COMPUTER AIDED PROGRAMMING

OF A CNC LATHE

by

ROBERT KEITH MOORE

B.Eng., Memorial University of Newfoundland, 1983

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF

THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

in

THE FACULTY OF GRADUATE STUDIES

Department of Mechanical Engineering

We accept this thesis as conforming

to the required standard

THE UNIVERSITY OF BRITISH COLUMBIA

September 1985

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Department of Mechanical Engineering

The University of British Columbia
1956 Main Mall
Vancouver, Canada
V6T 1Y3

Date Oct 7/1985

ABSTRACT

A software package and associated hardware have been developed which gives users of the ORAC CNC Training Lathe an easier and faster method of manufacturing on the lathe. The package, entitled ORACAP, uses the computing power of a mini-mainframe computer, a VAX 11/750, to assist in part design, program generation, program optimization, and program proving for the ORAC Lathe. The package is designed for users unfamiliar with computers such as students in an educational institute or workers in a production facility and uses command procedures to simplify the preparation and execution of the modules of the package. In addition to providing users with a valuable tool for manufacturing, ORACAP also gives users insight into the Computer Aided Design and Computer Aided Manufacturing methods used in industry, that is, it demonstrates the methods which allow production of a part from a very concise description of its geometry. Finally, ORACAP demonstrates the production advantages of a CAD/CAM system over conventional manufacturing methods for small to medium size batch production.

CONTENTS

TABLES

FIGURES

ACKNOWLEDGEMENT

The author would like to thank his supervisor, Dr. F. Sassani, for the guidance and assistance provided during the past two years. Many thanks are extended to Mr. Max Kean who provided much help and made many suggestions for all aspects of this work. Thanks are also due to the VAX Computer System Manager, Mr. Alan Steeves, who provided solutions for many hardware and software problems encountered during this project.

Assistance from the Machine Shop personnel is also greatly appreciated. Special thanks are due to Mr. Leonard Drakes for his expert comment on the design of the tool turret and for the time spent manufacturing it. Also, thanks to Mr. Dave Camp for his participation in the production time studies.

Finally, acknowledgement must be made to the Natural Sciences and Engineering Council of Canada who provided the scholarship which enabled this work.

CHAPTER 1

Introduction

1.1 History of Computer Numerical Control

Since machine tools were first invented, many modifications have been made to enable faster and easier production of machine products. One example of this is the turning lathe. It has evolved from a fully manual operation, which required a highly skilled operator, into a fully computer numerically controlled process. As parts became more complex and tolerances decreased, increased operator skill and time were required. Therefore, more sophisticated machines were needed to meet the demand. This resulted in the development of the automatic lathe which incorporated limit switches to assist part production, and copying machines which copied a master part by tracing its outline. These devices reduced the time and skill requirements while maintaining the flexibility required for small batch jobs. In 1952 the Massachusetts Institute of Technology contributed to the evolution of machine tools by developing the first NC (numerically controlled) milling machine [1]. This again reduced time and skill requirements. Next, the use of computers to directly control many NC machines (DNC - Direct Numerical Control) was introduced. This finally led to the development of micro-processors or computers to control individual NC machines (CNC - Computer Numerical Control). From this, it can be seen that the manufacturing industry has become increasingly reliant on computers and micro-processors for control of manufacturing equipment.

1.2 Advantages of Computer Numerical Control

Since 75% of parts produced by metalworking are in lots of less than 50 pieces [2], it is important to have systems which allow flexible automation. Traditional assembly lines are not flexible and are not capable of manufacturing a wide variety of products. However, NC and CNC machines are highly suitable for producing a variety of products since they allow automation of part production while keeping change-over costs low. Therefore today's trend is away from equipment which can only be used for mass-production of a few products and towards more sophisticated computer controlled equipment which can produce an infinite number of products.

Numerical control is advantageous for small to medium size batch production where similar parts are produced from similar materials using similar manufacturing steps [2]. Numerical Control is particularly suitable for:

1) production of parts with small lot sizes,
2) production of parts with complex geometries,
3) production of parts with close tolerances,
4) production of parts which require many operations,
5) production of parts which require much metal removal,
6) production of parts when changes to the design are likely,
7) production of parts which are expensive,
8) production of parts when 100% inspection is

required.

Use of numerical control in these situations leads to a reduction in production and non-production time, reduced fixturing time, reduced lead time and reduced inspection time. The addition of computer or micro-processor control allows much more flexibility while retaining the advantages of numerical control. In addition to these advantages CNC is capable of performing operations such as three-dimensional contour cutting which can not be performed as well by a manual operator [3]. Thus CNC allows part production with less skilled operators and reduced overhead costs.

1.3 NC and CNC Programming

The first NC programs consisted of simple codes with commands (such as G80, X100, or Y100) which "instructed" the machine to perform a single action or motion. These included G-functions which determined machining modes, M-functions which performed miscellaneous functions, and directional functions which caused motion along a specified axis. Feedrates and spindle-speeds could also be controlled. These instructions allowed such features as linear interpolation, circular interpolation, and thread-cutting.

Programs written in these "languages" could produce parts efficiently, but much time was required for programming and debugging of the NC code.

Gradually, many higher level programming languages, such as APT (Automatically Programmed Tools), were developed to facilitate the generation of the NC code. This allowed easier definition of the machine tool's path by referencing pre-defined points and lines which described the object. However, the path was still chosen by the NC programmer. The CNC machine tools used in many industries today use more sophisticated programming to perform similar functions, but few determine the machine tool's path directly from the drawing or from the CAD (Computer Aided Design) data file. That is, in many instances there is a gap between CAD and CAM (Computer Aided Manufacturing). Such is the case with the ORAC CNC Training Lathe which is currently being used by the Mechanical Engineering Department as well as in many educational institutions across Canada. Therefore, there was a need for a program which would combine Computer Aided Design with Computer Aided Manufacturing to produce a CAD/CAM package for the ORAC Lathe.

The steps necessary for part production on NC equipment can be summarized as follows:

1) Determination of the machining to be done by Numerical Control.
2) Determination of the machining sequence.
3) Creation of the NC code.
4) Verification of the NC code.
5) Execution of the NC code.

The first step can not be handled easily by a computer. However, each of the remaining steps can be accomplished with the assistance of a computer. For the particular case of the ORAC CNC Lathe, a CAD/CAM package has the following requirements:

1) Hardware and software to allow communication between the lathe and a host computer.

2) Software to create the ORAC CNC programs.

3) Software to select the appropriate cutting parameters.

4) Software to optimize the ORAC CNC programs.

5) Software to prove the ORAC CNC programs.

6) A tool turret (changer) to allow the use of more than one tool when machining.

7) An external controller to provide more efficient control over the loading and execution of ORAC CNC programs.

This thesis provides the first six of these requirements. The seventh is provided as part of a concurrent research project.

CHAPTER 2

ORAC CNC Training Lathe

2.1 Description

The ORAC CNC Training Lathe is a small 2-axis, micro-processor controlled lathe. Similarly to most lathes, it allows movement of a tool in two directions in a horizontal plane. These directions are defined relative to the workpiece by x and z directions as shown in Figure 1. The lathe can accommodate bar stock of diameters up to 40 mm and has travels of 95 mm and 350 mm in the x and z directions respectively. It is capable of both linear and circular interpolation and has built-in procedures for thread-cutting. The linear interpolation feature is limited to tapers with ratios less than 20:1. Circular interpolation is limited to radii between 2 and 3000 mm. Threading is limited to pitches between 0.35 mm and 3.5 mm. Feedrates are variable from an unknown lower limit (around 30 mm/min) to 1200 mm/min. Finally, spindle-speeds are variable up to 2000 rpm.

Movement along each axis is accomplished by a stepping motor operated by pulses generated by the micro-processor. Each pulse corresponds to a discrete distance along the axis.

The lathe uses an open-loop control system . This means that there is no position feedback and no compensation for position errors. However, open-loop control has been found to be suitable for many CNC applications and is used frequently since it is less expensive than closed-loop control which does have position feedback.

Top View

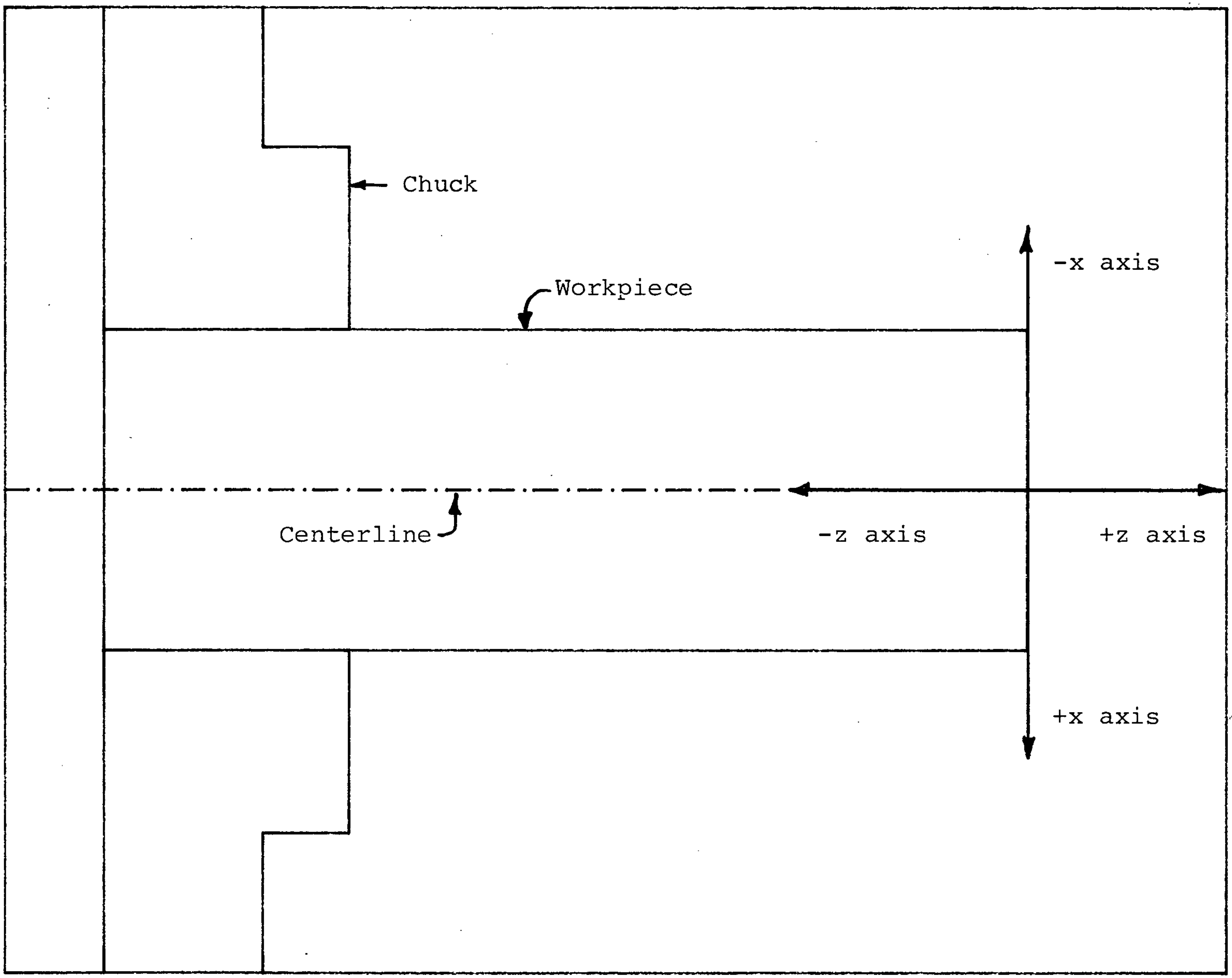


Figure 1. Axes of the ORAC Lathe.

The lathe is designed primarily for operation under micro-processor control, but it has controls for manual operation. These manual controls enable setting of the "floating zero point" and setting of the offsets necessary for each different tool. It also has controls to override the feedrate and spindle-speed during micro-processor controlled operation.

More information on the lathe can be found in the ORAC Programming Instruction and Maintenance Manual [4].

Note: The lathe does not have a zero point. Therefore the zero point must be set by facing the end of the workpiece to determine the z-axis location (i.e. the z-zero point) and a diameter must be turned and measured to accurately determine the location of the axis of rotation (i.e. the x-zero point).

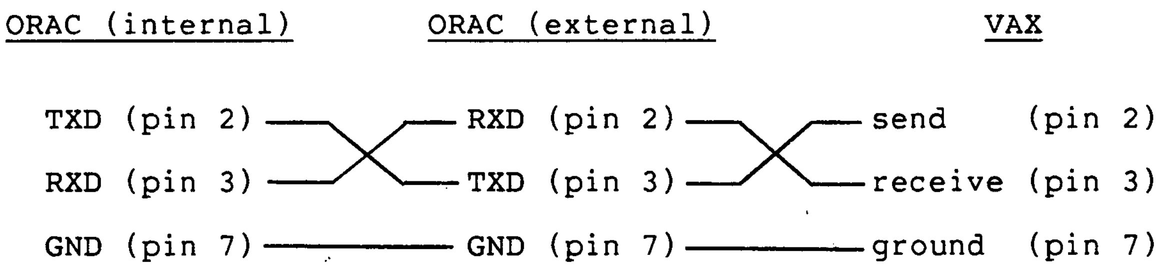
2.2 Interfacing the Lathe and the Computer

The lathe can be connected to a computer or another device by a standard RS 232C serial port, on which ORAC programs can be transmitted or received. The lathe is designed to use hardware handshaking to syncronize the transfer of programs between the two devices. Unfortunately, the Mechanical Engineering Department's VAX 11/750 computer uses only software handshaking which the ORAC does not recognize. Therefore, it was necessary to write several Fortran programs which use the VAX's QIOW (Queued Input / Output Wait) function to transfer data without hardware handshaking. It was found that the VAX could transmit data at 2400 baud which could be received by the lathe with no problems. However, a very efficient program was required to transmit from the lathe to the VAX at 2400 baud without errors.

ORAC programs are written using ASCII characters and are formatted using standard control characters to indicate the start and end of programs, the start and end of ORAC "pages", and the end of lines.

Note: A ORAC "page" refers to a block of data which contains a single machining instruction. It may contain from one to nine lines of information which defines the function to be performed.

The RS 232C port wiring of the lathe was originally slightly non-standard. In order to make connecting hardware more compatible, the transmit and receive wires on the RS 232C connector were switched. Therefore the lathe and the VAX were connected as follows:



where: TXD = Line on which data is transmitted.

RXD = Line on which data is received.

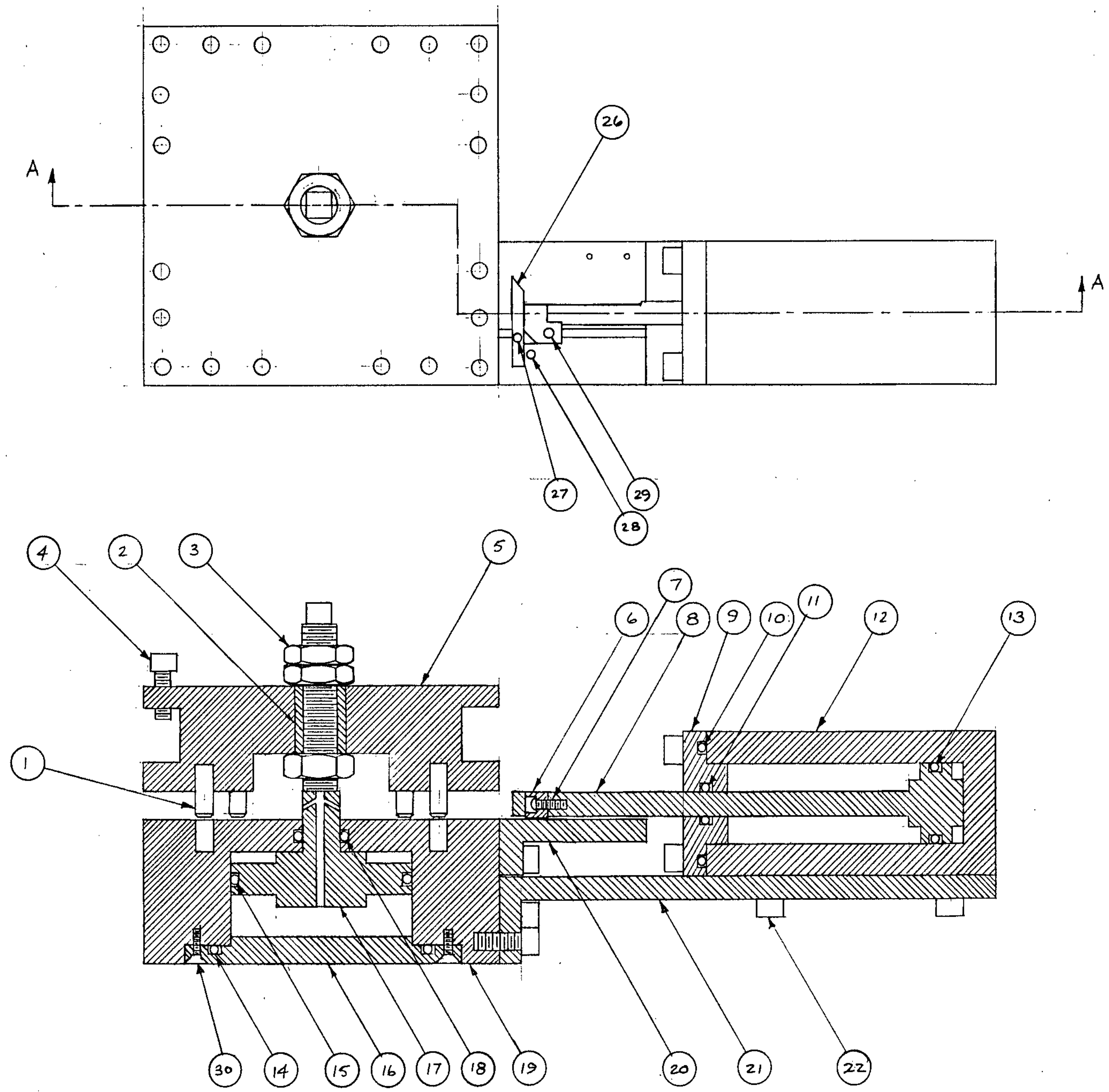
Pins 6 and 20 which are normally used by the ORAC Lathe for hardware handshaking were not used.

2.3 Modifications to the Lathe

Initially the ORAC Lathe was equipped with a quick-change toolpost and holder. However, in order to facilitate production, a toolholder or tool turret with a number of tools was required. Therefore, a pneumatic tool turret which holds four tools in any of its eight positions, equally spaced at 45 degrees, was designed. The tool turret was mounted on the cross-slide in place of the quick-change toolpost and top-slide. It is rotated in the x-z (horizontal) plane by a "rotate cylinder" and can be clamped down by the "clamp cylinder" in any of the eight positions. The "as built" design drawings of the tool turret are included in Figures 2, 3, 4 and 5.

The tool turret may be controlled by the auxiliary outputs from the lathe or by a separate interface board controlled by the VAX computer according to signals received from the auxiliary outputs. The tool turret design also incorporates five micro-switches which can be read by either the auxiliary input of the lathe or by the interface board. Holes have been drilled in the bottom of the tool holder in patterns which allow three micro-switches to indicate the position of the tool holder when clamped. The holes are positioned such that when the tool holder is clamped down the different open and closed combinations of the three switches indicate the position as a three digit binary number. The other two switches indicate the position of the clamp and rotate cylinders. One switch is activated when the clamp cylinder is down and the second indicates when the rotate cylinder is fully retracted.

