COMPUTER AIDED PART PROGRAM GENERATION AND SIMULATION OF TURNING PROCESSES

A MASTER'S THESIS

in

Mechanical Engineering
University of Gaziantep

By
Türkay DERELİ

June 1994

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Mazhar Ünsal
Director

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

Assoc. Prof. Dr. Sedat Bayseç
Chairman of the Department

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master Science.

Prof. Dr. İ. Hüseyin FİLİZ
Supervisor

Examining Committee in Charge

Prof. Dr. İ. Hüseyin FİLİZ

Assoc. Prof. Dr. Ali İhsan SÖNMEZ

Assist. Prof. Dr. Cengiz DOĞAN

ABSTRACT

COMPUTER AIDED PART PROGRAM GENERATION AND SIMULATION OF TURNING PROCESSES

DERELİ, Türkay
M. S. in Mechanical Engineering
Supervisor: Prof. Dr. İ. Hüseyin FİLİZ
June 1994, 128 pages

In this study, a computer program is developed for both the automatic generation of CNC part programs and tool paths, and their computer simulation, for lathe operations.

On the basis of the controller unit of a specific machine tool (i.e., Boxford 250B CNC lathe), the part program to proceed to actual production is automatically generated in standard G and M codes by processing a CAD/CAM data base. The generation can be performed by using either absolute or incremental co-ordinate system. Both the chip removal processes and tool paths (the plot of cutter locations) can be simulated for the verification and realisation of the codes proposed by the program. The generated part programs can also be printed, and stored for future use.

The program is written in C++ programming language on an IBM compatible personal computer. The use of the program is illustrated through examples.

Key Words: Computer Simulation Of Turning Operations, and
CNC Part Program Generation.

ÖZET

BİLGİSAYAR DESTEKLİ PARÇA PROGRAMI TÜRETİMİ VE TORNALAMA İŞLEMLERİNİN SİMÜLASYONU

DERELİ, Türkay
Yüksek Lisans Tezi, Makina Mühendisliği Bölümü
Tez yöneticisi: Prof. Dr. İ. Hüseyin FİLİZ
Haziran 1994, 128 Sayfa

Bu çalışmada, torna işlemlerine yönelik parça programlarının ve takım yollarının otomatik olarak türetilmesi, ve bunların bilgisayar ekranı üzerinde simüle edilmesi için bir bilgisayar programı geliştirilmiştir.

Boxford 250B CNC torna tezgahının kontrol ünitesi esas alınarak ve bir BDT/BDÜ veri tabanından faydalanılarak, üretimi gercekleştirecek parça programı otomatik olarak türetilmektedir. Türetme işlemi mutlak ya da artışlı koordinat sistemlerinde gerçekleştirilebilmektedir. Ayrıca, türetilen takım yolu ve parca programları bilgisayarda simüle edilerek kontrol edilebilmektedir. Yine türetilen bu kodlar basılabilmekte, ve gerektiğinde kullanılmak üzere saklanabilmektedir.

Program IBM uyumlu bir kişisel bilgisayarda, C++ programlama dili kullanılarak yazılmıştır. Programın kullanımı değişik örnekler ile gösterilmiştir.

Anahtar Kelimeler: Tornalama Operasyonlarının Bilgisayarda Benzetimi, CNC Parça Programı Türetimi.

ACKNOWLEDGEMENTS

I would like to express my thanks to Prof. Dr. İ. Hüseyin Filiz for his encouragement and suggestions during this research as the supervisor.

I also wish to thank to Mr. Cengiz Kayacan who criticised me throughout the study.

I am thankful to Assist. Prof. Dr. Ali Demirci for his helps in photographing.

Thanks are due to Research Assistants of Mechanical Engineering Department, especially to Mr. Adil Baykasoğlu for his helps in typing the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Item	Description
CAD	Computer Aided Design
CAD / D	Computer Aided Design / Drafting
CIM	Computer Integrated Manufacturing
CAPP	Computer Aided Process Planning
CAM	Computer Aided Manufacturing
NC	Numerical Control
CNC	Computer Numerical Control
FMS	Flexible Manufacturing System
CAI	Computer Aided Inspection
CAT	Computer Aided Testing
ASCII	American Standard Coding for Information Interchange
EIA	Electronic Industries Association
APT	Automatically Programmed Tools
TNR	Tool Nose Radius
MDI	Manual Data Input
LH	Left Hand
RH	Right Hand
TRP	Tool Reference Point
LISP	LISt Processing
PROLOG	PROgramming in LOGic
EOB	End Of Block
VDU	Visual Display Unit

CHAPTER I

INTRODUCTION

The wide spread usage of CAD systems and introduction of CNC machine tools into industry have provided considerable improvements in productivity and quality. In design and manufacturing, the primary objective is to design and produce the components with high quality in the shortest possible time, and at a competitive cost. In order to achieve this goal, the manual intervention must be completely eliminated from all of the design and manufacturing stages by implementing a CIM system which comprises a CAD/CAPP/CAM integration. Each individual system becomes an "island of automation" and is independent from the other without this integration. A typical CAD/CAPP/CAM integration is shown schematically in Figure 1.1.

CAPP (Computer Aided Process Planning) as an interface in the CAD/CAM integration consists basically the determination of processes and parameters required to convert a block into a finished product. The input to a CAPP system is the output of a CAD system which provides the drawing of the part to be produced.

A CAD system is mainly prepared by the user or commercially available package like AutoCAD. After the drawing is pre-processed by a feature-recognition system, the recognised part information is stored in a data file which can then be used as a CAD data base by process planning modules. Similarly, the input to a CNC machine tool is the Part Program that is the set of machining instructions. The output of the process planning modules is a CAD/CAM data base including the resulting process plan which is to be processed to generate part programs. Therefore, the feature-recognition system and CNC Code Generation System are considered as the sub-interfaces between CAD/CAPP and CAPP/CAM, respectively.

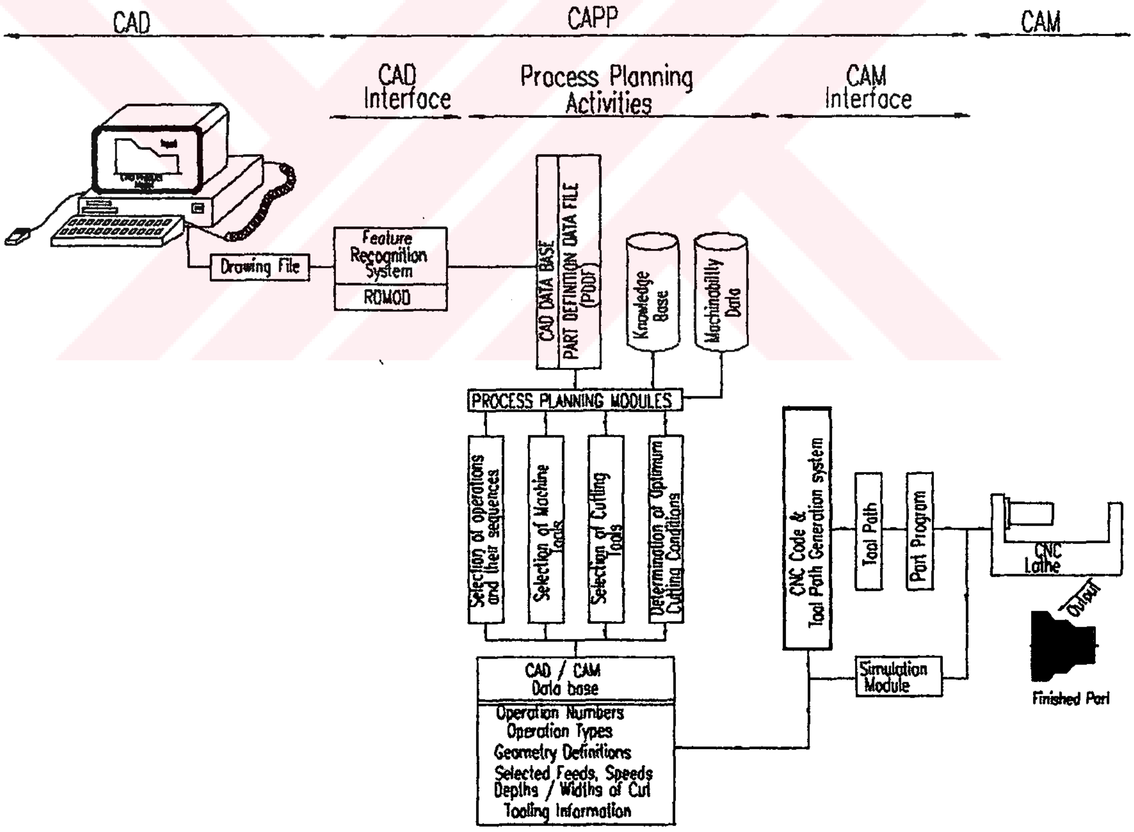


Figure 1.1 The CAD/CAPP/CAM Integration [1].

The ***CNC Part Program / Tool Path Generation*** is also considered as an essential step in finalising the CAD/CAM integration, since the part program is the only link providing the communication between the design and manufacturing stages.

Although the part programs are generated in the integrated environment, they must be verified before the manufacturing stage, since the debugging of CNC part programs on the shop-floor is expensive and dangerous. Simulation is a useful tool for the part programmer working with complex problems. It represents all machining processes and tool movements on the screen. Therefore, possibility of damage to the machine tool can be eliminated by the use of simulation packages prior to machining processes.

The aim of this study is to automate the generation of CNC part programs and tool paths for rotational parts to be manufactured with one set-up in an integrated CAD/CAM system.

The computer program prepared for this purpose takes a previously planned CAD/CAM data base including all machining information and a descriptive drawing data for each workpiece as the input. On the basis of controller unit of Boxford 250 B CNC lathe, it can then generate both the part program and tool path. The resulting part programs can be verified by animating the chip removal processes on the graphics screen. With the tool path display the tool motions could easily be seen. If required, hard copies of generated part programs and lists of tool paths can be printed. The program allows a user to use the software easily by presenting a user-friendly pulldown menu.

Chapter 2 summarises previous works on the *CNC part program generation and verification*. Chapter 3 determines the places of those topics in Computer Aided Process Planning applications as well as giving a brief introduction to Computer Integrated Manufacture. Chapter 4 gives a general review of Computer Numerical Control and part programming. Chapter 5 introduces the CNC words presented on the selected machine tool, and describes the methodology used in this study . Finally, Chapter 6 discusses the capability of the program and the results obtained from this study.

CHAPTER II

LITERATURE SURVEY

2.1 INTRODUCTION

This chapter gives a brief survey of the relevant literature related to the study reported in this thesis. The works done previously on the generation and verification of CNC part programs together with tool paths are reviewed in section 2.2 and 2.3, respectively.

2.2 PREVIOUS WORKS ON AUTOMATIC GENERATION OF CNC PART PROGRAMS

The innovation of numerically controlled machine tools has been the most significant development in manufacturing industry during the past 40 years. Before the advent of Numerical Control, machine tools were controlled by the skill of operators. With the developments of Numerical Control (NC), the optimisation of machining process has been introduced and inaccuracies have been largely eliminated. NC machines became Computer Numerical Control (CNC) machines through the development of inexpensive minicomputers and microprocessors with large memories. CNC became one of the most important production devices in the manufacturing industry. The preparation or development of the program of instructions for a particular part to be machined on these CNC machines is actually the task of the part programmers. However, due to variation in proficiency, the work done by the part programmers is sometimes not satisfactory especially for complex part geometries. Therefore, the automatic generation of CNC part programs without any human intervention has long been a challenge to both academia and industry.

It is very difficult to separate the work in the area of automatic generation of CNC codes from that in a CAPP system. Most of the CAPP systems have converted a cutter location file which contains the tool path generated on a computer assisted part programming language into a part program for a particular machine tool.

Turbo-CAPP [2] which is an intelligent process planning system is able to generate a CNC part program for each planned workpiece based on its geometric characteristics extracted from AutoCAD and the machining parameters stored in the knowledge base. The CNC part programs are generated specifically to run on a variety of CNC turning machines and centres.

TECHTURN (A Technically Oriented System for Turned Parts) generated a CNC part program with little manual intervention by using the outputs from EXCAP (Expert Computer Aided Process Planning). The generated codes have been tested on a MHP Turning Centre [3].

Wang and Lin [4] generated linear and circular tool paths after offsetting the part profile by using the boundary representation of the part from a feature recognition system that is an interface between Computer Aided Design and Computer Aided Process Planning.

Chin [5] developed a feature based knowledge system for CAD/CAM integration in which a cutter location file is generated and stored firstly and then it is converted into an NC code file. The G codes created in this system are for EMCO CNC lathe. However, only external shapes of a cylindrical part are defined and processed in his research.

On the basis of the geometric data and process route generated by processing planning module, ICAPP [6] can generate part programs in higher level languages like APT (Automatically Programmed Tools) and COMPACT II. The programs are then post-processed for a particular controller unit of a machine tool and controller combination.

Jia- Ming Shyu and Y.W Chen [7] have developed a mini Computer Integrated System for turning operations in which both cutting paths and CNC part programs are generated. The tool paths are calculated by using the macro instructions prepared previously in the system. CNC instructions are obtained from cutter location file through post-processors as in the case of previous reference.

Boer, C.R, Petitti, M., and Lombardi, M. [8] developed a CAPP/CAM expert system in which the tool paths translating the information contained in the synchronised solution in an APT-like cutter location file. This cutter location file is then read from a post-processor that generates a part program file in standard G codes used by the numerical control of a specific machine tool.

2.3. PREVIOUS WORKS ON VERIFICATION OF PART PROGRAM

CNC part programs generated either automatically or manually are prone to a variety of errors such as tool-fixture collisions. Therefore, they must be controlled prior to actual machining. There has been considerable research on CNC part program verification. The verification of user-supplied CNC part programs were earlier performed conventionally either by connecting a pencil equipped with a spring instead of cutter or by machining a soft part. In the 70's, the conventional verification methods and manual part programming were still the traditional ways of preparing and checking of the part programs, respectively. These methods were very time consuming and even dangerous to the machine tools. Therefore, the simulation packages being prepared for the verification of part programs became popular in the 80's.

The earlier tool paths were directly plotted. They are later displayed on monitor. Most verification systems now use a computer graphics display of the tool path. Initially, these graphics displays were two dimensional but three dimensional representations of the tool path have been developed recently.

The first automatic verification program has been proposed by Hunt and Voelcker [9]. Fujii and Speckert [10] have developed a CNC lathe simulation package using standard G codes for microcomputer application.

NCTEST [11] displayed the tool path and simulated the user supplied part programs written in standard EIA codes (G Codes). It has been used and tested in a new Computer Numerical Control programming course with positive results at the University of Calgary.

Sungurtekin and Voelcker [12] described a simulator using a model of stock. The simulator generated the cutter swept volumes and subtracted them from the workpiece to produce the intermediate workpiece. The results are then displayed and the process was repeated for the remaining motion statements in the CNC program. However, the system is inherently slow since it contains a lot of computation.

Ozdemir [13] developed a simulation program for proving the user - supplied part programs written for 1050 HLX CNC turning machine. The most important characteristics of the system were declared as that it could be applied for training of CNC part programmers in educational institutes and for trial runs in the industry. The program was written in Basic language on a personal computer with a monochrome monitor.

Unlusoy [14] implemented *"computer graphics"* on a previously developed software package for a feature -based process planning of rotational parts. Cutter location data that is the output by the planning system, was further processed and graphical layout was generated for the verification of NC part program output.

In addition to these works using user-supplied part programs to simulate, almost all integrated CAD/CAM systems (via CAPP) referenced in the previous section, have adopted their own graphic simulation modules to check the part programs and tool paths suggested by their respective generation modules.

CHAPTER III

COMPUTER INTEGRATED MANUFACTURING AND ITS ELEMENTS

3.1 INTRODUCTION

Computer Integrated Manufacturing and its enabling technologies are briefly described in this chapter. However, the main emphasis is given to Computer Aided Process Planning in order to show the importance and place of "*the generation and verification of CNC part programs / tool paths*".

3.2 COMPUTER INTEGRATED MANUFACTURING

Computer Integrated Manufacturing (CIM) systems have emerged as a result of developments in manufacturing and computer technology. It is actually the concept of making a fully-automated factory by combining existing computer technologies in order to manage and control the entire business. In other words, CIM is the computerisation of design, manufacturing, distribution, and financial functions into one coherent system by the use of computers. As the name implies, the computer plays an important role while integrating existing technologies, like Computer Aided Design and Computer Aided Manufacturing, and Computer Aided Process Planning in a typical CIM system.

3.3 COMPUTER AIDED DESIGN

Computer Aided Design (CAD) is a major element of a CIM system. CAD includes any type of design activity which makes use of the computer to develop, analyse or, modify an engineering design. It enables the user to design a component, product, tool, etc. "*Geometric modelling*", "*engineering analysis*", "*design review and evaluation*" and "automatic *drafting*" are the most important topics of tasks performed by a CAD system. The implementation of the CAD results in improvement in the quality, cost and efficiency of the design function.

3.4. COMPUTER AIDED PROCESS PLANNING

Computer Aided Process Planning (CAPP) is considered as the main link between the design stage and manufacturing stage of a part. It consists basically of determination of the processes required to produce the part and putting those processes in a suitable sequence. CAPP can also be defined as: "extracting operations and processing sequences from the part geometry described in CAD". Many different process sequences may be obtained for any part, taking machinability into account. Although the ultimate goal in Process Planning is to decrease total production time, the primary objective is to provide a suitable way to produce the most effective parts with high quality in an optimum time [15]. The cost is a compromising factor to be considered together with this primary objective. It is also important to define criteria used for deciding the process sequence and their relative weights in that decision. A typical Computer Aided Process Planning system includes the following steps or modules [16,17];

- Selection of machine tools
- Selection of set-ups
- Selection of work holding devices
- Determination of machining operations and their sequence
- Calculation of cutting conditions
- *Tool path / CNC part program generation*
- *The verification of the tool path / CNC part program*
- Down-loading the CNC code to machine through RS-232.

There are two basic approaches to the automated process planning[18];

- Variant approach
- Generative approach

In the variant approach, each part is classified according to its attributes and is coded using a coding system. These coded parts are grouped into part families depending upon their similarities. The code and matching process plan for each part are stored in the data base. When a process plan for a new design is required, the most suitable code is found firstly and a previously planned process plan is retrieved from data base.

In the generative approach, there are no process plans stored in the data base. In this case, the data base comprises all the machining information as well as tooling and process planning rules. Generative process planning system generates the required process plans by using the information stored in the data base, only. The generative approach is suitable for application of knowledge-based systems. A knowledge-based system for process planning must be capable of generating process plans for complex parts. It is also desirable to have a knowledge-based system that provides alternative process plans for a given part geometry.

3.4.1 Tool path and part program generation in CAPP

The automatic generation of the tool path and CNC part program is one of the major elements in the Computer Aided Process Planning applications. It also plays a key role being an interface between the CAPP and CAM.

On the basis of the geometric data developed for a workpiece to be machined and the process plan prepared internally, the automatic CNC code generation modules are capable of generating the cutter path and CNC part program for a specific machine tool.

3.4.2 Verification of tool path and CNC part program in CAPP

The animation of the machining processes and graphical display of the cutter path are two modes of simulation modules in CAPP applications. Most of them are capable of performing both of the modes.

The tool path simulation is a helpful tool permitting the user to visualise the motion of the cutting tool. The cutting tools are not shown in the tool path presentation. Therefore, the programmer can not see possible tool collisions. During the tool path display, rapid traverse movements can be represented by dotted lines, while the cutting motions by solid lines. In the case of colour screen, the colour of the path is changed between tool changes. An example to the tool path is shown in Figure 3.1. In this figure, the numbers 1, 2, and 3 describe the sequences and locations of tool replacements performed in the machining.

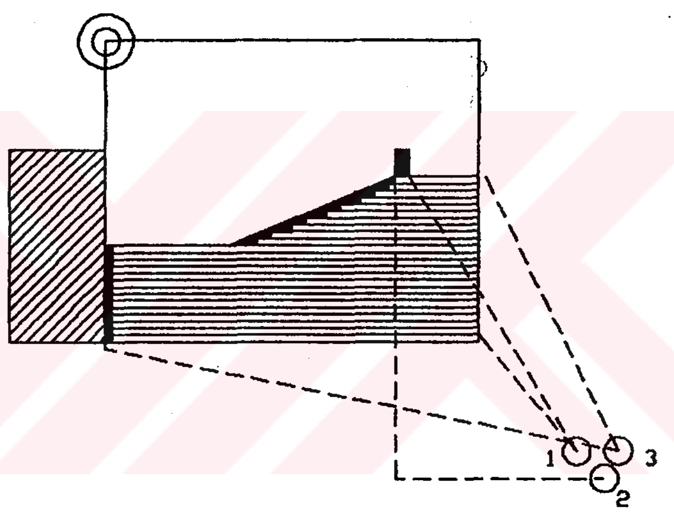


Figure 3.1 An example to the tool path display

The machining animation is useful for debugging, understanding, and finding errors in part programs in the shop-floor without interrupting the production cycle. There has been a dramatic increase in the use of "animation" for manufacturing analysis in the past few years. Most of the controller units now offer a version of their software with animation capabilities

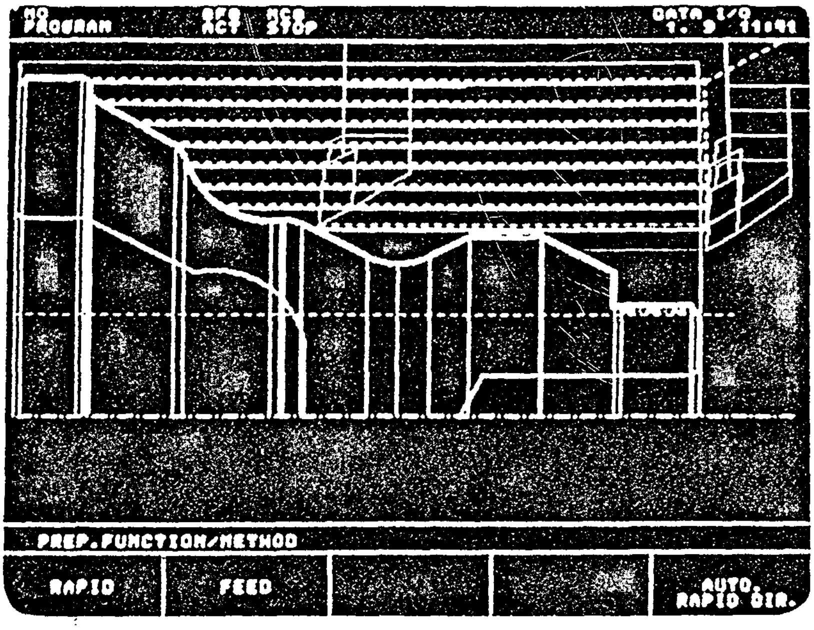


Figure 3.2 The view of machining animation [16].

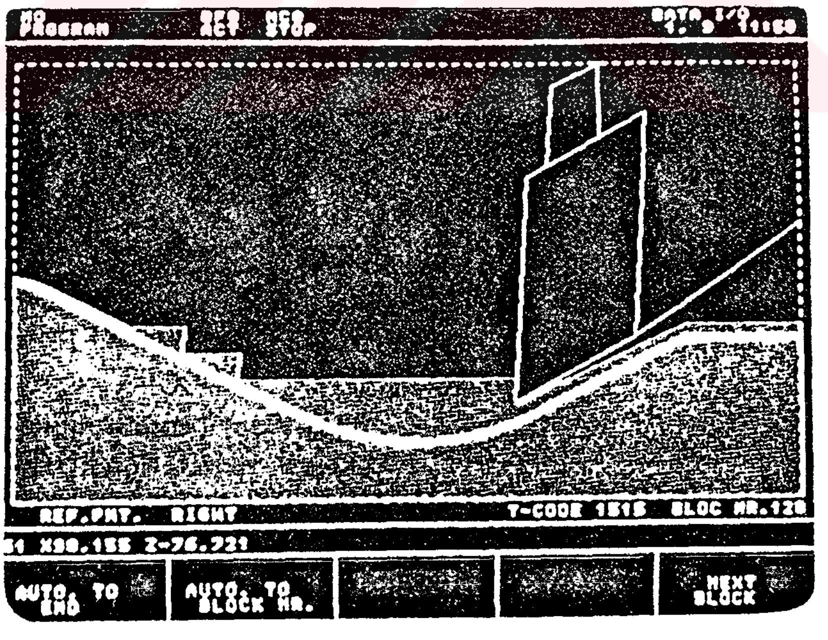


Figure 3.3 The view of the machining animation (zoomed) [16].

which increases the credibility of the model. The machining animation allows the user to simulate exactly what will happen and therefore to check almost everything including the tool collisions when the program is executed. The programmer will be able to check the tool collisions. Figure 3.2 and Figure 3.3 show two screens for typical external machining operations. In the latter, zooming is performed to get a neat understanding about the cutting.

3.5 COMPUTER AIDED MANUFACTURING

Computer Aided Manufacturing (CAM) is the term used commonly for all the computer-controlled activities that involve a process of converting raw material into finished products. CAM-I (Computer Aided Manufacturing - International, a non-profit firm based in U.S.A) defines the term CAM as follows:

"the effective utilisation of computer technology in the management, control, and operations of the manufacturing facility through either direct or indirect computer interface with the physical and human resources of the company".

