.1</td>
<td>General Calculation Approach</td>
<td>196</td>
</tr>
<tr>
<td>8.4.2</td>
<td>Calculation of KT, KR for a Given Set of Cutting Conditions</td>
<td>197</td>
</tr>
<tr>
<td>8.4.3</td>
<td>Calculation of KTC, KRC, P1, and P2</td>
<td>198</td>
</tr>
<tr>
<td>8.5</td>
<td>Examples of Material Constants Found To Date</td>
<td>200</td>
</tr>
<tr>
<td>8.6</td>
<td>Ball-End Cutter Geometry Simulation</td>
<td>208</td>
</tr>
<tr>
<td>9</td>
<td>SOFTWARE VALIDATION, DATA ACQUISITION AND EXPERIMENTAL RESULTS</td>
<td>211</td>
</tr>
<tr>
<td>9.1</td>
<td>Chapter Introduction</td>
<td>211</td>
</tr>
<tr>
<td>9.2</td>
<td>Supporting Software Validation</td>
<td>211</td>
</tr>
<tr>
<td>9.2.1</td>
<td>Supporting Software</td>
<td>211</td>
</tr>
<tr>
<td>9.2.2</td>
<td>Geometric Model Validation</td>
<td>212</td>
</tr>
<tr>
<td>9.2.3</td>
<td>NC Machine Model Validation</td>
<td>214</td>
</tr>
<tr>
<td>9.3</td>
<td>Force Data Acquisition Experimental Setup</td>
<td>214</td>
</tr>
<tr>
<td>9.4</td>
<td>Sampling Rate Selection</td>
<td>215</td>
</tr>
<tr>
<td>9.5</td>
<td>Mechanistic and Integrated Model Validation</td>
<td>219</td>
</tr>
<tr>
<td>9.5.1</td>
<td>Discrete Cutter Model Validation</td>
<td>219</td>
</tr>
<tr>
<td>9.5.2</td>
<td>Mechanistic Model Validation</td>
<td>220</td>
</tr>
<tr>
<td>9.5.3</td>
<td>Integrated Force Estimation Model Validation</td>
<td>232</td>
</tr>
<tr>
<td>9.6</td>
<td>Automatic Feedrate Selection Validation</td>
<td>233</td>
</tr>
<tr>
<td>10</td>
<td>CONCLUSIONS AND CONTINUING RESEARCH</td>
<td>257</td>
</tr>
<tr>
<td>10.1</td>
<td>Chapter Introduction</td>
<td>257</td>
</tr>
<tr>
<td>10.2</td>
<td>Summary</td>
<td>257</td>
</tr>
<tr>
<td>10.3</td>
<td>Suggestions for Continuing Research</td>
<td>261</td>
</tr>
<tr>
<td>10.3.1</td>
<td>Critical Enhancements</td>
<td>262</td>
</tr>
<tr>
<td>10.3.2</td>
<td>Suggested Enhancements</td>
<td>263</td>
</tr>
</tbody>
</table>
10.3.3 Research Possibilities ................................................. 265

10.4 Conclusions ..................................................................... 266

APPENDIX A  TROCHOIDAL CHIP THICKNESS CALCULATIONS .......... 268

APPENDIX B  MACHINE SPACE TO WORKPIECE SPACE COORDINATE
TRANSFORMATIONS ............................................................. 275

APPENDIX C  Z-BUFFER ELEMENT SPACING ANALYSIS .................. 282

APPENDIX D  PROTOCOL FOR MECHANISTIC CONSTANT DATA ACQUISITION .................................................. 288

APPENDIX E  SOFTWARE USERS GUIDE .................................... 292

LIST OF REFERENCES .............................................................. 299
LIST OF TABLES

Table 3.1: Axial disc parameters, the values of which are assumed constant for a given axial disc. The parameters contained at the top of the table define the discrete cutting tool, and are constant for a given cutter type. .............................. 51

Table 3.2: Disc parameters set for force estimation on the bottom of a flat end cutter. ...................... 64

Table 3.3: Cutting Conditions for the force plots of Figure 3.8. .............................................................. 74

Table 8.1: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (0.375 in. Diameter, 4 flute, Ball end cutter, HSS) ................................................. 201

Table 8.2: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (0.5 in. Diameter, 4 flute, Ball End Cutter, HSS) .................................................. 202

Table 8.3: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates 0.1875 in. Diameter, 4 flute, Ball End Cutter, Carbide) ................................. 203

Table 8.4: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (0.5 in. Diameter, 4 flute, Ball End Cutter, HSS) .................................................. 204

Table 8.5: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (0.5 in. Diameter, 4 flute, Ball End Cutter, Carbide) ................................. 205

Table 8.6: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (0.5 in. Diameter, 4 flute, Flat End Cutter, HSS) .................................................. 206

Table 8.7: Material constant comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates (1.0 in. Diameter, 4 flute, Flat End Cutter, HSS) .................................................. 206

Table 9.1: Validation of the mechanistic model for a 0.375" ball end mill cutting 6061 Al. In this research, the peak XY force magnitude is currently the value of interest when estimating milling forces. The measured XY force magnitude data is provided in Figure 9.5. ................................................................. 225

Table 9.2: Validation of the mechanistic model for a 0.1875" carbide ball end mill cutting 6061 Al. The peak XY force magnitude is shown above. ......................................................... 225
Table 9.3: Validation of the mechanistic model for a 0.5" carbide ball end mill cutting 15-5 Stainless Steel. The peak XY force magnitude is shown above. .............................................227

Table 9.4: Validation of the mechanistic model for a 1.0" HSS flat end mill cutting 6061 T6 aluminum. The peak XY force magnitude is shown above. ....................................................227

Table 9.5: A summary of the optimized feedrate constraint values and results for all operations of the commercial part program that had optimized feeds generated. ............255

Table D.1: Amplifier force limits for various Newton/Volt settings. ................................. 289

Table D.2: Sample data for obtaining calibration constants. ............................................290
LIST OF FIGURES

Figure 1.1: Some basic cut geometry definitions used to describe stock engagement during the milling process.................................................................3

Figure 1.2: During 'Down Milling', the motion of the flute due to rotation opposes the relative direction of travel, while during 'Up Milling' the flute travels in the direction of relative motion......................................................................................................................3

Figure 1.3: Various cutter profiles, and a visual depiction of the cutter helix angle \( \alpha_{helix} \)................................................................. 3

Figure 1.4: Visual Depiction of the Chip Thickness \( h \) and Feed-Per-Tooth \( f_{net} \)...........................................................................................5

Figure 1.5a: Two views of a simple 3-axis vertical CNC depiction.......................................................................................................................5

Figure 1.5b: Two views of a simple 5-axis CNC depiction...........................................................................................................................8

Figure 1.6a: 5-axis milling can produce more complex geometry than 3-axis milling. In 3-axis Milling (left), the cutting tool is constrained to a single orientation, unlike 5-axis Milling (right), where the cutter is free to rotate relative to the stock........................................8

Figure 1.6b: The use of 5-axis mills can improve milling efficiency. To produce identical surfaces to the same tolerance, the 5-axis milled part on the left requires 2/3 fewer tool passes than the 3-axis milled part on the right (count cusps)..................................................................8

Figure 1.7: Two different kinematic configurations used to achieve five-axis motion on a CNC mill. . . . 9

Figure 1.8: The principal directions of the milling force vectors calculated by the mechanistic model. . . . 18

Figure 1.9: Flowchart overview of the integrated modeling approach at automatic feedrate selection. . . . 18

Figure 2.1: In the orthogonal cutting model, radial and tangential force components are the estimated values as a result of chip formation.................................................................27

Figure 2.2: In discrete mechanistic models, cutting tools are represented as a set of discrete axial disc elements of thickness \( dZ \) ...............................................................................................................................27

Figure 2.3: The discrete cutter concept aids in modeling cutter geometry (variable helix angle, surface normal, radius, etc.) and simplifies representation of the region of intersection between the cutting tool and stock (the 'contact area'). .................................................................................................29

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.4: Definition of the Cutting Tool coordinate system. ...................................................... 29
Figure 2.5: The contact region is defined on the cutting tool via the definition of entrance and exit angles in Cutting Tool polar coordinates for each individual axial disc engaged in the stock. ................................................................. 34
Figure 2.6: This figure shows the contact area and relative chip thickness values for the partial immersion cut shown at the top of the page. ................................................................. 37
Figure 2.7: Cutter runout is a misalignment between the cutting tools central axis and the center of rotation, and leads to eccentricity in cutter rotation. ...................................................... 40
Figure 2.8: The milling force is estimated in the $X_{CT}, Y_{CT}$ plane, while the physical force vectors occur in 3 dimensions relative to the cutter flute geometry on the current axial disc. ................. 45
Figure 2.9: Note that the axial disc thickness $dZ$ used in the classical mechanistic model is not equal to the actual flute segment length $L_f$ except in cases where $\alpha_{helix} = 0$. ....................... 45
Figure 3.1: The axial discretization of flat and ball end mills for discrete cutter model definition using the cutter profile as a guideline. ................................................................. 52
Figure 3.2: Definition of the axial and radial components of the cutter ball profile normal vector. .... 60
Figure 3.3: The flute segment length $L_f$ for a given disc is a function of the helix angle, $\alpha_{helix}$, and the length of the cutter profile orthogonal to the cutter profile normal, $d_n$. ................. 60
Figure 3.4: Definition of the helix angles on the ball portion of the cutter makes use of the cutter ball flute definition angle, $\sigma$, both in transitioning from $\alpha_{helix}$ to $\alpha_{min}$ and for flute termination. ................................................................. 60
Figure 3.5: Several examples of the modeled ball end flute geometry rendered from discrete model. ... 63
Figure 3.6: The cutting flutes located on the bottom of axial disc ($i=0$) for a flat end mill are included in the force calculations when the cutter is travelling downwards relative to it's axis. ................................................................. 63
Figure 3.7: The calculated output feed is inhibited when the cutter first enters the stock (left), or when large directional changes are occur (right). ................................................................. 63
Figure 3.8: Force simulation plots showing milling force as a function of rotation angle for comparing different rotational increments. The cutting conditions are shown at top. ................. 75
Figure 4.1: In a Z-buffer model approach at NC simulation, the stock geometry is defined as a discrete set of uniformly spaced lines. ................................. 84

Figure 4.2: A traditional Z-buffer stores only the top of each element (left). In an extended Z-buffer, multiple levels can be stored, as required for 5-axis simulation (right). ................. 84

Figure 4.3: The Z-buffer model is defined to exist in the workpiece coordinate system. ....................... 84

Figure 4.4: The Zbuffer model elements are separated by a distance \( d_{xy} \) along the \( X_{wp} \) and \( Y_{wp} \) axes, and are indexed sequentially in \( X_{wp} \) from 0, and by \( N_x \) in \( Y_{wp} \). Each Z-buffer element has an associated area, \( A_{xy} \), but the element itself is a line that resides at the center of this area. .................................................. 86

Figure 4.5: Each linked list element contains a top height, and possibly a linked list of gap elements. ........................................................................................................... 86

Figure 4.6: The closed solid model STEs of this research always intersect a Z-buffer element at two locations. Only unusual, concave cutter shapes result in more intersections, shown at left. ................................................................. 86

Figure 4.7: In this figure is shown case 1, where no valid intersections occur, no Intersected Segments are created, and no updating of the Z-buffer model is required. Also shown is a reference chart that also applies to Figures 4.8 and 4.9. ......................... 90

Figure 4.8: This figure shows the two possible situations of case two, where either one (left) or no (right) valid intersections exist. In either case an intersected segment is created. If created, the intersected segment is subtracted from the model after being used in the calculation of the cut geometry parameters. ........................................ 90

Figure 4.9: This figure shows the five situations of case three (see text). Only in case 3a is no intersected segment created, and only cases 3a and 3e have no valid intersection locations. These define the specific cases for a boolean subtraction of an interior Z-buffer segment. .......................................................... 90

Figure 4.10: The Z-buffer model is easily rendered by creating a grid of nodes \( (n_i) \) that are used to define a list of connected quadrilaterals \( (Q_i) \). Normal vectors are then defined for each node as the average of the normals that are found with that
node for each quad.. ................................................................. 95

Figure 4.11: Once the quadrilaterals are defined, a meshed grid may be displayed. For the shaded
image to be rendered, normal vectors for each node on each quadrilateral are required. . . 95

Figure 4.12: When the polygons and normals have been defined, a lighting model may be applied
to render a shaded image of the geometry. The left image is a frameGrab from an
animation. ................................................................. 95

Figure 4.13: A bladed impeller or compressor wheel consists of a series of uniformly shaped
blades and cavities about a central hub. Only the cavities are milled out to create
the blades, the outer profile of the part results from a lathe turning used as raw
stock. .................................................................................... 97

Figure 4.14: A lathe profile may be used to generate a Z-buffer based 'surface of revolution'

defining the raw stock geometry for a bladed compressor or turbine wheel. ............. 97

Figure 4.15: Although a full geometric model of the raw stock may be generated (left),
it's more computationally efficient to model only the portion required to mill
one complete blade. ................................................................ 97

Figure 4.16: While polar coordinates could more efficiently model turned stock, it results in
non-uniform mesh spacing. This also results in a variable area represented by each
element, indicated by the size of the '+' above. ........................................... 100

Figure 4.17: To maintain a constant mesh spacing, turned stock is defined in standard

Workpiece coordinates. The model on the left has SX = SY = 1, while on the
right SX = SY = 0.9. .......................................................... 100

Figure 4.18: For a 3-axis flat end operation, DMin represents the minimum XWP,YWP move
distance that must occur for new material to be intersected (i.e. D > DMin) (left). As
shown on the right, continuous material removal may occur for a ball end mill or
a 5-axis flat end operation. .................................................... 104

Figure 4.19: The radial distance values found are a function of the start position in both Xwp
and Ywp, the distance traveled to the end position, and the direction of travel.
This is a result of the variable 'effective mesh spacing' seen orthogonal to the
Figure 5.1: The toolpath envelope defines the region swept in space by a single tool move and is used in intersection calculations for the simulation of material removal (left). The single cutter instance allows the flutes that engage the stock at a given rotational position to be identified using known flute geometry and the cut geometry of the current tool move (right).

Figure 5.2: The toolpath envelope is a closed solid constructed from a repeated set of geometric primitives. The example shown is for a ball end cutter, shown in 2D profile.

Figure 5.3: Basis vectors for the Toolpath Envelope Coordinates used in the calculation of intersection locations and cut geometry parameters. On the left is an isometric view, on the upper right is the view seen looking down $A$, and the lower right shows the view seen looking up $N$.

Figure 5.4: Although the toolpath envelope is modeled using a '3-axis approximation' with a constant cutter orientation ($A$), if the maximum change in orientation ($dA$) is restricted this can still accurately model a true 5-axis move.

Figure 5.5: Many typical 5-axis tool moves contain only minor variations in $A$ (far left set of cutters). However, there are cases where large variations in $A$ exist over a fairly short move. The most common of these are when the part being milled has a very small radius of curvature relative to the length of the tool move (middle set of cutters), and when the cutter has to tilt to prevent part interference (far right set of cutters).

Figure 5.6a: A 3-axis approximation of a 5-axis STE produces a geometric error $\varepsilon$ that is a linear function of the angle $dA$ and the distance $C$ from the center of rotation (for small $dA$).

Figure 5.6b: For a ball end cutter, the $C_{\text{Max}}$ value is typically measured up the axis of the cutter from the ball center. For a flat end cutter, if the highest point of axial contact is less than the cutter radius $R$, use $C_{\text{Max}} = R$ as error can also occur.
Figure 5.6c: The use of the average cutter orientation $A_{\text{avg}}$ minimizes the value of $\varepsilon$ for a given STE by providing the smallest possible value for $dA$. Similarly, the subdivision of moves that contain a large orientation change into a series of smaller "sub-moves" also reduces $\varepsilon$.

Figure 5.7a: Note the smoothness of this linked set of STEs (approx. 1000 tool moves) seen looking up $+Z_{wp}$. The dashed white line indicates the stock boundary, and the cutter is visible at right.

Figure 5.7b: In both images shown above, $R = 0.25''$, Z-buffer spacing = $0.0075''$, and a tool move with a net $dA$ of $60^\circ$ occurs over $2.75''$. On the left $\varepsilon = 0.075''$, and on the right $\varepsilon = 0.0075''$.

Figure 5.7c: Although all rotation is assumed to occur at the cutter location position, it is possible to accurately simulate rotations where the center of rotation is not at this position.

Figure 5.8: On the left an isometric view of an STE intersecting the Z-buffer model is shown. On the right is the view seen looking down the $Z_{wp}$ axis, with the toolpath envelope projected onto the $X_{wp},Y_{wp}$ plane. The bounds of the projected STE define the intersection zone containing the Z-buffer elements that could possibly intersect the current STE.

Figure 5.9: When intersecting the sphere that models the bottom of a ball end cutter in the final cutter position, it is possible to find no intersections (left-most image), one valid intersection and one invalid intersection that lies internal to the STE (middle image), or two valid intersections (right-most image).

Figure 5.10: On the left is a Z-buffer element intersecting an STE three times, twice on the bottom cylinder and once on a planar side. On the right is the same STE and Z-buffer element seen looking down the direction of travel. The physically valid intersections are simply the max and min (in $Z_{wp}$) intersection values in the case shown, or for any other set of intersections, and the remainder lie on portions of

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
the primitives internal to the STE. ................................................................. 128

Figure 6.1: The contact patch, shown shaded in the image on the right, defines the region
on the cutting tool where material removal may occur during a given tool move. ........ 133

Figure 6.2: The axial bounds $A_{\text{Max}}$ and $A_{\text{Min}}$ identify which axial discs engage the stock
material. Note that the bottom of the cutter does not have to engage the stock,
particularly milling steep faces with a ball end cutter, shown on left, or when
flank milling, shown on right. ................................................................. 133

Figure 6.3: Note that some axial discs bounded between $A_{\text{Max}}$ and $A_{\text{Min}}$ may not actually
intersect the current Z-buffer element, passing through gaps in the element instead. ...... 133

Figure 6.4: The entrance and exit angles bound the region of stock intersection on a given
axial disc for a given tool move. They may be easily calculated from the radial
depth of cut values $N_{\text{Max}}$ and $N_{\text{Min}}$, as shown in the image on the right. .... 136

Figure 6.5a: Traditionally defined entrance and exit angles often provide a conservative
representation of stock engagement, as shown above for the case of the shaded
axial disc on the far left. ................................................................. 136

Figure 6.5b: Error is also introduced by the traditional approach when ball end milling and
$A \cdot D < 0.$ .................................................................................... 136

Figure 6.6: Dividing axial discs into 4 quadrants, each with its own pair of entrance and
exit angles, allows for more accurate modeling of the cut geometry. .................. 138

Figure 6.7: Determination of the appropriate axial disc quadrant contacted by $P_z$ requires
knowledge of cutter location at the point of intersection when cutting with a ball
end mill and $A \cdot D < 0.$ ................................................................. 138

Figure 6.8: The quadrant method requires error compensation due to the discrete Z-buffer
stock model. ...................................................................................... 138

Figure 6.9: The effective mesh spacing $\varepsilon_{XY}$, seen at bottom looking down $D_{XY}$, at various
degrees of rotation. Note that the worst case conditions from 4 adjacent Z-buffer
elements are used, rather than simply 2 adjacent elements along $X_{WP}$ or $Y_{WP}$,
producing the max value of $\varepsilon_{XY}$. ...................................................... 142

xxi

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6.10: Assuming worst case conditions, $\varepsilon_N$ increases as $D_{XY}$ moves away from either
the $X_{WP}$ or $Y_{WP}$ principal direction, and as cutter tilt about $D$ increases. .................... 142

Figure 6.11: Pseudo-code displaying the integration of cut geometry calculation (in bold)
into the larger feed estimation framework. The method shown applies to a
single tool move. ................................................................. 145

Figure 6.12: When finding the axial location of a point $P_z$, the calculation must be performed
at the location where the point $P_z$ first contacts the surface of a cutter instance,
using the vector $Z_{CL}$ as shown on the left. If some other cutter position is used
the result is invalid, as shown on the right by $A_{em}$, calculated incorrectly using
the vector $(P_z - P_1)$. ................................................................. 147

Figure 6.13: Values for axial locations $A$ and distances traveled to the point of intersection $D_z$ are
known from the intersection calculations, where they are required to
bound intersections found with infinite primitives to the space actually occupied
by the current STE. ................................................................. 147

Figure 6.14: Issues in finding $DZ$ for some known $Z$=buffer location that is not an
intersection location ...................................................... 147

Figure 6.15: When $A$ and $D$ are not orthogonal, the axial location of $P_z$ projected down $D$ onto
the cutter surface is a function of its radial location; (note on left that $N$ crosses
several discs). Therefore $P_z$ must be indirectly mapped to $A$ via $A_P$, as shown on
the far right. ................................................................. 152

Figure 6.16: The mapping of $P_z$ along $-D$ and onto $A$ is a 2D operation in the $DA$ plane,
which neglects the 3D nature of the cutter surface in $D_P$ and $N$. .................... 152

Figure 6.17: The mapping of $P_z$ along $-D$ and onto $A$ is a 2D operation in the $DA$ plane,
which neglects the 3D nature of the cutter surface in $D_P$ and $N$. .................... 152

Figure 6.18: When $(D \cdot A) < 0$, the lowest axial position does not necessarily occur at the
intersection location $P_Z$ (i.e. at $d = DZ$) when the intersection occurs on the
'back' of the cutter. ...................................................... 155

Figure 6.19: When $(D \cdot A) < 0$, three situations may exist while identifying the axial bounds

xxii

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
of an intersected segment projected onto the cutter surface. Note that situation “1” also exists when both intersections occur on the front of the cutter. ................. 155

Figure 6.20: $R_{\text{Arc}}$ is the radius of the arc defined by the path of intersection in the $n_{\text{Arc}}$ plane, and its use compensates for rotation about $n_{\text{Arc}}$ as its value remains constant during rotation. The local coordinate system describes the relative orientation of the left and right-hand figures. .................................................. 157

Figure 6.21: Using $R_{\text{Arc}}$ and $n_{\text{Arc}}$, the calculation of $A_{\text{Min}}$ may be completed as shown above. ............ 157

Figure 6.22: Calculation of the radial distance $N$ from the known intersection location $P_z$. ............... 157

Figure 6.23: In a previous approach at finding the Max and Min bounds in the Axial and Normal directions, each axial disc element was intersected with the Z-buffer elements by modeling it as an individual STE. This provided accurate but inefficient results. However, this approach may prove useful in the modeling of cutters whose profiles must be represented in a piecewise manner, such as the odd shaped cutter shown at the right. ............................................................... 161

Figure 6.24: Comparison of the "dropped disc" STE series (left) vs. a single net toolpath STE (right); both as seen looking down the $D$ axis. Note that during operation the discs of the 'dropped disc' method are used individually for intersections, rather than together as shown. ................................................................. 161

Figure 6.25: In the 'swept circle' approach at identifying points of contact between an intersected segment and the appropriate axial discs, only a only a circle is swept at each axial disc location, rather than a net STE for each disc. This is done following an intersection calculation between an STE for the net toolpath and the Z-buffer model. ................................................................. 161

Figure 7.1: The relative velocity vector $V_{\text{rel}}$ describes the motion of the cutter relative to the stock, and is defined in the Machine Coordinate system of the G-code data. ............. 166

Figure 7.2: When purely linear motion occurs, the vector sum of the linear velocity components yields the net velocity vector independent of position or orientation. ............... 166

Figure 7.3: The linear velocity components produced by rotational motion depend on the
Figure 7.4: The radius vectors may be defined either as 3D Machine Coordinate vectors, or as
2D vectors in local 'Rotary Axis Coordinates'. In both cases the A axis is
shown as an example.

Figure 7.5: The table-on-table 5-axis arrangement used in this research. The stock translates in
X and Y, and rotates in A and B, while the cutting tool moves independently in Z.

Figure 7.6: The A rotary axis rides on the B rotary axis, and so the direction of A is a function
of B.

Figure 7.7: The use of Rotary Axis Coordinates can simplify linear velocity calculations as
the radius vector required is reduced to 2D and may be defined in terms of principal
axis positions.

Figure 7.8: The RotaryB cutter position provides the desired radius vector RB. Note that the
RotaryB coordinates do not rotate with the B axis (i.e. XB remains parallel to XM).

Figure 7.9: The cutter position in RotaryA provides the radius vector RA. While RotaryA rotates
with the B axis (on left), they not rotate with A (i.e. XA remains in the XM, ZM plane).

Figure 7.10: The relative velocity can vary with axial location on the cutting tool during 5-axis
milling as a result of the varying radius from the center of rotation.

Figure 7.11: Calculation of the YM component of linear velocity (which extends out of the
page) due to A axis rotation as a function of axial location and the cutter location
velocity for a table-on-table 5-axis arrangement.

Figure 7.12: Flow chart overview of 'standalone' relative velocity control software.

Figure 7.13: Constant velocity simulation results are shown for first pass in the 5-axis
"saddle surface." 

Figure 7.14: Plot of simulation results for a constant input velocity of 20 IPM and the
resultant relative velocity of the cutter past the stock.

Figure 7.15: Plot of simulation results for a constant relative velocity of 20 IPM and the
feed values required to maintain that velocity.

Figure 7.16: Simulation results of the feed values required for a constant relative velocity of 8 IPM.
Figure 7.17: Plot showing A axis saturation at +/- 1500 deg/min, which limits the ability to maintain the desired relative velocity of 20 IPM. ................................. 189

Figure 7.18: When the B axis hits its hard limit of =5 deg., the A axis must rotate 180 deg and the direction of milling is reversed for the cutter to maintain the desired cutting path. 189

Figure 8.1: On the ball portion of the cutting tool, the positive and negative radial distances change continually. Therefore the stepover distance may better describe the cut conditions. Note that on cylindrical cutter portions, the radial distance and the stepover distance are equal. ................................................................. 195

Figure 8.2: Certain sets of cutting conditions result in a near zero average X milling force. In the above case, the average force value is -0.1283 Lb. This should be avoided. 195

Figure 8.3: An axial depth of one radius includes only the cutter ball portion of the cutter in the cut, while an axial depth of one diameter, for a constant stepover, includes the same cutter ball portion, plus an equal axial length of the cylindrical portion of the cutter. 195

Figure 8.4: Graphical comparison of material constants for the case of the 0.375" ball end cutter of Table 8.1................................................................. 207

Figure 8.5: Regardless of the stepover distance $D_s$, the net radial depth $D_R$ of cut remains equal to it until an axial depth equal to the top of the cusps However, the individual positive and negative radial depth contributions vary. (The cutter is moving out of the page). 209

Figure 8.6: The stepover distance $D_s$ and the radius of the current axial disc may be used to calculate change in radial depth $D_R$ that occurs on the cutter ball. 209

Figure 9.1: The Z-buffer model was graphically rendered for the validation of intersection calculation and storage routines using controlled cutting simulations. Testing was performed with relatively few Z-buffer elements, as the operations function identically for all elements and this simplified the task of isolating problems. 213

Figure 9.2: Diagram of the system used for milling force data acquisition. 216

Figure 9.3: Comparison of the resolution of measured force data over 3 cutter revolutions for two spindle speeds, both are measured at the maximum sampling rate of 1000 Hz. 216

Figure 9.4a: Note if the sampling period shown above by the dashed line is selected, an inaccurate
estimate of peak and average force will result over all rotations. This is because the same angles are sampled every rotation of the cutter. .............................................................218

Figure 9.4b: Although the sampling period indicated above is similar to that of 9.4a, it does not sample the same angles every cutter rotation. This phase shifting between rotations allows for the collection of a wider range of data. .............................................................218

Figure 9.5: Simulation of the test cuts performed for the calculation of the material constants. Correct simulation is for the cutter to pass all the way through the stock model. ...............221

Figure 9.6: Samples of the measured XY force magnitude values presented in Table 9.1. A 2.5 second sampling of data taken at 1000 Hz is shown. Note that the scales differ between plots to maximize visual clarity. .............................................................225

Figure 9.7: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.1875" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter. .............228

Figure 9.8: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter. ...........229

Figure 9.9: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter. ...........230

Figure 9.10: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter. ...........231

Figure 9.11: Definition of the sinusoidal test cut shown in Z, a similar definition may also occur in Y. The figure on the bottom depicts a cross section of a test cut in progress. ...............234

Figure 9.12: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 6 IPM and a spindle speed of 3500 RPM. .................................235

Figure 9.13: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 15 IPM and a spindle speed of 7000 RPM. .................................236

Figure 9.14: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 18 IPM and a spindle speed of 4500 RPM. .................................237

Figure 9.15: Comparison of predicted and measured forces for a sinusoidal partial immersion cut at a constant feedrate of 18 IPM and a spindle speed of 4500 RPM. .................................238
Figure 9.16: Experimental forces for optimized feedrates versus a constant feedrate. The desired force was 75 Lb., however these results were obtained prior to the inclusion of runout.

Figure 9.17: Feedrate-controlled experimental force data for the sinusoidal slot cut of Figure 9.12. Data for the same cut at a constant 6 IPM is presented in Figure 9.12.

Figure 9.18: Feedrate-controlled experimental force data for the sinusoidal slot cut of Figure 9.13. Data for the same cut at a constant 15 IPM is presented in Figure 9.13.

Figure 9.19: Feedrate-controlled experimental force data for the sinusoidal partial immersion cut of Figure 9.15. Data for the same cut at a constant 18 IPM is presented in Figure 9.15.

Figure 9.20: Feedrate-controlled experimental force data for a sinusoidal partial immersion cut with a large amplitude in Z, ranging from a minimum depth of 0" to a maximum depth of 0.5625". The cutting and simulation conditions are also provided above.

Figure 9.21a: Force results for the cut conditions presented in Figure 9.18 (including feed values), only using a different, but identical, cutting tool.

Figure 9.21b: Force results for the cut conditions presented in Figure 9.20, except a stepover distance of 0.1125" (0.3 diameter) was used. The feed values were re-calculated for this distance.

Figure 9.22: Before and after images of the pre-finish operation for which adaptive feeds were generated. The central blade in the top image is the area of interest.

Figure 9.23: Simulation results for the manually optimized feedrates (including use of the NC machine model to maintain constant velocity), originally used during production of the part.

Figure 9.24: Simulation results for automatically generated feedrates to maintain a force of 50Lb., with a maximum feed of 55 IPM and no limits on chip thickness.

Figure 9.25: Simulation results for automatically generated feedrates to maintain a force of 50Lb., with a maximum feed of 70 IPM and a 0.001" limit on chip thickness.

Figure A.1: On the left, Trochoidal Surface S1 is created as flute fl rotates $2\pi/Nf$ radians.
and translates forward one feed per tooth distance. On the right, the Trochoidal Chip Thickness $h(\beta)$ at some angle $\beta$ is the difference between the surfaces generated by the current flute, $S_2$, and the initial flute, $S_1$. In this view the actual cutter radius vectors are shown, originating at the instantaneous cutter centers that were present when the surfaces were generated. Note that the angle to the radius vector that created $S_1$ lags the value of $\beta$ for which the chip thickness is desired. .......................... 270

Figure A.2: The trochoidal chip thickness is calculated as the magnitude of the difference between the vectors shown on the left. Note the lead/lag error in the angle for the corresponding $S_1$ and $S_2$ positions. On the right a means of obtaining the maximum lead/lag angle, $\varepsilon_{\text{MAX}}$, is shown (valid for $f \ll R$; not to scale). When the value of $\beta$ for $S_2$, $\beta_2$, $= 180^\circ$, the corresponding $\beta_1$ value is $(180^\circ - \varepsilon_{\text{MAX}})$; similarly when $\beta_2 = 0^\circ$, $\beta_1 = \varepsilon_{\text{MAX}}$. The value of $\beta_1$ is defined for intermediate values as $\beta_1 = \beta_2 - (\varepsilon_{\text{MAX}})(\cos(\beta(S_2)))$. ........................................ 270

Figure A.3: Trochoidal (Dark Line) Vs. Sinusoidal (Light Line) Chip Thickness as a function of rotation angle for $R = 0.09375''$, $f = 0.001$ in/tooth. Note that there is only a slight variation between the two approaches for these fairly common conditions. The maximum lead/lag angle in this case is $0.61$ deg. ...................................................... 273

Figure A.4: Trochoidal (Dark Line) Vs. Sinusoidal (Light Line) Chip Thickness as a function of rotation angle for $R = 0.09375''$, $f = 0.01$ in/tooth. Note the non-zero chip thickness values of the trochoidal form at 0 and 180 deg., as well as the overall increase in thickness from 0 to 90 deg. The maximum lead/lag angle in this case is $6.1$ deg. ............. 273

Figure B.1: In the table-on-table 5-axis machine configuration, the stock translates in X and Y, and rotates in A and B, while the cutting tool moves independently in Z. ............... 276

Figure B.2: The Rotary$_B$ coordinates remain parallel to Machine Coordinates at all times. Similarly, while Rotary$_A$ rotates on B, they do not rotate with A ($X_A$ remains in the $X_M$, $Z_M$ plane). .............................................................. 276

Figure B.3: Definition of the Machine Coordinate origin and the Rotary Table Coordinate origin for the Fadal table-on-table 5-axis configuration. ........................................ 278

Figure B.4: Definition of the workpiece offsets used to specify stock placement on the rotary
Figure B.5: Definition of the workpiece offsets used to specify stock placement on the rotary table.................................................................281

Figure C.1: Note that the portion of cutting tool indicated as not intersecting the discrete stock (highlighted in bold) generally increases with radial depth and is a function of mesh spacing $d_v$. This leads to errors in the entrance/exit angles, indicated by shaded region.................................................................285

Figure C.2: In the above image the Z-buffer elements are shown swept parallel to the direction of travel. Note that the chip thickness at flute $fl$ is approximately 28% of the sum of the chip thickness values for flutes $fl$ and $f2$ at the angle of disengagement, $\beta(e)$..............285

Figure D.1: Calibration cut direction.................................................................290

Figure D.1: Sample force plot showing transient and steady-state regions.................................................................291
LIST OF SYMBOLS

A is the cutter orientation vector
A(i) is the axial offset of the current disc from the cutter location position (in)
Adot is the machine A rotational axis velocity (deg/min)
Adot_{Max} is the maximum allowable A axis velocity (deg/min)
Adot is the rotational velocity vectors for the A axis (rad/min)
Adir is the machine A axis rotational direction vector
A_{Min} is the minimum axial location of contact between cutter and stock (in)
A_{Max} is the maximum axial location of contact between cutter and stock (in)
A_P is the component of the cutter orientation vector that lies orthogonal to D
A_{proj} is the location formed when the vector Zi is projected back along D and onto A_P (in)
A_{xy} is the area associated with a single z-buffer element (in^2)
Bdot is the machine B rotational axis velocity (deg/min)
Bdot_{Max} is the maximum allowable B axis velocity (deg/min)
Bdot is the rotational velocity vectors for the B axis (rad/min)
Bdir is the machine B axis rotational direction vector
C_{Max} is the axially highest location of stock contact on the cutter (in)
D is the net linear distance between the start and end positions for the move (in), and
dA is the distance the A axis must translate during the current tool move (deg)
dA is the angle between two possible orientation vectors (rad)
 dB is the distance the B axis must translate during the current tool move (deg)
 d_{Cyl} is the constant disc thickness orthogonal to the cutter cylinder (in),
 d_{Ball} is the constant disc thickness orthogonal to the ball cutter profile normal (in)
 dP is the total distance translated by a given axis for the current move (in)
 dP_{M}(n) is the differential amount moved by each axis for net tool move number n
 dT is the time taken for a single tool move to occur (min)

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
dT_M is the time it takes to complete the current tool move (min)
dX is the distance the X axis must translate during the current tool move (in)
dY is the distance the Y axis must translate during the current tool move (in)
dZ is the distance the Z axis must translate during the current tool move (in)
dZ is the axial disc thickness (in)
dZ(z) is the current axial disc thickness amount inside the summation (in)
dZBall(i) is the axial thickness of the current (i^{th}) axial disc (in)
d\lambda is the angular range covered by two milled cavities (one complete blade) (rad)
d\theta is the angular increment amount (rad)
d\sigma is the user defined angular increment for disc definition (rad)
d\psi is the cutter ball angular disc spacing amount (rad)
D is a unitized direction of travel vector for the cutter
D_{min} is the minimum radial (toolpath Normal dir) location of contact between cutter and stock (in)
D_{max} is the maximum radial (toolpath Normal dir) location of contact between cutter and stock (in)
D_P is the location of an intersection in the cutter surface in the cutting tool D_P direction (in)
D_P is the component of the direction of travel vector for the cutter that lies orthogonal to A
D_{XY} is the component of the cutter direction of travel that lies in the X_{wp}, Y_{wp} plane
Err is the percent difference between the estimated and actual peak force values
E_{ui} are the upper and lower ends the intersected segment (in)
f is the programmed feedrate value (in/min)
f(n) is the feed used for the current iteration (ipm)
f(n+1) is the input feed for the next iteration (ipm)
f_{calc} is the current iteration feed value that resulted in excessive chip thickness (ipm)
f_{calc}(m) is the calculated feed for current move (ipm)
f_{out} is the feed value output to the updated G-code file for the current tool move (ipm)
f_{t} is the feed-per-tooth value (in/tooth)
f_{net} is the net Feed-Per-Tooth value (in/tooth)
F_{desired} is the desired force magnitude (Lb)
\( F_{\text{Max}}(n) \) is the peak force magnitude for the current iteration (Lb)

\( F_{\text{avg}} \) is the average tangential force (Lbs)

\( F_x \) is the \( X_{CT} \) and vector cutting force acting on the cutter at cutter angle \( \Theta \) (lbf)

\( F_y \) is the \( X_{CT} \) and vector cutting force acting on the cutter at cutter angle \( \Theta \) (lbf)

\( F_{y-\text{avg}} \) is the average side force acting on the cutter (Lb)

\( h \) is the chip thickness (in)

\( h(\beta) \) is the chip thickness as a function of rotation angle (in)

\( h_{\text{avg}} \) is the average chip thickness over the net contact region (in)

\( h_{\text{Max}} \) is the maximum chip thickness value calculated during the current tool move (in)

\( h_{\text{Max,Desired}} \) is the user defined desired maximum chip thickness value (in)

\( \text{HelixIncr} \) is the desired helical increment per disc (rad)

\( i \) is the absolute index to the current axial disc element

\( i_{\text{Max/Min}} \) is the index to the upper or lower bounding discs

\( L_{\text{net}} \) is the net length of Z-buffer element removed (in)

\( \text{InStock} \) is a flag indicating if the current axial disc engages the stock during the current tool move

\( i_{\text{Zbuffer}} \) is the Index to the current Z-buffer element (beginning with 0)

\( j \) is an index to the current rotation angle

\( k \) is an index denoting the current flute

\( K \) is the Unit Power Consumption, a material constant (hp min / in3)

\( K_{T,} \) is the tangential force mechanistic material parameters (Lbf/in2)

\( K_{TC} \) is an empirically derived material constant (Lbf/(in)^2-P1)

\( K_{R} \) is radial force ratio amount mechanistic material parameters (Lbf/Lbf)

\( K_{RC,} \) is also an empirically derived material constant (Lbf/(in)^2 / Lbf

\( L_f \) is the length of the current flute segment on the current axial disc (in)

\( L_{f(i)} \) is the flute segment length of the ith disc (in)

\( L_s \) is the sum of the lengths of all gaps that lie within the intersected segment (in)

\( L_{si} \) is solid length of the the ith intersected segment (in)

\( M_{\text{element}} \) is the memory required for a single Z-buffer element (Kilobytes)
is the net memory required for all Z-buffer elements (Kilobytes)

nc is the outward surface normal direction of the cutting tool

is the axial component of the cutter surface unit normal for the current axial disc (unitless)

is the radial component of the cutter surface unit normal for the current axial disc (unitless)

Hc is the axial component of the cutter surface unit normal for the current axial disc (unitless)

is the number of blades on the part

Ncb is the number of disc elements present on the ball portion of the cutter

Nt is the number of flutes (or inserts) on the cutting tool (tooth)

Nm is the number of elements intersected during the move

Nset is the total number of Z-buffer elements in the model

Nsub is the number of sub-moves required (unitless)

NX is the number of Z-buffer elements in Xwp

NY is the number of Z-buffer elements in Ywp

NZ is the number of cutting flutes on cutting tool

N0 is the number of discrete rotational positions

P is the power consumed (hp)

Pcx,y,z is the location of the center of the sphere (in)

Pf, is the final tool position in workpiece coordinates

Pi is the initial G-code cutter location of a given axis for the current move (in)

Pi is the initial tool position in workpiece coordinates

PM is a machine position location that represents the current machine positional state

\{X_m, Y_m, Z_m, A, B\}

PM (n-1) is the start position of the net tool move (from G-code)

PM-sub(m) is the end position for all machine axes of the mth submove

[Pm] is the machine coordinate position values [X, Y, Z, A, B] found in the G-code file

[Pwp] is the workpiece coordinate cutter position and orientation [X, Y, Z, I, J, K]

PZ is a known point if intersection between the cutter and stock in workpiece space
$$P_1$$ is an empirically derived material coefficient (rad)
$$P_2$$ is an empirically derived material coefficient (rad)
$$Q$$ is the volumetric removal rate (in³/min)
$$R$$ is the cutter radius (in)
$$R(i)$$ is the current axial discs radius value (in)
$$R(i_{ball})$$ is the radius of disc i on the cutter ball (in)
$$R(i_{cyl})$$ is the radius of disc i on the cutter cylinder (in)
$$R_A$$ is the radius to a tool position from the A axis of rotation, defined orthogonal to A rotation (in)
$$R_B$$ is the radius to a tool position from the B axis of rotation, defined orthogonal to A rotation (in)
$$R_{MC}$$ indicates the workpiece coordinate space
$$R_{wp}(i)$$ is the radius to the $$i^{th}$$ Z-buffer element from the rotational origin (in)
$$S$$ is the sampling rate (Hz)
$$S_X$$ is user defined scaling factor in $$X_{wp}$$ that defaults to a value of 1.0
$$S_Y$$ is user defined scaling factor in $$Y_{wp}$$ that defaults to a value of 1.0
$$t$$ is the current sample time (sec)
$$t_0$$ is the sample time at which the first sample in the flute force profile occurs (sec)
$$[T]$$ is the transformation relation (function of machine kinematics and part setup)
$$TOL$$ is some small tolerance amount to account for round off error
$$V$$ is the surface cutting velocity (ft/min)
$$V$$ is the contribution to the net relative velocity vector from the machine linear axes (in/min)
$$V_A$$ is the contribution to the net relative velocity vector from the machine 'A' rotary axis (in/min)
$$V_A(A)$$ describes the A axis velocity contribution as a function of axial location (rad/min)
$$V_{ax,y,z}$$ is the linear velocity contributions from the machine A rotational axis (in/min)
$$V_B$$ is the contribution to the net relative velocity vector from the machine 'B' rotary axis (in/min)
$$V_B(A)$$ describes the B axis velocity contribution as a function of axial location (rad/min)
$$V_{bx,y,z}$$ is the linear velocity contributions from the machine B rotational axis (in/min)
$$V_M$$ is the volume removed during the current move (in³)
\( V_{\text{act}} \) is the net relative velocity vector of the cutter past the stock (in/min)

\( V_{\text{net}} \) is the magnitude of the net relative velocity vector of the cutter past the stock (in/min)

\( V_{\text{act}(i)} \) is the relative velocity for the \( i \)th axial disc (in/min)

\( V_{\text{net}x} \) is the X component of the net relative velocity vector (in/min)

\( V_{\text{net}y} \) is the Y component of the net relative velocity vector (in/min)

\( V_{\text{net}z} \) is the Z component of the net relative velocity vector (in/min)

\( V_{\text{RR}} \) is the volumetric removal rate (in³/min)

\( V_x \) is the individual machine linear Z axis velocity contribution (in/min)

\( V_y \) is the individual machine linear Z axis velocity contribution (in/min)

\( V_z \) is the individual machine linear Z axis velocity contribution (in/min)

\( X(i) \) denotes the \( X_{\text{wp}} \) base position of the \( i \)th element in Workpiece coordinates

\( X_{\text{dot}} \) is the \( X_{\text{wp}} \) base position of the \( i \)th element in Workpiece coordinates

\( X_{\text{dot max}} \) is the maximum allowable \( X \) axis velocity (in/min)

\( X_{\text{RO}} \) is the \( X_{\text{wp}} \) coordinate location of the rotational origin (in)

\( Y(i) \) denotes the \( Y_{\text{wp}} \) base position of the \( i \)th element in Workpiece coordinates

\( Y_{\text{dot}} \) is the \( Y_{\text{wp}} \) base position of the \( i \)th element in Workpiece coordinates

\( Y_{\text{dot max}} \) is the maximum allowable \( Y \) axis velocity (in/min)

\( Y_{\text{RO}} \) is the \( Y_{\text{wp}} \) coordinate location of the rotational origin (in)

\( Z \) is the summation index to the current axial disc

\( Z_{\text{CL}} \) is a vector from the cutter location position to a known intersection location

\( Z_{\text{dot}} \) is the \( Z_{\text{wp}} \) coordinate location of the rotational origin (in)

\( Z_{\text{dot max}} \) is the maximum allowable \( Z \) axis velocity (in/min)

\( Z_i \) is a vector between the initial tool position and a known intersection location

\( \alpha \) is the initial reference angle (rad)

\( \alpha_{\text{helix}}(z) \) is the current axial disc's cutter helix angle inside the summation (rad)

\( \alpha_{\text{helix}(i_{\text{Ball}})} \) is the helix angle on disc \( I \) of the cutter ball

\( \beta(i, j, k) \) is the angular position of the current flute (rad)

\( \beta_{\text{em}} \) is an entrance angle (rad)
\( \beta_{\text{ex}} \) is an exit angle (rad)

\( \delta \) is the angle of tilt of the cutter from \( Z_{\text{wp}} \) in the \( N,A \) plane

\( \delta_{\text{Dir}} \) is the angular change in direction between moves (Deg)

\( \varepsilon \) is the geometric error between the actual and as-modeled surface (in)

\( \varepsilon_{N} \) is the known separation in the cutter Normal direction, or 'allowable gap error' (in)

\( \varepsilon_{XY} \) is the known separation in the \( X_{\text{wp}}, Y_{\text{wp}} \) plane, or 'effective Z-buffer spacing' (in)

\( \gamma \) is the angular separation between adjacent flutes (rad)

\( \gamma_{\text{min}} \) is the minimum value between \( \gamma_{1} \) and \( \gamma_{2} \) (rad)

\( \gamma_{1} \) is the angle between \( D_{XY} \) and \( X_{\text{wp}} \) (rad)

\( \gamma_{2} \) is the angle between \( D_{XY} \) and \( Y_{WP} \) (rad)

\( \theta_{(j)} \) is the current cutter rotational position reference angle (rad)

\( \rho \) is a runout amount (in)

\( \sigma_{(i)} \) is the cutter ball flute definition angle for disc \( i \) on the cutter ball (rad)

\( \tau_{(i)} \) is an angular offset due to the cutter helix angle and axial location (rad)

\( \psi \) is the angle between \( A \) and \( AP \), and also \( D \) and \( DP \) (rad)

\( \Delta A \) is the distance the \( A \) axis must rotate during the current tool move (deg)

\( \Delta B \) is the distance the \( B \) axis must rotate during the current tool move (deg)

\( \Delta T \) is the time it takes to complete the current tool move (min)

\( \Delta X \) is the distance the \( X \) axis must translate during the current tool move (in)

\( \Delta Y \) is the distance the \( Y \) axis must translate during the current tool move (in)

\( \Delta Z \) is the distance the \( Z \) axis must translate during the current tool move (in)

\( \Lambda \) is the number of degrees of cutter rotation between samples (deg)

\( \Theta \) is the cutter rotation angle (deg.)

\( \Omega \) is the spindle speed (RPM)
ABSTRACT

DISCRETE MODELING OF SCULPTURED SURFACE MACHINING
FOR ROBUST AUTOMATIC FEEDRATE SELECTION

by

Jeffrey G. Hemmett

University of New Hampshire, May, 2001

Traditional feedrate selection techniques currently used in three and five-axis CNC machining reduces milling efficiency. Manually estimated feedrates tend to be conservative and constant, greatly increasing mill time. The goal of this research is to develop robust techniques and software tools for automatically generating optimized feedrates for use on three and five-axis CNC mills, to both simplify the feed selection process and to increase the safety and efficiency of the milling operation through milling process simulation.

The simulation software estimates milling force vectors for each tool move, and identifies a feedrate that maintains a desired peak force. The desired cutting force value may be selected to prevent cutter breakage, maintain part tolerance, or meet some other criteria. Other conditions are also considered, such as maximum allowable chip thickness and machine constraints. This allows for the generation of variable feedrates that are optimized for each tool move.

The software consists of three distinct portions: a discrete mechanistic model, a discrete geometric model, and a CNC machine model. The mechanistic model estimates cutting forces as a function of cut geometry, cutter/stock relative velocity, and material constants. The geometric model keeps track of the changing in-process stock geometry and provides the cut geometry parameters required by the mechanistic model. The CNC machine model calculates the cutter/stock relative velocity based on feed inputs, machine kinematics, and controller behavior. A feed value is calculated in an iterative manner for each tool move based on the force estimates. The results of this research have produced accurate force estimates during sculptured surface machining, and have also demonstrated that this approach at automatic feedrate selection...
is feasible. Testing of feedrate selection has included the five-axis milling of production turbomachinery in an industrial environment. An average improvement in efficiency of 20% has resulted from the use of the optimized feeds.
CHAPTER 1

INTRODUCTION

1.1 Introduction

The primary focus of this research is on the automatic selection of optimized feedrates to improve the efficiency of 3 and 5-axis sculptured surface machining. The methods are specifically developed for 5-axis sculptured surface machining, as this represents the most complex and difficult type of NC milling. However, 2, 21/2, 3, and 4-axis milling may be treated as special cases of five-axis milling by fixing the appropriate axes of the 5-axis model.

The use of optimized feeds reduces the machining time necessary to cut a given part, and can improve process reliability by reducing the chance of cutter breakage. The task of feedrate selection is also simplified. It is difficult to optimize feeds using traditional manual feedrate selection methods as a result of the complexity of the problem, combined with the large number of tool moves in a part program. Therefore, a software based analytical approach is used to provide a practical solution to this problem. Computer simulation of the milling process makes it possible to generate optimized feeds that vary with the cutting conditions of each tool move, and that are near optimal based on a set of user-defined constraints. These constraints include a maximum allowable force set to limit deflection and prevent breakage, maximum allowable axis velocities imposed by machine constraints, and a maximum allowable chip thickness value that limits tool wear during light cuts. Other constraints may also be specified as necessary.

This automated approach represents a shift in paradigm from traditional feedrate selection techniques. At the present time, feed values are typically manually defined, resulting in constant feed values that are overly conservative to protect the CNC machine and the part being milled. Without knowledge of the cutting conditions, manually defined feeds are set based on a single or limited set of perceived worst-case conditions that may occur during the milling process. At best, these feeds provide safe (due to their conservative nature) yet inefficient operation, and at worst they can lead to large part or cutter deflections, or even breakage. Automatic optimized feedrate selection attempts to overcome these limitations by
defining feeds that minimize mill time, while preventing cutter or part damage by selecting a unique feed value tailored to the conditions present during each tool move.

1.2 NC Milling Overview

A basic overview of milling and Computer Numerical Control (CNC, or simply NC) is now presented. This should aid in the understanding of some of the problems, and solutions, presented in this thesis, as well as define some of the basic terminology.

The milling operation is a material removal process, where small chips of material are removed from a piece of raw stock to obtain a desired shape. The raw stock may be in the form of a rectangular ingot or billet, or it may be the result of some prior manufacturing operation such as casting, turning, or previous milling. Whatever the state of the raw stock, it is at least slightly oversized so that excess material may be milled away. The benefits of milling include dimensional accuracy and excellent surface finish capabilities, combined with relatively low cost and faster process times (as compared to EDM, ECM, or some other manufacturing method of comparable accuracy).

The material removal process occurs over a set of discrete tool moves, or cuts. The 'cut geometry' describes the area of contact between the cutter and stock for each tool move, and is generally defined in terms of axial depth and radial depth. Axial depth defines how much of the tool, from the bottom of the cutter, is engaged in the stock along the axis of rotation. The radial depth describes how deeply the cutter is engaged in the stock material in a direction that lies orthogonal to both the cutter rotational axis and the direction of travel. See Figure 1.1 for a graphical representation of axial and radial depth. A case of full radial immersion is called a 'slot cut'. Notice that when partial immersion cutting (i.e. not slot cutting), two possible stock contact conditions can exist. These are referred to as Up (or Conventional) milling, and Down (or Climb) milling conditions, and are presented graphically in Figure 1.2.

Cutting tools are available in many shapes, and are denoted by their profile relative to the axis of rotation. The most commonly used cutters are ball end mills and flat end mills. 'Bull nosed' or 'round end' Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 1.1: Some basic cut geometry definitions used to describe stock engagement during the milling process.

Figure 1.2: During 'Down Milling', the motion of the flute due to rotation opposes the relative direction of travel, while during 'Up Milling' the flute travels in the direction of relative motion.

Figure 1.3: Various cutter profiles, and a visual depiction of the cutter helix angle $\alpha_{\text{helix}}$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
cutters, which are similar to a flat end cutter except there is a radius joining the sides and bottom of the cutter profile, are also frequently used. Cutters typically have a cylindrical shaft, but they may also exist in a 'tapered' format, where the sides of the profile lie at some angle to the axis of rotation. Shaped cutters are sometimes also used, with the profile of the cutter producing some desired shape, similar to a router bit. During milling, material is physically removed by sharp edges located on the rotating cutter. These sharp edges have a specific geometry on the cutting tool, and may either be an integral part of the cutting tool, known as "flutes", or they may be in the form of removable "inserts". The basic geometry of flutes or inserts is described on the cutter by the 'helix angle', defined relative to the axis of rotation. This angle may vary with axial location on the cutter. Figure 1.3 shows several cutter profiles as well as the helix angle definition.

The rotational velocity of the cutting tool (its Spindle Speed, $\omega$), and the relative velocity of the cutter past the stock $V_{net}$, combine to define the thickness of the chip being removed by a given flute at any instant during the rotation of the cutter. This value is referred to as the "chip thickness" ($h$). To facilitate chip thickness calculations, the spindle speed and relative velocity are often combined into a single "feed-per-tooth" value, $f_{net}$, defined as:

$$f_{net} = \frac{V_{net}}{\omega N_f}$$

where $f_{net}$ is the net Feed Per Tooth value (in/tooth), $V_{net}$ is the net cutter to stock relative velocity (in/min) $N_f$ is the number of flutes (or inserts) on the cutting tool (tooth), and $\omega$ is the spindle speed (RPM).

This value defines the distance the cutter moves forward during the period required for it to rotate the angular distance between two adjacent flutes, and is equal to the maximum theoretical chip thickness. See Figure 1.4 for a visual depiction of the chip thickness and feed-per-tooth concepts. In this figure, during the period required for the cutter to rotate 90° such that a flute at Pos. 1a on the dashed cutter (seen looking down the cutter axis) has moved to Pos. 1b on the solid cutter, the cutter translated forward...
Figure 1.4: Visual depiction of the maximum chip thickness $b_{Max}$ and feed-per-tooth $f_{set}$.

Figure 1.5a: Two views of a simple 3-axis vertical CNC depiction.
one feed-per-tooth value \( f_{\text{net}} \). The maximum chip thickness \( h \) is equal to \( f_{\text{net}} \) (Figure 1.4), along the direction of travel. The chip thickness then thins to zero away from the direction of travel, which may be described using:

\[
    h = f_{\text{net}} (n_T \cdot D)
\]

where

\( h \) is the chip thickness (in),

\( f_{\text{net}} \) is the net feed-per-tooth value (in/tooth),

\( n_T \) is the outward surface normal direction of the cutting tool, and

\( D \) is the cutting tool's direction of travel unit vector.

The chip is thickest along regions of the cutter that lie orthogonal to the direction of travel, and it thins as the cutter surface becomes parallel to the direction of travel (this effect is referred to as "chip thinning"). Therefore the chip thickness distribution is a function of cutter type, the regions on the cutter that intersect the stock, and the direction of travel of the cutter. These variables compose the "cutting conditions" required to estimate milling forces for a given tool move.

In Computer Numerical Controlled (CNC) milling, the motion of the cutting tool is controlled by a digital computer using feedback loops, with positional data input provided in the form of a data file. This allows a single machine to produce parts that vary greatly in size and geometry, limited only by physical machine constraints. Typically, a dedicated computer is used as the NC controller, but recently PC-based "open-architecture" controllers have also come into use. It is the job of the controller to dictate the motion of each degree of freedom present on the machine so that the net motion of the cutting tool occurs at the desired positions and velocities.

The most common type of CNC machine is a '3-axis' model. This name refers to the 3 degrees of freedom present on the machine, which form a three-dimensional Cartesian coordinate system. As shown in Figure 1.5a, during 3-axis milling the cutter is free to move along the X, Y, and Z axes, but is fixed in orientation. In the context of this dissertation, these coordinates will be referred to as the 'Machine Coordinate' system, \( X_M \), \( Y_M \), and \( Z_M \). Control of 3-axis machines is a relatively straightforward task, as...
the 3 degrees of freedom are decoupled. While these machines are useful in milling prismatic parts and simple sculptured surfaces, they are limited in their ability to efficiently produce complex geometries due to the fixed cutter orientation. However, they are important as they represent the vast majority of NC machines currently in use.

A second type of commonly available NC mill is a '5-axis' machine, which has the same Cartesian 3-axis basis to provide linear degrees of freedom, but also includes two rotary degrees of freedom (see Figure 1.5b). In this figure a rotary table mounted on a standard 3-axis CNC machine provides the rotational motion. The rotary axes are typically denoted "A" and "B", or "A" and "C", depending on the machines manufacturer and its physical configuration. The rotary degrees of freedom allow for changes in the cutting tool orientation relative to the stock being milled to allow for the creation of more complex geometries than possible on a 3-axis mill (see Figure 1.6a). The rotary motion also allows a flat end or bull nosed tool to be tilted such that the ellipse formed matches the local surface curvature, resulting in more efficient material removal (See Figure 1.6b), although this technology is still in the research stages [L98] [WHJ98].

Unfortunately, the presence of rotary motion also has several drawbacks as compared to 3-axis mills, including an increase in cost, increased programming difficulty, and reduced machine stiffness. There are also functional problems introduced by the rotary axes, such as coupled motion, variable velocity contributions to $V_{ax}$, singularity points (a combination of cutter location and orientation for which there is no unique solution for the required axis positions), and discrepancies between the desired and actual tool paths during large tool moves. The latter effect occurs because the CNC machine controller and rotary axes do not fully replicate the linearly interpolated toolpaths created by the CAM software used during toolpath definition. This interpolation error is usually not significant over short tool moves, but can introduce appreciable error for long ones. A model of the kinematic and controller characteristics for a given machine is necessary to simulate, identify, and eliminate these types of errors.

Manufacturers of 5-axis CNC mills use many different kinematic arrangements to include the rotational degrees of freedom, two of which are shown in Figure 1.7. This makes a completely generalized solution for automatic 5-axis feedrate selection difficult, as a unique model is required to represent a given physical machine configuration. Rotary motion results in position and velocity values that are a function of
Figure 1.5b: Two views of a simple 5-axis CNC depiction.

Figure 1.6a: 5-axis milling can produce more complex geometry than 3-axis milling. In 3-axis Milling (left), the cutting tool is constrained to a single orientation, unlike 5-axis Milling (right), where the cutter is free to rotate relative to the stock.

Figure 1.6b: The use of 5-axis mills can improve milling efficiency. To produce identical surfaces to the same tolerance, the 5-axis milled part on the left requires 2/3 fewer tool passes than the 3-axis milled part on the right (count cusps).
Figure 1.7: Two different kinematic configurations used to achieve five-axis motion on a CNC mill.
the radial distances from the center of rotation of each rotary axis to the current cutting tool position. These radii are a function of a given setup configuration, the tool length, and the current cutter position; information that is generally not considered by the current generation of controllers. This 'unknown radius' problem results in many of the velocity related difficulties experienced when 5-axis milling. However, using knowledge of the part setup, tooling information, machine kinematics, and controller behavior it is possible to modify the part programs "off-line", prior to their use. When these modified part programs are used, the desired cutting speeds are achieved through feedrate compensation. Similar compensatory methods could also be implemented directly in the controller.

The most common form of data input for CNC machine controllers are G-codes (or M&G-codes). G-codes consist of symbolic commands that instruct a machine to perform a specific action, such as "Turn Spindle On" or "Turn Coolant Off". They also provide user-defined values such as spindle speed, feedrate, and position data. Each line of G-code typically contains a single instruction, referred to as a "block" of G-code. The complete set of blocks required to mill a desired part is referred to as a "part program", and may consists of several different operations that use different tools and/or milling techniques. Generally the most common type of block in a G-code file is position data, and it is not uncommon to have tens of thousands, or even hundreds of thousands, of position data blocks in a part program. This is especially true in sculptured surface and/or 5-axis milling, where many small moves are typically used to ensure accurate positional control and geometric accuracy.

Below is an example of 8 blocks of G-code; all instructions following the asterisk should be interpreted as a comment:

```
G21           * English units
M03 S5000     * Spindle on clockwise at 5000 RPM
G92 X1.0 Y0.0 Z5.0 A0.0 B90.0 * Defines current machine state as coordinate reference
G0 Z2.5       * Move at Rapid Feed 2.5 inches down in Z, to Z=2.5
G1 X2.0 Y1.0 F5.0 * Linearly interpolate to X=2, Y=1 at a rate of 5 IPM
X3.0 Y2.0 F10.0 * Linearly interpolate to X=3, Y=2 at a rate of 10 IPM
X5.0 B80.0    * Linearly interpolate to X=5, B=80.0 (degrees) at a rate of 10 IPM
M02           * End program
```

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
There are several points that should be noted in this example. The first is that many commands are modal; that is, once they are defined in a given block, they remain constant in that setting until they are explicitly changed. Thus once a linear interpolation (G1) command, axis position, or feedrate is set, it does not have to be entered on each subsequent block until its state changes. If desired, though, a new feed may be entered for each tool move, a feature effectively used in this research. Also note that the part is generally not milled in some absolute coordinate system; the machine coordinate system must be defined based on a specific part setup configuration for each job. While this is necessary to simplify the task of manufacturing many different parts on one machine, it also contributes to the problems caused by the rotary motion as it contributes to the "unknown radius" problem in the controllers. Furthermore, not all controllers interpret commands identically. For example, a Fadal CNC controller interprets a G92 as a "Define Home Position", while a Boston Digital controller interprets it as "Start of Canned Cycle".

1.3 Milling Efficiency

Manufacturing via NC milling is a high overhead operation. The equipment is relatively expensive, with "inexpensive" NC mills costing on the order of $50,000 for a three axis machine, and two to three times that for an "inexpensive" 5-axis machine. Truly functional machines that are substantial enough to survive production work, set up and ready to mill, can cost appreciably more than that. Additionally, NC mills require a well-equipped shop, skilled operators and part programmers, consumables (stock, cutting tools, fluids), and maintenance. NC milling is also fairly commonplace, with many "job shops" bidding to manufacture parts for larger corporations. This not only increases competitiveness among the jobs shops, but also puts increased pressure on "in-house" milling operations to remain efficient. Jobs are often bid on in a 'per-part' or 'per-lot' basis, and so production efficiency greatly affects a shops financial performance in this high-overhead, competitive industry.

Efficiency improvements in the milling operation can be gained in two areas: the efficiency of the milling strategy, and the machining feedrate. Machining strategy refers primarily to toolpath generation and process definition. Efficiencies here can be found in cutter selection, toolpath optimization (how "well-
packed" the toolpaths are to minimize the passes required to produce a desired shape), and the definition of different operations (roughing, semi-finish, finish milling). There has been and continues to be much research in the area of 3 and 5-axis toolpath generation and process definition, and several robust commercial packages are currently available that allow users to generate accurate, efficient toolpaths from CAD data. The machining feedrate defines the velocities at which the machines operate. Efficiencies can be improved via feedrates through the use of the maximum velocities that can safely produce a part. There is also an interrelationship between the choice of machining strategy and the allowable feedrate, and a truly optimal CAM based process planner would consider this when defining toolpaths.

While feedrates can be varied over some range with acceptable results, the spindle RPM used to mill a part should remain fairly constant. The spindle speeds that are supplied in machining reference guides, such as the METCUT tables, are the results of years of empirical and theoretical research. These speeds are provided as a function of stock material, and are typically provided in units of "Surface Feed per Minute"; therefore the actual resultant spindle RPM is a function of the cutter radius. The surface velocity of the cutting flutes is important as it affects both the strain rate and friction present in the milling process, both of which produce large amounts of heat. The local temperatures on the cutting flutes can reach hundreds and even thousands of degrees Celsius [W89]. If allowed to become too high for extended periods, the heat can degrade the crystalline structure of the cutting tool material, leading to premature tool wear and even failure. Therefore the recommended spindle speeds, provided both for differing stock materials and cutter materials (due to the differing critical temperatures), should usually be used. This leaves feedrate as the primary variable for milling rate optimization.

The task of defining feedrates is not always easy. Even in traditional 3-axis milling, where the input feed value equals $V_{net}$, feed selection is complicated by the continually changing cut geometry. Skilled machinists, however, can identify the worst-case cutting conditions and set the feeds accordingly, based either on experience or tabular data, such as those supplied by METCUT. In 3-axis milling, these feed values may then be used with confidence, as $V_{net}$ remains constant and equal to the feed value specified to permit safe operation for the worst-case conditions. However, efficiency is sacrificed since the actual cutting conditions vary continually. The cutter could safely move faster in areas of less severe conditions.
(e.g. reduced axial depth of cut or partial immersion vs. full slot cuts), but it is not practical for a machinist to manually define variations in feedrate for each block of position data in a G-code.

The feed selection process becomes more complicated for 5-axis milling due to the addition of the rotational axes. Also, the geometries cut are typically more complicated, making accurate evaluation of the cutting conditions more difficult. Finally, many 5-axis mill controllers do not consider rotational information when calculating axis velocities, resulting in a widely varying cutter to stock relative velocity \( V_{nc} \), even for a constant input feedrate value. This greatly increases uncertainty in the identification of worst case conditions required for safe feed selection, as it is no longer a function of geometry alone.

The large volume of information that must be analyzed, combined with the ambiguities of identifying cutting conditions by inspection, make it prohibitively difficult to define optimized feedrates using traditional manual methods. In order to optimize feeds in a practical manner, an analytical method of feedrate definition robust enough for general application (i.e. reliable results are obtained regardless of cutter type, machine type, stock geometry, or toolpath geometry) must be developed, as will allow for numerical solution. Computer simulation makes it possible to analyze the cutting conditions present during for each block of position data in a part program, and set the feed value accordingly. To ensure safe and reasonable operation, additional constraints need to be defined; these include maximum and/or minimum allowable feed values, limits on the rate of feed change, and maximum chip thickness values. The chip thickness values are critical since they are directly related to temperature induced tool wear. Using a software based approach, the feeds may be fully optimized for an entire part program, maximizing the efficiency gains provided by optimized feedrates while maintaining safe milling procedures.

Another important cost is the "engineering hours" required to define optimized feeds. Although the feed optimization calculations are performed on a computer, the time required to define the appropriate input data must be considered. Ideally, the definition process should be as fast or faster than the traditional manual methods. Therefore, to maximize the overall time savings, the feed generation software should be simple to use, require a minimum of inputs, and be robust in operation. By robust it is meant that the software both be fault tolerant and stable, as well as producing reliable results so that little or no repeat
14 iterations are required to define the final feed values. Automatic feedrate selection provides a powerful tool for tuning and optimizing feed values selection for use in the final production part program.

1.4 Automatic Optimized Feedrate Estimation

A model that relates feedrate to cutting conditions is necessary for the definition of optimized feedrates using a computer. The most basic of these models employ "look-up tables", based on the tabular feeds/speeds information traditionally used for manual feedrate selection. This can provide the most conservative approach (assuming the worst case conditions are appropriately identified), and offer the smallest gain in efficiency as it would allow for very limited variation in the feeds, generally only differentiating between a fixed depth partial immersion and full slot cuts. While this may be sufficient for very simple prismatic parts with fairly uniform cutting conditions, it would not suffice for sculptured surface machining. While rudimentary, this concept is a small improvement over purely manual methods, and is commercially implemented in at least one CAM package, FeatureCam by Engineering Geometry Systems [JFHE00][F00].