PART NO	NAME OF PART	MATERIAL	NO. REQ'D
1	TOOL HOLDER PIN	MILD STEEL	8
2	TOOL HOLDER BUSHING		1
3	CLAMP CYLINDER PISTON NUT		3
4	10-24 x ¾ ALLEN SCREW		20
5	TOOL HOLDER		1
6	ROTATE CYLINDER PISTON END		1
7	4-40 UNC ALLEN SCREW		1
8	ROTATE-CYLINDER PISTON		1
9	ROTATE-CYLINDER CAP		1
10	#122 O-RING		1
11	#110 O-RING		1
12	ROTATE-CYLINDER BODY		1
13	# 115 O-RING		1
14	# 139 O-RING		1
15	# 133 O-RING		1
16	CLAMP-CYLINDER CAP		1
17	CLAMP-CYLINDER PISTON		1
18	# 112 O-RING		1
19	CLAMP-CYLINDER BODY		1
20	ROTATE-CYLINDER PISTON SUPPORT		1
21	ROTATE-CYLINDER SUPPORT		1
22	10-24 x ½" ALLEN SCREW		13
23	10-24 x 1" ALLEN SCREW (not shown)		4
24	10-24 UNC NUT (not shown)		4
25	2-56 UNC SCREW (not shown)		6
26	ROTATE-CYLINDER PISTON END LEVER		1
27	ROTATE-CYLINDER PISTON END LEVER PIN		1
28	ROTATE-CYLINDER SUPPORT PIN		1
29	ROTATE-CYLINDER PISTON END PIN		1
30	4-40 UNC SCREW		4



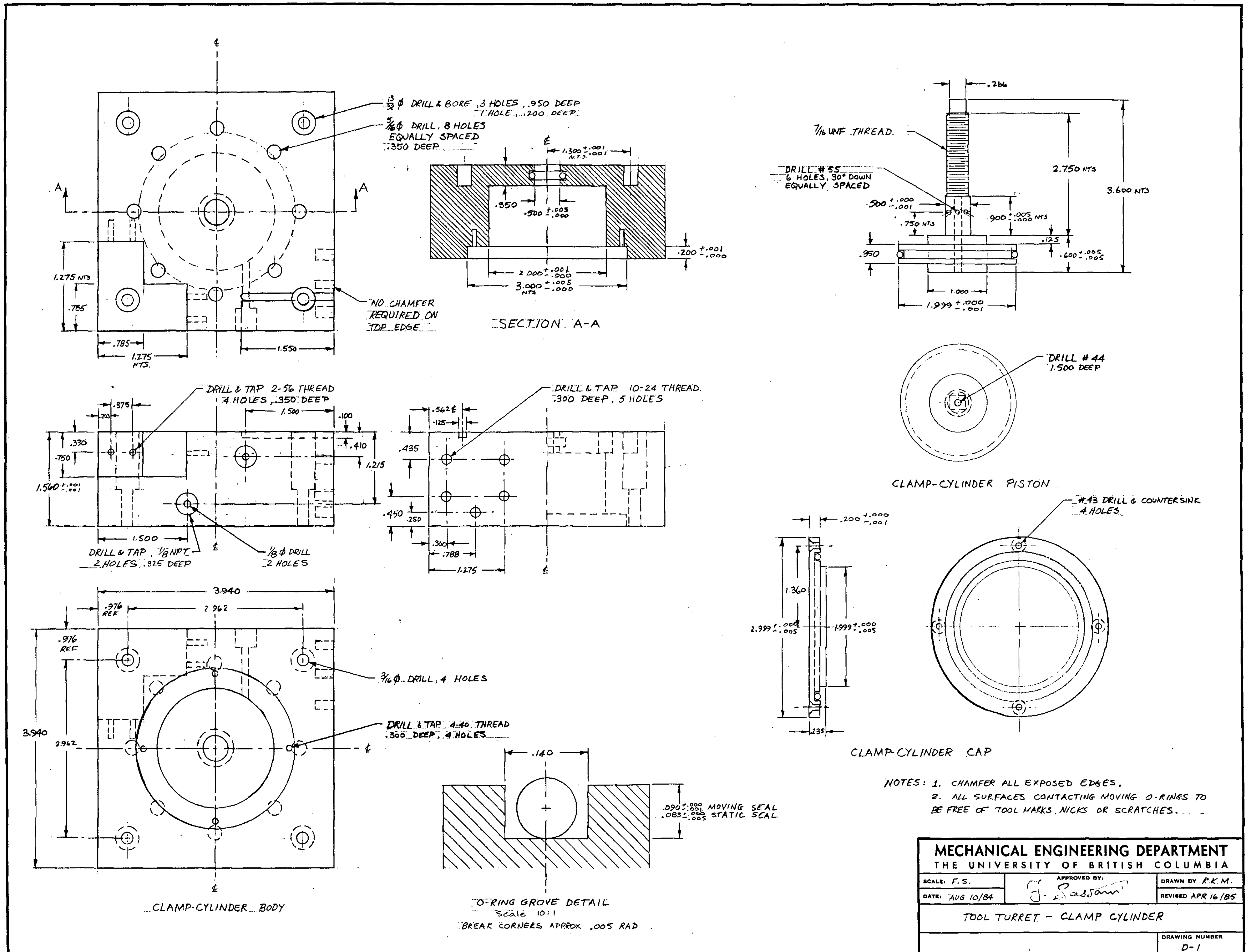
SECTION A-A

MECHANICAL ENGINEERING DEPARTMENT
THE UNIVERSITY OF BRITISH COLUMBIA

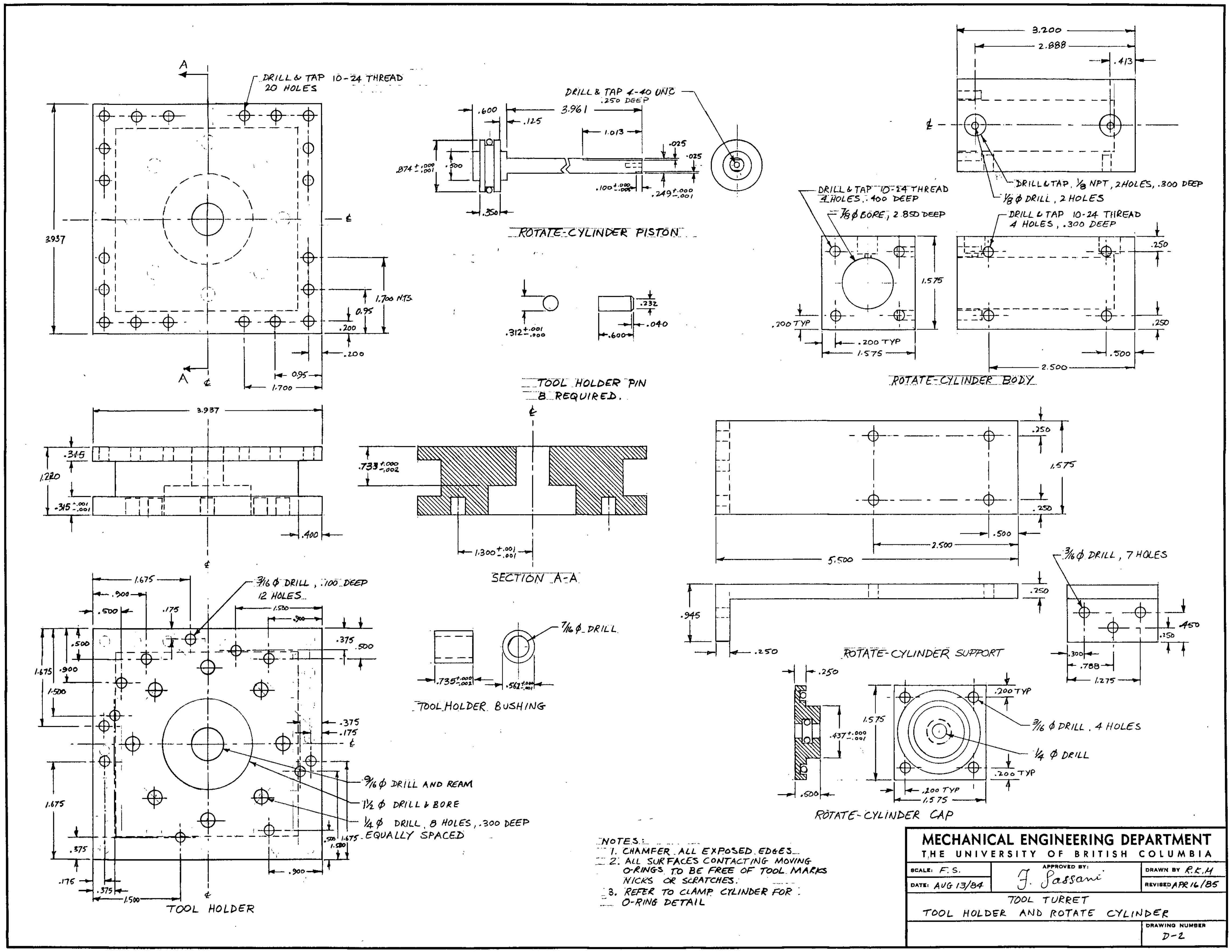
SCALE: F.S.	APPROVED BY: F. Sassani	DRAWN BY RKM
DATE: AUG 15/84		REVISED APR 22/85

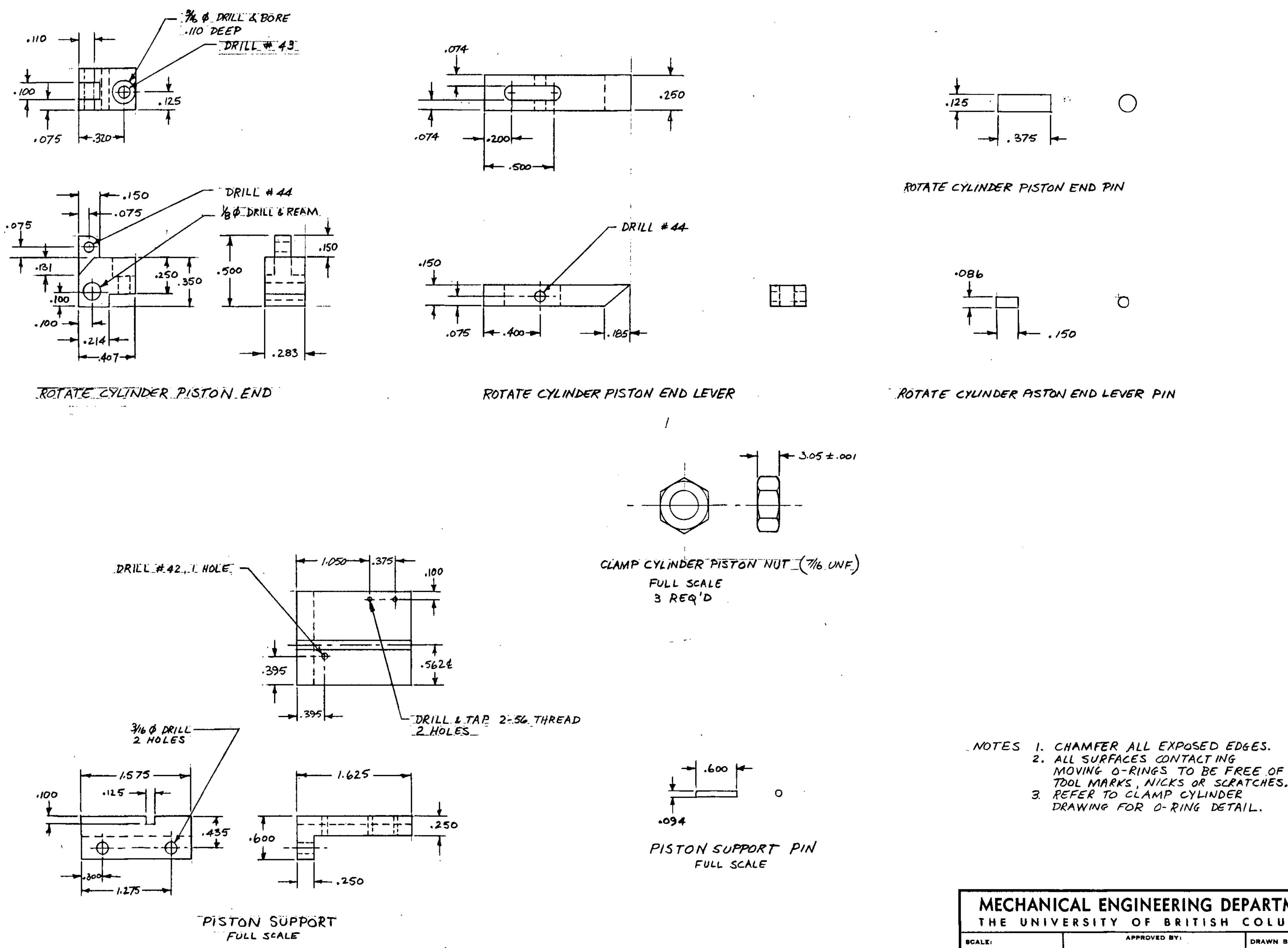
TOOL TURRET
ASSEMBLY DRAWING

DRAWING NUMBER A-1



SECTION A-A
CLAMP-CYLINDER BODY
O-RING GROVE DETAIL
Scale 10:1
BREAK CORNERS APPROX .005 RAD
CLAMP-CYLINDER PISTON
CLAMP-CYLINDER CAP
7/16 UNF THREAD.
DRILL # 44
1.500 DEEP
#43 DRILL & COUNTERSINK
4 HOLES
NO CHAMFER
REQUIRED ON
TOP EDGE
3/16 Ø DRILL, 4 HOLES
DRILL & TAP 4-40 THREAD
.300 DEEP, 4 HOLES
DRILL & TAP 2-56 THREAD
4 HOLES, .350 DEEP
DRILL & TAP 10-24 THREAD.
.300 DEEP, 5 HOLES
DRILL & TAP 1/8 NPT
2 HOLES .325 DEEP
1/8 Ø DRILL
2 HOLES
MOVING SEAL
STATIC SEAL
NOTES: 1. CHAMFER ALL EXPOSED EDGES.
2. ALL SURFACES CONTACTING MOVING O-RINGS TO BE FREE OF TOOL MARKS, NICKS OR SCRATCHES.
MECHANICAL ENGINEERING DEPARTMENT
THE UNIVERSITY OF BRITISH COLUMBIA
SCALE: F.S.
APPROVED BY:
DRAWN BY R.K.M.
DATE: AUG 10/84
REVISED APR 16/85
TOOL TURRET - CLAMP CYLINDER
DRAWING NUMBER
D-1


DRILL & TAP 10-24 THREAD 20 HOLES
TOOL HOLDER
SECTION A-A
ROTATE-CYLINDER PISTON
DRILL & TAP 4-40 UNC .250 DEEP
TOOL HOLDER PIN 8 REQUIRED.
TOOL HOLDER BUSHING
7/16 ϕ DRILL
ROTATE-CYLINDER BODY
DRILL & TAP 1/8 NPT, 2 HOLES, .300 DEEP
1/8 ϕ DRILL, 2 HOLES
DRILL & TAP 10-24 THREAD 4 HOLES, .400 DEEP
7/8 ϕ BORE, 2.850 DEEP
DRILL & TAP 10-24 THREAD 4 HOLES, .300 DEEP
ROTATE-CYLINDER SUPPORT
3/16 ϕ DRILL, 7 HOLES
ROTATE-CYLINDER CAP
3/16 ϕ DRILL, 4 HOLES
1/4 ϕ DRILL
3/16 ϕ DRILL, .100 DEEP 12 HOLES
9/16 ϕ DRILL AND REAM
1½ ϕ DRILL & BORE
1/4 ϕ DRILL, 8 HOLES, .300 DEEP EQUALLY SPACED
NOTES:
1. CHAMFER ALL EXPOSED EDGES
2. ALL SURFACES CONTACTING MOVING O-RINGS TO BE FREE OF TOOL MARKS NICKS OR SCRATCHES.
3. REFER TO CLAMP CYLINDER FOR O-RING DETAIL
MECHANICAL ENGINEERING DEPARTMENT
THE UNIVERSITY OF BRITISH COLUMBIA
SCALE: F.S.
DATE: AUG 13/84
APPROVED BY:
DRAWN BY R.K.H
REVISED APR 16/85
TOOL TURRET
TOOL HOLDER AND ROTATE CYLINDER
DRAWING NUMBER D-2



NOTES
1. CHAMFER ALL EXPOSED EDGES.
2. ALL SURFACES CONTACTING MOVING O-RINGS TO BE FREE OF TOOL MARKS, NICKS OR SCRATCHES.
3. REFER TO CLAMP CYLINDER DRAWING FOR O-RING DETAIL.

MECHANICAL ENGINEERING DEPARTMENT
THE UNIVERSITY OF BRITISH COLUMBIA

SCALE:	APPROVED BY:	DRAWN BY RKM
DATE: APR 18/85		REVISED

TOOL TURRET PARTS

DRAWING NUMBER D-3

The tool turret is operated by activating an 1/8" pneumatic valve which causes the clamp cylinder to lift the tool holder. While this cylinder is in the "up" position the second pneumatic valve may be operated and released. Each operate / release cycle causes the rotate cylinder to rotate clockwise by 45 degrees. After the tool holder has been rotated to the desired position, the clamp cylinder valve may be released, causing the clamp cylinder to return to the "down" position. When in this position, the tool holder is rigid, and the tool can then be used for machining.

The air flow to the cylinders is regulated by a flow-control valve which limits air flow to the entire system, and several check valves which allow sufficient air flow to operate the cylinders slowly. Slow operation of the piston is necessary in order to prevent damage to the rotate-cylinder piston shaft. Air is also directed through the clamp-cylinder piston over the lower surface of the tool turret when the clamp-cylinder piston is in the "up" position. This prevents metal chips from interfering with the proper seating of the tool when clamped. Refer to Figure 6 for a schematic of the pneumatic system. The air pressure must be maintained throughout the machining operation in order to keep the tool holder properly seated.

The tool turret was designed to hold the following tools:

1) A right-hand turning / facing tool which would be used for most machining operations.
2) A left-hand turning tool for areas inaccessible by the right-hand tool.
3) A threading tool for threading.

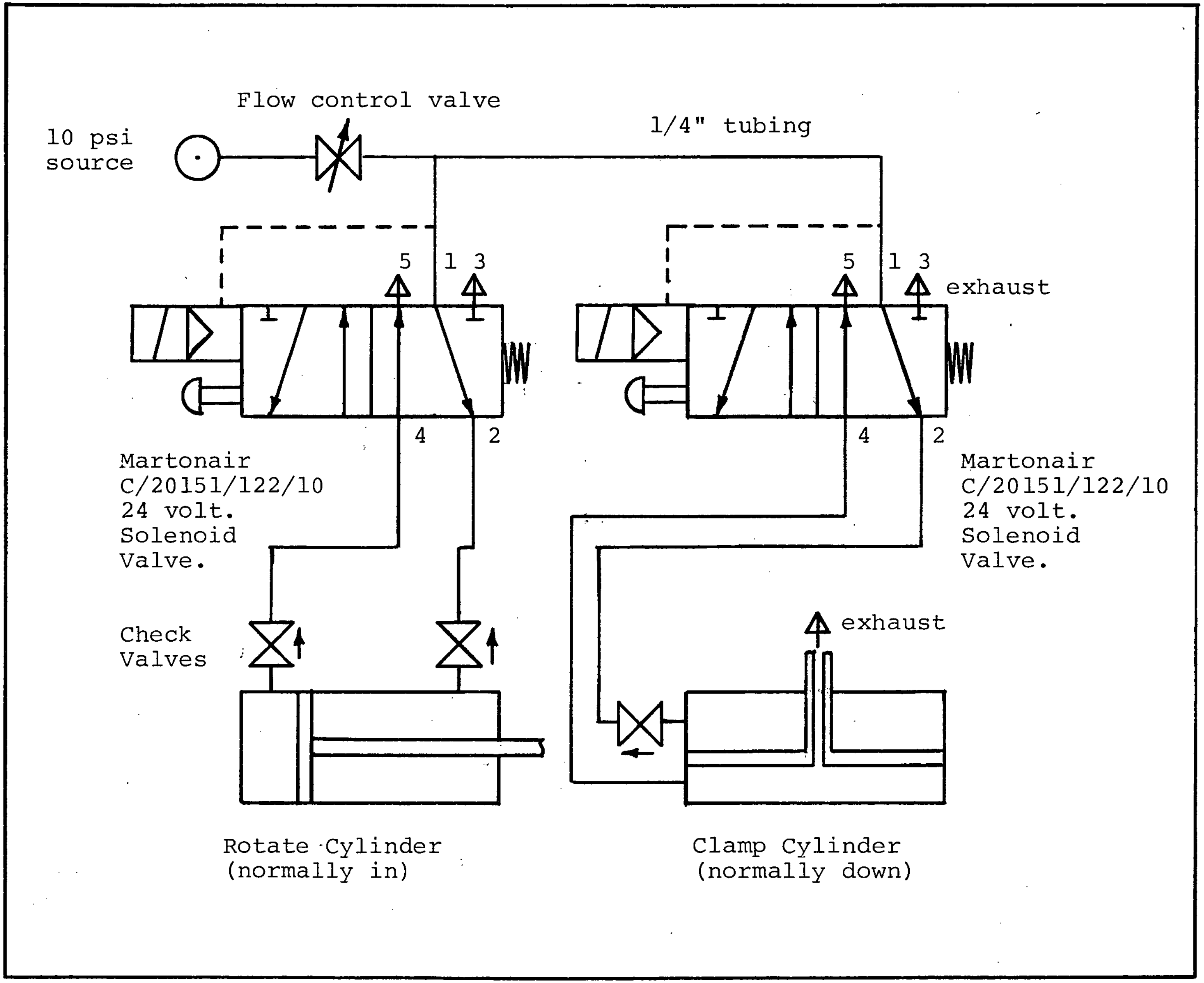


Figure 6. Schematic of Tool Turret Pneumatics.

4) A parting-off tool to part-off the object following machining.

However, the lathe was found to have a minimum feedrate which was too high for parting-off. Therefore, it was decided to use the fourth position for a second right-hand turning / facing tool which could be used for the initial facing of the rough bar stock. The final configuration for the tool holder is:

1) A right-hand turning / facing tool.
2) A left-hand turning tool.
3) A threading tool.
4) A right-hand turning / facing tool for rough cuts.

2.4 Programming the Lathe

Unlike most NC code, the CNC code used for ORAC programs looks much like that of a computer program. It consists of 17 instructions which determine the operation to be performed. Each of these instructions is stored on a separate ORAC "page". Up to 99 "pages" may be combined to form an "ORAC program". The ORAC program can then be executed page by page to produce the desired part. The instruction set, abbreviations, and purposes are listed below:

	Instruction	Purpose
INC	incremental format	Incremental programming mode
ABS	absolute format	Absolute programming mode

IN	inch units	Programming in inches
MM	millimeter units	Programming in millimeters
PDAT	program datum	Set location of program datum
PTP	point-to-point interpolation	Move by linear interpolation
CIRC	circular interpolation	Move by circular interpolation
THRD	threading	Create specified thread
DWEL	dwell	Delay program execution
AUXI	auxiliary input	Read auxiliary input signals
AUXO	auxiliary output	Create auxiliary output signals
CALL	call subroutine	Call appropriate subroutine
SUB	subroutine start	Start of subroutine
ESUB	end subroutine	End of subroutine
DO	start do-loop	Start of do-loop
EDO	end do-loop	End of do-loop
END	end program	End of program

Note: Guidelines must be followed when writing ORAC programs. They are contained in the ORAC Programming Instruction and Maintenance Manual [4].

