

# **PHYSICS BASED TURNING PROCESS SIMULATION**

by

Ahmet Gencoglu

A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

(Mechanical Engineering)

The University of British Columbia

(Vancouver)

August 2011

© Ahmet Gencoglu, 2011

# Abstract

The manufacturing planning of parts is currently based on experience and physical test trials. The parts are modeled, and Numerically Controlled (NC) tool paths are generated in Computer Aided Manufacturing (CAM) environment. The NC programs are physically tested, and if the process faults are found, the NC program is re-generated in the CAM environment. The objective of this thesis is to develop Virtual Turning System that predicts the part machining process ahead of costly physical trials.

Tool–workpiece engagement geometry is calculated along the tool path by a proposed polycurve method. The part geometry is imported as a stereolithography (STL) model from the CAM system, and the cross section around the turning axis is reconstructed. The tool and part cross sections are modeled by polycurves, which are constructed by series of arcs and lines. The tool–part geometries are intersected using boolean operations to obtain the engagement conditions.

The turning process is modeled by predicting the chip area and equivalent chord angle. The process forces are modeled proportional to the material dependent cutting force coefficients, depth of cut and equivalent chord length that depends on the nose radius and approach angle of the tool. The chatter stability of the process is examined using Nyquist criterion at each tool–workpiece engagement station along the path.

The virtual turning simulation simulates the forces and detects the chatter stability, and adjusts the feeds at each tool-part engagement station. The physical turning of parts with arbitrary geometry can be simulated, and cutting conditions that leads to most optimal machining operation is automatically determined without violating the limits of the machine tool and part.

# Table of Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Table of Contents</b> . . . . .	<b>iii</b>
<b>List of Tables</b> . . . . .	<b>vii</b>
<b>List of Figures</b> . . . . .	<b>viii</b>
<b>Glossary</b> . . . . .	<b>xii</b>
<b>Acknowledgements</b> . . . . .	<b>xiv</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
<b>2 Literature Review</b> . . . . .	<b>4</b>
2.1 Virtual Turning . . . . .	4
2.1.1 Workpiece Representation . . . . .	6
2.1.2 Tool Representation Model . . . . .	10
2.1.3 Tool–Workpiece Engagements . . . . .	12
2.2 Force Model . . . . .	16
2.3 Chatter Stability . . . . .	17
2.4 Optimization Algorithms . . . . .	19
2.4.1 Direct and Random Search . . . . .	19
2.4.2 Dichotomous Search and Interval Halving Methods . . . . .	21

2.4.3	Genetic Algorithms . . . . .	22
<b>3</b>	<b>Contour Turning Process Model . . . . .</b>	<b>23</b>
3.1	Angle Definitions for Contour Turning . . . . .	23
3.2	Tool–Workpiece Engagements . . . . .	24
3.3	Cutting Force Model . . . . .	27
3.3.1	Cutting Coefficients . . . . .	28
3.3.2	Orthogonal to Oblique Transformation . . . . .	29
3.3.3	Chip Flow Angle . . . . .	31
3.3.4	Effective Chip Thickness . . . . .	32
3.4	Tool Motion Classification . . . . .	33
3.5	Stability with Process Damping . . . . .	36
3.5.1	Process Damping Gains . . . . .	40
3.6	Feedrate Scheduling . . . . .	41
3.6.1	Design Variables . . . . .	42
3.6.2	Design Space . . . . .	42
3.6.3	Constraints . . . . .	43
3.6.4	Direct Search Algorithm . . . . .	44
3.6.5	Engagement Based Scheduling . . . . .	47
3.7	Summary . . . . .	50
<b>4</b>	<b>Turning Simulation Model . . . . .</b>	<b>51</b>
4.1	Vericut Application Program Interface . . . . .	52
4.2	Polycurve Representation . . . . .	53
4.2.1	Entity Definitions . . . . .	53
4.3	Workpiece Model . . . . .	55
4.3.1	Generator Curve Representation . . . . .	55
4.3.2	Intersection of a Triangle with a Plane . . . . .	55

4.3.3	Generator Curve of a Turning Workpiece . . . . .	61
4.4	Tool Model . . . . .	63
4.4.1	Parametric Tool Modeling . . . . .	63
4.4.2	User Defined Tool Geometry . . . . .	66
4.5	Tool–Workpiece Engagement Model . . . . .	67
4.5.1	Boolean Based Tool–Workpiece Engagements . . . . .	67
4.5.2	Processing Engagement Profiles . . . . .	73
4.6	Parsing NC-Code . . . . .	77
4.6.1	Spindle Speed and Cutting Speed . . . . .	79
4.7	Simulation Schemes . . . . .	80
4.7.1	Constant Engagement . . . . .	80
4.7.2	Non Axi–symmetric Workpiece . . . . .	81
4.8	Summary . . . . .	82
<b>5</b>	<b>Application and Experimental Verification . . . . .</b>	<b>83</b>
5.1	Virtual Turning Simulation . . . . .	83
5.2	Contour Turning Experiments . . . . .	84
5.2.1	Experimental Setup . . . . .	84
5.2.2	Tool–Workpiece Engagements . . . . .	87
5.2.3	Cutting Forces . . . . .	88
5.2.4	Feedrate Scheduling . . . . .	88
5.3	Chatter Stability Experiments . . . . .	92
5.3.1	Experimental Setup . . . . .	92
5.3.2	Structural Flexibility . . . . .	93
5.3.3	Results . . . . .	94
5.4	Summary . . . . .	99
<b>6</b>	<b>Conclusion . . . . .</b>	<b>100</b>

<b>Bibliography</b> . . . . .	<b>102</b>
<b>Appendices</b> . . . . .	<b>105</b>
A.1 Intersection Algorithms . . . . .	105
A.1.1 Line - Line Intersection . . . . .	105
A.1.2 Circle–Circle Intersection . . . . .	107
A.1.3 Line - Arc Intersection . . . . .	109
A.1.4 Extension of Intersection Algorithms for Line Segments and Arcs . . . . .	111
A.2 AISI 1045 Cutting Coefficients . . . . .	113
A.2.1 Cutting Edge Coefficients . . . . .	116
A.2.2 Curve Fitting . . . . .	118
A.3 NC-Commands for Hardinge Superslant Lathe . . . . .	122

# List of Tables

Table 2.1	Insert shapes defined in ISO 13399 [1] . . . . .	10
Table 4.1	Relational operations with real and floating point numbers . . . . .	72
Table 5.1	Geometric specifications of the tools used in experiments . . . . .	86
Table 5.2	Computational time required by the engagement models . . . . .	88
Table 5.3	Suggested feedrate values for contour turning experiment . . . . .	90
Table 5.4	Radial turning chatter experiments . . . . .	92
Table A.1	Coefficients for fitted surfaces . . . . .	116
Table A.2	Cutting coefficients for AISI 1045 steel with GC3205 grade tool [2] . . . . .	116
Table A.3	Cutting coefficients for AISI 1045 steel with KC8050 grade tool [3] . . . . .	118
Table A.4	Coefficients for edge coefficient of AISI 1045 steel with GC3205 grade tool . .	120
Table A.5	Coefficients for edge coefficient of AISI 1045 steel with KC8050 grade tool . .	121
Table A.6	NC-commands implemented in the G-code parser . . . . .	122

# List of Figures

Figure 2.1	Predictive models of machining operations for practical applications [4] . . . . .	5
Figure 2.2	Virtual turning workflow [5] . . . . .	6
Figure 2.3	Feature primitive representation proposed by Shengfang et al. [6] . . . . .	7
Figure 2.4	Workpiece representation by iso-boundaries [7] . . . . .	8
Figure 2.5	Workpiece representation by a 2D section obtained from a solid modeler [8] . . . . .	8
Figure 2.6	Turning workpiece representation by the generator curve of a rotational part [9] . . . . .	9
Figure 2.7	Workpiece representation using Dixel elements [10] . . . . .	9
Figure 2.8	Analytical definition of the cutting tool geometry [5] . . . . .	11
Figure 2.9	Outline of a grooving tool . . . . .	11
Figure 2.10	Modeling of the cutting edge with B-splines [11] . . . . .	12
Figure 2.11	Uncut chip area configurations considered by Atabey et al. [12] . . . . .	13
Figure 2.12	Chip geometry produced by a tool having a nose radius and approach angle during cylindrical turning [2] . . . . .	14
Figure 2.13	Chip geometry in contour turning [13] . . . . .	15
Figure 2.14	Feature decomposition method proposed by Zhou [5] . . . . .	16
Figure 2.15	Calculation of cutting forces with discrete orthogonal elements [14] . . . . .	17
Figure 2.16	Approximate chord for a cutter with a straight edge [15] . . . . .	18
Figure 2.17	Experimental verification of Colwell's chord approximation [16] . . . . .	19
Figure 2.18	Discretization of chip load by trapezoidal elements [17] . . . . .	20
Figure 2.19	Dichotomous searching method [18] . . . . .	21
Figure 2.20	Interval halving method [18] . . . . .	22

Figure 3.1	Contour turning process . . . . .	24
Figure 3.2	Tool–motion and surface angles in contour turning . . . . .	25
Figure 3.3	Tool–workpiece engagements . . . . .	25
Figure 3.4	Comparison of engagement maps in milling and turning . . . . .	26
Figure 3.5	Geometry of an oblique turning tool [14] . . . . .	29
Figure 3.6	Cutting forces during oblique cutting . . . . .	32
Figure 3.7	Changing engagement geometry along the tool path in contour turning . . . . .	33
Figure 3.8	Tool motion classification . . . . .	34
Figure 3.9	Force model modification for left and right handed cuts . . . . .	35
Figure 3.10	Chip thickness in contour turning . . . . .	37
Figure 3.11	Direct search algorithm . . . . .	46
Figure 3.12	Engagement based feedrate scheduling using search based algorithm . . . . .	47
Figure 3.13	Modification of tool–workpiece engagement geometry . . . . .	48
Figure 4.1	Polycurve entity definitions . . . . .	53
Figure 4.2	Generator curve for a solid part of revolution [9] . . . . .	56
Figure 4.3	Signed distance of a point from a plane in space . . . . .	56
Figure 4.4	Possible cases of triangle–plane intersection . . . . .	57
Figure 4.5	Euler angles [19] . . . . .	62
Figure 4.6	Parameters of a rhombic insert . . . . .	64
Figure 4.7	Definition points for rhombic inserts . . . . .	64
Figure 4.8	Geometric definition of a rhombic tool . . . . .	66
Figure 4.9	Use of swept polygon for boolean operation [9] . . . . .	67
Figure 4.10	Entity classification for boolean algorithm . . . . .	68
Figure 4.11	In-region test with 250 test points . . . . .	70
Figure 4.12	Errors in in-region test due to test–ray crossing vertices . . . . .	70
Figure 4.13	Boolean operations supporting holes in the workpiece model . . . . .	72
Figure 4.14	Positive and negative direction conventions for Green’s Theorem . . . . .	74

Figure 4.15	Area of the arc entity . . . . .	74
Figure 4.16	Area calculation by Green’s theorem . . . . .	74
Figure 4.17	Entities originating from tool curve are in cut . . . . .	76
Figure 4.18	Direction convention for the effective chord vector . . . . .	77
Figure 4.19	Flow-chart for NC-code parser algorithm . . . . .	78
Figure 4.20	Simulation scheme for constant engagement . . . . .	81
Figure 4.21	Multiple simulation planes per spindle revolution . . . . .	82
Figure 5.1	Structure of the virtual turning application . . . . .	85
Figure 5.2	Initial workpiece geometry for contour turning experiments . . . . .	86
Figure 5.3	Comparison of engagement model with Vericut Solution . . . . .	89
Figure 5.4	Cutting forces during contour turning . . . . .	90
Figure 5.5	Comparison of cutting forces before and after optimization . . . . .	91
Figure 5.6	Direct FRF of the workpiece with different overhang values. . . . .	93
Figure 5.7	Radial turning experiment #1 . . . . .	95
Figure 5.8	Radial turning experiment #2 . . . . .	96
Figure 5.9	Radial turning experiment #3 . . . . .	97
Figure 5.10	Radial turning experiment #4 . . . . .	98
Figure A.1	Line-Line intersection . . . . .	105
Figure A.2	Circle-Circle intersection . . . . .	107
Figure A.3	Line-Circle intersection . . . . .	109
Figure A.4	Quadratic surface fit for shear stress (AISI 1045 Steel) . . . . .	113
Figure A.5	Quadratic surface fit for friction angle (AISI 1045 Steel) . . . . .	114
Figure A.6	Quadratic surface fit for shear angle (AISI 1045 Steel) . . . . .	115
Figure A.7	Comparison of cutting edge coefficients for AISI 1045 steel . . . . .	117
Figure A.8	Curve fitting for cutting edge coefficients (GC3205 grade tool and AISI 1045 Steel) . . . . .	119

Figure A.9 Curve fitting for cutting edge coefficients (KC8050 grade tool and AISI 1045  
Steel) . . . . . 120

# Glossary

$A_c$  Uncut chip area in [ $mm^2$ ]

$a_c$  Depth of cut in [mm]

$\alpha_n$  Normal rake angle

$\alpha_o$  Equivalent orthogonal angle

$\beta_a$  Average friction angle in orthogonal cutting

$\Delta a_c$  Difference in the depth of cut parameter

$\Delta f_c$  Difference in the feedrate parameter

$f_c$  Tool movement during one complete spindle revolution [mm/rev]

$h$  Height of the approximate cutting chord in [mm]

$h_c$  Chip thickness

$h_{\text{eff}}$  Effective chip thickness in [mm]

$i$  Equivalent oblique angle

$\kappa_r$  Approach angle of the cutting tool

$L_c$  Total cutting edge Length

$l$  Tool edge length for a Rhombic Tool

$L_{eff}$  Length of the approximate cutting chord

$\epsilon_n$  Nose angle for a Rhombic Tool

$\phi_c$  Shear angle in orthogonal cutting

$\psi_r$  Side cutting angle of the cutting tool

$P$  Power required during machining in [W]

$r_n$  Nose radius of the tool

$n$  Spindle speed in rpm

$\psi_s$  Local surface angle of the in-process workpiece

$\tau_s$  Shear yield stress in orthogonal cutting

$\theta$  Approximate chord angle

$T$  Torque required during machining in [Nm]

$V_c$  Cutting speed or Surface speed in [m/min]

$w$  Width of the approximate cutting chord in [mm]

# Acknowledgements

First of all, I would like to express my lifelong gratitude to my supervisor Prof. Dr. Yusuf Altintas for allowing me to be a member of the University of British Columbia and Manufacturing Automation Laboratories. It was my privilege to conduct this research under his supervision and I am thankful for his wisdom, guidance and knowledge which he shared with me during the course of my studies.

I would also like to thank Prof. Dr. Steve Feng, for helping me with my research and for being a source of motivation.

The two years I spent in Vancouver has been a wonderful time thanks to my colleagues and friends. It has been both my honor and pleasure to be working in such a professional and yet fun atmosphere. I would like to thank members of the MAL for their support and friendship.

Finally, I would like to thank my brother Aytuğ, my father Aykut, my mother Melek and my fiancée Eda for their everlasting love and support. I would not be able to succeed without them. Therefore I want to dedicate this work to my family.

# 1 Introduction

The present manufacturing industry requires rapid and cost effective design and manufacturing methods to produce new products in small batches. The traditional manufacturing, which requires process planning, physical trial and re-design of the process in few iterations, is no longer suitable to produce parts cost effectively in short periods. Recently, the design and manufacturing operations have been replaced by their digital simulation, i.e. virtual models, in industry.

Virtual manufacturing is classified under two categories: Geometric and physics based simulations. Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) researchers developed graphics based simulation model of the machines and workpiece geometry. The collision of the tool, fixtures and machine tool can be simulated along the tool path graphically, and geometric errors can be corrected ahead of physical machining. The physics based simulation systems need to predict the forces, vibrations, and dynamic behavior of the machine tool and metal cutting process along the tool path. Manufacturing Automation Laboratory at the University of British Columbia has been one of the pioneering research centers which contributed significantly to the virtual simulation of milling processes. This thesis introduces the first Virtual Turning Process in the same laboratory using a new tool-part intersection model with computationally efficient turning mechanics and dynamics models.

Virtual turning has the following modules that have to be computationally efficient with a geometric accuracy within ten micrometers:

- Tool-part engagement computation
- Process mechanics model

- Chatter stability detection
- Feedrate scheduling

Each module has to be executed at each tool–part engagement station which can reach to over 100,000 on a typical aircraft engine disk.

Tool–workpiece engagement is identified by representing the symmetric – cylindrical geometric features by series of arcs and lines. Tool and part features are intersected using Boolean operations. The new, two dimensional engagement method is shown to be nine fold faster than commonly used Z–buffer methods based on solid geometry. Geometric accuracy is also improved by analytical calculation of the tool–workpiece engagements.

The turning process is commonly modeled by digitizing the curved cutting edge into small, differential elements, i.e. typically 20-30 segments. The chip thickness and the corresponding cutting force, torque and power are predicted for each differential element, and digitally summed to evaluate the total forces at each tool–workpiece engagement station which can be over 100,000. The traditional mechanics model, which requires 20-30 fold increase in computational time due to digital integration, is therefore not suitable to be used in virtual turning. An equivalent cutting edge concept, which was previously introduced by Colwell [15] in 1950s, is used in rapidly evaluating the cutting forces, torque and power in this thesis. The method is proven to have sufficient accuracy for virtual turning operations. The chatter is detected by applying computationally efficient Nyquist criterion to the dynamics of the turning process at each station.

Turning of parts with arbitrary geometry can be simulated, and cutting forces, torque and power can be predicted along the tool path by the proposed Virtual Turning System. Also, the cycle times can be reduced by searching feed and spindle speed values that do not violate the physical limits of the machine such as torque and power.

Henceforth, the thesis is organized as follows:

The literature related to the cutting mechanics and virtual turning is presented in Chapter 2. The process mechanics, chatter detection and feedrate scheduling models are given in Chapter 3 for contour turning operations. A new tool–part intersection model and its integration to the proposed

Virtual Turning system are presented in Chapter 4. The application and experimental verification of Virtual Turning system is presented in Chapter 5. The thesis is concluded in Chapter 6.

## 2 Literature Review

### 2.1 Virtual Turning

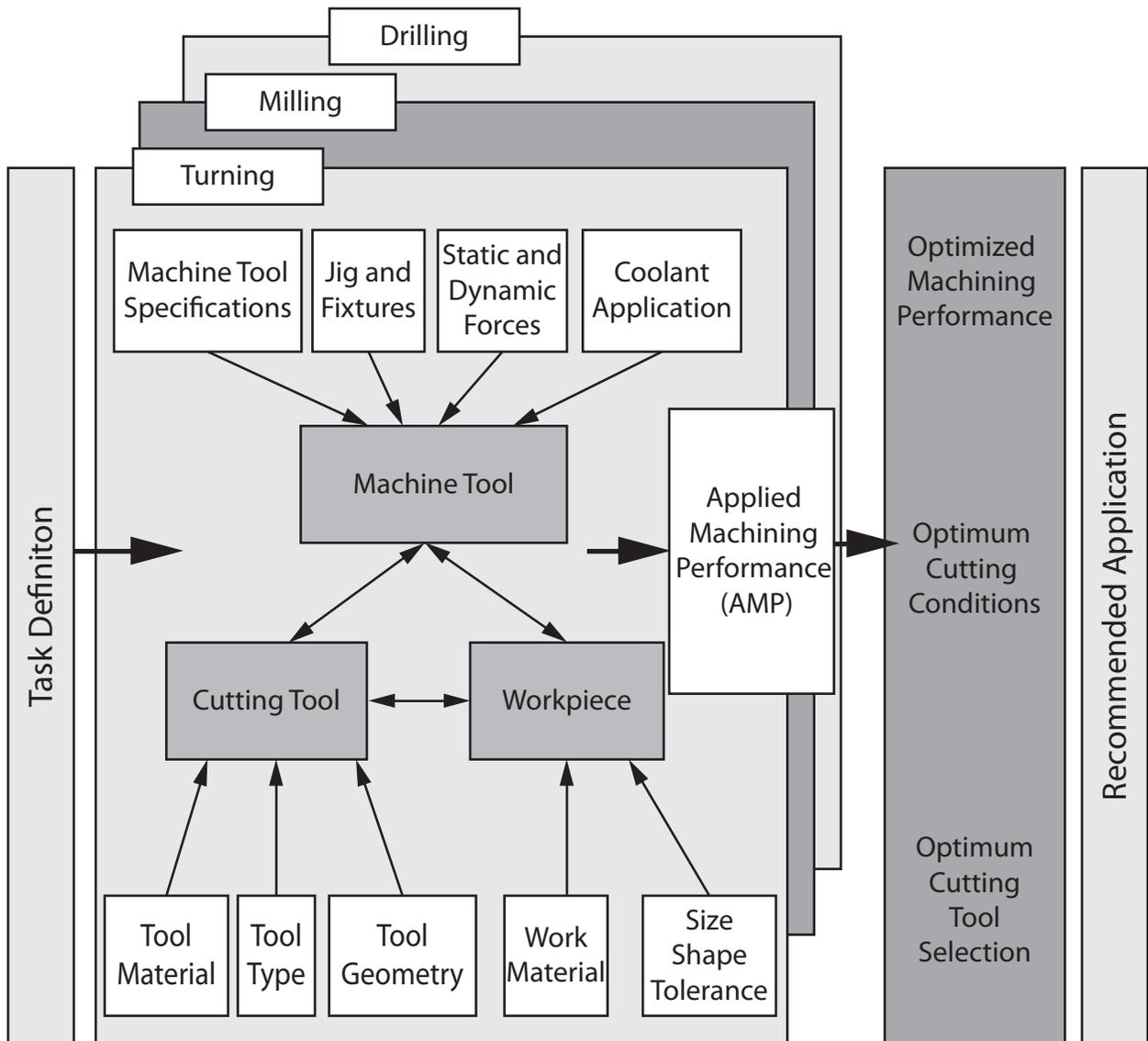
An accurate process model has to be developed to simulate the machining performance of a specified machining process. The machining performance can be evaluated either in technical or commercial aspects. Technical performance represents the physical outputs of the process such as cutting forces, power, torque and machine tool vibrations. Commercial performance is related to the operational costs, machining costs and productivity. The progress report of the CIRP working group ‘Modeling of Machine Operations’ lists the main motivations for modeling metal cutting operations as follows [4]:

- Design of Processes
- Optimization of Processes
- Control of Processes
- Simulation of Processes
- Design of Equipment

As Figure 2.1 demonstrates, the simulation of a machining process requires the definition of three main components:

- Cutting Tool
- Workpiece

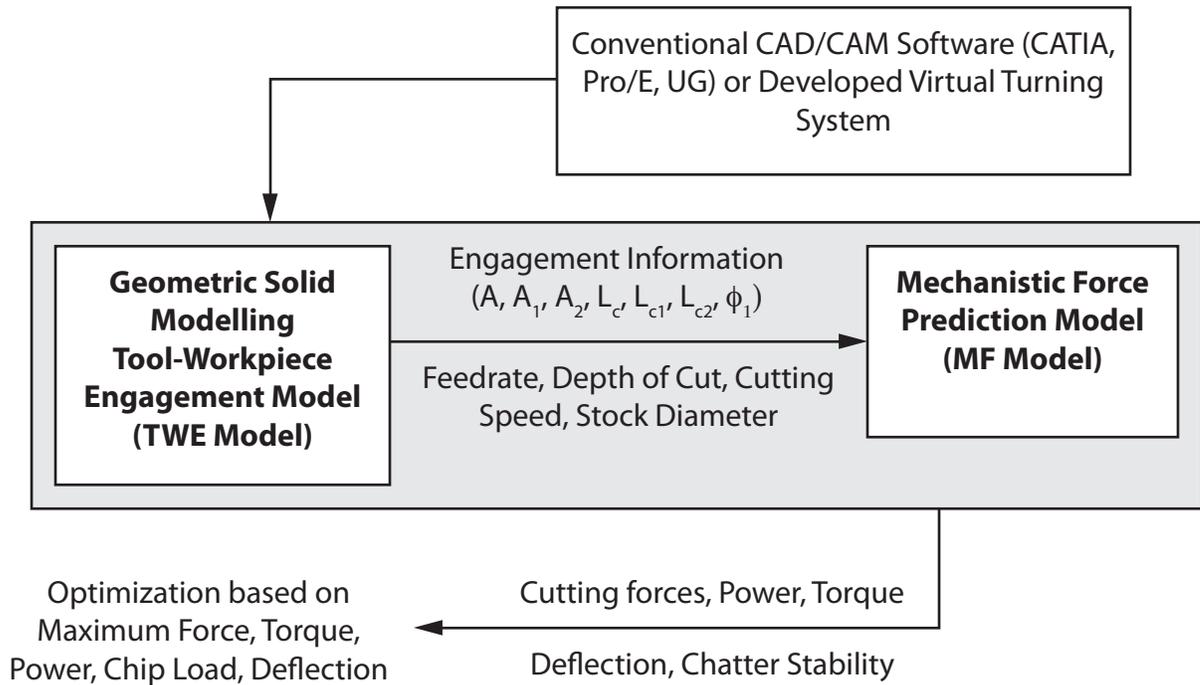
- Machine Tool



**Figure 2.1:** Predictive models of machining operations for practical applications [4]

For turning operations, a virtual machining simulation was proposed by Zhou [5]. The basic user input for this virtual machining system is CAD/CAM data, which can be obtained from available software packages. Since most processes are designed using one of these CAD/CAM environments, this is the most convenient choice of input data for the manufacturing industry. The workflow proposed in Figure 2.2 describes the tasks of the virtual turning system as follows:

- Geometric modeling of the Tool–Workpiece engagements (TWE Model)
- Force prediction model
- Optimization



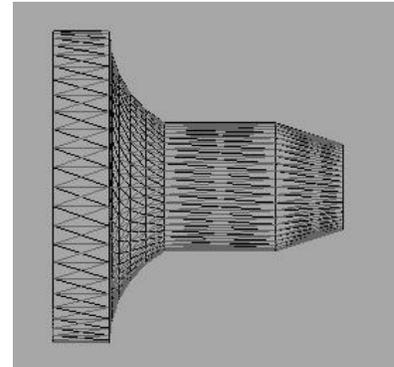
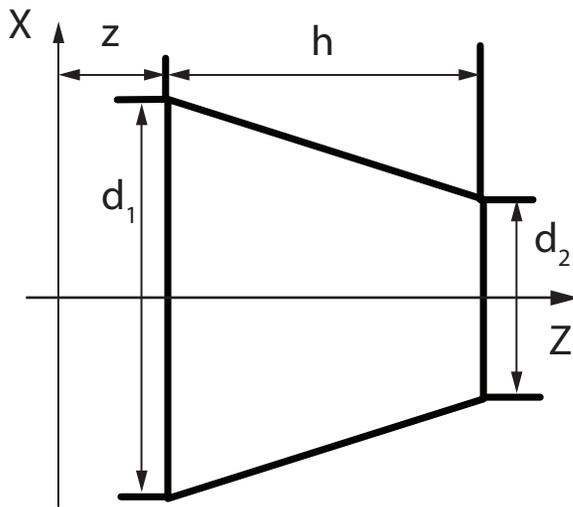
**Figure 2.2:** Virtual turning workflow [5]

The tool–workpiece engagements have to be resolved in the initial step of the virtual turning. The second step involves computation of the cutting forces, power and torque requirements for the given engagements. The final step evaluates errors, surface quality, stability of the cut, and optimizes the operation.

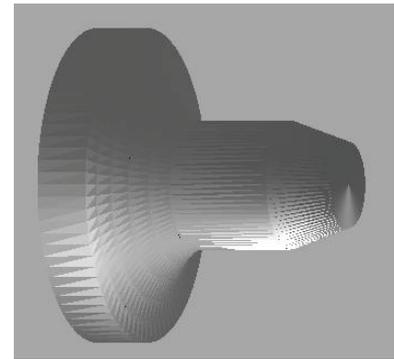
### 2.1.1 Workpiece Representation

A workpiece representation should have the following characteristics:

- Displayed on the screen efficiently
- Simple creation, manipulation and modification



a) Wireframe Model



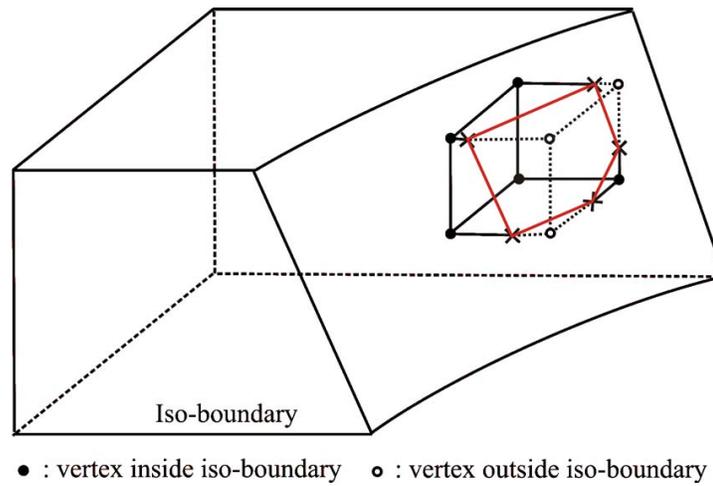
b) Solid Model

**Figure 2.3:** Feature primitive representation proposed by Shengfang et al. [6]

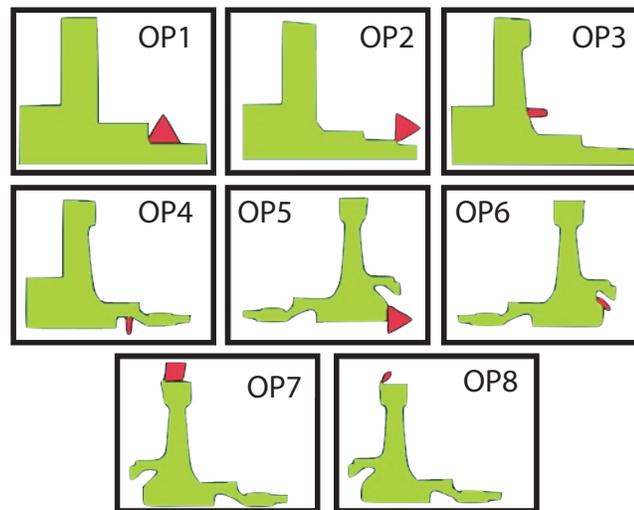
- Easy to implement and low computational load
- Memory efficient

Some studies have defined the workpiece as a solid model, using *Constructive Solid Geometry* [20], *Boundary representation*, as shown in Figure 2.4 [7], or *Solid primitives*, as shown in Figure 2.3 [6].

Solid model representations are relatively complex and have large memory requirements. These representations are necessary when dealing with milling operations where the cutting process is described with the tool motion in three or more axes. However, turning operations are axi-symmetric,



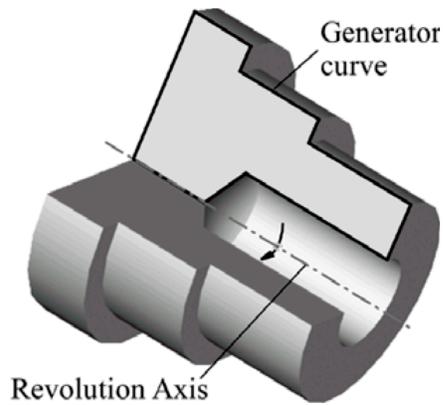
**Figure 2.4:** Workpiece representation by iso-boundaries [7]



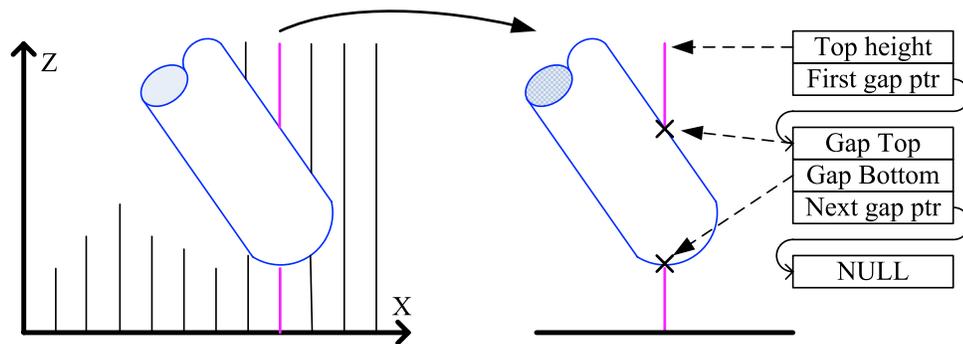
**Figure 2.5:** Workpiece representation by a 2D section obtained from a solid modeler [8]

therefore, it is possible to reduce the turning geometry into a two dimensional problem.

Model developed by Zhou et al. [8] proposed to use the cross sectional area of a given solid model to represent the workpiece as seen in Figure 2.5. Zhou used the ACIS solid libraries to manipulate the workpiece model. Because her approach still depends on the capabilities of the solid modeler, it cannot take full advantage of a two-dimensional solution.



**Figure 2.6:** Turning workpiece representation by the generator curve of a rotational part [9]



**Figure 2.7:** Workpiece representation using Dixel elements [10]

Li et al. [9] showed that the workpiece in a turning operation can be simply defined as the generator curve of a rotating geometry as demonstrated in Figure 2.6. The curve can be represented by a polygon, an analytical curve including non-linear segments, or by non-linear parametric curves such as splines.

A two dimensional workpiece can also be represented by dexels, which are usually used in milling simulations as the z-buffer method as described by Zhang et al. [10], see Figure 2.7. Dixel method can be reduced to represent two dimensional geometry by using a one-dimensional grid in-

stead of the two. One dimensional model can reduce the computational time spent by the intersection and modification algorithms. However, dixel based method can only represent the geometry in a discrete manner, and suffers from visualization difficulties.

## 2.1.2 Tool Representation Model

**Table 2.1:** Insert shapes defined in ISO 13399 [1]

Symbol Shape	Insert	Shape	Nose Angle
S		Square	90
T		Triangular	60
C		Rhombic (Diamond)	80
D			55
E			75
F			50
M			86
V			35
W			
H		Hexagonal	120
O		Octagonal	135
P		Pentagonal	108
L		Rectangular	90
A		Parallelogram	85
B			82
N/K			55
R		Round	-

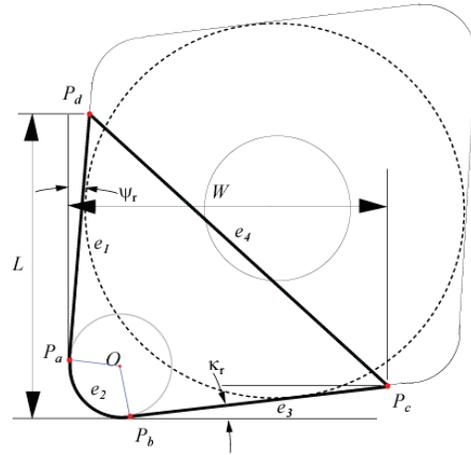
It is common practice in the industry to use multiple turning tools within a single turning operation, therefore, a versatile virtual turning system should be able to simulate the turning process with a wide variety of geometrically defined cutting tools.

Kaymakci [1] has investigated the geometry of standard cutters and inserts defined by ISO standards. Typical insert shapes mentioned in ISO 13399 standard are illustrated in Table 2.1.

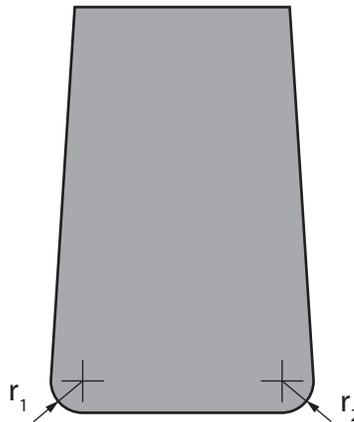
The tool geometry can be represented by the outline of the cutting edge as used by Zhou [5], see Figure 2.8. This study represents the tool profile as a series of lines and arcs which have been analytically defined as functions of typical tool design parameters such as tool nose radius, side and end cutting edge angles. However, this method is not generalized enough to support different

$r_c$  Tool nose radius  
 $\psi_r$  Side cutting edge angle  
 $\kappa_r$  End cutting edge angle  
 $L$  Tool height  
 $W$  Tool width

1. Tool center position  
 $O(X_0, Y_0)$
2. Tool nose arc edge  $e_2$  upper tangent point  $P_a$   
 $X_a = X_0 - r_c \cos(\psi_r)$   
 $Y_a = Y_0 + r_c \sin(\psi_r)$
3. Tool nose arc edge  $e_2$  lower tangent point  $P_b$   
 $X_b = X_0 + r_c \sin(\kappa_r + \sin(\psi_r)) \tan(\psi_r)$   
 $Y_d = Y_0 + L - r_c$
5. Tool straight edge  $e_3$  upper right point  $P$   
 $u = [W - r_c(1 + \sin(\kappa_r))] / \cos(\psi_r + \kappa_r)$   
 $X_e = u \cos(\kappa_r) + X_b$   
 $Y_e = u \sin(\kappa_r) + Y_b$



**Figure 2.8:** Analytical definition of the cutting tool geometry [5]



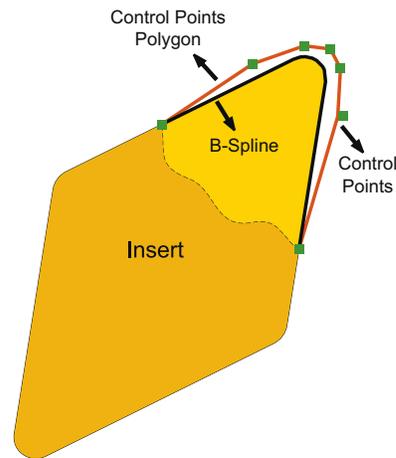
**Figure 2.9:** Outline of a grooving tool

tool geometries such as rectangular grooving tools where the cutting edge must be represented by two circular regions, and the interconnecting cutting edge, as shown in Figure 2.9.