For sculptured surface machining, where the cutting conditions vary continually, an analytical model that relates feedrate to current cutting conditions is desired. Even simple prismatic milling would benefit from this, as cusps left by previous operations, corners in the stock, and other areas of variable geometry would be considered. Over the course of the past three decades, several different types of analytical models have been developed, although primarily for research use. The most basic is the volumetric approach, where the feeds are set based on the volumetric material removal rate [E93][G87][NDW89][W88]. As the power consumed during a milling operation is a direct function of the torque loads placed on the cutter by tangential cutting forces, it follows that maintaining a constant power requirement leads to constant tangential forces. Thus the goal is to maintain a constant material removal rate. Calculating the volumetric material removal rate for each tool move for some initial feed value, and then multiplying both sides of the equation by a ratio that results in the desired removal rate, achieves this. This technique provides for more variability than the look up table approach, but is still fairly rudimentary.
Two commercially available applications of the volumetric approach at feedrate selection currently exist. CG-Tech has commercially implemented a feed selection program called Optipath, which adjusts feedrates as a function of material removal rate via a set of user-defined ratios [V00]. The reported success of this software demonstrates the desire in industry for automated process definition and their readiness to accept and implement the optimized feed concept. MasterCam has also recently developed automatic feedrate generation software called HighFeed [M00], also based on the volumetric approach. However, while the volumetric approach is a marked advance over manual or look-up table based feedrate selection, it does not adequately address all of the problems found in automatic optimized feedrate selection.

The primary drawback to the volumetric approach is predicts only average force values, rather than reflecting the continuously varying nature of the forces during cutter rotation. The peak values, which can be the most damaging if unchecked, typically are far greater than the average values. The force required to shear a chip from the stock is a function of the instantaneous chip thickness. To estimate the peak force values, the cutter type and direction of travel, the region of the cutter that is intersecting the stock material, and the current cutter rotation angle at any given instant during the cutter rotation must be considered, as they all affect the instantaneous chip thickness (as shown by Equation (1.2)). The volumetric model would set the same feed whether the cutter was contacting a large area of stock towards the radial edge of the cutter, or a smaller contact area in the radial center of the cutter, ignoring the variations in instantaneous chip thickness for these two cases. Although the average force values would be similar for both cases, the instantaneous values would be much higher in the latter case due to the greater chip thickness values. To remain conservative this approach must therefore set the feed assuming the higher peak forces.

This year CG-Tech has added a new feature to their product that also considers chip thinning by trying to maintain a constant maximum chip thickness during the milling operation. This marks another advance in commercial optimized feed selection software as the chip thinning concept does consider cut geometry, but it does not fully meet the eventual requirements of such a system. Ultimately, the true limiting factor on feedrate selection is excessive cutting forces, which leads to cutter and part deflection and damage. While the inclusion of chip thinning supports the theory that the instantaneous cutting force must be considered, it does not actually predict it. The force values can vary widely even for a constant maximum
chip thickness. Milling force is a non-linear function of chip thickness, tightly coupled to the type of stock material, the basic flute geometry on the cutting tool, and the total geometry of the contact region between the cutter and stock. All of this must be considered for truly optimized force-based feedrate selection.

Non-force based automatic feedrate generation schemes are not extensible to more advanced applications. Their use is limited to rough estimation of safe feedrate values, with little or no ability to advance with industrial requirements as companies acquire feedrate technology, implement it, and eventually desire more capabilities. Automatic feedrate generation based on force estimation marks only the beginning of a series of advancements. Once force based feedrate selection has been validated for accuracy, it may be applied in the modeling of cutter deflection for tolerance based feedrate selection. It may also be used in the modeling of machine dynamics to prevent chatter and other undesirable conditions. Finally, it can aid in predicting machine maintenance through the prediction of cyclic loading and other contributions that lead to bearing failure.

1.5 Milling Force Estimation and Automatic Optimized Feedrate Estimation

A milling force estimation model that relates the net instantaneous milling forces to feedrates must be employed to best estimate optimized feeds. Such a model must consider feedrate, spindle speed, cut geometry, cutting tool geometry, and material constants (to distinguish between different stock materials). These models are referred to as mechanistic models, as they use the mechanics of the milling process to estimate milling forces. Use of a mechanistic model in feedrate selection offers direct control over the true limiting factor (force).

There have been numerous attempts to develop an automatic feedrate generation scheme that use mechanistic models. In [TTIS89], Takata implemented a Z-buffer geometric model integrated with the mechanistic model of Devor and Kline for the investigation of cutting forces and cutter deflection. This work eventually led to a 2-1/2 axis process planner [T93]. Around the same time Fussell, Jerard, and Ersoy were researching a similar approach at automatic feedrate selection for 3-axis milling [FEJ92], [E93], which was later expanded by Durdag for true 3-axis capability [D93]. In Hemmett [H94], the combined the Z-
buffer/mechanistic model approach at automatic feedrate generation was proven feasible for application in 5-axis milling as well. More recently, there have been attempts to replace the Z-buffer geometric model with a solid model of the cutter and stock to provide the data required for mechanistic modeling. This approach has been used in Spense [SA94] and Bailey et. al. [BJSE96] for 2-1/2 axis process planning systems, and was expanded in Mounayri [MSE98] to include full 3-axis operation. This process has even been taken one step further [D93,W98], where the feeds are set to control the cutter deflection that is a result of the milling forces. Alternate approaches include varying the cutter path to directions that maintain minimum milling forces, potentially allowing for further reduced milling times as compared to methods that only vary the feeds on existing toolpaths [LM97].

1.6 Integrated Modeling Approach to Automatic Optimized Feedrate Estimation

One of the most useful mechanistic models is that proposed by Devor and Kline, originally developed to estimate cutting force vectors acting on a flat end cutter [KDL82][KD83]. This model, presented in Chapter 2, estimates milling force vectors in \( X_{CT} \) and \( Y_{CT} \), where \( X_{CT} \) and \( Y_{CT} \) form a right handed Cartesian coordinate system orthogonal to the cutter axis, and \( X_{CT} \) lies in the direction of travel (Figure 1.8). This discrete mechanistic model has been successfully applied in an inverse manner to generate feedrates necessary to maintain a constant cutting force magnitude during 3 and 5-axis sculptured surface machining [FHJ94][FJD95][H94]. The allowable force magnitude may be selected based on a number of criteria, including the prevention of cutter breakage, maintaining part tolerance by limiting cutter deflection, or meeting a machine constraint such as a maximum torque value allowed on a rotary table.

There are two approaches at solving for the feed value required to maintain a given force value. The first approach is a 'true' inverse method, where the mechanistic model is re-arranged for solution of the feed variable [E93]. As the relationship between feed and force is non-linear, a numerical approach such as the Newton-Raphson method must be implemented for solution of this equation. The primary drawback to this approach is that it only allows for independent solution of one force component, generally the \( Y_{CT} \) force as it tends to be most significant. As this approach does not allow for solution of a net \( X_{CT},Y_{CT} \) force, a
Figure 1.8: The principal directions of the milling force vectors calculated by the mechanistic model.

Figure 1.9: Flowchart overview of the integrated modeling approach at automatic feedrate selection.
second approach was developed that requires iterative solution of the mechanistic force models to solve for both $X_{CT}$ and $Y_{CT}$ forces, with the feed value being adjusted during each iteration until the desired net force magnitude is achieved [H94]. This approach, developed during earlier research, is implemented in this application as it provides the most milling force information with little additional computational overhead.

The data needed for mechanistic milling force estimation requires the use of additional supporting models. A geometric model of the stock and cutting tool, updated each tool move, is required to maintain a current representation of the in-process stock geometry. This is necessary for the calculation of cut geometry parameters that describe the contact area between the cutter and stock. A CNC machine model is also required to simulate the machine kinematics and controller behavior, allowing input feedrate commands to be translated into relative velocities between the cutting tool and stock.

The combined models of the software system are tied together in an integrated modeling approach, where the overall system is peripherally integrated such that the only links between components are for passing data. The modularity this provides allows for individual optimization of each component, and simplifies component replacement if a better method is developed. This also facilitates the application of the system to a variety of different cutting tool geometries and CNC machine configurations.

This integrated software scheme is shown in Figure 1.9. For each tool move contained in a G-Code file, the region of intersection between the cutter and stock is calculated by a geometric model, which is updated accordingly to represent material removal. If any stock intersection is found for the current tool move, geometric parameters that define the axial and radial depth of cut are then calculated from the intersection data for use in the mechanistic model. The CNC machine model is then used to calculate the relative velocity $V_{net}$ of the cutter past the stock. Once the geometric parameters and relative velocity have been found, the mechanistic model is invoked to estimate the feedrate necessary to maintain the desired force. If the force for the current input feed is not within the desired range, the model iterates on the feed value until the estimated force is within the desired range, or some other exit condition occurs. Other exit conditions include machine constraints, user imposed feed limits, or maximum allowable chip thickness values. For each move, the maximum chip thickness value calculated during force estimation is stored and checked against the allowable maximum during each iteration. Note that while the feed value is iterated, the
cut geometry parameters remain unchanged and only the relative velocity varies. At the termination of the feed calculation iterations, the current feed value is written out with the current G-code positional data block to an updated version of the part file. This procedure is repeated for each positional data block in the G-Code file, resulting in a set of near optimal feedrates for a given machining operation.

1.7 Summary of Advancements

The primary goal of this research is the development of an automatic feedrate selection system capable of industrial implementation for 3 and 5-axis sculptured surface machining. This requires a balance of accuracy, efficiency, ease of use, and robustness. It also requires a versatile solution capable of feed generation for a variety of machine and cutter types. Finally, this research should also provide a platform for future innovation, so that this technology can continue to develop with the demands of industry.

The advancements resulting from this research helped to meet these goals. This includes the development of a generalized mechanistic model that includes the effects of variable cutter geometry, which required explicit inclusion of the helix angle, cutter surface normal, and flute segment length in the mechanistic model. The existing chip thickness model was also modified into a form that was convenient and efficient to implement with this generalized model, and allows for the calculation of an average chip thickness and inclusion of cutter runout. The support of generalized cutter geometry also required the development of a generalized discrete representation of the cutting tool, as well as techniques to efficiently obtain necessary cutter data. The geometric model was also enhanced so that a contact area could be calculated on the cutter ball. The variable surface normal and cutter radius, combined with the ability for material removal to occur on the 'back' of the cutter ball, are now included in the contact area calculations. Validation of the ball end model was performed through numerous simulations and experimental cuts.

Explicit inclusion of the CNC machine model was also introduced to compensate for the rotary motion present in 5-axis milling. Rotary motion results in differences between the input feedrate and the cutter to stock relative velocity. The model calculates the correct velocity relationships regardless of NC machine type (assuming the correct model has been implemented), allowing for versatile application of the
automatic feedrate generation software. In this research, five different 5-axis NC machine configurations were modeled. These include a table-on-table arrangement where a 2-axis rotary tilt table is mounted on a 3-axis mill, and Boston Digital 405 and 505 5-axis mills, both in 'A axis' and 'C axis' configurations. Additionally, the CNC machine model was implemented as a standalone module to provide basic relative velocity control and also to validate the use of continually varying input feed commands.

There were also numerous efficiency developments made to the algorithms to reduce computation time without sacrificing accuracy. These are distributed throughout the system, and include the methods used to simulate cutter rotation, calculate relative velocities, calculate the cut geometry parameters and iterate to the required feed value.

Finally, identification of maximum allowable chip thickness as a requirement when optimizing feeds contributed greatly towards the practicality of the system. Limiting only the milling force is sufficient when heavier, consistent amounts of material are being removed. However, this situation is not typical in sculptured surface machining, where the cut conditions vary continuously. During lighter cuts an unacceptable chip thickness results while attempting to maintain a constant force level, resulting in premature tool wear. Direct control of the maximum allowable chip thickness therefore protects the cutter from excessive wear during lighter cuts, while the force constraint prevents breakage during heavier cuts.

The primary advancement made by this research is the compilation of the above developments with existing technology to form a working software system capable of automatic optimized feedrate selection in an industrial setting. The advancements mentioned above were specifically developed to meet the requirements for such a system, and therefore meet the stated goals of this research.

1.8 Thesis Overview

This thesis contains ten chapters. These chapters describe some of the available methods for automatic optimized feedrate selection, as well as the methods that were developed to support this endeavor. In particular, they describe one complete methodology that was developed to perform this task robustly and efficiently.
Chapter 1 provides the necessary background required for the understanding of the fundamentals of this research. It presents some of the basic terminology and relationships, as well as describing the need for this research and some of the reasons that the basic structure of the software system exists as it does. This provides the necessary foundation for chapters 2 through 7, where each of the components required for automatic optimized feed selection are presented in more detail.

Chapter 2 describes the mechanistic modeling process, and presents the model developed during this research. This includes techniques developed to improve the versatility of the model, making the overall system more practical for commercial implementation. Chapter 3 covers the discrete cutter representation and implementation details of mechanistic modeling and feedrate generation. Chapter 4 describes the geometric modeling techniques used to represent the in-process stock, while Chapter 5 describes the geometric modeling techniques required to model the cutting tool for material removal simulation. Chapter 6 covers the techniques used to calculate cut geometry data required for mechanistic modeling using data provided by the geometric model. Chapter 7 describes the NC machine model, and its role in calculating the cutter to stock net relative velocity, required for accurate chip thickness estimations. Chapter 8 describes the techniques developed to obtain the mechanistic material constants. These constants are found empirically, and the procedures and methods developed to calculate the required constants from experimental data are described. Chapter 9 presents the experimental methods used to validate the software, as well as some of the experimental results. Chapter 10 offers some conclusions that may be drawn as a result of the work performed, and also makes suggestions for future work.

Although the topic of this dissertation is automatic feedrate selection, note that force based feedrate selection and milling force estimation are inexorably linked. As the calculation of force constrained feed values merely consists of an iterative set of force estimations, the crux of the feed selection problem is force estimation itself. Therefore the bulk of the effort that went into this research was in force estimation techniques, including the calculation of the geometric descriptions necessary for accurate force estimation. The methods presented here therefore also apply to pure force estimation, in addition to the automatic feedrate generation possibilities they provide.
CHAPTER 2

MECHANISTIC MODELING FOR MILLING FORCE ESTIMATION AND FEEDRATE SELECTION

2.1 Chapter Introduction

The forces acting on a cutting tool during milling (the 'cutting forces') play a critical role in the milling process. These forces cause cyclic loading of the part and cutter as the cutter rotates and shears chips from the stock material. This loading results in deflections that can affect part tolerance and surface finish, and can also lead to part damage or cutter breakage. Therefore it is desirable to understand the processes that produce cutting forces, so that they may be controlled duringfeedrate optimization, and so mechanistic models were developed to perform this operation.

Mechanistic models estimate milling forces using the mechanics of the milling process as a foundation. Most practical mechanistic models estimate cutting forces as a function of the chip shearing forces exerted by the cutting flutes. Therefore these models have the following two basic requirements:

1) material parameter(s) that relate material removal to milling force to for a given material, and

2) a means of estimating the amount and location of material that is being sheared by the cutter.

The material parameters are generally treated as constants for a given material, varying only with tool changes (due to the effects of tool coatings and flute geometry). These constants are typically empirically derived for a given material and cutting tool combination. This leaves as a primary variable in mechanistic force equations the amount and location of material that is being sheared by the cutter during a tool move.

Feedrate has a dominant affect on the amount of material sheared per flute during a given tool move. While spindle speed, cut geometry (which defines the location of material removal on the cutter), cutter geometry, and machine kinematics can all affect material removal, these values are treated as constant during a given tool move for various reasons that are explained later. This leaves feedrate as the sole variable. This means that the mechanistic model has the critical job of relating feedrate, the variable that is to be defined, to milling force, the variable that is to be controlled.
This chapter describes some of the existing mechanistic models, and in particular discusses the model implemented during this research. As the mechanistic model forms the crux of force-based feedrate estimation, an accurate model is desired. The model must be capable of providing peak force estimates, as this must be limited for safe operation (as opposed to an average force). A model that can estimate instantaneous force vectors is also desirable, as this will allow for future expansion to provide tolerance-based feed selection through cutter deflection simulation. However, these requirements must be balanced against robustness and practical considerations. The model must be stable under all conditions, with conservative error in the force estimates. For a truly functional system, ease of use (including the calculation of material constants), a reasonable cutter definition requiring only readily available information, and computational efficiency must also be considered.

There are two distinct types of mechanistic models. The first type simply estimates the average milling force experienced during a tool move. This is typically performed using the volumetric material removal rate, which is based on a lumped average of the material removed (sheared) by the cutting flutes. The second type estimates instantaneous milling forces at any point during a single cutter rotation (as mentioned above, the forces are cyclic and therefore the same force values are repeated for subsequent cutter rotations as long as the cutting parameters remain unchanged). The second form is more complex in that the force values must calculated for many rotational positions of the cutting tool for a given tool move in order to obtain a full force profile for that move.

2.2 Volumetric Based Force Models

The most primitive forms of force estimation rely on volumetric removal rate. The volumetric relation is based on the average power consumption equation, traditionally used in the estimation of required milling spindle power. This power relation is defined as:

\[ P = K Q \]  \hspace{1cm} (2.1)

where P is the power consumed (hp),

\[ K \]  is the Unit Power Consumption, a material constant (hp min / in³), and
Q is the volumetric removal rate (in³/min).

For a given tool move, the calculation of Q requires knowledge of the cutter feedrate for move time estimation, and also the use of a geometric model for estimation of the volume of material removed during that move. From this relation, the average tangential cutting force (force tangent to the cutter as seen looking down the cutter axis; this is the force that results in torque on the cutter) may be calculated as:

\[
F_{t\text{-avg}} = 33000 \frac{P}{V}
\]

(2.2)

where \( F_{t\text{-avg}} \) is the average tangential force (Lbs),

P is the power consumed (hp), and

V is the surface cutting velocity (ft/min).

While this relation is of limited use for practical application, it has been demonstrated [W87, ST91] that the average side force on the cutter, \( F_{y\text{-avg}} \), which acts perpendicular to the direction of travel, is simply:

\[
F_{y\text{-avg}} = 0.5 F_{t\text{-avg}}
\]

(2.3)

The primary benefit of the volumetric model is its simplicity, which makes it fast and reliable, and it has also been shown to be reasonably accurate. It is also practical to use, as no cutter flute geometry definition is required, and values for K are readily available in many Machining Handbooks. This model is widely used by industry for the estimation of feed drive power requirements for NC mills [E93], and as stated in Chapter 1, has been commercially applied in a very simple attempt at optimized feedrate selection.

However, while this mechanistic modeling approach is capable of estimating average tangential forces, its accuracy is highly dependent on the value of K, which has varies widely in value in the documented listings [E93]. Also, for automatic force-based feedrate estimation, the average side force does not provide sufficient information for safe feed selection. The peak cutting force at some instant during a cutter rotation can greatly exceed the average value, and the force contribution in the direction of travel, neglected by this model, can also be substantial. Both of these issues are compounded in the presence of
appreciable cutter runout, which the model does not address. It also does not provide sufficient information for accurate cutter deflection modeling. Although useful where gross approximations are sufficient, such as in the estimation of power requirements, this model lacks the sophistication required for optimal force-based feedrate selection.

2.3. Instantaneous Mechanistic Modeling

The mechanistic model estimates instantaneous cutting force vectors rather than average milling force, and can simulate the effects of cutter tilt, runout, variable cutter geometry, and other physical variables that can affect the forces. While instantaneous mechanistic modeling is more complex than the volumetric approach, the relative difference in performance can be nominal, as in both models much of the computational effort goes into geometric simulation.

Instantaneous forces are generally calculated at a finite set of angular positions during cutter rotation, although time based models also exist. The benefit of calculating instantaneous cutting forces is that the full range of cutting forces present for a given set of cutting conditions may be estimated. This requires the forces to be estimated at a number of discrete rotational positions for a given tool move, as the magnitude and direction of the force vectors can vary appreciably as different flute segments engage the stock over varying rotation angles, particularly in the presence of cutter runout. This is achieved by assuming that cutting conditions remain constant over a single tool move, as using the worst case cut geometry conditions experienced during the move to provide a conservative estimate. This allows the full range of forces experienced during a given tool move to be estimated with a single cutter rotation.

Many instantaneous mechanistic models (referred to from now on as simply 'mechanistic models') are based on the orthogonal cutting model [TM75][KDL82][SD86]. In this model, the actual net force required to shear a chip from the stock material, composed of normal (shearing) and frictional components, and may be estimated by a set of radial and tangential force vectors (Figure 2.1). As shown in Figure 2.1, the magnitude of these force components is a function of the uncut chip thickness 'h'. The flute locating angle, $\beta$, is required to transform the force components into common X and Y coordinates for summation of the contributions from all engaged flutes. Some more recent mechanistic models use an oblique milling
Figure 2.1: In the orthogonal cutting model, radial and tangential force components are the estimated values as a result of chip formation.

Figure 2.2: In discrete mechanistic models, cutting tools are represented as a set of discrete axial disc elements of thickness dZ.
model as the physical foundation, which includes a third force component parallel to the flute edge to model chip flow and its additional frictional components [BYK94][YA96].

Several basic operations must be performed to estimate the milling force vectors at some given rotational position during a given tool move. First, the cutter rotation simulation must be defined and controlled. Second, the portions of the cutting flutes that engage the stock material must be identified at each rotational position, for the given cut geometry supplied by the geometric model. Third, the chip thickness that occurs along these flute portions must be calculated to find the area of the material being sheared from the stock. Finally, the force vectors that arise as a result of this shearing must be calculated at a set of discrete rotation angles for identification of the maximum force.

To simplify the integral mechanistic equations originally developed by Tlustly [TM75], mechanistic models are often presented in a discrete form suitable for numerical solution. In this approach the cutting tool is modeled as a set of discrete axial discs, as shown in Figure 2.2. Material removal occurs only on the cutting flutes, and so these discs may be more accurately thought of as a set of $N_f$ flute segments at some discrete axial location on the cutter, where $N_f$ is the number of flutes on any given disc element. As may be seen in Figure 2.2, the individual flute segment locations are defined on each axial disc by a 'flute locating angle', $\beta$. The discrete cutter model simulates cutter rotation by indexing the value of $\beta$ through a set of discrete locations for all flute segments in the model.

In the discrete cutter representation, all cutter geometry values are assumed constant over each disc, with all variation occurring in discrete steps between discs. This provides a means of representing, as a function of axial location, cutter specific parameters required for mechanistic modeling, such as variations in helix angle, cutter surface normal, number of flutes, and/or cutter radius (see Figure 2.3). Even the region of contact between the cutter and stock may be defined relative to the axial discs, as shown in Figure 2.3. This simplifies identification of the flute segments engaged in the stock material at a given angle, as it must only be determined if the current flute segment on a single axial disc lies within the bounds of stock contact. The net region of stock contact over all discs is defined as the 'contact area'.

At each rotational position, the normal and frictional forces developed on the tool lead to radial and tangential force components. These are estimated as a function of chip thickness, as shown in Figure 2.1. In some of the more sophisticated models, additional flank components are also explicitly estimated.
Figure 2.3: The discrete cutter concept aids in modeling cutter geometry (variable helix angle, surface normal, radius, etc.) and simplifies representation of the region of intersection between the cutting tool and stock (the 'contact area').

Figure 2.4: Definition of the Cutting Tool coordinate system.
These components are also included in the less sophisticated models, lumped with the pure milling forces via the empirical material constants. The individual milling force components may then be summed over all engaged flute segments to calculate a net force; alternately they may be applied as a distributed load for deflection modeling. Note that the as-calculated raw force components often lie in principal directions that are not convenient for summation or other practical application (e.g. deflection modeling), necessitating the use of some convenient global coordinate system.

2.4 Coordinate System Definition for Discrete Mechanistic Modeling

The mechanistic models implemented in this research calculate milling forces in vector form. For practical application, these vectors are defined in ‘Cutting Tool coordinates’, which consist of linked 3D Cartesian and polar coordinate systems. To provide a uniform basis regardless of CNC machine motion, these coordinate systems are defined relative to the cutter orientation and its direction of travel. Figure 2.4 presents the definition of these coordinate systems.

In the Cartesian system, the positive Z axis extends from the base of the cutter upwards along its rotational axis. The positive Y axis is defined as the cross product of Z into the direction of travel vector, D, (the direction vector associated with the net relative velocity), and the positive X axis is defined as the cross product of Y into Z. Note that the X axis lies along the direction of travel only when the cutter is moving orthogonal to its rotational axis. This system is referred to as the Cutting Tool Cartesian system in this research, indicated using $X_{CT}, Y_{CT}, Z_{CT}$.

The raw milling force components estimated by the mechanistic models extend radial and tangential directions relative to the cutter surface, as shown in Figure 2.1. These forces must be transformed into the Cutting Tool Cartesian system for practical application. The polar coordinate system provides a basis for identifying the flute locating angles that are required for performing these transformations. It also provides a convenient method for the estimation of chip thickness values, and a basis for cutter rotation simulation. In the polar system, the radius values are assumed constant and equal to the radius of the current axial disc, and the angular origin lies along $-Y_{CT}$. The angular degree of freedom is then defined using the right hand rule about $Z_{CT}$, as indicated by the flute locating angle $\beta$ shown in Figures 2.2 and 2.4.
2.5 Cutter Rotation Simulation

Cutter rotation simulation is defined to take place in cutting tool polar coordinates. To simulate cutter rotation, the cutting tool is first defined at some initial reference angle $\alpha$. The current angular location $\theta$ is then calculated relative to $\alpha$ by indexing an amount $d\theta$:

$$\theta(j) = \alpha + (j) (d\theta)$$  \hspace{1cm} (2.4)

where $\theta(j)$ is the current angular position (rad),
$\alpha$ is the initial reference angle (rad),
$j$ is an index to the current rotation angle, and
$d\theta$ is the angular increment amount (rad).

While the cutter rotation angle $\theta(j)$ may be calculated using Equation (2.4), the value of interest for mechanistic modeling is the flute locating angle $\beta$. In addition to the current cutter rotation angle, $\beta$ is a function of the current flute, cutter helix angle, and the axial location of interest:

$$\beta(i,j,k) = \theta(j) + (k)(\gamma) + \tau (i)$$  \hspace{1cm} (2.5)

where $\beta(i,j,k)$ is the angular position of the current flute (rad),
$\theta(j)$ is the current cutter rotational position reference angle (rad),
$k$ is an index indicating the current flute,
$\gamma$ is the angular separation between adjacent flutes (rad), and
$\tau (i)$ is an angular offset due to the cutter helix angle and axial location (rad).

Figure 2.2 shows an example of a specific angle $\beta$. The relation for $\gamma$, the angular separation between adjacent flutes is defined as:

$$\gamma = 2\pi/N_f$$  \hspace{1cm} (2.6)

where $N_f$ is the number of flutes at the current axial location.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The geometric relation that defines the angular offset that is a function of cutter helix angle is traditionally defined based on the discrete cutter model as provided by Kline and DeVor in [KDL81]:

\[ r(i) = (i)dZ \frac{\tan(\alpha_{\text{helix}})}{R} \]  

(2.7)

where \( i \) is the index to the current axial disc,

\( dZ \) is the axial disc thickness (in),

\( \alpha_{\text{helix}} \) is the cutter helix angle (rad), and

\( R \) is the cutter radius (in).

Equation (2.7) is sufficient for models of constant radius, constant helix angles, and constant axial disc thickness, which limits its usage to standard flat end mills. To include the variable radius and helix angles required for variable geometry cutter models (e.g. ball end), this relation must be expanded. This is performed through summation of the angular increments in flute locating angle due to helix angle across all axially lower discs to arrive at the net angular offset for the current disc:

\[ r(i) = \sum_{z=0}^{i} (dz(z))\frac{\tan(\alpha_{\text{helix}}(z))}{R(z)} \]  

(2.8)

where \( i \) is the absolute index to the current axial disc element,

\( z \) is the summation index to the current axial disc inside the summation,

\( dZ(z) \) is the thickness of the current axial disc inside the summation (in),

\( \alpha_{\text{helix}}(z) \) is the helix angle of the current axial disc inside the summation (rad), and

\( R(z) \) is the radius of the current axial disc inside the summation (in.).

2.6 The Contact Area

The axial discretization of the cutting tool provides a convenient means for specifying the contact area, which defines the region on the cutter surface that intersects the stock during a given tool move. First it must be determined if a given axial disc engages the stock material. Of the discs found to contact the
stock, the bounds of stock contact are then defined for each disc. This is performed through the definition of a pair of (or sets of pairs of) 'entrance' and 'exit' angles, defined in Cutting Tool Polar coordinates (Figure 2.5). An entrance angle indicates where a given flute, traveling in the direction of cutter rotation, enters the contact region, and an exit angle denotes where it leaves; all angular positions in between are assumed to contact the stock. During the cutter rotation simulation, Equations 2.4 and 2.5 are used to calculate the angular position of a given flute segment on a given axial disc. If this angular position lies in the region bounded by the entrance and exit angles, then that flute segment contributes to the milling force calculations. In order to calculate the contact area for any tool move during a given NC job, a geometric model of the stock is required. This provides both a medium for the calculation of the contact area, and keeps a current record of the continually changing stock geometry. The methods used for geometric modeling are presented in Chapters 4-6, with definition of the contact area presented in Chapter 6.

2.7 Chip Thickness Calculations

2.7.1 Chip Thickness

An essential factor in milling force prediction is accurate estimation of 'chip thickness'. Chip thickness refers to the width of material removed by a cutting flute from the stock material at some discrete rotational position, defined along the direction of the cutter surface normal vector (see Figure 1.1). Chip thickness is a function of cutter surface normal at the point of contact, and the rate and direction of travel of the cutting tool. The rate of travel of the cutter is included in the feed per tooth value.

The directional of travel component may be stated explicitly, as in Equation (1.2), but a more useful form is to include it via a flute locating angle $\beta$, which is defined relative to the direction of travel.

2.7.2 The Classical Chip Thickness Model

One of the most commonly used chip thickness approximation techniques used when mechanistic modeling [TM75][KDL82][KD82][BYK94] is the sinusoidal approach proposed by Martellotti [M45]:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 2.5: The contact region is defined on the cutting tool via the definition of entrance and exit angles in Cutting Tool polar coordinates for each individual axial disc engaged in the stock.
\[ h = f \sin (\beta) \]  
\hspace{1cm} (2.9)

where \( h \) is the chip thickness (in),
\( f \) is the feed-per-tooth value component that lies orthogonal to the cutter axis (in/tooth)
\( \beta \) is the cutter rotation angle of the current flute (rad).

This approximation works well for chip thickness estimation with a flat end cutter during 3-axis milling. However, as generalized 3 and 5-axis sculptured surface milling with ball or flat end cutters is the focus of this research, Equation (2.9) does not fulfill the current requirements. This is because Equation (2.9) assumes the cutter surface normal lies orthogonal to the cutter axis, which is true only for flat end mills.

### 2.7.3 The Generalized Chip Thickness Model

As introduced in Chapter 1, the generalized chip thickness relation that is used in this research multiplies the dot product between the direction of travel and cutter surface normal vectors (at the region of interest on the cutter) with the net feed-per-tooth value. This equation is repeated here:

\[ h = f_{\text{net}} (n_c \cdot D) \]  
\hspace{1cm} (1.2, repeated)

where \( h \) is the chip thickness (in),
\( f_{\text{net}} \) is the net feed-per-tooth value (in/tooth)
\( n_c \) is the outward surface normal direction of the cutting tool, and
\( D \) is the cutting tool's direction of travel unit vector.

This relation, presented in [CJ98], calculates chip thickness as the component of the net feed-per-tooth value that lies in the direction of the cutter surface normal at the location of interest. The generalized form of this relation makes it suitable for application in this research. However, while Equation (1.2) provides an excellent theoretical description of the ideal chip thickness value, it is not convenient to implement in a manner that directly suits the needs of discrete mechanistic modeling.

One method of defining Equation (1.2) suitable for discrete application, and as a function of the known value of \( \beta \), is to divide the direction of travel (\( D \)) and cutter surface normal (\( n_c \)) vectors into 'axial'
and 'radial' components. Axial components are defined to lie along the $Z_{CT}$ direction, while the radial components are defined to lie along $X_{CT}$. This allows the chip thickness relation of (1.2) to be defined as:

$$ h = f_{\text{net}}(D_R)(n_R)\sin(\beta) + (D_A)(n_A) $$

(2.10)

This equation represents a discrete form of Equation (1.2), as the parameters $f_{\text{net}}, D_R, D_A, n_R$ and $n_A$ may be stored as discrete values for each axial disc. This allows for convenient implementation of Equation (2.10) during the discrete mechanistic modeling process. Equation (2.10) also provides a means of conveniently calculating an average chip thickness value as an integral of $\beta$ using Equation (2.13), as discussed in section 2.8.2.

Figure 2.6 shows the results of this equation as a color map over the region of intersection on the surface of a cutting tool, for three different cutting conditions. In this figure, the images are read right to left, with the cutter on the right indicating the axial component of travel, and the remaining images showing side and front views of the contact region respectively. Relative chip thickness values are color indexed, with red indicating thicker chips and blue indicating thinner. The actual magnitude of the chip thickness distribution is a function of the current $f_{\text{net}}$ value.

### 2.7.4 Feed-Per-Tooth

The feed-per-tooth value is critical in any chip thickness calculation. This value indicates the linear distance the cutting tool has advanced during one tooth-passing cycle, and thus indicates the amount of material that may be sheared by a subsequent flute. The equation used in this research to calculate feed-per-tooth was originally defined in Chapter 1, but is repeated:

$$ f_{\text{net}}(i) = \frac{V_{\text{net}}(i)}{(\Omega \cdot N_{\ell}(i))} $$

(1.1 repeated)

where

- $f_{\text{net}}(i)$ is the net feed per tooth value for the current axial disc (in/tooth),
- $V_{\text{net}}(i)$ is the current disc's net cutter to stock linear relative velocity (in/min)
- $N_{\ell}(i)$ is the number of flutes (or inserts) on the current disc (tooth), and
- $\Omega$ is the spindle speed (RPM).
Figure 2.6: This figure shows the contact area and relative chip thickness values for the partial immersion cut shown at the top of the page. The images are read left to right, with the cutter on the left indicating the axial component of travel, and the remaining images showing front and side views of the contact region respectively. Relative chip thickness values are color indexed, with red indicating thicker chips and blue indicating thinner. The actual magnitude of the chip thickness distribution is a function of the current $f_{\text{cut}}$ value.
\( V_{\text{net}} \) represents the magnitude of the net relative velocity vector between the cutting tool and the stock material, \( V_{\text{net}} \). During 3-axis linear interpolation the value of \( V_{\text{net}} \) is generally equal to the input feedrate value. However, in the generalized case \( V_{\text{net}} \) is also a function of NC machine kinematics and controller behavior, and in this situation it can differ greatly in magnitude from the input feedrate command. These variations can be the result of rotational velocities in 5-axis milling, or of the controllers interpretation of a given command, such as G02/G03 (circular interpolation) or G0 (rapid traverse). As accurate chip thickness estimation is critical to the mechanistic model, a means of accurately calculating \( V_{\text{net}} \) is necessary. Therefore, an NC machine model that simulates the kinematics and controller behavior of a given machine is required for the calculation of \( V_{\text{net}} \) for a particular tool move.

### 2.7.5 Trochoidal Chip Thickness Models

The chip thickness calculation described by Equation (2.10) is only an approximation. The physical geometry of the surface that is cut by a single flute is actually trochoidal in nature, while Equation (2.10) assumes a purely circular surface in the \( X_{\text{CT}}, Y_{\text{CT}} \) plane, as evidenced by the sine term used in this plane. By trochoidal it is meant that the true path traced by a given cutter flute falls under the cycloidal family of curves, rather than circular, as a result of the cutting tools forward translation during rotation. Essentially Equation (2.10) assumes no forward motion occurs simultaneously with chip formation. Inclusion of the trochoidal effects results in a non-zero chip thickness at the \( \beta = 0^\circ \) and \( 180^\circ \) positions even when no axial motion is present. It also results in a maximum chip thickness region that is shifted to some angle greater than \( \beta = 90^\circ \), unlike the results provided by Equation (2.10), and the peak chip thickness in this region is greater than that found by Equation (2.10) at \( \beta = 90^\circ \) (note that the values provided exactly at \( \beta = 90^\circ \) are identical in both cases).

Methods have been developed to calculate chip thickness values based on the trochoidal surface model. In a trochoidal approach, the chip thickness is defined as the difference between the surface generated by the previous flute and that being created by the current flute [MA91]. For a relatively small feed-per-tooth value relative to the cutter radius, the error due to the approximation is minimal and Equation (2.10) is sufficient, as widely published for the case of a flat end cutter [KDL82][SD86][MA91].
On a ball-end cutter, as implemented in this research, R decreases to zero at the bottom of the cutting tool, and so to see what effects a trochoidal chip model would have for a ball end cutter a trochoidal model was implemented in this research. This is presented in Appendix A. As shown there, the trochoidal approach was found to offer no significant advantages even for the ball end cutter. This is due to the fact that the vast majority of the material removal occurs in areas where the $R \gg f$ condition applies as a result of chip thinning on the bottom of the cutter ball. Therefore the traditional, sinusoidal based chip thickness relation of (2.10) is retained in this research.

2.7.6 Cutter Runout

In addition to the cutter geometry effects included in Equation (2.10), there are additional physical considerations that can affect chip thickness. The most significant of these is cutter runout, or a physical misalignment of the cutter’s axial center relative to the axis of rotation (Figure 2.7), resulting in an eccentricity in the cutter rotation. Runout can not be avoided, only minimized, as manufacturing tolerances on the cutting tools and holders always results in some misalignment. The major effect of runout is that it causes the radius distance from the center of rotation to the individual flutes to vary from the ideal, constant, cutter radius value. This leads to variations in chip thickness for the different flutes, as the different radius values remove differing amounts of material. This in turn varies the force values. The flutes on the side of the cutter where the runout occurs experience smaller force values relative to no runout, and those on the opposite side experience larger force values relative to no runout. This increases both the peak force value as well as the range of the forces present during a single cutter rotation. Although the magnitude of the runout is typically small, on the order of $10^{-3}$ inches, its effects can be very noticeable, as this is typically the same order of magnitude for chip thickness values.

There exist several methods to include runout into the chip thickness calculations, depending on the chip thickness calculation approach taken. For trochoidal or other 'exact' surface-subtraction chip thickness calculation methods, there is an excellent approach presented in [MA91]. For approximate methods, such as those used in this research, the method suggested by Devor and Kline in [KD83] provides excellent results. This approach defines runout in terms relative to the Cutting Tool polar coordinates, as
Figure 2.7: Cutter runout is a misalignment between the cutting tools central axis and the center of rotation, and leads to eccentricity in cutter rotation.
shown in Figure 2.7, and allows for variation of both the magnitude and locating angle of the runout, necessary for accurate simulation.

In this approach, the variable radii from the actual center of rotation to the cutting flutes are calculated for each axial disc as a function of the runout amount, the runout locating angle, the disc radius, and \( \tau \). Recall that \( \tau \) is the angular offset of the current flute due to the cutter helix angle and axial location. At some given angular location of the flute during cutter rotation simulation, the difference between the radius value for the previous flute and the radius value for the current flute is added to the existing approximate chip thickness for the current flute. If there is no difference in radii, the value approximated by Equation (2.10) is used directly, otherwise the chip thickness value is modified by the addition of the difference in radii. In the case where the difference in radii is negative and greater than the existing approximate chip thickness value, a net negative chip thickness results. In this case it is assumed that the current flute does not engage the stock material at all, and the amount missed is added to the chip thickness value for the next flute that occupies that rotational position. Refer to [KD83] for a more complete development of the method used. The success of this method lead to adaptation by DeVor in [SD86] for use in vibratory milling conditions, where the deflection of the cutter during vibration has similar effects to cutter runout at any finite instant during the cutter rotation. The correct implementation of this method was initially validated in this research by replicating the experimental results provided in [KD83], as well as through the replication of data collected during the course of this research.

One important difference in this research from [KD83] is the inclusion of ball end cutters. The runout effect of runout is that the central axis of the cutter is shifted in a radial direction, such that rotation occurs about an axis parallel to the central axis, but separated by a distance equal to the runout amount. Therefore, only the component of chip thickness that lies normal to the axis of rotation, in the radial direction, should be included in the calculations that model the effects of runout. The axial component of chip thickness is not affected by runout. In Equation (2.10), the first term is the radial chip thickness component, \( h_R \), and the second term is the axial chip thickness component, \( h_A \), or, expressed individually:

\[
\begin{align*}
    h_R & = f_{net}(D_R)(n_R)\sin(\beta) \\
    h_A & = f_{net}(D_A)(n_A).
\end{align*}
\]
This issue does not arise with flat end mills ($n_a = 0$) or when traveling in only the $X_{CT}, Y_{CT}$ plane ($D_A = 0$), as was the case in [KD83], for $h_a = 0$ in these situations. However, it must be considered in the general case. This is achieved by including only $h_{R}$, calculated with Equation (2.10a), in the chip thickness calculations that include runout. This result is then added to $h_{A}$ to arrive at the final chip thickness estimate.

2.8 Discrete Mechanistic Modeling Methods

2.8.1 The Classical Discrete Mechanistic Model

The most common mechanistic approaches use as a foundation the methods pioneered by Tlustly and MacNeil [TM75] and popularized by Devor and Kline [KDL82][KD83] with their discrete numerical implementation of Tlustly's closed form integrations. Using the Kline and Devor discrete mechanistic model, the vector forces acting on a flat end mill may be estimated as:

\[ F_x(\theta) = \sum_{i=0}^{N_z} \sum_{k=0}^{N_f} \left( -K_R K_T dz h \sin(\beta) + K_T dz h \cos(\beta) \right) \]  
\[ F_y(\theta) = \sum_{i=0}^{N_z} \sum_{k=0}^{N_f} \left( K_R K_T dz h \cos(\beta) + K_T dz h \sin(\beta) \right) \]  

where $F_x, F_y$ are the $X_{CT}$ and $Y_{CT}$ force vectors acting on the cutter at angle $\Theta$ (lb), $\theta$ is the current rotation angle of the cutter (rad), $\beta$ is the locating angle of the current flute on the current disc (rad), $N_z, N_f$ are the number of axial disk elements and number of flutes on cutting tool, $K_T, K_R$ are material constants, (Lbfl/in$^2$) and (Lbfl/Lbf) respectively, $h$ is chip thickness (in), and $dz$ is the axial thickness of a disk element (in).

This model sums, over each flute on each disc of the cutting tool model, the vector force necessary to shear a chip from the stock by a flute segment which is engaged in the metal, for a given cutter rotation angle. The first term in each equation, excluding the trigonometric term, represents the radial component of force, while the second represents the tangential component. The trigonometric terms transform the radial and...
tangential forces to cutting tool Cartesian coordinates. This model has been successfully implemented for the estimation of both three-axis and five-axis milling forces during sculptured surface machining with a flat end mill [E93] [H94] [FJD95].

The cutting tool/stock material parameters, $K_T$ and $K_R$, reflect the normal shearing and frictional forces present for a given stock material and cutting tool combination. The $K_T$ value provides the ratio between the chip thickness and the tangential milling force, while $K_R$ provides the ratio between the tangential and radial forces. These values are calculated as a function of average chip thickness and empirically based material constants, presented in [KD83] as:

$$K_T = (K_{TC}) (h_{avg})^{-p_1}$$  \hspace{1cm} (2.12a)  

$$K_R = (K_{RC}) (h_{avg})^{-p_2}$$  \hspace{1cm} (2.12b)

where $K_{TC}$ is an empirically derived material constant (Lbf/(in)$^{3-p_1}$), $K_{RC}$ is also an empirically derived material constant (Lbf/(in)$^{p_2}$/Lbf), $P_1$, $P_2$ are empirically derived material coefficients (rad), and $h_{avg}$ is the average chip thickness over the net contact region (in).

The average chip thickness is formally defined as:

$$h_{avg} = \int_{\beta_{ex}}^{\beta_{en}} \frac{h_{avg}(\beta) \, d\beta}{(\beta_{en} - \beta_{ex})}$$  \hspace{1cm} (2.13)

where $\beta_{en}, \beta_{ex}$ are the entrance and exit angles respectively (rad), and $h(\beta)$ is the chip thickness as a function of rotation angle (in).

For the case of Equation (2.9), which provides instantaneous chip thickness for a flat end cutter with constant cutting conditions, and not including runout effects, this yields an exact solution of:

$$h_{avg} = f_t \frac{(\cos(\beta_{en}) - \cos(\beta_{ex}))}{(\beta_{en} - \beta_{ex})}$$  \hspace{1cm} (2.14)
A closed form solution for Equation (2.13) that includes runout has not been obtained. To include runout in the average chip thickness calculation, Devor and Kline suggest the empirical relation [KD83]:

\[ h_{avg} = \frac{f_t (\cos(\beta_{in}) - \cos(\beta_{out}))}{(\beta_{in} - \beta_{out})} + 0.43p \]  \hspace{1cm} (2.15)

where \( p \) is the runout amount (in).

For the case of non-constant cutting conditions, the average chip thickness is calculated and summed over all engaged axial discs and divided by the number of discs engaged. An individual value of \( h_{avg} \) is not calculated for each axial disc because the mechanistic material constants of (2.11) are calculated using average force data from the entire axial depth of cut, and it is not possible when calculating the constants to distinguish the relative contributions of different axial locations. Therefore a chip thickness value averaged over the entire cutter contact region is employed, as the empirical material constants in the model are based on that assumption.

### 2.8.2 The Generalized Discrete Mechanistic Model

Variable cutter geometry affects more than just chip thickness calculations, it also directly effects the force calculations. While Equation (2.11) has been successfully implemented for milling with a flat end tool, it is insufficient for estimating forces for ball end mills, as required by this research.

The primary milling forces of interest are those in the \( X_{CT}, Y_{CT} \) plane because they lie orthogonal to the cutter axis. As most cutters have a length that is significantly greater than their diameter, and as milling forces tend to exist at the free end of the cutter, these \( X_{CT}, Y_{CT} \) forces result in the majority of the cutter deflection. Therefore when estimating forces, it is the \( X_{CT}, Y_{CT} \) components that are often sought.

The physical milling forces, however, lie in principal directions that are a function of the cutter and flute geometry. Recall from Figure 2.1 that the resultant force due to the normal shearing and frictional forces acting on a flute (\( F_{net} \)), is broken into radial and tangential components. However, this resultant milling force often has an out of plane component that is not shown in Figure 2.1, which is the result of the cutters flute helix angle and/or the cutter surface normal direction. This is shown in Figure 2.8, where the in-plane radial and tangential forces are shown for a single axial disc on the left, and the source of the out
Figure 2.8: The milling force is estimated in the $X_{CT}, Y_{CT}$ plane, while the physical force vectors occur in 3 dimensions relative to the cutter flute geometry on the current axial disc. Therefore the tangential force must be scaled as a function of helix angle and the radial force must be scaled as a function of the cutter surface normal to arrive at the $X_{CT}, Y_{CT}$ components.

Figure 2.9: Note that the axial disc thickness $dZ$ used in the classical mechanistic model is not equal to the actual flute segment length $L_f$ except in cases where $\alpha_{helix} = 0$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
of plane components is shown on the right. Therefore the tangential force must be scaled as a function of helix angle, and the radial force must be scaled as a function of the cutter surface normal, to arrive at the $X_{CT}, Y_{CT}$ components. The presence of a helix angle results in an axial component in the tangential force vector, as the tangential force is defined orthogonal to the edge of the flute where the milling occurs. The tangent force lies in a plane tangent to the cutter surface at the point where it acts, and so is not affected by changes in the cutter profile surface normal. The radial force vector, however, is greatly affected by the surface normal direction, as it lies parallel to the cutter profile swept surface normal by definition. Therefore the $X_{CT}, Y_{CT}$ contribution made by the tangential force vector is considered a function of the cutter helix angle, and the $X_{CT}, Y_{CT}$ contribution made by the radial force vector is considered as a function of the cutter profile surface normal.

In traditional discrete mechanistic modeling approaches, the length over which the milling forces act is on a given axial disc is assumed to be the axial disc thickness $dZ$. This length value is required because the force vectors are calculated as a function of the area of material being sheared by a given flute. However, this area should be defined as the chip thickness being sheared multiplied by the length of cutting flute that is performing the shearing, rather than by the axial disc thickness. These two values are equal only when there is a zero helix angle, which is generally not the case for end mills (see Figure 2.9). Therefore, to accurately define the shear area, the length of the flute segment present on the current disc is required, which is a function of the helix angle on the current disc.

The classical mechanistic model can still provide reasonable force estimates while neglecting helix angle, cutter normal direction, and flute segment length, but only for flat end mills. Early development efforts modeled only flat end mills, where the helix angle and cutter normal typically remain constant over the length of the cutter. As a result of the constant helix angle, the flute segment length remains constant over all discs. This allows the surface normal and helix angle effects to be absorbed into the empirical constants, as they are uniform over all axial discs. In effect, these variables are pre-calculated and lumped with the mechanistic constants $K_t$ and $K_b$.

However, this will not work for a ball end cutter, or any other cutter type with a variable surface normal or helix angle. For a more generalized approach, the effects of helix angle and cutter normal must be explicitly included in the mechanistic model, as they directly effect the milling force estimates. To
include the effects of variable helix angle (including variable flute segment length) and cutter normal in the force estimations, the DeVor and Kline model was expanded into a more generalized solution:

\[
F_x(\theta) = \sum_{i=0}^{N_x} \sum_{k=0}^{N_f} \{ -K_R K_T L_f h \sin(\beta)(n_{CR}) + K_T L_f h \cos(\beta) \cos(\alpha_{helix}) \} 
\]

\[
F_y(\theta) = \sum_{i=0}^{N_x} \sum_{k=0}^{N_f} \{ K_R K_T L_f h \cos(\beta)(n_{CR}) + K_T L_f h \sin(\beta) \cos(\alpha_{helix}) \} 
\]

where \( F_x, F_y \) are the \( X_{CT} \) and \( Y_{CT} \) force vectors acting on the cutter at angle \( \theta \) (lbf),

\( \theta \) is the current rotation angle of the cutter (rad),

\( \beta \) is the locating angle of the current flute on the current disc (rad),

\( N_x, N_f \) are the number of axial disk elements and number of flutes on cutting tool,

\( K_T, K_R \) are material constants, \((\text{Lbf}^2/\text{in}^3)\) and \((\text{Lbf}/\text{Lbf})\) respectively,

\( h \) is chip thickness (in),

\( n_{CR} \) is the cutter surface normal radial component for the current axial disc (unitless),

\( \alpha_{helix} \) is the helix angle of the current flute on the current axial disc (rad), and

\( L_f \) is the length of the current flute segment on the current axial disc (in).

In this equation, the radial force is adjusted using \( n_{CR} \) and the tangential force is adjusted using \( \cos(\alpha_{helix}) \), such that only the components in the \( X_{CT}, Y_{CT} \) plane are considered.

The mechanistic constants \( K_T \) and \( K_R \) are calculated in the same manner as the classical Devor and Kline mechanistic model, using the Equation (2.12). However, the average chip thickness value used in Equations (2.12a,b) must include the effects of the variable cutter geometry, not present in Equations (2.14) or (2.15). A more suitable solution applicable to cutters of variable geometry requires an average chip thickness calculation that uses the generalized instantaneous chip thickness relation of (2.10). Substituting Equation (2.10) in Equation (2.13) yields a generalized solution for average chip thickness of:

\[
h_{avg} = f_{set} \left[ (D_R)(n_{CR})(\cos(\beta_{cn})-\cos(\beta_{cn}))/\left(\beta_{cn}-\beta_{cn}\right) + (D_\lambda)(n_{CA}) \right] + (D_R)(n_{CR})(0.43)\rho 
\]

\[
(2.17)
\]
for a given axial disc. The last term in Equation (2.17) accounts for the runout contribution in the radial direction only (recall runout occurs in the radial direction only).

Note that while the $K_T$ and $K_R$ variables are calculated in the same manner as for the Devor and Kline Equation, the same values of $K_{tc}$, $K_{rc}$, $P_1$, and $P_2$ (i.e. the mechanistic material constants) cannot be interchanged between equations. These values must be recalculated using the generalized model for a given stock/cutter combination in order to arrive at valid material constant values for use in Equation (2.17). Refer to Chapter 8 for more information regarding the calculation of mechanistic constants.

It should be noted that more detailed approaches at mechanistic modeling have been developed. These include models where the effects of the cutting tools rake and relief angles are included [YP91][YA96], the explicit inclusion of flank forces [YA96][BYK94], and the use of continuous 3D Bezier curves for the modeling of cutting flutes [MSE98]. While these models demonstrate an excellent theoretical understanding of the milling process, the ultimate practicality of these systems at the current time is suspect. This is because any increased accuracy they can provide is sensitive to small changes in cutter flute rake angle, relief angle, and other difficult to obtain information. Also, a greater number of mechanistic material constants are required to reflect the flank force components.

The primary difference between the more detailed mechanistic models and the simpler model developed during this research is the level of detail involved in the cutter geometry description. While efforts were made to increase the detail of the geometry description in this research, it was done on a practical scale. Typically the more detailed systems rely on accurate empirical measurements of a given cutting tool made using a CMM (Coordinate Measuring Machine), and without this very accurate cutter geometry information effective use of the more detailed models becomes difficult.

The primary practical benefit of these methods is that the added detail aids in reducing the dependency that a specific cutter has on a specific set of material constants. This de-coupling is desirable, as it would allow a small set of material constants to apply to a wide range of cutting conditions and cutter geometries. If the required detailed cutter information is made readily available at some point in the future, this benefit warrants further investigation of these methods as it would allow for a more generalized set of material constant, provided these benefits can be demonstrated.
CHAPTER 3

MECHANISTIC MODEL AND FEEDRATE SELECTION IMPLEMENTATION

3.1 Chapter Introduction

This chapter describes the methods used to implement the generalized discrete mechanistic model presented in Chapter 2. These methods affect the accuracy and efficiency of force estimation in a practical application. Due to the wide range of cutting conditions experienced during a typical milling operation, and variety of cutting tools available, the mechanistic model must be implemented in a generic manner that simplifies definition of the required parameters, supporting efficient solution. It is also critical that the implementation maximize the accuracy of the discrete mechanistic modeling approach.

The discrete cutter representation forms the foundation of the implementation. The discrete cutter contains the information required for mechanistic modeling, in a form that is simple to define and standardized for all disc elements. The stored consists falls into categories; user-specified data that remains constant over a simulation, and variable cut data that is calculated by the supporting software models.

The constant data consists of the cutting tool geometry, and the mechanistic material constants. The mechanistic material constants relate the forces produced by a given set of cutting conditions to the specific material type being milled. The cutting tool geometry defines the radius, helix angle, flute segment length, cutter profile normal, normal and axial disc thickness, and number of flutes, for each axial disc in the discrete model. This information is 'user-specified' rather than 'user-defined' in that a only a few specific global parameters are required, which are then used to calculate the local changing cutter geometry for each disc. Note that while the mechanistic material constants are truly constant at the present time, in the future variable 'material constants' may be used to reflect tool wear or large changes in cutting conditions. Even the cutter geometry itself may be variable to include the effects of cutter deflection. This chapter describes the definition and storage of information that falls into the above, 'constant', category.

The remaining data required for mechanistic modeling is variable between individual tool moves, and therefore requires continual updating by the supporting software models. This variable data includes
the definition of the contact area between the cutter and stock, and the definition of the cutter to stock relative velocity as a function of axial location along the cutter. Variations in these parameters are what give the mechanistic model the ability to reflect the changes in milling forces that result from varying cutting conditions. The calculation of this data, performed every tool move, is presented in the following chapters, but once calculated this data is stored in the discrete cutter model.

3.2 The Discrete Cutting Tool Representation

3.2.1 Discrete Cutter Description

The discrete cutter model represents the cutting tool as a series of axial discs. All mechanistic modeling parameters are assumed constant over each disc, including cutter geometry (radius, profile normal), individual flute segment data (number of flutes, helix angle), feed-per-tooth values, and contact area information. The force vectors calculated for each disc are also stored in the discrete model. Due to this level of detail, the discrete cutter model may be thought of as a 'micro-model' of the cutting tool, as opposed to a swept toolpath envelope required for intersection calculations (presented in Chapter 5), which may be thought of as a cutting tool 'macro-model'. Table 3.1 summarizes the data stored on each axial disc.

The discrete cutter model also is responsible for managing the cutter rotation simulation, required by the mechanistic model to reflect variations in force that occur as a result of cutter rotation. For a constant set of cutting conditions, these force variations are cyclic and are repeated over all rotations. Therefore only a single rotation simulation is required per tool move. Due to the large number of flute locating angles required for simulation of even a single rotation, these values are pre-calculated and stored in the discrete cutter model to improve computational efficiency.

In addition to cutter and flute geometry, the discrete cutter model also stores contact area information. This defines the region of contact between the cutter and stock for a given tool move. If a given disc intersects the stock material, it is assumed that the entire axial length of the disc is engaged, and so no axial bounds on the contact area exist for a single disc. The radial boundaries of the contact area are defined for a given disc in the Cutting Tool polar coordinates, and they are denoted as entrance and exit angles $\beta_{en}$ and $\beta_{ex}$ (note that four entrance/exit angle pairs are stored per disc, as explained in Chapter 6).
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Cut Geometry Parameters...</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Disc Index</td>
<td>K</td>
<td>(none)</td>
<td>Index uniquely identifying current disc</td>
</tr>
<tr>
<td>Disc Radius</td>
<td>R_d</td>
<td>(in)</td>
<td>Nominal Disc Radius orthogonal to cutter axis</td>
</tr>
<tr>
<td>Helix Angle</td>
<td>(\alpha_{\text{helix}})</td>
<td>(rad)</td>
<td>Helix angle of flutes on disc</td>
</tr>
<tr>
<td>Number of Flutes</td>
<td>N_f</td>
<td>None</td>
<td>Integer number of cutting flutes on disc</td>
</tr>
<tr>
<td>Cutter Profile Normal</td>
<td>(n_e)</td>
<td>None</td>
<td>2D unit vector defined relative to direction of travel</td>
</tr>
<tr>
<td>Axial Disc Thickness</td>
<td>(dZ)</td>
<td>(in)</td>
<td>Thickness of disc parallel to cutter axis</td>
</tr>
<tr>
<td>Flute Segment Length</td>
<td>L_f</td>
<td>(in)</td>
<td>Distance traversed along each flute segment on the disc</td>
</tr>
<tr>
<td>Axial Location</td>
<td>A</td>
<td>(in)</td>
<td>Axial distance of disc from cutter location position</td>
</tr>
<tr>
<td>Flute Rotational Position</td>
<td>(\beta_{\theta,k})</td>
<td>(rad)</td>
<td>Discrete rotational position array for flutes; function of current flute (k) &amp; current cutter rotational position (\theta).</td>
</tr>
<tr>
<td>Runout Radius</td>
<td>R_i</td>
<td>(in)</td>
<td>Radius to each of the flutes including the effects of cutter runout, (i = 1..N_f). See Chapter 2, &amp; [KD83]</td>
</tr>
<tr>
<td><strong>Cut Geometry and Force...</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stock Engagement</td>
<td>InStock</td>
<td>YES, NO</td>
<td>A binary flag indicating if disc engages the stock</td>
</tr>
<tr>
<td>Radial Depth of Cut</td>
<td>(N_{\text{max}}, N_{\text{min}})</td>
<td>(in)</td>
<td>The maximum and minimum depth of cut orthogonal to both the cutter axis and the direction of travel</td>
</tr>
<tr>
<td>Entrance &amp; Exit Angles, (up to four pairs per disc)</td>
<td>(\beta_{\text{em}}, \beta_{\text{ex}})</td>
<td>(rad)</td>
<td>Angular location of stock engagement and disengagement, defined relative to rotational direction</td>
</tr>
<tr>
<td>Feed-Per-Tooth</td>
<td>(f_t)</td>
<td>(in/tooth)</td>
<td>The net feed per tooth value experienced by disc</td>
</tr>
<tr>
<td>Milling Force Vectors</td>
<td>F_x, F_y</td>
<td>Lbs</td>
<td>Milling force contributions in the (X_{CT}, Y_{CT}) directions</td>
</tr>
</tbody>
</table>

Table 3.1: Axial disc parameters, the values of which are assumed constant for a given axial disc. The parameters contained at the top of the table define the discrete cutting tool, and are constant for a given cutter type.

These angles indicate where a flute first enters the contact area, and where it exits. This simplifies determination of engaged flutes at a given cutter rotation angle, a condition indicated by \(\beta_{\text{em}} \leq \beta \leq \beta_{\text{ex}}\), where \(\beta\) is the current flute locating angle. This chapter only discusses contact area storage; the methods used for contact area definition are presented in Chapter 6.

While the development of a generalized discrete cutter model presented here is specifically developed for application with ball or flat end cutters, it is not limited to only these types. As ball end cutters contain the full range of variation for radius, cutter profile normal, helix angle, and number of flutes, the methods presented here can be applied to model any cutter geometry in a discrete manner.

3.2.2 Axial Discretization and Disc Thickness Values

Axial discretization refers to the representation of a solid cutting tool as a set of axial disc elements. Figure 3.1 presents discrete models for ball and flat end cutters. On cylindrical portions of the
Figure 3.1: The axial discretization of flat and ball end mills for discrete cutter model definition using the cutter profile as a guideline.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
cutter, typically the majority of the cutter geometry values found in the upper portion of Table 3.1 are constant and equal to the corresponding values assigned to the cutting tool. This includes radius, helix angle, number of flutes, and cutter profile normal. However, these parameters tend to vary on the ball portion of a cutting tool as a function of axial location.

For a reasonable discrete approximation, the axial thickness $dZ$ of the disc elements must be sufficiently small so that the variation of parameter values between adjacent discs is not significant. However, disc thickness also has a direct effect on computational efficiency, as the mechanistic model sums forces over all engaged axial discs. Also, extremely thin axial discs contribute to the cumulative effects of floating point round-off and other computational forms of error.

As a starting point in defining a required disc thickness, it may be treated as a function of the cutter helix angle and radius. When a given disc contacts the stock during a tool move, this contact may only occur at a subset of the total rotation angles modeled; all flutes on the disc lie outside the contact area at the remaining angles. Also, when a flute segment is engaged, its angular position has a profound effect on chip thickness, and thus force contribution. As stock contact is determined using entrance and exit angles, the helix angle effectively defines which flutes on a given disc lie within the contact area at some rotation angle, as it defines the flute locating angle for a given disc (recall that this angle is assumed constant over the disc). One of the primary purposes of the cutter axial discretization is to model, in discrete angular increments, the spiral progression of the flutes about the cutter due to helix angle. Therefore, one purpose for selecting a desired disc thickness is to control the variation in helix angle that occurs between adjacent discs, i.e. limit the modeled discrete increment in helix angle. Assuming a constant radius and helix angle, Equation (2.7) may be used to derive a relation that defines disc thickness as a function of the desired helical increment:

$$dZ = \frac{(\text{HelixIncr})(R)}{\tan(\alpha_{\text{helix}})}$$

(3.1)

where $dZ$ is the disc thickness required to maintain 'HelixIncr' (in),

HelixIncr is the desired variation in helix angle over a given disc (rad),

$R$ is the cutter radius (in), and

$\alpha_{\text{helix}}$ is the cutter helix angle (rad).
For example, if a 2° helical increment were desired between discs on a 1" diameter cutter, the required axial disc thickness $dZ$ would be 0.03" (assuming the standard 30° helix angle).

Note that Equation (3.1) provides only a basic rule of thumb for calculating disc thickness, as there are parameters in addition to helix angle to consider. For example, the axial depth of cut plays a role as a result of the discrete axial stock engagement; a disc is either assumed to fully engage the stock in the axial direction, or it does not contact the stock at all. Therefore, while Equation (3.1) would predict a rather large allowable disc thickness for cutters with very shallow or zero helix angles, this could lead to overly conservative force estimates. The problem is further complicated as all disc related parameters remain constant over the length of the disc, not just helix angle (e.g. cutter profile normal, number of flutes, flute length, etc.). While a disc thickness may be acceptable with respect to helix angle, it could limit the variation of some other parameter. This could be an issue particularly for ball end cutters, where all these parameters vary. Therefore on the cutter ball, disc thickness values equal to or slightly less than that for the cylindrical section of the cutter are desired. Also note that, on the ball section, the definition of 'thickness' differs from the axial thickness that is applied to the cylinder section, as explained below.

When discretizing a cutter ball, the critical dimension for defining 'thickness' is the length of cutter profile that lies orthogonal to the cutter profile normal for that disc. This length is important as it ensures uniform chip thinning in the axial direction over the cutter ball. Recall that chip formation occurs in the cutter normal direction, as explained in Chapter 2. As a disc's profile normal is assumed constant over its length in the discrete model, the chip thickness calculated by Equation (2.10) is therefore also constant over its length (for some fixed locating angle value), although in reality the normal and chip thickness vary continuously. The use of axial discs that span equally divided lengths of the cutter profile ensure a uniform variation between thickness values on adjacent discs (at a given locating angle) and thus a uniform distribution of chip thickness values over the cutter ball surface.

For a flat end cutter, or the cylindrical portion of a ball end cutter, thickness of the disc orthogonal to the cutter profile, or the 'profile length', is equal to the axial thickness:
\[ d_{n_{CY}} = dZ \]  
(3.2a)

where \( d_{n_{CY}} \) is the constant disc thickness orthogonal to the cutter profile normal (in), and \( dZ \) is the user-specified axial disc thickness value (in).

Axial thickness is variable for a cutter ball, and is a function of the constant profile length value \('d_{n_{Ball}}', as shown in Figure 3.1. The value of \( d_{n_{Ball}} \) is defined using a constant angular increment \( d\sigma \), and is calculated as the arc-length spanned by \( d\sigma \) at the cutter ball surface, one cutter radius \( R \) from the ball center:

\[ d_{n_{Ball}} = R d\sigma \]  
(3.2b)

where \( d_{n_{Ball}} \) is the constant disc profile length orthogonal to the ball cutter profile normal (in), \( d\sigma \) is the user defined angular increment (rad).

Therefore a constant \( d\sigma \) is used to define the profile length on the cutter ball just as a constant \( dZ \) is specified for the definition of axial disc thickness on the cylindrical portion. Note that this differs from previously implemented ball end cutter models that were developed specifically for application with the Devor and Kline mechanistic model [E93].

The value of \( d\sigma \) is specified for cutter ball discretization, as opposed to \( d\eta \), because the cutter ball spans a finite distance along the cutter axis (one Radius), over which a great deal of variation in cutter and flute geometry occurs. Specifying \( d\sigma \) ensures that at least some minimum number of discs will be present over this finite distance, with a concentration at the cutter bottom where much of the milling occurs. The number of discs present on the cutter ball is denoted \( N_{D-Ball} \) and is a subset of the total number of discs modeling the cutter, \( N_D \).

In addition to disc thickness, the axial length of cutter modeled also contributes to the number of discs that are required. If the absolute maximum axial depth of cut during a milling operation is known, then that value may be used as the axial length of cutter that must be discretely modeled. If this value is not known, then either a conservative estimate or the fluted length of the cutter may be used. The modeled cutter length is therefore a known constant that is based on the requirements of a given milling operation.
While the above discussions provide some insight to the problem, variable cutter geometry and cutting conditions makes it very difficult to arrive at any concrete conclusions that apply to all cutters. Extensive testing with varying disc thickness values is needed to develop rules that balance performance and accuracy. In this research, fairly thin discs relative to the cutter and Z-buffer geometry have been used to ensure accuracy. The values used have usually been on the order of 0.01 - 0.05" for the dZ values of cylindrical cutter portions, and 2°-4° angular increments for dσ when discretizing a ball.

3.3 Definition of the Axial Disc Parameters

3.3.1 Axial Disc Definition

The axial discs of the discrete cutter model are all identified using an integer disc index value 'i'. This index starts at the axially lowest disc with a value of 0, and all discs are then sequentially assigned an index value one greater than the preceding disc up to the top-most disc, assigned an index of N_D - 1, as shown in Figure 3.1. Using these indices, the disc parameters are assigned values by looping through the discs from 0 to N_D-1. The discs that reside on the cutter ball may be easily identified, as their index is less than N_D-Ball (alternately, they may be identified by their negative axial offsets, defined shortly). Ball end cutters contain both ball and cylindrical portions, and so the discussion below will be for a ball end cutter; ignore the ball specific portion if a flat end cutter is to be modeled.

3.3.2 Axial Thickness

On a cylindrical cutter portion, the axial disc thickness dZ of all discs is constant and user defined. On a cutter ball it is a variable function of axial location due to the constant dσ_Ball requirement. Note that the very bottom of the cutter ball is not modeled (i.e. there is no σ = 0) as this disc would have a zero radius. Therefore the axial thickness may be defined as a function of the disc index 'i' using:

\[
\text{if } (i \geq N_{D-Ball}) \\
\quad dZ(i) = dZ \\
\text{else}
\]  

(3.3a)
\[ dZ(i) = R \left( \cos\left( (i+1)d\sigma\right) - \cos\left( (i+2)d\sigma \right) \right) \quad (3.3b) \]

where \( dZ(i) \) is the axial thickness of the current \((i^{th})\) axial disc (in),
\( dZ \) is the user defined axial disc thickness for the cylindrical cutter portion (in),
\( R \) is the cutter radius (in),
\( i \) is the current disc index value, and
\( d\sigma \) is the user defined angular increment for disc definition (rad).