The most important elements of CAM are;

- CNC manufacturing techniques
- Robotics
- Flexible Manufacturing Systems (FMS)
- Computer Aided Inspection (CAI)
- Computer Aided Testing (CAT).

CHAPTER IV

COMPUTER NUMERICAL CONTROL
AND PART PROGRAMMING

4.1 INTRODUCTION

The purpose of this chapter is to review briefly some of the important topics which may be included under the heading of Computer Numerical Control and part programming.

4.2 NUMERICAL CONTROL

Numerical Control which is abbreviated as NC is a system in which actions are controlled by means of numbers or coded instructions. Once these instructions are entered into the system, they are carried out automatically in a predetermined sequence without any human intervention. When the work changes, the program of instructions is also changed. This capability makes the Numerical Control process more flexible and favourable as compared with the traditional methods.

4.3 BASIC COMPONENTS OF AN NC SYSTEM

A typical Numerical Control system consists of three basic units. These are;

1) *Program of instructions*: It is known as "part program" and serves as an input to controller unit. It is the detailed set of instructions which tells the machine tool what to do.

2) *Controller unit*: It reads and interprets the program of instructions and converts it into mechanical actions of the machine tool. Typical elements of the controller unit includes the tape reader, a data buffer, signal output channels to the machine tool, and feedback channels from the machine tool.

3) *Machine tool*: The machine tool is the most expensive element in the NC system. It performs the commands coming from the controller unit.

4.4 NUMERICAL CONTROL METHOD

The following steps must be accomplished to utilise the Numerical Control in manufacturing;

a) Engineering drawing of the part must be interpreted in terms of manufacturing processes to be used,
b) Part program must be written.
c) A suitable input media representing the program of instructions must be prepared.
d) The instructions must be checked prior to production.

e) The part is then produced on a corresponding NC machine.

4.5 COMPUTER NUMERICAL CONTROL

Computer Numerical Control (CNC) is obviously an NC system using a minicomputer as the controller unit. The major difference from conventional NC is the replacement of the tool controller by a computer. NC punched tape which contains the program of instructions for a particular job is cycled through the reader for another part with the same configuration. In the case of CNC, the program is entered once and then stored in the memory. Compared to conventional NC, it offers more flexibility and advantages. Computer Numerical Control has therefore become an increasingly popular approach for small or specialised machining operations.

4.6 COORDINATE SYSTEM AND AXES OF MOVEMENTS

Machine tool motions are generally described in x, y, and z; Cartesian coordinates. Figure 4.1 illustrates the right hand convention. The rotational axes of a, b, and c are used to denote angles about the x, y, and z axis, respectively.

In naming the axis of a machine tool; the X and Y axis describe a plane orthogonal to the penetration axis of the tool. The tool penetration axis is usually the Z-axis, with negative Z representing penetration and positive Z representing withdrawal. In the X-Y plane, the axis with the greatest range of travel is commonly selected as the X-axis, while the other is taken as Y-axis. Two axes (X and Z) are enough to command the movement of the tool

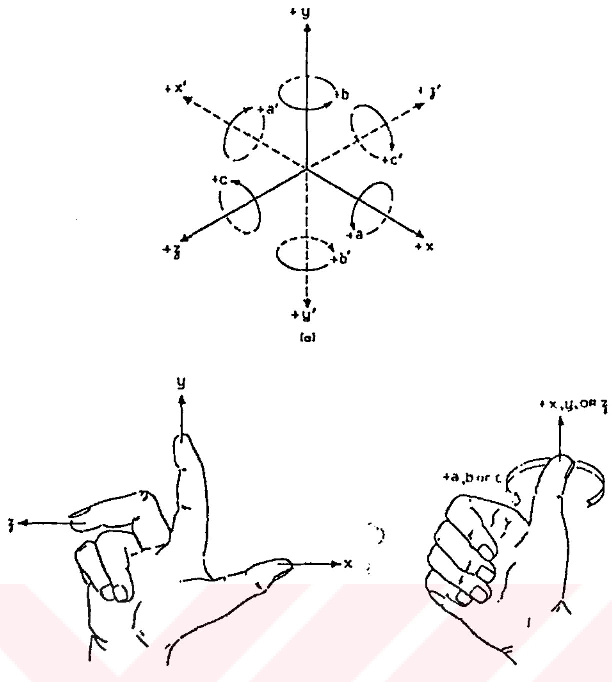


Figure 4.1 Right hand convention for Cartesian coordinates [20].

relative to the workpiece. The Z-axis is the axis of rotation of the workpieces. The X-axis defines the radial location of the cutting tool. Some auxiliary axes can also be defined on the CNC machine tools. Axes on a typical CNC lathe is shown in Figure 4.2.

4.7 CLASSIFICATION OF MACHINE TOOL CONTROL SYSTEMS

NC systems are classified into three groups according to machine tool control concerning the amount of control over the relative motion between the component and cutter. These are:

- Point-to-point positioning control systems
- Straight-cut control systems
- Contouring control systems

Figure 4.3 gives a demonstration for each type of control.

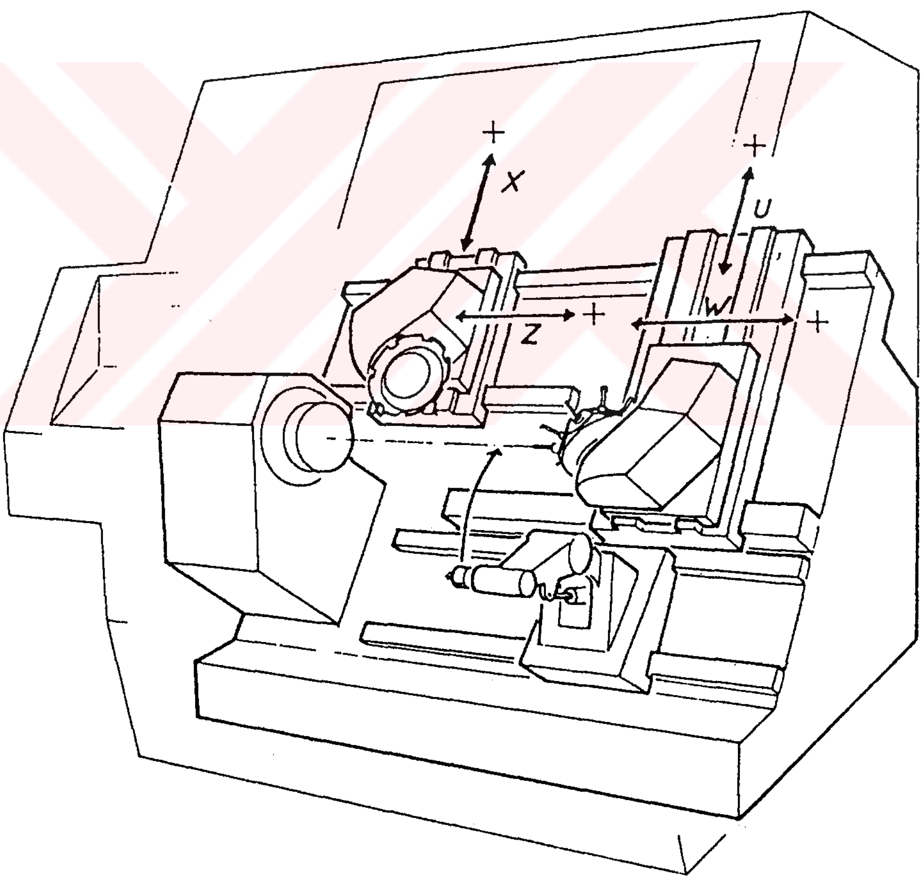


Figure 4.2 The axes on a CNC lathe [19].

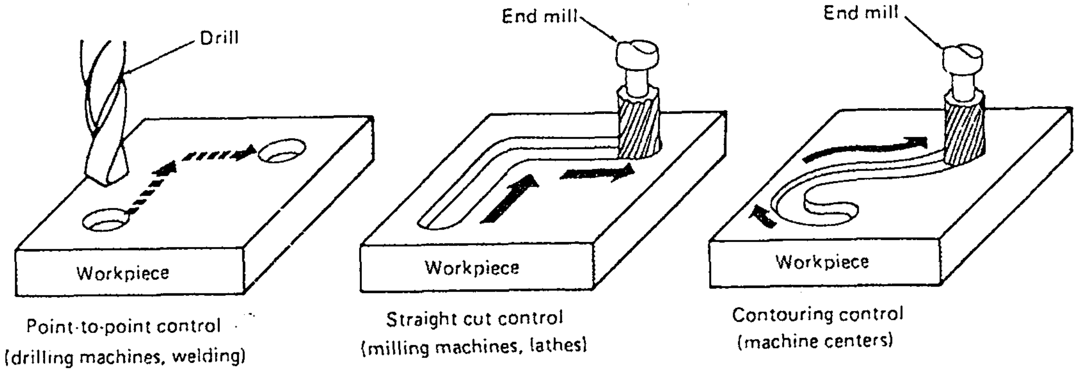


Figure 4.3 Types of control systems [21].

4.7.1 Point-to-point positioning systems

The objective in this system is to move the cutting tool to a predefined location. The speed or path by which this movement is accomplished is not important. Once the tool reaches the desired point, the machining operation is performed. Point-to-point systems are the cheapest machine tool control system. The most common applications of the point-to-point control are drilling, boring, spot welding, and sheet metal punching. In addition to the function of positioning, this system can control some auxiliary functions (M words) such as tool change (M06) and coolant on/off (M08) on a computer numerical control machine.

4.7.2 Straight-cut CNC systems

Straight-cut systems are capable of moving cutting tool parallel to the major axes at a controlled rate selected for a particular machining. It is therefore appropriate for performing milling operations of rectangular configurations. Angular cuts on the workpiece is not possible. A CNC machine capable of doing straight-cut movements is also capable of performing point-to-point movements.

4.7.3 Contouring (continuous path) CNC systems

Contouring is the most complex, flexible, and expensive type of machine tool control system. Continuous path systems have the capacity of controlling more than one axis of the machine tool simultaneously. The path of the cutting tool is continuously controlled to generate the desired geometry of component. Straight or plane surfaces at any orientation, circular paths, conical shapes, and mathematically definable shapes are possible under this type of control. Milling and turning operations are common examples of the use of contouring control.

Considering a two axes control system in which the machine tool table is moved in the xy plane, one can explain the basic differences between those three control types mathematically. The control is achieved over the x and y co-ordinates only with the point-to-point systems. With the straight-cut systems, it is provided for either dx/dt or dy/dt, but only one at a time. However, in the case of contouring systems, both of the rates can be controlled simultaneously.

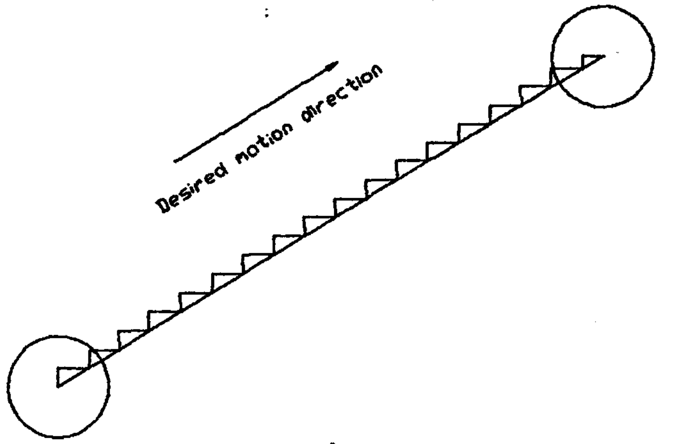
4.8 TYPES OF INTERPOLATIONS

Webster's New World Dictionary defined the word interpolation as; "to estimate a missing functional value by taking a weighted average of known functional values at neighbouring points". In the case of CNC, the controller can estimate the programmed path based on the information about starting and finishing points of the motion. When interpolating linearly, it makes actually a series of very small one axis movements from the starting point to the end point by generating a stairway-like path as shown in Figure 4.4. However, each step along the stairway is very small, and so the resulting path will appear to be a perfect straight line. The size of each step determines the resolution of the machine tool. Of course, the smaller the step, the better the resolution.

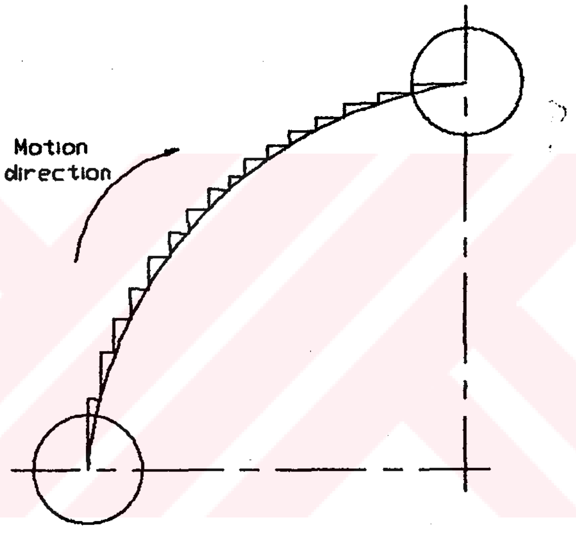
Linear, circular and parabolic interpolation commands have been commonly designed by the CNC control manufacturers. Linear interpolation represents the machining of a straight line between an initial and a final location. This feature is programmed by using the preparatory code of G01.

Circular interpolation as also illustrated in Figure 4.4 is the process of generating circular tool movements. G02 (clockwise) and G03 (counter clockwise) codes are used for programming a circular arc on CNC machines. In many cases, the circular interpolation is limited to only one quadrant of the circle in the machine tool system.

Other kinds of interpolation have much more specific applications and are seldomly used. For instance, parabolic interpolation is advantageous only with 5-axes machining, since the required blocks of information for simultaneous multi-axes movements are reduced dramatically.



a)



b)

Figure 4.4 Linear (a) and circular (b) interpolations

4.9 MODES OF PROGRAMMING

There are basically two modes of programming;

- Absolute programming
- Incremental programming

In absolute programming, all positions are referenced relative to the same origin selected for each axis prior to starting the program. This means that all motions defined in a part program are actually the locations of the particular point relative to the selected origin.

In incremental programming, the distance for the current position is measured from the last tool position. In this system, the control stores and processes only the additional path measurement. Therefore, the programming is more simple than programming absolute values, requiring a less complicated control design.

#	Absolute mode	Incremental mode
1	X1.0 Y1.0	X0.0 Y0.0
2	X2.0 Y1.0	X1.0 Y0.0
3	X2.5 Y0.5	X0.5 Y-0.5
4	X3.5 Y0.5	X1.0 Y0.0
5	X3.5 Y2.5	X0.0 Y2.0
6	X5.0 Y2.5	X1.5 Y0.0
7	X5.0 Y2.0	X0.0 Y-0.5
8	X6.0 Y2.0	X1.0 Y0.0
9	X7.0 Y3.5	X1.0 Y1.5

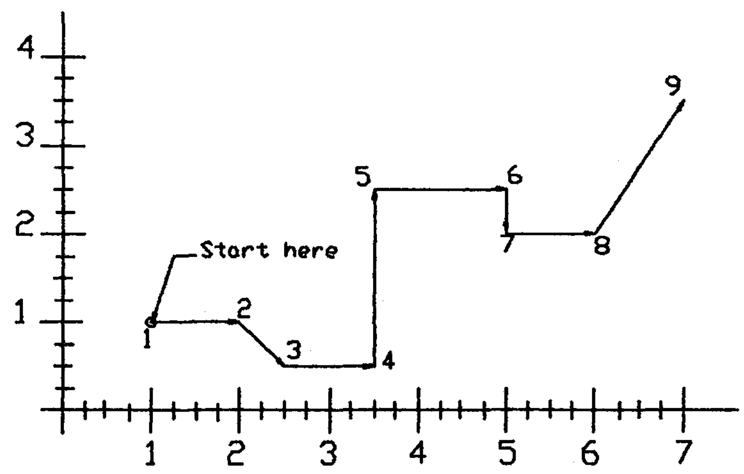


Figure 4.5 The explanation of absolute and incremental modes

A modification of a position can easily be done with an absolute system, since all distances are measured from the reference. In the case of incremental programming however, the part must be re-programmed from the point at which the original program has been modified. However, it is very easy to check possible geometric errors in an incremental part program, since the sum of the position commands for each axis must be equal to zero. The incremental systems are generally preferred in turning processes. Because, it allows "mirror-image programming" which can simplify the part programming of symmetrical parts [22]. Figure 4.5 stresses the difference between the incremental and absolute modes.

4.10 ZERO AND REFERENCE POINTS ON CNC MACHINES

Workpiece dimensioning is usually guided to some "zero points" and "reference points" defined on each type of CNC machines. Figure 4.6 shows these points to give a general understanding on the locations of them. Zero and datum reference points for CNC lathes are also specifically illustrated in Figure 4.7.

4.10.1 Machine zero point

The machine zero point (known as machine offset or home position) is at the origin of the machine co-ordinate system. It is fixed and can not be shifted. It should be quickly approached after the machine is turned on. The location of the machine zero is determined by the manufacturer. The machine offset on CNC machine tools plays a critical role as it is the exact distance from a datum at the spindle end of the machine to a point on the cross slide when the slides are at home positions. The X datum of machine zero point at the spindle end is always the centreline of the spindle. However, the Z datum point at the spindle end varies depending upon what type of work holding device has been installed. The datum is always at the front end of the device chosen.

The X and Z points for the cross slide depends on the type of the tooling system fitted. In the case of a disc turret, the X axis datum point is taken as the centreline of the internal disc stations and the Z axis datum is selected as the front face of the disc. These values for the machine offset have been generally set by machine tool manufacturer. On the other hand, it may be necessary to change these values when the type of work holding device is altered or when a collision involving the disc turret occurs.

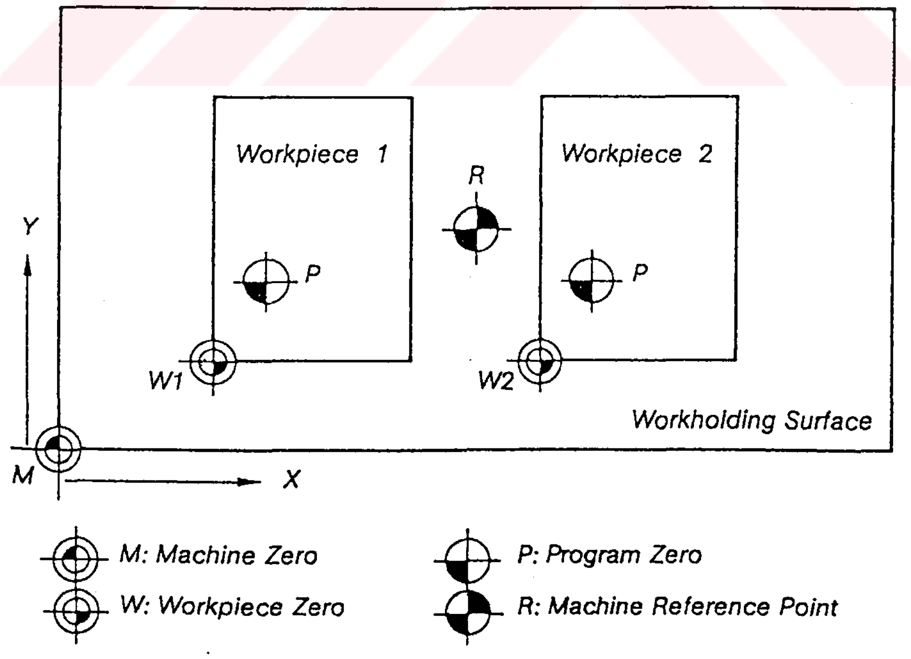


Figure 4.6 The convention used for reference points [19].

4.10.2 Machine reference point

In the cases where the machine zero point can not be approached due to the clamped workpiece or a fixture, a more convenient point be adopted as a fixed "machine reference point". The value in setting the axes, in this case, represents the difference between the machine zero point and machine reference point.

M = Machine tool zero point

A = Blocking point: can be identical with workpiece zero point

W = Workpiece zero point = program zero point

C = Control zero point: may be shifted to equal the workpiece zero point

B = Start point, to be determined in the program. This is from where the first tool starts the machining procedure

R = Reference point of the machine tool. Its position is determined by cams and measuring system. It is essential to know the distance between R and M in order to relate the position of the axis to that of the reference point.

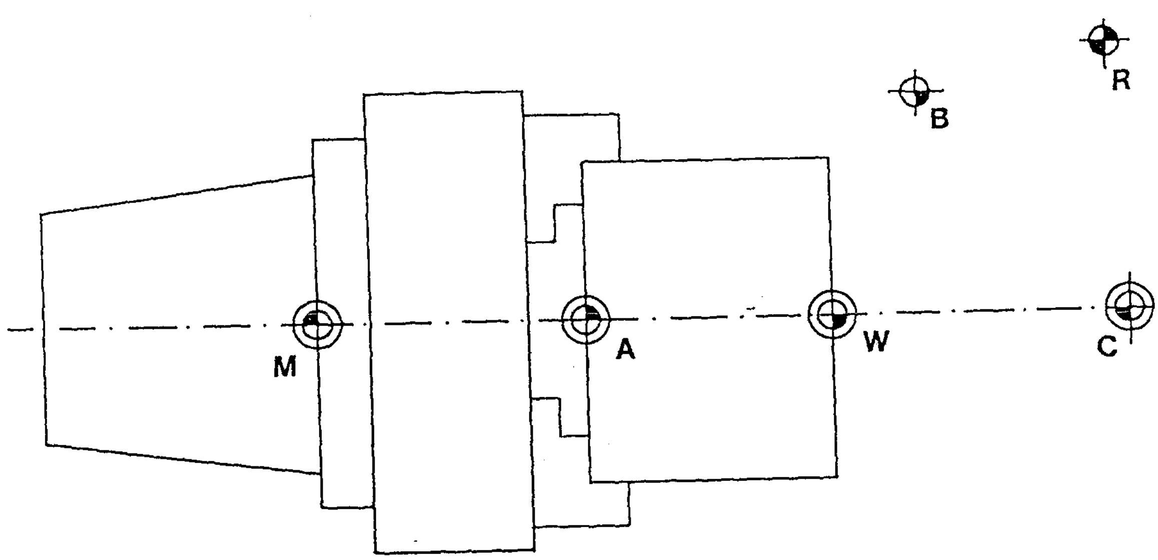


Figure 4.7 Zero and datum points for CNC lathes [19].

4.10.3 Workpiece zero point

The origin of the workpiece co-ordinate system is known as "workpiece zero point" and is freely selected by the part programmer. However, it is generally accepted as the intersection of the rotary axis with the reference face of the workpiece.

4.10.4 Program zero

"Program zero" is actually a starting point to the operations. It is determined by the part programmer and serves as a tool change position. It may be shifted by any value on each axis. It should not be too far from the front end of the workpiece.

4.11 TYPES OF COMPENSATIONS

Even if the part programs are prepared free of errors and as perfectly as possible, there is often a need for "compensation" depending upon the tooling and workpiece geometry. The compensation is a type of correction for the part programs and used for several different purposes on different types of CNC machines. For example, multi-axes machining centres have tool *length compensation* and *cutter radius compensation*, whereas electrical discharge machines need *wire radius compensation* and *wire taper compensation*. Turning centres and machines are generally of ***tool nose radius (TNR) compensation*** and ***offset (tool offset) compensation***. We will discuss in this section only the TNR and tool offset compensation, since we concentrated on *CNC lathes* in this thesis.

4.11.1 Tool offsets

Tool offsets are used for every kind of machining operation performed on a CNC turning machine. They are sometimes referred as dimensional tool offsets and allow the operator to adjust the position of tools without changing any tool set-up. Once the tools have been set in their holders, the control has to be aware of the position of the cutting tip of each tool. Therefore, the position of the first tool called by part program is taken as datum, and the positions of other tools are expressed as differences to this position. Most control provides an offset table to enter the offsets. Each offset includes two values; the X offset and Z offset. The programmer or user can complete the offset table by using the soft keys on the machine.

Tool offsets are vitally required to allow for tool wear. By using the offsets, one can easily adjust for problems in both axes. For instance, if a diameter is coming out incorrectly, it is obvious that the X axis offset would need an adjustment.

4.11.2 Tool Nose Radius (TNR) compensation

Single point cutting tools used extensively in turning and boring applications have a small radius on the cutting edge, as shown in Figure 4.8. They are different from the theoretical tools having no radius on cutting edge. Although these nose radii are considered to be quite small, they may cause some deviation from the required shape of the component in some applications.

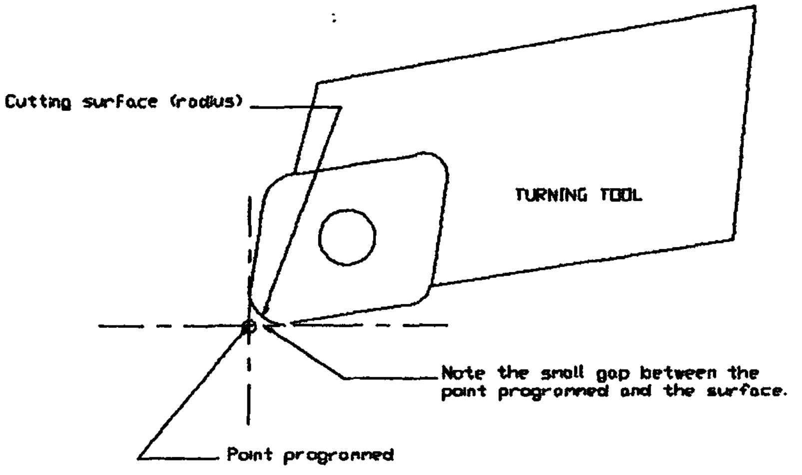


Figure 4.8 Tool nose radius on the cutting edge of a tool.