The cutting tool was defined as a polygon by Li et al. [9], which has the disadvantage of not being able to define non-linear curves, but only straight lines. Polygons increase the memory usage when a large number of small line segments are used to approximate circular regions of the tool.

Another possibility for defining the tool geometry is to use parametric non-linear curves.

Yussefian et al. [11] proposed to use B-spline curves to represent the turning tool geometry. B-spline curves are defined by their control points, which are arranged to form a *Control Points Polygon* as shown in Figure 2.10. A clipping algorithm is employed to find the intersections of the tool and the workpiece in order to define the chip boundaries that are used to calculate the resulting cutting forces.



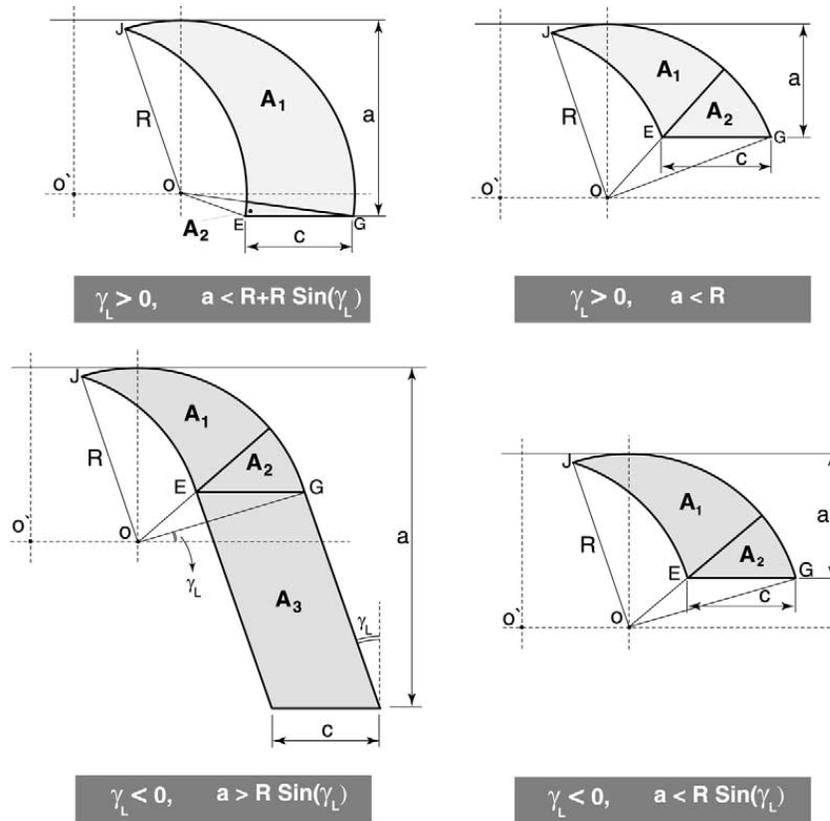
**Figure 2.10:** Modeling of the cutting edge with B-splines [11]

The cutting tool also can be represented volumetrically. It is the common approach in milling operations to define the cutting tool as a solid model, and generate the volume swept by the tool as it moves along the NC path [21][7].

### 2.1.3 Tool–Workpiece Engagements

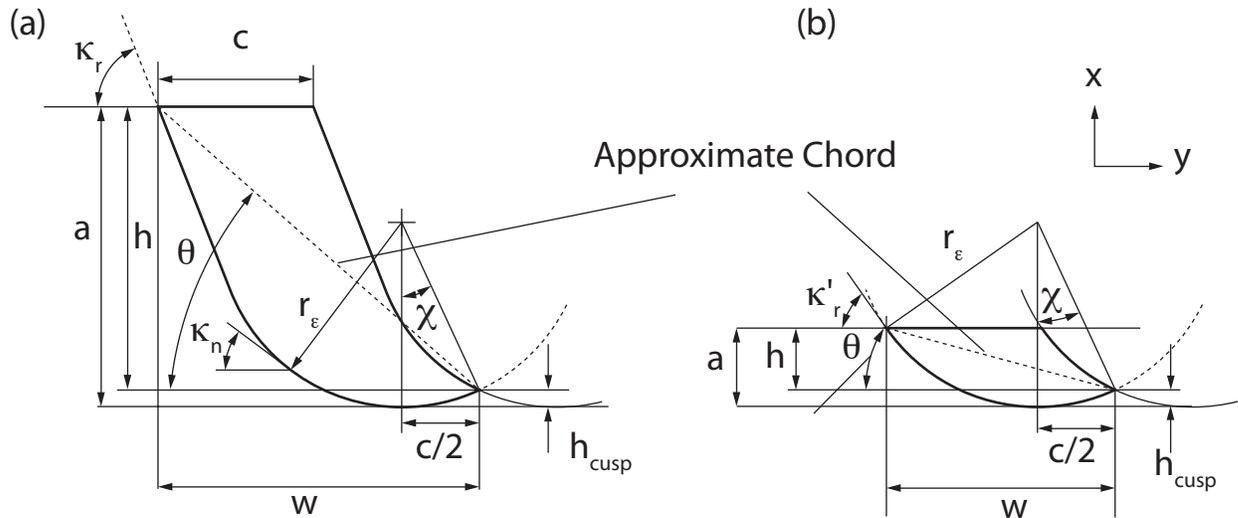
Unlike milling operations, where cutter workpiece engagements are three dimensional problems, turning engagements can be reduced to a two dimensional problem due to the axial symmetry of the process. Various studies aim to calculate the tool-workpiece engagements of the turning operation with a tool having a nose radius ( $r_n$ ) and side cutting angle ( $\psi_r$ ).

Atabey et al. [12] proposed to divide the uncut chip area into geometrical regions. Assuming a feedrate that is smaller than the nose radius of the tool, they reduced the number of possible engagement conditions to five, and used discrete elements to calculate the uncut chip area formed by the nose radius. Four of the five proposed configurations are shown in Figure 2.11. However, due to discretization of the chip load this method is not computationally favorable. Also, the configurations are defined only for straight cylindrical cuts, therefore, the solutions are not valid for contour turning, grooving or other turning operations.



**Figure 2.11:** Uncut chip area configurations considered by Atabey et al. [12]

Eynian and Altintas [2] developed a geometrical model of the chip load to predict the process. They considered two configurations, one where the cut is deeper than the nose radius, and another for the smaller cuts, as shown in Figure 2.12.



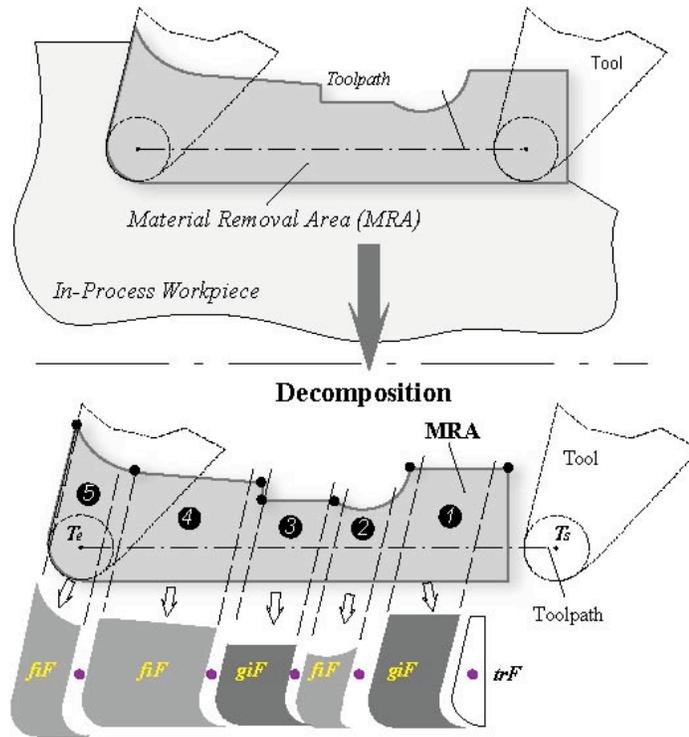
**Figure 2.12:** Chip geometry produced by a tool having a nose radius and approach angle during cylindrical turning [2]

Reddy et al. [13] developed geometrical models of the chip load for contour turning. Similar to Atabey et al. [12], they divided the chip load into zones as shown in Figure 2.13. However, this technique also relies on integration, which could be computationally costly and does not cover all possible configurations (such as the case where the depth of cut is smaller than the tool nose radius arc).

Zhou [5] used the ACIS kernel to define tool engagement by defining various engagement features, that are extracted from the workpiece by boolean operations. The analytically defined tool contour is swept along the tool path. Area swept by the tool, namely the tool swept area, is then intersected with the workpiece. Extracted engagements are decomposed by the proposed feature based method as illustrated in Figure 2.14. The feature decomposition is computationally complex and the solid modeler kernel brings additional computational load.

Some research has defined the tool workpiece engagement parameters (uncut chip area and cutting edge length) analytically, without any geometrical simulations. Ozdoganlar and Endres [22] developed analytical formulations where the motion in both radial and axial directions are considered. However, this approach does not account for the workpiece geometry, and does not present any possibility to modify and update the in-process workpiece.



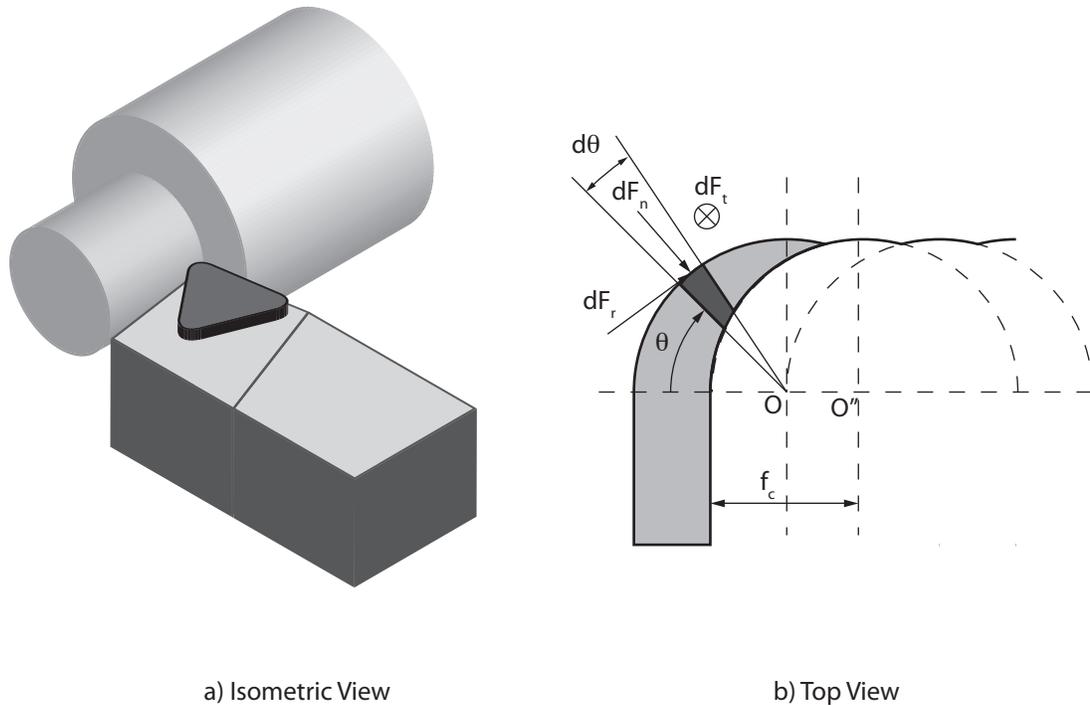


**Figure 2.14:** Feature decomposition method proposed by Zhou [5]

## 2.2 Force Model

The most prominent force models assume that the cutting forces are proportional to the uncut chip area and the cutting edge length. The coefficients that relate the chip load to the cutting forces are known as cutting coefficients [14]. There are several methods to evaluate the cutting coefficients, such as orthogonal to oblique transformation or mechanistic approach.

In reality the cutting tools are not always ideal and every cutting tool has a nose radius ( $r_n$ ), which thins the chip and changes the force direction. The cutting forces can be calculated by dividing the chip load into small discrete elements, which are assumed to cut the workpiece orthogonally. Forces are then calculated in the tangential, radial and feed directions as shown in Figure 2.15. These forces are transformed into measurement coordinates by using the local approach angle of each discrete element [14]. Yusefian et al. [11] proposed to use a discretization technique similar to Altintas [14], in order to obtain an effective chip flow angle for the boring process.



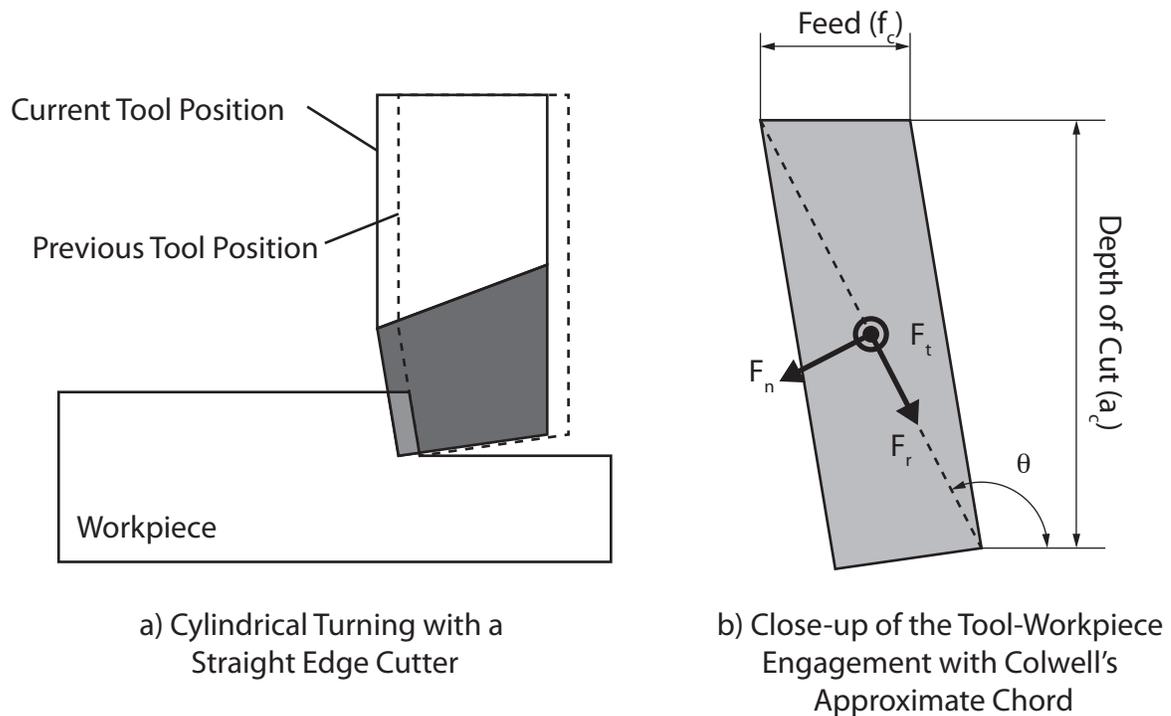
**Figure 2.15:** Calculation of cutting forces with discrete orthogonal elements [14]

Colwell [15] proposed the use of an equivalent chord, to which the chip flow is assumed to be perpendicular. The approximate chord is shown in Figure 2.16 for a straight edge tool, and in Figure 2.12 for a tool with a nose radius.

Eynian and Altintas [2] used a force model, which utilizes the approximate chord proposed by Colwell [15]. The approximate chord is calculated using the tool geometry and the process parameters such as depth of cut and feedrate as shown in Figure 2.12. The cutting forces are assumed to be equivalent to the cut where the approximate chord acts as the cutting edge. This approximation eliminates the necessity of using discretized elements and is proven to be sufficiently accurate by Eynian and Altintas [2] (see Figure 2.17).

## 2.3 Chatter Stability

Prediction of the regenerative chatter stability is one of the most important challenges in machining operations. Chatter instability in machining causes poor surface finish, tool wear, and damage to



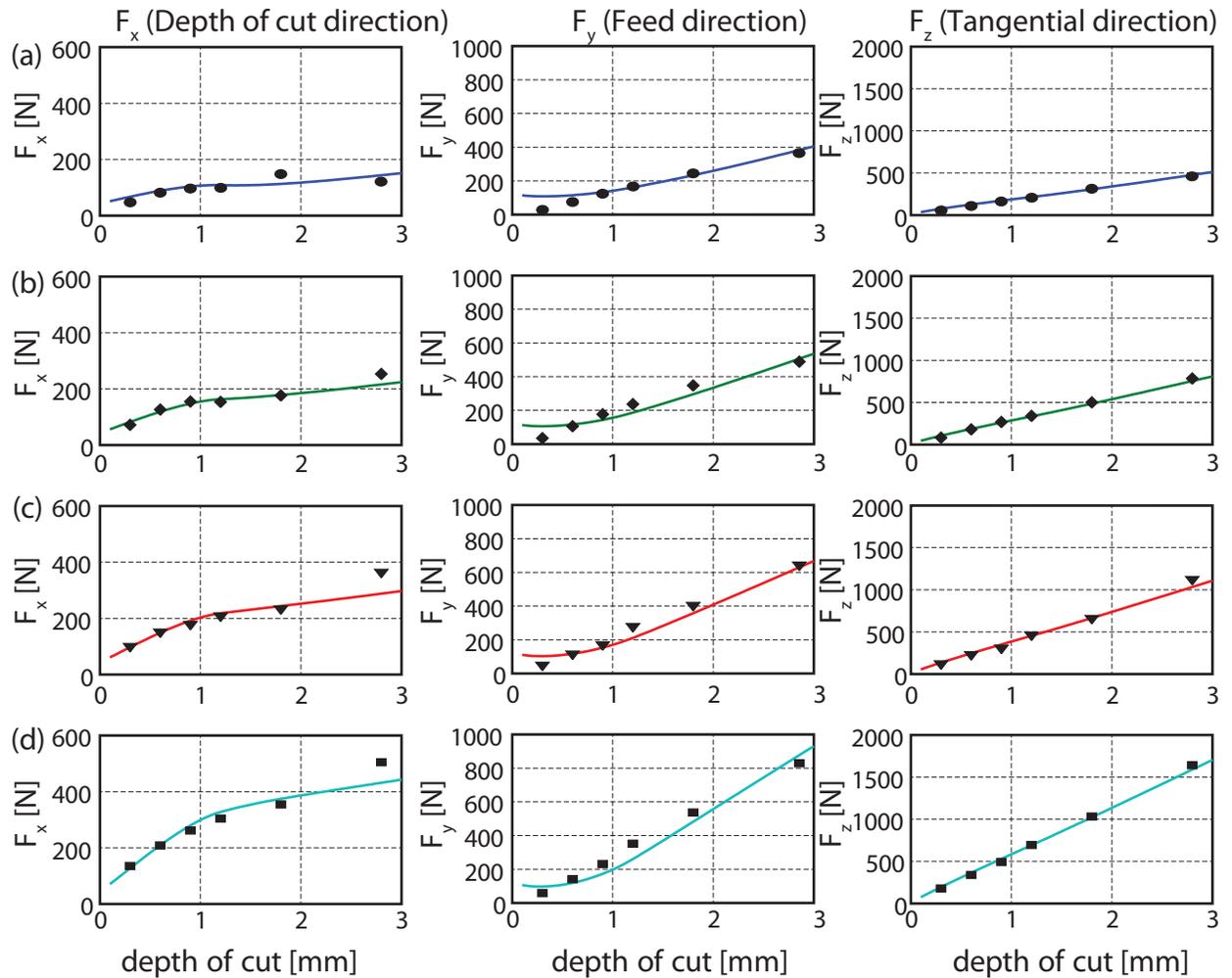
**Figure 2.16:** Approximate chord for a cutter with a straight edge [15]

the machine tool. Prediction of chatter stability enables the manufacturers to avoid scrapped parts, excessive tool wear, and damaged tools.

The method for obtaining the analytical stability lobes in orthogonal cutting process was developed by Tobias and Fishwick [28], Tlustý and Poláček [29], and is detailed by Altintas [14].

Ozlu and Budak [17] used this knowledge to develop a stability model for multi-dimension dynamic systems using a tool having a nose radius, in turning and boring. They discretized the uncut chip by using trapezoidal elements (Figure 2.18) and used a numerical solution for the eigenvalue problem. However, this approach does not take into account the full geometry of the chip (the cusp area) and requires extensive computational time.

Eynian and Altintas [2] used the approximate chord model and developed two regenerative chip models. Model I assumes a dynamically changing equivalent chip thickness. Model II defines the cutting forces as a function of both chip area and the chord length. Model II also considers the process damping effect. The stability of the obtained characteristic equations is evaluated by the



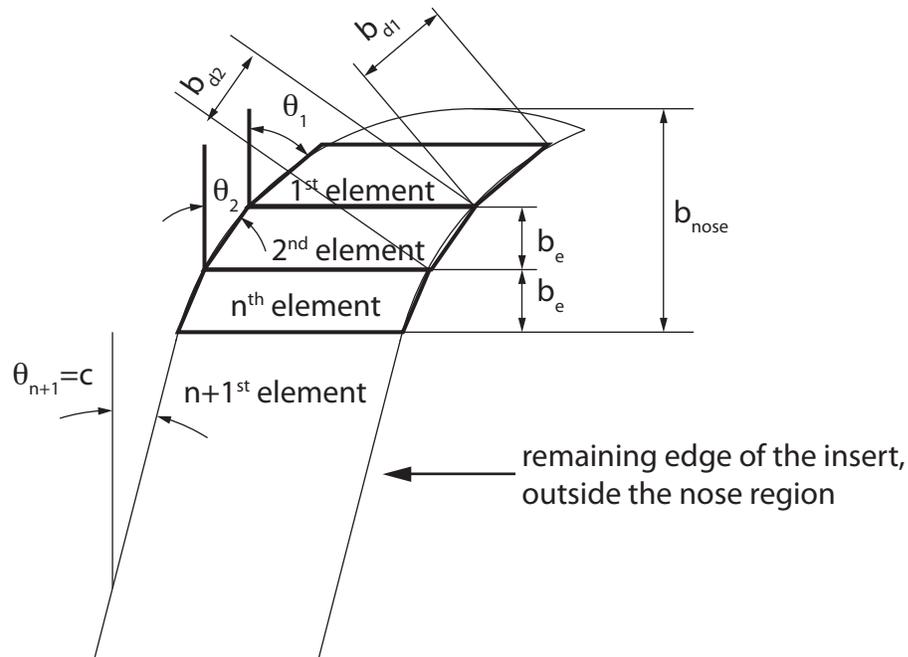
**Figure 2.17:** Experimental verification of Colwell's chord approximation [16]

Nyquist criterion.

## 2.4 Optimization Algorithms

### 2.4.1 Direct and Random Search

Search based algorithms use an objective function to evaluate the performance of a generated design vectors in the design space. If the optimization problem is constrained, the feasibility of each design vector is also checked. The optimum solution is chosen as the design vector with the highest performance, which does not violate any constraints.



**Figure 2.18:** Discretization of chip load by trapezoidal elements [17]

Direct search method generates and evaluates all possible design vectors within a design space, where the design variables are bounded, and discretized with a certain resolution.

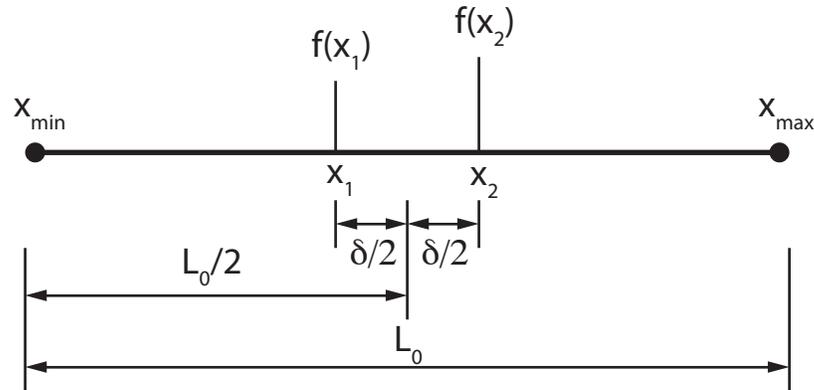
Random search method generates the design vectors by choosing a random value for each variable.

Basic procedure of search based optimization is described by Rao [18].

1. Generate the first trial design vector.
2. Verify whether the chosen design vector satisfies the constraints. The equality constraints are checked within a tolerance. If any constraint is violated, repeat from step 1.
3. Evaluate the objective function using the design vector. If the performance is improved in comparison to the current solution, store the values as the current solution, if not discard the design vector, and repeat from step 1.
4. Best design available after a specified number of trial designs is taken as the solution.

## 2.4.2 Dichotomous Search and Interval Halving Methods

A multi variable optimization problem can be reduced to a single dimensional optimization problem by fixing the rest of the design variables. The reduced problems can be solved by using simple one dimensional algorithms such as *Dichotomous Search* or *Interval Halving* methods [18].

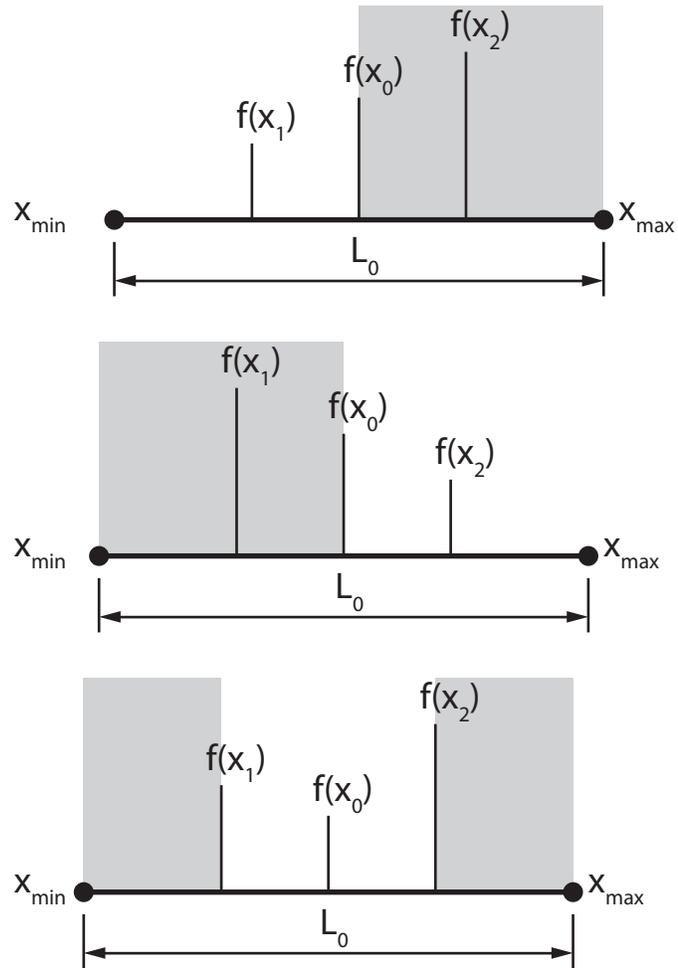


**Figure 2.19:** Dichotomous searching method [18]

In the dichotomous search method, two sets of experiments are needed. The design variable is set as close as possible to the mid point of the design space, separated by a chosen value  $\delta$ . The value ( $\delta$ ) should be selected in a way that it causes a significant variation in the objective function, that allows the interval of uncertainty to be halved with each pair of experiments, as shown in Figure 2.19.

Another similar one-dimensional optimization algorithm is the *Interval Halving* method. By dividing the interval of uncertainty into four smaller intervals at each step the interval where the possible optimal point lies can be halved as shown in Figure 2.20.

Both of these methods are only applicable to one dimensional optimization problems with a convex objective function. By setting one free design variable at each step, every design variable can be optimized one by one, until an optimum solution is reached.



**Figure 2.20:** Interval halving method [18]

### 2.4.3 Genetic Algorithms

Similar to the search based algorithms, genetic algorithms require the design spaces for each variable be expressed in discrete values. The possible intervals for the design variables must be discretized with a chosen resolution which determines the computational efficiency.

Genetic algorithms use the natural phenomena such as, reproduction, cross-over and mutation. Depending on the *survival of the fittest* theory, the optimal solution is obtained in a defined number of generations [18].

## 3 Contour Turning Process Model

Contour turning is a kind of turning process where the cutting tool moves along a curved tool path in the working-plane ( $X-Z$ ) of the machine tool. It is used to machine free-form surfaces of revolution as illustrated in Figure 3.1.

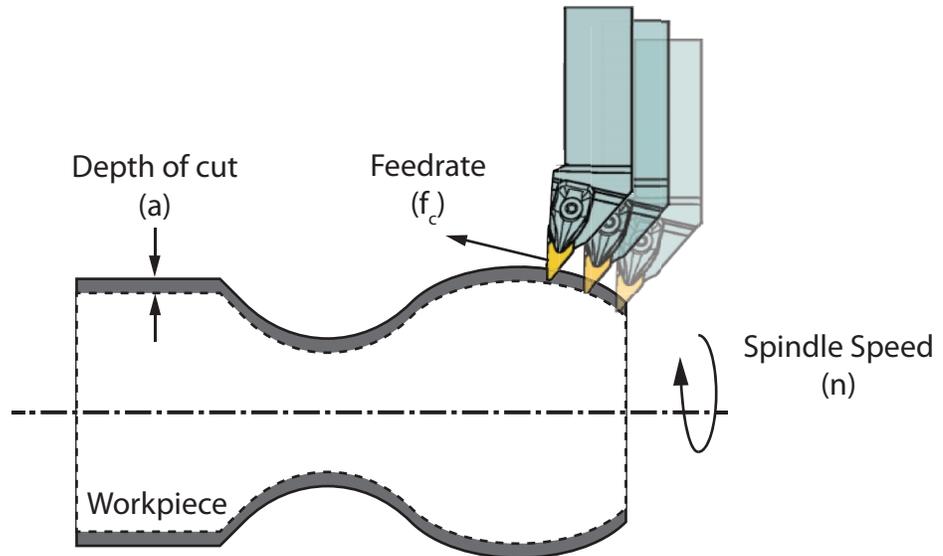
The process parameters, such as feedrate ( $f_c$ ) and spindle speed ( $n$ ), are defined in the same manner as they are for a straight cut. However, due to the free-form nature of the operation it is not always possible to define a static depth of cut value. This is true when the final part contour is different than the initial workpiece surface profile.

Contour turning is chosen as an example for the generalized turning process. During contouring, the cutting parameters and the uncut chip profile may vary along the tool path, therefore, it is necessary to repeat the simulation at small time intervals, leading to a large number of simulation steps. The development of computationally efficient algorithms is essential to reduce simulation time when a large number of tool paths are processed.

### 3.1 Angle Definitions for Contour Turning

Unlike a straight axial or radial turning process, the angle of the tool motion varies during contour turning. The instantaneous angle of the tool motion in the working plane is called the tool-motion angle ( $\psi_m$ ), as shown in Figure 3.2.

The inclination of the workpiece surface also changes along the tool-path depending on the in-process workpiece geometry. The local inclination angle of the surface is called the surface angle ( $\psi_s$ ), as shown in Figure 3.2.



**Figure 3.1:** Contour turning process

## 3.2 Tool–Workpiece Engagements

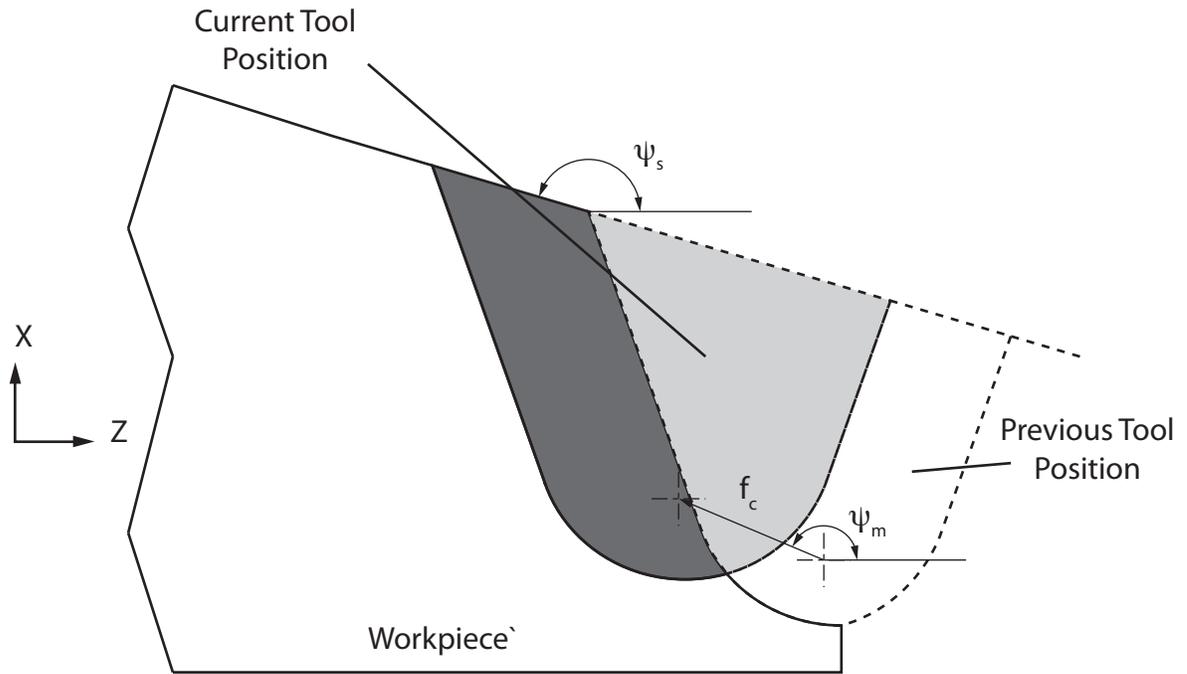
The variation of chip load along the tool path must be determined to simulate the physics of cutting operations. The chip load is defined as the region of the tool, which is instantaneously in-cut. This region is also equivalent to the intersection of the cutting tool and the workpiece at any given time during cutting. These intersections are called the tool–workpiece engagements, as shown in Figure 3.3.

In milling, the tool–workpiece engagement is the area on the tool surface, which is in contact with the workpiece (Figure 3.4a). These 3D engagements can be mapped into a two–dimensional coordinate system [30].

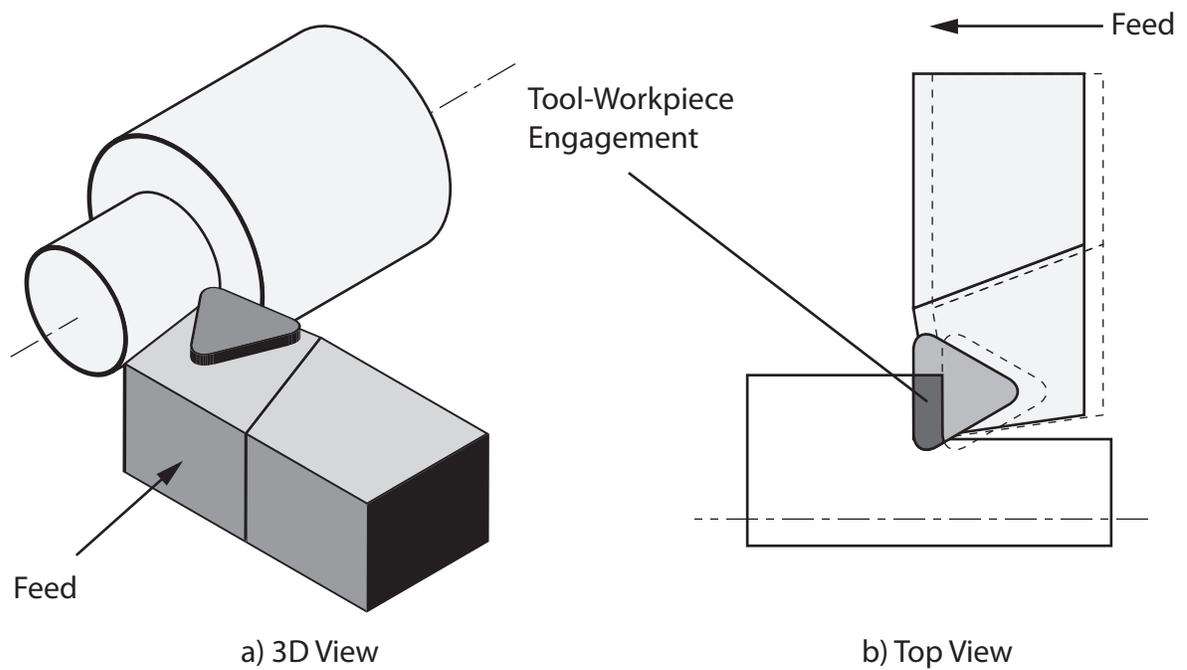
In turning processes, the engagement profile is the region on the cutting tool’s rake face, which is currently in contact with the uncut workpiece. This contact area is also defined as area swept by the tool edge profile during one complete spindle revolution, as shown in Figure 3.4b.

In straight turning processes, the uncut chip area is a function of the feedrate and the depth of cut, and therefore, does not vary along the tool path.