Note that the logic sequence checking the current disc index to find if the current disc is on the cylindrical portion \(( i \geq N_{D-Ball})\) or the ball portion \(( i < N_{D-Ball})\) of the cutter is common to all parameter definitions (for a flat-end cutter, \( N_{D-Ball} = 0 \)). To prevent repetition, two new variables will be introduced, \( i_{ cyl} \) and \( i_{ball} \). Those parameters defined using \( i_{cyl} \) assume that the \(( i \geq N_{D-Ball})\) condition is met and the parameter is being defined for a disc on the cutter cylinder, and those defined using \( i_{ball} \) assume that the current disc is known to exist on the cutter ball.

### 3.3.3 Axial Location

The axial location 'A' of a disc is assigned to the disc's axial center. It is defined relative to the cutter location position, which lies at the bottom of a flat end cutter at its radial center, or at ball center for a ball end cutter. Therefore, the axial location is positively valued on the cylindrical portion of the cutter, and negative on the ball portion. On a cylindrical cutter portion, the axial location is defined as:

\[
A(i_{cyl}) = (i - N_{D-Ball})dZ + dZ/2 \quad (3.4a)
\]

where \( i_{cyl} \) indicates that this only applies for indices to discs on the cylinder (i.e. \( i \geq N_{D-Ball} \)).

On a cutter ball, the axial location is defined as:

\[
A(i_{ball}) = -R \cos\left( (i+1)d\sigma \right) + dZ_{ball}(i)/2 \quad (3.4b)
\]
3.3.4 Disc Radius

On the cylindrical cutter portion, the axial disc radii are constant and equal to the cutter radius $R$:

$$R(i_{cy}) = R$$  \hspace{2cm} (3.5a)

where $R(i_{cy})$ is the radius of disc $i$ on the cutter cylinder (in).

On the ball portion, the disc radius is variable. Although the axial location of a given disc is defined at the disc center, the radius value assigned to a ball end disc is the minimum value that lies at the axial bottom of the disc element. This maximizes the entrance and exit angles calculated using the disc radius values (see Chapter 6). On the ball portion of the cutter, the radius value is defined as:

$$R(i_{ball}) = R \sin( (i+1)d\sigma)$$  \hspace{2cm} (3.5b)

where $R(i_{ball})$ is the radius of disc $i$ on the cutter ball (in),

$R$ is the cutter radius (in),

$i$ is the index to the current axial disc, and

$d\sigma$ is the user defined cutter ball angular increment (rad).

3.3.5 Cutter Profile Normal

The cutter profile defines the silhouette of the cutting tool, i.e. a cross sectional plane containing the cutters central axis, as shown in the discrete cutter images of Figure 3.1. The cutter profile normal, $n_r$, is a 2D unit vector that lies in the plane containing the cutter axis and the direction of travel, and defines the normal to the cutter profile for the current axial disc. This vector is required for chip thickness calculation using Equation (2.15), with $n_r$ defined in terms of axial and radial components. The axial component lies along the cutter axis, and the radial component lies orthogonal to this. On the cylindrical portion of the cutter, this vector is constant over all discs, and its axial and radial components are defined as:

$$n_r(i_{cy}) = 0$$  \hspace{2cm} (3.6a)

$$n_t(i_{cy}) = 1$$
where \( n_a \) and \( n_r \) are the axial and radial components of \( \mathbf{n}_c \) for disc \( i \) on the cutter cylinder.

On the ball portion of the cutter, the normal vector components are defined based on the cutter profile normal that exists at the axial center of the disc as shown in Figure 3.2, or:

\[
\begin{align*}
    n_a(i_{\text{bab}}) &= \cos((i+1.5)d\sigma) \\
    n_r(i_{\text{bab}}) &= \sin((i+1.5)d\sigma)
\end{align*}
\]

### 3.3.6 Cutting Flute Representation

The generalized discrete mechanistic model presented in Chapter 2 estimates force vectors acting on the cutting tool at a given cutter rotation angle, distributed over the axial discs of the discrete cutter model. On a given disc, force vectors are estimated as acting on the cutting flutes, and are calculated as a function of the area of material being sheared by a given flute. This area is defined as the chip thickness being sheared, multiplied by the length of cutting flute that is performing the shearing. To define the shear area, the length of the flute segments present on the current disc is required. Additionally, as the force vectors are assumed to act on the flute segments of the current disc, estimation of the directional components of the force vectors requires knowledge of the flute helix angle present on the current disc. Therefore a description of the flute segment length and the helix angle present on each axial disc is required. As with the other disc parameters, these values are assumed constant over the length of the disc, and additionally they are also assumed constant between all flutes on the disc. However, on the ball portion of the cutting tool, the number of flutes present on the cutter tends to decrease towards the bottom of the ball. Therefore the number of flutes present on a given disc is also required.

On the cylindrical portion of the cutting tool, the helix angle \( \alpha_{\text{helix}} \) is typically constant and equal to the value defined for the net cutting tool. The flute segment length is calculated as a function of the helix angle, as shown in Figure 3.3, using the relation:

\[
L_f(i) = \frac{dn}{\cos(\alpha_{\text{helix}})}
\]

\[
\text{where } L_f(i) \text{ is the flute segment length of the } i^{\text{th}} \text{ disc (in)},
\]
Figure 3.2: Definition of the axial and radial components of the cutter ball profile normal vector.

Figure 3.3: The flute segment length $L_f$ for a given disc is a function of the helix angle, $\alpha_{\text{helix}}$, and the length of the cutter profile that is orthogonal to the cutter profile normal, $dn$.

Figure 3.4: Definition of the helix angles on the ball portion of the cutter makes use of the cutter ball flute definition angle, $\sigma$, both in transitioning from $\alpha_{\text{helix}}$ to $\alpha_{\text{min}}$ and for flute termination.
dn is the length of the disc orthogonal to the cutter normal, = dZ on a cylinder (in), and 
\( \alpha_{\text{helix}} \) is the user provided cutter helix angle (rad).

The value of \( dn \) is used in Equation (3.7), as opposed to \( dZ \), so that this solution also applies to the cutter ball portion of ball end cutter tools, in which case the value of \( dn \) is calculated using Equation (3.2b).

On ball end cutters, the helix angle generally decreases uniformly from the constant \( \alpha_{\text{helix}} \) value of the cylindrical portion to some minimum value part way down the cutter ball. However, the value and start location of the minimum helix angle can vary appreciably between different manufacturers. To account for this variation, two user-defined variables are employed. The variable \( \alpha_{\text{min}} \) represents the value of the minimum helix angle, and 'HelixEnd' represents the location where it begins. As shown in Figure 3.4, the helix angle remains constant at \( \alpha_{\text{min}} \) below the HelixEnd position. HelixEnd is user defined, allowing the helix variations to be defined as a function of \( d\sigma \) through the use of a cutter ball definition angle '\( \sigma \)'.

\[
\sigma(i) = \pi/2 - (i+1) \ d\sigma 
\]

where \( \sigma(i) \) is the cutter ball flute definition angle for disc i on the cutter ball (rad).

Discs that have a cutter ball flute definition angle \( \sigma(i) \) greater than HelixEnd have their helix angles set to the \( \alpha_{\text{min}} \) value. Discs that lie between HelixEnd and \( \sigma = 0^\circ \) (at the point where the cylindrical portion of the cutter starts) have a variable helix angle, which varies between the user specified values of \( \alpha_{\text{helix}} \) (the maximum helix angle) and \( \alpha_{\text{min}} \) as a function of \( \sigma \), as shown in Figure 3.4.

Using \( \sigma \) to model the helix transformation, as opposed to axial location \( A \), allows for a smooth interpolation between \( \alpha_{\text{helix}} \) and \( \alpha_{\text{min}} \) with uniform variation in helix angle between discs. This smooth transition most closely models the physical flute geometry. For values of \( \sigma(i) \) less than HelixEnd, the transition between \( \alpha_{\text{helix}} \) and \( \alpha_{\text{min}} \) is achieved using the linear interpolation:

\[
\alpha_{\text{helix}}(i_{\text{ball}}) = \alpha_{\text{helix}} - (\sigma(i) / \text{HelixEnd} \ ) (\alpha_{\text{helix}} - \alpha_{\text{min}}) 
\]

where \( \alpha_{\text{helix}}(i_{\text{ball}}) \) is the helix angle for disc i of the cutter ball.
On standard four fluted ball end cutters, two of the four flutes typically terminate above the bottom of the cutting tool to aid in chip clearance as the radius diminishes. While this affects only a relatively small region on the bottom of the cutter, it doubles the feed-per-tooth value in that region and so should not be neglected, as this can lead to underestimation of forces during shallow cuts. Inclusion of this effect requires the use of a user-defined parameter 'FluteEnd'. This is an angular value, defined relative to \( \sigma \) like HelixEnd, which defines the location where two of the 4 cutter flutes terminate. All discs that have a flute definition angle \( \sigma(i) \) that lies below this angle are defined to have \( N_f = 2 \), while those that lie above it are assigned \( N_f = 4 \) (assuming a standard 4 fluted cutter). In terms of the flute index \( k \) used in the mechanistic model, flutes 1 and 3 are retained, while flutes 2 and 4 are terminated.

Figure 3.5 shows renderings of actual modeled ball end cutters; note the smooth transition in \( \alpha_{\text{helix}}(i) \) from \( \alpha_{\text{helix}} \) to \( \alpha_{\text{min}} \) and also the flute termination. The values for \( \alpha_{\text{helix}} \), \( \alpha_{\text{min}} \), HelixEnd, and FluteEnd may be determined using manufacturer data, measured data, or by inspection (compare physical cutter to on-screen display). The default values used are \( \alpha_{\text{helix}} = 30^\circ \), \( \alpha_{\text{min}} = 0^\circ \), HelixEnd = 70°, FluteEnd = 70°.

### 3.4 Simulation of the Bottom of a Flat End Cutter

When milling with a flat end cutter, it is possible to remove material with the bottom of the cutting tool. This is not a typical mode of operation for flat end cutters, but it does occur, particularly when transitioning between levels while performing a constant-Z-level operation. It is not a standard mode of operation as most flat end cutters have a 'dead zone' at the cutter location position, where no material removal occurs. Therefore when removing material with the bottom of the cutting tool, the cutter typically does not move directly in \(-Z_{CT}\), but rather at some angle to the \(X_{CT}, Y_{CT}\) plane.

The mechanistic constants, and the mechanistic model itself, are not specifically developed to simulate material removal with the bottom of a flat end cutter. However, as this case can occur, it should be modeled to ensure robust operation. If neglected, only the material removal that occurs on the shank of the cutter will be considered, which can allow the cutter to travel too rapidly when downward motion is present while milling with a flat end cutter. This is undesirable, and it is standard practice in industry to inhibit the feed when milling on the bottom of a flat end cutter.
Figure 3.5: Several examples of the modeled ball end flute geometry rendered from discrete model.

Figure 3.6: The cutting flutes located on the bottom of axial disc \((i=0)\) for a flat end mill are included in the force calculations when the cutter is travelling downwards relative to its axis.

Figure 3.7: The calculated output feed is inhibited when the cutter first enters the stock (left), or when large directional changes are occur (right).
The condition of a flat end cutter removing material with its bottom is identified through a combination of three criteria. The most basic is that the current cutting tool is a flat end cutter. The second criteria is that the direction of travel vector for the cutting tool, \( \mathbf{D} \), must have a negative component along the axis of the cutter, \( \mathbf{A} \), or \( \mathbf{D} \cdot \mathbf{A} < 0 \). The third criteria is that material removal is found to occur during the current tool move by the geometric model. If these conditions are met, the bottom of a flat end mill is modeled using methods identical to those used for the shank of the mill. This is achieved by modeling the bottom of the cutter as a unique axial disc, and passing this 'disc' to the mechanistic model for force estimation. The forces estimated are then included in the net force summation for the tool move.

The method developed to handle this condition is conservative in its force estimate, so that relatively low feeds will be selected for this uncommon condition. The bottom of a standard 4 fluted flat end cutter typically has flutes that extend linearly in a radial direction from the point of termination of the shank flutes almost to the cutter location point at the radial center of the cutter (see Figure 3.6). For a generic cutter with \( N_f \) flutes, \( N_f \) flutes reside on the cutter bottom. In this approach, these flutes are modeled like any other flute, using their existing geometry. The parameters set for the purpose of chip thickness and force estimation for these flutes are relatively simple, and are provided in Table 3.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D_z )</td>
<td>0</td>
</tr>
<tr>
<td>( N_c = [n_r,n_h] )</td>
<td>([0,-1])</td>
</tr>
<tr>
<td>( L_f )</td>
<td>Cutter Radius</td>
</tr>
<tr>
<td>( \alpha_{helix} )</td>
<td>0</td>
</tr>
<tr>
<td>( \beta_{min}, \beta_{max} )</td>
<td>180°, 0°</td>
</tr>
</tbody>
</table>

Table 3.2: Disc parameters set for force estimation on the bottom of a flat end cutter.

Note that Table 3.2 does not include a listing of all axial disc parameters; it only contains those directly related to chip thickness and force estimation. The material constants used are the same as for the shank of the cutter. The remaining parameters are not required in this case as they only serve to support the calculation of these variables.

The radial force component on the bottom of the cutter acts purely in a \( +Z_{CT} \) direction, and does not directly contribute to the net \( X_{CT}, Y_{CT} \) forces that are estimated. Although there is no direct contribution from the radial component of force to the net estimation, it may be assumed that this \( Z_{CT} \) force acting on the
cutter bottom increases friction between the bottom flutes and the stock material. This friction can indirectly increase the $X_{CT}$, $Y_{CT}$ milling forces, and so the milling forces estimated for the cutter bottom are multiplied by an arbitrary constant of 1.5 to ensure conservative results.

Note in Table 3.2 that the entrance and exit angles are forcibly set to 180° and 0° respectively. This is done to maximize the force estimation provided. It is possible for the cutter bottom to experience full 360° - 0° stock engagement, however as only the tangential component of force remains, this would result in complete force cancellation, i.e. a zero force estimate. Also, cutting can occur in the 360° - 180° range when $D \cdot A < 0$, this too is neglected as it could result in cancellation with forces calculated for the shank of the cutter. The net goal of this approach is to estimate worst case forces that will result in conservative feeds to effectively deal with this situation. This method is not intended for accurate force estimation, it is only intended to ensure safe operation when $D \cdot A < 0$ for a flat end cutter.

### 3.5 The Discrete Cutter Model and Applied Mechanistic Modeling

Once the discrete cutting tool model has been defined, it may be used in the implementation of the mechanistic model. As this topic has been covered in detail in other works [KDL82][KD83][E93], it will only be presented briefly here, and in the context of this application. In this application, the parameters that describe the discrete cutting tool geometry, contained in the upper section of Table 3.1, remain constant during the milling simulation for a given cutting tool. The remaining parameters are supplied for each new tool move by the supporting models (geometric, NC machine), to represent changes in cutting conditions.

The cut geometry parameters, which consist of the InStock flag and the entrance and exit angle values $\beta_{en}$ and $\beta_{ex}$, are supplied for each disc using information from the geometric model. The values of $\beta_{en}$ and $\beta_{ex}$ are used to identify which flute segments on each axial disc are engaged in the stock at the current rotation angle, and also to calculate the average chip thickness value using Equation (2.17).

The value of $f_{\text{net}}$ is calculated for each engaged disc using Equation (2.10) which requires the net relative velocity vector $V_{\text{net}}$ provided by the CNC machine model. Note that $f_{\text{net}}$ is also a function of the number of flutes present on the current disc. The value of $f_{\text{net}}$ is required for calculating values of chip thickness using Equation (2.10).
Each disc flagged as engaging the stock during a tool move (InStock = YES) is included in the force calculations for that move, which are performed at a series of discrete rotational positions. The flute locating angles are pre-calculated for each discrete rotational position and stored in the $\beta$ array on each axial disc. At a given rotation angle, these flute locating angles are compared against the values of $\beta_m$ and $\beta_m$ for that disc. All flute segments found to engage the stock at the current rotation angle have $X_{CT}, Y_{CT}$ force values calculated using Equation (2.16), which are summed over all flutes to arrive at the net $F_x,F_y$ values for each disc.

To arrive at $F_{mag}$, the net milling force acting on the cutter at a given cutter rotation angle, the $F_x,F_y$ forces calculated at are summed over all discs, and the magnitude is calculated. If the value of $F_{mag}$ exceeds the existing maximum force value for the entire tool move, $F_{max}$, the value of $F_{max}$ is reset to the current value of $F_{mag}$. Otherwise the value of $F_{mag}$ is overwritten at the next rotation angle (unless explicitly stored in an array to plot milling force vs. rotation angle). All angular dependant force values ($F_x,F_y, F_{mag}$) are initialized to zero for each new rotation angle, and $F_{max}$ is initialized to zero for each new tool move.

After forces have been calculated for all of the discrete rotational positions, the existing value of $F_{max}$ is used to represent the maximum net milling force for the tool move. This is the force to be controlled through variation of the input feedrate value as a result of automatic feedrate selection.

### 3.6 Automatic Feedrate Selection via Iterative Mechanistic Modeling

To solve for feed values that maintain a user-specified maximum force threshold, the mechanistic model is used in an iterative manner. In this approach, during each iteration the force estimate value $F_{max}$ is compared to the desired cutting force. If the values do not match (within a given tolerance), the feedrate value is adjusted appropriately, and $F_{mag}$ is recalculated. Iteration terminates when a feed value is found that produces an estimated force in the desired range (or some other exit condition is met; this is explained shortly). The feed value that exists at the termination of iterations is output to the updated G-code file.

The feed value calculated for the previous tool move is typically used as the initial feed value for the next tool move. In cases where the cutting conditions do not vary extensively, this can provide an
acceptable cutting force on the first iteration. For the first tool move in a G-code file an initial feed value specified in the input G-code file is used, and an arbitrary feed of 10 IPM is used if none is present.

If a force in the desired range is not estimated during the first iteration, a linear interpolation is used to solve for the second iterations feed value:

\[ f(n+1) = (f(n)) \left( \frac{F_{\text{Desired}}}{F_{\text{Max}}(n)} \right) \]  

where \( f(n+1) \) is the input feed for the next iteration (IPM), \( f(n) \) is the feed used for the current iteration (IPM), \( F_{\text{Desired}} \) is the desired force magnitude (Lb), and \( F_{\text{Max}}(n) \) is the peak force magnitude for the current iteration (Lb).

While the feed/force relation is non-linear, this can select the appropriate feed during the second iteration when small changes in cutting conditions occur. More importantly, in cases where the cutting conditions are changing appreciably, this method quickly adjusts the feed value into the appropriate operating range.

If the value of \( F_{\text{Max}} \) estimated during the second iteration does not fall in the desired range, all subsequent feed values are solved using a standard 1/2 interval search (also known as a bisection method). In this approach, first the required feed value is bounded between two feeds, one that produces too great a force, and one that produces too small a force (provided the correct feed is not found by chance in the process). Then the gap between the 'high feed' and the 'low feed' is continually cut in half until the required feed is obtained. This method and its application to feedrate selection is presented in detail in [H94].

There are other 'exit conditions' that will terminate the iterative solution in addition to the estimation of a force value within the desired range. All of these exit conditions require the resultant feed value to produce a force value that is less than the upper bound on the acceptable range, except one. The one exception to this rule is when the current feed value is less than 0.1 IPM, the minimum allowable feed on many NC controllers. In this case a feed value of 0.1 IPM is output regardless of the force.

The reason that 0.1 IPM is generally the minimum allowable feed is that many controllers have a 0.1 IPM resolution, which leads to the second exit condition. As the minimum allowable change in feedrate is 0.1 IPM, any iterations that result in changes in feed less than this amount are moot and result in
wasted computation time (as the resultant feed would have to be rounded down to the nearest 0.1 for output anyway). There are cases where a change in feedrate of 0.1 IPM results in a change in force magnitude that skips over the acceptable force range. To prevent unnecessary iterations in this case, the force value from the previous iteration is always stored, and the difference in feedrate between iterations is stored. If the difference in feeds is less than or equal to 0.1, and the current force is greater than the allowable range, and the previous force is less than the allowable range, then the iterations are exited and the previous feed value is output.

The third exit condition occurs if any of the individual machine axes exceeds an allowable limit, or if the current feed value exceeds a user defined maximum feed value. There are hard limits for the machine axes to prevent motor damage, and these are stored inside the NC controller. If the machine is asked to travel at a rate that causes one of these axes to travel faster than the allowable limit, the actual velocity produced is that which saturates that axis at its limit. Therefore when the axis velocities are calculated in the CNC machine model, they are checked against the machine limits (which must be provided). If any of the axes exceeds its limit, the feedrate that produces saturation on the limiting axis is calculated. This feed then has a force value calculated for it, and if this is less than the maximum allowable value the saturation feed is adequate, otherwise it is used as the starting point for a downward iteration.

Similarly, users can specify "MaxFeed", a maximum allowable feed, for added control. If the value of MaxFeed is exceeded, then MaxFeed is checked if to find if it produces a force less than the upper limit on the allowable force range; if it does then MaxFeed is output to the updated G-code file, else it is used as the starting point for a downward iteration. The default value for MaxFeed is 100 IPM.

The final exit condition is a maximum allowable chip thickness. While the milling forces estimated by the mechanistic model are a function of chip thickness, it is possible for acceptable milling force values to be calculated while producing unacceptable chip thickness values. This case occurs particularly when a small amount of material is being removed. In an effort to limit tool wear and flute damage caused by excessive chip loads, a user-defined maximum chip thickness value may be used. This method is actually quite similar to traditional methods where cutter feeds and speeds are selected from tabular data, the net result of which is control of the maximum chip thickness produced, only in this case...
the maximum chip thickness is controlled for every tool move. Note that this alone does not constitute a sufficient condition for feedrate selection, as during heavier cuts it could result in excessive milling forces.

The calculation of the maximum chip thickness is a straightforward process. At each discrete rotational position, the chip thickness value of all engaged flute segments is required for force calculation. Therefore, the maximum chip thickness present during a given tool move may be obtained by storing the maximum value calculated. If the value found during any given move is found to exceed the maximum allowable value, and the current force value is less than the than the upper bound on the acceptable force range, the feed iterations are exited. The output feed value is then adjusted to approximately the value that produces the maximum allowable chip thickness. As there is a linear relationship between chip thickness and feedrate at any given location on the cutter, the required feed may be calculated using:

\[
\text{\(f_{out} = f_{calc} \left( \frac{h_{MaxDesired}}{h_{Max}} \right)\)}
\]

where

- \(f_{out}\) is the feed value output to the updated G-code file for the current tool move (IPM),
- \(f_{calc}\) is the current iteration feed value that resulted in excessive chip thickness (IPM),
- \(h_{MaxDesired}\) is the user defined desired maximum chip thickness value (in), and
- \(h_{Max}\) is the maximum chip thickness value calculated during the current tool move (in).

3.7 Feed Inhibitors

When calculating feeds for practical application, transient conditions exist that require the feed calculated by the iterative model to be inhibited, or reduced in value by some percentage, as the mechanistic model assumes steady state cutting conditions. It also currently does not take CNC controller dynamics into account. There are mechanistic models that take dynamic effects into account [MA91][ESE98], but none are yet widely accepted as providing acceptable solutions over a wide range of conditions. Additionally, most of the dynamic models operate in the time domain and require iterative solution of force values (as opposed to just feed values), which greatly increases computation time. Therefore, in order to maintain an efficient and robust model, inhibitors are used. In the course of this discussion, the variable 'f\text{\(out\)' refers to the value actually written to the updated G-code file. The variable
'\(f_{\text{calc}}\)' refers to the 'calculated feed' that is suggested by the iterative model; note that \(f_{\text{out}} < f_{\text{calc}}\) when inhibitors are used.

The first feed inhibitor is stock engagement. If no material removal is noted for a given tool move, but the following tool move does engage the stock, then a stock engagement inhibit feed is activated. This results in one of two scenarios, both shown in the left-hand image of Figure 3.7.

In the first scenario, stock engagement occurs gradually. In this case, the maximum allowable feed is reached, or axis saturation occurs, before the desired force range is obtained, and a value of \(f_{\text{calc}}\) equal to the allowable maximum feed is set. In this condition, all lines of G-code, starting with the line where stock entrance occurs, are loaded into a temporary buffer until significant stock engagement occurs, or until the stock is exited again. If significant stock engagement occurs, the first value of \(f_{\text{calc}}\) that is less than the allowable maximum as a result of stock engagement triggers the buffer to be written out. This feed is multiplied by 0.5, the stock entrance inhibit amount, and attached as \(f_{\text{out}}\) to the first line of G-code stored in the buffer, which is then written out to the updated G-code file. The second line of G-code stored in the buffer then uses the full value of \(f_{\text{calc}}\) as its output feed, and the rest of the buffer is written out (thus all subsequent moves in the buffer are subject to \(f_{\text{calc}}\)). Feed calculation continues as normal once the buffer is output. If no significant stock engagement occurs before the stock is exited again, then the same output scheme is used with \(f_{\text{out}} = f_{\text{max}}\), the maximum allowable feed value.

In the second and more common stock engagement scenario, significant stock engagement occurs suddenly. In this case, the initial tool move with stock engagement uses \(f_{\text{calc}}\) multiplied by 0.5 as \(f_{\text{out}}\) for that move. It was found during testing that the feed should be gradually increased from this inhibited value, and so the 3 tool moves following stock engagement are also monitored. During these moves, the value of \(f_{\text{calc}}\) for the current move is only used as the value of \(f_{\text{out}}\) if it is less than \(f_{\text{out}}\) from the previous move; i.e. decreasing feeds are left unchanged. However, if during these moves the value of \(f_{\text{calc}}\) for the current move is greater than \(f_{\text{out}}\) from the previous move, inhibiting continues so that feeds increases gradually. When this occurs, \(f_{\text{out}}\) for the current move is set to \(f_{\text{calc}}\) for that move, minus 2/3 of the difference between the current value of \(f_{\text{calc}}\) and the value of \(f_{\text{out}}\) from the previous move; or:
\[ f_{\text{out}}(n) = f_{\text{calc}}(n) - 0.6666 \left( f_{\text{calc}}(n) - f_{\text{out}}(n-1) \right) \]  
\[ (3.12) \]

where \( f_{\text{out}}(n) \) is the output feed used during the current move (IPM),
\( f_{\text{calc}}(n) \) is the calculated feed for current move (IPM), and
\( f_{\text{out}}(n-1) \) is the output feed from the previous move (IPM).

The \( 2/3 \) amount was selected as it was found to provide a gradual increase over all three moves following stock engagement. Following the first four tool moves after stock engagement was detected, (including the original move where stock engagement first occurred), output feeds are no longer inhibited.

The second cause for feed inhibition is when large changes in the cutter direction of travel occur while the cutter is engaged in the stock material, as shown on the right in Figure 3.7. When a given tool move experiences a large directional change, the calculated feed value is inhibited if the output force value is greater than 10% of the desired force (i.e. inhibiting is ignored for very light cuts). The directional change is measured through calculation of the angle between the direction of travel vectors for adjacent tool moves. No feed inhibiting occurs for directional changes of less than 30°. If the angular change in direction lies between 30° and 60°, the feed inhibition amount is scaled linearly from 0.95 to 0.7 using:

\[ f_{\text{out}}(n) = f_{\text{calc}}(n) \left( 0.7 + 0.25 \left( \frac{60° - \delta_{\text{Dir}}}{30°} \right) \right) \]  
\[ (3.13) \]

where \( \delta_{\text{Dir}} \) is the angular change in direction between move \((n-1)\) and move \((n)\) (Deg).

For angular changes in direction greater than 60°, the maximum inhibition amount of 0.7 is used.

The final cause for feed inhibition is large changes in the output feed while the cutter is engaged in the stock material. This is achieved through the use of a user-defined parameter "MaxFeedChange", which represents the maximum allowable change in feed values between adjacent tool moves. If the difference between the previously output feedrate and the current output feed is greater than MaxFeedChange, then the current output feed value is re-set to be equal to the previously output feed value plus MaxFeedChange. The default value for MaxFeedChange is 20 IPM.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.8 Cutter Rotation Simulation

3.8.1 Cutter Rotation Simulation and Automatic Feedrate Selection

When using the mechanistic model, the force calculations take place at a series of discrete rotational positions, denoted \( \theta \), to simulate the rotation of the cutter. This rotation simulation is required to model variations in milling force as a function of rotation angle. The angular location of each flute segment is identified by a locating angle \( \beta \), which is a function of the current cutter rotational position \( \theta \), as well as cutter geometry. This angle is used to determine if a given flute segment is engaged in the stock at a given discrete cutter rotational position \( \theta \), as well as for chip thickness calculation.

As the flute locating angle \( \beta \) is a function of the specified cutter rotational positions and known flute geometry, a single finite set of repeated angular locations is required for cutter rotation simulation. Therefore, to improve the computational efficiency of the model, the values of \( \beta \) are pre-calculated for all flutes on all discs, and at all desired rotation angles. These values are stored in the \( \beta[j,k] \) array assigned to each axial disc (described in the following section). Also note that during the calculation of chip thickness using Equation (2.10) and milling forces using Equation (2.16), both the sine and cosine of the current \( \beta \) value is required. To further improve the computational efficiency of the model, these values are also pre-calculated and stored in the \( \beta[j,k] \) array.

3.8.2 Calculation of the Cutter Rotation Simulation Array

The \( \beta \) array is two dimensional, as the angles are calculated as a function of the current rotation angle (indicated by the index 'j'), and also of the current flute (indicated by the index 'k'). The effects of axial location and helix angle are included as a unique \( \beta[j,k] \) array is defined for each axial disc.

The array is filled using Equations (2.4-2.6, 2.8) (repeated below for completeness), starting with the bottom most disc \( (i=0) \), at zero rotation angle \( (j=0) \) and at flute zero \( (k=0) \). This location is arbitrarily defined as \( \beta=0 \), with all other angular locations being defined relative to this. The remaining flutes on the remaining discs then have their values of \( \beta \) calculated using:
\[ \beta[j,k] = \theta(j) + (k)(\gamma) + \tau(i) \] (2.5)

where \( \theta(j) = (j)(d\theta) \),

\[ \gamma = \frac{2\pi}{N_f}, \text{ and} \]

\[ \tau(i) = \sum_{z=0}^{N_f} (dZ(z)) \left( \tan(\alpha_{el}(z)) / R(z) \right). \] (2.8)

The flute index \( k \) is incremented from 0 to \( N_f-1 \) on each disc. Assuming runout is present, the cutter rotation angle index \( j \) is incremented from 0 to \((360 / d\theta) - 1\). The '1' is required as 0° and 360° are redundant. The value of \( d\theta \) in Equation (2.4), which defines the discrete rotational increment, is arbitrary and dependant on how 'fine' a rotational resolution is desired, as explained in the following section.

3.8.3 Cutter Rotation Simulation Optimization

To improve the efficiency of the feedrate calculation operation, it was desired to reduce the total number of angular positions that must be checked for peak force identification. However, it was also desired to maintain a given level of accuracy while performing this reduction. The actual number of force calculations present during a given cutter rotation simulation is largely a function of the number of discrete rotational positions present in the model, determined by the value of the rotational increment amount \( d\theta \).

By using a 'coarse' rotational increment, with a large angular change between each force calculation, the total number of force calculations required to cover the full 360 degree rotation may be minimized. However, neglecting the angles that fall between the coarse increments loses a great deal of force information. Conversely, the use of a fine rotational increment increases the amount of force data available for peak force identification, but at increased computational cost. Therefore it is desirable to develop an 'intelligent' cutter rotation simulation that uses variable rotational increments. This allows peak forces to be estimated with a minimum of force calculations at unique cutter rotation angles.

Based on observations of estimated force plots, it was determined that 1° rotational increments are fine enough to provide adequate coverage in identifying peak force values. The Figure 3.8 shows a comparison of estimated milling force plots (\( X_{CT}Y_{CT} \) magnitude) using differing rotational increments for a simple partial immersion cut. The conditions for the cut are listed in Table 3.3. The upper chart shows very fine 0.1° rotational increments plotted against 1° rotational increments, while the lower chart shows 0.1°
degree rotational increments are plotted against a coarser 8° rotational increment. Notice that the 1° increments follow the trace of the 0.1° rotational increments almost exactly, even in regions where a great deal of variation is present, such as the peak force locations. Compare this to the 8° increments, where the peak values in each of the force spikes are missed. However, notice that the 8° increments do a good job of tracking major trends in force variation. Based on these observations, an approach at peak force calculation that uses a variable rotational increment is utilized.

In this adaptive method, both coarse and fine rotational increments are used. First, the cutter is rotated 360° using coarse 8° rotational increments, with the force values calculated and the corresponding angles being written to an array. This requires 360/8-1, or a total of 44 force calculations (the ‘-1’ is used as force calculations at both 0° and 360° would be redundant). From this coarse rotation, a peak force value is identified. A second rotation with fine 1° increments then takes place, but it is limited to a range of +/- 40° about the peak force angle identified during the course iteration. However, note that the values from the coarse iteration are stored, and so force values for 11 of the 80 angular locations in the fine rotation are already known, resulting in only 69 locations that require additional calculations. Thus 1° resolution may be achieved using only 113 force calculations, as opposed to 359 force calculations if a full rotation were simulated at a 1° resolution.

<table>
<thead>
<tr>
<th>Cutter Type</th>
<th>Ball End</th>
<th>Axial Depth of Cut</th>
<th>0.1875'' (Ball Only)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cutter Material</td>
<td>HSS</td>
<td>Radial Depth of Cut</td>
<td>0.09375'' (1/4 Diam.), Down Milling</td>
</tr>
<tr>
<td>Cutter Diameter</td>
<td>0.375''</td>
<td>Spindle Speed</td>
<td>2200 RPM</td>
</tr>
<tr>
<td>Number of Flutes</td>
<td>4</td>
<td>Feedrate</td>
<td>2.8 IPM</td>
</tr>
<tr>
<td>Helix Angle</td>
<td>30°</td>
<td>Stock Material</td>
<td>6061 Al</td>
</tr>
<tr>
<td>FluteEnd</td>
<td>75°</td>
<td>KTC (Ball Value)</td>
<td>16139.6</td>
</tr>
<tr>
<td>Min Helix Angle</td>
<td>0°</td>
<td>KRC (Ball Value)</td>
<td>0.1387</td>
</tr>
<tr>
<td>Cutter Runout</td>
<td>0.00015''</td>
<td>P1 (Ball Value)</td>
<td>0.3729</td>
</tr>
<tr>
<td>Runout Locating Angle</td>
<td>143°</td>
<td>P2 (Ball Value)</td>
<td>0.1817</td>
</tr>
</tbody>
</table>

Table 3.3: Cutting Conditions for the force plots of Figure 3.8.

The location of the peak force value can vary on every tool move, so the array of flute locations β[i,j] must be calculated using the finer of the two rotational increments, in this case 1°, for a full cutter rotation. Also, the array β[i,j] is implemented such that the cyclical nature of rotations is considered. If an angular location less than 0° or greater than (360°-d0) is requested, the request should return the corresponding...
Figure 3.8: Force simulation plots showing milling force as a function of rotation angle for comparing different rotational increments. The cutting conditions are shown at top.
value that is bounded in this range. For example, a request for an angle of -12° should return the flute locating angle and trigonometric values for 348°, or a request for 393° should return 33°. This can occur during the fine iteration of ±40° about the peak values when the peak region identified is less than 40° from 0° or 360°.

Note that the values shown of 1° and 8° are arbitrary, and while they work well they are not required values. If values other than 1° and 8° are to be used, the course rotational increment should be an integer multiple of the fine rotational increment to maximize the optimization.

Finally, note that the location of the peak force can change with the value of input feedrate. Therefore a full coarse rotation must take place during each iteration, although efficiency gains would result from being able to use the same peak force location, or even peak force region, between all iterations for a given tool move. This is a result of the mechanistic constants $K_T$ and $K_R$, which identify the tangential and radial force components acting on the cutter, as they are a function of average chip thickness (which is a function of the current feed value). As the input feed value changes, the ratios of $K_T$ and $K_R$ also change. As both the tangential and radial components of force can contribute to the $X_{CT}$ and $Y_{CT}$ net force vectors, the change in ratio between $K_T$ and $K_R$ results in changes in the force values calculated at different locating angles. This effect is enhanced when the feed value changes dramatically between tool moves, but no generalized set of conditions that results in this occurrence have been identified.
CHAPTER 4

GEOMETRIC MODELING OF IN-PROCESS STOCK FOR MILLING FORCE ESTIMATION

4.1 Chapter Introduction

This chapter describes the techniques developed during this research to model the in-process stock geometry for milling force estimation and automatic feedrate selection. The in-process stock model defines the current geometric state of the stock during milling simulation, and requires updating after each tool move to reflect the changing geometry. In addition to providing an accurate geometric representation of the in-process stock, the geometric model also provides data for the calculation of cut geometry parameters required for mechanistic modeling.

4.2 In-Process Stock Geometric Model Requirements

The in-process stock model is central to any automatic feedrate definition scheme, as the geometry that defines any given cut must be known so that a feed may be defined to meet the current conditions. In order to provide this information for a given tool move, the state of the stock geometry as a result of all prior tool moves must be known. Therefore the ability to accurately represent the changing stock geometry over the course of a milling simulation is a basic requirement of the in-process stock model. The stock model should also provide data necessary for the calculation of the contact area, which defines the region of intersection between the cutter and stock for a given tool move.

Due to its critical role, the stock model must be robust as well as accurate. Without reliable, consistent, and accurate geometric simulation, optimized feedrate definition is not feasible. A geometric error could result in inaccurate feeds not only for the tool move during which the error occurred, but also on subsequent tool moves passing over the erroneous region. Practical considerations must also be taken into account, including ease of use, computational efficiency, flexibility in the definition of the “raw stock”, and model storage. There can be $10^3$ to $10^5$ tool moves in a typical part program, particularly for 5-axis and/or
sculptured surface machining, and so computational efficiency is critical. Also, the stock model should be able to accurately represent any raw stock form, e.g. a turning or casting. Finally, as many machining jobs consist of several operations, a means of storing the model to disc for later use is desirable.

As in most modeling situations, a balance exists between accuracy, efficiency, and robustness. While some such surface or solid based approaches can theoretically provide increased geometric accuracy as compared to discrete methods, they do so at reduced computational efficiency and robustness. Similarly, the magnitude of the discrete subdivisions in the approximate methods can be reduced to improve accuracy, but at an efficiency cost. In this balance, robust behavior is the most critical. There is a finite limit to the accuracy required, as a result of the accuracy the CNC machines are capable of, and the effects of cutter or part deflection. While there has been some successful research that includes cutter deflection for simple cutting conditions [SD86][LLB98][D93], no robust, simple methods with proven accuracy have been developed for generalized milling conditions. When the physical effects of positional accuracy, ball-screw backlash, cutter runout, axis runout, thermal expansion, part setup accuracy, and deflections are taken into account, the practical realizable accuracy is on the order of 0.001". Therefore, among the modeling approaches that can provide the minimum acceptable level of geometric accuracy while maximizing robustness, it is desirable to select the most computationally efficient approach.

4.3 In-Process Stock Geometric Model Selection

4.3.1 Solid Modeling Approach

One approach currently being researched for representing the in-process stock is solid modeling. In this approach, material removal is simulated by performing a Boolean subtraction of a solid model of the swept toolpath envelope from a solid model of the stock for each tool move. While this method may well prove to be a convenient and accurate means of performing geometric NC simulation in the future, efficiency and robustness issues need to be addressed before this method will be generally applicable.

In the solid modeling method, the stock and toolpath solid models are typically modeled using the boundary representation (B-rep) approach, the de-facto standard for solid modeling today [MSE98][T93][SA94][B96]. B-rep models are defined using a set of surfaces constrained such that all adjacent surfaces...
edges are joined at their common boundary, forming a closed volume. Attempts at NC simulation have also been made using constructive solid geometry, or CSG [VH81][W87][W88]. In this approach, the stock is defined as a finite set of individual geometric primitives. The B-rep models used in most commercial solid modeling software presently available have largely replaced the use of this method, as it provides less flexibility in the geometries it can efficiently and accurately represent.

The primary benefit of solid modeling is geometric accuracy. However, any actual increases in geometric accuracy of this approach versus the alternative approaches are difficult to demonstrate, particularly in light of the finite accuracy of the NC milling process itself. Additionally, while the solid modeling methods are fairly exact in theory, implementation details often result in approximations. Exact analytic solutions are not always possible for all milled surfaces, particularly if the condition of closure for the solid is to be met (the boundaries of all surfaces must join), resulting in approximate representations of the geometry for all but the simplest cases. Approximations are also often used to address memory and computational efficiency issues. The use of these approximations in solid modelers is mentioned in the solid model based NC simulation of Mounayri [MSE98]. It should be noted, however, these approximations are generally on the order of the theoretically achievable tolerances in NC milling, as the toolpaths are generally defined using CAD data represented in this form (the actual tolerances that may be obtained are variable and a function of the NC mill).

A more critical drawback to the use of solid modeling for generalized NC simulation is that it lacks robustness in large-scale applications. While solids based approaches work well in the definition of CAD models, NC simulation is beyond the scope of current technology. The in-process stock is much more complex than the design geometry. In a large design part definition there may be several hundred primary and joining surface definitions. NC milling simulation requires solid modeling systems to perform thousands of Boolean subtractions on a single part, potentially resulting in thousands of unique surfaces. Solid modelers must also ensure closure of the resulting solid while continually adding, subtracting, trimming, and connecting thousands of small surfaces. These requirements are beyond the scope of their current capabilities. Data management of large solid representations can also be error prone.

Computational efficiency is also a concern. While the computational effort of the Z-buffer approach is O(N), where N is the number of tool moves, the computational effort of solids based
approaches have been reported to be $O(N^4)$ [VH81]. While this may prove to be a worst case approximation, only under unusual conditions could solids based modeling approach $O(N)$. When milling a planar surface with a flat end mill, there comes a point where no additional surfaces are generated, and all new surfaces replace existing ones. At this point there is fixed number of surface operations per tool move, and the solids based method approaches $O(N)$. However, in most real-world milling operations involving complex shapes and different cutters, as the milling progresses new surfaces are continually being added to model the in-process stock, increasing the order of effort as more complex surface/surface intersections are required. In addition to process simulation, rendering of the in-process stock for graphical simulation becomes increasingly difficult and computationally expensive as the number of surfaces grows. While continual advances in computer technology will no doubt increase the performance of solids based approaches, all other methods will benefit as well.

Even with continuing advances in computer hardware and solid modeling technology, the method still has sufficient disadvantages that prevent practical implementation at this time. However, it may be possible that significant advances could one day make it practical. Therefore it should be noted that the overall layout of this software system is modular to allow easy replacement of any given portion.

4.3.2 Discrete Geometry Representation

There are many discrete methods available for the purpose of modeling the in-process stock. These methods include Octree method [RSR83], Voxel and Dexec approaches [S95][TWY97], and the highly successful Z-buffer style methods [V86][WW86][AEF87][JHDS89][JD91][E93] [CJ98]. One common benefit of discrete approaches is that they are computationally simpler than the solid modeling approach. Often, discrete methods require intersection calculations between simple geometric primitives, allowing simple and robust closed form solutions. This simplicity provides extremely robust behavior, and also results in excellent computational efficiency. One of the goals of this research is the development of a milling process simulation package suitable for use in an industrial five-axis machine shop environment, using current computer technology, and so the simple, robust nature of the discrete methods is desirable.

The primary disadvantage of these methods is their discrete representation of the geometry, which can lead to a loss of geometric accuracy. However, the error introduced can be controlled in many cases,
and therefore be limited to acceptable levels. Properly implemented, discrete methods can yield accuracy on the order of the physical tolerances on the actual machined part.

Another disadvantage of discrete approaches is that there is no "off the shelf" solution available for use in NC simulation. The relatively simple nature of CAD model definitions lends itself to solid modeling, and so many commercial solid modeling packages have been developed and marketed for product design. The APIs provided in these packages may also be used for NC simulation. There is no corresponding availability of APIs for discrete modelers, as the primary users of the technology are commercial NC simulators who do not divulge their proprietary techniques. Therefore the discrete methods require a substantial development effort. One benefit of this, however, is that the geometric modeler can be tailored to the needs of NC simulation and cut geometry data calculation.

The method selected for geometric stock modeling in this research is the extended Z-buffer approach [H94][CJ98], which is similar to Dexel and Ray Casting methods [MR92]. In this approach, the stock is modeled as a set of uniformly spaced parallel lines (see Figure 4.1). Material removal simulation is performed through calculation of intersections between these lines and a geometric representation of the swept toolpath envelope. The method is 'extended' as internal gaps may be stored in the model elements, while traditional Z-buffer methods allow only a single intersection (top) height. This is necessary for 5-axis milling simulation because the cutting tool orientation is variable relative to the stock (see Figure 4.2).

The extended Z-buffer geometric modeling method is very similar to the display Z-buffer used in graphics drivers for CRTs, and derives its name from this technology. Some Z-buffer models used for NC simulation actually use the display Z-buffer for model storage, e.g. the commercially available package 'Vericut' by CGTech of Irvine, CA. The advantage of this approach is efficient rendering, as the stock model is the rendered model, and no additional processing is required. The drawback is that control over geometric accuracy is diminished, as it becomes a function of the stock size, number of screen pixels, and part orientation on the screen. Definition using the graphics Z-buffer also prevents dynamic transformation of the stock model when the geometry is rendered on the screen, as the milling simulation must be repeated for the view to be changed.
Figure 4.1: In a Z-buffer model approach at NC simulation, the stock geometry is defined as a discrete set of uniformly spaced lines.

Figure 4.2: In traditional Z-buffers only the top of each element is stored (left). In an extended Z-buffer, multiple levels can be stored, as required for 5-axis simulation (right).

Figure 4.3: The Z-buffer model is defined to exist in the workpiece coordinate system.
While the graphics-based Z-buffer approach has its advantages for visual NC verification, it is not best suited for mechanistic NC simulation. For this purpose, a separate Z-buffer representation is preferred, as this allows for user-defined Z-buffer element spacing.

The Z-buffer approach is very robust, requiring only simple line intersections performed between the stock and a model of the volume swept by a single tool move (the 'swept toolpath envelope', or STE) to model material removal. The geometric form of the STE affects the overall complexity of the intersections, but line intersections define the simplest type of intersection with whatever form is used. The STE is defined in this research using a CSG approach, reducing all intersections to a finite set of line/ geometric primitive cases, as described in the following chapter. The robust behavior also carries over to model storage, both in RAM and on disc, as only a finite set of points in space indicating the bounds of each Z-buffer element, as well as any gaps they may contain, must be stored. Assuming each element extends in a Z direction, the X,Y location of each element may be implied by the index that accesses the element (as demonstrated in the next section), and so only a set of points in Z is required to define each element.

Another benefit of the Z-buffer approach is computational efficiency. Using this method, there is no appreciable decrease in performance as the simulation progresses. The total number of Z-buffer elements remains constant throughout the simulation, and so assuming that on average the same number of elements are intersected during any given tool move, the computational effort remains constant.

The robust and computationally efficient nature of Z-buffer modeling has resulted in it becoming the leading method for commercial NC simulation and process definition software. In addition to the previously mentioned Vericut package by CGTech, there are other commercial implementations of Z-buffer models. One of these is CHIPS (Computer Hybridized Interference Protection System), a 3 and 5-axis toolpath generation software package developed and used by Ford Motor Company for the production of body panel stamping dies. Other examples include Z-Master and Omega, toolpath generation and verification software originally developed by the research group of Byoung Choi at KAIST and now commercially implemented in Korea. Z-buffer models are also used in many commercial CAM packages for rendering, automatically identifying 'rest material' missed by the cutter during a milling operation, and primitive forms of volumetric material removal based automatic feedrate selection.
4.4 Definition of the In-Process Stock Geometric Model

4.4.1 The Z-buffer Model

In the extended Z-buffer approach, the stock is modeled as a set of parallel rays uniformly spaced in a square grid pattern. For convenience, this grid is defined in Cartesian 'Workpiece Coordinates', with the plane of the grid forming the X_{wp}, Y_{wp} plane, and the Z-buffer elements extending in the Z_{wp} direction (see Figure 4.3). When viewed from above along the -Z_{wp} direction, the lower left corner of this rectangle is defined as the X_{wp}, Y_{wp} origin. The Z_{wp} origin is defined as a X_{wp}, Y_{wp} plane relative to which all Z_{wp} values are specified, and only positive values are considered valid. Therefore in this implementation the Z-buffer model is defined to exist in a positively valued space, with the bottom, lower, left hand corner of stock existing at (0,0,0). The point where a given Z-buffer element contacts the Z_{wp} origin in the X_{wp}, Y_{wp} plane is referred to as its 'base position'. This positive space restriction simplifies the definition and use of the model. It also aids in the definition of non-rectangular stock models through the assignment of zero heights to elements that lie outside of the area actually occupied by the stock. These elements are read as NULL and are not included in any calculations. This positive space definition is purely arbitrary and may be discarded if needed, as the intersection calculations apply equally well in negative space.

The Z-buffer is referred to as “Extended” because multiple Z positions may be stored along a single Z-buffer element. The top-most Z value of a given element represents the top height of the stock at a given X_{wp}, Y_{wp} location, and any additional Z values (always present in pairs) represent “gaps” in the otherwise “solid” Z-buffer element. This is a requirement for 5-axis milling simulation, as the cutter is free to obtain any orientation relative to the Z-buffer elements. This allows the stock material to be undercut, as shown in on the right in Figure 4.2.

When defining the Z-buffer model, first a rectangular region representing the outermost bounds of the raw stock in X_{wp} and Y_{wp} are defined, using dimensions denoted D_{X} and D_{Y} respectively. A user-defined “mesh size” d_{XY} that indicates the grid spacing, or distance between adjacent elements in X_{wp} and Y_{wp}, is then used to define the required number of individual Z-buffer elements:

\[ N_{X} = \frac{D_{X}}{d_{XY}} \]  \hspace{2cm} (4.1)

\[ N_{Y} = \frac{D_{Y}}{d_{XY}} \]  \hspace{2cm} (4.2)
\[ N_{\text{net}} = N_x \cdot N_y \]  

(4.3)

where \( N_x, N_y \) are the number of Z-buffer elements in \( X_{wp}, Y_{wp} \), respectively, and \( N_{\text{net}} \) is the total number of elements in the model.

The total number of elements required, \( N_{\text{net}} \), is then allocated as an array in memory. Each element has a unique index into this array, and these indices are defined relative to the workpiece coordinates. The indices are numbered sequentially in the \( X_{wp} \) direction starting with 0 at the workpiece coordinate origin, and they are indexed by the amount \( N_x \) in the \( Y_{wp} \) direction (see Figure 4.4).

An area \( A_{XY} \) in the \( X_{wp}, Y_{wp} \) plane is associated with each Z-buffer element, and this may be calculated from the mesh size, as shown in Figure 4.4:

\[ A_{XY} = d_{XY}^2 \]  

(4.4)

Multiplication of this area with the net length of all Z-buffer elements intersected by the toolpath envelope during a given tool move yields the volume amount of material removed during that move. Dividing this volume by the move time value yields an accurate approximation of the average volumetric removal rate during that tool move. This information may then be applied in some of the simpler, volumetric-based feedrate calculation schemes, or for other purposes.

4.4.2 Individual Z-buffer Elements

Each Z-buffer element is composed of two components, a ‘Top’ value and a list of ‘Gaps’. The Top value represents the maximum height of the stock model at the \( X_{wp}, Y_{wp} \) location of that element. A “Gap” denotes a region in the element below the Top where a section of the Z-buffer element is removed. Gaps consist of a GapTop and a GapBottom, and also a pointer to the next Gap in the list (if any). Gaps are stored in order of decreasing \( Z_{wp} \). If there are no Gaps present in an element, the Gap pointer defaults to NULL; similarly setting the Next pointer to NULL within a given gap element denotes that it is the last element the Gap list (see Figure 4.5).
The Zbuffer model elements are separated by a distance $d_{xy}$ along the $X_{wp}$ and $Y_{wp}$ axes, and are indexed sequentially in $X_{wp}$ from 0, and by $N_X$ in $Y_{wp}$. Each Z-buffer element has an associated area, $A_{xy}$, but the element itself is a line that resides at the center of this area.

Each linked list element contains a top height, and possibly a linked list of gap elements.

The closed solid model STEs of this research always intersect a Z-buffer element at two locations. Only unusual, concave cutter shapes result in more intersections, shown at right.
The Z-buffer base positions are not stored to conserve memory, as they may be easily inferred from the unique index that identifies each element. Note an additional \((d_{XY}/2)\) amount is included in the position values during this calculation, as the elements are defined to lie at the center of the area they occupy. The \(X_{wp}\) and \(Y_{wp}\) base position for a given element may then be calculated from the index using:

\[
\text{Row}_X = \text{floor}\left(\frac{I_{Z\text{buffer}}}{N_X}\right) \quad (4.5a)
\]

\[
X_{wp} = (I_{Z\text{buffer}} - \text{Row}_X \times N_X) \times d_{XY} + d_{XY}/2 \quad (4.5b)
\]

\[
Y_{wp} = \text{Row}_X \times d_{XY} + d_{XY}/2 \quad (4.5c)
\]

where \(\text{Row}_X\) is the current \(X_{wp}\) row of elements that the index resides in, and \(I_{Z\text{buffer}}\) is the Index to the current Z-buffer element (beginning with 0).

### 4.4.3 Updating the Z-buffer Model and Intersected Segment Definition

"Updating the Z-buffer Model" refers to the methods used to alter the Top and Gap values as a result of intersection calculations between the Z-buffer elements and the STE, and is required for material removal simulation. While material removal simulation may be generically classified as a Boolean subtraction, some implementation details will now be provided. These details are critical to efficiency and robustness, and also describe the definition of intermediate data structures, ‘intersected segments’, required for efficient calculation of the cut geometry parameters. Note that the scope of updating the Z-buffer model is limited to a single element. Intersection location(s) are found between the toolpath and a single element, which is then updated prior to intersection calculations with the following element.

For simple, single setup 3-axis milling simulation, only a top height value exists for each Z-buffer element. If an intersection location is found that lies below the current top height of a Z-buffer element, the existing Top value is replaced by the new, lower \(Z_{wp}\) value. This process of comparing relative heights is referred to as ‘intersection validation’, as many intersections that are calculated lie above the current top height, and are therefore invalid. Intersections found to be invalid are ignored; only valid intersections are used in updating the Z-buffer and in contact area calculation. During 3-axis simulation, every intersection calculation results in either zero or one valid intersections, and an accurate representation of the in-process stock is kept by continually updating the top height values of the model with all valid intersections found.
For the case of generalized 5-axis milling, zero, one, or two valid intersections may occur for standard, non-concave cutting tools. The toolpath modeling methods are specifically designed to simplify the task of identifying valid intersections. As presented in Chapter 5, the volume swept by each tool move is represented as a discrete, error controlled, three-axis approximation, modeled using a closed solid representation (the STE). This assures two intersections between the toolpath and a Z-buffer element will occur, one where the infinite line modeling the Z-buffer element enters the STE, and one where it exits, as shown on the left in Figure 4.6. For most common cutter geometries there will be only two intersection locations, but for concave geometries, such as some routers or shapers, there could be more, as shown on the far right in Figure 4.6. However, notice that even in this case the intersections will always exist in pairs, and the methods presented here may be applied to each intersection pair found. During 5-axis milling simulation the conditions for intersection validity are that the intersection occurs below the top of a Z-buffer element, and that it occurs within a solid portion of the element. Although both intersections between an STE and a Z-buffer element may occur below the top, they may occur either partially or wholly within an existing gap in the element, resulting in zero or one valid intersections.

From a given pair of intersections between the Z-buffer element and the STE, an “intersected segment” is defined. The intersected segment defines the portion of Z-buffer element that lies entirely within the STE, and represents the material removal contribution from that element. While the intersected segment is eventually subtracted from the Z-buffer model, prior to this it is used to improve the efficiency of cut parameter calculations. The entire intersected segment is known to contact the cutter, which simplifies the calculation of volumetric removal rate and contact area.

As the intersected segment lies along $Z_{upp}$, it is convenient to think of the bounds as being “higher” or “lower”. The upper and lower ends of the intersected segment are initially set to the calculated upper and lower intersection locations. After this initial segment has been created, the ends of this segment are adjusted so that both are valid, defining the actual intersected segment. This is achieved by testing for 3 primary cases, and several sub-cases:
1) *The lower intersection lies above the top height of the element (see Figure 4.7):*

In this case the current element does not intersect the STE, and so no intersected segment is defined and the Z-buffer element requires no updating.

2) *The lower intersection lies below the top height, and the upper intersection lies above it (Figure 4.8):*

In this common case, the upper end of the intersected segment is defined as the current element top height. The lower end is checked for validity and redefined as necessary to specify the final intersected segment, which encompasses two sub-cases. In either case, existing gaps that are interior the region occupied by the intersected segment are included in it and deleted from the Z-buffer element.

2a) If the lower intersection is found to reside in a solid section of Z-buffer, the definition of the ends of the intersected segment is completed by assigning its lower end the value of the lower intersection. The Z-buffer model is then updated by redefining the top height of the element to the lower intersection value.

2b) If the lower intersection is found to reside in a gap, the definition of the ends of the intersected segment is completed by assigning its lower end the value of the gap top. The Z-buffer model is then updated by assigning the top height of the element the value of the gap bottom, and that gap is deleted from the model.

The element is now completely updated, and the intersected segment defined.

3) *Both the upper and lower intersections lie below the top height (Figure 4.9):*

In this case the two intersections either form a new gap or expand an existing gap in the element, which encompasses five sub-cases. In any case where an intersected segment is formed, existing gaps that are interior to region occupied by the intersected segment are included in it and deleted from the Z-buffer element.

3a) If both the upper and lower intersections lie completely within a single existing gap, the current element does not intersect the STE, and so no intersected segment is defined and the Z-buffer element requires no updating.

3b) If neither the upper nor the lower intersections lie within an existing gap, the intersected segment is defined by assigning to its upper and lower ends the values of the upper and lower intersection.
Figure 4.7: This figure shows case 1, where no valid intersections occur, no Intersected Segments are created, and no updating of the Z-buffer model is required. Also shown is a reference chart that also applies to Figures 4.8 and 4.9.

Figure 4.8: This figure shows the two possible situations of case two, where either one (left) or no (right) valid intersections exist. In either case an intersected segment is created. If created, the intersected segment is subtracted from the model after being used in the calculation of the cut geometry parameters.

Figure 4.9: This figure shows the five situations of case three (see text). Only in case 3a is no intersected segment created, and only cases 3a and 3e have no valid intersection locations. These define the specific cases for a boolean subtraction of an interior Z-buffer segment.
locations respectively. The Z-buffer model is then updated through the definition of a new gap whose upper and lower ends are assigned the upper and lower intersection values respectively.

3c) If only the upper intersection lies within an existing gap, the intersected segment is defined by assigning to its upper end the value of gap bottom, and to its lower end the value of the lower intersection location. The Z-buffer element is then updated through gap expansion by redefining the gap bottom to be the value of the lower intersection (the gap top remains unchanged).

3d) If only the lower intersection lies within an existing gap, the intersected segment is defined by assigning to its upper end the value of the upper intersection location, and to its lower end the value of the gap top. The Z-buffer element is then updated through gap expansion by redefining the gap top to be the value of the upper intersection (the gap bottom remains unchanged).

3e) If both the upper and lower intersections lie within separate existing gaps, the intersected segment is defined by assigning to its upper end the value of the upper gaps bottom, and to its lower end the value of lower gaps top. The Z-buffer element is then updated through gap expansion by redefining the upper gap bottom to be the value of the lower gaps bottom, and the lower gap is deleted from the model.

The element is now completely updated, and the intersected segment defined.

4.4.4 Estimation of Volumetric Material Removal Rate

Once the intersected segment has been defined, calculation of the net length of 'solid' Z-buffer element removed is a trivial operation. This is done by subtracting the lower end of the intersected segment from the upper end to obtain the gross length removed, and then subtracting from that value the net length of all gaps present in the intersected segment:

\[ L_Z = E_u - E_l - L_G \]  

(4.6)

where \( L_Z \) is the net length of Z-buffer element removed (in),

\( E_{ul} \) are the upper and lower ends of the intersected segment (in), and

\( L_G \) is the sum of the lengths of all gaps that lie within the intersected segment (in).
L_z is useful as it may then be used to calculate net volume of material removed during a given tool move:

$$V_M = \sum_{i=1}^{N_M} A_{xy} L_{z_i}$$  \hspace{1cm} (4.7)

where $V_M$ is the volume removed during the current move (in$^3$), $N_M$ is the number of elements intersected during the move, $L_{z_i}$ is solid length of the the $i$th intersected segment (in), and $A_{xy}$ is the area associated with a single z-buffer element (in$^2$).

This in turn is used in calculating a volumetric removal rate:

$$VRR = V_M/dT_M$$  \hspace{1cm} (4.8)

where $VRR$ is the volumetric removal rate (in$^3$/min), and $dT_M$ is the time required to complete the current tool move (min).

The CNC machine model, described in Chapter 6, is used to calculate the move time $dT_M$. The value of $VRR$ provides a simple means of roughly approximating the current load on the cutter, as well as providing a means of estimating power requirements.

4.5 Geometric Model Verification

As the geometric model creates and stores the in-process stock geometry, graphical rendering of this geometry provides a convenient means of validating its accuracy. The stored geometry should look identical to the geometry physically created when the part is milled, with any unexpected visual variations indicating error in the geometric model. This error can be the result of known modeling approximations (e.g. faceting in vertical walls due to the discrete Z-buffer spacing), or software 'bugs'.

Rendering of the stock geometry is simplified by the ready availability of computer graphics APIs, which are typically resident on most computer systems for running the graphics applications built into the
operating system. The most common API currently used today is the OpenGL library, developed from the GL "Graphics Language" originated by Silicon Graphics Inc. The dominance of OpenGL is a result of it being bundled with the Microsoft Windows 95/98/2000/NT operating systems, and so it is present on most 'PC' based systems available today. These graphics libraries also have the benefit of platform independence, and will also run on UNIX-based systems provided the OpenGL libraries in X windows form, GLX, are present. In addition to the basic lighting operations, OpenGL also provides other techniques required for realistic rendering, such as hidden surface removal and shading models (required for a true shaded image, rather than simply a 'flat lit' model with no blending of reflectance between polygons). There are also additional higher-level APIs available. These simplify the use of OpenGL by predefining commonly used routines from the lower level OpenGL functions, such as the OpenGL utility library (GLU), and the Open Inventor Toolkit. Also available are APIs that aid in interacting with the rendered images by providing event loops and callback registration to manage user inputs, such as Glut. This is useful for actions that require user interaction such as 3D transformation of the rendered object (translation, rotation, and scaling), pausing during animation, displaying a wireframe vs. a shaded image, displaying the cutter shape rather than the stock geometry, etc. There is a great deal of literature currently available regarding the features and application of OpenGL, the two most widely used of which are the texts of Neider et. al. [NDW93] and Kempf et. al. (Ed.) [KF96].

The OpenGL libraries operate using standard computer graphics rendering techniques (one source of information regarding this topic is the text of Foley et. al. [FDFH90]). These techniques are based on the concept of a lighting model, in which shaded images of the object to be rendered are generated by calculating the light reflected by the object to the viewing screen from some light source. Therefore, assuming a light source and viewing direction have been defined, what is required for rendering an object is a surface definition for that object, and a description of its optical qualities (color, opacity, diffusivity, etc.). In OpenGL, as in most graphics applications, surface data is required in a discrete manner via a tessellated surface representation consisting of a set of polygons, with normal vectors being required at the polygon vertices. The Z-buffer model is well suited to supplying the basic input for graphical rendering, as it consists of a set of points in space. By limiting the display surface to a single layer of the Z-buffer model (e.g. only the top heights), these points may be used as the vertices of the required polygons. Note that
when gaps exist they are not correctly rendered in this approach, but as graphical rendering was implemented primarily as a validation tool, the sophisticated meshing required to render gaps is beyond the scope of this research.

The polygonal shape used in this application is quadrilaterals, as this is the natural shape of the $X_{wp}$, $Y_{wp}$ grid of Z-buffer elements (see Figure 4.10). Note that each quadrilateral could be bisected diagonally to form a list of triangles, but this would double the number the polygons present in the rendering, which would detrimentally affect performance (the computational effort of the lighting calculations is a linear function of the number of polygons). While this can result in non-planar surfaces, Gouraud shading is used to ensure proper blending of the light intensity across the polygon.

After the quadrilaterals have been defined, the resultant geometry may be viewed as a meshed surface (see Figure 4.11). For the lighting models to be applied, normal vectors must be calculated for each of the quadrilaterals. Also, as Gouraud shading is to be applied, an average normal vector must be defined at each verticie for each quadrilateral. This is performed as follows: For each node, a normal is calculated for each adjoining quad by taking the vector cross products of the polygon sides. The total number of vectors calculated for a given node is then averaged by taking the vector sum of the individual normal vectors and normalizing the result. This is shown in Figure 4.10 on the right, using the ‘right hand rule’ to ensure a proper outward normal is calculated (note that the normals calculated are not shown in this figure, only sides of the polygons used to calculated them). When this has been performed for all nodes in the model, a smooth shaded rendering of the geometry may be displayed by application of the appropriate lighting model, as shown in Figure 4.12.

Note on the left in Figure 4.12 an image of the cutting tool is present. As the Z-buffer model is updated for every tool move, and the cutter location relative to the Z-buffer model is also known, an animated display showing the in-process milling may be created. This is performed by updating the quadrilaterals (and normal vectors) that have their nodal positions changed as a result of the intersection calculations during the current move, and re-displaying the model. Repeating these operations for every tool move, and displaying a geometric model of the cutter in the appropriate known location, allows for animated viewing of the milling simulation. This can prove useful when debugging the geometric model.
Normal Vector $\mathbf{V}_{11}$ for Node 11 is the average of the normals from this node for all surrounding quads ($\mathbf{V}_{11,\text{Quad}}$):

\[
\mathbf{V}_{11,4} = (\mathbf{n}_{10} - \mathbf{n}_{11}) \times (\mathbf{n}_6 - \mathbf{n}_{11}) \\
\mathbf{V}_{11,5} = (\mathbf{n}_6 - \mathbf{n}_{11}) \times (\mathbf{n}_{12} - \mathbf{n}_{11}) \\
\mathbf{V}_{11,9} = (\mathbf{n}_{12} - \mathbf{n}_{11}) \times (\mathbf{n}_{16} - \mathbf{n}_{11}) \\
\mathbf{V}_{11,8} = (\mathbf{n}_{16} - \mathbf{n}_{11}) \times (\mathbf{n}_{10} - \mathbf{n}_{11})
\]

\[
\mathbf{V}_{11} = \| \mathbf{V}_{11,4} + \mathbf{V}_{11,5} + \mathbf{V}_{11,9} + \mathbf{V}_{11,8} \|
\]

Figure 4.10: The Z-buffer model is easily rendered by creating a grid of nodes ($n_i$) that are used to define a list of connected quadrilaterals ($Q_i$). Normal vectors are then defined for each node as the average of the normals that are found with that node for each quad.

Figure 4.11: Once the quadrilaterals are defined, a meshed grid may be displayed. For the shaded image to be rendered, normal vectors for each node on each quadrilateral are required.

Figure 4.12: When the polygons and normals have been defined, a lighting model may be applied to render a shaded image of the geometry. The left image is a frame-grab from an animation.
Only an overview of the procedure is presented here. For a more rigorous description of the full procedure used to render Z-buffer models, refer to [H94]. In that application, triangles are used for the polygons and flat shading as opposed to smooth shading was implemented, and so only one normal per polygon was required. However, many of the required operations, such as the basic definition of individual (not averaged) outward normal vectors and the definition of the sides of the stock model, are identical.

4.6 Custom Z-buffer Model Definition

The basic Z-buffer model previously described defines the raw stock as a rectangle in \( X_{wp}, Y_{wp} \) with a constant height in \( Z_{wp} \). While this case is not uncommon, raw stock is also frequently in the form of a casting, or has had some previous operation performed on it such as turning or prior milling. For accurate material removal simulation in these cases, it is necessary that the raw stock be accurately represented.

In the case of this research, the primary non-rectangular stock of interest was lathe turnings. Our industrial research partner primarily manufactures bladed turbine and compressor wheels, which are manufactured though the 5-axis milling of a lathe turning that has a geometrically correct outer profile of the finished part (see Figure 4.13). Therefore a model of a lathe turning is required for the first milling operation; for all subsequent milling operations the raw stock is the lathe turned stock minus the material removed by all prior-milling operations.

