

DEVELOPMENT OF POSTPROCESSOR, SIMULATION AND
VERIFICATION SOFTWARE FOR A FIVE-AXIS CNC MILLING MACHINE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ENDER CENGİZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
MECHANICAL ENGINEERING

SEPTEMBER 2005

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science

Prof. Dr. Kemal İDER
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science

Prof. Dr. Mustafa İlhan GÖKLER
Supervisor

Examining Committee Members:

Prof. Dr. R.Orhan YILDIRIM (METU, ME)

Prof. Dr. Mustafa İlhan GÖKLER (METU, ME)

Prof. Dr. S.Engin KILIÇ (METU, ME)

Asst. Prof. Dr. Melik DÖLEN (METU, ME)

Prof. Dr. Can Çoğun (GAZI UNIV,ME)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ender Cengiz

ABSTRACT

DEVELOPMENT OF POSTPROCESSOR, SIMULATION AND VERIFICATION SOFTWARE FOR A FIVE-AXIS CNC MILLING MACHINE

CENGİZ, Ender

M.Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. Mustafa İlhan GÖKLER

September 2005, 145 pages

Five-axis CNC milling machine tools bring great facility to produce complex workpieces with increased dimensional accuracy and better surface quality in shorter machining times. However, kinematics of five-axis machine tools has a complex form which makes it difficult to operate these machine tools properly. The difficulty arises from the complexity of NC-Code generation and tool path verification. Collision of machine tool or setup components with each other is a severe problem in five-axis machining operations and usually results from inadequate postprocessors or insufficient collision checking due to absence of well-prepared simulation and verification programs. Five-axis CNC machine tool owners may get rid of this problem by purchasing commercial postprocessors, simulation and verification programs. However, these programs are expensive and small and medium enterprises (SME's) usually cannot afford the costs of these programs. In the related libraries of commercial programs, there is great number of CNC machine tools, which is generally unnecessary for SME's. An alternative to overcome this problem is to develop particular program, which is capable of postprocessing, simulating and verifying milling operations, for each certain five-axis CNC machine tool.

In this study, a software named “*Manus 1.0*”, which performs postprocessing and simulation processes, has been developed for the high speed “Mazak Variaxis 630-5X” CNC five-axis machine tool, located in METU-BILTIR Center. Moreover, tool path verification algorithms have been developed to detect collisions. The software has been written in Borland C++ Builder5.0. The developed program has been tested in sample milling operations and satisfactory results have been achieved.

Keywords: Five-Axis Postprocessor, Milling Simulation, Tool Path Verification, Collision Detection, Collision Warning

ÖZ

BEŞ-EKSEN CNC FREZE TEZGAHI İÇİN SON-İŞLEMCİ, SİMÜLASYON VE DOĞRULAMA YAZILIMI GELİŞTİRİLMESİ

CENGİZ, Ender

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Mustafa İlhan GÖKLER

Eylül 2005, 145 sayfa

Beş-eksen CNC freze tezgahları, kompleks parçaların artan boyutsal doğruluklarla ve daha iyi yüzey kaliteleriyle, daha kısa kesme sürelerinde üretilmesinde büyük kolaylık getirmiştir. Fakat, beş-eksen tezgahların kinematiği, bu makinelerle uygun şekilde çalışmayı zorlaştıran, karmaşık bir yapıya sahiptir. Bu zorluk, NC kodlarının yaratılmasında ve takım yolu doğrulamasındaki karmaşıklıktan dolayı ortaya çıkar. Beş-eksen tezgahlarda tezgah parçalarının veya düzenek parçalarının birbirine çarpması çok ciddi bir problemdir ve genellikle yetersiz son-işlemcilerden veya iyi hazırlanmış bir simülasyon ve doğrulama programı olmadığı için yetersiz şekilde gerçekleştirilen çarpma kontrol işlemlerinden kaynaklanmaktadır. Beş-eksen tezgah sahipleri bu durumdan, ticari son-işlemci, simülasyon ve doğrulama programları olarak kurtulabilirler. Fakat, bu programlar pahalıdır ve küçük ve orta boy işletmeler (KOBİ'ler) genellikle bu yazılımların maliyetini karşılayamaz. Ticari yazılımların ilgili kütüphanelerinde KOBİ'ler için genellikle gereksiz olan oldukça fazla sayıda CNC tezgahı bulunmaktadır. Bu problemin üstesinden gelmenin bir alternatifi, her bir belirli

beş-eksen CNC tezgah için son-işlemeyi gerçekleştirebilen, beş-eksen freze işlemlerinin simülasyon ve doğrulamasını yapabilen özel yazılımların geliştirilmesidir.

Bu çalışmada, ODTÜ-BİLTİR Merkezinde yer alan “Mazak Variaxis 630 5X” CNC beş-eksen freze tezgahı için, son-işleme ve kesme simülasyon işlemlerini gerçekleştiren “*Manus 1.0*” isimli bir yazılım geliştirilmiştir. Ayrıca, oluşabilecek çarpmaları bulmak için takım yolu doğrulama algoritmaları geliştirilmiştir. Yazılım Borland C++ Builder5.0 dilinde yazılmıştır. Geliştirilen yazılım bazı örnek frezeleme işlemlerinde denenmiş ve tatmin edici sonuçlar elde edilmiştir.

Anahtar Sözcükler: Beş-Eksen Son-İşlemci, Freze Simülasyonu, Takım Yolu Doğrulama, Çarpma Bulma, Çarpma İkazı

To My Family,

ACKNOWLEDGEMENTS

I express sincere appreciation to Prof. Dr. Mustafa İlhan Gökler for his guidance, systematic supervision and insight throughout the study.

I wish to thank to Mr. Serdar Güler and Mr. Yiğit Yıldırım. The support of them is gratefully acknowledged.

I also would like to thank to METU-BİLTİR Research & Application Center for the facilities provided for my work.

Special thanks go to my colleagues, Özkan İlkün, Arda Çelik, Emine Ün, İlker Durukan, Mehmet Maşat, Ömer Köktürk, Pelin Sarı, Sevgi Saraç and Halit Şahin for their valuable support and aid; to my senior colleagues Barış Civelekoğlu, Barış Karagözler, Mehmet Tunç and Sertan Gülbahar for their support and guidance.

I offer sincere thanks to my parents, Gülsüm and Şükrü Cengiz, my sister Hande Cengiz and my brother Özgür Cengiz for their encouragement and faith in me.

TABLE OF CONTENTS

ABSTRACT.....	IV
ÖZ	VI
ACKNOWLEDGEMENTS	IX
TABLE OF CONTENTS.....	X
LIST OF TABLES	XIV
LIST OF FIGURES	XV
LIST OF SYMBOLS	XX
CHAPTERS	
1.INTRODUCTION	1
1.1 Five-Axis Milling.....	1
1.2 Types of Five-Axis Milling Machine Tools	3
1.3 The Principles of Performing Five-Axis Milling.....	8
1.3.1 Modeling of Workpieces	10
1.3.2 Tool Path Generation for Milling Operations	10
1.3.3 Postprocessing	12
1.3.4 Milling Simulation and Collision Detection	15
1.4 Scope of the Thesis	21
2.GENERAL STRUCTURE OF THE DEVELOPED SOFTWARE.....	23
2.1 Introduction	23
2.2 Computer Environment Used For Software Development	23

2.3 Software Requirements and Capabilities	24
2.4 Overview of the Software Structure.....	26
2.5 Postprocessing Part of the Software.....	27
2.5.1 User Definitions Software Component	28
2.5.2 CL-Data Compiler Software Component.....	29
2.5.3 Geometric Calculations Software Component.....	29
2.5.4 NC-Code (G-Code) Generation Software Component	30
2.6 Simulation Part of the Software	30
2.6.1 User Definitions Software Component	30
2.6.2 Motion File Compiler Software Component.....	32
2.6.3 Simulation Display Software Component.....	32
2.6.3.1 Simulation Play Control	34
2.6.3.2 Simulation View Control.....	35
2.6.3.3 View Display Settings	36
2.7 Verification Part of the Software	38
2.7.1 User Definitions Software Component	38
2.7.2 Collision Check Computation Software Component.....	39
2.7.3 Collision Check Results Software Component	39
2.8 Closure	40
3.POSTPROCESSOR PART OF THE SOFTWARE	41
3.1 Introduction.....	41
3.2 A-Axis Calculations	47
3.2.1 Algorithm for Detecting A-Axis Sign Changes	48
3.2.2 Algorithm for Detecting Limit Exceeding in A-Axis	50
3.3 C-Axis Calculations	53
3.3.1 Algorithm for Finding Number of Rotations in C-Axis.....	54
3.4 X-Axis Calculations	56
3.5 Y-Axis Calculations	56
3.6 Z-Axis Calculations	58
3.7 Closure	58

4.SIMULATION PART OF THE SOFTWARE	59
4.1 Introduction	59
4.2 Displaying CAD Models in Manus 1.0.....	60
4.3 Displaying Simulation Components	64
4.3.1 Machine Tool Components	64
4.3.2 Setup Components.....	65
4.3.2.1 Workpiece Model	65
4.3.2.2 Tooling Equipments	66
4.3.2.3 Fixtures and Clamping Elements	68
4.4 Positional Adjustments of Machine Tool and Setup Components	71
4.4.1 Positional Adjustment of Workpiece Model	71
4.4.2 Positional Adjustment of Fixtures.....	72
4.4.3 Positional Adjustment of Clamping Elements	74
4.5 Motion Generation	75
4.6 Displaying Machining Simulation	77
4.7 Closure	78
5.VERIFICATION PART OF THE SOFTWARE.....	79
5.1 Introduction	79
5.2 Mathematical Background for Collision Detection	80
5.2.1 Ray-Cylinder Intersection	82
5.2.1 Ray-Sphere Intersection	83
5.3 Algorithm for Detecting Collisions of Flat endmill.....	84
5.4 Algorithm for Detecting Collisions of Ball Endmill.....	88
5.5 Collision Warnings	92
5.6 Closure	93
6.SAMPLE APPLICATIONS	94
6.1 Introduction	94
6.2 Machining Setup Preparation.....	94
6.3 Sample Milling Operations	99
6.3.1 Application-1	99

6.3.2 Application-2.....	101
6.3.3 Application-3.....	103
6.3.4 Application-4.....	105
6.3.5 Application-5.....	107
6.4 Discussion on Machining Results	109
6.5 Closure	111
7.CONCLUSIONS AND FUTURE WORK	112
7.1 Conclusions.....	112
7.1.1 Capabilities of the Developed Software.....	114
7.1.2 Limitations of the Developed Software	116
7.2 Future Work	117
REFERENCES	120
APPENDICES	
A.TECHNICAL SPECIFICATIONS OF MAZAK VARIAXIS 630-5X	125
B.LIBRARY OF MANUS 1.0 FOR TOOL HOLDERS.....	128
C.LIBRARY OF MANUS 1.0 FOR CLAMPING ELEMENTS	136
D.THE COLLISION CHECK PROCESS.....	141

LIST OF TABLES

TABLES

Table 1.1 Classification of Five-Axis Machine Tools	5
Table 3.1 Conditions for A-Axis Sign Changes	50
Table 3.2 Forms of C-Axis calculation equations	54
Table 3.3 Forms of X-Axis calculation equations	56
Table 3.4 Forms of Y-Axis calculation equations	57
Table A.1 Technical features of Mazak Variaxis 630-5X	125

LIST OF FIGURES

FIGURES

Figure 1.1 Vertically hidden (unreachable) zone.....	2
Figure 1.2 Scallop reduction in five-axis milling	3
Figure 1.3 Rotary Axis Labeling	4
Figure 1.4 Structure of a machine tool having rotating/tilting table at B, C axes	6
Figure 1.5 Linear and rotary axes of “Mazak Variaxis 630 5X” machine tool	6
Figure 1.6 Structure of a machine tool having rotating/tilting spindle head at A, C axes	7
Figure 1.7 Structure of a machine tool with rotating table at C-Axis and rotating spindle head at A-Axis	8
Figure 1.8 Flow chart showing of five-axis milling preparation processes.....	9
Figure 1.9 Selection of machine tool type in G-Post module of ProEnginner	14
Figure 1.10 Simulation views from Manus 1.0 (a) collision occurs, (b) collision does not occur.....	16
Figure 1.11 A sample simulation view from Vericut	17
Figure 1.12 A sample simulation view from EdgeCAM	18
Figure 1.13 A sample simulation view from Esprit	18
Figure 2.1 Schematic diagram of the software structure	27
Figure 2.2 Machining reference coordinate system.....	29
Figure 2.3 Sample simulation views from Manus 1.0 (a) Front-Upper-Right view (b) Front-Lower-Right view (c) Back-Upper-Left view	33
Figure 2.4 Simulation Play Control window in Manus 1.0	34
Figure 2.5 Simulation Control Buttons.....	35
Figure 2.6 “View Display Settings” window of Manus 1.0 software.....	36

Figure 2.7 Sample simulation views from Manus 1.0 (a) table support is opaque (b) table support is translucent	37
Figure 2.8 Sample simulation view where cutting tool collides to a nut.....	40
Figure 3.1 Steps of postprocessing process	41
Figure 3.2 Sample simulation views from Manus 1.0 where (a) $A=0^\circ$ (no table tilting motion) (b) $A=-90^\circ$ and (c) $A=30^\circ$	43
Figure 3.3 Sample simulation views from Manus 1.0 where (a) $C=0^\circ$ (no table rotation) (b) $C=-45^\circ$ and (c) $C=-90^\circ$	44
Figure 3.4 The position of the cutting tool specified in the sample motion command line.....	45
Figure 3.5 Simulation view for “Mazak Variaxis 630 5X” in Manus 1.0	46
Figure 3.6 Phase reverse algorithm for avoiding tool interference	51
Figure 3.7 Tool interference (gouge) during the phase reverse process	52
Figure 4.1 A sample STL formatted CAD model	60
Figure 4.2 The definition of chord height	61
Figure 4.3 Sample STL formatted CAD models with (a) medium chord height, (b) fine chord height	62
Figure 4.4 A sample triangular facet in STL file format	63
Figure 4.5 General triangular facet forming program code	64
Figure 4.6 Machine tool predefined components in Manus 1.0	65
Figure 4.7 “Workpiece Model Select” window of Manus 1.0 software.....	66
Figure 4.8 “Tool Settings” window of Manus 1.0 software	67
Figure 4.9 Tooling equipment length.....	67
Figure 4.10 “Tooling Preview” window of Manus 1.0 software.....	68
Figure 4.11 “Fixture Settings” window of Manus 1.0 software	70
Figure 4.12 “Clamping Element Settings” window of Manus 1.0 software.....	71
Figure 4.13 “Machine Setup Preview” window of Manus 1.0 software displayed after workpiece location is determined.....	72
Figure 4.14 “Machine Setup Preview” window of Manus 1.0 software displayed after fixture location is determined.....	73
Figure 4.15 “Machine Setup Preview” window of Manus 1.0 software displayed after the locations of clamping elements are determined	75

Figure 4.16 Schematic description of point generation process on motion path ...	77
Figure 5.1 Finding ray equations for triangular facets.....	81
Figure 5.2 A ray intersection with cylinder where t values are greater than 1	86
Figure 5.3 A ray intersection with cylinder where one t value is in the interval of 0 and 1	86
Figure 5.4 The end point of the ray segment is above the cylinder base plane	87
Figure 5.5 The ray segment is completely below the cylinder base plane	87
Figure 5.6 The ray segment is completely inside the cylinder	88
Figure 5.7 A ray intersection with sphere where t values are greater than 1	90
Figure 5.8 A ray intersection with sphere where one t value is in the interval of 0 and 1	90
Figure 5.9 The end point of the ray segment is above sphere minimum Z-Axis value.....	91
Figure 5.10 The ray segment is below sphere minimum Z-Axis value.....	91
Figure 5.11 The ray segment is completely inside the sphere	92
Figure 5.12 Protective shields around cutting tools.....	93
Figure 6.1 Workpiece model prepared for sample machining operations	95
Figure 6.2 Dimensions of sample workpiece.....	95
Figure 6.3 Machining setup (View-1).....	96
Figure 6.4 Machining setup (View-2).....	96
Figure 6.5 Machining setup (View-3).....	97
Figure 6.6 Prepared machining setup in Manus 1.0 (View-1).....	97
Figure 6.7 Prepared machining setup in Manus 1.0 (View-2).....	98
Figure 6.8 Prepared machining setup in Manus 1.0 (View-3).....	98
Figure 6.9 Sample simulation view from ProEngineer for “Application-1”	100
Figure 6.10 Sample portion of generated NC-Code for “Application-1”	100
Figure 6.11 Sample simulation view from Manus 1.0 for “Application-1”	101
Figure 6.12 Sample simulation view from ProEngineer for “Application-2”	102
Figure 6.13 Sample portion of generated NC-Code for “Application-2”	102
Figure 6.14 Sample simulation view from Manus 1.0 for “Application-2”	103
Figure 6.15 Sample simulation view from ProEngineer for “Application-3”	104
Figure 6.16 Sample portion of generated NC-Code for “Application-3”	104

Figure 6.17 Sample simulation view from Manus 1.0 for “Application-3”	105
Figure 6.18 Sample simulation view from ProEngineer for “Application-4”	106
Figure 6.19 Sample portion of generated NC-Code for “Application-4”	106
Figure 6.20 Sample simulation view from Manus 1.0 for “Application-4”	107
Figure 6.21 Sample simulation view from ProEngineer for “Application-5”	108
Figure 6.22 Sample portion of generated NC-Code for “Application-5”	108
Figure 6.23 Sample simulation view from Manus 1.0 for “Application-5”	109
Figure 6.24 Optic scanner result (scale is in millimeter)	111
Figure A.1 Machine Coordinate System and table center locations on “Mazak Variaxis 630-5X”	126
Figure A.2 X-Axis, Z-Axis strokes and rotation axis location of C-Axis on “Mazak Variaxis 630-5X”	127
Figure A.3 Y-Axis stroke and rotation axis location of A-Axis on “Mazak Variaxis 630-5X”	127
Figure B.1 “HSK A 63 ER16X120M E” (a) photograph (b) CAD model	129
Figure B.2 Dimensions of “HSK A 63 ER16X120M E”	129
Figure B.3 “HSK 63-2/20-160” (a) photograph (b) CAD model	130
Figure B.4 Dimensions of “HSK 63-2/20-160”	130
Figure B.5 “HSK A63 ER16X160 M E” (a) photograph (b) CAD model	131
Figure B.6 Dimensions of “HSK A63 ER16X160 M E”	131
Figure B.7 “HSK A63 ER32X120 E” (a) photograph (b) CAD model	132
Figure B.8 Dimensions of “HSK A63 ER32X120 E”	132
Figure B.9 “HSK A63 ER32X100 E” (a) photograph (b) CAD model	133
Figure B.10 Dimensions of “HSK A63 ER32X100 E”	133
Figure B.11 “HSK A63 ER32X80 E” (a) photograph (b) CAD model	134
Figure B.12 Dimensions of “HSK A63 ER32X80 E”	134
Figure B.13 “HSK A63 EM25X110 E” (a) photograph (b) CAD model	135
Figure B.14 Dimensions of “HSK A63 EM25X110 E”	135
Figure C.1 Clamping element components	136
Figure C.2 “Clamp Type-1” (a) photograph (b) CAD Model	137
Figure C.3 “Clamp Type-2” (a) photograph (b) CAD Model	137
Figure C.4 “Clamp Type-3” (a) photograph (b) CAD Model	138

Figure C.5 Dimensions of step clamps in METU-BILTIR Center.....	138
Figure C.6 “Block Type-1” (a) photograph (b) CAD Model.....	139
Figure C.7 “Block Type-2” (a) photograph (b) CAD Model.....	139
Figure C.8 “Block Type-3” (a) photograph (b) CAD Model.....	140
Figure C.9 Dimensions of step blocks in METU-BILTIR Center.....	140
Figure D.1 Flow chart of collision check process.....	142

LIST OF SYMBOLS

SYMBOL

A	: machine tool current A-Axis position
A_0	: machine tool previous A-Axis position
A_c	: second degree variable of quadratic equation of ray cylinder intersection
A_s	: second degree variable of quadratic equation of ray sphere intersection
B_c	: first degree variable of quadratic equation of ray cylinder intersection
B_s	: first degree variable of quadratic equation of ray sphere intersection
C	: machine tool current C-Axis position
C_0	: machine tool previous C-Axis position
C_c	: zeroth degree variable of quadratic equation of ray cylinder intersection
C_s	: zeroth degree variable of quadratic equation of ray sphere intersection
CTV	: calculated “t” values
CX	: X-Axis location of a point on cylinder surface
CX_0	: X-Axis location of cylinder center axis
CY	: Y-Axis location of a point on cylinder surface

CY_0	: Y-Axis location of cylinder center axis
D	: direction vector of ray
D_x	: X-Axis component of ray direction vector
D_y	: Y-Axis component of ray direction vector
D_z	: Z-Axis component of ray direction vector
I	: current tool orientation vector X-Axis component
I_0	: previous tool orientation vector X-Axis component
J	: current tool orientation vector Y-Axis component
J_0	: previous tool orientation vector Y-Axis component
K	: current tool orientation vector Z-Axis component
K_0	: previous tool orientation vector Z-Axis component
P	: collision check approach tolerance
$P(t)$: point on a ray
R	: radius of cutting tool
S	: Starting point of ray
S_x	: X-Axis component of ray starting point
SX	: X-Axis location of a point on sphere surface
SX_0	: X-Axis location of sphere center
S_y	: Y-Axis component of ray starting point
SY	: Y-Axis location of a point on sphere surface
SY_0	: Y-Axis location of sphere center
S_z	: Z-Axis component of ray starting point
SZ	: Z-Axis location of a point on sphere surface
SZ_0	: Z-Axis location of sphere center
t	: ray equation variable

U	: tool tip X-Axis position
V	: tool tip Y-Axis position
W	: tool tip Z-Axis position
WCS	: workpiece coordinate system
WCX	: workpiece coordinate system X-Axis component
WCY	: workpiece coordinate system Y-Axis component
WCZ	: workpiece coordinate system Z-Axis component
X	: machine tool current X-Axis position
Y	: machine tool current Y-Axis position
Z	: machine tool current Z-Axis position
Δ	: discriminant

CHAPTER 1

INTRODUCTION

1.1 Five-Axis Milling

Today, five-axis CNC milling is an extensively used technology for manufacturing valuable parts of complex 3D sculptured surfaces in aerospace, automotive, and die/mold industries and many researchers have dedicated their efforts towards expanding the use of five-axis machine tools. It is certain that this trend will continue to increase, because of the variety of advantages that five-axis machining brings to productivity and product quality, and also because of the essential features that it gets to produce more complex parts.

Five-axis machine tools have ability to simultaneously position and orient cutting tools and workpiece in a coordinate system defined in the workspace by adjusting the positions of five motion axes. This characteristic makes the machine tools very versatile.

The most important advantage of five-axis machining is the ability to machine complex workpieces in single setup. This increases machining productivity if compared to executing the operation in a series of setups, besides significantly reduces the time and cost consumed for preparing fixtures. In addition, in three-axis machine tools, there is always a possibility of incorrect alignment when the part is each time remounted using multiple setups.

Another significant benefit of five-axis machining is that shorter cutting tools can be utilized in the five-axis machine tools, since the cutting tool orientation can be adjusted with respect to the workpiece surface normal and therefore the spindle head can be lowered more towards the workpiece. This reduces the risk of unwanted vibration during machining. Thus, cutting speeds

can be increased, which causes reduction in production times. In addition, breakages and early wears on cutting tools resulting from tool vibration are lowered with shorter cutting tools. This allows higher quality surface finishes, and eliminates the need for time-consuming hand finishing.

Five-axis machining can also decrease the machining times consumed in the drilling of holes on 3D curved surfaces. Drilling these holes is difficult and extremely time consuming when performed in three-axis CNC machine tools, since it may require several setups. However, in a five-axis machine tool, the cutting tool can be positioned in desired orientations for each hole.

In some workpieces, vertically hidden (i.e. unreachable) zones, which can not be machined in three-axis machine tools, exist as indicated in Figure 1.1. These hidden surfaces can easily be machined in five-axis machine tools as shown in this figure [1].

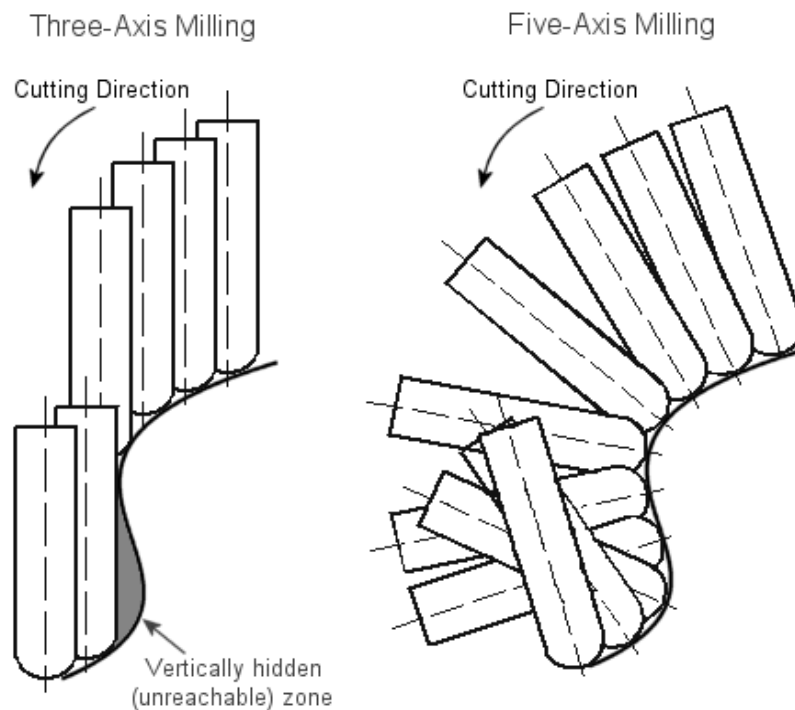


Figure 1.1 Vertically hidden (unreachable) zone

The scallops formed with ball endmills during three-axis finish milling operations are also problem as shown in Figure 1.2. Five-axis machine tools allow the user to keep the cutting tool in any desired orientation with respect to machined surface. This is especially useful in die and mold finish milling operations where high quality surfaces are required. In five-axis machine tools, instead of ball endmills, flat endmill tools can be utilized for finish milling operations. The scallops can be reduced by proper adjustments on the cutting tool orientation as shown in Figure 1.2 [1].

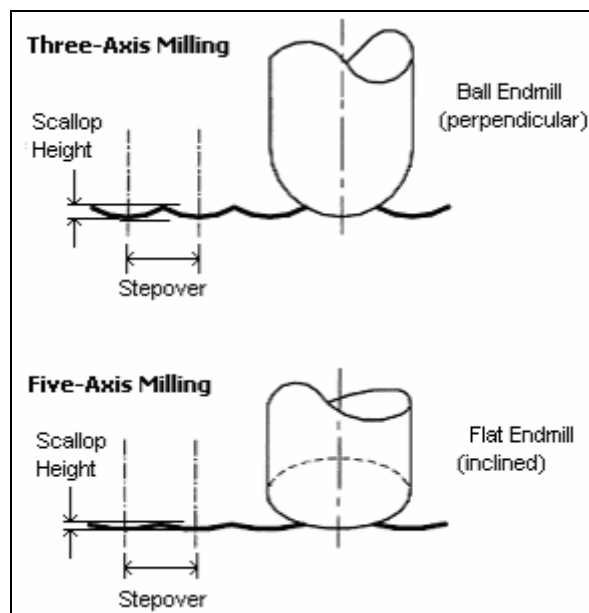


Figure 1.2 Scallop reduction in five-axis milling

1.2 Types of Five-Axis Milling Machine Tools

A five-axis machine tool has five degree of freedom. This five degree of freedom is achieved with five independently working motors. These five motors set the machine in motion in five separate motion axes. In most of the five-axis machine tools, these motion axes are three linear axes (X, Y and Z), and two rotary axes [2].

The three linear axes are the same for all five-axis machine tools. Linear motions are executed in these three linear axes; X, Y and Z axes. The orientation of these linear axes may depend on the machine tool configurations. In addition to that, these three linear motions can be activated differently depending on the machine tool configurations. In some machine tools, all of these linear motions are activated by spindle head, while in some other machine tools; some or all linear motions can be performed by the table instead. The variations in five-axis machine tools mainly depend on the differences in rotary axes. By convention, rotary axes are labeled according to the orientation of the rotation axes as shown in Figure 1.3.

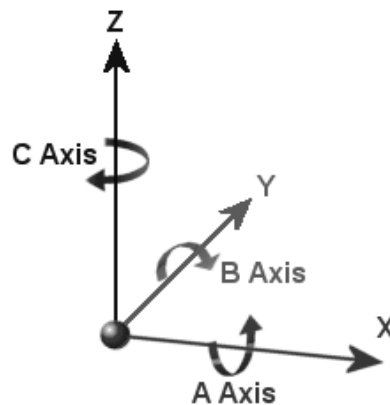


Figure 1.3 Rotary Axis Labeling

The five-axis machine tools are generally classified according to their two rotary axes configurations. Excluding machines with non-orthogonal axes (e.g. the so-called “nutating” arrangements), there are basically three types of five-axis machine tools classified according to the actuation types of rotary axes [3]. These are;

- a. Five-Axis machine tools with the rotating and tilting table,
- b. Five-Axis machine tools with the rotating and tilting spindle head,
- c. Five-Axis machine tools with the rotating table and the tilting spindle head.

All five-axis machine tool configurations adapt one of these configuration types. These three main classes can be categorized in twelve subgroups according to the names of the rotary axes as shown in Table 1.1.

Table 1.1 Classification of Five-Axis Machine Tools

Classification-1	Classification-2	Rotary/Tilting Axes
Rotating and Tilting Table	Type-1	A-Axis (table) and B-Axis (table)
	Type-2	A-Axis (table) and C-Axis (table)
	Type-3	B-Axis (table) and C-Axis (table)
Rotating and Tilting Spindle Head	Type-4	A-Axis (spindle head) and B-Axis (spindle head)
	Type-5	A-Axis (spindle head) and C-Axis (spindle head)
	Type-6	B-Axis (spindle head) and C-Axis (spindle head)
Rotating Table, Tilting Spindle Head	Type-7	A-Axis (table) and B-Axis (spindle head)
	Type-8	A-Axis (table) and C-Axis (spindle head)
	Type-9	B-Axis (table) and A-Axis (spindle head)
	Type-10	B-Axis (table) and C-Axis (spindle head)
	Type-11	C-Axis (table) and A-Axis (spindle head)
	Type-12	C-Axis (table) and B-Axis (spindle head)

a) Five-Axis Machine Tools with the Rotating and Tilting Table: In these types of five-axis machine tools, both rotary motions are provided by a rotating and tilting table. Rotating and tilting axes can be different depending on the machine tool configuration. There are three different configurations (Type-1, Type-2 and Type-3) of this type of machine tools as shown in Table 1.1. The structure of a five-axis machine tool having a rotating and tilting table is shown in Figure 1.4. The machine tool shown in this figure is Type-3 in Table 1.1 and has rotary axes at B and C.

The five-axis “Mazak Variaxis 630 5X” CNC machine tool that is located in METU-BILTIR Center has a rotating and tilting table and it is a Type-2 machine tool. That means rotating axes of this machine tool are A and C axes. The three linear motions are provided via spindle head. All the linear axes and rotary axes of “Mazak Variaxis 630 5X” machine tool are shown in Figure1.5 and machine technical specifications are given in Appendix A.

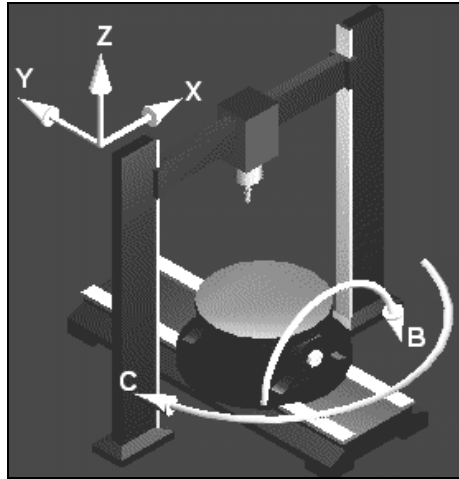


Figure 1.4 Structure of a machine tool having rotating/tilting table at B, C axes [4]

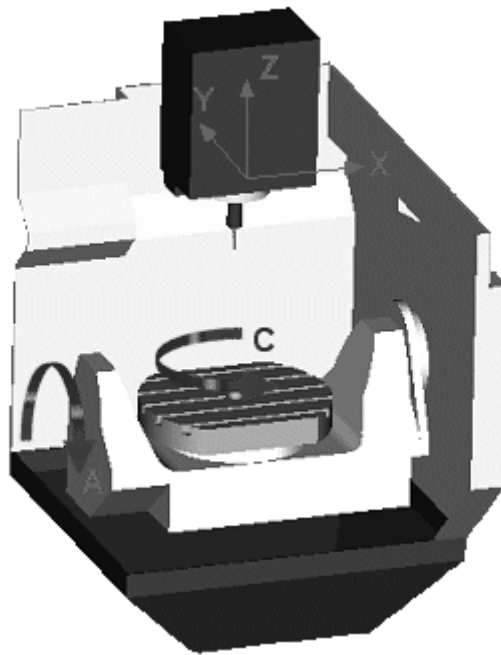


Figure 1.5 Linear and rotary axes of “Mazak Variaxis 630 5X” machine tool

Five-axis machine tools having rotating and tilting tables are known as the most popular five-axis machine tools since they can realize five degree of freedom just by an added rotating and tilting mechanism on the table of three-axis milling machines. This makes them easier to be designed and produced for CNC machine tool producers if compared to other two types. The major advantages of this type are as follows; the load imposed on the spindle is less than other types

since the spindle does not tilt, in addition, the circular shapes and profiles can be machined quite well with a rotating table about the C-axis. However, heavy workpieces are not adequate for this type due to large torque needed for rotating and tilting table [5].

b) Five-Axis Machine Tools with the Rotating and Tilting Spindle Head:

In these types of five-axis CNC milling machines, the two rotary motions are provided via spindle head. The major advantage of this type of machine tools is that heavy parts can be machined more easily when compared to other two main classes, since there is no need to rotate or tilt heavy workpieces on the table [5]. The structure of a five-axis machine tool having a rotating and tilting spindle head is shown in Figure 1.6. The machine tool shown in this figure is Type-5 in Table 1.1 and has rotary axes at A and C. In addition to that, as can be seen from Figure 1.6, all three linear motions in X, Y and Z axes are provided by the table instead of the spindle head.

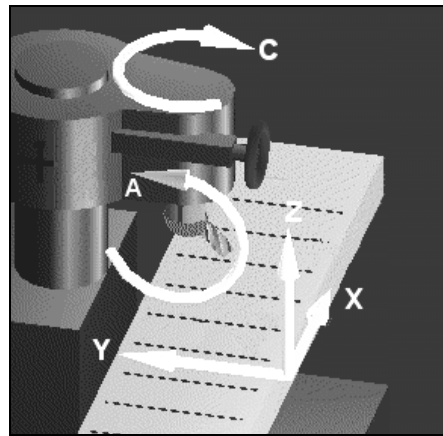


Figure 1.6 Structure of a machine tool having rotating/tilting spindle head at A, C axes [4]

c) Five-Axis Machine Tools with the Rotating Table and the Rotating Spindle Head: In these types of five-axis CNC milling machines, one rotary motion is provided by the machine table, while the other rotary (tilting) motion is provided via the spindle head. There are six different types having this configuration as shown in Table 1.1. The structure of a five-axis machine tool

having a rotating table and a rotating spindle head is shown in Figure 1.7. The machine tool shown in this figure is Type-11. The rotary axes of this machine are A-Axis (spindle head) and C-Axis (table). In this sample, the linear motions in X and Y axes are provided by the table, while the Z-Axis motion is provided by the spindle head.

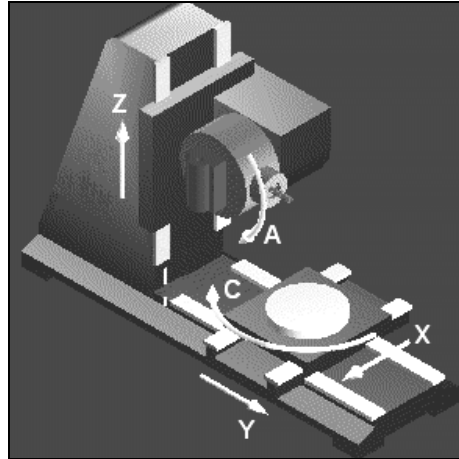


Figure 1.7 Structure of a machine tool with rotating table at C-Axis and rotating spindle head at A-Axis [4]

1.3 The Principles of Performing Five-Axis Milling

The five-axis milling technology although mainly depends on the developments in the CNC machine tool design industry, the success of five-axis machining also depends on the software tools which enable to model the workpiece, to generate the cutter tool path and NC-Codes and to simulate and verify the milling processes. A five-axis machine tool has limited use in the absence of these special purpose software tools. To illustrate the way of how to work with five-axis machine tools and the principles of performing five-axis milling operations, a flow chart is prepared that is shown in Figure 1.8.