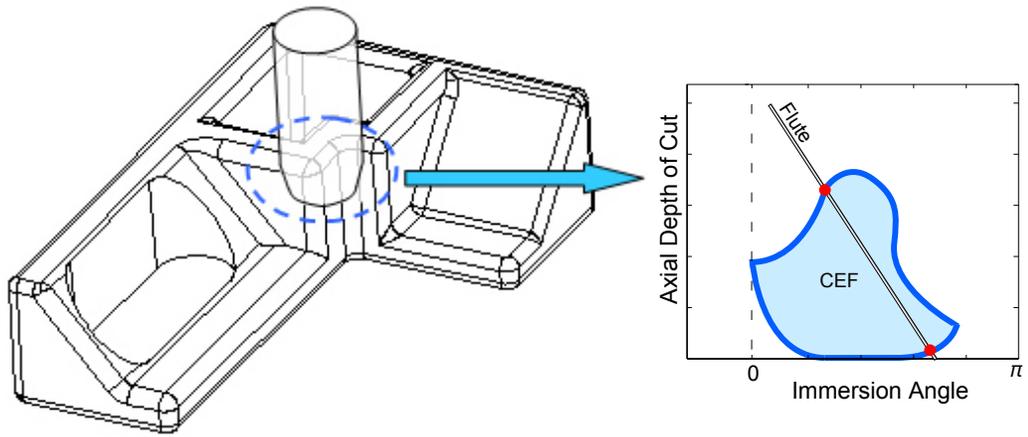
Unlike straight turning processes, when the tool path is a contoured profile, the tool–workpiece engagements vary along the tool path. For contour turning simulations, an effective method to



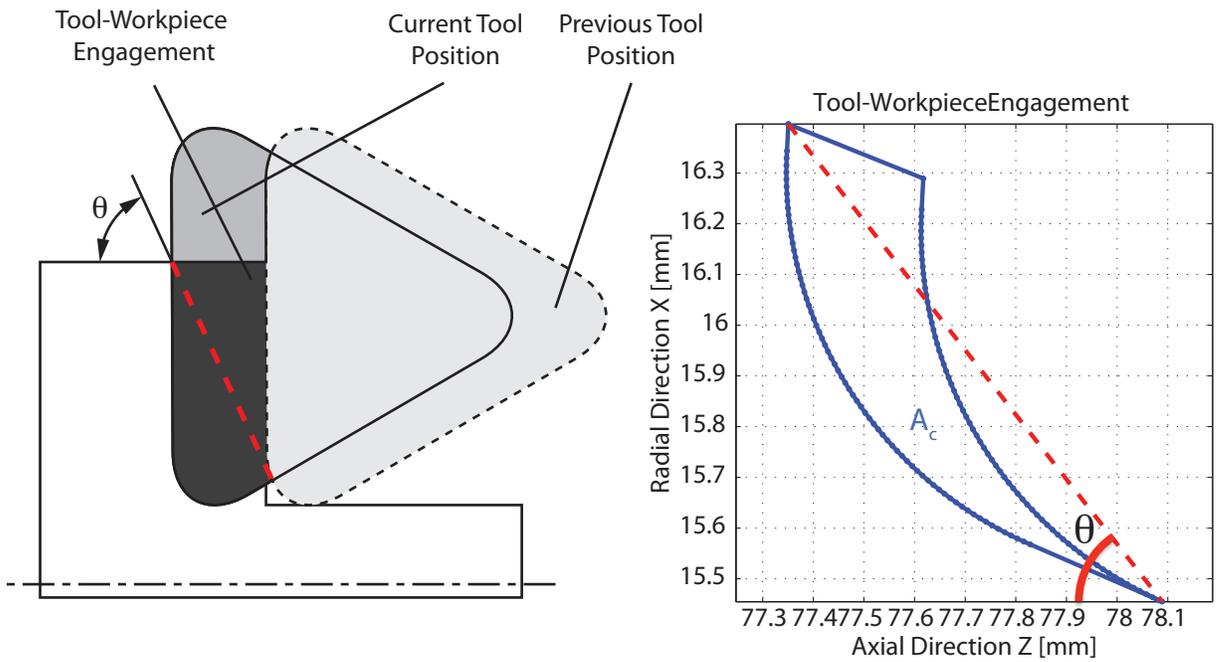
**Figure 3.2:** Tool-motion and surface angles in contour turning



**Figure 3.3:** Tool-workpiece engagements



(a) Engagement maps in milling [30]



(b) Engagement maps in turning

**Figure 3.4:** Comparison of engagement maps in milling and turning

calculate the time varying tool–workpiece engagements is developed and presented in Chapter 4.

### 3.3 Cutting Force Model

One of the major objectives of turning process simulation is the prediction of cutting forces. The cutting forces are modeled proportional to the uncut chip area, and the total length of the cutting edge that is in–cut. Cutting forces  $F_t$ ,  $F_n$  and  $F_r$  act on the tangential, normal and radial directions respectively as shown in Figure 3.6. A widely accepted, semi-mechanistic model is presented by Altintas [14].

$$\begin{aligned}
 F_n &= K_{nc}A_c + K_{ne}L_c \\
 F_r &= K_{rc}A_c + K_{re}L_c \\
 F_t &= K_{tc}A_c + K_{te}L_c
 \end{aligned} \tag{3.1}$$

or expressed in matrix form

$$\begin{Bmatrix} F_n \\ F_r \\ F_t \end{Bmatrix} = \begin{bmatrix} K_{nc} & K_{ne} \\ K_{rc} & K_{re} \\ K_{tc} & K_{te} \end{bmatrix} \begin{Bmatrix} A_c \\ L_c \end{Bmatrix} \tag{3.2}$$

where  $A_c$  is the total uncut chip area or in other words the area enclosed by tool–workpiece engagement profile.  $L_c$  is the total length of the tool edge actively in cut.

The terms  $K_{tc}, K_{nc}, K_{rc}$  are the *cutting force coefficients* in the tangential, normal and radial directions respectively. The cutting force coefficients account for the shearing forces during the turning process.  $K_{te}, K_{ne}, K_{re}$  are called the *edge force coefficients* which account for the forces which do not contribute to the shearing process. The edge force coefficients are known to be sensitive to changes in tool wear, temperature, tool coating and use of lubrication.

### 3.3.1 Cutting Coefficients

#### 3.3.1.1 Cutting force coefficients

Cutting force coefficients for a given material are obtained experimentally by conducting a series of orthogonal cutting tests as explained in Section A.2. Cutting forces are measured to obtain process parameters under different cutting conditions.

The material dependent properties are obtained for each workpiece material, and stored in databases as a function of cutting parameters [31].

$$\begin{aligned}\tau_s &= f(V_c, h_c, \alpha_n) \\ \phi_c &= f(V_c, h_c, \alpha_n) \\ \beta_a &= f(V_c, h_c, \alpha_n)\end{aligned}\tag{3.3}$$

where  $\tau_s$ ,  $\phi_c$  and  $\beta_a$  are shear yield stress, shear angle and average friction angle respectively. The cutting parameters are cutting speed ( $V_c$ ), chip thickness ( $h_c$ ), and normal rake angle ( $\alpha_n$ ).

#### 3.3.1.2 Edge force coefficients

Edge force coefficients are specific to each workpiece–tool material pair. The edge force coefficients are calibrated using a mechanistic approach, and their average values for a specific workpiece material are stored in databases. The edge force coefficients for AISI 1045 steel paired with two different tool material are presented in Section A.2.

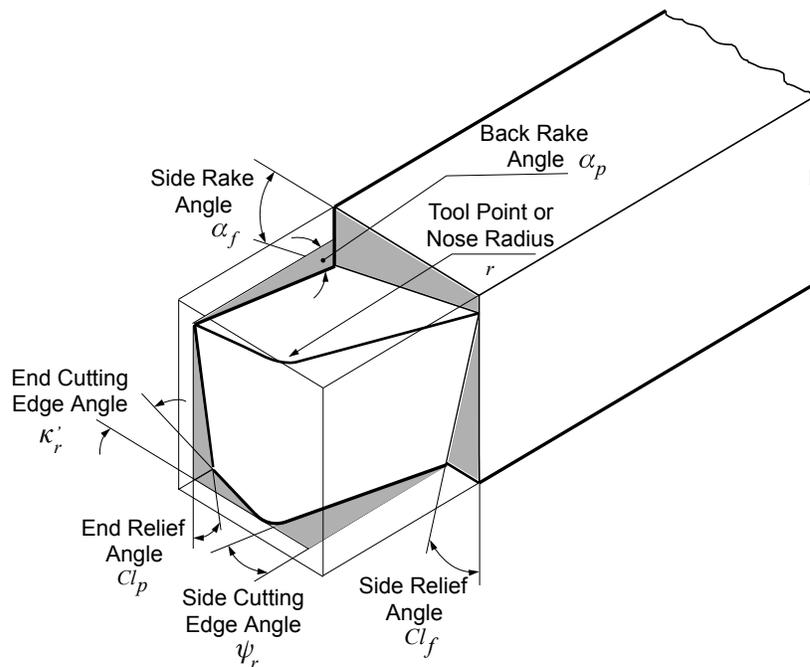
#### 3.3.1.3 Orthogonal cutting

In orthogonal cutting, the cutting force coefficients can be expressed as follows:

$$\begin{aligned}
K_t &= \tau_s \frac{\cos(\beta_a - \alpha_r)}{\sin \phi_c \cos(\phi_c + \beta_a - \alpha_r)} \\
K_n &= \tau_s \frac{\sin(\beta_a - \alpha_r)}{\sin \phi_c \cos(\phi_c + \beta_a - \alpha_r)} \\
K_r &= 0
\end{aligned} \tag{3.4}$$

However, most turning tools have oblique geometry, therefore, the corresponding cutting force coefficients are calculated by *Orthogonal to Oblique Transformation* method which is explained in the following section.

### 3.3.2 Orthogonal to Oblique Transformation



**Figure 3.5:** Geometry of an oblique turning tool [14]

For analyzing the mechanics of oblique cutting, normal rake and oblique angle of the cutting tool must be defined. Since most tools and inserts do not have simple geometries, it is common to define oblique angles by using equivalent angles [32]. For the tool shown in Figure 3.5, equivalent

oblique angle ( $i$ ), equivalent orthogonal angle ( $\alpha_o$ ) and normal rake angle angle ( $\alpha_n$ ) are defined by

$$\begin{aligned}\tan \alpha_o &= \tan \alpha_f \cos \psi_r + \tan \alpha_p \sin \psi_r \\ \tan i &= \tan \alpha_p \cos \psi_r + \tan \alpha_f \sin \psi_r \\ \tan \alpha_n &= \tan \alpha_o \cos i\end{aligned}\tag{3.5}$$

Once all the required cutting parameters are obtained, cutting coefficients for a given material can be calculated as follows:

$$\begin{aligned}K_{nc} &= \frac{\tau_s}{\sin \phi_n \cos i} \frac{\sin(\beta_n - \alpha_n)}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2 \eta \sin^2 \beta_n}} \\ K_{rc} &= \frac{\tau_s}{\sin \phi_n} \frac{\cos(\beta_n - \alpha_n) \tan i - \tan \eta \sin \beta_n}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2 \eta \sin^2 \beta_n}} \\ K_{tc} &= \frac{\tau_s}{\sin \phi_n} \frac{\cos(\beta_n - \alpha_n) + \tan i \tan \eta \sin \beta_n}{\sqrt{\cos^2(\phi_n + \beta_n - \alpha_n) + \tan^2 \eta \sin^2 \beta_n}}\end{aligned}\tag{3.6}$$

where the material dependent properties,  $\tau_s$ ,  $\phi_c$  and  $\beta_a$  are obtained from Equation 3.3. The chip flow angle is assumed to be equal to the oblique angle due to Stabler's chip flow rule [33].

$$i = \eta\tag{3.7}$$

Also the friction angle and the shear angle are assumed to be the same as orthogonal cutting for oblique cutting.

$$\beta_n = \beta_a\tag{3.8}$$

$$\phi_n = \phi_c\tag{3.9}$$

In contour turning operations, cutting parameters such as the cutting speed and the chip thickness vary with time, therefore, the cutting coefficients are calculated at each simulation step. This allows the cutting coefficients to vary as the chip geometry changes along the tool path.

### 3.3.3 Chip Flow Angle

In an ideal orthogonal turning operation with a straight cutting edge, cutting forces are generated in two principal directions, namely the normal direction, which is perpendicular to the cutting edge, and the tangential direction, which is the direction of the cutting speed.

Most turning tools have a nose radius and side cutting edge angle. The direction of the resultant force is affected by the oblique geometry of the tool, therefore, the resultant force has to be expressed in a three dimensional coordinate system. In addition to the normal and tangential forces, radial forces are considered as shown in Figure 3.6.

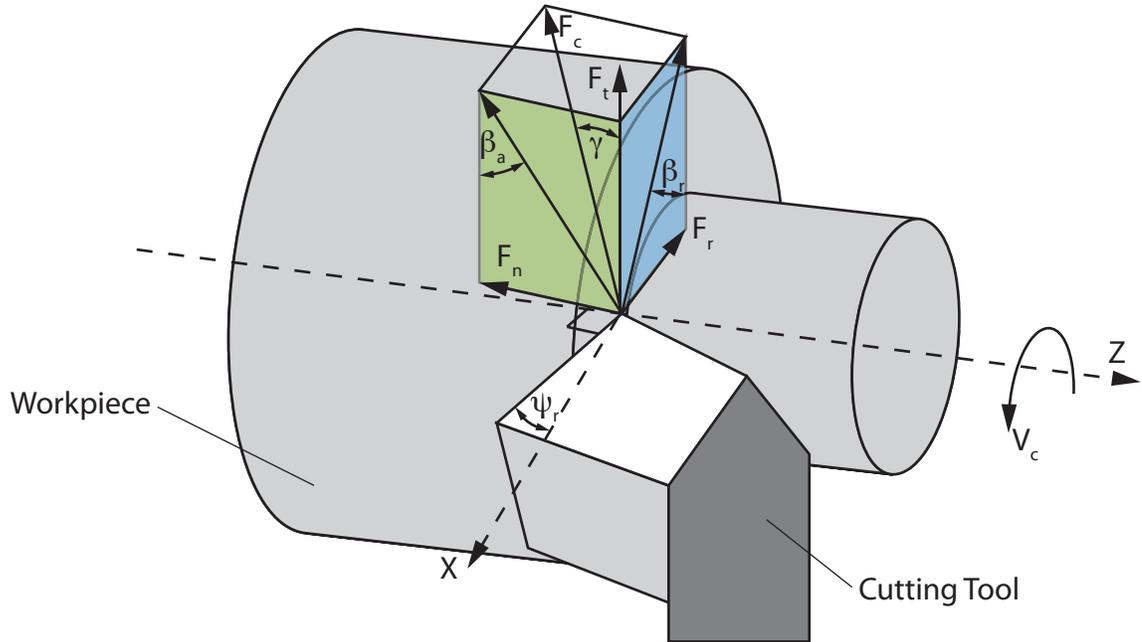
The oblique geometry is accounted for, by using the orthogonal to oblique transformation as explained in Section 3.3.2. It is still necessary, however, to find the direction of the resultant force on the rake face, or in other words the angle of the chip flow. There are numerous models to predict the chip flow angle [17, 12]. However, most of these models discretize the chip load into small elements, and superpose cutting forces generated by these small elements to find the direction of the resultant cutting force. The use of discrete elements increases the computational load, which is a major drawback for simulation purposes.

To estimate the chip flow direction in a more efficient manner, the Colwell effective chord approximation is employed in the virtual turning system.

#### 3.3.3.1 Effective chord approximation

In order to avoid discretizing the chip geometry into small elements, a single straight cutting edge is assumed as proposed by Colwell [15]. He explains that the effect of the tool nose radius can be modeled by using an approximate cutting edge, that is chosen to be the major diameter of a given tool-workpiece engagement profile, as shown in Figure 2.16 and Figure 3.4b.

The length of the approximate chord ( $L_{eff}$ ) and the approximate chord angle ( $\theta$ ) are obtained



**Figure 3.6:** Cutting forces during oblique cutting

from the engagement profile geometry. Normal force is assumed to be acting in the direction perpendicular to the effective chord, while the radial force is assumed to be acting along the chord (Figure 2.16).

### 3.3.4 Effective Chip Thickness

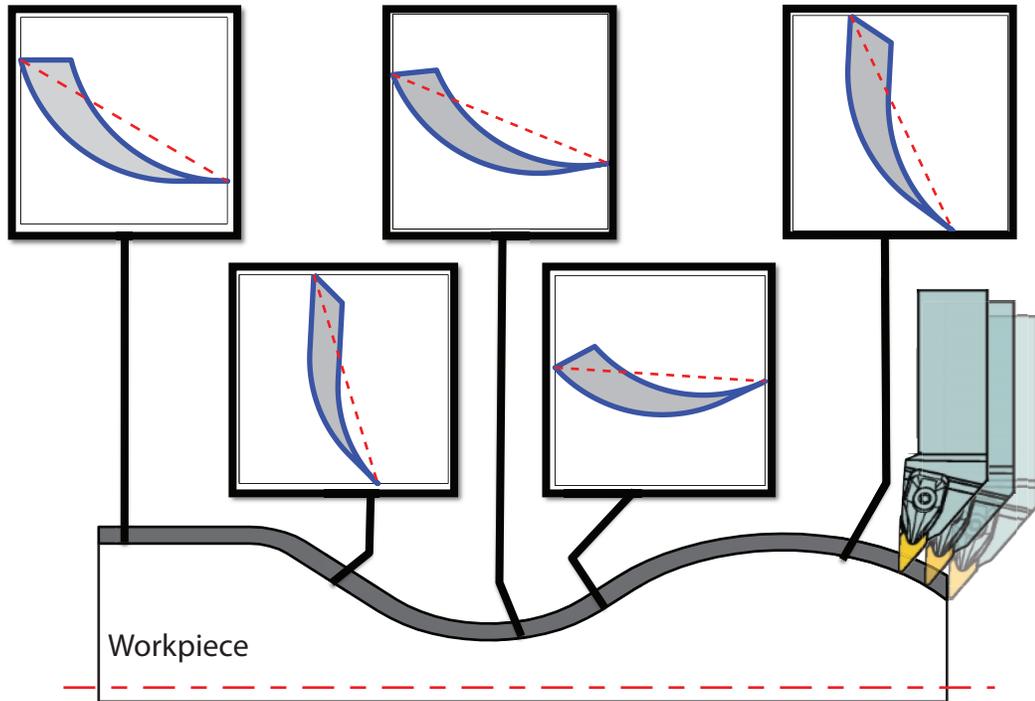
When the tool path is a free-form curve, as in the case of contour turning, the chip thickness varies along the tool path, even if the depth of cut and the feedrate remain constant. The variation of chip thickness is shown in Figure 3.7. The chip thickness ( $h_c$ ) is assumed to be the thickness of the uncut chip profile measured perpendicularly to the approximate chord as suggested by Colwell [15].

When the surface inclination angle ( $\psi_s$ ) is known, the chip thickness can be calculated as

$$h_c = -f_c \sin(\theta + \psi_s) \quad (3.10)$$

where  $\psi_s$  is the instantaneous surface inclination as shown in Figure 3.2.

In a general turning process it is assumed that there is no a priori knowledge about the work-



**Figure 3.7:** Changing engagement geometry along the tool path in contour turning

piece surface profile. A more generalized method for estimating the chip thickness is used. The effective chip thickness is calculated by dividing the uncut chip area, to the total cutting edge length [34].

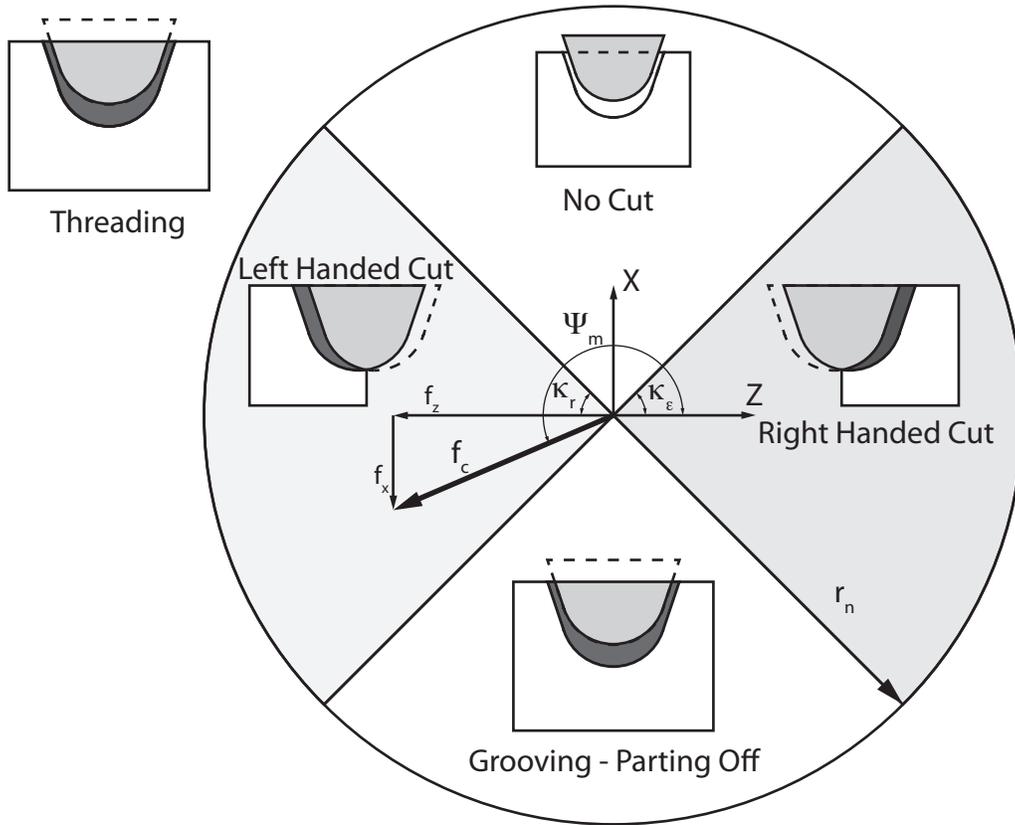
$$h_{\text{eff}} = \frac{A_c}{L_c} \quad (3.11)$$

where  $L_c$  is the cutting edge length, and  $A_c$  is the uncut chip area.

### 3.4 Tool Motion Classification

For a general purpose turning simulation, the force model must be generalized enough to simulate a range of turning operations. Depending on the tool motion, five different cutting operations are defined, similar to the four quadrant division proposed by Reddy [13]. The type of cut is determined by the instantaneous tool motion angle and feedrate (Figure 3.8).

The instantaneous tool motion vector is written as



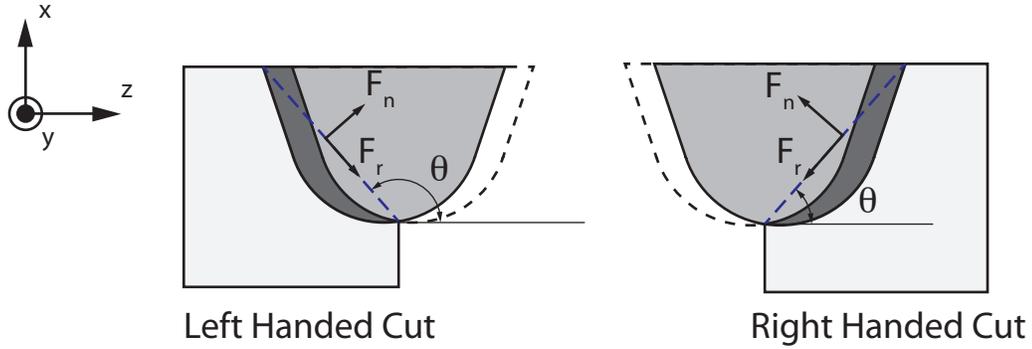
**Figure 3.8:** Tool motion classification

$$f_c = \begin{Bmatrix} f_z \\ f_x \end{Bmatrix} \quad (3.12)$$

where  $f_z$  and  $f_x$  are the axial and radial components of the feed respectively. The angle ( $\psi_m$ ) and magnitude ( $f_c$ ) of this vector in relation to the tool geometry defines the type of cut.

$\|f_c\| > r_n$  Feedrate value is larger than the tool nose radius. All three edges of the tool are in contact with the workpiece. This type of feedrates are generally used for threading operations.

$-\kappa_r < \psi_m < \kappa_\epsilon$  Tool is cutting with the tool nose radius and right edge. This type of cut is classified as a Right Handed Cut.



**Figure 3.9:** Force model modification for left and right handed cuts

$\kappa_\epsilon \leq \psi_m \leq \pi - \kappa_r$  Tool is leaving the workpiece, no edges are in contact, therefore, there is no cut.

$\pi - \kappa_r < \psi_m < \pi + \kappa_\epsilon$  Tool is cutting with the tool nose radius and left edge. This type of cut is classified as a Left Handed Cut.

$\pi + \kappa_\epsilon \leq \psi_m \leq -\kappa_r$  Tool is cutting with all three edges, this type of cut occurs in grooving or parting-off processes.

In these inequalities  $\kappa_r$  and  $\kappa_\epsilon$  are the edge approach angles for left and right hand side angles respectively, and  $r_n$  is the tool nose radius.

These regions are represented by a circle of radius equivalent to the tool nose radius,  $r_n$ . The circle is split into four regions by the angles of  $\kappa_\epsilon$  and  $\kappa_r$ , as shown in Figure 3.8. By placing the feedrate vector  $f_c$  at the center of this circle, the type of cut is determined depending on the region in which the end point lies.

The virtual turning system considers, the left and right handed cuts. The validity and applicability of the models with grooving, parting-off, and threading operations must be further studied.

When using the approximate cutting edge, the cutting forces are expressed as normal, radial and tangential directions ( $F_n$ ,  $F_r$ ,  $F_t$ ). These forces are transformed into machine-tool coordinate system as:

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [C_{nm}]^T \begin{Bmatrix} F_n \\ F_r \\ F_t \end{Bmatrix} \quad (3.13)$$

It can be seen from Figure 3.9 that depending on the type of cut, the transformation matrix  $[C_{nm}]^T$  in Equation 3.13 needs to be modified. Following the coordinate system given in Figure 3.9, the transformation matrices for the two cases are written as:

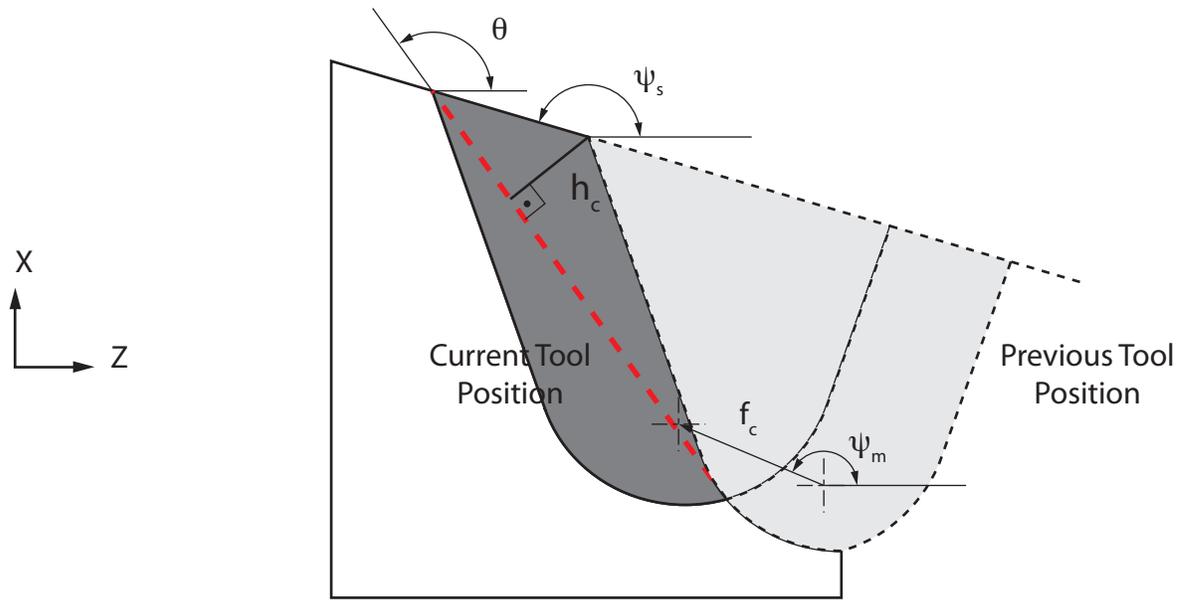
$$[C_{nm}]^T = \begin{cases} \begin{bmatrix} -\cos \theta & -\sin \theta & 0 \\ 0 & 0 & -1 \\ \sin \theta & -\cos \theta & 0 \end{bmatrix} & \text{for Left Handed Cut} \\ \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ 0 & 0 & 1 \\ -\sin \theta & -\cos \theta & 0 \end{bmatrix} & \text{for Right Handed Cut} \end{cases} \quad (3.14)$$

### 3.5 Stability with Process Damping

The chatter stability solution for axial turning process has been developed by Eynian and Altintas [2] who proposed two separate models. The first model considered dynamically changing chip thickness. The second model incorporates the variance in the uncut chip area, and the effective cutting edge during the cutting process. It also considers the process damping forces.

In order to develop a computationally efficient model, a modified version of the regenerative chip thickness model is used to determine chatter stability. The model is modified to include process damping forces due to flank penetration into workpiece at slow cutting speeds [2, 35, 36].

Dynamic cutting forces consist of two components; cutting and process damping forces. Cutting forces are due to the chip formation process of the turning. These forces are calculated using



**Figure 3.10:** Chip thickness in contour turning

the mechanistic force model explained in Section 3.3.

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [F]_{\text{Cutting}} + [F]_{\text{Process Damping}} \quad (3.15)$$

Process damping forces are expressed as a function of the vibration velocity, and the process damping gain matrix  $[J_v]$  as follows:

$$[F]_{\text{Process Damping}} = [J_v] \begin{Bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{Bmatrix} \quad (3.16)$$

The calculation of process damping gains  $[J_v]$  is explained in Section 3.5.1. Replacing Equation 3.13 and Equation 3.16 in Equation 3.15:

$$\begin{aligned}
\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} &= [C_{nm}]^T \begin{Bmatrix} F_n(t) \\ F_r(t) \\ F_t(t) \end{Bmatrix} + [J_v] \begin{Bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{Bmatrix} \\
\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} &= [C_{nm}]^T \begin{Bmatrix} K_{na} \\ K_{ra} \\ K_{ta} \end{Bmatrix} L_c h_c(t) + [J_v] \begin{Bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{Bmatrix}
\end{aligned} \tag{3.17}$$

The chip thickness changes due to the tool motion perpendicular to the effective chord. The dynamic chip thickness is expressed as a function of tool motion in the X and Z directions and the approximate chord angle. The difference between the previous vibrations and the current vibrations gives the dynamic chip thickness equation as:

$$h_c = c \sin \theta - \{[-x(t) \cos \theta + z(t) \sin \theta] - [-x(t - \tau) \cos \theta + z(t - \tau) \sin \theta]\} \tag{3.18}$$

where  $\tau$  is the spindle period.

Dropping the static term, the dynamically varying chip thickness is written in the matrix form

$$h_c = \begin{Bmatrix} \cos \theta & 0 & -\sin \theta \end{Bmatrix} \begin{Bmatrix} x(t) - x(t - \tau) \\ y(t) - y(t - \tau) \\ z(t) - z(t - \tau) \end{Bmatrix} \tag{3.19}$$

replacing this term in Equation 3.17 gives

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [C_{nm}]^T \begin{Bmatrix} K_{na} \\ K_{ra} \\ K_{ta} \end{Bmatrix} L_c \begin{Bmatrix} \cos \theta & 0 & -\sin \theta \end{Bmatrix} \begin{Bmatrix} x(t) - x(t - \tau) \\ y(t) - y(t - \tau) \\ z(t) - z(t - \tau) \end{Bmatrix} + [J_v] \begin{Bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{z}(t) \end{Bmatrix} \tag{3.20}$$

Expressing the dynamic force equation in Laplace domain we obtain

$$\begin{aligned} \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} &= (1 - e^{-s\tau})[J] \begin{Bmatrix} x(s) \\ y(s) \\ z(s) \end{Bmatrix} + s[J_v] \begin{Bmatrix} x(s) \\ y(s) \\ z(s) \end{Bmatrix} \\ \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} &= [(1 - e^{-s\tau})[J] + sJ_v] \begin{Bmatrix} x(s) \\ y(s) \\ z(s) \end{Bmatrix} \end{aligned} \quad (3.21)$$

Cutting tool vibrations can be expressed as a function of the cutting forces and the frequency response function of the flexible structure. The tool vibrations are expressed as

$$\begin{Bmatrix} x(s) \\ y(s) \\ z(s) \end{Bmatrix} = [\Phi(s)] \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} \quad (3.22)$$

Replacing the cutting tool vibrations into Equation 3.21 yields the equation for dynamic cutting forces as

$$\begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} = [(1 - e^{-s\tau})[J] + sJ_v] [\Phi(s)] \begin{Bmatrix} F_x \\ F_y \\ F_z \end{Bmatrix} \quad (3.23)$$

The characteristic equation of the system can be obtained by rearranging the above equation

$$\det \{ ([I]_{3 \times 3} - ((1 - e^{-s\tau})[J] + sJ_v) [\Phi(s)]) \} = 0 \quad (3.24)$$

where

$$J = [C_{nm}]^T \begin{Bmatrix} K_{na} \\ K_{ra} \\ K_{ta} \end{Bmatrix} L_c \begin{Bmatrix} \cos \theta & 0 & -\sin \theta \end{Bmatrix} \quad (3.25)$$

The process damping gain matrix  $J_v$  is evaluated for each engagement profile, as explained in the following section.

For every engagement profile the matrices  $J$  and  $J_v$  are calculated. When the frequency response function  $\Phi(s)$  is known for the flexible workpiece and tool, the stability of the characteristic equation given above can be determined by using the Nyquist criterion. The frequency response function of the structure is measured by experimental modal analysis techniques, such as impact hammer testing. The Nyquist criterion allows the detection of chatter and possible chatter frequencies.

### 3.5.1 Process Damping Gains

The process damping gain matrix is given by Eynian and Altintas [2]. After necessary coordinate transformations, the contribution from a straight edge is given as

$$[J_v]_{line} = -L_w^2 \frac{K_{sp} L_{line}}{2V_c} \begin{bmatrix} \cos^2 \kappa_r & 0 & \sin \kappa_r \cos \kappa_r \\ -\mu_c \cos \kappa_r (\cos \kappa_r + \sin \kappa_r) & 0 & -\mu_c \sin \kappa_r (\cos \kappa_r + \sin \kappa_r) \\ \sin \kappa_r \cos \kappa_r & 0 & \sin^2 \kappa_r \end{bmatrix} \quad (3.26)$$

where  $L_{line}$  is the length and  $\kappa_r$  is the inclination angle of the straight edge.

Contribution of an arc is given as

$$[J_v]_{arc} = -L_w^2 \frac{K_{sp}}{2V_c} \int_{\theta_0}^{\theta_1} \begin{bmatrix} \cos^2 \kappa_n & 0 & \sin \kappa_n \cos \kappa_n \\ -\mu_c \cos \kappa_n (\cos \kappa_n + \sin \kappa_r) & 0 & -\mu_c \sin \kappa_n (\cos \kappa_n + \sin \kappa_n) \\ \sin \kappa_n \cos \kappa_n & 0 & \sin^2 \kappa_n \end{bmatrix} r_n d\kappa_n \quad (3.27)$$

$$[J_v]_{arc} = -L_w^2 \frac{K_{sp}}{2V_c} \begin{bmatrix} \frac{\kappa_n}{2} + \frac{\sin 2\kappa_n}{4} & 0 & \frac{\sin^2 \kappa_n}{2} \\ -\mu_c \left( \frac{\kappa_n}{2} + \frac{\sin 2\kappa_n}{4} + \frac{\sin^2 \kappa_n}{2} \right) & 0 & -\mu_c \left( \frac{\sin^2 \kappa_n}{2} + \frac{\kappa_n}{2} - \frac{\sin 2\kappa_n}{4} \right) \\ \frac{\sin^2 \kappa_n}{2} & 0 & \frac{\kappa_n}{2} - \frac{\sin 2\kappa_n}{4} \end{bmatrix}_{\theta_0}^{\theta_1} r_n \quad (3.28)$$

where  $\theta_0$  and  $\theta_1$  are the limit angles of an arc and  $r_n$  is the radius.

$K_{sp}$  is an experimentally identified contact force coefficient,  $L_w$  is the flank wear,  $V_c$  is the cutting speed and  $\mu_c$  is the coefficient of friction which is assumed to be 0.3 for steels [36].

The damping gain coefficients ( $[J_v]$ ) are calculated by adding the contribution of each edge segment currently in-cut. Cutting edge segments which are in-cut are obtained by using the engagement model detailed in Chapter 4.

### 3.6 Feedrate Scheduling

The ultimate goal of virtual machining is to provide the industry with means of increasing productivity, in other words, decreasing the machining cycle times.

Conventionally, the cutting parameters are chosen based on on the experience of the machinists and engineers, or values suggested in handbooks and catalogues by the tool manufacturers [30]. The strict tolerances and surface requirements force the industry to use conservative feeds and speeds, causing a loss in productivity.

For contour turning, a feedrate scheduling approach is developed where feasible cutting conditions, which do not violate the machine tool physics, are identified.

### 3.6.1 Design Variables

- Feedrate ( $f_c$ ): Large feeds increase the cutting forces, surface roughness, and chip thickness. For each workpiece–tool pair suggested maximum feedrate values are provided by the tool manufacturer. The amplitude of the feed mark can be limited by setting a maximum feedrate  $(f_c)_{max}$ .  
When the feed is chosen too small, cutting tool rubs against the workpiece surface, preventing chip formation. Therefore a minimum limit can be chosen by applying a minimum chip thickness criteria.
- Spindle Speed ( $n$ ): The choice of spindle speed influences the tool–life and the required power during machining. Depending on the workpiece and cutting tool material, minimum and maximum limits for the spindle speed are suggested by manufacturer specifications.

### 3.6.2 Design Space

After the boundaries for the design variables are set, the design space is generated by using a specific resolution for each design variable. Smaller resolutions improve the simulation precision, while increasing the computational time.