In cylindrical and face turning; the tool nose radius does not affect the contour being machined, since the point of the tool comes into contact with the workpiece as illustrated in Figure 4.9. However, the point does not come into contact with the workpiece when conical and circular surfaces are machined. So, the surface being machined does not match the required dimensions.

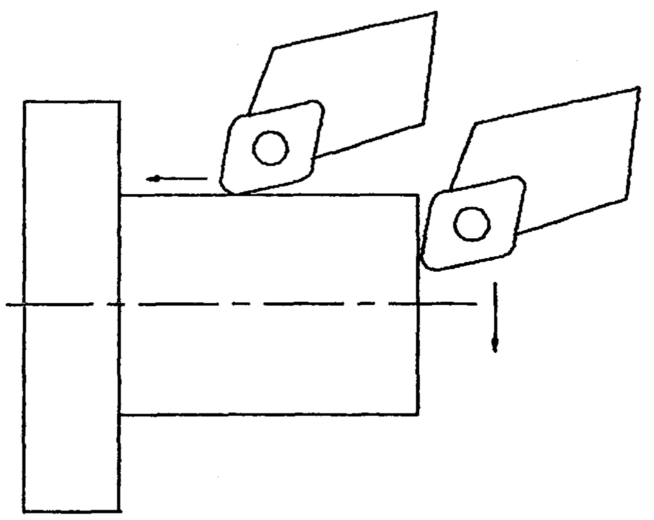


Figure 4.9 The operations not requiring TNR compensation.

The nose radius compensation gives the programmer an easy method of ensuring in which the tip radius remains in correct contact with the workpiece. It is too easy to program the TNR compensation (if included in the control) on the CNC lathes. To do this, one first instates compensation by using either G41 or G42; second gives the motion commands, and finally cancels the compensation by G40. G41 denotes "TNR compensation for left", and used in left-hand/internal and right-hand/external turning operations. G42 stands for "TNR compensation for right", and used in left-hand/external and right-hand/internal turning processes.

The radii of the tools being called by the program can also be input prior to the machining and are stored in the control's memory together with the values of X and Z offsets. It is well worth to point out that there is actually no need for the TNR compensation when simple chamfers and circular arcs having small radii are to be machined.

4.12 CNC PART PROGRAMMING

Part programming includes the planning and specification of processing steps to be performed for machining a particular component. The part program defines the sequence of operations required to fabricate the workpiece; gives the coordinate positions of the operations, specifies the spindle speed, and feed and determines also whether the same tool continues the next operation or whether a tool change is required. These items are put together in a specified order to define a block of information needed to execute an operation. These machining instructions for the part are prepared in machine-usable form and provided to the machine tool by means of a suitable input media.

4.13 INPUT MEDIA

The machining operations to be performed on a CNC system are defined by PART PROGRAM. It can be fed into the system manually or by means of punched tapes, magnetic tapes, or by diskettes shown in Figure 4.10.

Manual input is not commonly used on CNC systems, since it is a time consuming method. It is only employed when the program will be used once.

Punched tapes are predominantly used by CNC systems as an input media. There are eight regular columns of holes running in the lengthwise direction of the tape. There is also a ninth column of holes between the third and fourth columns as shown in Figure 4.11. These are smaller than others and used for feeding the tape. The coding, known as BINARY SYSTEM, is

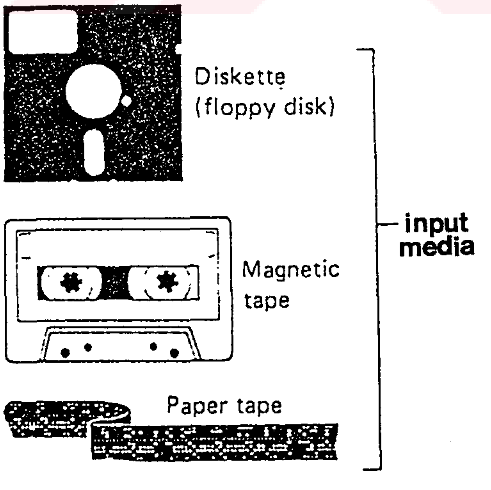


Figure 4.10 Input Media for feeding part programs [21].

provided by either the presence or absence of a hole in the various positions. There are two possible conditions for each hole; either the presence or absence. Four digits are required in the Binary System to -

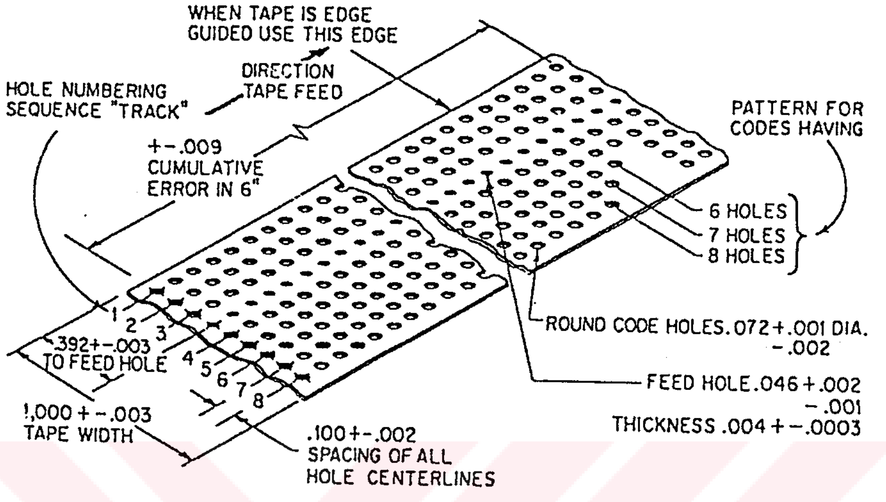


Figure 4.11 Perforated Tape [23].

represent any of the single digit numbers in the decimal system. However, there are eight columns present on the tape. The reason is because alphabetical letters, plus and minus signs and other symbols are also needed in CNC tape coding.

To minimise the possibilities of errors, a parity check system has been implemented in the standardised coding. The parity check code for EIA consists of an extra hole in fifth column so that the number of holes across the tape will always be odd. The parity check code for ASCII (American Standard Coding for Information Interchange) requires even number of holes across the tape in every row. In this case, holes in eight columns are used for parity adjustment whenever it is required.

In most CNC applications, codes are recorded on half-inch wide cartridged magnetic(MAG) tape. Although MAG type tapes are speedy and cheaper than punched tapes, they have never gained the wide acceptance since the codes on it can not be seen by the naked eye. The alphanumeric codes are shown by dashes.

ASCII and EIA punched tape codes together with MAG type are illustrated in Figure 4.12.

4.14 CNC WORDS

There are different types of words used in the formation of part program blocks;

4.14.1 N Words

This letter address is used to specify a sequence number. It allows the programmer to organise each line in a program by numbers. No decimal point is allowed with the N word.

4.14.2 G Words

These words are also known as "*preparatory functions*". They are used to prepare the controller for instructions that are to follow. For instance, G00 is used with the contouring systems to prepare for a point-to-point operation, whereas G01 to prepare for the linear interpolation. G70, G71, G90, G91 words stand for inch, metric, absolute, incremental mode selections.

CHARACTER	MAG TAPE CODING							(ISO) ASCII									EIA								
	1	2	3	4	5	6	7	1	2	3	o	4	5	6	7	8	1	2	3	o	4	5	6	7	8
0	—			—							o		O	O						o			O		
1	—						—	O			o		O	O		O	O			o					
2		—					—		O		o		O	O		O		O		o					
3	—	—						O	O		o		O	O			O	O		o		O			
4			—				—			O	o		O	O		O			O	o					
5	—		—					O		O	o		O	O			O		O	o		O			
6		—	—						O	O	o		O	O				O	O	o		O			
7	—	—	—				—	O	O	O	o		O	O		O	O	O	O	o					
8				—			—				o	O	O	O		O				o	O				
9	—			—				O			o	O	O	O			O			o	O	O			
a	—				—	—	—	O			o				O		O			o			O	O	
b		—			—	—	—		O		o				O			O		o			O	O	
c	—	—			—	—		O	O		o				O	O	O	O		o		O	O	O	
d			—		—	—	—			O	o				O				O	o			O	O	
e	—		—		—	—		O		O	o				O	O	O		O	o		O	O	O	
f		—	—		—	—			O	O	o				O	O		O	O	o		O	O	O	
g	—	—	—		—	—	—	O	O	O	o				O		O	O	O	o			O	O	
h				—	—	—	—				o	O			O					o	O		O	O	
i	—			—	—	—		O			o	O			O	O	O			o	O	O	O	O	
j	—					—			O		o	O			O	O	O			o		O		O	
k		—				—		O	O		o	O			O			O		o		O		O	
l	—	—				—	—	O			o	O			O	O	O	O		o				O	
m			—			—		O		O	o	O			O				O	o		O		O	
n	—		—			—	—		O	O	o	O			O		O		O	o				O	
o		—	—			—	—	O	O	O	o	O			O	O		O	O	o				O	
p	—	—	—			—					o		O		O		O	O	O	o		O		O	
q				—		—		O			o		O		O	O				o	O	O		O	
r	—			—		—	—		O		o		O		O	O	O			o	O			O	
s		—			—			O	O		o		O	O	O			O		o		O	O		
t	—	—			—		—			O	o		O		O	O	O	O		o			O		
u			—		—			O		O	o		O		O				O	o		O	O		
v	—		—		—		—		O	O	o		O		O		O		O	o			O		
w		—	—		—		—	O	O	O	o		O		O	O		O	O	o			O		
x	—	—	—		—						o	O	O		O	O	O	O	O	o		O	O		
y				—	—			O			o	O	O		O					o	O	O	O		
z	—			—	—		—		O		o	O	O		O		O			o	O		O		
+								O	O		o	O		O						o		O	O	O	
—						—	—	O		O	o	O		O						o				O	
/	—				—			O	O	O	o			O		O	O			o		O	O		
.	—	—		—	—	—	—		O	O	o	O		O			O	O		o	O		O	O	
,	—	—		—	—					O	o	O		O		O	O	O		o	O	O	O		
%			—	—	—		—	O		O	o			O		O	O	O		o	O				
&					—	—					o									o					
(											o	O		O					NOT	o		ASSIGNED			
)											o	O		O		O			NOT	o		ASSIGNED			
:									O		o	O	O	O					NOT	o		ASSIGNED			
CR								O		O	o	O				O				o					O
DELETE								O	O	O	o	O	O	O	O	O	O	O	O	o	O	O	O	O	
SPACE											o			O		O				o		O			
=											o									o					
TAB								O			o	O						O	O	o	O	O	O		
											o									o					

Figure 4.12 The MAG, ASCII, and EIA punched tape codes [24].

There are more than ninety-nine G words standardised. On the other hand, all of them may not be installed even on lace models of CNC machines. Only one motion G word is allowed in a block. That is, only one of G00, GO1, G02, or G03 is allowed in a command. No decimal point is allowed with the G word. A complete list of G words are given in Table A.1, in the Appendix.

4.14.3 X, Y, and Z Words

These words are used to designate a movement along the X, Y, and Z axis, respectively. The movement is dependent on whether the control is currently in absolute or incremental mode. In the absolute mode, the end point specified is relative to the program zero point. In the incremental mode, the motion begins from the current position of the tool.

4.14.4 I, J, and K words

These letter addresses are used to specify the arc in a circular movement. The I, J, and K words specify the distance and direction of the starting point of arc to the centre of the arc.

4.14.5 F Words

This word specifies the feed in a machining operation. It enables the programmer to specify the rate at which the axes will traverse during machining. The type of CNC machine tool determines the designation for the F word.

4.14.6 S Words

This letter designates a spindle speed. The type of the CNC machine again determines the designation for S Word.

4.14.7 T Words

This word specifies a tool (or a tool station handling a cutting tool) to be used for a machining operation. It would only be needed for CNC machines having a tool turret or automatic tool changing magazine. The format for the T Words varies according to the types of CNC equipment. Some of them use an "*M word - I word*" combination for designating the tool changing operation.

4.14.8 M Words

The M words are used to specify a series of miscellaneous functions available on a CNC machine. M words can simply be considered as programmable ON/OFF switches that turn-on and turn-off functions such as coolant or spindle. A complete list of M words are given in Table A.2, in the Appendix.

4.15 CNC DATA FORMATS

A tape format means the organisation of CNC words within a block. There are three types of formats currently in use;

- Word address format
- Tab sequential format
- Fixed block format

The word address system is the most commonly used and most flexible format among the three. Each word can be preceeded by a character denoting the word address letter. The words can be arranged in any order.

In the tab sequential formatting, the address of the words are the same with those in the word address formatting. However, no address predecessor is required, since the words are inserted in a fixed and prescribed sequence and separated from each others by a "TAB".

Fixed address format is the least flexible of the formats. The words must be of the same format and length. Both plus and minus signs and zero distance movement must be specified. Below is an example, in which two holes are drilled subsequently by using a feed of 3,5 ipm and a speed of 675 rpm, for each type of formats.

<u>Word address format:</u> N001 X2.000 Y4.000 F3.50 S675 EOB
N002 X5.000 EOB

<u>Tab-sequential format:</u> 001 TAB 2.00 TAB 4.00 TAB 3.50 TAB 675 EOB
002 TAB 5.00 TAB TAB TAB EOB

<u>Fixed block format:</u> 001 +02.000 +04.000 3.50 675 EOB
002 +05.000 +04.000 3.50 675 EOB

4.16 METHODS OF CNC PART PROGRAMMING

There are several different methods of CNC part programming. These are classified into five groups [25];

- Manual part programming
- Computer assisted part programming
- Manual Data Input (MDI)
- CNC part programming using CAD
- Computer automated part programming

4.16.1 Manual part programming

Manual part programming was widely used in a variety of machining operations, until microprocessor based CNC controllers became available on a large scale in late 70's. Manual part programming of CNC machines involves the manual selection of cutting tools, clamping devices, fixtures, cutting parameters like feed and speed, and the cutting sequence based on the technical drawing of the component to be produced.

The part programmer's job is to specify the machining instructions on a form called a "manuscript". The manuscript is a listing of the positions of the tool relative to the workpiece that the machine must follow in order to perform the operation. It also includes other commands such as speeds, feeds and tooling information. There are various forms of manuscripts as shown in Figure 4.13 depending upon the machine tool and the tape format to be used. The sheets for programming and description of operations for

Boxford 250 CNC lathe are also given in Figure A.1, and Figure A.2, respectively in the Appendix.

The manual part programming is basic to writing all CNC programs. Therefore, one should start his training in manual part programming, since it is absolutely necessary to understand, read, and change the part programs on CNC machines. Figure 4.14 shows a typical procedure of the manual part programming.

4.16.2 Computer assisted part programming

For relatively simple parts, the manual part programming is a suitable application. However, in the case of complex parts and complicated three axes contouring applications, the manual part programming becomes an extremely tedious task and prone to error. The part programmer's job is very difficult and time consuming. Therefore, in these instances it is much more appropriate to use a computer to assist in the part programming process.

Computer assisted part programming does not eliminate the part programmer from the programming procedure. The part programmer translates each element of the workpiece *such as point, line, arc, circle* into a mathematical expression or a statement by using a pre-selected computer assisted part programming language, like APT (Automatically Programmed Tools). After defining the workpiece geometry, the part programmer specifies the tool paths that the cutting tool will follow to machine the part by using again the CNC programming language.

NC Part Programming Manuscript

Two-Axis PTP or Contouring Machine
Word Address Format

Part No. ______________ Date ______________

Part Name ______________ Prepared by ______________

n-WORD	g-WORD	x-WORD	y-WORD	f-WORD	s-WORD	t-WORD	m-WORD	EOB	COMMENTS

NC Part Programming Manuscript

Two-Axis Point-to-Point Machine
Tab Sequential Format

Part No. ______________ Date ______________

Part Name ______________ Prepared by ______________

SEQUENCE NO.	TAB/EOB	x-COORD	TAB/EOB	y-COORD	TAB/EOB	m-WORD	TAB/EOB	COMMENTS

NC Part Programming Manuscript

Two-Axis Point-to-Point Machine
Fixed Block Format

Part No. ______________ Date ______________

Part Name ______________ Prepared by ______________

SEQUENCE NO.	x-COORD	y-COORD	m-WORD	COMMENTS

Figure 4.13 Manuscripts for different applications [23].

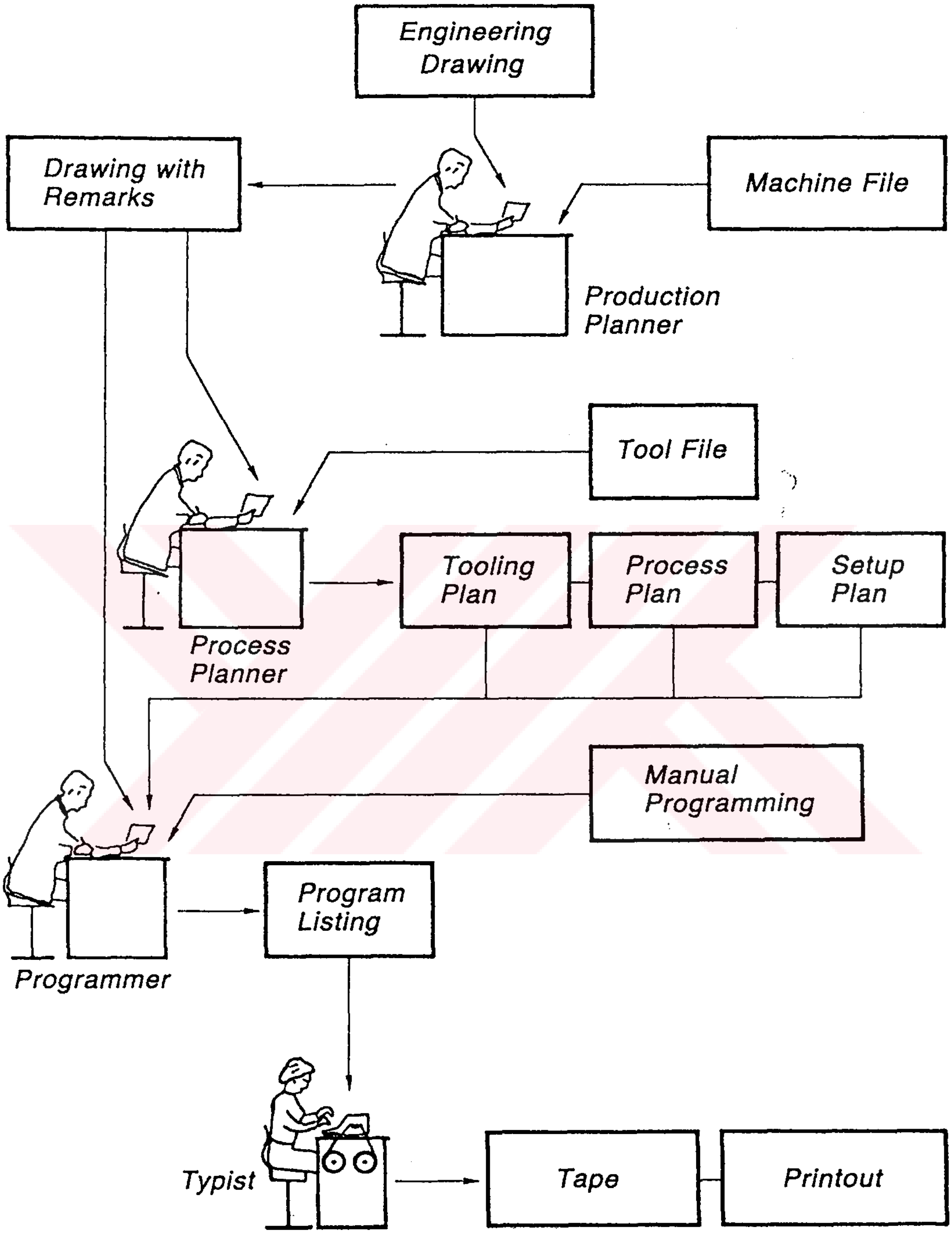


Figure 4.14 Information flow of the manual part programming [19].

The computer then receives the data which is prepared by the part programmer, and perform all the necessary calculations for post-processor to code the CNC tape. The post-processor prepares the punched tape for a specific machine tool. A typical information flow in Computer assisted part programming is given in Figure 4.15.

In parallel with the growth of the CNC industry, many computer assisted languages are developed. All of these use "English-like" words to describe the part, tool, type of operation and other auxiliary operation required for machining operation. The "English-like" words may sometimes be abbreviated to single letters. Computer assisted programming languages are more accurate and flexible than manual part programming. They also provide considerable time savings in programming. APT, AUTOSPOT (Automatic System for Positioning Tools), and SPLIT (Sundstrand Processing Language Internally Translated) are three of the earliest CNC programming language. However, there are more than one hundred computer assisted CNC languages like COMPACT II, ADAPT, EXAPT available on the market.

4.16.3 Manual Data Input (MDI)

CNC machines are currently being offered with sophisticated controllers enabling direct programming on the shop-floor. The CNC part program-like commands are entered directly into the machine tool controller unit through the keyboard on the control panel of a CNC machine as shown in Figure A.3 and Figure A.4, in the Appendix.

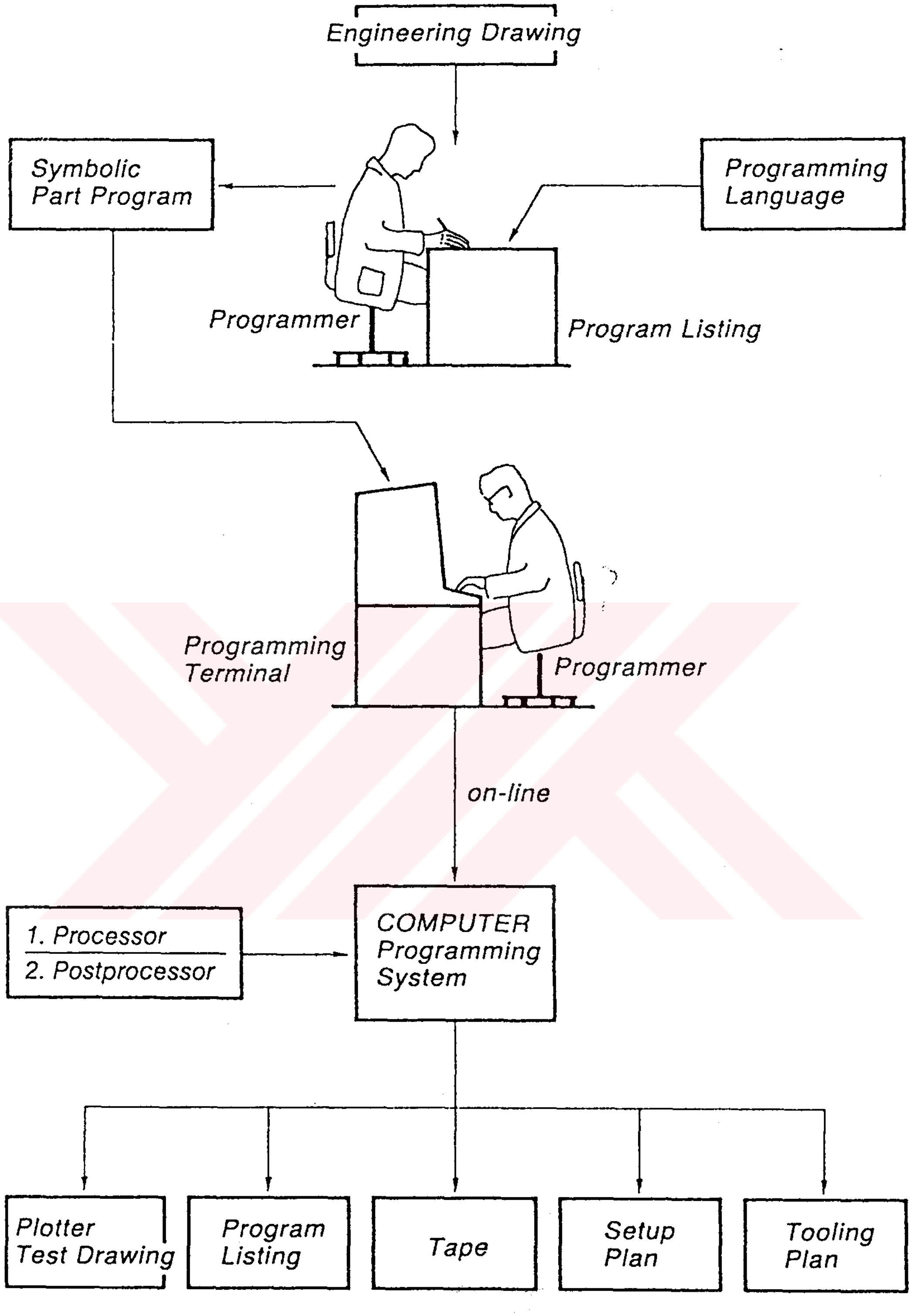


Figure 4.15 The flowchart of the Computer Assisted Part Programming [19].