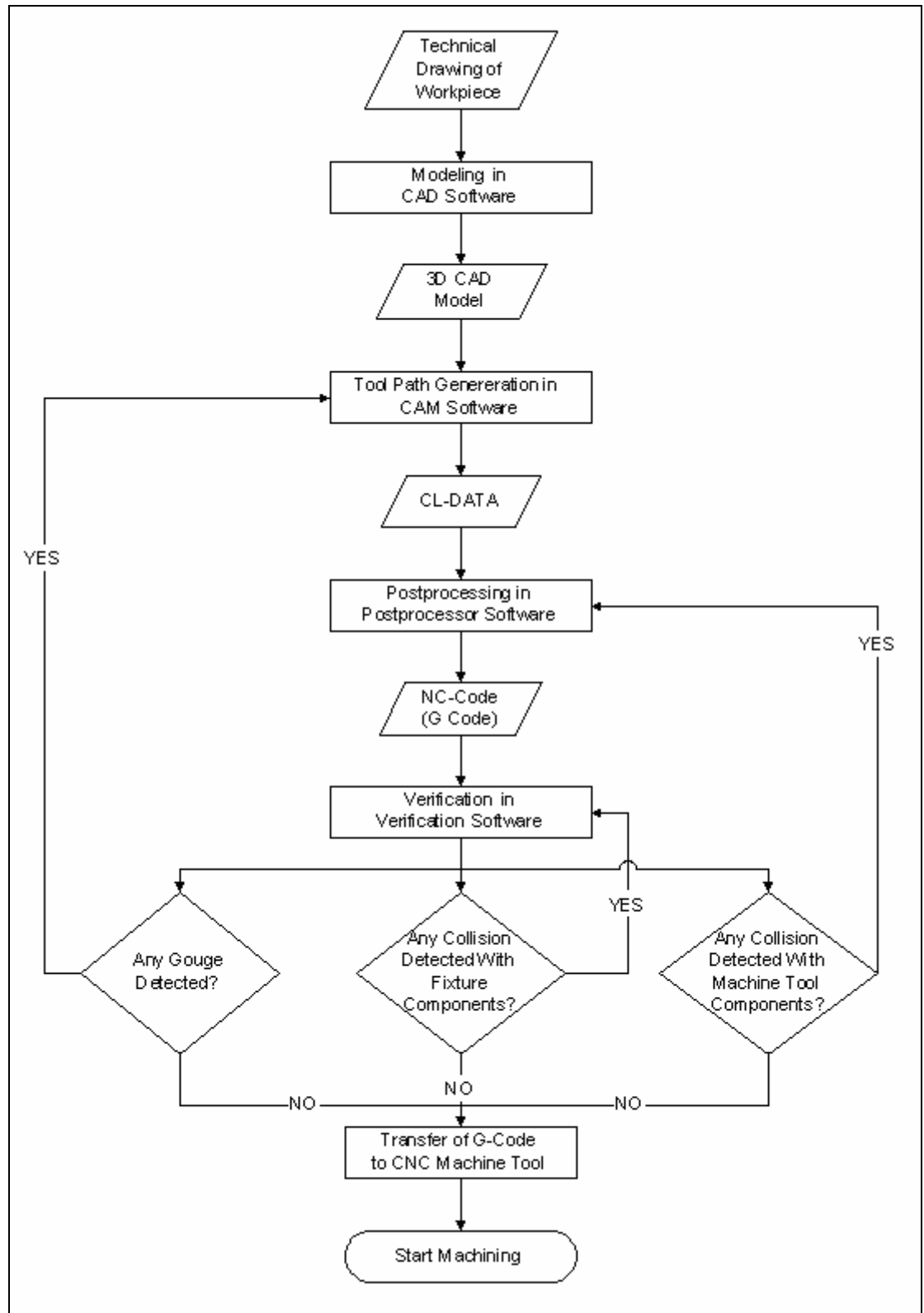


Figure 1.8 Flow chart showing of five-axis milling preparation processes

All the steps needed to work with five-axis machine tools will be described in the following subsections.

1.3.1 Modeling of Workpieces

As shown in Figure 1.8, the first step of machining preparation process is the creation of a 3D model of the required workpiece by using Computer Aided Design (CAD) software. According to the existing technical drawing or the given dimensional information, the 3D model is prepared. Today, there are several commercially available CAD software programs [6, 7 and 8] on the market and CAD modeling becomes a standard operation not only in manufacturing industry, and also in many areas involving engineering studies.

1.3.2 Tool Path Generation for Milling Operations

Since CNC machine tools can capable of producing more complex workpieces than conventional ones, tool path generation for CNC machine tools has become important. Tool path generation for sophisticated parts is complicated; therefore special software programs are needed to achieve this task. The software programs that are developed for this purpose are called Computer Aided Manufacturing (CAM) software programs. By means of these CAM programs, complex parts can be machined more easily.

Today there are many commercially available CAM programs in the market. Some of these programs are standalone CAM programs, like Tebis [9], EdgeCAM [10] and PowerMILL [11] and some of them are a part of a CAD/CAM program like ProEngineer [6], Unigraphics [7] and Catia [8]. The workpiece model, which is prepared using CAD software program, is transferred to CAM software in a suitable data transfer format. After that, the raw material, which the workpiece will be machined from, is defined. The initial raw material can be in various forms. It can be a forged, cast or a block material having a cylindrical or prismatic geometry.

All the necessary parameters are input to the CAM programs, these parameters are; the selection of suitable tool and tool diameter, the depth of cut and width of cut for each cuts, the selected surfaces to be machined for finishing operations and the cutting feed of the tool. In addition to this basic information, CAM software also allows the user to specify some other specific information for advanced applications. For instance, with a conventional milling machine tool, it depends on the operators' ability to adjust tool and workpiece first touch conditions; the plunge feed, the approach distance, the approach type and approach angle of the tool to the workpiece. This is almost impossible to adjust properly in a conventional milling machine. However, CNC machine tools can capable of adjusting these parameters with proper selections in CAM programs. While creating the tool path for machining, the CAM software user can easily specify this specific information.

Tool path generation is much more complicated for five-axis milling operations because of the nature of the five-axis milling processes. In a three-axis machine tool, the cutting tool always orients along Z-Axis with respect to the machine coordinate system. However, in a five-axis machine tool while performing advanced finishing operations, the cutting tool can be in any orientation with respect to the workpiece. The tool can be positioned exactly perpendicular or in any desired orientation as to the workpiece surfaces in the working space.

In CAM software programs, the user can adjust the position of the cutting tool in desired orientations for multi-axis milling operations. And several different combinations of different milling strategies with different cutting tool orientations are in use in industry for various purposes like to increase tool life or to increase surface quality [12, 13].

By means of CAM programs, the user can create the required tool path for machining operation. The generated tool path is in the CL (Cutter Location) Data Format [14]. And all CAM programs can create the tool path in this format. All the cutting information is stored in CL-Data file. The spindle speed, the cutting,

plunge and retract feedrates, the coolant condition and most importantly the motions that will be performed are all included in CL-Data file.

1.3.3 Postprocessing

As described in the previous section, the CL-Data file created using any CAM program includes the machining information, however, this format is not suitable to use in any machine tools directly. The CNC Machine tools recognize NC-Code (a.k.a. G-Code) format [15] for machining. Therefore CL-Data has to be converted to G-Code with an operation named “*postprocessing*”. Postprocessing is a step that converts the CL-Data format into G-Code format. The place of postprocessing in preparing machining codes is shown in Figure 1.8. The G-Code created will directly be used for machining. The G-Code is the recommended standard (EIA RS 274 D) for numerically controlled machines developed by the Electronic Industry Association (EIA).

Some other important issues must be performed while performing postprocessing. In the CL-Data format, machine tool parameters are not specified. Some of these machine tool parameters are; machine tool motion axes, travel limits of these motion axes and positions of rotating axes. These parameters are very important for postprocessing. For instance, when a five-axis CNC machine tool is considered which has a rotating and tilting table, the travel limits on rotational axes may cause changing of the required machining code (i.e. G-Code) extensively. For the same job, a machine tool with A-Axis limits of -90° and 0° will need a different G-Code and a machine tool with A-Axis limits of -30° , $+90^{\circ}$ will need a completely different G-Code. If the same G-Codes is used in different machine tools without considering the machine parameters, the result will be a fatal error, which can cause a collision between spindle, cutting tool, table or workpiece. The broken cutting tools, deformed workpieces and most importantly damaged machine tool components will cause high costs and big losses in productivity. Therefore, these machine parameters must be taken into account while creating a proper postprocessor for multi-axis milling operations [16].

The machine parameters are specific for each five-axis CNC machine tool. Even for machine tools with similar configurations, these machine parameters vary reasonably. Furthermore, today there are many five-axis machine tools, which have completely different configurations, on the market and these machine tools have totally different machine parameters. This means that any machining code that will be prepared have to be specific for the particular machine tool whether the created tool paths are the same or not. These circumstances make five-axis machining works more sophisticated and troublesome and make the postprocessing an important machine code preparation step and it is obvious that without a proper postprocessor, it is almost impossible to operate five-axis CNC machine tools.

There are commercially available software packages which are used to provide postprocessors for specific machine tools. Some of those are standalone postprocessor preparation programs like IntelliPost [17], CAM-POST [18] and IMSpost [19]. The required postprocessor can be configured by using these packages. CAM programs have also modules for postprocessing. These are interactive programs where the user can define technical specifications of the machine tool, and the package program creates the required postprocessor. The package program used in ProEngineer is G-Post [20]. In Figure 1.9, a sample view from the G-Post module of ProEngineer is shown. In this figure, a dialog box of G-Post is shown where the user can define machine tool type.

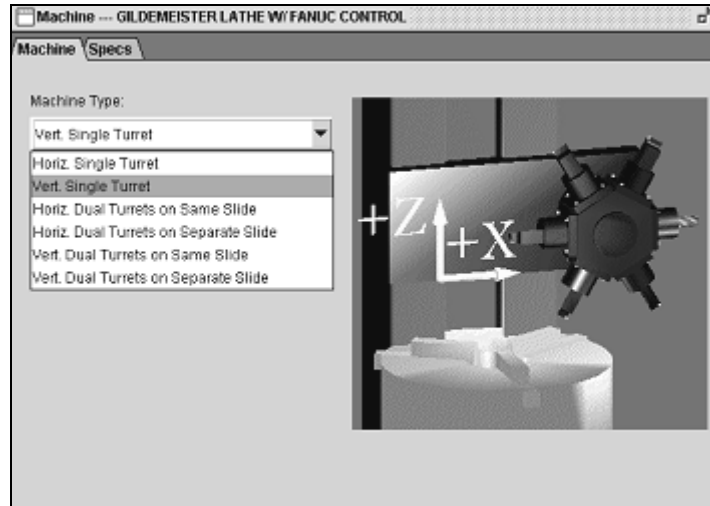


Figure 1.9 Selection of machine tool type in G-Post module of ProEngineer [6]

Postprocessor preparation capabilities of these programs are generally sufficient when postprocessors have to be designed for three-axis CNC milling machine tools or two-axis CNC lathes. However, their capabilities are limited and they are insufficient for complex five-axis postprocessor design. A postprocessor has been prepared by the author during this study in the G-Post module of ProEngineer for high speed “Mazak Variaxis 630-5X” five-axis milling machine tool, which is located in METU-BILTIR Center. However, it is observed that this postprocessor generates G-Codes with numerous fatal errors. The major error encountered, when G-Post is used to create G-Code files, occurs during the “phase reverse process” performed when the calculated positions of machine table at rotary axes pass beyond the limits. The “phase reverse process” will be described in detail in Chapter 3. The other problems that are observed mostly occur because of the “bugs” in the software. The specifications of this postprocessor generated using G-Post module are described in detail in Chapter 3 with the detailed description of the problems encountered.

Researches have been performed to overcome these difficulties encountered in multi-axis machining operations [5,21-23]. These researches are mainly focus on developing new strategies to avoid gouges or collisions occurred due to wrong postprocessing. Y.H. Jung et al. [5] studied on an algorithm which prevents the problems encountered during “phase reverse process”. J.P. Kruth et

al. [21] studied on developing a postprocessor integrated with a simulation package. In this software developed if any collisions are detected by the simulation package, the postprocessor automatically recalculates the machining axes positions in the alternate regions. M. Munlin et al. [22] develop an algorithm to optimize tool paths during postprocessing. Takeuchi and Watanabe [23] make researches on five-axis collision free tool path generation and postprocessing.

1.3.4 Milling Simulation and Collision Detection

In Figure 1.8, the place of tool path verification in machine code preparation process is shown. The purpose of tool path verification is to control the machining activity in a simulation environment before starting the job on the machine tool. The reason of simulating the machining activity is to detect and prevent any unwanted movements that may cause collisions between machine tool and setup components. The collision between cutting tool and workpiece is known as “gouging” and has a special importance, since gouging causes the deformation of workpiece and the cutting tool. By simulating the machining activity before starting the job on the machine tool, the user can realize any collisions and gouges via visual inspection. However, in most of the cases, it is hard to recognize collisions via visual inspection, therefore collision detection algorithms should be developed, which can be used in verification software programs. In addition to the visual inspection, by means of collision detection capabilities of verification software programs [24-27], user can precisely control the tool path. If user is sure that the postprocessor used has no defects, then the only reason of gouging is the error in generated tool paths. In such a case, the user goes back the tool path generation step as shown in Figure 1.8 and tries to correct the tool path, then starts to follow the same steps. If any collision is detected, the corrections should be done according to the collision type and reason. Collisions may occur because of the prepared machining setups. For instance, if a collision is detected between spindle and table, the probable reason for that is the insufficient fixture height where the workpiece is mounted on, as shown in Figure 1.10(a). In Figure 1.10(b), the height of the fixture is increased and the collision is prevented.

By increasing the fixture, the location of the workpiece is changed. That means postprocessing step has to be activated again as shown in Figure 1.8. Similarly, collisions with fixture components may also be eliminated by correcting the position of fixtures or clamping elements. In these cases, the verification process has to be reactivated as shown in Figure 1.8.

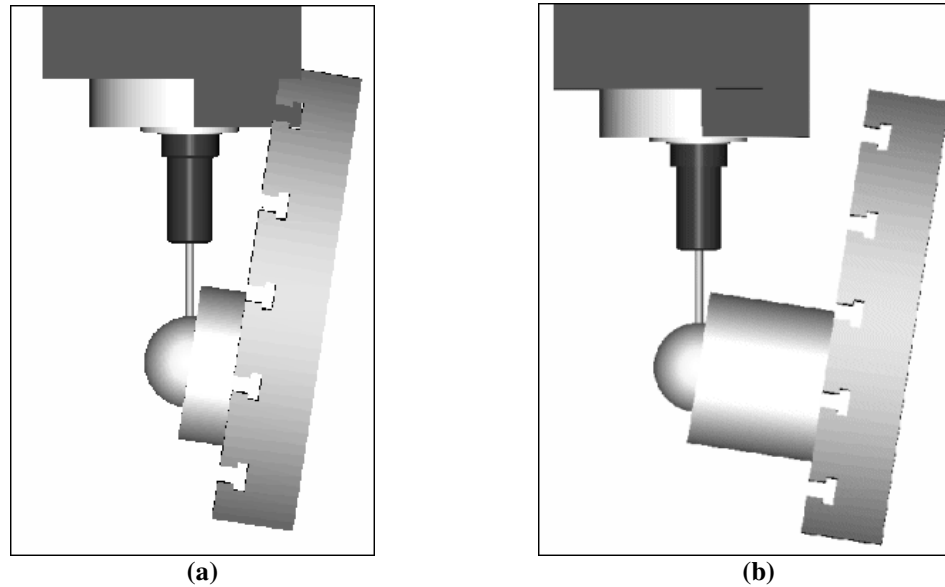


Figure 1.10 Simulation views from Manus 1.0 (a) collision occurs, (b) collision does not occur

The commercially available verification software and some CAM software can capable of simulating the tool paths in computer environment. Vericut [24] is one of the most popular verification programs in industry. The other examples are Predator Virtual CNC [25], NCsimul [26] and MachineWorks [27]. In Figure 1.11, a sample simulation view prepared for a Cincinnati milling center in Vericut is shown. Milling operations can be virtually simulated according to the data in the G-Code file. The simulation shows the milling operation from the raw material in start to the end of the workpiece model formation. With this visualization, the user can detect any significant and noticeable errors like whether the depth of cut and width of cut values are truly adjusted, whether the right tool is selected and most importantly whether there are any gouges or collisions. All these can be controlled in this simulation environment.

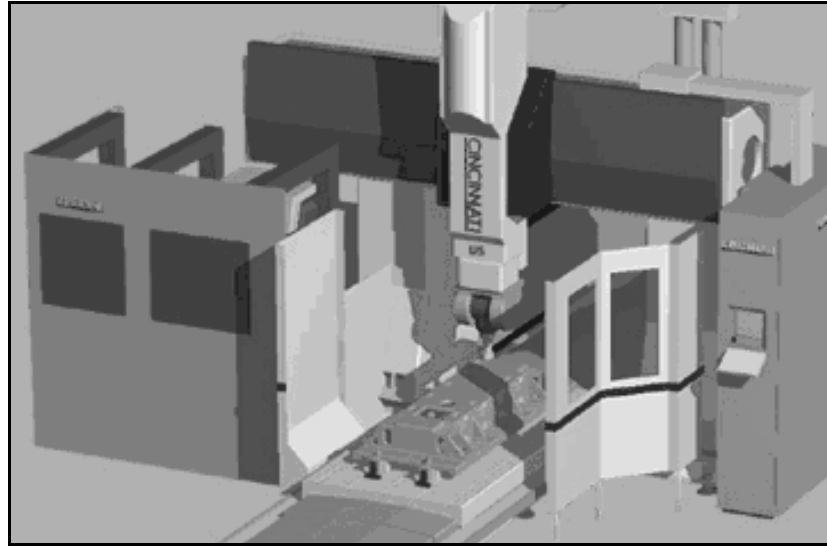


Figure 1.11 A sample simulation view from Vericut [24]

Verification programs can detect collisions for the machine tools existing in their machine tool library modules and bring great simplicity in controlling tool paths. However, these packages are very expensive and especially uneconomical for the small and medium enterprises (SMI) and university research centers that have limited number of CNC machine tools.

Visual inspection, collision and gouge detection capabilities of some CAM programs can be treated as sufficient for five-axis milling operations. These software programs involve the full machine setup components of some well-known machine tools and simulate exact milling operations. However, the machine tool libraries of these CAM programs are much more limited than machine tool libraries of the verification programs. The examples of these CAM software programs can be given as EdgeCAM [10], PowerMILL [11] and Esprit [28]. Sample simulation views taken from EdgeCAM and Esprit simulation environment are shown in Figure 1.12 and Figure 1.13 respectively. The machine tool shown in these figures are turning centers with milling capabilities.

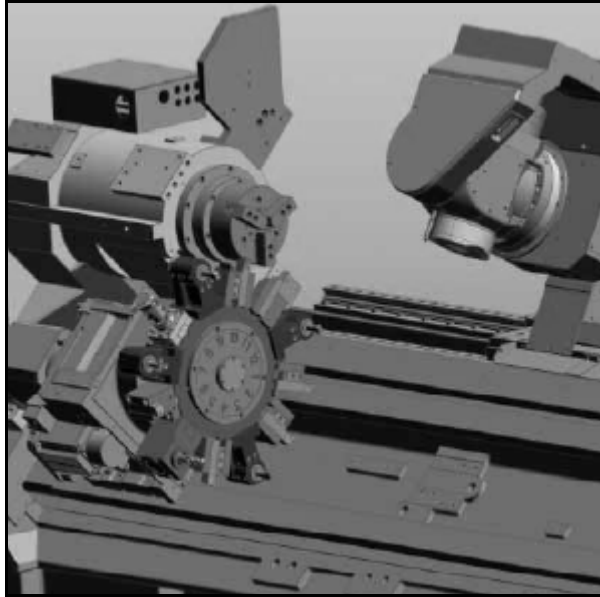


Figure 1.12 A sample simulation view from EdgeCAM [10]

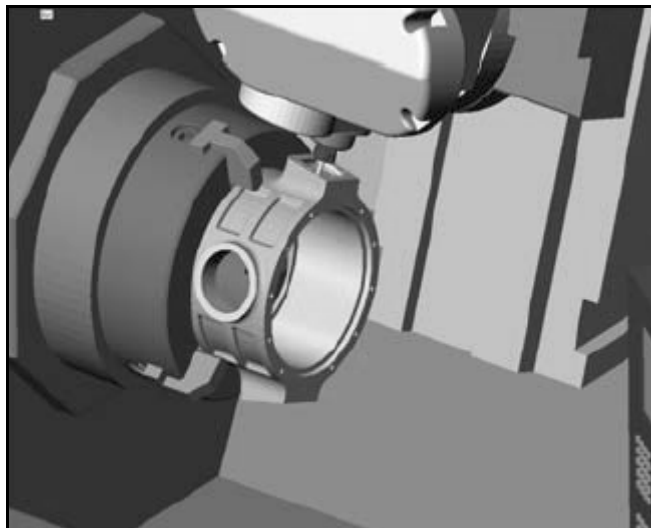


Figure 1.13 A sample simulation view from Esprit [28]

ProEngineer [6], Unigraphics [7] and Catia [8] are the most commonly used CAD/CAM packages. However, they have very limited NC simulation and verification capabilities. In these software programs, while gouge checking can be performed, it is impossible to detect collisions occurred during five-axis milling operations. While creating the tool path for five-axis milling operations, these CAD/CAM programs do not take into account the machine tool kinematics and it

is assumed that the rotating and tilting motions will be provided by cutting tool while workpiece stays as fixed. This makes the detection of collisions between machine components completely impossible.

Verification of tool path in five-axis milling operations needs advanced studies and the researchers have devoted their efforts to improve methods for detecting collisions faster and more precisely. Different methods have been developed for collision detection. One of these methods is the G-Buffer method developed by Saito and Takahashi [29]. They perform the tool path verification using G-buffer method. In this method, an array is constructed which is named as G-Buffer. This array contains geometric properties like depth (z) value and surface normal of the workpiece surface for each pixel. In order to find the overcut amount of the cutting tool, the depth values of the cutting tool during machining are calculated and these calculated depth values are subtracted from the G-buffer depth values of the workpiece for each pixel. If there is no calculated overcut then this means the collision does not occur between cutting tool and workpiece. This method can only be used in three-axis machining of surfaces which stay visible after projection is performed in X-Y plane. There is no possibility of using this method in five-axis tool path verification.

Van Hook [30] uses rectangular solid elements called “dexels” for NC simulation process. The dixel structure is formed by scanning the surface data of the displayed object. At each dixel, the farthest and nearest Z values and the color of the dixel are stored. In this approach, the cutting tool is treated as an object with negative dixel structure having different colors. The milling simulation process is performed by subtracting these cutting tool negative dexels from the workpiece dexels. The problem in that approach is that the construction of the dexels is limited by the viewing direction. Once the dexels are constructed, the coordinate system cannot be changed. Therefore, this approach can be usable in only three-axis milling simulations. Huang and Oliver [31] extend this approach by changing the coordinate system definition. In this approach, each dixel information is stored relative to an independent dixel coordinate system and this coordinate system can be transformed into either the world coordinate system or

the screen coordinate system via coordinate transformations. Based on this extended dixel representation, five-axis tool motions can be simulated and collisions between machine components may also be detected.

Sarma, and Adachi [32] worked on an algorithm which uses Constructive Solid Geometry (CSG) representation for the cutting tool geometry and utilizes a point-cloud representation technique for the workpiece. The equations that represent the cutting tool geometry are formed by applying binary space partition (BSP-Tree) method. The number of points in the point-cloud shows how accurately the workpiece model is approximated. In this algorithm, when any points from the point-cloud that represents workpiece fall inside the BSP-Tree defined cutting tool geometry, the collision occurs. However, in this method, it is very difficult to define the complex tool geometries and it also loses its efficiency when the point number in the point-cloud is increased.

E. L. J. Bohez et al. [33] develops a tool path verification method based on the sweep plane algorithm. In this method, by taking parallel slices from the workpiece with constant intervals, the intersections of the cutting tool with the blank part are computed. Then, the machined geometry of the part is obtained by displaying the resultant slices on the computer screen. This method also allows performing collision detection between modeled machine components by calculating the overcut regions in the same manner. The efficiency of this method depends on the slice numbers and the intervals between these slices.

The swept volume terminology is also used in combination with constructive solid geometry approach [34-36]. In this terminology, the cutting tool motion is defined and the volume occupied by the cutting tool during machining operation is calculated. This volume is named as the swept volume and after calculating this volume, it is subtracted from the created workpiece solid geometries.

1.4 Scope of the Thesis

As explained in previous sections, although five-axis CNC milling machine tools brings great flexibility in the production of complex parts, it is not that easy to properly operate them. Proper use of these machine tools first needs advance knowledge on tool path creation methods using CAM programs, needs a presence of a well-prepared postprocessor that meets the requirements of the particular CNC machine tool, and it also needs a complete machine simulation and verification software with collision and gouge detection capabilities that eliminates any collision or gouging possibility of machine components.

Similar to the requirements of the CNC users in industry, in order to operate properly the five axis high speed “Mazak Variaxis 630-5X” machine tool located in METU BILTIR Center, a postprocessor, simulation and verification software is needed. To fulfill the requirements of this particular machine tool, it is initially prepared a postprocessor by using G-Post module of ProEngineer; however as explained previously in Section 1.3.3, this postprocessor is insufficient and causes gouges and collisions during five-axis milling operations. The problems encountered will be described in detail in Chapter 3. In addition, verifying the generated tool is also a problem like many five-axis machine tool owners experienced. The tool path verification capabilities of CAD/CAM packages are very limited as described previously in Section 1.3.4. Hence, additional precautions should be taken to eliminate any collisions while operating the “Mazak Variaxis 630-5X” machine tool located in METU-BILTIR Center.

As stated earlier in Section 1.3.4., commercially available postprocessor, simulation and verification packages are very expensive. Therefore, the goal of this thesis study is to develop a software that is capable of postprocessing, simulating and verifying cutting tool paths for the high speed “Mazak Variaxis 630-5X” five-axis CNC machine tool located in METU-BILTIR Center. The features of the software will be described in detail in next chapters. In Chapter 2, the general structure of the developed software will be described. The properties the postprocessing part of the software and developed algorithms for

postprocessing process are explained in Chapter 3. The techniques used for generating simulation process in the developed software are represented in Chapter 4. In Chapter 5, the tool path verification capabilities of the software and the developed algorithms for collision detection are described. In Chapter 6, the sample applications performed on the “Mazak Variaxis 630-5X” machine tool are represented.

CHAPTER 2

GENERAL STRUCTURE OF THE DEVELOPED SOFTWARE

2.1 Introduction

In this chapter, the computer environment used for the software development, the necessary requirements and the capabilities of the software are described. It is also given the general structure of the developed software and the working principle of the software is described by giving all the functional software components.

2.2 Computer Environment Used For Software Development

The software developed in this thesis study is named as “Manus 1.0” and throughout this thesis, this name will be used. Manus 1.0 software has been developed on a PC based Microsoft Windows 2000 operating system, which is also compatible with Windows 95, Windows NT, Windows 98 and Windows XP.

There exist various alternative platforms for software development [37]. The software development platform, that will be used, has to involve both the compiling environment and graphical library tools, which are used for simulation processes. Borland C++ Builder 5.0 programming language [38] has been preferred to be used for developing the Manus 1.0 software, since it uses full advantage of the high-speed ANSI/ISO C++ technology and provides a highly productive visual development environment.

The graphical library tools are special software libraries developed by various organizations to simplify programming when drawing and modeling

processes are performed. In order to develop a software capable of displaying graphical objects and performing milling simulations, a graphical library tool is needed. Graphical library tools are generally called as Graphics API (Application Programming Interface). Graphics API's are embedded in programming environment and software developer uses the functions in the Graphics API to draw graphical objects on the screen. The Graphics API preferred to be used in this thesis study is *Silicon Graphics OpenGL* graphical library [39], since it is the most commonly used Graphics API in simulation programming processes. This Graphics API does not only work on Windows operating system, but also supports other widely used operating systems. It can be embedded in Borland C++ Builder, which has been used as programming language in this thesis study. The most important feature of OpenGL is its high speed and because of this, it is widely used in software development industry [39].

2.3 Software Requirements and Capabilities

The minimum hardware requirements for the developed software are as follows:

1. Windows operating system
2. Intel Zeon processor
3. 256 MB RAM (512 MB RAM is recommended)
4. Video card with 3D support (OpenGL compatible)
5. 150 MB Hard disk space

The capabilities of the developed software can be summarized as follows:

1. The software can read and process the CL-Data generated by CAM module of ProEngineer.
2. The software can generate NC-Codes (G-Codes) for three and five-axis milling processes.

3. The user can save the generated NC-Code into a desired file in computer memory.
4. The software allows the user to select and use the workpiece CAD models that are prepared in any CAD programs. Any workpiece models that are converted to STL ASCII format can be perceived by the Manus 1.0 software.
5. The necessary machine tool components like table, table support, spindle head, machine inner walls and operating door are predefined in the Manus 1.0 simulation environment.
6. The tooling information can be defined in Manus 1.0.
7. The Manus 1.0 software has a library for the tool holders that are currently used in METU-BILTIR Center. The user selects one of these predefined tool holders from this software library.
8. The fixtures that are used in the machining setup can be defined and placed in Manus 1.0 simulation environment.
9. The Manus 1.0 software has a library for the clamping elements that are currently used in METU-BILTIR Center. The user can select the required clamping elements from this software library and place them in desired locations using special software tools.
10. Simple geometric elements like cylindrical or prismatic objects can be modeled in Manus 1.0 without using any CAD software.
11. The simulation play controls are available in Manus 1.0 software.
12. Dynamic zooming, rotating and translating of the view vector can be performed during the machining simulation.
13. The machine setup and structural component view settings can be adjusted as opaque, translucent or invisible in Manus 1.0 during simulation.

14. The tool path is automatically displayed during the simulation. The cutting motions are displayed in red color; the rapid motions are displayed in yellow color.
15. The collision checking algorithm is developed for the cutting tools. The Manus 1.0 software can detect the collisions of the cutting tool with any machine tool or setup components.
16. The collisions that are detected are reported on a report window.
17. The collisions are also displayed during simulation by changing the color of the collided machine tool or setup component.
18. In Manus 1.0 software, it is also developed an algorithm for the not exactly positioned fixtures and clamping elements. The software detects the risky approaches to that misplaced fixtures and clamping elements and warns the user.

All these features of Manus 1.0 software will be described in Chapters 2, 3, 4 and 5 in detail.

2.4 Overview of the Software Structure

The schematic diagram of the software structure is shown in Figure 2.1. There exists fundamentally three main software parts in Manus 1.0. These are *Postprocessing Part*, *Simulation Part* and *Verification Part*. In all these parts, specific processes are performed. Each part has its own software components where these processes are activated. These components are functional, specific purpose components and each component carries out a special function. These software components will be described in the following sections.

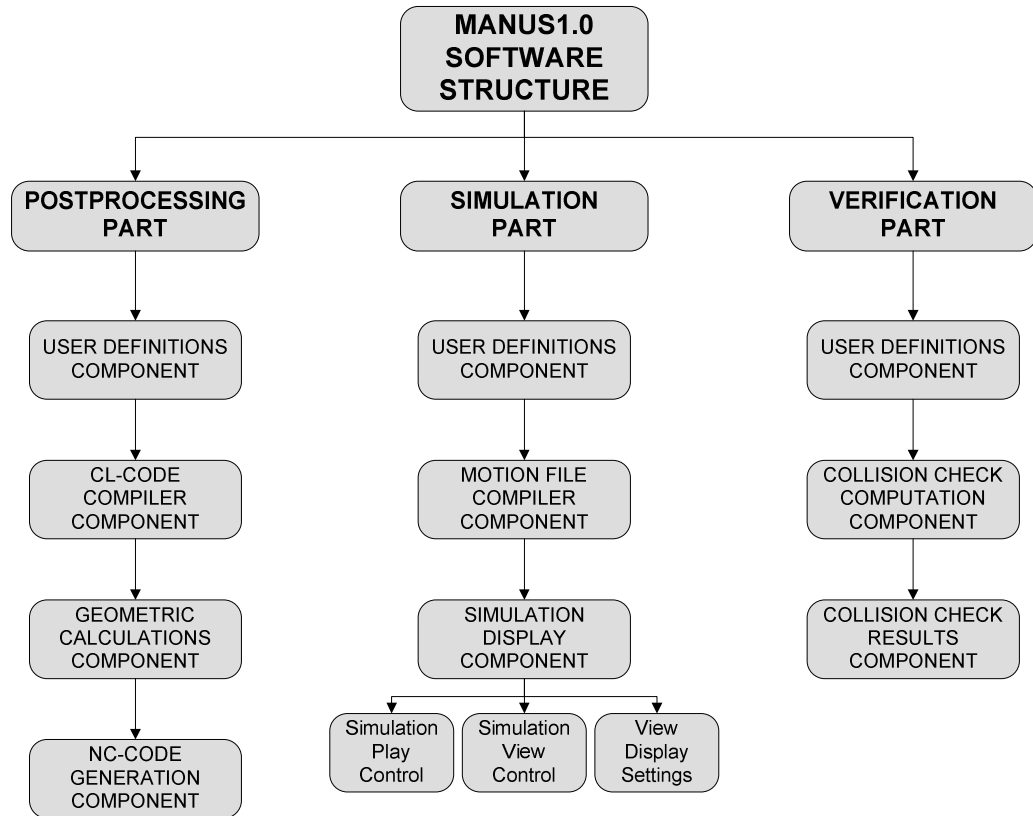


Figure 2.1 Schematic diagram of the software structure

2.5 Postprocessing Part of the Software

One of the main goals of the software is to postprocess the CL-Data and to generate the required NC-Code (G-Code) for machining. Postprocessing is performed by the postprocessing components shown in Figure 2.1. The brief explanations of these software components and some important features of the *Postprocessing Part* of the software are described in the following subsections. The detailed description of the *Postprocessing Part* of the software is given in Chapter 3.

2.5.1 User Definitions Software Component

For postprocessing to be performed, CL-Data files which will be postprocessed and some related machining parameters have to be defined by the user. The CL-Data file is the output of CAM software. The particular CL-Data has to be selected by the user by activating “*Open CL-Data*” menu item in Manus 1.0. The related machining parameters are “Workpiece Coordinate System (WCS)” parameters and are defined in “*WCS Selection and Settings*” menu item.

Workpiece Coordinate System (WCS) Selection and Settings: The machining reference coordinate system for the workpiece should be determined while generating tool path in CAM software. All the CL-Data and the motion command lines are generated according to this machining reference coordinate system. A sample machining reference coordinate system which is adjusted during tool path generation in CAM module of ProEngineer is shown in Figure 2.2. WCS is the coordinate system which shows the location of this machining reference coordinate system on the machine tool. Therefore, it is associated with every workpiece mounted on the machine tool. Once the location of the WCS is determined in the machine coordinate system which is specific for each CNC machine tool, it does not change due to any tilting or rotating motions and stays as fixed during machining operation. The location of the machine coordinate system for “Mazak Variaxis 630 5X” is given in technical specifications of the machine tool in Appendix A.

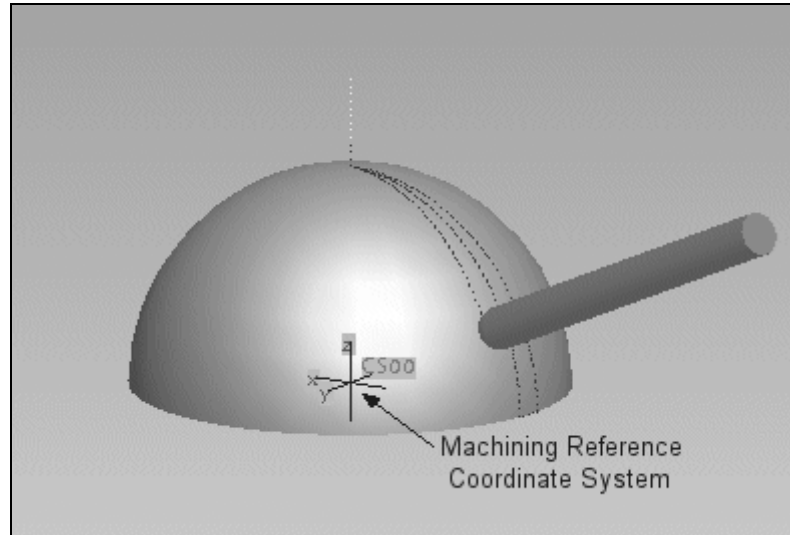


Figure 2.2 Machining reference coordinate system

G-Code commands G54 through G59 are used to designate the location of WCS [40]. There are six of these G-Code commands. WCS locations are used during machining operation just by calling the designated G-Code command in G-Code file. In Manus 1.0, the selected G-Code command on machine tool and the measured location of the WCS is entered to the program by activating the “WCS *Selection and Settings*” menu item.

2.5.2 CL-Data Compiler Software Component

CL-Data Compiler Software Component is used to read the cutting information stored in CL-Data file which includes the position and orientation of the cutting tool for every single step of cutting motion and some machining related parameters. The information gathered is sent to the *Geometric Calculations Software Component*.

2.5.3 Geometric Calculations Software Component

This software component forms the skeleton of *Postprocessing Part* of the software. Geometric calculations, which are performed to find the exact locations

of workpiece and cutting tool during cutting, are executed in this software component. The postprocessing algorithm together with geometric calculations is given in Chapter 3.

2.5.4 NC-Code (G-Code) Generation Software Component

After all geometric calculations are performed; the NC-Code is generated and displayed in the software screen. In Manus 1.0 software, the user can also save this generated NC-Code in the computer memory to make it transferable to the NC unit of the machine tool. The “Mazak Variaxis 630-5X” machine tool can only perceive the NC-Codes that have “EIA” extensions. In Manus 1.0, the user can create a file with the “EIA” extension in the computer memory and can save the generated NC-Code in this file. “*Save G-Code*” or “*Save As G-Code*” menu items are employed for this purpose.

2.6 Simulation Part of the Software

The other objective of the developed software is to simulate the milling operations that are performed in “Mazak Variaxis 630-5X” machine tool. The simulation of the milling operations are achieved by the software components shown in Figure 2.1. The brief explanations of these software components and some important features of the *Simulation Part* of the software are described in the following subsections. The detailed description of the *Simulation Part* of the software is given in Chapter 4.