Design space is generated by

$$\begin{aligned} f_c &= \left\{ (f_c)_{min} \quad (f_c)_{min} + \Delta f_c \quad \dots \quad (f_c)_{max} - \Delta f_c \quad (f_c)_{max} \right\} \\ n &= \left\{ (n)_{min} \quad (n)_{min} + \Delta n \quad \dots \quad (n)_{max} - \Delta n \quad (n)_{max} \right\} \end{aligned} \quad (3.29)$$

where  $(f_c, n)_{max}$  and  $(f_c, n)_{min}$  are the upper and lower boundaries of the variables respectively.  $\Delta f_c$ ,  $\Delta n$  are the resolution for each design variable.

### 3.6.3 Constraints

Constraints which limit the machining operations include, chatter stability, machine tool capabilities, such as maximum force, power, torque limits, and performance constraints, such as chip breakability and form errors [30],[14].

For contour turning process, the following constraints are considered:

- Force Constraints :  $F \leq F_{max}$

The simulated cutting forces are compared against the maximum allowable forces, and the solution is only accepted if the forces are smaller than the maximum allowable value.

- Power Constraint :  $P \leq P_{max}$

Power required during machining is calculated

$$P = F_y \frac{V_c}{60} \quad (3.30)$$

$$P = F_y \frac{2\pi rn}{60 \times 10^3} \quad (3.31)$$

where  $V_c$  is cutting speed,  $F_y$  is the tangential cutting force,  $n$  is spindle speed, and  $P$  is power in [W].

- Torque Constraint :  $T \leq T_{max}$

Required spindle torque in turning can be estimated as a function of current tool position and simulated cutting forces, as follows [14]:

$$T = F_y \left( \frac{d-a}{2} \right) \quad (3.32)$$

where  $F_y$  is the tangential cutting force,  $d$  is the workpiece diameter,  $a$  is the depth of cut in cylindrical turning, and  $T$  is torque in [Nm]. For process simulation, Equation 3.32 is rewritten by replacing the workpiece diameter with the current tool position.

$$T = F_y \left( \frac{x_{\text{tool}}}{2} \right) \quad (3.33)$$

where  $x_{\text{tool}}$  is the radial coordinate of the current tool position that is obtained by parsing the NC-code as explained in Section 4.6.

- Chip Thickness:  $h_{\min} < h_{\text{eff}} < h_{\max}$

A minimum chip thickness constraint prevents rubbing of the tool, therefore ensures chip formation. A maximum limit is also chosen as large chip thickness values cause tool chipping and tool breakage [30].

Chip thickness is approximated as

$$h_{\text{eff}} = \frac{A_c}{L_c} \quad (3.34)$$

where  $A_c$  is uncut chip area and  $L_c$  is cutting edge length.

- Stability Constraint

Chatter stability of the cut is checked using the Nyquist criterion, as explained in Section 3.5. The stability model allows for the detection of any cutting conditions that causes chatter instability.

If chatter is detected at an engagement map, a new depth of cut value is recommended, by searching for a value which leads to stable cutting conditions. NC-commands causing chatter instability are marked for revision.

### 3.6.4 Direct Search Algorithm

The cutting forces in contour turning are a function of cutting speed ( $V_c$ ), feedrate ( $f_c$ ), depth of cut ( $a_c$ ), and engagement and tool geometries:

$$F = f(V_c, f_c, a_c, \text{Engagement Geometry, Tool Geometry}) \quad (3.35)$$

The term *Engagement Geometry* is used to indicate that the resulting cutting forces are also a function of the tool, workpiece and tool path geometries. It should be noted that the engagement geometry itself is a function of the design variables.

$$\text{Engagement Geometry} = f(f_c, a_c) \quad (3.36)$$

Equation 3.35 and Equation 3.36 reveal that the cutting forces are a complex function of the process parameters, and chatter stability is determined only for the simulated cutting conditions. Hence, it is not possible to use conventional and linear optimization techniques. Instead, a feasible solution is searched by using a direct search algorithm as detailed in the following section.

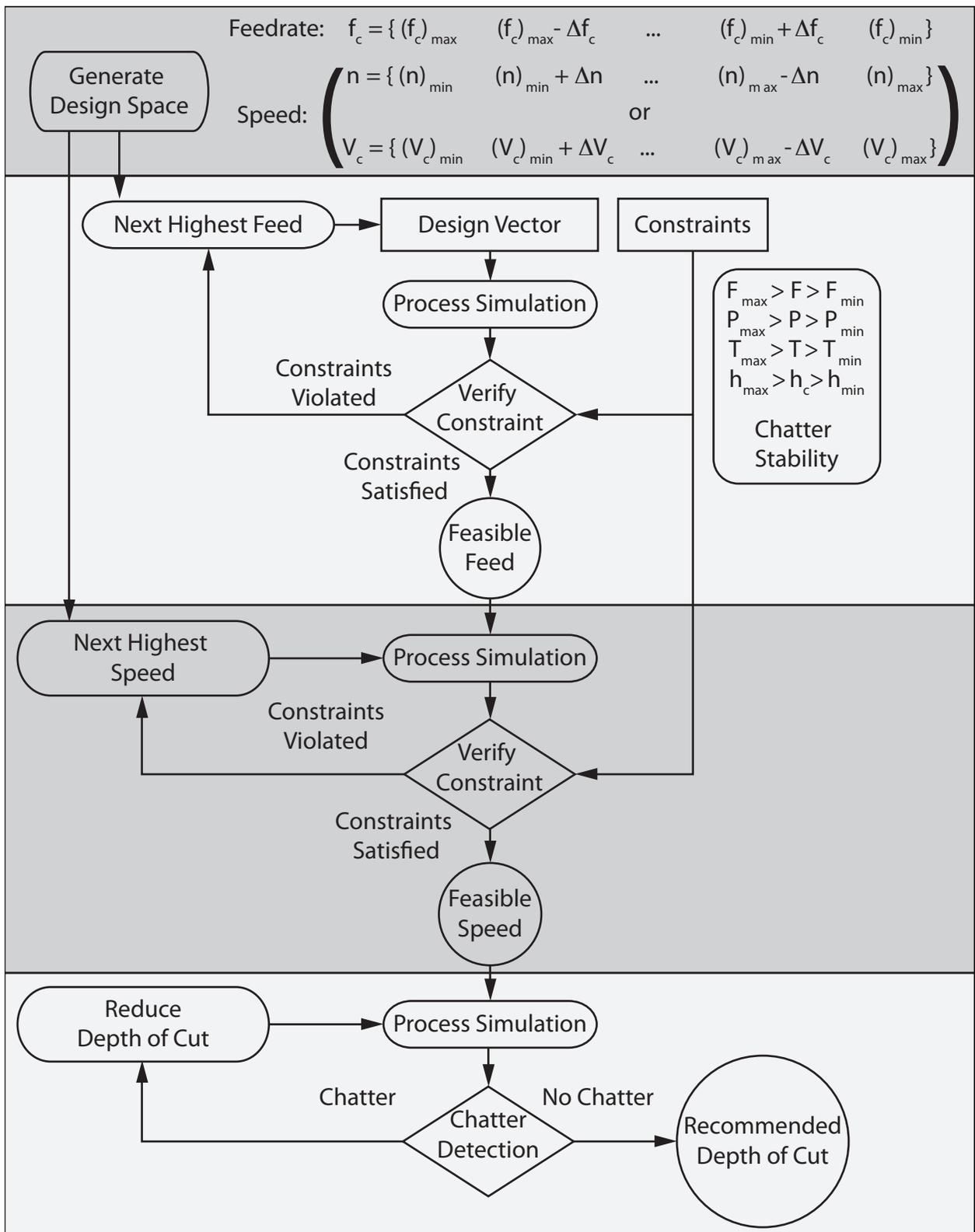
A direct search algorithm is used to search for the largest feedrate value, which does not violate the machine tool constraints. The performance of the process is evaluated, and the constraints are checked using the presented process model.

Most contour turning geometry consist of smooth and continuous features, therefore the feeds can be modified at big intervals. Each NC command of the part program is assigned a new feedrate value. This approach prevents sudden changes in the feedrate which may violate the dynamics of the machine drives, and cause poor finish.

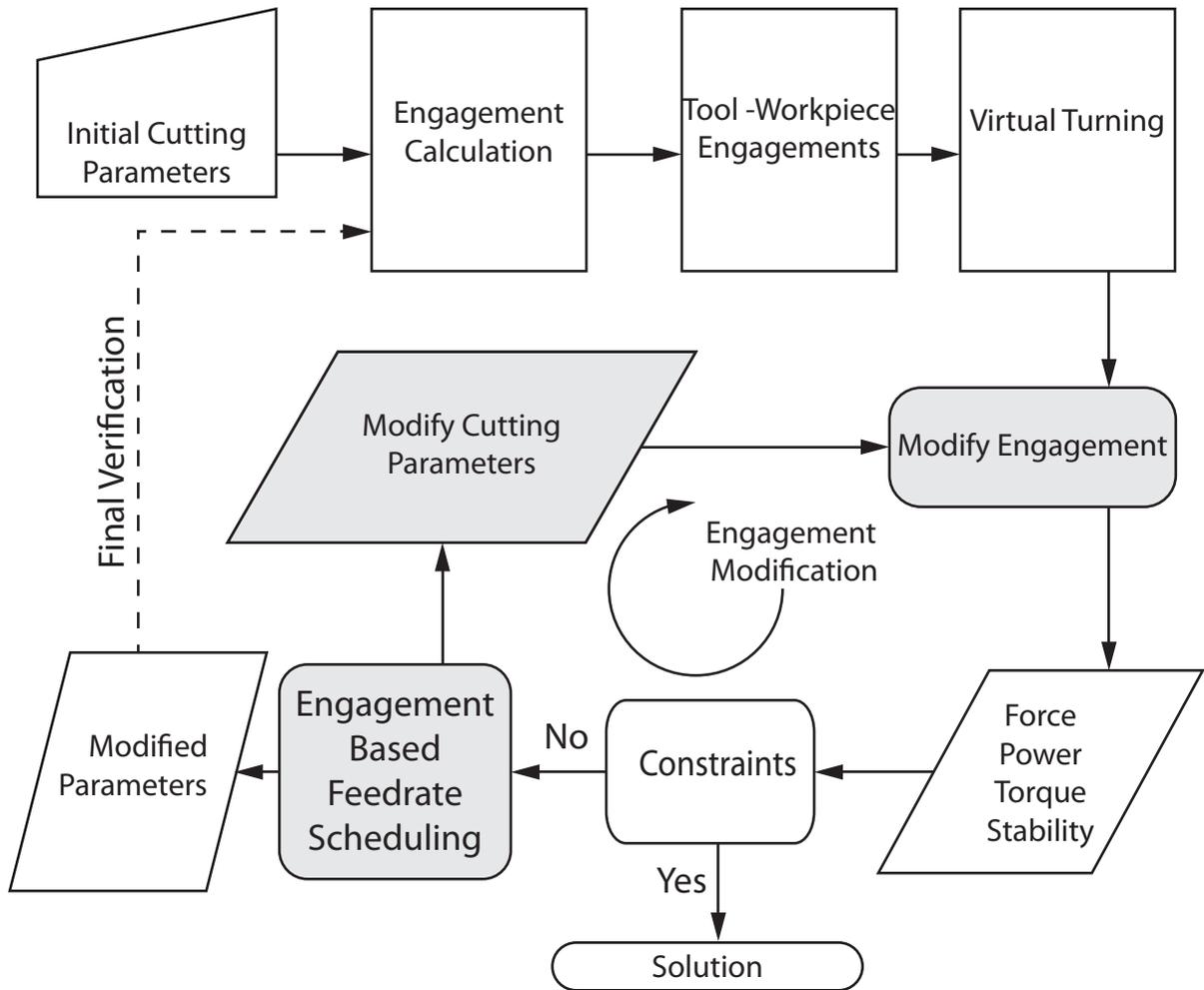
The optimization algorithm (Figure 3.11) is as follows :

1. The entire design space is generated using Equation 3.29.
2. Simulation is run with the maximum allowable feedrate value. Cutting forces, power and torque requirements, and chip thickness are calculated, and stability is checked.
3. If the constraints are satisfied, current feedrate is assigned for the current engagement map. If the constraints are violated, a smaller feedrate value is chosen and the procedure is repeated until a solution, which satisfies all the constraints, is found.

After the feedrate scheduling, a secondary step searches the highest feasible spindle speed. The direct search is repeated by using the spindle speed as the design variable.



**Figure 3.11:** Direct search algorithm



**Figure 3.12:** Engagement based feedrate scheduling using search based algorithm

Finally, the chatter stability is checked for the modified cutting conditions. If the cut is unstable, the depth of cut value is gradually reduced and a new chatter-free condition is identified. The engagement position causing chatter is marked, and the maximum feasible depth of cut value is recommended to the NC planner for revision.

### 3.6.5 Engagement Based Scheduling

At every step of the search algorithm, the cutting forces, power and torque are evaluated, and chatter stability is solved to ensure chatter-free operation, which requires the process simulation

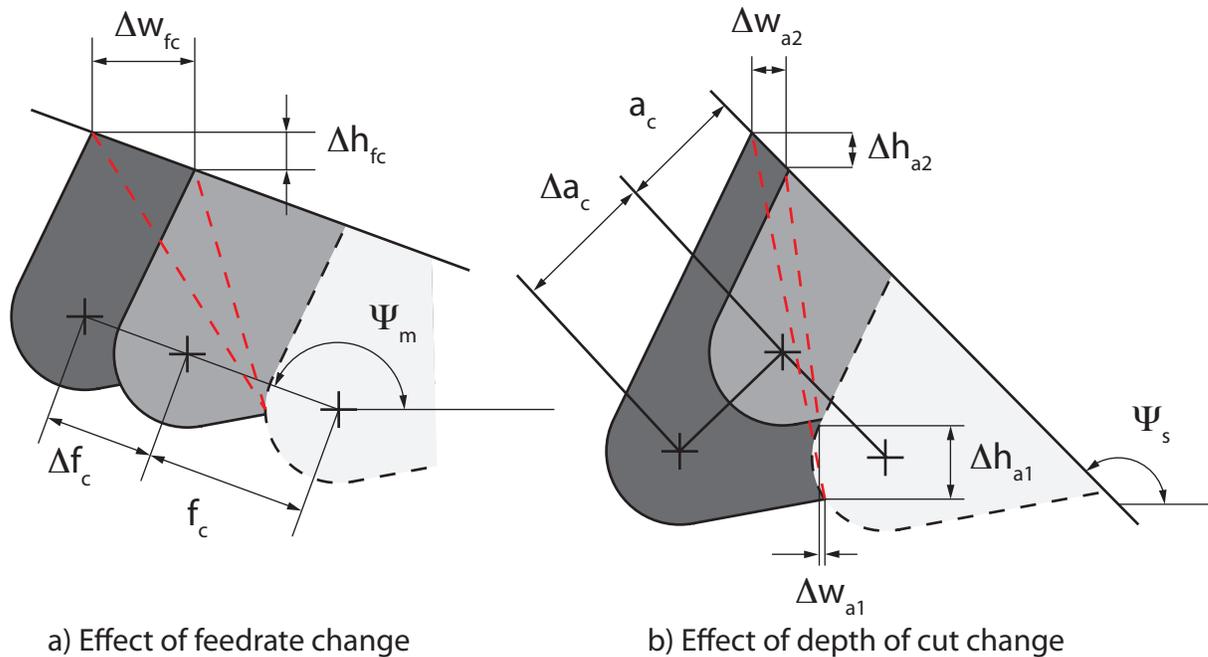
to be repeated a large number of times. This is not a feasible approach due to time constraints.

The most time consuming task is the calculation of the engagement profiles. Therefore, an analytical solution for engagement modification is developed.

Initial engagements are obtained using the engagement model explained in Chapter 4. The change in the engagement parameters is calculated as a function of the change of cutting parameters, and the parameters are optimized through the direct search method.

The final solution is verified by simulating the process entirely with the chosen cutting parameters (Figure 3.12).

### 3.6.5.1 Modified engagement geometry



**Figure 3.13:** Modification of tool–workpiece engagement geometry

For a given engagement profile, height ( $h$ ), width ( $w$ ), and length ( $L_{eff}$ ) values of the approximate chord is obtained as explained in the Section 3.3.3.1. Using the initial approximate chord geometry, the modified chord (Figure 3.13) is calculated as follows:

## Feedrate

The difference in the height and width of the approximate chord due to the change in the feedrate ( $\Delta f_c$  in Figure 3.13a) is approximated as:

$$\Delta w_{f_c} = \Delta f_c \cos(\psi'_s) \quad (3.37)$$

$$\Delta h_{f_c} = \Delta f_c \sin(\psi'_s)$$

with

$$\psi'_s = \pi - \psi_s$$

where  $\Delta f_c$  is the difference in the feedrate and ( $\psi_s$ ) represents the surface angle.

The modified approximate edge is now defined as:

$$\begin{aligned} w' &= \Delta w_{f_c} + w & h' &= \Delta h_{f_c} + h \\ L'_{eff} &= \sqrt{w'^2 + h'^2} & \theta' &= \tan^{-1} \frac{w'}{h'} \end{aligned} \quad (3.38)$$

The uncut chip area is assumed to be linearly proportional to the feedrate ( $f_c$ ). Therefore the modified chip load can be calculated as

$$A'_c = A_c \frac{f'_c}{f_c} \quad (3.39)$$

## Depth of cut

The difference in the height and width of the approximate chord due to the change in the depth of cut ( $\Delta a_c$  in Figure 3.13b) is approximated as:

$$\begin{aligned}
\Delta w_{a_c1} &= |\Delta a_c \tan(\psi'_s - \kappa_r)| \cos(\pi - \psi'_s) & \Delta h_{a_c1} &= |\Delta a_c \tan(\psi'_s - \kappa_r)| \sin(\pi - \psi'_s) \\
\Delta w_{a_c2} &= \Delta a_c \cos\left(\frac{3}{2}\pi - \psi'_s\right) & \Delta h_{a_c2} &= \Delta a_c \sin\left(\frac{3}{2}\pi - \psi'_s\right) \\
\Delta w_{a_c} &= \Delta w_{a_c1} + \Delta w_{a_c2} & \Delta h_{a_c} &= \Delta h_{a_c1} + \Delta h_{a_c2}
\end{aligned} \tag{3.40}$$

Writing Equation 3.38 for depth of cut:

$$\begin{aligned}
w' &= \Delta w_{a_c} + w & h' &= \Delta h_{a_c} + h \\
L'_{eff} &= \sqrt{w'^2 + h'^2} & \theta' &= \tan^{-1} \frac{w'}{h'}
\end{aligned} \tag{3.41}$$

Similar to Equation 3.39 the chip area is assumed to be linearly proportional to the depth of cut ( $a_c$ )

$$A'_c = A_c \frac{a'_c}{a_c} \tag{3.42}$$

### 3.7 Summary

Process model for contour turning operations is presented in this chapter. Once the engagement geometry of the cut is known, the cutting forces, power and torque requirements, and the process stability is evaluated using a semi-mechanistic material model. Based on the presented process model a feedrate scheduling method is proposed. Feedrate scheduling aims to improve the overall productivity by modifying the feedrate and spindle speed commands in the NC-code while ensuring the physical limitations of the machine tool are not violated.

## 4 Turning Simulation Model

A process model for contour turning operations is presented in Chapter 3, where the physics of cutting is defined by the tool–workpiece engagements. Generating the engagement maps is the essential task of turning simulations. Two methods for obtaining the engagement data are presented in this chapter.

The first method, which is detailed in the section one of this chapter, obtains the engagement maps by using a commercially available software. Implementation of this method is simple and fast, since all necessary models for tool, workpiece, and numerical control are handled by the external application. However, this approach suffers from compatibility and efficiency problems due to the dependency on a third party software.

To eliminate this dependency a new simulation model is developed specifically for contour turning simulations. The model is developed by considering computational implementation. Because the ultimate objective of this system is to form a process planning tool that can be used in a computer environment, computational efficiency, ease of implementation and visualization tasks are important.

The general geometric representation adopted by the new simulation model is presented in section two. The workpiece and tool models are described in sections three and four, respectively.

Section five details the tool–workpiece intersection model that is based on two–dimensional boolean operations. Process parameters that form the inputs of the force model are obtained by processing the generated tool–workpiece engagement maps.

Section six details simulation schemes that allow the user to decide between the simulation resolution and computational time.

The last section of the chapter explains briefly how the NC-code is parsed by the virtual turning system. The NC-code is used to generate the tool locations along the tool–path, as well as to determine cutting parameters such as cutting speed, feedrate and spindle speed.

## 4.1 Vericut Application Program Interface

Tool–workpiece engagement profiles are obtained directly from Vericut, which is a machining verification software capable of simulating CNC machine tools. With this software package, an application program interface (API) library is provided under the name of *Vericut Development Tools*. Version 7.0.3 of the development tools provides an API function which allows the extraction of tool–workpiece engagements during turning operations.

A custom API is developed in C++ that runs alongside the NC–simulation. An output file is created, which is processed by the virtual turning system, to estimate the cutting forces, torque and power requirements. The output file created by the Virtual Turning API, contains the resulting tool–workpiece engagement profiles for each simulation step.

The main API function which allows the engagement profile of the simulated turning operation to be extracted is the “opapi get turning contact profile” function. Invoking this function at every complete spindle revolution allows the correct tool–workpiece profiles to be obtained.

*Output data structure* The data fields of API output file are detailed as follows

**Step Number:** Number of the current simulation step

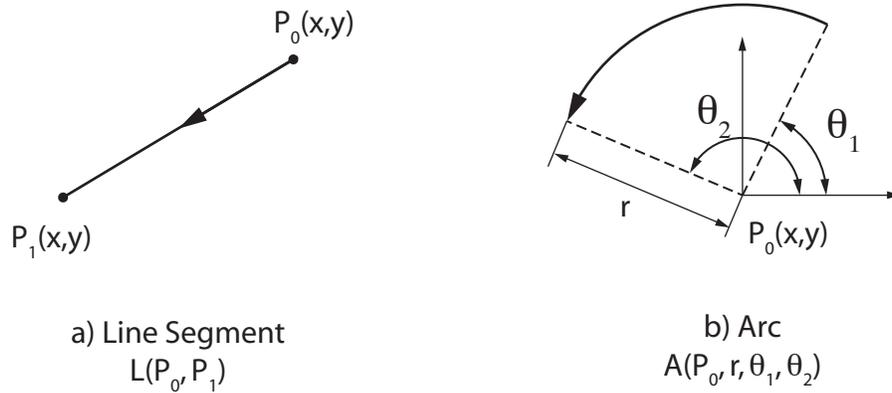
**Loop Number:** Number of closed loops defining the engagement profile

**Spindle Speed:** Spindle speed at the current simulation step, in rotations per minute

**Feedrate:** Feedrate at the current simulation step

**Total Time:** Total tool motion time elapsed, in seconds

$x_p$  : The X coordinate of the point or the centre point of the arc



**Figure 4.1:** Polycurve entity definitions

$z_p$  : The Z coordinate of the point or the centre point of the arc

$r_p$  : If zero, the point represents a corner of the boundary, if non-zero it is the radius of the arc

$\Delta x$  : The radial tool motion, in diameter coordinates

$\Delta z$  : The axial tool motion

## 4.2 Polycurve Representation

For the new simulation model, a general two dimensional curve representation scheme is proposed to represent the workpiece and tool geometries. Profiles are represented as series of line segments and arcs, which form a closed curve. Inclusion of circular elements in the boundary allows the geometry to be defined accurately, and with less elements, compared to polygons. This representation is referred to as the polycurve representation throughout this text.

Geometric models and boolean operations for polygons are extended to accommodate the inclusion of circular elements. Modifications to the polygon models are explained in the following sections.

### 4.2.1 Entity Definitions

Polycurves are represented by using two geometric entity types; namely line segments and arcs. Each entity is defined by geometric parameters as illustrated in Figure 4.1. Entity data is formatted

and stored using the following data structure.

#### 4.2.1.1 Data structure for entities

A data structure for the storage of the geometric entities is presented. The data structure combines the geometric properties required to define an arc or line segment, and extra data fields for entity classification.

**Entity.Type** Defines the type of the current entity as 'LINE' or 'ARC'

**Entity.Layer** Defines if the entity is part of the workpiece contour ('IPW'), or part of the tool contour ('TOOL')

**Entity.x (LINE)**  $\{x_0, x_1\}$  X-coordinates of the end-points

**Entity.x (ARC)**  $x_0$  X-coordinates of the center point

**Entity.y (LINE)**  $\{y_0, y_1\}$  Y-coordinates of the end-points

**Entity.y (ARC)**  $y_0$  Y-coordinates of the center point

**Entity.r (ARC)**  $r$  Radius of the arc

**Counter-Clockwise** if  $r > 0$

**Clockwise** if  $r < 0$

**Entity.t (ARC)**  $\{\theta_1, \theta_2\}$  Start and end angles of the arc

**Entity.Begin** The beginning point of the entity regardless of type

**Entity.End** The ending point of the entity regardless of type

## **4.3 Workpiece Model**

Workpiece model is an essential component of the virtual turning. Among the variety of workpiece representations, which were presented in Chapter 2, generator curve representation is adopted for the virtual turning system, due to its computational simplicity and efficiency. The main advantage of the generator curve representation is that it represents workpiece geometry in two-dimensions, which facilitates generation, modification and visualization of the in-process workpiece .

If the workpiece is not perfectly axi-symmetric, the workpiece model is extended to accommodate the non-symmetric features of the workpiece as presented in Section 4.7.

### **4.3.1 Generator Curve Representation**

Revolved solid geometries can be represented by their axis of revolution, and their cross-sections, as shown in Figure 4.2. This representation is called the Generator Curve Representation [9].

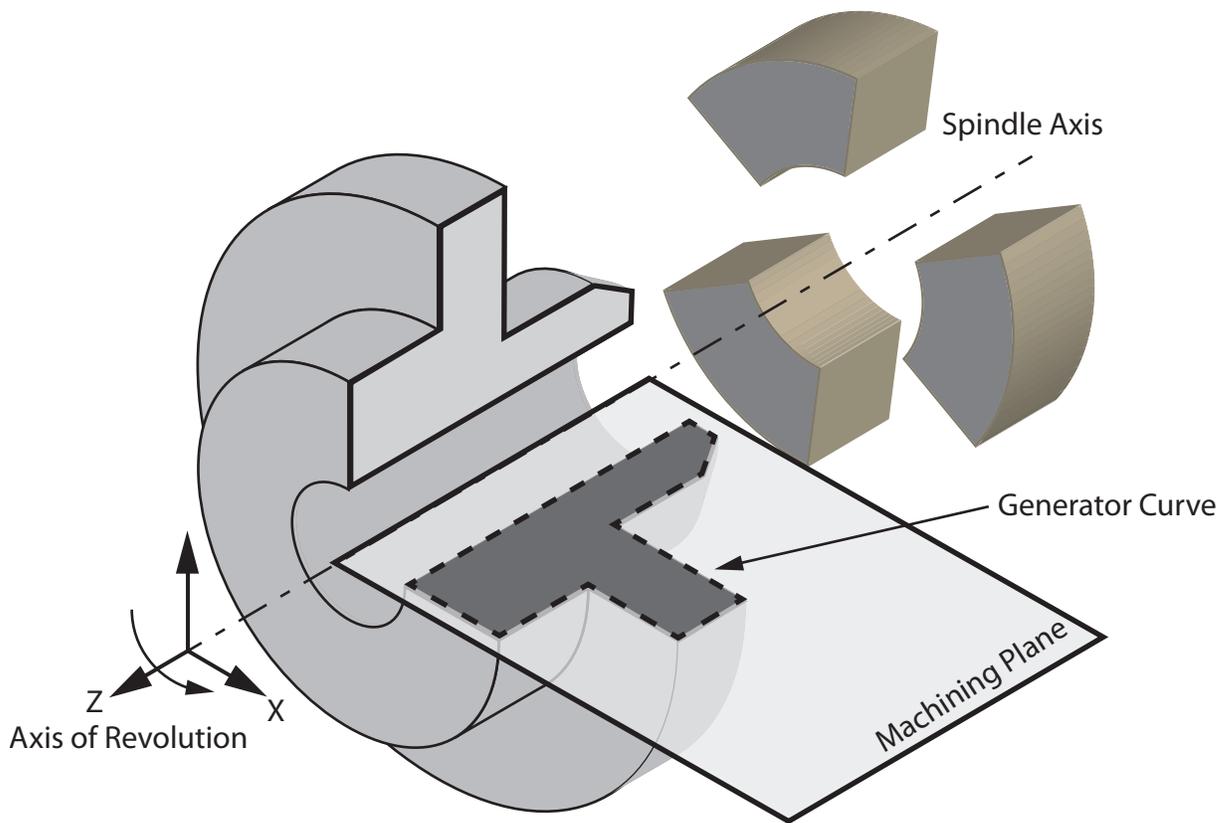
The CAD data of the workpiece is converted to the generator curve representation, before it is used in the virtual turning simulation. STL file format is chosen as the user input because it is a neutral file format that generates water tight solids. STL parts consist of triangular facets, therefore, the edges of an stl solid are linear.

The generator curve of the solid part is extracted from the solid model by intersecting the triangular facets with the machining-plane. The axis of revolution for the generator curve representation coincides with the spindle axis of the machine-tool (Figure 4.2).

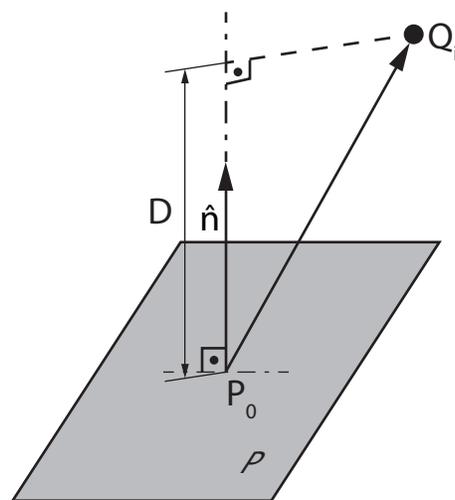
The elements of an STL solid are triangles in three-dimensional space. Finding the intersection of the solid model with a plane is reduced to the calculation of the intersection between a plane and a triangular facet. The cross-section is reconstructed by collecting the resulting intersection edges.

### **4.3.2 Intersection of a Triangle with a Plane**

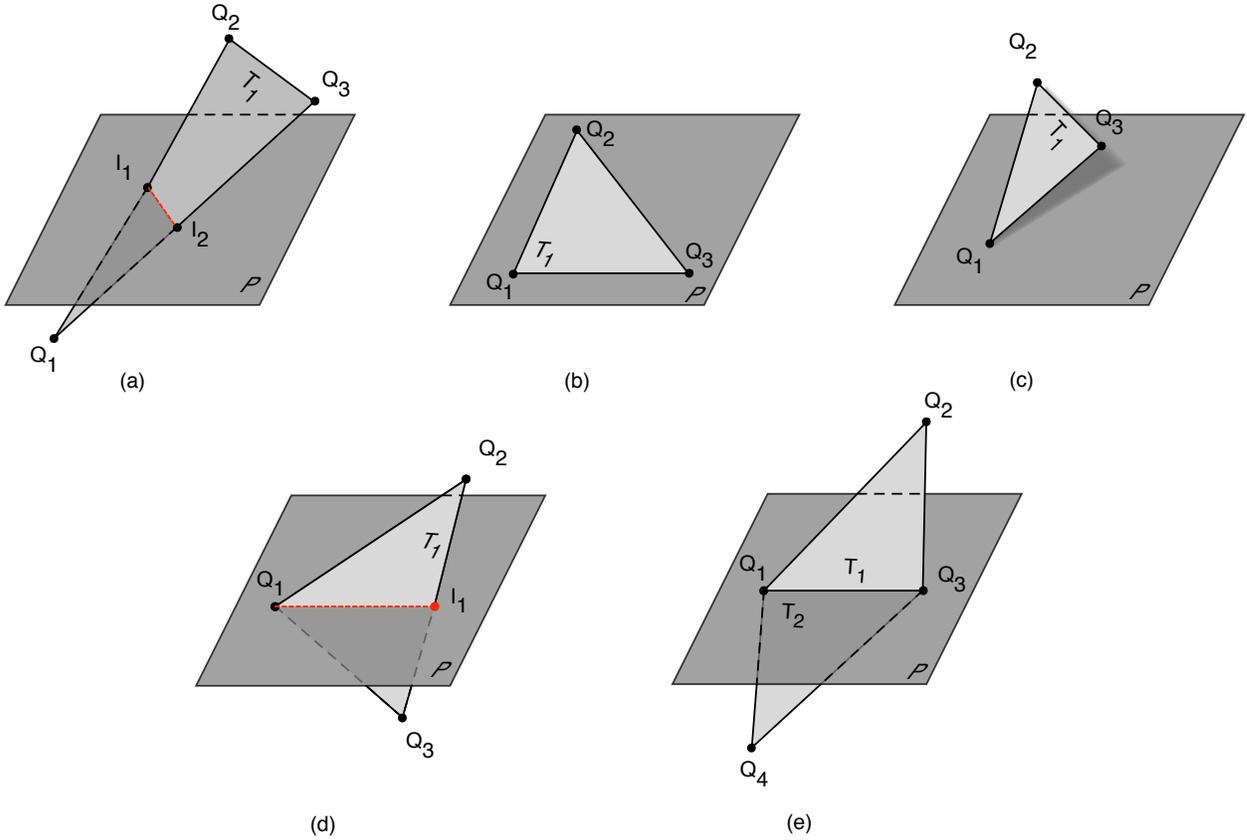
To classify the intersection problem, the vertices of the triangle must be located with respect to the plane, by calculating the signed distance between the plane and the point (Figure 4.3).



**Figure 4.2:** Generator curve for a solid part of revolution [9]



**Figure 4.3:** Signed distance of a point from a plane in space



**Figure 4.4:** Possible cases of triangle–plane intersection

Let  $T_1$  be the triangle with vertices  $Q_1(x_1, y_1, z_1)$ ,  $Q_2(x_2, y_2, z_2)$ ,  $Q_3(x_3, y_3, z_3)$  and  $P$  a plane in three dimensional space (Figure 4.4).

$P$  is defined as follows:

$$ax + by + cz + d = 0 \tag{4.1}$$

The distance between point  $i$  and the plane is given by the equation

$$D = \frac{ax_i + by_i + cz_i + d}{\sqrt{a^2 + b^2 + c^2}} \tag{4.2}$$

The distance can also be expressed by using the dot product

$$D = \frac{|P_0Q_i| \cdot \hat{n}}{\|\hat{n}\|} \quad (4.3)$$

where  $\hat{n}$  is the normal vector defining the plane

$$\hat{n} = \begin{Bmatrix} n_x \\ n_y \\ n_z \end{Bmatrix} = \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \quad (4.4)$$

Location of a point with respect to the plane is defined by the sign of the distance

$$\text{sign}(D) = \text{sign}(|P_0Q_i| \cdot \hat{n}) \quad (4.5)$$

where  $P_0$  is an arbitrary point on the plane. As a convention, this point is chosen as the machine–tool coordinate system origin.

- $\text{sign}(D) > 0$  The point is on the positive side of the plane
- $\text{sign}(D) < 0$  The point is on the negative side of the plane
- $\text{sign}(D) = 0$  The point is on the plane

Once all three vertices are located, the intersection problem is solved using the cases below:

**(a)** Two vertices lie on one side of the plane ( $Q_2, Q_3$ ), while the third one is on the other side ( $Q_1$ ).

The intersection points for the two edges that crosses from one side of the plane to the other are calculated (Figure 4.4a).

$$I_1 = P \cap |Q_1Q_2| \quad (4.6)$$

$$I_2 = P \cap |Q_1Q_3|$$

The intersection edge becomes

$$P \cap T = |I_1 I_2| \quad (4.7)$$

**(b)** All vertices lie on one side of the plane, or they all lie on the plane itself in which case the intersection is the entire triangular facet (Figure 4.4b). There is no apparent intersection edge.

**(c)** One vertex ( $Q_1$ ) lies on the plane while the other two ( $Q_3, Q_2$ ) are on the same side (Figure 4.4c).

$$P \cap T = Q_1 \quad (4.8)$$

Since there is only a single point on the plane, therefore, there are no apparent intersection edges.

**(d)** One vertex ( $Q_1$ ) lies on the plane while the other two ( $Q_2, Q_3$ ) lie on different sides. A single intersection point ( $I_1$ ) is calculated. The intersection edge is the line segment between the point on the plane and the calculated intersection point (Figure 4.4d).

$$I_1 = P \cap |Q_2 Q_3| \quad (4.9)$$

$$P \cap T = |Q_1 I_1|$$

**(e)** Two vertices of the triangular facet lie on the plane ( $Q_1 Q_3$ ). The intersection edge is the line segment between these two points (Figure 4.4e).

$$P \cap T = |Q_1 Q_3| \quad (4.10)$$

### 4.3.2.1 Intersection of a line with a plane

For cases (a) and (c), the intersection of a line with the cross-section plane is calculated as follows.

Let the normal vector  $\hat{n}$  and origin  $P_0$  define a plane in three-dimensional space. Also let  $P_{in}$  and  $P_{out}$  be two points which are on different sides of this plane.

