

THE APPLICATION OF A POINT-TO-POINT NUMERICAL CONTROL
MACHINE FOR CONTINUOUS CONTOUR CUTTING

THE APPLICATION OF A POINT-TO-POINT NUMERICAL CONTROL
MACHINE FOR CONTINUOUS CONTOUR CUTTING

By

MOSES S. WONG, B.Eng.

Thesis

Submitted To Faculty of Graduate Studies
in Partial Fulfilment of the Requirements

For The Degree

Master of Engineering

McMaster University

December, 1971

MASTER OF ENGINEERING (1971)
(Production)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: THE APPLICATION OF A POINT-TO-POINT NUMERICAL
CONTROL MACHINE FOR CONTINUOUS CONTOUR CUTTING

AUTHOR: Moses S. Wong, B.Eng. (McGill), P.Eng.

SUPERVISOR: Professor J.L. Duncan

NUMBER OF PAGES: *vi*, 113

SCOPE AND CONTENTS: This thesis is a feasibility study of the application of a point-to-point numerical control (N/C) machine for contour cutting work. It comprises three basic parts which are divided into ten chapters.

The first part deals with the primary objectives of this project and the N/C machine's hardware and software descriptions.

The second part introduces the utilization of computer-aided programming and presents an original translation program necessary for this particular work.

The last part gives several examples and describes the outcomes of the study, and is concluded with some objective discussions.

ACKNOWLEDGEMENTS

The author wishes to express his sincere gratitude to Professor J.L. Duncan for his guidance and constant encouragement during the course of preparation of this thesis.

The author would also like to thank Dr. A.R. Elliott of the Electrical Engineering Department for his thorough instructions on the utilization of the PDP-8/L computer and for allowing generous access to the computer facilities.

Appreciations are here expressed to the personnel of the machine shop for their hearty co-operations and enduring patience throughout the project.

The author is also indebted to Mr. Salmon, Chairman of the Mechanical Engineering Department of Mohawk College, for allowing liberal use of their metrology facilities.

The author wishes to acknowledge the support given to this program by NRC and CIM.

Finally, many thanks to Miss Lynn Paquette for her expert typing of this manuscript.

TABLE OF CONTENTS

<u>CHAPTER</u>	<u>PAGE</u>
1. INTRODUCTION	1
2. DESCRIPTION OF THE MOOG HYDRA-POINT MACHINING CENTRE	7
3. APPLICATION OF THE POINT-TO-POINT MACHINE FOR CONTOUR CUTTING	17
4. SIMPLIFIED DESCRIPTION OF THE PDP-8/L COMPUTER SYSTEM	30
5. AN ORIGINAL PROGRAM FOR THE TRANSLATION OF ASCII INTO EIA USING THE PDP-8/L COMPUTER	41
6. PROGRAMS FOR DERIVATION OF COORDINATES OF SIMPLE CURVES	55
7. MACHINING OF TEST PARTS	66
8. MACHINING OF LARGE SCALE MODELS OF A FLUID LOGIC ELEMENT.....	78
9. PERFORMANCE TESTS OF THE FLUIDIC BISTABLE MODELS	87
10. DISCUSSIONS AND CONCLUSIONS	92
REFERENCES	97

TABLE OF CONTENTS (Continued)

<u>APPENDICES</u>		<u>PAGE</u>
APPENDIX A	THE ADVANTAGES OF NUMERICAL CONTROL MACHINES TOOLS	100
APPENDIX B	PAL PRINTOUT OF THE EIA CONVERSION PROGRAM	105
APPENDIX C	FORTRAN CIRCLE PROGRAM	108
APPENDIX D	FORTRAN ELLIPSE PROGRAM	111

CHAPTER I
INTRODUCTION

Numerical control is a relatively new form of automation. It is generally applied in metal cutting machine tools although forming, drafting, assembling, wire-wrapping, welding and inspection machines are making extensive use of numerical control systems nowadays.

Since its inception in the late 1940s, numerical control has gained great momentum during the last twenty years. The popularity of numerical control machines lies not in the fact that they are novel but almost solely in their ability to permit the manufacture of superior products far more economically than with manual control or other conventional means of manufacturing methods in certain types of parts and for certain quantities of production.