2.6.1 User Definitions Software Component

For milling simulation to be activated, *WCS center location*, *workpiece selection*, *tooling settings* and *fixture settings* parameters have to be defined.

WCS Settings: If the WCS location is not defined, the workpiece cannot be located on the table correctly and simulation cannot be performed truly. This

parameter has already been described in the *User Definitions Software Component of Postprocessing Part* of the software.

Workpiece Selection: The second user defined parameter for simulation is the selection of the workpiece file which the milling operations will be performed on. The workpiece model can be prepared beforehand by using any CAD software. The prepared model should be saved in STL format to be used in Manus 1.0 software. In Manus 1.0 software, the user selects the workpiece model by using “*Workpiece Select and Preview*” menu item and this selected model is used in the simulation.

Tooling Settings: The cutting tool and tool holder types, the length and the diameter of the cutting tool have to be defined in the “Tooling Settings” menu item of Manus 1.0 software.

Fixture Settings: The last parameters that have to be defined for simulation are the fixture settings parameters, which are defined in “*Fixture Settings*” menu item. The fixtures used in the setup, which are modeled by using any CAD software, can be used in Manus 1.0. These fixture models must be converted to STL format in CAD software in order to make them perceivable by Manus 1.0. The selection and positioning of these fixture models are performed in “*Fixture Settings*” menu item. In Manus 1.0, it is also possible to model simple fixture elements like cylindrical or prismatic objects without using commercial CAD programs. In addition, not only fixtures used in the setup can be defined in Manus 1.0, but it is also possible to define the clamping elements used in the setup. In Manus 1.0, the user can select the proper clamping elements from the library of the software. Then, these clamping elements are positioned in proper places. The clamping elements defined in the software library include the ones currently available in METU-BILTIR Center. Machining setup preparation capability of Manus 1.0 will be described in detail in Chapter 4.

2.6.2 Motion File Compiler Software Component

Motion File Compiler reads the motion file generated in *Postprocessing Part*. While creating the NC-Code in *NC-Code Generation Software Component*, the Manus 1.0 also automatically generates the necessary motion code and writes this information in a file. *Motion File Compiler* processes this motion code file and simulation is performed according to the information gathered by *Motion File Compiler*.

2.6.3 Simulation Display Software Component

The simulation of milling operation is performed on a separate window in Manus 1.0 software. The software screen is automatically opened by starting the simulation process.

For realistic simulation and for perceptible viewing directions, the main components of “Mazak Variaxis 630-5X” like table, table support, spindle head, machine inner walls and operating door are modeled as shown in Figure 2.3. The walls and the operation door of the machine tool may become invisible according to the view vector orientation in order to ensure that the machining zone always becomes visible.

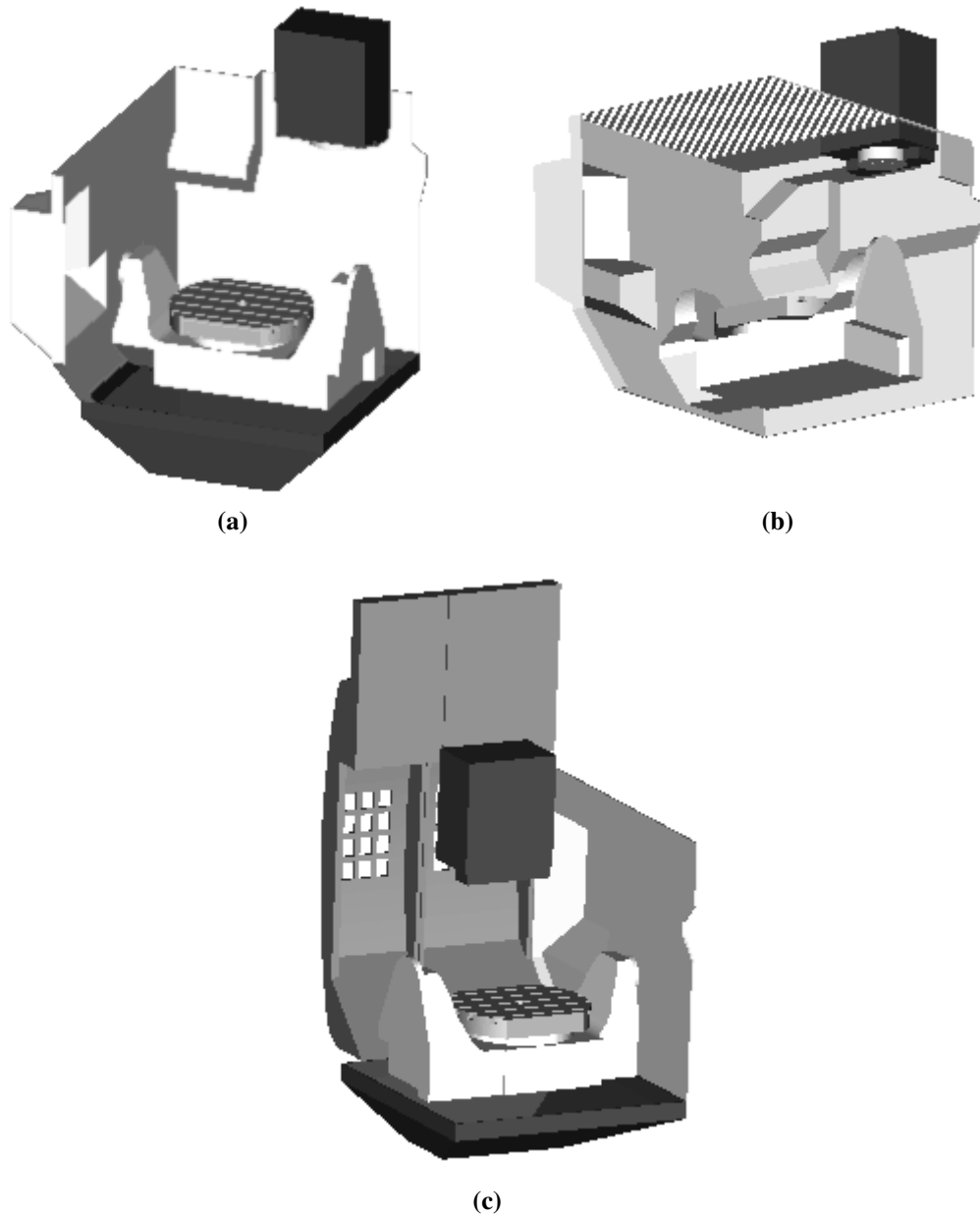


Figure 2.3 Sample simulation views from Manus 1.0 (a) Front-Upper-Right view (b) Front-Lower-Right view (c) Back-Upper-Left view

Simulation Display Software Component involves three operational subcomponents. These are *Simulation Play Control*, *Simulation View Control* and *View Display Settings*.

2.6.3.1 Simulation Play Control

The simulation play control is performed by a separate control window as shown in Figure 2.4.

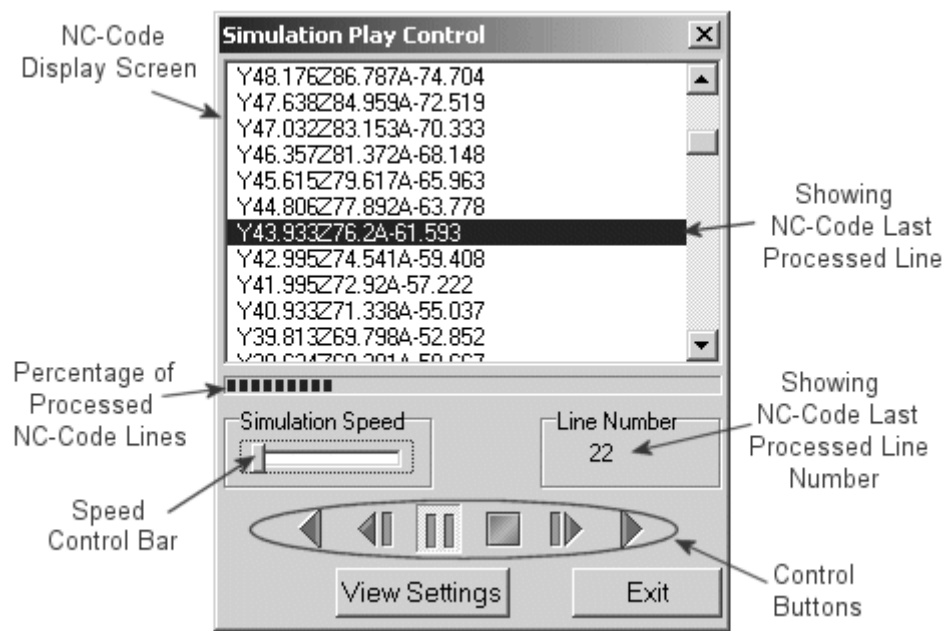


Figure 2.4 Simulation Play Control window in Manus 1.0

The NC-Code, which the simulation will be performed for, is displayed on this control window. During simulation, NC-Code is executed line by line. The currently proceed line is shown as blue colored on the NC-Code display screen. The line number of the currently processed NC-Code line is also displayed on the control window as shown in the figure. During simulation, the user can check the percentage of the processed NC-Code lines via the progress bar indicated in the figure. The simulation speed can be adjusted by using the speed control bar. There are 10 step intervals of speed control bar and the user can increase the simulation speed.

There are also six control buttons on the simulation play control window as shown in Figure 2.5.

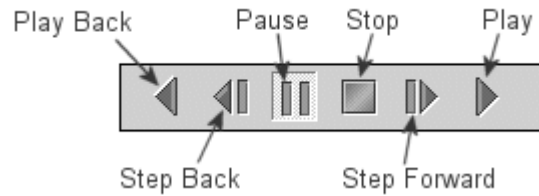


Figure 2.5 Simulation Control Buttons

Play Button: It is the simulation start button. By clicking on this button, the simulation is started with a simulation speed defined by the speed control bar.

Pause Button: The simulation can be paused by clicking on this button.

Stop Button: Clicking on the stop button resets the simulation and simulation goes back to start position, which is the first line of NC-Code.

Play Back Button: It is the simulation reverse playing button. By clicking on this button, the simulation begins to play reverse with a simulation speed defined by the speed control bar.

Step Forward Button: CNC machine tools generally can capable of running the NC-Code in “line by line mode” or in other words “single block mode”. To simulate the machining operation more realistic, “*single block mode*” playing property is also added to Manus 1.0. By activating this button, the simulator is automatically paused after the particular line of the NC-Code is executed. To run the simulation in a “single block mode”, this button has to be pressed for every single line of NC-Code to proceed.

Step Back Button: This button does the same job with the step forward button, but in reverse order.

2.6.3.2 Simulation View Control

During simulation, usually it is needed to look the machining zone from different directions. Therefore, the simulation software must have the ability to change the view direction in any orientation desired. When this property is not supported by the simulation software, in some cases it becomes impossible to

visually track the tool path. In Manus 1.0 software, it is possible to dynamically change the view vector during simulation. The table center of the machine tool is taken as the rotation center for the view vector rotations. It is also possible to dynamically translate this view vector in the screen plane.

The other requirement expected in simulation software is the zooming capability and this feature is available in Manus 1.0 software. The absence of this property limits the user to focus on any desired location and this prevents the user to visually control small gouges or collisions that can occur during simulation.

2.6.3.3 View Display Settings

All machine components are initially displayed as visible and opaque in Manus 1.0. In order to make the simulation more effective, the view display settings capability is added to the Manus 1.0 software. The view display properties of machine tool and setup components can be adjusted by activating the “View Settings” button in “Simulation Play Control” window. The “View Display Settings” window is opened by activating this button as shown in Figure 2.6.

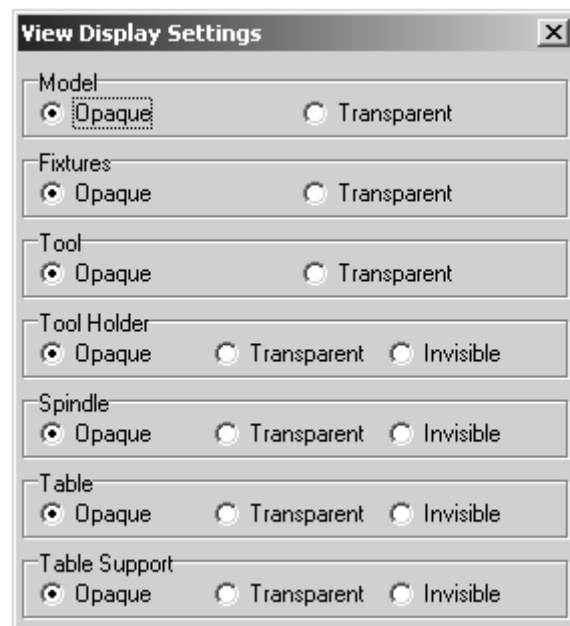


Figure 2.6 “View Display Settings” window of Manus 1.0 software

Many times during simulation, machining zone becomes hidden depending on the view direction. The table support, table, spindle, fixtures or clamping elements may behave as obstacles and cause the machining zone to be invisible. Figure 2.7(a) shows a sample simulation view, where the machining zone can not be seen because of the view vector orientation, since the table support behaves as an obstacle in front of the machining zone. In order to see the machining zone in this view vector orientation, the table support must be displayed as translucent or have to be invisible. Most of the commercial verification programs do not have this capability. In Manus 1.0 software, this can be achieved via adjusting the view display settings. To make the machining zone visible regardless of view vector orientation, view display settings of the machine tool components can be adjusted as translucent or invisible. In Figure 2.7(b), same sample simulation view shown in Figure 2.7(a) is given, but this time view display setting of the table support is adjusted as translucent. Here in this view, the cutting zone can be seen although the table support again is in front of the cutting zone.

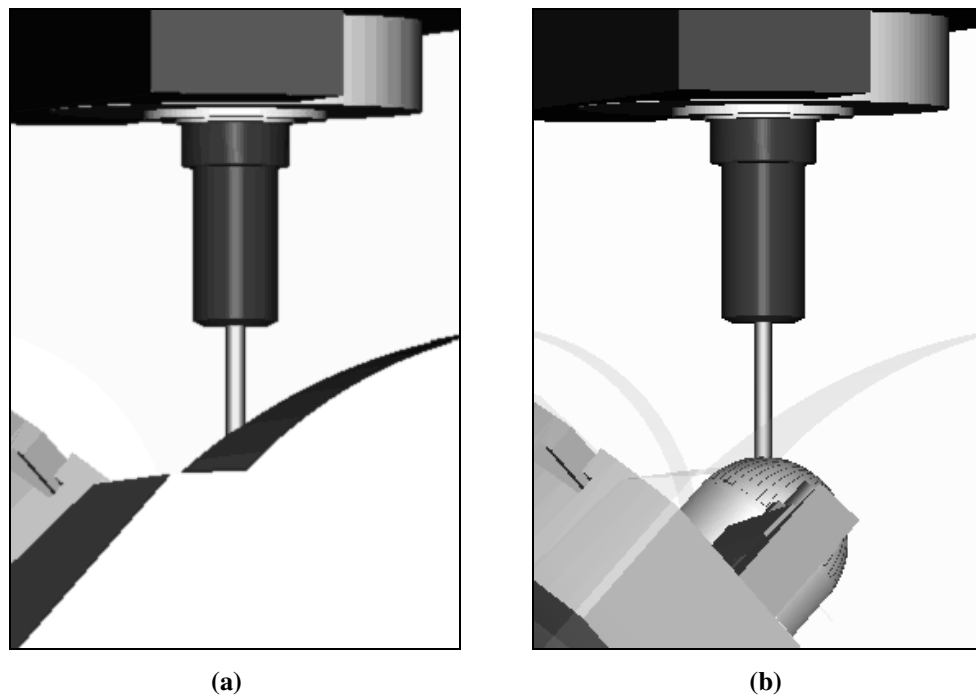


Figure 2.7 Sample simulation views from Manus 1.0 (a) table support is opaque (b) table support is translucent

Similar view blocking problems may occur with the other machine setup components like table, spindle head, fixtures or clamping elements and even with workpiece model itself. Same view display setting adjustments can be performed on these components.

2.7 Verification Part of the Software

The other objective of the developed software is to verify the tool path generated for “Mazak Variaxis 630-5X” machine tool. The gouges and collisions of the cutting tool with machine tool and setup components are found in this part of the software. The verification of the tool path is achieved by the software components shown in Figure 2.1. The brief explanations of these software components and some important features of the *Verification Part* of the software are described in the following subsections. The detailed description of the *Verification Part* of the software is given in Chapter 5.

2.7.1 User Definitions Software Component

For the tool path verification, the collision checking parameters have to be defined firstly. These parameters are *collision check approach tolerance* parameters defined for table, fixtures and clamping elements.

Collision Check Approach Tolerance: Manus 1.0 software can capable of detecting gouges and collisions occurred between cutting tool and machine tool components like table, fixtures and clamping elements. While performing tool path verification on any verification software; fixtures and clamping elements have to be defined in correct geometry and position as much as possible. Many times exactly adjusting the positions of these components in the simulation environment is hard to achieve and time consuming. However, if these components are not positioned exactly in the simulation software, some possible collisions can not be determined. Manus 1.0 software provides flexibility to the

user to define warning regions. The user may enter the collision check approach tolerance (P) value which is the minimum allowable distance between the cutting tool and the related machine tool or setup components. For instance, if the user wants to learn whether the cutting tool approaches closer than 5mm to fixtures or clamping elements, then this approach tolerance value should be entered to the software and the software will warn the possible risky approaches that are calculated. In Manus 1.0, the approach tolerance values for the table, fixtures and clamping elements can be separately input by activating the “*Collision Check Approach Tolerance*” menu item. To the best knowledge of the author, this is a special feature that is available only in Manus 1.0 software.

2.7.2 Collision Check Computation Software Component

This software component forms the skeleton of *Verification Part*. All collision check calculations are performed in this software component. The collision check algorithm together with mathematical background is given in Chapter 5.

2.7.3 Collision Check Results Software Component

After collision check calculations are performed, the results are displayed as a report and the user can check the results in a report window. The results that are displayed on this window are as follows;

- collision detected NC-Code line numbers,
- collision type for every collisions detected,
- collide components for every collisions detected

In addition to this report, collisions are also displayed during simulation. The collision displaying strategy of Manus 1.0 is to change the color of component which the cutting tool collides to. This collision displaying strategy makes Manus 1.0 simulation capability more effective for visually controlling

tool paths during simulation. Just at the time of collision during simulation, the component, that is collided, takes the color of red. In Figure 2.8, a sample simulation view is shown. In this figure, the cutting tool collides to a nut which is a clamping element, and the nut takes the color of red just at the time of collision. The cutting tool view display property is adjusted as translucent in order to make the collision zone visible. In addition, if the cutting tool enters the zone defined by the “collision check approach tolerance” value, the related machine setup components take the color of pink as shown in Figure 2.8.

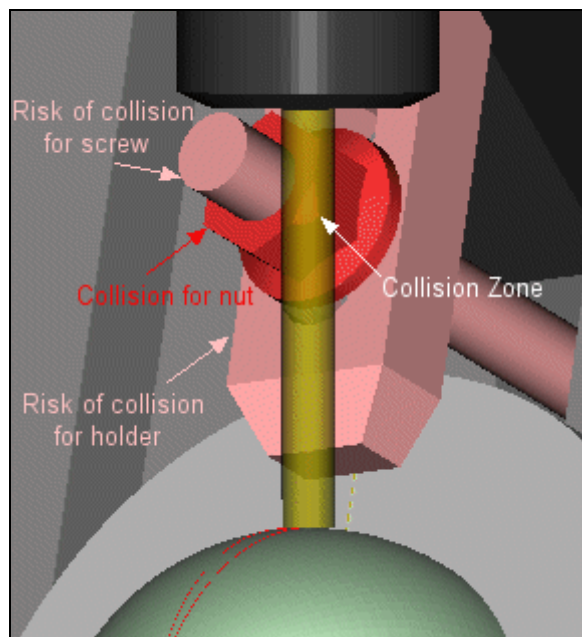


Figure 2.8 Sample simulation view where cutting tool collides to a nut

2.8 Closure

All the software features and the functions of the software components were described in this chapter. The computer environment used for the software development and the necessary requirements were also given in this chapter. In addition, the capabilities of the software were described. It was also described the general structure of the developed software and the working principle of the software was described by giving all the functional software components.

CHAPTER 3

POSTPROCESSOR PART OF THE SOFTWARE

3.1 Introduction

The postprocessing process is performed in a sequence by software components described in Chapter 2 and the steps of the postprocessing process are summarized in Figure 3.1.

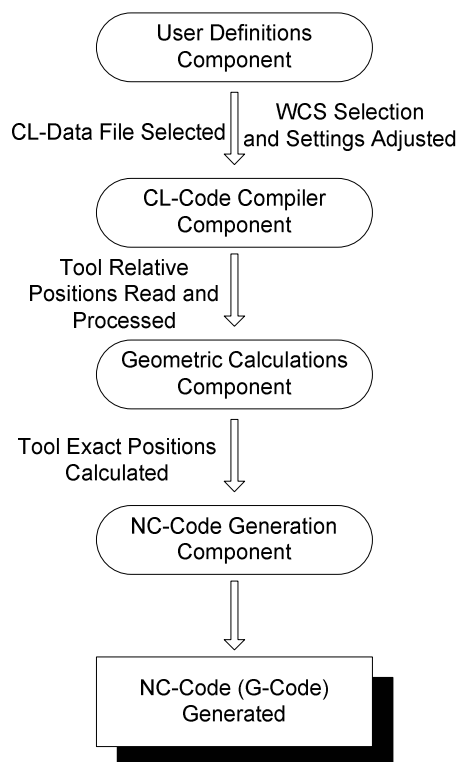


Figure 3.1 Steps of postprocessing process

Geometric calculations are performed in the *Geometric Calculations Software Component* of the *Postprocessing Part* of the software and these calculations form the basics of the postprocessing process. The cutting information in the CL-Data file includes the instantaneous locations of the cutting tool during cutting process with respect to a stationary workpiece. In order to obtain the same relative motion defined in the CL-Data file, the table (i.e. workpiece) and the spindle head (i.e. cutting tool) have to perform some synchronize combinations of motions in five-axis “Mazak Variaxis 630-5X” CNC machine tool. The motions that have to be performed both by the table and the spindle head must be calculated according to the information in the CL-Data file. The algorithm used in geometric calculations will be described in the following sections.

In this particular CNC machine tool, the cutting tool (in the spindle) has three degree of freedom and can move in X, Y and Z axes in the machine coordinate system. Spindle head does not rotate or tilt, the cutting tool axis vector always orients parallel to the machine Z-Axis. The other two axes of the machine tool are A and C axes. The A-Axis motion is the tilting motion performed by the table as shown in Figure 3.2 in different tilting orientations.

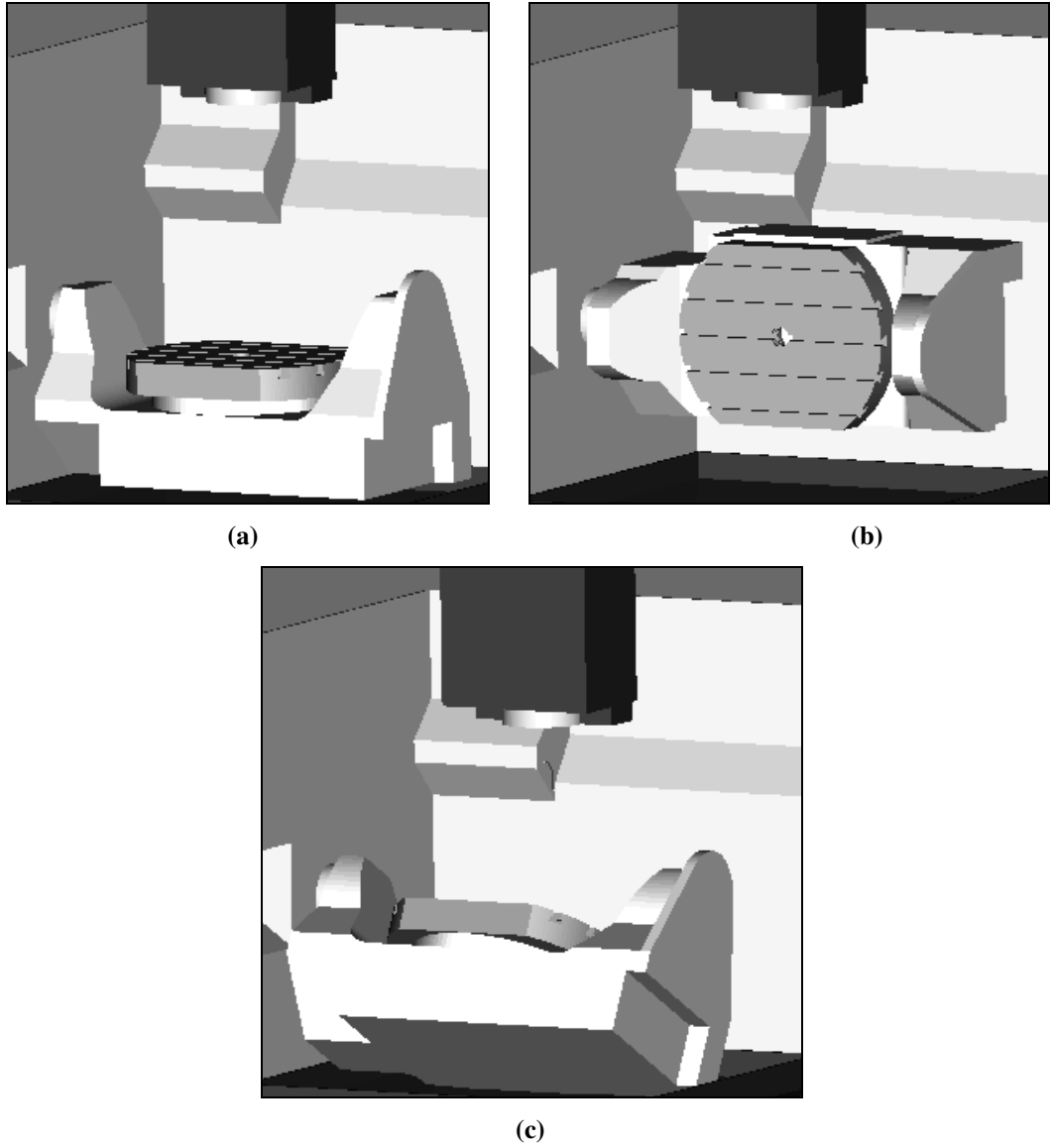


Figure 3.2 Sample simulation views from Manus 1.0 where (a) $A=0^\circ$ (no table tilting motion) (b) $A=-90^\circ$ and (c) $A=30^\circ$

The C-Axis motion is the rotating motion of the table as shown in the three sample views given in Figure 3.3.

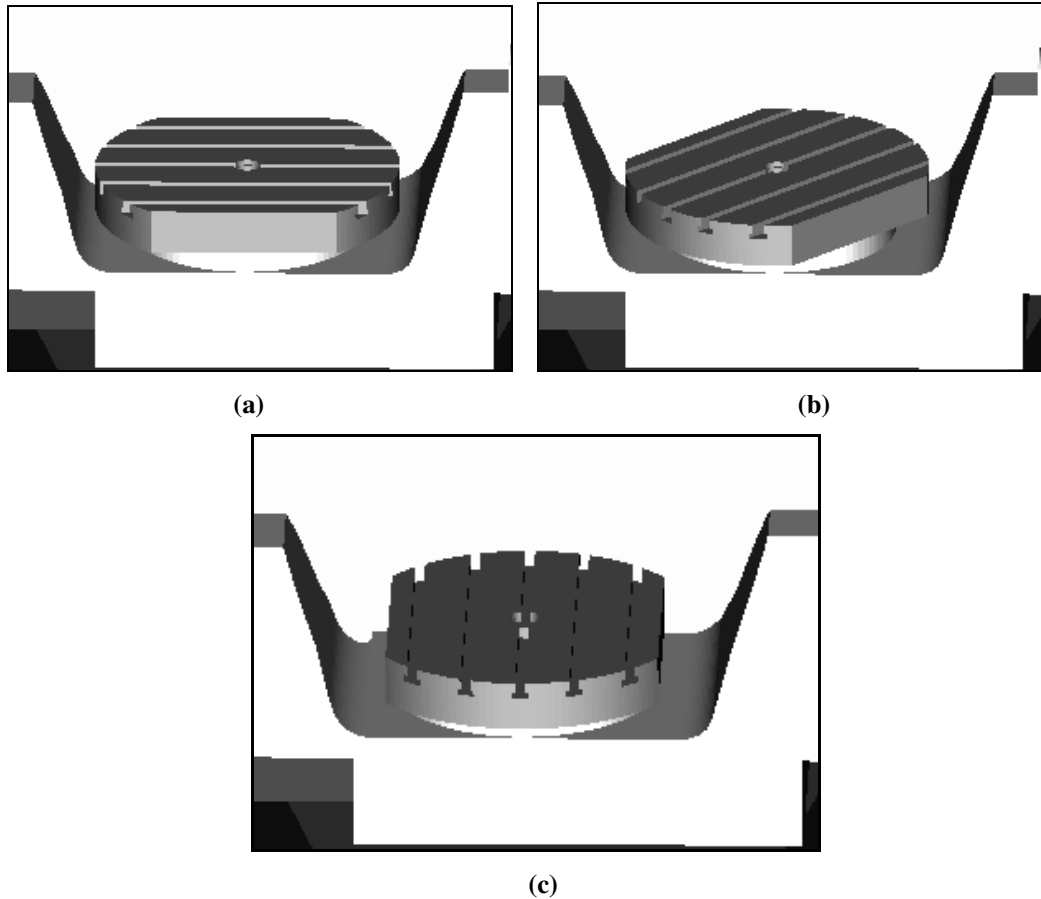


Figure 3.3 Sample simulation views from Manus 1.0 where (a) $C=0^\circ$ (no table rotation) (b) $C=-45^\circ$ and (c) $C=-90^\circ$

In order to find the position of the cutting tool in X,Y and Z axes of the machine coordinate system and the position of the workpiece in A and C axes, their relative locations with respect to each other have to be determined first. This is done in *CL-Data Compiler Component* of the software. The CL-Data includes the machining information and the motion commands. The motion commands in the CL-Data start with the “GOTO” command and for multi-axis milling operations, it will be in the form of;

GOTO / U, V, W, I, J, K

The U, V and W parameters are the relative positions of cutting tool with respect to the stationary workpiece in X,Y and Z axes respectively. The I, J and K values are the tool axis vector components in X, Y and Z axes respectively. A sample CL-Data motion line is given as follows;

```
GOTO / -15.7213624650, 0.000, 47.4640786516, $
-0.3144272890, 0.0000000000, 0.9492815599
```

The U, V, W and I, J, K parameters take the following values in this sample; U=-15.7213624650mm, V=0, W=47.4640786516mm, I=-0.3144272890, J=0, K=0.9492815599. The continuation sign (“\$”) in this representation means that the motion line is not ended and the below line belongs to this motion line command. The position of the cutting tool for this sample CL-Data motion line is shown in Figure 3.4.

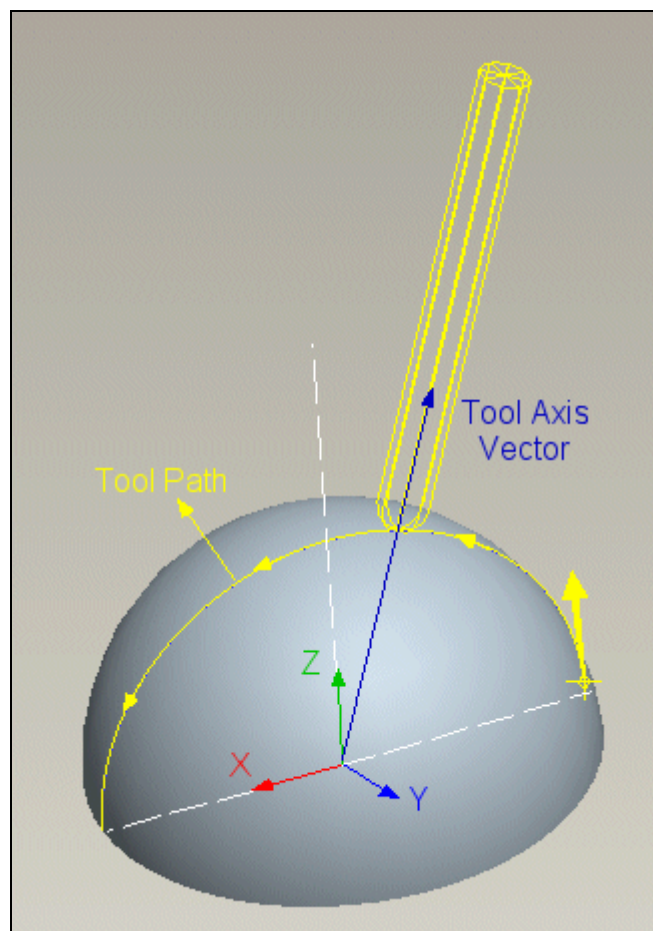


Figure 3.4 The position of the cutting tool specified in the sample motion command line

In “Mazak Variaxis 630-5X”, the spindle head makes no tilting and rotating motion and the cutting tool always orients parallel to the machine Z-Axis. This means a situation as shown in Figure 3.4 will never occur in “Mazak

Variaxis 630-5X". Instead, the workpiece will make rotating and tilting motions while the cutting tool orients parallel to machine Z-Axis as shown in Figure 3.5. In this figure, same instantaneous position of the cutting tool given in Figure 3.4 is shown in the real machining environment of "Mazak Variaxis 630-5X" machine tool.

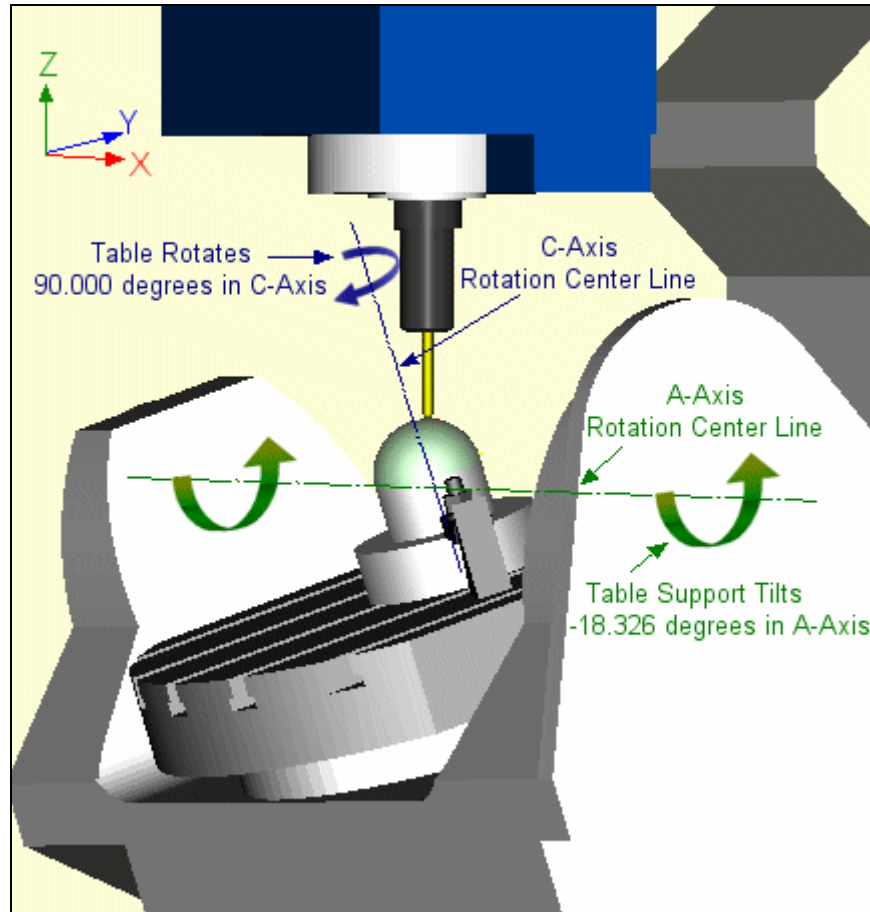


Figure 3.5 Simulation view for "Mazak Variaxis 630 5X" in Manus 1.0

In Figure 3.5, while the cutting tool lies along the machine Z-Axis, the table tilts -18.326° in A-Axis and rotates $+90^\circ$ in C-Axis and the cutting tool is positioned at: $X=0$, $Y=-15.736$ mm, $Z=47.466$ mm according to the specified WCS. In this particular case, the WCS is set as: $WCX=-315$ mm, $WCY=-315$ mm, $WCZ=-640$ mm in machine coordinate system.

The generated machining code (G-Code) depends on the WCS location determined. Changing the position of the raw material means the WCS location is changed which causes a modification in the machining code. For instance if the raw material is mounted on a different place than the previous sample, whose WCS location is given as; WCX=-300 mm, WCY=-300 mm, WCZ=-600 mm, then the cutting tool has to take a position of X=0, Y=-57.553 mm, Z=40.72 mm for the same CL-Data motion line.

3.2 A-Axis Calculations

The A-Axis motion is a tilting motion performed by the table as shown in Figure 3.2. The formula derived to find the table A-Axis location can be given as;

$$\mathbf{A} = \pm \arccos (\mathbf{K}) \quad (3.1)$$

where K term is defined in the CL-Data file.

“Mazak Variaxis 630-5X” machine tool can make a tilting motion in A-Axis in the interval of -120° and +30° as given in Appendix A. This means that there is a second alternative for every A-Axis location in the interval of ±30°. This is shown on Eq 3.1 with a ± sign in front. Every point represented by plus A-Axis value can also be represented by minus A-Axis value or vice versa by performing proper adjustments in C-Axis. A-Axis value normally gets minus sign in the milling operations in “Mazak Variaxis 630-5X”.