To model a lathe turning, the raw stock defined as a surface of revolution using a 2D 'lathe profile' NURBS curve (see Figure 4.14). The maximum radius of the part, \( R_{max} \), may be determined from the lathe profile. Many lathe profiles also have a minimum radius value, \( R_{min} \), as the center of the wheel is typically bored out. The shape of the profile is known as it is required to generate the actual turned part. The turning may be modeled by defining a location in \( X_{wp}, Y_{wp} \) for the central axis, which extends in \( Z_{wp} \) to reside. This location is referred to as the 'rotational origin', denoted in Workpiece Coordinates by \( X_{RO}, Y_{RO} \). After the lathe profile and rotational origin are defined, it is relatively simple to create a 3D Z-buffer model for the turned part. This is done by looping sequentially through all Z-buffer elements, defining for each element a radius \( R_{wp} \) in \( X_{wp}, Y_{wp} \) from the rotational origin to the base position of the current element:
Figure 4.13: A bladed impeller or compressor wheel consists of a series of uniformly shaped blades and cavities about a central hub. Only the cavities are milled out to create the blades, the outer profile of the part results from a lathe turning used as raw stock.

Figure 4.14: A lathe profile may be used to generate a Z-buffer based 'surface of revolution' defining the raw stock geometry for a bladed compressor or turbine wheel.

Figure 4.15: Although a full geometric model of the raw stock may be generated (left), it's more computationally efficient to model only the portion required to mill one complete blade.
\[ R_{WP}(i) = ( (X(i) - X_{RO})^2 + (Y(i) - Y_{RO})^2 )^{1/2} \]  \hspace{1cm} (4.9)

where \( R_{WP}(i) \) is the radius to the \( i \)th Z-buffer element from the rotational origin (in), and
\( X(i), Y(i) \) denote the base position of the \( i \)th element in Workpiece coordinates.

This radius, along with the lathe profile, is then used to find the corresponding height (\( Z_{wp} \)) of the stock at that radius value, as shown in Figure 4.14. If the value of \( R_{WP}(i) \) is greater than \( R_{Max} \) or less than \( R_{Min} \), a value of 0 is returned, indicating a "NULL" Z-buffer element.

The location of the rotational origin and the \( X_{wp}, Y_{wp} \) dimensions of the raw stock are defined as follows. As the value of \( R_{Max} \) is known, the size of the required raw stock may be defined automatically. To simulate the entire turned stock, \( D_x \) and \( D_y \) (the \( X_{wp} \) and \( Y_{wp} \) dimensions of the raw stock) are both set equal to \( 2(R_{Max} + d_{xy}) \). The additional mesh distance is added to ensure the lathe profile lies within the bounds of the discrete model. The rotational origin is then defined at the stock center, and so \( X_{RO} \) and \( Y_{RO} \) equal \( (R_{Max} + d_{xy}) \). This results in the geometry shown on the left in Figure 4.15. However, modeling of the entire turned stock is detrimental to computational performance, as the majority of the Z-buffer elements are redundant. Typically, only a single blade requires simulation as a result of part symmetry.

The turbines (and compressors) are manufactured by milling away all material from the turnings that occupies the cavities between adjacent blades. The material bounding the cavities on the finished part consists of blade surfaces on either side, and also the hub surface. As all blade shapes on the wheel are uniform, all cavity geometry is also identical. Therefore the part programs (roughing, finish milling, etc.) need only be defined to produce a single blade shape and cavity, which are then repeated about the perimeter of the stock to arrive at the finished part. Therefore a geometric model is required for only a fraction of the net part, as feeds require definition only once per part program. Note this requires the milling of two cavities, so both sides of a single blade are defined, as shown on the right in Figure 4.15.

Ideally in this application, the Z-buffer elements would be distributed in a polar coordinate system. This would allow Z-buffer elements to be defined only over the radial range of the actual lathe profile, and also only over the angular range of two cavities, calculated as:
\[ d\lambda = 2 \left( \frac{2\pi}{N_b} \right) \]  

(4.10)

where \( d\lambda \) is the angular range covered by two milled cavities (one complete blade) (rad), and \( N_b \) is the number of blades on the part.

While this would minimize the number of Z-buffer elements required, note that it results in variable mesh spacing. As the Z-buffer elements are distributed at uniform \( \lambda \) and \( R \) increments, the linear distance between the vectors will increase in the \( \lambda \) direction with increasing values of \( R \), as shown in Figure 4.16. While methods could be developed to increase the Z-buffer density as a function of radial distance, the mesh spacing would still be variable. The Z-buffer management code developed during this research does not support variable mesh spacing, as it is required only for a subset of all possible parts; however, implementation of such an approach could be beneficial for the dedicated simulation of turbomachinery.

To maintain compatibility with all supporting software developed for a generalized milling situation, the uniform spacing of the Z-buffer elements is maintained. This also simplifies the software, resulting in greater reliability. Using this constraint, the values \( D_X \) and \( D_Y \) may be defined as:

\[ D_X = S_X (R_{\text{Max}}) + 2d_{XY} \]  

(4.11a)

\[ D_Y = S_Y (R_{\text{Max}}) + 2d_{XY} \]  

(4.11b)

where \( S_X \) and \( S_Y \) are user defined scaling values that default to a value of 1.0.

While the stock size is variable, the rotational origin is permanently fixed in space, always defined at:

\[ X_{RO} = Y_{RO} = (R_{\text{Max}} + d_{XY}) \]  

(4.12)

The default situation provided by Equations (4.11) and (4.12) is that 1/4 (90°) of the total turned stock is modeled. This is generally sufficient for most parts, which have 8 blades or more (refer to Equation (4.10)). If this size is insufficient, \( S_X \) and/or \( S_Y \) must be increased to ensure all milled material is reflected in the stock model (\( S_X = S_Y = 2.0 \) models the entire stock). Similarly, when the part contains more than 8 blades and \( R_{\text{Min}} \) is non-zero, \( S_X \) and/or \( S_Y \) may be decreased to minimize the size of the modeled stock.
Figure 4.16: While polar coordinates could more efficiently model turned stock, it results in non-uniform mesh spacing. This also results in a variable area represented by each element, indicated by the size of the '+' above.

Figure 4.17: To maintain a constant mesh spacing, turned stock is defined in standard Workpiece coordinates. The model on the left has $S_X = S_Y = 1$, while on the right $S_X = S_Y = 0.9$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In the case of castings, a similar method may be employed, although not implemented in this research. For 3D castings, the stock model can be defined from the 3D CAD model used to define the casting, in the same manner that the 2D “lathe profile” is used to define a surface of revolution. In this approach, the 3D CAD surface data is used to identify the required $Z_{wp}$ value for each Z-buffer element. This is performed by mapping the $X_{wp}, Y_{wp}$ grid of Z-buffer base positions onto the surface data, and obtaining the corresponding $Z_{wp}$ values for each Z-buffer element. This querying of CAD data may be performed using ‘macro’ utilities provided in some CAD package, or by using CAD vendor supplied APIs that allow data access or even perform the desired function (returning a Z value for an input X,Y coordinate), or it may require development of dedicated software to perform the operation if no other means is provided. Many CAD packages today provide some form of “open architecture”, allowing the attributes of the CAD surface data to be accessed.

In the event that a primitive CAD package is used that does not provide any of the above functionality, or the CAD data for the raw stock is unavailable, it is still possible to define a Z-buffer model of the geometry. This is achieved by simulating without feedrate generation the first milling operation(s) that remove the outer layer of material from the stock, using a standard rectangular Z-buffer model. This would be done strictly for the purpose of defining the starting geometry for subsequent operations, which then could have feedrates automatically generated. If this software is used by the part-programmers, an initial operation that mills away a thin layer of material from the part could be intentionally programmed to support this approach. This approach could also be used to simulate parts where the ‘raw stock’ is the result of previous milling operations.

4.7 Z-Buffer Model File Storage

As many machining jobs consist of several milling operations, a means of storing the Z-buffer stock model in a file format is desirable. This simplifies re-calculation of feeds if changes are made in any given operation by localizing the simulation to only the affected operation. It also provides a convenient means of sharing and archiving geometric machining simulation data. Currently the Z-buffer data file is
written out in ASCII format. However, if used regularly it is recommend that this be changed to a binary format. This will read and write to disc more efficiently, and require less disc space.

The task of reading and writing the geometric model to disc is easily accomplished with a Z-buffer style model. Prior to writing out the actual elements, first a file header containing the values of $d_{xy}$, $N_{net}$, $N_{x}$, and $N_{y}$ is written out. The value of $N_{net}$ is required so that the correct amount of memory may be allocated when reading in the Z-Buffer file. The values of $d_{xy}$ and $N_{x}$ are required so that the base positions of each element can be calculated using Equation (4.5), and $N_{y}$ is included for completeness.

The Z-buffer model consists of a sequential list of elements, and on each element is a sequential list of $Z_{wp}$ locations. Therefore the task of writing the Z-buffer model to a disc file is simply a matter of looping through the elements, and for each element performing an inner loop to writes out the $Z$ locations it contains. A delimiter is used to separate unique elements, but the locations on each element are written out in a known format so no delimiter is required.

As the data being written to the file is numeric, and all valid Z-buffer data exists in a positively valued space, currently a '-1' is used for the delimiter. For each element, first the top height is written out, followed by any gaps that may be present in that element. The gaps are written out in order of decreasing $Z_{wp}$ location. A sample file segment appears below, with comments (that aren't truly in the file) in italics.

<table>
<thead>
<tr>
<th>0.0075 720000 1200 600 (file header: $d_{xy}$, $N_{net}$, $N_{x}$, $N_{y}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.4565 -1 5.4563 -1 5.4565 -1 5.4558 -1 5.4555 -1 5.4553 4.4833 4.0241 3.2454 3.0214 -1 5.4548……</td>
</tr>
<tr>
<td>Top1  Top2  Top3  Top4  Top5  Top6  gap1  gap2  Top7</td>
</tr>
</tbody>
</table>

Reading in a Z-buffer file is equally simple. As the exact number of elements is known from information in the file header, the memory for the individual elements, not including gaps, is first allocated in bulk. The elements are then read in sequentially, just as they were written out. For each element, all $Z$ locations in the element are read until a '-1' is encountered, with gap elements being dynamically allocated and added to the gap list as necessary. The first location read is the top height. If another positive value follows that, it is known that a gap is present and memory is allocated. The gap is known to contain both an upper and lower position, and both are read in and attached to the end of the list. This routine of allocating gaps and reading in the upper and lower positions then continues for all gaps in the element.
4.8 Z-buffer Element Spacing and Computer Memory Issues

The definition of $d_{wp}$, the Z-buffer element spacing, has a direct effect on the accuracy of the geometric simulation, as well as on the calculation of the geometric parameters. It is very difficult to arrive at an analytical solution for an ideal value of $d_{wp}$ for 5-axis milling. Ersoy presents an equation for mesh spacing based on an allowable error and the tool move length, but this was developed to limit the error in volume removal calculation during 3-axis milling with a flat end cutter [E93]. While reducing error in the volumetric simulation will indeed control the overall error in the model, it becomes more complicated in the case of 5-axis milling. The Z-buffer elements are discrete in $X_{wp}$ and $Y_{wp}$ and continuous in $Z_{wp}$, so when the cutter is tilted relative to $Z_{wp}$ (or when 3-axis milling with a ball end cutter), the material removal simulation is continuous regardless of the distance moved. This is shown in the simplified 2D view of Figure 4.18. This makes it difficult to accurately apply Ersoys equation, as the volumetric removal error is no longer strictly a function of tool move length. Additionally, this error calculation is based on volumetric considerations, while the force model of this research relies more on geometric accuracy. The entrance and exit angles that bound the contact area defining the region of contact between the cutter and stock must be accurately calculated from the Z-buffer element spacing provided.

It is virtually impossible to relate force estimation error to mesh spacing. The tilt of the cutter in the direction of travel and about the direction of travel, the cutter type, the axial and radial depths of cut, the $X_{wp}, Y_{wp}$ direction of travel, and the distance traveled all affect the error produced by a discrete stock model. An example of how some of these dependencies affect the radial distance calculation is shown in Figure 4.19. These conditions are all interdependent and predicting error accurately as a function of some of these parameters is only possible when some of the other variables are fixed. The actual combination of parameters present during any given mode is highly dependent on the current cutting conditions, which are themselves variable. While it is possible to fix a set of conditions and assume a worst case scenario, this generally results in a required Z-buffer element spacing that is unrealistically small, at least for practical implementation on current generation computers. While this will produce a mesh size that is capable of dealing with the worst-case condition analyzed, it is generally far too conservative for the general situations encountered. An example of such an analysis is presented in Appendix C.
Figure 4.18: For a 3-axis flat end operation, $D_{\text{Min}}$ represents the minimum $X_{\text{WP}}, Y_{\text{WP}}$ move distance that must occur for new material to be intersected (i.e. $D > D_{\text{Min}}$) (left). As shown on the right, continuous material removal may occur for a ball end mill or a 5-axis flat end operation.

Figure 4.19: The radial distance values found are a function of the start position in both $X_{\text{WP}}$ and $Y_{\text{WP}}$, the distance traveled to the end position, and the direction of travel. This is a result of the variable 'effective mesh spacing' seen orthogonal to the direction of travel.
Due to the difficulty in analytically defining an ideal mesh size that provides adequate performance with regards to both performance and accuracy, a more 'rule of thumb' based approach has been implemented. In this approach, the smallest feasible spacing distance for a given stock size is used, based primarily on the available memory of the computer system on which the simulation is being run. The rule of thumb is that the mesh size be at most $0.1R$, where $R$ is the smallest cutter radius used for a given milling simulation. Accurate force estimates have been achieved with this spacing. In general a smaller mesh size is implemented if possible, as this will always yield more accuracy over a broader range of cases.

As mentioned above, the main impediment to allowing for extremely small values of $d_y$ on geometrically large stock models is the amount of RAM that is available to store the model. The current design of the Z-buffer structure is intended to be memory efficient. Inferring the base positions, rather than storing the values, immediately reduces the minimum memory required by almost $2/3$. Also, since machining accuracy levels generally do not extend past 4 decimal places, and the Z-location values are not solved for in a highly iterative manner, machine round-off error is not a concern and the use of double precision values is not required. Therefore 4 byte floating-point values may be used for the top height and gap locations. Also, by storing gaps as a linked list with a pointer to the first gap, only a 4 byte pointer is required when no gaps are present, and so each element requires 8 bytes as a result. This translates into 320 Kilobytes per square inch of modeled stock with a fairly conservative 0.005" mesh size (roughly the thickness of a piece of paper). A final means of conserving memory is in the gap definition. As the gap contains both the upper and lower end positions, plus a pointer to the next gap, each requires 12 bytes. This is superior to the method of using a linked list of individual locations, where both the top and the bottom of the gap are separate elements in the linked list, as this method would require 16 bytes per gap. However, it still demands a good deal of memory. If it is assumed that on average every element in a given model will contain one gap, the elemental memory requirement is now 20 bytes per element. This translates into 800 Kilobytes per square inch of modeled stock with a fairly conservative 0.005" mesh, or 1.12 square inches per Megabyte. In general, the nominal memory requirements for a Z-buffer model may be calculated as:

$$M_{net} = \left( \frac{(D_x)(D_y)}{d_y^2} \right) M_{element}$$ \hspace{1cm} (4.13)

where $M_{net}$ is the net memory required for all Z-buffer elements (Kilobytes), and
$M_{\text{element}}$ is the memory required for a single Z-buffer element (Kilobytes).

Not surprisingly, this shows that increasing mesh spacing or decreasing stock size reduces the memory requirements. It also shows that increasing mesh spacing has a greater effect than decreasing stock size, except in instances where both sides may be reduced by the same amount. This indicates the benefit of minimizing the modeled stock to the absolute smallest area required for accurate simulation, as well as the costs incurred by greatly reducing the mesh spacing.

If large, finely meshed models are required, the Z-buffer could be further optimized. The $Z_{wp}$ height values, currently stored as part of a continuum, could be discretized and stored as short integers requiring only 2 bytes. This is achieved by dividing the range of real valued $Z$ values required for the model by the number of discrete integer values provided by a short integer. As the workpiece space is defined as having no negative coordinate values, unsigned short integers may be used to represent the $Z$ positions. Unsigned integers include all integers between 0 and 65536, which should provide adequate $Z$ resolution in most cases for the purpose of cut geometry calculation. For example, a model requiring a relatively large $Z$ range of 36" would have a resolution of 0.00055" $\left( \frac{36"}{65536 \text{ values per short int}} \right) = 0.00055$. This is more than sufficient, as it is on the order of magnitude of the absolute accuracy of the best CNC machines. If this approach were implemented in a purely 3-axis simulator (no gap pointers required), it would require only 80K per square inch of stock. Therefore a relatively large 3'x4' piece of stock, such as a stamping die, could be modeled to 0.005" resolution using only 138.25 Megs of RAM, which would be feasible utilizing current PC technology.

If the memory requirements exceed the available RAM, a disc-paging scheme would have to be set up, so that only a portion of the stock model is in RAM at any one time, with the remainder residing on the hard drive. As this would dramatically decrease the performance of the overall system, the paging should be implemented in an intelligent manner such that the portion of the model read into memory has a significant amount of milling simulated on it before being swapped out for another piece of the model. This may require a pre-analysis of the current cut sequence to determine its range of travel in $X_{wp}$, $Y_{wp}$. Also, the Z-buffer model should be stored in binary format to minimize the read/write time.
CHAPTER 5

TOOLPATH ENVELOPE MODELING AND INTERSECTION CALCULATIONS

5.1 Chapter Introduction

This chapter describes techniques used to construct a geometric model of the cutting tool for the purpose of intersection calculation. The ‘cutter representation’ implemented actually models the swept toolpath envelope (STE), defining the surface of the volume swept by the cutter during a tool move. The STE is used in intersection calculations, rather than a single instance of the cutter positioned at discrete locations, to improve efficiency. A very large number of discrete positions would be required to accurately model the geometry of a single tool move, while generally a single STE is sufficient. Note, however, that a single instance of the cutting tool identical to that defined for geometric modeling is used during calculation of the cut geometry parameters. Both forms of the cutter model may be seen in Figure 5.1.

The basic requirement for the STE is that it accurately represent the surface created by the cutting tool moves during a single tool move. The STE representation should also lend itself to robust, accurate, and efficient intersection calculations. An exact representation is desirable from an accuracy standpoint. However, an equally important requirement is model stability over the wide range of possible move geometries, particularly during five-axis milling simulation. Computational efficiency is also a valid concern, as a typical G-code file can contain thousands of tool moves, each requiring a unique STE model. An approximate model with controllable error is therefore acceptable to balance the varied requirements.

5.2 Toolpath Envelope Modeling Techniques

5.2.1 Solid Modeling of the Swept Toolpath Envelope

The STE may be thought of as consisting of two individual cutter instances (at the initial and final cutter positions), connected by the volume swept by the cutter between these positions. Note the final cutter position for a given tool move is the initial position of the following move, providing continuity. As
Figure 5.1: The toolpath envelope defines the region swept in space by a single tool move and is used in intersection calculations for the simulation of material removal (left). The single cutter instance allows the flutes that engage the stock at a given rotational position to be identified using known flute geometry and the cut geometry of the current tool move (right).
opposed to the in-process stock model, the limited geometric complexity of the STE allows a solid model to work well in this application. The use of a solid model provides a means for robust intersection calculation. For each tool move, all Z-buffer elements that intersect the STE solid model are individually updated to remove the portions that lie interior to it, reducing material removal simulation to a finite set of line intersections. The exact form of these intersections is dependent on the type of STE solid model employed. In order to meet the goal of robust, efficient operation, these repeated calculations should be made as simple as possible, as they represent the lowest level of operation in the geometric model.

5.2.2 CSG Solid Modeling of the Toolpath Envelope

The simple geometric nature of most cutter types allows for use of Constructive Solid Geometry, or CSG, for toolpath modeling. This approach at STE representation was first presented for 5-axis milling simulation with a flat end mill and using Surface Point Sets [JHDS89], and later was successfully implemented for use with flat end mills and Extended Z-buffers [H94]. Using this approach, the STE is represented by a set of geometric primitives, each representing a specific portion of the STE. Note a single primitive type may represent several geometrically identical portions of the STE, as demonstrated by the multiplicity below. For a ball end cutter, the required primitives are (shown in profile in Figure 5.2):

- 2 planes to represent the swept sides of the STE,
- 3 cylinders;
  - 1 to represent the swept bottom of the STE between end positions, and
  - 2 to represent the upper portion of the cutter (above the ball end) at the end positions
- 2 spheres to represent the lower portion of the cutter (the ball end) at the end positions, and
- 1 swept circle for bounding the top of the STE.

For a flat end cutter, discs replace the spheres at the end positions, and a swept circle replaces the cylinder modeling the bottom of the STE. While only ball and flat end cutters are modeled in this research, the use of a CSG model also extends to other common standard cutter types (e.g. bull nosed and tapered cutters), as these also may be modeled using geometric primitives (toroids and conics). This suite of primitives could also model most non-standard cutter geometries by bounding the primitive to exist only along the axial portion of the cutter they represent.
Figure 5.2: The toolpath envelope is a closed solid constructed from a repeated set of geometric primitives. The example shown is for a ball end cutter, shown in 2D profile.

Figure 5.3: Basis vectors for the Toolpath Envelope Coordinates used in the calculation of intersection locations and cut geometry parameters. On the left is an isometric view, on the upper right is the view seen looking down A, and the lower right shows the view seen looking up N.
The primary benefits of the CSG approach are speed and reliability, as all intersection calculations are reduced to a repeated set of distinct line/primitive intersections. The primitives are simply predefined shapes, with only specific parametric variables (radius, length, orientation) varying; this use of a small set of predefined shapes, combined with the stability of line/primitive intersection calculations, greatly enhances speed and reliability. Although some primitives occur multiple times in a STE, only a single intersection algorithm is required for each primitive. These benefits become even more apparent when one considers the thousands of tool moves over which the primitives are repeated.

The primary drawback of this approach is that it is approximate when modeling 5-axis toolpaths. To simplify STE definition and subsequent intersection calculations, it is assumed all primitives are linear. Therefore, during 5-axis simulation, curvature in the path of travel and variations in cutter orientation are neglected. This approach is referred to as a '3-axis approximation', as it essentially treats all tool moves as if they were 3-axis. No error is introduced during 3-axis simulation with this approach.

5.2.3 B-rep Solid Modeling of the Toolpath Envelope

The alternative to CSG toolpath envelope modeling is a boundary representation, or B-rep. In the B-rep approach, the outer boundary of the STE volume is defined by a closed set of surface models. These models may either be analytical, piecewise continuous surface models (e.g. non-uniform rational B-splines, or NURBs), a tessellated surface representation such as a set of triangles, or a combination of the two. The surfaces of the B-rep may be defined as sweeps and surfaces of revolution created from a cutter profile definition and tool move data. Another method of performing this operation is through the use of a Sweep Differential Equation (SDE), which has been adapted for NC milling simulation in [BLW96]. However, this approach yields a set of boundary points rather than a true closed volume, which must be joined to form the continuous surface required for boundary representation.

The primary benefit of the B-rep approach is accuracy. During 5-axis milling the cutter orientation is continually changing, and this variation is not reflected in the CSG approach. In the B-rep approach, as the geometry of the constituent parts of the STE are not predefined, they may include the effects of non-linear geometry present in the toolpath, which can vary greatly for different tool moves.
The drawbacks to the B-rep approach are a decrease in speed and reliability, both a result of increased model complexity. Accurate definition of the cutter profile is difficult as it varies with cutter orientation, and generating sweeps from this variable profile is also complex. Also, the wide range of machine motions possible can lead to singularities in the equations required for STE definition. This complexity reduces the robustness of the method. Also, the toolpath envelope components must be defined as a function of the current tool move geometry; this results in a notable efficiency disadvantage when compared to the pre-defined geometry of the CSG approach. The variable and more complex geometry extend the efficiency and robustness losses to the intersection calculations. Finally, calculation of the geometric cut parameters is also made more complicated. One means of describing these variables is through projection of the intersected volume back onto a single instance of the cutter, and with a 5-axis STE representation using B-reps this projection is no longer linear.

The B-rep approach is generally the method implemented when a solids based stock model is employed, as the solids engine that defines the stock may also be used in the definition of the STEs and in the calculation of the intersections [ESE98][SA94]. This method has also been used in the modeling of deflected toolpaths [LLB98], and has been implemented for use with a discrete stock model in [Q93]. While the potential for greater geometric accuracy exists using this approach for 5-axis simulation, none of the literature presents data demonstrating an increased accuracy when performing NC milling simulation for force estimation. The error bounding methods presented in Section 5.3.3 provides an accurate means to control the error introduced by the 3-axis approximated STE, limiting error to acceptable levels. It has been demonstrated in by Quinn [Q93] that the benefits of the increased simplicity of the error-controlled CSG approach greatly outweigh the nominal geometric accuracy gains of the B-rep approach.

5.3 Toolpath Envelope Definition

5.3.1 Toolpath Envelope Definition: Boundary Points and Orientation

To accurately model a given tool move, the primitives used to define the STE model must be located and oriented properly relative to the stock model. The tool move information in the G-code file is provided in ‘machine coordinates’. This is a global coordinate system defined relative to the CNC machine.
that describes cutter motion relative to the kinematics of a given machine. The use of machine coordinate data is required for accurate geometric modeling and relative velocity calculation during 5-axis milling simulation. It is not possible to use CLDAT or some other part relative toolpath coordinates, as this only defines unique cutter positions relative to the stock, and does not describe the motion between them. To facilitate calculation of intersection locations and cut geometry parameters, it is convenient to define the toolpath model in workpiece coordinates (defined in Chapter 4) through the transformation:

\[
[P_{wp}] = [T] [P_M] 
\]

where \([P_{wp}]\) is the workpiece coordinate cutter position and orientation \([X,Y,Z],[I,J,K]\), \([T]\) is the transformation relation (function of machine kinematics and part setup), and \([P_M]\) is the machine coordinate position values \([X,Y,Z,A,B]\) found in the G-code file.

The transformation relation \([T]\) is dependent on the kinematics of a particular CNC machine. For the machine type used in this research, a Fadal 3-axis vertical milling center with a Jones and Shipman two-axis rotary tilt table, definition of this transformation is presented in Appendix B.

For each G-code position command read, a workpiece coordinate cutter position and orientation are calculated using Equation (5.1). To define a STE, two adjacent cutter positions are required, as the STE models a full tool move from \(P_{wp}(n)\) to \(P_{wp}(n+1)\), where \(n\) indicates the position number in the G-code file. These bounds are denoted \(P_i\) and \(P_f\), where \(P_i\) is the initial cutter position \((P_{wp}(n))\), and \(P_f\) is the final cutter position \((P_{wp}(n+1))\) for a given tool move. Notice that the final cutter position \(P_f\) for one tool move becomes the initial cutter position \(P_i\) on the following tool move.

The \((I,J,K)\) components of the cutter orientation \(A\) are calculated from the rotary axis positions as part of the transformation of Equation (5.1). The transformation provides a cutter orientation, denoted \(A_n\), at the position being transformed, \(P_n\). However, this orientation may differ from that at \(P_i\) (the \(P_f\) location for the prior move). Therefore, as the 3-axis approximation used in this research allows only a single orientation to exist for a given STE, error is introduced into the toolpath approximation. If the orientation \(A_i\) were used, there would be zero error at \(P_n\), which would increase to a maximum at \(P_f\) (assuming \(A_i\) and \(A_f\) are not equal). To minimize this error, an average cutter orientation value, \(A_{avg}\), is used. This causes the
error to exist at both \( P_i \) and \( P_f \), with the zero error condition existing half way between the two, at the location where the average orientation physically occurs. This method reduces both the maximum and average error present by a factor of up to 50%, as explained in Section 5.5.3.

For the remainder of this thesis, the vector \( A_{avg} \) is referred to simply as \( A \), unless noted otherwise. The method used to define \( A_{avg} \) is presented in Appendix B.

### 5.3.2 Toolpath Envelope Definition: Toolpath Coordinate System

To provide a foundation for calculations relative to the STE, it is necessary to define some basis vectors. These vectors are not to be confused with the toolpath coordinates of Chapter 2. Although the directions are similar, that system was defined to facilitate the calculation and summation of milling force vectors, while these serve to facilitate the calculation of intersections and cut geometry. In the toolpath envelope coordinate system, the origin is defined at \( P_i \) for the current STE. This lies at the bottom, radial center of a flat end cutter, or ball center for a ball end cutter. Four unit direction vectors, shown graphically in Figure 5.2, extend from it, and are used in the calculation of intersections and cut geometry parameters.

The first of these is the vector \( D \), which indicates the direction of travel of the cutting tool in Workpiece Coordinates during the current tool move. It is defined as:

\[
D = \frac{(P_f - P_i)}{\|P_f - P_i\|} \tag{5.2}
\]

The variable 'd' is used to denote a distance traveled from \( P_i \) along \( D \), and the value \( D = \|P_f - P_i\| \) defines the net distance moved, such that:

\[
P_f = P_i + dD \text{ for } d = D \tag{5.3}
\]

The second vector, denoted \( A \), runs along the cutters rotational axis. This vector defines the cutter Axial direction, and indicates cutter orientation. The variable 'a' is used to denote the distance traveled along the cutter axis from any point along \( D \). That is, the equation:
\[ P = P_i + dD + aA \]  \hspace{1cm} (5.4)

indicates the position of some location along the cutter axis at an axial height of 'a' along the cutter and at a distance 'd' from the initial position (i.e. it defines the plane of motion of the cutter axis). The vector \( A \) is defined as the \((I,J,K)\) component of \( P_w \) calculated during the transformation of Equation (5.1).

The third vector is referred to as the normal vector, denoted \( N \), as it defines the normal direction to the planar sides of the STE in Workpiece Coordinates. It is calculated as vector product of the toolpath cutter orientation vector \( A \) with the direction of travel vector \( D \):

\[ N = (A \times D) \]  \hspace{1cm} (5.5)

The above three vectors are not necessarily orthogonal, as \( A \) may have a component in \( D \). Therefore, for an orthogonal basis two additional vectors are calculated, \( A_p \) and \( D_p \). The vector \( A_p \) is calculated as the vector product of the direction of travel vector \( D \) with the toolpath Normal vector \( N \):

\[ A_p = (D \times N) \]  \hspace{1cm} (5.6)

Unlike \( A \), \( A_p \) lies orthogonal to \( D \), which often proves useful. Similarly, the component of \( D \) that lies orthogonal to \( A \), denoted \( D_p \), is also calculated:

\[ D_p = (N \times A) \]  \hspace{1cm} (5.7)

\( A, A_p, D, \) and \( D_p \) all lie in a plane, whose normal is \( N \). These five vectors are shown in Figure 5.3.

Note that these relations degenerate when the cutter travels along its own axis (\( |D \cdot A| = 1 \)). This condition is not a problem, but it must be identified prior to the calculation of intersections and cut geometry parameters. When the cutter is traveling upwards along its axis (\( D \cdot A = +1 \)), it is not possible for a flat or ball end cutter to intersect any stock material and no further calculations are required for that move. In the case where the cutter is traveling downward along its axis (\( D \cdot A = -1 \)), it is possible to
intersect the stock, but a STE is not required. In this case a single instance of the cutting tool, with a modeled length equal to the cutter length plus the distance traveled during the move, is sufficient. As this operation does not represent a standard end-milling operation, conservative feeds that are 50% of the calculated feed are set; refer to Section 3.4 for the determination of cutting conditions in this situation for a flat end cutter. This condition also provides an aid to automatic identification of drilling operations, allowing for future inclusion of the drilling force estimation models of Armarego [AW84] [AZ96].

5.3.3 Toolpath Envelope Error Bounding

There is one notable problem with the STE description provided in the approach described above. During 5-axis milling, motion of the rotary axes can result in a continuously variable cutter orientation $A$ as the cutter traverses from $P_j$ to $P_f$, resulting in non-planar swept sides to the STE. This motion can also result in a non-linear path of motion ($D$). The 3-axis approximated toolpath model used in this research assumes a constant orientation, and a linear path of travel; therefore error is introduced into the geometric model during 5-axis simulation. To provide acceptable geometric accuracy, it is desirable to control the maximum amount of error introduced. This section describes the error control methods employed in this research, which are similar to those proposed by Quinn [Q93], although developed independently.

The assumption that $D$ is linear generally introduces negligible error. Many toolpath generation algorithms in CAM software also assume a linear interpolation, neglecting the non-linear effects introduced by the rotary axes because they are machine dependent. Using current methods, the CAM software has no knowledge of the physical geometry of the NC mill to be used. Machine information is used only during the post-processing of the NC toolpath data. Therefore, the cutter positions and orientations required to produce a desired geometry are defined as discrete values in the CAM software, relative to the part geometry, with little or no accurate knowledge of the path that will be taken between them. The post processor then maps these values to machine space based on the kinematic arrangement of the axes on a given machine type. A typical 5-axis toolpath file will consist of many small moves in order to reduce the chordal deviation between the desired and actual toolpath that can result from post processing. This inherent design trait reduces the error introduced by the linear approximation for $D$, as the CAM software makes a similar linear assumption when defining the toolpaths. Additionally this also benefits the constant
A approximation used in this research. Although the value of $A$ is assumed variable in the CAM software, it is generally evaluated only at discrete locations (the cutter positions provided). This, combined with the linear approximation that takes place in the CAM software, generally results in a value of $A$ that varies little over the course of a single move. Figure 5.4 shows a gross approximation of how many small linear tool moves can closely approximate a true continuous sweep. In effect, the 3-axis approximation used in the STE is a form of discrete simulation.

While most toolpaths contain only minor variations in $A$ between adjacent tool moves, for a truly robust approach at NC simulation it is not sufficient to rely on this assumption. While linearity in the direction of travel is generally a valid assumption, as similar approximations take place when the toolpaths are defined, there are cases where large variations in $A$ occur during a single tool move, and in some classes of parts this can be a fairly frequent occurrence. For example, this occurs when the cutter is traversing an area with a very small radius of curvature relative to the distance traveled while attempting to maintain an orientation that lies at some constant angle relative to the surface normal. Large variations in $A$ over a single tool move also result when the cutter is forced to tilt to prevent interference with some other portion of the stock. Figure 5.5 displays both of these conditions when milling a small portion of a single stage axial flow turbine.

The average orientation that occurs over the course of a tool move ($A_{avg}$) is used to represent the orientation of the STE in order to passively minimize the error of the 3-axis approximated STE. The use of $A_{avg}$ causes the geometric error to exist at both $P_1$ and $P_f$ in equal magnitude, with the zero error condition existing half way between these two points (at $P = P_1 + (D/2)D$). This method reduces both the maximum and average error present factor of up to one half when compared to other passive methods. As shown in Figure 5.6a, for small angles the geometric error that results from the constant orientation approximation may be defined as:

$$\varepsilon = C_{Max} \, dA$$  \hspace{1cm} (5.8)

where $\varepsilon$ is the geometric error between the actual and as-modeled surface (in), $C_{Max}$ is the furthest location of stock contact from the center of rotation in an axial or radial direction (in), and
Figure 5.4: Although the toolpath envelope is modeled using a '3-axis approximation' with a constant cutter orientation ($A_i$), if the maximum change in orientation ($dA$) is restricted this can still accurately model a true 5-axis move.

Figure 5.5: Many typical 5-axis tool moves contain only minor variations in $A$ (far left set of cutters). However, there are cases where large variations in $A$ exist over a fairly short move. The most common of these are when the part being milled has a very small radius of curvature relative to the length of the tool move (middle set of cutters), and when the cutter has to tilt to prevent part interference (far right set of cutters).
Figure 5.6a: A 3-axis approximation of a 5-axis STE produces a geometric error $\varepsilon$ that is a linear function of the angle $dA$ and the distance $C$ from the center of rotation (for small $dA$).

Figure 5.6b: For a ball end cutter, the $C_{\text{Max}}$ value is typically measured up the axis of the cutter from the ball center. For a flat end cutter, if the highest point of axial contact is less than the cutter radius $R$, use $C_{\text{Max}} = R$ as error can also occur on the bottom of the cutting tool.

Figure 5.6c: The use of the average cutter orientation $A_{\text{Avg}}$ minimizes the value of $\varepsilon$ for a given STE by providing the smallest possible value for $dA$. Similarly, the subdivision of moves that contain a large orientation change into a series of smaller "sub-moves" also reduces $\varepsilon$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
dA is the angle between two possible orientation vectors (rad).

In the 3-axis approximation, \( \mathbf{P}_i(n) = \mathbf{P}_i(n+1) \), and all changes in cutter orientation are modeled as occurring between adjacent STEs, at the \( \mathbf{P}_i(n) \), \( \mathbf{P}_i(n+1) \) location. Therefore the cutter is always modeled as rotating about a known cutter location position, providing knowledge of the position where maximum error occurs. The value of \( C_{\text{Max}} \) is known in the axial direction, as it is required to estimate the region of possible intersection around a given STE (Section 5.6.3). In the case of a ball end cutter, this always provides a sufficient value, but note that only contact axially higher than the cutter location position (ball center) need be considered as no error occurs on the cutter ball (see Figure 5.6.b). In the case of a flat end cutter, if the maximum axial contact occurs less than one cutter radius from the base, then the radius value should be used for \( C_{\text{Max}} \) instead of the axial location, as error can also occur with the bottom of a flat end cutter, as shown in Figure 5.6.b. Once \( C_{\text{Max}} \) is known, the angle \( dA \) may be calculated between any two adjacent orientation vectors \( \mathbf{A}_1 \) and \( \mathbf{A}_2 \) using their scalar product (recall they are unitized):

\[
dA_{1,2} = \cos^{-1}(\mathbf{A}_1 \cdot \mathbf{A}_2)
\]

(5.9)

The error introduced by the 3-axis approximation is minimized using \( \mathbf{A}_{\text{avg}} \) as follows. When \( \mathbf{A}_i \) or \( \mathbf{A}_f \) is used to represent the orientation, the value of \( dA \) is maximized, as it is equal to the total angle between \( \mathbf{A}_i \) and \( \mathbf{A}_f \) (\( dA_{i,f} \)). However, when \( \mathbf{A}_{\text{avg}} \) is used, the value of \( dA \) is one-half that value, as the angle between \( \mathbf{A}_{\text{avg}} \) and \( \mathbf{A}_i \) or \( \mathbf{A}_f \) is equal to one half of the total angle between \( \mathbf{A}_i \) and \( \mathbf{A}_f \) by definition, reducing \( \varepsilon \) by a factor of two. This is shown in the first two images of Figure 5.6c. Note that if some orientation other than \( \mathbf{A}_{\text{avg}} \) were used, the error would only increase.

While this passive method minimizes error in the 3-axis approximated STE, unacceptable values of \( \varepsilon \) can still result when large variation in \( \mathbf{A} \) occurs during a tool move, when the cutter contacts the stock far up its axis, or a combination these. Therefore an active mode of error compensation is also required. This implemented through subdivision of the net move into a series of sub moves, which are linearly interpolated from the initial to the final cutter position, resulting in a smooth transition over the course of
the net tool move. $A$ varies gradually between each sub-move, reducing the value of $dA$, and thus $\varepsilon$. This is shown in the image on the far right in Figure 5.6c.

For the purposes of geometric verification of a set of NC toolpaths, the ideal value for $\varepsilon$ would be on the order of NC mill accuracy, which has real-world values on the order of 0.0005"-0.001". This would produce on average zero error (i.e. the NC mill itself contains the same level of uncertainty). However, the computational cost of this level of accuracy can be far too great, as it greatly multiplies the number of tool moves that must be simulated. However, for the purposes of force estimation a more reasonable value for $\varepsilon$ would be on the order of the mesh spacing for the Z-buffer model, as this marks the inherent level of geometric accuracy in the overall geometric model. It is very difficult to quantify in general how a given value of $\varepsilon$ affects the overall geometric accuracy or computation time, as it is a function of the part geometry being milled. For example, if cutting occurs only on the ball portion of a ball end cutter, there is zero error regardless of cutter tilt, allowing a very large allowable error to be used as no subdivision of the moves is necessary. Alternatively, for physically small parts or small part files, it may be practical to use a very small allowable error as the computation time is not significant. Due to this variability in the effects of a given value of $\varepsilon$, its value is user-definable, with a default value equal to the Z-buffer model mesh spacing. This allowable geometric error is defined $\varepsilon_{\text{allowable}}$.

Each sub-move is modeled as an individual STE. To obtain the parameters necessary to define the sub-move STEs, it is necessary to first determine the number of sub-moves required using Equation 5.8. For each tool move, the value of $dA_{\text{r}}$ is calculated, and used with $C_{\text{max}}$ which is already known from the calculation of the intersection zone (an intersection zone for the net toolpath is first calculated, and if no stock intersection occurs it is not necessary to subdivide the move). Note that the value $(dA_{\text{r}}/2)$ is used for $dA$ in Equation 5.8, as only half of the total angle is required due to the use of $A_{\text{Avg}}$. Once $\varepsilon$ is calculated, it is compared to $\varepsilon_{\text{allowable}}$. If $\varepsilon < \varepsilon_{\text{allowable}}$ no subdivision is necessary, but if $\varepsilon > \varepsilon_{\text{allowable}}$ then the number of sub-moves required is calculated as:

$$ N_{\text{sub}} = \left\lfloor \frac{\varepsilon}{\varepsilon_{\text{allowable}}} \right\rfloor $$           \hspace{1cm} (5.10)

where $N_{\text{sub}}$ is the number of sub-moves required (unitless).
Once $N_{\text{sub}}$ is known, it is possible to define the sub-move STEs. This is performed using G-code positional data, as the linear interpolation between positions occurs at the machine level rather than the part level. If the subdivision took place in workpiece space, it could be possible that the resulting motion may not match the physical machine motion. Each individual machine axis must start and finish its motion in unison with the other axes, in order to define the geometry swept by a given NC mill. The kinematic relationships that define machine motion are not considered in workpiece space, and therefore they may be violated. It is also not sufficient to use the kinematic equations as a means of validating that a subdivision in the workpiece domain does not violate the machine constraints. As the transformation of Equation 5.1 maps 5 machine parameters $(X_M, Y_M, Z_M, A, B)$ into 6 workpiece parameters $(X_{WP}, Y_{WP}, Z_{WP}, I, J, K)$ using the kinematic relations, it is possible that the inverse transformation could map a single set of workpiece space variables into multiple mathematically valid solutions in the machine space. Therefore it is necessary to subdivide the move in the Machine domain (G-code coordinates), and then apply the transformation of Equation (5.1) a total of $N_{\text{sub}}$ times to define the orientation and final tool position for each sub-move STE (which then becomes the initial tool position for the next sub-move). The move is therefore subdivided using the differential amount moved by each axis:

$$dP_M(n) = (P_M(n) - P_M(n-1))$$

where $dP_M(n)$ is the differential amount moved by each axis for net tool move number $n$ (it consists of values for $dX$, $dY$, $dZ$, $dA$, $dB$).

The final tool position is then defined for each sub-move (in machine coordinates) as:

$$P_{M_{\text{sub}}}(m) = P_M(n-1) + m \left( \frac{dP_M(n)}{N_{\text{sub}}} \right)$$

where $P_{M_{\text{sub}}}(m)$ is the end position for all machine axes of the $m^{th}$ sub-move, $P_M(n-1)$ is the start position of the net tool move (from G-code), and $m$ is the index to the current sub-move, $1 \leq m \leq N_{\text{sub}}$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Each $P_{N\text{-sub}}$ is then transformed using Equation (5.1) for definition of the sub-move STEs in Workpiece coordinates, identical to the case where the machine coordinate data is read from a G-code file.

Once a sub-move STE is defined, intersection calculations, force estimations, and feedrate selection operations are performed as if it were actually a unique tool move. When subdivision is required, it is necessary to remain aware it is performed for geometric accuracy only. Relative velocity calculation requires use of net tool move information, as that is the information sent to the machine controller being modeled. This allows for a single calculation for relative velocity regardless of the number of sub-moves. Similarly, when estimating milling forces or calculating feedrates, the most conservative values that result from the complete set of sub-moves must be identified and stored as the representative value for the net move. That is, over a given set of sub-moves, the highest force value estimated, or lowest feed value required to maintain a desired force, should be treated as the absolute value for the net move.

Figure 5.7a shows a linked set of 3-axis approximated STEs extracted from a 5-axis milling simulation. Note that no faceting or other signs of linearization appear in the image, demonstrating the smooth geometric transitions provided by this method (the variable width of the linked STEs is due to the varying axial depth of cut). There are approximately 1000 STEs present, some requiring subdivision.

Figure 5.7b graphically shows the effects of geometric error that can result when $\varepsilon$ is too large. Three tool moves are shown in both images: a 0.5" linear move into the stock with the cutter vertical, followed by a single 2.75" tool move with $\Delta A = 60^\circ$ in N, and finally another 0.5" linear move with the cutter fixed at 60° from vertical as the cutter exits the stock. $R = 0.25"$, and the Z-buffer spacing is 0.0075" in both cases. An allowable $\varepsilon$ of 0.075" was set in the left image, providing only 12 'sub moves' over which the cutter rotation may be distributed. An allowable $\varepsilon$ of 0.0075" was set on the right, providing for 112 sub-moves. (note this value of $\varepsilon$ is on the order of the Z-buffer element spacing).

Finally, Figure 5.7c demonstrates STE subdivision allows for accurate simulations of cutter rotations about a center of rotation other than the cutter location position, although the 3-axis approximated STE assumes all rotation to occur about this position between tool moves. In this figure, the cutter first rotates $+45^\circ$ in a single tool move, followed by a $-90^\circ$ rotation, with the center of rotation occurring 1" above the cutter location position ($R = 0.25"$). This is made possible by allowing the toolpath direction of travel $D$
Figure 5.7a: Note the smoothness of this linked set of STEs (approx. 1000 tool moves) seen looking up +Zw. The dashed white line indicates the stock boundary, and the cutter is visible at right.

Figure 5.7b: In both images shown above, R = 0.25", Z-buffer spacing = 0.0075", and a tool move with a net dA of 60° occurs over 2.75". On the left ε = 0.075", and on the right ε = 0.0075".

Figure 5.7c: Although all rotation is assumed to occur at the cutter location position, it is possible to accurately simulate rotations where the center of rotation is not at this position.
to oppose the direction that dA is incremented. The actual direction of travel of each axial disc is found during relative velocity calculation. Those that lie axially below the center of rotation are actually travelling in +D, and a positive velocity value is calculated, while those above the center of rotation produce a negative velocity value. If a disc lies exactly at the center of rotation, a zero velocity value is obtained.

As the error introduced by the 3-axis approximation is managed through passive and active error control, it is possible to use this method to accurately represent 5-axis swept STEs. By allowing the cutter orientation to remain constant, the speed and reliability of the model is greatly enhanced, while limiting error to acceptable levels.

5.4 Intersection Calculations

5.4.1 Intersection Calculation Overview

The STE is composed of a set of geometric primitives in the CSG approach. This allows material removal simulation to be performed using only line-primitive intersection calculations. This provides an excellent foundation for NC simulation for force estimation, as it is very robust, efficient, and also lends itself well to the calculation of the cut geometry parameters. Another benefit of this approach is that it is simple to maintain and expand. By defining the primitives in a parametric manner consistent with the toolpath envelope coordinate basis vectors, it is possible to calculate intersections for a wide variety of cutter sizes and move geometries with a limited set of intersection calculation algorithms.

When calculating intersections between the STE and the stock, all Z-buffer elements that could potentially intersect the STE are checked for intersections with each primitive defining the STE. For example, if modeling material removal with a flat end cutter, the required algorithms intersect a line with:

- a cylinder (called 2 times per Z-buffer element to model the two end positions),
- a disc (called 2 times per Z-buffer element to model the bottom of the two end positions),
- a plane (called 2 times per Z-buffer element to model the swept sides), and
- a swept circle (called 2 times per Z-buffer element to model the top and bottom bounds).

Although 8 different intersection calculations are required per Z-buffer element to find all possible intersections in the above case, only 4 different algorithms are required. This not only ensures robust
behavior through the use of compact and well tested code, it also offers an excellent opportunity for pre-
calculation of common variables, and other minor forms of optimization. As the intersection code is very 
low level, noticeable performance enhancements can be achieved with even small amounts of optimization.

When modeling a ball end cutter, spheres are used to model the bottom of the cutter at \( P_i \) and \( P_f \).
For improved efficiency, only the sphere at \( P_f \) is required, as it is identical to the sphere at \( P_i \) for the 
following move. This does not hold for a flat end cutter, which has a bottom modeled by a disc. However,
if the cutter is tilted in the direction of travel \( (A \cdot D > 0) \), only the disc at \( P_i \) is required. If tilted backward 
relative to the direction of travel \( (A \cdot D < 0) \), only the disc at \( P_f \) is required. The end that does not require a 
unique disc lies internal to the STE, covered by the swept circle modeling the swept bottom of the STE and 
joining the two ends.

### 5.4.2 Bounding Intersection Calculations

Mathematically, it is not known prior to the intersection calculations if a given Z-buffer element 
actually intersects any of the geometric primitives (those which do not intersect a primitive produce 
physically meaningless results, e.g. complex numbers). A means of identifying only those elements that can 
potentially intersect the STE was therefore implemented, greatly reducing computation time by requiring 
only a small subset of the total number of Z-buffer elements to be checked for intersections. The subset of 
Z-buffer elements checked for intersections is referred to as the 'intersection zone'. To define the 
intersection zone, the STE is projected down along \( Z_v \) onto the \( X_v, Y_v \) plane. The maximum and 
minimum \( X_{wp}, Y_{wp} \) bounds of the projection define the intersection zone (see Figure 5.8). To further 
improve efficiency, rather than projecting the net axial length of the STE (entered by the user as a fluted 
length of the cutter), a conservative estimate of the length of cutter engaged in the stock is used. This is 
determined by estimating the axially highest point of contact that the cutter has with the stock, using the 
known heights of the Z-buffer elements and the known STE definition.

Implementation details for the intersection zoning procedure are described by Hemmett in [H94], 
although some variations exist in this research due to the inclusion of ball end cutters. For flat-end cutters, 
the extrema of the base of the projected STE are calculated using the cutter radius value multiplied by some 
fraction that is typically \(< 1\) in order to account for the effects of cutter tilt relative to \( Z_{wp} \). This is necessary
Figure 5.8: On the left an isometric view of an STE intersecting the Z-buffer model is shown. On the right is the view seen looking down the $Z_{wp}$ axis, with the toolpath envelope projected onto the $X_{wp}, Y_{wp}$ plane. The bounds of the projected STE define the intersection zone containing the Z-buffer elements that could possibly intersect the current STE.

Figure 5.9: When intersecting the sphere that models the bottom of a ball end cutter in the final cutter position, it is possible to find no intersections (left-most image), one valid intersection and one invalid intersection that lies internal to the STE (middle image), or two valid intersections (right-most image).

Figure 5.10: On the left is a Z-buffer element intersecting an STE three times, twice on the bottom cylinder and once on a planar side. On the right is the same STE and Z-buffer element seen looking down the direction of travel. Note that the physically valid intersections are simply the max and min (in $Z_{wp}$) intersection values in the case shown, or for any other set of intersections, and the remainder lie on portions of the primitives internal to the STE.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
as the projected radius of a disc (the base of a flat end cutter) varies as the normal to that disc is tilted relative to the plane of projection. However, when a ball end cutter is being modeled, a full radius value is always used regardless of tilt. This is because the projected radius of a sphere (the base of a ball end cutter) remains unchanged as the sphere is rotated. This ball-end tilt correction is implemented in the context of [H94] by constraining the parameters $R_c_x$ and $R_c_y$ of Equation (4.11) in [H94] to be equal to $R$, the cutter radius, when modeling a ball end cutter.

5.4.3 Intersection Calculations

All intersection calculations are reduced to simple line primitive intersection calculations in the approach used in this research. These are well understood and are not presented here. The methods implemented are derived from those developed by Jerard et. al. [JHDS89], in which geometric primitive are intersected with surface normal vectors. Hemmett provides an example of this approach using a Z-buffer model for flat end mills. Another excellent source of information regarding the line/primitive NC simulation approach may be found in Chapter 12 of [CJ98].

Two notable variations from [JHDS89] exist in this research. First, [JHDS89] intersects the STE with a Surface Point Set, where the intersected vectors lie normal to the milled surface. In this research the STE is intersected with a Z-buffer model, allowing for optimization based on the constant [0,0,1] vector orientation. The second variation is that [JHDS89] does not include ball end cutters, which requires the addition of a new geometric primitive element, a sphere, to model the cutter bottom at $P_r$. It also requires application of the cylinder primitive to model the volume swept by the cutter ball between $P_i$ and $P_f$. In [JHDS89] a method for performing cylinder/line intersections is presented to model discrete instances of the cutting tool shaft at $P_i$ and $P_f$. In that case the cylinder extends along $A$ and is bounded as $(0 <= a <= L)$, where $L$ denotes the cutters modeled length and 'a' indicates a location along $A$. When modeling the bottom of a STE created by a swept ball end cutter, the cylinder is defined to extend between $P_i$ and $P_f$ along $D$ (as opposed to $A$), and is bounded by $(0 <= d <= D)$, where $D$ is the total distance traveled during the move (see Figure 5.1).
5.4.4 Line/Sphere Intersection Calculations

The new geometric primitive required to model a ball end cutter at \( P_f \) is a sphere. The equation of a sphere may be defined as:

\[
((X_{wp} - PC_x)^2 + (Y_{wp} - PC_y)^2 + (Z_{wp} - PC_z)^2) = R^2 \tag{5.13}
\]

where \( X_{wp}, Y_{wp}, Z_{wp} \) define a point on the surface of the sphere (in), \( PC_x, y, z \) is the location of the center of the sphere in workpiece space (in), and \( R \) is the sphere radius (in).

In Equation 5.13, substitution of \( P_f \) for \( PC \), and a \( X_{wp}, Y_{wp} \) Z-buffer base position for \( X \) and \( Y \), allows for solution of \( Z \), which corresponds to the \( Z_{wp} \) intersection location(s) along the Z-buffer element:

\[
Z_{wp} = P_{fz} \pm \sqrt{dZ} \tag{5.14}
\]

where \( dZ = \sqrt{R^2 - (X_{wp} - P_{fx})^2 - (Y_{wp} - P_{fy})^2} \).

Only real-valued solutions for \( dZ \) are valid. No intersection with the sphere exists when \( dZ \) is complex, as this indicates \((X_{wp} - P_{fx})^2 + (X_{wp} - P_{fx})^2 > R^2\), i.e. the Z-buffer element is not within the radius of the sphere in the \( X_{wp}, Y_{wp} \) plane. Solutions for \( dZ \) typically occur in pairs, one where the Z-buffer element enters the sphere, and one where it exits (except in the rare case where the Z-buffer element lies exactly one radius in \( X_{wp}, Y_{wp} \) from the sphere center). In some cases both intersections are physically valid, that is both lie on the external boundary of the STE, and in others only one intersection is valid, with the other occurring internal to the STE (see Figure 5.9). In either case, both intersections are stored as possible valid intersections. Once all possible intersections have been calculated, intersection validation is performed to identify those that occur on the outer surface of the STE, and are therefore physically possible.

5.4.5 Identification of Valid Intersections

The fundamental benefit of line-primitive intersection is they are inherently stable and efficient. Either (an) intersection(s) exists between the line and the primitive, or a solution is not possible. However,
not all of the calculated intersections are valid. Only intersections on the outer boundary of the STE can physically occur, with the remainder occurring internal to the STE (see Figure 5.10).

In the CSG approach used in this research, the swept circle modeling the top of the STE is not required for physical accuracy, but rather to ensure a closed volume. This greatly simplifies the identification of 'valid' intersections. When calculating intersections, the Z-buffer element is treated as an infinite line. As the STE is a closed volume, at least two points of contact with the Z-buffer element will always exist on the boundary of the STE (entrance/exit intersections). These boundary intersections translate into the maximum and minimum intersection locations on the Z-buffer element. Therefore, when calculating intersections with a given Z-buffer element, only the maximum and minimum values are stored as true intersections. The remaining intersections exist with portions of the primitives internal to the STE. These do not physically occur and are ignored, being only an artifact of the toolpath modeling scheme.

Without the top on the STE, it is no longer a closed volume, and the 'two intersection' rule no longer applies. This makes the identification of valid intersections much more ambiguous. However, this applies strictly to the general 5-axis case. When 3-axis milling, only a single intersection value can exist, assuming the Z-buffer elements are aligned parallel to the cutter axis, and so it is only necessary to save the minimum $Z_{wp}$ values. Also note that the above statements only apply to standard cutting tool geometries. Shaped cutters such as routers may produce STEs that may have several boundary intersections. However such cutting tools comprise an insignificant portion of those typically used.
CHAPTER 6

CALCULATION OF THE CUT GEOMETRY PARAMETERS

6.1 Chapter Introduction

The cut geometry parameters describe the bounds of the contact area between the cutting tool and
workpiece during a tool move. These parameters are defined relative to the discrete cutter model, presented
in Chapter 3, which represents the cutter as a set of discrete axial disc elements. This model provides a full
description of the cutter and flute geometry, and also stores the cut geometry data, all on a per disc basis.
For a given disc and cutter rotation angle, only those flute segments that lie within the contact area are
included in the force contribution from that disc A net force estimate is obtained by summing over all flute
segments that engage the stock. This entire process is then repeated over all remaining discrete cutter
rotation angles, so that the peak force for the move may be identified. Therefore an accurate description of
the contact area via the cut geometry parameters is critical for accurate force estimation, as it determines
which flute segments are included during force summation, allowing the milling force predictions to
include the effects of variable cut geometry, variable flute geometry, and cutter rotation.

To calculate the cut geometry parameters for a given tool move, the volume of material intersected
by the cutter can be projected back along the direction of travel, onto the cutter at the initial tool position.
This projected volume results in a contact area, the bounds of which are the cut geometry parameters.
However, for improved efficiency using limited information, contact area definition requires more than a
simple projection.

The first part of this chapter provides a brief description of the cut geometry parameters and how
they are used for contact area definition. The majority of the chapter then describes the techniques used to
efficiently define these parameters for a given tool move using intersection calculation data.
6.2 The Contact Area Geometric Description

6.2.1 The Cut Geometry Parameters and Discrete Mechanistic Modeling

In any desirable milling operation, contact between the cutting tool and stock occurs only along the cutting tools' flutes. As the cutter rotates, differing portions of each flute contact the stock, shearing material from it. While flute contact varies continually over a single rotation, the region of possible contact remains constant for a given tool move (assuming constant cutting conditions). The contact area describes this region of possible contact between the cutter and stock for a given tool move (see Figure 6.1), and this area is defined by its bounds via the 'cut geometry parameters', or simply the 'cut geometry'. Note that the contact area remains constant over all rotation angles, although it may vary between moves.

The cut geometry is an integral part of the mechanistic modeling approach for milling force estimation. To improve computational efficiency, these parameters are therefore defined in terms of existing variables in the mechanistic model. The two principal degrees of freedom for the flute segments in the mechanistic model are axial location and the angular position. The axial degree of freedom is discrete, defined by the current axial disc, over which all parameters remain constant. The rotational degree of freedom is continuous, although the cutter rotation simulation is discrete, as the cutter helix angle results in flute segments at intermediate rotational positions. Therefore, the basic requirement for contact area definition is that it provides discrete (i.e. axial disc based) bounding in the axial direction, and continuous, real valued angular bounding in the rotational direction.

6.2.2 The Axial Bounds ('Axial Depth of Cut')

Traditionally, the axial depth of cut is a single value that describes the length of cutting tool engaged in the stock material, measured from the cutter bottom. This single value was sufficient during the original development of mechanistic models, as it applied only to relatively simple 3-axis cuts where the cutter bottom always engaged the stock [KD83][KDL84]. In the general case however, and in particular during 5-axis milling, the bottom portion of the cutting tool may not perform any milling. While this is not the norm, it does occur, and is a common occurrence for some styles of milling, such as 5-axis flank milling where the part is milled by the side of the cutting tool rather than its end (see Figure 6.2).
Figure 6.1: The contact patch, shown shaded in the image on the right, defines the region on the cutting tool where material removal may occur during a given tool move.

Figure 6.2: The axial bounds $A_{\text{Max}}$ and $A_{\text{Min}}$ identify which axial discs engage the stock material. Note that the bottom of the cutter does not have to engage the stock, particularly milling steep faces with a ball end cutter, shown on left, or when flank milling, shown on right.

Figure 6.3: Note that some axial discs bounded between $A_{\text{Max}}$ and $A_{\text{Min}}$ may not actually intersect the current $Z$-buffer element, passing through gaps in the element instead.
Therefore, in this research the traditional axial depth of cut value is replaced by two values that define the minimum and maximum axial points of contact with the stock, denoted $A_{\text{Min}}$ and $A_{\text{Max}}$ respectively.

Axial bound definition is only required relative to the current intersected segment, so that the discs which pass through it may be identified for inclusion in the calculation of the cut geometry parameters. Recall that for each Z-buffer element intersected by the STE, an intersected segment is defined, which describes the portion of the Z-buffer element intersected. Projecting the ends of the intersected segment onto the cutter, and noting the axial location of these projections, identifies the values of $A_{\text{Min}}$ and $A_{\text{Max}}$. Once $A_{\text{Min}}$ and $A_{\text{Max}}$ have been calculated, they are used only to identify the axial discs that exist at the axial locations of these bounds. These discs are tagged as intersecting the stock material (InStock = YES, see Chapter 3), and radial distance values are calculated for them (explained shortly). These discs then serve as implicit axial bounds, with the discs between them being identified using the indices of the bounding discs. This set of discs is known to pass through the intersected segment during the current tool move, provided they do not pass through a gap in the segment, and they are tagged as intersecting the stock material and require radial distance calculations (see Figure 6.3). 'Global' $A_{\text{Min}}$ and $A_{\text{Max}}$ for the net move are not stored, rather they are implied by the minimum and maximum disc indices calculated, which may be directly applied in the mechanistic model axial summation.

Once the values of $A_{\text{Min}}$ and $A_{\text{Max}}$ are known, the corresponding axial disc indices are calculated. If a given axial bound is positive it exists on a cylindrical portion of the cutting tool, where the disc thickness remains constant, and the index may be calculated as:

$$i_{\text{Max/Min}} = \text{floor} \left( \frac{A_{\text{Max/Min}}}{dZ} \right) + N_{D-\text{Ball}} \quad (6.1a)$$

where $i_{\text{Max/Min}}$ is the index to the upper or lower bounding discs,

$A_{\text{Max/Min}}$ is the positive valued max or min axial bound value (in),

$dZ$ is the constant cylindrical axial disc thickness value (in), and

$N_{D-\text{Ball}}$ is the number of disc elements present on the ball portion of the cutter.

Note for a flat end cutting tool, $N_{D-\text{Ball}} = 0$, and so this relation still holds. If a given axial bound is negative, then it is known to exist below the Cutting Tool Axial Coordinate origin, valid only on the ball portion of
cutting tool (in the context of this research). In this case the axial disc thickness value is variable, but each disc has a constant arc-length orthogonal to the cutter profile surface normal, defined using a constant angular spacing $d\psi$. Therefore, on the ball end portion of a cutter, the disc indices are calculated using:

$$i_{\text{Max/Min}} = \text{floor} \left( \frac{\sin^{-1} \left( \frac{A_{\text{Max/Min}}}{R_c} \right)}{d\psi} \right) + N_{\text{D-Ball}}$$

(6.1b)

where $i_{\text{Max/Min}}$ is the index to the upper or lower bounding discs, $A_{\text{Max/Min}}$ is the negatively valued max or min axial bound value (in), $R_c$ is the cutter ball radius value (in), $d\psi$ is the cutter ball angular disc spacing amount (rad), and $N_{\text{D-Ball}}$ is the number of disc elements present on the ball portion of the cutter.

For robust operation with the finite precision of numerical computations, the index must be checked to ensure $0 \leq i_{\text{Max/Min}} \leq N_D$ so that potential memory faults during disc element access may be avoided.

6.2.3 Entrance and Exit Angles and the Radial Bounds

As the flute locations on a given disc are described in terms of the angular coordinate $\beta$ (see Chapter 2), the cut geometry should be similarly described to simplify comparison between flute angular location and contact area bounds. Therefore, for each disc that intersects the stock material, an entrance and exit angle (or sets of entrance and exit angles), are required to define the contact area for that disc. An entrance angle denotes the point where flute first engages the stock material in the direction of cutter rotation, and an exit angle denotes the point where contact ends (see Figure 6.4).

The entrance and exit angles are calculated using the maximum and minimum radial depth of cut, denoted $N_{\text{Max}}$ and $N_{\text{Min}}$ as they exist in the Toolpath Normal direction (see Figure 6.4):

$$\beta_{\text{entr/exit}} = \sin^{-1} \left( \frac{N_{\text{Max/Min}}}{R_D} \right) + \pi/2$$

(6.2a)

where $\beta_{\text{entr/exit}}$ are the entrance and exit angle values respectively (rad), $N_{\text{max/\text{min}}}$ are the corresponding maximum and minimum radial depths of cut (in), and...
\[ \beta = \alpha + 90^\circ, \text{ where } \alpha = \sin^{-1}(N / R) \]

Figure 6.4: The entrance and exit angles bound the region of stock intersection on a given axial disc for a given tool move. They may be easily calculated from the radial depth of cut values \( N_{\text{Max}} \) and \( N_{\text{Min}} \), as shown in the image on the right.

Figure 6.5a: Traditionally defined entrance and exit angles often provide a conservative representation of stock engagement, as shown above for the case of the shaded axial disc on the far left.

Figure 6.5b: Error is also introduced by the traditional approach when ball end milling and \( \mathbf{A} \cdot \mathbf{D} < 0 \).
$R_D$ is the radius value of the current axial disc (in).

Note the suffix order is important in Equation (6.2a), as $\beta_{en}$ is generally a function of $N_{\text{Max}}$, also expressed $\beta_{en}(N_{\text{Max}})$, and $\beta_{ex}(N_{\text{Min}})$. Exceptions to this rule are described shortly. As shown by Equation (6.2a), it is trivial to calculate $\beta_{en}$ and $\beta_{ex}$ from $N_{\text{Max}}$ and $N_{\text{Min}}$. The problem is therefore to devise an efficient means of calculating $N_{\text{Max}}$ and $N_{\text{Min}}$ values for each engaged axial disc from the intersection data.

6.2.4 Axial Disc Quadrants and the Definition of Cut geometry parameters

Traditionally, a single set of entrance and exit angles is used in the mechanistic model, requiring only one set of radial distances per engaged disc to be calculated [KD83][KDL84][E93][H94]. This provides a conservative estimate of the contact area by combining the worst-case conditions that exist during the tool move into a single set of angles [H94]. For example, if a single pair of radial distance values is calculated when the material between them contains a gap, the resultant entrance and exit angles do not reflect the fact that they do not bound continuously solid material (See Figure 6.5a). While conservative results are desirable in the presence of error, accuracy is preferred. Additionally, the use of a single entrance/exit angle pair can lead to non-conservative error when simulating a ball end cutter and $(A \cdot D) < 0$, as in this situation material removal on the 'back side' of the cutter would not be considered (See Figure 6.5b). The back of the cutter is defined as 180°-360° in cutting tool Polar coordinates. Therefore, to improve overall accuracy as well as to allow for accurate 5-axis ball end simulation, 4 quadrants are defined on each disc, and $\beta_{en}, \beta_{ex}$ are defined for each (Figure 6.6). Note Quadrants III and IV lie on the back of the cutter, and can only intersect stock when cutting with a ball end mill and $(A \cdot D) < 0$.

The use of quadrants requires up to four sets of $N_{\text{Max}}$ and $N_{\text{Min}}$ be identified and stored for each engaged axial disc, one set per quadrant, but only those quadrants that contact the stock require radial distance calculations. For each quadrant, a radial distance value $N$ is calculated for each Z-buffer element that intersects it, but only $N_{\text{Max}}$ and $N_{\text{Min}}$ are stored. When the final values of $N_{\text{Max}}$ and $N_{\text{Min}}$ are known, they are used in the entrance and exit angles calculation. For Quadrants I and II, $\beta_{en}$ is a function of $N_{\text{Max}}$ (i.e. $\beta_{en}(N_{\text{Max}})$) and $\beta_{ex}(N_{\text{Min}})$, and so Equation (6.2a) is used (shown on the right in Figure 6.6; note Quadrant I
Figure 6.6: Dividing axial discs into 4 quadrants, each with its own pair of entrance and exit angles, allows for more accurate modeling of the cut geometry.

Figure 6.7: Determination of the appropriate axial disc quadrant contacted by $P_Z$ requires knowledge of cutter location at the point of intersection when cutting with a ball end mill and $A \cdot D < 0$.