With the use of MDI unit, commands entered from the keyboard and display screen can only be executed once. If the command must be executed a second time, the operator must enter MDI commands a second time, too. This is one disadvantage of using MDI. However, this can be avoided by saving the MDI commands into a part program file. The manual data input unit can even be used for machining a workpiece. The operator can make machining commands in the same way as in a CNC part program, since all CNC words are available in MDI unit. On the other hand, one must be careful while machining workpieces by using MDI, because, the commands will be executed as entered. If the operator makes a mistake, it would be a disaster [26].

With the introduction of MDI units, the programming procedure is simplified to allow the operators rather than part programmers to do the programming.

4.16.4 CNC part programming by CAD

The definition of the part geometry and specification of the tool motions are the two basic duties of the part programmer in both the manual and computer assisted part programming, as discussed in the previous sections. The difference between the two is that the machining instructions are written in "English-like" statements of the CNC part programming language with the latter.

In the case of CAD-part programming, the work done manually by the part programmer in the computer assisted part programming is performed on computer by using the interactive CAD techniques. Remaining procedure is the same. This type of programming methods provides a graphic display of the workpiece and the tool paths. They eliminate the time consumed in the geometry definition. The part programmer can develop a computer graphics model for a workpiece and store it in the memory. The model contains all the specifications of the part and can be visualised on the graphics monitor. The more sophisticated systems are capable of making immediate graphic display of the cutter path for verification.

4.16.5 Computer Automated Part Programming

In the case of Computer Automated Part Programming (or Computer Aided Part Programming), the computer not only assists but also automates the part programming. The interactivities in "*CNC part programming by CAD*" and human assistance in "*computer assisted part programming*" techniques are completely eliminated from the part programming procedure. These systems possess sufficient logic and decision-making capability to accomplish CNC part programming with no human assistance for a given geometric model of a workpiece.

Computer Aided Part Programming is closely related to Computer Aided Process Planning applications and is commonly considered as a basic module in their information flow [27, 28]. On the basis of the part design produced by the CAD system and process plan prepared in the CAPP system, the part programs for CNC machines are automatically generated.

CHAPTER V

COMPUTER AIDED PART PROGRAM GENERATION AND GRAPHICAL SIMULATION OF TURNING PROCESSES

5.1 INTRODUCTION

This chapter introduces the ***Computer Aided Part Program Generation and Graphical Simulation of Turning Processes***. In section 5.2, turning processes are classified. Section 5.3 discusses the specifications of the Boxford 250 B CNC lathe selected as the target machine tool - controller combination in this work. The computer program prepared and method used in the programming are introduced in section 5.4. Execution of the program is explained in Section 5.5.

5.2 CLASSIFICATION OF TURNING OPERATIONS ASSOCIATED WITH A CNC LATHE

TECHTURN [29] has classified ***the machining processes which can be performed on a CNC lathe*** into eight broad groups;

1) Forward and backward turning,

2) Forward and backward boring,

3) In and out facing,

4) External and internal grooving,

5) External and internal threading,

6) Drilling,

7) Reaming,

8) Centre drilling.

Group 1 to group 3 are considered as the "basic turning operations" and the remaining are called "the auxiliary turning operations". The forward and backward turning includes external processes of cylindrical, conical, profile, and recessing operations in left hand (LH) and right hand (RH). Moreover, the forward and backward boring involves internal operations of those in the forward and backward turning. However, in this case, the right hand movements are limited due to possibility of tool-fixture collisions. Facing is considered to be the first operation in any turning application. It is performed in order to establish a reference end position for axial dimensions [30, 31]. The *In facing* denotes a facing operation in which cutting is from the outside to inward. For out facing, the cutting is vice-versa. A recess is usually machined by two (or more than two) tools of opposite hands, whereas a groove is by just one tool [32, 33].

In this study, the machining operations are denoted by simple abbreviations. The first character states whether if it is an internal or external process. The next three characters stand for the abbreviation of machining operation. The extension expresses the hand of the cutting tool. A complete list of turning operations and their acronyms used in this thesis are given in Table 5.1.

Table 5.1 The turning operations and their acronyms used in this thesis

NO	ACRONYM	THE TURNING OPERATION
1	ECNC_LH	External conical turning left-hand
2	ECNC_RH	External conical turning right-hand
3	ECYL_LH	External cylindrical turning left-hand
4	ECYL_RH	External cylindrical turning right-hand
5	EFCN_LH	External facing left-hand
6	EGRV_LH	External grooving left-hand
7	EPRF_LH	External profile turning left-hand
8	EPRF_RH	External profile turning right-hand
9	ERCS_LH	External recess turning left-hand
10	ERCS_RH	External recess turning right-hand
11	ETHR_LH	External threading left-hand
12	ETHR_RH	External threading right-hand
13	ICNC_LH	Internal conical turning left-hand
14	ICNC_RH	Internal conical turning right-hand
15	ICYL_LH	Internal cylindrical turning left-hand
16	IGRV_LH	Internal grooving left-hand
17	IPRF_LH	Internal profile turning left-hand
18	IPRF_RH	Internal profile turning right-hand
19	IRCS_LH	Internal recess turning left-hand
20	IRCS_RH	Internal recess turning right-hand
21	ITHR_LH	Internal threading left-hand
22	ITHR_RH	Internal threading right-hand
23	IDRL	(Internal) drilling
24	IRMN	(Internal) reaming

5.3 SPECIFICATIONS OF SELECTED MACHINE TOOL AND CONTROL

5.3.1 Machine capacity and properties

The Boxford 250B is an ideal CNC lathe for flexible batch production of small components and second operation work. It has a 3 HP AC spindle motor. Spindle speeds are programmable and variable from 100 to 3500 rpm. The rapid traverse feed rate is 3 m/min. Feed rates for cutting can be programmed from 10 to 999 mm/min. The X axis and Z axis travels are limited to 200 mm and the maximum spindle bore diameter is 35 mm. The system resolution is 0.0025 mm for X axis, and 0.0050 mm for Z axis.

The machine is fitted with the standard Boxford machine tool control. The most of G words and canned cycles of roughing, finishing, deep hole drilling, and threading are implemented on the system. However, the G words for tool nose compensation are not included. The part programs can be loaded to the machine through RS 232 serial interface as well as from the keyboard.

5.3.2 Program Format

Input of the program is via control system and is in tabular format. A typical format of the part program for the Boxford 250 lathe is shown below [34].

LINE G M X Z I K F S

Where:

LINE	is block number
G	is preparatory code
M	is miscellaneous function code
X	is X co-ordinate to change depth of cut
Z	is Z co-ordinate measured along the axis of the billet
I	is an additional information as needed
K	is an additional information as needed
F	is the feed rate
S	is the spindle speed

5.3.3 G and M codes supported by Boxford 250 Lathe

The G and M functions implemented on the Boxford 250 CNC lathe are as follows [34]:

a) G codes:

CODE NO	*for FUNCTION*
G00	*Rapid traverse positioning*
G01	*Linear interpolation*
G02	*Circular interpolation(clockwise)*
G03	*Circular interpolation(counter clockwise)*
G04	*Time dwell*
G50	*Set maximum spindle speed*
G70	*Programmable inch dimensions*
G71	*Programmable metric dimensions*
G80	*Canned cycle-cancel*
G81	*Canned cycle-roughing diameter*
G82	*Canned cycle-facing or grooving*
G83	*Canned cycle-peck drilling*

G84	*Canned cycle-thread cutting*
G90	*Absolute data input*
G91	*Incremental data input*
G94	*Feed/min*
G95	*Feed/rev*
G96	*Constant surface speed*
G97	*Constant rpm.*

b) M codes:

CODE NUMBER	*for FUNCTION*
M02	*Marks the end of the program*
M03	*Starts spindle forward. Enter spindle speed S.*
M04	*Starts spindle reverse. Enter spindle speed S.*
M05	*Stops spindle.*
M06	*Tool change. Enter I = tool type number (1 - 8)*
M08	*Coolant ON*
M09	*Coolant OFF*
M23	*Corner rounding ON*
M24	*Corner rounding OFF*
M30	*Marks the end of the program repeated quantities.*
M39	*Closes automatic chuck*
M40	*Opens automatic chuck*
M99	*Continuation code*

5.3.4 Further explanations of G codes

5.3.4.1 G00 - Rapid Movement

This code moves the tool along the X axis and then the Z axis to the specified X and Z values as quickly as possible. It is not used for cutting metals. The machine tool control has a default value of 00 in the G column for every new program.

5.3.4.2 G01 - Linear Interpolation

This code moves the cutter at the specified feed rate in a straight line along each axis or diagonally from its present position to the new X and Z values.

5.3.4.3 G02 - Clockwise Circular Interpolation

This code moves the tool clockwise in a circular arc in a single quadrant. The following extra information should be entered in the block with G02.

X value = the diameter to which movement to be made.

Z value = the Z co-ordinate to which movement to be made.

I value = the distance from the arc centre(incremental)to the tool position at the start of the cut along the X axis.

K value = The distance from the arc centre (incremental)to the tool position at the start of the cut along the Z axis.

5.3.4.4 G03 - Counter clockwise Circular Interpolation

This code moves the tool counter clockwise in a circular arc in a single quadrant. The information entered in the block with the G03 is same as with those in the clockwise circular interpolation.

5.3.4.5 G04 - Time Dwell

This code is used when there is a need for pausing the operation. The execution can be delayed up to one hundred seconds. For instance, a G04 word may be included in a part program before an tool replacement statement, if machine tool do not have a tool turret. By this way the operator is allowed to change the tools in a certain time period.

5.3.4.6 Canned Cycles

Canned cycles may be defined as a set of reprogrammed instructions stored away in computer memory. The word "canned" has probably been borrowed from canned goods, which one usually stores away for later use. Because the instructions represented a set of routine-like repetitive patterns, the word "cycle" was found to best express what was taking place. These canned cycles are filed away under a G code address. To a large extend the G codes are standardised. G81, G82, G83, and G84 are the four of them and stand for rough cylindrical turning, facing (grooving), drilling, and threading cycles.

5.3.4.6.1 Outside Diameter - fixed cycle roughing

This code enables several cuts to be taken automatically provided that the Z axis movements are the same. The starting point in the X axis is at the 2 mm stand-off point and the tool is programmed to this location on the previous line. The following additional information is needed in the block with G81.

I value = total depth of cut (the stand off is excluded)
Z value = total length of travel in the Z axis
I value = number of cuts required
F value = feed rate

5.3.4.6.2 G82 for facing

This code enables several cuts to be taken automatically provided that the X axis movements are the same. The starting point in the Z axis is at a stand-off point and the tool is programmed to this location on the previous line. Total depth of cut, total cut width, number of cuts, and feed rate should be entered in the X, Z, I, and F columns in the block with G82, respectively.

5.3.4.6.3 G82 for grooving

This canned cycle is used to cut rectangular section of grooves having a width larger than "2 * width of the cutting tool". Before entering the canned cycle routine, it is necessary to create a slot to the required depth. This is accomplished using codes G00 and G01. After cutting this narrow groove the tool is taken to its stand-off position. The following information should be entered in the block with this code.

X value = the difference between the stand off diameter and the diameter of the bottom of the groove.

Z value = the additional width of the groove.

I value = number of cuts.

F value = feed rate.

5.3.4.6.4 G83 - Peck Drilling

This code is used to drill holes deeper than twice the drill diameter. From the stand-off point (X=0, Z=2), the first movement takes the drill in the -Z direction a distance equal to twice the drill diameter. After withdrawing from the hole in order to remove swarf, the drill returns and cuts further in -Z direction by an amount which determined by " K * first depth". This pecking is repeated until the total depth -Z is reached. Total depth of hole, drill diameter, a reduction value for pecking, and feed rate should be specified in the block with G83.

5.3.4.6.5 G84 Canned cycle for thread cutting

This code is used for both external and internal threads. For externals, the starting point for the threading cycle in the X axis is 2 mm greater than the outside diameter of the point on the diameter of the material to be cut. for internals, the stand off point is 2 mm smaller. This position is programmed on the line previous to the G84 being called. Total length from stand off, depth of thread, number of cuts, and pitch should be instated in the Z, I, K, and F columns in the block with G84.

5.4 COMPUTER PROGRAM

The computer program is written in C++ computer programming language and compiled with Borland's Turbo C++ (Version 3.1) compiler on an IBM compatible personal computer. The program is mainly composed of two sections;

- Generation module
- Simulation module

The simplified flowcharts of programs prepared for the generation and simulation modules are given in Figure 5.1.a and Figure 5.1.b, respectively.

5.4.1 The generation module

The generation module generates both part program and tool path by processing a previously prepared CAD/CAM data base[1] for a given part geometry. The CAD/CAM data base and part geometry are supplied by formatted text files. An example part and the file defining this part are given in Figure 5.2 and Table 5.2, respectively. The work done in the generation module is to prepare some machining instructions (and to determine paths in the case of tool path generation) by considering the rules of machine tool control and the geometry of workpiece to be machined.

[1]This data base is the output of a Ph. D. study on CAPP by Cengiz Kayacan (continuing at the Department of Mechanical Engineering, University of Gaziantep).

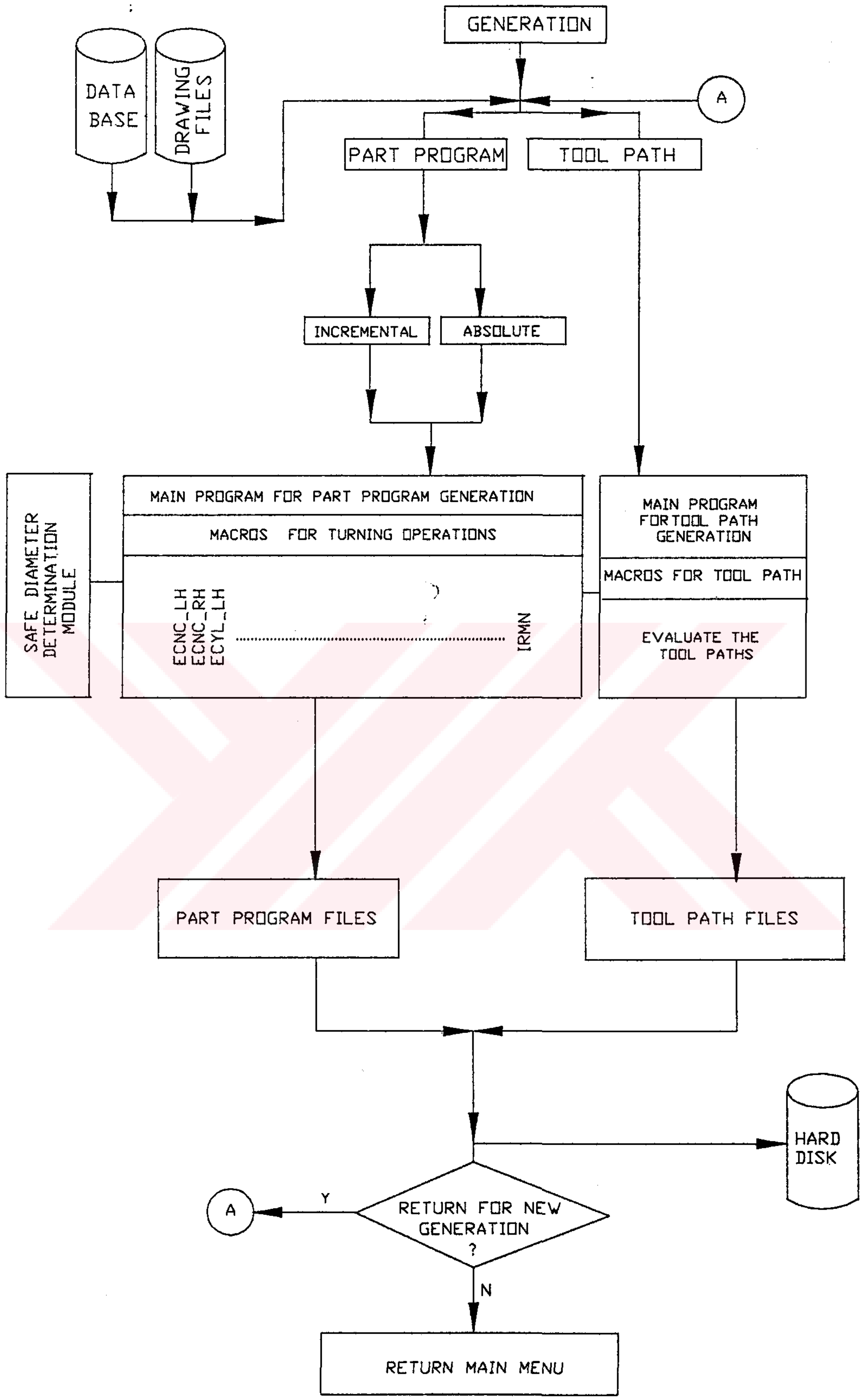


Figure 5.1.a The simplified flowchart of the generation module

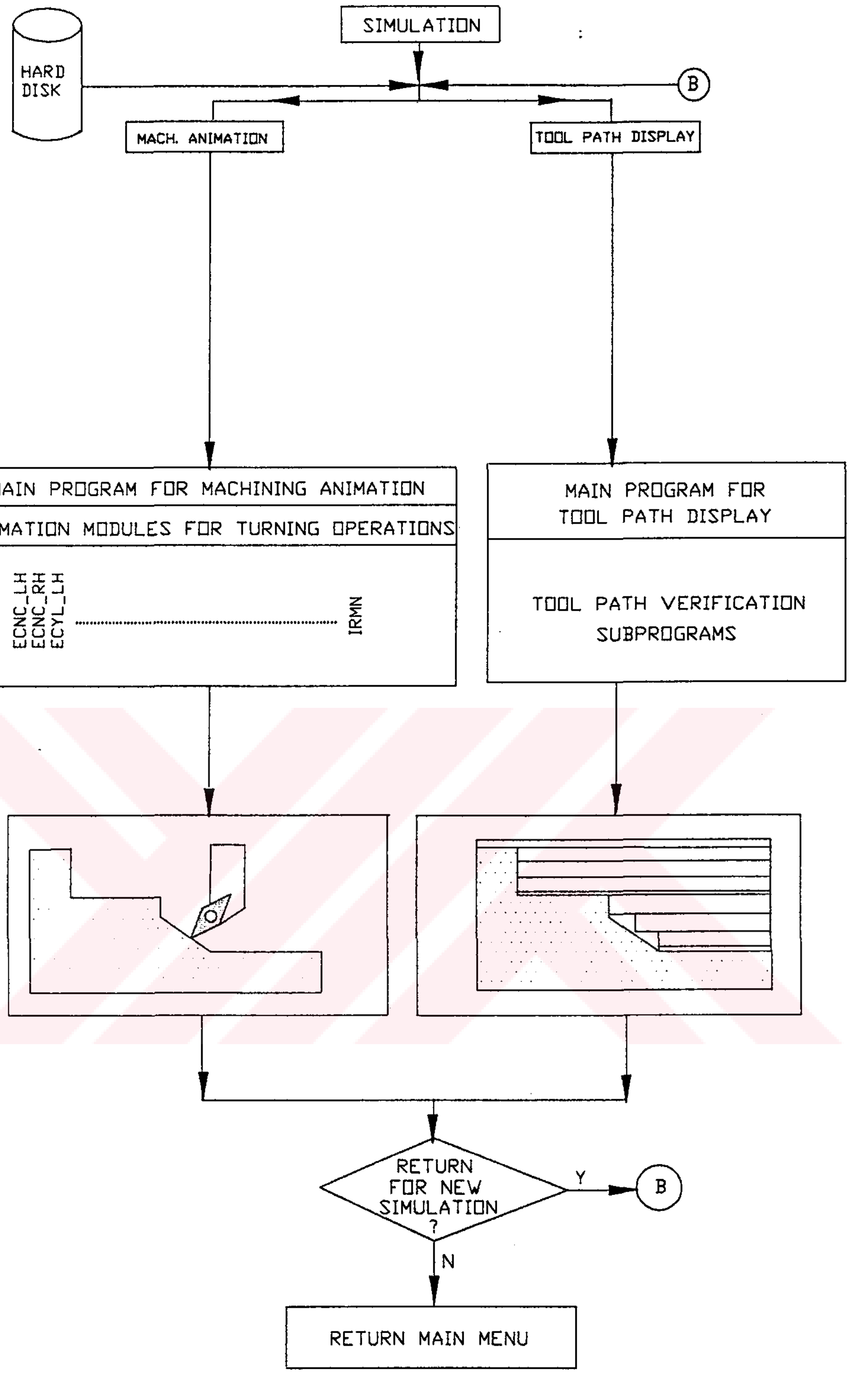
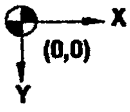


Figure 5.1.b The simplified flowchart of the simulation module



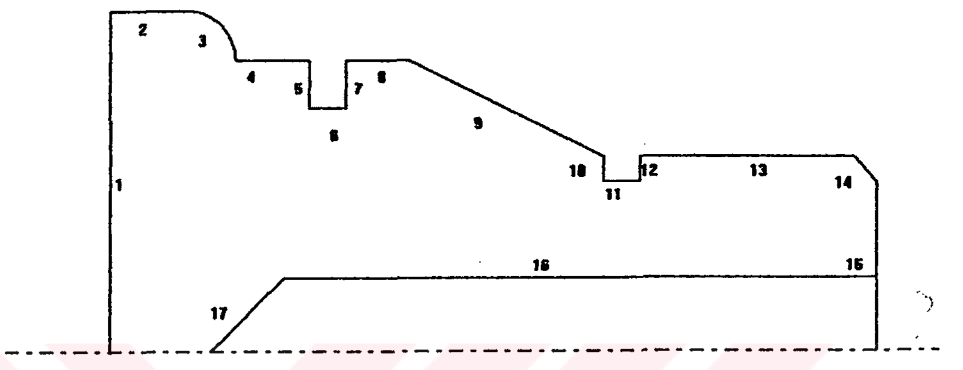


Figure 5.2 The drawing of the upper half of the part to be produced

Table 5.2 The text file obtained from the part drawing shown in Figure 5.2

DSN	X1	Y1	X2	Y2	RAD
1	30	240	30	130	0
2	30	130	50	130	0
3	50	100	0	90	30
4	80	160	100	160	0
5	100	160	100	180	0
6	100	180	110	180	0
7	110	180	110	160	0
8	110	160	120	160	0
9	120	160	160	170	0
10	160	170	160	180	0
11	160	180	170	180	0
12	170	180	170	170	0
13	170	170	210	170	0
14	210	170	220	180	0
15	220	180	220	240	0
16	220	230	100	230	0
17	100	230	90	240	0

The X1, Y1 and X2, Y2 in the drawing file of the part as shown in Table 5.2 stand for the starting and finishing coordinates of contours defining the part surface. However, in the case of arcs, these numbers denote the centre coordinates, start angle and end angle, respectively. The RAD defines the radius of arc.

The CAD/CAM data base for the given part shown in Figure 5.3 consists of all the information necessary for machining as tabulated in Table 5.3. It includes operation numbers (no), and operation types (opr), geometry definitions (x1, y1, x2, y2, mx, my, sa, ea, rad), feeds in mm/min (f), speeds in RPM (spd), depth or width of cut (doc), tooling information (t) and surface number (sn). The x1, y1 and x2, y2 in geometry definitions define the left-top and right-bottom coordinates of the area to be machined for cylindrical, face, recess, and groove turning operations. For all others, these point out the starting and finishing coordinates of the cutting operation. The next five columns stand for the centre coordinates (mx, my), start angle, end angle, and radius of arc, respectively.