The G-Post module of ProEngineer [20] and the CNK5XPP postprocessor [41] which is developed by CNK software company for “Mazak Variaxis 630 5X” takes A-Axis movement range between 0° and -90° disregarding the plus moves and also disregarding the moves over -90° up to -120°. These postprocessors use the above approach and any plus movements are represented by their minus movement case alternatives.

At first glance, it seems that positive movements on A-Axis are meaningless and any plus moves can be represented with their minus case alternatives. Although, with this approach, many milling operations can be

performed quite well, it is not possible to obtain expected results in some cases. For instance, if the cutting tool has to make a continuous cutting motion starting from -30° and ending at $+20^\circ$ in A-Axis, the G-Post and CNK5XPP postprocessor have to interrupt this continuous motion just at the position A-Axis plus movements start. When the plus moves start, in order to lift the cutting tool from the cutting zone, CNK5XPP postprocessor guides the cutting tool to make a retract motion (Retracting motion is not available in G-Post postprocessor). Then the CNK5XPP postprocessor positions the cutting tool and the workpiece in the minus case alternative position and the remaining cutting process starts at that new location. This is a typical “phase reverse process” that will be described in detail in Section 3.2.2. Although this cutting strategy of CNKPP5X postprocessor can be treated as acceptable, actually interrupting the cutting process especially in finishing operations is undesirable in milling processes. These unnecessary interruptions on cutting processes may increase the machining times excessively. In addition, this interruption can affect the cut surface quality unfavorably and even may cause some small gouges just when the tool contacts the cut surface after interruption. The phase reverse process is not available in G-Post postprocessor; this will be described in Section 3.2.2.

Manus 1.0 software uses the whole range of the machine tool A-Axis limits. The A-Axis limit of Manus 1.0 software is in the interval of $+30$ degrees and -120 degrees. This widened range increases the possibility of uninterrupted longer continuous cutting motions to occur.

3.2.1 Algorithm for Detecting A-Axis Sign Changes

In the default case, A-Axis tilting motions are all performed through minus signed region (0° to -120°) of A-Axis. Every tilting motion that has to be done is desired to be succeeded in the minus signed region. It is practical to complete all the machining operation in only one region (minus signed region). The plus signed region cannot be selected as the first choice for machining

operations since its range (+30° to 0° degrees) is narrower than the minus signed region.

The plus signed region is only used when there is a continuous motion that will be interrupted if it is forced to stay in the minus signed region. The interruption can be avoided by continuing the cutting motion in the plus signed region. However, it is needed to be clarified in which cases sign change algorithm has to be performed.

In order to apply the sign change algorithm, the particular CL-Data motion line has to indicate a cutting motion. This algorithm is not utilized on rapid motions. The second condition controlled is that one step previous position of the table at A-Axis (A_0) must be as close as possible to 0° (not tilted) position of the table since the sign change algorithm is used to avoid the interruptions on the continuous cutting motions. An interval is selected for possible A_0 values by considering several different tolerance values arranged on CAM software. It is observed that the maximum gap size between A-Axis step intervals do not exceed 3° even when the tolerance value of 0.01 millimeters is selected. This tolerance value is the default value that is set in the CAM software and for multi-axis milling operations, it is treated as coarse and even less than 0.0001 millimeters are normally used. Therefore, A_0 value has to be in the interval of $-3^\circ \leq A_0 \leq 3^\circ$ for A-Axis sign change algorithm has to be employed.

The third condition that has to be satisfied is a combination of several conditions and these conditions form the skeleton of the algorithm and are described in Table 3.1.

Table 3.1 Conditions for A-Axis Sign Changes

Condition-1	$C_0=0$ and $((J_0<0 \text{ and } J>0) \text{ or } (J_0>0 \text{ and } J<0))$
Condition-2	$C_0=m*180$ and $((J_0<0 \text{ and } J>0) \text{ or } (J_0>0 \text{ and } J<0))$
Condition-3	$C_0=m*90$ and $((I_0<0 \text{ and } I>0) \text{ or } (I_0>0 \text{ and } I<0))$
Condition-4	$C_0=m*270$ and $((I_0<0 \text{ and } I>0) \text{ or } (I_0>0 \text{ and } I<0))$
Condition-5	$((I_0<0 \text{ and } I>0) \text{ or } (I_0>0 \text{ and } I<0))$ and $((J_0<0 \text{ and } J>0) \text{ or } (J_0>0 \text{ and } J<0))$

In Table 3.1, C_0 is the one step previous position of the table at C-Axis. I_0 , J_0 and K_0 are one step previous components of tool orientation vector and “m” is the multiplier component and in default case it is set as 1. “m” term will be described in Section 3.3.1.

If any of the conditions given in Table 3.1 is satisfied, then the software perceives that there is a continuous motion and changing the region (passing minus sign region from plus sign region or vice versa) will ensure that the cutting motion will be uninterrupted.

3.2.2 Algorithm for Detecting Limit Exceeding in A-Axis

Most five-axis machines having rotating and tilting tables have unsymmetrical rotation limit about A-axis as in the case of “Mazak Variaxis 630-5X” machine tool. Therefore, when a calculated orientation of the table violates the operation limit of the machine at plus side (over $+30^\circ$ in “Mazak Variaxis 630-5X”), the post-processor has to choose the negative conjugate posture of the “plus A-Axis position”. This is called as “phase reverse process”[5] and described in Figure 3.6.

The process works as follows: First the cutting tool is lifted up to a Z value higher than the previous location as shown in “Step 1” in Figure 3.6. The second step is to position the cutting tool on the alternate location or in other

words arranging the cutting tool X-Axis and Y-Axis location. The next step will be arranging the A-Axis and C-Axis location of the machine tool. The cutting tool makes an approach movement just above a predefined distance (Δd) from the calculated alternative Z location. All these motions mentioned are rapid motions whose feedrate are adjusted from the machine tool panel. Finally the cutting tool makes a plunge motion with a feed defined for plunge moves in CAM software. This plunge movements ends at the calculated Z-Axis location where the cutting motion will restart. All these steps are shown in Figure 3.6.

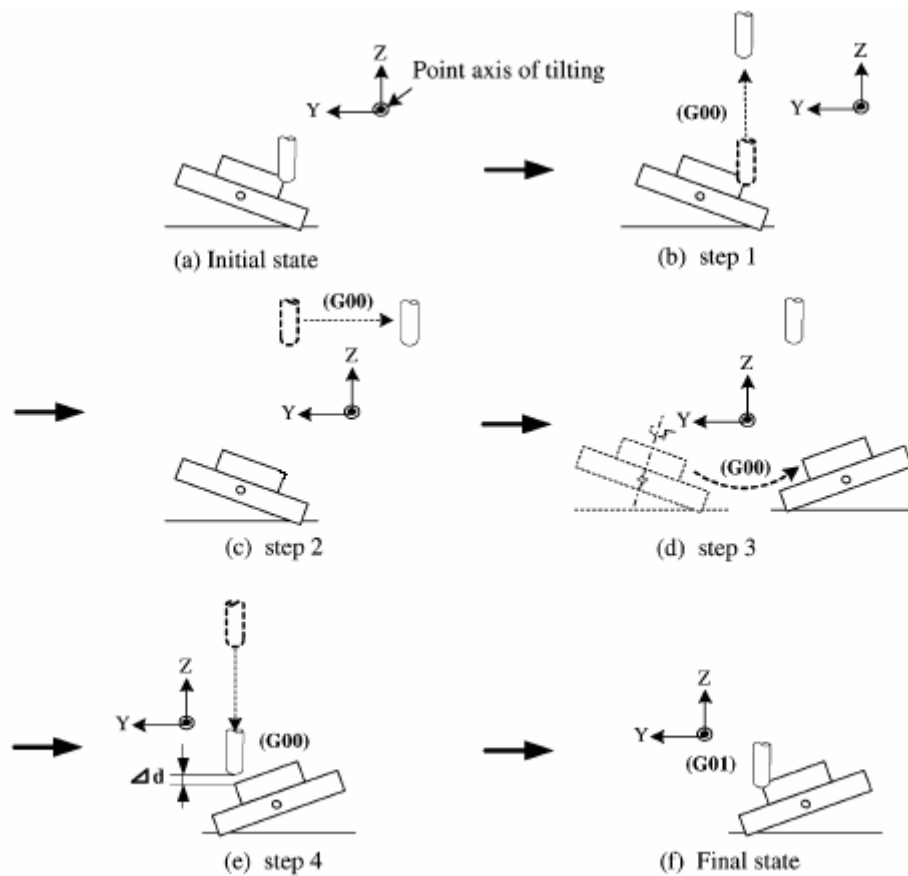


Figure 3.6 Phase reverse algorithm for avoiding tool interference [5]

The A-Axis working zone of “Mazak Variaxis 630-5X” is in the interval of $+30^\circ$ and -120° as mentioned previously. If the calculated position of A-Axis violates this range from the plus side, the “phase reverse process” has to be

performed. If the calculated position of A-Axis exceeds $+30^\circ$; this means that it is not possible to go further in the plus signed region. The cutting process has to be interrupted and the cutting tool and the workpiece have to be positioned in the alternative location. Finding alternative location for locations exceeding $+30^\circ$ is achieved by changing the sign of the A-Axis angle value and by adding or subtracting 180° to the C-Axis angle value. For instance, if the calculated location of A-Axis and C-Axis are; $A = +32^\circ$ and $C = +16^\circ$, the alternative position that will be found in minus signed region will be; $A = -32^\circ$ and $C = 196^\circ$ (or -164°).

Retracting motion is not present in the postprocessor prepared by using G-Post [20] which is a module of ProEngineer as shown in Figure 3.7. The cutting tool is positioned in the minus case alternative location without any retraction of the cutting tool. During this process, while the table rotates and tilts to the new orientation, the cutting tool moves linearly to the new position, which may cause gouge (interference) or collision. This unpredictable movement will result in breakage of the cutting tool and destruction of the workpiece. This is a very dangerous situation and makes G-Post module of ProEngineer unreliable.

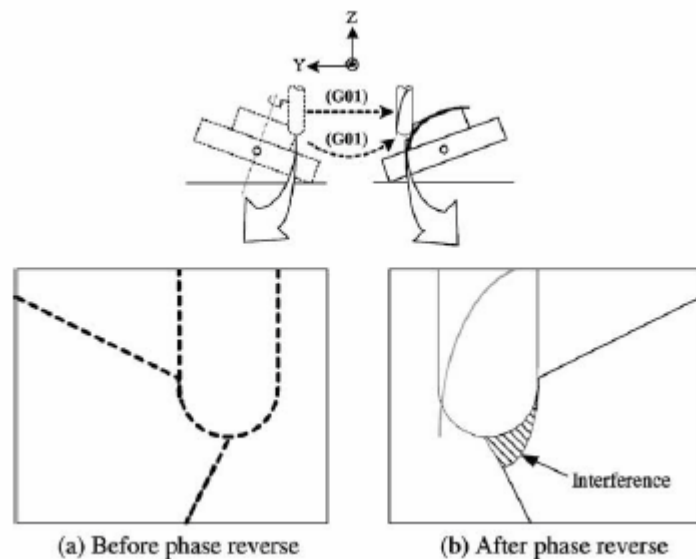


Figure 3.7 Tool interference (gouge) during the phase reverse process [5]

3.3 C-Axis Calculations

The C-Axis motion is a rotating motion performed by the table as shown in Figure 3.3. Ten different formulae are derived to find C-Axis location as given in Eq 3.2- Eq 3.11. One of these following formulae is used to find the C-Axis location depending on the circumstances.

$$C = (m - 1) * 360 + \text{atan2}(I, J) \quad (3.2)$$

$$C = (m - 1) * 360 - 180 + \text{atan2}(I, J) \quad (3.3)$$

$$C = (m - 1) * 360 - 360 + \text{atan2}(I, J) \quad (3.4)$$

$$C = (m - 1) * 360 + 270 \quad (3.5)$$

$$C = (m - 1) * 360 + 90 \quad (3.6)$$

$$C = (m + 1) * 360 + 360 - \text{atan2}(I, J) \quad (3.7)$$

$$C = (m + 1) * 360 + 180 - \text{atan2}(I, J) \quad (3.8)$$

$$C = (m + 1) * 360 - \text{atan2}(I, J) \quad (3.9)$$

$$C = (m + 1) * 360 + 90 \quad (3.10)$$

$$C = (m + 1) * 360 + 270 \quad (3.11)$$

where I and J terms are defined in CL-Data file. “atan2” is the two argument \tan^{-1} function.

The C-Axis rotation limit of “Mazak Variaxis 630” machine tool is in the interval of $\pm 10000^\circ$ and this corresponds to 27 full rotations plus 280° rotations in one direction. Therefore C can get values higher than 360° up to 10000° for some milling operations. The value of the C-Axis location of table depends on multiplier “m” term. In default case, “m” term is set as 1 and can get integer values in the intervals of $-27 \leq m < 0$ and $0 < m \leq 27$.

The presented equations are selectively used in the postprocessing algorithm depending on the sign of the multiplier “m” term and the sign of I and J (tool vector components). Table 3.2 represents the forms of equations employed depending on the value of I and J components.

Table 3.2 Forms of C-Axis calculation equations

	Condition-1	Condition-2	Form of Equation
Form-1	$m > 0$	$(I \geq 0 \text{ and } J > 0)$	Eq (3.2)
Form-2	$m > 0$	$(I \geq 0 \text{ and } J < 0)$	Eq (3.3)
Form-3	$m > 0$	$(I < 0 \text{ and } J < 0)$	Eq (3.3)
Form-4	$m > 0$	$(I < 0 \text{ and } J > 0)$	Eq (3.4)
Form-5	$m > 0$	$(I > 0 \text{ and } J = 0)$	Eq (3.5)
Form-6	$m > 0$	$(I < 0 \text{ and } J = 0)$	Eq (3.6)
Form-7	$m < 0$	$(I \geq 0 \text{ and } J > 0)$	Eq (3.7)
Form-8	$m < 0$	$(I \geq 0 \text{ and } J < 0)$	Eq (3.8)
Form-9	$m < 0$	$(I < 0 \text{ and } J < 0)$	Eq (3.8)
Form-10	$m < 0$	$(I < 0 \text{ and } J > 0)$	Eq (3.9)
Form-11	$m < 0$	$(I > 0 \text{ and } J = 0)$	Eq (3.10)
Form-12	$m < 0$	$(I < 0 \text{ and } J = 0)$	Eq (3.11)

3.3.1 Algorithm for Finding Number of Rotations in C-Axis

The maximum allowable rotation amount in C-Axis is $\pm 10000^\circ$. In order to ensure a continuous machining process, this rotation capability has to be fully activated. The postprocessor developed by using G-Post can only capable of generating tool paths with rotations of $\pm 1000^\circ$ in C-Axis. This causes interruptions on continuous tool paths and excessively affects the machining surfaces and also increases the machining times. In G-Post postprocessor, although it is possible to define C-Axis limits, the software output always become the same ($\pm 1000^\circ$ of rotation capability). This is a limitation of the G-Post

software. It is also observed that the generated code starts to become erroneous just after the $\pm 1000^\circ$ zone is exceeded.

In Manus 1.0 software, an algorithm has been developed to ensure to use the full rotation capability of “Mazak Variaxis 630-5X”. The algorithm mainly depends on finding the rotation amount, in other words the “m” term. The default rotation amount of the table in C-Axis is zero and the default “m” term at zero rotation is 1 which corresponds to a machining zone in the interval of 0° and $+360^\circ$ in C-Axis. The algorithm works as follows: for instance at a particular location during machining, C-Axis value becomes 359° . In the following step, if calculated C-Axis value occurs to be 1 degree, this does not correspond to a 358° counter clockwise rotation. A wrong processing that causes a 358° counter clockwise rotation will generate a totally improper cutting path which will cause workpiece deformation or tool breakage. Instead a 2° clock wise rotation has to be performed for a true uninterrupted cutting path. By performing so, a full rotation in C-Axis becomes completed; therefore the table enters to the second rotation zone where C-Axis values have to be in the interval of $+360^\circ$ and $+720^\circ$. When the table enters the second rotation zone, the multiplier “m” term becomes 2. This means that the C-Axis value in NC-Code will be 361° instead of 1° . This process is called as “C-Axis machining zone changing process” and has to be activated for correct milling operations. Similar processes are activated when going back to the first rotation zone. If the one step previous C-Axis value is 361° and the current C-Axis value is calculated as 719° , this corresponds to a 2° rotation in counter clockwise direction which makes the table reenter the first rotation zone and makes “m” term again 1. This means that the output C-Axis value will be 359° instead of 719° .

The only necessary condition needed to activate “C-Axis machining zone changing process” is;

$$(C-C_0) > 180^\circ \text{ or } (C-C_0) < -180^\circ$$

180° is selected because any rotations above 180° in one motion line cannot be possible in the developed postprocessing part of Manus 1.0 software. The maximum allowable rotation at C-Axis only occurs during “phase reverse

process” and in “phase reverse process”, the table rotates 180° at C-Axis as described in the Section 3.2.2.

3.4 X-Axis Calculations

In this section, the formulae derived and algorithm developed to find the exact location of the cutting tool in X-Axis will be described. The formulae developed to find cutting tool X-Axis location can be given as;

$$X = + U * \cos(\text{atan2}(I, J)) - V * \sin(\text{atan2}(I, J)) - 315 - WCX \quad (3.12)$$

$$X = - U * \cos(\text{atan2}(I, J)) + V * \sin(\text{atan2}(I, J)) - 315 - WCX \quad (3.13)$$

where U, V, I and J are all defined in CL-Data file and WCX is defined by the user in *User Definitions Component* of the software. “atan2” is the two argument \tan^{-1} function.

Table 3.3 represents the forms of equations employed depending on the value of I and J components.

Table 3.3 Forms of X-Axis calculation equations

	Condition	Form of Equation
Form-1	(I>0) or (I=0 and J>=0)	Eq (3.12)
Form-2	(I<0) or (I=0 and J<0)	Eq (3.13)

3.5 Y-Axis Calculations

The formulae derived to find the cutting tool Y-Axis location can be given as;

$$Y = -U * K * \sin(\text{atan2}(I, J)) - V * K * \sin(\text{atan2}(I, J)) + W * I * \sin(\text{atan2}(I, J)) + W * J * \cos(\text{atan2}(I, J)) - B1 - 315 - WCY \quad (3.14)$$

$$Y = +U * K * \sin(\text{atan2}(I, J)) + V * K * \sin(\text{atan2}(I, J)) - W * I * \sin(\text{atan2}(I, J)) - W * J * \cos(\text{atan2}(I, J)) - B1 - 315 - WCY \quad (3.15)$$

where U, V, W, I, J and K are all defined in CL-Data file and WCY is defined by the user in *User Definitions Component* of the software. The B1 term is added to the formulation because of the location of A-Axis rotation center line as can be shown in machine technical specifications in Appendix A. A-Axis rotation center line is located along X-Axis at points + 49.954 mm in Z-Axis and -0.012 mm in Y-Axis with respect to the table center. Term B is formulated as;

$$B1 = \cos\left(-A * \frac{\pi}{360}\right) + \text{atan2}(-0.012, 49.954) * \sqrt{(2 * ((49.954 * 49.954) + (-0.012 * -0.012)) * (1 - \cos(-A * \frac{\pi}{360})))} \quad (3.16)$$

where A term shows the tilting amount of the table (i.e. workpiece) in A-Axis.

Table 3.4 represents which forms of equations are employed depending on the value of I and J components.

Table 3.4 Forms of Y-Axis calculation equations

	Condition	Form of Equation
Form-1	(I>0) or (J=0 and J>=0)	Eq (3.14)
Form-2	(I<0) or (I=0 and J<0)	Eq (3.15)

3.6 Z-Axis Calculations

The formula developed to find cutting tool Z-Axis location can be given as;

$$\mathbf{Z} = \mathbf{U} * \mathbf{I} + \mathbf{V} * \mathbf{J} + \mathbf{W} * \mathbf{K} - \mathbf{B2} - 740 - \mathbf{WCZ} \quad (3.17)$$

where WCZ is defined by the user in *User Definitions Component* of the software. The B2 term is added to the formulation because of the location of A-Axis rotation center line. Term B2 is formulated as;

$$\mathbf{B2} = \sin \left((-\mathbf{A} * \frac{\pi}{360}) + \text{atan2}(-0.012, 49.954) \right) * \sqrt{(2 * ((49.954 * 49.954) + (-0.012 * -0.012)) * (1 - \cos(-\mathbf{A} * \frac{\pi}{360})))} \quad (3.18)$$

Eq 3.17 has only one form and is not affected by signs of I or J components of tool orientation vector.

3.7 Closure

In this chapter, the postprocessing process was described. The algorithms developed and the equations that are necessary to find the locations of the machine tool axes were also given. Moreover, it was also described the special software features which were used to overcome gouging problems in “phase reverse process”.

CHAPTER 4

SIMULATION PART OF THE SOFTWARE

4.1 Introduction

When building the simulation environment in Manus 1.0 software, firstly the machine tool components and their motion axes have been defined. These machine tool components are the table support, the table and the spindle head of the machine tool. These are modeled and placed in the simulation environment considering the machine tool kinematics. In order to form the exact machining environment in the simulation screen, the motions and rotations of these machine tool components are all generated by considering this machine tool kinematics.

The second step of creating the virtual machining environment is to model and place the workpiece, cutting tool and other setup components. These setup components are the fixtures and clamping elements that are used for positioning and clamping the workpiece on the machine table. The locations of these setup components can change depending on the applications and therefore the software must allow the user to select and place these components on correct positions.

The other important issue that has to be considered is the generation of the motions for milling simulation. The linear motions and rotations of the machine tool components have to be already determined in *NC-Code Generation Software Component* in *Postprocessing Part* of the software before starting the simulation process.

When developing Manus 1.0 software, all these circumstances have been considered and special features have been added to the software to achieve these tasks. These features will be described in the following sections.

4.2 Displaying CAD Models in Manus 1.0

The CAD models of all the displayed machine components in Manus 1.0 can be prepared in any CAD software. When transferring these prepared models to the simulation environment, some particular subjects have to be considered.

The most important matter that has to be concerned is that all the models prepared in CAD software have to be converted to STL (stereolithography) format after the geometric model preparation is finished. STL is a triangular facet based representation that approximates surface and solid entities [42]. The STL file consists of a list of triangular facet data. Each triangular facet is uniquely identified by a unit normal (a vector perpendicular to the triangle and with a length of 1.0) and by three vertices. The normal and each vertex are specified by three coordinates, so there is a total of 12 numbers stored for each facet.

Since the STL formatted model is a triangular approximation of the CAD model, a model that is transformed to STL format will be displayed with triangular approximations as shown in Figure 4.1.

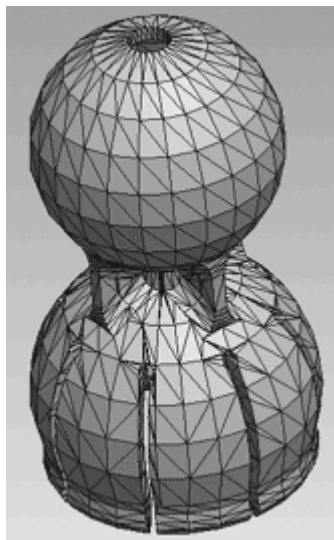


Figure 4.1 A sample STL formatted CAD model [43]

CAD programs often contain optional settings that will give the designer some level of control over the quality of the STL file output. The most common settings or variables pertain to "Chord Height" and "Angle Control." "Angle Control" parameter allows the user to adjust the accuracy of small-radius curves. The setting ranges from 0 (least accurate) to 1 (maximum accuracy).

"Chord Height" is a term that numerically describes the maximum distance between the actual part surface and the tessellated surface of the STL file that describes the part. As shown in Figure 4.2.

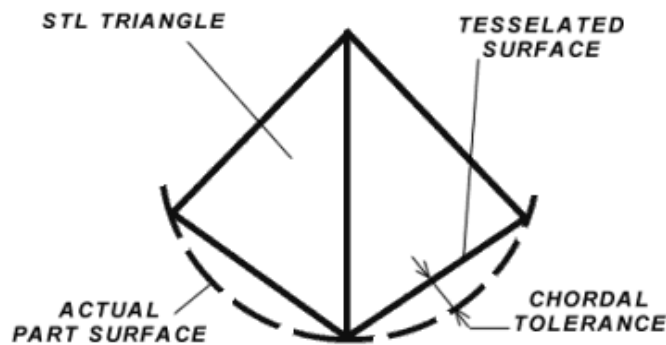


Figure 4.2 The definition of chord height [43]

The smaller the chord height, the more triangles required to describe a particular section of the part geometry. As the number of triangles increases and the relative triangle size decreases, the shape begins to represent more accurately the actual model as shown in Figure 4.3.

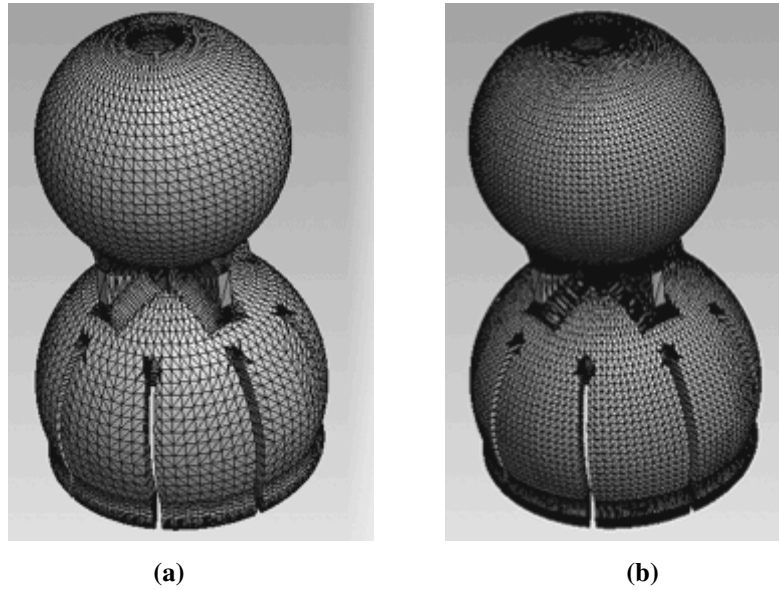


Figure 4.3 Sample STL formatted CAD models with (a) medium chord height, (b) fine chord height [43]

Although the chord height value is not critical for machine tool components like machine inner walls, operating door and table support, all other components which will be displayed in the simulation screen of Manus 1.0 have to be prepared with maximum allowable accuracy for collision detection processes. Therefore minimum allowable chord height and angle control value of 1 are used in generating the STL files of these components.

A typical STL triangular faced is shown in Figure 4.4. The facet normal vector coordinates are displayed in this figure. The facet normal vectors are used to form the lighting conditions on the displayed components. In the absence of a well-prepared lighting in the simulation environment, the components that are displayed somewhat become unrecognizable. Therefore for every triangular facet displayed in the simulation screen, the normal vectors of these facets are also taken into account. The normal vector designation function of OpenGL Graphics API is the “glNormalf” function [44]. The normal vector coordinates given in the STL file are used in this function as follows;

```
glNormalf (normal vector X component, Y component, Z  
component) ;
```

```

facet normal 0.000000e+00 -1.000000e+00 0.000000e+00
  outer loop
    vertex -9.900000e+01 -1.850000e+01 1.880000e+01
    vertex -1.300000e+01 -1.850000e+01 0.000000e+00
    vertex -1.300000e+01 -1.850000e+01 1.880000e+01
  endloop
endfacet

```

Figure 4.4 A sample triangular facet in STL file format

In order to draw the triangular facets on the simulation screen, the “glVertex3f” function of OpenGL graphics library is used [44]. The coordinates of each vertex is given in the STL file as shown in Figure 4.4. These coordinate values for each vertex is used in the same manner as “glNormalf” function while programming. Each three vertices of the triangular facet have to be designated in the program code. The OpenGL graphics API automatically draws lines to connect these vertices and to form the triangle. A vertex drawing format is as follows;

```

glVertex3f (vertex X coordinate, Y coordinate, Z
coordinate);

```

A triangular facet can be displayed in the simulation screen with demonstrating both its normal vector and three vertices. The start and end of each triangular facet drawing code has to be defined in the programming code. “glBegin” and “glEnd” functions are used for this purpose [44]. A general triangular forming program code is given in Figure 4.5.

```
glBegin();  
    glNormalf (normal vector X component, Y component, Z component);  
    glVertex3f (vertex1 X coordinate, vertex1 Y coordinate, vertex1 Z coordinate);  
    glVertex3f (vertex2 X coordinate, vertex2 Y coordinate, vertex2 Z coordinate);  
    glVertex3f (vertex3 X coordinate, vertex3 Y coordinate, vertex3 Z coordinate);  
glEnd();
```

Figure 4.5 General triangular facet forming program code

4.3 Displaying Simulation Components

The simulation components can be classified in two main groups. These are the machine tool components and setup components. In order to generate a well-prepared simulation software, these components have to be displayed in their exact locations.

4.3.1 Machine Tool Components

Although the machine tool involves many structural components, only the ones that are required for machining simulation have been modeled in Manus 1.0 software. These displayed machine tool components are the machine inner walls, machine operating door, table support, table and spindle head of the machine tool as shown in Figure 4.6. All the machine tool components that are displayed in the simulation screen are already present in the Manus 1.0 software database. This means that when the simulation screen of Manus 1.0 software is activated, the machine view comes with these predefined machine tool components.

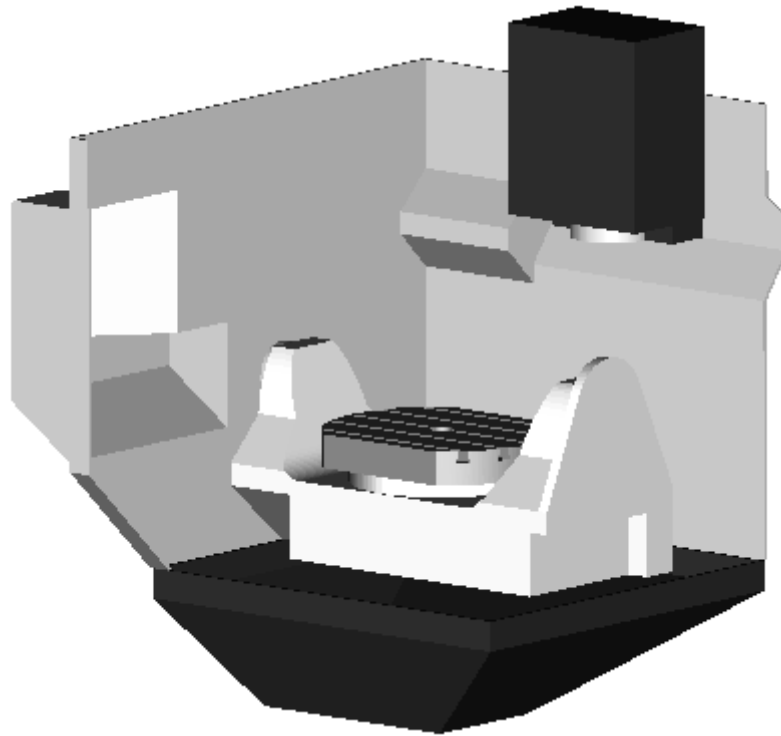


Figure 4.6 Machine tool predefined components in Manus 1.0

4.3.2 Setup Components

The second group of components needed for simulation display is the setup components. These are the user defined components and include the workpiece, the cutting tool, the tool holder, the fixtures and the clamping elements used in the setup.

4.3.2.1 Workpiece Model

The workpiece model, after prepared and transformed to STL format by using any CAD software, can be displayed in Manus 1.0 software screen. For selecting the proper workpiece model, the “Workpiece Select and Preview” menu item of Manus 1.0 is activated. The user can select and also preview the selected workpiece model, in the displayed “Workpiece Select and Preview” window. A

sample view from the “Workpiece Select and Preview” window is shown in Figure 4.7.



Figure 4.7 “Workpiece Model Select” window of Manus 1.0 software

4.3.2.2 Tooling Equipments

The tooling equipments are the cutting tool and the tool holder. The cutting tool type can be selected in Manus 1.0 software. The cutting tool types are the flat and the ball endmill tools, which are the most commonly used cutting tools in the milling operations. The user can select any of these cutting tools and can specify the diameter of the cutting tool by activating the “Tooling Settings” menu item of Manus 1.0. The sample view from “Tooling Settings” window is shown in Figure 4.8. The user also specifies the tool holder in the same environment. There are seven different tool holders currently used for “Mazak Variaxis 630-5X” machine tool in METU-BILTIR Center. All these currently

available tool holders are predefined in software database and the user can select any of these tool holders while adjusting tooling equipments. The tool holders used in “Mazak Variaxis 630-5X” are all given in Appendix B. The length of the tooling equipment is adjusted in “Tooling Settings” window. The length of the tooling equipment does not have to be confused with the length of the cutting tool. Tooling equipment length is the total length of the tooling equipment including the cutting tool effective length and the tool holder length outside the spindle head bearing as shown in Figure 4.9. Tooling length or (tooling equipment length) is measured via measurement probe of the machine tool.



Figure 4.8 “Tool Settings” window of Manus 1.0 software

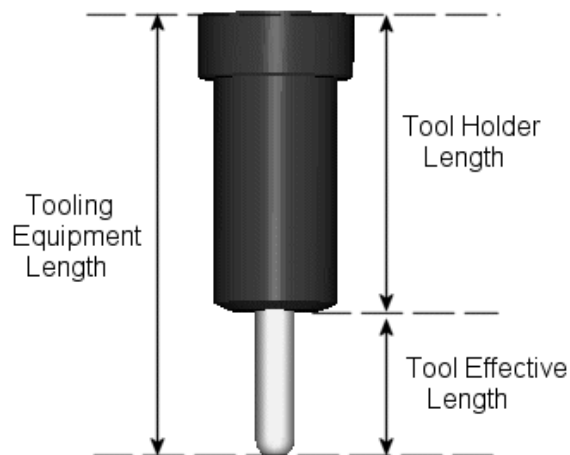


Figure 4.9 Tooling equipment length

At the same instant when the user specifies the tooling information, the defined tooling setup is displayed in the “Tooling Preview” window and any changes made to tooling settings parameters are automatically displayed in this preview window. The sample view from the “Tooling Preview” window is shown in Figure 4.10.

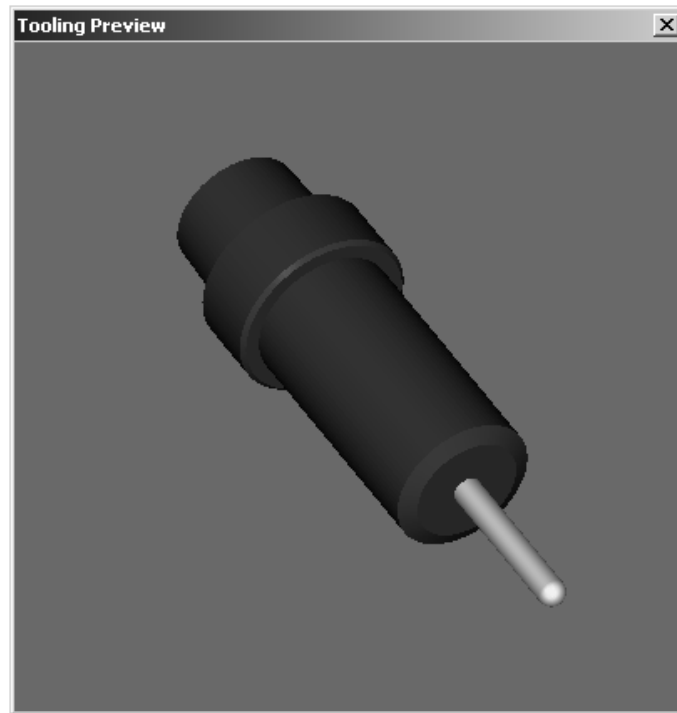


Figure 4.10 “Tooling Preview” window of Manus 1.0 software

4.3.2.3 Fixtures and Clamping Elements

The fixtures are used in the milling operations to position the workpiece in predefined locations, thus allowing the operators to clamp the workpieces every time at the same location. This is especially important in mass production. However, regardless of the production amounts, there has to be a fixture and clamping element set prepared for every machining operation to support the workpiece during machining. The fixtures used in the machining setups have unique forms and for every different machining setup, a new distinct fixture and

clamping element set have to be prepared. These sets may have primitive or advanced forms depending on the applications. In order to prepare a milling simulation environment, it is compulsory to define these fixture components. In the absence of these components, collisions cannot be completely found and it becomes meaningless trying to detect collisions.

The Manus 1.0 software allows the users to define the fixtures and clamping elements in the simulation environment. As in the case of workpiece selection, the fixtures used in the setup can be selected via activating the “Fixture Settings” menu item. The “Fixture Settings” window is shown in Figure 4.11. The “Fixture Select” sub item in this window is utilized for selection of the required fixturing elements. The fixtures have to be converted to STL format in order to make them perceivable by the Manus 1.0 software. Up to seven different fixtures can be selected and positioned in the simulation environment.

Manus 1.0 software also allows the user to create simple fixture components like cylindrical and prismatic objects without the use of a CAD software program. The user defines the type of the geometry and enters the required dimensions to create the CAD models of these objects in Manus 1.0.