2.5 Setup of the Lathe

Prior to execution of ORAC programs, it is necessary to define the zero position, in the x-z plane, for each tool to be used. This is accomplished by touching or machining a face and a known diameter, with each tool, as instructed by the lathe. The difference in the zero position of the tool and the zero position of the reference tool 0, determines the tool offset for that tool. Subsequently all zero positions may be redefined by relocating the zero position of tool 0 provided that the tool offsets have not been lost or changed. For future use, the tool offsets may be recorded on a tape using the ORAC's mini-cassette recording unit. This allows easier setup of the tools, since all that is required then is to define the zero position for tool 0, and reload the saved tool offsets from the tape, as instructed by the lathe.

CHAPTER 3

Overview of ORACAP Program

3.1 ORACAP Options

ORACAP is an acronym for ORAC Computer Aided Programming. It is a software package, developed during the course of this work, which allows easier and faster use of the ORAC CNC Lathe. The package consists of ten options which are executed by running a controlling command procedure. Nine of these options are Fortran programs and the remaining option is a command procedure. The options are presented in an option menu and are executed when selected as explained in Chapter 4. The options of the package allow the user to:

1) Input or change the dimensions of an object to be machined on the lathe (i.e. the "DESIGN" Option).
2) Generate the CNC instructions necessary to machine the part (i.e. the "GENERATE" Option).
3) Optimize the CNC instructions by eliminating extra instructions and combining inefficient ones (i.e. the "OPTIMIZE" Option).
4) Assemble the CNC instructions into an ORAC program or programs (i.e. the "ASSEMBLE" Option).
5) Create a Fortran program or programs to prove the ORAC program or programs (i.e. the "PROVE" Option).
6) Compile, link and run the program or programs to

prove the ORAC program (i.e. the "PLOT" Option).

7) Down-load the ORAC program or programs from the computer to the lathe (i.e. the "VAXTOORAC" Option).
8) Up-load an ORAC program from the lathe to the computer (i.e. the "ORACTOVAX" Option).
9) Compress an ORAC program into the corresponding CNC instructions (i.e. the "COMPRESS" Option).
10) Write CNC instructions directly (i.e. the "WRITE" Option).

3.2 ORACAP Capabilities and Limitations

The ORACAP program is capable of handling objects to be externally machined on one or both ends without the use of the tailstock. The program includes all ORAC machining features such as linear interpolation, circular interpolation, and threading. These operations are limited by the same constraints as for the ordinary ORAC programming.

The ORAC lathe is capable of handling arcs up to 180 degrees in one program page. However, ORACAP requires circular sections to be contained in only one quadrant. This eliminates several problems in the plotting subroutines.

Also objects with parabolic sections can be machined. This is accomplished by approximating the parabolic curve with straight line segments.

The program also allows control of the automatic tool turret, if desired, and can generate the instructions necessary for parting-off. However, the lathe can not machine at the slow speed necessary for parting-off. Therefore, this feature can not be used.

Since the ORAC lathe is limited to 99 program steps, it is sometimes necessary to create more than one program to machine a desired part. Therefore, for each end, various parts of the machining instructions are created and stored in separate files and then assembled into the minimum number of ORAC programs. Each ORAC program is contained in a separate file.

All ORACAP options can use either Imperial or Metric units. However, the "GENERATE" program creates ORAC programs which use only metric measurements. The "PROVE" and "DESIGN" programs have been written so that they can use either system of measurement. In all other options the type of units is unimportant.

3.3 Instruction and Program Formats

In order to save disk storage space and reduce execution time, the "DESIGN", "GENERATE", "OPTIMIZE" and "WRITE" options store the machining instructions in a compressed format, similar to that used for NC instructions. Therefore, they are refered to as "CNC instructions". In this format, each ORAC "page" occupies only one line. The "PROVE", "VAXTOORAC" and "ORACTOVAX" options operate on "ORAC Programs" where one "page" may occupy one to nine lines. The "ASSEMBLE" and "COMPRESS" options allow conversion from one format to the other.

3.4 Use of Subdirectories

ORACAP uses the VAX's subdirectories to separate files into groups with similar functions. The subdirectories and their contents are:

"[ME258.LATHE.COM]" - Command procedure files which enable the setup and running of ORACAP.

"[ME258.LATHE.FOR]" - Fortran source code, object files, and the task images for each Fortran option.

"[ME258.LATHE.DAT]" - Files created or used by the "DESIGN" option of ORACAP.

"[ME258.LATHE.CNC]" - Files containing CNC instructions and ORAC programs.

"[ME258.LATHE.PLT]" - Fortran source code, object files, and task images created by the "PROVE" and "PLOT" options of ORACAP.

These directory names are generated by ORACAP when needed. The ".DAT", ".CNC", and ".PLT" directories will not be used for output if the "directory" mode is not selected as explained in section 4.3.

3.5 Terminals

ORACAP has been designed to run primarily on a Digital VT101 terminal with Retro-Graphics enhancements which allow it to emulate a Tektronix 4027 terminal. However, most options can be executed on any standard computer terminal. Plotting during the "DESIGN" and "GENERATE" options must be done on a terminal or plotter which can interpret TX 4010 or 4027 plotting signals. Plotting for the Proving option can also be done on a terminal which can interpret TX 4010 or 4027 plotting signals or on an HP 7221A plotter.

Note: ORACAP and the Fortran options generate escape sequences which are used for various VT100 and VT101 display functions. Therefore, execution of the programs on HP or TX terminals will result in the escape sequence characters being displayed on the screen. The characters will have no effect on program execution.

CHAPTER 4

ORACAP Program Execution

4.1 Use of Example Parts

This chapter describes the various parts of the ORACAP package and their execution. Throughout, a typical part, which can be machined on a lathe, is used to illustrate formats and for clarification of the filenames being used. One of the parts chosen for these examples was a Queen from a chess set. It was chosen since it includes a machining problem (i.e. an area which is inaccessible by a right-hand turning / facing tool) and also since it requires more than 99 steps to machine the part. Since several final ORAC programs are necessary, this example shows how a complex part would be handled by the ORACAP program. A summary of files associated with this part and their sizes is contained in Appendix A. Appendix B contains examples of some of these files. However, the chess piece example does not demonstrate several features, such as parabolic sections and threading. Therefore these features are illustrated using an ORACAP Master Part. The summary table for this part is contained in Appendix C while the example files are contained in Appendix D.

4.2 Setup of ORACAP Program Modules

Prior to running the "ORACAP.COM" command procedure, all Fortran subroutines must be compiled and placed in the library file "[ME258.LIBRARY]LATHELIB.OLB". This can be accomplished using the command procedure "ADDLIB.COM" which is used by typing "ADD "filename"". If all subroutines are to be compiled, the command procedure "ADDALL.COM" may be used. All Fortran programs must be compiled and linked with these subroutines. The task images must be created in the directory "[ME258.LATHE.FOR]". Again, this can be done individually using the command procedure "COMPILE.COM" which can be executed by typing "COMP "filename"". All programs can be compiled using the "COMPALL.COM" command procedure. Finally, all of the above operations can be accomplished by typing "SETUP ""account", where "account" is the account name (i.e. "ME258"). This executes the command procedure "SETUP.COM". These procedures have already been executed and will only have to be executed if a task image has been deleted. These command procedures are contained in the "[ME258.LATHE.COM]" directory and are listed in Appendix F on the enclosed microfiche.

Note: In order to set up the ORACAP package in a different account, all references to the "ME258" directories must be changed. Also, the subroutine FILENAME will have to be modified to properly create the new directory names.

4.3 Program Modes and Filenames

"ORACAP.COM" can then be executed by typing "ORACAP". The program first prompts for a partname or a <CR> (carriage return). The response is stored in the file "PARTNAME.TMP", and it is used to determine the program mode. If a partname is entered, "part" mode will be used, and the Fortran programs will use this name to create the filenames needed for the various data files. "Part" mode is used for executing the ORACAP options sequentially or independently. If <CR> is entered, "file" mode will be used, and the subsequent programs will prompt for the necessary filenames or required information. "File" mode is used primarily for special applications, such as creating a particular file, by-passing an undesirable option, or for debugging.

The program then asks the user if he wishes to include directory names in the file specifications when they are created. If this "directory" mode is selected, a specific directory name such as "[ME258.LATHE.DAT]" is placed in front of the filename so that the files are created in the specified directory. If the "directory" mode is not selected the current directory is used to contain created files. This mode can only be used when executing the program from an account which has "read" and "write" access to the ME258 account. This mode is independent of the "part" and "file" modes.

The examples used throughout this thesis assume that the "directory" mode is chosen and that the ME258 file specifications are used. That is, the examples give the names of the directory specifications which will be used if this mode is chosen and if in "file" mode, no directory specification is given.

The default format for file specifications created in "part" mode is "Directory" + "Partname" + "End" + "File" + "Extension", where:

"Directory" is a directory name.

(eg. "[ME258.LATHE.DAT]")

"Partname" is the partname which may contains up

to 7 characters. (eg. "QUEEN")

"End" is a letter denoting the end. (eg. "A")

"File" is a digit denoting one of a series of files.

(eg. "1")

"Extension" is a 4 character file extension.

(eg. ".CAD")

The default format for file specifications created in "file" mode is "Directory" + "Filename" + "Extension", where:

"Directory" is a directory name.

(eg. "[ME258.LATHE.DAT]")

"Filename" is the filename which may contains up to 9 characters. (eg. "QUEENA1")

"Extension" is a 4 character file extension.

(eg. ".CAD")

Note: "End" and "File" are automatically generated by the program in "part" mode when required. They are not required for the "ORACTOVAX", "WRITE", or "COMPRESS" options, and therefore the files created by "part" and "file" modes are the same. Also, when running options independently in the "file" mode, it may be necessary to include the required "End" or "File" in the filename, if the files were created in the "part" mode, or are to be used by a program running in "part" mode.

4.4 Terminal Selection

Next, ORACAP requests the user to select a terminal type. There are four choices:

1) HP - HP terminal / plotter; for proof plotting on the HP 7221 plotter.
2) RG - VT101 in Retro-Graphics 4027 mode; for VT101 terminals with Retro-Graphics enhancements which emulate a Tektronix 4027 terminal.

3) TX - VT100 or VT101 in Retro-Graphics 4010 mode; for VT terminals with Retro-Graphics enhancements which emulate a Tektronix 4010 terminal. This mode is also used when the TX 4662 plotter is connected to one of these terminals.

4) VT - VT100 or VT101; for terminals with no graphics.

The selected terminal type is stored in the file "TERMINAL.TMP". The RG terminal mode is most desireable when executing ORACAP since it displays more information to the user. The examples use the TX mode to allow plotting on the TX 4662 plotter.

4.5 Option Selection

ORACAP then allows selection of an option. The options and the programs executed by selecting the option are:

OPTION MENU	PROGRAM
0) RESTART PROGRAM	
1) DESIGN SHAPE TO BE MACHINED	DESIGN.FOR
2) GENERATE ORAC INSTRUCTION FILE	GENERATE.FOR
3) OPTIMIZE ORAC INSTRUCTION FILE	OPTIMIZE.FOR
4) ASSEMBLE INSTRUCTION FILE INTO PROGRAM	ASSEMBLE.FOR
5) PROVE ORAC PROGRAM	PROVE.FOR
6) PLOT PROOF OF ORAC PROGRAM	PLOT.COM
7) TRANSFER PROGRAM FROM COMPUTER TO ORAC	VAXTOORAC.FOR

8) TRANSFER PROGRAM FROM ORAC TO COMPUTER — ORACTOVAX.FOR

9) COMPRESS ORAC PROGRAM — COMPRESS.FOR

10) WRITE ORAC INSTRUCTION FILE — WRITE.FOR

11) EXIT FROM PROGRAM

Note: The listings of these Fortran programs can be found in Appendix F on the enclosed microfiche.

Following execution of an option, the user is prompted for the next option. Valid responses are:

"Y", which continues with the next option.

"N", which returns to the main menu.

"E", which exits from the ORACAP program.

"1" - "11", which executes the entered option.

4.5.1 DESIGN Option

The "DESIGN.FOR" Fortran program enables the user to create data files which describe the part to be machined. It first asks if a new part is to be created. If so, the type of material, diameter of bar stock, length of the final part, and the number of ends to be machined, must be entered. Also, at this point the Parting-off option or signal generation for automatic machining may be selected. Parting-off is the separation of the finished piece from the remaining bar stock. Automatic machining is using an external micro-processor, currently being designed in a separate project, to control the lathe and tool changer.

Next, the part dimensions must be entered. The program assumes that the user has a drawing of the part, dimensioned in a way similar to the examples shown in Figures 7 and 8. If the object is to be machined at only one end, it is dimensioned from from a datum at the end of the part which will be next to the chuck. If the object is to be machined on both ends, it is dimensioned from a datum at the end of the part to be clamped first (i.e. the end which is to be machined last). The dimensions measured along the z-axis of the part are referred to as positions. At each position, the diameter must be known. In order to input these dimensions, the object is divided into sections. The positions and diameters, as well as any other required information, can then be given for each section.

There are 6 section types which are identified by the numbers in the following list:

1) Linear sections which can be described by two positions and two diameters.
2) Concave (circular) sections which can be described by two positions, two diameters, and a radius.
3) Convex (circular) sections which can also be described by two positions, two diameters, and a radius.
4) Threaded sections which require two positions, a base diameter, an outside diameter, and a pitch.
5) Parabolic sections constructed from parabolas whose axes are parallel to the z-axis which can be described by three positions and three diameters.
6) Parabolic sections constructed from parabolas

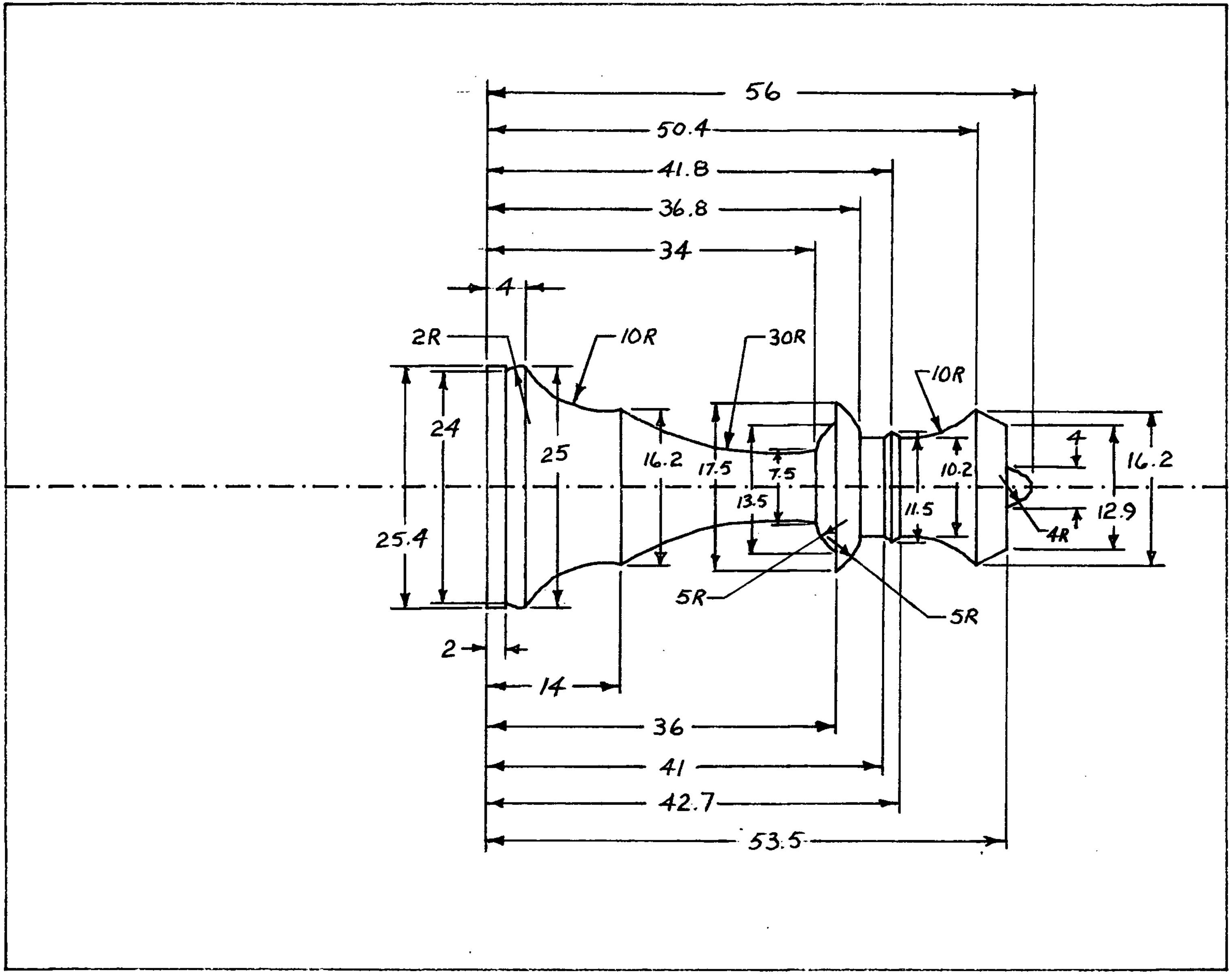


Figure 7. Dimensioned Drawing of Example Chess Piece.

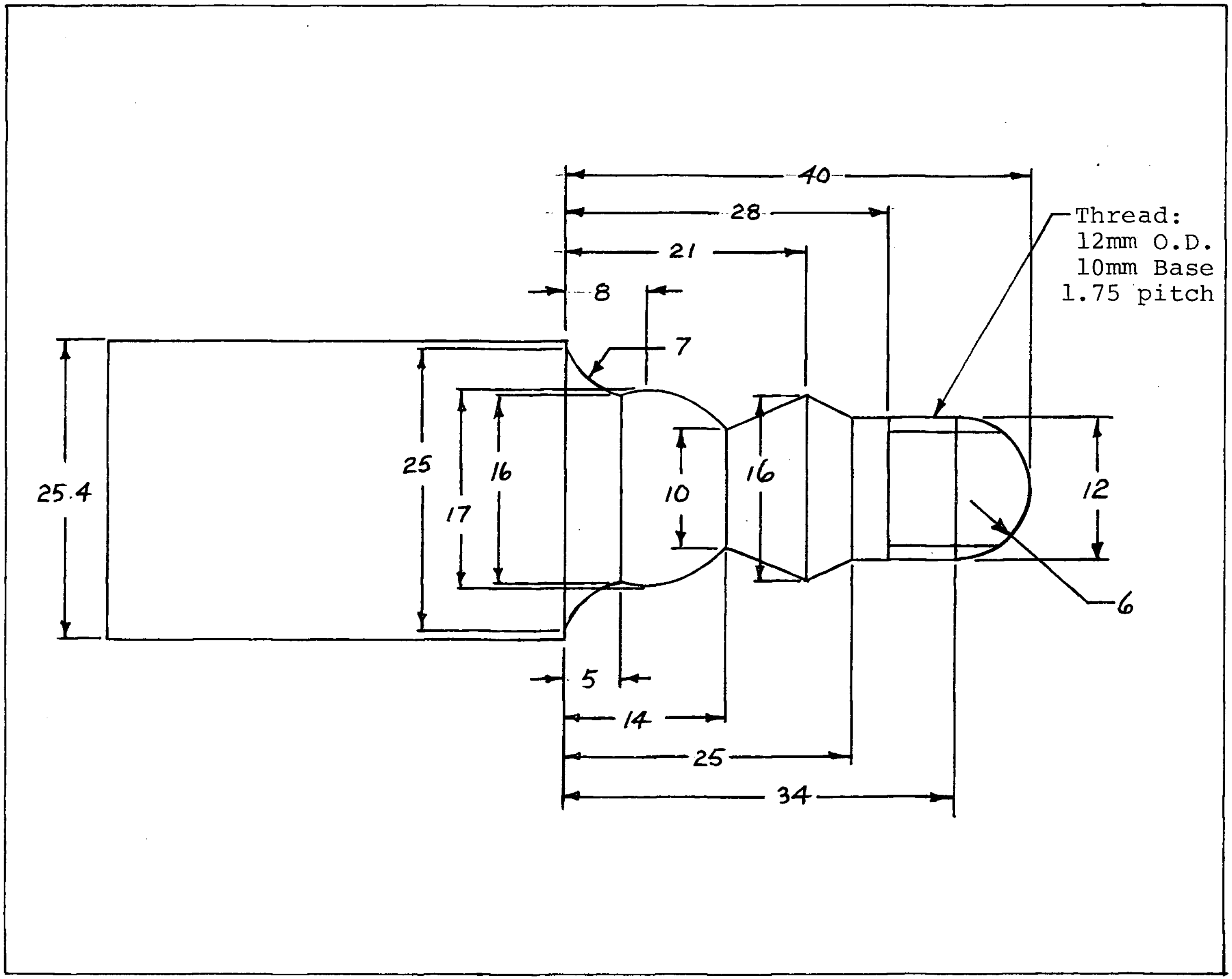


Figure 8. Dimensioned Drawing of ORACAP Master Piece.

whose axes are parallel to the x-axis which can also be described by three positions and three diameters.

For parabolic sections, the first and second points define the start and end locations. The third point can be any other point on the parabola, including points which are represented by negative positions or diameters.

For many parts, the starting dimensions of a section are the same as the ending dimensions of the previous section. Therefore, input subroutines are used to facilitate the entry of the data. The ending dimensions of the previous section are shown on the terminal and may be selected by pressing the return key. Once the part dimensions have been input, they can be stored in a data file, checked for inaccessible areas (by the subroutine "CHECK"), and / or plotted for verification (by the subroutine "MAINPLOT"). Examples of such plots are shown in Figures 9 and 10. Any individual section, or all sections can then be changed. Again the input subroutines are used. The old dimensions, which are shown on the screen, may be entered by pressing the return key, or new values may be entered in their place. This new data may then be stored, checked, and / or plotted as before.