For each turning operation listed in Table 5.1, *macros containing all CNC statements required to perform the machining* are written. As an example, the macro prepared for the external cylindrical turning (ecyl_lh) is given below.

```
/*******************************************************************************************/
                                        gen_ecyl_lh( )
/*******************************************************************************************/
{
eksor=1;
radior=1;

/*————————————— LEVEL ONE ——————————————————————*/

if(nrpx==trpx && nrpy==trpy)
{
 fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod,kod6,NULL,NULL,2,2,NULL,NULL);
```

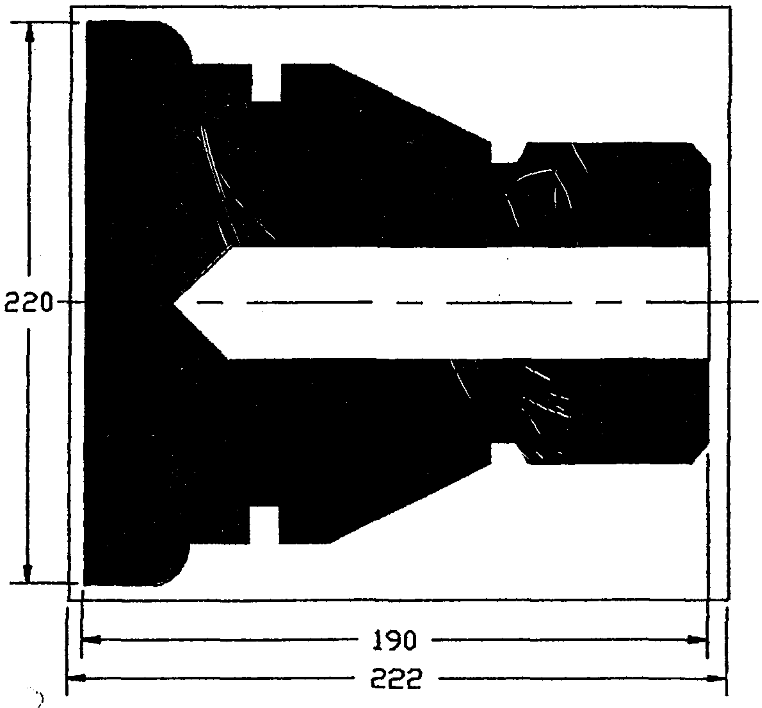


Figure 5.3 The complete view of the component to be produced

Table 5.3 The CAD/CAM data base for the example part

no	opr	x1	y1	x2	y2	mx	my	sa	ea	rad	t	doc	sn	f	spd
1	efcn_lh	220	120	222	240	0	0	0	0	0	5	2	0	60	3000
2	ecyl_lh	31	120	220	130	0	0	0	0	0	2	2	0	60	3000
3	ecyl_lh	68	130	220	132	0	0	0	0	0	2	2	0	60	3000
4	ecyl_lh	68	132	220	134	0	0	0	0	0	2	2	0	60	3000
5	ecyl_lh	74	134	220	136	0	0	0	0	0	2	2	0	60	3000
6	ecyl_lh	74	136	220	138	0	0	0	0	0	2	2	0	60	3000
7	ecyl_lh	78	138	220	140	0	0	0	0	0	2	2	0	60	3000
8	ecyl_lh	78	140	220	142	0	0	0	0	0	2	2	0	60	3000
9	ecyl_lh	80	142	220	160	0	0	0	0	0	2	3	0	58	3000
10	ecyl_lh	144	160	220	163	0	0	0	0	0	2	3	0	58	3000
11	ecyl_lh	144	163	220	166	0	0	0	0	0	2	3	0	58	3000
12	ecyl_lh	160	166	220	168	0	0	0	0	0	2	2	0	60	3000
13	ecyl_lh	160	168	220	170	0	0	0	0	0	2	2	0	60	3000
14	ecnc_lh	120	160	160	170	0	0	0	0	0	25	0	9	57	2950
15	eprf_lh	50	130	80	160	50	160	0	90	30	25	0	3	57	2950
16	egrv_lh	160	170	170	180	0	0	0	0	0	28	0	11	53	2850
17	egrv_lh	100	160	110	180	0	0	0	0	0	28	0	6	53	2850
18	ethr_lh	170	170	220	170	0	0	0	0	0	6	1.23	0	1	700
19	idrl	90	240	220	130	0	0	0	0	0	22	0	0	50	2000

```
fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod,kod4,NULL,NULL,NULL,NULL,NULL,
spd[on]);

 if((y2[on]-y1[on]) == doc[on])
  {
  xyolu=y2[on]-trpy;
  zyolu=trpx-x2[on]-eksor;
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,-xyolu,
  -zyolu,NULL,NULL,NULL,NULL);
  kesz=x2[on]-x1[on]+eksor;
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod1,kod,NULL,
  -kesz,NULL,NULL,feed[on],NULL);
  }

 if((y2[on]-y1[on]) > doc[on])
  {
  noc=(y2[on]-y1[on])/doc[on];
  xcyolu=y1[on]-trpy-radior;
  zcyolu=trpx-x2[on]-eksor;
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,-xcyolu,
  -zcyolu,NULL,NULL,NULL,NULL);
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod81,kod,(y2[on]-y1[on]),
  -(x2[on]- x1[on]+eksor),noc,NULL,feed[on],NULL);
  }
}


if(nrpx < trpx || nrpy > trpy)
{

 if((y2[on]-y1[on]) == doc[on])
  {

        if( nrpx < x2[on] )
        {
        xyolu=radior;
        zyolu=(x2[on]-nrpx)+eksor;
        fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,xyolu,zyolu,NULL,NULL,
         NULL,NULL);
        xyolu=(y2[on]-y1[on])+radior;
        fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,
         -xyolu,NULL,NULL,NULL,NULL,NULL);
        }

        if( nrpx > x2[on] )
        {
        xyolu=y2[on]-nrpy;
        fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,
         xyolu,NULL,NULL,NULL,NULL,NULL);
        }
  kesz=(x2[on]-x1[on])+eksor;
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod1,kod,NULL,
  -kesz,NULL,NULL,feed[on],NULL);
  }

  if((y2[on]-y1[on]) > doc[on])
  {
  noc=(y2[on]-y1[on])/doc[on];
        if( nrpx < x2[on] )
        {
        xcyolu=radior;
        zcyolu=(x2[on]-nrpx)+eksor;
        fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,xcyolu,zcyolu,NULL,NULL,
```

```
 NULL,NULL);
 }

 if( nrpx > x2[on] )
 {
 xcyolu=(y1[on]-nrpy)-radior;
 fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,
  -xcyolu,NULL,NULL,NULL,NULL,NULL);
 }

 fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod81,kod,(y2[on]-y1[on]),
 -(x2[on]-  x1[on]+eksor),noc,NULL,feed[on],NULL);
 }
}

/*--------------- LEVEL TWO ------------------------------*/

if(tot[on+1] == tot[on])
{

  if((y2[on]-y1[on]) == doc[on]){ nrpx=x1[on]; nrpy=y2[on] ;}

  if((y2[on]-y1[on]) >  doc[on]){ nrpx=x2[on]+eksor;
                                   nrpy=y1[on]-radior;
                                 }
}

if(tot[on+1] != tot[on])
{
  if((y2[on]-y1[on]) == doc[on]){ gdonx=y2[on]-trpy;
                                  gdonz=trpx-x1[on];
                                  }
  if((y2[on]-y1[on]) >  doc[on]){ gdonx=y1[on]-trpy-radior;
                                  gdonz=trpx-x2[on]-eksor;
                                  }

  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod0,kod,gdonx,gdonz,NULL,NULL,NULL,
  NULL);

  nrpx=trpx;
  nrpy=trpy;
  fprintf(stream,"%5s%5s%5d%5d%5d%5d%5d%5d\n",kod,kod5,NULL,NULL,NULL,NULL,NULL,
  NULL);
}
return 0;
}

/*************************************************************************************/
```

Each macro has basically two levels for determination of the machining instructions as illustrated in Figure 5.4. In the first level, the set of instructions are determined by the location of the cutting tool. If any tool is at tool reference point (TRP), it is simply positioned for cutting after some-

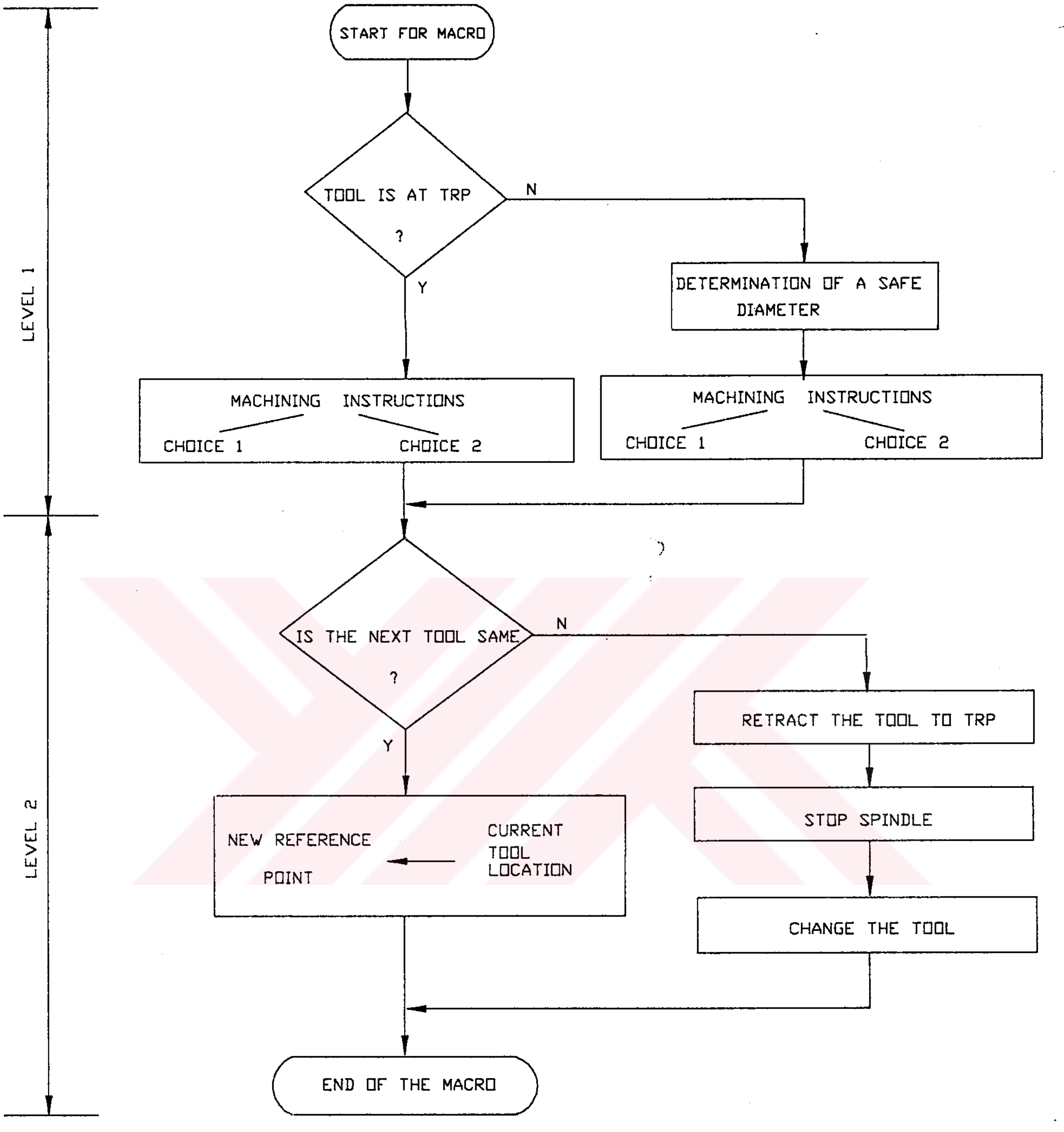


Figure 5.4 The levels in the determination of machining instructions

functions and then machining is started with the related G word. If the tool is not at TRP, the maximum intermediate and minimum intermediate diameter of the component between the current and target co-ordinates become important to prevent tool collisions for external and internal operations, respectively. A safe diameter is first determined by means of "*Safe diameter determination sub-program*" for the retraction of the tool. In this sub-program, the maximum or minimum diameters between the specified co-ordinates are found by checking the part geometry and considering the type of operations. After the tool is brought to its new location (with movements of m1 and m2) to prevent any tool collision with the workpieces shown in Figure 5.5, the next program blocks for positioning and cutting (the movements of m3, m4, and cutting from 3 to 4) are appended to the part program file opened for the workpiece. For internal operations, the safe diameter can generally be accepted as the centre-line or the minimum intermediate diameter can be found in a similar way as above.

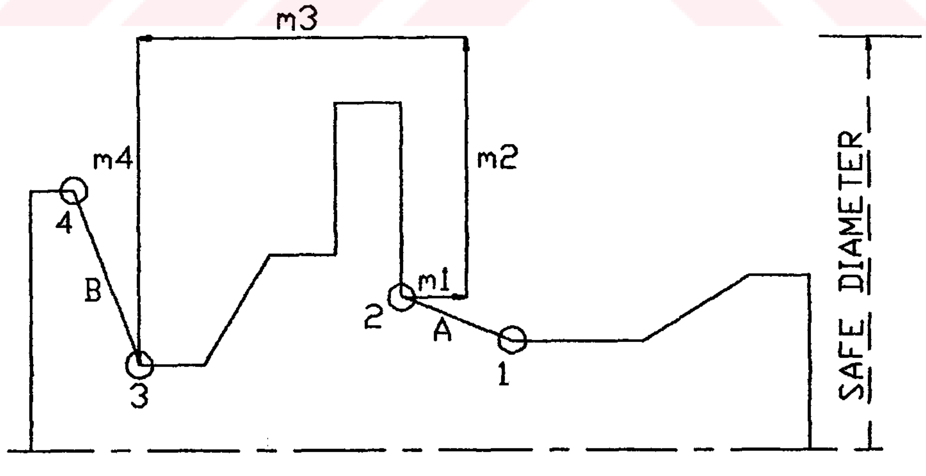


Figure 5.5 The determination of the safe diameter.

In the second level, the tooling information plays a critical role in the determination of the machining commands. If the subsequent operation requires the same tool, the current location is identified as the new reference point of the tool. No machining instructions are given before exiting the macro. On the other hand, if the next tool is different; the tool is first retracted to the tool changing position with rapid traverse commands and the spindle is stopped by calling M05. The tool is then changed as requested in the data base.

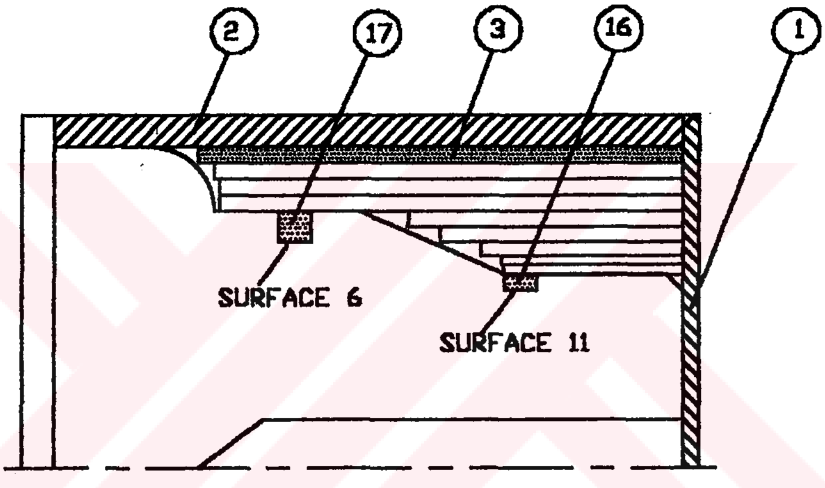


Figure 5.6 The areas to be machined.

The generation process is continued by scanning the data base starting from the first operation and adding the CNC statements into a part program file. For the data base given in Table 5.3; the first operation is declared as the external facing. The x1, y1 (220, 120) and x2, y2 (222, 240) are the upper-left and right-bottom corner coordinates of the area (1) to be machined as shown in Figure 5.6. The total width of the area is calculated by "x2-x1" and is found to be 2 mm. Since the allowable depth of cut given in the "doc" column is also 2 mm, the macro determines that this operation is going to be performed with a linear interpolation (G01). After the positioning statements are given, the motion command is appended to the program. X axis movement is calculated from; [-(y2-y1)+1mm stand-off] and it is equal to -121. The tool used in the next operation is then controlled. Since the next tool is (tool-2) being different from the current tool (tool-5), the necessary statements for retracting the tool to TRP and stopping of the spindle are defined. The CNC statements for this operation is described from line 30 to line 90 in the part program listed in Table 5.4. Then, number of operation (no) is increased by one and the determination of CNC codes for the second operation is started.

The second operation is the external cylindrical turning with a left hand tool. The x1, y1 and x2, y2 again stand for the upper-left and right bottom coordinates of the area (2) to be machined as shown in Figure 5.6. In this case, however, total depth of cut (10 mm) is not equal to allowable depth of cut (2 mm) given in the same row of the data base. Therefore, the ecyl_lh macro immediately determines that this operation is going to be performed with a canned cycle (G81) and calculates the number of cuts by dividing the total depth of cut to allowable depth of cut. The Z axis movement is calculated by; [-(x2-x1) + 1 mm stand-off], and is equal to -190. After the cutting commands are specified, the next tool in the data base is controlled. Since the third operation uses the same tool (tool-2), the current location of the tool which is the starting point of the canned cycle, is identified as the reference point for subsequent operations. No other instructions are given in this case before exiting the macro. The CNC codes for this operation is described from line 100 to line 130 in the part program listed in Table 5.4. The third operation is also external cylindrical turning. However, in this case G01 is used to cut the specified machining area (3) shown in Figure 5.6, since the total depth of cut and allowable depth of cut given in the third row of data base are equal to each other. (See line 150 in Table 5.4).

For some successive operations of conical, profiling, and grooving using the same cutting tool, it is necessary to calculate a diameter between the respective surfaces for the retraction of the tool. For example, sixteenth operation (egrv_lh) finishes with the specification of the reference point for the seventeenth operation. This point is the left-bottom corner of the groove (16) shown in Figure 5.6. In the seventeenth operation, the tool must first be positioned for cutting at a stand-off location which is 1 mm away from the upper-left corner of the groove (16). In order to determine the safe diameter between the surfaces 11 and 6, the safe diameter determination sub-program is called by the grooving macro, and the diameter is found by searching the drawing file of the part tabulated in Table 5.2 between the related surfaces. In this case, the surface 8 is found to be as the entity having the maximum y coordinate (160). Subtracting 1 mm stand-off value from this yields 159. Since the surface 11 has the y coordinate of 180, 180 minus 159 gives 21. Similarly, subtracting the x coordinate of the left corner of the surface 6 (100 mm) from the x coordinate of the left corner of the surface 11 (160 mm), the Z axis movement for positioning is obtained as 60 mm. Then the tool is positioned for cutting the groove (operation 17) by the block; "G00 X21 Z-60".

The process is terminated after all operations in the data base is scanned. The incremental and absolute part programs generated for the master part are listed in Table 5.4 and Table 5.5, respectively. More examples of part programs generated for different part geometries are given in the Appendix.

Table 5.4 The list of part program (incremental) generated for example part

N	G	M	X	Z	I	K	F	S
10	91							
20	71							
30		06			5	1		
40		04						3000
50	00		-79	-120				
60	01		-121				60	
70	00			120				
80	00		200					
90		05						
100		06			2	2		
110		04						3000
120	00		-79	-119				
130	81		10	-190	5		60	
140	00		-13					
150	01			-153			60	
160	00		1	153				
170	00		-3					
180	01			-153			60	
190	00		1	153				
200	00		-3					
210	01			-147			60	
220	00		1	147				
230	00		-3					
240	01			-143			60	
250	00		1	147				
260	00		-3					
270	01			-143			60	
280	00		1	143				
290	00		-3					
300	01			-143			60	
310	00		1	143				
320	81		18	-141	6		58	
330	00		-22					
340	01			-77			58	
350	00		1	77				
360	00		-4					
370	01			-77			58	
380	00		1	77				
390	00		-3					
400	01			-61			60	
410	00		1	61				
420	00		-3					
430	01			-61			60	
440	00		130	180				
450		05						
460		06			25	3		
470		04						2950
480	00			-179				
490	00		-130	-1				
500	01		10	-40			57	
510	00			1				
520	00		1	-40				
530	00		-1	-1				
540	03		30	-30		30	57	
550	00		90	290				
560		05						
570		06			4	4		

580		04						2950
590	00			-180				
600	00		-129					
610	01		-11				53	
620	00		21	-60				
630	01		-21				53	
640	00		140	240				
650		05						
660		06			5	5		
670		04						700
680	00		-129	-115				
690	84			-55	1.23	10	1	
700	00		129	115				
710		05						
720		06			22	6		
730		03						2000
740	00		-200	-118				
750	83			-130	20	0.9	50	
760	00			118				
770	00		200					
780		05						
790		30						

Table 5.5 The list of part program (absolute) generated for the example part

N	G	M	X	Z	I	K	F	S
10	90							
20	71							
30			400	120	5	1		
40		04						3000
50	00		242	0				
60	01		0				60	
70	00			120				
80	00		400					
90		05						
100		06			2	2		
110		04						3000
120	00		242	1				
130	81		20	-190	5		60	
140	00		216	1				
150	01			-152			60	
160	00		218	1				
170	00		212					
180				-152			60	
190	00		214	1				
200	00		208					
210	01			-146			60	
220			210	1				
230			204					
240				-146			60	
250	00		206	1				
260	00		200					
270	01			-142			60	
280			202	1				
290			196					
300				-142			60	
310	00		198	1				
320	81		36	-141	6		58	
330			154	1				

340	01			-76			58	
350	00		156	1				
360	00		148					
370	01			-76			58	
380	00		150	1				
390	00		144					
400	01			-60			60	
410	00		146	1				
420	00		140					
430	01			-60			60	
440	00		400	120				
450		05						
460		06			25	3		
470		04						2950
480	00		400	-59				
490	00		140	-60				
500	01		160	-100			57	
510	00			-99				
520	00		162	-139				
530	00		160	-140				
540	03		220	-170		30	57	
550	00		400	120				
560		05						
570		06			4	4		
580		04						2850
590	00		400	-60				
600	00		142					
610	01		120				53	
620	00		162	-120				
630	01		120				53	
640	00		400	120				
650		05						
660		06			5	5		
670		04						700
680	00		142	5				
690	84			-55	1.23	10	1	
700	00		400	120				
710		05						
720		06			22	6		
730		03						2000
740	00		0	2				
750	83			-130	20	0.9	50	
760	00		0	120				
770	00		400					
780		05						
790		30						

In the tool path generation, the method and logic used are almost the same as that in the generation of the part programs. In this case, the information about the tooling, speed, feed etc., in the data base has no importance. Only the X and Z axis movements of the tools involved in machining of the workpiece are calculated by means of the tool path generation macros prepared for the turning processes as tabulated in Table 5.1. The generation is performed by processing the data base prepared for a given part. The X and Z movements are calculated according to type of operations and appended into a 'pth' extension file. The entire tool path generated for the example part is listed in Table 5.6. It can easily be seen from that file that the X and Z axis codes are similar to those in the part program files for one-cut operations and rapid traverse movements. However, for multi-cut operations requiring canned cycles (G81, G82, G83, G84), the actual path which the tool follows while machining the workpiece is also determined. Following is the macro prepared for the tool path generation of facing operation.

```
/*****************************************************************************/
  pth_efcn_lh( )
/*****************************************************************************/
{
eksor=1;
radior=1;

if((x2[on]-x1[on]) == doc[on])
 {
  xyolu=y1[on]-trpy-radior;
  zyolu=trpx-x1[on];
  fprintf(stream,"%5s%5d%5d\n",kod0,xyolu,-zyolu);
  kesx=y2[on]-y1[on]+radior;
  fprintf(stream,"%5s%5d%5d\n",kod1,kesx,NULL);
  gdonx=y2[on]-trpy;
  gdonz=trpx-x1[on];
  fprintf(stream,"%5s%5d%5d\n",kod0,NULL,gdonz,);
  fprintf(stream,"%5s%5d%5d\n",kod0,-gdonx,NULL);
 }

if((x2[on]-x1[on]) > doc[on])
 {
  noc=(x2[on]-x1[on])/doc[on];
  xcyolu=y1[on]-trpy-radior;
  zcyolu=trpx-x2[on]-eksor;
  fprintf(stream,"%5s%5d%5d%5d\n",kod0,xcyolu,-zcyolu);

     kessay=1;
     while (kessay < noc)
     {
     fprintf(stream,"%5s%5d%5d\n",kod82,NULL,-(doc[on]+eksor));
     fprintf(stream,"%5s%5d%5d\n",kod82,(y2[on]-y1[on]+radior),NULL);
     kessay=kessay+1;
```

```
        fprintf(stream,"%5s%5d%5d\n",kod82,NULL,eksor);
        fprintf(stream,"%5s%5d%5d\n",kod82,-(y2[on]-y1[on]+radior),NULL);
        }
        fprintf(stream,"%5s%5d%5d\n",kod82,NULL,-(doc[on]+eksor));
        fprintf(stream,"%5s%5d%5d\n",kod82,(y2[on]-y1[on]+radior),0);
        fprintf(stream,"%5s%5d%5d\n",kod82,NULL,(x2[on]-x1[on]+eksor));
        fprintf(stream,"%5s%5d%5d\n",kod82,-(y2[on]-y1[on]+radior),NULL);

    gdonx=xcyolu;
    gdonz=zcyolu;
    fprintf(stream,"%5s%5d%5d\n",kod0,-gdonx,gdonz);
   }
 nrpx=trpx;
 nrpy=trpy;

 return 0;
 }
 /****************************************************************************/
```

Table 5.6 The tool path file generated for the part shown in Figure 5.2

G CODE	X	Z
00	-79	-120
01	-121	0
00	0	120
00	200	0
00	-79	-119
81	-3	0
81	0	-190
81	1	0
81	0	190
81	-3	0
81	0	-190
81	1	0
81	0	190
81	-3	0
81	0	-190
81	1	0
81	0	190
81	-3	0
81	0	-190
81	1	0
81	0	190
81	-3	0
81	0	-190
81	11	0
81	0	190
00	-13	0
01	0	-153
00	1	153
00	-3	0
01	0	-153
00	1	153

00	-3	0
01	0	-153
00	1	153
00	-3	0
01	0	-147
00	1	147
00	-3	0
01	0	-147
00	1	147
00	-3	0
01	0	-143
00	1	143
00	-3	0
01	0	-143
00	1	143
81	-4	0
81	0	-141
81	1	0
81	0	141
81	-4	0
81	0	-141
81	1	0
81	0	141
81	-4	0
81	0	-141
81	1	0
81	0	141
81	-4	0
81	0	-141
81	1	0
81	0	141
81	-4	0
81	0	-141
81	1	0
81	0	141
81	-4	0
81	0	-141
81	19	0
81	0	141
00	-22	0
01	0	-77
00	1	77
00	-4	0
01	0	-77
00	1	77
00	-3	0
01	0	-61
00	1	61
00	-3	0
01	0	-61
00	130	180
00	0	179
00	-130	-1
01	10	-40
00	0	1
00	1	-40
00	-1	-1

03	30	-30
00	90	290
00	0	-180
00	-129	0
01	-11	0
00	21	-60
00	0	0
01	-21	0
00	140	240
00	-129	-115
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	1	0
84	0	55
84	-2	0
84	0	-55
84	11	0
84	0	55
00	129	115
00	-200	-118
83	0	-42
83	0	42
83	0	-62
83	0	62
83	0	-92

83	0	92
83	0	-132
83	0	132
00	0	118
00	200	0

5.4.2. The simulation module

In Computerised Numerical Control, the part program is the only key providing the interrelation between the man and machine. When the instructions are supplied to the machine, the controller unit reads and interprets the set of commands, and executes them. That is, some actions had been already taught to the machine, and they have been kept in the memory. These pre-taught functions (G and M functions, etc.) determine the machining capability of the machine tool. Although there are many standard functions, it is not a necessity for a machine to have all these functions. For example, a machining centre may have all G words, while another may be capable of performing only a few function. On the other hand, it is obvious that the higher number of those pre-taught functions, the more intelligent the CNC machine is. In this study, all the actions performed by the use of G and M words on the Boxford 250 CNC lathe which are summarised in the previous sections have been defined by preparing graphical simulation sub-programs for each. The sub-programs are stored in the memory and ready for the execution. Hence, an imaginary CNC lathe is installed on the computer.