Figure 6.8: The quadrant method requires error compensation due to the discrete Z-buffer stock model.
contains -N). However this relationship is reversed in Quadrants III and IV, such that \( \beta_{\text{en}}(N_{\text{Max}}) \) and \( \beta_{\text{en}}(N_{\text{Min}}) \). This requires use of the alternate relation:

\[
\beta_{\text{en/loc}} = \sin^{-1}\left( \frac{(N_{\text{Max}})}{R_{D}} \right) + \pi/2
\]

(6.2b)

For each radial distance stored, the quadrant that it occurs in must be identified. For the case of an intersection with a cylinder, or with a sphere when \( A \cdot D >= 0 \), quadrant assignment is simplified as only quads I and II are valid. In this case negative values of \( N \) belong to quadrant I, and positive values to quadrant II (calculation of \( N \) is covered shortly). The situation is more complex when cutting on the ball portion of a ball end tool and \( A \cdot D < 0 \), as quads III and IV are also valid. In this case, it must be first determined whether the intersection occurs on the front (quadrant I or II) or on the rear (quad III or IV) of the cutter. This is achieved by moving the cutter from \( P_1 \) along \( D \), until the point \( P_z \) resides on the cutter surface (see Figure 6.7). \( P_z \) is defined as a known point along the current intersected segment that contacts the cutter surface during the current tool move. The variable \( D_z \) is used to represent the distance along \( D \) that the cutter traveled from \( P_1 \) to the point of contact with \( P_z \). The value of \( D_z \) is found either during intersection calculation, described in Jerard et. al. [JHDS89], or during calculation of the axial bounds, described in the following section. Assuming for now that \( D_z \) is known, the cutter location position where the cutter first contacts \( P_z \) (i.e. \( P_z \) lies on the cutter surface), denoted \( P_{\text{CL}}(D_z) \), may be calculated as:

\[
P_{\text{CL}}(D_z) = P_1 + D_z D
\]

(6.3)

When this position is known, a vector \( ZCL \) may be defined as:

\[
ZCL = P_z - P_{\text{CL}}(D_z)
\]

(6.4)

As this calculation is required only on the ball portion of the cutter \( \| ZCL \| = R \), the cutter ball radius, as it extends from the ball center to some point on the ball surface. The vector \( ZCL \) is then used to determine
whether the intersection occurred on the front or rear half of the cutter ball. This is done by noting the sign of the scalar product of \( ZCL \) with the vector \( D_p \) (recall from Chapter 5 that \( D_p = N \times A \)):

\[
Q = \text{SignOf}(ZCL \cdot D_p) \tag{6.5}
\]

where \( Q = +1 \) indicates intersection on the front of the cutter (Quads I or II), and \( Q = -1 \) indicates intersection on the back of the cutter (Quads III or IV).

Equation (6.5) is shown graphically on the right in Figure 6.7. Determination of the final quadrant is then performed using the sign of the radial distance, as previously described.

There are other points to note in the right hand image of Figure 6.7. Looking down \( N \), as shown, the axial direction \( A \) defines a line between the front and back of the cutter. The dotted line represents the transition from 'ingress' to 'egress' on the cutter surface. The ingress region, where the cutter is moving into the stock and material removal may occur, is on the +\( D \) side of this line. This line is the side profile of a plane that passes through the cutter location position and whose normal is \( D \).

### 6.2.5 Error Control for Axial Disc Quadrants

When simulating the milling of solid stock across more than one quadrant, a continuous contact area must be enforced between the adjacent quadrants. This does not occur automatically as a result of the discrete Z-buffer mesh spacing. The entrance and exit angles are calculated from the radial distance values in a given quadrant, which are separated between quadrants by a value that is a function of \( d_{xy} \), the mesh spacing distance. This is especially a concern between quadrants I and II, where this condition commonly occurs, as this region can make the greatest contribution to the force estimates. As shown in Figure 6.8, the discrete Z-buffer spacing results in \( N_{\text{Max}} < 0 \) in Quadrant I and \( N_{\text{Min}} > 0 \) in Quadrant II, although zero radial distance values are required in both to model solid stock engagement. Therefore, some form of compensation is required to compensate for the 'gap error'.

In the case of Figure 6.8, the distance between \( N_{\text{Min,I}} \) and \( N_{\text{Max,I}} \) is equal to the Z-buffer element spacing \( d_{xy} \). This condition may therefore be compensated for using the assertion:
If \(( N_{\text{Min-II}} - N_{\text{Max-I}}) \leq \varepsilon_{N} + \text{TOL} \) \hspace{1cm} (6.6)

\[ N_{\text{Min-II}} = N_{\text{Max-I}} = 0; \]

where \( \varepsilon_{N} \) is the 'allowable gap error', in this case \( \varepsilon_{N} = d_{XY} \), and \( \text{TOL} \) is some small tolerance amount to account for round off error (~\( d_{XY}/100 \)).

In Figure 6.8, \( \varepsilon_{N} = d_{XY} \) as a result of the cutter travelling in a principal direction in Workpiece space, with zero cutter tilt about \( D \) (\( N \) is parallel to \( X_{wp}, Y_{wp} \)). Therefore the effective mesh spacing in \( X_{wp}, Y_{wp} \) orthogonal to \( D \) is also that of a principal direction, which is \( d_{XY} \) by definition. It is now desired to define a means of calculating \( \varepsilon_{N} \), the 'allowable gap error', in the general case.

For convenience, define the effective mesh spacing in \( X_{wp}, Y_{wp} \) and orthogonal to \( D \) as '\( e_{XY} \)'. The value of \( \varepsilon_{N} \) is calculated as a function of \( e_{XY} \), and so the maximum value of \( e_{XY} \) that could possibly exist must be determined to ensure \( \varepsilon_{N} \) is maximized. This is achieved by determining the maximum separation between four adjacent Z-buffer elements that form a square in both principal directions (see Figure 6.9).

The value of \( e_{XY} \) is calculated using the angles \( \gamma_{1} \) and \( \gamma_{2} \) of Figure 6.10, where \( \gamma_{1} \) is the angle between \( D_{XY} \) and the \( X_{wp} \) direction, \( \gamma_{2} \) is the angle between \( D_{XY} \) and \( Y_{wp} \), and \( D_{XY} \) is the \((X_{wp}, Y_{wp})\) component of \( D \).

Using these angles, \( e_{XY} \) is calculated as:

\[ e_{XY} = d_{xy} / \cos (\gamma_{\text{min}}) \] \hspace{1cm} (6.7)

where \( \gamma_{\text{min}} \) is the minimum value between \( \gamma_{1} \) and \( \gamma_{2} \) (rad).

Thus, starting at \( \gamma_{1} = 0 \), \( e_{XY} \) starts as the mesh spacing \( d_{xy} \). Its value then increases to a maximum of \( (1.414) \) \( d_{xy} \) at \( \gamma_{1} = \gamma_{2} = 45^\circ \), and subsequently decreases again as \( D_{XY} \) approaches \( Y_{wp} \).

While \( e_{XY} \) is defined in the \( X_{wp}, Y_{wp} \) plane, the gap error \( \varepsilon_{N} \) is the result of a difference in values between \( N_{\text{Max-I}} \) and \( N_{\text{Min-II}} \) defined along \( N \). Therefore \( e_{XY} \) must be projected onto \( N \) to calculate \( \varepsilon_{N} \) (note in Figure 6.8, \( e_{XY} \) is parallel to \( N \) and maps onto \( N \) with a 1:1 ratio, resulting in \( \varepsilon_{N} = d_{XY} \)). This is performed
Figure 6.9: The effective mesh spacing $\varepsilon_{XY}$, seen at bottom looking down $D_{XY}$, at various degrees of rotation. Note that the worst-case conditions from 4 adjacent $Z$-buffer elements are used, rather than simply 2 adjacent elements along $X_{wp}$ or $Y_{wp}$, producing the max value of $\varepsilon_{XY}$.

Figure 6.10: Assuming worst-case conditions, $\varepsilon_N$ increases as $D_{XY}$ moves away from either the $X_{wp}$ or $Y_{wp}$ principal direction, and as cutter tilt about $D$ increases.
using $\delta$, the angle of tilt about $D_{XY}$ between $N$ and the $X_{wp}, Y_{wp}$ plane, as shown in the right hand pair of images of Figure 6.10. The projection of $\varepsilon_{XY}$ onto $N$, which is $\varepsilon_N$, is calculated as:

$$\varepsilon_N = \frac{\varepsilon_{XY}}{\cos(\delta)} \quad (6.8)$$

When there is no tilt of $A$ about $D_{XY}$, then $\varepsilon_N = \varepsilon_{XY}$; this is always the case in 3-axis milling. More importantly, note that when $A$ is orthogonal to $Z_{wp}$ ($\delta = 90^\circ$), $\varepsilon_N$ is infinite; this condition must be noted to prevent a division by zero error. This orientation allows the entire $N$ vector, bounded in length by the current disc radius, to lie completely between the $Z$-buffer elements that define $\varepsilon_{XY}$. To visualize this, imagine the cutter of Figure 6.10 to be tilted $20^\circ$ more to the right than shown such that the portion of $N$ bounded by the cutter radius lies completely between the $Z$-buffer elements. This condition provides insufficient information to make a reasonable approximation of $\varepsilon_N$, and so no gaps are allowed between quadrants I and II. This provides a conservative approach during an uncommon condition. Note the mesh spacing shown in Figure 6.10 is greatly exaggerated. Generally $d_{xy} < R/15$, where $R$ is the cutter radius, and so this condition only occurs when $A$ is very nearly orthogonal to $Z_{wp}$, which is not typical.

Once calculated, the value of $\varepsilon_N$ is used in the assertion of Equation (6.6). If $N_{Min-II} - N_{Max-I} > \varepsilon_N + \text{TOL}$, it is assumed an actual gap in the material exists, as it exceeds the maximum theoretical $Z$-buffer separation. Otherwise it is assumed the separation is an artificially induced result of the discrete $Z$-buffer spacing, and both $N_{Min-II}$ and $N_{Max-I}$ are reset to zero for a continuous frontal contact area.

### 6.3 Cut Geometry Parameter Calculation Overview

An overview of the procedure used to calculate the cut geometry parameters will now be provided. The cut geometry parameters are calculated using an 'intersected segment', which describes the portion of a single $Z$-buffer element intersected by the STE during a given tool move. The contribution the current $Z$-buffer element makes to the cut geometry parameters must be calculated prior to updating the $Z$-buffer model (i.e. deleting the intersected segment from the geometric model to simulate material removal), as all
geometric information contained in the intersected segment is deleted. Figure 6.11 provides a 'pseudo-code' description of cut geometry calculation and its integration into the larger feedrate selection framework.

When calculating the cut geometry parameters, first the ends of the intersected segment are used to define \( A_{\text{Max}} \) and \( A_{\text{Min}} \), which are then used to identify the bounding axial discs. Once the bounding discs are known, they and the discs between them are tagged as intersecting the stock and have a value of \( N \) calculated, provided they do not pass through a gap in the Z-buffer element. The values of \( N \) are calculated for the bounding discs using the known end positions of the intersected segment, while the point of intersection between the intersected segment and all intermediate discs must be determined in order to perform the calculation for \( N \). When \( N \) is calculated, the quadrant that it applies to is determined, and \( N \) is compared to the existing \( N_{\text{Min}} \) and \( N_{\text{Max}} \) values stored in that quadrant. These values are then updated as necessary. After all intersections have taken place for the current move, gap error compensation is performed and the entrance and exit angles are calculated. Once this has been done for all quadrants on all discs that contact the stock during the current tool move, the contact area definition is complete, described in terms of discrete axial bounds and continuous rotational bounds as desired.

To calculate radial distances and axial positions, a point of intersection between the intersected segment and the cutter must be known. This point, denoted \( P_z \), may be a known location, such as an end of the intersected segment or a bound on a gap, or it must be solved for. Generally the majority of the \( P_z \) locations fall into the latter category. For computational efficiency, only one \( P_z \) location per intersected disc is desired, and so it must be solved for as a function of axial location. In some cases the known \( P_z \) locations also require additional calculations to determine the axial height at which they occur.

The problem then is two-fold. First, a method of obtaining the desired cut geometry parameters \( A_{\text{Max}}, A_{\text{Min}}, \) and \( N \) as a function of \( P_z \) must be defined. This provides a uniform means of geometric parameter calculation for all axial discs using a 1 dimensional parameter. Second, an efficient means of calculating the per-disc intersection location \( P_z \) must be developed. Efficiency is critical in both cases, as these routines are the most frequently called of all the geometric modeling routines, on the order of \( (\text{Number of Discs Engaged}) \times (\text{Number of Z-buffer elements intersected}) \) per tool move.
Transform Tool Move from Machine to Workpiece Coordinates; *(Chapter 5)*
Define STE; *(Chapter 5)*
Define Intersected Zone; *(Chapter 5)*
Loop through potentially affected Z-buffer Elements of Intersected Zone ...

{  
  Calculate intersections with STE; *(Chapter 5)*
  If valid intersections exist...
  
  Define Intersected Segment; *(Chapter 4)*

  **Identify Max, Min Axial Bounds** $A_{\text{Max}}$, $A_{\text{Min}}$; identify corresponding Discs
  Loop through Discs affected by current Intersected Segment...

  {  
    Identify intersection location $P_z$ between Disc and Intersected Segment
    If valid intersection exists (disc does not pass through gap in segment)...

    {  
      Calculate Radial Distance $N$
      Determine Quadrant Radial Distance applies to
      Store only Max, Min Radial Distance in Each Quadrant

      next Disc;
    }
  }

  Update Z-buffer Model to reflect material removal; *(Chapter 4)*
  next Z-buffer element;
}

**Loop through Discs affected during Current Tool Move...**

{  
  Calculate Allowable Gap Error, adjust Radial Distances as required;
  Calculate Entrance/Exit Angles for each quadrant from Radial Distance values;
  Calculate Relative Velocity *(Chapter 7)*
  next Disc;
}

**Estimate Required Feed Value** *(Chapter 3)*

{  
  Define initial feed estimate;
  Loop to Desired Force...

  {  
    Estimate Milling Force; *(Chapter 2)*
    Compare to Desired Force, Adjust Feed as necessary; *(Chapter 3)*
  }
}

next Tool Move;

Figure 6.11: Pseudo-code displaying the integration of cut geometry calculation (in bold) into the larger feed estimation framework. The method shown applies to a single tool move.
6.4 Calculation of the Axial Bounds

6.4.1 Axial Bounds

Axial bounds are used to identify the discs that can potentially contact the current intersected segment. They are calculated by noting where the end positions of the intersected segment contact the cutting tool. This makes them dependent on the STE geometry, and hence cutter type. For a ball end cutter there are four specific cases that must be considered, and three for a flat end cutter. However, the cylindrical portion of a ball end cutter is identical to that on a flat end cutter, and so some of the techniques are common to both cutter types.

Note that while an STE is used for intersection calculation, a single instance of the cutting tool is required for axial bound calculation. This is necessary because A and D are coupled in the non-orthogonal cutting tool coordinates, and so the axial location of an intersection becomes a function of distance traveled from $P_i$ to the point of intersection. Therefore the cutter location at the point of intersection, $P_{Cl}(D_z)$, must be used in the calculation of the axial bounds. When the cutter is at $P_{Cl}(D_z)$, the calculated point of intersection $P_z$ lies on the surface of the cutter instance, providing the true axial location of the intersection location (see Figure 6.12). Only when the cutter is at the actual position along D where the intersection occurs is the coupling between A and D cancelled. If the axial location is calculated when the cutter is at some other location along D, for example $P_i$, the coupling between A and D leads to erroneous results for the axial position calculated, as indicated by $A_{err}$ in Figure 6.12.

6.4.2 Identification of Axial Bounds on a Flat End Cutter

Three cases exist when solving for axial bounds on a flat end cutter. The first case exists when one end of the intersected segment is a known intersection location with the bottom, swept circle portion of the STE. This case is trivial. Any intersections found with the swept circle are known to occur at an axial depth of zero, or $A_z = 0$. $A_z$ is defined as the axial point of contact of $P_z$ on the cutting tool.

The remaining two cases both involve the upper cylindrical portion of the cutting tool. In one case $P_z$ is a known intersection location, calculated with the upper cylindrical portion of the STE. In the other case $P_z$ is either a top height or a bound on a gap in the element, for which the axial location is desired. As both of these cases are shared with ball end cutters they will be presented in the following section.
Figure 6.12: When finding the axial location of a point $P_z$, the calculation must be performed at the location where the point $P_z$ first contacts the surface of a cutter instance, using the vector $ZCL$ as shown on the left. If some other cutter position is used the result is invalid, as shown on the right by $A_{err}$, calculated incorrectly using the vector $(P_z - P_t)$.

Figure 6.13: The geometric primitives extend infinitely along their respective degrees of freedom, requiring the location of each intersection to be calculated along each D.O.F. (e.g. $D_z$ along $D$) to ensure they occur within the space physically occupied by the current STE.

Figure 6.14: Issues in finding $D_z$ for some known Z-buffer location that is not an intersection location.
6.4.3 Identification of Axial Bounds on a Ball End Cutter

There are four primary cases to be considered when identifying the axial location of an intersection with a ball end cutter. The first case exists when \( P_Z \) is a known intersection location calculated between a Z-buffer element and the upper, cylindrical portion of the cutting tool. The second case exists when \( P_Z \) is a known intersection location calculated between a Z-buffer element and the sphere that models the bottom of the cutting tool at the final cutter position. The third case exists when \( P_Z \) is a known intersection location calculated between a Z-buffer element and the cylinder used to model the bottom, swept portion of the STE. The fourth case exists when \( P_Z \) is some other point on the Z-buffer element, such as the top of the Z-buffer element or one of the bounds on a gap in the element.

**CASE 1:** \( P_Z \) is an intersection location calculated with the cylindrical portion of the net STE, which consists of two planar sides in the swept region and two cylinders at the end positions. Note this case also applies to flat end cutters. In this case the axial location is known, having been calculated during intersection calculation to ensure the intersection lies within the valid axial bounds of the cutting tool. The intersection calculations take place with cylinders and planes that extend infinitely in \( A \) (see Figure 6.13), and so the intersections found must be bounded to the axial space physically occupied by the STE [JHDS89][H94]. Therefore the axial locations are known values in this case.

**CASE 2:** \( P_Z \) is a known intersection location calculated with the sphere modeling the cutter bottom at \( P_f \). In this case, the axial location of the intersection is calculated using the scalar product:

\[
A_Z = (P_Z - P_f) \cdot A
\]  
(6.9a)

**CASE 3:** \( P_Z \) is a known intersection location calculated with the cylinder modeling the bottom, swept portion the STE. When used to model the swept bottom of a ball end cutter the cylinder has no axial degree of freedom, as it extends along \( D \), and so no axial location is found during intersection calculation. It's also not possible to use Equation (6.9a); which holds only when either the intersection occurred at \( P_b \) or when \( D \) and \( A \) are orthogonal \((D \cdot A) = 0\). When \( A \) and \( D \) are not orthogonal, axial location is a function of \( D \). This is accounted for through the use of \( P_{cl}(D_Z) \), which includes the distance traveled from \( P_t \) to the point of intersection, and allows for the definition of a generalized solution for axial location:

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[ A_Z = (P_Z - P_{CL}(D_Z)) \cdot A \] (6.9b)

The value of \( D_Z \) required for calculation of \( P_{CL}(D_Z) \) is determined during the intersection calculations, as it’s required to ensure that the infinite cylinder that models the swept bottom of the ball end STE is appropriately bounded in \( D \) (see Figure 6.13). This allows for simple solution of Equation (6.9b) after first calculating \( P_{CL}(D_Z) \) using the known \( D_Z \) value in Equation (6.3). Also, \( P_Z - P_{CL}(D_Z) \) is a known vector, previously defined as \( Z_{CL} \). Notice that Equation (6.9a) is actually a special case of Equation (6.9b), with the intersection occurring when \( P_{CL}(D_Z) = P_i \).

**CASE 4:** the intersected segment end location(s) is(are) not a known intersection, but rather the Z-buffer element top, or a bound on a gap in the segment. This case is a common occurrence, as it results when the Z-buffer terminates element inside the STE (i.e. the entire top of the Z-buffer element is removed). This case also applies to flat end cutters; in this situation ignore all cutter ball references.

The axial location may be obtained in this case using Equation (6.9b). However, this first requires calculation of \( P_{CL}(D_Z) \), i.e. the location of the cutter when \( P_Z \) (the known point along the Z-buffer element, in this case the top height, or bound on a gap) first contacts the surface of the cutting tool during the move. It is not known in advance whether this point of initial contact occurs on the ball or cylindrical portion of the cutter. As the majority of the material removal occurs on the ball portion of a ball end cutter, it is first assumed the ball contacts \( P_Z \). If the ball portion is not found to contact \( P_Z \), then the intersection is known to have occurred with cylindrical cutter portion.

To determine if \( P_Z \) contacts the cutter ball, a sphere with the cutters radius is swept along \( D \) from \( P_i \) to where the sphere first contacts \( P_Z \). The distance traveled along \( D \) is \( D_Z \), which is required to calculate \( P_{CL}(D_Z) \). Note when the sphere is at \( P_{CL}(D_Z) \), \( P_Z \) lies on the surface of the sphere, and so \( P_Z \) is a distance \( R \) (the cutter radius) from the spheres center at \( P_{CL}(D_Z) \). Also note \( P_{CL}(D_Z) = P_i + D_ZD \), as defined in Equation (6.3). Setting the magnitude of \( (P_Z - P_{CL}(D_Z)) \) equal to \( R \), and substituting for \( P_{CL}(D_Z) \), yields:

\[ \| (P_Z - P_i - D_ZD) \| = R \] (6.10)

Note that the only unknown in this equation is \( D_Z \). \( D_Z \) is solved for by first by defining the vector:
\[ \mathbf{Z} = \mathbf{P}_z - \mathbf{P}_i \]  

(6.11)

This is substituted into Equation (6.10), which may then be solved the square of the magnitude and expanded, yielding the relation:

\[
\begin{align*}
D_z^2 & \ (D(x)^2 + D(y)^2 + D(z)^2 - 2 \ D_z \ [D(x) \ Z_i(x) + D(y) \ Z_i(y) + D(z) \ Z_i(z)] \\
& + (Z_i(x)^2 + Z_i(y)^2 + Z_i(z)^2) = R^2
\end{align*}
\]  

(6.12a)

The coefficient of the "\(D_z^2\)" term is the square of the magnitude of the unitized direction vector, and has a value of 1. The coefficient for the "\(-2D_z\)" term is the dot product of \(\mathbf{Z}_i\) and \(\mathbf{D}\), \((\mathbf{Z}_i \cdot \mathbf{D})\), and the last term in the equation is dot product of \(\mathbf{Z}_i\) with itself (i.e. its magnitude squared). Rewriting, the relation arrived at is:

\[
D_z^2 - 2 \ (\mathbf{Z}_i \cdot \mathbf{D}) \ D_z + (\mathbf{Z}_i - \mathbf{Z}_i)^2 - R^2 = 0
\]  

(6.12b)

Equation (6.12b) may be solved for \(D_z\), providing two solutions. The lesser value is desired, as it indicates where \(\mathbf{P}_z\) first contacts the sphere, the greater value is the exit location on the back of the sphere (shown on the left in Figure 6.14). As \(\mathbf{Z}_i\) and \(\mathbf{D}\) extend in the same direction, \((\mathbf{Z}_i - \mathbf{D})\) is positive; as is the result of the square root. Therefore the minimum value of \(D_z\) may be solved for as:

\[
D_z = (\mathbf{Z}_i \cdot \mathbf{D}) - \ [ \ (\mathbf{Z}_i \cdot \mathbf{D})^2 - (\mathbf{Z}_i \cdot \mathbf{Z}_i) \ ]^{1/2}
\]  

(6.13)

In Equation (6.13), no real-valued solution exists if \([(\mathbf{Z}_i \cdot \mathbf{D})^2 - (\mathbf{Z}_i \cdot \mathbf{Z}_i)] < 0\). This indicates the sphere never contacts \(\mathbf{P}_z\) during its sweep along \(\mathbf{D}\), and \(\mathbf{P}_z\) must contact the cylindrical cutter portion.

If \((\mathbf{Z}_i \cdot \mathbf{D})^2 - (\mathbf{Z}_i \cdot \mathbf{Z}_i) \geq 0\), then \(\mathbf{P}_z\) contacts the sphere and \(D_z\) may be obtained. This value may then be used in Equation (6.3) to calculate \(\mathbf{P}_{CL}(D_z)\), allowing for solution of \(A_z\) using Equation (6.9b). The sign of \(A_z\) must then be checked to verify \(A_z \geq 0\), indicating \(\mathbf{P}_z\) lies on the outer surface of the cutter, and is valid. Otherwise the contact occurs on the upper half of the sphere, which is internal to the cutter and is...
surrounded by the cylindrical portion of the cutter, as shown on the right in Figure 6.14. In this case the value of $A_z$ found with the sphere is invalid, and $A_z$ must be calculated on the cutter cylinder.

If a valid value of $A_z$ doesn't exist with the swept sphere, or if modeling a flat end cutter, then $P_z$ first intersects the cylindrical portion of the cutter. While a swept cylinder could be used to solve for $A_z$, a simpler approach is used to improve efficiency. This approach requires a simple 2D projection, and compensation for the coupling between $A$ and $N$ that exists on the surface of the cylinder when viewed down $D$ when $(A \cdot D) \neq 0$ (see Figure 6.15). Only when $A$ and $D$ are orthogonal may the vector $Z_i$ of Equation (6.17) be projected along $-D$ and onto $A$ using $(Z_i \cdot A)$, solving for $A_z$ directly. However, $Z_i$ may be projected along $-D$ and onto $A_p$, which is orthogonal to $D$ (see Chapter 5), using:

$$A_p = Z_i \cdot A_p \quad (6.14)$$

Noting that $A$ and $A_p$ are unit vectors, the angle $\psi$ between them may be calculated as:

$$\psi = \cos^{-1}(A_p \cdot A) \quad (6.15)$$

Noting that $(A_p \cdot A) = \cos(\psi)$, the $A_p$ location may be transformed onto $A$ using:

$$A_0 = A_{ni} / (A_p \cdot A) \quad (6.16)$$

The projection of $P_z$ onto $A_p$, and the transformation from $A_p$ onto $A$ to arrive at $A_0$, are 2D operations in the $A,D$ plane. As shown in Figure 6.16, this neglects the 3D cutter geometry, and $A_0$ generally requires correction to arrive at the desired value of $A_z$. Recall from Chapter 5 that $D_p = (N \times A)$, and the cutter surface is defined by a cylinder along $A$, at a constant radius $R$ from $A$ in the $N,D_p$ plane. Equation (6.16) calculates the projection of $A_{ni}$ onto the $A$ axis, in the $A,D$ plane at $N = 0$ (hence the notation "$A_{ni}$"). However, this point of intersection actually occurs on the cutter surface, at a distance $R$ from $A$, generally at some non-zero radial distance (see Figure 6.16, 6.17). A generic solution for $A_z$ must be found that considers intersections at any point on the front of the disc, i.e. $N \neq 0$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 6.15: When \( A \) and \( D \) are not orthogonal, the axial location of \( P_z \) projected down \( D \) onto the cutter surface is a function of its radial location; (note on left that \( N \) crosses several discs). Therefore \( P_z \) must be indirectly mapped to \( A \) via \( A_P \), as shown on the far right.

Figure 6.16: The mapping of \( P_z \) along \(-D\) and onto \( A \) is a 2D operation in the \( D,A \) plane, which neglects the 3D nature of the cutter surface in \( D_P \) and \( N \).

Figure 6.17: The mapping of \( P_z \) along \(-D\) and onto \( A \) is a 2D operation in the \( D,A \) plane, which neglects the 3D nature of the cutter surface in \( D_P \) and \( N \).
Because \( A_0 \) lies directly on the \( A \) axis in 3D space, it is valid only at \( D_P = 0 \), where \( D_P \) intersects \( A \). If the projection of \( P_Z \) along \(-D\) is traced from \( D_P = 0 \) to \( D_P = R \), as shown in Figure 6.16, the value of axial location changes and can cross several discs due to the non-orthogonal nature of \( D \) and \( A \). Note in this figure that different axial discs are intersected between \( D_{P0} \) and \( D_{P1} \), \( D_{P1} \) and \( D_{P2} \), and \( D_{P2} \) and \( D_{P3} \). Therefore, to ensure that the correct axial disc is identified, the actual axial location where the projection of \( P_Z \) enters the cutter surface must be determined. This will differ from the value provided by Equation (6.16) as a function of \( D_P \). The cutter surface is defined relative to \( D_P \) and \( N \) using the relation:

\[
N^2 + D_P^2 = R^2
\]  

(6.17)

This allows \( D_P \) to be calculated as a function of \( N \), a known value. This relationship may be seen in Figure 6.17, which shows the same situation as Figure 6.16, only viewed along \(-A\) as opposed to \( N \). In this view it becomes obvious the radial distance \( N \) where the projection of \( P_Z \) along \(-D\) intersects the cutter surface actually affects the axial location on cutter surface.

Once \( D_P \) has been calculated, the true axial location may be found. Noting that the angle \( \psi \) between \( A \) and \( A_P \) is equal to the angle between \( D \) and \( D_P \), and referring to the right image in Figure 6.17, the desired axial location \( A_z \) may be found using:

\[
A_z = A_0 + D_P \tan(\psi)
\]

(6.18)

Combining Equations (6.11, 6.14-6.18) yields the final, integrated solution for \( A_z \):

\[
A_z = \frac{(P_Z-P_P) \cdot A}{(A_P \cdot A) + (R^2-N^2)\tan(\cos^{-1}(A_P \cdot A))}
\]

(6.19)

This relationship is more efficient than performing a direct 3D projection of \( P_Z \) along \(-D\), onto the cutter cylinder. Additionally, some parameters in this equation remain constant over the tool move, namely \((A_P \cdot A)\) and \(\tan(\cos^{-1}(A_P \cdot A))\), reducing the required calculations. Also, \((P_Z-P_P)\) is known, required for calculation of the radial distance \( N \). The independent solution for \( N \) is presented in the following section.
6.4.4 Minimum Axial Location for Ball End Cutters

While the above four cases accurately calculate the axial location where the ends of the intersected segment contact the cutter, it is possible that neither of these locations are the minimum axial point of contact ($A_{\text{min}}$). When at least one intersection occurs on the cutter ball, and the cutter is traveling such that $D$ has a component in $-A$, ($(D \cdot A) < 0$), additional calculations are required to ensure the axially lowest point of contact with the current intersected segment is identified. This may be visualized by projecting the intersected segment along $-D$ and onto the cutter surface at $P_n$ as shown in Figure 6.18. Note in this figure that the point of intersection between the intersected segment and the cutter ball traces a semicircular arc along the ball surface. The ends of this arc correspond to where the ends of the intersected segment contact the cutter ball, but $A_{\text{min}}$ occurs between these points, at some unknown location along the intersected segment. This is a result of calculating the intersection location $P_z$ using a swept circle that lies orthogonal to the direction of travel (i.e. a cylinder). As the normal to this swept circle extends along $D$, when $(D \cdot A) < 0$ the lower half of this swept circle (with which intersections are found) moves onto the 'back' of the cutter (Quadrants III and IV), pivoting about $N$ and effectively up $A$, as shown in profile in Figure 6.18. Therefore, $A_{\text{min}}$ may not occur at the calculated intersection locations, but at some intermediate point that existed while the Z-buffer element was being trimmed, and so an alternate solution for $A_{\text{min}}$ is required.

There are three primary situations that must be considered when simulating a ball end cutter and $(D \cdot A) < 0$, as shown in Figure 6.19. The case that exists for a given cutting scenario cannot be determined in advance, and so the axial location of both ends of the intersected segment must always be calculated. Once the axial bounds of both ends are known, the quadrant they occur in must be determined, as this provides the means for identifying the situation that currently exists (see Figure 6.19). If both ends of the intersected segment contact either the front or the back of the cutter (Quads I, II or Quads III, IV; situation "1" in Figure 6.19), it is not necessary to calculate $A_{\text{min}}$ as the lower of two calculated axial locations may be used. If one end of the intersected segment contacts the cutter front and the other the back (situations "2" and "3" in Figure 6.19), the location of $A_{\text{min}}$ occurs along the arc between the two known positions, at some unknown point of intersection $P_{\text{min}}$ on the intersected segment (see Figure 6.20). In this case $A_{\text{min}}$ must be calculated as described below. Note in all three situations $A_{\text{min}}$ is always defined as the greater of the two calculated axial locations, as may be seen when comparing the situations of Figure 6.19.
Figure 6.18: When $(\mathbf{D} \cdot \mathbf{A}) < 0$, the lowest axial position does not necessarily occur at the intersection location $P_z$ (i.e. at $d = D_z$) when the intersection occurs on the "back" of the cutter.

Figure 6.19: When $(\mathbf{D} \cdot \mathbf{A}) < 0$, three situations may exist while identifying the axial bounds of an intersected segment projected onto the cutter surface. Note that situation "1" also exists when both intersections occur on the front of the cutter.
The axially lowest position on the arc traced on the surface of the cutter ball during intersection is $A_{\text{Min}}$. This arc may be thought of as the intersection between a static sphere and a plane formed by sweeping the Z-buffer element along $-D$. The normal to this plane, and therefore the arc, is:

$$n_{\text{Arc}} = \frac{Z_{WP} \times D}{||Z_{WP} \times D||} \quad (6.20)$$

where $n_{\text{Arc}}$ is the unit normal to the plane containing the arc traced by the Z-buffer element.

Note when $D$ and $Z_{WP}$ are parallel, the cutter is descending directly down on the Z-buffer element and no arc is formed, and so Equation (6.20) is not required.

The plane of the arc and $D$ are parallel, and so the distance between them is constant. This distance may be calculated using a known point on each, along with the direction of minimum distance ($n_{\text{Arc}}$):

$$d_{\text{Arc}} = (P_z - P_{CL}(D_z)) \cdot n_{\text{Arc}} \quad (6.21)$$

where $d_{\text{Arc}}$ is the distance from the plane of the arc to the direction of travel vector,

$P_z$ is one of the known end locations on the arc, and

$P_{CL}(D_z)$ is the corresponding cutter location position at the point of intersection.

Note that $P_z - P_{CL}(D_z)$ is the previously calculated vector $Z_{CL}$. Using $d_{\text{Arc}}$, the arc radius may be calculated. The magnitude of $(P_z - P_{CL}(D_z))$ is equal to the sphere radius $R$ ($P_z$ lies on the sphere surface, $P_{CL}(D_z)$ lies at its center), while $d_{\text{Arc}}$ exists in a direction orthogonal to the plane of the arc ($n_{\text{Arc}}$), and the arc exists on the surface of the sphere, therefore the constant in-plane arc radius must be:

$$R_{\text{Arc}} = (R^2 - d_{\text{Arc}}^2)^{0.5} \quad (6.22)$$

where $R_{\text{Arc}}$ is the 2D radius of the arc (in).

This is shown on the right in Figure 6.20, where the cutter ball is viewed along the vector $(A \times n_{\text{Arc}})$. This places $A$ and $n_{\text{Arc}}$ in the plane of the page, as well as $P_{\text{Min}}$, at the perimeter of the spheres 2D projection.
Figure 6.20: \( R_{\text{Arc}} \) is the radius of the arc defined by the path of intersection in the \( n_{\text{Arc}} \) plane, and its use compensates for rotation about \( n_{\text{Arc}} \) as its value remains constant during rotation. The local coordinate system describes the relative orientation of the left and right-hand figures.

\[ A_{\text{Min}} = -R \cos(\phi) \]

Must solve for \( \phi \), which exists with \( A \) in plane of page

\[ \phi = \xi + \zeta \]

\[ \xi = \tan^{-1}(d_{\text{Arc}}/R_{\text{Arc}}) \]

\[ \zeta = \pi - \cos^{-1}(A \cdot n_{\text{Arc}}) \]

Figure 6.21: Using \( R_{\text{Arc}} \) and \( n_{\text{Arc}} \), the calculation of \( A_{\text{Min}} \) may be completed as shown above.

Figure 6.22: Calculation of the radial distance \( N \) from the known intersection location \( P_z \).
Once the values of \( n_{\text{Arc}}, d_{\text{Arc}}, \) and \( R_{\text{Arc}} \) are known, they may be used to calculate \( A_{\text{Min}} \) as shown in Figure 6.21. In this figure, \( P_{\text{Min}} \) and \( A \) both lie in the plane of the page, and so \( A_{\text{Min}} \) may be calculated as:

\[
A_{\text{Min}} = -R \cos (\phi)
\]  

(6.23)

where \( \phi \) is the angle between \( A \) and the vector \( R \) from the sphere center to \( P_{\text{Min}} \) (rad).

The value of \( \phi \) is obtained as the summation of two angles, \( \xi \) and \( \zeta \). The angle \( \xi \) lies between the vertical direction and a vector extending from the sphere center to \( P_{\text{Min}} \), and may be calculated as a function of the known horizontal and vertical components \( R_{\text{Arc}} \) and \( d_{\text{Arc}} \) (see Figure 6.21):

\[
\xi = \tan^{-1}(d_{\text{Arc}}/R_{\text{Arc}})
\]  

(6.24)

The angle \( \zeta \) lies between vertical and \( A \), and may be calculated with the cosine property of the dot product (and recalling that both \( A \) and \( n_{\text{Arc}} \) are unit vectors):

\[
\zeta = \pi - \cos^{-1}(A \cdot n_{\text{Arc}})
\]  

(6.25)

### 6.5 Calculation of the Radial Bounds

Following identification of the axial bounds of intersection between the cutter and the current intersected segment, a radial distance value is calculated for each disc within these bounds. Assuming that a point of contact between the current intersected element and the current axial disc \( P_z \) is known, the radial distance from \( D \) to that point along \( N \) may be calculated using the scalar product:

\[
N = N \cdot (P_z - P_1)
\]  

(6.26)

This relation is displayed in Figure 6.22. As \( N \) and \( D \) are orthogonal, no coupling exists, and it is not necessary to use \( P_{\text{CL}}(D_2) \) in this equation, although the intersection with \( P_z \) may not have occurred at \( P_1 \).
Due to the discrete Z-buffer element spacing, it is possible for the STE to extend further into the stock along N than calculated. Although modeled as lines, Z-buffer elements represent a volume and have a square area associated with them, with the line at the squares center. To compensate for this, the final values of $N_{\text{Max}}$ and $N_{\text{Min}}$ are adjusted by $(+d_{XY}/2)$ and $(-d_{XY}/2)$ respectively. A 'smart' error compensation method, similar to that used in preventing gaps in the contact area between disc quadrants, was attempted, but was not found beneficial. In particular, when $\mathbf{A}$ and $Z_{\text{wp}}$ are nearly orthogonal, any axial disc even marginally engaged in the stock was modeled as having full engagement. This effect is not a problem when addressing the quadrant issue, as in that case the potential 'extra material' is included only when cutting is known to occur on both quadrants I and II on the current disc, it proves overly conservative in this case, where the entire disc may be modeled as cutting the stock even if a single intersection is found. Due to the relatively tight Z-buffer element spacing, and the conservative nature of the quadrant/gap algorithms when $\mathbf{A}$ and $Z_{\text{wp}}$ are near orthogonal, the simple half-mesh adjustment method was found to be sufficient.

6.6 Calculation of the Per-Disc Intersection Location $P_Z$

6.6.1 Per-Disc Intersection Locations

It is simple to calculate the radial distance value to some Z-buffer/Disc intersection location $P_Z$ once that location is known. While this location has been calculated for the ends of the intersected segment, it is not known for the discs that lie between the axial bounds, and it must be calculated for each disc.

One simple approach of obtaining the $P_Z$ locations is to implement a Monte Carlo approach, where a great deal of $P_Z$ locations are randomly defined along the intersected segment, for which $N$ values and axial locations are calculated using Equations (6.26) and (6.7) respectively. However, this method can result in no $P_Z$ being defined for some discs that do intersect the stock, requiring use of a large number of $P_Z$ locations. This is very inefficient, resulting in multiple calculations of $N$ per disc, when only one $N$ value is required (per intersected segment). Therefore more efficient methods that consider the current cutter shape and orientation are desired, so that $P_Z$ may be calculated as a function of axial location.
6.6.2 The Dropped-Disc Approach to Per-Disc Intersection Calculation

In earlier research [H94], each axial disc of the discrete cutter model was used to define a unique STE for intersection calculation. Starting with the highest axial disc that could possibly engage the stock, an STE containing only the volume swept by the current disc was defined and intersected with the Z-buffer elements. The final entrance and exit angles for the disc were subsequently calculated. This was then repeated for all remaining discs, incrementing down the cutter axis (Figure 6.23).

The primary benefits of this approach were its simplicity and robustness in providing the data required for calculation of the entrance and exit angles for each disc, independent of the cut geometry calculations for other discs. Because intersections were calculated with an STE defined specifically for each disc, radial distance calculation was straightforward. The required location $P_z$ was the calculated intersection location, and the max and min radial distances were identified for that disc alone, allowing for immediate calculation of entrance and exit angles. Also, no explicit axial location calculation was required, as the axially highest disc that intersects the stock was defined as the axial depth of cut.

While the method works well, it is inefficient, as a complete STE description was required for each disc element. This requires an intersection zone and a full set of intersections be performed $N_D$ times per move, where $N_D$ is the number of discs that could potentially intersect the stock. Additionally, $N_D$ was estimated using the value of $C_{max}$ defined in Chapter 5, which provides a conservative estimate for the upper axial bound prior to any intersection calculations. Therefore the top-most STEs typically did not intersect the stock, although they had to be defined and checked for intersections. Another drawback to this method was reduced geometric accuracy when modeling ball end cutters, as the constant disc radius of the discrete cutter model produces 'stepping' or 'terracing' in the geometric model when modeling cutters with axial curvature. This lead to inaccuracies in the cut geometry parameters for ball end cutters, particularly in regions previously created using the dropped disc method (see Figure 6.24).

6.6.3 The Swept Circle Approach to Per-Disc Intersection Calculation

While the above method provides accurate results for $P_z$, a new approach was developed in order to improve efficiency. In this approach, an STE of sufficient axial length to include all discs that contact the stock is defined and intersected once with all Z-buffer elements affected by the current tool move. For each
Figure 6.23: In a previous approach at finding the Max and Min bounds in the Axial and Normal directions, each axial disc element was intersected with the Z-buffer elements by modeling it as an individual STE. This provided accurate but inefficient results. However, this approach may prove useful in the modeling of cutters whose profiles must be represented in a piecewise manner, such as the odd shaped cutter shown at the right.

Figure 6.24: Comparison of the "dropped disc" STE series (left) vs. a single net toolpath STE (right); both as seen looking down the D axis. Note that during operation the discs of the 'dropped disc' method are used individually for intersections, rather than together as shown.

Figure 6.25: In the 'swept circle' approach at identifying points of contact between an intersected segment and the appropriate axial discs, only a circle is swept at each axial disc location, rather than a net STE for each disc. This is done following an intersection calculation between an STE for the net toolpath and the Z-buffer model.
element intersected, an intersected segment is defined for use in the calculation of cut geometry parameters. The axial bounds of contact between the intersected segment and the discrete cutter model are then calculated. The $P_z$ locations for the bounding discs are known, allowing for calculation of radial distance, and the $P_z$ locations for the remaining discs are calculated through intersection with a single geometric primitive, the swept circle (see Figure 6.25). Using this approach, the number of intersection calculations is minimized, and the calculation of cut geometry parameters is limited to discs known to intersect the stock.

Once the bounding discs are have been identified, a loop through all discs that lie between them is performed, and a swept circle is defined for each. Defining a circle of the appropriate radius for the current disc, and sweeping this circle along $D$, allows for calculation of the point of intersection known to exist with the intersected segment. This provides the desired $P_z$ location, as well as $D_z$, required in some calculations, and the algorithm used is the same required to model the bottom of flat end cutters and bound the tops of all STEs. In place of the initial tool position $P_i$, generally used in this equation, the value of the 'initial tool position' for the current axial disc is calculated as:

$$P_i(i) = P_i + A(i)A$$  \hspace{1cm} (6.27)

where $P_i(i)$ is the initial 'cutter position' of the current ($i^{th}$) axial disc, and $A(i)$ is the axial offset of the current disc from the cutter location position.

Once $P_i(i)$ is known, the swept circle intersection code is called and a maximum of two intersection locations and corresponding $D_z$ values are calculated. If two values are found, the intersection location belonging to the lesser $D_z$ value is always used for $P_z$, as it corresponds to an intersection on the leading edge of the circle, where it first intersected the segment. The remaining intersection location may also be required, as explained below. If a single value is found, indicating the sweep was too short for the intersected segment to contact both the leading and trailing edge of the circle, the intersection location is always used as $P_z$ if it occurs with the leading edge of the circle (in Quadrants I, II). If the single intersection occurs on the trailing edge of the circle (Quadrants III, IV), the method described below is also required. Note that prior to any additional calculations that use these intersection locations, they should be checked against gaps in the intersected segment to ensure that they occur with solid material, and are valid.
Intersections with the trailing edge of the circle can only be valid when the current disc resides on the cutter ball and \((D \cdot A) < 0\). In this case, the validity of trailing edge intersections is determined using the intersection location \(P_z\) and the corresponding \(D_z\) value to calculate the vector \(\mathbf{ZCL}\) (Equation 6.4). If \((\mathbf{ZCL} \cdot \mathbf{D}) > 0\), then the intersection occurs on the ingress side of the cutter, at a location where the cutter is moving into the stock, and the intersection on the trailing edge is valid and can be used in the calculation of \(N\). If \((\mathbf{ZCL} \cdot \mathbf{D}) < 0\), the intersection is invalid as it exists on the egress portion of the cutter. If \((\mathbf{ZCL} \cdot \mathbf{D}) = 0\), technically the intersection occurs, but moot as it results in a zero chip thickness.

This method is more efficient than the dropped disc approach as fewer calculations are required. The greatest increase in efficiency is due to a large decrease in the actual number of intersection calculations required. Under this approach, each Z-buffer element in the intersection zone and each of the 5 geometric primitives in the STE are intersected only once per tool move, as opposed to \(N_D\) times per move in the dropped disc method. Additionally, it is known that each swept circle will intersect the intersected segment, and so no additional intersection zone calculations are required. A number of other supporting calculations required per STE is also reduced, such as STE definition, intersection validation, and updating of the Z-buffer model. It should be noted, however, that the dropped-disc method could still provide an important means to calculate intersections and cut geometry parameters for cutting tools that have a complex profile. Representing such tools piecewise by a series of individual STEs may provide the only feasible method of modeling them (see Figure 6.23).
CHAPTER 7

CNC MACHINE MODELING AND RELATIVE VELOCITY CONTROL FOR
5-AXIS CNC MILLING

7.1 Chapter Introduction

This chapter presents the techniques used for modeling the CNC machine. This is required for parsing the G-code file in the appropriate manner, performing Machine to Workpiece coordinate transformations, and for calculation of the cutter to stock relative velocity values. The machine model consists of a kinematic model and a CNC controller model. The machine kinematics define the motion of a given machine, which is an attribute of the physical construction of the machine. The kinematic model is required to perform the coordinate transformations, and to relate individual axis velocities to net relative velocity. The CNC controller consists of hardware and software that parses the input G-code data, and ensures that the NC mill carries out the commands contained in that file. The controller model is required for G-code parsing, and to calculate individual axis velocities based on input commands. Both machine kinematics and controller behavior are typically unique to a given brand and model of CNC machine, so a unique derivation of a machine-specific CNC machine model is generally required.

7.2 Machine Coordinate to Workpiece Coordinate Transformations

The ‘Machine coordinate’ cutter positions contained in the G-code file must be transformed into the ‘Workpiece coordinates’ of the Z-buffer model for material removal simulation. Workpiece coordinates are used to provide a common basis for the intersection calculations, independent of machine type. In 3-axis milling, \( Z_M \) and \( Z_{WP} \) are typically parallel by definition, as the stock model only represents the top surface of the milled part, and so in this case the transformation is purely translational. In 5-axis milling, however, the cutter is free to change its orientation relative to the stock through the use of rotary axes. However, this rotational motion contributes to cutter motion while changing the orientation, and so 5-axis milling simulation requires both translational and rotational transforms to map the cutter location and orientation from machine into workpiece space.
milling simulation requires both translational and rotational transforms to map the cutter location and orientation from machine into workpiece space.

The transformation from machine to workpiece coordinates is a function of the machine coordinate origin location, the fixed kinematic arrangement of the axes, and the current state of the axes. These transformation routines are included in a self-contained machine model responsible for both coordinate transformation and relative velocity calculation to limit redundancy and aid in ensuring the appropriate routines are called for the current machine type. A separate model is required for each machine type to be simulated, and the user must specify the current machine type. An example of the required transformations for a Fadal table-on table 5-axis arrangement is presented in Appendix B. Once the transformation has been performed for a given tool move, the results may be used in the definition of the swept toolpath envelope (STE), as described in Chapter 5.

7.3 The Relative Velocity Vector $V_{net}$

The relative velocity vector, $V_{net}$, describes the difference in velocity that exists between the cutting tool and stock during a given tool move (see Figure 7.1). Accurate relative velocity calculation is critical as it directly affects chip thickness, which in turn directly affects the cutting force estimates. In reality it is not the feedrate that is being controlled to maintain desired force thresholds, but the relative velocity; feedrate only provides a CNC machine input through which relative velocities may be controlled.

To simplify calculations, $V_{net}$ is defined relative to machine coordinates, as the machine kinematics are most easily described in this system because this system is unique to each machine type. In this system both the cutter and stock may be in motion relative to the floor that the NC machine is mounted on. However, only the magnitude of $V_{net}$ is of interest in mechanistic modeling. This allows all relative motion to be modeled in workpiece coordinates, where the stock is fixed and all motion occurs in the cutter. The directional component is included in workpiece coordinates through the direction vector $D$, calculated independently from G-code data, as presented in Chapter 5.

In 3-axis milling, $V_{net}$ is typically equal to the input feedrate value. However, in 5-axis milling, the presence of rotary axes results in a variable $V_{net}$ that can differ greatly from the input feedrate.
Figure 7.1: The relative velocity vector $V_{\text{net}}$ describes the motion of the cutter relative to the stock, and is defined in the Machine Coordinate system of the G-code data.

Figure 7.2: When purely linear motion occurs, the vector sum of the linear velocity components yields the net velocity vector independent of position or orientation.

Figure 7.3: The linear velocity components produced by rotational motion depend on the distance from the center of rotation (left), and the current angular position (right).
Additionally, the magnitude of $V_{net}$ can vary as a function of axial location on the cutter. Therefore, a nominal value of $V_{net}$ is first calculated for the cutter location position (the position indicated in the G-code file), and this value is assumed constant during the current move. The magnitude of $V_{net}$ is then calculated for other axial locations relative to the cutter location value (also considered constant during the move).

In addition to milling force control, the relative velocity calculations may be used in a stand-alone fashion to provide predictable relative velocities. In this mode, the machine model is used to calculate variable feedrates that compensate for machine kinematics and controller behavior to arrive at a user-defined constant relative velocity.

7.4 Calculation of the Relative Velocity Vector

7.4.1 General Description

Calculation of $V_{net}$ requires the use of a CNC machine model, which contains a model of the machine kinematics and the CNC controller. The kinematic model is used to determine the value of $V_{net}$ as a function of the individual axis velocities, and the current kinematic state of the machine. The controller model is needed to calculate the individual axis velocities as a function of the input feedrate value and tool move data. Determining the linear velocity vector produced by each of the machine axes, and summing them, calculates $V_{net}$. Because they are fundamentally different, velocity contributions from linear axes and rotational axes are presented separately. Note both machine kinematics and controller behavior are specific to a given type of NC mill, and so a unique model is generally required for each machine type.

7.4.2 Linear Axis Contributions

In machinery with purely linear degrees of freedom, e.g. 3-axis NC mills, calculation and control of $V_{net}$ is straightforward. A velocity vector is defined for each degree of freedom (i.e. each machine axis), the sum of which provides $V_{net}$ (See Figure 7.2). Even in this simple case, it is necessary to have a kinematic model that describes the machine motion in order to calculate the individual velocity vectors. However, the kinematic model is static over all tool moves due to the constant relationship between the linear axes.
In order to calculate the individual velocity contributions for each axis, the CNC controller model is required. The controller calculates axis velocities as a function of input feedrate, machine constraints (e.g. saturation speeds), and algorithms that consider machine kinematics. Controllers may also include algorithms for other machine specific considerations that manufacturers think may be helpful. For example, Boston Digital 3200 series controllers use different values for the Y-axis distance traveled, dependent on the direction of A axis rotation, rather than using the actual distance moved. This must all be included in the controller model to ensure that accurate axis velocity values are calculated.

It is fairly trivial to ensure accurate relative velocity control in machines with purely linear DOFs. In these machines, \( V_{\text{net}} \) is generally equal to the input feedrate value, except when a given axis saturates at its maximum allowable speed. To arrive at \( V_{\text{net}} \), first the move time is calculated using:

\[
\Delta T = \frac{D}{f} \tag{7.1}
\]

where \( \Delta T \) is the time taken for a single tool move to occur (min),

\( D \) is the net linear distance between the start and end positions for the move (in), and

\( f \) is the programmed input feedrate value (in/min).

The individual axis velocities are then calculated by dividing the distance each axis must translate by the required move time. For example, the linear velocity of the X-axis, \( V_X \), may be calculated using:

\[
V_X = \frac{\Delta X}{\Delta T} \tag{7.2}
\]

where \( \Delta X \) is the distance the X axis translates during the current tool move (in).

The velocities for the remaining axes are then calculated in an identical manner.

Note that these velocity calculations do not consider axis acceleration. The acceleration limits are defined values in CNC controllers, and they could be obtained either from the manufacturer or through experimentation. It was found during this research that when no ‘rapid’ moves were present, the estimated versus predicted mill times generally experienced less than 5% deviation, indicating the small role of acceleration when feeds do not vary appreciably. However, when a large number of rapid moves were
present, the deviation became significant, increasing to as much as 20%, with the percentage deviation increasing as a function of the number of 'rapid' moves present in a part file. This signifies that the effects of acceleration can become significant when large variations in feed are present (the 'rapid' moves could be set to either 150 or 300 IPM, while typical feeds were on the order of 5-25 IPM). While not implemented in this research, inclusion of acceleration represents an area where additional work could be performed.

7.4.3 Rotary Axis Contributions

The use of 5-axis machines introduces rotational degrees of freedom as a result of the rotary axes required to vary cutter orientation. This complicates velocity control, as the velocity contribution from the rotary axes varies as a function of both the radial distance from the center of rotation, and the current angular position (see Figure 7.3). Additionally, these radial distances and angular positions typically change every tool move. To deal with these effects, the current state of the NC machine must be considered for accurate calculation of $V_{net}$ during the current tool move. Unfortunately, most CNC controllers do not consider machine kinematics when calculating axis velocities. Many 5-axis controllers use the same algorithm for axis velocity calculation originally developed for 3-axis mills, where a move time $\Delta T$ is calculated, based on the linear move distance and input feed and ignoring the rotary axes contributions. Therefore, as $V_{net}$ is a function of the variable kinematic state of the machine, a varying relative velocity typically results from a constant input feedrate, resulting in a loss of velocity control in 5-axis machines. However, through modeling of the machine kinematics and controller behavior, the relative velocity that results from a given feedrate value, for a given tool move, may be calculated.

7.5 Kinematic Modeling for Calculation of Relative Velocities

7.5.1 General Definition

The kinematic model describes the interdependencies between axes, and the current state of each axis, for each tool move. Note that $V_{net}$ is only constant over a tool move during 3-axis milling (neglecting acceleration). When rotary axes are present, $V_{net}$ can vary continuously as a result of varying distances to the center of rotation. Therefore the set of constant $V_{net}$ values provided by the kinematic model (one per
move) is actually a discrete approximation. Typically, 5-axis tool moves are short to maintain geometric accuracy, and so the variation in \( V_{net} \) is generally small. However, to minimize error, the radius values and angular positions present at the middle of the move are used when calculating \( V_{net} \). This provides an 'average' representation of the kinematic state during the move, and is calculated using:

\[
P_M = P_i + \frac{\Delta P}{2}
\]  

\( P_M \) is the 'average' machine position for a given axis (\( X_M, Y_M, Z_M, A_M \) or \( B_M \)), 

\( P_i \) is the initial G-code cutter location of that axis for the current move (in), and 

\( \Delta P \) is the total distance translated by the axis during the current tool move (in).

As explained in Chapter 5, large rotational moves are subdivided into a set of smaller moves during simulation in order to maintain geometric accuracy. When calculating the radius values and angular positions needed by the kinematic model, the average machine position for each sub-move is used. However, the individual axis velocities calculated by the controller model require the use of tool move data for the net move, as this is the information supplied to the actual CNC controller. These axis velocities are then assumed constant over the net move, with all variation occurring in the kinematic data only.

Although applied in workpiece coordinates, \( V_{net} \) is calculated in machine coordinates because only the magnitude of \( V_{net} \) is desired, which is independent of the coordinate system. To calculate \( V_{net} \), its components are defined through summation of the contributions from each machine axis in machine space:

\[
V_{netX} = V_x + V_{Ax} + V_{Bx}
\]  

\( 7.4a \)

\[
V_{netY} = V_y + V_{Ay} + V_{By}
\]  

\( 7.4b \)

\[
V_{netZ} = V_z + V_{Az} + V_{Bz}
\]  

\( 7.4c \)

where \( V_{netX}, V_{netY}, V_{netZ} \) are the principal direction components of \( V_{net} \) (in/min), 

\( V_x, V_y, V_z \) are the individual linear axis velocity contributions (in/min), and 

\( V_{Ax,y,z}, V_{Bx,y,z} \) are the linear velocity contributions from the rotary axes (in/min).
Note that some of these contributions may be zero. This commonly occurs when a rotary axis rotates about a principal direction, in which case it can make no velocity contribution in that direction.

The linear velocity contributions from the rotary axes may be found using the vector products:

\[ \mathbf{V}_A = (\mathbf{Adot} \times \mathbf{R}_A) = \{V_{Ax}, V_{Ay}, V_{Az}\} \]  
\[ \mathbf{V}_B = (\mathbf{Bdot} \times \mathbf{R}_B) = \{V_{Bx}, V_{By}, V_{Bz}\} \]  

where \( \mathbf{Adot} \), \( \mathbf{Bdot} \) are rotational velocity vectors for the A and B axes, (rad/min), and \( \mathbf{R}_A \), \( \mathbf{R}_B \) are radius vectors from the axis of rotation to the cutter location position (in).

The task of calculating the rotary axes velocity contributions is simplified by finding the location of the center of rotation for the A and B axes, denoted \( \mathbf{P}_A \) and \( \mathbf{P}_B \) respectively. Using A as an example, this allows the required radius vector for that axis to be calculated as (see Figure 7.4, on left):

\[ \mathbf{R}_A = (\mathbf{P}_M - \mathbf{P}_A) \]  

where \( \mathbf{P}_M \) is the current (average) cutter location position in machine coordinates, and \( \mathbf{P}_A \) is the location of the center of rotation in machine coordinates.

While Equations (7.5) and (7.6) provide a valid solution for the relative velocity contributions at a single position along the cutter axis, they are inefficient when the relative velocity of multiple axial locations is desired. In this case it is more efficient to solve for the linear velocity contributions as a function of axial location, using the component of \( \mathbf{R}_A \) that lies in the plane of A rotation. Only this component results in linear velocity, and it may easily be solved for as a function of axial location, as presented in section 7.7. This is the method actually implemented.

The in-plane radius vector (i.e. radial distance) may be defined through the use of a 'Rotary Axis' coordinate system. This system has its origin at some known point on the axis of rotation (either \( \mathbf{P}_A \) or \( \mathbf{P}_B \)), one principal axis along the axis of rotation, and the remaining two principal axes at a known orientation relative to machine coordinates. The radial distance components are then simply the Rotary Axis Coordinate locations of \( \mathbf{P}_M \), as shown on the right in Figure 7.4. This allows for a solution of the two...
Figure 7.4: The radius vectors may be defined either as 3D Machine Coordinate vectors, or as 2D vectors in local 'Rotary Axis Coordinates'. In both cases the A axis is shown as an example.

Figure 7.5: The table-on-table 5-axis arrangement used in this research. The stock translates in X and Y, and rotates in A and B, while the cutting tool moves independently in Z.

Figure 7.6: The A rotary axis rides on the B rotary axis, and so the direction of A is a function of B.
components of linear velocity that exist in the Rotary Axis coordinates through scalar multiplication of the rotational velocity magnitude with the Rotary Axis components of radial distance. Using A as an example, this scalar operation is performed as:

\[ V_{AX} = Adot \cdot Ry_A \]  \hspace{1cm} (7.7a)
\[ V_{AY} = -Adot \cdot Rx_A \]  \hspace{1cm} (7.7b)

where \( V_{AX}, V_{AY} \) are the linear velocity components from the A axis velocity vector (in/min), \( Adot \) is the magnitude of the A axis rotational velocity vector (rad/min), and \( Rx_A, Ry_A \) are the \( X_{RT} \) and \( Y_{RT} \) rotary table components of \( R_A \) (in).

The signs on the velocity vectors are assigned to define the velocity of the cutter relative to the stock. These velocity vectors must then be transformed back into machine coordinates for summation in Equation (7.4).

While slightly more tedious, this approach provides an efficient solution when \( V_{net} \) is desired at multiple axial positions on the cutter. This is achieved by transforming the cutter orientation into Rotary Axis coordinates, allowing for solution of \( Rx \) and \( Ry \) as a function of axial location. Additionally, the Rotary Axis cutter locations are pre-calculated as intermediate values during the Machine to Workpiece coordinates transformation, and so no additional calculations are required (see Appendix B).

7.5.2 Application Example: The Table-on-Table 5-axis Configuration

In this research, a ‘table-on-table’ 5-axis configuration, in which a 2-axis rotary tilt table is mounted on a standard 3-axis CNC mill, is used (see Figure 7.5). In this arrangement, B rotates about \( Y_M \), and so the B axis rotational vector, in machine coordinates, is:

\[ \mathbf{B}_{RM} = \{0,1,0\} \]  \hspace{1cm} (7.8)

As shown in Figures 7.5 and 7.6, A is chained with B, riding on a platform that rotates with B and remains orthogonal to B at all times. The A axis rotation vector is therefore constrained to the \( X_M, Z_M \) plane, and is
a function of the B axis location. The origin of B axis rotation (B=0) is defined as the position where \( \mathbf{A}_{\text{Ret}} = \{0,0,1\} \) in machine coordinates, and so the A axis rotation vector is defined in machine coordinates as:

\[
\mathbf{A}_{\text{Ret}} = \{\sin(B), 0, \cos(B)\}
\]  \(7.9\)

These vectors indicate the directions of the rotational velocity vectors in Equations (7.5a,b). While this method is replaced by the local Rotary Axis coordinate system method, these vectors are still useful as they describe the orientation of the Rotary Axis coordinates in Machine space. This knowledge allows the velocity vectors calculated in Rotary Axis space to be transformed back to Machine space for application.

The radial distances required for the linear velocity are calculated through transformation of the cutter location into Rotary Axis coordinate systems for the A and B axes, denoted Rotary\textsubscript{A} and Rotary\textsubscript{B} respectively (see Figure 7.7). Definition of these systems, and the cutter location transformation, is presented in detail in Appendix B. Conveniently, these transformations are intermediate steps in the transformation of cutter positions from Machine to Workpiece coordinates. As shown in Figure 7.8, the in-plane radius vector from the center of B rotation is simply the cutter location in Rotary\textsubscript{B} coordinates:

\[
\mathbf{R}_B = \{X_B, 0, Z_B\}
\]  \(7.10\)

Note the Rotary\textsubscript{B} system is defined to remain fixed relative to Machine coordinates, and does not rotate with B. Therefore the Rotary\textsubscript{B} principal directions may be defined parallel to Machine coordinates, allowing the Rotary\textsubscript{B} velocity vectors to require no additional transformation to Machine coordinates. The Machine coordinate B axis velocity vector required by Equation (7.4) is therefore:

\[
\mathbf{V}_B = \{-Z_B, 0, X_B\} \ (B\text{dot})
\]  \(7.11\)

Note a +B rotation with a +Z radius results in a negative relative velocity of the cutter past the stock. Similarly, the in-plane radius vector \( \mathbf{R}_A \) is the simply the Rotary\textsubscript{A} cutter location (see Figure 7.9):
Figure 7.7: The use of Rotary Axis Coordinates can simplify linear velocity calculations as the radius vector required is reduced to 2D and may be defined in terms of principal axis positions.

Figure 7.8: The RotaryB cutter position provides the desired radius vector $\mathbf{R}_B$. Note that the RotaryB coordinates do not rotate with the B axis (i.e. $X_B$ remains parallel to $X_M$).

Figure 7.9: The cutter position in RotaryA provides the radius vector $\mathbf{R}_A$. While RotaryA rotates with the B axis (on left), they do not rotate with A (i.e. $X_A$ remains in the $X_M, Z_M$ plane).
Using $R_A$, the Rotary$_A$ linear velocity vectors may be calculated through scalar multiplication with the magnitude of $\mathbf{Adot}$. However, in this case the mapping of these velocity vectors to Machine coordinates is not so direct. This is where the vector $A_{Rot}$ may be used. As $A_{Rot}$ describes the Machine coordinate orientation of a principal axis in Rotary$_A$ coordinates (the axis of $A$ rotation), it may be used to map the radial distance back into Machine Coordinates. First, note $Y_A$ and $Y_M$ remain parallel at all times, and the Rotary$_A$ coordinates do not rotate with $A$ (i.e. $X_A$, $Z_A$ remain in the $X_M$,$Z_M$ plane). As the velocity vector requiring rotational transformation ($\mathbf{Xdot}_A$) is orthogonal to $A_{Rot}$, the Machine coordinate $A$ axis velocity contribution may be calculated using:

$$
\mathbf{V}_A = \{(Y_A)\cos(B), -X_A, (Y_A)\sin(B)\} \ (\mathbf{Adot})
$$

(7.13)

This may be seen in Figure 7.9. The linear velocity contributions from the $A$ and $B$ axes are now known, and may be combined with the linear axis velocities in equation (7.4) to arrive at $\mathbf{V}_{set}$.

### 7.6 The Controller Model

#### 7.6.1 Individual Axis Velocity Calculation

In this research, a Fadal 3-axis vertical CNC machining center equipped with a Jones and Shipman two-axis rotary tilt table is used for 5-axis machining (see Figure 7.5). The Fadal controller is relatively simple, designed to drive the 3-axis machine, and it has no built in velocity compensation for the rotary axes. Some controllers, such as Boston Digital’s 3200 series, have a crude form of rotational velocity compensation. However, even in this case a fixed radius from the center of rotation is assumed, providing only nominal improvement.

The Fadal controller sets axis velocities in 5-axis mode the same as it does in 3-axis mode. This is performed using the net distance moved by the linear axes, $D_{xy}$, which is calculated as:
\[ D_{xyz} = (\Delta X^2 + \Delta Y^2 + \Delta Z^2)^{0.5} \]  
(7.14)

where \( \Delta X, \Delta Y, \) and \( \Delta Z \) are the distances moved by the individual linear axes (in), e.g.:

\[ \Delta X(m) = X_M(m) - X_M(m-1) \]  
(7.15)

where \( m \) is an index indicating the current move number.

The time required for this move to occur, \( \Delta T \), may then be calculated using \( D_{xyz} \) and the input feed value:

\[ \Delta T = D_{xyz} / f \]  
(7.16)

where \( f \) is the current input feedrate value (in/min).

The individual axis velocities may then be defined using the value of \( \Delta T \):

\[ Xdot = \Delta X / \Delta T \]  
(7.17)

\[ Ydot = \Delta Y / \Delta T \]

\[ Zdot = \Delta Z / \Delta T \]

\[ Adot = \Delta A / \Delta T \]

\[ Bdot = \Delta B / \Delta T \]

where \( Xdot, Ydot, \) and \( Zdot \) are the linear axis velocities (in/min), and \( Adot, Bdot \) are the rotary axis velocities (deg/min).

Note that the rotational velocities must be re-defined in units of radians per minute when calculating \( V_{rot} \). Only in situations where purely rotational motion is present does the move time consider rotational motion. In this situation, the angular distance moved by the axis with maximum rotation is divided by the current feed value, which is assumed to be degrees per minute. The move time calculated is then used in the definition of the individual axis rotational velocities.

Once calculated, the axis velocities are used to define \( V_{rot} \), provided none exceeds the maximum allowable value for that axis. For accurate simulation of machine behavior, the calculated axis velocities must be compared against their maximum allowable values. When one or more axis exceeds its maximum allowable velocity, the axis that most exceeds its maximum is used to limit all axis velocities. This is
achieved through calculation of the $\Delta T$ value that results in the limiting velocity, constraining the axis to its allowable maximum. For example, if only $A$ exceeds its maximum allowable velocity (denoted $A_{\text{dotMax}}$), the limiting $\Delta T$ is calculated as:

$$
\Delta T = \frac{\Delta A}{A_{\text{dotMax}}}
$$

All axis velocities would then be set using this limiting value of $\Delta T$, applying it in Equation (7.17).