The data for the part is stored in the following data files:

"[ME258.LATHE.DAT]partname.CAD" - This file contains the number of sections, the dimensions of all sections, as well as the material, and flags indicating parting-off or signal generation options.

ORACAP Option #1 - Design

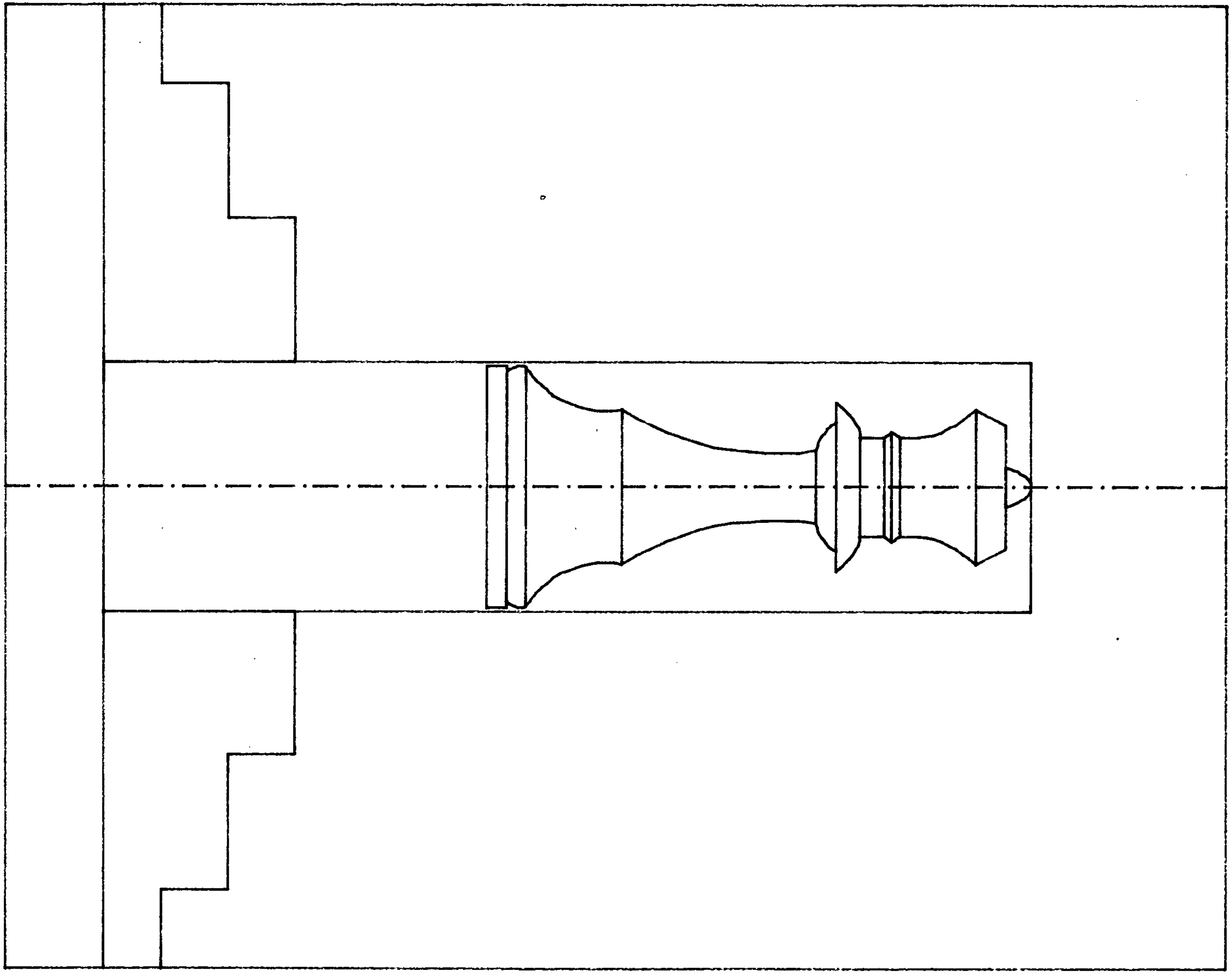


Figure 9. Plot of Example Chess Piece Sections.

ORACAP Option #1 - Design

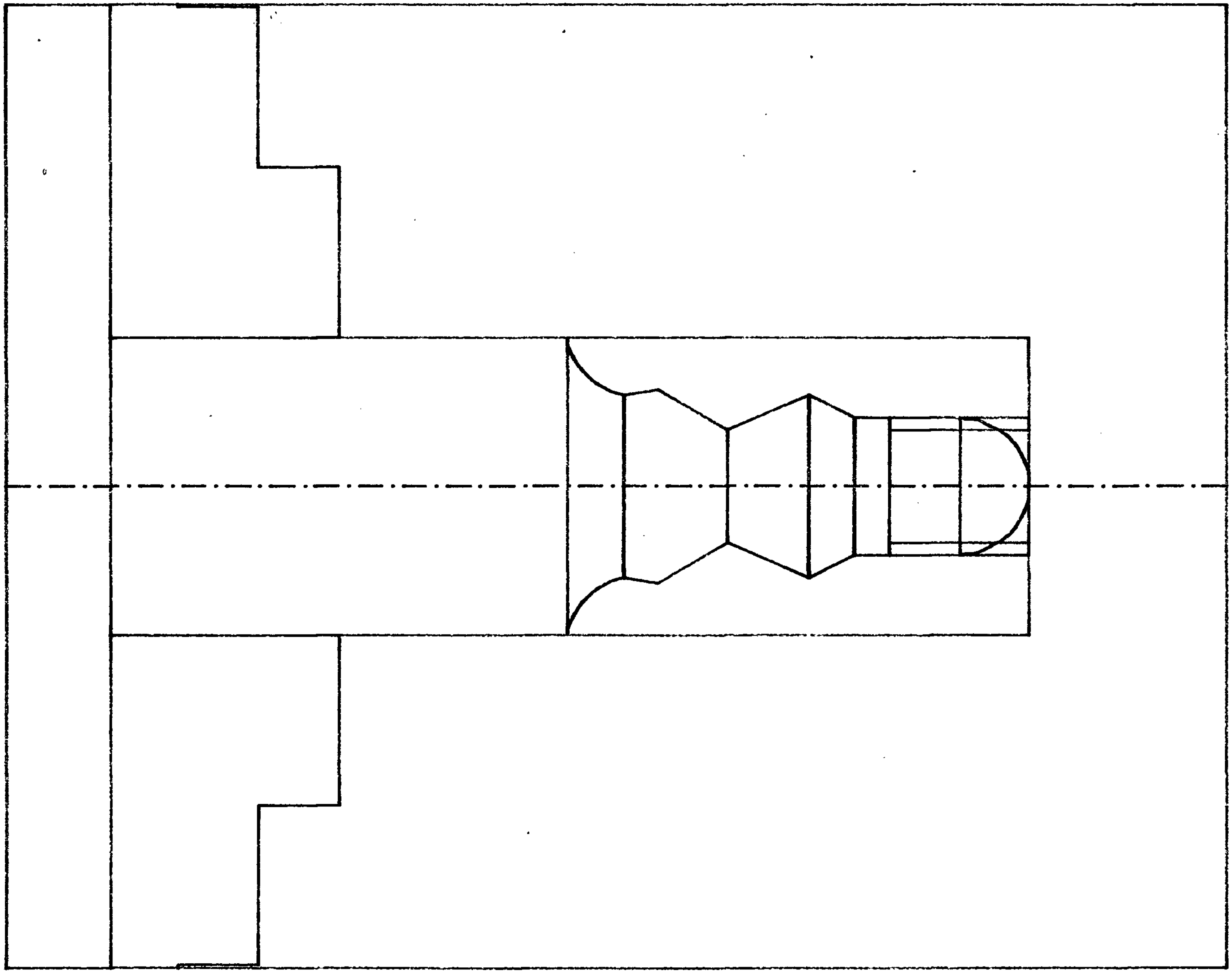


Figure 10. Plot of ORACAP Master Part Sections.

"[ME258.LATHE.DAT]partname.SIZ" - This file contains the diameter of the bar stock, and the length of the final part.

"[ME258.LATHE.DAT]partname.END" - This file contains the number of ends to be machined.

Note: These files are used as the input files for subsequent modifications. The ".CAD", ".SIZ", and ".END" file for the chess piece can be found in Appendix B, and are named "QUEEN.CAD", "QUEEN.SIZ", and "QUEEN.END" respectively. The corresponding files for the ORACAP Master Part example are "MASTER.CAD", "MASTER.SIZ", and "MASTER.END" and are contained in Appendix D.

If both ends of the part are to be machined, the ".CAD" file is split and two new files are created. The "split point" (or position which divides the object into two ends) may be entered or determined by the program. If it is determined by the program, it is placed at the beginning of the first section which is at least 30 mm from the clamped end of the object. This allows sufficient space (10 mm) to ensure that the tool will not collide with the chuck. This assumes that the limit switch has been set correctly. A distance of 40 mm from the clamped end is used for automatic machining. This allows 20 mm for gripping by the chuck and 20 mm for gripping by a robot after machining. The created files are "[ME258.LATHE.DAT]partnameA.CAD" and "[ME258.LATHE.DAT]partnameB.CAD" where A and B correspond to the first and second ends to be machined.

Note: Refer to the following subroutines:

Special purpose subroutines: CHECK and MAINPLOT.

General purpose subroutines: COMPARE, FILENAME, INTINPUT, and

REALINPUT.

4.5.2 GENERATE Option

GENERATE is mostly an organizational program in that it organizes part sections and calls the appropriate subroutines necessary for machining. The ORAC lathe is not capable of machining parabolic sections. Therefore the parabolic sections are broken into a number of linear sub-sections with edges which approximates the parabolic curve. Since the number of sub-sections determines the quality of the parabola, it is desireable to have a large number of sub-sections. However, each additional sub-section increases the number of ORAC pages and the time required to machine the part. Therefore, a compromise must be made when selecting the number of sub-sections. Typically, a sub-section will be about 1 mm in width as in the ORACAP Master Part. The contents of the arrays containing the section information must be shifted to allow inclusion of the newly generated sub-sections. Also, if the final z-position of the last section is less than the overall length of the part, a section will be added in order to machine the excess bar stock. Finally, a 3 mm section, equal to the thickness of the parting-off tool, will be added to the clamped end of the part if the Parting-off option has been selected.

The program uses numerous subroutines which, if required, generate CNC instructions for the various parts of the ORAC program. These include starting instructions, instructions for rough cuts, instructions for final cuts, instructions for threading, instructions for parting-off, instructions for tool changes, and finally instructions for ending the program. Each of the above program sections is placed in a separate output file when created, so that it can later be recognized as a distinct instruction set that begins and ends at the program datum. This separation of instructions is necessary as a result of the ORAC Lathe's limited memory. It enables easier assembly of the CNC instructions into the final ORAC program or programs. The number of CNC instructions files created is placed in the output data file "[ME258.LATHE.DAT]partname.FIL" or "[ME258.LATHE.DAT] filename.FIL" again depending on the execution mode. The data file containing the number of output files for the chess piece example is "[ME258.LATHE.DAT]QUEEN.FIL" and is contained in Appendix B. Similarly, the file for the ORACAP Master Part is "[ME258.LATHE.DAT]MASTER.FIL" and is contained in Appendix D.

The Fortran program "GENERATE.FOR" uses the ".CAD" file created by "DESIGN.FOR" to generate the CNC instructions necessary to machine a part. The program uses the file "[ME258.LATHE.DAT]partname%.CAD" (where "%" is the optional end specification) for input when in "part" mode and the default file "[ME258.LATHE.DAT]filename.CAD" when in "file" mode. Output is placed in a number of files specified by "[ME258.LATHE.CNC] partname%#.ANC" or "[ME258.LATHE.DAT]filename%#.ANC" (where "%" is the end specification and "#" is the file number), again depending on the execution mode. In both example cases there are five ".ANC" files, namely "QUEEN1.ANC" through "QUEEN5.ANC" and "MASTER1.ANC" through "MASTER5.ANC". The ".ANC" files for the chess piece example part and the ORACAP Master Part example are contained in Appendix B.6 and D.6 respectively.

The CNC instruction files produced are created as follows:

First, the program reads the material type from the ".CAD" file and then uses the MATERIAL subroutine to find the appropriate cutting parameters from the "material" file. This file contains a table of cutting, non-cutting and parting-off feedrates, and cut increments (depths) for rough cuts, final cuts, and for threading. The program first looks in the current directory for the file "MATERIAL.DAT". If it is not found it uses the "material" file in "[ME258.LATHE.DAT]". This allows each user to have one or several distinct "material" tables. A new table can be created by copying the "[ME258.LATHE.DAT] MATERIAL.DAT" file into the desired user's directory. New materials may be added to either of these files by using the VAX's editor and following the existing format. The existing "MATERIAL.DAT" file is shown in Appendix E.

Note: The data in this file has been determined by use of the ORAC Lathe. Attempts to determine the machining parameters directly from a material's machinability or horsepower requirements were made, but these attempts produced ambiguous results. It was felt that this was a result of the ORAC's small size and low HP. Therefore, a table of machining parameters was created to provide the necessary data. Additions to this table must be made based on use of the ORAC as well as previous experience with other lathes.

Second, the instructions for the beginning of the program are produced. These steps specify the units and programming format to be used, and the location of the program datum.

Incremental format was chosen, since do-loops are used, and do-loops require incremental programming. By keeping all programming incremental the format specification is only required once which saves several program steps. However this means that it will be much harder to "read" the CNC instructions produced. Since the user should never have to correct or modify these instructions, the advantages of using incremental format seemed to outweigh those of the absolute format. Millimeter units are used and the program datum location is selected based on the bar stock diameter and length.

Next, the program creates the rough cuts for the areas accessible by the right-hand tool. An example of an area inaccessible by a right-hand tool is shown in Figure 11. As much as possible of an area such as this is machined by the right-hand tool. The remainder is left for the left-hand tool.

Initially the final diameters of all sections are increased by the amount to be taken off by two finishing cuts as specified for that material in the "MATERIAL.DAT" file. The program then starts with the last section at the outside diameter of the bar stock and proceeds back towards the first section until it detects a section which has a diameter greater than the diameter being machined. If no such section is found the search is continued at the next diameter. When a protruding section is found the program creates the instructions needed to machine to the previous diameter and if a section with a smaller diameter exists on the other side of the protruding section it divides the object into two intervals which will be considered separately. If new intervals are not necessary it continues with the next

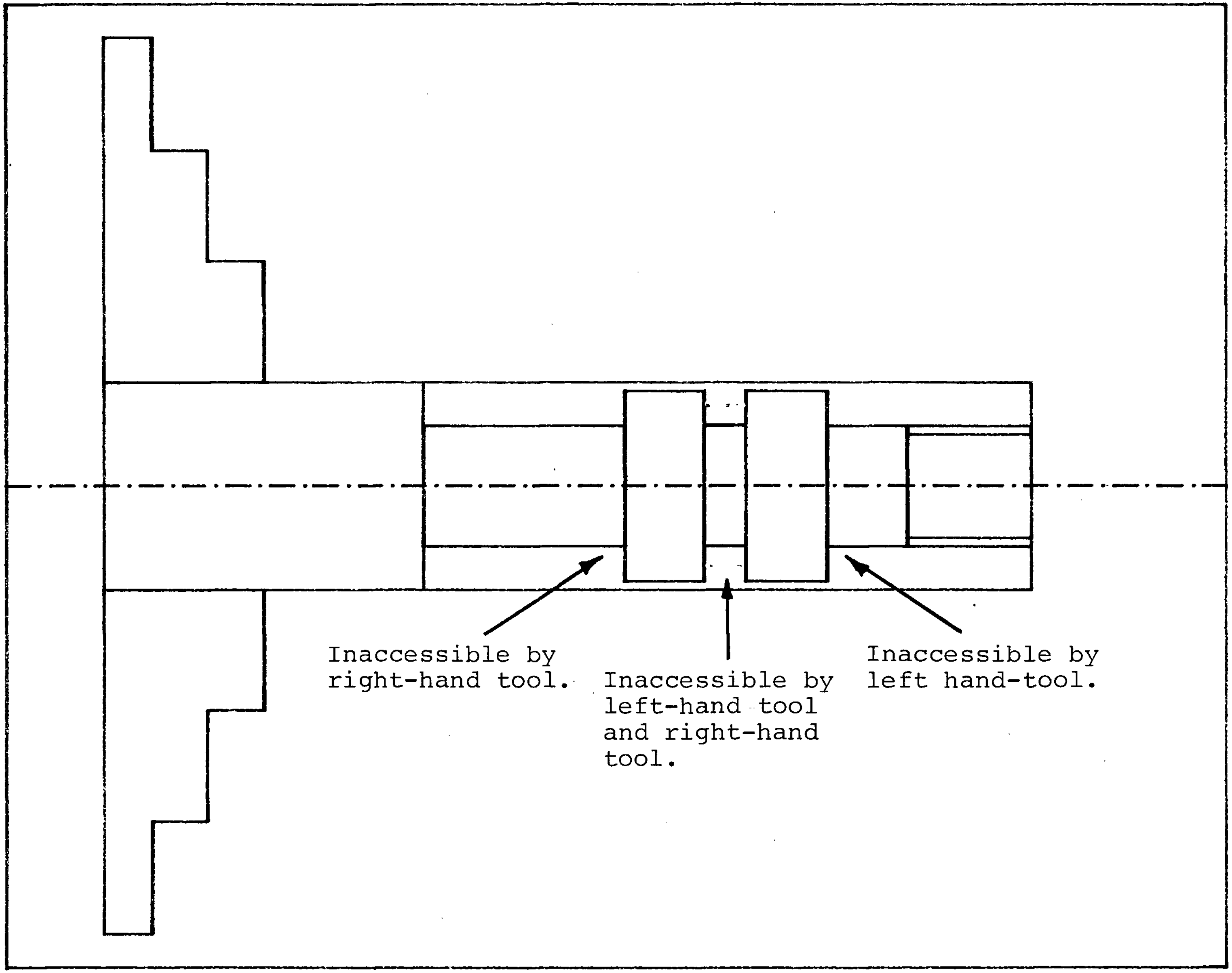


Figure 11. Example of Inaccessible Area.

diameter. If new intervals are required the program starts with the last section of the first new interval and searches as before until another protruding section is found. It again creates the instructions to machine to the previous diameter and if necessary divides the interval into two new intervals as above. This continues until all sections of each interval and all intervals have been machined to within the dimension of the modified final diameter. The two right-hand finishing cuts are then produced. This reduces most of the object to the desired final diameters with a surface finish which is dependent on the depth of the final cut increments.

During the previous step the cuts necessary for any inaccessible areas are determined and stored in an array. Now the rough left-hand cuts are produced from this information followed by the finishing cuts for these areas. The cuts are produced in this order in order to minimize the number of tool changes required.

Finally, CNC instructions are produced for threading and parting-off if required.

Note: The rough cuts may be plotted as they are generated in order to give the user visual confirmation of the tool path being produced. However, this should not be considered to be sufficient proof of the created tool path and therefore the "PROVE" and "PLOT" options should still be used for verification of the final cutting paths. The plots showing the rough cuts for the example parts are shown in Figures 12 and 13.

Note: Refer to the following subroutines:

Special purpose subroutines: BEGINPROG, ENDPROG, LEFTCUT, ORACAP Option #2 - Generate

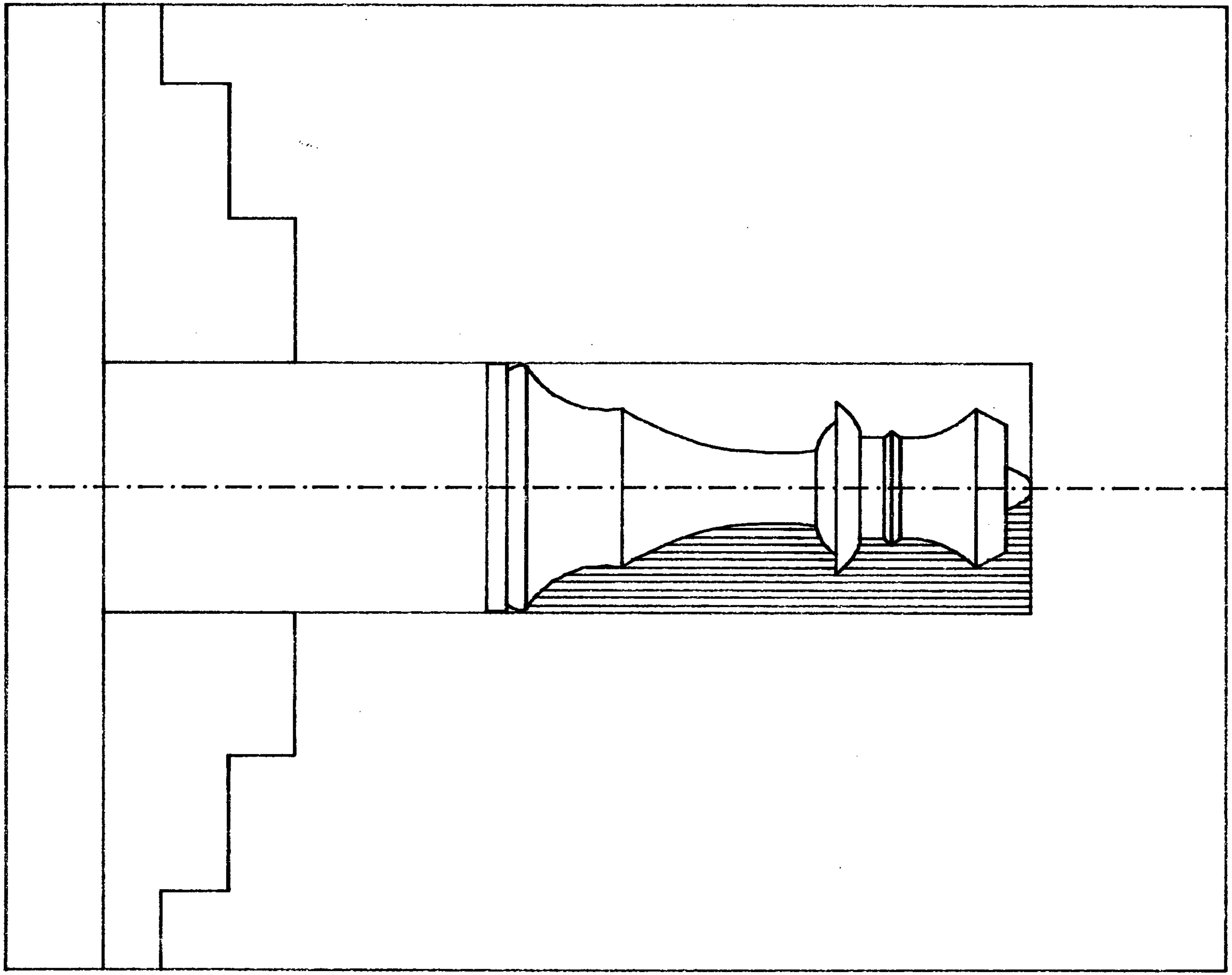


Figure 12. Rough Cuts Required for Example Chess Piece.