The simulation module prepared for this purpose is mainly of two modes;

- Machining animation
- Tool path display

The absolute and incremental part programs, and the tool paths generated in the generation module are verified by means of this module, prior to the actual machining. The part programs and cutter locations which are provided by the formatted text files are the sources for the simulation process.

In the machining animation, the machining processes, rapid traverse movements or tool replacements are animated on the screen of the computer in accordance with the CNC statements generated previously. For each turning operation listed in Table 5.1, a simulation macro is prepared by using the C++ environment. The macros prepared for cutting operations are capable of animating particular machining processes, and include the words of G01, G02, G03, G81, G82, G83, and G84. To animate a chip removal process; the area of the material being machined is painted in the background colour of screen, while the cutting tool is repeatedly redrawn along a tool path according to the G code. As an example, the macro prepared for the simulation of external cylindrical turning with G01 is given below.

```
/********************************************/
ext_cyl_lh_simulation_with_01( )
/********************************************/

{
 tox=getx()+z[ps];
 toy=gety()-x[ps];

                yy=gety()-20;
                xx=getx();
```

```
        while (xx > tox)
        {
        xx=xx-1;
        putimage(xx,yy,tool,XOR_PUT);
        delay(delay_time);
        putimage(xx,yy,tool,XOR_PUT);
        setfillstyle(SOLID_FILL,15);
        bar(xx,yy,getx(),gety());
        }
        putimage(xx,yy,tool,XOR_PUT);
        putimage(xx,yy,tool,XOR_PUT);

moveto(tox,toy);
return 0;
}
```

In the simulation of G00 (rapid traverse positioning), only the cutting tool is repetitively redrawn on the screen between two coordinates. The works done by M words are simulated by just high-lighting a switch in the machine tool status area on the monitor. However, the simulation module for M06 also animates the tool replacement on the tool reference point.

The animation process is performed simply by scanning the generated part program files and determining the characteristic of each block. In the part program given in Table 5.4, the first two lines stand for the incremental and metric system selections. For these, respective switches are illuminated in the machine tool status area. Upon executing the third line, tool-5 is drawn at the tool reference point. Line 40 illuminates the switch "spindle reverse" on the machine tool status area. When the fifth row of the part program is scanned, the cutting tool is rapidly moved from current position to the target position which is 79 mm away in the - X axis, and 121 mm away in the -Z axis from the TRP, on the screen. Line 60 contains G01 and X-121. Since the cutting is required only in the X axis and the tool selected for this operation (tool-5) is used for the facing operation, the efcn_lh macro is called. This macro takes the X word as input, and simulates the facing operation on the screen. After the tool is retracted to

TRP by processing the lines 70 and 80; and spindle is stopped, the tool-2 is selected from the tool turret, and the spindle is turned by scanning the lines 90, 100, and110, respectively; *"external cylindrical turning (left hand) with G81"* macro is called for the execution. This macro takes the X, Z, I words as the input, and animates the canned cycle on the screen. The cutting is performed in five cuts as stated in the part program file. The simulation process is terminated after all lines in the part program is scanned. An example to the simulation of threading operation is illustrated in Figure 5. 7.

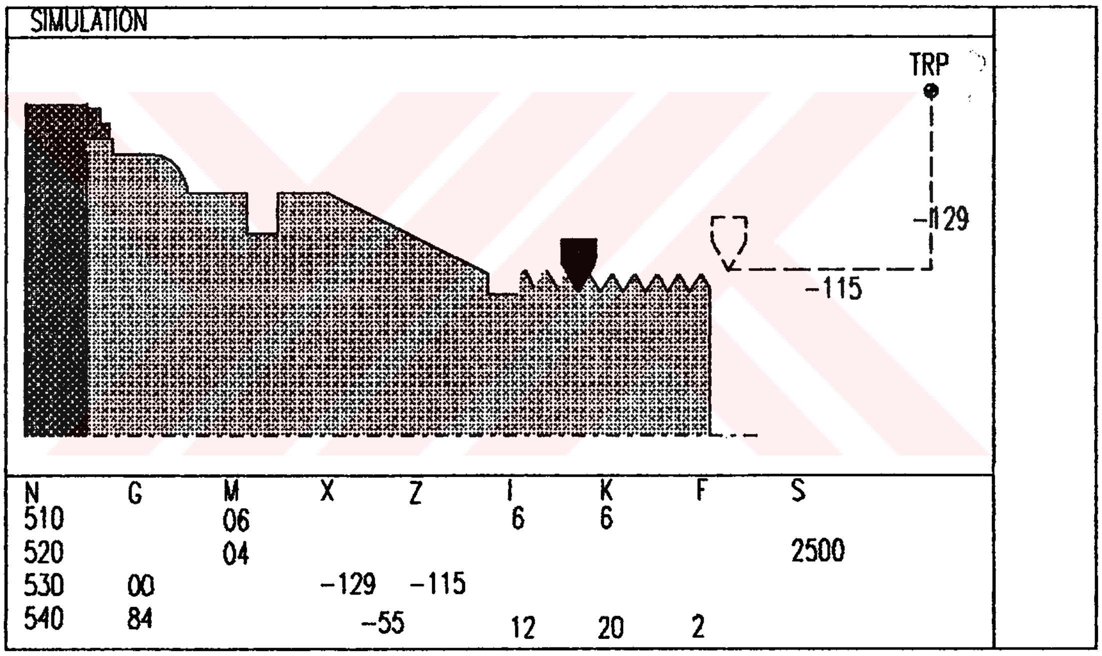


Figure 5.7 The simulation of the threading operation

The simulation of machining processes can be done as rapidly as possible or performed in a real-time basis. The former is useful for quick checking of the part programs. However, if some information about the actual machining is required, the real-time mode may be selected. In this case, the tools are moved by the feed rates given in the part program. Hence, the time elapsed in the simulation is going to be equal to that in the actual machining.

In the tool path verification mode, the cutter location files which are generated in the generation module and stored in the hard-disk are again retrieved and the path of the tools are drawn depending upon the type of the operations. The display of the rapid traverse includes the drawing of the lines parallel to X and Z axes of the lathe. For cutting commands, it may be a straight line along the X axis or Z axis for many machining processes *(like facing, grooving, recessing, cylindrical turning, drilling)*, or a diagonal line for conical turning operation, or an arc for circular interpolations. The colours of the paths for different processes or movements are changed for easy visualisation.

5.5 RUNNING OF THE PROGRAM

Upon executing the program, the user-friendly pop-up / pull down menu, whose routine is written on Clipper 3.2, is immediately displayed on the screen. The program can be used easily by selecting suitable keys from the pull down menu and giving the necessary inputs as requested.

Both the incremental and absolute part programs, and the tool paths can be generated by entering their respective pop-up menus after pulling GENERATE menu down, as shown in Figure 5.8. In each case, the user is required to give a drawing or data base file name and warned for possible errors. When the generation process is finished, the system returns automatically to the main menu.

The simulation module can be run by pulling SIMULATE menu down as shown in Figure 5.9. Either "machining animation" or "tool path verification" can be selected from that menu. A part program or tool path file name is required to start execution in the "machining animation" and "tool path verification", respectively. The verification starts with the display of the turret configuration in which all the tools called by the entered part program is fitted in their holders. A second turret set-up for the applications calling more than eight tools is also reminded to the user (Figure 5.10). The remaining tools are also displayed as shown in Figure 5.11. In the case of tool path simulation however, turret configurations are not shown, since the cutting tools have no importance in the process and are not displayed during the simulation. Figure 5.12 shows an example of tool path display. The block and the finished product shape (target part) in half section are then automatically drawn on the screen in the simulation area as shown in Figure 5.13. Pressing enter key now starts the animation. The part program blocks and machine tool status are also displayed in their corresponding screen areas. The examples for the simulation of external cylindrical turning, drilling, internal grooving and internal threading operations are given in Figure 5.14, 5.15, 5.16, and 5.17, respectively. When all the operations are completed, the information about total cutting time, total production time, tool replacement time, and total time elapsed in the traversing are displayed in the message area, as shown in Figure 5.18. The resulting part is then shown in full section (Figure 5.19). The part programs and tool paths which are stored in the text files can easily be printed through PRINT menu. By the use of HELP menu as illustrated in Figure 5.20, one can have brief information about the program format, preparatory codes, miscellaneous functions, axes directions, modes of programming, tool library and tool offsets. Figure 5.21 to 5.27 show those menu items.

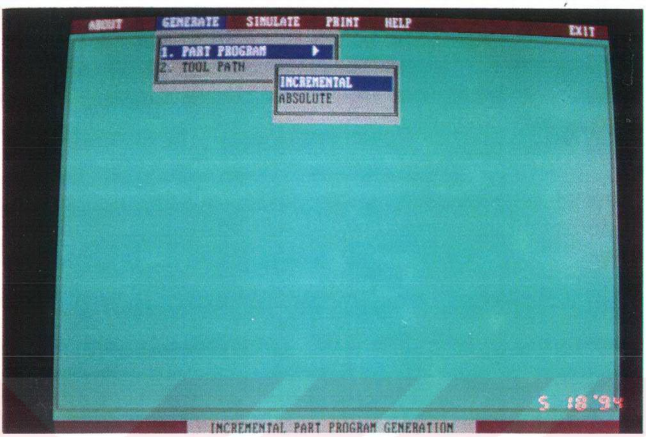


Figure 5.8 Generate Menu

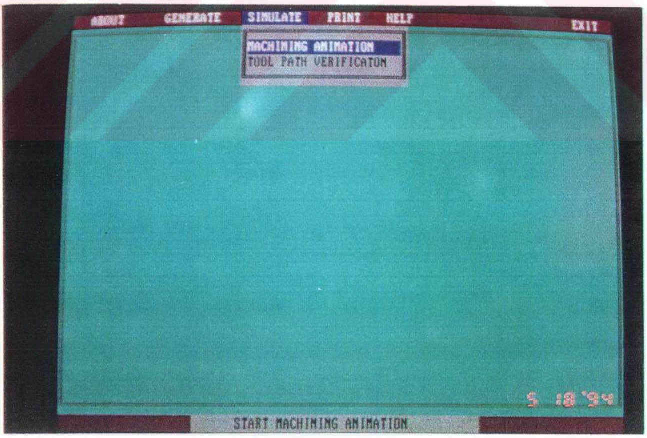


Figure 5.9 Simulate Menu

Figure 5.10 The Second Turret Set-up Call

Figure 5.11 The Turret Configuration

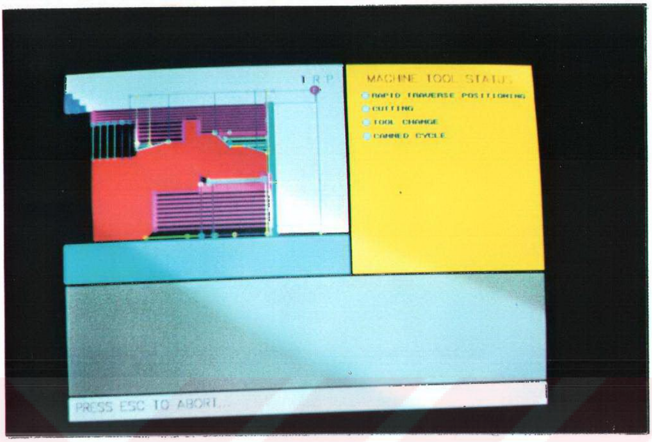


Figure 5.12 The tool path display

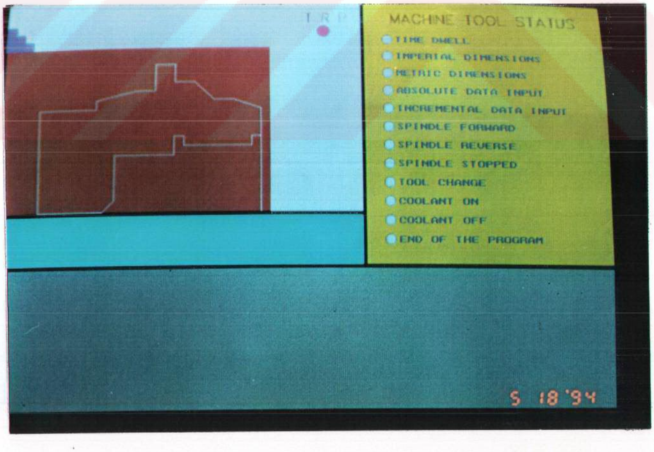


Figure 5.13 The finished part and the block

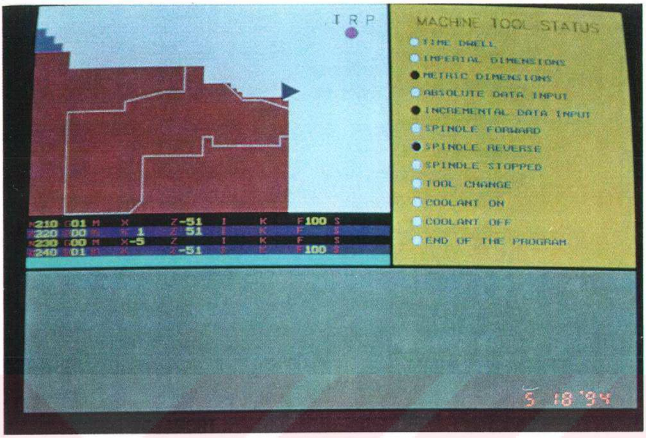


Figure 5.14 The simulation of external cylindrical turning

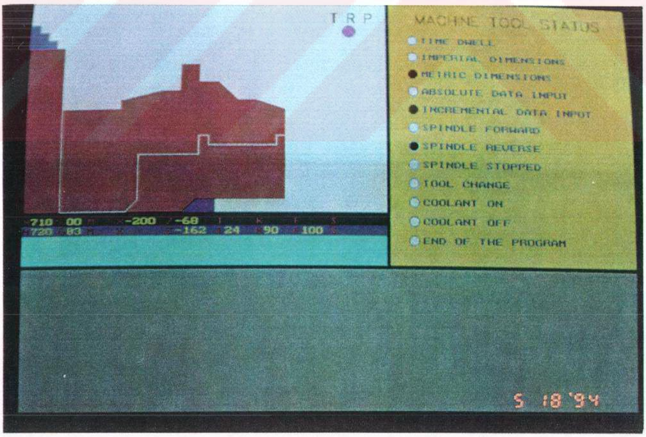


Figure 5.15 The simulation of drilling operation

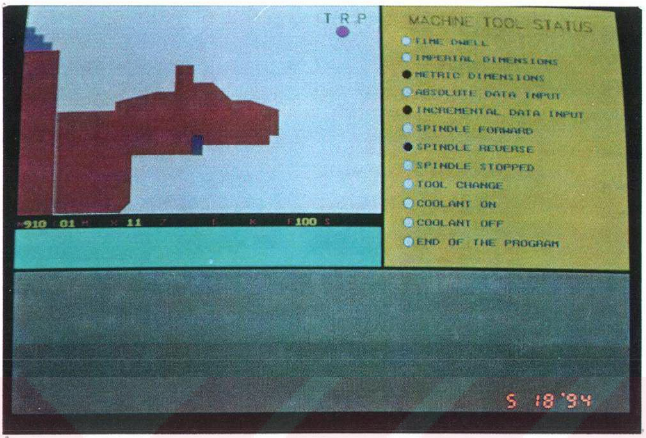


Figure 5.16 The simulation of internal grooving operation

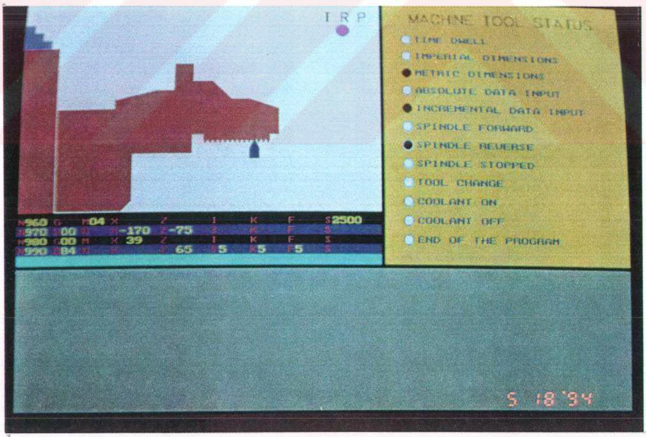


Figure 5.17 The simulation of internal threading operation

Figure 5.18 The time information about the machining processes

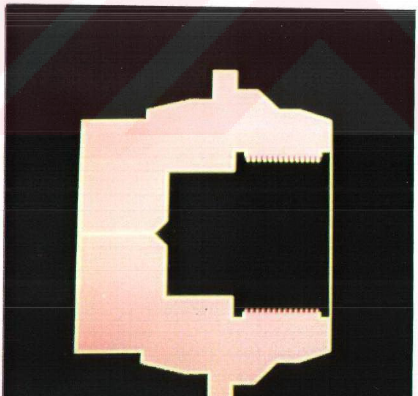


Figure 5.19 The complete view of the finished product

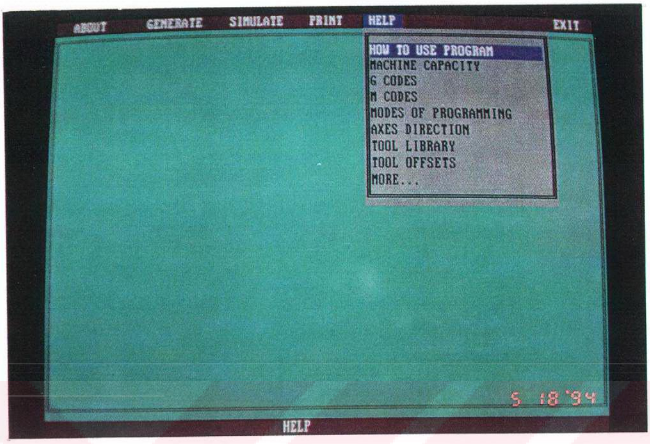


Figure 5.20 Help menu

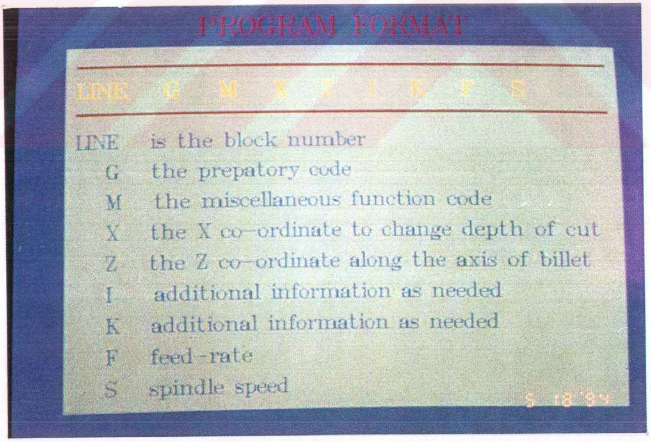


Figure 5.21 Program format

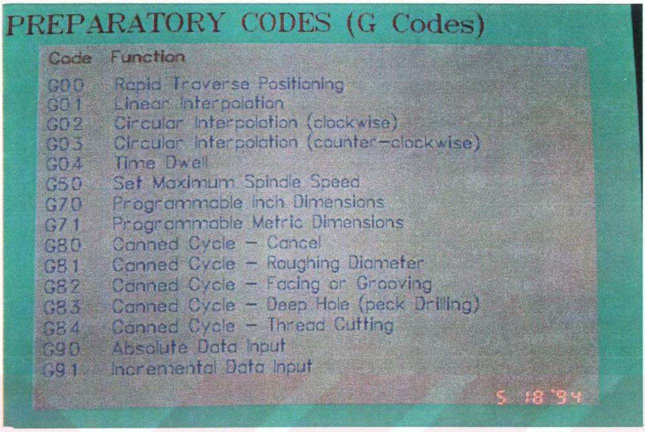


Figure 5.22 Preparatory codes

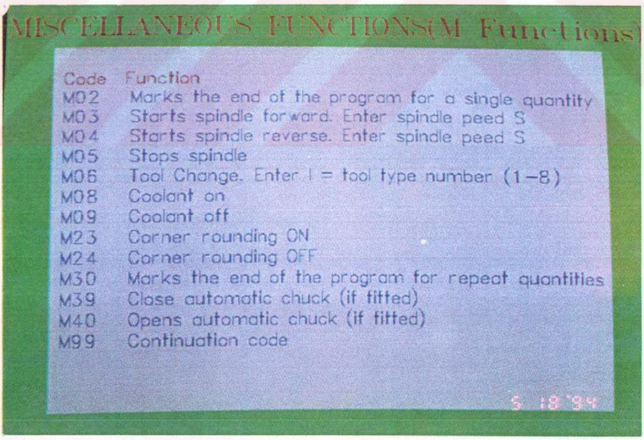


Figure 5.23 Miscellaneous functions

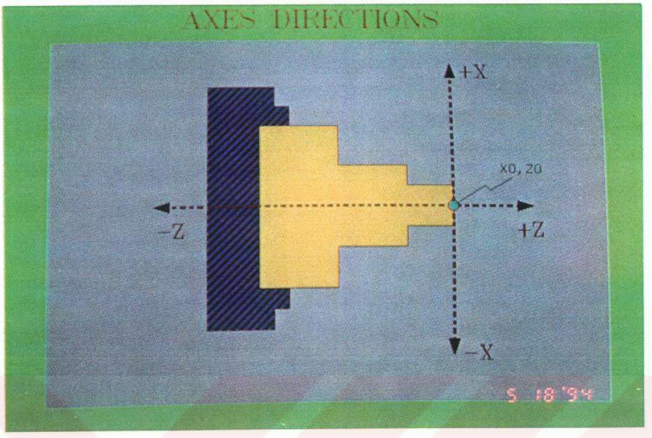


Figure 5.24 Axes directions

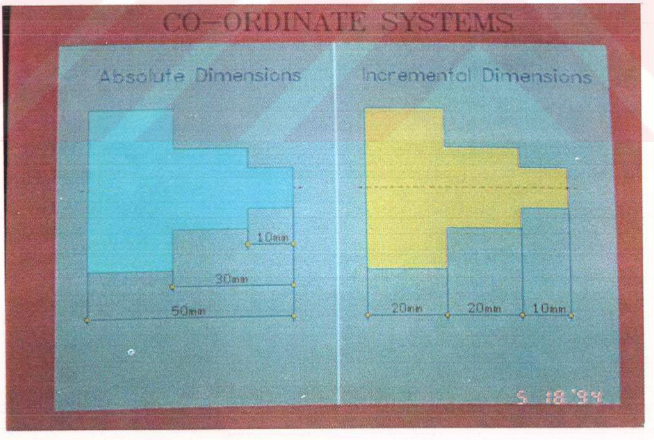


Figure 5.25 Co-ordinate systems

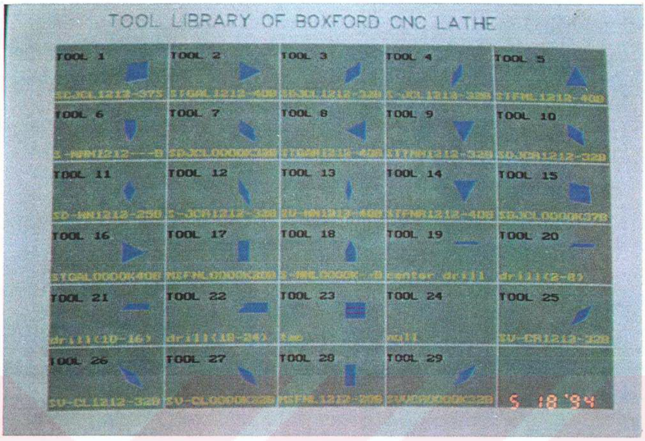


Figure 5.26 The tool library of the Boxford CNC lathe

Figure 5.27 Tool offsets

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1 INTRODUCTION

The evolution of the prepared computer program and problems faced in the development of the study are discussed in section 6.2. Conclusion of the work is given in section 6.3. Section 6.4 is devoted to the recommendations for future study.