Let  $P'$  represent the position vector of the intersection point. Therefore,  $P'$  is a point on both the plane and the line. Writing in vector form,

$$(P' - P_0) \cdot \hat{n} = 0 \quad (4.11)$$

The parametric line equation is written

$$P' = t(P_{in} - P_{out}) + P_{out} \quad (4.12)$$

where  $t$  is the line equation parameter with

$$0 < t < 1$$

Combining Equation 4.12 with Equation 4.11

$$\begin{aligned} (t(P_{in} - P_{out}) + P_{out} - P_0) \cdot \hat{n} &= 0 \\ t(P_{in} - P_{out}) \cdot \hat{n} &= (P_0 - P_{out}) \cdot \hat{n} \\ t &= \frac{(P_0 - P_{out}) \cdot \hat{n}}{(P_{in} - P_{out}) \cdot \hat{n}} \end{aligned} \quad (4.13)$$

Finally, intersection point ( $P'$ ) is calculated by replacing  $t$  in Equation 4.12.

$$P' = \frac{(P_0 - P_{out}) \cdot \hat{n}}{(P_{in} - P_{out}) \cdot \hat{n}} (P_{in} - P_{out}) + P_{out} \quad (4.14)$$

#### 4.3.2.2 Processing intersection data

When an edge lies exactly on the cross-section plane (Figure 4.4e), it appears twice in the set of intersection edges, due to the fact that all the edges are shared by exactly two facets. The resultant intersection edges are checked, and any redundant edges are removed from the cross-section profile.

### 4.3.3 Generator Curve of a Turning Workpiece

In turning processes, by convention the machining takes place in X-Z plane of the machine tool coordinates, where X-axis is the radial coordinate, and Z-axis is the axial coordinate. This plane is called the machining-plane (Figure 4.2).

Therefore, the machining-plane is defined in the machine-tool coordinate system as

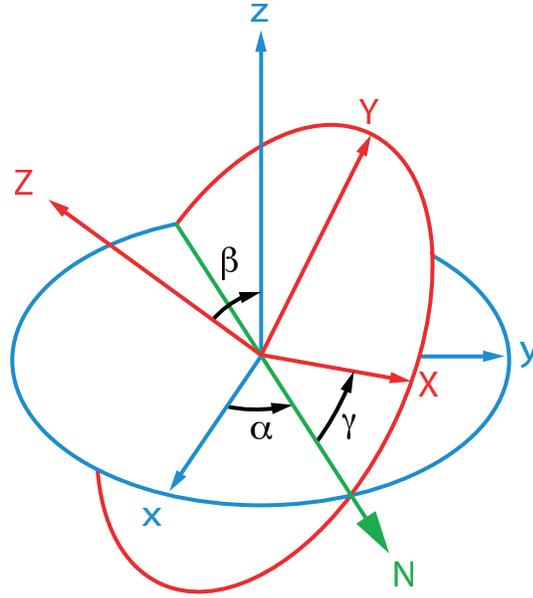
$$P_0 = (0, 0, 0)$$
$$n = \{0, 1, 0\}^T$$

where  $P_0$  is the point of origin and  $n$  is the normal vector.

If the CAD data supplied by the user does not follow this convention, the input data is modified by a model transformation. The workpiece coordinate system is re-oriented using rotations and translations, before the generator curve is extracted.

#### 4.3.3.1 Model transformation

Using the *Euler Angles* convention (Figure 4.5) rotations in the space are described by three angles  $(\alpha, \beta, \gamma)$ . The rotation matrix is written as



**Figure 4.5:** Euler angles [19]

$$R = \begin{bmatrix} \cos \beta \cos \gamma & -\cos \alpha \sin \gamma + \sin \alpha \sin \beta \cos \gamma & \sin \alpha \sin \gamma + \cos \alpha \sin \beta \cos \gamma \\ \cos \beta \sin \gamma & -\cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\sin \alpha \cos \gamma + \cos \alpha \sin \beta \sin \gamma \\ -\sin \beta & \sin \alpha \cos \beta & \cos \alpha \cos \beta \end{bmatrix} \quad (4.15)$$

Therefore, the vertices of the solid model is transformed by

$$\begin{Bmatrix} X \\ Y \\ Z \end{Bmatrix} = R \begin{Bmatrix} x \\ y \\ z \end{Bmatrix} + \begin{Bmatrix} a \\ b \\ c \end{Bmatrix} \quad (4.16)$$

where  $a, b$  and  $c$  are the translational distances in machine–tool coordinate system  $(X, Y, Z)$  and  $R$  is the rotational transformation matrix given in Equation 4.15.

## 4.4 Tool Model

### 4.4.1 Parametric Tool Modeling

Primary geometry of cutting tools is defined by standard types such as diamond, round or rectangle. The exact tool geometry is defined by parameters such as the nose radius ( $r_\epsilon$ ), nose angle ( $\epsilon_n$ ), and tool size ( $l$ ).

Parametric tool model follows the same tool selection concept. First, the primary tool type is selected by the user from a list of standard geometries (Table 2.1). Depending on the tool geometry chosen, a set of parameters is used to generate the cutting edge geometry. The oblique angles such as side rake angle ( $\alpha_f$ ), back rake angle ( $\alpha_p$ ) and approach angle ( $\kappa_r$ ) are imposed by the tool holder geometry.

Tool and tool holder specifications are available in manufacturer catalogues.

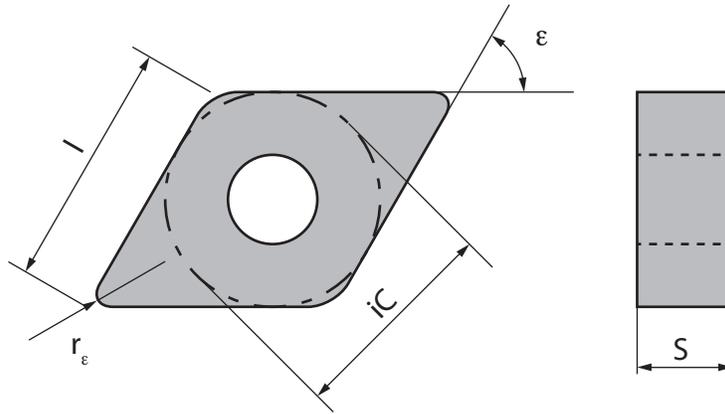
The cutting edge contour is constructed using the input parameters. As an example, the tool generation process for rhombic inserts is given in the following section. Rhombic inserts cover all the diamond shaped cutters with varying nose angles. (Types C, D, E, F, M and V in Table 2.1)

#### 4.4.1.1 Rhombic inserts

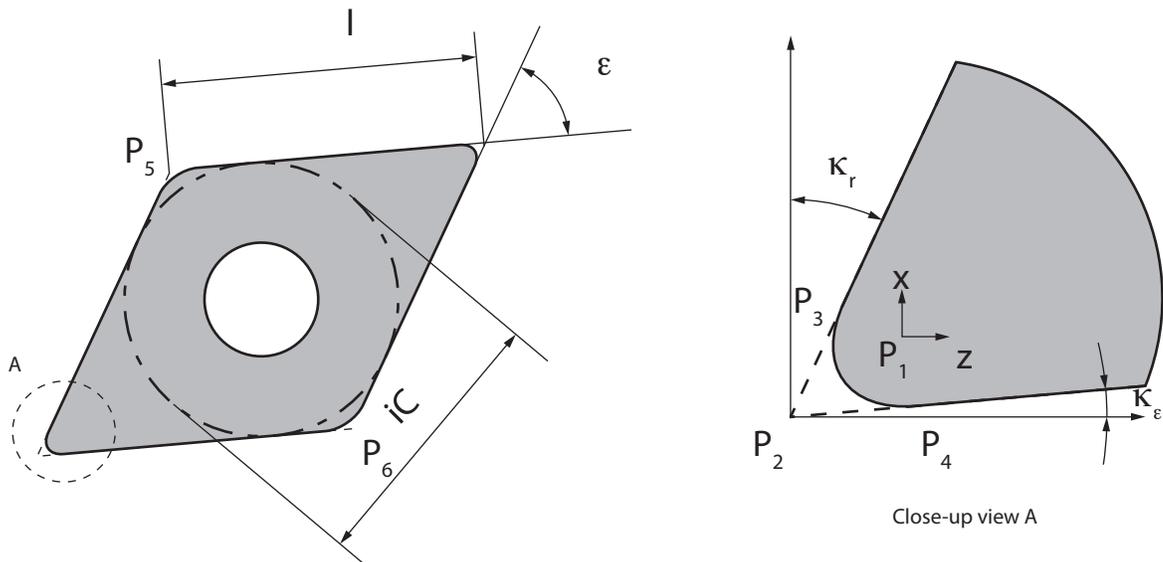
Rhombic inserts are diamond shaped tools, which are widely used for radial, axial, and contour turning. They are classified into smaller groups according to their nose angle ( $\epsilon_n$ ).

By defining the nose angle as one of the parameters, a general model for rhombic tools is developed. The parameters of the model are as follows (Figure 4.6):

- $\epsilon_n$ : Nose angle
- $l$ : Tool edge length
- $r_n$ : Nose radius
- $\kappa_r$ : Approach angle



**Figure 4.6:** Parameters of a rhombic insert



**Figure 4.7:** Definition points for rhombic inserts

#### 4.4.1.2 Definition points

Points of the cutting edge must be defined, including any point which does not directly contribute to the geometry, but is used as a tool driven point.

For example, point  $P_2$  in Figure 4.7, which is the apparent intersection of the major and minor cutting edges, is included in the model, although it is not a part of the cutting tool edge geometry. This point is commonly used as a tool-driven point due to its ease of detection during part setup.

Let  $P_1$  be the origin of the tool coordinate system. The remaining control points are defined as

follows (Figure 4.8):

$$\begin{aligned}
P_1 &= P(0,0) \\
P_2 &= P\left(\cos\left(\frac{3\pi}{2} - \frac{\varepsilon}{2} - \kappa_r\right) \frac{r_\varepsilon}{\sin\left(\frac{\varepsilon}{2}\right)}, \sin\left(\frac{3\pi}{2} - \frac{\varepsilon}{2} - \kappa_r\right) \frac{r_\varepsilon}{\sin\left(\frac{\varepsilon}{2}\right)}\right) \\
P_3 &= P(\cos(\pi - \kappa_r) r_\varepsilon, \sin(\pi - \kappa_r) r_\varepsilon) \\
P_4 &= P\left(\cos\left(\kappa_\varepsilon - \frac{\pi}{2}\right) r_\varepsilon, \sin\left(\kappa_\varepsilon - \frac{\pi}{2}\right) r_\varepsilon\right) \\
P_5 &= P\left(P_{2x} + \cos\left(\frac{\pi}{2} - \kappa_r\right) L, P_{2y} + \sin\left(\frac{\pi}{2} - \kappa_r\right) L\right) \\
P_6 &= P\left(P_{2x} + \cos\left(\frac{\pi}{2} - \kappa_r - \varepsilon\right) L, P_{2y} + \sin\left(\frac{\pi}{2} - \kappa_r - \varepsilon\right) L\right)
\end{aligned}$$

The limit angles of the arc defining the nose radius are given by

$$\begin{aligned}
\theta_1 &= \pi - \kappa_r \\
\theta_2 &= \kappa_\varepsilon - \frac{\pi}{2} \\
\kappa_\varepsilon &= \frac{\pi}{2} - (\kappa_r + \varepsilon_n)
\end{aligned}$$

where, (Figure 4.7)

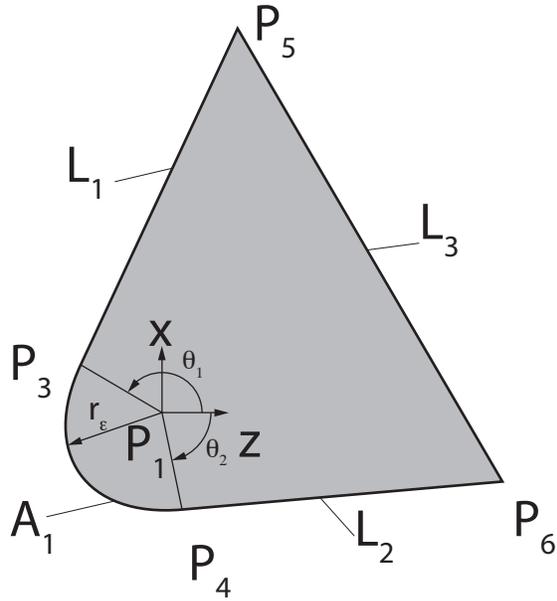
$\kappa_r$  is the approach angle defined by the user

$\kappa_\varepsilon$  is the angle between the secondary edge and Z axis

$\varepsilon_n$  is the nose angle that defines the tool type

#### 4.4.1.3 Cutting edge construction

The cutting edge is generated using the entities defined in Section 4.2. For simplicity, only one half of tool edge is modeled, since practically the depth of cut in turning is much smaller than the tool edge length ( $l$ ). Entities of the tool contour are defined as (Figure 4.8):



**Figure 4.8:** Geometric definition of a rhombic tool

$$A_1 = A(P_1, r_\epsilon, \theta_1, \theta_2)$$

$$L_1 = L(P_3, P_5)$$

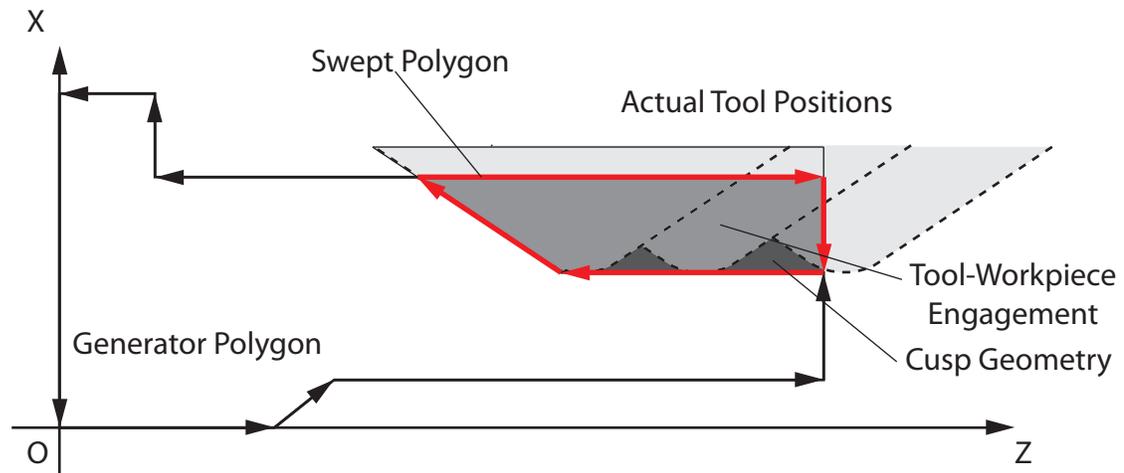
$$L_2 = L(P_4, P_6)$$

$$L_3 = L(P_5, P_6)$$

#### 4.4.2 User Defined Tool Geometry

Any tool geometry that can be converted to the polycurve representation, can be used as the cutting tool. This allows the use of arbitrary tool geometries, provided that the CAD data is available.

When a solid model of the tool is provided, the cross-section of the tool on the machining-plane is obtained using the intersection method explained in Section 4.3.2. Alternatively, if the two-dimensional drawing of the tool is provided, CAD data is reformatted into polycurve representation.



**Figure 4.9:** Use of swept polygon for boolean operation [9]

## 4.5 Tool–Workpiece Engagement Model

### 4.5.1 Boolean Based Tool–Workpiece Engagements

Li et al. [9] proposed to use boolean operations on polygons to simulate turning process, where both the workpiece and the tool–swept area are defined with polygons. Tool–workpiece engagements are calculated as the intersection of two polygons.

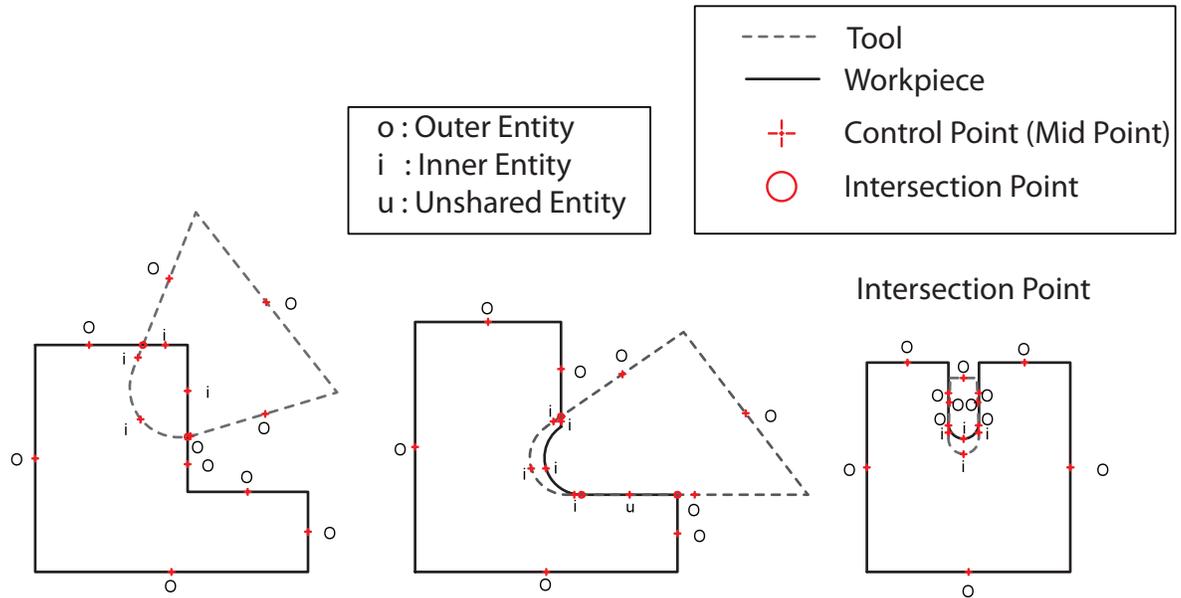
The main disadvantage of this model is that, the arcs are discretized by lines. When a tool swept polygon is used as one of the operands the cusp region is ignored (Figure 4.9). As a result, the number of elements in the resulting in–process workpiece model does not increase significantly. However, in virtual turning, the simulation is run at spindle period intervals. Feed marks left on the workpiece are generated, therefore, a large number of elements is required by the polygon representation.

The boolean operations are extended for profiles having circular entities to reduce the number of elements needed in tool–workpiece engagement maps, and in–process workpiece.

#### 4.5.1.1 Boolean operations on polycurves

The following properties on the operands are assumed.

Operand profiles



**Figure 4.10:** Entity classification for boolean algorithm

- Are closed curves without any self-intersections, which is true for the cross-section of any valid solid object.
- Can have holes.
- Can be convex or concave.

The main steps of the boolean algorithm are as follows:

1. Calculate all occurring intersections between the entities, by running intersection algorithms for Line–Line, Line–Arc and Arc–Arc intersections. A review of these intersection algorithms is presented in Section A.1.
2. Split the entities at points of intersection.
3. Classify each entity of one region as being in, out, or on the other region (see Figure 4.10).

The classification process is as follows:

*Entity classification* Once all the entities are split at the intersection points, each entity is classified as *outer*, *inner* or *unshared*. A fourth class, *shared*, is possible with general boolean algorithms but this class does not occur in machining simulations as the tool profile can not be located inside the workpiece.

The classification is performed by choosing a test–point on the entity, which is chosen as the mid–point, for convention. Entities of the cutting tool profile are tested against the in–process workpiece profile and vice versa. This test classifies the entities as follows:

**Outer** Entity is outside the other region

**Inner** Entity is inside the other region

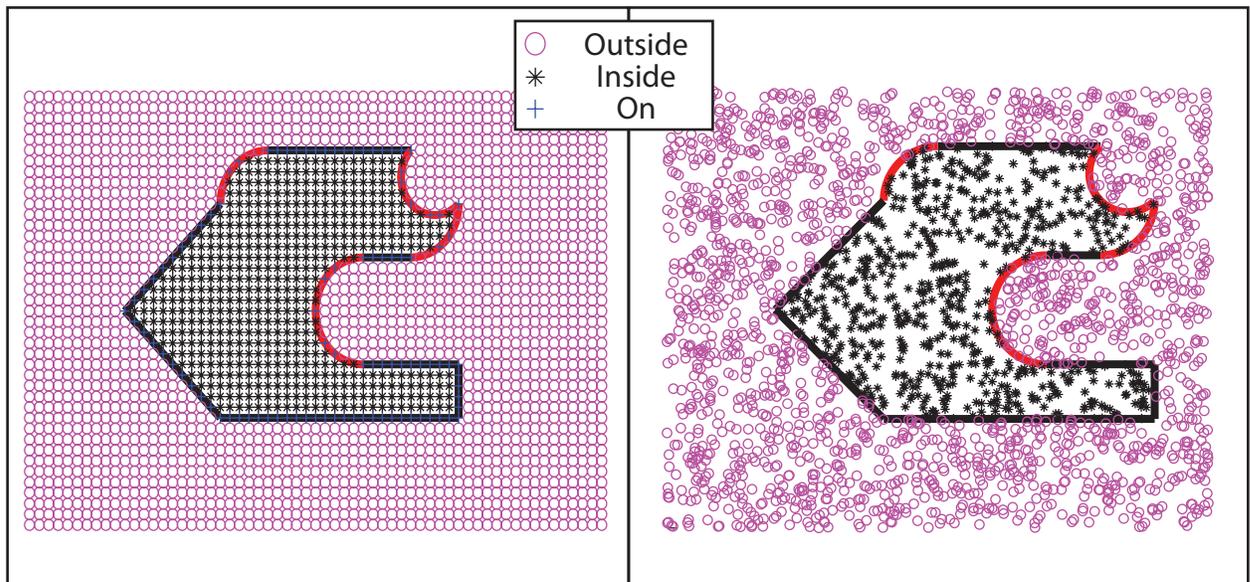
**Shared** Entity is on the boundary of the other region and both regions lie on the same side of the entity

**Unshared** Entity is on the boundary of the other region and the regions lie on different sides of the entity

*In-Region test* Point in–region test checks whether a given point lies outside, inside or on a given region.

An arbitrary ray originating from within a closed boundary, crosses the closed boundary at an odd number of times. Similarly, an arbitrary ray originating from outside a closed boundary, crosses the boundary at an even number of times. Location of the test point is determined by counting the number of intersections between the test ray and the region boundary.

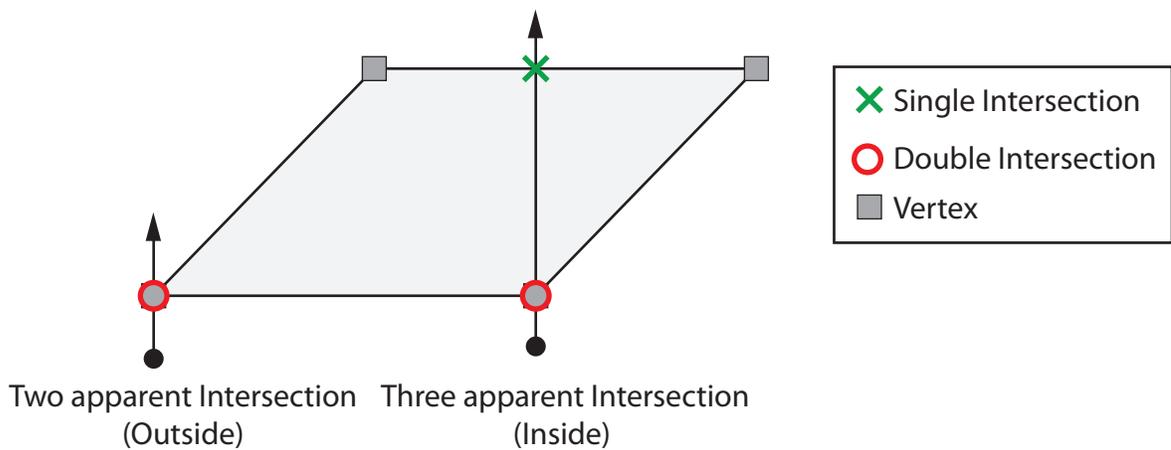
In–region test yields accurate results as long as the generated ray does not intersect with any vertex. When a polygon is represented by its vertices, which is the most common application, a ray passing through a vertex will apparently intersect the polygon twice, resulting in the test to fail (Figure 4.12). To tackle this problem, a test ray which crosses a vertex is assumed to be slightly at the side of the vertex. In this case the ray is shifted by a sufficiently small amount ( $\epsilon$ ).



(a) Point in Region Test with Rectangular Grid

(b) Point in Region Test with Random Point

**Figure 4.11:** In-region test with 250 test points



**Figure 4.12:** Errors in in-region test due to test-ray crossing vertices

The results of the in-region test are verified in Figure 4.11, where 250 points were tested against a sample region. In Figure 4.11a, test points form a regular grid. In Figure 4.11b, the test points are generated randomly.

*Intersection* The engagement profile is the intersection of the current tool profile with the current in-process workpiece cross-section.

The intersection of the two profiles is defined as the collection of entities that are classified as either inner or shared from Region A, and the ones that are classified as inner from Region B (Figure 4.13a).

$$A \cap B = A(\text{inner}) \cup A(\text{shared}) \cup B(\text{inner}) \quad (4.17)$$

*Difference* The resulting in-process workpiece is the last workpiece profile minus the current engagement. Therefore, the in-process workpiece profile is obtained by calculating the difference of in-process workpiece cross-section and the cutting edge profile.

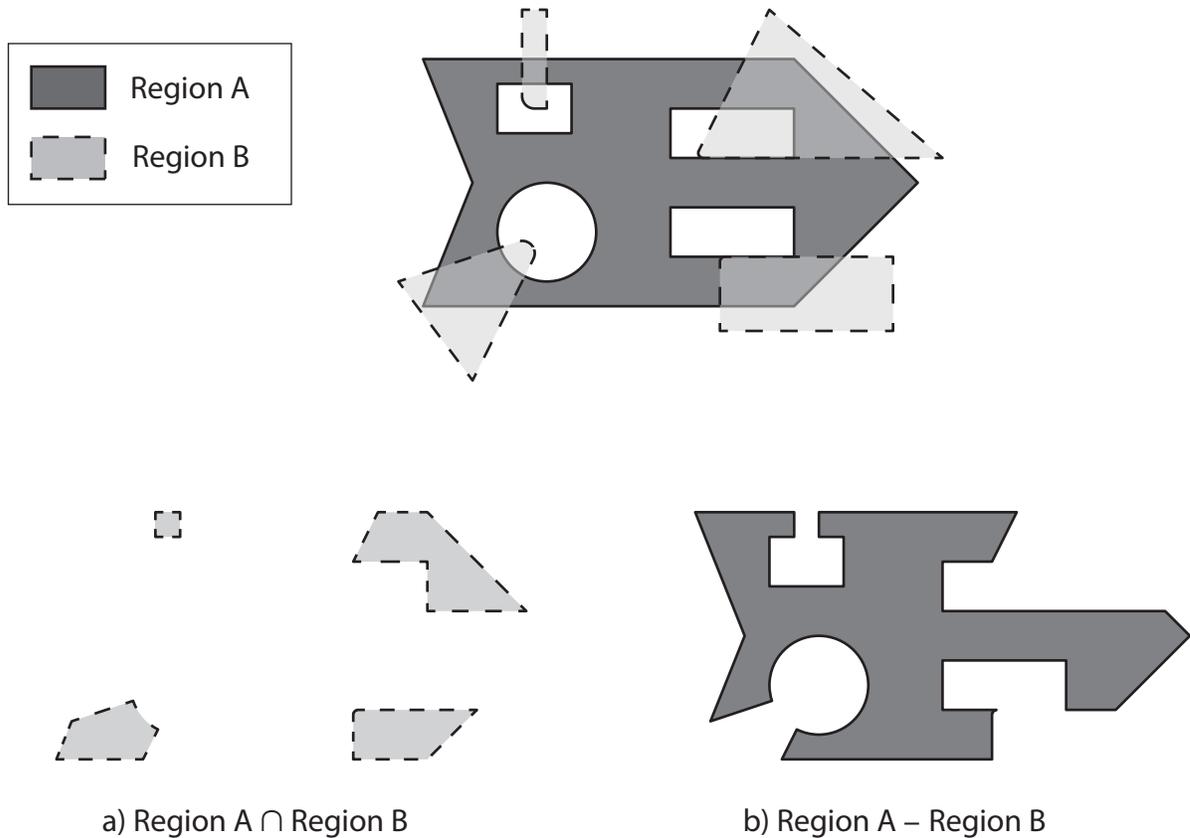
The difference of two regions is the collection of entities marked as either outer or unshared from Region A and the entities marked as inner from Region B (Figure 4.13b).

$$A - B = (A(\text{outer}) \cup A(\text{unshared})) \cup B(\text{inner}) \quad (4.18)$$

#### 4.5.1.2 Numerical robustness

One of the major problems of the boolean algorithms is the floating point errors. Computer algorithms are not capable of representing real numbers exactly due to memory limitations and their binary nature [37], therefore, care must be taken during implementation, especially when defining relational operations.

Since real numbers cannot be expressed exactly, the relational operations are defined with a tolerance as shown in Table 4.1.



**Figure 4.13:** Boolean operations supporting holes in the workpiece model

**Table 4.1:** Relational operations with real and floating point numbers

Relational Operation	Real Numbers	Floating Point
$>$	$A > B$	$A > B + \epsilon$
$<$	$A < B$	$A < B - \epsilon$
$=$	$A = B$	$ A - B  < \epsilon$

The geometrical values can be limited to a given precision, such as the maximum number of digits allowed by input files (i.e. stl file). All digitally calculated physical values are rounded off to this chosen precision.  $\epsilon$  must be chosen as a value sufficiently smaller than the precision of the simulation, and larger than the data-type precision.

## 4.5.2 Processing Engagement Profiles

The engagement profiles obtained from the boolean operation are processed to obtain the parameters required by the force and stability models.

### 4.5.2.1 Uncut chip area

The area enclosed by an engagement map is calculated using the Green's Theorem.

*Green's theorem:* Green's theorem expresses a double integral as a line integral around its boundary. A very common application of this theorem is the calculation of the area enclosed by a closed profile. The general form of the Green's theorem is expressed as:

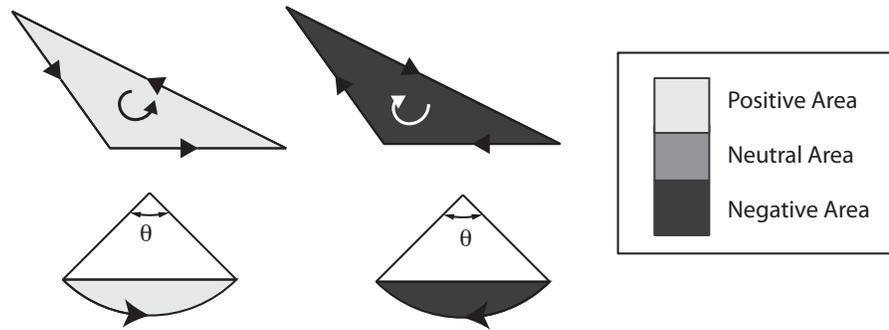
$$\int \int_R \left( \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} \right) dx dy = \oint_c (f dy - g dx) \quad (4.19)$$

For discrete points, such as the ones defining a closed polygon, the Green's integral (Equation 4.19) takes the form of a summation.

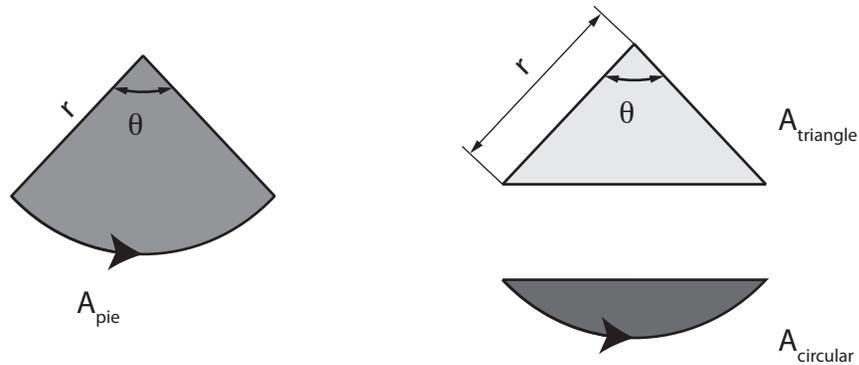
$$A_c = \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) \quad \text{where } n + 1 = 1 \quad (4.20)$$

where  $n$  is the number of vertices on the polygon, and  $(x_i, y_i)$  are the coordinates of the  $i^{\text{th}}$  vertex.

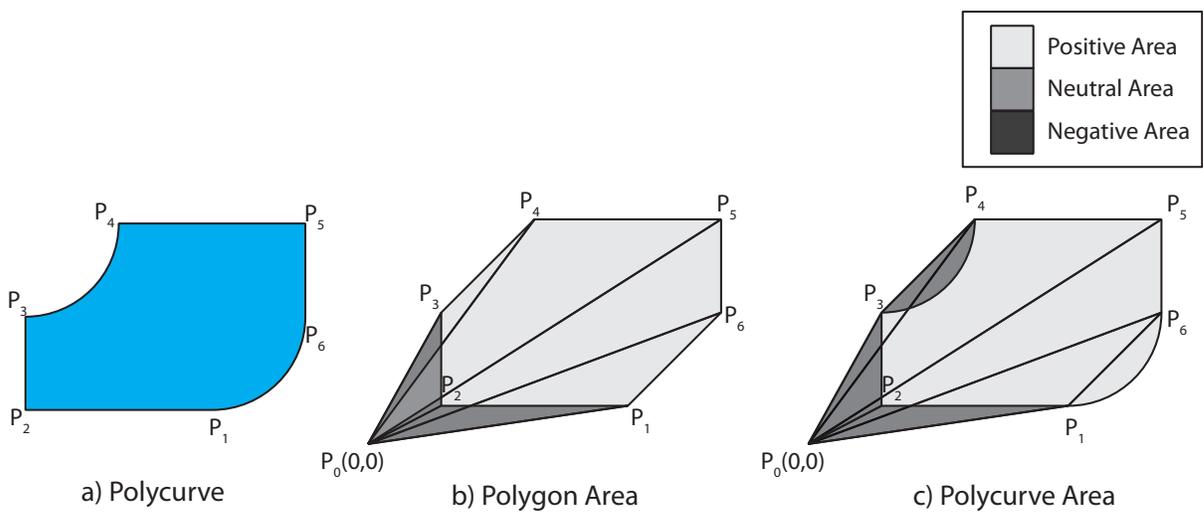
This expression gives the area enclosed by a polygon, calculated as a summation of signed triangular areas. Each triangle is formed by the origin of the coordinate system and two consecutive vertices of the polygon (Figure 4.16b).



**Figure 4.14:** Positive and negative direction conventions for Green's Theorem



**Figure 4.15:** Area of the arc entity



**Figure 4.16:** Area calculation by Green's theorem

*Area of polycurves:* The area of the engagement profile represented by a polycurve (Figure 4.16a) is calculated in two steps. The first step is to evaluate the polygon area, excluding the arc elements (Figure 4.16b), which is calculated using Equation 4.20. Conventionally, counter-clockwise direction is chosen as positive.

In the second step, the contribution of each circular region is calculated and added to the area of the polygon following the same positive direction convention (Figure 4.16c).

The area of the circular region is the area of the pie shaped region minus the triangular region, as shown in Figure 4.15. The circular region's area is calculated as follows:

$$\begin{aligned}
 A_{\text{circular}} &= A_{\text{pie}} - A_{\text{triangle}} \\
 A_{\text{circular}} &= \frac{\theta}{2\pi} \pi r^2 - r^2 \frac{\sin \theta}{2} \\
 A_{\text{circular}} &= r^2 \left( \frac{\theta - \sin \theta}{2} \right) \tag{4.21}
 \end{aligned}$$

The total area can be expressed as the sum of the polygon area and the sum of the circular regions.

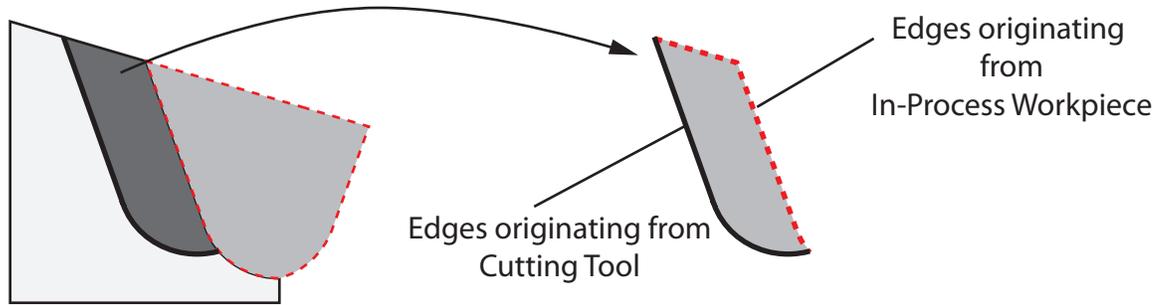
$$\begin{aligned}
 A_c &= A_{\text{polygon}} + A_{\text{circular}} \\
 A_c &= \frac{1}{2} \sum_{i=1}^n (x_i y_{i+1} - x_{i+1} y_i) + \sum_{j=1}^m \left( r_j^2 \left( \frac{\theta_j - \sin \theta_j}{2} \right) \right) \tag{4.22}
 \end{aligned}$$

where  $n$  is the number of vertices in the polygon and  $m$  is the number of arc elements in the polycurve.

Due to the positive direction convention, the uncut chip area can be negative. Since there are no self-intersections in the engagement profile, the absolute value of the uncut chip area is used.

#### 4.5.2.2 Cutting edge length

Edges of the engagement profile are marked as originating from either the tool profile or the in-process workpiece. Edges originating from the tool profile constitute the cutting edge currently in



**Figure 4.17:** Entities originating from tool curve are in cut

cut, therefore, finding the length of the cutting edge is just a matter of calculating the total length of these edges.