ORACAP Option #2 - Generate

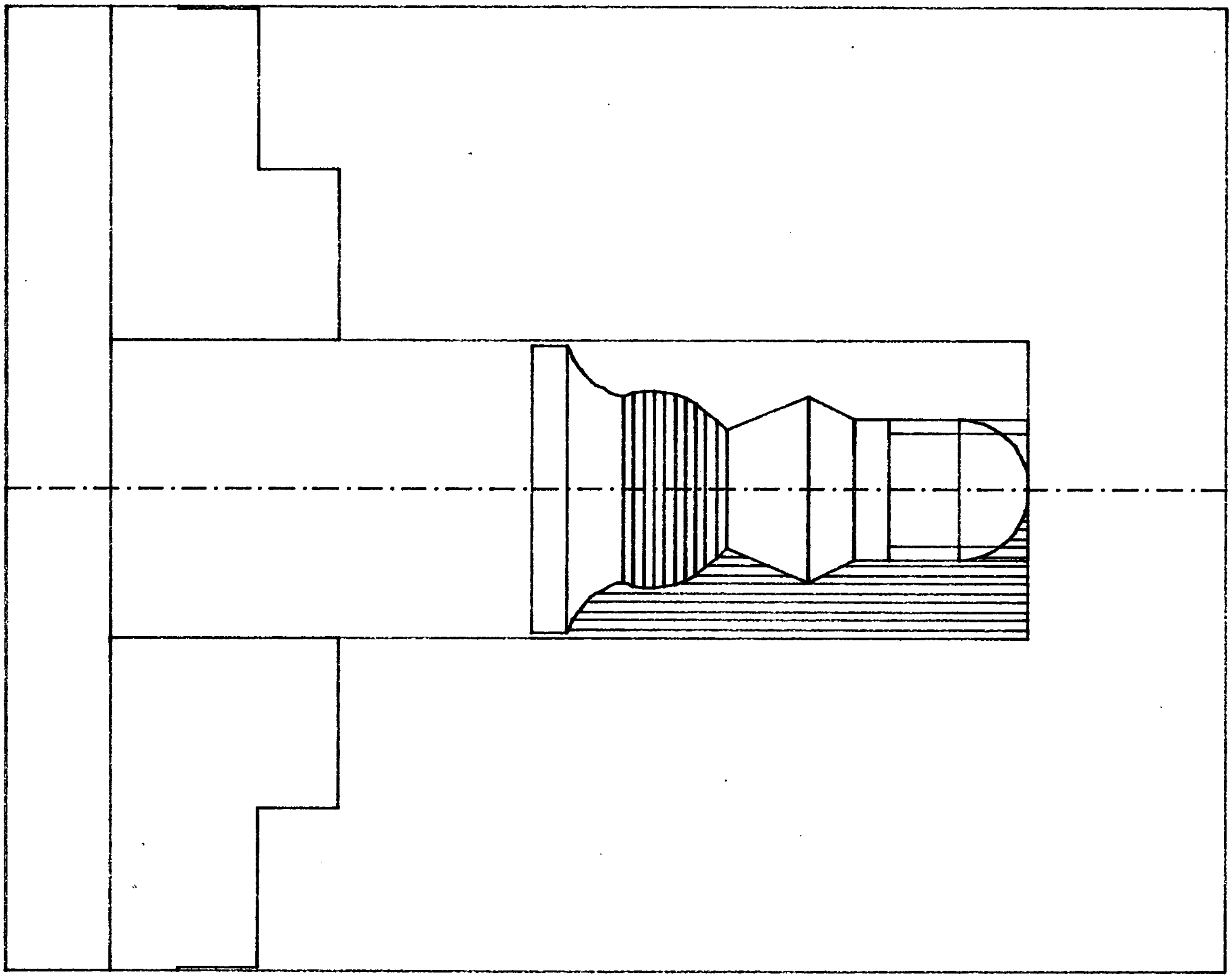


Figure 13. Rough Cuts Required for ORACAP Master Part.

LEFTFIN, MAINCUT, MAINPLOT, MATERIAL, PARTOFF, RIGHTFIN, and THREADCUT.

General purpose subroutines: COMPARE, FILENAME, and PARASOLVE.

4.5.3 OPTIMIZE Option

The "OPTIMIZE.FOR" program optimizes CNC instruction files to eliminate machining instructions which are unnecessary or inefficient. This is important since this can reduce the machining time required to produce an object as well as the number of program steps that are needed to machine the object. This is especially critical for the ORAC Lathe since programs are limited to 99 pages (steps). Thus minimizing the number of pages may also reduce the number of ORAC programs which must be assembled, proved, and down-loaded to the Lathe. In the above context, the terms optimize and minimize refer to obtaining an instruction file with fewer machining instructions, but not necessarily the optimum or minimum number of machining instructions. That is, the term optimize as used here refers to progressing towards an optimum solution, but no comparisons or tests have been made to determine if the solution obtained is truly optimal.

The program optimizes the default file "[ME258.LATHE.CNC] filename.ANC" and creates a new file "[ME258.LATHE.CNC] filename.BNC" if the program is running in "file" mode (i.e., the partname was not specified). If a partname was specified ("part" mode), it optimizes the files "[ME258.LATHE.CNC]partname%#.ANC" and creates the output files "[ME258.LATHE.CNC]partname%#.BNC" where "%" indicates the end and "#" represents the file number.

For the example chess piece, the files "QUEEN1.ANC" through "QUEEN5.ANC" are optimized to give the files "QUEEN1.BNC" through "QUEEN5.BNC". An example ".BNC" file is contained in Appendix B.7. For the ORACAP Master Part, the files "MASTER1.ANC" through "MASTER5.ANC" are optimized to give the files "MASTER1.BNC" through "MASTER5.BNC". Again, an example ".BNC" file is contained in Appendix D.7.

The number of input files is contained in the file "[ME258.LATHE.DAT]partname.FIL" or "[ME258.LATHE.DAT] filename.FIL". This will also be the number of output files.

The program optimizes instruction sequences by examining groups of up to five CNC point-to-point (PTP) instructions at a time. It first reads in instructions until four consecutive non-zero PTP instructions are found. (PTP instructions with x = 0.0 and z = 0.0 are ignored). It then checks these instructions for two equal and opposite movements along the x-axis. A positive motion followed by an equal negative motion will be eliminated. Similarly, other simple inefficiencies, and more complex ones, such as loops, can be eliminated or simplified by combining them with other instructions, before writing the new instructions to the output file. For example, a positive z-motion followed by a positive x-motion, a negative z-motion, a negative x-motion and another negative z-motion, may be simplified to three instructions. That is, a positive z-motion followed by a negative x-motion combined with a negative z-motion and followed by a negative z-motion. In other words, the second, third and fourth instructions may be replaced by one instruction.

The number of steps and storage space saved can be seen by comparing the ".ANC" and ".BNC" file data in the summary tables in Appendix A and Appendix C. For the chess piece example, the number of CNC instructions is reduced from 203 to 160. For the ORACAP Master Part, the number of instructions is reduced from 151 to 113. This shows that the "OPTIMIZE" option results in the reduction of program steps by about 25%. In these examples the final number of ORAC programs is not affected by the optimization.

Note: Refer to the following subroutines:
General purpose subroutines: COMPARE, FILENAME, and LENGTH.

4.5.4 ASSEMBLE Option

The Fortran program "ASSEMBLE.FOR" assembles CNC instruction files into ORAC program files. The program reads the CNC instructions created by the optimize or write program and assembles them into ORAC programs. When executing in "part" mode, the program determines the number of files, the number of ends, and whether automatic signal generation is required, from the appropriate data files. It then uses the default files "[ME258.LATHE.CNC]partname%#.BNC" for input. When executing in "file" mode, it prompts to determine if automatic signal generation is wanted and assumes that the number of ends and the number of files are both one. In this case "[ME258.LATHE.CNC] filename.BNC" is the default input file.

The program assembles the ORAC program by first determining the function for an instruction and then reading the appropriate parameters associated with that function. It then creates an ORAC page from this data. The ORAC page contains statements, such as x and z distances, spindle-speeds,and feedrates, as well as the control characters necessary for formatting. For example, the start of text (STX) character ^B (ASCII 2) is used to begin a page. The created page will closely resemble one created using the keypad of the lathe, except for the inclusion or exclusion of a few unimportant spaces.

If automatic signal generation has been chosen, it will place the appropriate auxiliary input and output signals in front of tool change instructions and in front of the end instruction. These signals will indicate the desired tool change, or the end of the program, to the external micro-processor which will be used to control the lathe.

The output is placed in the "[ME258.LATHE.CNC] partname%#.CNC" files or the "[ME258.LATHE.CNC]filename.CNC" file again depending on the execution mode.

Note: These files can be printed in a compressed format by using the command procedure "PRCNC.COM" which can be executed by typing "PRCNC". This executes the Fortran program "PRCNC.FOR", which compresses the entered file, and then prints and deletes the compressed file.

If the program is executing in "part" mode the program may have a number of input files and a number of output files. The CNC instructions are read from all input files consecutively until all files have been exhausted. The ORAC pages are placed in the output files in groups, as received from the input files, until the contents of an input file is too large to be placed in the current output file. If this occurs, the auxiliary output signals for ending (if desired) and an end statement page are created, the output file is closed, and a new output file is opened. The first three ORAC instructions which have been stored previously in the temporary file "[ME258.LATHE.DAT]START.TMP" are then written to the new output file. File closure and creation continues in this manner until all input files have been processed. Finally, the number of output files created is stored in the file "[ME258.LATHE.DAT]partname.OUT" or "[ME258.LATHE.DAT] filename.OUT". For the examples used, the five ".BNC" files are assembled into two ".CNC" files, namely "QUEEN1.CNC" and "QUEEN2.CNC" for the chess piece example and "MASTER1.CNC" and "MASTER2.CNC" for ORACAP Master Part. The first three pages of "QUEEN1.CNC" and "MASTER1.CNC" and the files "QUEEN.OUT" and "MASTER.OUT" are contained in Appendix B and D.

Note: Refer to the following subroutines:

General purpose subroutines: COMPARE, FILENAME, FILELEN, and LENGTH.

4.5.5 PROVE Option

Once an ORAC program has been created, it is desireable to "PROVE" the program by plotting the tool path on paper. This enables the user to detect errors without damaging tools or the workpiece.

The proving of ORAC programs was more complicated than expected due to the ORAC's use of subroutines and do-loops. Within the subroutines, the format, units, tool number, feedrate, spindle-speed, etc. may be changed, thus affecting the operation of the main ORAC program. Therefore each "page" of the "PROVE" program must be executed in the same order as the ORAC program. Two methods of achieving this were considered:

1) Creation of a Fortran program which would resemble the ORAC program but plot the tool path on a plotter instead of machining it.
2) Reading, backspacing and re-reading the appropriate instructions, as necessary, while executing the "PROVE" program.

The first option would require computer time to compile, link and run the Fortran program, while the second option would require time to execute the additional "disk" operations. The first of the two options was selected since this would allow the graphs to be reproduced by simply rerunning the created programs. Therefore, the "PROVE" Option runs the Fortran program "PROVE.FOR" which creates a Fortran plotting program which mimics the actions of the ORAC machining program.

The "PROVE" program can operate from either of the ORACAP's execution modes. "Part" mode is used to prove ORAC programs which have been designed, generated, optimized, and assembled by the corresponding options of ORACAP. "File" mode is used to prove ORAC programs which may not have been created in this manner and therefore, may not have the associated data files. For example, "file" mode would be used to prove programs written on the lathe itself, or generated by the "WRITE" and "ASSEMBLE" options.

The input files and output files (i.e. the created Fortran programs) used in "part" mode are "[ME258.LATHE.CNC] partname%#.CNC" and "[ME258.LATHE.PLT]partname%#.P$$" where "%"

is a letter indicating the end of the part, "#" is a digit indicating the file number, and "$$" is the two letter code for the terminal type. The number of input files is contained in the file "[ME258.LATHE.DAT]partname%.OUT". "File" mode uses the file "[ME258.LATHE.CNC]filename.CNC" for input and "[ME258.LATHE.PLT] filename.P$$" for output.

The "PROVE" program generates plotting instructions for either the HP Plotter, TX plotter, or Retro-Graphics Terminals in 4010 or 4027 mode, as specified in the file "TERMINAL.TMP", created by the ORACAP program. For each ORAC program page, a set of instructions is produced to show the tool motion associated with the program step. The resulting Fortran program will closely resemble the original ORAC program in both structure and program flow. The Fortran programs for the examples are "QUEEN1.PTX", "QUEEN2.PTX", "MASTER1.PTX", and "MASTER2.PTX". The first three pages of "QUEEN1.PTX" is contained in Appendix B.9 while the first three pages of "MASTER1.PTX" are contained in Appendix D.9.

Note: Refer to the following subroutines:

Special purpose subroutines: FORMAT

General purpose subroutines: COMPARE, CONVERT, and FILENAME.

4.5.6 PLOT Option

The "PLOT" option is used to compile, link, and execute the Fortran program created by the "PROVE" option. The option is not a Fortran program, but is a command procedure, contained in the file "[ME258.LATHE.COM]PLOT.COM". The program compiles and links the Fortran program "[ME258.LATHE.PLT]partname%#.P$$" when run in "part" mode and the program "[ME258.LATHE.PLT] filename.P$$" when run in "file" mode. When the ".EXE" file created is executed, the tool motion will be plotted.

Figures 14, 15, 16 and 17 contain the "proofs" of the ORAC programs for the chess piece example. Figure 14 and 15 contains the rough cuts for the right hand tool, while Figure 16 and 17 contains the finishing right-hand cuts, the rough left-hand cuts, and the finishing left-hand cuts.

In order to further illustrate the use of the "PROVE" and "PLOT" options, a copy of "MASTER1.CNC" has been modified by changing one z-value in a Point-to-Point instruction in order to create a deliberate error. When these options are executed, the plot produces a toolpath (shown in Figure 18) which indicates to the user that the part will not be machined as desired, and thus prevents damage to the tool or lathe.

The "PLOT" option also calculates an estimate of the machining time, based on the feedrates and distance travelled by the tool. This figure is displayed after plotting.

Compiling and linking of the Fortran code produced by the "PLOT" Option may require some time during heavy VAX usage. Therefore, it is important that unnecessary compile options, such as "/list", are not used since this will considerably increase the time required for these operations.

Note: Refer to the following subroutines:

Special purpose subroutines: CHANGEHP, CHANGERG, FRAMEHP, and FRAMERG.

ORACAP Option #6 - Plot

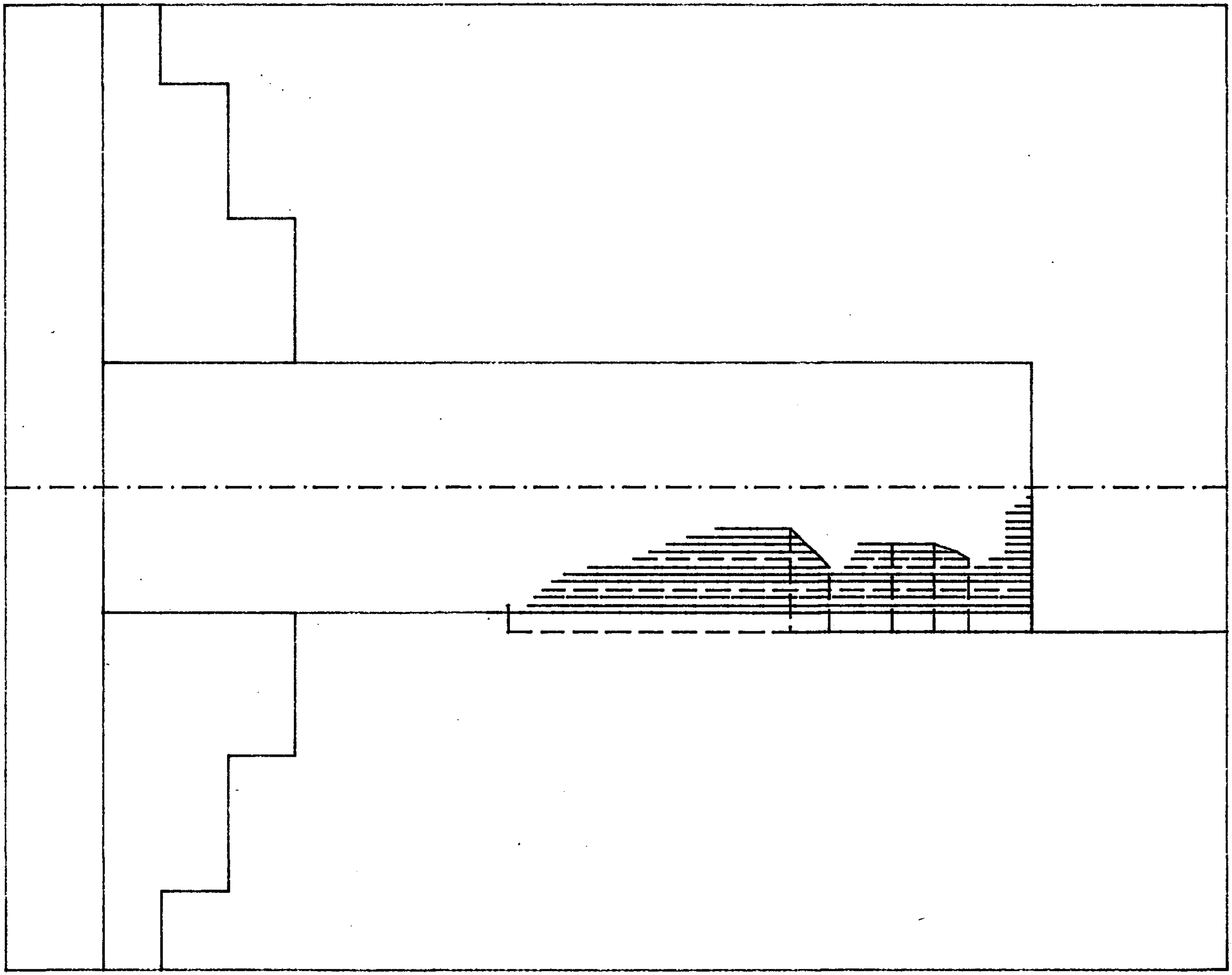


Figure 14. Proof of Rough Cuts for Example Chess Piece.