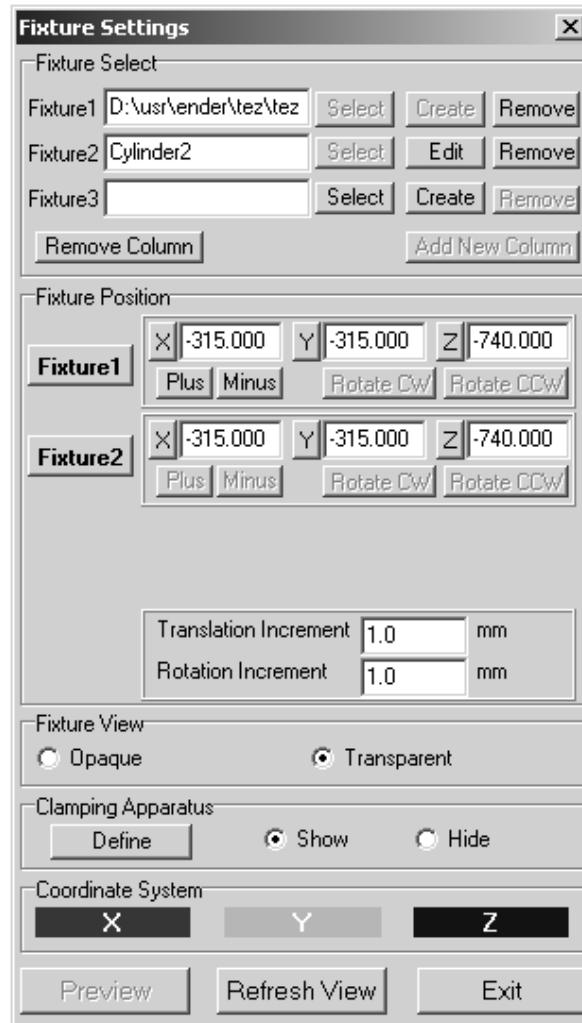


Figure 4.11 “Fixture Settings” window of Manus 1.0 software

Defining clamping elements is also very important when preparing the machining setup in the simulation environment. Manus 1.0 software has a library for clamping elements and this library involves all the clamping elements used in “Mazak Variaxis 630-5X” machine tool. The user can select the proper clamping element set via activating the “Clamping Element Settings” sub window inside the “Fixture Settings” window. The “Clamping Element Settings” window is displayed in Figure 4.12. The clamping element sets used in “Mazak Variaxis 630-5X” are all given in Appendix C.

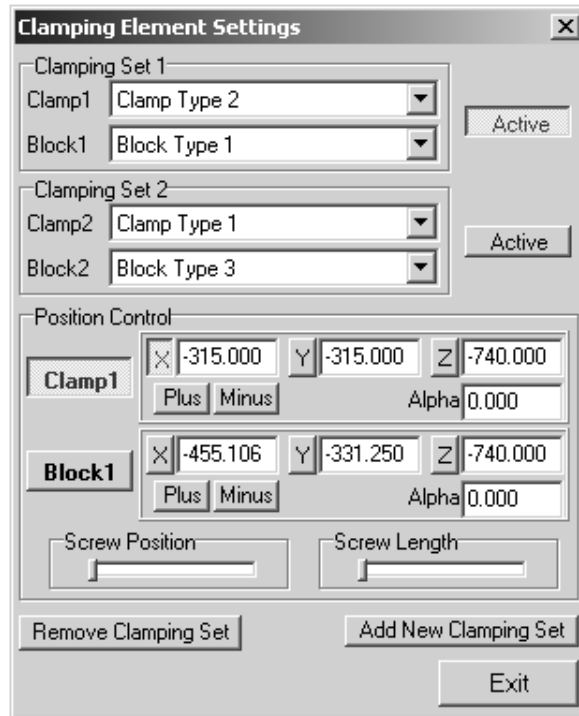


Figure 4.12 “Clamping Element Settings” window of Manus 1.0 software

4.4 Positional Adjustments of Machine Tool and Setup Components

In order to simulate milling operation, the positional adjustments of the setup components have to be performed. The locations of machine tool components are fixed and predefined in Manus 1.0. However, locations of setup components have to be arranged by the user during machining setup preparation. Manus 1.0 software allows the user to define the locations of these components.

4.4.1 Positional Adjustment of Workpiece Model

The WCS location value is entered to the “WCS Select and Preview” menu item of the software, thus the selected workpiece is displayed in this entered location. This location can be controlled via activating the “Machine Setup Preview” menu item. “Machine Setup Preview” window allows user to view and control the locations of machining setup components prior to starting the

machining simulation. A sample simulation view from “Machine Setup Preview” window is shown in Figure 4.13. The selection and positional adjustment of the workpiece model are completed in this sample shown in Figure 4.13.

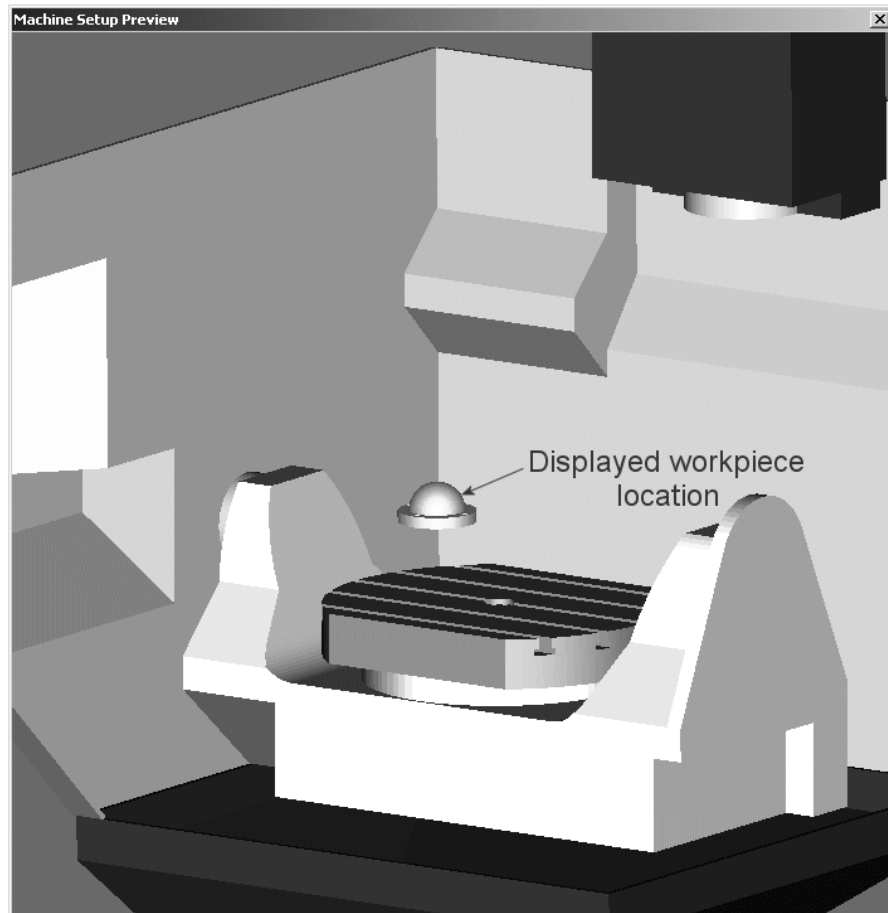


Figure 4.13 “Machine Setup Preview” window of Manus 1.0 software displayed after workpiece location is determined

4.4.2 Positional Adjustment of Fixtures

As described in the previous sections, the user can define the required fixtures in the simulation environment of Manus 1.0 software. The locations of these fixtures also have to be defined in the simulation screen. In order to define the fixtures and their locations on the machine table, the “Fixture Settings” menu item has to be activated. The “Fixture Settings” window is displayed by clicking

on this menu item. In addition to this window item, the “Machining Setup Preview” window is also displayed just by activating “Fixture Settings”. In the default case all the selected or created fixtures come to the simulation environment in the machine table center location. In “Fixture Settings” window, the user is allowed to adjust the locations of these fixtures. The “Fixture Position” sub item of this window is utilized for positional adjustment of the fixtures. The “Fixture Position” sub item is shown in Figure 4.11. Another sample view from “Machine Setup Preview” window is displayed in Figure 4.14, which displays the machine setup just after the selection and positional adjustments of the fixtures used in the setup have finished.

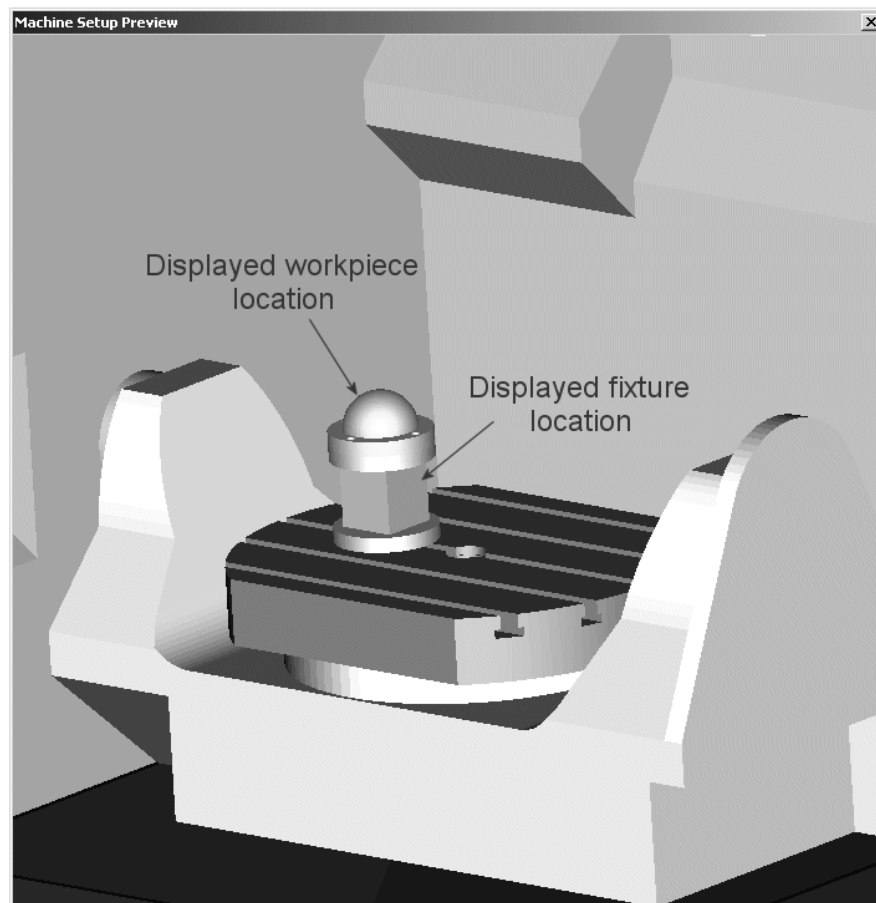


Figure 4.14 “Machine Setup Preview” window of Manus 1.0 software displayed after fixture location is determined

4.4.3 Positional Adjustment of Clamping Elements

In order to define the locations of clamping elements in the simulation environment, the “Clamping Elements Settings” window is activated via clicking on the “Define” button in “Clamping Elements” sub item of “Fixture Settings” window. As in the case of fixture position adjustment, the clamping elements selected from software library come to the simulation screen at the machine table center location. The proper adjustments on location can be performed by activating the “Position Control” sub item inside “Clamping Elements Settings” window. The “Position Control” sub item is shown in Figure 4.12. With the control elements inside “Position Control” sub item, the user can define the location of the clamping elements including the clamps, the support blocks and the clamping bolts and nuts. The clamping set can be in any orientation with respect to the machine coordinate system. Therefore, angular orientation of the clamping apparatus set also has to be defined. The Manus 1.0 software gives full control to user to adjust the location of these components in any desired orientation. Another sample view from “Machine Setup Preview” window is displayed in Figure 4.15, which displays the machining setup just after the selection and positional adjustments of the clamping elements used in the setup have finished.

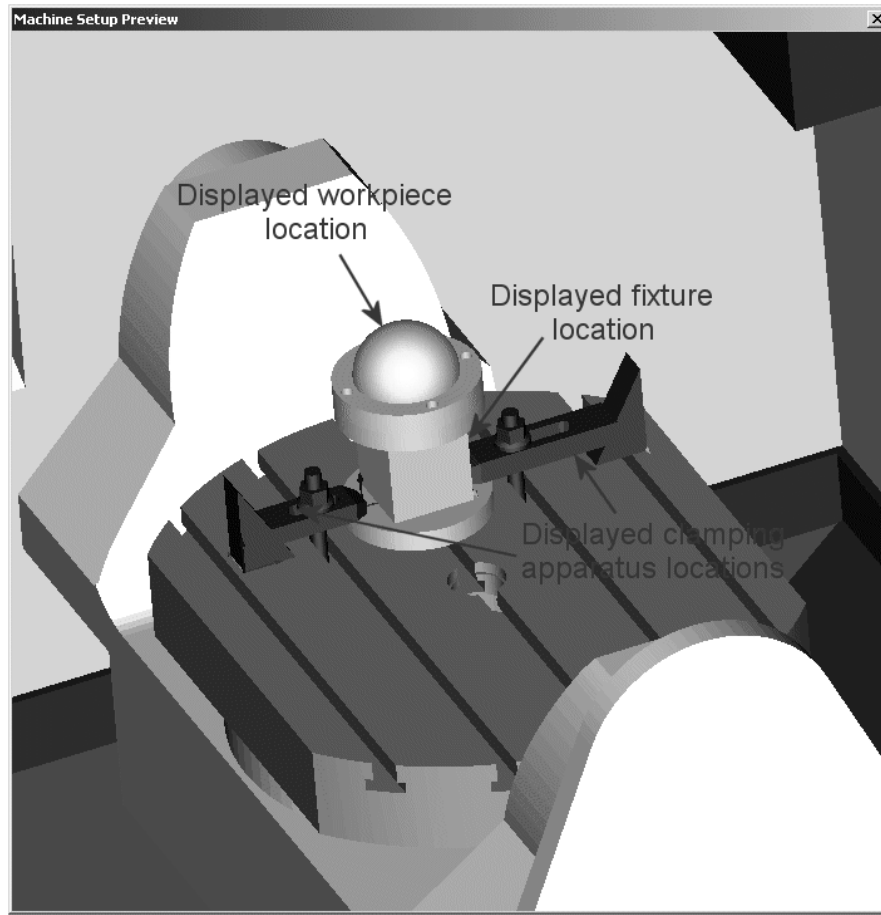


Figure 4.15 “Machine Setup Preview” window of Manus 1.0 software displayed after the locations of clamping elements are determined

4.5 Motion Generation

The simulation of milling operations can be achieved by generating the exact motions of machine components in the simulation screen. Therefore, the machine tool kinematics has to be understood completely. The “Mazak Variaxis 630-5X” machine tool has three linear axes: X, Y and Z axes. The motions at these three linear axes are activated via spindle head. The other two axes are rotary axes which are generated by the table rotational motions. These two rotary axes are A and C axes for “Mazak Variaxis 630-5X”. The rotation axes of these rotary axes have a great importance in machine tool kinematics, since they affect the movements of whole machine tool components. The A and C rotation axes are

given in “Machine Technical Specifications” in Appendix A. The other important subject is the machine tool axis limitations. Both the linear axes and rotary axes have limited motion ranges (a.k.a. “travel”) given in Appendix A.

The Manus 1.0 software is generated by considering all these machine tool kinematics and the simulated motions are the exact motions of the machine tool components. The motion information necessary to simulate milling operations is created in the *NC-Code Generation Software Component* during the postprocessing process. Therefore, the generated motion code is the same as the G-Code motion code. However, the G-Code determines only the start and final location of the machine components for every single line of code and in between it is not the function of postprocessor or G-Code itself to adjust the travel path of the cutting tool. For instance, a G-Code motion line set is given as;

```
G1X0.Y54.348Z150. C0A-20.  
G1X1.2Y54.395Z150.437. C0A-22.461
```

The current line (first line) shows the start location of motion for machine tool components and the subsequent line (second line) indicates the motion end location for these components. The motions that the machine tool components will generate for these motion code line sets cannot be determined by the postprocessor or G-Code, it only depends on the machine tool kinematics. The necessary movements in between these start and end locations have to be determined for a proper machining simulation. This can only be achieved by finding the motion paths of these components. After finding the motion paths of these machine components, points have to be generated on these paths with equal intervals and these points are used for generating the motions in between these start and end locations of the motion code line sets. In Manus 1.0, an algorithm has been developed to find these motion paths and points. The algorithm generates points on the motion path for every motion axes. The schematic description of the point generation process for a rotary axis is given in Figure 4.16.

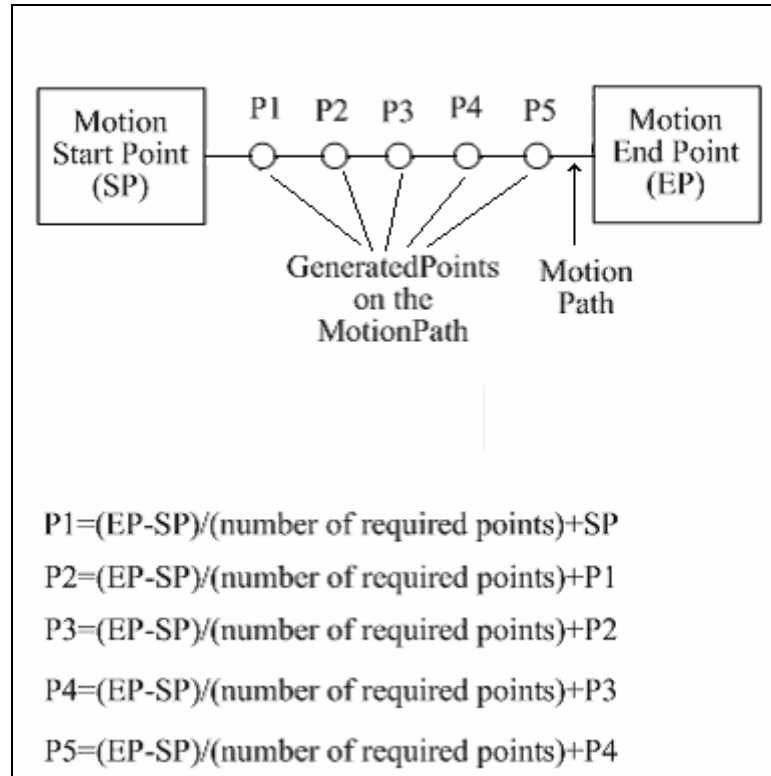


Figure 4.16 Schematic description of point generation process on motion path

4.6 Displaying Machining Simulation

Machining simulation in Manus 1.0 is displayed in a separate window called “Simulation” window. This simulation window is the exact copy of “Machine Setup Preview” window. The “Simulation Play Control” window is used to control the simulation process. The “Simulation Play Control” window was shown in Figure 2.4 and its sub items were described in Section 2.5.3.1 of Chapter 2.

The second simulation control window is the “View Display Settings” window. All machine components are initially displayed as visible and opaque in Manus 1.0. The visibilities of these machine tool and setup components can be adjusted by using this window shown in Figure 2.6. The reason for developing component view settings adjustment capability in Manus 1.0 was described in Section 2.5.3.3 of Chapter 2.

Manus 1.0 software also allows the user to control dynamically the view direction in simulation environment as described in Section 2.4.3.2 of Chapter 2. The user can look the machining zone in any orientation during simulation by using mouse and keyboard buttons. By clicking on mouse middle button and also by moving the cursor on the simulation screen, the view direction vector can be rotated. The view vector rotation center always lies on the machine table center. As described previously in this chapter, the necessary machine tool components are modeled in Manus 1.0 software environment. The inner walls and the machine-operating door have been also modeled to clarify the machining view direction. However, these walls can make the machining zone invisible when the view vector direction is changed. In order to prevent this, an algorithm has been developed in Manus 1.0, to make these walls invisible when the view vector orientation is changed. Some sample views from simulation screen were given in Figure 2.3 in Chapter 2. In these views, the view vector orientations are changed and consequently the walls that come in front are disappeared.

The other dynamic view manipulation during simulation can be performed by translating the view vector on the screen plane. This translation can be performed by clicking on the middle mouse button and the “Shift” key on keyboard. After clicking on this button and the key, any cursor movements will translate the view vector in the cursor moved direction. In addition, the view vector can be made closer or farther by clicking on the middle mouse button and “Alt” key on keyboard. When these button and key are clicked and cursor is moved up or down, the zooming effect is generated on the simulation screen.

4.7 Closure

In this chapter, the simulation part of the software, the CAD model displaying strategy in Manus 1.0 and the capability of the software in machine setup preparation were described. In addition, the motion generation principle that is used in machining simulation was also given. Finally, the machining simulation displaying features were illustrated.

CHAPTER 5

VERIFICATION PART OF THE SOFTWARE

5.1 Introduction

The tool path verification process has to be executed to ensure the correctness of the tool path generated in *Postprocessing Part* of the software. In the absence of a tool path verification process, it can be very dangerous to execute the generated tool path file (G-Code file) in real machining operations. As described in Section 1.3.4 of Chapter 1, severe problems may occur like unpredictable gouges or collisions between machine components. Gouging and especially collisions are serious problems that may give harm to the workpiece, the cutting tool or even machine tool components like table and spindle head. Therefore, these problems have to be detected and solved prior to executing the machining code in the real machining environment.

Two main control methods are carried out in Manus 1.0 for tool path verification process. These are the visual control method over tool path which is performed during machining simulation process and the geometric control method.

The visual control over tool path is executed during simulation process. While the machining operation is running on the simulation screen, the user traces the simulation process and visually tries to detect any gouges or collisions of the cutting tool, the tool holder and the spindle head with other machine tool and setup components. If no specialized features are embedded into the verification software, it is only possible to recognize major gouges or collisions by visual inspection method and relatively minor gouges or collisions cannot be recognized with this method. However, any gouges or collisions regardless of their sizes may

give rise to great harms to setup and machine tool components. The geometric control process solves the problem of unrecognizable gouges and collisions. The control over tool path is carried out by performing some numerical computations. In Manus 1.0, currently gouge and collision detection algorithms have been developed for collisions of cutting tool with the workpiece, fixtures, clamping elements and the table. The collision detection algorithms for the tool holder and the spindle head have not been currently available in Manus 1.0 software. The numerically computed gouges and collisions are reported in Manus 1.0 and the results are displayed in a report window prior to executing the simulation process. Via this report window, the user is informed about in which NC-Code line the collision or gouging occurs and which machine tool and setup components are collided with the cutting tool.

In order to enhance the software collision checking capability during visual inspection process, a different concept is used. This is the integration of the visual control method with the geometric control method. By this way, the calculated collisions and gouges are displayed during simulation process. This is achieved by changing the colors of the components that are colliding with the cutting tool. The colliding component takes the color of red just at the time collision occurs as described in Section 2.6.3. This enhances the visual inspection capabilities of the users and makes the geometric collision detection more apprehensible. Another important feature of Manus 1.0 software is the collision warning capability in risky circumstances as discussed in Section 2.6.1.

The flow chart and description of the collision check process is given in Appendix D.

5.2 Mathematical Background for Collision Detection

All machine components have surfaces in the form of triangular facets as described in Section 4.2. Each triangular facet has three edges and three vertices. In order to find the collisions between the cutting tool and the machine tool and setup components, the vertices and the edges that are forming these triangular

facets are utilized. Each edge of the triangle can be represented as a ray by utilizing the ray equation. The ray equation is defined as follows [45]:

$$\mathbf{P}(t) = \mathbf{S} + \mathbf{D}(t) \quad (5.1)$$

where “P(t)” is a 3D point on the ray at a distance “t” along the ray, “S” is the starting point, and “D” is the direction vector. Finding the ray equations for each edge of a given triangle is performed as described in the illustration given in Figure 5.1.

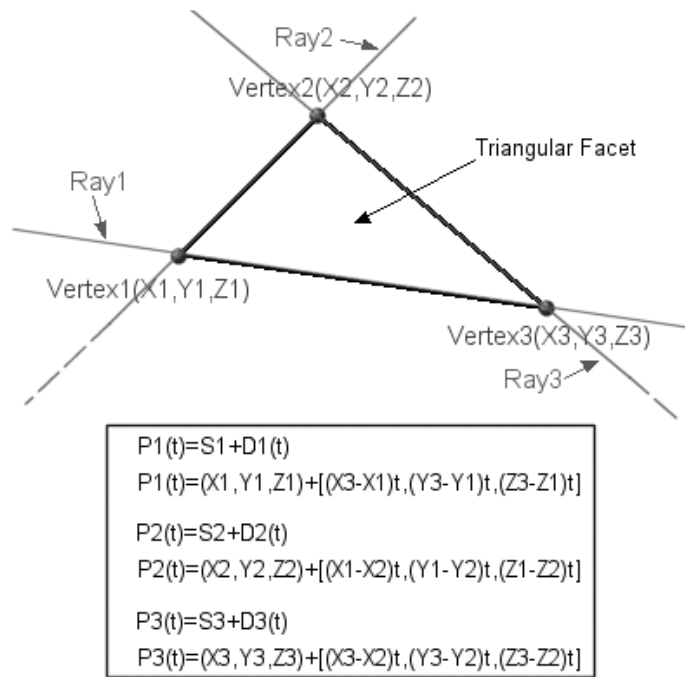


Figure 5.1 Finding ray equations for triangular facets

A flat endmill tool has a cylindrical geometry and in order to perform collision checking algorithm, the cylinder equation has to be utilized. The equation of an infinite cylinder that orients along Z-Axis is defined as follows:

$$(\mathbf{CX} - \mathbf{CX}_0)^2 + (\mathbf{CY} - \mathbf{CY}_0)^2 = \mathbf{R}^2 \quad (5.2)$$

where “CX” and “CY” are the X and Y-Axis coordinates of a point on the cylinder surface respectively, “CX₀” and “CY₀” show the X and Y-Axis location of the cylinder axis and “R” is the radius of the cutting tool.

A ball endmill tool consists of a hemisphere and a cylinder section. The cylinder equation Eq. 5.2 is utilized for the cylindrical section of the ball endmill tool. The sphere equation for the hemi-spherical section of the ball endmill tool is as follows;

$$(SX - SX_0)^2 + (SY - SY_0)^2 + (SZ - SZ_0)^2 = R^2 \quad (5.3)$$

where “SX”, “SY” and “SZ” are the X, Y and Z-Axis coordinates of a point on the sphere surface respectively, “SX₀”, “SY₀” and “SZ₀” show the X, Y and Z-Axis location of the sphere center and “R” is the radius of the cutting tool.

Tool path verification process utilizes these equations for detecting gouges and collisions. There are two algorithms that are developed for collision detection process. The first of these algorithms is developed to find the collision of flat endmill tools and the cylindrical section of ball endmill tools with other setup and machine tool components. The “ray-cylinder intersection” equations are employed for this purpose. The second algorithm is developed to find gouges and collisions of hemi-sphere section of the ball endmill tool with setup and machine tool components. The “ray-sphere intersection” equations are employed for this purpose.

5.2.1 Ray-Cylinder Intersection

In order to apply the ray-cylinder intersection process, the exact locations of the cutting tool and the machine components have to be already determined. The ray equations of the triangular facets for every single machine components are formed by employing Eq 5.1. The cylinder equation of the flat endmill cutting tool is formed by employing Eq 5.2. The next step is to form the ray-cylinder intersection equations. This is accomplished by solving the ray equation Eq. 5.1 and the cylinder equation Eq. 5.2 together. For ray-cylinder intersection to occur, the P(t) value of the ray equation Eq 5.1 must coincide with the cylinder surface. When these equations are solved together, a quadratic equation will be formed as given below;

$$\mathbf{A}_c \cdot t^2 + \mathbf{B}_c \cdot t + \mathbf{C}_c = 0 \quad (5.4)$$

where

$$\mathbf{A}_c = \mathbf{D}_x^2 + \mathbf{D}_y^2 \quad (5.5)$$

$$\mathbf{B}_c = 2 \cdot (\mathbf{S}_x \cdot \mathbf{D}_x + \mathbf{S}_y \cdot \mathbf{D}_y) \quad (5.6)$$

$$\mathbf{C}_c = \mathbf{S}_x^2 + \mathbf{S}_y^2 - \mathbf{R}^2 \quad (5.7)$$

where “ \mathbf{D}_x ” and “ \mathbf{D}_y ” are the X and Y axes components of the ray direction vector respectively, “ \mathbf{S}_x ” and “ \mathbf{S}_y ” are the X and Y axes components of the start point. The “ t ” value in Eq 5.4 has to be calculated to find the position of the intersection of the infinite ray with the infinite cylinder.

The “ t ” value of Eq 5.4 is found as follows;

First the discriminant of the quadratic equation is found:

$$\Delta = \mathbf{B}_c^2 - 4 \cdot \mathbf{A}_c \cdot \mathbf{C}_c \quad (5.8)$$

Then, the “ t ” values can be found as:

$$t_1 = \frac{-\mathbf{B}_c + \sqrt{\Delta}}{2 \cdot \mathbf{A}_c} \quad (5.9)$$

$$t_2 = \frac{-\mathbf{B}_c - \sqrt{\Delta}}{2 \cdot \mathbf{A}_c} \quad (5.10)$$

If the calculated discriminant value is less than zero, this means the “ t ” values cannot be calculated. This has special meanings that will be described in the following sections.

5.2.1 Ray-Sphere Intersection

Similar to generation of ray-cylinder intersection equations, the ray-sphere intersection equations can be formed by solving the ray equation Eq 5.1 and

sphere equation Eq. 5.3 together. For collision to occur, the $P(t)$ value of the ray equation Eq 5.1 must coincide with the sphere surface. When these equations are solved together, a similar quadratic equation will be formed as given below;

$$\mathbf{A}_s \cdot t^2 + \mathbf{B}_s \cdot t + \mathbf{C}_s = 0 \quad (5.11)$$

where

$$\mathbf{A}_s = \mathbf{D}_x^2 + \mathbf{D}_y^2 + \mathbf{D}_z^2 \quad (5.12)$$

$$\mathbf{B}_s = 2 \cdot (\mathbf{S}_x \cdot \mathbf{D}_x + \mathbf{S}_y \cdot \mathbf{D}_y + \mathbf{S}_z \cdot \mathbf{D}_z) \quad (5.13)$$

$$\mathbf{C}_s = \mathbf{S}_x^2 + \mathbf{S}_y^2 + \mathbf{S}_z^2 - \mathbf{R}^2 \quad (5.14)$$

where “ \mathbf{D}_x ”, “ \mathbf{D}_y ” and “ \mathbf{D}_z ” are the X, Y and Z axes components of the ray direction vector respectively, “ \mathbf{S}_x ”, “ \mathbf{S}_y ” and “ \mathbf{S}_z ” are the X, Y and Z axes components of the start point. The “ t ” value in Eq 5.11 has to be calculated to find the position of the intersection of the infinite ray with the sphere. It is found as the same manner in ray-intersection process. First the discriminant of the quadratic equation is found:

$$\Delta = \mathbf{B}_s^2 - 4 \cdot \mathbf{A}_s \cdot \mathbf{C}_s \quad (5.15)$$

Then, the “ t ” values can be found as:

$$t_1 = \frac{-\mathbf{B}_s + \sqrt{\Delta}}{2 \cdot \mathbf{A}_s} \quad (5.16)$$

$$t_2 = \frac{-\mathbf{B}_s - \sqrt{\Delta}}{2 \cdot \mathbf{A}_s} \quad (5.17)$$

5.3 Algorithm for Detecting Collisions of Flat endmill

The ray-cylinder intersection equations are employed to find the collisions of the flat endmill tool with triangular facets of the machine tool and setup components. However, the ray-cylinder intersection equations are general term equations which are used for detecting intersections of infinite rays and infinite

cylinders. In the case of collision detection during machining simulation, there are no infinite objects; the cylinder represents a flat endmill tool which has a definite fixed length and the ray represents an edge of a triangular facet. Therefore the start and end points of this ray segment and cylinder has to be taken into account during collision checking in the milling operations.

The “ t ” values calculated using Eq 5.9 and Eq 5.10 shows the intersection points of an infinite ray with an infinite cylinder. “ t ” can get any real value between minus infinity to plus infinity. If the discriminant Eq 5.8 of the quadratic equation Eq 5.4 is less than zero, the calculated “ t ” values will be complex numbers with imaginary parts. This is an important situation that means the infinite ray does not intersect the infinite cylinder, or in other words these geometric objects are oriented parallel in the 3D space. Thus, the first step of calculating the ray-cylinder intersection is to check whether the discriminant of the quadratic equation is less than zero or not. If the calculated discriminant value is less than zero, the ray-cylinder intersection test is canceled for that specific triangular facet ray.

The ray equation Eq 5.1 is constructed for triangular facets as described in Figure 5.1. While the start point “ S ” of the ray is taken as one vertex of the triangle, the direction vector “ D ” is calculated by using the other vertex of the triangle. Thus, the zero value of “ t ” for the triangular facet ray represents the start point “ S ” of the ray segment and the value of one represents the end point (the other vertex) of the ray segment. This means that when the calculated “ t ” value is in the interval of zero and one, the point calculated using Eq 5.1 represents a point on edge of the triangular facet. This is the second check for the calculated “ t ” value. Any “ t ” values outside this range do not represent a point on the edge of the triangle and therefore the calculated intersection becomes invalid for these cases. This situation is described in Figure 5.2 and Figure 5.3 for the sample ray – cylinder intersections.

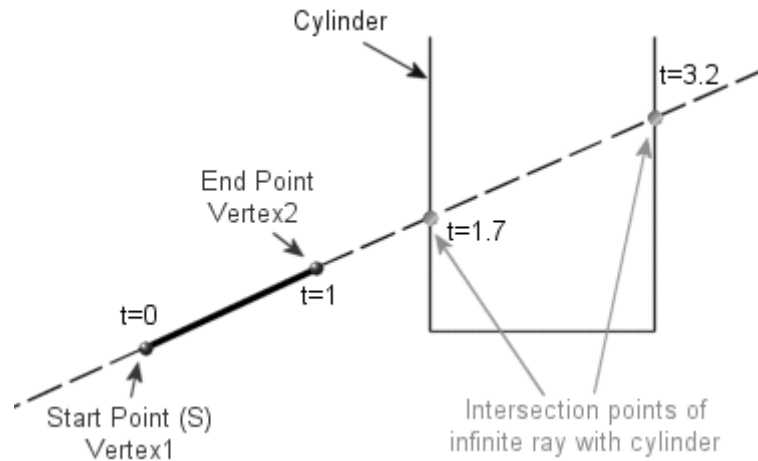


Figure 5.2 A ray intersection with cylinder where t values are greater than 1

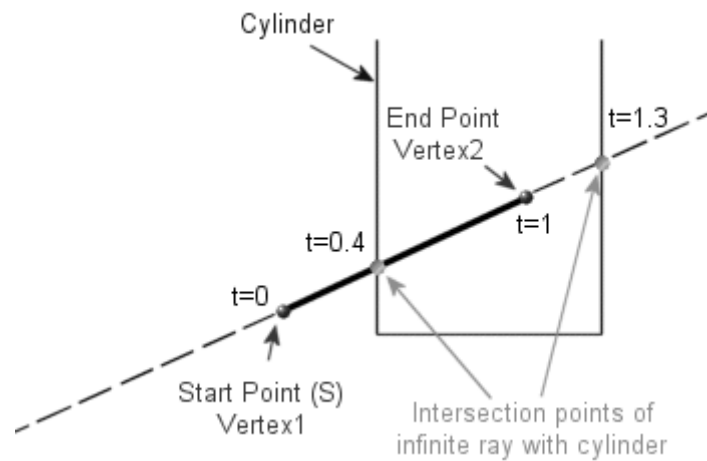


Figure 5.3 A ray intersection with cylinder where one t value is in the interval of 0 and 1

In Figure 5.2, a ray intersection with cylinder is shown. In this figure the calculated “ t ” values are greater than 1, which mean the intersection points are outside of the region bounded by triangle vertices. This indicates that the intersection does not really occur between the triangular facet edge and the cylinder, since the calculated intersection points are outside the edge of the triangle. In the second example shown in Figure 5.3, one of the calculated “ t ” values is inside the region bounded by the vertices. This shows that a real intersection occurs between triangular facet edge and the cylinder and this calculated ray-cylinder intersection point corresponds to a collision between a machine component surface and the flat endmill tool.

There are similar other conditions that have to be validated for collision checking. The first of these is the cylinder boundaries. The flat endmill tool will have definite position on Z-Axis and any triangular facet edges below this Z-Axis value will be automatically disregarded for collision check calculations. Therefore, the Z-Axis values at the start or end points of the ray segment must be greater than the cylinder base location Z-Axis value for collision check calculations to be performed. This situation is described in Figure 5.4 and 5.5.

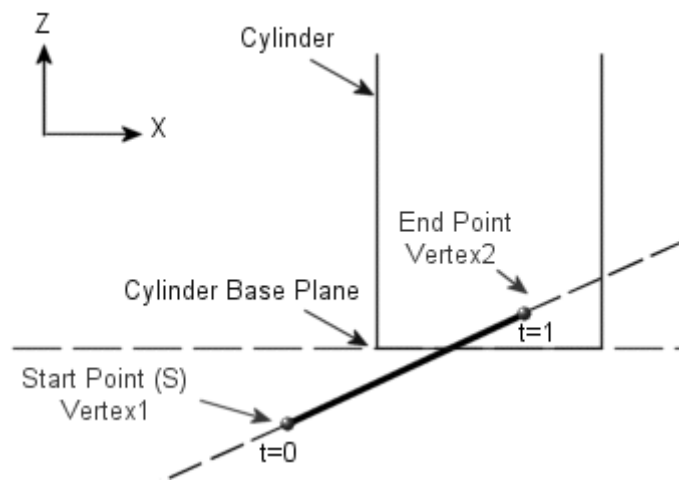


Figure 5.4 The end point of the ray segment is above the cylinder base plane

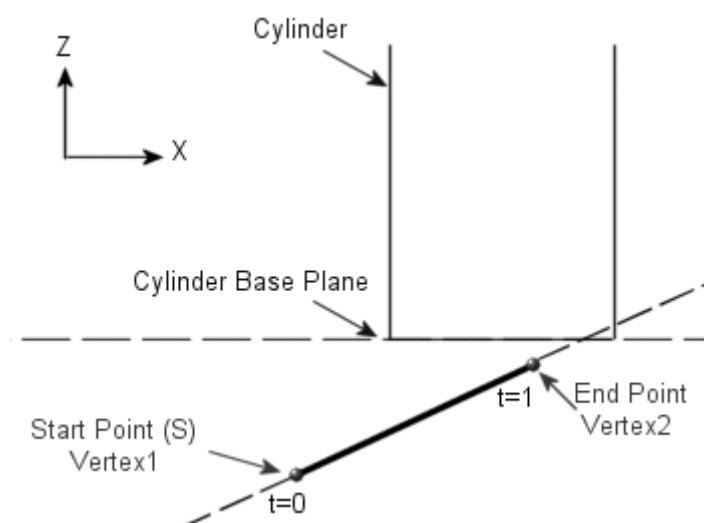


Figure 5.5 The ray segment is completely below the cylinder base plane

In Figure 5.4, the collision check calculations are performed for the ray segment, since its Z-Axis value at end point is greater than the cylinder base Z-Axis value. However, the ray segment shown in Figure 5.5 is disregarded in collision check calculations, since it is completely below the cylinder base plane.