For a linear edge, the coordinates of the end points are known, therefore the edge length is calculate as follows:

$$L_{line} = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (4.23)$$

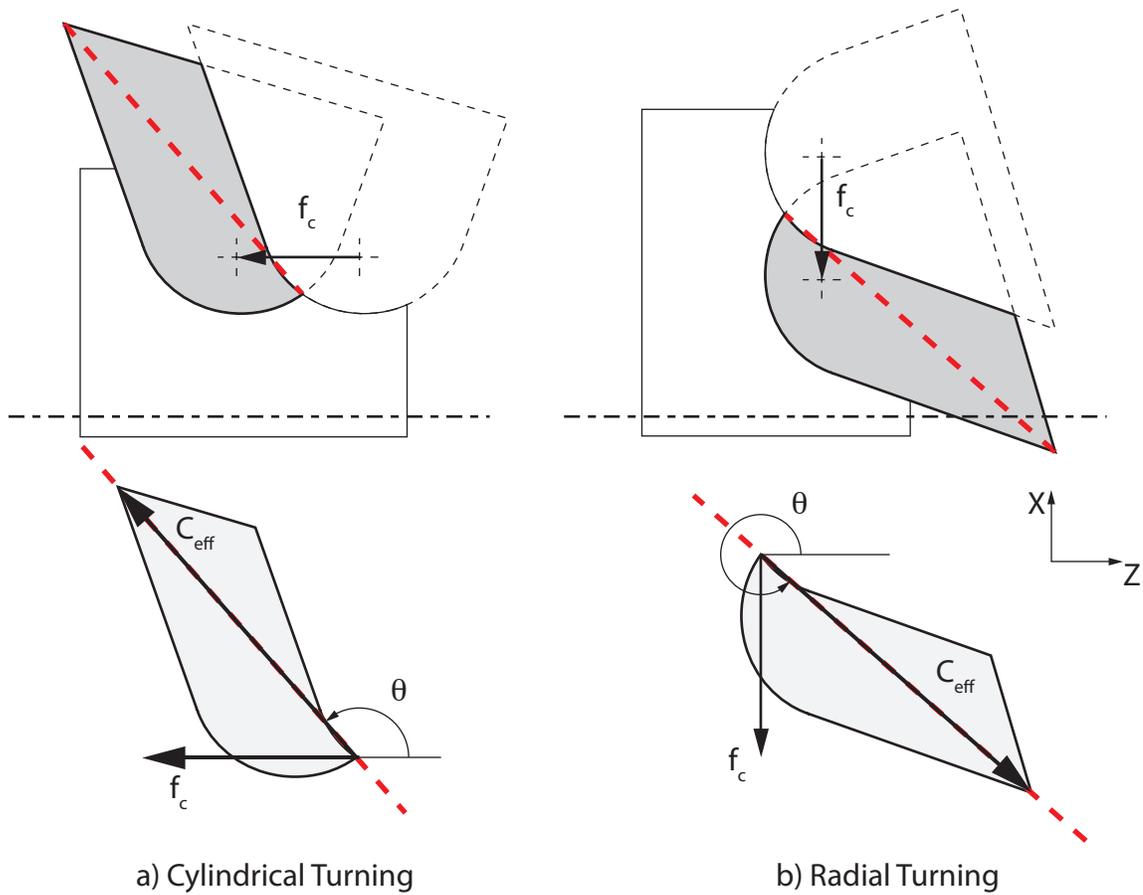
The length of an circular edge is calculated by using the limiting angles and the radius values.

$$L_{arc} = (\theta_1 - \theta_0) r \quad (4.24)$$

#### 4.5.2.3 Approximate chord angle

The approximate chord is the major diagonal of the uncut chip area [15]. The approximate chord angle ( $\theta$ ) is the angle between the approximate chord and the positive Z axis of the machine tool coordinate system. However, during software implementation, this definition is insufficient because the direction of the effective chord is undetermined. To completely define the approximate chord, a convention for its direction is assumed.

The orientation of the chord is given by the line connecting the vertices of the engagement profile that are furthest away from each other, and the direction is chosen, so that it forms an acute angle with the feed vector (Figure 4.18).



**Figure 4.18:** Direction convention for the effective chord vector

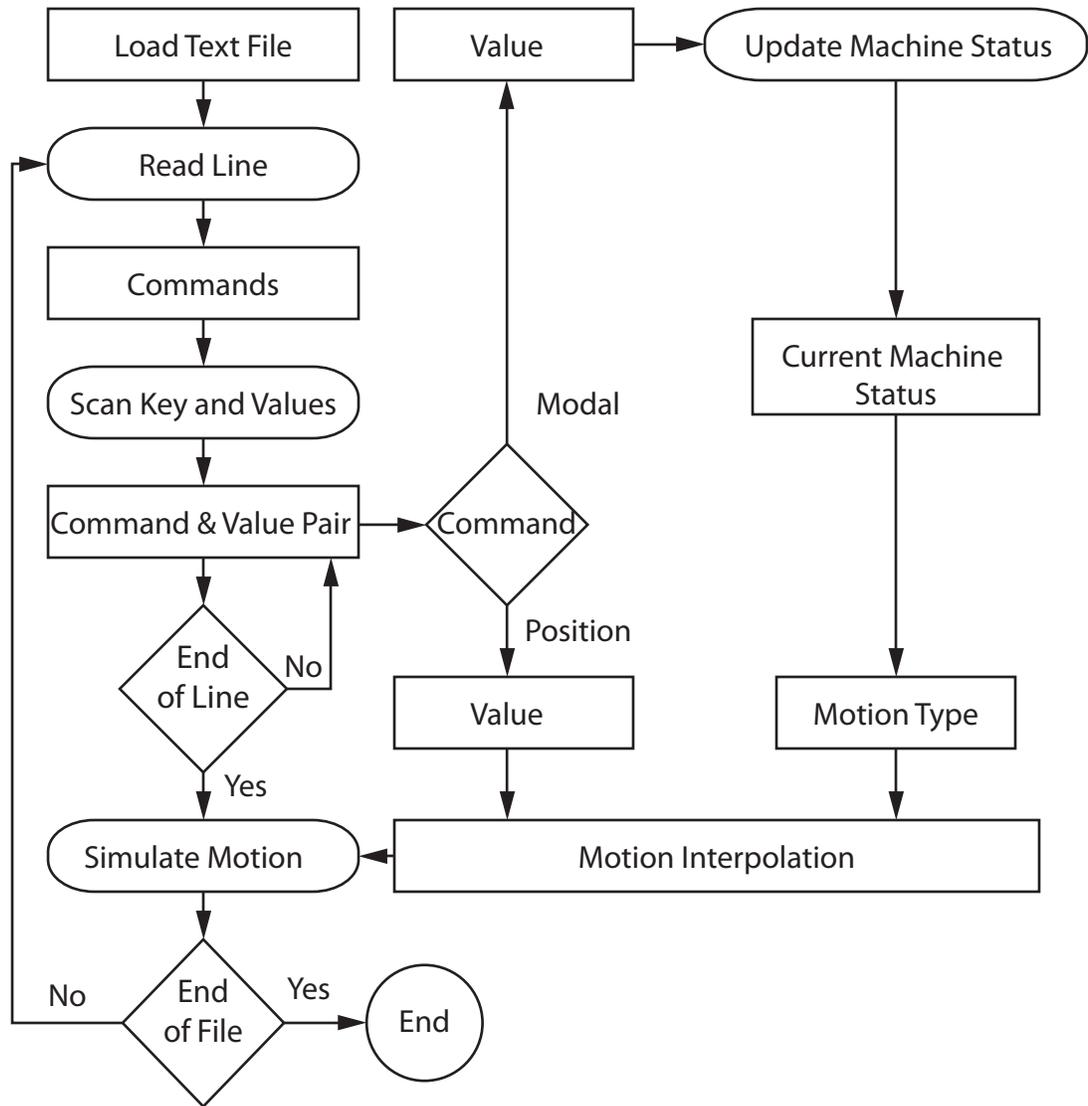
$$\{f_c\} \cdot \{C_{\text{eff}}\} > 0 \quad (4.25)$$

where,  $\{f_c\}$  is the feed vector and  $\{C_{\text{eff}}\}$  is the approximate chord vector.

This convention ensures that the approximate chord vector points away from the nose of the tool, allowing the chord angle ( $\theta$ ) to be calculated systematically.

## 4.6 Parsing NC-Code

NC-codes are generated by CAM systems in a machine independent form, namely the APT files. The machine independent codes are converted to machine specific G-codes using post-processors. Turning simulations have the ability to use either APT or G-codes.



**Figure 4.19:** Flow-chart for NC-code parser algorithm

For virtual turning, G-codes are chosen as the user input. A basic, single purpose parser, which reads NC programs stored as text files, is developed for the GN6 series controller on the Hardinge Superslant lathe. The flow-chart for the parser algorithm is shown in Figure 4.19. Compatible commands, and expected values are given in Section A.3.

## 4.6.1 Spindle Speed and Cutting Speed

Spindle speed ( $n$ ) is the rotational speed of the machine tool spindle, usually expressed in revolutions per minute. Cutting speed ( $V_c$ ), or the surface speed is the tangential velocity of the rotating workpiece at the tool contact point. Modern CNC lathes support both, Constant Spindle Speed and Constant Surface Speed modes.

*Constant spindle speed mode* When the constant spindle speed mode is active, the controller keeps rotational speed of the spindle constant.

In contour operations, the tool is free to move in both axial and radial directions. Since the cutting speed is dependent on the tool location in radial axis, it should be calculated for each step of the simulation. Cutting speed is calculated as follows:

$$\begin{aligned} V_c &= n\pi d \\ &= n\pi x_{avg} \\ &= n\pi \frac{(x_{start} + x_{end})}{2} \end{aligned} \quad (4.26)$$

where  $V_c$  is cutting speed in [mm/min],  $n$  is spindle speed in [rpm].  $x_{start}$  and  $x_{end}$  are the starting and ending locations of the tool in the radial direction of the machine tool coordinate, given in diameter coordinates.

*Constant surface speed mode* The CNC controller adjusts the spindle speed to provide a constant cutting speed. Rearranging Equation 4.26, the spindle speed is calculated for a given tool position as follows:

$$\begin{aligned}
n &= \frac{\pi d}{V_c} \\
n &= \frac{\pi x_{avg}}{V_c} \\
n &= \frac{\pi(x_{start} + x_{end})}{V_c}
\end{aligned}
\tag{4.27}$$

## 4.7 Simulation Schemes

Turning simulations require different levels of complexities for different processes. It is possible for the tool–workpiece engagements to remain constant, or vary in time depending on the tool motion and workpiece geometry. In addition, workpieces can have non axi–symmetric features.

Therefore, simulation schemes that allow for a variable level of complexity in virtual turning are proposed.

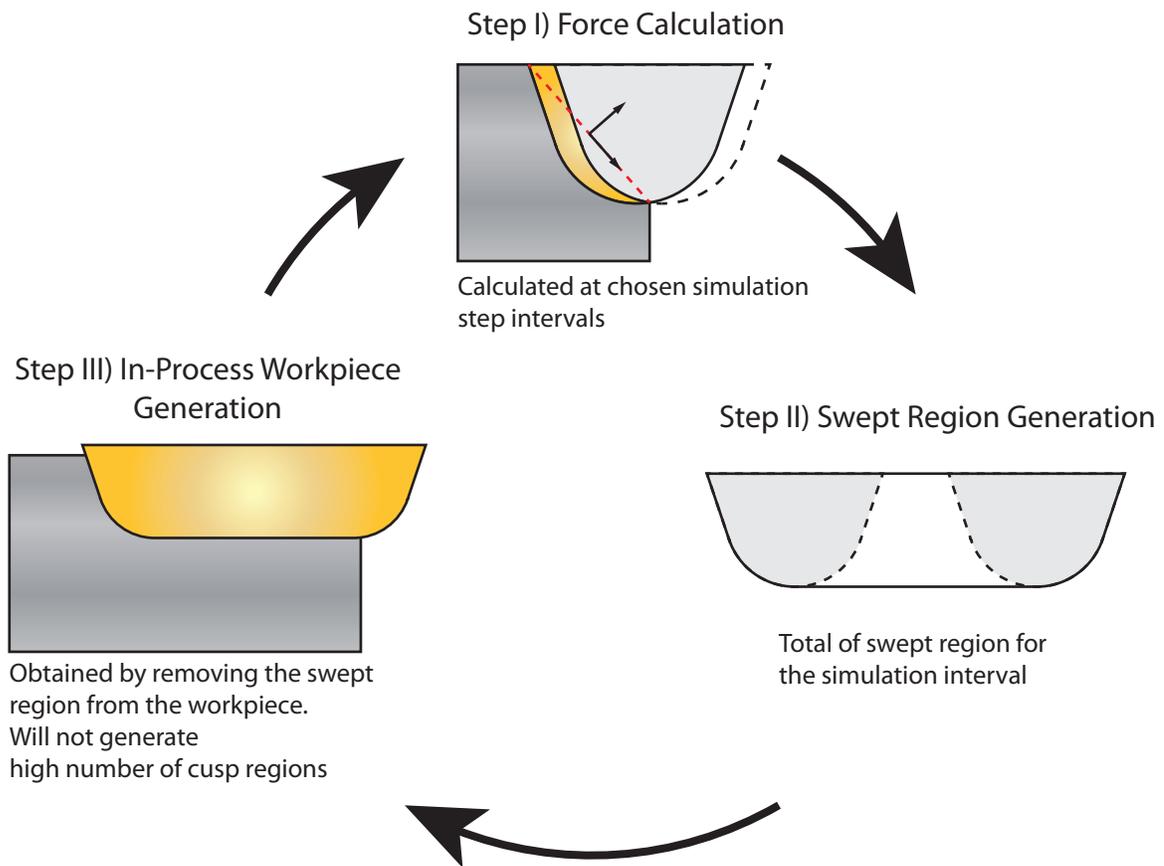
### 4.7.1 Constant Engagement

During linear cutting motions, it is possible that the tool–workpiece engagement remains constant, therefore, it is redundant to simulate the cutting process at spindle period steps.

The entire linear motion command can be simulated by subtracting the tool swept region from the in–process workpiece, which allows the process simulation to be run once for the entire NC-command [9, 5]. The use of tool swept region also generates an in–process workpiece with a smaller number of elements, reducing the memory requirements (Figure 4.9).

The algorithm to evaluate tool–workpiece engagement is detailed as follows (Figure 4.20):

1. For each motion command in the NC-Code, the motion is discretized into spindle period simulation steps. However, the cutting forces are calculated only at every  $n^{th}$  simulation step, where  $n$  is a user-defined value, representing how small or big the time step of the simulation is.
2. After cutting forces are calculated, the area swept by the tool in the next  $n$  steps is generated.



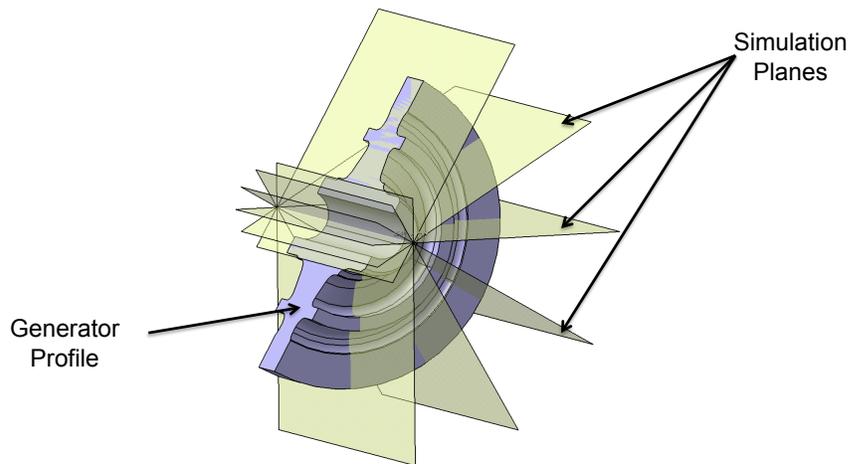
**Figure 4.20:** Simulation scheme for constant engagement

3. The in-process workpiece is generated by using boolean operations between the workpiece and the generated swept area. The process is repeated until the entire tool motion is simulated.

#### 4.7.2 Non Axi-symmetric Workpiece

Simulation of a workpiece having non axi-symmetric features requires simulation steps smaller than the spindle period to capture the changes within one revolution of the spindle.

This is achieved by running the process simulation on multiple simulation planes (Figure 4.21). Each of these simulation planes represents the machining-plane as the spindle rotates. A separate



**Figure 4.21:** Multiple simulation planes per spindle revolution

in-process workpiece profile is stored in memory for each simulation plane, increasing the total computational time linearly .

## 4.8 Summary

In this chapter, a new simulation model based on two dimensional boolean operations is developed. Polygon representation is expanded to include circular edges. This new representation is used to define tool and workpiece cross-section profiles on the machining plane.

Tool-workpiece engagement is calculated as the intersection of tool and workpiece profiles, while the in-process workpiece is defined as their difference. The boolean operations on the proposed geometric representation are developed.

Controller behavior is simulated by parsing the NC program. At each simulation step, tool location is obtained, and engagement profile is calculated by the new tool-workpiece engagement model. Also, simulation schemes which allow the adjustment of simulation resolution is presented.

# 5 Application and Experimental Verification

A process model for contour turning is presented in Chapter 3, and simulation models for virtual turning are presented in Chapter 4. A virtual turning simulation, which integrates these models, is developed into a software application.

In the first section of the chapter, the details of the computer implementation is presented.

In the following sections, the validity of the virtual turning application is verified experimentally. A sample part with a free-form surface geometry is machined to verify the tool-workpiece engagements, cutting forces, and feedrate scheduling. Additionally, face turning experiments are conducted to verify the chatter detection capabilities.

Process simulations are created in the virtual turning application, and the simulated results are compared against the measurements.

## 5.1 Virtual Turning Simulation

The virtual turning application consists of six modules, which are developed in Matlab environment. Generated data is stored and transferred between the modules using the native data format (.mat files).

Simple graphical user interfaces for each module are built using the GUIDE toolbox. Core functions of the system are converted into C language using the CODER toolbox to increase the computational performance.

The structure of the virtual turning application is illustrated in Figure 5.1. The following core Matlab modules represent the developed virtual turning system.

**VTStl** generates the generator curve representation from the solid workpiece geometry, as detailed in Section 4.3. Solid model is reoriented, and intersected with the machining plane to obtain the cross-section.

**VTTool** module generates the tool data from the user input, as detailed in Section 4.4. Tool data includes the cutting edge contour and the oblique angles of the tool holder.

**NCLoader** interprets the NC programs written with G-codes, as detailed in Section 4.6.

**VTTurning** generates the tool-workpiece engagement maps, using boolean algorithms as presented in Section 4.5. Engagement map data includes the chip load geometry and the cutting conditions. VTTurning module is responsible for updating the in-process workpiece geometry during the cut. Visualization of the in-process workpiece, tool motion and engagement profiles, are also handled.

**VTResult** module calculates the cutting forces, power and torque requirements, and determines the chatter stability using the process model presented in Chapter 3.

**OptiSearch** module improves the feeds and speeds, using the initial engagements generated by the virtual turning system.

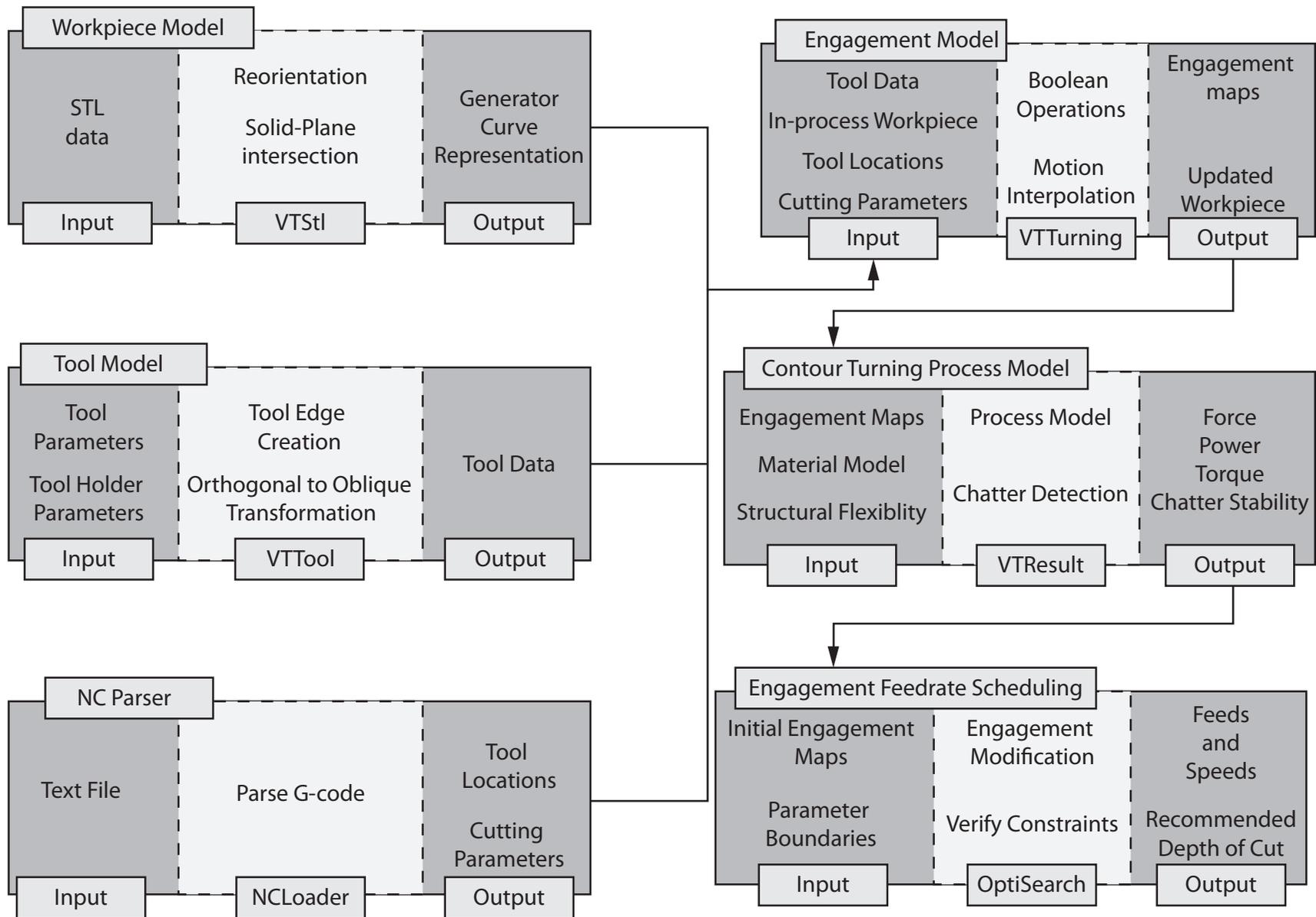
## **5.2 Contour Turning Experiments**

### **5.2.1 Experimental Setup**

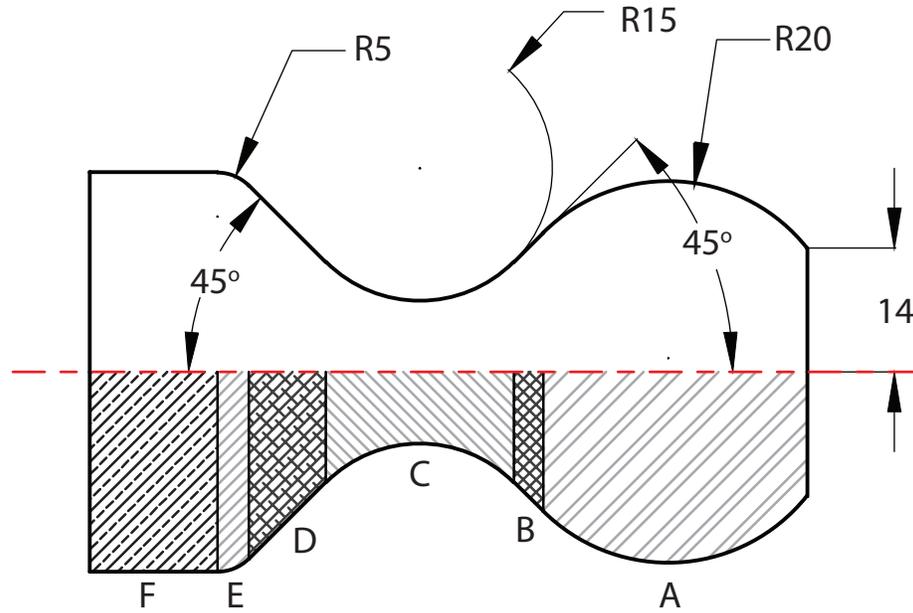
Experiments are conducted on a Hardinge Superslant CNC lathe, which is controlled by a GN 6 Series Fanuc controller. Cutting forces are measured with a three axes Kistler 9121 dynamometer installed on the turret of the lathe.

#### **5.2.1.1 Cutting tools and tool holders**

Sandvik Coromant VNGA 160408 inserts on a DVJNL 2020K 16 tool holder is used for the contour turning experiments. The specifications of the tools and holders are presented in Table 5.1.



**Figure 5.1:** Structure of the virtual turning application



**Figure 5.2:** Initial workpiece geometry for contour turning experiments  
Workpiece material: AISI 1045 Steel

**Table 5.1:** Geometric specifications of the tools used in experiments

		Nose Radius	Edge Length	Approach Angle	Side Rake Angle	Back Rake Angle
Cutting Tool	Tool Holder	$r_n$	$l$	$\kappa_r$	$\alpha_f$	$\alpha_p$
CNMA120404	DCLNL 2020K 12	0.4 mm	12 mm	5°	-6°	-6°
CNMA 120408	DCLNL 2020K 12	0.8mm	12mm	5°	-6°	-6°
CNMA120416	DCLNL 2020K 12	1.6 mm	12 mm	5°	-6°	-6°
VNGA 160408	DVJNL 2020K 16	0.8 mm	16 mm	3°	-4°	-13°

### 5.2.1.2 Workpiece

The initial workpiece is a free-form, AISI 1045 steel part. Initial workpiece geometry is shown in Figure 5.2. Orthogonal cutting data for the AISI 1045 steel is given in Section A.2.

The part geometry includes six geometrical features as shown in Figure 5.2:

A) Convex circular interpolation

- B) Linear cut with negative slope
- C) Concave circular interpolation
- D) Linear cut with positive slope
- E) Convex circular interpolation with small radius (Fillet)
- F) Straight axial cut

Workpiece geometry and the tool–paths are generated in UGS NX software. For Vericut simulations, geometric models are saved in IGES format, while in virtual turning simulations the STL data format is used.

### 5.2.1.3 Cutting parameters

Cutting parameters for contour turning experiment are chosen as

$$a_c = 0.5 \text{ mm} \qquad f_c = 0.1 \frac{\text{mm}}{\text{rev}} \qquad V_c = 120 \frac{\text{m}}{\text{min}} \qquad (5.1)$$

Machining is performed in constant surface speed mode, therefore, the cutting speed ( $V_c$ ) is controlled by the CNC, while the spindle speed ( $n$ ) varies during the process.

### 5.2.2 Tool–Workpiece Engagements

Tool–workpiece engagement maps are generated using both the Vericut API, and the boolean–based engagement model presented in Section 4.5.

The boolean model is verified by comparing the engagement maps from two models. Figure 5.3 shows that the models generate very similar engagement maps. Small discrepancies emerge due to the difference in geometric representation, and tool–motion interpolation.

Both models are found to generate the engagement maps of the simulated process accurately, however, the boolean–based model is significantly faster. Computation times required by the mod-

**Table 5.2:** Computational time required by the engagement models

	Vericut API	Boolean based
Number of engagement maps	776	783
Computation time	90 sec	10 sec

els on a standard desktop computer are compared in Table 5.2. The proposed polycurve tool–part intersection method is nine fold faster in comparison to existing commercial code.

### 5.2.3 Cutting Forces

Cutting forces are simulated using the virtual turning application. Simulated forces match the experimental data closely, as shown in Figure 5.4.

### 5.2.4 Feedrate Scheduling

The cycle time is reduced by using the proposed engagement–based feedrate scheduling method detailed in Section 3.6. Workpiece, cutting tool, cutting parameters and the experimental setup are given in Section 5.2.1.

#### 5.2.4.1 Constraints

Maximum cutting force, torque and power constraints are applied.

$$F_{max} = 500\text{N}$$

$$T_{max} = 20\text{Nm}$$

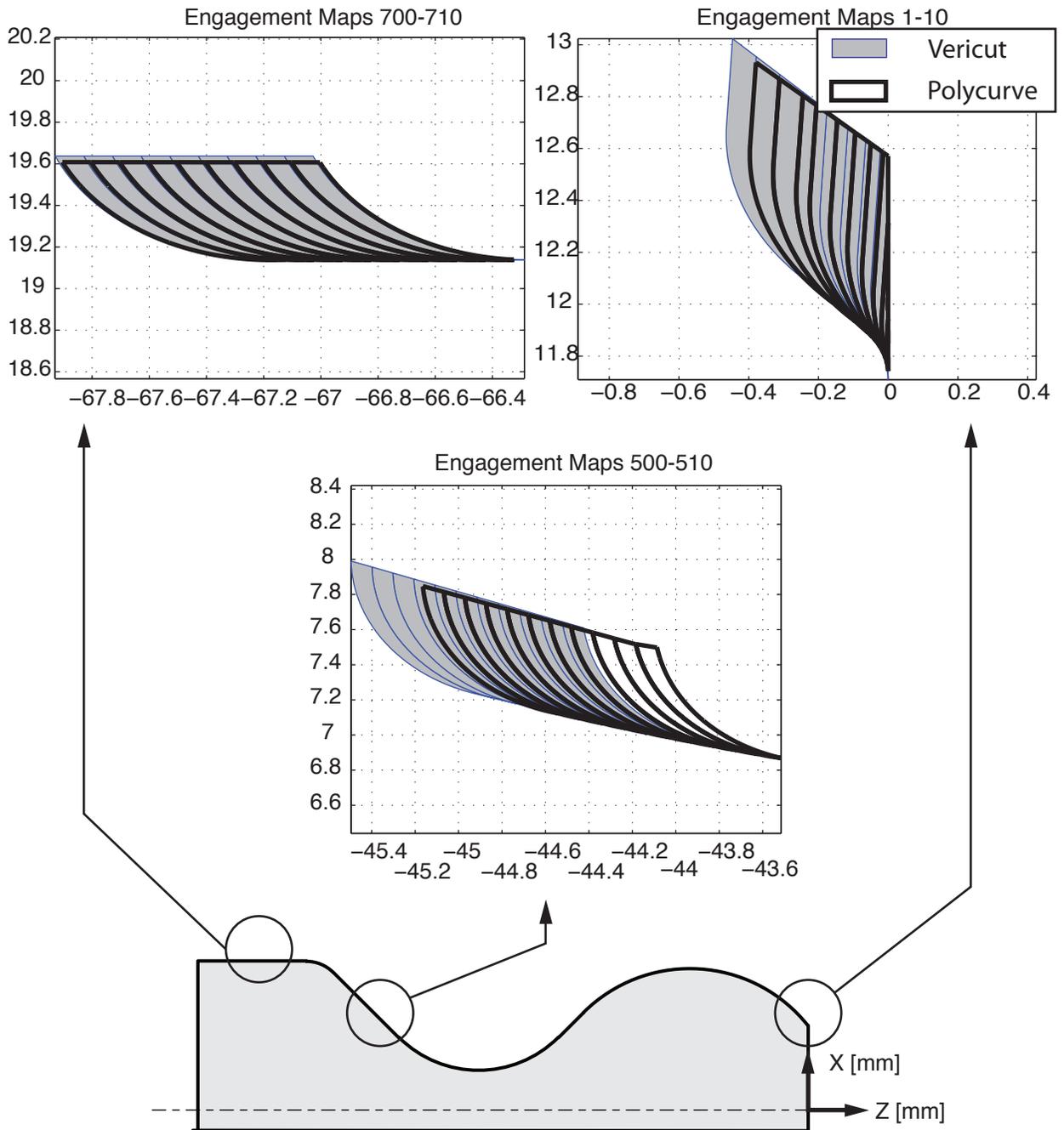
$$P_{max} = 1000\text{W}$$

#### 5.2.4.2 Results

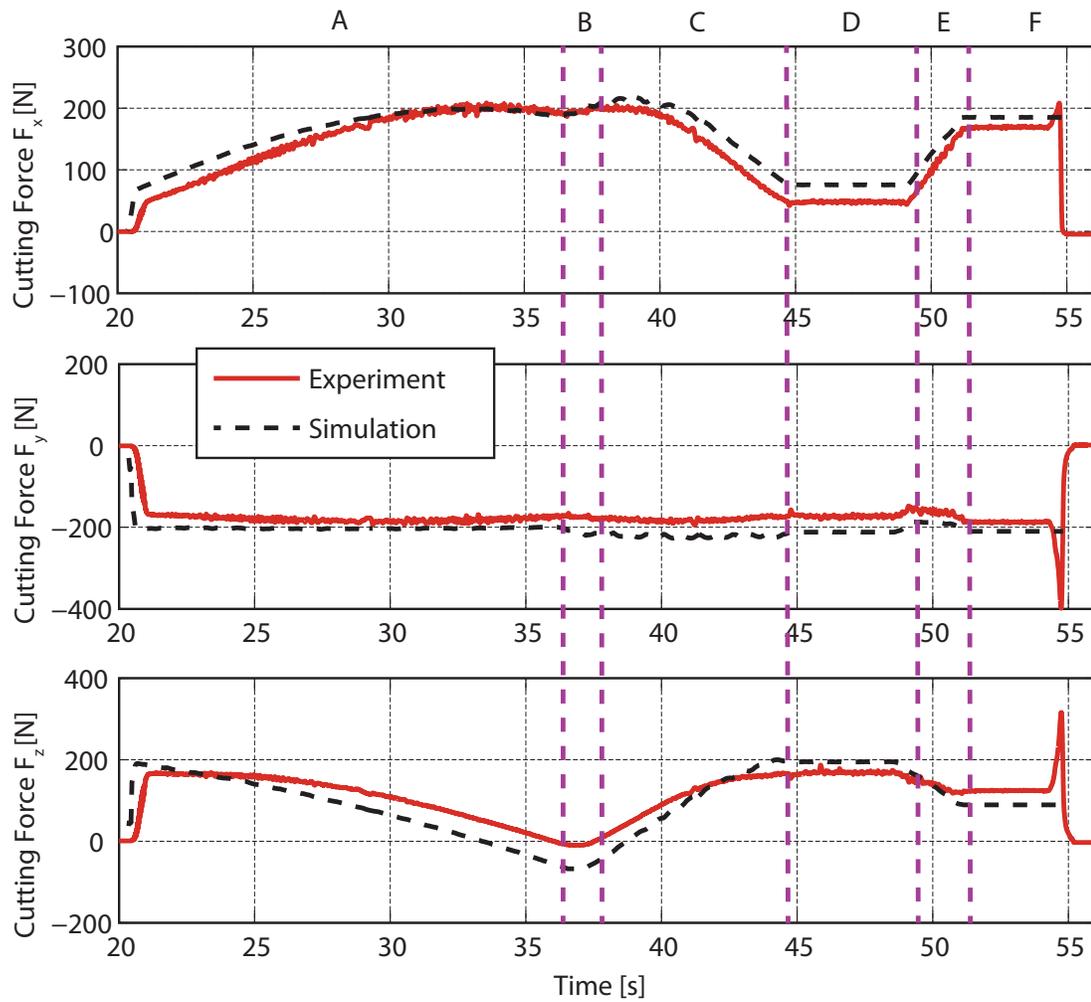
The highest, feasible feedrate for each NC command (Figure 5.2) is searched. The results are shown in Table 5.3.

The total cycle time has been reduced from 33 seconds to 8 seconds after the scheduling operation, resulting in a simulated productivity improvement of approximately four times.