Note that saturation modeling is not required when only relative velocity control is desired. In this situation, when saturation occurs the machine cannot maintain the desired relative velocity, the limiting velocities will result by default regardless of input feed. However, it is required for accurate estimation of milling time, which is useful as the estimates may be used in the bidding and scheduling of NC jobs. This time may also be used in the validation of new machine models, through comparison of predicted to actual mill times. When defining axis velocities, the controller model calculates the time required to complete each move. Summation of these times provides an accurate estimate of the net milling time. If axis velocity saturation is not considered, move time estimates will be too short.

7.6.2 G-code Data Acquisition

In addition to axis velocity calculation, the controller model is also required for the parsing of G-code files, and translation of the machine specific position data contained in the G-code into a generic format used by the remaining software. Although G-code commands are fairly standardized, it is not uncommon for a given code to be interpreted differently by different controllers. It is important that the controller software correctly interpret commands for the machine being simulated. Once a specific G-code command has been read, it is translated into a generic command format that is then passed to the other routines of the simulator. This allows a single version of the simulator to be used for all machine types.

When reading a G-code file, each line is loaded individually in a text format, and parsed for recognized commands. Recognized commands are immediately translated into the generic format, while unknown commands trigger an error sequence to terminate program execution. This is done as a precaution to ensure accurate geometric modeling, because skipping an unrecognized positional command would
result in an incorrect STE definition when the next recognized position information was read. Once translated, some commands are ignored, as they are irrelevant to relative velocity control (e.g. coolant on/off). Other commands are used to define model parameters, (e.g. coordinate system origin, spindle speed). However, the vast majority of information in a G-code file is machine tool position data.

When machine position data is read, the axis values are copied from text into real-valued variables for translation into Workpiece coordinate cutter location and orientation. The distance moved by each axis is also calculated at this time through subtraction of the previous position data for each axis. As the current line of G-code is eventually written out to the updated G-code file that contains the newly calculated feedrates, the text format is required so that each line may be output as text. This ensures the original position data is not corrupted. All calculated feeds are appended to this line prior to output.

Prior to altering feed values for a tool move, it must be checked to ensure it's a valid change. During some moves, such as approaches/departures to/from the stock, re-vectors (when the tool re-orient itself in the air between cutter passes), and ‘rapid’ tool moves, the feed does not require adjustment. For example, during approaches/departures, there is typically no axis rotation (therefore \( V_{nt} = \) input feed), and a specific feed is used to ensure positional accuracy. No force estimation or feed calculation is performed for these moves, although geometric modeling is still required. Other known conditions may also be flagged and checked during G-code input, such as slot cuts (which require a reduced output feed to limit tool wear).

### 7.7 Relative Velocity as a Function of Axial Location

The presence of rotary motion on 5-axis mills causes \( V_{Net} \) to vary with axial location, as the radial distance to the center of rotation typically varies along the cutter axis (except when the cutter orientation and axis of rotation are parallel) (see Figure 7.10). It is therefore desirable to model \( V_{Net} \) as a function of axial location, i.e., \( V_{Net}(a) \), so that the variations may be reflected in the feed-per-tooth estimates when using the mechanistic model. This allows for a better chip thickness distribution approximation along the cutter axis, which directly affects the milling force estimates for each axial disc. However, the method presented earlier calculates \( V_{Net} \) only at a single axial location, the cutter location position. While this
Figure 7.10: The relative velocity can vary with axial location on the cutting tool during 5-axis milling as a result of the varying radius from the center of rotation.

\begin{align*}
V_y(a_0) &= -R_{CL} \cdot Adot \\
V_y(a_i) &= -(R_{CL} - dR) \cdot Adot \\
V_y(a_i) &= -(R_{CL} - a_i \sin(B)) \cdot Adot \\
V_y(a_i) &= -R_{CL} \cdot Adot + a_i \cdot \sin(B) \cdot Adot \\
V_y(a_i) &= V_{CL} + \sin(B) \cdot Adot \cdot a_i \\
V_y(a_i) &= V_{CL} + (\text{Trig}_{A,Y} \cdot Adot) \cdot a_i
\end{align*}

Figure 7.11: Calculation of the $Y_M$ component of linear velocity (which extends out of the page) due to A axis rotation as a function of axial location and the cutter location velocity for a table-on-table 5-axis arrangement.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
approach may be used repeatedly to solve for \( V_{Net} \) at multiple axial locations, a more efficient method is
needed.

An efficient means of calculating \( V_{Net}(a) \) is to first determine \( V_{Net} \) at the cutter location position, \( V_{CL} \), and then solve for \( V_{Net}(a) \) as a function of \( V_{CL} \) and the radial distance from rotational centers. \( V_{Net}(a) \) may be solved for relative to \( V_{CL} \) using a relation of the form:

\[
V_{net}(a) = V_{CL} + (dR_A(a)A\dot{a} + dR_B(a)B\dot{a})
\]

(7.19)

where \( dR_A(a) \), \( dR_B(a) \) are functions describing the variations radial distance from the center of A and B rotation to the current axial location (in),

\( A\dot{a}, B\dot{a} \) are the magnitudes of the A and B axis rotational velocities (rad/min), and

\( a \) is the current axial location (in).

The exact functions required to define \( V_{net}(a), V_{CL}, dR_A(a), \) and \( dR_B(a) \) are machine dependent. These functions are based on the current kinematic state of a particular machine, but once defined for a given tool move, Equation (7.19) may be applied uniformly to all discs (and for all feed iterations). The use of Equation (7.19) reduces calculations for other axial locations to only the difference in radii that exists at those locations, denoted \( dR_A(a) \) and \( dR_B(a) \) for A and B respectively. An example of their derivation, including relevant coordinate systems, is provided in the following section.

A single component (i.e. \( X_m, Y_m, \) or \( Z_m \)) of the net radial distance from a center of rotation to the current axial location may be calculated as a function of the radial distance to the cutter location position \( (R_{CL}) \), the axial offset from the cutter location position to the current axial disc \( (a) \), and trigonometric relations that project the current axial offset into the plane of rotation (“Trig”):

\[
R(a) = (R_{CL} + \text{Trig} \cdot a)
\]

(7.20)

where \( R(a) \) is the net radial distance along a principal direction to the current disc (in).

An equation of this form is required for each principal direction that contains a component in the plane of rotation, yielding the desired relations \( dR_A(a) \) and \( dR_B(a) \). Equation (7.19) may then be re-written as:
\[ V_{\text{set}}(i) = V_{\text{CL}} + (\text{Trig}_A \cdot \text{Adot} + \text{Trig}_B \cdot \text{Bdot}) \cdot a, \quad (7.21) \]

Where \( \text{Trig}_{A,B} \) are vectors mapping the axial offset \( a \) onto principal directions in the planes of \( A \) and \( B \) rotation.

Note Equation (7.21) is a discrete form of Equation (7.19), providing \( V_{\text{set}} \) for each axial disc. More importantly, this equation separates the three primary sources of change that necessitate the re-calculation of velocity: change in kinematic state, change in feed, and change in axial location. Kinematic variation occurs every tool move, and the method presented above allows parameters dependent on kinematics (\( \text{Trig}_{A,B} \)) to be isolated and calculated once per move. \( V_{\text{CL}} \) must be recalculated each feed iteration, which also provides the axis velocities, but these remain constant over all axial discs. Once these calculations have been performed, compensating for axial variations is simply a matter of solving Equation (7.21) for each engaged disc, with the \( V_{\text{CL}} \) and \( (\text{Trig}_A \cdot \text{Adot} + \text{Trig}_B \cdot \text{Bdot}) \) terms remaining constant over all discs.

The development of \( \text{Trig}_{A,B} \) will now be presented for the Fadal table-on-table 5-axis arrangement. In this setup, the cutter orientation remains fixed along \( Z_M \), \( B \) remains fixed to rotate about \( Y_M \), and \( A \) is chained to (rides on) \( B \), rotating about an axis orthogonal to \( Y_M \) (see Figure 7.6). This results in no axial variation in the \( Z_M \) velocity component, as the cutter axis remains fixed in this direction. Changes in velocity can only occur in \( X_M \) and \( Y_M \) which lie orthogonal to the axial variation in \( Z_M \). Therefore \( \text{Trig}_{A,Z} \) and \( \text{Trig}_{B,Z} = 0 \) (i.e. axial variations do not affect the velocity component in \( Z_M \)).

As \( B \) rotates about \( Y_M \), \( B \) rotation produces linear velocity components only in \( X_M \) and \( Z_M \), and therefore \( \text{Trig}_{B,Y} = 0 \) (note the \( Z_M \) velocity component produced by \( B \) rotation also doesn’t vary with axial location, as mentioned above, and is only included in \( V_{\text{CL}} \)). As the cutter axis lies along \( Z_M \) fixed in the plane of \( B \) rotation, variations in the \( X_M \) velocity component due to \( B \) rotation varies proportionally with changes in axial location. Any change in axial location increases the radial distance from the center of rotation by an equal amount, and \( \text{Trig}_{B,X} = -1 \). This may be seen in Figure 7.10, where the \( X_M \) velocity vectors extend orthogonal to the axis of the cutter. The negative sign is required as the stock rotates with \( B \), and therefore a \(+B \) rotation (and a \(+Z_M \) axial location) results in a \(-X_M \) velocity of the cutter relative to the stock.
Because \( A \) is chained with \( B \), \( A \) rotation is capable of producing velocity components in \( X_M, Y_M, \) and \( Z_M \). As mentioned above, the \( Z_M \) velocity component does not vary with axial location. \( B \) rotates about \( Y_M \), and the \( B \) axis origin (\( B=0 \)) is defined to exist where the axis of \( A \) rotation lies along \( Z_M \). Calculation of \( \text{Trig}_{A,X} \) and \( \text{Trig}_{A,Y} \) occurs in \( \text{Rotary}_A \) coordinates to simplify solution, allowing the radii to be defined orthogonal to the axis of \( A \) rotation (\( \text{Rotary}_A \) coordinates are shown in Figure 7.7). The radius in the \( Y_A \) direction can produce \( X_M \) and \( Z_M \) velocity components, dependent on the current \( B \) location. The \( Y_A \) direction remains orthogonal to \( Z_M \) and therefore the cutter axis, for all \( B \), and so changes in axial location do not affect the \( X_M \) component of velocity produced by \( A \) rotation; therefore \( \text{Trig}_{A,X} = 0 \). The radius in \( X_A \) produces the \( Y_M \) velocity component from \( A \) rotation. When \( B = 0 \), the \( X_A \) direction lies parallel to \( X_M \), however as \( B \) rotates away from the origin, the \( X_A \) direction obtains a \( Z_M \) component. The \( Y_M \) component of velocity therefore changes with axial location as a function of the sine of \( B \), and \( \text{Trig}_{A,Y} = \sin(B) \) (see Figure 7.11). In summary, for the Fadal table-on-table machine configuration,

\[
\text{Trig}_A = \{0, \sin(B), 0\} \tag{7.22a}
\]

\[
\text{Trig}_B = \{-1, 0, 0\} \tag{7.22b}
\]

These relations may now be used in Equation (7.21), along with \( V_{CL} \), to solve for \( V_{rel}(i) \), the relative velocity of the cutter past the stock for each axial disc when modeling the Fadal milling center.

### 7.8 Standalone Relative Velocity Control

#### 7.8.1 Software implementation of Relative Velocity Control

In addition to providing the cutter velocity data required for chip thickness estimation, relative velocity calculation methods may also be implemented in a 'standalone' fashion. In this mode, variable feedrates that maintain a desired relative velocity during 5-axis milling are calculated, effectively canceling the unpredictable variations in velocity that result from rotary motion. This greatly reduces uncertainties in the feed definition process, and the improved control results in reduced mill times and fewer broken tools. During relative velocity control, only the velocity at the cutter location position is of interest.
The compensation software may be implemented either ‘on-line’, in the machine controller, or ‘off-line’, where the G-codes are processed on a computer prior to usage. When implemented ‘off-line’, a feed value that produces the desired relative velocity \( V_{\text{Des}} \) is calculated and appended to positional commands in the G-code file as required, and this G-code is later used to mill the desired part. In the ‘on-line’ case, the G-code is input as usual to the controller, which then performs the relative velocity control during milling. Most controllers today have a closed architecture, making control algorithms inaccessible, and so on-line relative velocity control is currently not practical. However, with the recent increase in commercially available open-architecture controllers, this may not always be the case.

In the off-line approach, position commands are identified and analyzed individually. For each move, the required feed values are solved for iteratively. Note that during relative velocity control only the magnitude of \( V_{\text{net}} \), i.e. \( V_{\text{net}} \), is of interest. If the initial ‘seed’ feedrate doesn’t produce a \( V_{\text{net}} \) equal to \( V_{\text{Des}} \) (within some allowable tolerance), the feed is adjusted appropriately, and the process continues. When \( V_{\text{Des}} \) is achieved, the current feed value is appended to the current line of G-code, which is then output to an ‘updated’ version of the G-code file (provided it differs from the previously output feed value). To minimize the iterations required, the ‘seed’ feedrate used during the first iteration is equal to the feed value calculated for the previous tool move, as this can result in \( V_{\text{Des}} \) during the first iteration in some cases. The value of \( V_{\text{net}} \) is used as the seed value for the first tool move in the G-code file.

Typically, \( V_{\text{Des}} \) is not achieved on the first iteration, and so it is necessary to iteratively adjust the feedrate such that \( V_{\text{Des}} \) is approached. The most direct method for arriving at \( V_{\text{Des}} \) requires only two iterations. The first iteration uses the seed value described above. During the second iteration, a new feed is calculated via linear interpolation from the first iterations feed value, using the ratio of \( V_{\text{Des}} \) to \( V_{\text{net}} \):

\[
\text{Feed}(i) = \text{Feed}(i-1) \times \frac{V_{\text{Des}}}{V_{\text{net}}(i-1)}
\]

(7.23)

where \( i \) indicates the number of the current iteration, 2 in this case.

This typically results in \( V_{\text{Des}} \) during the second iteration, as many controllers calculate axis velocities as a linear function of a single move time value. This in turn is a linear function of feedrate, and therefore
relative velocity is generally a linear function of feedrate for a given tool move. Note that the relation is non-linear between moves as a result of the rotary motion, which introduce trigonometric relationships.

Unfortunately, the linear relationship of Equation (7.23) does not hold in the general case, even over a single move. While the relationship between feedrate and axis velocities is typically linear, in some controllers this relationship is only piecewise linear, achieved through the use of ‘IF’ statements in the controller software (this is the case in the Boston Digital controller). Nonlinear conditions also exist to meet machine-imposed or user-defined constraints, such as maximum allowable axis velocities. Therefore, a more general method of solution is required to ensure robust operation, although linear interpolation should always be used on the first iteration to maximize efficiency (even in instances where it does not provide \( V_{\text{Des}} \), it should provide an appropriate ‘seed’ feed value for the more generic iterative solver). The general iterative approach used is the bisection method. This method was also implemented for automatic feedrate selection, and is discussed in Chapter 3. The flow chart of Figure 7.12 provides a graphical presentation of the basic operation of the relative velocity control software.

### 7.8.2 Results of the Standalone Model

Simulation results from standalone relative velocity control will now be presented for the Fadal table-on-table arrangement. These results are generated for one pass of the cutting tool during the milling of the ‘saddle surface’ shown in Figure 7.13. This surface has full three-dimensional variation in both cutter location and orientation, and areas of both rapid and gradual changes in cutter orientation.

Figure 7.14 displays simulation results for the \( V_{\text{Net}} \) values produced by a constant input feedrate of 20 IPM. Note \( V_{\text{Net}} \) varies continuously, and it never achieves the 20 IPM of the input feedrate. Note in particular the spike in velocity that occurs at approximately 18 seconds. This rapid increase in velocity could result in cutter or part damage if unnoticed when test cutting the part. This spike indicates the worst-case conditions that exist during the cut. Using traditional feed selection techniques, a skilled machinist would identify this condition and limits the feedrate to produce acceptable results over this region. Unfortunately this feed would generally apply to the entire milling job, or a large portion of it, as it is not practical to manually identify worst-case velocity conditions for each tool move. Due to this limitation, the overall cut occurs at a slow feedrate, resulting in a loss in efficiency.
Start Program

Read User Data And Open Input, Output Gcode Files

Read 1 Line of G-code, Store Position Data

Feed Modifiable Command?

Yes (Position Data, Initial Feed)

CNC Machine Model
Kinematic Model

Controller Model

(New Feed)

(Axis Velocities)

(Radii)

Calculate Relative Velocity

VRel

In Desired Range?

(VDes +/- Tol)

Yes

Feed Value Differ From Previous?

No

Yes

Append New Feed To Current Line of Gcode

Write Current Line of Gcode To Updated File

Last G-code Command?

No

Yes

End Program

Figure 7.12: Flow chart overview of 'standalone' relative velocity control software.

Figure 7.13: Constant velocity simulation results are shown for first pass in the 5-axis "saddle surface".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 7.14: Plot of simulation results for a constant input velocity of 20 IPM and the resultant relative velocity of the cutter past the stock.

Figure 7.15: Plot of simulation results for a constant relative velocity of 20 IPM and the feed values required to maintain that velocity.

Figure 7.16: Simulation results of the feed values required for a constant relative velocity of 8 IPM.
Figure 7.15 displays simulation results for the feedrates necessary to maintain a constant 20 IPM relative velocity. As a result of relative velocity control, the milling time was reduced 68%, from 25 seconds to 8 seconds. Unfortunately, this does not represent a realistic comparison. Although the constant 20 IPM feedrate may have produced acceptable results, a relative velocity of 20 IPM is 43% greater than the maximum $V_{\text{Net}}$ of 14 IPM that is produced by the constant 20 IPM input feedrate. A 20 IPM relative velocity is therefore not an acceptable alternative to a constant 20 IPM input feedrate in this situation.

A more realistic basis for comparison would be to define $V_{\text{Des}}$ somewhere in the 4 to 14 IPM range that was experienced during the constant 20 IPM feedrate. A reasonable estimate for a safe relative velocity value is 8 IPM. Figure 7.16 displays simulation results for the feedrates necessary to maintain this constant 8 IPM. Note that the milling time is approximately 18 seconds, which represents a 29% reduction in milling time as compared to the constant 20 IPM feedrate. This is substantial, as some parts can take hundreds of hours to mill, at prices in the range of $100.00/hour. Additionally, the maximum relative velocity is reduced 42%, from 14 to 8 IPM, resulting in less stress on the cutter and part.

While the constant relative velocity of 20 IPM in Figure 7.15 does not provide a realistic comparison versus a constant 20 IPM feedrate, it does provide an example of axis velocity saturation modeling. Note that the relative velocity occasionally dips below the desired range of 20 IPM ($\pm 0.25$) during the course of the cut. The cause of some of these reductions in relative velocity is readily apparent in the feedrate values, also shown in that figure. The maximum allowable (non-'rapid') input feedrate is 100 IPM with the Fadal controller. When feeds exceeding this value are required to maintain $V_{\text{Des}}$, the input feed is restricted to its maximum and $V_{\text{Des}}$ cannot be achieved. The remaining reductions in relative velocity, those near 0 sec. and also near 3.5 sec. in particular, are not so apparent. These are the result of A axis velocity saturation, as shown in Figure 7.17. The maximum rotational velocity of the Jones and Shipman rotary table is 1500 deg./min. At the start of the cut, the A axis must rotate from its home position at $A = 0^\circ$ to the initial rotational position required at the start of the move. This large rotational move requires an axis velocity greater than 1500 deg./min, as calculated by the controller model to meet the requirements for the current input feedrate value and tool move length. However, the desired rotational velocity cannot be achieved and the A axis rotational velocity saturates at 1500 deg./min, with the remaining axes inhibited accordingly. Saturation occurs again in the middle of the pass, when the A axis
Figure 7.17: Plot showing A axis saturation at +/- 1500 deg/min, which limits the ability to maintain the desired relative velocity of 20 IPM.

Figure 7.18: When the B axis hits its hard limit of =5°, the A axis must rotate 180° and the direction of milling is reversed for the cutter to maintain the desired cutting path.
must reverse direction as a result of physical machine constraints. The physical limitation is that the B axis
it is only capable of $-5^\circ$ rotation, although it may rotate $+105^\circ$. Therefore, when 'more' than $-5^\circ$ of rotation
is required to achieve the desired cutter orientation, the A axis must rotate $\sim180^\circ$, and the direction be cut
reversed, so that the required B axis tilt achieved in the positive direction (see Figure 7.18). This large A
rotation relative to the other axes again results in high A axis velocities.

In addition to lab testing, this method was also implemented for production use in an industrial
environment. The relative velocity control problem arose during talks with our industrial partner, a
manufacturing company specializing in the 5-axis milling of turbomachinery. For this implementation,
Boston Digital model 405, 505, and 605 CNC machines were modeled. This required the development of
software that would control the relative velocities for these various machine types, and tailored to work
within the specific milling paradigms used at the site. This included the identification of tool moves for
which feeds were not to be changed (e.g. approaches, retracts, re-vectors), among other things. While the
CNC machine models differs between the Fadal and various Boston Digital machines, the same software
kernels are used to parse user data, calculate relative velocity, iterate to the desired relative velocity, and
perform other common tasks. This demonstrates the robustness and modularity of the integrated modeling
approach; only new machine models are required to support additional CNC machine types.

As a result of relative velocity control, our partner is reporting average milling time reductions on
the order of 25%. Constant relative velocities have also reduced cutter breakage, which further increases
the actual time savings, and reduces part damage. Additionally, the process of selecting feedrates has been
simplified, which results in reduced time spent defining feeds with greater confidence in the results.
CHAPTER 8

MECHANISTIC MATERIAL CONSTANT CALCULATION

8.1 Chapter Introduction

This chapter describes the methods used to obtain the mechanistic material constants required by the generalized discrete mechanistic model. These constants relate the area of stock material being sheared by a cutting flute segment to the tangential and radial components of cutting force that act on the flute.

The material constants are empirically derived, and are unique to a specific stock material and cutting tool. Through measurement of the forces experienced during a series of controlled cutting tests, it is possible to back-calculate the constants required by the mechanistic model. These tests must be repeated for each combination of cutting tool and stock material that require milling force estimation or feedrate generation. This approach of using empirically derived constants makes it possible to apply the same mechanistic model to many different material types. The mechanistic material constants are also a function of the current cut conditions, and so the tests should be prepared to approximate a typical cutting condition that is present for type of milling to be simulated (e.g. rough cutting or finish milling).

8.2 The Mechanistic Material Constants

In the generalized discrete mechanistic model of Chapter 2, the parameters $K_T$ and $K_R$ that are required by Equation 2.16 relate the area of material being sheared by the cutter to tangential and radial cutting force components. The parameter $K_T$ determines the current cutting force component that lies tangent to the cutter for a given flute segment, per unit length, as a function of the current chip thickness. It essentially represents specific cutting energy for a particular cut [KDL82]. The parameter $K_R$ determines radial force component acting on the flute segment as a ratio of the tangential force value.

While the values of $K_T$ and $K_R$ are capable of estimating the cutting forces present for a given set of cutting conditions, they are not sufficient to describe the forces produced over a broad range of

191

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
conditions, and are particularly sensitive to changes in feedrate. As a result, $K_T$ and $K_R$, various methods were developed to calculate the material constants as a function of various parameters that define the cutting conditions, such as axial depth of cut, radial depth of cut, feedrate, etc. [KDL82]. The function that was finally settled upon is an exponential function of the average chip thickness, which combines the effects of several cut parameters into a single value [KD83]. Using this method, the values of $K_T$ and $K_R$ are found as shown in Equation (2.12) of chapter 2, repeated here:

$$K_T = (K_{TC}) (h_{avg})^{-P_1}$$  \hspace{1cm} (2.12a)

$$K_R = (K_{RC}) (h_{avg})^{-P_2}$$  \hspace{1cm} (2.12b)

While $K_T$ and $K_R$ actually appear in the mechanistic model, it is the values of $K_{TC}$, $K_{RC}$, $P_1$, and $P_2$ that are the desired mechanistic material constants (the 'material constants') that are empirically obtained. These material constants are then used in Equation (2.12) to solve for the values of $K_T$ and $K_R$ required for a given set of cutting conditions, as indicated by the average chip thickness value.

The use of empirically based material constants is not limited to the DeVor and Kline approach. Other mechanistic modeling approaches also require the use of material constants derived from empirical cutting force data [YP90],[YS93],[BYK94],[YA96]. Although an ideal approach would relate the milling force coefficients to intrinsic material properties (e.g. shear strength, yield strength, absolute strength, and/or elasticity), the complexities of the milling process have so far prevented the development of such relations. It has been found that the primary variables affecting the process parameters are related to variations in chip thickness [T76],[KDL82],[KD83] and cutter geometry [AW85],[BYK94], and therefore the majority of the methods calculate constants as function of the empirical data and these parameters.

**8.3 Data Acquisition for Mechanistic Constant Calculation**

The material constants are calculated from measured cutting force data that is collected during cuts made under controlled conditions. The force data is measured in both the X and Y directions as defined in
the mechanistic model, where X indicates the direction of travel of the cutting tool, and Y lies orthogonal to X and the central axis of the cutting tool. These are the force directions predicted by the mechanistic model.

Even with the use of Equation (2.12), it is difficult to obtain very accurate force estimates over a wide range of cutting conditions. To improve the accuracy of the constants, the empirical data is obtained over an operating range of conditions similar to those for which the force estimates are desired. These conditions include spindle speed, feedrate, and axial and radial depths of cut. If force estimation or automatic feedrate selection is desired for a given part (or class of parts), the constants are calculated using average conditions found during the major operations (e.g. rough or finish milling) for that part. The spindle speed selected for the test cuts should also represent the actual cutting conditions of interest.

The range of feedrates used during data collection could be obtained either from prior experience, or using the METCUT "CUTDATA" computerized machining data system [M84] (or similar). The latter method could supply a means for automating test cut generation. The CUTDATA system provides acceptable surface speed & feed-per-tooth values, specified primarily to limit cutter wear and prevent damage, for different stock and cutting tool materials and various cutting conditions. This data is provided for specific given axial and radial depths of cut. Note that these depths are generally known quantities, as toolpaths are commonly specified to maintain some depth of cut criteria. If the typical range of these depths is known for a given job, the CU DATA information may be used to bound the feed values that may be set during automatic feedrate generation to an acceptable range. The feed-per-tooth values supplied by the CUTDATA system can also provide maximum allowable chip thickness information, as the feed-per-tooth value indicates the maximum chip thickness that could result from a given feedrate and spindle speed.

When performing the test cuts, extra precautions are taken to limit sources of error in the force data. This includes a minimizing cutter runout, so that an accurate representation of the average forces is attained. While runout compensation is included in the mechanistic constant calculation software, minimizing physical runout provides a better measure of the average force data, producing more accurate constants. This is of particular concern as the actual runout value and locating angle are typically not known, and approximate values are generally used. The runout parameters may then be treated as ‘fine tuning’ variables in the mechanistic model in order to achieve accurate force estimates for different cutting tools (of the same size and type), or to aid in obtaining conservative results.
The empirical data collection method is kept simple to reduce error and aid in practical application of the technology. This includes a minimizing the amount of data required to calculate accurate constants. The method developed for data acquisition explicitly considers the inclusion of ball end cutters, which require a slightly different approach at constant calculation, as presented in section 8.4. It was found that obtaining force data over four cuts for a flat end cutter, or eight test cuts for a ball end cutter, using partial immersion straight line cuts, provides adequate data to calculate constants for a given operating range.

For a ball end cutter, two sets of four test cuts each are performed. One set occurs at an axial depth of one cutter radius (i.e. on the cutter ball only), and the other occurs at twice that depth (i.e. the cutter ball, plus an equal axial length of the cylindrical portion). At a given axial depth, the feedrate is varied between each of the 4 cuts, and the radial depth is held constant at some value that approximates the typical cutting conditions that feeds are to be generated for. The same feeds and radial depths are used at both axial depths. Note that when using a ball end cutter, the term 'stepover distance' is more appropriate than 'radial depth'. The stepover distance is the distance that the cutting tool is stepped over in a direction orthogonal to the direction of travel between two adjacent test cuts. The stepover distance can provide a more general description of the cutting conditions as the radial depth of cut can vary as a function of axial location due to the variable ball radius (Figure 8.1). On a flat end cutter, the radial distance remains constant over the axial length of the cutter, and is equal to the stepover distance.

When using a flat end cutter only four test cuts are made, each at a different feed value that is defined to span the desired operating range. If the flat end cutter is to be used in constant Z-level, constant stepover roughing, a common usage, the axial and radial depths used in the test cuts are easily obtained, being the known change in Z between levels and the stepover distance. If the cutter is to be used for some technique that results in highly variable cutting conditions, an axial depth of one radius and a radial depth that is in the range of the average radial depth of cut is used, similar to the case for ball end mills.

During the course of this research, the primary industrial application was feedrate generation for rough and semi-finish machining of bladed parts. To remain in an operating range similar to the conditions found during these operations, a radial depth of 0.7 times the cutter diameter was applied; the axial depths used were as described above. If cutting conditions used in a given job differ greatly from these conditions, then the axial and radial depths for the test cuts should be adjusted accordingly.
Stepover Distance

Figure 8.1: Radial distances vary continually in the positive and negative Normal direction on the cutter ball, and so the stepover distance may better describe the cut conditions. Note that on cylindrical cutter portions, the radial distance and the stepover distance are equal.

Figure 8.2: Certain sets of cutting conditions result in average X milling force that is around zero. In the above case, the average force value is -0.1283 Lb. This should be avoided.

Figure 8.3: An axial depth of one radius includes only the cutter ball portion of the cutter in the cut, while an axial depth of one diameter, for a constant stepover, includes the same cutter ball portion, plus an equal axial length of the cylindrical portion of the cutter.
It should be noted that certain cutting conditions result in near-zero X direction (i.e. the direction of travel) average forces as a result of force cancellation (see Figure 8.2). This produces inaccurate material constant values and should be avoided. If the cutting conditions to be simulated result in near zero average X forces, the step-over distance should be adjusted slightly to eliminate this condition. For the case of a ball end cutter, ball depth cuts at a radial depth of cut of 0.5-0.6x Diameter result in this condition.

A complete description of the protocol used in performing the test cuts required for collecting empirical cutting force data for the purpose of mechanistic constant calculation is presented in Appendix D.

8.4 Mechanistic Constant Calculation

8.4.1 General Calculation Approach

Once the empirical data has been collected, calculation of the mechanistic constants may occur. This is separated into two distinct steps. First, values of $K_T$ and $K_R$ are calculated as a function of the force data and the corresponding cutting conditions. Second, values of $K_{TC}$, $K_{RC}$, $P_1$, and $P_2$ are calculated as a function of the average chip thickness produced by a given set of cutting conditions, and the corresponding $K_T$ and $K_R$ values.

The average of the force data collected during each test cut, along with the cutting conditions specific to each cut, is used to calculate unique values of $K_T$ and $K_R$ for that cut. Using the data collection method presented above for a ball end cutter, this results in eight $K_T$, $K_R$ pairs, four at the axial depth of one cutter radius, and four at twice this depth. Four $K_T$, $K_R$ pairs exist for a flat end cutter.

The method for mechanistic material constant calculation for ball end cutters differs slightly from that for flat end cutters, in order to better isolate the contributions from the different regions on a ball end cutter (i.e. ball and shank). Previously, a single set of constants was generated and applied to the entire axial length of the cutter. While this approach is retained for flat end cutters, two sets of material constants are generated for ball end cutters. One set applies to the cutter ball, and the other to the cylindrical shank. This helps isolate the effects of the differing cutter and flute geometries on these two portions of the cutter. This separation between the two sets of constants occurs in the calculation of $K_T$ and $K_R$ for the two regions, as opposed to during the final $K_{TC}$, $K_{RC}$, $P_1$, $P_2$ calculations, and is performed as follows.
Recall that an identical set of feed values is used at the axial depth of one radius (i.e. ball depth only) and the axial depth of one diameter (i.e. ball and cylinder). For the ball depth cuts, the as-measured average forces at the four different feedrates are used in the calculation of $K_T$ and $K_R$ values that apply to the cutter ball only. The cutting force contributions due to the cylindrical portion of the cutter are then isolated, and used in the calculation of $K_T$ and $K_R$ values that apply only to force estimation on the cutter cylinder. The average force values acting on the cylindrical portion of the cutter are calculated by subtracting the forces generated at an axial depth of one radius from the forces generated at a radial depth of one diameter. This effectively subtracts the forces acting on the ball portion of the cutter from the net forces, leaving only those forces acting on the cylindrical portion (see Figure 8.3). The differential average force values are then used in the calculation of $K_T$ and $K_R$ for the cylindrical portion of the cutter.

Once $K_T$ and $K_R$ have been calculated for all test cuts, they are used in the calculation of the material constants $K_{TC}$, $K_{RC}$, $P_1$, and $P_2$. All four pairs of $K_T$ and $K_R$ are used in the calculation of a single set of constants for a flat end cutter. For ball end cutters, the four sets of $K_T$ and $K_R$ calculated for the cutter ball portion are used to generate one set of material constants, and the remaining four sets calculated for the cylindrical portion are used to generate a second set. Both sets of material constants are used during force estimation, with set usage being determined by the axial location of the current disc requiring force estimation (i.e. ball portion or cutter shank).

8.4.2 Calculation of $K_T$, $K_R$ for a Given Set of Cutting Conditions

The values of $K_T$ and $K_R$ are solved for using the mechanistic model, treating the measured forces and known cutting conditions as input data. Recall that the mechanistic model predicts instantaneous force values at a specific cutter rotation angle, for a given set of cut conditions. However, as the mechanistic model represents a numerical solution to a closed form integral, a discrete form of the mean value theorem may be applied to obtain $K_T$ and $K_R$ using the average of the measured force data. This is applied in the case of the discrete mechanistic model through summation of the forces experienced at all discrete rotation
angles, divided by the number of discrete rotational positions, or:

\[
\overline{F_x} = \frac{1}{N_\theta} \left[ \sum_{j=1}^{N_\theta} F_x(j) \right] \\
\overline{F_y} = \frac{1}{N_\theta} \left[ \sum_{j=1}^{N_\theta} F_y(j) \right]
\]

(8.1a)

(8.1b)

where \( \overline{F_x} \) and \( \overline{F_y} \) are the average force values (Lb),

\( N_\theta \) is the number of discrete rotational positions, and

\( F_x \) and \( F_y \) are the instantaneous force values of Equation (2.16) (Lb).

The use of Equations 8.1a,b allow for solution of \( K_T \) and \( K_R \) for each test cut. As \( K_T \) and \( K_R \) are not a function of rotation angle, they may be removed from inside the summation. When this is done and the summation is executed, Equations 8.1a and 8.1b form a linear system of equations from which \( K_T \) and \( K_R \) are obtained.

8.4.3 Calculation of \( K_{TC} \), \( K_{RC} \), \( P_1 \), and \( P_2 \)

Once \( K_T \) and \( K_R \) have been calculated for each test cut in a given data set (e.g. a ball end cutter ball, a ball end shank, or a flat end cutter), the values of the material constants \( K_{TC} \), \( K_{RC} \), \( P_1 \), and \( P_2 \) may be obtained. The material constants are calculated as a function of the \( K_T \) and \( K_R \) values, and an estimation of the average chip thickness value, \( h_{avg} \) during each test cut in the data set. The value of \( h_{avg} \) is estimated for each test cut using Equation (2.17). Note that when calculating \( h_{avg} \) on the cutter ball portion of a ball end cutter, the entrance and exit angles are variable due to the variable radius value. Therefore a value of \( h_{avg} \) must be calculated for each axial disc on the cutter ball, and these are then summed and divided by the number of cutter ball discs to arrive at a value of \( h_{avg} \) for the net cutter ball.

For a given data set, the values of \( K_T \) and \( h_{avg} \) from each test cut are used to find the value of \( P_1 \), while \( K_R \) and \( h_{avg} \) are used to obtain \( P_2 \). Previously the force data from pairs of test cuts were used, allowing two sets of \( K_T \) and \( K_R \) to be obtained. The value of \( P_1 \) or \( P_2 \) was then found using the ratio of the \( K_T \) or \( K_R \).
values respectively, as applied in Equation (2.12). For example using two values of $K_T$ and $h_{avg}$ from two different test cuts, denoted by the additional subscripts 1 and 2, this ratio method appears as:

$$\frac{K_{T1}}{K_{T2}} = \frac{K_T C \left(h_{avg1}\right)^{-P_1}}{K_T C \left(h_{avg2}\right)^{-P_1}}$$  \hspace{1cm} (8.2)

If the feedrate is doubled between the two test cuts, the feed per tooth value, and therefore the average chip thickness, also doubles. In this situation, the value of $P_1$ found using the ratio method would be:

$$P_1 = \frac{\ln\left(\frac{K_{T1}}{K_{T2}}\right)}{\ln(0.5)}$$  \hspace{1cm} (8.3)

Note that this method calculates the slope between two points on a log-log plot of $K_T$ versus $h_{avg}$, as may be clearly seen when Equation (8.3) is rewritten in the form:

$$P_1 = \frac{\ln(K_{T1}) - \ln(K_{T2})}{\ln(h_{avg1}) - \ln(h_{avg2})}$$  \hspace{1cm} (8.4)

A similar procedure is then performed using $K_R$ to obtain the value of $P_2$. Once the values of $P_1$ and $P_2$ are known, $K_{TC}$ and $K_{RC}$, may be calculated using Equation (2.12).

One drawback to the above method is that scatter appears in plots of $\ln(K_T)$ and $\ln(K_R)$ versus $\ln(h_{avg})$ [KLD82][F87]. This scatter is likely the result of small variations in cutting conditions due to tool wear, changes in cutter runout, and deflection, as well as other slight variances. Therefore, relying on a single slope value to provide the constants limits the range of accuracy provided by them.

In order to improve accuracy over a wider range of cuts, force data from all four test cuts in a data set is used in this research. This is achieved using a linear least squares fit of the data provided, finding the average, minimum error slope that simultaneously best fits all data sets. There are many sources of derivations of the least squares method in both matrix and scalar formats [K72] [B91], and as such it will not be presented here.
Through application of the least squares method, a solution for $P_1$ is obtained:

$$
P_1 = \frac{\sum_{n=1}^{N} \ln(h_{avg-n}) \sum_{n=1}^{N} \ln(K_{T-n}) - N \sum_{n=1}^{N} \ln(h_{avg-n}) \ln(K_{T-n})}{N \sum_{n=1}^{N} \left[ \ln(h_{avg-n}) \right]^2 - \left( \sum_{n=1}^{N} \ln(h_{avg-n}) \right)^2} \tag{8.5}
$$

where $n$ suffix is an index indicating the current data set, and $N$ is the total number of data sets included in the interpolation.

In this application, $N=4$ as there are four test cuts per data set. Once the value of $P_1$ has been calculated, it may be used in Equation (2.12) for the solution of a value of $K_{TC}$ for each data set, using the known values of $K_T$ with their corresponding $h_{avg}$ values. The average of the N values of $K_{TC}$ is then taken to arrive at the final value for this parameter. The values of $P_2$ and $K_{RC}$ are then solved for in an identical manner, only using values of $K_R$ in place of $K_T$ in Equations (8.5) and (2.12).

### 8.5 Examples of Material Constants Found To Date

The use of the above method has proven very accurate at reproducing the non-linear relationship between $h_{avg}$ and $K_T$, $K_R$, as demonstrated in Tables 8.1- 8.7. These tables encompass three stock materials (6061 Al, 6061 T6 Al, and 15-5 Stainless Steel), two cutter materials (High Speed Steel and Carbide), two cutter types (flat end and ball end), and several cutter sizes ranging from 0.1875" to 1.0" in diameter. The tables provide measured average force data, the resulting $K_T$ and $K_R$ values, and the values of $P_1$, $P_2$, $K_{TC}$ and $K_{RC}$ that are required as input by the mechanistic model when force estimates are desired for cutting conditions similar to those used to generate the constants. Also provided are the values of $K_T$ and $K_R$ that result from application of the $K_{TC}$, $K_{RC}$, $P_1$, $P_2$ values calculated, along with $h_{avg}$ in Equation (2.12). Notice that the percent difference between the estimated values of $K_T$ and $K_R$ and those calculated directly from the measured force data are within 5% in all cases. Figure 8.4 shows graphically the estimated versus directly calculated values of $K_T$ and $K_R$ for the case of the 0.375" ball end cutter (the data for this case is provided in Table 8.1).
### CUTTER BALL PORTION, KT ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.000258</td>
<td>1.59e-4</td>
<td>-6.4</td>
<td>19</td>
<td>18134</td>
<td>0.3577</td>
<td></td>
<td>416463</td>
<td>413936</td>
</tr>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>2.76e-4</td>
<td>-8.9</td>
<td>30.4</td>
<td>18134</td>
<td>0.3577</td>
<td></td>
<td>336805</td>
<td>340208</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>3.78e-4</td>
<td>-9.8</td>
<td>38.9</td>
<td>18134</td>
<td>0.3577</td>
<td></td>
<td>302808</td>
<td>304027</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>4.94e-4</td>
<td>-12.6</td>
<td>48.3</td>
<td>18134</td>
<td>0.3577</td>
<td></td>
<td>278348</td>
<td>276165</td>
</tr>
</tbody>
</table>

### CUTTER BALL PORTION, KR ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_R$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.000258</td>
<td>1.59e-4</td>
<td>-6.4</td>
<td>19</td>
<td>0.1501</td>
<td>0.1731</td>
<td></td>
<td>0.1501</td>
<td>0.1731</td>
</tr>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>2.76e-4</td>
<td>-8.9</td>
<td>30.4</td>
<td>0.1501</td>
<td>0.1731</td>
<td></td>
<td>0.1501</td>
<td>0.1731</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>3.78e-4</td>
<td>-9.8</td>
<td>38.9</td>
<td>0.1501</td>
<td>0.1731</td>
<td></td>
<td>0.1501</td>
<td>0.1731</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>4.94e-4</td>
<td>-12.6</td>
<td>48.3</td>
<td>0.1501</td>
<td>0.1731</td>
<td></td>
<td>0.1501</td>
<td>0.1731</td>
</tr>
</tbody>
</table>

### CYLINDER PORTION, KT ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.000258</td>
<td>2.25e-4</td>
<td>-3.5</td>
<td>14</td>
<td>19822</td>
<td>0.3185</td>
<td></td>
<td>292895</td>
<td>287522</td>
</tr>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>4.08e-4</td>
<td>-3.5</td>
<td>21.1</td>
<td>19822</td>
<td>0.3185</td>
<td></td>
<td>228009</td>
<td>238056</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>5.67e-4</td>
<td>-3.8</td>
<td>28.5</td>
<td>19822</td>
<td>0.3185</td>
<td></td>
<td>216894</td>
<td>214290</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>7.49e-4</td>
<td>-3.5</td>
<td>34.7</td>
<td>19822</td>
<td>0.3185</td>
<td></td>
<td>198318</td>
<td>196089</td>
</tr>
</tbody>
</table>

### CYLINDER PORTION, KR ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_R$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>0.000258</td>
<td>2.25e-4</td>
<td>-3.5</td>
<td>14</td>
<td>0.055</td>
<td>0.2795</td>
<td></td>
<td>0.055</td>
<td>0.2795</td>
</tr>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>4.08e-4</td>
<td>-3.5</td>
<td>21.1</td>
<td>0.055</td>
<td>0.2795</td>
<td></td>
<td>0.055</td>
<td>0.2795</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>5.67e-4</td>
<td>-3.8</td>
<td>28.5</td>
<td>0.055</td>
<td>0.2795</td>
<td></td>
<td>0.055</td>
<td>0.2795</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>7.49e-4</td>
<td>-3.5</td>
<td>34.7</td>
<td>0.055</td>
<td>0.2795</td>
<td></td>
<td>0.055</td>
<td>0.2795</td>
</tr>
</tbody>
</table>

Table 8.1: Material Constant Comparison: calculated from force data vs. $K_T$, $P_1$ and $K_R$, $P_2$ estimates.

- **Stock Material**: 6061 Aluminum
- **Cutting Tool**: 0.375 in. Diameter, 4 flute, Ball End Cutter, HSS
- **Modeled Runout**: 0.0001" @ 0°
- **Spindle Speed**: 775 RPM
- **Conditions**: Axial depth 0.1875" for both ball and cylinder, Step-over 0.2625", Coolant On
### CUTTER BALL PORTION, KT ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>havg (in)</th>
<th>Fxavg (Lb)</th>
<th>Fyavg (Lb)</th>
<th>KTC</th>
<th>P1</th>
<th>K_T</th>
<th>Est. K_T</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>3.05e-4</td>
<td>-12.8</td>
<td>46.4</td>
<td>25395</td>
<td>0.3354</td>
<td>400373</td>
<td>383739</td>
<td>-4.15</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>4.07e-4</td>
<td>-14.6</td>
<td>56.3</td>
<td>25395</td>
<td>0.3354</td>
<td>339338</td>
<td>348338</td>
<td>2.65</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>5.23e-4</td>
<td>-16.4</td>
<td>68.1</td>
<td>25395</td>
<td>0.3354</td>
<td>305932</td>
<td>320116</td>
<td>4.64</td>
</tr>
<tr>
<td>6.2</td>
<td>0.002</td>
<td>9.74e-4</td>
<td>-30</td>
<td>119.1</td>
<td>25395</td>
<td>0.3354</td>
<td>266791</td>
<td>259828</td>
<td>-2.61</td>
</tr>
</tbody>
</table>

### CUTTER BALL PORTION, KR ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>havg (in)</th>
<th>Pxavg (Lb)</th>
<th>Pyavg (Lb)</th>
<th>KRC</th>
<th>P2</th>
<th>K_R</th>
<th>Est. K_R</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>3.05e-4</td>
<td>-12.8</td>
<td>46.4</td>
<td>0.402</td>
<td>0.0461</td>
<td>0.5973</td>
<td>0.584</td>
<td>-2.24</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>4.07e-4</td>
<td>-14.6</td>
<td>56.3</td>
<td>0.402</td>
<td>0.0461</td>
<td>0.5741</td>
<td>0.5762</td>
<td>0.38</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>5.23e-4</td>
<td>-16.4</td>
<td>68.1</td>
<td>0.402</td>
<td>0.0461</td>
<td>0.5482</td>
<td>0.5696</td>
<td>3.9</td>
</tr>
<tr>
<td>6.2</td>
<td>0.002</td>
<td>9.74e-4</td>
<td>-30</td>
<td>119.1</td>
<td>0.402</td>
<td>0.0461</td>
<td>0.5636</td>
<td>0.5535</td>
<td>-1.8</td>
</tr>
</tbody>
</table>

### CYLINDER PORTION, KT ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>havg (in)</th>
<th>Fxavg (Lb)</th>
<th>Fyavg (Lb)</th>
<th>KTC</th>
<th>P1</th>
<th>K_T</th>
<th>Est. K_T</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>4.36e-4</td>
<td>-7.1</td>
<td>33.3</td>
<td>17849</td>
<td>0.3538</td>
<td>271824</td>
<td>275745</td>
<td>1.44</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>5.96e-4</td>
<td>-9.2</td>
<td>44.7</td>
<td>17849</td>
<td>0.3538</td>
<td>254557</td>
<td>246961</td>
<td>-2.98</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>7.78e-4</td>
<td>-8.8</td>
<td>51.6</td>
<td>17849</td>
<td>0.3538</td>
<td>220987</td>
<td>224704</td>
<td>1.68</td>
</tr>
</tbody>
</table>

### CYLINDER PORTION, KR ANALYSIS

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>havg (in)</th>
<th>Pxavg (Lb)</th>
<th>Pyavg (Lb)</th>
<th>KRC</th>
<th>P2</th>
<th>K_R</th>
<th>Est. K_R</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.6</td>
<td>0.000516</td>
<td>4.36e-4</td>
<td>-7.1</td>
<td>33.3</td>
<td>0.1575</td>
<td>0.1595</td>
<td>0.5352</td>
<td>0.541</td>
<td>1.09</td>
</tr>
<tr>
<td>2.3</td>
<td>0.000742</td>
<td>5.96e-4</td>
<td>-9.2</td>
<td>44.7</td>
<td>0.1575</td>
<td>0.1595</td>
<td>0.5267</td>
<td>0.5148</td>
<td>-2.27</td>
</tr>
<tr>
<td>3.1</td>
<td>0.001</td>
<td>7.78e-4</td>
<td>-8.8</td>
<td>51.6</td>
<td>0.1575</td>
<td>0.1595</td>
<td>0.4871</td>
<td>0.4933</td>
<td>1.27</td>
</tr>
</tbody>
</table>

Table 8.2: Material Constant Comparison: calculated from force vs. KTC, P1 and KRC, P2 estimates.
Stock Material: 6061 Aluminum
Cutting Tool: 0.5 in. Diameter, 4 flute, Ball End Cutter, HSS
Modeled Runout: 0.000167" @ 103°
Spindle Speed: 775 RPM
Conditions: Axial depth 0.25" for both ball and cylinder, Step-over 0.35", Coolant On
<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>0.000258</td>
<td>1.7e-4</td>
<td>-2.2</td>
<td>7</td>
<td>95085</td>
<td>0.1317</td>
<td>306177</td>
<td>298198</td>
<td>-2.61</td>
</tr>
<tr>
<td>4.8</td>
<td>0.000516</td>
<td>2.86e-4</td>
<td>-3</td>
<td>12.1</td>
<td>95085</td>
<td>0.1317</td>
<td>268870</td>
<td>278424</td>
<td>3.55</td>
</tr>
<tr>
<td>6.9</td>
<td>0.000742</td>
<td>3.88e-4</td>
<td>-3.5</td>
<td>16.5</td>
<td>95085</td>
<td>0.1317</td>
<td>257234</td>
<td>267496</td>
<td>3.99</td>
</tr>
<tr>
<td>9.3</td>
<td>0.001</td>
<td>5.04e-4</td>
<td>-3.9</td>
<td>23.1</td>
<td>95085</td>
<td>0.1317</td>
<td>270288</td>
<td>258422</td>
<td>-4.39</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_P$</th>
<th>$P_2$</th>
<th>$K_P$</th>
<th>Est. $K_P$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>0.000258</td>
<td>1.7e-4</td>
<td>-2.2</td>
<td>7</td>
<td>0.0355</td>
<td>0.3346</td>
<td>0.641</td>
<td>0.6477</td>
<td>1.05</td>
</tr>
<tr>
<td>4.8</td>
<td>0.000516</td>
<td>2.86e-4</td>
<td>-3</td>
<td>12.1</td>
<td>0.0355</td>
<td>0.3346</td>
<td>0.5488</td>
<td>0.5441</td>
<td>-0.86</td>
</tr>
<tr>
<td>6.9</td>
<td>0.000742</td>
<td>3.88e-4</td>
<td>-3.5</td>
<td>16.5</td>
<td>0.0355</td>
<td>0.3346</td>
<td>0.5031</td>
<td>0.4914</td>
<td>-2.31</td>
</tr>
<tr>
<td>9.3</td>
<td>0.001</td>
<td>5.04e-4</td>
<td>-3.9</td>
<td>23.1</td>
<td>0.0355</td>
<td>0.3346</td>
<td>0.4403</td>
<td>0.4502</td>
<td>2.25</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>0.000258</td>
<td>2.36e-4</td>
<td>-1.3</td>
<td>7.4</td>
<td>10620</td>
<td>0.4108</td>
<td>340915</td>
<td>328182</td>
<td>-3.73</td>
</tr>
<tr>
<td>4.8</td>
<td>0.000516</td>
<td>4.18e-4</td>
<td>-1.5</td>
<td>10.3</td>
<td>10620</td>
<td>0.4108</td>
<td>237524</td>
<td>259433</td>
<td>9.22</td>
</tr>
<tr>
<td>6.9</td>
<td>0.000742</td>
<td>5.78e-4</td>
<td>-1.3</td>
<td>14.1</td>
<td>10620</td>
<td>0.4108</td>
<td>231891</td>
<td>227195</td>
<td>-2.03</td>
</tr>
<tr>
<td>9.3</td>
<td>0.001</td>
<td>7.6e-4</td>
<td>-1.1</td>
<td>16.9</td>
<td>10620</td>
<td>0.4108</td>
<td>208065</td>
<td>202994</td>
<td>-2.44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_P$</th>
<th>$P_2$</th>
<th>$K_P$</th>
<th>Est. $K_P$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>0.000258</td>
<td>2.36e-4</td>
<td>-1.3</td>
<td>7.4</td>
<td>0.0742</td>
<td>0.228</td>
<td>0.4851</td>
<td>0.4983</td>
<td>2.71</td>
</tr>
<tr>
<td>4.8</td>
<td>0.000516</td>
<td>4.18e-4</td>
<td>-1.5</td>
<td>10.3</td>
<td>0.0742</td>
<td>0.228</td>
<td>0.4628</td>
<td>0.4374</td>
<td>-5.5</td>
</tr>
<tr>
<td>6.9</td>
<td>0.000742</td>
<td>5.78e-4</td>
<td>-1.3</td>
<td>14.1</td>
<td>0.0742</td>
<td>0.228</td>
<td>0.401</td>
<td>0.4063</td>
<td>1.33</td>
</tr>
<tr>
<td>9.3</td>
<td>0.001</td>
<td>7.6e-4</td>
<td>-1.1</td>
<td>16.9</td>
<td>0.0742</td>
<td>0.228</td>
<td>0.3746</td>
<td>0.3817</td>
<td>1.9</td>
</tr>
</tbody>
</table>

Table 8.3: Material Constant Comparison: calculated from force data vs. $K_T$, $P_1$ and $K_P$, $P_2$ estimates.

Stock Material: 6061 Aluminum
Cutting Tool: 0.1875 in. Diameter, 4 flute, Ball End Cutter, Carbide
Modeled Runout: 0.000125" @ 22°
Spindle Speed: 2325 RPM
Conditions: Axial depth 0.09375" for both ball and cylinder, Step-over 0.2325", Coolant On
### Cutter Ball Portion, $K_T$ Analysis

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{Xavg}$ (Lb)</th>
<th>$F_{Yavg}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.000511</td>
<td>2.63e-4</td>
<td>-26.1</td>
<td>85.3</td>
<td>31693</td>
<td>0.3745</td>
<td>708050</td>
<td>695153</td>
<td>-1.82</td>
</tr>
<tr>
<td>20</td>
<td>0.001023</td>
<td>4.93e-4</td>
<td>-32.7</td>
<td>126.5</td>
<td>31670</td>
<td>0.3745</td>
<td>539092</td>
<td>549091</td>
<td>3.41</td>
</tr>
<tr>
<td>30</td>
<td>0.001534</td>
<td>7.23e-4</td>
<td>-44.1</td>
<td>168.8</td>
<td>31693</td>
<td>0.3745</td>
<td>472032</td>
<td>475641</td>
<td>0.76</td>
</tr>
<tr>
<td>40</td>
<td>0.002045</td>
<td>9.53e-4</td>
<td>-53.3</td>
<td>208.7</td>
<td>31693</td>
<td>0.3745</td>
<td>438312</td>
<td>428854</td>
<td>-2.16</td>
</tr>
</tbody>
</table>

### Cutter Ball Portion, $K_R$ Analysis

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{Xavg}$ (Lb)</th>
<th>$F_{Yavg}$ (Lb)</th>
<th>$K_R$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.000511</td>
<td>3.81e-4</td>
<td>-26.1</td>
<td>85.3</td>
<td>0.292</td>
<td>0.0915</td>
<td>0.6321</td>
<td>0.6207</td>
<td>-1.79</td>
</tr>
<tr>
<td>20</td>
<td>0.001023</td>
<td>4.93e-4</td>
<td>-32.7</td>
<td>126.5</td>
<td>0.292</td>
<td>0.0915</td>
<td>0.5642</td>
<td>0.586</td>
<td>3.86</td>
</tr>
<tr>
<td>30</td>
<td>0.001534</td>
<td>6.06e-4</td>
<td>-44.1</td>
<td>168.8</td>
<td>0.292</td>
<td>0.0915</td>
<td>0.5681</td>
<td>0.568</td>
<td>-0.41</td>
</tr>
<tr>
<td>40</td>
<td>0.002045</td>
<td>7.19e-4</td>
<td>-53.3</td>
<td>208.7</td>
<td>0.292</td>
<td>0.0915</td>
<td>0.5598</td>
<td>0.5517</td>
<td>-1.46</td>
</tr>
</tbody>
</table>

### Cylinder Portion, $K_T$ Analysis

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{Xavg}$ (Lb)</th>
<th>$F_{Yavg}$ (Lb)</th>
<th>$K_T$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.000511</td>
<td>3.93e-4</td>
<td>-12.7</td>
<td>69.2</td>
<td>31977</td>
<td>0.3718</td>
<td>599767</td>
<td>590045</td>
<td>-1.62</td>
</tr>
<tr>
<td>20</td>
<td>0.001023</td>
<td>7.55e-4</td>
<td>-11.9</td>
<td>100.9</td>
<td>31977</td>
<td>0.3718</td>
<td>448235</td>
<td>463148</td>
<td>3.33</td>
</tr>
<tr>
<td>30</td>
<td>0.001534</td>
<td>1.11e-3</td>
<td>-7.9</td>
<td>132.3</td>
<td>31977</td>
<td>0.3718</td>
<td>400335</td>
<td>400466</td>
<td>0.03</td>
</tr>
<tr>
<td>40</td>
<td>0.002045</td>
<td>1.47e-3</td>
<td>-10.1</td>
<td>161.7</td>
<td>31977</td>
<td>0.3718</td>
<td>366605</td>
<td>360814</td>
<td>-1.58</td>
</tr>
</tbody>
</table>

### Cylinder Portion, $K_R$ Analysis

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{Xavg}$ (Lb)</th>
<th>$F_{Yavg}$ (Lb)</th>
<th>$K_R$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.000511</td>
<td>5.73e-4</td>
<td>-12.7</td>
<td>69.2</td>
<td>0.0728</td>
<td>0.2457</td>
<td>0.5016</td>
<td>0.4996</td>
<td>-0.4</td>
</tr>
<tr>
<td>20</td>
<td>0.001023</td>
<td>7.49e-4</td>
<td>-11.9</td>
<td>100.9</td>
<td>0.0728</td>
<td>0.2457</td>
<td>0.43</td>
<td>0.4257</td>
<td>-1.0</td>
</tr>
<tr>
<td>30</td>
<td>0.001534</td>
<td>9.26e-4</td>
<td>-7.9</td>
<td>132.3</td>
<td>0.0728</td>
<td>0.2457</td>
<td>0.3693</td>
<td>0.3867</td>
<td>4.71</td>
</tr>
<tr>
<td>40</td>
<td>0.002045</td>
<td>1.10e-3</td>
<td>-10.1</td>
<td>161.7</td>
<td>0.0728</td>
<td>0.2457</td>
<td>0.3721</td>
<td>0.361</td>
<td>-3.0</td>
</tr>
</tbody>
</table>

Table 8.4: Material Constant Comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates.

**Stock Material:** 6061 T6 Aluminum

**Cutting Tool:** 0.5 in. Diameter, 4 flute, Ball End Cutter, HSS

**Modeled Runout:** 0.000075" @ 97.5°

**Spindle Speed:** 4889 RPM

**Conditions:** Axial depth 0.25" for both ball and cylinder, Step-over 0.35", Coolant On
<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.00075</td>
<td>4.13e-4</td>
<td>-63.4</td>
<td>151.9</td>
<td>10850</td>
<td>0.5572</td>
<td>836841</td>
<td>833433</td>
<td>-0.41</td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>5.26e-4</td>
<td>-69.4</td>
<td>174.9</td>
<td>10850</td>
<td>0.5572</td>
<td>726306</td>
<td>728685</td>
<td>0.33</td>
</tr>
<tr>
<td>10</td>
<td>0.00125</td>
<td>6.38e-4</td>
<td>-73</td>
<td>194.3</td>
<td>10850</td>
<td>0.5572</td>
<td>648812</td>
<td>653996</td>
<td>0.8</td>
</tr>
<tr>
<td>12</td>
<td>0.0015</td>
<td>7.51e-4</td>
<td>-80.3</td>
<td>216</td>
<td>10850</td>
<td>0.5572</td>
<td>601636</td>
<td>597394</td>
<td>-0.71</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_{RC}$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.00075</td>
<td>4.13e-4</td>
<td>-63.4</td>
<td>151.9</td>
<td>0.23</td>
<td>0.1593</td>
<td>0.7974</td>
<td>0.7956</td>
<td>-0.23</td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>5.26e-4</td>
<td>-69.4</td>
<td>174.9</td>
<td>0.23</td>
<td>0.1593</td>
<td>0.7662</td>
<td>0.7656</td>
<td>-0.08</td>
</tr>
<tr>
<td>10</td>
<td>0.00125</td>
<td>6.38e-4</td>
<td>-73</td>
<td>194.3</td>
<td>0.23</td>
<td>0.1593</td>
<td>0.7345</td>
<td>0.7423</td>
<td>1.06</td>
</tr>
<tr>
<td>12</td>
<td>0.0015</td>
<td>7.51e-4</td>
<td>-80.3</td>
<td>216</td>
<td>0.23</td>
<td>0.1593</td>
<td>0.7286</td>
<td>0.7233</td>
<td>-0.73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_T$</th>
<th>Est. $K_T$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.00075</td>
<td>6.05e-4</td>
<td>-94.3</td>
<td>172.3</td>
<td>22193</td>
<td>0.4951</td>
<td>876410</td>
<td>870156</td>
<td>-0.71</td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>7.82e-4</td>
<td>-118.1</td>
<td>203.1</td>
<td>22193</td>
<td>0.4951</td>
<td>763029</td>
<td>766537</td>
<td>0.46</td>
</tr>
<tr>
<td>10</td>
<td>0.00125</td>
<td>9.58e-4</td>
<td>-139.4</td>
<td>229.4</td>
<td>22193</td>
<td>0.4951</td>
<td>681317</td>
<td>693001</td>
<td>1.71</td>
</tr>
<tr>
<td>12</td>
<td>0.0015</td>
<td>1.11e-3</td>
<td>-155.7</td>
<td>260.1</td>
<td>22193</td>
<td>0.4951</td>
<td>646411</td>
<td>637330</td>
<td>-1.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>$h_{avg}$ (in)</th>
<th>$F_{X_{avg}}$ (Lb)</th>
<th>$F_{Y_{avg}}$ (Lb)</th>
<th>$K_{RC}$</th>
<th>$P_2$</th>
<th>$K_R$</th>
<th>Est. $K_R$</th>
<th>% Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>0.00075</td>
<td>6.05e-4</td>
<td>-94.3</td>
<td>172.3</td>
<td>-0.1363</td>
<td>0.9744</td>
<td>0.9841</td>
<td>0.99</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.001</td>
<td>7.82e-4</td>
<td>-118.1</td>
<td>203.1</td>
<td>-0.1363</td>
<td>1.0268</td>
<td>1.019</td>
<td>-0.76</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.00125</td>
<td>9.58e-4</td>
<td>-139.4</td>
<td>229.4</td>
<td>-0.1363</td>
<td>1.0681</td>
<td>1.0477</td>
<td>-1.91</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.0015</td>
<td>1.11e-3</td>
<td>-155.7</td>
<td>260.1</td>
<td>-0.1363</td>
<td>1.0537</td>
<td>1.0722</td>
<td>1.75</td>
<td></td>
</tr>
</tbody>
</table>

Table 8.5: Material Constant Comparison: calculated from force data vs. $K_{TC}$, $P_1$ and $K_{RC}$, $P_2$ estimates.
Stock Material: 15-5 Stainless Steel
Cutting Tool: 0.5 in. Diameter, 4 flute, Ball End Cutter, Carbide
Modeled Runout: 0.000175" @ 92°
Spindle Speed: 2000 RPM
Conditions: Axial depth 0.25" for both ball and cylinder, Step-over 0.35", Coolant On

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>( h_{avg} ) (in)</th>
<th>( F_{X_{avg}} ) (Lb)</th>
<th>( F_{Y_{avg}} ) (Lb)</th>
<th>( K_T )</th>
<th>( P_1 )</th>
<th>( K_T ) Est.</th>
<th>( K_T ) % Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.0005</td>
<td>4.46e-4</td>
<td>-2.7</td>
<td>40</td>
<td>395</td>
<td>0.8938</td>
<td>398626</td>
<td>390158</td>
</tr>
<tr>
<td>7.5</td>
<td>0.00075</td>
<td>6.15e-4</td>
<td>-5.5</td>
<td>43.8</td>
<td>395</td>
<td>0.8938</td>
<td>282766</td>
<td>292656</td>
</tr>
<tr>
<td>10</td>
<td>0.001</td>
<td>7.85e-4</td>
<td>-7</td>
<td>48.7</td>
<td>395</td>
<td>0.8938</td>
<td>233574</td>
<td>235516</td>
</tr>
<tr>
<td>12.5</td>
<td>0.00125</td>
<td>9.54e-4</td>
<td>-8.8</td>
<td>53.2</td>
<td>395</td>
<td>0.8938</td>
<td>201805</td>
<td>235778</td>
</tr>
</tbody>
</table>

Table 8.6: Material Constant Comparison: calculated from force data vs. \( K_T \), \( P_1 \) and \( K_{RC} \), \( P_2 \) estimates.

Stock Material: 6061 Aluminum
Cutting Tool: 0.5 in. Diameter, 4 flute, Flat End Cutter, HSS
Modeled Runout: 0.00025" @ 12°
Spindle Speed: 2500 RPM
Conditions: Axial depth 0.25", Step-over 0.3", Coolant On

<table>
<thead>
<tr>
<th>Feed (ipm)</th>
<th>Feed/Tooth (in/tooth)</th>
<th>( h_{avg} ) (in)</th>
<th>( F_{X_{avg}} ) (Lb)</th>
<th>( F_{Y_{avg}} ) (Lb)</th>
<th>( K_R )</th>
<th>( P_2 )</th>
<th>( K_R ) Est.</th>
<th>( K_R ) % Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.0005</td>
<td>4.46e-4</td>
<td>-2.7</td>
<td>40</td>
<td>17045</td>
<td>0.3063</td>
<td>0.3077</td>
<td>0.3077</td>
</tr>
<tr>
<td>7.5</td>
<td>0.00075</td>
<td>6.15e-4</td>
<td>-5.5</td>
<td>43.8</td>
<td>17045</td>
<td>0.3063</td>
<td>0.2804</td>
<td>0.2804</td>
</tr>
<tr>
<td>10</td>
<td>0.001</td>
<td>7.85e-4</td>
<td>-7</td>
<td>48.7</td>
<td>17045</td>
<td>0.3063</td>
<td>0.2615</td>
<td>0.2615</td>
</tr>
<tr>
<td>12.5</td>
<td>0.00125</td>
<td>9.54e-4</td>
<td>-8.8</td>
<td>53.2</td>
<td>17045</td>
<td>0.3063</td>
<td>0.2462</td>
<td>0.2462</td>
</tr>
</tbody>
</table>

Table 8.7: Material Constant Comparison: calculated from force data vs. \( K_{TC} \), \( P_1 \) and \( K_{RC} \), \( P_2 \) estimates.

Stock Material: 6061-T6 Aluminum
Cutting Tool: 1.0 in. Diameter, 4 flute, Flat End Cutter, HSS
Modeled Runout: 0.00075" @ 12°
Spindle Speed: 1681 RPM
Conditions: Axial depth 0.25", Step-over 0.4", Coolant On
Figure 8.4: Graphical comparison of material constants for the case of the 0.375" ball end cutter of Table 8.1.
8.6 Ball-End Cutter Geometry Simulation

The variable radius on the ball portion of a ball end cutting tool results in variations in the entrance and exit angles in this region, even under constant cutting conditions. Cusps left by prior passes of the cutter further compound these variations. As may be seen in Figure 8.5, the radial depth distance remains constant in magnitude and location only on the cylindrical portion of the cutter, and the value of the radial depth in this region is equal to the stepover distance. On the ball portion of the cutter, the net radial depth amount remains constant and equal to the stepover distance only at axial locations that lie between the ball center and an axial location equal to the top of cusps. Even in this region, the individual positive (+N, where N is the toolpath Normal direction; see Chapter 4) and negative (-N) contributions continually vary, affecting the entrance/exit angle values. Furthermore, from this region downwards in the axial direction, the net radial distance diminishes to a minimum of zero at the bottom of the cutter ball (see Figure 8.5). These variations in radial depth have a profound effect on the entrance and exit angles.