ORACAP Option #6 - Plot

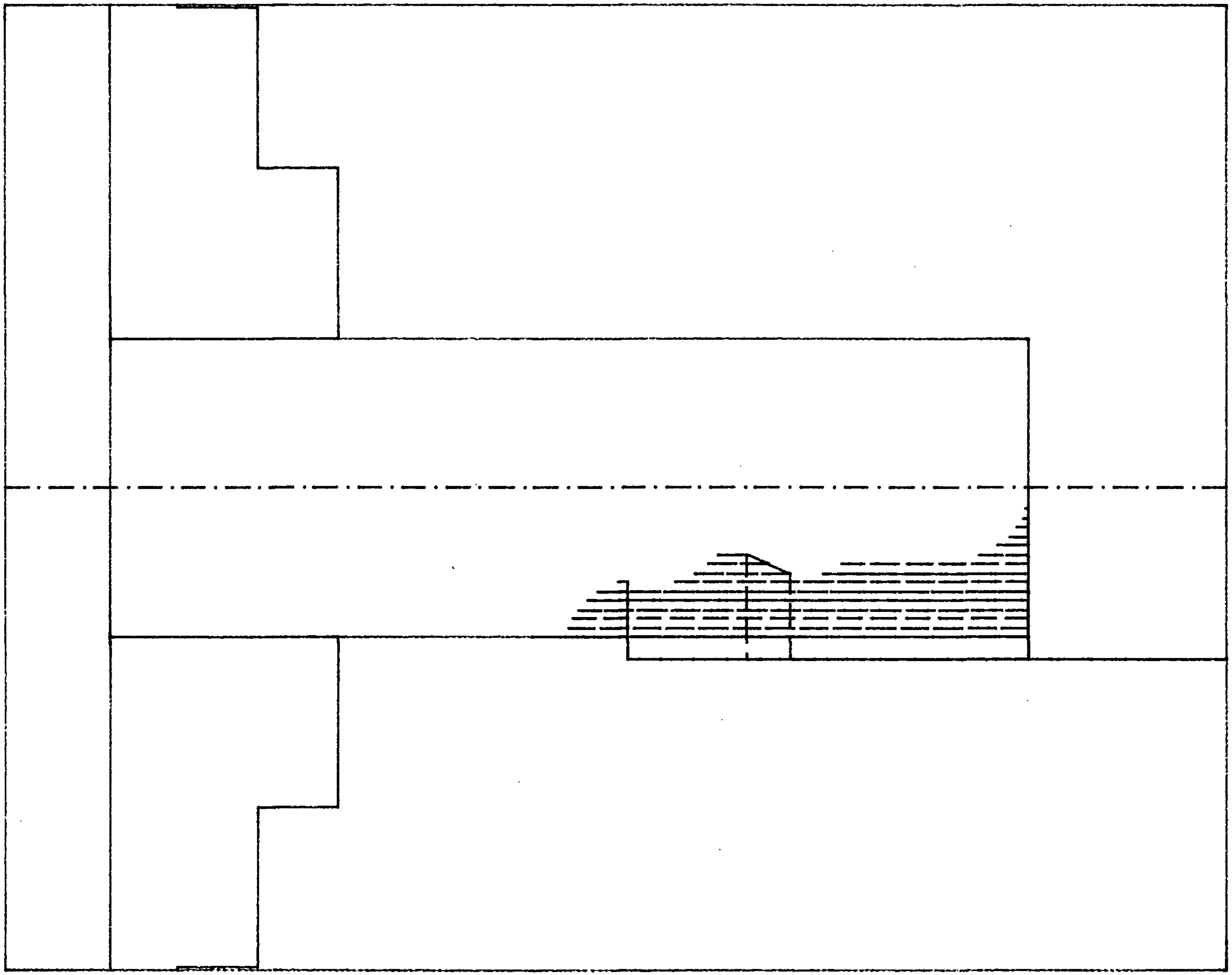


Figure 15. Proof of Rough Cuts for ORACAP Master Part.

ORACAP Option #6 - Plot

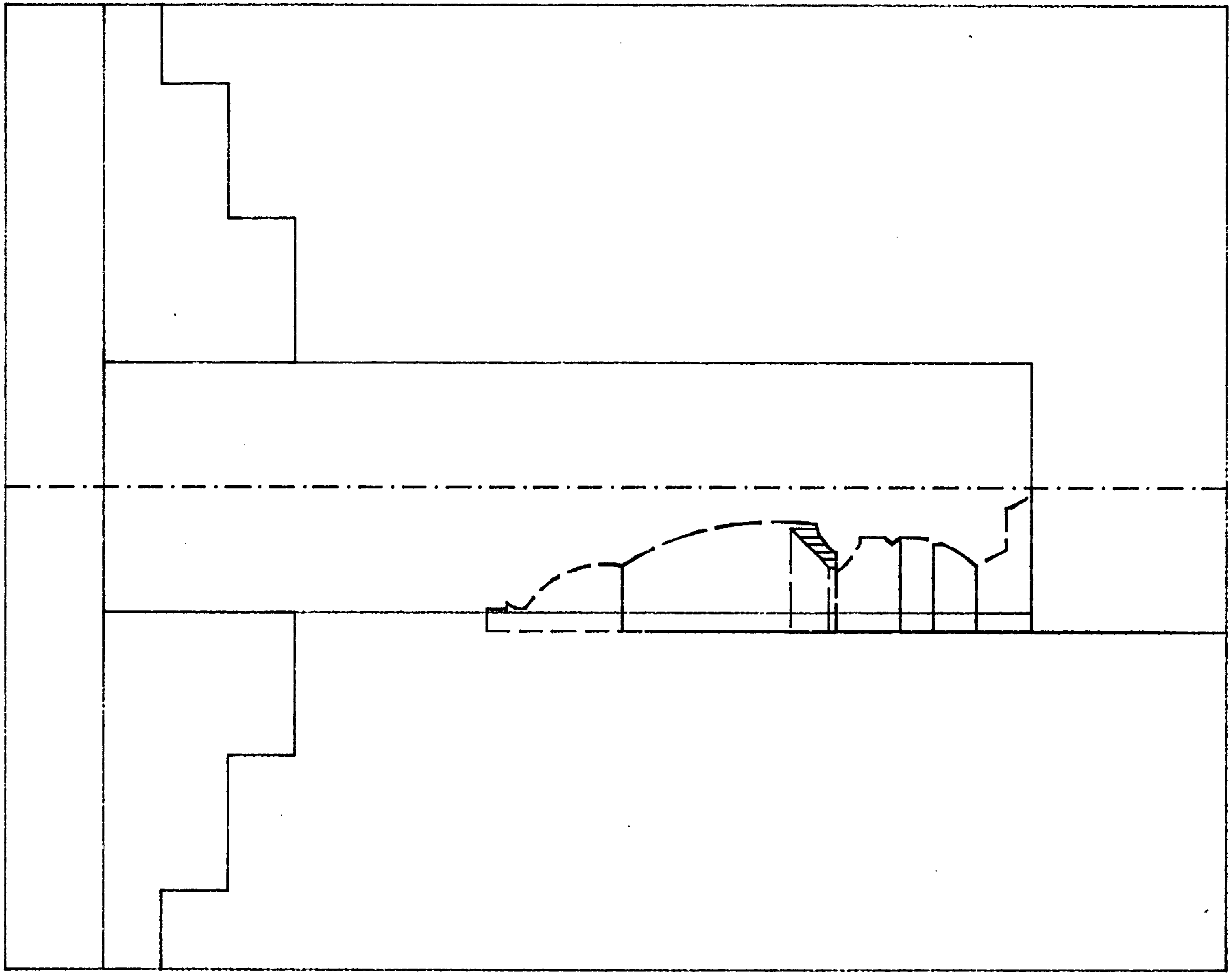


Figure 16. Proof of Remaining Cuts for Example Chess Piece.

ORACAP Option #6 - Plot

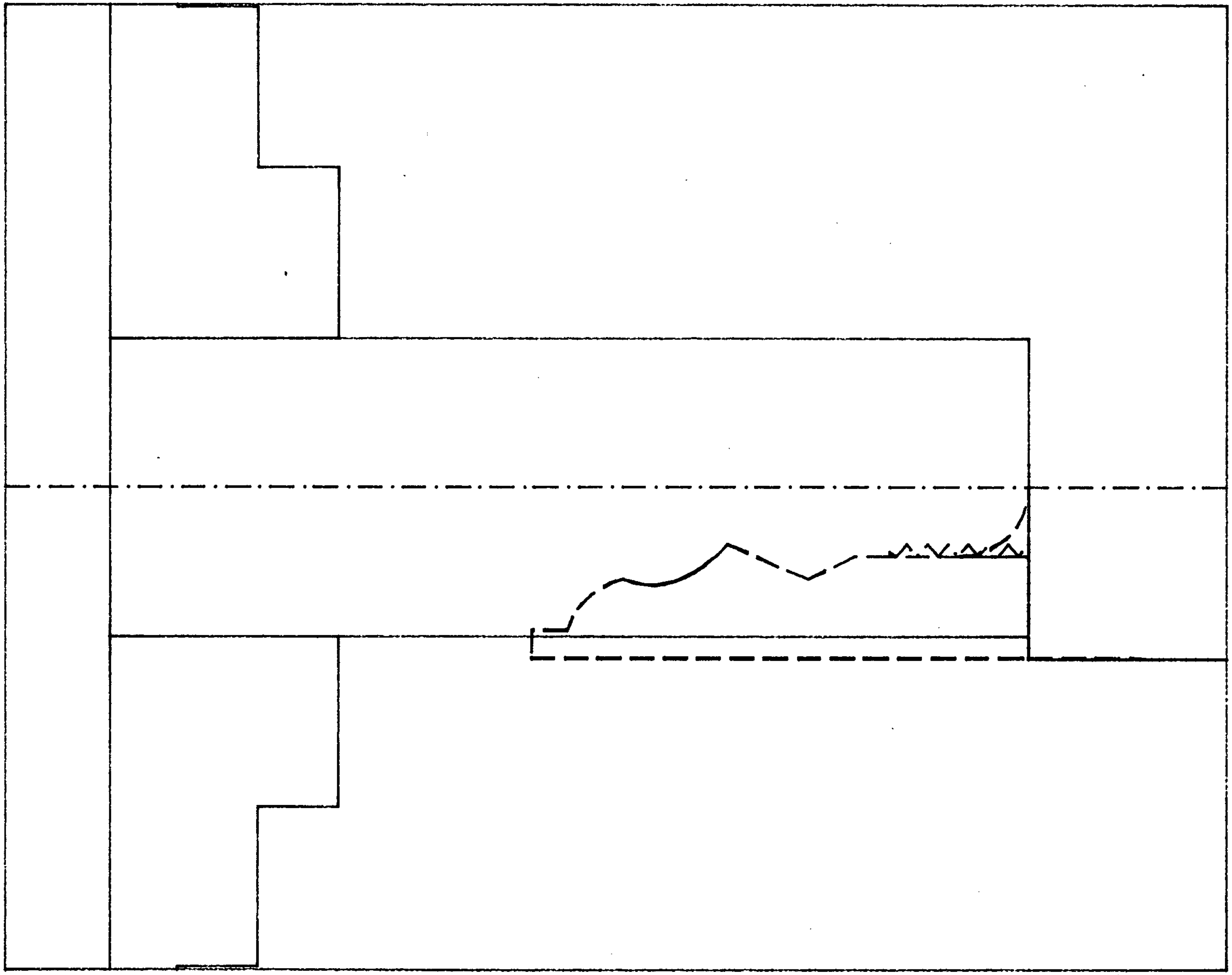


Figure 17. Proof of Remaining Cuts for ORACAP Master Part.

ORACAP Option #6 - Plot

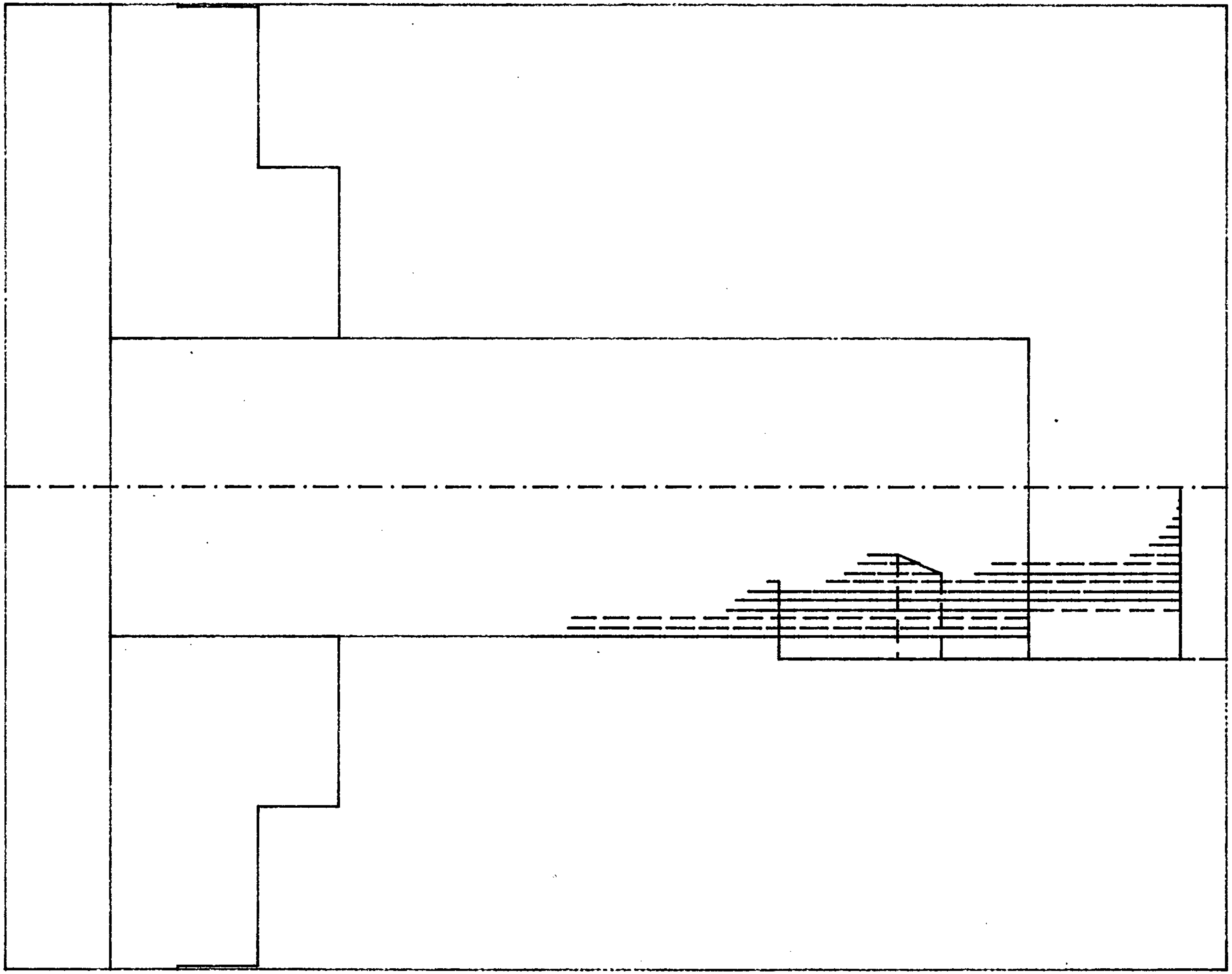


Figure 18. Proof of ORACAP Master Part with Errors.

4.5.7 VAXTOORAC Option

Once the ORAC program has been proved, it can be down-loaded, using the Fortran program "VAXTOORAC.FOR". This program transfers the contents of the input file "[ME258.LATHE.CNC]partname.CNC" or "[ME258.LATHE.CNC] filename.CNC" to the terminal line associated with the logical name "lathe". The ORAC program is passed one character at at time to the lathe, along the RS232C link, using the VAX's QIOW function. The transfer is ended by the EOT (end of transmission) control character ^D.

Note: Refer to the following subroutines:

General purpose subroutines: COMPARE, and FILENAME

4.5.8 ORACTOVAX Option

If a program exists in the ORAC Lathe's memory, it can be transferred to a computer file by use of the Fortran program "ORACTOVAX.FOR". This program uses the VAX's QIOW function to read ASCII characters placed on the terminal line assigned to the logical name "lathe". The characters are transferred one by one until the EOT (end of transmission) control character ^D is found. Once read the characters are placed in the file "[ME258.LATHE.CNC]partname.CNC" if "part" mode is being used, and in the file "[ME258.LATHE.CNC]filename.CNC" if "file" mode is being used.

Note: Refer to the following subroutines:

General purpose subroutines: COMPARE, and FILENAME

4.5.9 COMPRESS Option

Once an ORAC program has been written by "WRITE.FOR" (option 10) or up-loaded by "ORACTOVAX.FOR" (option 8), it may be desirable to have the ORAC program compressed into CNC instructions. This can be accomplished by the Fortran program "COMPRESS.FOR". The "COMPRESS" Option uses the input file "[ME258.LATHE.CNC]partname.CNC" and the output file "[ME258.LATHE.CNC]partname.BNC" if in "part" mode and "[ME258.LATHE.CNC]filename.CNC" and "[ME258.LATHE.CNC] filename.BNC" if in "file" mode.

Note: Refer to the following subroutines:

Special purpose subroutines: STRING and FORMAT

General purpose subroutines: COMPARE, CONVERT, and FILENAME

4.5.10 WRITE Option

In order to enable easier input of ORAC programs, written by a user without the use of ORACAP, "WRITE.FOR" was written. This program creates the ORAC program pages in a way which is similar to program entry using the keypad of the lathe, and stores the instructions in the file "[ME258.LATHE.CNC] partname.CNC" or "[ME258.LATHE.CNC]filename.CNC" again depending on the execution mode.

The program also uses input subroutines which display the previous value of a variable, and allows that value to be re-entered by pressing the return key.

Note: Refer to the following subroutines:

General purpose subroutines: BYTEINPUT, COMPARE, FILENAME, INTINPUT, and REALINPUT.

4.6 Description of ORACAP Sub-Programs

The Fortran code for the subroutines called by the ORACAP modules are contained in the directory "[ME258.LATHE.SUB]". Listings of the Fortran code can be found in Appendix F on the enclosed microfiche. The compiled object files are contained in the library file "[ME258.LIBRARY]LATHELIB.OLB". A short descriptions of these subroutines follow:

ADE2NB* : Used to convert an ascii string (i.e. a byte variable) to a real number.

ARCMID : Used to determine the "midpoint" of an arc. That is, a point on the arc which is equidistant from the endpoints.

BEGINPROG : Produces the starting instructions of an ORAC program.

BYTEINPUT : Used to display a prompt and the previous value of a byte variable, so that the value may be re-entered by pressing the return key.

CENTERCC : Uses the endpoints and radius of an arc to determine the center of the arc, and the angles which define the start and end of a concave arc.

CENTERCV : Uses the endpoints and radius of an arc to determine the center of the arc, and the angles which define the start and end of a convex arc.

CHANGEHP : Changes plotting parameters (eg. pen number) of the HP graphics language in order to indicate different feedrates and tools.

CHANGERG : Changes plotting parameters (eg. line type) of the IGL graphics language in order to indicate different feedrates and tools.

CHECK : Checks sections of a designed object for inaccessible areas (i.e. areas which can not be machined with the present ORAC tools).

COMPARE : Compares two strings and returns a logical value indicating if they are equal.

CONVERT : Determines the integer value of a byte variable.

ENDPROG : Produces the ending instructions of an ORAC program.

FILELEN : Determines the length of a specified file.

FILENAME : Constructs a complete file specification from supplied and default information.

FORMAT : Removes and adds blanks of an ORAC program in order to produce an ORAC program in a standard format.

FRAMEHP : Used to set up plotting on the HP 4221 plotter. Initializes plotting, draws a frame, the chuck of the lathe, and the workpiece.

FRAMERG : Used to set up plotting on Tektronix terminals using IGL. Initializes plotting, draws a frame, the chuck of the lathe, and the workpiece.

INTARCCC : Finds the intersection of the cutting path and a concave circular section.

INTARCCV : Finds the intersection of the cutting path and a convex circular section.

INTINPUT : Used to display a prompt and the previous value of an integer variable, so that the value may be re-entered by pressing the return key.

INTLINE : Finds the intersection of the cutting path and a linear section.

LEFTCUT : Produces instructions, for rough cuts, for areas accessible by a left-hand cutting tool.

LEFTFIN : Produces instructions, for final cuts, for areas accessible by a left-hand cutting tool.

LEFTPLOT : Produces plotting instructions to show tool path being created by a left-hand cutting tool.

LENGTH* : Determines the length of a byte variable.

MAINCUT : Determines the cutting path for the main rough cuts necessary to outline the object.

MAINPLOT : Plots the geometric shape of the object during program execution.

MATERIAL : Determines feedrates and cutting depths associated with a particular material from a user supplied table.

PARASOLVE* : Determines the equation of a parabola from a set of three points.

PARMID : Determines the "midpoint" of a parabola. This is a point on the parabola which has the x value half-way between the x values of the endpoints, or the z value half-way between the z values of the endpoints.

PARTOFF : Produces instructions for parting off an object after machining.

REALINPUT : Used to display a prompt and the previous value of a real variable, so that the value may be re-entered by pressing the return key.

RIGHTCUT : Produces instructions, for rough cuts, for areas accessible by a right-hand cutting tool.

RIGHTFIN : Produces instructions, for final cuts, for areas accessible by a right-hand cutting tool.

RIGHTPLOT : Produces plotting instructions to show tool path being created by a right-hand cutting tool.

SPINSPEED : Determines spindle-speed for cutting based on the diameter being cut.

THRDSPEED : Determines spindle-speed for cutting threads based on the pitch of the thread.

THREADCUT : Produces instructions for cutting threads.

TOOLSIG : Produces instructions for auxiliary input and output required for tool changer control and end of program control.

Note: Subroutines marked with an asterisk (*) were not written by the author of this thesis.

CHAPTER 5

Conclusion

5.1 Contributions of ORACAP

In addition to providing the users of the ORAC Lathe with a package which allows them to design, generate, optimize, assemble, prove, down-load, up-load, compress and write programs for the ORAC Lathe, ORACAP presents a noval method of defining an object and producing the required machining instructions from that information. It also demonstrates a method of writing multiple programs for a CNC machine tool which does not have sufficient memory for all machining to be performed by one program.

As a result of the modular nature of ORACAP, parts of it may be used in conjunction with other CAD or CAD/CAM packages provided the necessary software for such interaction is written. For example, it may be desirable to use a more versatile CAD package. The information produced by such a package could be used by the GENERATE option of ORACAP, provided the required information is prepared in the same format as produced by the DESIGN option. Or, if it is desirable to use only the ASSEMBLE option to assemble the ORAC programs from instructions produced by another CAD/CAM package, the CNC instruction file would have to be in a form usable by the ASSEMBLE program. Finally it may be desirable to use ORACAP to write programs for another lathe. In that case, the ASSEMBLE option could be replaced or modified to produce programs in the form required by that lathe.

5.2 Comparison of Various Manufacturing Methods

The ORACAP program will allow design and manufacture of small externally machinable objects in much less time than possible using the existing methods available in the Mechanical Engineering Department. For instance, the chess piece example used throughout this paper, could be manufactured in several ways:

1) By an experienced machinist on a manual lathe.
2) By use of the unmodified ORAC CNC Lathe by an experienced operator.
3) By use of the ORAC CNC Lathe with modifications and using the ORACAP software.
4) By use of the ORAC CNC Lathe with modifications, using the ORACAP software, under external microprocessor control.