It is not always required that the calculated “t” values (CTV) should be in between 0 and 1 as shown in Figure 5.3. There is a special case in which the ray segment is completely inside the cylinder as shown in Figure 5.6. In this case, the intersection points will be outside the bounded region of ray segment as shown in the figure and the CTV will not be in the interval of 0 and 1. However, this is special case that represents a collision. The condition for this case is that when one of the CTV is greater than 1, then the other “t” value must be lower than 0 for collision to occur.

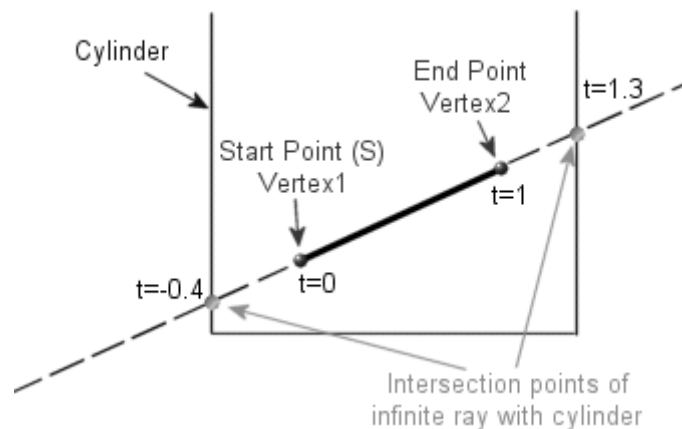


Figure 5.6 The ray segment is completely inside the cylinder

5.4 Algorithm for Detecting Collisions of Ball Endmill

The ball endmill tool consists of two geometric sections. The first section is the cylindrical section and the second one is the hemi-spherical section. For the cylinder section of the ball endmill tool, the same algorithm is employed given in Section 5.3. In order to find the collisions of the hemi-spherical section, a

different algorithm has been developed. In this algorithm, the ray-sphere intersection equations Eq 5.11 to Eq 5.17 are utilized. The ray-sphere intersection equations are general term equations which are used for detecting intersections of infinite rays with sphere. In the case of collision detection during machining simulation, the boundaries for the ray equation have to be determined as in the case of ray-cylinder intersection.

The “t” values calculated using Eq 5.16 and Eq 5.17 shows the intersection points of an infinite ray with a sphere. “t” can get any real value between minus infinity to plus infinity. If the discriminant Eq 5.15 of the quadratic equation Eq 5.11 is less than zero, the calculated “t” values will have imaginary parts. This situation is the same described in Section 5.3; which shows the infinite ray does not intersect the sphere. Thus, the first step of calculating the ray-sphere intersection is to check whether the discriminant of the quadratic equation is less than zero or not. If the calculated discriminant value is less than zero, the ray-sphere intersection test is canceled for that specific triangular facet ray.

Similar conditions can be employed for ray-sphere intersection process. Firstly, the CTV has to be checked to investigate any possible collisions. The “t” value must be in the interval of 0 and 1 for the ray segment (edge of the triangular) to intersect with the sphere. Any “t” values outside this range do not represent a point on the edge of the triangle and therefore the calculated intersection becomes invalid for these cases. This situation is described in Figure 5.7 and Figure 5.8 for sample ray –sphere intersections.

In Figure 5.7, a ray intersection with sphere is shown. In this figure, the CTV are greater than 1, which means the intersection points are outside of the region bounded by the triangle vertices. This indicates that the intersection does not really occur between triangular facet edge and the sphere, since the calculated intersection points are outside the edge of the triangle. In the second example shown in Figure 5.8, one of the calculated “t” values is inside the region bounded by vertices. This shows that a real intersection occurs between the triangular facet

edge and the sphere and this calculated ray-sphere intersection point corresponds to a collision between a machine component surface and the ball endmill tool.

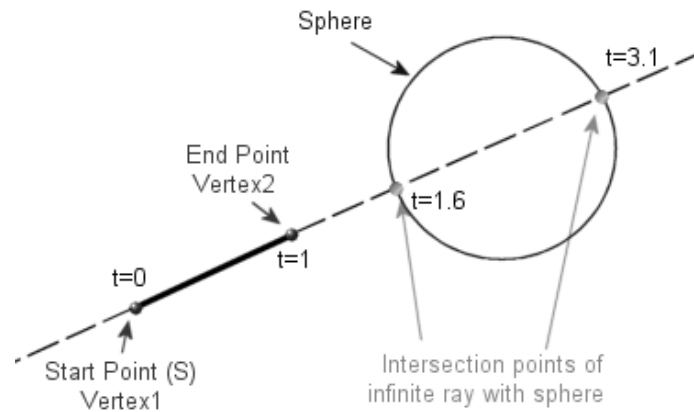


Figure 5.7 A ray intersection with sphere where t values are greater than 1

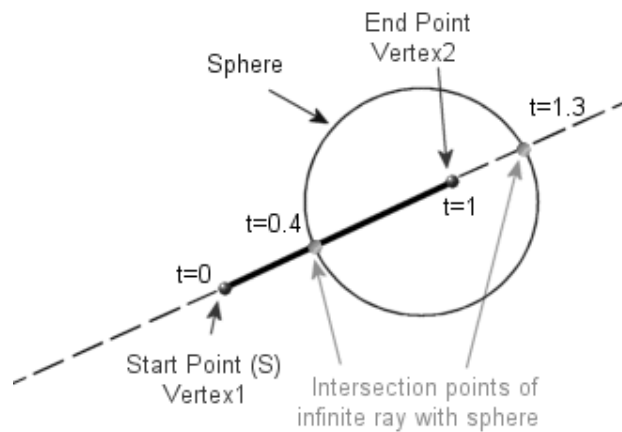


Figure 5.8 A ray intersection with sphere where one t value is in the interval of 0 and 1

There are similar other conditions that have to be validated for collision checking. The first of these is the sphere minimum Z-Axis value. The ball endmill tool will have definite position on Z-Axis and any triangular facet edges below this Z-Axis value will be automatically disregarded for collision check calculations. Therefore, the Z-Axis values at the start or end points of the ray segment must be greater than the sphere minimum Z-Axis value for collision

check calculations to be performed. This situation is described in Figure 5.9 and 5.10.

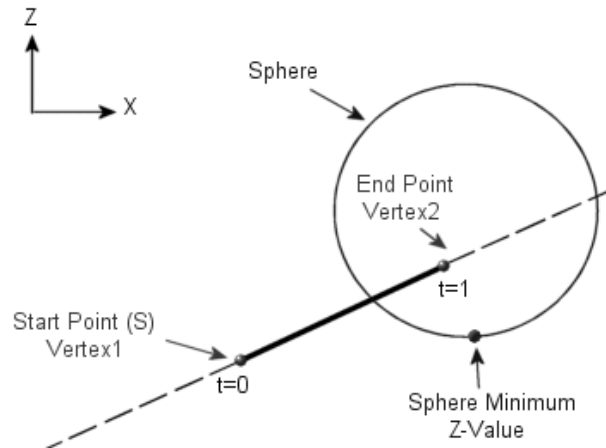


Figure 5.9 The end point of the ray segment is above sphere minimum Z-Axis value

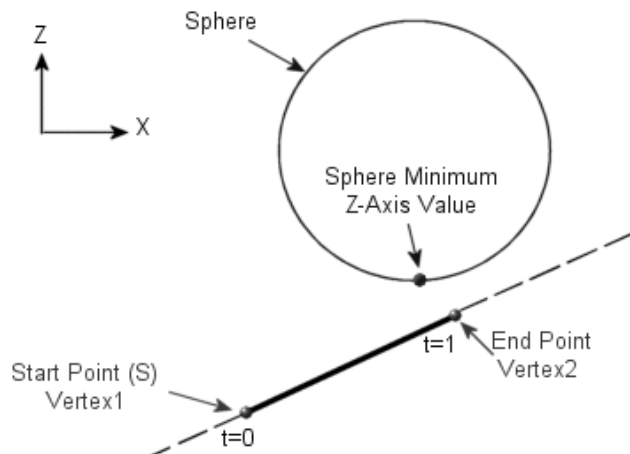


Figure 5.10 The ray segment is below sphere minimum Z-Axis value

In Figure 5.9, the collision check calculations are performed for the ray segment, since its Z-Axis value at end point is greater than the sphere minimum Z-Axis value. However, the ray segment shown in Figure 5.10 is disregarded in collision check calculations, since it is completely below the sphere minimum Z-value.

It is not always required the CTV in between 0 and 1 as shown in Figure 5.8. There is a special case in which the ray segment is completely inside the

sphere as shown in Figure 5.11. In this case, the intersection points will be outside the bounded region of the ray segment as shown in the figure and the CTV will not be in the interval of 0 and 1. However, this is special case that represents a collision. The condition for this case is that the when one of the CTV is greater than 1, then the other “t” value must be lower than 0 for collision to occur.

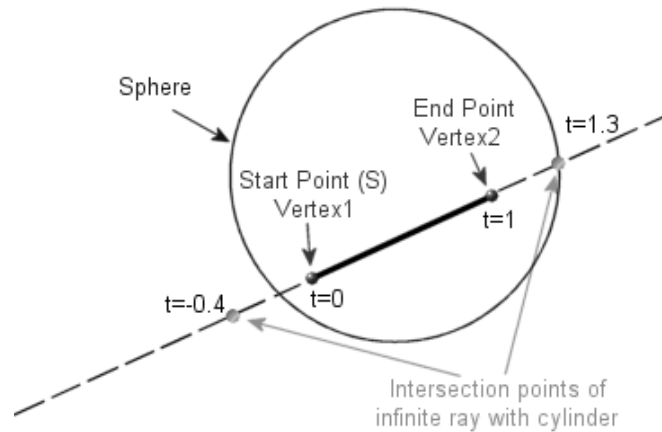


Figure 5.11 The ray segment is completely inside the sphere

5.5 Collision Warnings

Collision risks that can occur due to misplaced fixtures and clamping elements are found by creating a protective shield (warning region) around the cutting tool. When fixtures or clamping elements defined in the setup, enter in that zone, Manus 1.0 software warns the user via reporting and during simulation process. The protective shield around the cutting tool can be formed as shown in Figure 5.12. The protective shields have the same geometry with the related cutting tools. The protective shield for a flat endmill tool has a cylindrical geometry as shown in the figure. “P” is the “*Collision Check Approach Tolerance*” value which is defined by the user. The diameter of the protective cylinder is found by adding “2P” value to the cutting tool diameter and the protective cylinder end is lowered “P” value as shown in the figure. The protective shield for a ball endmill tool has two geometrical sections as shown in

Figure 5.12. The cylindrical part is the same as the flat endmill protection shield. The protection shield of the hemi-spherical section has a spherical geometry with a diameter “ $2P$ ” greater than the ball endmill tool diameter.

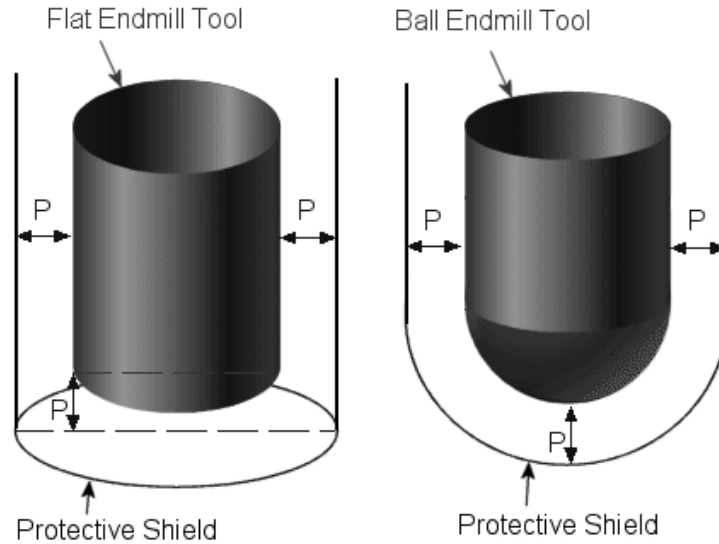


Figure 5.12 Protective shields around cutting tools

The same ray-cylinder intersection and ray-sphere intersection equations Eq 5.1 through Eq 5.17 and the same algorithms, which are used for flat endmill and ball endmill tools, are utilized for detecting collisions of these protective shields around the cutting tools. These equations and algorithms are employed with the new cylinder and sphere diameter values of the protection shield.

5.6 Closure

In this chapter, the verification part of the software described in detail. As the subtitles, the mathematical background which is necessary for collision detection calculations and the algorithms developed for collision detection of flat and ball endmill tools were explained. Moreover, the collision warning feature of the developed software was explained. The working principle of the collision warning process was given.

CHAPTER 6

SAMPLE APPLICATIONS

6.1 Introduction

The sample applications have been performed for various three-axis and five-axis milling operations for the sample workpiece given in Figure 6.1 by considering different surfaces on the workpiece. A machining setup is prepared to perform these sample milling operations. In all these applications, the tool paths are generated using CAM module of ProEngineer. The postprocessing is performed using Manus 1.0 software. In addition, all machining operations are simulated and verified in Manus 1.0 software. The machinings have been performed on the “Mazak Variaxis 630-5X” CNC machine tool and the particular machining processes are recorded by video camera for comparison of the simulation with the real machining operations.

6.2 Machining Setup Preparation

Prior to starting to define machining related parameters, the WCS value has to be determined and defined in the “Workpiece Settings” window of the software. The selected WCS is WCX=-314.094 mm, WCY=-308.739 mm, WCZ=-559.829 mm and G54 command is selected to define this WCS.

A workpiece model is prepared in CAD module of ProEngineer as shown in Figure 6.1. The dimensions of the workpiece are given in Figure 6.2. The machining reference coordinate system is also shown in the figure. The material of workpiece is “cast polyamide”.

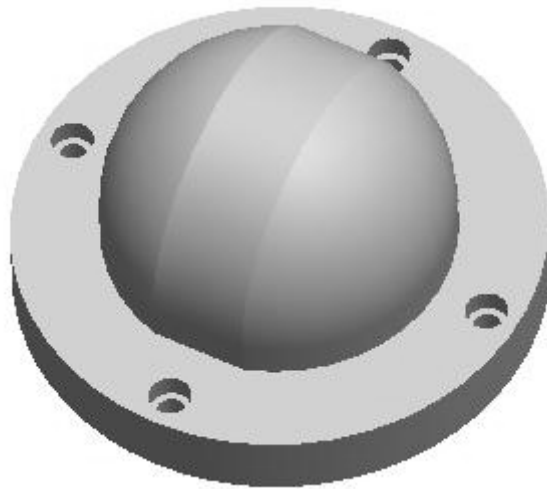


Figure 6.1 Workpiece model prepared for sample machining operations

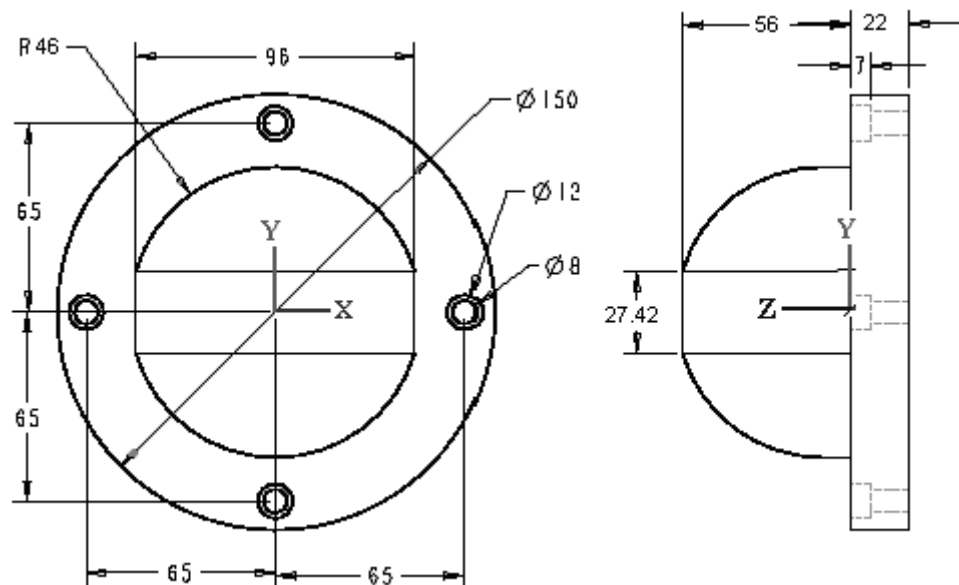


Figure 6.2 Dimensions of sample workpiece

The machining setup prepared is shown in Figure 6.3 through Figure 6.5. This setup is prepared to perform the three-axis and simultaneous five-axis finish milling operations. By employing the features of Manus 1.0 software, this machining setup has been prepared in Manus 1.0 simulation environment as shown in Figure 6.6 through Figure 6.8.



Figure 6.3 Machining setup (View-1)

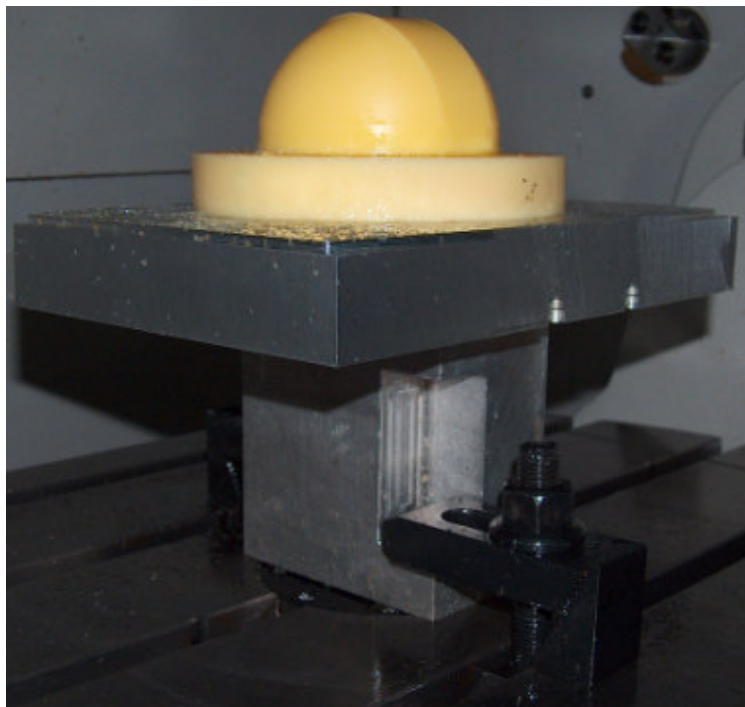


Figure 6.4 Machining setup (View-2)

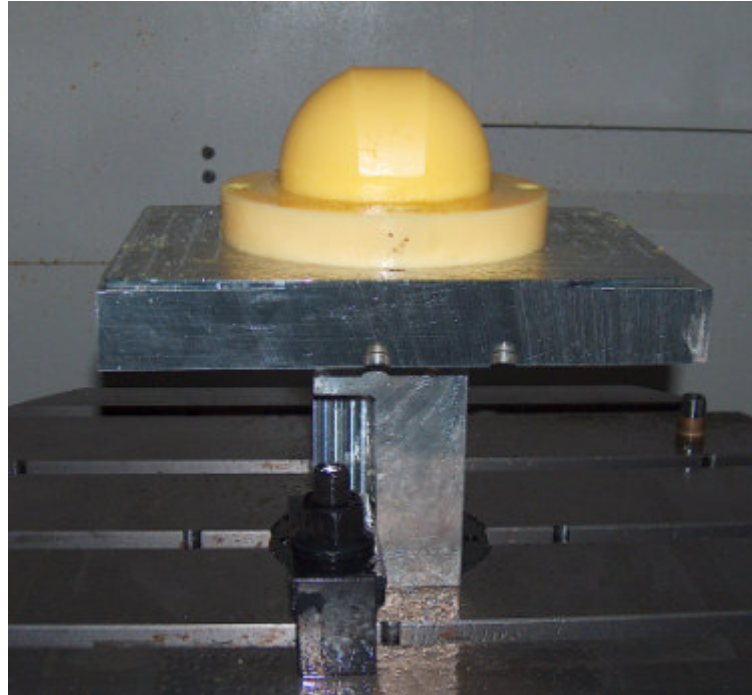


Figure 6.5 Machining setup (View-3)

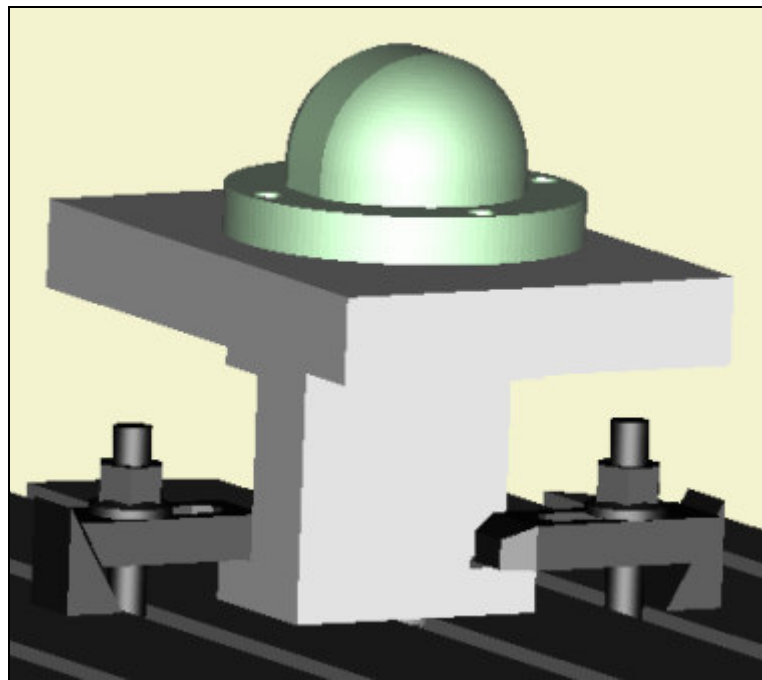


Figure 6.6 Prepared machining setup in Manus 1.0 (View-1)

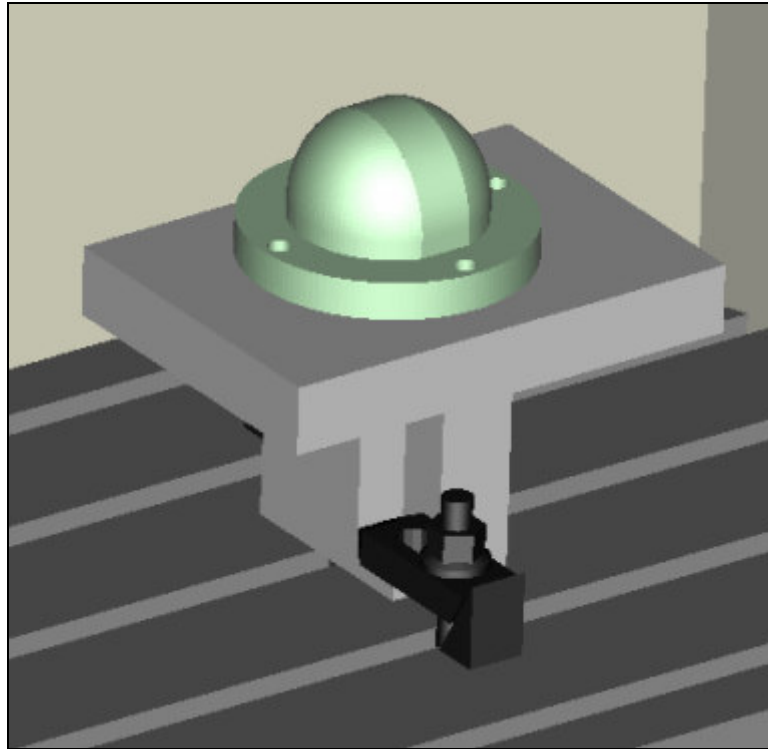


Figure 6.7 Prepared machining setup in Manus 1.0 (View-2)

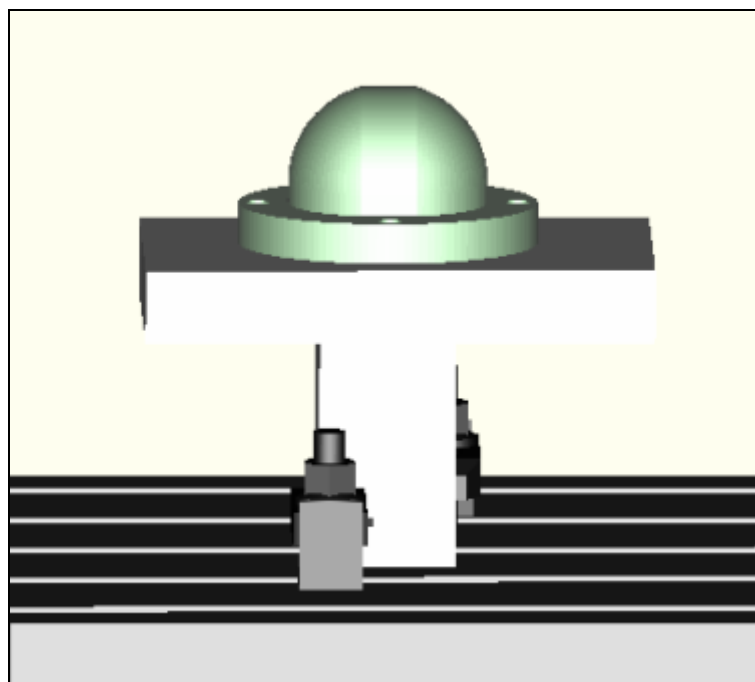


Figure 6.8 Prepared machining setup in Manus 1.0 (View-3)

6.3 Sample Milling Operations

Five sample milling operations are performed as a case study. In the following subsections, these operations will be described.

6.3.1 Application-1

Manus 1.0 postprocesses, simulates and also verifies conventional three-axis milling operations as well as five-axis simultaneous milling operations. The sample cutting performed in this case study is a three-axis finish milling operation. The cutting parameters of this operation are adjusted as;

- Cutting tool is ball endmill.
- Cutting tool diameter=12 mm
- Cutting tool length=174.337 mm
- Cutting feed=6000 mm/m
- Step over (width of cut)=0.25 mm
- Spindle speed=10000 rpm
- No coolant

The tool path is generated in CAM module of ProEngineer. The sample view from tool path simulation in CAM module of ProEngineer is shown in Figure 6.9. The NC-Code (G-Code) is generated using Manus 1.0 and it has over 37000 lines of NC-Codes, a small portion of this NC-Code is given in Figure 6.10. The milling simulation sample view from Manus 1.0 is given in Figure 6.11.

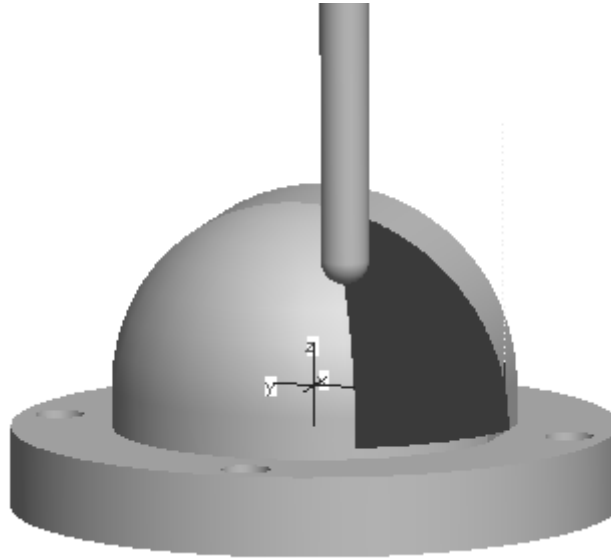


Figure 6.9 Sample simulation view from ProEngineer for “Application-1”

```

O1001
T10M6
S10000M3
G90G54X0Y0
G0X-15.681Y-52.539
Z70.
Z10.389
G1 Z5.389F6000.
X-15.783Y-52.881Z3.511
X-15.867Y-53.162Z1.622
X-15.932Y-53.38Z-.275
X-15.978Y-53.536Z-2.18
X-16.006Y-53.63Z-4.089
X-16.016Y-53.661Z-6.
X-16.35Y-53.56
X-16.34Y-53.525Z-3.99
X-16.308Y-53.422Z-1.983
X-16.256Y-53.25Z.02
X-16.182Y-53.009Z2.014
X-16.088Y-52.699Z3.998
X-15.973Y-52.322Z5.97
X-15.837Y-51.877Z7.926
X-15.681Y-51.366Z9.864
X-15.682Y-50.829Z11.508
X-15.68Y-50.237Z13.14
X-15.877Y-50.868Z11.216
X-16.053Y-51.431Z9.268
X-16.207Y-51.927Z7.301
X-16.341Y-52.354Z5.316
X-16.452Y-52.712Z3.316
X-16.542Y-53.Z1.304

```

Figure 6.10 Sample portion of generated NC-Code for “Application-1”

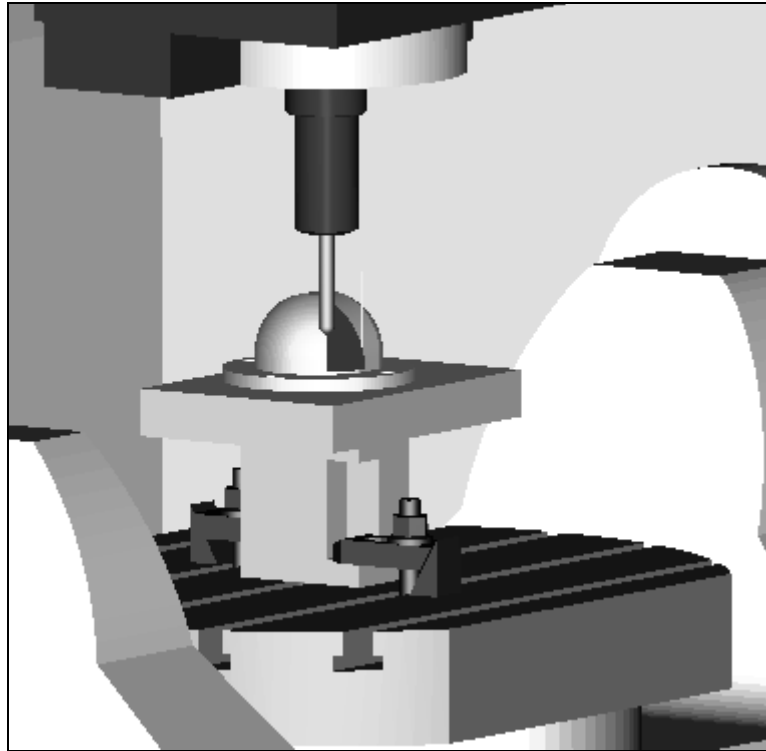


Figure 6.11 Sample simulation view from Manus 1.0 for “Application-1”

6.3.2 Application-2

This sample milling operation is a simultaneous five-axis finish milling operation performed on the right curved surface of the workpiece. The cutting parameters of this operation are adjusted as;

- Cutting tool is ball endmill.
- Cutting tool diameter=12 mm
- Cutting tool length=174.337 mm
- Cutting feed=6000 mm/m
- Step over (width of cut)=1 mm
- Spindle speed=10000 rpm
- No coolant

The sample view from tool path simulation in CAM module of ProEngineer is shown in Figure 6.12. The NC-Code (G-Code) is generated using Manus 1.0. A small portion of this NC-Code is given in Figure 6.13. The milling simulation sample view from Manus 1.0 is given in Figure 6.14.

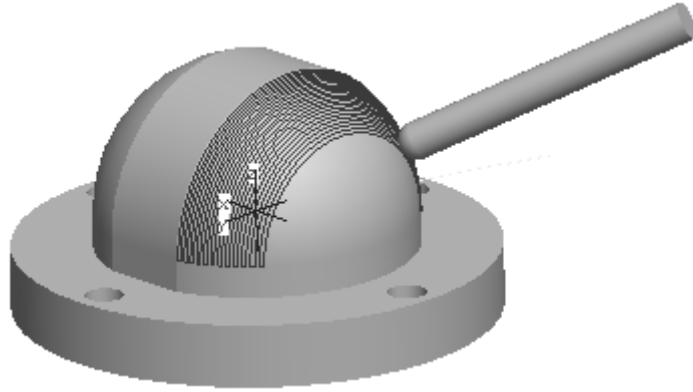


Figure 6.12 Sample simulation view from ProEngineer for “Application-2”

```

O1002
T10M6
S10000M3
G90G54X0Y0
M43
M46
G0X.069Y-131.319
C162.897A-90.
Z-31.285
Z-76.285
G1 Z-81.285F6000.
X.07Y-131.464Z-76.695C162.887A-87.899
X.074Y-131.44Z-72.105C162.852A-85.798
X.081Y-131.248Z-67.521C162.791A-83.699
X.09Y-130.889Z-62.949C162.706A-81.601
X.102Y-130.361Z-58.384C162.596A-79.499
X.117Y-129.665Z-53.838C162.461A-77.395
X.134Y-128.805Z-49.334C162.3A-75.298
X.155Y-127.79Z-44.899C162.111A-73.217
X.179Y-126.617Z-40.515C161.892A-71.142
X.206Y-125.29Z-36.185C161.645A-69.072
X.235Y-123.829Z-31.967C161.371A-67.031
X.268Y-122.262Z-27.914C161.072A-65.045
X.303Y-120.604Z-24.024C160.746A-63.113
X.341Y-118.856Z-20.275C160.394A-61.224
X.382Y-117.039Z-16.688C160.017A-59.387
X.425Y-115.175Z-13.28C159.619A-57.613
X.47Y-113.281Z-10.055C159.199A-55.906

```

Figure 6.13 Sample portion of generated NC-Code for “Application-2”

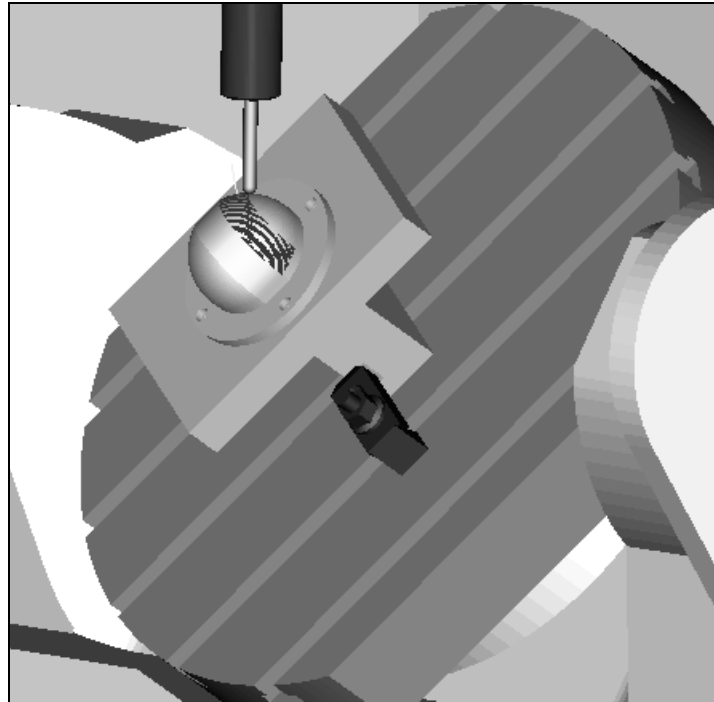


Figure 6.14 Sample simulation view from Manus 1.0 for “Application-2”

6.3.3 Application-3

This sample milling operation is a simultaneous five-axis finish milling operation performed on the left curved surface of the workpiece with a different cutting strategy from the previous sample. The cutting parameters of this operation are adjusted as;

- Cutting tool is ball endmill.
- Cutting tool diameter=12 mm
- Cutting tool length=174.337 mm
- Cutting feed=4000 mm/m
- Step over (width of cut)=0.5 mm
- Spindle speed=8000 rpm
- No coolant

The sample view from tool path simulation in CAM module of ProEngineer is shown in Figure 6.15. The NC-Code (G-Code) is generated using Manus 1.0. A small portion of this NC-Code is given in Figure 6.16. The milling simulation sample view from Manus 1.0 is given in Figure 6.17.