The simulated cutting forces, power and torque values are shown in Figure 5.5. The simula-



**Figure 5.3:** Comparison of engagement model with Vericut Solution  
 See Figure 5.2 for the part geometry. Cutting conditions:  $a_c = 0.5mm$   $f_c = 0.1mm/rev$   
 $V_c = 120m/min$



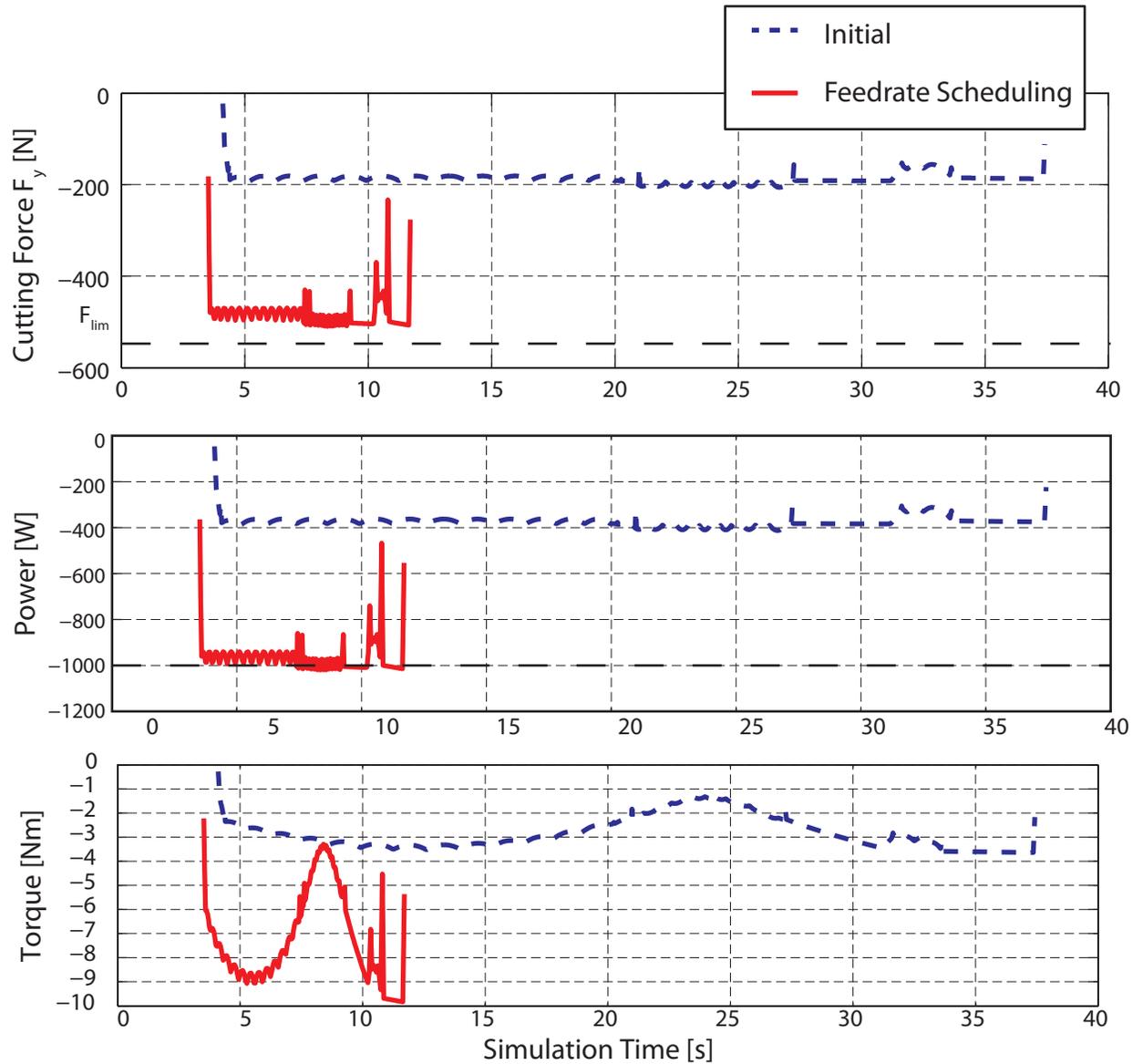
**Figure 5.4:** Cutting forces during contour turning

See Figure 5.2 for the part geometry. Workpiece material: AISI 1045 Steel. Cutting conditions:

$$a_c = 0.5\text{mm} \quad f_c = 0.1\text{mm/rev} \quad V_c = 120\text{m/min}$$

**Table 5.3:** Suggested feedrate values for contour turning experiment

Region	Feedrate [mm/rev]
A	0.41
B	0.41
C	0.37
D	0.42
E	0.46
F	0.43



**Figure 5.5:** Comparison of cutting forces before and after optimization Initial; see Figure 5.4 for cutting conditions. After feedrate scheduling; see Table 5.3 for the feedrate values.

tion shows that tangential cutting force and the power is maximized without violating the chosen constraints.

## 5.3 Chatter Stability Experiments

A series of face turning experiments verify the stability solution, which is presented in Section 3.5. The frequency response of the flexible structure is obtained experimentally by impact hammer testing. The engagements and stability solution by Nyquist criterion are simulated by virtual turning application.

### 5.3.1 Experimental Setup

Sandvik Coromant CNMA 120408 KR inserts on a DCLNL 2020K 12 tool holder is used. Tool specifications are given in Table 5.1.

Workpiece is an AISI 1045 steel bar, with a diameter of 41.275 mm, and an initial length of 150 mm.

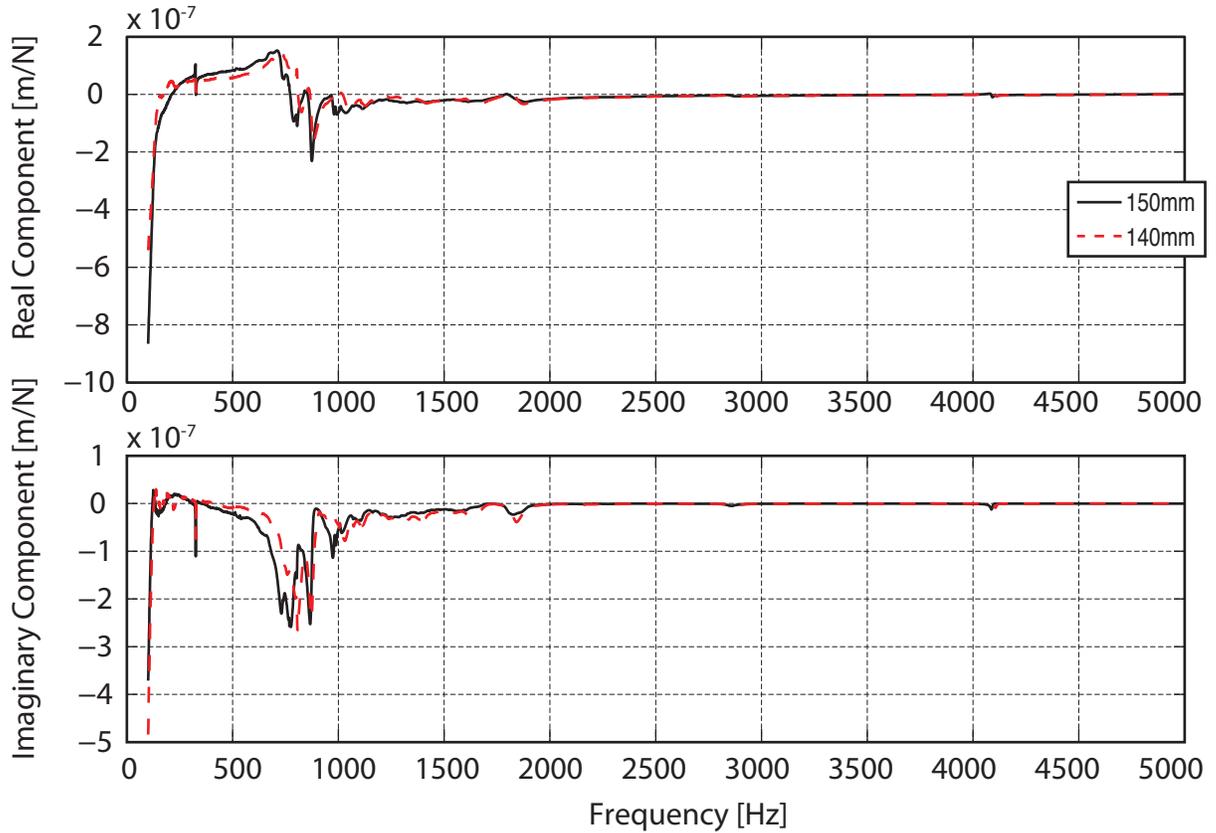
Three facing experiments are conducted with increasing depth of cut values. The cutting parameters for the experiments are given in Table 5.4.

**Table 5.4:** Radial turning chatter experiments

Workpiece: AISI 1045 Steel. See Figure 5.6 for workpiece FRF

Experiment	$n$	$V_c$	$f_c$	$a_c$	Chatter
	[rpm]	[m/min]	[mm/rev]	[mm]	[Hz]
Radial 1	Variable	75	0.1	1	No Chatter
Radial 2	Variable	75	0.1	2	No Chatter
Radial 3	Variable	75	0.1	3	800
Radial 4	Variable	75	0.1	4	800

### 5.3.2 Structural Flexibility



**Figure 5.6:** Direct FRF of the workpiece with different overhang values.

AISI 1045 Steel bar with  $D = 41.275mm$

In this set of experiments the tool holder assembly is rigidly clamped. Since the major cutting forces in facing operations act along the rigid tool holder shaft, this component is assumed to be fully rigid.

The workpiece is clamped with an overhang to increase the flexibility. Workpiece geometry is given as

$$D_{\text{workpiece}} = 41.275mm \quad L_{\text{initial}} = 150mm \quad L_{\text{final}} = 140mm \quad (5.2)$$

The flexibility of the workpiece is measured at the end of the long workpiece, where the cutting takes place. Since the overhang of the workpiece is reduced from 150mm to 140mm during the experiments, its FRF is measured both before and after the cut, as shown in Figure 5.6. The change in the dynamic response of the workpiece is found to be negligible, therefore, the measurements from the workpiece having 140mm overhang is used to simulate the process.

The dynamics of the workpiece are assumed to be identical in X and Y directions. The workpiece is assumed to be rigid along the Z direction (spindle axis). For simplicity, the cross transfer functions were not considered in these experiments.

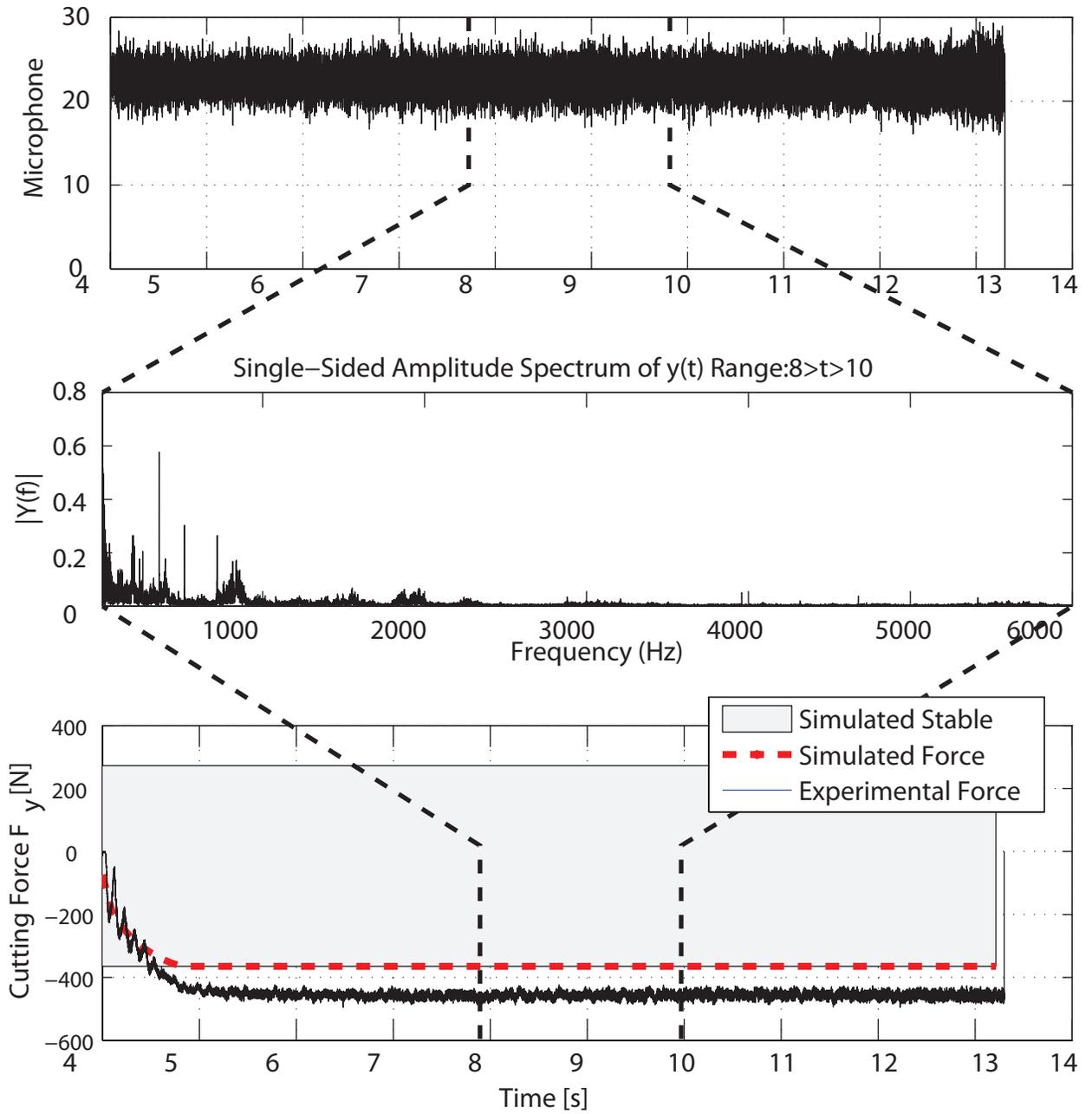
### **5.3.3 Results**

In experiment #1 (Figure 5.7), FFT of the microphone signal does not contain any vibration frequencies, and the cutting forces are stable during the cut. No chatter instability is observed. This result verifies the virtual turning simulation, which does not indicate chatter for this process.

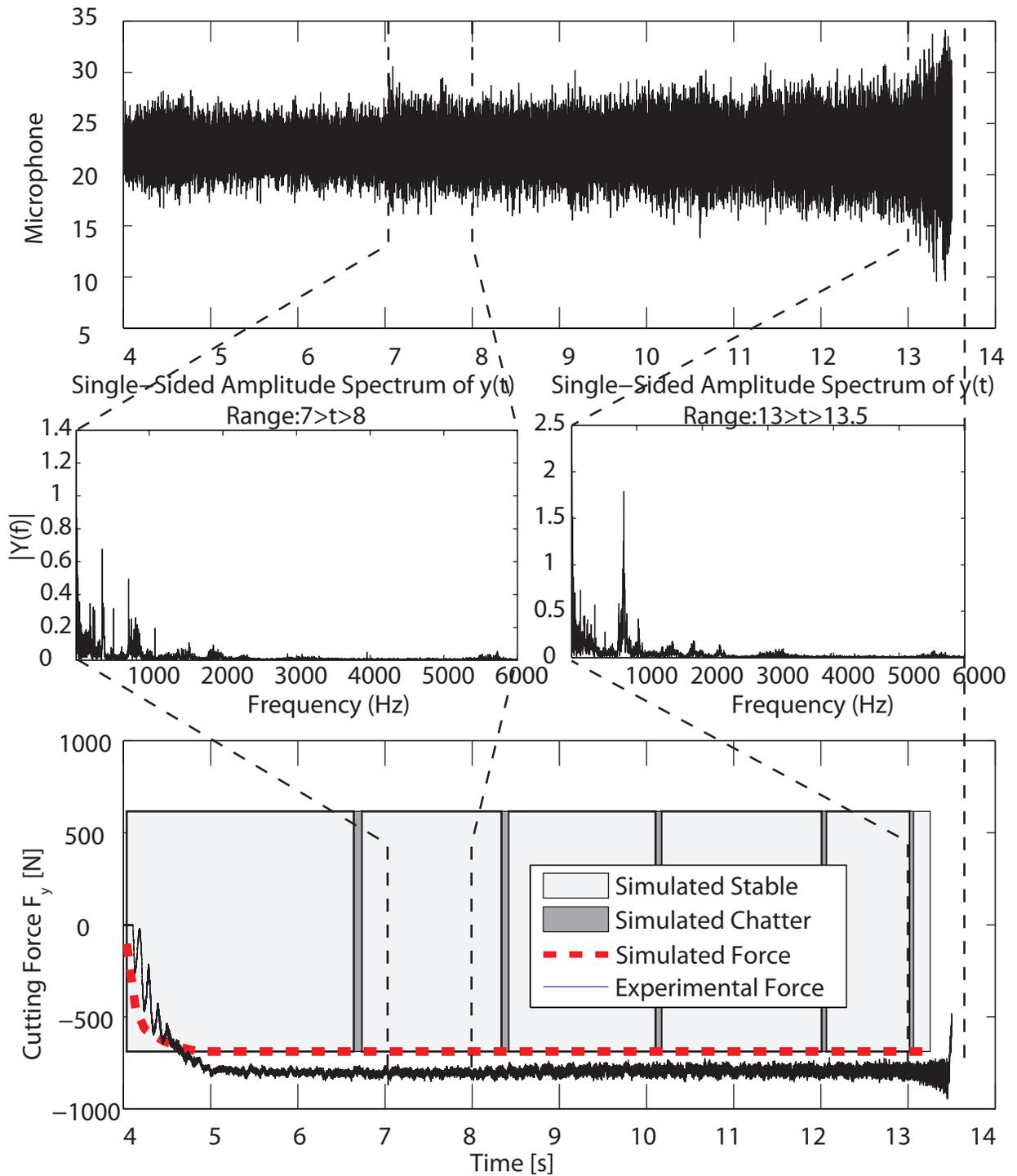
In experiment #2 (Figure 5.8), simulation predicts small regions of instability, however, these regions are too small for chatter to develop. Between the 7-8 seconds, no chatter is predicted by the simulation, as verified by the amplitude spectrum of the microphone signal. Around the regions where chatter is simulated, the FFT of the microphone signal reveals vibrations around 800 Hz, but no growth of the cutting forces is observed. It is concluded that there are no chatter vibrations, in accordance with the simulation results.

In experiment #3 (Figure 5.9), the simulation predicts chatter around 780 Hz, all along the cut. FFT of the microphone signal reveals vibrations around 800Hz, which corresponds to the natural frequency of the first bending mode of the workpiece. The cutting forces slightly grow at several regions of the cut. It is concluded that, there are slight chatter vibrations during this cut.

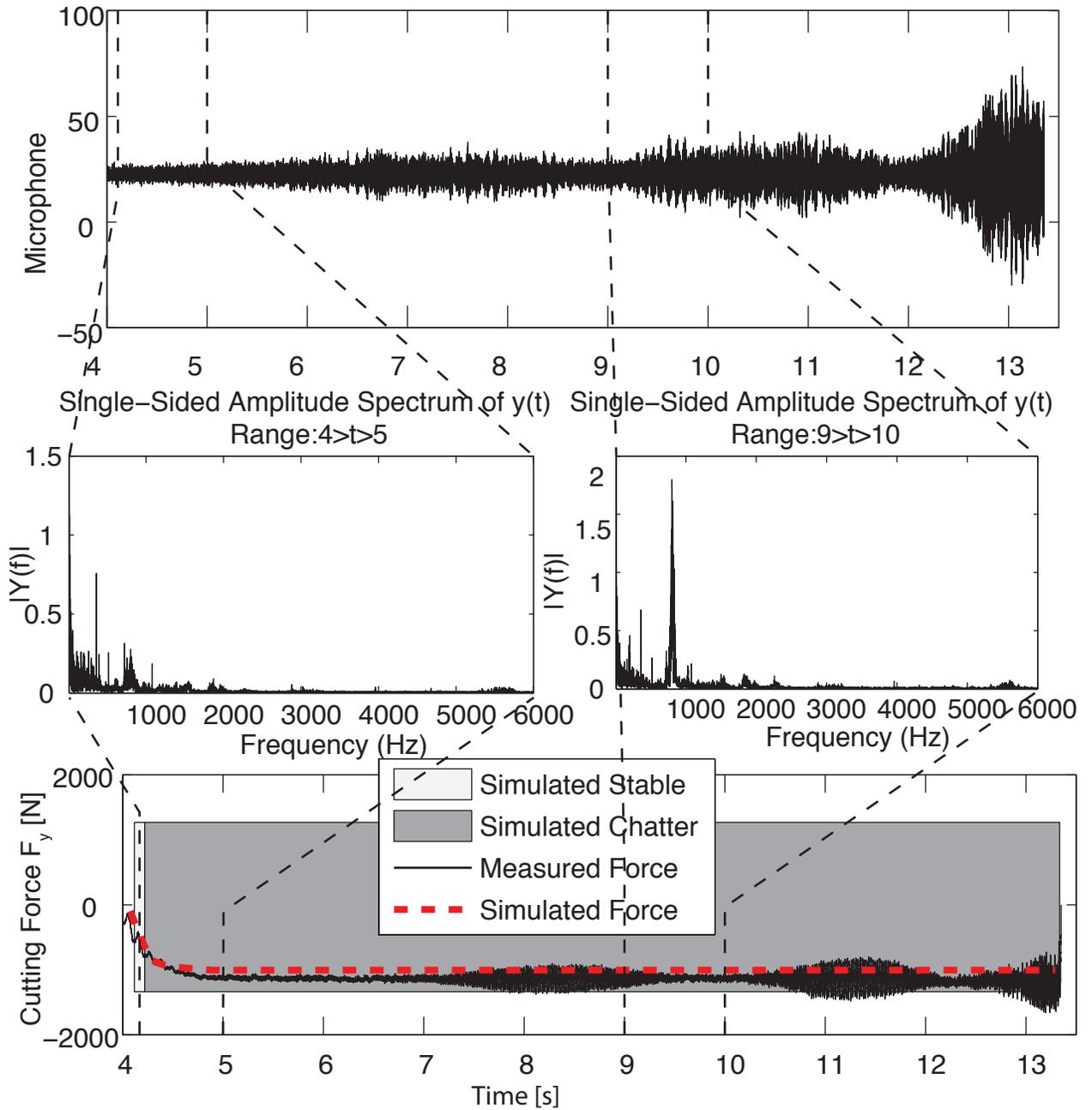
In experiment #4 (Figure 5.10), the microphone signal reveals chatter around 800 Hz, along with growing cutting forces. 800Hz is the first natural frequency of the workpiece as shown in Figure 5.6. Under these cutting conditions the process is unstable, as predicted by the virtual turning simulation.



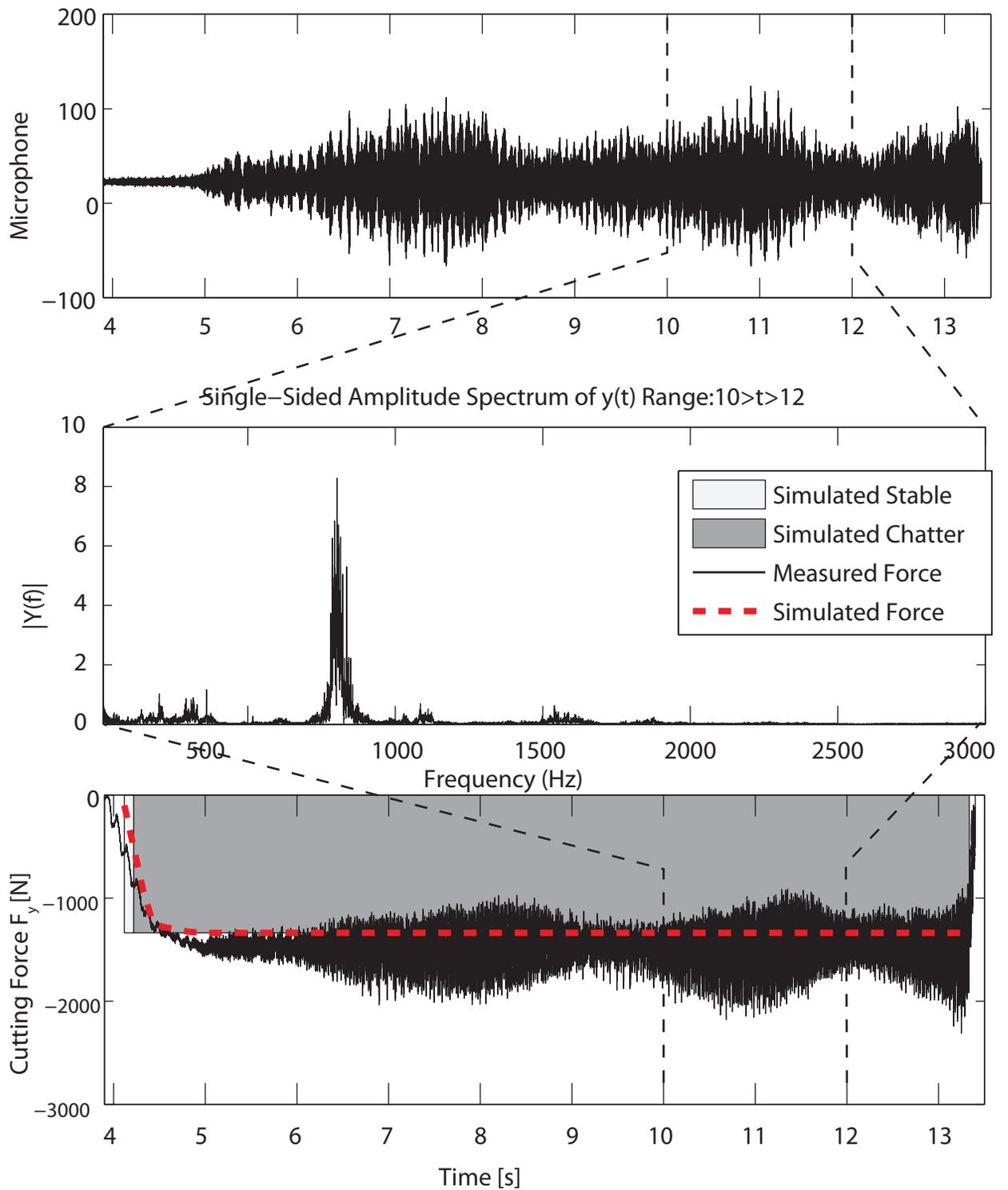
**Figure 5.7:** Radial turning experiment #1  
 Facing process of AISI 1045 steel bar. Workpiece diameter  $D = 41.275$  Cutting conditions:  
 $a_c = 1.0\text{mm}$   $f_c = 0.1\text{mm/rev}$   $V_c = 75\text{m/min}$



**Figure 5.8:** Radial turning experiment #2  
Facing process of AISI 1045 steel bar. Workpiece diameter  $D = 41.275$  Cutting conditions:  
 $a_c = 2.0\text{mm}$   $f_c = 0.1\text{mm/rev}$   $V_c = 75\text{m/min}$



**Figure 5.9:** Radial turning experiment #3  
Facing process of AISI 1045 steel bar. Workpiece diameter  $D = 41.275$  Cutting conditions:  
 $a_c = 3.0\text{mm}$   $f_c = 0.1\text{mm/rev}$   $V_c = 75\text{m/min}$



**Figure 5.10:** Radial turning experiment #4  
Facing process of AISI 1045 steel bar. Workpiece diameter  $D = 41.275$  Cutting conditions:  
 $a_c = 4.0\text{mm}$   $f_c = 0.1\text{mm/rev}$   $V_c = 75\text{m/min}$

## **5.4 Summary**

A virtual turning application is developed in Matlab environment. Machining simulations are constructed using the developed modules and the results are compared against experiments. Contour turning experiments are conducted to verify cutting force simulations. The engagement maps are generated accurately by both the third party API and the developed boolean based model, while the latter proved to be nine fold faster. It was shown that the cutting force simulations are sufficiently accurate. Using the engagement based optimization method the machining process was optimized. The machining performance was almost tripled, while satisfying the machine tool and process limitations. Also the chatter detection capabilities of the virtual turning application is verified by a series of face turning experiments.

## 6 Conclusion

A computationally efficient virtual turning process simulation based on a new tool–workpiece engagement model is introduced in this thesis. The system has tool–workpiece engagement, process mechanics and dynamics, and feedrate scheduling modules which constitute the main contributions of the thesis.

Tool–workpiece engagement boundaries along the segmented tool path needs to be calculated in order to solve the equations which describe the process physics. While solid model based algorithms are most accurate, they take considerable amount of computation time which makes them impractical to use in virtual turning. A typical turning of a gas turbine shaft may require over 100,000 tool-part engagement calculations, which can take hours unless computationally efficient models are introduced. It has been more common to use z–buffer models which are computationally faster but the engagement conditions are not identified accurately because the solid models are digitized at discrete mesh points. Since the turning parts are modeled by the revolution of their cross–sections, a simple but efficient polycurve method is introduced in the thesis. Workpiece and tool edge intersection boundaries are analytically calculated by Boolean operations. Using two dimensional analytical geometry allows nine fold increase in computational efficiency while improving the geometric accuracy in comparison to popular z-buffer methods. With the best of author’s knowledge, this thesis introduced the application of polycurve method first time in the virtual machining literature.

The process forces must also be rapidly calculated in virtual machining. Past research identified the cutting force distribution along the cutting edge, which required computationally costly digital integration of differential force distribution. Without sacrificing from the accuracy of force

prediction, this thesis applied Colwell's chord angle and line approximation in solving the mechanics of cutting. The method has proven to predict the varying forces along a part, while avoiding costly and slow iterative methods.

Feeds and speeds of the cutting process are improved while respecting torque, power, force and depth of cut limits. A direct search method borrowed from the literature is adopted here.

The virtual turning system is already functional, but further research is needed before its use in industry. Boolean operations may sometimes crash due to their numerical instabilities. The optimization model is needed to be further enhanced.

# Bibliography

- [1] M. Kaymakci, “Generalized Modeling Of Metal Cutting Mechanics,” Master’s thesis, Vancouver, Canada, Aug. 2009.
- [2] M. Eynian and Y. Altintas, “Chatter Stability of General Turning Operations With Process Damping,” *Journal of Manufacturing Science and Engineering*, vol. 131, no. 4, p. 041005, 2009.
- [3] S. Centmayer, “General Modeling of Mechanics and Kinematics of Metal Cutting,” Master’s thesis, Universität Stuttgart, 2007.
- [4] C. A. van Luttervelt, T. H. C. Childs, I. S. Jawahir, F. Klocke, P. K. Venunod, Y. Altintas, E. J. A. Armarego, D. Dornfeld, I. Grabec, J. Leopold, B. Lindstrom, D. Lucca, T. Obikawa, Shirakashi, and H. Sato, “Present Situation and Future Trends in Modelling of Machining Operations Progress Report of the CIRP Working Group ‘Modelling of Machining Operations’,” *CIRP Annals - Manufacturing Technology*, vol. 47, no. 2, pp. 587–626, 1998.
- [5] J. Zhou, “Virtual Turning System,” Master’s thesis, Vancouver, Canada, Jun. 2005.
- [6] Z. Shengfang, Z. Na, S. Zhihua, and J. Xiaoli, “Study on turning simulation technology based on parametric feature modeling,” in *Mechanic Automation and Control Engineering (MACE), 2010 International Conference on*, 2010, pp. 49–51.
- [7] J. C. J. Chiou, “Virtual Clay: An Enhanced Marching Cubes Algorithm for In-Process Geometry Modeling,” *Journal of Manufacturing Science and Engineering*, vol. 129, no. 3, pp. 566–574, 2007.
- [8] J. Zhou, D. Yip-Hoi, and X. Huang, “A Hybrid Analytical, Solid Modeler and Feature-Based Methodology for Extracting Tool-Workpiece Engagements in Turning,” *Journal of Computing and Information Science in Engineering*, vol. 7, no. 3, pp. 192–202, 2007.
- [9] J. Li, W. B. Lee, Y. Yao, C. F. Cheung, and S. To, “Workpiece representation for virtual turning,” *International Journal Of Advanced Manufacturing Technology*, vol. 25, no. 9-10, pp. 857–866, 2005.
- [10] L. Zhang, J. Feng, Y. Wang, and M. Chen, “Feedrate scheduling strategy for free-form surface machining through an integrated geometric and mechanistic model,” *The International Journal of Advanced Manufacturing Technology*, vol. 40, pp. 1191–1201, 2009.

- [11] N. Z. Yussefian, B. Moetakef-Imani, and H. El-Mounayri, "The prediction of cutting force for boring process," *International Journal Of Machine Tools & Manufacture*, vol. 48, no. 12-13, pp. 1387–1394, 2008.
- [12] F. Atabey, I. Lazoglu, and Y. Altintas, "Mechanics of boring processes—Part I," *International Journal Of Machine Tools & Manufacture*, vol. 43, no. 5, pp. 463–476, 2003.
- [13] R. G. Reddy, S. G. Kapoor, and R. E. DeVor, "A Mechanistic Force Model for Contour Turning," *Journal of Manufacturing Science and Engineering*, vol. 122, no. 3, pp. 398–405, 2000.
- [14] Y. Altintas, *Manufacturing Automation*, 2000.
- [15] L. V. Colwell, "Predicting the Angle of Chip Flow for Single-Point Cutting Tools," *Transactions of the ASME*, Feb. 1954.
- [16] M. Eynian, "Chatter Stability of Turning and Milling with Process Damping," Ph.D. dissertation, The University of British Columbia, Vancouver, Canada.
- [17] E. Ozlu and E. Budak, "Analytical Modeling of Chatter Stability in Turning and Boring Operations—Part I: Model Development," *Journal of Manufacturing Science and Engineering*, vol. 129, no. 4, pp. 726–732, 2007.
- [18] S. S. Rao, *Engineering Optimization - Theory and Practice (4th Edition)*, 2009.
- [19] L. Brits, "Eulerangles," Wikipedia, the free encyclopedia.
- [20] Y. Yao, H. Zhao, J. Li, and Z. Yuan, "Modeling of virtual workpiece with machining errors representation in turning," *Journal Of Materials Processing Technology*, vol. 172, no. 3, pp. 437–444, 2006.
- [21] H. Narita, K. Shirase, H. Wakamatsu, A. Tsumaya, and E. Arai, "Real-time cutting simulation system of a milling operation for autonomous and intelligent machine tools." *International Journal of Production Research*, vol. 40, no. 15, pp. 3791–3805, 2002.
- [22] O. B. Ozdoganlar and W. J. Endres, "An Analytical Representation of Chip Area for Corner-Radiused Tools Under Both Depth-of-Cut and Feed Variations," *Journal of Manufacturing Science and Engineering*, vol. 122, no. 4, pp. 660–665, 2000.
- [23] M. I. Shamos and D. Hoey, "Geometric intersection problems," in *Proceedings*, 1976, pp. 208–215.
- [24] J. Bentley and T. Ottmann, "Algorithms for Reporting and Counting Geometric Intersections," *Computers, IEEE Transactions on*, vol. C-28, no. 9, pp. 643–647, 1979.
- [25] M. Rivero and F. R. Feito, "Boolean operations on general planar polygons," *Computers & Graphics*, vol. 24, no. 6, pp. 881–896, 2000.
- [26] F. Martínez, A. J. Rueda, and F. R. Feito, "A new algorithm for computing Boolean operations on polygons," *Computers & Geosciences*, vol. 35, no. 6, pp. 1177–1185, 2009.

- [27] L. J. Simonson, "Industrial strength polygon clipping: A novel algorithm with applications in VLSI CAD," *Computer-Aided Design*, vol. 42, no. 12, pp. 1189–1196, 2010.
- [28] S. A. Tobias and W. Fishwick, "The Chatter of Lathe Tools Under Orthogonal Cutting Conditions," *Transactions of the ASME*, Jul. 1958.
- [29] J. Tlustý and M. Poláček, *The stability of machine tools against self excited vibrations in machining*. Proceedings of the ASME International Research in Production Engineering, 1963.
- [30] S. D. Merdol, "Virtual Three-Axis Milling Process Simulation and Optimization," Ph.D. dissertation, Vancouver, Canada, Apr. 2008.
- [31] E. Budak, Y. Altintas, and E. J. A. Armarego, "Prediction of Milling Force Coefficients From Orthogonal Cutting Data," *Journal of Manufacturing Science and Engineering*, vol. 118, no. 2, pp. 216–224, 1996.
- [32] E. J. A. Armarego and R. H. Brown, *The Machining of Metals*, 1969.
- [33] G. V. Stabler, "The Chip Flow Law and Its Consequences," *Advances in Machine Tool Design and Research*, pp. 243–251, 1964.
- [34] E. Oberg, F. D. Jones, H. L. Horton, and H. H. Ryffel, *Machinery's Handbook*, 2008.
- [35] R. Y. Chiou and S. Y. Liang, "Chatter stability of a slender cutting tool in turning with tool wear effect," *International Journal Of Machine Tools & Manufacture*, vol. 38, no. 4, pp. 315–327, Mar. 1998.
- [36] D. WU, "A New Approach of Formulating the Transfer-Function for Dynamic Cutting Processes," *Journal of Engineering for Industry-Transactions of the Asme*, vol. 111, no. 1, pp. 37–47, 1989.
- [37] D. Goldberg, "What every computer scientist should know about floating-point arithmetic," *ACM Computing Surveys*, 1991.
- [38] E. W. Weisstein. Line-Line Intersection. [Online]. Available: <http://mathworld.wolfram.com/Line-LineIntersection.html>
- [39] ——. Circle-Circle Intersection. [Online]. Available: <http://mathworld.wolfram.com/Circle-CircleIntersection.html>
- [40] ——. Circle-Line Intersection. [Online]. Available: <http://mathworld.wolfram.com/Circle-LineIntersection.html>

# Appendices

## A.1 Intersection Algorithms

Using two entity types there are three possible intersection types that may occur in the geometry.

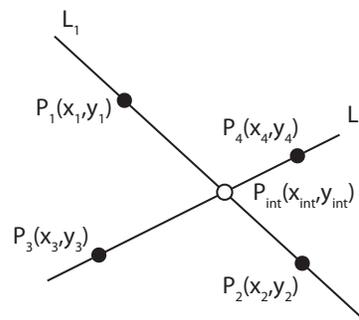
- Line – Line Intersection
- Line – Arc Intersection
- Arc – Arc Intersection

Simple geometric algorithms for these intersections are widely available and relatively simple.

### A.1.1 Line - Line Intersection

The intersection of the two lines are calculated [38].

Let two lines be defined by two points.