The following tables list necessary operations and estimates of time requirements for each of the above cases. Table I is based on a batch size of one, while Table II is based on a batch size of 25. The time estimates for manual machining are based on the time required to actually machine one chess piece. The time required for machining 25 pieces is based on the time required to machine a form tool which could be repeatedly used to machine the parts. The time requirements for cases 2, 3, and 4 are based on prior use of the lathe.

Note: The tables do not include the time required for parting-off of the final object, since the ORAC is not capable of this function.

Table I. Production Times for Batch Size of 1

	Case 1	Case 2	Case 3	Case 4
Programming time	-	2 hrs	0.2 hrs	0.2 hrs
Debugging time:	-	2 hrs	0.1 hrs	0.1 hrs
Setup time:	-	0.2 hrs	0.2 hrs	0.1 hrs
Machining time:	4 hrs	0.2 hrs	0.2 hrs	0.2 hrs
Total:	4 hrs	4.4 hrs	0.7 hrs	0.6 hrs

Table II. Production Times for Batch Size of 25

	Case 1	Case 2	Case 3	Case 4
Programming time:	-	0.08 hrs	0.008 hrs	0.008 hrs
Debugging time:	-	0.08 hrs	0.004 hrs	0.004 hrs
Setup time:	0.08 hrs	0.2 hrs	0.2 hrs	0.004 hrs
Machining time:	0.5 hrs	0.2 hrs	0.2 hrs	0.2 hrs
Total per part:	0.58 hrs	0.56 hrs	0.412 hrs	0.216 hrs

These tables show that with increased use of Computer Aided Programming and Computer Aided Production, it is possible to achieve substantial savings in time and expenditure.

5.3 Uses of ORACAP Package

The ORACAP Package is beneficial in several ways. First, it may be used for design and machining with substantial savings in time and expenditure as demonstrated above. Second, various options may be used for educational laboratories as follows:

1) The "WRITE" option may be used to input CNC instructions which have been manually written by following the instructions in the ORAC Programming Instruction and Maintenance Manual [4]. This allows several users to enter CNC instructions into the computer at the same time.
2) The "ASSEMBLE" option may be used to assemble the CNC instructions produced in (1) into the ORAC program format used by the lathe.
3) The "PROVE" and "PLOT" options may be used to prove the program on a graphics terminal or plotter prior to execution on the lathe.
4) The "VAXTOORAC" option may be used to transfer a program from the VAX computer to the ORAC lathe.
5) The "ORACTOVAX" option may be used to transfer a program from the ORAC lathe to the VAX computer.

Thus the ORACAP package provides a valuable tool for the introduction of Computer Aided Design / Computer Aided Manufacturing to students in a laboratory environment.

REFERENCES

[1] Pusztai, Joseph and Michael Sava, Computer Numerical Control, Reston Publishing Company, Inc., Reston, Virginia, 1983.

[2] Groover, Mikell P., Automation, Production Systems, and Computer-Aided Manufacturing, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1980.

[3] Koren, Yoram, Computer Control of Manufacturing Systems, McGraw-Hill Book Company, New York, 1983.

[4] ORAC Programming Instruction and Maintenance Manual, Denford Machine Tools Limited, Brighouse, West Yorkshire, 1983.

BIBLIOGRAPHY

El-Midany, T.T., H. Eskicioglu, and B.J. Davies, "Interactive Operation Sequence Planning for Turned Parts (AUTOCAP) and Non-rotational Parts (ICAPP)," Seminar on Computer Aided Design in Industry, Belgrade, Yugoslavia, Sept. 1980.

Groover, Mikell P., Automation, Production Systems, and Computer Aided Manufacturing, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1980.

Hannam, R.G. and J.C.S. Plummer, "Capturing Production Engineering Practice within a CADCAM System, "International Journal of Production Research, 1984, vol.22, no. 2, pp. 267-280.

Koren, Yoram, Computer Control of Manufacturing Systems, McGraw-Hill Book Company, New York, 1983.

Pusztai, Joseph and Michael Sava, Computer Numerical Control, Reston Publishing Company, Inc., Reston, Virginia, 1983.

Roberts, Arthur D. and Richard C. Prentice, Programming for Numerical Control Machines, 2nd ed., McGraw-Hill, Inc., New York, 1978.

Steudel, Harold J., "Computer-Aided Process Planning: Past, Present and Future," International Journal of Production Research, 1984, vol. 22, no. 2, pp. 253-256.

APPENDIX A

ORACAP Chess Piece Example Table

Table III. File Sizes for Chess Piece Example

Option	Filename	ORAC Pages	Size (blocks)
Design	QUEEN.CAD		2
	QUEEN.END		1
	QUEEN.FIL		1
	QUEEN.OUT		1
	QUEEN.SIZ		1
total	5		6
Generate	QUEEN1.ANC	124	10
	QUEEN2.ANC	25	3
	QUEEN3.ANC	0	0
	QUEEN4.ANC	36	3
	QUEEN5.ANC	18	2
total	5	203	18
Optimize	QUEEN1.BNC	91	8
	QUEEN2.BNC	25	3
	QUEEN3.BNC	0	0
	QUEEN4.BNC	26	3
	QUEEN5.BNC	18	2
total	5	160	16
Assemble	QUEEN1.CNC	92	23
	QUEEN2.CNC	72	18
total	2	164	41
Prove	QUEEN1.PTX	92	101
	QUEEN2.PTX	72	97
total	2	164	198

APPENDIX B

ORACAP Chess Piece Example Files

B.1 [ME258.LATHE.DAT]QUEEN.CAD

```
aluminum 12  PART= F  AUTO= F  INCH= F       ! matl,nsect,flags
  1  1                                       ! sect,type
   0.000  25.400   2.000  25.400             ! p1,d1,p2,d2
  2  3                                       ! sect,type
   2.000  24.000   4.000  25.000   2.000     ! p1,d1,p2,d2,rad
  3  2                                       ! sect,type
   4.000  25.000  14.000  16.200  10.000     ! p1,d1,p2,d2,rad
  4  2                                       ! sect,type
  14.000  16.200  34.000   7.500  30.000     ! p1,d1,p2,d2,rad
  5  3                                       ! sect,type
  34.000   7.500  36.000  13.500   5.000     ! p1,d1,p2,d2,rad
  6  3                                       ! sect,type
  36.000  17.500  38.600  10.200   5.000     ! p1,d1,p2,d2,rad
  7  1                                       ! sect,type
  38.600  10.200  41.000  10.200             ! p1,d1,p2,d2
  8  3                                       ! sect,type
  41.000  10.200  41.800  11.500   2.000     ! p1,d1,p2,d2,rad
  9  3                                       ! sect,type
  41.800  11.500  42.700  10.200   2.000     ! p1,d1,p2,d2,rad
 10  2                                       ! sect,type
  42.700  10.200  50.400  16.200  10.000     ! p1,d1,p2,d2,rad
 11  1                                       ! sect,type
  50.400  16.200  53.500  12.900             ! p1,d1,p2,d2
 12  3                                       ! sect,type
  53.500   4.000  56.000   1.000   4.000     ! p1,d1,p2,d2,rad
```

```
Where:  matl is the material
        nsect is the number of sections
        flags are flags indicating options:
                PART= T indicates parting-off option
                AUTO= T indicates automatic machining
                INCH= T indicates inch units
        sect is the section number
        type is the section type
        p1 is position one
        d1 is the diameter at position one
        p2 is position two
        d2 is the diameter at position two
        rad is the radius
```

B.2 [ME258.LATHE.DAT]QUEEN.END

```
1
```

B.3 [ME258.LATHE.DAT]QUEEN.FIL

```
5
```

B.4 [ME258.LATHE.DAT]QUEEN.OUT

```
2
```

B.5 [ME258.LATHE.DAT]QUEEN.SIZ

```
  26.200  56.000
```

B.6 [ME258.LATHE.CNC]QUEEN1.ANC

```
INC
MM
PDAT  15.100  20.000
PTP    0.000 -20.000 1000 1 1094
PTP    0.000   0.000 1000 1 1214
PTP   -2.000   0.000   80 1 1214
PTP    0.000 -56.000   80 1 1214
PTP    0.000  56.000 1000 1 1214
PTP    2.000   0.000 1000 1 1214
PTP    0.000   0.000 1000 1 1262
PTP   -2.800   0.000   80 1 1262
PTP    0.000 -51.754   80 1 1262
PTP    0.000  51.754 1000 1 1262
PTP    2.800   0.000 1000 1 1262
PTP    0.000   0.000 1000 1 1310
PTP   -3.600   0.000   80 1 1310
PTP    0.000 -51.116   80 1 1310
PTP    0.000  51.116 1000 1 1310
PTP    3.600   0.000 1000 1 1310
PTP    0.000   0.000 1000 1 1358
PTP   -4.400   0.000   80 1 1358
PTP    0.000 -50.323   80 1 1358
PTP    0.000  50.323 1000 1 1358
PTP    4.400   0.000 1000 1 1358
PTP    0.000   0.000 1000 1 1406
PTP   -5.200   0.000   80 1 1406
PTP    0.000 -49.312   80 1 1406
PTP    0.000  49.312 1000 1 1406
PTP    5.200   0.000 1000 1 1406
PTP    0.000   0.000 1000 1 1454
PTP   -6.000   0.000   80 1 1454
PTP    0.000 -47.930   80 1 1454
PTP    0.000  47.930 1000 1 1454
PTP    6.000   0.000 1000 1 1454
PTP    0.000 -53.845 1000 1 1262
PTP   -2.800   0.000   80 1 1262
PTP    0.000  -0.155   80 1 1262
PTP    0.000   0.155 1000 1 1262
PTP    2.800   0.000 1000 1 1262
PTP    0.000  53.845 1000 1 1502
PTP   -6.800   0.000   80 1 1502
PTP    0.000 -19.101   80 1 1502
PTP    0.000  19.101 1000 1 1502
PTP    6.800   0.000 1000 1 1502
PTP    0.000 -20.610 1000 1 1502
PTP   -6.800   0.000   80 1 1502
PTP    0.000 -24.870   80 1 1502
PTP    0.000  24.870 1000 1 1502
PTP    6.800   0.000 1000 1 1502
PTP    0.000  -0.800 1000 1 1550
PTP   -7.600   0.000   80 1 1550
PTP    0.000 -19.315   80 1 1550
PTP    0.000  19.315 1000 1 1550
```

```
 54  PTP     7.600    0.000 1000 1 1550
 55  PTP     0.000   -0.800 1000 1 1598
 56  PTP    -8.400    0.000   80 1 1598
 57  PTP     0.000  -16.994   80 1 1598
 58  PTP     0.000   16.994 1000 1 1598
 59  PTP     8.400    0.000 1000 1 1598
 60  PTP     0.000   -0.800 1000 1 1646
 61  PTP    -9.200    0.000   80 1 1646
 62  PTP     0.000  -14.417   80 1 1646
 63  PTP     0.000   14.417 1000 1 1646
 64  PTP     9.200    0.000 1000 1 1646
 65  PTP     0.000   -0.800 1000 1 1694
 66  PTP   -10.000    0.000   80 1 1694
 67  PTP     0.000  -11.438   80 1 1694
 68  PTP     0.000   11.438 1000 1 1694
 69  PTP    10.000    0.000 1000 1 1694
 70  PTP     0.000   -0.800 1000 1 1742
 71  PTP   -10.800    0.000   80 1 1742
 72  PTP     0.000   -7.649   80 1 1742
 73  PTP     0.000    7.649 1000 1 1742
 74  PTP    10.800    0.000 1000 1 1742
 75  PTP     0.000   24.610 1000 1 1550
 76  PTP    -7.600    0.000   80 1 1550
 77  PTP     0.000   -4.172   80 1 1550
 78  PTP     0.000    4.172 1000 1 1550
 79  PTP     7.600    0.000 1000 1 1550
 80  PTP     0.000    0.000 1000 1 1598
 81  PTP    -8.400    0.000   80 1 1598
 82  PTP     0.000   -2.669   80 1 1598
 83  PTP     0.000    2.669 1000 1 1598
 84  PTP     8.400    0.000 1000 1 1598
 85  PTP     0.000    0.000 1000 1 1646
 86  PTP    -8.400    0.000 1000 1 1646
 87  DO     5
 88  PTP    -0.800    0.000   80 1 1646
 89  PTP     0.000   -2.500   80 1 1646
 90  PTP     0.000    2.500 1000 1 1646
 91  EDO
 92  PTP    12.400    0.000 1000 1 1646
 93  PTP     0.000    0.000 1000 1 1886
 94  PTP   -13.200    0.000   80 1 1886
 95  PTP     0.000   -1.608   80 1 1886
 96  PTP     0.000    1.608 1000 1 1886
 97  PTP    13.200    0.000 1000 1 1886
 98  PTP     0.000    0.000 1000 1 1934
 99  PTP   -14.000    0.000   80 1 1934
100  PTP     0.000   -0.395   80 1 1934
101  PTP     0.000    0.395 1000 1 1934
102  PTP    14.000    0.000 1000 1 1934
103  PTP     0.000   -6.430 1000 1 1550
104  PTP    -7.600    0.000   80 1 1550
105  PTP     0.000  -11.916   80 1 1550
106  PTP     0.000   11.916 1000 1 1550
107  PTP     7.600    0.000 1000 1 1550
108  PTP     0.000   -1.131 1000 1 1598
```

109	PTP	-8.400	0.000	80	1	1598
110	PTP	0.000	-10.293	80	1	1598
111	PTP	0.000	10.293	1000	1	1598
112	PTP	8.400	0.000	1000	1	1598
113	PTP	0.000	-2.435	1000	1	1646
114	PTP	-9.200	0.000	80	1	1646
115	PTP	0.000	-4.176	80	1	1646
116	PTP	0.000	4.176	1000	1	1646
117	PTP	9.200	0.000	1000	1	1646
118	PTP	0.000	-4.225	1000	1	1646
119	PTP	-9.200	0.000	80	1	1646
120	PTP	0.000	-3.321	80	1	1646
121	PTP	0.000	3.321	1000	1	1646
122	PTP	9.200	0.000	1000	1	1646
123	PTP	0.000	0.000	1000	1	1646
124	PTP	0.000	34.222	1000	1	1646

B.7 [ME258.LATHE.CNC]QUEEN1.BNC

```
INC
MM
PDAT  15.100   20.000
PTP    0.000  -20.000  1000  1  1094
PTP   -2.000    0.000    80  1  1214
PTP    0.000  -56.000    80  1  1214
PTP    0.000   56.000  1000  1  1214
PTP   -0.800    0.000    80  1  1262
PTP    0.000  -51.754    80  1  1262
PTP    0.000   51.754  1000  1  1262
PTP   -0.800    0.000    80  1  1310
PTP    0.000  -51.116    80  1  1310
PTP    0.000   51.116  1000  1  1310
PTP   -0.800    0.000    80  1  1358
PTP    0.000  -50.323    80  1  1358
PTP    0.000   50.323  1000  1  1358
PTP   -0.800    0.000    80  1  1406
PTP    0.000  -49.312    80  1  1406
PTP    0.000   49.312  1000  1  1406
PTP   -0.800    0.000    80  1  1454
PTP    0.000  -47.930    80  1  1454
PTP    0.000   47.930  1000  1  1454
PTP    6.000    0.000  1000  1  1454
PTP    0.000  -53.845  1000  1  1262
PTP   -2.800    0.000    80  1  1262
PTP    0.000   -0.155    80  1  1262
PTP    0.000    0.155  1000  1  1262
PTP    2.800    0.000  1000  1  1262
PTP    0.000   53.845  1000  1  1502
PTP   -6.800    0.000    80  1  1502
PTP    0.000  -19.101    80  1  1502
PTP    0.000   19.101  1000  1  1502
PTP    6.800    0.000  1000  1  1502
PTP    0.000  -20.610  1000  1  1502
PTP   -6.800    0.000    80  1  1502
PTP    0.000  -24.870    80  1  1502
PTP    0.000   24.870  1000  1  1502
PTP   -0.800   -0.800    80  1  1550
PTP    0.000  -19.315    80  1  1550
PTP    0.000   19.315  1000  1  1550
PTP   -0.800   -0.800    80  1  1598
PTP    0.000  -16.994    80  1  1598
PTP    0.000   16.994  1000  1  1598
PTP   -0.800   -0.800    80  1  1646
PTP    0.000  -14.417    80  1  1646
PTP    0.000   14.417  1000  1  1646
PTP   -0.800   -0.800    80  1  1694
PTP    0.000  -11.438    80  1  1694
PTP    0.000   11.438  1000  1  1694
PTP   -0.800   -0.800    80  1  1742
PTP    0.000   -7.649    80  1  1742
PTP    0.000    7.649  1000  1  1742
PTP   10.800    0.000  1000  1  1742
```

```
54  PTP    0.000   24.610 1000 1 1550
55  PTP   -7.600    0.000   80 1 1550
56  PTP    0.000   -4.172   80 1 1550
57  PTP    0.000    4.172 1000 1 1550
58  PTP   -0.800    0.000   80 1 1598
59  PTP    0.000   -2.669   80 1 1598
60  PTP    0.000    2.669 1000 1 1598
61  PTP    8.400    0.000 1000 1 1598
62  PTP   -8.400    0.000 1000 1 1646
63  DO    5
64  PTP   -0.800    0.000   80 1 1646
65  PTP    0.000   -2.500   80 1 1646
66  PTP    0.000    2.500 1000 1 1646
67  EDO
68  PTP   -0.800    0.000   80 1 1886
69  PTP    0.000   -1.608   80 1 1886
70  PTP    0.000    1.608 1000 1 1886
71  PTP   -0.800    0.000   80 1 1934
72  PTP    0.000   -0.395   80 1 1934
73  PTP    0.000    0.395 1000 1 1934
74  PTP   14.000    0.000 1000 1 1934
75  PTP    0.000   -6.430 1000 1 1550
76  PTP   -7.600    0.000   80 1 1550
77  PTP    0.000  -11.916   80 1 1550
78  PTP    0.000   11.916 1000 1 1550
79  PTP   -0.800   -1.131   80 1 1598
80  PTP    0.000  -10.293   80 1 1598
81  PTP    0.000   10.293 1000 1 1598
82  PTP   -0.800   -2.435   80 1 1646
83  PTP    0.000   -4.176   80 1 1646
84  PTP    0.000    4.176 1000 1 1646
85  PTP    9.200    0.000 1000 1 1646
86  PTP    0.000   -4.225 1000 1 1646
87  PTP   -9.200    0.000   80 1 1646
88  PTP    0.000   -3.321   80 1 1646
89  PTP    0.000    3.321 1000 1 1646
90  PTP    9.200    0.000 1000 1 1646
91  PTP    0.000   34.222 1000 1 1646
```

B.8 [ME258.LATHE.CNC]QUEEN1.CNC

```
PAGE 01 INCREMENTAL-FORMAT.G91
PAGE 02 MM-UNITS
PAGE 03 PROGRAM-DATUM
X  15.100
Z  20.000
PAGE 04 POINT-TO-POINT.G00,G01
X   0.000
Z -20.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1094
PAGE 05 POINT-TO-POINT.G00,G01
X  -2.000
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 06 POINT-TO-POINT.G00,G01
X   0.000
Z -56.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 07 POINT-TO-POINT.G00,G01
X   0.000
Z  56.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 08 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 09 POINT-TO-POINT.G00,G01
X   0.000
Z -51.754
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 10 POINT-TO-POINT.G00,G01
X   0.000
Z  51.754
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 11 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1310
```

```
PAGE 12 POINT-TO-POINT.G00,G01
X   0.000
Z -51.116
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1310
PAGE 13 POINT-TO-POINT.G00,G01
X   0.000
Z  51.116
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1310
PAGE 14 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 15 POINT-TO-POINT.G00,G01
X   0.000
Z -50.323
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 16 POINT-TO-POINT.G00,G01
X   0.000
Z  50.323
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 17 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 18 POINT-TO-POINT.G00,G01
X   0.000
Z -49.312
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 19 POINT-TO-POINT.G00,G01
X   0.000
Z  49.312
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 20 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 21 POINT-TO-POINT.G00,G01
```

```
X   0.000
Z -47.930
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 22 POINT-TO-POINT.G00,G01
X   0.000
Z  47.930
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 23 POINT-TO-POINT..G00,G01
X   6.000
Z   0.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 24 POINT-TO-POINT.G00,G01
X   0.000
Z -53.845
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 25 POINT-TO-POINT.G00,G01
X  -2.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1262
.
.
etc.
.
.
PAGE 90 POINT-TO-POINT.G00,G01
X   9.200
Z   0.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1646
PAGE 91 POINT-TO-POINT.G00,G01
X   0.000
Z  34.222
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1646
PAGE 92 END-PROGRAM..M02
```