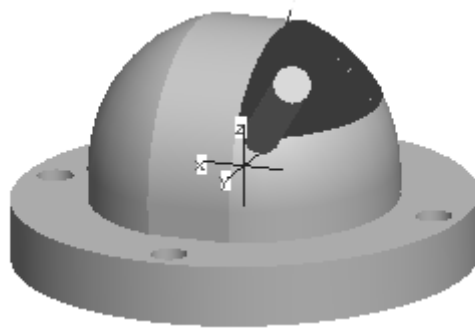


Figure 6.15 Sample simulation view from ProEngineer for “Application-3”

```
O1003
T10M6
S8000M3
G90G54X0Y0
M43
M46
G0X4.918Y-44.835
C104.754A-16.832
Z93.931
Z48.931
G1 Z43.931F4000.
X5.065Y-44.471Z44.041C101.065
X5.187Y-44.099Z44.153C97.376
X5.284Y-43.721Z44.268C93.687
X5.355Y-43.337Z44.384C89.998
X5.4Y-42.95Z44.501C86.309
X5.419Y-42.56Z44.619C82.62
X5.412Y-42.171Z44.737C78.931
X5.379Y-41.782Z44.855C75.242
X5.274Y-42.371Z44.684C69.411A-17.403
X5.351Y-42.771Z44.559C73.272
X5.4Y-43.175Z44.432C77.132
X5.42Y-43.581Z44.305C80.993
X5.411Y-43.988Z44.177C84.854
```

Figure 6.16 Sample portion of generated NC-Code for “Application-3”

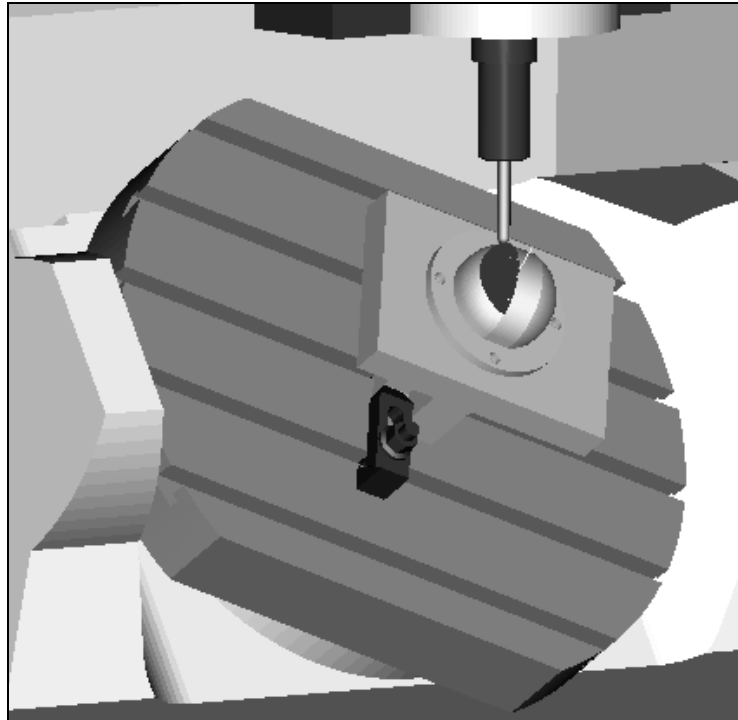


Figure 6.17 Sample simulation view from Manus 1.0 for “Application-3”

6.3.4 Application-4

This sample milling operation is a simultaneous five-axis finish milling operation performed on the middle curved surface of the workpiece. The cutting parameters of this operation are adjusted as;

- Cutting tool is ball endmill.
- Cutting tool diameter=12 mm
- Cutting tool length=174.337 mm
- Cutting feed=4000 mm/m
- Step over (width of cut)=0.5 mm
- Spindle speed=8000 rpm
- No coolant

The sample view from tool path simulation in CAM module of ProEngineer is shown in Figure 6.18. The NC-Code (G-Code) is generated using Manus 1.0. A small portion of this NC-Code is given in Figure 6.19. The milling simulation sample view from Manus 1.0 is given in Figure 6.20.

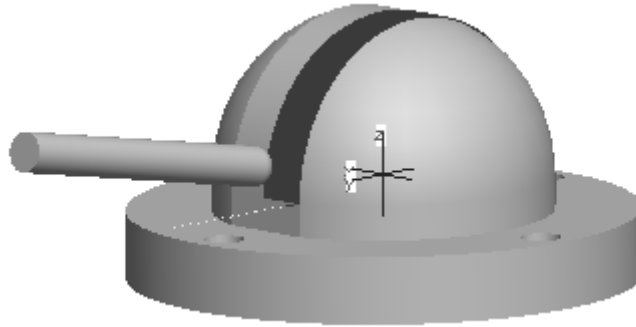


Figure 6.18 Sample simulation view from ProEngineer for “Application-4”

```
O1004
T10M6
S8000M3
G90G54X0Y0
M43
M46
G0X-14.Y-131.319
C0A-90.
Z-19.758
Z-65.773
G1 Z-70.773F4000.
Y-130.997Z-66.102A-87.857
Y-130.501Z-61.446A-85.714
Y-129.83Z-56.812A-83.571
Y-128.987Z-52.206A-81.429
Y-127.973Z-47.635A-79.286
Y-126.788Z-43.105A-77.143
Y-125.435Z-38.622A-75.
Y-123.915Z-34.194A-72.857
Y-122.23Z-29.825A-70.714
Y-120.383Z-25.522A-68.571
Y-118.377Z-21.292A-66.429
Y-116.214Z-17.139A-64.286
Y-113.897Z-13.07A-62.143
Y-111.43Z-9.09A-60.
Y-108.815Z-5.206A-57.857
Y-106.057Z-1.422A-55.714
```

Figure 6.19 Sample portion of generated NC-Code for “Application-4”

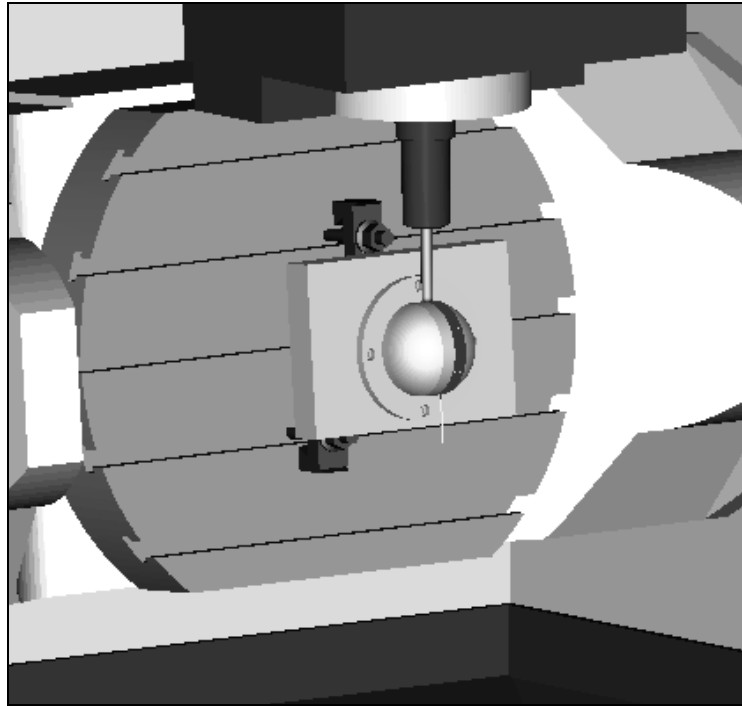


Figure 6.20 Sample simulation view from Manus 1.0 for “Application-4”

6.3.5 Application-5

This sample milling operation is a simultaneous five-axis finish milling operation performed on the middle curved surface of the workpiece with a different cutting strategy from the previous application. The cutting tool used in this application is a flat endmill tool. The step over is increased to 2.5 mm. The cutting parameters of this operation are adjusted as;

- Cutting tool is flat endmill.
- Cutting tool diameter=12 mm
- Cutting tool length=201.674 mm
- Cutting feed=3000 mm/m
- Step over (width of cut)=2.5 mm
- Spindle speed=5000 rpm

The sample view from tool path simulation in CAM module of ProEngineer is shown in Figure 6.21. The NC-Code (G-Code) is generated using Manus 1.0. A small portion of this NC-Code is given in Figure 6.22. The milling simulation sample view from Manus 1.0 is given in Figure 6.23.

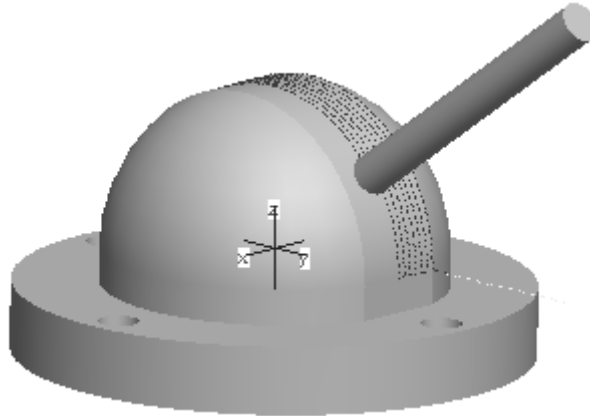


Figure 6.21 Sample simulation view from ProEngineer for “Application-5”

```
O1005
T14M6
S5000M3
G90G54X0Y0
M43
M46
G0X-14.Y-131.319
C0A-90.
Z-19.758
Z-65.773
G1 Z-70.773F3000.
Y-130.997Z-66.102A-87.857
Y-130.501Z-61.446A-85.714
Y-129.83Z-56.812A-83.571
Y-128.987Z-52.206A-81.429
Y-127.973Z-47.635A-79.286
Y-126.788Z-43.105A-77.143
Y-125.435Z-38.622A-75.
Y-123.915Z-34.194A-72.857
Y-122.23Z-29.825A-70.714
Y-120.383Z-25.522A-68.571
Y-118.377Z-21.292A-66.429
Y-116.214Z-17.139A-64.286
Y-113.897Z-13.07A-62.143
Y-111.43Z-9.09A-60.
Y-108.815Z-5.206A-57.857
Y-106.057Z-1.422A-55.714
```

Figure 6.22 Sample portion of generated NC-Code for “Application-5”

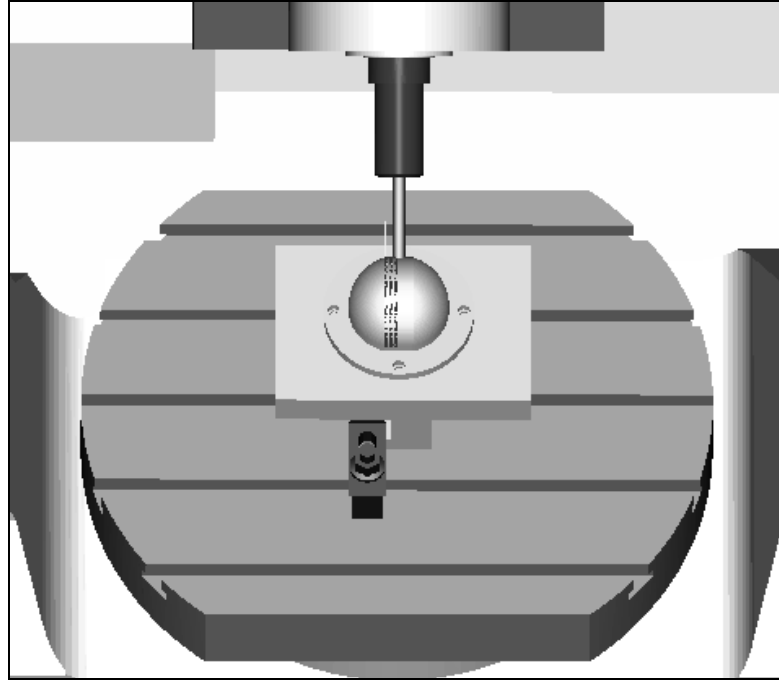


Figure 6.23 Sample simulation view from Manus 1.0 for “Application-5”

6.4 Discussion on Machining Results

The collision check algorithms were activated for these five different applications in Manus 1.0. No gouges or collisions were detected between the cutting tools and the machine tool or setup components. The collision check approach tolerances were defined for both fixtures and clamping elements as 10 mm and the software gave no collision warnings for these applications. All these sample applications were recorded via video camera to compare the real motions of the machine tool components with the simulation display results. These checks were performed to find if the software simulated the machining operations properly or not. When comparing these video displays with simulation displays, it was observed that machining simulations in Manus 1.0 are correctly performed for these applications.

The measurements were performed to compare the machined workpiece with the desired model by using the optic scanner located in METU-BILTIR Center. By performing these measurements, it was observed that the workpiece

was produced in correct dimensions and no gouges were detected, as shown in Figure 6.24. It was observed excessive material left on the edges where the three curved surfaces have met. This is an expected result, since it is usually observed in the generated tool paths that the cutting tool does not fully approach the ends of the cut surfaces (i.e. the boundary edges of the surface). Therefore scallops formed on these edges. The maximum excess material remained on the machined surface is approximately 200 μm at these edges.

The elapsed time while performing postprocessing process and during collision check process were calculated for “Application-4”. The postprocessing time was found as 43 seconds. The collision check process time depends on the length of the generated NC-Code and also depends on the number of the triangular facets that form the surfaces of the displayed models in Manus 1.0. The number of triangular facets forming workpiece model is 5489. The number of triangular facets for table is 1476. The total number of triangular facets for the fixture components and the clamping elements used in the setup is 1371. The total number of triangular facets is therefore found as 8336. The number of NC code lines for “Application-4” is 1634. The computer which the Manus 1.0 software is activated has the following features:

- HP workstation x4000
- Intel Xeon processor
- 1 GB RAM

The collision check process time is calculated as 1100 seconds (18.3 minutes) for this particular application.

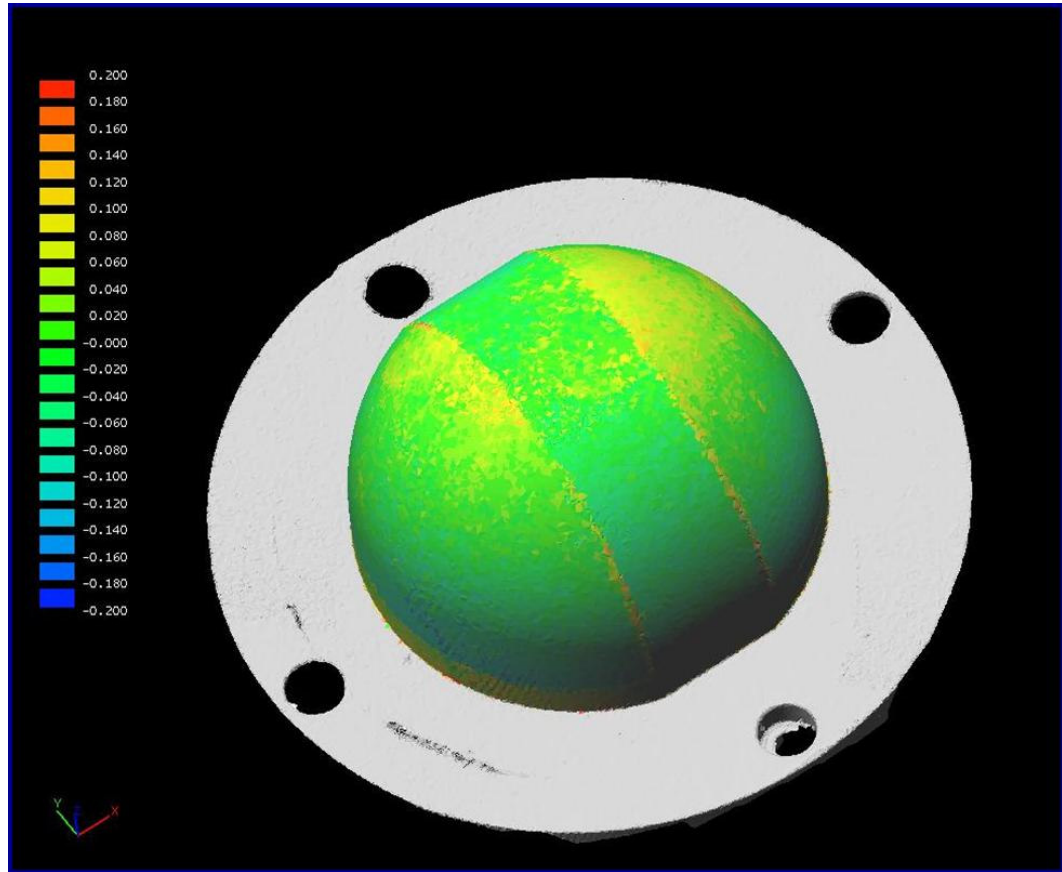


Figure 6.24 Optic scanner result (scale is in millimeter)

6.5 Closure

In this chapter, the sample applications that were performed in “Mazak Variaxis 630-5X” were described. Five different applications were done. In all these applications, the Manus 1.0 software was used. These applications were recorded via video camera in order to compare the real machining views with the simulation display results. The machined workpiece was also analyzed and measured by using the optic scanner located in METU-BILTIR Center.

CHAPTER 7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The aim of this study is to develop a software named “**Manus 1.0**”, which is capable of postprocessing the CL-Data files, simulating and verifying milling operations for a particular five-axis CNC machine tool. The software is developed for the high speed “Mazak Variaxis 630-5X” five-axis machine tool located in METU-BILTIR Center. Prior to the usage of this developed software, this machine tool could not be efficiently used due to the absence of a proper postprocessor and verification software. The absence of this software caused unpredictable gouges and collisions to occur during machining operations. This software has been developed to overcome these difficulties. The software is developed in Borland C++ Builder5.0. Over 29000 lines of codes have been written and 14 forms have been prepared to form the “Manus 1.0” software.

One of the features of the software is the software postprocessing part. The major problem solved in Manus 1.0 is the phase reversing problem encountered in G-Post module [20] of ProEngineer. In G-Post postprocessor, the retracting motion of the cutting tool during “phase reverse process” does not take place. This causes deformation of the workpiece, breakage of the cutting tool or even destruction of the machining setup components to occur in “phase reverse process”. In Manus 1.0 software, this problem has been solved by an algorithm developed and embedded in the software.

The second major advantage of Postprocessing Part of Manus 1.0 software is the WCS settings part. The WCS is not taken into consideration in G-Post. Therefore while creating the tool path in CAM module, user always has to show the position of the machine table center. If Manus 1.0 software is utilized for

postprocessing, defining the table center location is not needed to be performed during the tool path generation process, since in Manus 1.0; user is allowed to define the WCS location.

In some cases, it is observed that erroneous NC-Code lines occur in G-Post postprocessor. These faulty NC-Codes may give great harms to workpiece, cutting tool, setup components or even machine tool components like table and spindle head. The postprocessing part of Manus 1.0 software also solves the problems related with G-Post module faults.

The simulation and verification parts of the software have many features for properly simulating and verifying the milling operations. Simulation and tool path verification capabilities of Manus 1.0 is far beyond the simulation capabilities of ProEngineer, Unigraphics and Catia that are previously used for tool path verification in METU BILTIR Center. The exact model of the “Mazak Variaxis 630-5X” machine tool is prepared in Manus 1.0 including full machining kinematics. This allows detection of any possible gouges or collisions that can occur during machining. The tool path is also displayed during simulation operation. In NC simulation module of ProEngineer, dynamic zooming, translating and rotating features are not available. This makes it difficult to track the tool path during simulation. In Manus 1.0, it is possible to dynamically zoom, translate and rotate the view vector during simulation.

In addition to the visual inspection capabilities, algorithms have been developed for detecting gouges and collisions of cutting tool with workpiece, fixtures, clamping elements and machine tool components. This collision detection feature of Manus 1.0 provides safer machining operations to be performed. In addition to that collision detection capability, it is developed a new algorithm that is only available in Manus 1.0. This algorithm is responsible from warning possible collisions that can occur due to misplaced setup components. This is a new concept that is first named in this thesis study. In real case applications, it is time-consuming and very complicated to place the fixture and clamping elements in exact locations and these setup components are always placed in approximate locations. These misplacements can cause unpredictable

collisions between cutting tool and setup components. In Manus 1.0, it is possible to define an approach tolerance between cutting tool and setup components. The software warns the user when the cutting tool approaches these components closer than the defined tolerance. This allows the user to recognize any possible collision risks and solve the problem of misplaced components.

The developed program has been tested in sample three-axis and five-axis milling operations as described in Chapter 6 and satisfactory results have been achieved.

7.1.1 Capabilities of the Developed Software

The capabilities of the developed software can be summarized as follows:

1. The software can read and process the CL-Data generated by CAM module of ProEngineer.
2. The software can generate NC-Codes (G-Codes) for three and five-axis milling processes.
3. The user can save the generated NC-Code into a desired file in computer memory.
4. The software allows the user to select and use the workpiece CAD models that are prepared in any CAD programs. Any workpiece models that are converted to STL ASCII format can be perceived by the Manus 1.0 software.
5. The necessary machine tool components like table, table support, spindle head, machine inner walls and operating door are predefined in the Manus 1.0 simulation environment.
6. The tooling information can be defined in Manus 1.0.
7. The Manus 1.0 software has a library for the tool holders that are currently used in METU-BILTIR Center. The user selects one of these predefined tool holders from this software library.

8. The fixtures that are used in the machining setup can be defined and placed in Manus 1.0 simulation environment.
9. The Manus 1.0 software has a library for the clamping elements that are currently used in METU-BILTIR Center. The user can select the required clamping elements from this software library and place them in desired locations using special software tools.
10. Simple geometric elements like cylindrical or prismatic objects can be modeled in Manus 1.0 without using any CAD software.
11. The simulation play controls are available in Manus 1.0 software.
12. Dynamic zooming, rotating and translating of the view vector can be performed during the machining simulation.
13. The setup and machine tool component view settings can be adjusted as opaque, translucent or invisible in Manus 1.0 during simulation.
14. The tool path is automatically displayed during the simulation. The cutting motions are displayed in red color; the rapid motions are displayed in yellow color.
15. The collision checking algorithm is developed for the cutting tools. The Manus 1.0 software can detect the collisions of the cutting tool with any setup or machine tool components.
16. The collisions that are detected are reported on a report window.
17. The collisions are also displayed during simulation by changing the color of the collided machine setup or structural component.
18. In Manus 1.0 software, it is also developed an algorithm for the not exactly positioned fixtures and clamping elements. The software detects the risky approaches to that misplaced fixtures and clamping elements and warns the user.

7.1.2 Limitations of the Developed Software

The limitations of the developed software can be summarized as follows:

1. The software can read and process only the CL-Data generated by CAM module of ProEngineer. The CL-Data files that are generated by other CAM programs can not be used in Manus1.0.
2. The software can generate NC-Codes (G-Codes) for three and five-axis milling processes. The drilling operations can not be postprocessed and simulated in Manus 1.0.
3. The models, which are desired to be displayed in the simulation environment of Manus 1.0, have to be converted to STL ASCII format. The other model transfer formats can not be used in Manus 1.0.
4. Only the G-Codes that are postprocessed in Manus 1.0 can be simulated and verified in the software.
5. The collision detection algorithms that are developed in Manus 1.0 give approximate results depending on the surface quality of the generated STL files, since the STL formatted model is a triangular approximation of the CAD model. If low quality STL files are used in the collision detection process, some possible gouges and collisions can not be detected.
6. In the STL model, the triangular facets generated on planar surfaces can be considerably large. This causes undetectable gouges and collisions since the collision detection process in Manus1.0 is performed by considering only the edges of the triangular facets. The region inside these edges are not taken into consideration in collision detection process in Manus 1.0.
7. The tool path verification part of the software is developed for detecting collisions and gouges of the cutting tool with the workpiece, machine table, fixtures and clamping elements. The tool holder and spindle head collision detection algorithms are not currently present in this study.

8. The collision detection process in Manus 1.0 takes into account all triangular facets forming the STL model and in most of the cases the number of these facets may exceed even 10000. In the developed collision detection algorithm in Manus 1.0, the quick facet eliminations for not risky triangular facets are not present. This considerably slows down the collision detection process.
9. In Manus 1.0 software, collisions and gouges can be recognized via visual inspection and collision detection algorithms, however no solutions are recommended for preventing these detected collisions and also no optimizations are performed on the cutting tool path to avoid detected gouges.
10. The Manus 1.0 software tool library allows the user to define two different types of cutting tools. These are flat endmill and ball endmill tools. Other cutting tools can not be defined in Manus 1.0.

7.2 Future Work

In order to increase the capabilities of Manus 1.0 software, following capabilities should be added:

1. For future development, an interaction of the Manus 1.0 with some of the commonly used CAD/CAM software packages (ProEngineer, Unigraphics and Catia) may be prepared. By this way, the postprocessing, simulation and tool path verification processes can be performed in the same environment with tool path generation process.
2. Manus 1.0 can currently recognize CL-Data files prepared by ProEngineer. Perceiving capability of any other CL-Data files that are prepared by other CAD/CAM programs like Unigraphics and Catia can be added to the software to extend the capabilities of the Manus 1.0.
3. Only the G-Codes that are postprocessed in Manus 1.0 can be simulated and verified in the software. G-Code processing capability can be added

to the software, to allow the users to simulate and verify G-Codes that are prepared by other postprocessors.

4. The model format recognized by Manus 1.0 software is the STL format. The models that are transformed to STL format in CAD software can be used in Manus 1.0. New algorithms can be developed for detecting other model formats. The most preferred model transfer formats currently used in the market are STEP and IGES formats. These can be read and processed by the software if proper algorithms are developed.
5. The collision detection gives approximate results depending on the surface quality (chord height) of the generated STL files, since the STL formatted model is a triangular approximation of the CAD model. The improvements in the STL model can enhance the accuracy of the collision detection.
6. The collision detection process can be accelerated by developing quick facet elimination methods for not risky triangular facets.
7. The collision amount can be calculated and displayed in the software to inform the user. By considering the collision amount, the user can take preventive actions more easily.
8. The clamping elements and tool holders used in the software library are limited with the currently available ones in METU BILTIR Center. This library can be extended by adding new clamping elements and tool holders and also the software can be improved by adding the ability for the creation of user defined clamping elements and tool holders.
9. The tool path verification part of the software is developed for detecting collisions and gouges of the cutting tool with the workpiece, machine table, fixtures and clamping elements. The tool holder and spindle head collision detection algorithms are not currently present in this study. These algorithms can be developed and embedded in the software for complete tool path verification.

10. Algorithms can be developed to optimize cutting tool path when any gouges or collisions detected. It can also be developed specific features to show possible solutions to the user in order to prevent the detected gouges and collisions.
11. The Manus 1.0 software is developed for three and five-axis milling operations. The drilling postprocessing and simulation capability and also drilling tools are not currently available in Manus 1.0 software. These properties can be added to Manus 1.0 software to make it workable for drilling operations.
12. The Manus 1.0 software tool library allows the user to define two different types of cutting tools. These are flat endmill and ball endmill tools. A feature can be added to the software to allow the user to define other special cutting tools.
13. The machining time by considering the cutting and rapid feedrates can be calculated and displayed in the software screen to inform the user. In addition, the elapsed time and remaining time of machining operation can be calculated and displayed in the software screen.
14. This software can be adapted to the other five-axis CNC machine tools.

REFERENCES

- [1] Hwang J.S., “Five-Axis NC machining of Sculptured Surfaces” Ph.D. Thesis, University of Birmingham, United States, August 1997.
- [2] Rajah M.G., “5-Axis Machining”, *web site*: http://www.symbiosisonline.com/oct03_axis.htm, *last accessed*: May 2005
- [3] Hagiz G., “5-axis machining”, *web site*: <http://numeryx.com/cnc/5axes.htm>, *last accessed*: April 2005
- [4] G-Post (ProEngineer NC Postprocessing module) manual
- [5] Jung Y.H., Lee D.W., Kim J.S., Mok H.S., “NC post-processor for 5-axis milling machine of table rotating/tilting type”, *Journal of Materials Processing Technology*, 130–131 (2002) 641–646
- [6] PTC Product Development Company, *web site*: <http://www.ptc.com>, *last accessed*: December 2004
- [7] UGS Product Lifecycle Management (PLM) Solutions, *web site*: <http://www.ugs.com>, *last accessed*: December 2004
- [8] Dassault Systemes, *web site*: <http://www.3ds.com/home>, *last accessed*: December 2004
- [9] Tebis AG Die CAD/CAM Experten, *web site*: <http://www.tebis.com>, *last accessed*: January 2005
- [10] CAD/CAM Software Designed by Pathtrace EdgeCAM, *web site*: <http://www.edgcam.com>, *last accessed*: January 2005
- [11] CAD/CAM Software from Delcam, *web site*: <http://www.delcam.com>, *last accessed*: January 2005

- [12] Lim T.S., Lee C.M., Kim S.W., Lee D.W., “Evaluation of cutter orientations in 5-axis high speed milling of turbine blade”, *Journal of Materials Processing Technology* 130–131 (2002) 401–406
- [13] Baptista R., Antune Simoes J.F., “Three and Five axes milling of sculptured surfaces”, *Journal of Materials Processing Technology* 103 (2000) 398-403
- [14] The Official Generative Machining ASCII CLDATA Output Specification August 9, 2000 Version 8.1.1
- [15] Eskicioğlu H., “Lecture Notes on Numerical Control”, Middle East Technical University, Mechanical Engineering Department, 73-92, Ankara, 1987
- [16] Kurt M., “Nümerik Kontrollü Tezgahlar ve Sistemler”, İTÜ Makina Fakültesi, Asilteknik Yayın, 168-170, İstanbul, 1986
- [17] Software Magic, *web site*: <http://www.softwaremagic.com>, *last accessed*: January 2005
- [18] ICAM Technologies, *web site*: <http://www.icam.com>, *last accessed*: January 2005
- [19] NC Postprocessing Software IMS Software, *web site*: <http://www.ims-software.com>, *last accessed*: January 2005
- [20] Austin NC, *web site*: <http://www.austinnc.com>, *last accessed*: January 2005
- [21] Kruth J.P., Lauwers B., Klewais P., Dejonghe P., “NC-postprocessing and NC-simulation for five-axis milling operations with automatic collision avoidance”, K.U.Leuven, Division PMA, Celestijnenlaan 300 B, B-3001 Leuven, Belgium.

- [22] Munlin M., Makhanov S.S., Bohez E.L.J., "Optimization of rotations of a five-axis milling near stationary points", *Computer-Aided Design* 36 (2004) 1117–1128.
- [23] Takeuchi Y. and Idemura T., "Generation of five-axis control collision-free tool path and post processing for NC-data", *Ann. CIRP* 41(1), 539-542 (1992).
- [24] Vericut Software, *web site*: <http://www.cgtech.com>, *last accessed*: February 2005
- [25] Predator Software, *web site*: <http://www.predator-software.com>, *last accessed*: March 2005
- [26] NCSimul Software, *web site*: <http://www.ncsimul.com>, *last accessed*: February 2005
- [27] MachineWorks Software, *web site*: <http://www.machineworks.com>, *last accessed*: February 2005
- [28] Esprit CAM Software, *web site*: <http://www.dptechnology.com>, *last accessed*: February 2005
- [29] Saito T., Takashi T., "NC Machining with G-buffer method". *Computer Graphics*, Volume 25, Number 4. July, 1991
- [30] Van Hook T., "Real-time shaded NC milling display", *Computer Graphics* Volume 20, Number 4, August, 1986
- [31] Huang Y., Olive J.H., "Integrated Simulation, error assessment, and tool path correction for five-axis NC machining" *Journal of Manufacturing Systems* Volume 14, Number 5, 331-334, 1995
- [32] Ho S., Sarma S., and Adachi Y., "Real-time interference analysis between a tool and an environment", *Computer-Aided Design*, 33(13):935–947, November 2001.

- [33] Bohez E. L. J., Minh N. T. H., Kiatsrithanakorn B., Natasukon P., Ruei-Yun H., and Son L. T.. “The stencil buffer sweep plane algorithm for 5-axis CNC tool path verification”, *Computer Aided Design*, 35(12):1129–1142, October 2003.
- [34] Abdel-Malek K, Seaman W and Yeh H, “NC Verification of up to 5 Axis Machining Process Using Manifold Stratification” *Journal of Manufacturing Science and Engineering*, Vol.122, pages1-11
- [35] Salihler S., “Developed of Simulation and Verification Software For CNC Turning Machines”, M. S. Thesis, Middle East Technical University, Ankara, Turkey, 2001
- [36] Susuzlu T. “NC Verification for Turning Operations”, M. S. Thesis, Middle East Technical University, Ankara, Turkey, 2001
- [37] Ferguson A., “History of Computer Programming Languages”, *web site*: http://www.princeton.edu/~ferguson/adw/programming_languages.shtml, *last accessed*: September 2004
- [38] Borland C++ Builder 6, *web site*: <http://www.borland.com/us/products/cbuilder>, *last accessed*: September 2004
- [39] SGI: OpenGL, *web site*: <http://www.sgi.com/products/software/opengl>, *last accessed*: September 2004
- [40] Operating Manual of “Mazak Variaxis 630-5X” milling machine.
- [41] CNK Software, *web site*: <http://www.5axispostprocessor.com>, *last accessed*: February 2005
- [42] Wah W.H., “Introduction to STL format”, *web site*: http://rpdrc.ic.polyu.edu.hk/old_files/stl_introduction.htm, *last accessed*: November 2004

- [43] Sicam - Rapid Models - Stl File Format,
web site: <http://www.sicam.com/Main/TechInformation/StlFileFormat.asp>,
last accessed: November 2004
- [44] The OpenGL Programming Guide - The Redbook, 4th Edition, Version 1.4
- [45] Funkhouser T., "Ray Casting", *web site:*
<http://www.cs.princeton.edu/courses/archive/fall00/cs426/lectures/raycast/raycast.pdf>, *last accessed:* February 2005

APPENDIX A

TECHNICAL SPECIFICATIONS OF MAZAK VARIAXIS 630-5X

The technical specifications of five-axis “Mazak Variaxis 630-5X” CNC machine tool are given in Table A.1. The descriptions of some of these specifications are shown in Figure A.1 through Figure A.3.