Accurate entrance and exit angles are required for each axial disc during mechanistic constant calculation to determine which flute segments lie within the contact area at a given rotation angle, and thus make a force contribution, and also to calculate $h_{gap}$. During mechanistic constant calculation, an analytical calculation of the contact area is performed, rather than relying on the Z-buffer geometric model. This is done to minimize error in this critical calculation, which can be the result of both errors due to the approximate Z-buffer model, as well as user error in defining the geometric model. Equally important, this is also done to provide an independent verification of the overall model. That is, when the constants calculated are applied under cutting conditions identical to the test cuts, the Z-buffer model should produce a contact area description that results in force estimates acceptably close in value to the test cut forces. If errors are present in the geometric model, or contact area calculation routines, this will aid in discovering them. Therefore, an analytic relation that describes the geometric variations on the cutter ball is desired, so that the appropriate entrance and exit angle values may be found for mechanistic constant calculation.

When down-milling on the cutter ball, the radial depth component that extends in $-N$ continually increases in magnitude down the cutter axis, until the top of the cusp is reached. At this point the radial depth remains equal to the (decreasing) radius value. This may be seen in Figure 8.6, where radial distance $D_r$ increases down the axis of the cutter ball until the top of the cusp is reached, at which point it remains
Figure 8.5: Regardless of the stepover distance $D_s$, the net radial depth $D_r$ of cut remains equal to it until an axial depth equal to the top of the cusps. However, the individual positive and negative radial depth contributions vary. (The cutter is moving out of the page).

Figure 8.6: The stepover distance $D_s$ and the radius of the current axial disc may be used to calculate change in radial depth $D_r$ that occurs on the cutter ball.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
equal to the cutter radius as it diminishes to zero at the cutter bottom. This results in a continual increase in the entrance angle, from the value that exists at the axially highest point of stock engagement, to 180° at the top of the cusp height. Note that the radial depth component in +N remains equal to the current radius for all axial locations, and therefore the exit angle remains unchanged at 0° over the entire axial depth of cut. The reverse is true when up-milling, with the exit angle remaining fixed at 180° and the entrance angle gradually decreasing to 0° at the top of the cusps. Compensating for variations in radial depth on the cutter ball is fairly straightforward once the relationship between the disc radii and the stepover distance has been defined. This will now be presented for the case of down-milling. Reverse the roles of the positive and negative radial depth values for up-milling.

When down-milling, note that the negative radial depth, \(-D_r\), remains constant and equal to the radius of the current axial location for all axial locations on the ball that are engaged in the stock. Also note that the net radial depth remains constant and equal to the stepover distance, \(D_s\), at all points above the axial location of the top of the cusps. Below this axial location, both the positive and negative radial depths are equal to the radius, and the net radial depth decreases from \(D_s\) to zero. Therefore, the positive radial depth, \(+D_r\), may be calculated as:

\[ +D_r = D_s - R \]

(8.6)

where \(R\) is the radius value of the disc to which \(+D_r\) is to be applied (in).

This relation may be seen in Figure 8.6. If the value of \(+D_r\) calculated is greater than \(R\), as it will be for discs that lie below the top of the cusp, there is no problem. If a radial depth greater than \(R\) is passed the entrance and exit angle calculation routines, they set the entrance / exit angle to 180°/ 0° respectively.
SOFTWARE VALIDATION, DATA ACQUISITION, AND EXPERIMENTAL RESULTS

9.1 Chapter Introduction

This chapter describes the methods used to validate the automatic feedrate selection software system, and presents some of the results obtained. The primary components of the software system are the geometric model, the NC machine model, and the mechanistic model. The mechanistic model calculates milling force estimates as a function of material constants and cutting conditions, while the geometric and NC machine models provide continually updated cutting condition data. Accurate development of each of these component models, and proper integration of these models into the overall system, is necessary for the automatic feedrate selection system to operate correctly.

A large portion of the validation effort was spent analyzing the force predictions provided by the mechanistic model, as accurate force estimation is a basic requirement for automatic feedrate selection. Accurate force estimates also indicate that the supporting models are providing the proper data, and are properly integrated into the system (these models were also tested individually). The methods used to measure physical milling force data are also presented, along with a sampling of the data collected.

9.2 Supporting Software Validation

9.2.1. Supporting Software

In this implementation, the geometric and NC machine models support the mechanistic model, providing cutting condition data required for force estimation. As accurate output is required from the supporting models for accurate force estimation and feedrate selection, it is important to validate the accuracy of the supporting models prior to use in the integrated system. Additionally, each supporting model is independently of value. The geometric model is useful for NC toolpath verification, allowing for the 'virtual milling' of a part prior to cutting any metal, so that flaws in the NC part program may be
identified. The NC machine model is useful as it provides cutter-to-stock relative velocity control when 5-axis milling (and is currently implemented at an industrial site performing this task). This independent standalone operation further imposes the need for minimizing error in the supporting models, while also providing the means necessary for independent validation.

9.2.2 Geometric Model Validation

The geometric model is validated primarily through graphical imaging, including rendering of the stock geometry model, the cutter geometry, and the cut geometry parameters (i.e. the ‘contact patch’). During the initial implementation (and when bugs are identified in the software), situations are defined where the values of the intersection locations and contact patch bounds are known, and these are compared against the solution provided by the geometric model (generally using only small number of Z-buffer elements to simplify the task). However, once the basic accuracy of the implementation has been validated for a basic set of conditions, computer graphics are used to identify remaining theoretical or software 'bugs'. This approach is very useful, as the shear volume of numerical data that would have to be analyzed for validation would be overwhelming, even for a simple part file.

When validating the milling simulation, an animated rendering is used to ensure that the appropriate cutter positions and orientations are attained, and that the intersection calculations are performed correctly. Two methods are used during this validation. In one method, specific test cases involving a small set of specific tool moves and a small set of Z-buffer elements are used to validate the intersection calculations and storage in the Z-buffer model. In these cases, the Z-buffer model is rendered directly (see Figure 9.1). In the second method, entire 3 and 5-axis G-code part files that produce a known geometry are simulated, such as in Figures 4.11, 4.12, 4.15, 7.11, and 9.5. This is useful for finding unforeseeable errors that are not explicitly tested for, but arise during the essentially random selection of tool moves contained in a wide variety of part files. The graphical display of the geometric model is also valuable for validating the geometric approximations and error bounding required as a result of the ‘3-axis approximation’ of the swept toolpath envelope, as shown in Figures 5.7a,b,c. Note that the animated renderings also aid in NC machine model validation, as accurate geometric results require proper parsing of the G-code file as well as accurate coordinate transformation.
Figure 9.1: The Z-buffer model was graphically rendered for the validation of intersection calculation and storage routines using controlled cutting simulations. Testing was performed with relatively few Z-buffer elements, as the operations function identically for all elements and this simplified the task of isolating problems.
9.2.3 NC Machine Model Validation

The NC machine model is validated in several ways. The first is through graphical rendering. Animated graphical renderings can be shown in both part-relative and machine-relative points of view. In the part-relative point of view, only the cutter moves, relative to the milled stock, which remains stationary. In the machine-relative point of view the actual machine motion is simulated, and the stock and cutter move relative to each other as they would on the actual NC mill. In order for part geometry and the relative motion between the cutter and stock to appear properly, these animations require an accurate model of the machine kinematics for proper coordinate transformation, as well as proper parsing of the G-code file.

The combined kinematic and controller model that is used to calculate cutter relative velocities is validated through timing tests, and also through application. As the controller model sets axis velocities based on a desired time interval over which the move occurs, specific tool moves can be defined and simulated, and then timed on the mill for comparison. Then, using the known distance the axes are required to travel during the move, and the known time required to complete the move, the velocities of the axes during the move are validated. Additionally, the NC machine model has been commercially implemented for performing relative velocity control during 5-axis milling. Its success in this application demonstrates accuracy over a wide range of varying conditions. Commercial application also provided a medium to obtain feedback from users, so that subtle errors could be identified and fixed.

9.3 Force Data Acquisition Experimental Setup

To validate force estimates provided by the mechanistic model, it is necessary to measure empirical force data for comparison. Mounting the stock material to a Kistler Type 9257B piezoelectric load cell allowed this. This load cell has three independent degrees of freedom along which force data may be measured, and these were aligned with the $X_M$, $Y_M$, $Z_M$ coordinates, so they lie parallel to the CNC machines linear axes. The majority of the data was collected when milling on a Fadal Vertical Milling center, but data collection also occurred on a Boston Digital 1000 series CNC machine.

The force signals are routed from the load cell to a Kistler Type 5004 charge amplifier so that the small charge generated by the load cells piezoelectric crystals can be transformed into a voltage signal. The
charge amplifier generates an output voltage relative to the input signal from the load cell at the last 'reset' of the charge amplifier. For accurate force measurements it must only be reset under no-load conditions. The output signal decays exponentially following a reset, limiting the amount of time over which data acquisition may occur before another reset is required. The time constant on this decay is variable via a control knob on the charge amplifier, which was left in the 'Long' position. This was empirically found to allow for about 90 seconds of data acquisition before the measured signal decayed to 95% of its original value. The manufacturer provided time constant information, but it is variable and dependant on unspecified conditions (for the "Long" setting it stated "up to 100,000 sec").

From the charge amplifier, the force signals are routed to a Keithley-Metrabyte DAS 20 A/D data acquisition board mounted in a Pentium based PC. Labtec Notebook data acquisition software loaded on this PC managed the data acquisition process, allowing for the definition of sampling rates, sample times, data file definition, etc. Figure 9.2 shows a schematic of the entire data acquisition system.

9.4 Sampling Rate Selection

One drawback of the DAS 20 / Labtec Notebook combination is that it imposes a 1KHz limit on the sampling rate. This does not pose a severe restriction, as peak force data is typically desired, but it does make it difficult to obtain accurate plots of force versus cutter rotation angle for spindle speeds exceeding approximately 1500 RPM. At this speed there are only 40 samples per rotation, providing only 10 samples per tooth passing cycle for a 4-fluted cutter. Higher spindle speeds result in too sparse a data set to accurately reconstruct the variable forces that result from individual flute engagement during cutter rotation. This is demonstrated in Figure 9.3. In this figure, the individual tooth profiles are clearly visible in the data taken at a spindle speed of 832 RPM (three revolutions are shown). However, the data shown for the 6000 RPM spindle speed shows no clearly visible tooth profiles or distinctly repeated patterns over the course of three revolutions, as the measured data is too sparse. Aliasing is also a concern in this situation.

Average force values are of interest when calculating the mechanistic constants, while peak forces are of interest when predicting forces or setting feedrates. Therefore, some precautions must be taken when defining the sampling rate in order to maximize the possibility of sampling at a time when a peak force
Figure 9.2: Diagram of the system used for milling force data acquisition.

Figure 9.3: Comparison of the resolution of measured force data over 3 cutter revolutions for two spindle speeds, both are measured at the maximum sampling rate of 1000 Hz.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
occurs, and increase the range of data sampled to improve the average force calculation. This is achieved by selecting a sampling rate that does not result in continual sampling at the same rotation angle(s). The number of degrees of rotation between samples is calculated using:

$$A = (\Omega)(360 \text{ deg/rot}) \left(\frac{1}{60 \text{ min/sec}}\right) / S$$

(9.1)

where \(A\) is the number of degrees of cutter rotation between samples (deg), \(\Omega\) is the spindle speed (RPM), and \(S\) is the sampling rate (Hz).

From this, the number of samples per rotation, \(N_s\), is calculated:

$$N_s = \text{floor}(360 / A).$$

(9.2)

For example, if sampling forces at 1KHz with a spindle speed of 4000 RPM, data acquisition occurs every 24° of cutter rotation, and exactly 15 samples are taken during a full rotation. This results in no variation in the angles being sampled between rotations, i.e. every rotation a sample takes place at 0°, 24°, 48°, 72°... 336°, and this cycle is then repeated on the next rotation. As the cutting forces due to individual flute engagement are cyclic, theoretically the same force values will be obtained on every rotation in this situation (assuming constant cutting conditions). This repeated set of forces may not be indicative of the actual peak or average forces, and could provide misleading results (see Figure 9.4a).

Due to slight variances in spindle speed as a result of variable loads, the actual rotation angle at which sampling occurs may not remain identical between rotations. However it is possible to greatly reduce the amount of useful information contained in a sampled data set as a result of “improper” sampling rates. Therefore, note that the selection of a slightly different sampling rate results in a phase shift in the sampling times relative to the rotation angle, causing sampling to occur at different angles during each rotation of the cutter, as shown in Figure 9.4b. Over the course of several rotations, the sampling of different rotation angles provides a more complete set of force data. This allows for more accurate calculation of average force, and increases the probability that a sample will occur at or near a local peak force value. For
Figure 9.4a: Note if the sampling period shown above by the dashed line is selected, an inaccurate estimate of peak and average force will result over all rotations. This is because the same angles are sampled every rotation of the cutter.

Figure 9.4b: Although the sampling period indicated above is similar to that of 9.4a, it does not sample the same angles every cutter rotation. This phase shifting between rotations allows for the collection of a wider range of data.
example, a cutter rotating at 4000 RPM that is sampled at 987 Hz experiences a sample every 24.32°, which
imparts a phase shift of 4.74° per rotation. On the first rotation 0° is sampled, followed by samples every
24.32°. On the second rotation, this multiple causes the first sample to occur at 4.74°, and on the third
rotation the first sample is at 9.48°, etc. After 5 rotations, the phase shifting has come full cycle and data
has been sampled at an average interval of 4.74°. At 4000 rpm 5 rotations occur in 0.075 sec. At a relatively
high feedrate of 20 IPM (or a feed per tooth of 0.00125 in/tooth), the cutter would cover a distance of
0.025” during this period, and so the cutting conditions would remain fairly constant. This sampling rate
therefore allows for an effective 4.74° sampling period. The phase shift amount \( \phi \) may be solved for as:

\[
\phi = (N_2 + 1) \lambda - 360°.
\] (9.3)

There is no single equation for the definition of a desirable sampling frequency, as it is a function
of the general magnitude of the sample time as well as spindle speed. If measuring two force values for 15
seconds, a sampling rate in the range of 1000 Hz may be acceptable. However, if measuring three forces
and three axis velocities over 90 seconds, a sampling rate in the range of 500 Hz may be preferred to reduce
the total amount of data collected. In our setup, very large amounts of data are difficult to analyze (MS
Excell has a 64,000 line limit) and they reduce the robustness of the data acquisition system (the systems
crashes with a ‘cache overflow’ when a large amount of data is collected too rapidly). Once the spindle
speed and the general magnitude of the sample time is known for a given set of test cuts, Equations (9.1-
9.3) may be used to ‘fine tune’ the sampling rate to ensure some phase shifting occurs in the sampled data.
A dedicated Excell spreadsheet program was used in this research for this ‘sample rate tuning’ task.

9.5 Mechanistic and Integrated Model Validation
9.5.1 Discrete Cutter Model Validation

Graphical rendering is also helpful in validating the geometry-based portions of the mechanistic
model. This includes three different mechanistic parameters; contact area, chip thickness, and discrete
cutter geometry. Contact area, and chip thickness are rendered simultaneously, as shown in Figure 2.6. The
flute geometry of the discrete cutter model may also be included in this display, as shown in Figures 3.5 and 9.6. Chip thickness is indicated by a color map distributed over the contact area, with red indicating areas of maximum chip thickness, and then fading through orange, yellow, green, and blue as the chip thickness values thin to zero. This color map indicates the relative chip thickness that would exist when a cutting flute passes through that region of the contact area. Note that Figure 3.5 shows a tool move in which no stock intersection occurred, and so no contact area or chip thickness information is displayed.

In addition to static renderings, animation also plays a role in mechanistic model validation. Rendering of the geometry-based portions of the mechanistic model can be performed while stepping through a milling simulation, one tool move at a time. When this is done, the cutter motion and material removal may be compared to the contact area produced by toggling between a 'stock view' and a 'discrete cutter view' mode, as demonstrated in Figure 9.5 (note that in actual operation the discrete cutter model and the stock geometry would be viewed independently). Animation can also occur local to the discrete cutter rendering. In addition to being able to toggle the modeled flute geometry on/off, the rendered flutes can also be animated to simulate cutter rotation, using the discrete cutter rotation angle positions stored on each axial disc. This is done to validate the accuracy of the cutter rotation simulation.

9.5.2 Mechanistic Model Validation

Once the supporting models have been validated, the mechanistic model is tested. This is performed through comparison of measured to estimated force data for a given set of cutting conditions. Note this comparison also continues validation of the other components of the software system. The mechanistic model relies on the geometric and NC machine model for input data, and so force estimate errors can be the result of problems in any of these. This is why independent validation of the supporting models is first required. However, it is not possible to identify all errors in the supporting models, as some problems only occur under unusual circumstances, or are introduced during model integration. Therefore, forces are calculated for relatively simple cutting conditions, such as constant feed, straight-line slot and partial immersion cuts, during the initial mechanistic model validation. It is easy to manually verify that accurate cutting condition information is being passed to the mechanistic model for these cuts, and so any errors that exist must be local to the mechanistic model (assuming 'good' material constants are used).
CORRECT SIMULATION METHOD; PASS THROUGH STOCK
Note that this leaves the appropriate contact area, as shown in the inset.

INCORRECT SIMULATION METHOD; TERMINATE CUT IN STOCK
Note that this leaves an incorrect contact area, as shown in the inset.
This is the result of stock intersection at the end of the cut.

Figure 9.5: Simulation of the test cuts performed for the calculation of the material constants. Correct simulation is for the cutter to pass all the way through the stock model.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
When calculating material constants, milling force data is collected for a short set of cuts that have a known, simple cut geometry and minimized cutter runout. This provides an excellent source of force data for initial validation of the mechanistic model. While the primary purpose of these test cuts is to obtain average $X_C T$ and $Y_C T$ force values for mechanistic constant calculation, the full range of force values present during a given cut is measured for the calculation of these average forces. Recall that milling force vectors are defined in Cutting Tool Coordinates, in which $X_C T$ lies in the direction of cutter travel, and $Y_C T$ is defined as the cross product of the cutter rotational axis with $X_C T$.

In the context of this research, the force value of interest is the peak $X_C T,Y_C T$ magnitude that occurs during a single tool move, as described in Chapter 3. Note that each test cut has constant cutting conditions, and is therefore representative of a single tool move. Therefore, the peak $X_C T,Y_C T$ force magnitude is calculated for each test cut performed during material constant calculation, for comparison against the estimated value for each cut. The estimates are obtained by defining an appropriate stock model and G-code file that reproduce the cutting conditions present during data acquisition. For efficiency, the G-code file is defined to contain the cutter motions, feedrates, and spindle speed value used during all the test cuts in a complete data set (defined in Chapter 8), and the stock model is defined to be large enough to contain all required test cuts. The image at the top of Figure 9.5 shows graphically the results of such a simulation performed in the correct manner. Also shown in this image is the contact area; note that it is easy to validate that the contact area is correct (the shape of the contact area during the test cuts is presented in Chapter 8).

When simulating the test cuts, it is important that the cutter pass through the entire width of the stock model in a single move. Failure to do so results in an incorrect estimation of the cutting conditions, as a result of the conservative modeling approach that was implemented. When the cutter stops inside of the stock material, the size of the contact area increases at the end of the toolpath envelope as the cutter contacts additional material in this region, as shown in the bottom of Figure 9.5. While this increased contact area physically exists, it occurs only briefly at the very end of the entire cut. However, this worst-case condition is applied to the entire cut during simulation to ensure conservative results.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
The force estimates obtained from the simulations are then compared to the empirical force data. The error in the force estimates is defined as the percent difference between the estimated force value and the empirical force value relative to the empirical force value, or:

\[
\%\text{Err} = \left( \frac{F_{\text{Estimated}} - F_{\text{Measured}}}{F_{\text{Measured}}} \right) \times 100.
\]  

(9.4)

Error of up to 10% is generally considered acceptable under these constant cutting conditions [KDL82].

Error in the force estimates is not always due to implementation error; it can also result from an incorrect runout definition. This can easily be corrected. If all the estimates for a set of test cuts differ uniformly relative to the empirical data (high or low), this can indicate that an improper runout description was used. When this occurs, the estimates are shifted into the range of the empirical data through adjustment of the runout definition (refer to Chapter 2 or [KD83] for a further description of runout). Increasing the amount of runout increases peak force estimates; the converse is true for reduced runout. The runout locating angle is also used to 'fine tune' the estimates, although it has less of an overall effect than the runout amount. On a 4 fluted cutter model, the flutes originate at 0°, 90°, 180°, and 270°, and so a locating angle defined near one of these values provides a small increase in peak force, as it causes one flute to experience larger chip thickness values than the others (generally the majority of the material removal occurs near the cutter bottom). Defining a locating angle near 45°, 135°, 225°, or 315° tends to reduce the effects of runout, distributing it over two flutes, slightly decreasing the force estimates. Adjustment of the locating angle position has enhanced effects for ball end cutters, as the shallow helix angle on the cutter ball causes a greater length of flute to lie in the region of a given angular position.

If runout adjustment can't bring the all estimates to within 10% of the empirical data, it is assumed that an error exists in the model, or that the material constants are incorrect. Once the basic definition and implementation of the mechanistic model has been validated, the constants are the primary source of error. If acceptable force matching cannot be achieved for a set of test cuts, the constants calculated from those cuts are discarded. This generally occurs when the test cuts are performed with excessive cutter runout, a dull or damaged cutter, or in the presence of 'chatter'. Avoidance of these conditions generally allows acceptable material constants, which provide accurate force estimates for the test cuts, to be calculated.
Table 9.1 presents a comparison of estimated versus measured force data for the test cuts performed to calculate constants for a 0.375" HSS ball end cutter milling 6061 Aluminum. Figure 9.6 presents plots of the empirical force data for this case. The value of interest in these plots is the peak force magnitude. The test cuts are performed under constant cutting conditions, and so the value of the peak magnitude is fairly constant over time within each plot. However, notice through a comparison of the plots that the peak value becomes less uniform over time as force magnitude increases. This is most likely due to dynamic effects and cutter deflections, which increase with milling force. In this situation, the ‘peak force’ is defined to be the largest value attained, with the exception of outliers (i.e. random force spikes that account for less than 0.05% of all samples). Tables 9.2 through 9.4 present similar comparisons of estimated vs. measured peak force magnitude for a range of stock materials and cutter types.

While matching peak magnitude values is an important part of model validation, it provides little information regarding the overall physical accuracy of the mechanistic model. While the material constants are empirical, the mechanistic model itself is based on the physical mechanics of the metal cutting process, estimating $X_{CT}$ and $Y_{CT}$ force values at some discrete cutter rotation angle using orthogonal cutting theory. Therefore, if properly defined and implemented, the model should be capable of reproducing the variations in force that occur over the course of a cutter rotation. Plots of milling force vs. rotation angle are referred to as ‘flute force profiles’, or simply ‘flute profiles’, in this research.

Some flute force profiles are presented in Figures 9.7-9.10, along with a description of the cutting conditions and simulation values used to achieve these results. While the profiles do not provide an exact match between the estimated and measured data, the magnitudes and trends of both the $X_{CT}$ and $Y_{CT}$ force components are very similar. It is difficult to get an exact match due to the finite accuracy inherent during modeling as a result of the use of approximate flute geometry and a discrete model. Additionally, physical effects such as cutter deflection, machine dynamics, variable frictional forces, data sampling rate, and the forward motion of the cutter during rotation all contribute to the force variations. Note that even within a single measured data plot, the force profiles are not identical over the three cutter rotations shown. It is also very tedious to obtain the flute force profiles, as it requires fine adjustment of the runout amount and locating angle, which are varied individually but are dependant on each other.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
### 0.375" HSS Ball End Mill, 775 RPM, Runout 0.0007" @ 12.5°, Cutting 6061 Al

*(See Table 8.1 for the analysis of the material constants)*

<table>
<thead>
<tr>
<th>CYLINDER PORTION MATERIAL CONSTANTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{TC}$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>19822</td>
<td>0.3185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUTTER BALL PORTION MATERIAL CONSTANTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{TC}$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>18134</td>
<td>0.3577</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radial Depth (in)</th>
<th>Axial Depth (in)</th>
<th>Feed (IPM)</th>
<th>Estimated Peak Mag (Lb)</th>
<th>Measured Peak Mag (Lb)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2625</td>
<td>0.1875</td>
<td>0.8</td>
<td>30.848</td>
<td>30.98</td>
<td>-0.426</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.1875</td>
<td>1.6</td>
<td>44.685</td>
<td>44.48</td>
<td>0.461</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.1875</td>
<td>2.3</td>
<td>53.172</td>
<td>55.82</td>
<td>-4.744</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.375</td>
<td>3.1</td>
<td>61.816</td>
<td>65.8</td>
<td>-6.055</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.375</td>
<td>0.8</td>
<td>51.329</td>
<td>47.53</td>
<td>7.993</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.375</td>
<td>1.6</td>
<td>72.333</td>
<td>69.84</td>
<td>3.570</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.375</td>
<td>2.3</td>
<td>86.552</td>
<td>86.25</td>
<td>0.350</td>
</tr>
<tr>
<td>0.2625</td>
<td>0.375</td>
<td>3.1</td>
<td>101.441</td>
<td>108.13</td>
<td>-6.186</td>
</tr>
</tbody>
</table>

Table 9.1: Validation of the mechanistic model for a 0.375" ball end mill cutting 6061 aluminum. In this research, the peak XY force magnitude is currently the value of interest when estimating milling forces. The measured XY force magnitude data is provided in Figure 9.5.

### 0.1875" Carbide Ball End Mill, 2325 RPM, Runout 0.000125" @ 12.5°, Cutting 6061 Al

*(See Table 8.3 for the analysis of the material constants)*

<table>
<thead>
<tr>
<th>CYLINDER PORTION MATERIAL CONSTANTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{TC}$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>10620</td>
<td>0.4108</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CUTTER BALL PORTION MATERIAL CONSTANTS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{TC}$</td>
<td>$P_1$</td>
</tr>
<tr>
<td>95085</td>
<td>0.1317</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Radial Depth (in)</th>
<th>Axial Depth (in)</th>
<th>Feed (IPM)</th>
<th>Estimated Peak Mag (Lb)</th>
<th>Measured Peak Mag (Lb)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1312</td>
<td>0.0938</td>
<td>2.4</td>
<td>14.651</td>
<td>14.6</td>
<td>0.35</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.0938</td>
<td>4.8</td>
<td>21.372</td>
<td>19.8</td>
<td>7.94</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.0938</td>
<td>6.9</td>
<td>26.549</td>
<td>25.4</td>
<td>4.52</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.0938</td>
<td>9.3</td>
<td>32.126</td>
<td>33.1</td>
<td>-2.94</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.1875</td>
<td>2.4</td>
<td>26.775</td>
<td>25.4</td>
<td>5.41</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.1875</td>
<td>4.8</td>
<td>37.272</td>
<td>34.3</td>
<td>8.66</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.1875</td>
<td>6.9</td>
<td>44.575</td>
<td>43.4</td>
<td>2.71</td>
</tr>
<tr>
<td>0.1312</td>
<td>0.1875</td>
<td>9.3</td>
<td>51.723</td>
<td>53.3</td>
<td>-2.96</td>
</tr>
</tbody>
</table>

Table 9.2: Validation of the mechanistic model for a 0.1875" carbide ball end mill cutting 6061 Al. The peak XY force magnitude is shown above.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 9.6: Samples of the measured XY force magnitude values presented in Table 9.1. A 2.5-second sampling of data taken at 1000 Hz is shown. Note that the scales differ between plots to maximize visual clarity.
### 0.5" Carbide Ball End Mill, 2000 RPM, Runout 0.000125" @ 2.0°, Cutting 15-5 Stainless Steel

(See Table 8.5 for the analysis of the material constants)

<table>
<thead>
<tr>
<th>Radial Depth (in)</th>
<th>Axial Depth (in)</th>
<th>Feed (IPM)</th>
<th>Estimated Peak Mag (Lb)</th>
<th>Measured Peak Mag (Lb)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.35</td>
<td>0.25</td>
<td>6</td>
<td>246.6</td>
<td>238</td>
<td>3.59</td>
</tr>
<tr>
<td>0.35</td>
<td>0.25</td>
<td>8</td>
<td>266.0</td>
<td>265</td>
<td>0.39</td>
</tr>
<tr>
<td>0.35</td>
<td>0.25</td>
<td>10</td>
<td>283.4</td>
<td>285</td>
<td>-0.55</td>
</tr>
<tr>
<td>0.35</td>
<td>0.25</td>
<td>12</td>
<td>299.4</td>
<td>315</td>
<td>-4.96</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5</td>
<td>6</td>
<td>498.0</td>
<td>460</td>
<td>8.25</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5</td>
<td>8</td>
<td>556.5</td>
<td>520</td>
<td>7.03</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5</td>
<td>10</td>
<td>609.4</td>
<td>580</td>
<td>5.07</td>
</tr>
<tr>
<td>0.35</td>
<td>0.5</td>
<td>12</td>
<td>658.2</td>
<td>635</td>
<td>3.65</td>
</tr>
</tbody>
</table>

Table 9.3: Validation of the mechanistic model for a 0.5" carbide ball end mill cutting 15-5 Stainless Steel. The peak XY force magnitude is shown above.

### 1.0" HSS Flat End Mill, 1681 RPM, Runout 0.0025" @ 95°, Cutting 6061-T6 Aluminum

(See Table 8.7 for the analysis of the material constants)

<table>
<thead>
<tr>
<th>Radial Depth (in)</th>
<th>Axial Depth (in)</th>
<th>Feed (IPM)</th>
<th>Estimated Peak Mag (Lb)</th>
<th>Measured Peak Mag (Lb)</th>
<th>% Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.25</td>
<td>25</td>
<td>151.935</td>
<td>149.2</td>
<td>1.83</td>
</tr>
<tr>
<td>0.4</td>
<td>0.25</td>
<td>35</td>
<td>172.465</td>
<td>171</td>
<td>0.86</td>
</tr>
<tr>
<td>0.4</td>
<td>0.25</td>
<td>45</td>
<td>192.005</td>
<td>190.8</td>
<td>0.63</td>
</tr>
<tr>
<td>0.4</td>
<td>0.25</td>
<td>55</td>
<td>210.715</td>
<td>221.4</td>
<td>-4.83</td>
</tr>
</tbody>
</table>

Table 9.4: Validation of the mechanistic model for a 1.0" HSS flat end mill cutting 6061 T6 aluminum. The peak XY force magnitude is shown above.
Figure 9.7: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.1875" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 9.8: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter.
Figure 9.9: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter.
Figure 9.10: Comparison of predicted and measured flute force profiles for a partial immersion slot cut of 0.09375" axial depth, 0.1125" radial depth for a 0.1875" ball end cutter.
The estimated flute force profiles are generated through calculation of $X_{CT}$ and $Y_{CT}$ force estimates at $1^{\circ}$ intervals over a $360^{\circ}$ range. These estimates are then duplicated twice to arrive at the data for the three rotations shown, as calculation beyond $360^{\circ}$ would simply produce identical results. The empirical force data is sampled as a function of time, but plotted as a function of rotation angle. The rotation angle is approximated from the known sample time using the relation:

$$\Theta(t) = (\Omega)(360 \text{ degree/revolution})(0.01666 \text{ sec/min}) (t - t_0)$$

(9.5)

where $\Theta$ is the cutter rotation angle (deg.),

$\Omega$ is the spindle speed (RPM),

$t$ is the current sample time (sec), and

$t_0$ is the sample time at which the first sample in the flute force profile occurs (sec).

9.5.3 Integrated Force Estimation Model Validation

While static cutting conditions and flute force profiles are useful for validation of the mechanistic model and the basic implementation of the integrated model, it only does so for a limited set of conditions. To fully validate the integrated model, force estimates must be compared to empirical data measured during sculptured surface machining, when the conditions are varying continually. For testing purposes, this is achieved using sinusoidal cutter paths. These cutter paths are defined in machine coordinates such that the net cutter motion occurs in $X_M$, with motion in $Z_M$ and/or $Y_M$ occurring as a sine function to produce continually varying cutting conditions. The use of sinusoidal paths is selected as they may be easily repeated. Using this approach, cutter motion in $Z_M$ is defined using:

$$Z_M = Z_{\text{offset}} + A_Z \sin(2\pi X_M / Fr_Z)$$

(9.6)

Where $Z_M$ is the Machine coordinate Z position placed in the G-code file,

$Z_{\text{offset}}$ is an offset that shifts the location of the sinusoid,

$A_Z$ is the amplitude of the sinusoid in Z,

$X_M$ is the machine coordinate X position for which the Z location is desired, and

$Fr_Z$ is the frequency of the sinusoid in Z.
An identical equation is used to define cutter motion in $Y_M$:

$$Y_M = Y_{\text{offset}} + A_Y \sin\left(2\pi \frac{X_M}{F_{r_Y}}\right)$$

(9.7)

Different values for offset, amplitude, and frequency may be defined in $Y_M$ and $Z_M$. The location $X_M = 0$ is defined as the edge of the stock where the cutter enters it in the $X_M$ direction. The location $Z_M = 0$ is defined as the cutter location where the cutting tool contacts the surface of the stock; note for a ball end cutter this is a distance of one radius from the stock surface. Figure 9.11 graphically shows the definition of a sinusoidal slot cut in $Z_M$, the process is similar in $Y_M$.

Figures 9.12 - 9.15 provide a comparison of measured vs. estimated $X_MY_M$ force magnitudes for some sinusoidal cuts, along with a description of the cutting conditions and model parameters used. Figure 9.12 presents data for a 0.5" HSS ball end cutter with a sinusoid only in the $Z_M$ direction. Note both the measured and estimated force data are also sinusoidal, tracking the depth of cut. Figures 9.13, 9.14, and 9.15 present force data for a 0.1875" carbide ball end cutter, with 9.13 and 9.14 showing slot cut data and 9.15 showing data for a partial immersion cut. The partial immersion cut was performed by shifting the cutting tool -0.1125" in $Y_M$, or 0.6 of its diameter, from the sinusoidal slot cut. Again, the estimated force data tracks the empirical data in all three figures. Note that a subtle trend in the empirical data due to $Y_M$ motion is tracked by the estimates. In Figures 9.14 and 9.15, the central of the three main force peaks experiences a sharper transition from increasing to decreasing force (i.e. it is “more pointy”) relative to the force peaks on either side. This is most noticeable in Figure 9.15, but may also be seen in Figure 9.14. Also note in Figure 9.15 that the estimates track the central peak as being the greatest in magnitude.

9.6 Automatic Feedrate Selection Validation

Recall that feedrates are selected primarily to maintain a desired force threshold during automatic feedrate selection, although other constraints include the definition of a maximum allowable feedrate, and/or a maximum allowable chip thickness. Feedrate selection validation is therefore accomplished through comparison of simulated to measured force values, both performed using the feeds calculated by
Figure 9.11: Definition of the sinusoidal test cut shown in Z, a similar definition may also occur in Y. The figure on the bottom depicts a cross section of a test cut in progress.
Figure 9.12: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 6 IPM and a spindle speed of 3500 RPM.
Figure 9.13: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 15 IPM and a spindle speed of 7000 RPM.
Figure 9.14: Comparison of predicted and measured forces for a sinusoidal slot cut at a constant feedrate of 18 IPM and a spindle speed of 4500 RPM.
Sinusoidal Partial Immersion Cut Force Data

**CUTTING CONDITIONS - FORCE ESTIMATION**

<table>
<thead>
<tr>
<th>Cutting Tool</th>
<th>Spindle Speed (RPM)</th>
<th>Feedrate (IPM)</th>
<th>Stock Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1875&quot; Carbide Ball</td>
<td>4500</td>
<td>18</td>
<td>6061 Aluminum</td>
</tr>
</tbody>
</table>

**SINUSOIDAL CUT DESCRIPTION**

- X: Primary Cut Direction, 5.75" Cut Length in X Partial Immersion 0.1125" Stepover, Downmill
- Y: Amplitude 0.2815 (3R) Period: 3" Y_{offset} = -0.1125" (0.6D) stepover per cutter pass
- Z: Amplitude 0.0703 (0.75R) Period: 2.25" Z_{offset} = -0.1172" (1.25R)

**SIMULATION CONDITIONS; Cylinder Portion**

<table>
<thead>
<tr>
<th>K_{TC}</th>
<th>P_1</th>
<th>K_{RC}</th>
<th>P_2</th>
<th>Disc Definition</th>
<th>Helix Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>10620</td>
<td>0.4108</td>
<td>0.0742</td>
<td>0.2280</td>
<td>0.01&quot;</td>
<td>30°</td>
</tr>
</tbody>
</table>

**SIMULATION CONDITIONS; Ball Portion**

<table>
<thead>
<tr>
<th>K_{TC}</th>
<th>P_1</th>
<th>K_{RC}</th>
<th>P_2</th>
<th>HelixEnd</th>
<th>MinHelix</th>
<th>FluteEnd</th>
<th>Disc Defn</th>
</tr>
</thead>
<tbody>
<tr>
<td>95085</td>
<td>0.1317</td>
<td>0.0355</td>
<td>0.3346</td>
<td>70°</td>
<td>0°</td>
<td>70°</td>
<td>2°</td>
</tr>
</tbody>
</table>

**SIMULATION CONDITIONS; Common**

<table>
<thead>
<tr>
<th>Runout</th>
<th>Runout Locating Angle</th>
<th>Z-buffer Spacing</th>
<th>Total Number of Z-buffer Elements</th>
<th>Total Number of Tool Moves</th>
<th>Simulation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00012&quot;</td>
<td>105.5°</td>
<td>0.005&quot;</td>
<td>349461</td>
<td>276</td>
<td>Pentium 300 MHz</td>
</tr>
</tbody>
</table>

Figure 9.15: Comparison of predicted and measured forces for a sinusoidal partial immersion cut at a constant feedrate of 18 IPM and a spindle speed of 4500 RPM.
the software (as opposed to simply comparing measured forces to the desired threshold value). This section contains some examples of such comparisons. During these tests, an arbitrarily chosen force tolerance of 0.5 Lb. and an arbitrary maximum allowable feed value of 40 IPM were used (both values are user-definable in the software, although defaults are provided). The desired force threshold was varied between the tests, and is provided with each of the individual data sets. As these tests do not include extended light cuts (this is not the focus of this research), maximum chip thickness values were not defined.

Figure 9.16 presents a direct comparison of force data for a sinusoidal slot cut made with a 0.5" ball end cutter, similar in geometry to the cut depicted in Figure 9.12. The cut parameters are provided at the top of the figure. The top-most two plots show on the left the measured forces for this cut made at a constant 6 IPM, and on the right with the corresponding depth of cut. Not surprisingly the forces experienced track the depth of cut. Using traditional methods, the feed value must be manually selected to provide for safe milling conditions for the worst-case scenario, namely the maximum depth of cut. Directly below the 6 IPM constant feed force plot is a plot of the measured forces for the exact same cut made using optimized feeds, automatically generated by the integrated model. The scales on both plots are identical, indicating that the use of optimized feeds reduced the peak force experienced by 16.9% while reducing the milling time by 25.9%. The bottom-most plot in the figure presents the optimized feed values used, along with the depth of cut, vs. time. Note the sinusoidal path of the cutter is skewed in this image due to the use of the optimized feedrates. Half of the total distance is traversed during the first 10 seconds, as opposed to 40 seconds for the constant feedrate. That is because the depth of cut drops from 0.25" (1 radius) to zero and back to 0.25" over the first half, allowing for the use of higher feeds; in fact the feed saturates at the maximum allowable value (40 IPM). The majority of the time savings occur in this region. However, the second half of the cut, where the maximum cut depth of 0.5" occurs, takes place at a feed less than 6 IPM, causing this portion of the cut to require more time than for the constant feed case (approximately 50 seconds versus 40), but it produces a lower peak force, and the net time savings are still appreciable. It should be noted that while the feeds selected in this case maintained a peak $X_{CT}, Y_{CT}$ magnitude of 100 Lb, the desired force threshold for this cut was 75 Lb. This data was collected prior to the inclusion of runout modeling, causing the simulation to assume ideal conditions and over-estimate the required feeds.
COMPARISON OF OPTIMIZED FEEDS TO CONSTANT FEEDS
For a Sinusoidal Slot Cut, Milled in 6061 Al at 2500 RPM with a 0.5" Ball End Cutter

SINUSOIDAL CUT DESCRIPTION
X: Primary Cut Direction, 8.25" Cut Length in X Full Slot Cut
Y: Amplitude 0 (no Y motion)
Z: Amplitude 0.25 (1 Radius) Period: 8"

Optimized Feed rate (Limited to 40 IPM)

<table>
<thead>
<tr>
<th>Feedrate Mode</th>
<th>Feed Value (IPM)</th>
<th>Peak Force (Lb)</th>
<th>Milling Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant</td>
<td>6.0</td>
<td>134.1</td>
<td>85</td>
</tr>
<tr>
<td>Optimized</td>
<td>See Chart</td>
<td>111.4</td>
<td>63</td>
</tr>
</tbody>
</table>

Percent Reductions -16.9% -25.9%

Figure 9.16: Experimental forces for optimized feedrates versus a constant feedrate. The desired force was 75 Lb., however these results were obtained prior to the inclusion of runout.
Many tests were performed after the inclusion of cutter runout, and these demonstrate a better ability to maintain the desired force magnitude. Figures 9.17 - 9.20 present some examples of this data. Figure 9.17 presents the data and feed values required to maintain a force threshold of 100 Lb for a 0.5" ball end cutter performing a sinusoidal slot cut with oscillations in Z only. Compare the data in this figure to that of Figure 9.12, which contains data for the same cut at a constant 6 IPM. Note in both Figures 9.17 and 9.16 that the peak forces occur near the start and end of the cuts, this is most likely the result of transient conditions that exist during stock (dis)engagement. While the feed is inhibited at stock entry to reduce transient effects, this does not provide an exact solution to the problem. The feed inhibition method was implemented at the request of machinists at our industrial research partner's site, and while the stock engagement is smoother as a result, it is not as a result of explicit transient force control.

The effects of transients and cutting dynamics are more apparent in Figures 9.18-9.20. In these cuts sinusoidal oscillations are defined in both YCT and ZCT, which leads to a more continual 'transient state' as a result of the continually varying conditions. Note that while the force data in these figures remains in the vicinity of the desired magnitude, it does not track as closely as the simpler, 2D cuts presented in Figures 9.16 and 9.17. In addition to the axial depth variations, this is also possibly the result of the varying location of the contact area on the surface of the cutter. Another contributor may be lack of inclusion of the machine acceleration/deceleration, which occurs in three degrees of freedom in these cuts. However, these plots still demonstrate an ability to track a desired force level as compared to a constant input feed. This may be seen through a comparison of Figure 9.18 with 9.13, and also through comparison of Figure 9.19 with 9.15, as both pairs of Figures represent common cutting conditions.

A comparison of the results for optimized vs. constant feedrate is presented in Figures 9.16-9.19. Note the average time milling time reduction is on the order of 20%. The actual amount of milling time 'saved' is dependant on part geometry, the desired force level, the maximum allowable feed, and other parameters that effect the output feed selected by the model (e.g. machine constraints). Manually defined feeds are set for safe operation during worst-case conditions. Therefore parts with a fairly constant cut geometry will see less time reduction than those that experience greater variation, as milling time is reduced using optimized feeds by increasing the feeds when more favorable conditions exist.
ADAPTIVE FEEDS GENERATED USING THE CONDITIONS IN FIGURE 9.12
For a Sinusoidal slot cut with Z variations milled with a 0.5" Ball End Cutter

<table>
<thead>
<tr>
<th>Desired Maximum XY Mag.</th>
<th>Maximum Allowable Feed</th>
<th>Simulation Time, Pentium 300 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>100.0 Lb.</td>
<td>40.0 IPM</td>
<td>23 sec</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Mill Time (sec)</th>
<th>Peak XY Mag. (Lb.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constant 6 IPM</td>
<td>88.16</td>
<td>141.0</td>
</tr>
<tr>
<td>Variable Feed</td>
<td>77.22</td>
<td>110.6</td>
</tr>
<tr>
<td>Percent Difference Relative to Constant Feed</td>
<td>-8.35%</td>
<td>-21.56%</td>
</tr>
</tbody>
</table>

Figure 9.17: Feedrate-controlled experimental force data for the sinusoidal slot cut of Figure 9.12. Data for the same cut at a constant 6 IPM is presented in Figure 9.12.
Feedrate Controlled to 35 Lb. Maximum

ADAPTIVE FEEDS GENERATED USING THE CONDITIONS IN FIGURE 9.13
For a Sinusoidal slot cut with Y and Z variations milled with a 0.1875" Ball End Cutter

<table>
<thead>
<tr>
<th>Desired Maximum XY Mag.</th>
<th>Maximum Allowable Feed</th>
<th>Simulation Time, Pentium 300 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>35.0 Lb.</td>
<td>40.0 IPM</td>
<td>31 sec</td>
</tr>
<tr>
<td></td>
<td>Mill Time (sec)</td>
<td>Peak XY Mag. (Lb.)</td>
</tr>
<tr>
<td>Constant 15 IPM (Figure 9.13)</td>
<td>39.9</td>
<td>40.6</td>
</tr>
<tr>
<td>Variable Feed (This Figure)</td>
<td>30.4</td>
<td>38.5</td>
</tr>
<tr>
<td>Percent Difference Relative to Constant Feed</td>
<td>-23.8%</td>
<td>5.2%</td>
</tr>
</tbody>
</table>

Figure 9.18: Feedrate-controlled experimental force data for the sinusoidal slot cut of Figure 9.13. Data for the same cut at a constant 15 IPM is presented in Figure 9.13.
ADAPTIVE FEEDS GENERATED USING THE CONDITIONS IN FIGURE 9.15
For a Sinusoidal Partial Immersion cut with Y and Z variations milled with a 0.1875" Ball End Cutter

<table>
<thead>
<tr>
<th>Desired Maximum XY Mag.</th>
<th>Maximum Allowable Feed</th>
<th>Simulation Time, Pentium 300 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.0 Lb.</td>
<td>40.0 IPM</td>
<td>21 sec</td>
</tr>
<tr>
<td>Constant 18 IPM (Figure 9.15)</td>
<td>Mill Time (sec)</td>
<td>Peak XY Mag. (Lb.)</td>
</tr>
<tr>
<td>Variable Feed (This Figure)</td>
<td>22.7</td>
<td>50.5</td>
</tr>
<tr>
<td>Percent Difference Relative to Constant Feed</td>
<td>18.0</td>
<td>46.3</td>
</tr>
<tr>
<td></td>
<td>-20.7%</td>
<td>-8.3%</td>
</tr>
</tbody>
</table>

Figure 9.19: Feedrate-controlled experimental force data for the sinusoidal partial immersion cut of Figure 9.15. Data for the same cut at a constant 18 IPM is presented in Figure 9.15.
CUTTING CONDITIONS - AUTOMATIC ADAPTIVE FEEDRATE SELECTION

<table>
<thead>
<tr>
<th>Cutting Tool</th>
<th>Spindle Speed</th>
<th>Max Feedrate</th>
<th>Desired Force</th>
<th>Stock Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.375&quot; HSS Ball</td>
<td>4500 RPM</td>
<td>40 IPM</td>
<td>100 Lb.</td>
<td>6061 Aluminum</td>
</tr>
</tbody>
</table>

SINUSOIDAL CUT DESCRIPTION

X: Primary Cut Direction, 8.25" Cut Length in X Partial Immersion 0.225" Stepover, Downmill

Y: Amplitude 0.375 (2R) Period: 3" $Y_{offset} = -0.225" (0.6D)$ stepover

Z: Amplitude 0.28125 (1.5R) Period: 4" $Z_{offset} = -0.1875" (R)$

SIMULATION CONDITIONS; Cylinder Portion

<table>
<thead>
<tr>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_{RC}$</th>
<th>$P_2$</th>
<th>Disc Definition</th>
<th>Helix Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>19822</td>
<td>0.3185</td>
<td>0.055</td>
<td>0.2795</td>
<td>0.01&quot;</td>
<td>30°</td>
</tr>
</tbody>
</table>

SIMULATION CONDITIONS; Ball Portion

<table>
<thead>
<tr>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_{RC}$</th>
<th>$P_2$</th>
<th>HelixEnd</th>
<th>MinHelix</th>
<th>FluteEnd</th>
<th>Disc Defn</th>
</tr>
</thead>
<tbody>
<tr>
<td>18134</td>
<td>0.3577</td>
<td>0.1501</td>
<td>0.1731</td>
<td>70°</td>
<td>0°</td>
<td>70°</td>
<td>2°</td>
</tr>
</tbody>
</table>

SIMULATION CONDITIONS; Common

<table>
<thead>
<tr>
<th>Runout Locating Angle</th>
<th>Runout Spacing</th>
<th>Total Number Z-buffer Elements</th>
<th>Total Number Tool Moves</th>
<th>Simulation Time Pentium 300 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00015&quot;</td>
<td>105.5&quot;</td>
<td>0.0075&quot;</td>
<td>293967</td>
<td>39 sec (one pass)</td>
</tr>
</tbody>
</table>

Figure 9.20: Feedrate-controlled experimental force data for a sinusoidal partial immersion cut with a large amplitude in Z, ranging from a minimum depth of 0" to a maximum depth of 0.5625". The cutting and simulation conditions are also provided above.
In these test cuts, the majority of the forces experienced using optimized feeds are slightly less than the desired threshold. This is most likely the result of three factors. First, the contact area is calculated using the worst case conditions experienced during a tool move, a conservative approximation. Second, the cutter runout amount and locating angles are selected to slightly over-estimate forces as compared to the force values measured for mechanistic constant calculation, as may be seen in Figures 9.12 - 9.15. Finally, cutter deflection is not considered. For the slotting and down milling that occurs during these cuts, the milling forces deflect the cutter away from the stock, decreasing the maximum chip thickness experienced.

While the results provided above demonstrate the systems capability to maintain a desired force level, these represent some of the better results obtained. Variations in cutter wear, dynamic conditions, the cut conditions relative to those used when calculating the material constants, and other parameters that affect the milling forces, all contribute to variations in the forces produced by a given set of variable feeds. The measured force values can vary appreciably from the desired force level, even between two identical cuts. This may be seen in Figure 9.21a, where exact same cutting conditions as presented in Figure 9.18 are used, but the results are appreciably different. The only variation between these two cuts is that the experimental force values shown in 9.18 were acquired immediately following a tool change, and those in Figure 9.21a were acquired when using a used cutting tool. Figure 9.21b also demonstrates the variations that can exist in the experimentally measured versus predicted forces. In this figure the cutting conditions are the same as presented in Figure 9.20, except that a stepover distance of 0.1125" (0.3 diameter) was used. The difference here is most likely the result of the large deviation from the cutting conditions present during data acquisition for material constant calculation, where a stepover distance of 0.2625" was used.

In general application, the measured forces can vary appreciably from the desired values. This is primarily the result of cutter deflection and other milling dynamics, which can greatly increase milling forces (particularly during 'chatter'). In these controlled test cases, which are similar to the conditions present at the site of our industrial research partner, these dynamic effects do not play a major role. However, these effects can occur frequently during general application. Currently this automatic feedrate generation system does not predict chatter, or other dynamic effects. However, it has been demonstrated through industrial application that even when the desired force threshold can not be maintained, safe operation can occur at reduced milling times as a result of the increased control over feedrate.
Figure 9.21a: Force results for the cut conditions presented in Figure 9.18 (including feed values), only using a different, but identical, cutting tool.

Figure 9.21b: Force results for the cut conditions presented in Figure 9.20, except a stepover distance of 0.1125" (0.3 diameter) was used. The feed values were re-calculated for this distance.
9.7 Practical Implementation of Automatic Feedrate Selection

A primary goal of this research is to develop a practical feedrate selection system capable of simplifying the feedrate selection process and improving process times in an industrial environment. While laboratory testing is required for validating the underlying theories and basic implementation of this system, industrial testing is also essential. This is necessary to validate the system for robustness, computational efficiency, ease of use, and to provide feedback regarding other 'real-world' issues.

This industrial testing took place at the site of our research partner, who manufactures turbomachinery using 5-axis CNC milling. Usage of the automatic feedrate generation system on one of their production parts is presented below. Included in this description are details of the procedures currently required for use of this system; a summary of this methodology is provided at the end of this section.

The sample application is a pre-finish operation on a centrifugal compressor blade, providing a great deal of surface variation due to the cusps left by the prior roughing operation (see Figure 9.22). The part material is 15-5 stainless steel, the cutting tool is a 0.25" carbide ball end cutter, and milling occurred on a Boston Digital 505 5-axis CNC mill. Unfortunately, it was not possible to measure force data during these tests. The fixture required for mounting the part on the mill’s rotary table made it difficult to attach the load cell. Also, it is impossible to effectively measure the estimated $X_{CT}, Y_{CT}$ forces during 5-axis milling with the current data acquisition system. These forces lie orthogonal to the cutter axis, which tilts at continually varying angles relative to the load cell as a result of the 5-axis motion. This causes the $X_{CT}, Y_{CT}$ forces to contribute variably to the $X_M, Y_M, Z_M$ directions measured by the load cell. The necessary transformation requires accurate knowledge of the cutter position and orientation, which is not available due to the closed architecture of the BD3200 controller. It is recommended that a spindle mounted load cell be acquired, which would eliminate this issue.

This example part was selected for two primary reasons. First, the tough 15-5 stainless steel is more capable of bringing out system flaws than aluminum. Second, manually optimized feeds exist for this part, allowing for comparison to current best industry practice. An experienced machinist had selected the feeds during prototyping to maximize performance. This included use of the standalone CNC machine model developed during this research (refer to Chapter 7) to ensure a constant cutter velocity relative to the...
Figure 9.22: Before and after images of the pre-finish operation for which adaptive feeds were generated. The central blade in the top image is the area of interest.
stock. In this case, manual experimentation determined that feeds selected to maintain a constant velocity of 24 IPM would provide the optimal balance between safety and performance.

The first step performed for automatic feedrate generation was to obtain the parameters required to limit the feedrates generated. To do this, the existing G-code part file, containing the existing feed values, was simulated to obtain milling force estimates (see Figure 9.23 for the simulation results). The resulting maximum force estimates are approximately 70 Lbs., but these values occur infrequently. A more typical peak force value is between 45 and 50 Lbs, and so 50 Lbs was selected for the desired force threshold. In cases where an existing production part file does not exist, a similar part could be used to obtain these values. These parameters could also be obtained using METCUT data, where the chip thickness values are provided directly, and the recommended feeds and speeds for a given set of cutting conditions could be simulated to arrive at the corresponding force values. Note that with regular use, the values required for a given class of part and stock material will become apparent. This is an area that deserves further research to allow for true 'pushbutton' automatic feedrate selection.

Note in Figure 9.23 that the mill time per blade using manually defined feeds is 13 min., 18 sec. Once the force threshold was determined as described above, a new version of the part program containing optimized feeds was generated. These feeds are defined to maintain a 50 Lbs. milling force, with no feed exceeding 55 IPM. The 55 IPM limit was selected as it's 10% greater than the maximum manually defined feed. Figure 9.24 provides the force simulation results for the optimized part file, and shows a 24.8% mill time reduction and a 31.5% peak force reduction. The optimized part file was then used to mill several blades. While the optimized part file appeared to work well during simulation, it resulted in appreciable tool wear in application. This made it impractical to implement these feed values for production.

A leading cause of tool wear is chip load, or the maximum chip thickness experienced by the cutter. High chip loading produces a great deal of thermal energy, which leads to cutter degradation [NDW89]. Therefore, when excessive cutter wear was encountered, the maximum simulated chip thickness values for the manually and automatically defined feeds were examined (refer to the maximum chip thickness plots of Figures 9.23 and 9.24). The manually specified feeds were found to produce a typical peak chip thickness in the region of 0.00088" (although it briefly reaches as high as 0.0017" towards the end of the cut). However, the optimized feeds were found to frequently produce a maximum chip thickness
Figure 9.23: Simulation results for the manually optimized feedrates (including use of the NC machine model to maintain constant velocity), originally used during production of the part.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure 9.24: Simulation results for automatically generated feedrates to maintain a force of 50 Lb., with a maximum feed of 55 IPM and no limits on chip thickness.
value more than twice this amount. It is not uncommon for a small area on the cutter to contact the stock during pre-finish operations, resulting in large chip thickness values when maintaining the desired 50 Lbs. force. Unfortunately, this high chip loading results in rapid tool wear, and could even result in catastrophic flute failure in some cases as a result of the force concentration. While methods have been developed to predict flute failure due as a function of feedrate [G90] [E93], these require knowledge of cutter geometry and material properties well beyond what is readily available, making them impractical for industrial application. Therefore the more direct approach of allowing for a user-defined maximum allowable chip thickness was implemented in order to control the problems related with high chip loading.

A value of 0.001" was selected as the maximum chip thickness, as this is slightly greater than the typical value experienced using manually defined feeds. Also, the maximum allowable feed was increased to 70 IPM, as the resulting feed-per-tooth values are less of a concern with chip thickness control. A second optimized part program was then generated to maintain a constant 50 Lbs. when possible, while satisfying these constraints. While limiting the maximum allowable chip thickness did result in a slight mill time increase relative to optimized feeds with no chip thickness control, the rapid tool wear was eliminated. If the 0.001" maximum chip thickness value did not produce satisfactory results, it could be further reduced. Note that the chip thickness could also be increased to just below the onset of tool wear, to further optimize the process. Furthermore, limiting both milling force and maximum chip thickness still provided an 18.2% mill time reduction vs. manually defined feeds (see Figure 9.25), an appreciable reduction as these savings are made against the current best industry practice. Much of this reduction was due to surface cusps left by the roughing operation, as the cutter velocity varied appreciably as it crossed them, while the manually defined feeds are limited to provide safe operation at the maximum depth of cut.

There is another small benefit provided by the use of a chip thickness control. The generation of optimized feedrates with no chip thickness limitations required a computation time of 8 min., 24 sec. on a 300 MHz Pentium based PC (see chart in Figure 9.24). However, when the maximum chip thickness was limited to 0.001", a computation time of 7 minutes, 14 seconds was required, or a 13.9% reduction in computation time (see chart in Figure 9.25). This is the result of a reduced number of iterations required before an exit condition is met in the feed generation loop. When a maximum allowable chip thickness condition is met, the software no longer iterates on the estimated force, reducing computation time.
**SIMULATION RESULTS**

<table>
<thead>
<tr>
<th></th>
<th>Max Feed</th>
<th>Max Force</th>
<th>Max Chip Thickness</th>
<th>Mill Time (Per Blade, x17)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIMULATED VALUE</td>
<td>70 IPM</td>
<td>50 Lb.</td>
<td>0.001&quot;</td>
<td>10 min, 9 sec.</td>
</tr>
<tr>
<td>% Difference from Manual Feeds</td>
<td>+40.0%</td>
<td>-31.5%</td>
<td>-70.0%</td>
<td>-18.2%</td>
</tr>
</tbody>
</table>

**CUTTING CONDITIONS**

<table>
<thead>
<tr>
<th>Cutting Tool</th>
<th>Spindle Speed</th>
<th>Feedrate (IPM)</th>
<th>Stock Material</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125&quot; Carbide Ball End</td>
<td>4500 RPM</td>
<td>Variable to maintain 50 Lb, limited to 55 IPM Max</td>
<td>15-5 Stainless Steel</td>
</tr>
</tbody>
</table>

**SIMULATION CONDITIONS; Cylinder Portion**

<table>
<thead>
<tr>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_{rc}$</th>
<th>$P_2$</th>
<th>Disc Definition</th>
<th>Helix Angle</th>
</tr>
</thead>
<tbody>
<tr>
<td>31693.4</td>
<td>0.3745</td>
<td>0.2944</td>
<td>0.0915</td>
<td>0.01&quot;</td>
<td>30°</td>
</tr>
</tbody>
</table>

**SIMULATION CONDITIONS; Ball Portion**

<table>
<thead>
<tr>
<th>$K_{TC}$</th>
<th>$P_1$</th>
<th>$K_{rc}$</th>
<th>$P_2$</th>
<th>HelixEnd</th>
<th>MinHelix</th>
<th>FluteEnd</th>
<th>Disc Defn</th>
</tr>
</thead>
<tbody>
<tr>
<td>22162.2</td>
<td>0.3961</td>
<td>0.023</td>
<td>0.3638</td>
<td>70°</td>
<td>0°</td>
<td>70°</td>
<td>2°</td>
</tr>
</tbody>
</table>

**SIMULATION CONDITIONS; Common - FEEDRATE SELECTION**

<table>
<thead>
<tr>
<th>Runout (inches)</th>
<th>Runout Locating Angle</th>
<th>Z-buffer Spacing</th>
<th>Total Number Z-buffer Elements</th>
<th>Total Number Tool Moves</th>
<th>Simulation Time Pentium 300 MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00025</td>
<td>45°</td>
<td>0.0075&quot;</td>
<td>614656</td>
<td>6454</td>
<td>7 min. 14 sec.</td>
</tr>
</tbody>
</table>

Figure 9.25: Simulation results for automatically generated feedrates to maintain a force of 50Lb., with a maximum feed of 70 IPM and a 0.001" limit on chip thickness.
The example updated part file containing the optimized feeds is now being used in production, and the optimized feedrate generation system is being implemented as regular practice at our research partner's site. In addition to the semi-finish operation described above, two roughing operations and a second semi-finish operation also had optimized feeds generated; the results are summarized in Table 9.5. The minimum mill time reduction occurred during the first roughing operation, a result of fairly constant cutting conditions that minimize potential for increased velocity. The time savings increases in later operations when cusps left by prior operations provide increased geometric variation in the milled surface. An overall average improvement of 13.9% (weighted by the respective mill times for each operation) was achieved by the software system developed during this research, as compared to current industry best practice.

<table>
<thead>
<tr>
<th>SUMMARY OF OPTIMIZED FEEDRATE CONSTRAINTS AND RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-----------</td>
</tr>
<tr>
<td>ROUGH 1</td>
</tr>
<tr>
<td>ROUGH 2</td>
</tr>
<tr>
<td>SEMI-FINISH 1</td>
</tr>
<tr>
<td>SEMI-FINISH 2</td>
</tr>
</tbody>
</table>

Table 9.5: A summary of the optimized feedrate constraint values and results for all operations of the commercial part program that had optimized feeds generated.

The procedure for use of the existing software in a practical application is provided below; Appendix E contains a users guide for software operation. Note that in future research some of these procedures should be automated, but at this time they must be manually performed.

1. Determine material constants for the combination of cutter, stock material, and cut conditions.
   Test cuts must be performed if no data for similar materials is available. Note that over time, libraries of material constants may be built, and this procedure will become less necessary.

2. Determine the desired force level and maximum allowable chip thickness.
   This is currently performed through force simulation of the part to be optimized, or of a similar part, or using knowledge from the previous feedrate generation for a similar part.
3. Define an appropriate stock model for the simulation.

Currently rectangular stock or turned stock may be defined. The turned stock is defined through
the use of a NURBs curve that describes the 2D profile of the turning.

4. Define an appropriate cutter representation for the simulation.

The cutter definition is described in more detail in Chapter 3. It was found in this research that
typically cylindrical cutter portions have helix angles of 30°. On ball portions, excellent
simulation results were achieved when the 30° helix angle was linearly interpolated to some
minimum value at the location where two of the flutes terminate; the minimum helix and flute
termination location are user defined. Over time libraries of cutter descriptions may be built.

5. Define the appropriate cutter runout amount and locating angle for the simulation.

These may be obtained when matching simulated force data to the test cuts performed to obtain
the material constants, or they may be selected for conservative results. In the course of this
research, simulated runout amounts in the range of 0.0001" - 0.00025" were found to provide
adequate force matching for ball end mills, while flat end mills required runout values between
0.00025" and 0.001". Using the upper ends of these ranges should help ensure conservative
estimates. The locating angle is less critical and may be selected arbitrarily.

6. Enter the required information into a file and run the automatic feedrate generation software.

Libraries of feedrate generation files may be built to serve specific classes of parts, with all of
the appropriate data already contained in them.
CHAPTER 10

CONCLUSIONS AND CONTINUING RESEARCH

10.1 Chapter Introduction

The primary goal of this research is the development of an automatic feedrate selection system that is based on mechanistic modeling principals, and is capable of practical implementation in an industrial environment for 3 and 5-axis sculptured surface machining. By practical it is meant that the system is reliable, computationally efficient, and easy to use, and that its use improves the efficiency to the milling process. These goals require robust and efficient modeling methods, and generalization of these methods for the broad applications encountered in a shop environment. The resulting system is capable of application in 3 and 5-axis milling, and may be applied to a variety of cutting tool types, CNC machine types, stock geometries, setup configurations, and cutting conditions. This system is currently being used in industry to aid in the production of turbomachinery, one of the most complex five-axis production milling operations, providing an excellent test case for the implementation and validation of this system.

This chapter summarizes the research that was performed during the development of this mechanistic based feedrate selection system, and draws some conclusions based on the results achieved. Suggestions for future research, and for enhancements to the feedrate selection system, are also provided.

10.2 Summary

There were several mechanistic based feedrate selection models in existence prior to this research, but all have limitations that make commercial implementation impractical. Most of the early models [KDL82] [KD83] and some of the more recent models [BYK94] rely on manual data input of the cut geometry, rather than obtaining it automatically from a geometric model. This prevents practical implementation, as it would be prohibitively difficult to manually calculate and enter the contact area between the cutter and the changing stock geometry for every tool move in a part program. While some of
the more recent applications do implement a geometric model for cut geometry simulation, many rely on analytical surface data to provide the cut geometry description and stock modeling [MA91] [YP93] [EDE98]. This leads to efficiency and robustness concerns, as all but the simplest of part geometries, milled with only a flat end cutter, would require thousands of small surface definitions to accurately reflect the milled part geometry to the required level of detail. This is not feasible with current surface modeling and computer technology. Finally, none of literature to date has mentioned the explicit inclusion of a CNC machine model. The inclusion of this model is necessary for accurate 5-axis simulation. Additionally, special exit conditions from the feed calculation loop, e.g. a maximum allowable chip thickness, are also required for a practical implementation.

An integrated modeling approach is implemented in this research, in which the component models (CNC machine, geometric, and mechanistic) are separate but share data in a closely integrated manner. Of the component models, the identification and definition of the CNC machine model was a particular benefit of this research. The machine model simulates the kinematic and controller behavior of a specific CNC machine, and is required for G-code translation, machine coordinate to part coordinate transformations, and for calculating the cutter velocity relative to the stock. Explicit inclusion of a unique machine model greatly simplifies the task of performing automatic feedrate generation on a variety of machine styles.

The CNC machine model is also independently valuable from its role in mechanistic model based feedrate generation, as it also provides a means of maintaining a constant cutter to stock relative velocity when 5-axis milling. This is achieved through the calculation of variable feeds that result in a constant relative velocity, providing much greater velocity control than was previously possible. The use of these variable feeds demonstrates the capability of the CNC controller to handle a large number of variable feeds, which is necessary for force-based automatic feedrate generation.

Simulation of the material removal process is a basic requirement of any automatic feedrate generation method, as feedrate requirements vary with cutting conditions. Therefore, geometric modeling was a critical component of this research, required to maintain a record of the current state of the stock during milling. An extended Z-buffer model of the stock geometry was implemented in this research. This approach combines the required accuracy with computational efficiency, and provides robust behavior, both when performing material removal simulation and during calculation of the cut geometry data. The Z-
buffer model was expanded during this research to allow for the use of non-prismatic stock models. Currently, stock models that are the result of a lathe turning operation are supported, and the methods developed may be applied to the definition or other non-prismatic stock, such as castings. The Z-buffer model was also expanded to allow for the use of ball end cutters during 3 and 5-axis milling simulations. The ability to read and write the Z-buffer models to a file was also implemented, so that different milling operations could be performed individually on the same part, and intermediate stages stored to disc. Finally, the definition of the 'intersected segment', defined in Chapter 4, was found to greatly simplify the tasks of updating the Z-buffer model (to reflect material removal) and calculating the cut geometry parameters. A shaded image rendering of the in-process milling, as well as graphical rendering of the individual Z-buffer elements, was also developed. This proved very useful for model validation.

Other geometric developments include the ability to calculate cut geometry parameters for ball end cutters, and efficiency improvements were made in the calculation of all cut geometry parameters. This required that the cut geometry parameters be defined in quadrants, so that the 'back' of the cutter could be appropriately modeled. It also required cut geometry calculations using intersection data from the net toolpath envelope. Previously, each axial disc had been modeled as an individual swept toolpath for the calculation of these parameters. This change was necessary to ensure accurate geometric simulation for all cutter geometries, and additionally resulted in an improvement in computational efficiency.

The addition of ball end cutters also required modification to the mechanistic modeling methods. The basic form of the classical discrete mechanistic model was used in this research as a result of its proven accuracy, robust behavior, and readily obtainable input parameters. However, the existing form of this model was specific to flat end mills, and therefore changes were necessary to accommodate a more generalized cutter description. The cutter radius, cutter surface normal, helix angle, flute length, number of flutes, and axial disc thickness are all variable on a ball end cutter, and so these parameters were explicitly included in the mechanistic model. Inclusion of these parameters required the development of a simple means of defining them in the discrete model, using a minimum of input data. Inclusion of these parameters also necessitated alterations to the average and instantaneous chip thickness calculations, including the mechanism for modeling cutter runout. Graphical rendering of the discrete cutting tool,
including the contact area, a chip thickness color map, and the cutter flute geometry, was developed to aid in validating the discrete cutter model.