B.9 [ME258.LATHE.PLT]QUEEN1.PTX

```
      BYTE TERM(5)
      INTEGER FEED1,FEED2,FEEDMX,TOOL1,TOOL2
      LOGICAL ABSOL,INCH,SPEN
      COMMON /MAIN/XPOS,ZPOS,ABSOL,INCH
      COMMON /SUB/FEED1,FEED2,FEEDMX,TOOL1,TOOL2, SPEN,IPEN
      DATA FEED2/0/,TOOL2/0/,PI/3.1415926/,TIME/0.0/
      FEEDMX=  200
      SPEN=.TRUE.
      CALL FRAMERG(  26.200,   96.000,4010)
CC  PAGE   1  INCREMENTAL FORMAT
      ABSOL=.FALSE.
CC  PAGE   2  MM-UNITS
      INCH=.FALSE.
CC  PAGE   3  PROGRAM-DATUM
      XNEW= -15.100
      ZNEW=  20.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/1000
      XPOS=XNEW
      ZPOS=ZNEW
      CALL MOVE(ZPOS,XPOS)
CC  PAGE   4  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2= 1000
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW= -20.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   5  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2=   80
      TOOL2=1
      CALL CHANGERG
      XNEW=   2.000
      ZNEW=   0.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
```

```
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   6  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2=   80
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW= -56.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   7  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2= 1000
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW=  56.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   8  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2=   80
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.800
      ZNEW=   0.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
```

```
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
      .
      .
      etc.
      .
      .
CC  PAGE  91  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2= 1000
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW=  34.222
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE  92  END-PROGRAM
      CALL MAKCUR
      TYPE *
      ACCEPT   100, TERM
      CALL NEWPAG
      CALL GRSTOP
      TERM(1)=32
      TERM(2)=27
      TERM(3)='"'
      TERM(4)='0'
      TERM(5)='g'
      TYPE   100, TERM
  100 FORMAT (1X,5A1)
      TYPE   150,TIME
  150 FORMAT (1X,'ESTIMATED MACHINING TIME IS: '
      ,F7.1,' MINUTES')
      STOP
      END
```

APPENDIX C

ORACAP Master Part Example Table

Table IV. File Sizes for Master Part Example

Option	Filename	ORAC Pages	Size (blocks)
Design	MASTER.CAD		1
	MASTER.END		1
	MASTER.FIL		1
	MASTER.OUT		1
	MASTER.SIZ		1
total	5		5
Generate	MASTER1.ANC	111	9
	MASTER2.ANC	28	3
	MASTER3.ANC	0	0
	MASTER4.ANC	5	1
	MASTER5.ANC	7	1
total	5	151	14
Optimize	MASTER1.BNC	75	7
	MASTER2.BNC	26	3
	MASTER3.BNC	0	0
	MASTER4.BNC	5	1
	MASTER5.BNC	7	1
total	5	113	12
Assemble	MASTER1.CNC	76	19
	MASTER2.CNC	41	10
total	2	117	29
Prove	MASTER1.PTX	76	85
	MASTER2.PTX	41	48
total	2	117	133

APPENDIX D

ORACAP Master Part Example Files

D.1 [ME258.LATHE.DAT]MASTER.CAD

```
AL6061T6  7  PART= T  AUTO= F  INCH= F       ! matl,nsect,flags
  1  2                                       ! sect,type
   0.000  25.000   5.000  16.000   7.000     ! p1,d1,p2,d2,rad
  2  5                                       ! sect,type
   5.000  16.000  14.000  10.000   8.000  17.000  10.000
                                       ! p1,d1,p2,d2,p3,d3,nsub
  3  1                                       ! sect,type
  14.000  10.000  21.000  16.000             ! p1,d1,p2,d2
  4  1                                       ! sect,type
  21.000  16.000  25.000  12.000             ! p1,d1,p2,d2
  5  1                                       ! sect,type
  25.000  12.000  28.000  12.000             ! p1,d1,p2,d2
  6  4                                       ! sect,type
  28.000  12.000  40.000  10.000   1.750     ! p1,od,p2,bd,pit
  7  3                                       ! sect,type
  34.000  12.000  40.000   0.000   6.000     ! p1,d1,p2,d2,rad
```

Where: matl is the material
nsect is the number of sections
flags are flags indicating options:
- PART= T indicates parting-off option
- AUTO= T indicates automatic machining
- INCH= T indicates inch units

sect is the section number
type is the section type
p1 is position one
d1 is the diameter at position one
p2 is position two
d2 is the diameter at position two
p3 is position three
d3 is the diameter at position three
rad is the radius
od is the outside diameter
bd is the base diameter
pi is the pitch

D.2 [ME258.LATHE.DAT]MASTER.END

```
 1
```

D.3 [ME258.LATHE.DAT]MASTER.FIL

```
 5
```

D.4 [ME258.LATHE.DAT]MASTER.OUT

```
 2
```

D.5 [ME258.LATHE.DAT]MASTER.SIZ

```
  26.200  40.000
```

D.6 [ME258.LATHE.CNC]MASTER1.ANC

```
INC
MM
PDAT  15.100  20.000
PTP    0.000 -20.000 1000 1 1094
PTP    0.000   0.000 1000 1 1214
PTP   -2.000   0.000   80 1 1214
PTP    0.000 -43.000   80 1 1214
PTP    0.000  43.000 1000 1 1214
PTP    2.000   0.000 1000 1 1214
PTP    0.000   0.000 1000 1 1262
PTP   -2.800   0.000   80 1 1262
PTP    0.000 -39.863   80 1 1262
PTP    0.000  39.863 1000 1 1262
PTP    2.800   0.000 1000 1 1262
PTP    0.000   0.000 1000 1 1310
PTP   -3.600   0.000   80 1 1310
PTP    0.000 -39.471   80 1 1310
PTP    0.000  39.471 1000 1 1310
PTP    3.600   0.000 1000 1 1310
PTP    0.000   0.000 1000 1 1358
PTP   -4.400   0.000   80 1 1358
PTP    0.000 -38.938   80 1 1358
PTP    0.000  38.938 1000 1 1358
PTP    4.400   0.000 1000 1 1358
PTP    0.000   0.000 1000 1 1406
PTP   -5.200   0.000   80 1 1406
PTP    0.000 -38.216   80 1 1406
PTP    0.000  38.216 1000 1 1406
PTP    5.200   0.000 1000 1 1406
PTP    0.000   0.000 1000 1 1454
PTP   -6.000   0.000   80 1 1454
PTP    0.000 -37.202   80 1 1454
PTP    0.000  37.202 1000 1 1454
PTP    6.000   0.000 1000 1 1454
PTP    0.000   0.000 1000 1 1502
PTP   -6.800   0.000   80 1 1502
PTP    0.000 -30.384   80 1 1502
PTP    0.000  30.384 1000 1 1502
PTP    6.800   0.000 1000 1 1502
PTP    0.000 -34.590 1000 1 1502
PTP   -6.800   0.000   80 1 1502
PTP    0.000  -0.917   80 1 1502
PTP    0.000   0.917 1000 1 1502
PTP    6.800   0.000 1000 1 1502
PTP    0.000  34.590 1000 1 1550
PTP   -7.600   0.000   80 1 1550
PTP    0.000 -17.680   80 1 1550
PTP    0.000  17.680 1000 1 1550
PTP    7.600   0.000 1000 1 1550
PTP    0.000   0.000 1000 1 1598
PTP   -8.400   0.000   80 1 1598
PTP    0.000 -16.080   80 1 1598
PTP    0.000  16.080 1000 1 1598
```

54	PTP	8.400	0.000	1000	1	1598
55	PTP	0.000	0.000	1000	1	1646
56	PTP	-9.200	0.000	80	1	1646
57	PTP	0.000	-4.253	80	1	1646
58	PTP	0.000	4.253	1000	1	1646
59	PTP	9.200	0.000	1000	1	1646
60	PTP	0.000	0.000	1000	1	1694
61	PTP	-10.000	0.000	80	1	1694
62	PTP	0.000	-2.595	80	1	1694
63	PTP	0.000	2.595	1000	1	1694
64	PTP	10.000	0.000	1000	1	1694
65	PTP	0.000	0.000	1000	1	1742
66	PTP	-10.800	0.000	80	1	1742
67	PTP	0.000	-1.657	80	1	1742
68	PTP	0.000	1.657	1000	1	1742
69	PTP	10.800	0.000	1000	1	1742
70	PTP	0.000	0.000	1000	1	1790
71	PTP	-11.600	0.000	80	1	1790
72	PTP	0.000	-1.016	80	1	1790
73	PTP	0.000	1.016	1000	1	1790
74	PTP	11.600	0.000	1000	1	1790
75	PTP	0.000	0.000	1000	1	1838
76	PTP	-12.400	0.000	80	1	1838
77	PTP	0.000	-0.564	80	1	1838
78	PTP	0.000	0.564	1000	1	1838
79	PTP	12.400	0.000	1000	1	1838
80	PTP	0.000	0.000	1000	1	1886
81	PTP	-13.200	0.000	80	1	1886
82	PTP	0.000	-0.258	80	1	1886
83	PTP	0.000	0.258	1000	1	1886
84	PTP	13.200	0.000	1000	1	1886
85	PTP	0.000	0.000	1000	1	1934
86	PTP	-14.000	0.000	80	1	1934
87	PTP	0.000	-0.074	80	1	1934
88	PTP	0.000	0.074	1000	1	1934
89	PTP	14.000	0.000	1000	1	1934
90	PTP	0.000	0.000	1000	1	1982
91	PTP	-14.800	0.000	80	1	1982
92	PTP	0.000	-0.002	80	1	1982
93	PTP	0.000	0.002	1000	1	1982
94	PTP	14.800	0.000	1000	1	1982
95	PTP	0.000	-20.540	1000	1	1550
96	PTP	-7.600	0.000	80	1	1550
97	PTP	0.000	-8.200	80	1	1550
98	PTP	0.000	8.200	1000	1	1550
99	PTP	7.600	0.000	1000	1	1550
100	PTP	0.000	-1.867	1000	1	1598
101	PTP	-8.400	0.000	80	1	1598
102	PTP	0.000	-5.230	80	1	1598
103	PTP	0.000	5.230	1000	1	1598
104	PTP	8.400	0.000	1000	1	1598
105	PTP	0.000	-1.867	1000	1	1646
106	PTP	-9.200	0.000	80	1	1646
107	PTP	0.000	-2.461	80	1	1646
108	PTP	0.000	2.461	1000	1	1646

109	PTP	9.200	0.000	1000	1	1646
110	PTP	0.000	0.000	1000	1	1646
111	PTP	0.000	44.273	1000	1	1646

D.7 [ME258.LATHE.CNC]MASTER1.BNC

```
INC
MM
PDAT  15.100   20.000
PTP    0.000  -20.000  1000  1  1094
PTP   -2.000    0.000    80  1  1214
PTP    0.000  -43.000    80  1  1214
PTP    0.000   43.000  1000  1  1214
PTP   -0.800    0.000    80  1  1262
PTP    0.000  -39.863    80  1  1262
PTP    0.000   39.863  1000  1  1262
PTP   -0.800    0.000    80  1  1310
PTP    0.000  -39.471    80  1  1310
PTP    0.000   39.471  1000  1  1310
PTP   -0.800    0.000    80  1  1358
PTP    0.000  -38.938    80  1  1358
PTP    0.000   38.938  1000  1  1358
PTP   -0.800    0.000    80  1  1406
PTP    0.000  -38.216    80  1  1406
PTP    0.000   38.216  1000  1  1406
PTP   -0.800    0.000    80  1  1454
PTP    0.000  -37.202    80  1  1454
PTP    0.000   37.202  1000  1  1454
PTP   -0.800    0.000    80  1  1502
PTP    0.000  -30.384    80  1  1502
PTP    0.000   30.384  1000  1  1502
PTP    6.800    0.000  1000  1  1502
PTP    0.000  -34.590  1000  1  1502
PTP   -6.800    0.000    80  1  1502
PTP    0.000   -0.917    80  1  1502
PTP    0.000    0.917  1000  1  1502
PTP    6.800    0.000  1000  1  1502
PTP    0.000   34.590  1000  1  1550
PTP   -7.600    0.000    80  1  1550
PTP    0.000  -17.680    80  1  1550
PTP    0.000   17.680  1000  1  1550
PTP   -0.800    0.000    80  1  1598
PTP    0.000  -16.080    80  1  1598
PTP    0.000   16.080  1000  1  1598
PTP   -0.800    0.000    80  1  1646
PTP    0.000   -4.253    80  1  1646
PTP    0.000    4.253  1000  1  1646
PTP   -0.800    0.000    80  1  1694
PTP    0.000   -2.595    80  1  1694
PTP    0.000    2.595  1000  1  1694
PTP   -0.800    0.000    80  1  1742
PTP    0.000   -1.657    80  1  1742
PTP    0.000    1.657  1000  1  1742
PTP   -0.800    0.000    80  1  1790
PTP    0.000   -1.016    80  1  1790
PTP    0.000    1.016  1000  1  1790
PTP   -0.800    0.000    80  1  1838
PTP    0.000   -0.564    80  1  1838
PTP    0.000    0.564  1000  1  1838
```

54	PTP	-0.800	0.000	80	1	1886
55	PTP	0.000	-0.258	80	1	1886
56	PTP	0.000	0.258	1000	1	1886
57	PTP	-0.800	0.000	80	1	1934
58	PTP	0.000	-0.074	80	1	1934
59	PTP	0.000	0.074	1000	1	1934
60	PTP	-0.800	0.000	80	1	1982
61	PTP	0.000	-0.002	80	1	1982
62	PTP	0.000	0.002	1000	1	1982
63	PTP	14.800	0.000	1000	1	1982
64	PTP	0.000	-20.540	1000	1	1550
65	PTP	-7.600	0.000	80	1	1550
66	PTP	0.000	-8.200	80	1	1550
67	PTP	0.000	8.200	1000	1	1550
68	PTP	-0.800	-1.867	80	1	1598
69	PTP	0.000	-5.230	80	1	1598
70	PTP	0.000	5.230	1000	1	1598
71	PTP	-0.800	-1.867	80	1	1646
72	PTP	0.000	-2.461	80	1	1646
73	PTP	0.000	2.461	1000	1	1646
74	PTP	9.200	0.000	1000	1	1646
75	PTP	0.000	44.273	1000	1	1646

D.8 [ME258.LATHE.CNC]MASTER1.CNC

```
PAGE 01 INCREMENTAL-FORMAT.G91
PAGE 02 MM-UNITS
PAGE 03 PROGRAM-DATUM
X  15.100
Z  20.000
PAGE 04 POINT-TO-POINT.G00,G01
X   0.000
Z -20.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1094
PAGE 05 POINT-TO-POINT.G00,G01
X  -2.000
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 06 POINT-TO-POINT.G00,G01
X   0.000
Z -43.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 07 POINT-TO-POINT.G00,G01
X   0.000
Z  43.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1214
PAGE 08 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 09 POINT-TO-POINT.G00,G01
X   0.000
Z -39.863
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 10 POINT-TO-POINT.G00,G01
X   0.000
Z  39.863
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1262
PAGE 11 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1310
```

```
PAGE 12 POINT-TO-POINT.G00,G01
X   0.000
Z -39.471
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1310
PAGE 13 POINT-TO-POINT.G00,G01
X   0.000
Z  39.471
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1310
PAGE 14 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 15 POINT-TO-POINT.G00,G01
X   0.000
Z -38.938
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 16 POINT-TO-POINT.G00,G01
X   0.000
Z  38.938
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1358
PAGE 17 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 18 POINT-TO-POINT.G00,G01
X   0.000
Z -38.216
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 19 POINT-TO-POINT.G00,G01
X   0.000
Z  38.216
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1406
PAGE 20 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 21 POINT-TO-POINT.G00,G01
```

```
X   0.000
Z -37.202
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 22 POINT-TO-POINT.G00,G01
X   0.000
Z  37.202
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1454
PAGE 23 POINT-TO-POINT.G00,G01
X  -0.800
Z   0.000
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1502
PAGE 24 POINT-TO-POINT.G00,G01
X   0.000
Z -30.384
FEEDRATE   80
TOOL-NO 1
SPINDLE-SPEED 1502
PAGE 25 POINT-TO-POINT.G00,G01
X   0.000
Z  30.384
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1502
.
.
etc.
.
.
PAGE 74 POINT-TO-POINT.G00,G01
X   9.200
Z   0.000
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1646
PAGE 75 POINT-TO-POINT.G00,G01
X   0.000
Z  44.273
FEEDRATE 1000
TOOL-NO 1
SPINDLE-SPEED 1646
PAGE 76 END-PROGRAM..M02
```

D.9 [ME258.LATHE.PLT]MASTER1.PTX

```
         BYTE TERM(5)
         INTEGER FEED1,FEED2,FEEDMX,TOOL1,TOOL2
         LOGICAL ABSOL,INCH,SPEN
         COMMON /MAIN/XPOS,ZPOS,ABSOL,INCH
         COMMON /SUB/FEED1,FEED2,FEEDMX,TOOL1,TOOL2, SPEN,IPEN
         DATA FEED2/0/,TOOL2/0/,PI/3.1415926/,TIME/0.0/
         FEEDMX=  200
         SPEN=.TRUE.
         CALL FRAMERG(  26.200,  80.000,4010)
CC   PAGE    1  INCREMENTAL FORMAT
         ABSOL=.FALSE.
CC   PAGE    2  MM-UNITS
         INCH=.FALSE.
CC   PAGE    3  PROGRAM-DATUM
         XNEW= -15.100
         ZNEW=  20.000
         IF (INCH) XNEW=XNEW*25.4
         IF (INCH) ZNEW=ZNEW*25.4
         DIST=SQRT(XNEW**2+ZNEW**2)
         TIME=TIME+DIST/1000
         XPOS=XNEW
         ZPOS=ZNEW
         CALL MOVE(ZPOS,XPOS)
CC   PAGE    4  POINT-TO-POINT INTERPOLATION
         FEED1=FEED2
         TOOL1=TOOL2
         FEED2= 1000
         TOOL2=1
         CALL CHANGERG
         XNEW=   0.000
         ZNEW= -20.000
         IF (INCH) XNEW=XNEW*25.4
         IF (INCH) ZNEW=ZNEW*25.4
         IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
         IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
         TIME=TIME+DIST/FEED2
         IF (ABSOL) XPOS=XNEW
         IF (ABSOL) ZPOS=ZNEW
         IF (.NOT.ABSOL) XPOS=XPOS+XNEW
         IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
         CALL DRAW(ZPOS,XPOS)
CC   PAGE    5  POINT-TO-POINT INTERPOLATION
         FEED1=FEED2
         TOOL1=TOOL2
         FEED2=   80
         TOOL2=1
         CALL CHANGERG
         XNEW=   2.000
         ZNEW=   0.000
         IF (INCH) XNEW=XNEW*25.4
         IF (INCH) ZNEW=ZNEW*25.4
         IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
         IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
```

```
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   6  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2=   80
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW= -43.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   7  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2= 1000
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW=  43.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE   8  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2=   80
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.800
      ZNEW=   0.000
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
```

```
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
      .
      .
      etc.
      .
      .
CC  PAGE  75  POINT-TO-POINT INTERPOLATION
      FEED1=FEED2
      TOOL1=TOOL2
      FEED2= 1000
      TOOL2=1
      CALL CHANGERG
      XNEW=   0.000
      ZNEW=  44.273
      IF (INCH) XNEW=XNEW*25.4
      IF (INCH) ZNEW=ZNEW*25.4
      IF (ABSOL) DIST=SQRT((XNEW-XPOS)**2+(ZNEW-ZPOS)**2)
      IF (.NOT.ABSOL) DIST=SQRT(XNEW**2+ZNEW**2)
      TIME=TIME+DIST/FEED2
      IF (ABSOL) XPOS=XNEW
      IF (ABSOL) ZPOS=ZNEW
      IF (.NOT.ABSOL) XPOS=XPOS+XNEW
      IF (.NOT.ABSOL) ZPOS=ZPOS+ZNEW
      CALL DRAW(ZPOS,XPOS)
CC  PAGE  76  END-PROGRAM
      CALL MAKCUR
      TYPE *
      ACCEPT    50, TERM
      CALL NEWPAG
      CALL GRSTOP
      TERM(1)=32
      TERM(2)=27
      TERM(3)='"'
      TERM(4)='0'
      TERM(5)='g'
      TYPE    50, TERM
   50 FORMAT (1X,5A1)
      TYPE   100,TIME
  100 FORMAT (1X,'ESTIMATED MACHINING TIME IS: '
      ,F7.1,' MINUTES')
      STOP
      END
```

APPENDIX E

Data File for ORACAP Materials

E.1 [ME258.LATHE.DAT]MATERIAL.DAT

```
ABCDEFGH 1234 1234 1234   1.234   1.234   1.234  AB
AL6061T6 1000   80    5   0.800   0.080   0.040  MM
ALUMINUM 1000   80    5   0.800   0.080   0.040  MM
ALUMSOFT 1000   80    5   1.000   0.100   0.050  MM
ALUMHARD 1000   80    5   0.600   0.060   0.030  MM
ST1018   1000   80    2   0.500   0.050   0.020  MM
STEEL    1000   80    2   0.500   0.050   0.020  MM
al6061t6 1000   80    5   0.800   0.080   0.040  mm
aluminum 1000   80    5   0.800   0.080   0.040  mm
alumsoft 1000   80    5   1.000   0.100   0.050  mm
alumhard 1000   80    5   0.600   0.060   0.030  mm
st1018   1000   80    2   0.500   0.050   0.020  mm
steel    1000   80    2   0.500   0.050   0.020  mm
ALUMINUM   40    3    1   0.030   0.003   0.002  IN
aluminum   40    3    1   0.030   0.003   0.002  in
!
!       FORMATTED TABLE CONTAINING FEEDRATE AND
!       DEPTHS FOR VARIOUS MATERIALS TO BE MACHINED
!       ON THE ORAC CNC LATHE.
!
!       COLUMN 1 - 8 CHARACTER MATERIAL NAME
!       COLUMN 2 - HIGH FEEDRATE FOR TRAVEL
!       COLUMN 3 - LOW FEEDRATE FOR CUTTING
!       COLUMN 4 - FEEDRATE FOR PARTING OFF
!       COLUMN 5 - DEPTH OF ROUGH CUTS
!       COLUMN 6 - DEPTH OF FINISHING CUTS
!       COLUMN 7 - DEPTH OF THREAD CUT INCREMENT
!       COLUMN 8 - INCH OR MM UNITS
!
```