6.2 DISCUSSION

The "full automation", "unmanned factory", and "CAD/CAM integration are the most popular terms or concepts in the world of manufacturing, although some authorities claim that these are just dreams impossible to achieve. On the other hand, there are numerous works on the CAD/CAM integration implemented for CIM applications, and all of them use a process planning system as integrator. Indeed, the CAPP is a unique tool removing the slash between CAD and CAM. However, the generation of

CNC part programs and tool path, and their verifications are described as the finalising steps of the process planning systems, almost in all researches. In this study, a computer program has been developed for the automatic generation of CNC codes and tool paths. A simulation program also has been prepared for the verification of those generated codes and paths.

There are mainly five steps involved to complete the study;

- Selection of a computer programming language
- Selection of machine tool and controller combination
- Writing an algorithm for the generation of CNC codes and paths
- Writing an algorithm for the simulation of generated codes
- Converting those algorithms into programs

The first two are considered as the first stage and independent from the researcher, while the remaining three are considered as the second stage and dependent on the researcher.

The algorithm and language of programming are the two predominant factors in the design of a software [35]. The computer programming languages have been actually accepted as CIM technologies serving the objective of flexibility [36]. It is obvious that if a language supports the hardware, the performance of the programs will be increased considerably. In Reference [13], it is mentioned about increased time of working due to GwBasic and advised finally to use another language to overcome this problem. Basic is no longer used in the process planning activities. Alting and Zhang [37] have made a literature survey on the programming languages for CAPP procedures and seen that, PASCAL, C, LISP, and PROLOG are now being used increasingly in that area. Among them, PASCAL and C have a special importance because of their close relationship with UNIX operation system used commonly for machining centres, PROLOG and LISP occupy more memory space and have low speed. So, they are generally used for the steps requiring a highly logical programming before the CNC code generation in the CAPP procedures. In this study C is selected as the programming language, since it is fast, and offers a handy environment to work, and has superior graphical capabilities for simulation processes [38]. The availability of compilers and manual of C played also an important role in this selection.

The selection of the machine tool and controller unit combination *which can be used as the "base" for the format of the generated part programs* is another important consideration, in this work. Unfortunately, the CNC industry has so far been unable to come up with a compatible coding standard. The problem is that each machine tool / control is different and machine readable codes (G and M codes) running the equipment are non-standard. Each control uses different codes and functions specific to a model. Therefore, the same G word may represent different functions on different machines. Part programs written for one control type can not be used for another, even if the machine is the same kind and size, because of those incompatible codes. Besides, the number of words supported by the machine tool control determines the capabilities of the machine. For instance, although there are more than one hundreds G words standardised, many CNC machines only support twenty or thirty of them. However, it is well worth to point out that there wasn't so many alternatives to be selected as the target machine for this study. The only CNC machine that would be used for testing the part programs was the BOXFORD 250 CNC lathe provided by World Bank project for Gaziantep Higher Vocational School, University of Gaziantep. The Boxford 250 is fitted with the standard Boxford CNC control and offers the most of G words and M words as discussed in the preceding sections. However, the words related to tool nose compensation (G40, G41, G42) are not implemented on the control. Therefore, we could not generate any TNR code in the part programs, consider them when calculating the tool paths, and normally simulate its function on the monitor. Besides, although the machine tool originally offers a choice of eight station programmable disc turret, it was not available on the machine. However, we have implemented the turret in our system, since the automation will not be considered without it. All those problems may be overcome by specifying the properties of different machine tool controllers into the system and presenting a simple selection mechanism at the beginning of the part program generation algorithm. Of course, same thing must be applied to the simulation module.

Most of the studies related to process planning, and therefore CNC code generation, use commercially available CAD packages like AutoCAD and AutoLISP to draw the required part geometry. Most commercially available CAD systems use wire-frame models [39]. The parts are drawn in their environment, and saved into the files with DWG (DraWinG) extension. These drawings are then converted into ASCII codes, and stored in the files with DXF (Data eXchange Files). These DXF files are used as the CAD data base. The work done by the researcher is to write a program to extract the information stored in those DXF files. However, in this study, required data for the part geometry is taken from the text file generated by a Computer Aided Drafting package which is designed for a Ph.D. study on CAPP (continuing at the Department of Mechanical Engineering, University of Gaziantep). Since "the generation and simulation of CNC codes" is an integral part of the CAPP study, we did not actually look for any alternative CAD system. The drafting package which will later be integrated into the process planning study enables the designer to draw the part geometry by using simple geometrical entities (like arc and line) through the support of mouse and keyboard [40]. The two dimensional wire-frame model is used to form the shape of the workpiece. Because of the symmetry of the rotational parts, it will be adequate to draw only the upper half of the component.

The solid modelling and three dimensional representations were not preferred in the simulation process as in the case of design of the part, because of inadequacy of computer hardware. It was actually impossible to make such a work on a 386DX-40 personal computer. High memory, high graphic resolution and speed as well as Z-buffering capability are strictly needed in the solid modelling, since too many Boolean subtraction are required for the volume generation. All those properties can only be found on a Work Station. However, although Takata [41] used a SUN4 engineering work station in his 3_D cutting simulation study, he complained from the lower time of simulation and stated finally that the time consumed in the simulation was roughly twice of the actual machining.

6.3 CONCLUSION

The main purpose of this study was to link the Computer Aided Process Planning with the Computer Aided Manufacturing. This has been achieved by using the required part geometry data and process plan associated with the component. The functions of CAPP/CAM interface, including both the automatic generation and verification of CNC part programs together with tool paths, are fulfilled by this work.

With the computer program developed for the generation and simulation of CNC codes, it would be possible to reduce the times of programming and checking that will result an increase in productivity which is generally considered as the ultimate goal of the CIM.

The generated part programs have been tested satisfactorily on the Boxford 250 B CNC lathe, and positive results have been obtained. The experience learned from this study can also be applied to prismatic parts requiring milling operations.

6.4 RECOMMENDATIONS FOR FURTHER STUDY

The work presented in this thesis can be extended to the following fields;

- The generated machining instructions can be converted into different formats readable by different CNC machine tools by preparing an interface (post-processor) program for each.
- Three dimensional representations using solid modelling techniques to display a part during the machining process can be developed for the simulation applications, if sufficient computer hardware is provided.

- CNC data generation systems can be incorporated with the image processing technology. From a scanner or photograph image of the workpiece, the CNC part programs can automatically be generated.

- The part programs can directly be transmitted to the machine tool through the RS 232-C serial communication interface.

LIST OF REFERENCES

1. Filiz, I.H., Dereli, T., and Kayacan, M.C., " Automatic Generation of CNC Part Programs for Rotational Parts", paper to be published.

2. Wang, H.P. and Wysk, R.A., "Intelligent Reasoning for Process Planning", Computers in Industry, No.8, pp. 293-309, 1987.

3. Joseph, A.T., Kalta, M. and Davies, B.J., "Automatic Generation of NC Programs for Turned Components from CAD Product Models", Proceedings of the 28th International MATADOR Conference, pp 289-294, Manchester, April 18-19, 1990.

4. Wang, H.P., Lin, C.A., "Automated Generation of NC Part Programs for Turned Parts Based on 2-D Drawing Files", Int. J. Adv. Manufacturing Technology, Vol. 2, No. 3, pp. 23-36, 1987.

5. Chin, S.C., "Developing A Feature Based Knowledge System For CAD/CAM Integration", Computers & Industrial Engineering, Vol.15, No 1-4, pp 34-40, 1988.

6. Ssemakula, M.E., "Process Planning System in the CIM Environment", Computers & Industrial Engineering, Vol.19, No 1-4, pp 452-456, 1990.

7. Jia-Ming S. and Chen, Y.W., "A Mini CIM System for Turning", Annals of the CIRP, Vol.36, No.1, pp 277-280, 1987.

8. Boer, C.R, Petitti, M., and Lombardi, F., "A CAPP/CAM Expert System for a High Productivity, High Flexibility CNC Turning Center", Annals of the CIRP, Vol.39, No.1, pp 481-483, 1991.

9. Hunt, W.A. and Voelcker, H.B., "An Exploratory Study of Automatic Verification of Programs for Numerically Controlled Machine Tools", Tech. Memo 34, Production Automation Project, University of Rochester, January 1982.

10. Fujii, T. and Speckert, R.E., "Computer Graphics: A way of teaching CNC Programming With or Without Machine Tools", ASEE/EDGD Midyear Meeting, West Lafayette, Indiana, November 24-26, 1985.

11. Norrie, D., Roy, G., Fauvel G. and Guo, D., "Microcomputer Simulation of A CNC Machining Center and Its Application", Computer Modelling and Simulation of Manufacturing Processes, ASME, Materials Divisions, Vol. MD20, pp 217-223, 1990.

12. Sungurtekin, U. and Voelcker, H., "Graphical Simulation and Automatic Verification of NC Machining Programs", Proceedings of IEEE Int. Conference on Robotics and Automation, San Francisco, California, April, 1986.

13. Ozdemir, Ahmet, "CNC'li Torna Tezgahlarina Hazirlanan Is Uretme Programinin Bilgisayar Yardimi ile Simulasyonu", M. Sc. Thesis, School of Natural and Applied Sciences, Gazi University, 1988.

14. Unlusoy, Mete, "Implementation of Computer Graphics on Computer Aided Process Planning System for Rotational Parts", M.S. Thesis, METU, February, 1989.

15. Filiz, I.H., Kayacan, M.C. and Dereli, T., "Dönel Is Parçalarinin Bilgisayar Destekli Tasarimi ve Islem Sirasi Planlamasi (Computer Aided Design and Operation Sequencing for Turned Parts)", Proceedings of 6st National Symposium On Theory Of Machines, Vol.1, pp 279-288, Trabzon, September 15-17, 1993.

16. Ham, I., and Lu, S.C., "Computer-Aided Process Planning: The Present and the Future", Annals of the CIRP, Vol.37, No.2, pp 591-600, 1988.

17. Cho, K.K., Lee, S.E., and Ahn, J.H., "Development of Integrated Process Planning and Monitoring System for Turning Operation", Annals of the CIRP, Vol.40, No.1, pp 423-427, 1991.

18. Kusiak, Andrew, Intelligent Manufacturing Systems, Prentice Hall, Inc., 1990.

19. Hans B.K., Flexible Automation, Becker Publishing Company, Havant, Hampshire, 1986.

20. Roberts, A.D., and Prentice, R.C., Programming For Numerical Control Machines, McGraw-Hill, Inc., 1970.

21. Degarmo, E.P, Black, J.T, and Kohser, R.A., Materials and Processes in Manufacturing, Macmillan Publishing Company, 1988.

22. Niebel, B.W, Draper, A.B, and Wysk, A.R, Modern Manufacturing Process Engineering, McGraw-Hill, Inc, 1989.

23. Groover, Mikell. P., Automation, Production Systems, and Computer-Aided Manufacturing, Prentice-Hall, Inc.,1980.

24. Sava, Michael and Pusztai, Joseph, Computer Numerical Control Programming, Prentice-Hall International, Inc, 1990.

25. Groover, Mikell. P., and Emory W. Zimmers, CAD/CAM Computer Aided Design and Manufacturibg, Prentice-Hall, Inc.,1984.

26. Lynch, Mike, Computer Numerical Control for Machining, McGraw-Hill, Inc, 1992.

27. Furth, B.I., "Automated Process Planning", NATO Advanced Study Series - Computer Aided Manufacturing, Vol. F49, pp 37-53, 1988.

28. Peklenik, J., and Sluga, A., "Contribution for Development of a Generative CAPP-System Based on Manufacturing Process Technology", Annals of the CIRP, Vol.38, No.1, pp 417-420, 1989.

29. Hinduja, S. and Barrow, G., "TECHTURN - A Technically Oriented System for Turned Components", Proceedings of International

Conference on Computer Aided Production Engineering, pp 255-260, Edinburgh, April, 1986.

30. Plummer., J.C.S., "Using a CAD/CAM System, To Link The Design, Drawing and Manufacture of Turned Parts", PhD. Thesis, Manufacturing and Machine Tool Division, Department of Mechanical Engineering, UMIST, 1990.

31. Kirecci, Ali, "Computer Integrated Manufacturing as Regard Group Technology Approach", M. Sc. Thesis, School of Natural and Applied Sciences, METU, 1988.

32. Hinduja, S. and Huang, H., "OP-PLAN: An Automated Operation Planning System for Turned Components", Journal of Engineering Manufacture, Vol.203, pp 145-158, 1989.

33. Van Houten, M., "Strategy in Generative Planning of Turning Processes", Annals of the CIRP, Vol.35, No.1, pp 331-335, 1986.

34. Boxford 250 B CNC Machine & Programming Manual, Boxford Ltd.,1991.

35. Filiz, I.H., Kayacan M.C. and Dereli T., "CNC Tornalama Islemleri için C++ ile Hazirlanan Bir Simülasyon Yazilimi (A Simulation Study for CNC Lathe Operations)", Proceedings of International Symposium On Computer Applications, pp 63-70, Konya, June 9-10, 1993.

36. Kochan, A. and Cowan, D., Implementing CIM, IFS (publications) Ltd., UK, 1986.

37. Alting, L. and Zhang, H., " Computer Aided Process Planning: The State-Of-The Art Survey", Int.J.Prod.Research, Vol.27, No.4, pp 553-585, 1989.

38. Stevens, Roger. T., The C Graphics Handbook, Academic Press, Inc.,1992.

39. Wang, H.P, and Wysk, R.A., " AIMSI: A Prelude To a New Generation of Integrated CAD/CAM Systems", Int. J. Prod. Research, Vol. 26, No.1, pp. 119-131, 1988.

40. Filiz, I.H., Kayacan, M.C. and Dereli, T., "Romod: A Product Modelling System for Rotational Parts", Paper to be published.

41. Takata, S., Tsai, M.D., and Inui, M., "A Cutting Simulation System For Machinability Evaluation Using a Workpiece Model", Annals of the CIRP, Vol.38, No.1, pp 417-420, 1989.

APPENDIX

Table A.1 PREPARATORY FUNCTIONS (G-CODES)

Standart	Special	Turning Centers
G00	G00	Positioning (rapid traverse)
G01	G01	Linear Interpolation (cutting Feed)
G02	G02	Circular Interpolation (clockwise)
G03	G03	Circular Interpolation (counterclockwise)
G04	G04	Dwell cycle
G10	G10	Offset value setting by program
G20	G70	Inch data input
G21	G71	Metric data input
G22	G22	Safety zone programming
G23	G23	Programmed crossing through safety zone
G27	G27	Reference point return check
G28	G28	Return to reference point
G29	G29	Return from reference point
G30	G30	Return to 2nd reference point
G31	G31	Skip cutting
G34	G34	Variable lead thread cutting
G36	G36	Automatic tool compensation X axis
G37	G37	Automatic tool compensation Z axis
G40	G40	Tool nose radius compensation cancel
G41	G41	Tool nose radius compensation left
G42	G42	Tool nose radius compensation right
G50	G92	Programming of absolute zero point
G66	G66	Modal user macro call command
G68	G68	Mirror image for double turrets ON
G69	G69	Mirror image for double turrets OFF
G70	G72	Finishing cycle
G71	G73	Stock removal in turning
G72	G74	Stock removal in facing
G73	G75	Pattern repeating
G74	G76	Peck-drilling in Z axis
G75	G77	Grooving in X axis
G76	G78	Thread cutting cycle
G90	G20	Cutting cycle A
G92	G21	Thread cutting cycle
G94	G24	Cutting cycle A
G96	G96	Constant surface speed control on
G97	G97	Constant surface speed control cancel
G98	G94	Feed per minute
G99	G95	Feed per revolution
—	G90	Absolute Programming
—	G91	Incremental Programming

Table A.2 MISCELLANEOUS FUNCTIONS (M-CODES)

M-Code	Mill/Turn
M00	Program Stop
M01	Optional Stop
M02	End of program and tape rewind
M03	Spindle Start CW
M04	Spindle Start CCW
M05	Spindle stop
M06	Tool change
M08	Coolant ON
M09	Coolant OFF
M10	Polarity change (probing)
M11	Read X-measurement (probing)
M12	Read Y-measurement (probing)
M13	Read Z-measurement (probing)
M19	Spindle orient/stop
M21	Mirror image X
M22	Mirror image Y
M23	Mirror image OFF
M27	Reset register (probing)
M30	End of program and memory rewind
M38	Measurement correction to CNC
M41	Low range (mill) / Deviation measurement
M42	High range (mill) / Comparative measurement
M44	Centre Location Measurement (probing)
M45	Multisurface centre locations (probing)
M46	Hole centre (probing)
M47	Shaft centre
M48	Override cancel OFF (mill)
M49	Override cancel ON (mill)
M98	Go to subroutine
M99	Return from subroutine

<u>Note:</u> If one examines a number of specific G and M codes in Tables A.1 and A.2, it will be seen that the same code may represent a number of different functions. Therefore an end user must always follow the specific manual for a machine tool at all times.

DRAWING NO.	BOXFORD PROGRAMMING SHEET								MATERIAL:-
	TITLE.			WRITTEN BY:-					SHEET NO. ____ OF ____
SEQUANCE DESCRIPTION	SEQUANCE NUMBER	PREP. FUNCTION	MISC. FUNCTION	AXIS CO-ORDINATES				FEED RATE	SPINDLE SPEED
	N	G	M	X	Z	I	K	F	S

Figure A.1 Programming Sheet of Boxford 250 CNC Lathe

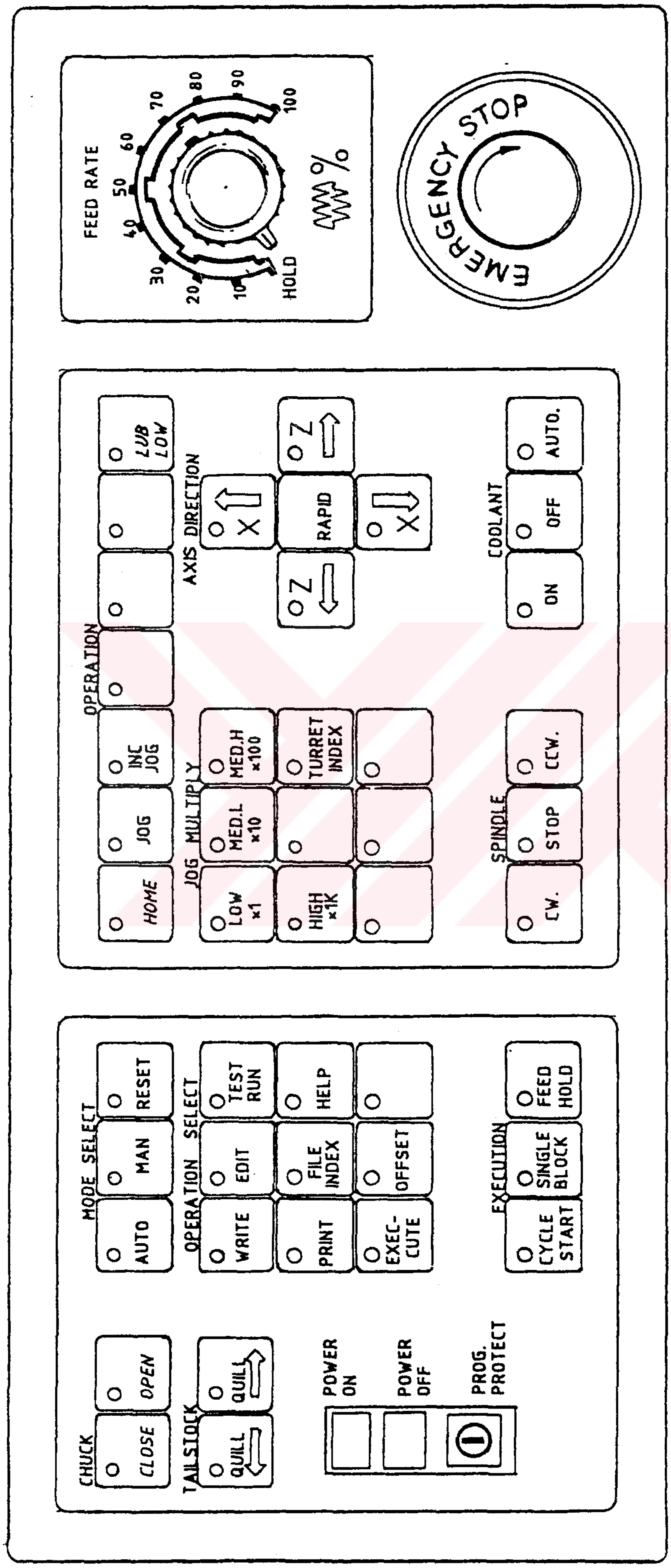
OPERATIONS SHEET

BILLET | MATERIAL __________ TYPE NO. | TITLE

____ O/DIA. ____ I/DIA. ___ STICKOUT

OPERATION NUMBER	OPERATION DESCRIPTION	SPINDLE		TOOL	TOOL
		SPEED	RANGE	FEED	NO.

Figure A.2 Operations Sheet of Boxford 250 CNC Lathe



Machine Function Keyboard

250B Boxtronic

Figure A.3 Machine Function Keyboard

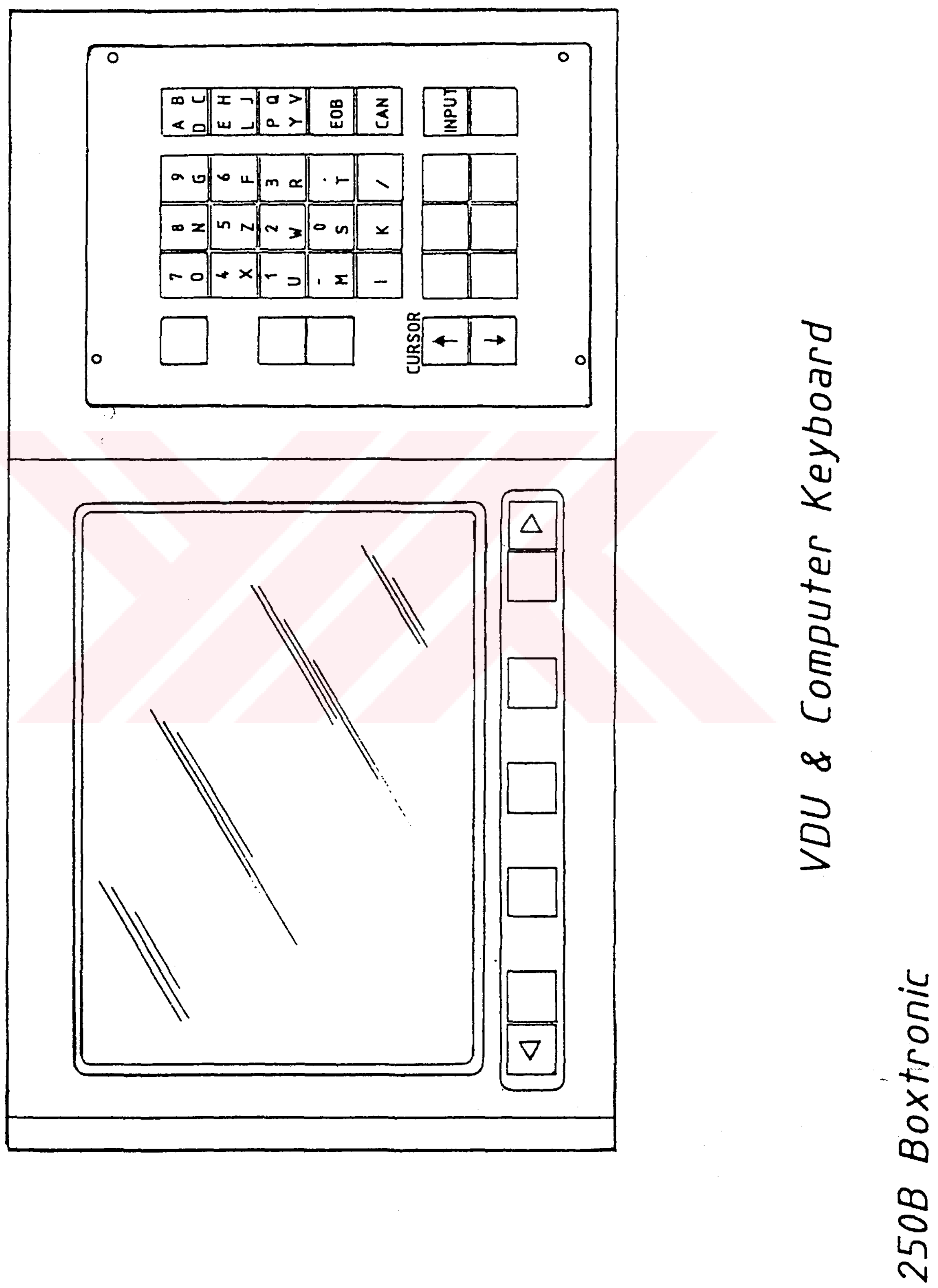


Figure A.4 Visual Display Unit and Keyboard of Boxford 250 Lathe

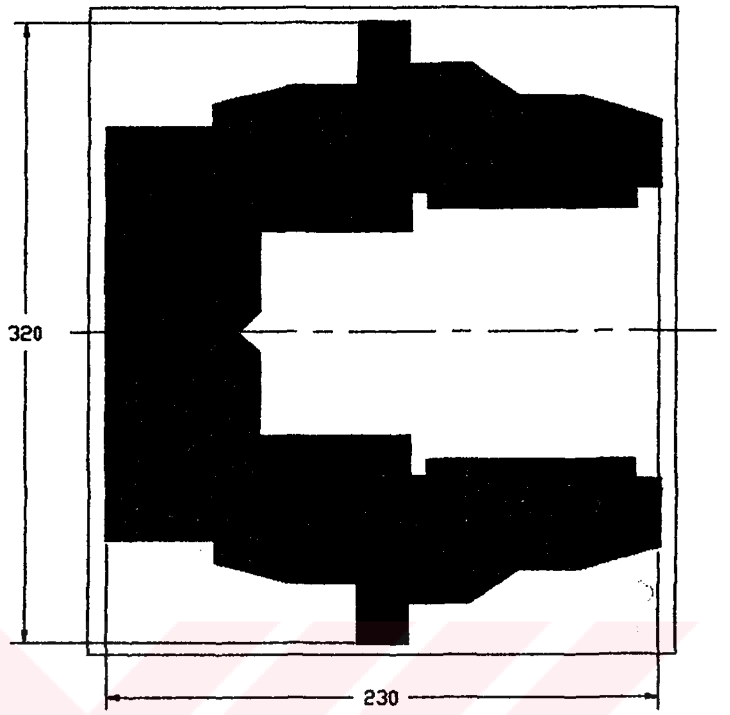


Figure A.5 Example part II

Table A.3 The text file obtained from the example part II

DSN	X1	Y1	X2	Y2	RAD
1	60	240	60	80	0
2	60	80	100	100	0
3	100	100	210	100	0
4	230	100	90	180	20
5	230	80	290	80	0
6	290	80	290	160	0
7	290	160	240	160	0
8	240	160	240	180	0
9	240	180	200	190	0
10	200	190	190	190	0
11	190	190	190	180	0
12	190	180	180	180	0
13	180	180	180	190	0
14	180	190	130	190	20
15	130	190	130	180	0
16	130	180	100	180	0
17	100	180	100	190	0
18	100	210	90	180	20
19	80	210	80	230	0
20	80	230	70	240	0
21	70	240	60	240	0