**Figure A.1:** Line-Line intersection

$$\begin{aligned}
P_1 &= (x_1, y_1) & P_2 &= (x_2, y_2) \\
P_3 &= (x_3, y_3) & P_4 &= (x_4, y_4)
\end{aligned} \tag{A.1}$$

Writing the linear equations in the general form the lines are represented by

$$\begin{aligned}
L_1 &\rightarrow A_1x + B_1y = C_1 \\
L_2 &\rightarrow A_2x + B_2y = C_2
\end{aligned} \tag{A.2}$$

where

$$\begin{aligned}
A_1 &= y_2 - y_1 & B_1 &= x_1 - x_2 & C_1 &= A_1x_1 + B_1y_1 \\
A_2 &= y_3 - y_3 & B_2 &= x_3 - x_4 & C_2 &= A_2x_3 + B_2y_3
\end{aligned} \tag{A.3}$$

Two line equation can now be written in the matrix form as

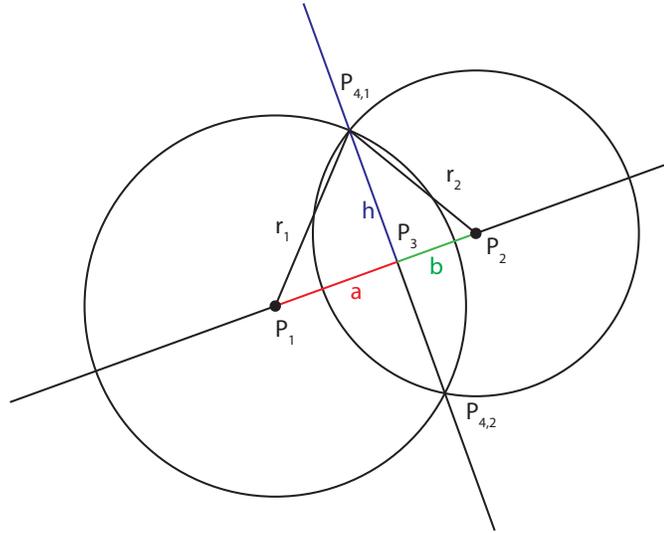
$$\begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{A.4}$$

Solving for x and y

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} A_1 & B_1 \\ A_2 & B_2 \end{bmatrix}^{-1} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{A.5}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{A_1B_2 - A_2B_1} \begin{bmatrix} B_2 & -B_1 \\ -A_2 & A_1 \end{bmatrix} \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \tag{A.6}$$

Therefore the point of intersection is



**Figure A.2:** Circle-Circle intersection

$$P_{int} = \begin{bmatrix} \frac{B_2C_1 - B_1C_2}{\Delta} \\ \frac{A_1C_2 - A_2C_1}{\Delta} \end{bmatrix} \quad (\text{A.7})$$

with

$$\Delta = A_1B_2 - A_2B_1 \quad (\text{A.8})$$

### A.1.2 Circle–Circle Intersection

The intersection points of two circles is calculated for two circles with centers  $P_1(x_1, y_1)$  and  $P_2(x_2, y_2)$  and radii  $r_1$  and  $r_2$  respectively [39]. The distance between two circles

$$\begin{aligned} d &= \|P_2 - P_1\| \\ &= \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \end{aligned} \quad (\text{A.9})$$

Depending on this value the intersection conditions are checked

$$d \geq r_1 + r_2 \rightarrow \text{No Intersection}$$

$$d < |r_1 - r_2| \rightarrow \text{No Intersection}$$

$$d = 0 \wedge r_1 = r_2 \rightarrow \text{Circles Coincident}$$

Using the triangle

$$r_1^2 = h^2 + a^2 \tag{A.10}$$

$$r_2^2 = h^2 + b^2 \tag{A.11}$$

Since

$$a + b = d \tag{A.12}$$

$$b = d - a \tag{A.13}$$

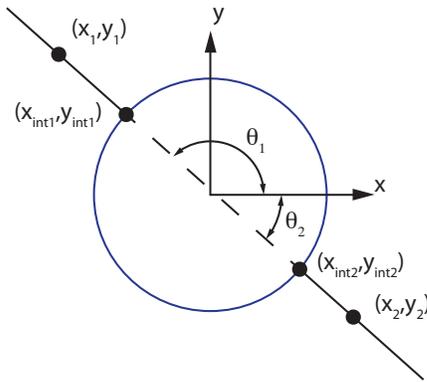
it follows that

$$a = \frac{(r_2^2 - r_1^2 - d^2)}{-2d} \tag{A.14}$$

$$h = \sqrt{r_1^2 - a^2} \tag{A.15}$$

$$\Rightarrow P_3 = (P_2 - P_1) \frac{a}{d} + P_1 \tag{A.16}$$

$$\Rightarrow P_4 = \begin{bmatrix} x_3 \pm \frac{h}{d}(y_2 - y_1) \\ y_3 \mp \frac{h}{d}(x_2 - x_1) \end{bmatrix} \tag{A.17}$$



**Figure A.3:** Line-Circle intersection

### A.1.3 Line - Arc Intersection

Intersection of a line and circle is calculated [40]. Let  $L_1$  be the line defined by its two points.

$$P_1(x_1, y_1)$$

$$P_2(x_2, y_2)$$

and the circle centered at the origin with a radius of  $r$ .

$$d_x = x_2 - x_1 \quad (\text{A.18})$$

$$d_y = y_2 - y_1 \quad (\text{A.19})$$

$$d_r = \sqrt{d_x^2 + d_y^2} \quad (\text{A.20})$$

$$D = \begin{vmatrix} x_1 & x_2 \\ y_1 & y_2 \end{vmatrix} = x_1 y_2 - x_2 y_1 \quad (\text{A.21})$$

The discriminant

$$\Delta \equiv r^2 d_r^2 - D^2 \quad (\text{A.22})$$

defines the incidence such as

$$\Delta < 0 \rightarrow \text{No Intersection}$$

$$\Delta = 0 \rightarrow \text{Tangent}$$

$$\Delta > 0 \rightarrow \text{Intersection at two discrete points}$$

and the coordinates of the intersection point is given by

$$x = \frac{D d_y \pm \text{sgn}^*(d_y) d_x \sqrt{r^2 d_r^2 - D^2}}{d_r^2} \quad (\text{A.23})$$

$$y = \frac{-D d_x \pm |d_y| \sqrt{r^2 d_r^2 - D^2}}{d_r^2} \quad (\text{A.24})$$

where the signum function is described as

$$\text{sgn}^*(x) \begin{cases} -1 & \text{for } x < 0 \\ 1 & \text{otherwise} \end{cases}$$

## A.1.4 Extension of Intersection Algorithms for Line Segments and Arcs

Since the intersection algorithms are defined only for complete circles and infinite lines, the calculated intersection points must be checked to make sure they lie on the line segments or arcs represented by the entities. This can be accomplished by verifying the intersection point  $P_{int}$  belongs to both entities.

$$P_{int} \subset \text{Entity 1} \wedge P_{int} \subset \text{Entity 2} \quad (\text{A.25})$$

### A.1.4.1 Line segments

The calculated intersection point  $P_{int}(x, y)$  lies on the line segment if and only if

$$\begin{aligned} \min(x_1, x_2) \leq x \leq \max(x_1, x_2) \\ \min(y_1, y_2) \leq y \leq \max(y_1, y_2) \end{aligned} \quad (\text{A.26})$$

For the intersection of two line segments, the intersection point must lie on both line segments to be a valid intersection point. Therefore using the equation above,

$$\begin{aligned} \min(x_1, x_2) \leq x \leq \max(x_1, x_2) \wedge \min(y_1, y_2) \leq y \leq \max(y_1, y_2) \wedge \dots \\ \min(x_3, x_4) \leq x \leq \max(x_3, x_4) \wedge \min(y_3, y_4) \leq y \leq \max(y_3, y_4) \end{aligned} \quad (\text{A.27})$$

which can be simplified into

$$\begin{aligned} \max[\min(x_1, x_2), \min(x_3, x_4)] \leq x \leq \min[\max(x_1, x_2), \max(x_3, x_4)] \\ \wedge \\ \max[\min(y_1, y_2), \min(y_3, y_4)] \leq y \leq \min[\max(y_1, y_2), \max(y_3, y_4)] \end{aligned} \quad (\text{A.28})$$

#### A.1.4.2 Arc

The arc entities are defined by a simple circle entity, and the corresponding start and end angles ( $\theta_i$  and  $\theta_f$ ) respectively. Once the intersection point is located, the corresponding angular position of the intersection point can be calculated by atan2 function.

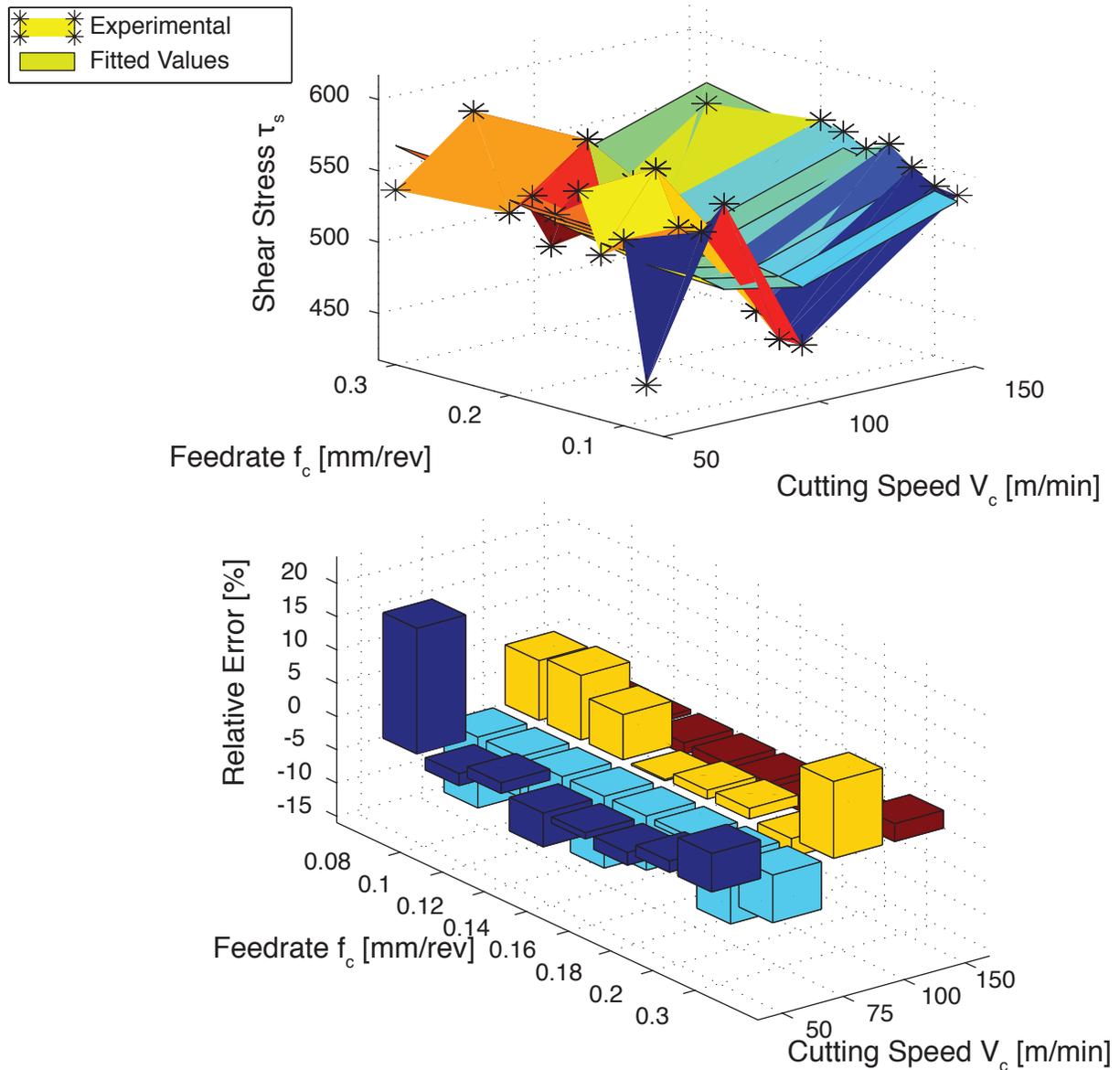
$$\begin{aligned}\Delta x &= (x - x_o) \\ \Delta y &= (y - y_o) \\ \theta &= \tan^{-1} \left( \frac{\Delta y}{\Delta x} \right)\end{aligned}\tag{A.29}$$

The point  $P_i$  lies on the arc if and only if

$$\theta_i \leq \theta \leq \theta_f$$

For Line-Arc intersection, this condition must be checked along with the check for the line segment. For Arc-Arc intersection the same condition must be satisfied by both arcs for the intersection point to be valid.

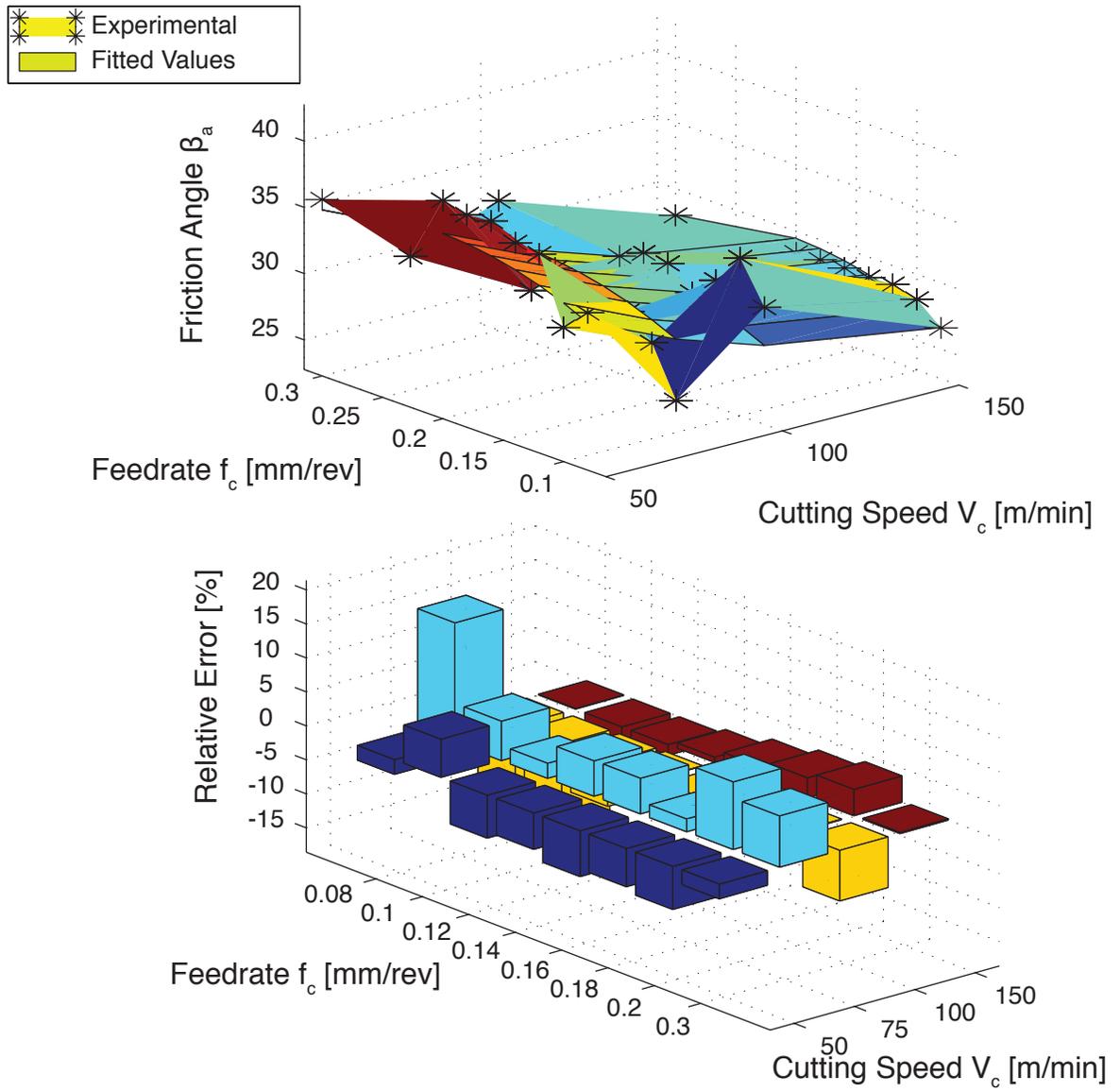
## A.2 AISI 1045 Cutting Coefficients



**Figure A.4:** Quadratic surface fit for shear stress (AISI 1045 Steel)

Physics of an oblique turning process can be defined by using parameters such as shear stress ( $\tau_s$ ), friction angle ( $\beta_a$ ) and shear angle ( $\phi_c$ ). These values can be estimated by using data stored in orthogonal cutting databases. For this purpose, the orthogonal cutting data are represented commonly as a function of cutting parameters.

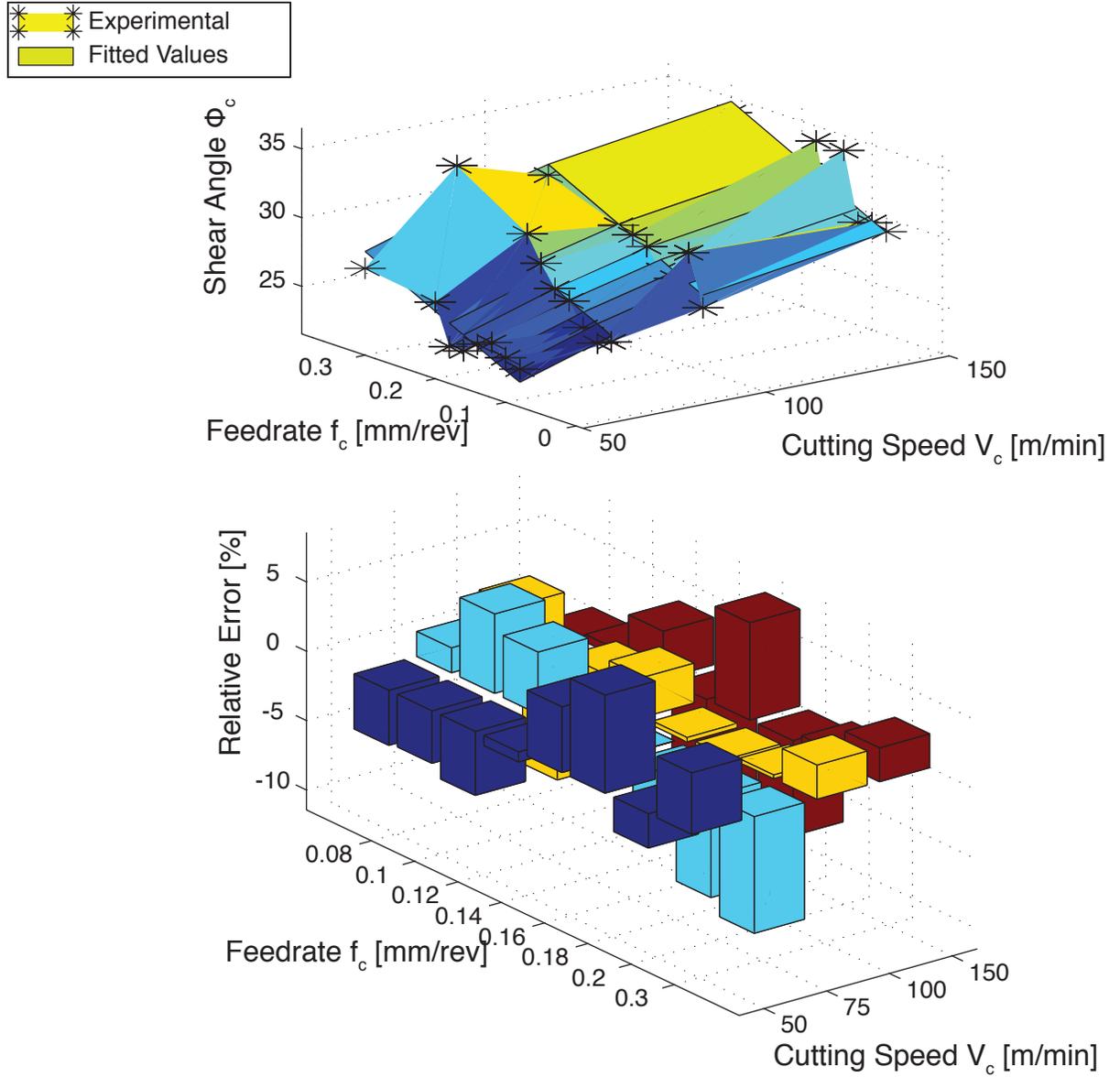
For AISI 1045 steel, these values are obtained by varying the chip thickness ( $h_c$ ) and cutting



**Figure A.5:** Quadratic surface fit for friction angle (AISI 1045 Steel)

speed ( $V_c$ ).

$$\begin{aligned}
 \tau_s &= f(h_c, V_c) \\
 \beta_a &= f(h_c, V_c) \\
 \phi_c &= f(h_c, V_c)
 \end{aligned}
 \tag{A.30}$$



**Figure A.6:** Quadratic surface fit for shear angle (AISI 1045 Steel)

These parameters can be modeled as a quadratic surface as given in Equation A.31 by surface fitting over the experimental data. The coefficients in these equations are calibrated directly from orthogonal cutting experiments [3].

$$[\tau_s, \beta_a, \phi_c] = C_1 + C_2 V_c^2 + C_3 V_c + C_4 h_c^2 + C_5 h_c \quad (\text{A.31})$$

These coefficients are obtained using a least squares method. The obtained results are shown in

Figure A.4, Figure A.5, Figure A.6.

The coefficients for the fitted surfaces are given in Table A.1.

**Table A.1:** Coefficients for fitted surfaces

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	Error
$\tau_s$	639.03	0.0152	-3.094	30.34	153.04	4.81%
$\phi_c$	15.82	-0.0004	0.13	1.67	25.47	3.53%
$\beta_a$	37.49	0.0006	-0.20	-173.98	72.43	4.82%

## A.2.1 Cutting Edge Coefficients

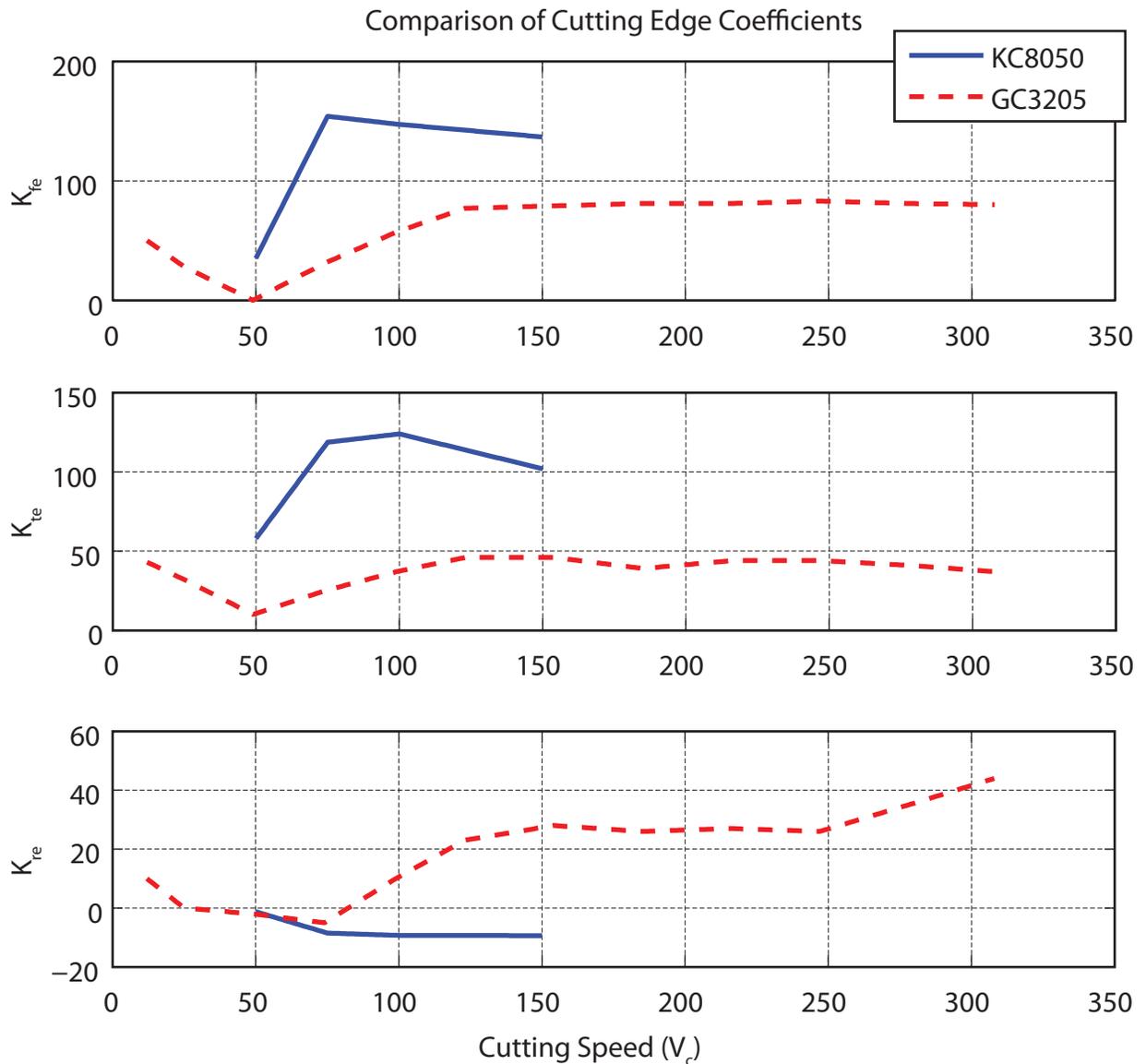
Cutting edge coefficients, which account for the portion of cutting forces which are not related to the shearing process, are also calibrated by cutting experiments. However, unlike cutting coefficients which are modeled to be only workpiece material dependent, edge coefficients are specific to the workpiece–tool material pair [14].

For AISI 1045 steel two sets of experiments were performed in previous studies. The first set of experiments, have been conducted with *Sandvik GC3205* grade inserts. The results are given in Table A.2 [2].

**Table A.2:** Cutting coefficients for AISI 1045 steel with GC3205 grade tool [2]

n	$V_c$	$K_{fe}$	$K_{fa}$	$K_{re}$	$K_{ra}$	$K_{te}$	$K_{ta}$
[rpm]	[m/min]	[N/mm]	[MPa]	[N/mm]	[MPa]	[N/mm]	[MPa]
100	12	50	1816	10	-25	43	3512
200	25	28	2458	0	-5	32	3776
400	49	0	4359	-2	188	10	4811
600	74	31	3364	-5	172	25	4152
800	99	57	2509	10	53	37	3623
1000	123	77	1937	23	-52	46	3235
1250	154	79	1773	28	-81	46	3102
1500	185	81	1639	26	-85	39	3294
1750	216	81	1580	27	-115	44	2937
2000	247	83	1544	26	-124	44	2881
2250	278	81	1619	35	-219	41	2929
2500	308	80	1698	44	-293	37	3042

The second set of experiments were conducted by Centmayer who used AISI 1045 Steel work-



2

**Figure A.7:** Comparison of cutting edge coefficients for AISI 1045 steel

piece with Kennametal KC8050 grade cutting tools. Experimental results are presented in Table A.3 [3].

Since this grade of tool is very similar to the Sandvik GC6050, these coefficients are assumed to be applicable to the GC6040 grade as well.

It can be seen that the cutting edge coefficients obtained from each study is very different as shown in Figure A.7. Therefore it is necessary to run cutting tests for each workpiece and tool

**Table A.3:** Cutting coefficients for AISI 1045 steel with KC8050 grade tool [3]

$V_c$	$K_{fe}$	$K_{te}$	$K_{re}$
[m/min]	[N/mm]	[N/mm]	[N/mm]
50	35.12	58.033	-1.2361
75	154	118.69	-8.4721
100	147.21	123.93	-9.2675
150	136.72	101.97	-9.3731

material combination. The edge coefficients are also sensitive to the tool coating, tool wear and chipping.

In the orthogonal databases, data for commonly used workpiece - tool material pairs can be stored. Such an process will unavoidably, increase the number of orthogonal cutting tests required to construct an orthogonal database.

### A.2.2 Curve Fitting

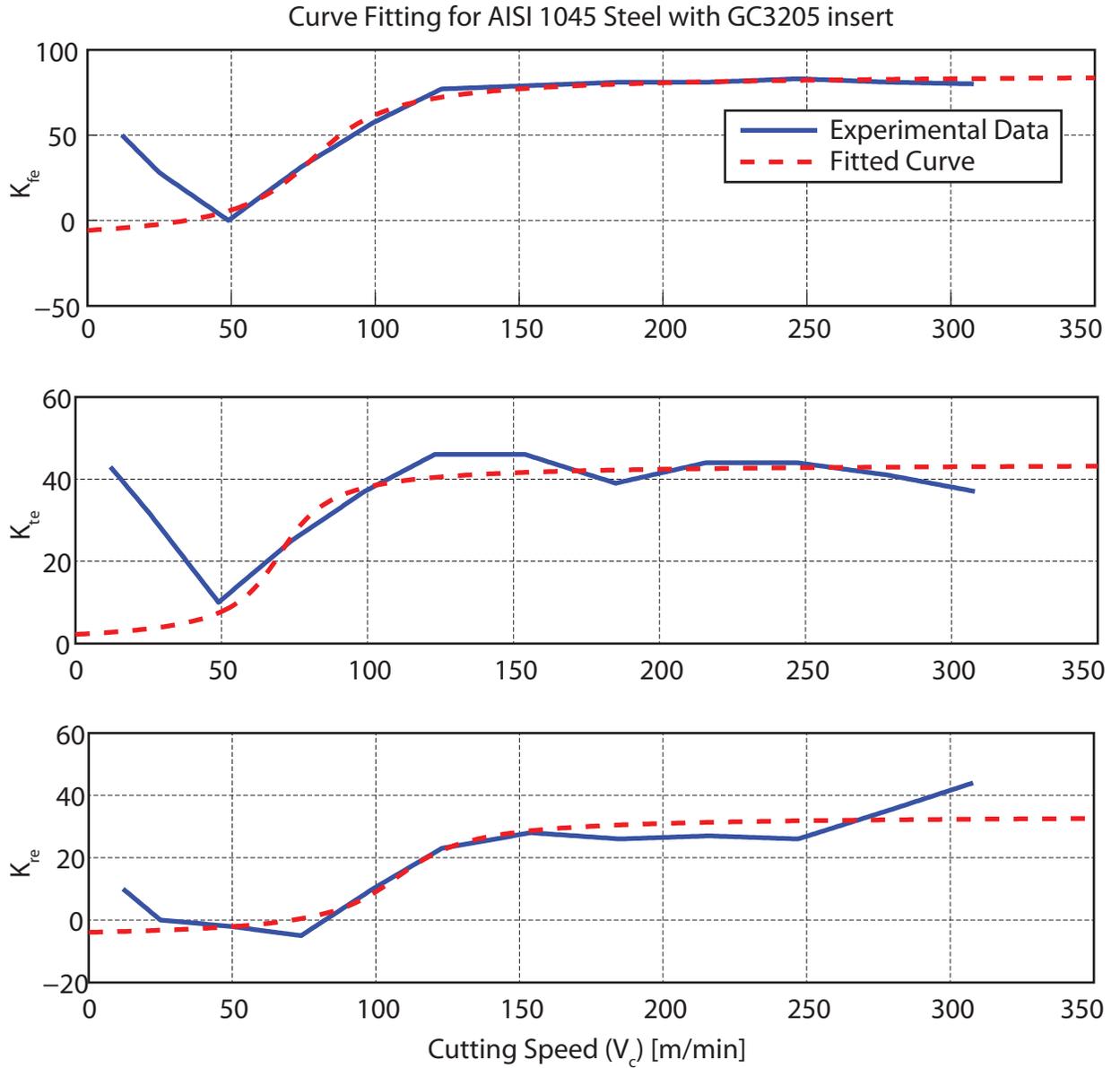
In order to store these experimental data in a format which can be used in a simulation system, the acquired data is fitted with a curve. However a polynomial curve fitting does not yield satisfactory representation of the cutting coefficients.

Therefore a function which can represent the overall shape of the curve was chosen. The fitted curve is in the form of an arctangent function as shown in Equation A.32

$$K_e = C_1 \left[ \text{atan} \left( \frac{V_c - C_2}{C_3} \right) \right] + C_4 \quad (\text{A.32})$$

The experimental data (Table A.2 and Table A.3) are fitted by using a least squares method. In order for the fitted curve to preserve the overall shape of the experimental values, intermediate data points are created by interpolation between the given data points.

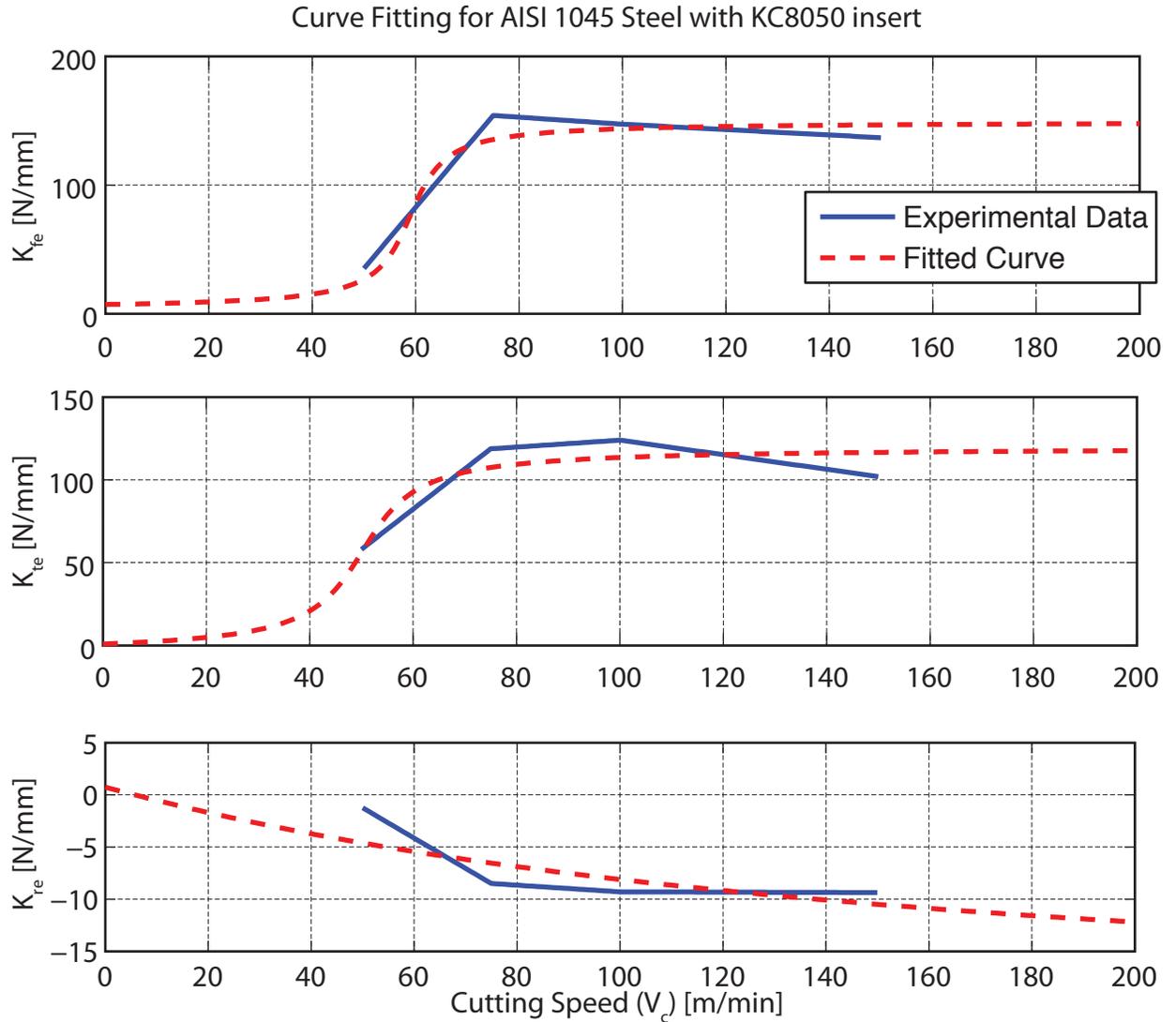
Also, the cutting edge coefficients were assumed to reach a value of zero when the cutting speed goes to zero.



**Figure A.8:** Curve fitting for cutting edge coefficients (GC3205 grade tool and AISI 1045 Steel)

$$(K_e)_{V_c=0} = 0 \quad (\text{A.33})$$

The resulting fitted curves are compared to the experimental values in Figure A.8 and Figure A.9 for GC3205 and KC8050 grades respectively. The coefficients for each tool grade are found as



**Figure A.9:** Curve fitting for cutting edge coefficients (KC8050 grade tool and AISI 1045 Steel)

shown in Table A.4 and Table A.5 respectively.

**Table A.4:** Coefficients for edge coefficient of AISI 1045 steel with GC3205 grade tool

	$C_1$	$C_2$	$C_3$	$C_4$
$K_{fe}$	14.060	69.967	12.979	21.711
$K_{te}$	31.833	78.128	20.653	35.946
$K_{re}$	12.642	107.350	19.143	13.690

**Table A.5:** Coefficients for edge coefficient of AISI 1045 steel with KC8050 grade tool

	$C_1$	$C_2$	$C_3$	$C_4$
$K_{fe}$	46.486	58.968	4.967	76.219
$K_{te}$	39.804	50.355	7.883	57.179
$K_{re}$	-27.267	-102.09	110.65	21.059

### A.3 NC-Commands for Hardinge Superslant Lathe

The NC-commands and the required data formats for the Hardinge Superslant lathe are obtained from the user manual. The commands listed in Table A.6 are implemented within the NC-parser algorithm.

**Table A.6:** NC-commands implemented in the G-code parser

Command	Value	Type	Definition
G	00	Modal	Rapid Motion
G	01	Modal	Linear Motion
G	02	Modal	CW Circular Interpolation
G	03	Modal	CCW Circular Interpolation
G	96	Modal	Constant Surface Speed Mode
X	XXX.xxx	Position	X-coordinate in diameter mode
Z	XXX.xxx	Position	Z-coordinate
I	XXXX.xxxx	Position	Circular interpolation arc center (X incremental)
K	XXXX.xxxx	Position	Circular interpolation arc center (Z incremental)
F	XXX.xxx	Modal	Feedrate
S	XXXX	Modal	Spindle speed (or Cutting speed if CSS mode is active)
T	XX	Modal	Tool selection