The resolution of the rotational simulation in the mechanistic model was also improved, while simultaneously improving computational efficiency. This was achieved using a variable rotation angle. In this approach, a course angular increment (8°) is used to locate the peak force region, and then a fine (1°) rotational increment is used to identify the actual peak force within this region.

Other mechanistic modeling developments include methodology improvements for deriving the mechanistic material constants. The use of a least squares method was added for averaging several empirical data sets with minimum error. Methods were also added to support mechanistic constant calculations for a ball end mill. This included dividing the cutter into two distinct portions, the cutter ball and the cutter cylinder, and calculating different mechanistic material constants for each. This better reflects the differing flute geometries on these two portions of the cutter. The use of a ball end cutter also necessitated the development of analytical methods for obtaining the cut geometry without the use of a Z-buffer model when calculating mechanistic model parameters. This provides an exact representation of the contact area during calculation of the constants in order to minimize error, and also to reduce the chance for user error in defining the cutting conditions present during force data collection.

Finally, several improvements were made to the automatic feedrate selection system as a result of testing in an industrial environment. The primary of these is the addition of ‘chip thinning’, or limiting the maximum allowable chip thickness. While the mechanistic model is adequate to control feeds when a substantial amount of material is being removed, it can calculate feeds that result in too high of a chip load during light cuts, resulting in accelerated cutter wear. Control over the maximum allowable chip thickness limits these effects, allowing for robust automatic feed calculation regardless of cutting conditions. Chip thinning also improves computational efficiency by exiting the force estimation loop when the maximum value is obtained. Efficiency gains were also achieved through the explicit inclusion of other exit conditions from the iterative feed calculation loop, as presented in Chapter 3.

Computational efficiency gains were also made in the algorithms that calculate the feed value for each force calculation iteration (recall that this approach calculates the feedrate iteratively, such that the resultant force estimate approaches some desired value). This improvement requires use of the output feed
from the previous tool move as the initial feed value for the current move, as this can immediately provide the correct solution for relatively constant cutting conditions. If this is not the case, then the second iteration uses a linearly interpolated feed, based on the ratio of the desired to the currently estimated force values. Although the relationship between feedrate and force is nonlinear, when large changes in feed are required between tool moves this greatly reduces the number of iterations required in the bisection solver by placing the feed estimate in the appropriate range. As the relationship between feedrate and force is more linear over small changes in feed, this also increases the probability of selecting the appropriate feed value on the second iteration when only small changes in feed are required.

The main advancement made in this research, however, is the integration of these methods into a working system that is capable of automatically generating feedrates that are optimized to the unique cutting conditions present during each tool move. This offers greatly improved control over feedrate selection, particularly for geometrically complex jobs such as 5-axis sculptured surface machining. This system has the potential to reduce the time required to select feed values for a given job, and also to provide more confidence in the feed values selected. Another benefit from the use of optimized feedrates is decreased milling time, with times savings of 5-50% experienced, dependant on part geometry and the constraint values selected. A real savings of 13.9% was achieved on a production five-axis milled sculptured surface part, as compared to the best practice of an experienced shop using advanced relative velocity control software that was developed earlier during this research. The net result from the use of this system is increased manufacturing efficiency, with greater confidence in the feed values selected.

10.3 Suggestions for Continuing Research

The suggestions for future research have been divided into three sections. The first section lists critical enhancements that would immediately make these methods more valuable in an industrial environment. The second section lists enhancements that would offer less critical benefits, and may also be implemented in the existing software. The third section is more research oriented, covering areas where this technology could be exploited or enhanced only after a substantial research effort.
10.3.1 Critical Enhancements

The most beneficial enhancement for a commercial implementation would be the support of an increased number of cutting tool geometries. Currently ball end and flat end mills may be simulated, and while these represent the majority of cutting tools available, many other styles are also frequently used. A direct expansion of the modeling methods used in this research would allow for the inclusion of 'bull nosed' and tapered cutters, which should be the first types added. Their addition would allow the system to model all of the geometries used by a large majority of the NC market. Insert style cutters could be added next, with the inserts being modeled as discontinuous flute segments using existing flute geometry parameters. Finally, 'shaped' cutters such as routers, dovetails, and other free-form geometry could be added using a boundary representation of the swept toolpath envelope composed of NURBS, or some other standard surface representation created by lofting the cutter profile.

Another basic requirement, although not necessarily an enhancement, is to benchmark the software developed during this research against the commercially available automatic feedrate generation software. This includes HighFeed by Mastercam [M00], and Vericut Optipath by CGTech [V00]. First, it is desired to see if these packages operate effectively in a five-axis sculptured surface environment, which represents the most complex conditions for CNC milling. Also, both packages implement a volumetric approach at automatic feedrate selection, which effectively estimates average force values. This leads to the assumption that these packages would not be able to optimize feeds to the same degree as the mechanistic approach developed during this research, which estimates peak force values. Finally, the use of these packages could provide insight to features and attributes that did not arise during the course of this research but are desirable in a commercial application of automatic feedrate selection.

Another practical enhancement would be the automatic identification of full immersion slot cuts. While this is not an issue in some classes of parts, it can be critical in others. Some parts require either no or only a small number of slot cut, with the majority of the cuts being partial immersion cuts that are stepped over from the edge of the stock or the occasional slot cut. When slot cuts occur infrequently their presence is not a concern. In other classes of parts, such as turbine and compressor wheels, slot cuts occur more frequently, as a cutting cycle involving slots is repeated at a number of layers of material removal, and at a number of locations on the part. This relatively large number of slot cuts contributes significantly
to tool wear. Although the mechanistic model can adjust feed automatically for current cutting conditions, it is difficult to differentiate between a full slot cut and a partial immersion with a 'deep' radial depth (greater than one radius) using only mechanistic model and maximum chip thickness data. Although the peak forces are similar in both cases, significantly more tool wear can occur in the case of the full slot cut. The situation is similar in the case of the maximum chip thickness, as this occurs directly in the direction of travel at a flute locating angle of $\beta = 90^\circ$, a position that is present for both full slot cuts and deep partial immersion cuts. Therefore the same maximum chip thickness value is returned in either case, and they cannot be differentiated. Currently a flag in the G-code file is used to indicate the presence of a slot cuts so that the feed may be inhibited accordingly, but automatic identification through the analysis of the contact area would provide a superior method. Ideally, adjustment of the mechanistic cutting constants to the current cutting conditions, so that the subtle differences in force would be reflected in the feeds calculated, would be desirable over simply inhibiting the feed by some percentage.

10.3.2 Suggested Enhancements

One important topic for investigation is automation of the feedrate selection system. Currently, an existing production part file is simulated prior to feedrate selection in order to obtain the milling force and chip thickness constraints required by the feedrate selection system. This prevents true 'black box' operation, where only toolpath and part geometry information are required. While a database of the required data could be built over time using the current method, this presents an obstacle to automation. Similarly, pre-calculation of the material constants is also required. To simplify automatic feedrate selection, it is desirable to obtain as many input parameters as possible in an automated manner.

One possible source of automation is through the use of METCUT data. This method is similar to the current approach of 'pre-simulating' a production part known to have acceptable forces and chip thickness in order to obtain this information, except that no existing production part is required. Under this approach, the user would be required to enter the material constants, material type, cutter information, a stock description, and a G-code file. Geometric simulation would then be performed, and the contact areas present during each tool move would be stored. These contact areas would then be analyzed to arrive at a range of average cutting conditions that could be looked up in the METCUT CUTDATA tables. Force
simulation would then automatically occur using the cut descriptions contained in the METCUT data which most closely match the cutting conditions, in order to arrive at values for chip thickness and milling force that are known to be acceptable. Once this information is known, feedrate estimation would occur. This would require only the use of the CNC machine model for estimation of relative velocity values, which would then be used with the previously stored contact area data to perform automatic feedrate selection.

Another area of both theoretical and practical interest is milling process dynamics. As the cutting tool rotates, the force vectors acting on it change cyclically as a result of the varying flute locations. These continually varying forces result in variable deflections in the system. Cutter deflection is typically the primary motion of interest, but appreciable deflection can also occur in the machine and stock, which may also require modeling depending on the situation (e.g. part deflection may dominate in thin walled parts). Cutting dynamics are of interest as they directly affect the milling forces experienced. This is particularly true when a condition referred to as 'regenerative waviness' [E93] exists. This occurs when the tooth passing frequency matches a dominant mode of the system, leading to a regenerative growth in the deflections through forced vibration. When the cutting process becomes unstable due to such regenerative feedback, an audible ringing, referred to as chatter, results. The milling forces increase greatly when chatter conditions exist, up to several times the values that are present under stable conditions, as a result of the large variations in chip thickness that occur as the cutter vibrates. In addition to the danger of the increased forces, the effects of the cutter vibrations are also clearly visible in the milled part in the form of a scalloped or wavy surface, which is often an unacceptable surface finish for the part.

One approach that is used to predict the effects of milling dynamics is to perform a discrete time simulation. This method has been implemented by several researchers, independent of feedrate generation schemes, with a good deal of success [MA91][ESE98]. However, an iterative solution is required for each time step to arrive at an equilibrium solution, and the changes in force that occur as a result of the rapidly rotating cutter necessitate the use of a very small time step, relative to the milling time. Therefore this method is not practical for industrial application using current computer technology. However, faster methods also exist. These involve the concept of a 'stability lobe', as developed by Tlusty [T90]. In this approach, the dynamic effects are pre-calculated and mapped to a chart of feedrate vs. spindle speed, defined for a given set of cutting conditions. Stable operating regions, or 'stability lobes', are indicated on
these charts (i.e. feeds and speeds that will not result in chatter for a given set of cutting conditions). These charts could be used in some rapid look-up implementation (e.g. a hash table) to verify that the currently selected feedrate, and the neighboring feedrates, will not result in chatter. If necessary, the feeds can be decreased into a stable region (the feeds should not be increased, as this would be contrary to the exit condition which resulted in that feed value). The spindle speed could also be adjusted, although this typically has a slower response time on the machine, and may even require cutter rotation to cease temporarily while drive gears are changed. Note that an isolated feed in the unstable region for a typical tool move is most likely tolerable, as there would not be sufficient time for significant chatter to develop.

A final area of practical extension for this research is cutter wear prediction. Using this system, a contact area and chip thickness distribution is generated for the discrete cutter model during every tool move in a simulation, and the move time is also known. This information can be used to estimate tool wear, as it allows for calculation of the amount of material removed, the maximum and average thickness of the material removed, and the rate of material removal, for each disc element on the discrete cutter model. These parameters are directly related to tool wear, and relationships could be developed that estimate the progression of tool wear during the milling process using this information.

10.3.3 Research Possibilities

One area of research that could be undertaken using this software is the development of an integrated process definition system. Currently, toolpaths are generated using CAM software, separate from the feedrate definition process. Care is generally taken during toolpath definition to ensure a fairly well defined and consistent set of cutting conditions, which are necessary to minimize the effort required for manual feedrate selection (although typically several iterations may be required to arrive at reasonable feeds). However, these toolpaths do not necessarily optimize total milling efficiency, which contains both spatial and velocity components. The spatial component refers to the net length of all toolpaths in a part file, and the velocity component refers to the variable rate at which the cutter moves along these toolpaths. Under the current method, attempting to maintain consistent cutting conditions during toolpath definition imposes a constraint unrelated to spatial efficiency, and the subsequent use of these constant toolpaths in feedrate definition limits velocity maximization to a single set of conditions. Only when toolpath definition
and feedrate specification are integrated can both spatial and velocity parameters be optimized such that the milling time is minimized. This would be the goal of an integrated process definition system.

Another excellent application for this research not yet in common usage is the solution of both the geometric and the discrete models using parallel processing. As the solution of these models requires a repeated set of identical, independent operations, parallel processing could greatly enhance the performance of this system. A parallel processing approach at Z-buffer modeling (referred to in that research as ‘ray casting’) was previously performed by Menon et al. [MR92], but no similar effort has been published for mechanistic modeling. In mechanistic modeling, identical calculations are performed over all discrete axial discs during calculation of the contact area and milling forces. These calculations are also repeated over all discrete rotation angles during force calculation. If implemented on a multiple processor computer, these discrete tasks could be divided between the processors, to arrive at a substantial improvement in efficiency. This could be of particular benefit during automatic feedrate generation for sculptured surface machining, which typically requires several iterations of force calculation for most tool moves.

10.4 Conclusions

The primary goal of this research was the development of an automatic feedrate selection system for generalized milling conditions that could be readily implemented as a standard practice in industry, using currently available computer and software technology. While two known commercial implementations of automatic feedrate generation are currently available, they are based primarily on volumetric material removal methods, which provide only a crude ability to estimate required feeds. The volumetric approach is also limited in terms of future expansion, as it provides no additional information regarding the milling process outside of feed estimates (i.e. no force vectors). Therefore a more advanced approach at automatic feedrate generation was desired. To meet this goal, this research applied a mechanistic modeling based approach to the feedrate generation problem, and a working system with significant potential for application and future research was the result.

On a practical level, this research has successfully demonstrated the success of using an integrated modeling approach at automatic feedrate selection. This method combines a geometric model and CNC
machine model in support of a generalized discrete mechanistic model. The discrete Z-buffer approach has proven to be an efficient and accurate tool for the geometric modeling of material removal, and also for providing the data required for the calculation of the cut geometry parameters. The CNC machine model has provided accurate relative velocity values during five-axis milling, required for chip thickness. The generalized form of the discrete mechanistic model, and the supporting methods developed to provide the mechanistic material constants, have proven capable of accurate force estimation for a variety of cutting conditions. Finally, the integration of these three separate models into a unified system has proven effective through laboratory testing and industrial implementation at calculating optimized feed values, regardless of the part geometry, cutter type, or CNC machine style.

The success of this method has implications beyond the scope of feedrate selection. In addition to allowing for the selection of feedrate values, it also provides estimates of milling force vectors, which may be used to model cutter and part deflection, and in analyses of machine and cutter dynamics. Also provided is detailed cutter/stock contact area and chip thickness information, which could prove useful in predicting cutter wear on a 'macro level'. The chip thickness, force, and contact area information could also be used to study the thermal aspects of the milling process, which has the greatest effect on tool wear, providing for a 'micro-level' analysis of this phenomena.

While there is still more work that must be performed for this approach to become fully mature, the net result of this research is a set of methods already implemented into a working software system. This system is ready for application both in research and industrial environments, and can provide substantial benefits to both.
The physical geometry of the surface cut by a single cutting tool flute is trochoidal in nature as a result of the forward motion of the cutter during rotation. This forward motion is ignored in the idealized surface assumed when calculating the chip thickness using the dot product approximation.

It is not possible to use a direct calculation as is performed by the dot product approximation when calculating a chip thickness based on a trochoidal model. Rather, the surfaces generated by two subsequent cutting flutes are defined, and the distance between them is calculated at some discrete angular position to yield a chip thickness value. The 'discrete angular position' criterion is used as it follows the form of the mechanistic model, where forces are calculated as a function of rotation angle. There are also discrete time based trochoidal cutting models [MA91], developed for use in dynamic simulations.

The distance translated forward by the cutter may be defined as a function of the current flutes rotation angle using:

\[ x(\beta) = f((\pi - \beta)/2\pi)(N_f) \]  \hspace{1cm} (A.1)

where \( x(\beta) \) is the distance traveled by the cutter center as a function of \( \beta \) (in), \( \beta \) is the rotation angle of the current flute (rad), and \( N_f \) is the number of flutes at the current axial location.

Using this relation, it is possible to define the surface 'S' left by a given flute as a function of rotation angle. The X and Y components of the surface 'S' are defined for what we will call the initial flute, or flute 1, as:
\[ X_l(\beta_1) = R \sin (\beta_1) + x(\beta_1) \tag{A.2a} \]
\[ Y_l(\beta_1) = R \cos (\beta_1) \tag{A.2b} \]

where \( X_l(\beta_1), Y_l(\beta_1) \) define the surface \( S_1 \) left by the initial cutting flute (in),

- \( R \) is the cutter radius (in),
- \( \beta_1 \) is the current flute rotation angle (rad) of flute 1, and
- \( x(\beta_1) \) is the cutter translation as a function of \( \beta_1 \) (in).

Equation A.2 defines the surface created by a flute on the cutting tool as it both rotates about the cutter axis from the 180° position, and translates forward from a defined origin; this is shown on the left in Figure A1.

To define a chip thickness, the difference between this surface and the path traced by the subsequent flute, \( S_2 \), must be calculated, as shown on the right in Figure A1. For this operation to be performed, it is necessary to define a 'lead/lag' angle that aligns the rotation angle of the second flute that creates \( S_2 \) with the initial flute's rotation angle. This is required as the chip thickness value is calculated as the difference between \( S_1 \) and \( S_2 \) at some given cutter rotation angle \( \beta_2 \), which is defined relative to \( S_2 \), and the corresponding location along \( S_1 \) included in the difference was not generated at the angle \( \beta_2 \). This may be seen in Figures A1 and A2, and is the result of cutter translation during rotation, which causes the second flute to start at a \( \beta_2 = 180^\circ \) that corresponds to some value of \( \beta_1 \) that is less than 180° (as seen on the right in Figure A2). Therefore, for the purpose of calculating the trochoidal chip thickness value as a function of a single discrete rotation angle (\( \beta_2 \)), a lead/lag angle, denoted \( \epsilon \), must be defined such that:

\[ \beta_1 = \beta_2 + \epsilon(\beta_2) \tag{A.3} \]

where \( \epsilon(\beta_2) \) is the desired lead/lag angle function (rad).

The angle \( \epsilon \) may be viewed as the angle between two radius vectors, such as may be seen on the right in Figure A1. Starting at \( \beta_2 = 180^\circ \), where the maximum lag between \( \beta_1 \) and \( \beta_2 \) exists, the value of \( \epsilon \) reduces to zero as \( \beta_2 \) rotates to 90°; that is, the lag \( \beta_1 \) has relative to \( \beta_2 \) diminishes to zero as both vectors rotate.
Figure A.1: On the left, Trochoidal Surface S1 is created as flute f1 rotates $2\pi/N_f$ radians and translates forward one feed per tooth distance. On the right, the Trochoidal Chip Thickness $h(\beta)$ at some angle $\beta$ is the difference between the surfaces generated by the current flute, S2, and the initial flute, S1. In this view the actual cutter radius vectors are shown, originating at the instantaneous cutter centers that were present when the surfaces were generated. Note that the angle to the radius vector that created S1 lags the value of $\beta$ for which the chip thickness is desired.

Figure A.2: The trochoidal chip thickness is calculated as the magnitude of the difference between the vectors shown on the left. Note the lead/lag error in the angle for the corresponding S1 and S2 positions. On the right a means of obtaining the maximum lead/lag angle, $\epsilon_{\text{MAX}}$, is shown (valid for $f \ll R$; not to scale). When the value of $\beta$ for S2, $\beta_2$, = 180°, the corresponding $\beta_1$ value is $(180^\circ - \epsilon_{\text{MAX}})$; similarly when $\beta_2 = 0^\circ$, $\beta_1 = \epsilon_{\text{MAX}}$. The value of $\beta_1$ is defined for intermediate values as $\beta_1 = \beta_2 - (\epsilon_{\text{MAX}})(\cos(\beta(S2)))$.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
such that they are aligned with the X axis, with both $\beta_1$ and $\beta_2$ at $90^\circ$. From there, $\beta_1$ leads $\beta_2$ to a maximum lead, equal in magnitude to the maximum lag, at $\beta_2 = 0^\circ$. This is modeled using the relation:

$$
\varepsilon(\beta_2) = (\varepsilon_{\text{MAX}})(\cos(\beta_2)) \quad (A.4)
$$

where $\varepsilon_{\text{MAX}}$ is the maximum lead/lag angle value (rad).

The value of $\varepsilon_{\text{MAX}}$ may be calculated using equation A.2a, by setting the $X(\beta)$ amount equal to the feed-per-tooth value and solving for $\beta$. This process is simplified in that the resultant value of $\beta$ is generally rather small, allowing a small angle approximation of the sine term. Using this method is $\varepsilon_{\text{MAX}}$ defined as:

$$
\varepsilon_{\text{MAX}} = \frac{f}{(R + f\gamma)} \quad (A.5)
$$

where $f$ is the feed per tooth value (in/tooth), $R$ is the cutter radius at the current axial location (in), and $\gamma$ is the cutter rotation during one feed-per-tooth period ($=2\pi/N_f$; rad).

Equations A.4 and A.5 provide insight into the amount of error included in the dot product approximation for the chip thickness shown in Equation 1.2. When no lead/lag angle is present, the results from the trochoidal model are identical to that provided by Equation 1.2. In this case the distance between the surfaces that is returned is constant and equal to the feed per tooth value, which must then be projected onto a vector in the radial direction at the desired angle, yielding the identical results. All of the error is therefore introduced by neglecting $\varepsilon(\beta_2)$, which has a maximum amount described by A.5. As the value of $f\gamma$ is generally much smaller than $R$, the dominant relationship is $f / R$. Therefore it may be stated that the error is negligible for $f \ll R$, a necessary condition for use of the dot product approximation.

Equations A.4 and A.5 also provide the remaining information required for the definition of a trochoidal chip thickness value. The chip is defined as the difference between two surfaces created by two adjacent flutes, with the first surface being described by A.2a,b. The second surface is nearly identical in form and is defined as:
\[ X_2(\beta_2) = R \sin (\beta_2) + x(\beta_2) + f \]  
(A.6a)

\[ Y_2(\beta_2) = R \cos (\beta_2) \]  
(A.6b)

where \( X_1(\beta) \), \( Y_1(\beta) \) define the surface \( S_1 \) left by the initial cutting flute (in), \( R \) is the cutter radius (in), \( \beta \) is the current flute rotation angle (rad), and \( x(\beta) \) is the cutter translation as a function of \( \beta \) (in).

Note the primary differences between Equation A.6 and Equation A.2 are that Equation A.6 calculates the surface relative to \( \beta_2 \), and the \( X \) term (A.6a) has an additional feed-per-tooth amount added to it to account for the translation that occurred between flutes. Using Equations A.4 and A.5 it is possible to calculate the value of \( \beta_1 \) that corresponds to the current \( \beta_2 \) value, allowing for the calculation of chip thickness solely as a function of \( \beta_2 \). The chip thickness value is solved for as:

\[ h(\beta_2) = ((X_2(\beta_2)-X_1(\beta_2))^2 + (Y_2(\beta_2)-Y_1(\beta_2))^2)^{1/2} \]  
(A7)

Some results of the trochoidal chip calculation process described here are shown in Figures A.3 and A.4. Figure A.3 shows the results for the fairly small radius value of 0.09375" and a feed-per-tooth value of 0.001 in/tooth. These conditions represent a fairly common scenario, and notice that there is only a very small difference between the dot product and trochoidal chip thickness values. These differences will decrease as the cutter radius value increases. The maximum lead/lag angle in this case is 0.61°, well within the range of acceptability for the small angle approximation. Figure A.4 shows the results for the same radius value, but a rather large feed-per-tooth value of 0.01 in/tooth. Even in this somewhat extreme case, the maximum lead/lag angle in this case is 6.1°, still within the range of acceptability for the small angle approximation. While these conditions may not be a common scenario, it does demonstrate some of the effects of the trochoidal chip thickness model. Most notable is that the chip thickness value is non-zero at 0° and 180°. This further implies that for linear motion orthogonal to the cutter axis, the bounds of rotation must be extended beyond the 0° and 180° limits required by the dot product method in this case; i.e. some

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure A.3: Trochoidal (Dark Line) Vs. Sinusoidal (Light Line) Chip Thickness as a function of rotation angle for $R = 0.09375''$, $f = 0.001$ in/tooth. Note that there is only a slight variation between the two approaches for these fairly common conditions. The maximum lead/lag angle in this case is 0.61 deg.

Figure A.4: Trochoidal (Dark Line) Vs. Sinusoidal (Light Line) Chip Thickness as a function of rotation angle for $R = 0.09375''$, $f = 0.01$ in/tooth. Note the non-zero chip thickness values of the trochoidal form at 0 and 180 deg., as well as the overall increase in thickness from 0 to 90 deg. The maximum lead/lag angle in this case is 6.1 deg.
cutting occurs on the 'back' of the cutter. Also note that the average chip thickness is greater from 90° to 0° than from 180° to 90°, which differs from the symmetry found in the dot product approach. That said, note that the net effect on chip thickness values is still rather small, and as the greatest difference occurs at the smallest chip thickness values, the peak milling force does not change.

Although the trochoidal method presented here can provide slightly more accurate chip thickness values than the dot product approach actually implemented, the actual variations are negligible for most reasonable cutting conditions, and the trochoidal approach as currently developed is too limited for practical application. The primary limitation is that currently the surfaces calculated are those that lie in a plane orthogonal to the axis of the cutter. This means that contributions of axial components of cutter motion are not included in the chip thickness calculation. For practical application a model that includes surface components that lie in the axial direction as well must be derived.
APPENDIX B

MACHINE SPACE TO WORKPIECE SPACE COORDINATE TRANSFORMATIONS

This appendix provides an example of the machine coordinate to workpiece coordinate transformation that is required to map cutting tool locations and orientations from the machine-specific coordinates of the G-code data to the local coordinates of the Z-buffer model. This transformation is specific to a given machine style, and use of this transformation prior to other calculations helps limit the machine specific dependencies that exist during feedrate calculation. The example provided is for a Fadal table-on-table 5-axis mill, the kinematic style of which may be seen in Figure B.1.

The stock rides on a 2-axis rotary tilt table in the Fadal arrangement, allowing the translation between machine and workpiece coordinates to be simplified by the definition of two additional coordinate systems attached to the tilt table. These are referred to as the rotary axis coordinates, denoted RotaryA and RotaryB for the A and B axes respectively. These systems separate the rotational transformations into individual operations, which is important for two reasons. First, this helps simplify the linear translations required to locate the workpiece and machine coordinate origins relative to each other. Second, definition of these systems aids in the calculation of cutter to stock relative velocity values.

The rotary coordinate systems are shown in Figure B.2. The RotaryB system remains fixed relative to machine coordinates, not rotating with the B axis. The A axis is chained with the B axis, riding on a platform that rotates with B. The RotaryA coordinates rotate with the B axis are fixed relative to the A axis, similar to the relationship between RotaryB and B.

The workpiece coordinate origin is defined as the lower left hand corner of the stock model, as seen when viewed from above. The location of this origin relative to machine coordinates is dependant on setup of a specific job, as well as on the location of the machine coordinate origin, which itself is often arbitrary. The machine coordinate origin may be defined in the G-code file with a G92 command, which defines the machine coordinate origin as the current cutter location at the time of execution.
Figure B.1: In the table-on-table 5-axis machine configuration, the stock translates in X and Y, and rotates in A and B, while the cutting tool moves independently in Z.

Figure B.2: The RotaryB coordinates remain parallel to machine Coordinates at all times. Similarly, while RotaryA rotates on B, they do not rotate with A (X_A remains in the X_M, Z_M plane).
Alternatively, the machine origin may be defined through a manually entered set of \(X_M, Y_M, Z_M\) offsets that performs the same operation as a G92. Both the manually entered offsets and the G92 location (contained in the G-code file) will be referred to as the 'G92 offsets'. In either case, the cutting tool is manually jogged to some position relative to the workpiece, referred to as the 'Home Position'. The G92 offsets are then entered into the CNC controller to define the location of the machine coordinate origin relative to the cutter position. This is shown by the G92 offsets displayed in Figure B.3.

The cutter location is first translated from machine coordinates to Rotary_B coordinates. This is done relative to the Home Position through the use of three 'Rotary Table' offset values that are defined in \(X_M, Y_M,\) and \(Z_M\). These offsets, denoted \(RT_{X,Y,Z}\), define the distance from the Home Position to the center of A axis rotation for \(B=0\). In addition to these variable offsets is a constant offset value, \(C_{RT}\), which is a physical attribute of the rotary table. This offset represents the distance from the center of B-axis rotation to the surface on the A axis where the stock is mounted, and for the Jones and Shipman rotary table this has a value of 5.561". As shown in this Figure B.3, the Rotary_B Coordinates may be located with respect to machine coordinates by the relations:

\[
\begin{align*}
X_B &= X_M - G92_X + RT_X \\
Y_B &= Y_M - G92_Y + RT_Y \\
Z_B &= Z_M - G92_Z + RT_Z + C_{RT}
\end{align*}
\]

where \(X_B, Y_B, Z_B\) are the Rotary_B coordinates of the cutter position, \(X_M, Y_M, Z_M\) are the machine coordinate Locations, \(G92_X, G92_Y, G92_Z\) are the machine coordinates of the home position, \(RT_X, RT_Y, RT_Z\) are the user specified Rotary Table offset values, and \(C_{RT}\) is a constant that is a physical attribute of the rotary table.

Note that the home position indicated by \(G92_{X,Y,Z}\) is either contained in the G-code file (the G92 location), or specified by the user (the value of the offsets manually entered to the controller during setup).

From the Rotary_B coordinates, the cutter location is next transformed into Rotary_A coordinates. The transform from Rotary_B is composed of a translation in \(Z_B\) followed by a rotation about \(B\):

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Figure B.3: Definition of the machine Coordinate origin and the Rotary Table Coordinate origin for the Fadal table-on-table 5-axis configuration.

Figure B.4: Definition of the workpiece offsets used to specify stock placement on the rotary table.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
\[
X_A = X_b \cos(B) - (Z_b - C_{RT}) \sin(B) \tag{B.2}
\]
\[
Y_A = Y_b
\]
\[
Z_A = X_b \sin(B) + (Z_b - C_{RT}) \cos(B)
\]

where \(X_A, Y_A, Z_A\) are the Rotary\(_A\) coordinates of the cutter position.

The workpiece coordinates of the cutter location are then obtained through a transformation from the Rotary\(_A\) coordinates. This is performed by a rotational transformation in \(A\), followed by a translation performed in the Rotary\(_A\) coordinates. The translation is required as the workpiece coordinate origin lies at the lower left hand corner of the stock model when viewed from above, but the Rotary\(_A\) origin lies at the center of \(A\) axis rotation. Therefore workpiece offsets are required to simulate "stock placement" on the surface of the rotary table (see Figure B.4). Once these have been defined, the transformation from Rotary\(_A\) coordinates to workpiece coordinates is performed as:

\[
X_{wp} = X_A \cos(A) + Y_A \sin(A) + WP_X \tag{B.3}
\]
\[
Y_{wp} = -X_A \sin(A) + Y_A \cos(A) + WP_Y
\]
\[
Z_{wp} = Z_A + WP_Z
\]

where \(X_{wp}, Y_{wp}, Z_{wp}\) are the workpiece coordinates of the cutter position, and \(WP_X, WP_Y, WP_Z\) are the workpiece stock offsets.

The \(WP_Z\) offset shifts the cutter locations up and down along the \(A\) axis. It is only required if the stock is elevated from the surface of the rotary table, or if a different than actual stock 'height' in \(Z_{wp}\) is modeled.

Now that the cutter position has been transformed from machine to workpiece coordinates, the cutter orientation must also be transformed. In the Fadal arrangement, the cutter has a permanent \([0,0,1]\) orientation in machine coordinates, as its axis of rotation is fixed in \(Z_M\). However, the orientation varies relative to the stock due to rotations in \(A\) and \(B\). Additionally, the orientation varies continuously throughout the tool move, so the swept toolpath envelope used to model a tool move has a single fixed
The orientation vector is calculated using the angular positions that occur at the middle of the tool move, calculated as:

\[ A_{\text{mid}} = A_i + 0.5 (A_f - A_i) \]
\[ B_{\text{mid}} = B_i + 0.5 (B_f - B_i) \]

where \( A_{\text{mid}}, B_{\text{mid}} \) are the angular positions at the middle of the tool move,

\( A_i, B_i \) are the angular positions from the G-code file for the initial tool position, and

\( A_f, B_f \) are the angular positions from the G-code file for the final tool position.

The initial and final tool positions refer to the two cutter locations that bound the ends of a swept toolpath envelope. Using the average rotation angles, the cutter orientation vector for a given toolpath envelope is calculated as:

\[ A_x = -\sin(-B_{\text{mid}}) \cos(-A_{\text{mid}}) \]
\[ A_y = \sin(-B_{\text{mid}}) \sin(-A_{\text{mid}}) \]
\[ A_z = \cos(-B_{\text{mid}}) \]

where \( A_{XYZ} \) are the workpiece coordinate components of \( A \), the cutter orientation vector.

This is shown graphically in Figure B.5 on the following page.
Figure B.5: Definition of the workpiece offsets used to specify stock placement on the rotary table.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
APPENDIX C

Z-BUFFER ELEMENT SPACING ANALYSIS

There is no well-defined analytical means of bounding force estimation error introduced as a result of the discrete spacing distance of the Z-buffer elements, d_{XY}, in the geometric model. The value of d_{XY} has a definite affect on the force estimations, as it is the intersection of the cutting tool with the Z-buffer model that provides the contact area definition required for mechanistic modeling. However, the relationship between d_{XY} and the resultant geometric parameters can vary greatly with cutting conditions, and so it is very difficult to derive a realistic set of worst-case conditions (with respect to vector spacing effects) that are actually encountered during a milling operation. Therefore it is necessary to rely on theoretically possible worst-case conditions, yet these prove overly conservative for the majority of cutting conditions.

There is no drawback to being overly conservative from a pure accuracy standpoint, as a smaller d_{XY} will always provide more accurate results over a broader range of cutting conditions. The drawback is that the practicality of the force estimation model is greatly reduced, as a result of limitations in current computer hardware. The computational load incurred by force estimation increases O(n^2) with respect to reductions in d_{XY}, a result of the increased number of vectors in the model. However, this is not the primary concern due to the efficiency of the algorithms implemented. The primary drawback, as addressed in Chapter 4, is the available memory on the computer system for storing the vectors. If not carefully managed this can increase the computation time by magnitudes of order as a result of memory paging, or in the extreme can result in system failure. While a mesh spacing defined via a worst-case analysis exceeds the practical limits of current generation computers for large milling jobs, this will not always be the case. Therefore an example of 'worst-case' error limiting analysis for a given set of cutting conditions will be presented. Note that this is in contrast to the less rigorous mesh spacing selection methods presented in Chapter 4, which were found to work well within the current hardware constraints for typical jobs; these will also maximize computational efficiency regardless of the available hardware.
One means of calculating $d_{xy}$ is based on limiting error in the entrance and exit angles for a given set of cutting conditions. These angles are calculated using radial distance locations similar to a radial depth of cut, found by noting the radial depth where each Z-buffer element intersects the STE. The maximum and minimum radial distance values found from all intersections for the current move are treated as the absolute radial depths for the current tool move, and are used in the calculation of the entrance and exit angles (see Chapter 6).

For the generic condition of cutting along a principal direction in Workpiece space with no cutter tilt, the condition that results in the largest deviation between the actual and as-calculated radial distance value is when one side of the STE misses intersecting a row of Z-buffer elements by a small amount. This is shown occurring for the minimum radial distance on the right in Figure 4.19. This results in a radial distance value approximately equal to $(R - d_{xy})$. As explained in Chapter 6, the entrance and exit angles are found as a function of radial distance using a relation of the form:

$$\beta = \sin^{-1}\left( \frac{N}{R} \right) + \frac{\pi}{2}$$  \hspace{1cm} (C.1)

where $\beta$ is the entrance or exit angle (rad),

$N$ is the radial depth of cut (in), and

$R$ is the radius value (in).

In the case shown on the right in Figure C.1, $N = R - d_{xy}$. Note that the value of $d_{xy}$ may be defined as a fraction of the radius value, or:

$$d_{xy} = \kappa R$$ \hspace{1cm} (C.2)

where $\kappa$ is some value less than 1.

In this case the value of $N$ may be re-written as $N = R(1 - \kappa)$ (note the angle is disengaging in the $-N$ direction). Replacing $\kappa$ with $d_{xy}/R$, as $d_{xy}$ is eventually desired, allows Equation (C.1) to be re-written as a function of the mesh spacing $d_{xy}$ using:
\[ \beta = \sin^{-1}(d_{xy}/R - 1) + \pi/2 \]  

(C.3)

What is desired is a limitation in the error in the force calculations, not in the angle calculations, so this must now be related to force. Recall from Chapter 2 that for the case of a flat end cutter travelling in X_{wp}, Y_{wp}, the chip thickness value \( h \) may be defined as:

\[ h = f_i \sin(\beta) \]  

(2.9)

For a given set of cutting conditions, assume the value of the net force magnitude acting on the cutter is a linear function of chip thickness (i.e. the forces vary linearly with chip thickness over the surface of the cutter). This allows a function for the calculation of net force magnitude acting on the cutter as a linear function of chip thickness to be defined as:

\[ F_{net} = C f_i \sin(\beta) \]  

(C.4)

where \( C \) is some constant (\( = K_\gamma dZ (Kr^2+1)^{0.5} \) for Devor and Kline)

In the typical case of a four-fluted cutter, two cutting flutes an angular distance of \( \pi/2 \) would engage the stock at any given time during the full slot cut of Figure C.1. Therefore the total force contributions from both flutes could be described as:

\[ F_{net} = C f_i (\sin(\beta) + \sin(\beta + \pi/2)) \]

or simply

\[ F_{net} = C f_i (\sin(\beta) + \cos(\beta)) \]  

(C.5)

One means of limiting error in the simulation as a result of mesh size is to control the maximum change in force predicted by (C.5) as a result of one of the flutes exiting the contact patch as a result of the discrete
Figure C.1: Note that the portion of cutting tool indicated as not intersecting the discrete stock (highlighted in bold) generally increases with radial depth and is a function of mesh spacing $d_{xy}$. This leads to errors in the entrance/exit angles, indicated by shaded region.

$\hat{P}(e) = \frac{h_{int}}{h_{net}}$

1.29 ft

$e = 28\%$

$\beta(e) = 21.04^\circ$

$d_{xy} = 0.0666 R$

$(d_{xy} = 1/15 R)$

$e = 28\%$

$\beta(e) = 21.04^\circ$

$h(f_1) = 0.36 f_t$

$h(f_2) = 0.93 f_t$

$h_{net} = 1.29 f_t$

$e = h(f_1) / h_{net}$

Figure C.2: In the above image the Z-buffer elements are shown swept parallel to the direction of travel. Note that the chip thickness at flute $f_1$ is approximately 28\% of the sum of the chip thickness values for flutes $f_1$ and $f_2$ at the angle of disengagement, $\beta(e)$.
spacing of the Z-buffer elements. This is the case shown on the right in Figure C.1. Define the allowable
error $\varepsilon$ as a percentage representing the maximum change in force due to flute disengagement as a result of
discrete Z-buffer element spacing. This would mean that the mesh spacing would have to be defined such
that the disengaging flute was only contributing the ratio $\varepsilon$ of $F_{net}$ at the point of disengagement, while the
remaining flute contributes $1-\varepsilon$. Note that due to the definition of the flute-locating angle $\beta$, the cosine term
would still be the larger contributor that remains engaged while the sine term would disengage. Therefore,
this condition is defined by setting the sine term as the ratio of $\varepsilon$ to the sum of both terms:

$$\sin(\beta) = \varepsilon \left( \sin(\beta) + \cos(\beta) \right)$$  \hspace{1cm} (C.6)

and applying the appropriate trigonometric functions, and combining with Equation (C.3) yields:

$$\tan(\sin^{-1}(d_{XY}/R - 1) + \pi/2)^{-1} + 1 = \varepsilon^{-1}$$  \hspace{1cm} (C.7)

This may be solved for $d_{xy}$ as shown below:

$$\tan(\sin^{-1}(d_{XY}/R - 1) + \pi/2)^{-1} = \varepsilon^{-1} - 1$$
$$\tan(\sin^{-1}(d_{XY}/R - 1) + \pi/2)^{-1} = (\varepsilon^{-1} - 1)^{-1}$$
$$\tan(\sin^{-1}(d_{XY}/R - 1) + \pi/2) = \varepsilon / (1 - \varepsilon)$$

$$d_{XY} = (1 - \cos(\tan^{-1}(\varepsilon / (1 - \varepsilon)))) R$$  \hspace{1cm} (C.8)

Therefore, if a maximum allowable change in force due to 'artificial' flute disengagement were to be limited
to 5% ($\varepsilon = 0.05$), this would require a mesh spacing of $d_{xy} = 0.00138R$, and the disengagement would occur
at $\beta = 3.013^\circ$ (assuming the worst case conditions of the right hand image in Figure C.1). Examples on a
viewable scale are shown in Figure C.2; note that the mesh spacing shown is similar to that used in actual
simulations, namely $d_{xy}$ is $O(0.1R)$, as described in Chapter 4.
The mesh size solved for above is far too small to be practically implemented at the current time. However, this does not pose a problem. The sizing assumes a single, specific set of conditions, and these conditions may rarely if ever occur. Note, however, that variations on the method presented above may be used to analyze any specific set of conditions desired. In general the actual error encountered at any given rotational position is highly dependent on the current cutting conditions, which vary continually. More importantly, note that the model uses the maximum force value found during the cutter rotation as the representative force for that move. Except during very light cuts, this peak value generally occurs at some angular location much different from the location where the flute disengages the stock, and is much greater in magnitude than the force contribution calculated just prior to flute disengagement. Therefore the effects of this level of accuracy on the force estimate for the move would in most cases be negligible, while detracting from the goal of computational efficiency. Additionally, the peak force calculated is the result of multiple flute engagement, over a range of flute locating angles on each flute as a result of the helix angle, unlike the single rotational position for the flutes assumed above. Cutter runout also has a major effect on the force values. Therefore, while the ideal implementation would use the very fine mesh size indicated by some worst-case error limitation, accurate force results may still be obtained even if a much larger value of \( d_r \) is used. Provided that the mesh spacing is adequate to provide a reasonable estimate of the peak force experienced during the move, the subtler variations in force may be sacrificed for now to obtain the required efficiency. This is important as it allows for practical implementation of the force estimation and feedrate selection software using currently available computer hardware.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
APPENDIX D

PROTOCOL FOR MECHANISTIC CONSTANT DATA ACQUISITION

1. Preparation for Calibration Cuts
   1.1. Preparation of the Test Piece
       1.1.1. Clamp a material big enough in size to make several slot and straight cuts.
       1.1.2. Drill holes on the part to match a set of holes on the mid-plate. If the holes on the mid-plate are not suitable, drill and tap holes on the mid-plate to match the holes you drill on the test piece.
       1.1.3. Mill the top surface of the part using a 1" end-mill. This surface is going to be at the bottom during mounting so a good surface finish is essential.
       1.1.4. Turn the part over and counter sink the holes if the part thickness is too much for bolt length. Ideally at least 4-5 threads of the bolt should be holding on to the mid-plate.
       1.1.5. De-burr the edges of the part using a file. Also use fine sandpaper, clean the bottom surface until a shiny surface finish is obtained. Wipe the metal dust off and clean the part.
       1.1.6. The test piece is now ready to be mounted on the mid-plate.
   1.2. Preparation of the Load Cell - Test Piece Assembly
       1.2.1. Clean the machine: The cleanliness of the machine is very important. Even with a tiny chip between parts, the bolts will get loose during cutting and a lot of vibration will be observed in the force measurements. Vacuum the area where the load cell is to be mounted and use air hose to get rid of the metal chips. Use grease solvents in the area to get rid of the coolant residues. Dry the area thoroughly with the air hose.
       1.2.2. Mount the load cell: The load cell has to be mounted with the tapered mounting pieces and bolts with hexagonal wide-end nuts. The load cell has to be secure in both X and a Y direction even if cutting is only in one direction.
       1.2.3. Mount the mid-plate: Clean the load cell and the mid-plate with grease solvent and bolt the mid-plate on the load cell. Dry both parts before mounting using the air hose.
       1.2.4. Mount the test piece: Clean the test piece and dry it. Place it on the mid-plate, the milled surface of the test piece touching the mid-plate. Fasten it using the bolts. Use soft washers so that the bolts do not damage the test piece while tightening. Do not use hard washers as they might damage the cutting tool in case of an accidental contact.
       1.2.5. Clean the top surface of the test piece by milling it with a 1" face mill. This surface will be used as $Z=0$ by changing the fixture offset.
       1.2.6. The piece is ready to be tested on.
   1.3. Preparing the Tools to be Tested
       1.3.1. Load the tool(s) on the spindle.
       1.3.2. You may choose to set the tool offsets relative to the other pre-loaded tools or compensate for the difference in the fixture offset.
       1.3.3. Clamp the linear dial indicator.
       1.3.4. Lower the tool and touch it to the dial indicator at the end of the tools cylindrical section. Measure the run-out by rotating the tool with your hand. You may have to unfasten the tool on the chuck and fasten it again. If the run-out is close to 0.001", you can try hitting the tool using the hammer and putting a plastic piece between the tool and the hammer. Make sure the final run-out is less than or equal to 0.001".

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
1.4. Getting the Equipment Ready for Testing

1.4.1. Turn the power of the amplifier on and wait for a few minutes. Make sure the switches of the amplifier points to "remote" to enable the remote control unit. Keep the remote unit switch in the reset position and release the switch just before you start sampling to minimize the drift of the signal. Also check the amplifier scale and make sure it is at the desired level. Table D.1 shows the force limits for different settings.

<table>
<thead>
<tr>
<th>Newtons per Volt</th>
<th>100</th>
<th>200</th>
<th>500</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max. Force (lb)</td>
<td>224</td>
<td>449</td>
<td>1124</td>
<td>2248</td>
</tr>
</tbody>
</table>

Table D.1: Amplifier force limits for various Newton/Volt settings.

1.4.2. Check the load cell polarity. Run some tests by pushing on the load cell in X, Y and Z directions and checking the results for the sign. The forces should be opposite in sign to the direction you are pushing, thus representing the force on the tool, and not on the work piece.

1.4.3. Stir the coolant tank behind the machine to have a homogenous coolant. Make sure you wait 5 minutes after stirring to let the chips in the tank settle. Else the chips will clog the coolant lines. Then turn on the coolant and let it run for 5 minutes before starting any tests.

1.4.4. Turn the spindle on and let it run at the testing speed for a few minutes to heat up the machine.

2. File Convention for Saving the Calibration Data

2.1. Directory Structure
The calibration data files are saved in I: drive on the controller PC. The directory structure classifies the files according to the material, tool, and kind of cut (in several ways). The sub directories for a sample cut are shown below:
I:\Calibration Constants
  └ Stainless Steel (15-5 or 17-4)
    └ 0.25in ball-end
      └ Down Mill
        └ 0.25in axial try1
        └ 0.25in axial try2
        └ 0.5in axial try1

The same structure should be used for saving of calibration files to make the data easy to find in the future.

2.2. Filename Details
The file name consists of the details of the cut. For example:
6ipm 0.25axial 0.35radial 2000rpm date02.25.2000.out
This file name gives the feedrate, axial depth of cut, radial depth of cut (or slot cut), spindle speed, and the date the data was collected. The last "out" is used for file type association. This filename convention should be used to make the data easy to find in the future.

3. Choosing the Spindle Speed, Feedrates and Cut Length

3.1. The operating spindle speed and feedrates can be chosen from a known G-code file that has been used before or from an experimental table (i.e. Metcut).
3.2. Choose four feedrates for the tests. For example if the operating feedrate is limited to a maximum of 20ipm, choose 5, 10, 15, and 20. This will help find the constants that match our operating range.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
3.3. Find the minimum cut length required to get a steady state data for each feedrate using the following formula:

\[
\text{Cut Length} = 0.25" + \text{Feedrate (ips)} \times 15 \text{ sec}
\]

The 0.25" extra is for getting rid of the transient forces caused by the entry into the stock.

4. Making the Cuts

4.1. Positioning the cut

Position the tool at one corner of the cutting area so that after the first slot cut, the rest of the cuts will be down-mill. Figure D.1 shows a sample starting point for calibration cuts in the X-direction.

![Figure D.1: Calibration cut direction.](image)

The Y=0 position should be set at the start of the slot cut, pointed by the arrow. The radial steps should be taken downward (for cuts in Y direction, X=0 should be set at that point).

4.2. Cutting and Taking the data

The first cut will be a slot cut at one ball depth (or one radius depth for flat-end cutters).

4.2.1. Write a simple G-code in the program window of OpenCNC to move the cutter across the part, to be safe, at the lowest chosen feedrate value. It should look something like this: "G1 X7.0 F3.0". Make sure you will exit the workpiece at the other end. Position the tool as instructed in previous section and switch to run mode.

4.2.2. Run the Sampler program from the desktop of the controller PC. Check the amplifier settings to make sure they match the settings on the amplifier. The default is selected as 500 N/V. Set the sampling time to 15 seconds. Click "Start Sampling" button. Actual sampling doesn't start until the machine starts moving. The program will stop responding to user input while sampling. It will resume after 15 seconds of sampling.

4.2.3. Press the "Operate Reset" button on the load cell remote control and press "Cycle Start" button on the operator panel attached to the control case.

4.2.4. After the run stops, the data will be saved under the filename specified in the Sampler program (e:\test.out by default). You may have to rename the data to match the requirements defined under section 2. File Convention for Saving the Calibration Data.

4.2.5. You may choose to process the data and go to section 4.3. Using the Collected Data, or you may do the rest of the cuts by repeating the previous steps for the following settings:

- 0.7 x radius radial step-over, 1 x radius axial depth for the selected feedrate settings.
- 0.7 x radius radial step-over, 2 x radius axial depth for the selected feedrate settings.

4.3. Using the Collected Data

You may want to prepare a table to write the findings from this section; a sample is shown below:

<table>
<thead>
<tr>
<th>Feedrate (ipm)</th>
<th>Avg. X (lb.)</th>
<th>Avg. Y (lb.)</th>
<th>Mag Peak (lb.)</th>
<th>Mag Min (lb.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>70</td>
<td>170</td>
<td>250</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>200</td>
<td>300</td>
<td>130</td>
</tr>
</tbody>
</table>

Table D.2: Sample data for obtaining calibration constants.
4.3.1. Open the data file in Excel by selecting tab delimited text option. You may want to create an Excel macro for the following procedures, since they will be repeated for every cut.

4.3.2. Draw the graphs of X and Y force columns.

4.3.3. Create a new column showing the magnitude of the force total of X and Y forces. Also create a graph of this column. Find the average peak and minimum forces for this column.

4.3.4. Find the averages of the X and Y force columns throughout the steady state section of the data, in other words where the transient forces have damped out. A sample Y force graph and the steady state section are shown in Figure D.2. The transient region is selected large to be conservative.

![Y Force Lbs](image)

Figure D.2: Sample force plot showing transient and steady state regions.

4.3.5. After finding the values listed in Table D.2, you can calculate the mechanistic constants using a computer program developed for this purpose.
This appendix contains the instructions required for the operation of the software developed during this research. These instructions are broken into two parts for the two primary programs developed, MasterFeed and CalcMechConists. MasterFeed provides four options; relative velocity control, geometric simulation, force estimation, and automatic feedrate selection. Instructions for each are presented in the MasterFeed section. CalcMechConists performs only a single function, the calculation of the mechanistic material constants from empirical force data.

**Program I: MasterFeed.** Used for geometric simulation, force estimation, relative velocity control, and automatic feedrate selection.

This software requires two input data files for operation. One of these is the G-code file that contains the part program to be simulated. The other file is a user data file that contains the commands and values required for the operation of the software. This majority of this guide is dedicated to describing these command and values. The user data file must be created prior to the running of the program. If a mistake exists in the user data file, or an unknown command is contained in the G-code file, automatic error checking in the software will notify the user of the unrecognized information, and provide the file name and line number which it corresponds to. Note that the user data file contains the G-code file information, and so only the user data file name is explicitly required to initiate program execution.

The program may be executed in one of two manners, command line based or prompted data entry based. The command line based method is useful when the program is to be from a batch file, and also simplifies the task of software development as the command line data may be entered into MS VC++.

In the prompted data entry based method, the user enters

```bash
> /path/MasterFeed
```

in a DOS shell, where the '>' is NOT typed, it only indicates the DOS prompt, and 'path' indicates the directory path to the executable. Alternatively, the user may select the MasterFeed application icon in an MS windows shell. The user is then prompted for the required user data file name (and path).

In the command line based method, the user enters

```bash
> /path/MasterFeed -userdata USER_DATA_FILE_PATH/USER_DATA_FILE_NAME
```

in a DOS shell, or the in a windows shell the user clicks on the icon for a batch file that contains this command. In place of `USER_DATA_FILE_PATH` the path to the desired user data file, relative to the location of the MasterFeed application, should be entered. In place of `USER_DATA_FILE_NAME` the appropriate file name should be entered.

If the user data file cannot be opened using the specified path and filename, the user is notified of this error in the DOS shell, and program execution terminates. Otherwise the DOS shell displays the options selected by the user, and the values of the data entered, and program execution commences.
MasterFeed Software Options

There are 4 primary applications contained in the MasterFeed program; Kinemax, Render/Animate, WriteForce, and AutoFeed. Kinemax simulates the NC machine kinematics and controller behavior, and as a result it creates a copy of the input G-code file that contains variable feedrates that result in a constant relative velocity of the cutter past the stock when 5-axis milling. Render/Animate performs a geometric simulation of the milling process, and as a result provides graphical imaging of the milled stock; Render shows the final stock geometry, and Animate shows the in-process milling and the final stock. WriteForce performs a geometric and mechanistic simulation of the milling process for the purpose of force estimation. Based on the simulation results it creates a data file containing the peak estimated force value for each tool move and the G-code line number of that move, as well as the corresponding net machine runtime, volumetric removal rate, and maximum chip thickness values. AutoFeed performs a geometric simulation of the milling process, combined with an iterative mechanistic simulation, for the purpose of automatic feedrate selection. As a result of the simulation, a copy of the input G-code file that contains feedrates optimized for the unique cutting conditions of each tool move is created.

Input File Information

Any time a data file is required for input, or when specifying the path and filename for a Z-buffer model that is to be saved, this is performed using a string that contains the required path and the desired filename. This appears after the appropriate command as

FILE_PATH/FILE_NAME

In place of FILE_PATH the path to the desired user data file, relative to the location of the directory containing the MasterFeed application, should be entered. In place of FILE_NAME the desired file name and file extension should be entered.

Output Files: Copies of G-code Files containing Updated Feedrate Information

Both Kinemax and AutoFeed generate copies of the input G-code file, and these new files contain updated feedrate information. In both cases, the management of the new files is identical. The new G-code information is written to a file that has the exact same path and file name root as the input G-code, with the file extension changed to '.UPD' for UPDate'd. For example, if the input G-code file existed in a subdirectory of the MasterFeed Directory entitled "PartFiles", and the original filename was titled "fender.cnc", the input Gcode would be indicated by "PartFiles\fender.cnc". The newly generated copy of this file that contains the updated G-code information would then be "PartFiles\fender.UPD". If an existing version of "PartFiles\fender.UPD" exists, this is changed to "PartFiles\fender.bak" as a backup file in case of error. Only 1 backup is generated, which is overwritten by each subsequent backup. If "PartFiles\fender.UPD" is provided as the input file, then no new output file is created.

Output Files: Other Files Generated by the Software

This case covers file types other than updated G-code files that are created by the software, e.g. force data files, relative velocity data, etc. These files are written out in a format appropriate for reading into the Microsoft Excel spreadsheet where applicable. Except in cases where the user is required to specify a file path and filename for a desired output file, the file generated is given same path and filename as the input G-code file, with followed by the extension ".Dat" (note that the input G-code should not use this extension).

Software Option Interoperability

Note that both WriteForce and AutoFeed options require a geometric model. Because of this, the Render and Animate features may be used simultaneously with either of these two options. This is performed by including the appropriate rendering commands in the user data file along with the WriteForce or AutoFeed commands.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Note, however, that while both WriteForce and AutoFeed perform mechanistic modeling, they can not be used simultaneously. If estimated forces are desired for a newly created G-code file that contains optimized feed values, they must be calculated using a second, separate run of the software. This is because the feed inhibiting and in particular the buffering of G-code data upon stock entry, the feed values output are often reduced from those that were used in the final force calculation.

Data Input: Software Option User Data File Requirements
A description of the commands and data required for each of the software options will now be provided. As many of the commands and values are common to several options, a list containing the options and values required will be referenced as appropriate. This list is provided at the end of this section. All command are entered in the user data file using the following protocol:

**COMMAND VALUE**
where "COMMAND" is the command word entered into the user data file, and "VALUE" indicates a value that must accompany the command, if any.

These commands will be presented below in the following format:

**COMMAND_NAME (VALUE_TYPE)  text description of command**
where "COMMAND_NAME" is the command word entered into the user data file, "(VALUE_TYPE)" indicates the type value that must accompany the command (integer, real, character), if any, and "text description of command" provides a description of the command listing and value. The commands and command groups will be listed according to the software option for which they apply.

1) The KINEMAX option
Kinemax creates a copy of the input G-code file that contains variable feedrates that result in a constant relative velocity of the cutter past the stock when 5-axis milling. Kinemax is the simplest of the four applications, requiring only a G-code filename, cutter radius, and CNC machine Type; no stock model is necessary.

**Kinemax Specific Commands:**
- **KINEMAX** (no value) - calculate constant relative velocities for the input G-code file
- **VELOCITY** (real) - the desired relative velocity value, (IPM)
- **FEEDTOL** (real) - the allowable range about VELOCITY that calculated feeds are considered acceptable
- **SCALE** (real) - scale all feeds generated by the amount indicated; may be ran standalone also
- **MAXFEED** (real) - maximum allowable output feedrate
- **SLOTFEEDPERCENT** (real) - scale tool moves identified as slot cuts by the indicated amount
- **WRITEFEED** (no value) - indicates that a data file containing the input feedrates and corresponding axis velocities and net relative velocity is desired.

♦ Other Required Command Groups (see Command Group List):
A) CNC Machine Description

2) The ANIMATE / RENDER option
Render/Animate provides graphical imaging of the milled stock; Render shows the final stock geometry, and Animate shows the in-process milling and the final stock. All instructions for manipulating the stock model once rendered may be obtained by clicking the right mouse button to bring up a command menu; clicking on a desired command will execute it, and instructions for keyboard operation are provided.
• **Animate/Render Specific Commands:**
  
  **ANIMATE** (no value) - this indicates a dynamic rendering of the in-process milling is desired. A static rendering is provided when the animation is complete.
  
  **WIREFRAME** (no value) - indicates a wireframe representation of the cutter is desired during animation.
  
  **RENDER** (no value) - this indicates that only a static rendering of the part geometry that exists at the end of the current G-code file is desired.
  
  The below parameters adjust the view of the initial rendering that appears on the screen. The default view is straight down Z (the axis of the tool on a 3-axis vertical machining center), with X extending horizontally across the screen, and Y extending vertically. The below angular definitions then adjust relative to this, with signed values that adhere to the right hand rule.
  
  **X_ROT** (real) - the amount of rotation desired about the X axis (deg.)
  
  **Y_ROT** (real) - the amount of rotation desired about the Y axis (deg.)
  
  **Z_ROT** (real) - the amount of rotation desired about the Z axis (deg.)
  
• **Other Required Command Groups (see Command Group List):**
  
  A) CNC Machine Description
  
  B) Cutting Tool Definition
  
  C) Input G-code File Specification
  
  D) Stock Model Definition
  
  3) **The WRITEFORCE option**
  
  WriteForce creates a data file containing the force data for each tool move is desired. This information includes, for each move, the G-code line number, the corresponding net machine run time, the peak force for that move (XY Magnitude), the volumetric removal rate, and the maximum chip thickness value.
  
  • **WriteForce Specific Commands:**
    
    **WRITEFORCE** (no value) - this indicates that a force data file is desired
  
  • **Other Required Command Groups (see Command Group List):**
    
    A) CNC Machine Description
    
    B) Cutting Tool Definition
    
    C) Input G-code File Specification
    
    D) Stock Model Definition
    
    E) Mechanistic Model Definition
  
  4) **The AUTOFEED option**
  
  AutoFeed creates copy of the input G-code file that contains feedrates optimized for the unique cutting conditions of each tool move is created. The optimization takes place within a user defined set of constraints, which compose the majority of the commands that are listed below.
  
  • **WriteForce Specific Commands:**
    
    **AUTOFEED** (no value) - this indicates that the autofeed option is desired
    
    **FORCENET** (real) - the desired force magnitude (acting orthogonal to the cutter axis) (Lb.)
    
    **FORCETOL** (real) - the acceptable tolerance about the desired force value within which a calculated force value is deemed acceptable (Lb.)
    
    **MAXCHIPTHICKNESS** (real) - the maximum allowable chip thickness value (in.)
    
    **MAXFEED** (real) - the maximum allowable output feedrate while material removal is occurring (IPM)
MAXFEEDAIR (real) - the maximum allowable output feedrate while no material removal is occurring and the cutter is travelling through air (IPM)

SLOTFEEDPERCENT (real) - scale tool moves identified as slot cuts by the indicated amount

♦ Other Required Command Groups (see Command Group List):
  A) CNC Machine Description
  B) Cutting Tool Definition
  C) Input G-code File Specification
  D) Stock Model Definition
  E) Mechanistic Model Definition

Software Option User Data File Command Groups

The below command groups are common to several software options, and so are listed here to limit redundancy. Typically all commands in a group are required unless otherwise specified.

A) CNC Machine Description
   FADAL (no value) - indicates a Fadal VMC with a 2-axis rotary tilt table is to be modeled
   BD405 (no value) - indicates a Boston Digital type 405 machine is to be modeled
   BD505 (no value) - indicates a Boston Digital type 505 machine is to be modeled

   The below parameters apply only to Boston Digital NC machines.
   A_AXIS (real) - indicates an A axis configuration (rotation about +X)
   C_AXIS (real) - indicates a B axis configuration (rotation about +Z)

B) Cutting Tool Definition
   RADIUS (real) - the radius value of the current cutting tool(in.)
   FLUTED_LENGTH (real) - the length of the cutter that contains milling flutes. Note that this defines the axial length of cutter used in the discrete model (in.)
   TOOL_LENGTH (real) - the total length of the cutter, used primarily for rendering (in.)
   NUM_FLUTES (integer) - the number of flutes present on a cutter
   HELIX (real) - the helix angle on the cylindrical portion of the cutter(deg.)
   FLAT_END (no value) - indicates that a flat end cutter is to be modeled
   BALL_END (no value) - indicates that a ball end cutter is to be modeled

   The below parameters apply only to the cutter ball portion of ball end cutters. They apply to the variations in the cutting flutes that occur on this section of the cutter.
   BALL_HELIX_END (real) - the angle, measured from a plane extending through the ball center and orthogonal to the cutter axis (the cutter ball definition angle in chapter 3), where variation in the helix angle ends on the cutter ball.
   BALL_MIN_HELIX (real) - the minimum helix angle value attained on the cutter ball. The helix angle varies from the value on the cylindrical to this minimum value, which occurs at the BALL_HELIX_END location. Below this point the helix is constant at this minimum value.
   BALL_FLUTE_END (real) - the cutter ball definition angle at which 2 of the flutes on a 4 fluted cutter ball terminate (see chapter 3 for defn of this angle)

   The below variables are model cutter runout, and are only required when mechanistic modeling or calculating the empirically derived mechanistic material constants.
   RUNOUT (real) - the modeled amount of cutting tool runout (see chapter 2) (in.)
   RUNOUT_LOC_ANG (real) - the locating angle for the cutter runout (see chapter 2)(deg.)

C) Input G-code File Specification

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
GCODE (character) - indicates the path and file name of the desired input G-code file, relative to the home directory of the MasterFeed executable

D) Stock Model Definition

MESH (real) - the Z-buffer element spacing distance in Xwp, Ywp, (in.)
WIDTH (real) - the size of a rectangular stock model in the Xwp direction (in.)
DEPTH (real) - the size of a rectangular stock model in the Ywp direction (in.)
HEIGHT (real) - the size of a rectangular stock model in the Zwp direction (in.)
The below values shift the location of the stock material in workspace space (see Appendix B). The default position is the 'front, lower, left corner' of rectangular stock or only corner that defines a right handed coordinate system, with all angular offsets defaulting to zero. The angular offsets are applied to the axes present on the currently modeled machine, and so their effects will vary between machine types..
WP_OFFSET_X (real) - shifts the location of the stock material along Xwp (in.)
WP_OFFSET_Y (real) - shifts the location of the stock material along Ywp (in.)
WP_OFFSET_Z (real) - shifts the location of the stock material along Zwp (in.)
WP_OFFSET_A (real) - shifts the location of the stock material about the A rotary axis (deg.)
WP_OFFSET_B (real) - shifts the location of the stock material about the B rotary axis (deg.)
The below values are required only for FADAL simulations, or in any other case where no fixed machine coordinate system is used. They define the distance between workpiece coordinate origin and the machine coordinate origin (see Appendix B).
RT_OFFSET_X (real) - defines the relative location of the machine coordinate X axis
RT_OFFSET_Y (real) - defines the relative location of the machine coordinate Y axis
RT_OFFSET_Z (real) - defines the relative location of the machine coordinate Z axis

The below commands save a Z-buffer model to the hard disc, or load one from the hard disc.
SAVESTOCK (character) - saves the z-buffer model to the path and filename specified
LOADSTOCK (character) - loads the z-buffer model from the path and filename specified
NOTE: The default offsets that existed when the model was created are read in from the Z-buffer file. New offsets may be defined, but if so all offset values must be specified, not only those that changed.

The below parameters apply only to Boston Digital NC machines when milling turbomachinery with repeated blade geometry.
The raw stock for these machining operations is generally a lathe turning. These may be defined using a 'lathe profile' composite NURBS curve that defines the 2D profile of the turned part. Note the mesh size must be specified, but the size of the stock is determined automatically.
TURNSTOCK (character) - indicates a turned stock model is desired, and is followed by the path and name of the file containing the lathe profile description.
The default size of a turned stock model is 1/4 of the entire turning, or one radius in X and Y from the hub center. The below values are ratios that increase or decrease this amount depending on requirements. Ideally the minimum stock size that fully encompasses all material removal on one blade and one cavity is desired. A value > 1.0 increases the modeled size (e.g. 2.0 models a full diameter), and < 1.0 decreases it.
TURNING_RATIO_X (real) - scaling in Xwp of the default radius distance that is modeled
TURNING_RATIO_Y (real) - scaling in Ywp of the default radius distance that is modeled

When mechanistic modeling, two cavities must be milled so that the operations taking place on the blade surface are modeled correctly. Only one of the two is required.
BLADE_OFFSET (real) - angular distance between adjacent blades (deg.), =360/Num Blades
NUM_BLADES (integer) - number of uniformly spaced blades on the part (not incl. splitters)

E) Mechanistic Model Definition

DISC (real) - the desired axial disc thickness on a cylindrical cutter portion
BALL_KTC (real) - the KTC value for the ball portion of the cutter
BALL_KRC (real) - the KRC value for the ball portion of the cutter
BALL_P_ONE (real) - the Pt value for the ball portion of the cutter

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
**Program II: CalcMechConsts.** Used for the calculation of the mechanistic material constants.

This software performs only a single function, it calculates the mechanistic material constants required by the generalized discrete mechanistic model contained in MasterFeed and used during force estimation and optimized feedrate selection. This Software requires a single user data file for operation, which contains the cutting tool description, feeds and spindle speeds, and the cut geometry description. The data is listed in two segments. First common data that applies to all test cuts, i.e. spindle speed, cutter description and runout amounts, are entered. Then the number of force empirical force data sets is entered. Finally, the data sets are entered, this includes feedrate, cut geometry, and measured average force values. A unique number that identifies it precedes each data set; these are typically numbered sequentially from 1. All values other than forces are modal; they are held constant until a new value is entered, and so only values that change must be entered. For example, if the cutting conditions do not change for the first four data sets, then these conditions only need to be entered for the first data set.

In addition to the commands provided below, the commands for a cutting tool description, defined above in B) Cutting Tool Definition, are also required.

**Test Speed (real)** - the spindle speed used when making the test cuts (RPM)

**NumSets (integer)** - the total number of data sets to be read from the file

**Set (integer)** - indicates a new data set is to be read, set number is arbitrary

**MinRadial (real)** - minimum radial depth, measured from axis of rotation along \( Y_{cr} \) (in.)

**MaxRadial (real)** - maximum radial depth, measured from axis of rotation along \( Y_{cr} \) (in.)

**Axial (real)** - axial depth, measured from bottom of tool along axis (in.)

**TestFeed (real)** - feedrate value for current set of empirical force data (IPM)

**ForceX (real)** - the average X force for the current data sets test cut conditions (Lb.)

**ForceY (real)** - the average Y force for the current data sets test cut conditions (Lb.)

**ForceZ (real)** - the average Z force for the current data sets test cut conditions (Lb.)

*NOTE:* The Z-force is currently not used, may be neglected.
LIST OF REFERENCES


Tlusty, J., Author's lecture notes on "Structural Dynamics of Production Machinery" course offered at the University of Florida, Spring, 1990.