Table A.4 The CAD/CAM data base for the example part II

no	opr	x1	y1	x2	y2	mx	my	sa	ea	rad	t	doc	sn	f	spd
1	efcn_lh	290	60	292	240	0	0	0	0	0	5	2	0	60	3000
2	ecyl_lh	60	60	290	80	0	0	0	0	0	2	4	0	57	3000
3	ercs_rh	82	80	210	83	0	0	0	0	0	10	3	0	58	3000
3	ercs_rh	82	83	210	86	0	0	0	0	0	10	3	0	58	3000
4	ercs_rh	92	86	208	89	0	0	0	0	0	10	3	0	58	3000
4	ercs_rh	92	89	208	92	0	0	0	0	0	10	3	0	58	3000
5	ercs_rh	100	92	209	106	0	0	0	0	0	10	4	0	57	3000
5	ercs_rh	100	96	209	100	0	0	0	0	0	10	4	0	57	3000
6	ecnc_rh	60	80	100	100	0	0	0	0	0	25	0	2	55	2900
7	eprf_rh	210	100	230	80	230	100	90	180	20	26	0	4	55	2900
8	idrl_lh	70	240	290	230	0	0	0	0	0	22	0	0	50	2000
9	icyl_lh	80	210	290	230	0	0	0	0	0	16	4	0	56	3000
10	icyl_lh	84	206	290	210	0	0	0	0	0	16	4	0	56	3000
11	icyl_lh	86	202	290	206	0	0	0	0	0	16	4	0	56	3000
12	icyl_lh	88	198	290	202	0	0	0	0	0	16	4	0	56	3000
13	icyl_lh	92	194	290	198	0	0	0	0	0	16	4	0	56	3000
14	icyl_lh	100	190	290	194	0	0	0	0	0	16	4	0	56	3000
15	icyl_lh	214	188	290	190	0	0	0	0	0	16	2	0	60	3000
16	icyl_lh	224	186	290	188	0	0	0	0	0	16	2	0	60	3000
17	icyl_lh	234	182	290	186	0	0	0	0	0	16	4	0	55	3000
18	icyl_lh	240	178	290	182	0	0	0	0	0	16	4	0	55	3000
19	icyl_lh	240	172	290	175	0	0	0	0	0	16	3	0	57	3000
19	icyl_lh	240	175	290	178	0	0	0	0	0	16	3	0	57	3000
20	icyl_lh	240	166	290	169	0	0	0	0	0	16	3	0	57	3000
20	icyl_lh	240	169	290	172	0	0	0	0	0	16	3	0	57	3000
21	icyl_lh	240	160	290	163	0	0	0	0	0	16	3	0	57	3000
21	icyl_lh	240	163	290	166	0	0	0	0	0	16	3	0	57	3000
22	icnc_lh	240	180	200	190	0	0	0	0	0	27	0	14	50	2800
23	iprf_lh	100	190	80	210	100	210	90	180	20	27	0	23	50	2900
24	igrv_lh	180	180	190	190	0	0	0	0	0	17	2	17	50	2000
25	igrv_lh	100	180	130	190	0	0	0	0	0	17	10	21	55	2850
26	ithr_lh	180	190	130	190	0	0	0	0	0	18	0	19	1	800

Table A.5 The list of part program generated for the example part II (incremental)

N	G	M	X	Z	I	K	F	S
10	91							
20	71							
30		06			5	1		
40		04						3000
50	00		-19	-39				
60	01		-181		2		60	
70	00			50				
80			200					
90		05						
100		06			2	2		
110		04						3000
120	00		-19	-49				
130	81		20	-231	5		57	
170	00		19	49				
180	00			-258				
190		05						
200		06			10	3		
210		04						3000
220	00		-39					
230	01		-4				58	
240	01			128			58	
250	00		1	-128				

260	01		-4				58	
270	01			128			58	
280	00		1	-118				
290	01		-4				58	
300	01			116			58	
310	00		1	-116				
320	01		-4				58	
330	01			116			58	
340	00		1	-108				
350	01		-5				57	
360	01			109			57	
370	00		1	-109				
380	01		-5				57	
390	01			109			57	
400	00		60	131				
410		05						
420		06			25	4		
430		04						2900
440	00			-239				
450	00		-60	-1				
460	01		20	-40			55	
470	00			1				
480	00		40	279				
490		05						
500		06			26	5		
510		04						2900
520	00			-130				
530	00		-60					
540	02		20	20		20	55	
550	00			-1				
560	00		40	111				
570		05						
580		06			22	6		
590		04						2000
600	00		-200	-48				
610	83			-220	20	90	50	
620	00			48				
630	00		200					
640		05						
650		06			16	7		
660		04						3000
670	00		-191	-49				
680	81		20	-211	5		56	
690	00		25					
700	01			-207			56	
710	00		-1	207				
720	00		5					
730	01			-205			56	
740	00		-1	205				
750	00		5					
760	01			-203			56	
770	00		-1	203				
780	00		5					
790	01			-199			56	
800	00		-1	199				
810	00		5					
820	01			-191			56	
830	00		-1	191				
840	00		3					
850	01			-77			60	

860	00		-1	77				
870	00		3					
880	01			-67			60	
890	00		-1	67				
900	00		5					
910	01			-57			55	
920	00		-1	57				
930	00		5					
940	01			-51			55	
950	00		-1	51				
960	00		4					
970	01			-51			57	
980	00		-1	51				
990	00		4					
1000	01			-51			57	
1010	00		-1	51				
1020	00		4					
1030	01			-51			57	
1040	00		-1	51				
1050	00		4					
1060	01			-51			57	
1070	00		-1	51				
1080	00		4					
1090	01			-51			57	
1100	00		-1	51				
1110	00		4					
1120	01			-51			57	
1130	00			100				
1140	00		120					
1150		05						
1160		06			27	8		
1170		04						2800
1180	00		-170	-100				
1190	00		30					
1200	01		-10	-40			50	
1210	00		-20	-100				
1220	00		20					
1230	03		-20	-20	20		50	2900
1240	00			260				
1250	00		170					
1260		05						
1270		06			17	9		
1280		04						2000
1290	00		-170	-160				
1300	00		19					
1310	01		11				50	
1320	00		-30	-60				
1330	00		19					
1340	01		11				55	
1350	00		-11	-10				
1360	01		11				55	
1370	00		-11	10				
1380	82		11	-10			55	2850
1390	00		-19	220				
1400	00		170					
1410		05						
1420		06			18	10		
1430		04						800
1440	00		-170	-155				
1450	00		19					

1460	84			45	1.23	10	1
1470	00		-19	155			
1480	00		170				
1490		05					
1500		30					

Table A.6 The tool path file generated for the example part II

#	X	Z	#	X	Z	#	X	Z
1	19	-39	66	0	222	131	-50	0
2	0	-6	67	0	48	132	20	-20
3	181	0	68	-200	0	133	30	260
4	0	1	69	191	-49	134	-200	0
5	-181	0	70	-5	0	135	200	-160
6	0	-6	71	0	-211	136	-49	0
7	181	0	72	1	0	137	-11	0
8	0	11	73	0	211	138	60	-60
9	-181	0	74	-5	0	139	-49	0
10	-19	39	75	0	-211	140	-11	0
11	19	-49	76	1	0	141	11	-10
12	5	0	77	0	211	142	-11	0
13	0	-231	78	-5	0	143	11	10
14	-1	0	79	0	-211	144	0	-20
15	0	231	80	1	0	145	-11	0
16	5	0	81	0	211	146	11	0
17	0	-231	82	-5	0	147	0	20
18	-1	0	83	0	-211	148	49	220
19	0	231	84	1	0	149	-200	0
20	5	0	85	0	211	150	200	-155
21	0	-231	86	-5	0	151	-49	0
22	-1	0	87	0	-211	152	-2	0
23	0	231	88	21	0	153	0	-55
24	5	0	89	0	211	154	1	0
25	0	-231	90	-25	0	155	0	55
26	-1	0	91	0	-207	156	-2	0
27	0	231	92	1	207	157	0	-55
28	5	0	93	-5	0	158	1	0
29	0	-231	94	0	-205	159	0	55
30	-21	0	95	1	205	160	-2	0
31	0	231	96	-5	0	161	0	-55
32	-19	49	97	0	-203	162	1	0
33	0	-258	98	1	203	163	0	55
34	39	0	99	-5	0	164	-2	0
35	7	0	100	0	-199	165	0	-55
36	0	128	101	1	199	166	1	0
37	-1	-118	102	-5	0	167	0	55
38	7	0	103	0	-191	168	-2	0
39	0	116	104	1	191	169	0	-55
40	-1	-108	105	-3	0	170	1	0
41	9	0	106	0	-77	171	0	55

42	0	109	**107**	1	77	**172**	-2	0
43	-60	131	**108**	-3	0	**173**	0	-55
44	0	-239	**109**	0	-67	**174**	1	0
45	60	-1	**110**	1	67	**175**	0	55
46	-20	-40	**111**	-5	0	**176**	-2	0
47	0	1	**112**	0	-57	**177**	0	-55
48	-40	279	**113**	1	57	**178**	1	0
49	0	-130	**114**	-5	0	**179**	0	55
50	60	0	**115**	0	-51	**180**	-2	0
51	-20	20	**116**	1	51	**181**	0	-55
52	0	-1	**117**	-7	0	**182**	1	0
53	-40	111	**118**	0	-51	**183**	0	55
54	200	-48	**119**	1	51	**184**	-2	0
55	0	-42	**120**	-7	0	**185**	0	-55
56	0	42	**121**	0	-51	**186**	1	0
57	0	-62	**122**	1	51	**187**	0	55
58	0	62	**123**	-7	0	**188**	-2	0
59	0	-92	**124**	0	-51	**189**	0	55
60	0	92	**125**	0	100	**190**	11	0
61	0	-137	**126**	-120	0	**191**	0	5
62	0	137	**127**	200	-100	**192**	49	155
63	0	-204	**128**	-60	0	**193**	-200	0
64	0	204	**129**	10	-40			
65	0	-222	**130**	50	-100			

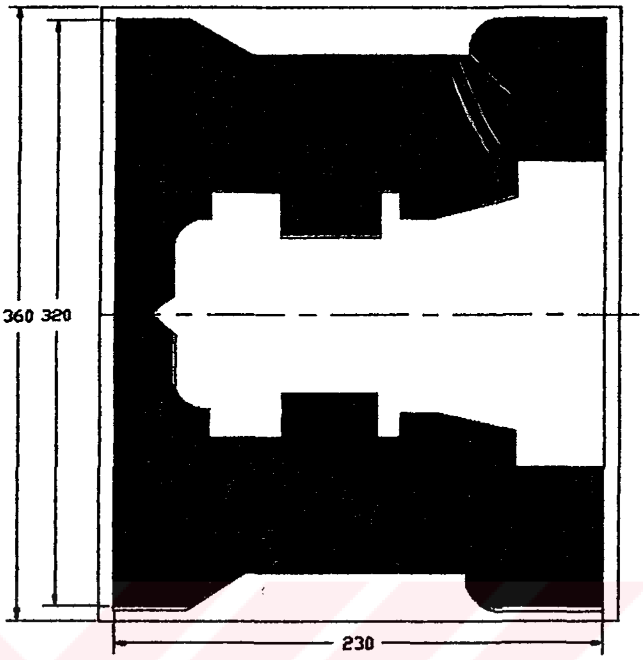


Figure A.6 Example part III

Table A.7 The text file obtained from the example part III

DSN	X1	Y1	X2	Y2	RAD
1	40	240	40	130	0
2	40	130	100	130	0
3	100	130	100	120	0
4	100	120	140	110	0
5	140	110	160	110	0
6	160	110	160	80	0
7	160	80	180	80	0
8	180	80	180	100	0
9	180	100	200	100	0
10	200	100	220	120	0
11	220	120	240	120	0
12	240	120	270	130	0
13	270	130	270	160	0
14	270	160	260	160	0
15	260	160	260	170	0
16	260	170	190	170	5
17	190	170	190	160	0
18	190	160	180	160	0
19	180	160	180	180	0
20	180	180	120	180	0
21	120	180	120	228	0
22	120	228	108	240	0
23	108	240	40	240	0

Table A.8 The CAD/CAM data base for the example part III

no	opr	x1	y1	x2	y2	mx	my	sa	ea	rad	t	doc	sn	f	spd
1	efcn_lh	270	60	272	240	0	0	0	0	0	5	2	0	60	3000
2	ecyl_lh	40	60	270	80	0	0	0	0	0	2	4	0	57	3000
3	ecyl_lh	180	80	270	100	0	0	0	0	0	2	4	0	57	3000
4	ecyl_lh	208	100	270	106	0	0	0	0	0	2	2	0	60	3000
5	ecyl_lh	214	106	270	110	0	0	0	0	0	2	4	0	57	3000
6	ecyl_lh	220	110	270	113	0	0	0	0	0	2	3	0	58	3000
7	ecyl_lh	220	113	270	116	0	0	0	0	0	2	3	0	58	3000
8	ecyl_lh	220	116	270	120	0	0	0	0	0	2	4	0	57	3000
9	ecyl_lh	256	120	270	124	0	0	0	0	0	2	4	0	57	3000
10	ecnc_lh	240	120	270	130	0	0	0	0	0	25	0	12	57	2950
11	ecnc_lh	200	100	220	120	0	0	0	0	0	25	0	10	57	2950
12	egrv_lh	40	80	100	130	0	0	0	0	0	28	10	0	53	2850
13	ecyl_rh	100	80	160	110	0	0	0	0	0	8	3	0	58	3000
14	ecyl_rh	100	110	116	114	0	0	0	0	0	8	4	0	57	3000
15	ecnc_rh	100	120	140	110	0	0	0	0	0	26	0	4	57	2950
16	idrl_lh	108	240	270	228	0	0	0	0	0	22	0	0	50	2000
17	icyl_lh	120	180	270	228	0	0	0	0	0	16	3	0	58	3000
18	icyl_lh	180	170	270	180	0	0	0	0	0	16	2	0	60	3000
19	icyl_lh	260	160	270	170	0	0	0	0	0	16	2	0	60	3000
20	igrv_lh	180	160	190	170	0	0	0	0	0	17	10	18	53	2850
21	ithr_lh	260	170	190	170	0	0	0	0	0	18	1.23	16	1	700

Table A.9 The list of part program generated for part III (incremental)

N	G	M	X	Z	I	K	F	S
10	91							
20	71							
30		06			5	1		
40		04						3000
50	00		-19	-70				
60	82		-181		5		60	
70	00			70				
80	00		200					
90		05						
100		06			2	2		
110		04						3000
120	00		-19	-69				
130	81		20	-231	5		57	
140	00		-20					
150	81		20	-91	5		57	
160	00		-20					
170	81		6	-63	3		60	
180	00		-11					
190	01			-57			57	
200	00		1	57				
210	00		-4					
220	01			-51			58	
230	00		1	51				
240	00		-4					
250	01			-51			58	

260	00		1	51				
270	00		-5					
280	01			-51			57	
290	00		1	51				
300	00		-5					
310	01			-15			57	
320	00		84	84				
330		05						
340		06			25	3		
350		04						2950
360	00			-69				
370	00		-90	-1				
380	01		10	-30			57	
390	00		1					
400	00		-1	-20				
410	01		20	-20			57	
420	00			1				
430	00		60	139				
440		05						
450		06			28	4		
460		04						
470	00			-250				
480	00		-39					
490	01		-51				53	
500	00		51	-10				
510	01		-51				58	
520	00		51	10				
530	82		51	-40	4		57	
540	00		39	250				
550		05						
560		06			8	5		
570		04						3000
580	00			-241				
590	00		-39					
600	81		30	61	10		57	
610	00		-35					
620	01			17			57	
630	00		74	224				
640		05						
650		06			26	6		
660		04						2950
670	00			-241				
680	00		-80	1				
690	01		10	40			57	
700	00			1				
710	00		70	201				
720		05						
730		06			22	7		
740		04						2000
750	00		-200	-68				
760	83			-162	24	0.9	50	
770	00			68				
780	00		200					
790		05						
800		06			16	8		
810		04						3000
820	00		-189	-69				
830	81		48	-151	16		58	
840	00		48					
850	81		10	-91	5		60	

860	00		10					
870	81		10	-11	5		60	
880	00			69				
890	00		131					
900		05						
910		06			17	9		
920		04						2850
930	00		-170	-160				
940	00		39					
950	01		11				53	
960	00		-50	160				
970	00		170					
980		05						
990		06			18	10		
1000		04						700
1010	00		-170	-75				
1020	00		39					
1030	84			65	1.23	10	1	
1040	00		-39	75				
1050	00		170					
1060		05						
1070		30						

Table A.10 The list of part program generated for part III (absolute)

N	G	M	X	Z	I	K	F	S
10	90							
20	71							
30			400	70				
40		06			5	1		
50		04						3000
60	00		362	0				
70	82		0		5		60	
80	00		400	70				
90		05						
100		06			2	2		
110		04						3000
120	00		362	1				
130	81		40	-231	5		57	
140	00		322	1				
150	81		40	-91	5		57	
160	00		282	1				
170	81		12	-63	3		60	
180	00		260	1				
190	01			-56			57	
200	00		262	1				
210	00		254					
220	01			-50			58	
230	00		256	1				
240	00		248					
250	01			-50			58	
260	00		250	1				
270	00		240					
280	01			-50			57	
290	00		242	1				

200	00		232					
310	01			-14			27	
320	00		400	70				
330		05						
340		06			25	3		
350		04						2950
360	00		400	1				
370	00		220	1				
380	01		240	-30			57	
390	00			-50				
400	01		280	-70			57	
410	00			-69				
420	00		400	70				
430		05						
440		06			28	4		
450		04						2850
460	00		400	-180				
470	00		322					
480	01		220				53	
490	00		322	-190				
500	01		220				58	
510	00		322	-180				
520	82		102	-40	4		57	
530	00		400	70				
540		05						
550		06			8	5		
560		04						3000
570	00		400	-171				
580	00		322					
590	81		60	61	10		57	
600	00		252	-171				
610	01			-154			57	
620	00		400	70				
630		05						
640		06			26	6		
650		04						2950
660	00		400	-171				
670	00		240	-170				
680	01		260	-130			57	
690	00			-131				
700	00		400	70				
710		05						
720		06			22	7		
730		04						2000
740	00		0	2				
750	83			-162	24	90	50	
760	00		0	70				
770	00		400					
780		05						
790		06			16	8		
800		04						3000
810	00		22	1				
820	81		96	-151	16		58	
830	00		118	1				
840	81		20	-91	5		60	
850	00		138	1				
860	81		20	-11	5		60	
870	00		118	70				
880	00		400					
890		05						

900		06			17	9		
910		04						2850
920	00		30	-90				
930	00		138					
940	01		160				53	
950	00		30	70				
960	00		400					
970		05						
980		06			18	10		
990		04						700
1000	00		30	-5				
1010	00		138					
1020	84			65	1.23	10	1	
1030	00		30	70				
1040	00		400					
1050		05						
1060		30						

Table A.11 The tool path file generated for the example part III

#	X	Z	#	X	Z	#	X	Z
1	19	-59	89	-84	84	177	0	-151
2	0	-3	90	0	-69	178	1	0
3	181	0	91	90	-1	179	0	151
4	0	1	92	-10	-30	180	-7	0
5	-181	0	93	-1	0	181	0	-151
6	0	-3	94	1	-20	182	1	0
7	181	0	95	-20	-20	183	0	151
8	0	1	96	0	1	184	-7	0
9	-181	0	97	-60	139	185	0	-151
10	0	-3	98	0	-250	186	1	0
11	181	0	99	39	0	187	0	151
12	0	1	100	51	0	188	-7	0
13	-181	0	101	-51	-10	189	0	-151
14	0	-3	102	51	0	190	49	0
15	181	0	103	-51	10	191	0	151
16	0	1	104	0	-20	192	-48	0
17	-181	0	105	51	0	193	-3	0
18	0	-3	106	-51	0	194	0	-91
19	181	0	107	0	-10	195	1	0
20	0	11	108	51	0	196	0	91
21	-181	0	109	-51	0	197	-3	0
22	-19	59	110	0	-10	198	0	-91
23	19	-59	111	51	0	199	1	0
24	5	0	112	-51	0	200	0	91
25	0	-231	113	0	-10	201	-3	0
26	-1	0	114	51	0	202	0	-91
27	0	231	115	-51	0	203	1	0
28	5	0	116	0	50	204	0	91

29	0	-231	117	-39	250	205	-3	0
30	-1	0	118	0	-241	206	0	-91
31	0	231	119	39	0	207	1	0
32	5	0	120	7	0	208	0	91
33	0	-231	121	0	61	209	-3	0
34	-1	0	122	-1	0	210	0	-91
35	0	231	123	0	-61	211	11	0
36	5	0	124	7	0	212	0	91
37	0	-231	125	0	61	213	-10	0
38	-1	0	126	-1	0	214	-3	0
39	0	231	127	0	-61	215	0	-11
40	5	0	128	7	0	216	1	0
41	0	-231	129	0	61	217	0	11
42	-21	0	130	-1	0	218	-3	0
43	0	231	131	0	-61	219	0	-11
44	20	0	132	7	0	220	1	0
45	5	0	133	0	61	221	0	11
46	0	-91	134	-1	0	222	-3	0
47	-1	0	135	0	-61	223	0	-11
48	0	91	136	7	0	224	1	0
49	5	0	137	0	61	225	0	11
50	0	-91	138	-31	0	226	-3	0
51	-1	0	139	0	-61	227	0	-11
52	0	91	140	35	0	228	1	0
53	5	0	141	0	17	229	0	11
54	0	-91	142	-74	224	230	-3	0
55	-1	0	143	0	-241	231	0	-11
56	0	91	144	80	1	232	11	0
57	5	0	145	-10	40	233	0	11
58	0	-91	146	0	1	234	0	69
59	-1	0	147	-70	201	235	-131	0
60	0	91	148	200	-68	236	200	-160
61	5	0	149	0	-50	237	-69	0
62	0	-91	150	0	50	238	-11	0
63	-21	0	151	0	-74	239	80	160
64	0	91	152	0	74	240	-200	0
65	20	0	153	0	-110	241	200	-75
66	3	0	154	0	110	242	-69	0
67	0	-63	155	0	-164	243	-2	0
68	-1	0	156	0	164	244	0	-75
69	0	63	157	0	68	245	1	0
70	3	0	158	-200	0	246	0	75
71	0	-63	159	189	-69	247	-2	0
72	-1	0	160	-7	0	248	0	-75
73	0	63	161	0	-151	249	1	0
74	3	0	162	1	0	250	0	75
75	0	-63	163	0	151	251	-2	0
76	-7	0	164	-7	0	252	0	-75
77	0	63	165	0	-151	253	1	0
78	11	0	166	1	0	254	0	75
79	0	-57	167	0	151	255	-2	0
80	-1	57	168	-7	0	256	0	-75
81	7	0	169	0	-151	257	1	0
82	0	-51	170	1	0	258	0	75

83	-1	51	**171**	0	151	**259**	-2	0
84	5	0	**172**	-7	0	**260**	0	-75
85	0	-51	**173**	0	-151	**261**	6	0
86	-1	51	**174**	1	0	**262**	0	5
87	5	0	**175**	0	151	**263**	69	75
88	0	-15	**176**	-7	0	**264**	-200	0