Table A.1 Technical features of Mazak Variaxis 630-5X

ITEM			VARIAXIS 630-5X Standard
Capacity	Max. workpiece dimensions	Diameter	730 mm
		Height	500 mm
	X-axis stroke (saddle right/left movement)		630 mm
	Y-axis stroke (column right/left movement)		765 mm
	Z-axis stroke (spindle head up/down movement)		510 mm
	A-axis (table tilt) travel amount		-120° to +30° (axial control)
	C-axis (table rotation) travel amount		+/-360° (axial control)
	Distance from the spindle nose to the table top face		230 to 740 mm (9.06 to 29.13 in.)
	Table center position wrt machine coordinate system	X-Axis	-315mm (12.40 in.)
		Y-Axis	-315mm (12.40 in.)
		Z-Axis	-740 mm (29.13 in.)
	C-Axis rotation axis position wrt machine coordinate system	X-Axis	-315mm (12.40 in.)
		Y-Axis	-315mm (12.40 in.)
	A-Axis rotation axis position wrt table center	Y-Axis	-0.012 mm (0.000472 in.)
		Z-Axis	49.954 mm (1.97 in.)
	Table loading capacity (evenly distributed)		500 kgf
Spindle	Spindle speed		35 to 25000 rpm
	Number of spindle speed change ranges		1 speed
FeedRate	Rapid traverse rate	X/Y/Z	50 m/min
		A/C	18000°/min
	Maximum cutting feedrate	X/Y/Z	50 m/min
		A/C	9000°/min
	Simultaneously controlled axes		5 axes
	Index positioning time	A	1.1 s/90°
		C	1.1 s/90°
Table	Table dimensions		Ø630 mm x 500 mm wide
	T-slots + reference hole/ Tap + reference hole on the table top face	Nominal dimensions	(ISO R 299) 18 mm
		Number of T-slots	5
		Intervals	100 mm
		Reference hole	φ60 ^{+0.018} ₀ mm
	Height from the machine bottom to table top		950 mm

Table A.1 Technical features of Mazak Variaxis 630-5X (continued)

ATC, tool magazine	Tool storage capacity		30 tools
	Tool selection method		Fixed magazine pocket numbers
	Retention bolt type		Yamazaki ANSI type
	Maximum diameter of tool in the tool magazine	Unconditional	φ80 mm (3.15 in.)
		Without adjacent tools	φ125 mm (4.92 in.)
Accuracy	Maximum tool length (from the reference face of taper)		360 mm (14.17 in.)
	Positioning accuracy (pitch error compensation function used)	X, Y & Z	±2.5 μm
		A	±10 s
		C	±5 s
	Repeatability	X, Y & Z	±0.7 μm
		A	±5 s
		C	±3 s

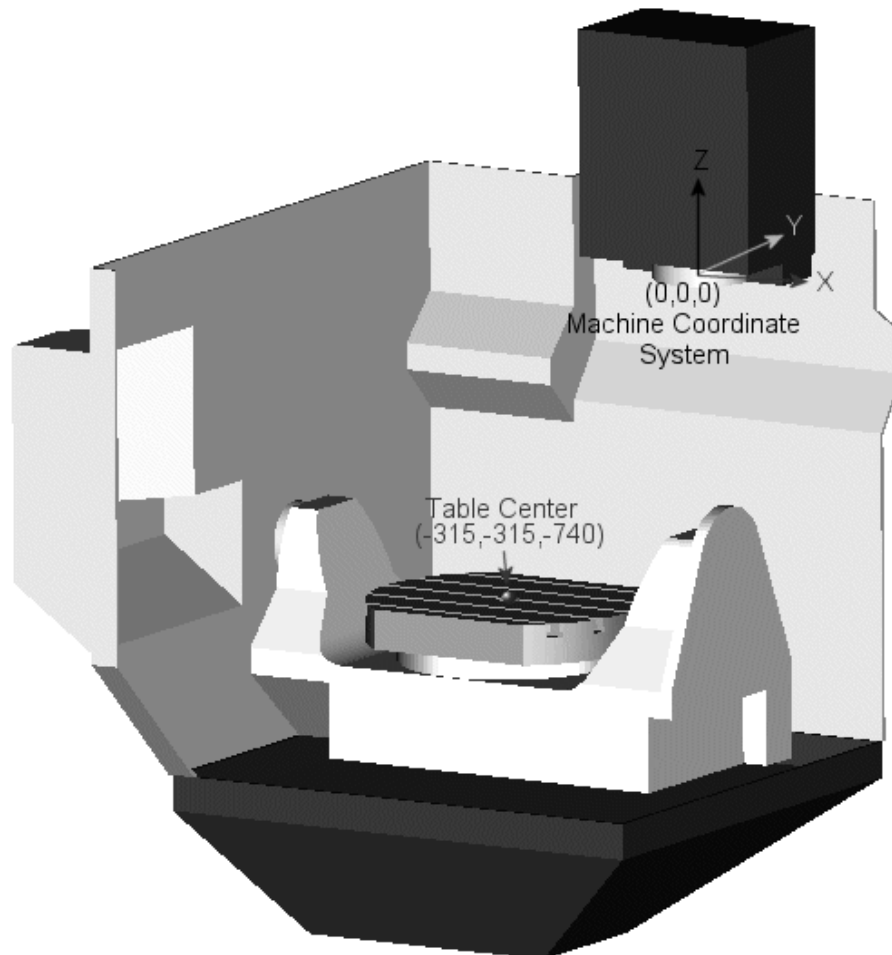


Figure A.1 Machine Coordinate System and table center locations on “Mazak Variaxis 630-5X”

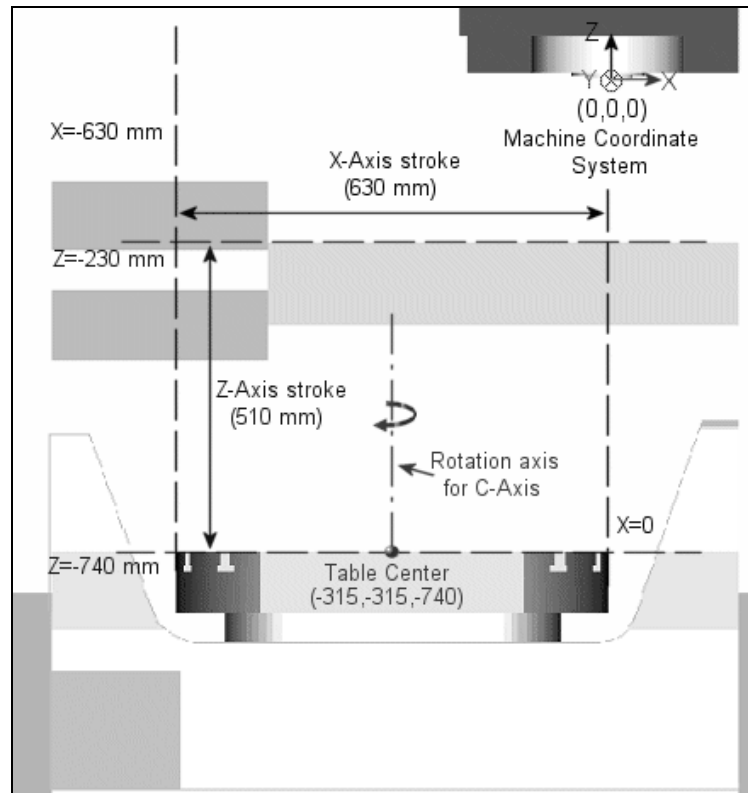


Figure A.2 X-Axis, Z-Axis strokes and rotation axis location of C-Axis on “Mazak Variaxis 630-5X”

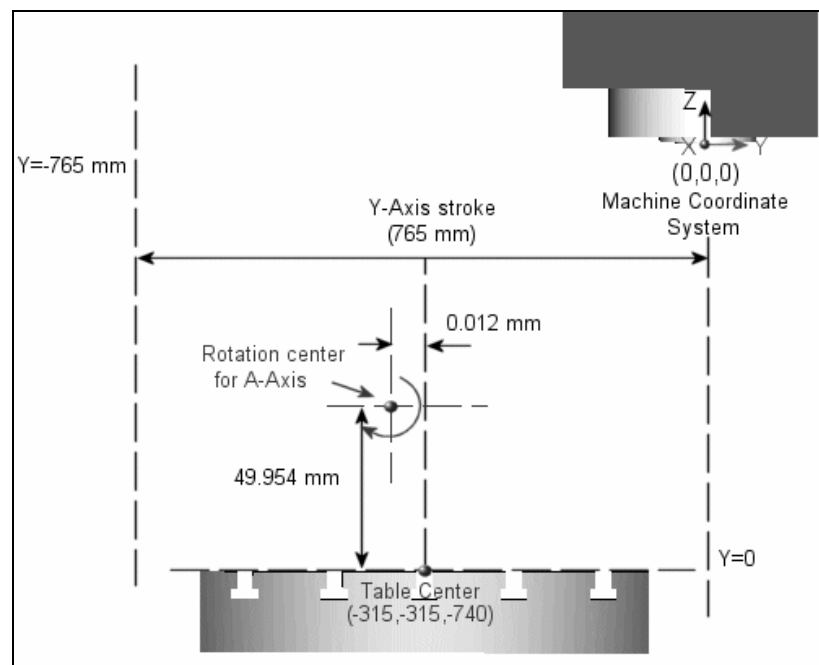


Figure A.3 Y-Axis stroke and rotation axis location of A-Axis on “Mazak Variaxis 630-5X”

APPENDIX B

LIBRARY OF MANUS 1.0 FOR TOOL HOLDERS

In this Appendix, the tool holders that are currently used for machining operations in “Mazak Variaxis 630-5X” machine tool in METU-BILTIR Center are represented. These tool holders are all DIN 69893 standard tool holders that are shown in Figure B.1 through B.14. These tool holders are;

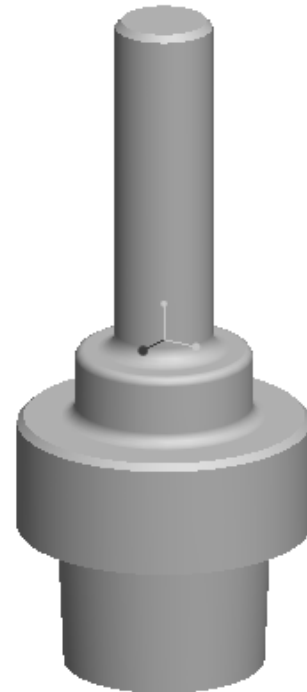
1. HSK A 63 ER16X120M E
2. HSK 63-2/20-160
3. HSK A63 ER16X160 M E
4. HSK A63 ER32X120 E
5. HSK A63 ER32X100 E
6. HSK A63 ER32X80 E
7. HSK A63 EM25X110 E

These tool holders that are given are all defined in Manus 1.0 software. The CAD models are prepared and embedded in Manus 1.0 software. The tool holders, their CAD models and dimensional information are given in this Appendix.

1. HSK A 63 ER16X120M E



(a)



(b)

Figure B.1 “HSK A 63 ER16X120M E” (a) photograph (b) CAD model

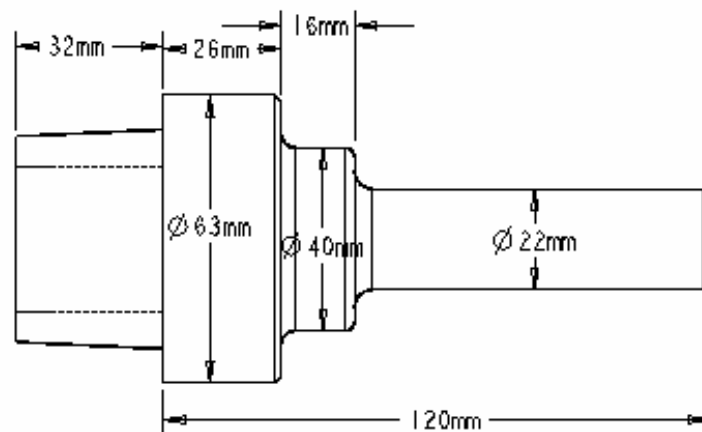
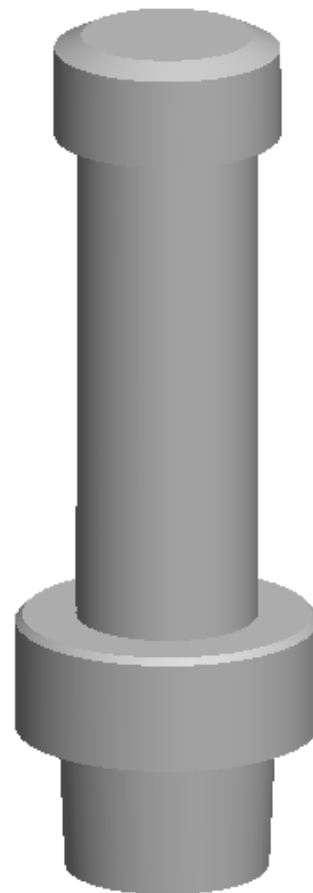


Figure B.2 Dimensions of “HSK A 63 ER16X120M E”

2. HSK 63-2/20-160



(a)



(b)

Figure B.3 “HSK 63-2/20-160” (a) photograph (b) CAD model

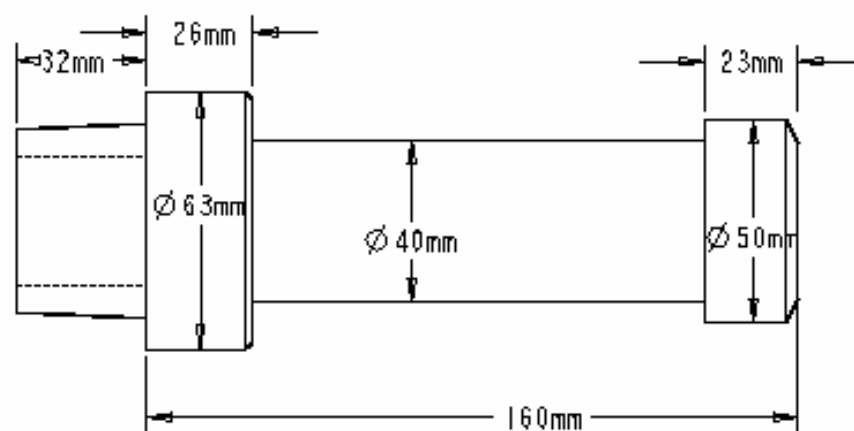


Figure B.4 Dimensions of “HSK 63-2/20-160”

3. HSK A63 ER16X160 M E



(a)



(b)

Figure B.5 “HSK A63 ER16X160 M E” (a) photograph (b) CAD model

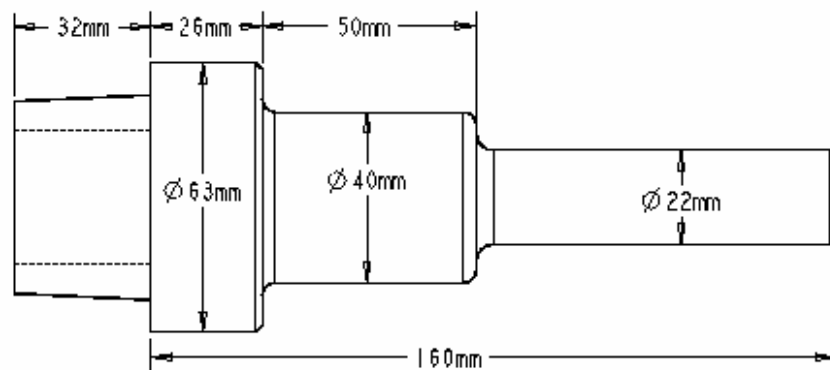
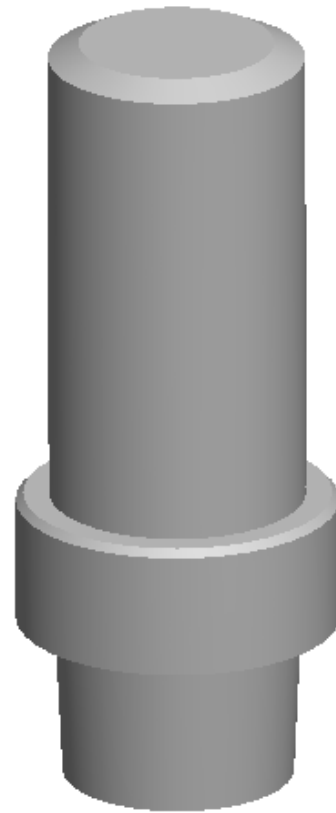


Figure B.6 Dimensions of “HSK A63 ER16X160 M E”

4. HSK A63 ER32X120 E



(a)



(b)

Figure B.7 “HSK A63 ER32X120 E” (a) photograph (b) CAD model

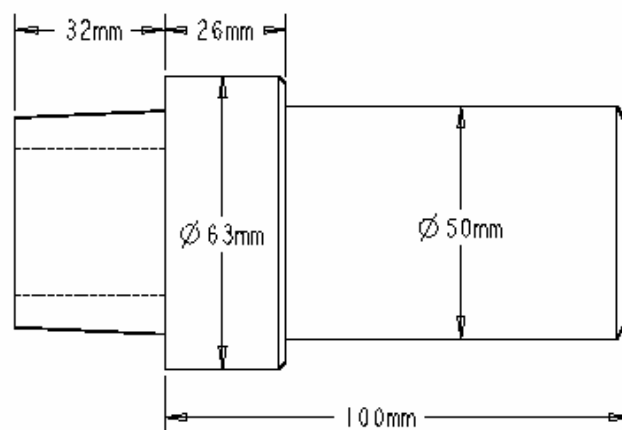
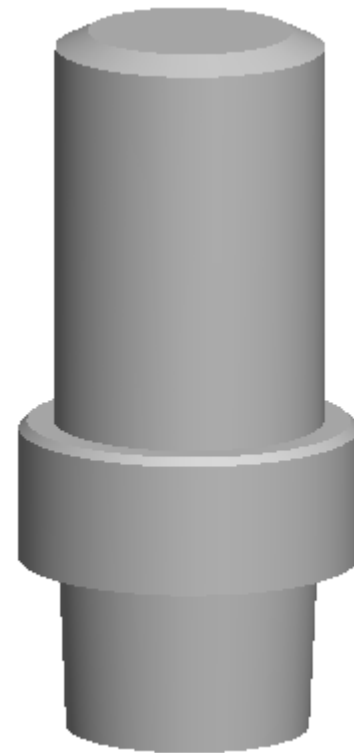


Figure B.8 Dimensions of “HSK A63 ER32X120 E”

5. HSK A63 ER32X100 E



(a)



(b)

Figure B.9 “HSK A63 ER32X100 E” (a) photograph (b) CAD model

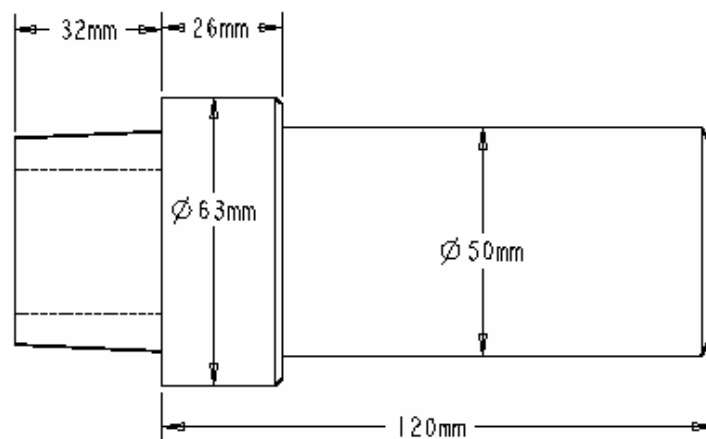
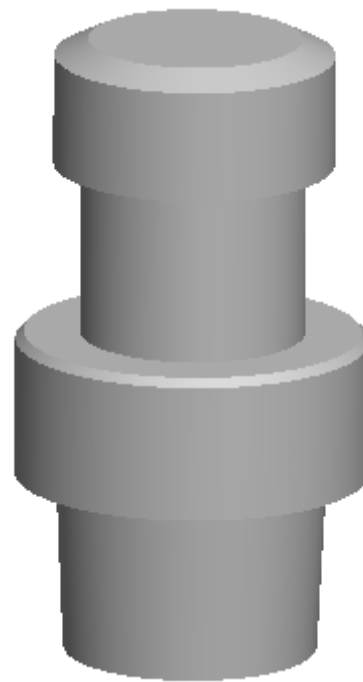


Figure B.10 Dimensions of “HSK A63 ER32X100 E”

6. HSK A63 ER32X80 E



(a)



(b)

Figure B.11 “HSK A63 ER32X80 E” (a) photograph (b) CAD model

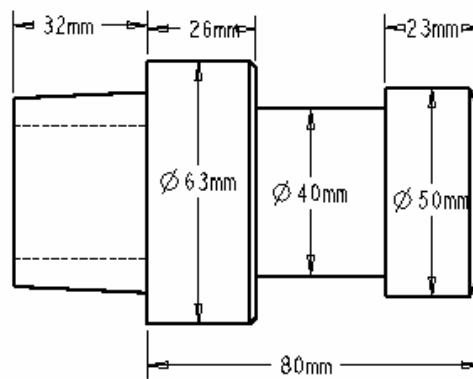
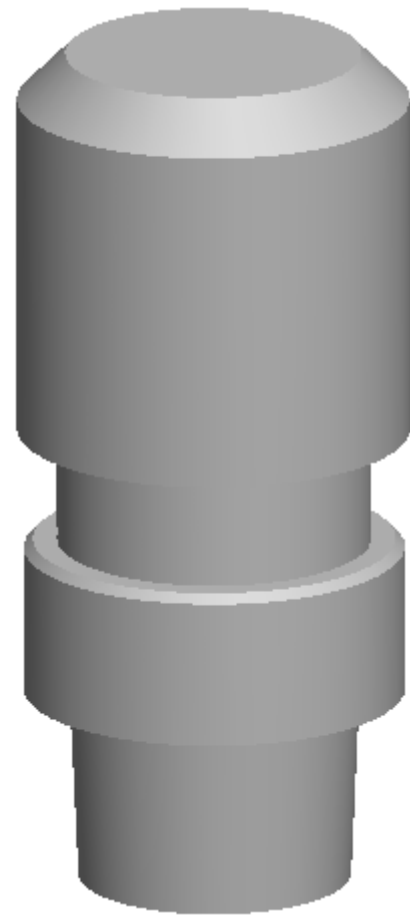


Figure B.12 Dimensions of “HSK A63 ER32X80 E”

7. HSK A63 EM25X110 E



(a)



(b)

Figure B.13 “HSK A63 EM25X110 E” (a) photograph (b) CAD model

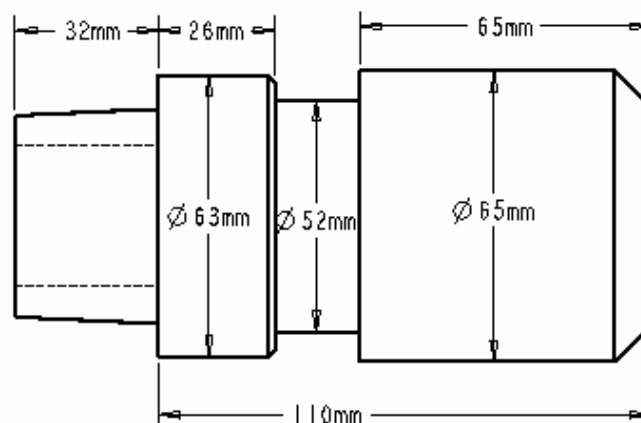


Figure B.14 Dimensions of “HSK A63 EM25X110 E”

APPENDIX C

LIBRARY OF MANUS 1.0 FOR CLAMPING ELEMENTS

The clamping elements that are currently used for machining operations in “Mazak Variaxis 630-5X” machine tool in the METU-BILTIR Center are represented. A clamp set has two major components. The step clamp component and the step block component. These are all in DIN 6314Z (step clamp), DIN 6500 (step block) standards. These components are shown in Figure C.1. The other components of clamping set are; the clamping studs, T nuts and flange nuts. The clamping stud, the flange nut and T-nut are standard clamping parts having M16 (metric16) threads. The clamping stud and the flange nut are also modeled in Manus 1.0. The T-nut is not modeled since it enters to the T-slot of the machine table and becomes completely invisible as shown in Figure C.1. Therefore, T-nut has no effect on milling simulation and collision detection.

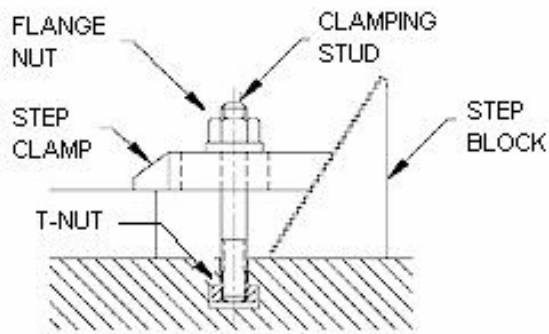


Figure C.1 Clamping element components

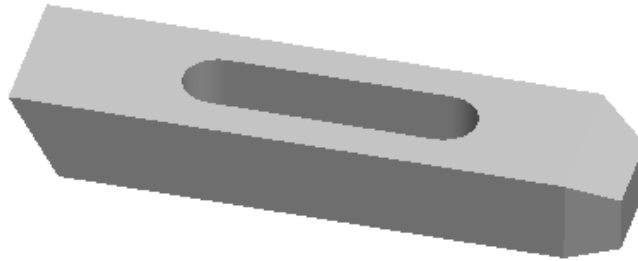
In the clamping set that is currently used in METU-BILTIR Center, there are three types of step clamp and also three types of step blocks.

1. STEP CLAMPS

The step clamps in METU-BILTIR Center are given in Figure C.2 through C.4 and their dimensions are given in Figure C.5.



(a)



(b)

Figure C.2 “Clamp Type-1” (a) photograph (b) CAD Model



(a)

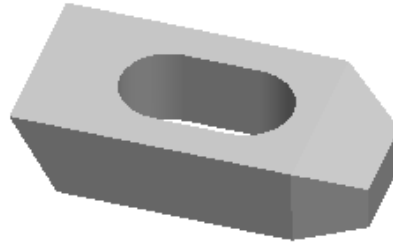


(b)

Figure C.3 “Clamp Type-2” (a) photograph (b) CAD Model

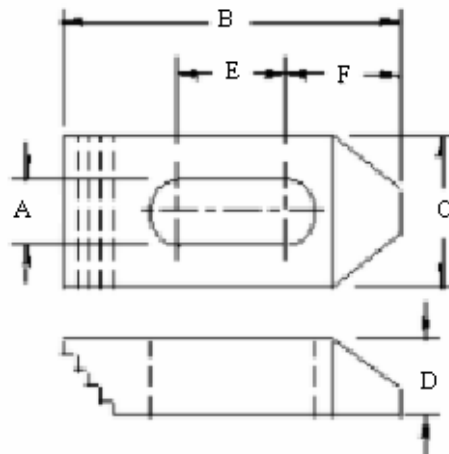


(a)



(b)

Figure C.4 “Clamp Type-3” (a) photograph (b) CAD Model



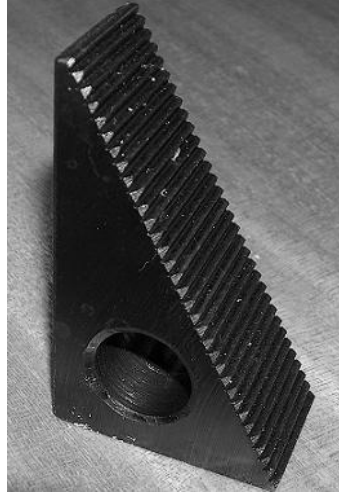
Dimensions are in millimeter

STEP CLAMP TYPES	DIMENSIONS						WEIGHT (LBS/100PCS)
	STUD SIZE (A)	(B)	(C)	(D)	(E)	(F)	
CLAMP TYPE-1	M16	152	38	22	49	34	160
CLAMP TYPE-2	M16	102	38	19	30	31	87
CLAMP TYPE-3	M16	64	32	16	14	25	33

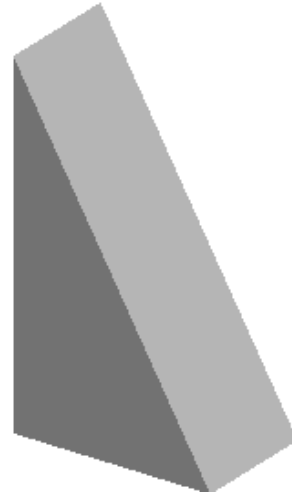
Figure C.5 Dimensions of step clamps in METU-BILTIR Center

1. STEP BLOCKS

The step blocks in METU-BILTIR Center are given in Figure C.6 through C.8 and their dimensions are given in Figure C.9.

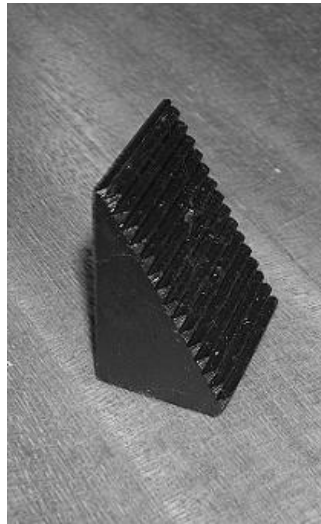


(a)



(b)

Figure C.6 “Block Type-1” (a) photograph (b) CAD Model

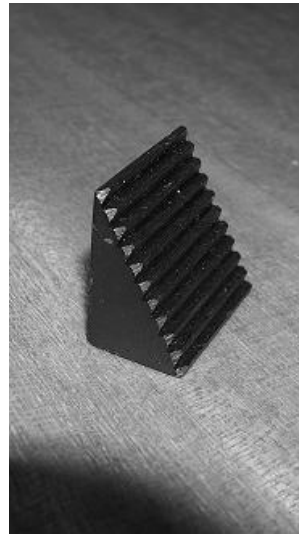


(a)



(b)

Figure C.7 “Block Type-2” (a) photograph (b) CAD Model



(a)



(b)

Figure C.8 “Block Type-3” (a) photograph (b) CAD Model

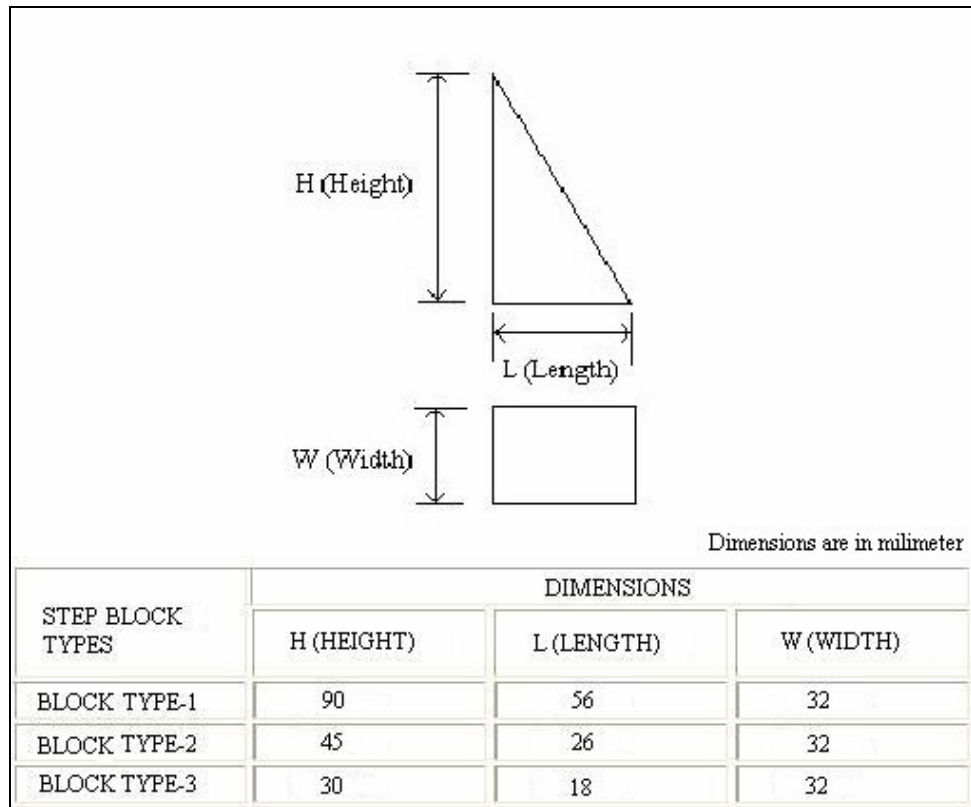


Figure C.9 Dimensions of step blocks in METU-BILTIR Center

APPENDIX D

THE COLLISION CHECK PROCESS

The flowchart of collision check process is shown in Figure D.1. The process contains a general loop (“Loop-1” as shown in Figure D.1). The collision detection process is performed for every CL-Data motion lines. Therefore, the “Loop-1” takes one CL-Data motion line at each step from the beginning to the end of the CL-Data file.

Inside “Loop-1”, there is another loop “Loop-2”. The collision check process is performed for every triangular facet of the desired model. In “Loop-2”, one triangular facet is taken at each step to perform the collision check process. The “Loop-2” has a form as follows;

```
stl_line=1;
while (stl_satir<50000){
collision_detection_process_for_triangle;
stl_line+=1;}
```

The “stl_line” variable is an integer showing the processed triangular facet. It is assumed that there can be maximum 50000 triangular facets in an STL model. If there are less, the process is finished without waiting the stl_satir to become 50000, by additional controls. The “collision_check_process_for_triangle” is performed for every three edges of each triangular facet. Therefore a third loop “Loop-3” is formed inside “Loop-2” which has a form as follows;

```
for (int i=0;i<3;i++){
collision_detection_process_for_edge;}
```

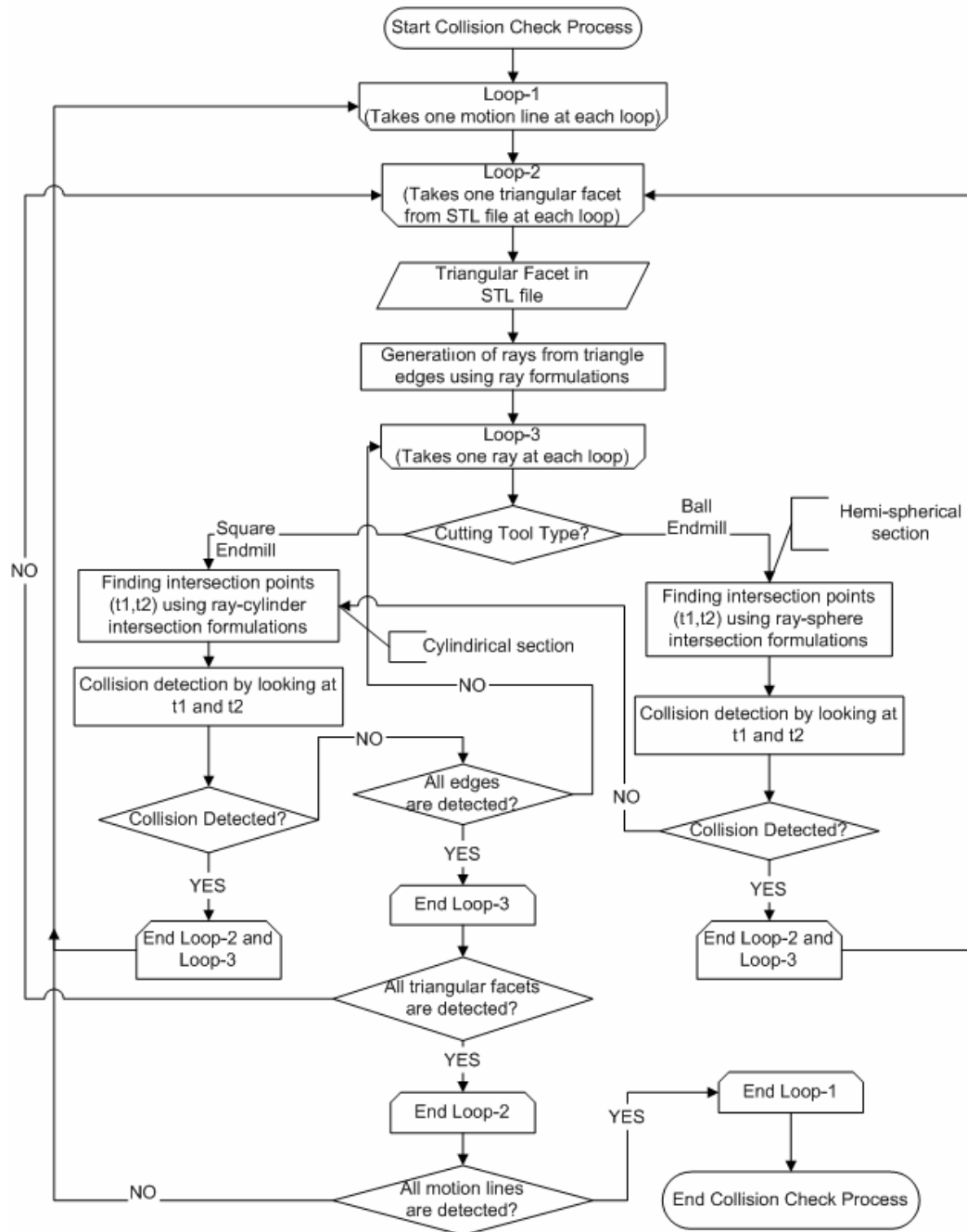


Figure D.1 Flow chart of collision check process

The “collision_detection_process_for_edge” is performed as follows;

1) First the location of the workpiece is found for the C-Axis rotation. This is accomplished by finding the new locations of the triangular facet vertices that are forming the workpiece model.

```
model_x0[i]=Matrix3[0]*(ntri4[stl_satir].vertices[0]+  
WCX+315)+Matrix3[4]*(ntri4[stl_satir].vertices[1]+WCY+  
315);
```

```
model_y0[i]=Matrix3[1]*(ntri4[stl_satir].vertices[0]+W  
CX+315)+Matrix3[5]*(ntri4[stl_satir].vertices[1]+WCY+  
315);
```

```
model_z0[i]=ntri4[stl_satir].vertices[2]+WCZ+740;
```

where “model_x0[i]”, “model_y0[i]” and “model_z0[i]” show the X,Y and Z axes locations of the triangular facet edges just after the C-Axis rotation. “Matrix3” is a four by four matrix (Matrix3[4,4]) that is formed to find the location of the workpiece after the C-Axis rotation is finished. That means the “Matrix3” matrix contains the rotation information for C-Axis rotation. “ntri4[stl_satir].vertices [i] is a struct formed to show the original location of the triangular facet vertices of the workpiece before any rotations.

2) Then the locations of these triangular facet vertices are found for the A-Axis rotation as follows;

```
model_x[i]=model_x0[i];
```

```
model_y[i]=Matrix2[5]*(model_y0[i]+0.012)+Matrix2[9]*  
(model_z0[i]-49.954);
```

```
model_z[i]=Matrix2[6]*(model_y0[i]+0.012)+Matrix2[10]*  
(model_z0[i]-49.954);
```

where “model_x[i]”, “model_y[i]” and “model_z[i]” show the location of the triangular facet edges just after the C-Axis and A-Axis rotations are finished. “Matrix2” is a four by four matrix (Matrix2[4,4]) that is formed to find the location of the workpiece after the A-Axis rotation is finished. That means the “Matrix2” matrix contains the rotation information for A-Axis rotation.

3) After that, the cutting tool position has to be determined. This is performed in a reverse logic. Instead of the cutting tool, the workpiece is moved as follows;

```
model_x[i]=model_x[i]-motion_c.e[2];
model_y[i]=model_y[i]-motion_c.e[3];
```

where “motion_c[2]” and “motion_c[3]” show the location of the cutting tool at X and Y axes. By performing this process, the workpiece new location is determined.

4) After determining the correct location of the workpiece and the cutting tool with respect to each other, the “collision_check_process_for_edge” can be started. First the vertices are classified according to their Z-Values. The vertices that have maximum, medium and minimum Z-Values are found by using the functions of C++;

```
max_z0=max(model_z[0],model_z[1]);
min_z0=min(model_z[0],model_z[1]);
max_z=max(max_z0,model_z[2]);
min_z=min(min_z0,model_z[2]);
if (max_z==model_z[2])
    mid_z=max_z0;
else if (min_z==model_z[2])
    mid_z=min_z0;
else
    mid_z=model_z[2];
```

where “max_z” is the Z-Value of the vertex that has the maximum Z-Value, “mid_z” is the Z-Value of the vertex that has the medium Z-Value and “min_z” is the Z-Value of the vertex that has the minimum Z-Value. The X and Y values of these vertices are also classified by using the same concept.

5) Then it is started to check the triangular edges for collision with the algorithms described in Chapter 5.

6) To increase the calculation performance, first the edge (ray segment), which is formed with the vertices having maximum Z-Value and the medium Z-Value, is checked since it is most probable to find a collision with the edge that has bigger Z values (or in other words with the edges that are closer to cutting tool). If a collision is detected, “Loop-2” and “Loop-3” are ended as shown in Figure D.1. If no collision is detected with the corresponding triangle edge, then the second most critical edge (ray segment), which has formed with the vertices having maximum Z-Value and minimum Z-Value, is checked for collision. If a collision is detected, “Loop-2” and “Loop-3” are ended. If no collision is detected with the corresponding triangle edge, then the third (last) edge (ray segment), which has formed with the vertices having medium Z-Value and minimum Z-Value, is checked for collision. If collision is detected, “Loop-2” and “Loop-3” are ended. If no collision is detected, “Loop-3” is ended and “Loop-2” continues to work until all triangular facets are processed as shown in Figure D.1.

7) “Loop-2” is ended after all triangular facets are controlled for collision,.

8) Similarly after all CL-Data motion lines are processed, “Loop-1” is also ended as shown in Figure D.1. After that, the collision detection process is finished and the results are displayed in the “Collision Results” window automatically.