Contrary to the popular belief, numerical control machines are NOT generally designed for automation in large quantity production; they are, in fact, more suitable for small lot production and are characterized by their ability to reduce substantially set-up time and unproductive man-hours in short

manufacturing runs. In some cases, it has been found more economical to manufacture one part by numerical control means than by conventional methods. Machining prototypes and experimental models, where quick design changes are often called for, has utilized the versatility of numerical control machines with the consequent of savings in cost as well as in time.

Basically there are two forms of numerical control systems, point-to-point and continuous contouring; the former is less expensive but has very wide applications, from simple drill press and punch press to complicated electronic microcircuit board assembly operations and has accounted for 80 percent of the numerical control installations. The name is virtually self-explanatory; this type of machine can perform precision positioning operations automatically where the exact path that the cutter takes in moving from one point to another is immaterial; providing of course, the time it takes is reasonably short and the cutter does not collide with the fixture or the work-piece in doing so.

On the other hand, continuous contouring is an extremely sophisticated form of automatic control

whose application in contour or skin milling has made possible a new dimension of machining which was once believed to be impossible by conventional methods. Its ability to employ more than three axes (up to five or six axes are possible) of machining enables the SCULPTURING of a workpiece in a single set-up rather than the two or three normally required. This greatly enhances the manufacture of curved and difficult parts such as turbine blades, aircraft wing structures and engine carbureters.

Unlike point-to-point control, continuous contouring control is provided with precisely computed information about the desired path to be followed in addition to the point-to-point accuracy. The word "computed" is used because the information is normally obtained through the use of a high-speed electronic computer either as a preprogrammed subroutine or as part of the built-in control package.

The determination of the contouring path between two defined points is done by a number of interpolation techniques. The simplest form is "linear interpolation". More sophisticated versions such as circular, parabolic and even cubic interpolation methods are also available. Basically, all the inter-

polation techniques are used to provide the data for the velocities of the combined axes as well as the intermediate coordinates necessary to curve-fit the desired contour. Linear interpolation is used where tangents can be drawn along a curve within an allowable error band. For more precision work, however, higher order interpolations are required.

It is expected then, that continuous contouring machines are more expensive than point-to-point machines, as almost 40 percent of the dollars invested have been for the contouring type even though 80 percent of the numerical control machines are point-to-point. It should be emphasized here, however, that it is both possible and practical to utilize continuous contouring machines for performing point-to-point operations whereas point-to-point systems are restricted, in general to simple operations.

It is possible however that the usefulness of point-to-point machines could be increased considerably if inexpensive means could be found of performing limited contour machining. Such an application would be economically valid provided it did not require increased capital investment and provided also that the actual

machining time in the contouring application was not excessive. It is also important that programming costs for contour machining using a point-to-point machine should be reasonable.

In this thesis, a technique which fulfils these requirements is presented. Programming and machining times for making a number of parts are given.

The machine available was a standard 3-axis vertical machining centre with standard drilling, boring, tapping and straight-edge milling capabilities; additional facilities such as a 24-tool carousel and pneumatic tape punch are also fitted.

The attempt to utilize the point-to-point machine for contour cutting requires the controlling of the cutter movements in very small discrete steps about the desired path (a kind of microscopic point-to-point application) and as the number of instructions is inevitably greater this means a deviation from the standard point-to-point programming practice.

The machine is provided with a number of discrete X-Y Feed Rates which allows only limited Feed

Rate combinations for linear slope interpolation; it is possible, however, to make use of the Feed Rates independently to provide interpolation in either X or Y direction for any desired curvature which will be dealt with in depth in the subsequent Chapters.

The thesis describes all aspects of this project in the following Chapters:

Chapter 2 gives a more detailed description of the Moog Hydra-Point Machining Centre and Chapter 3, the Application of the Machine for Contour Cutting.

Chapters 4, 5, and 6 indicate the utilization of computer-aided programming to produce control tapes for machining particular as well as general contour paths.

Chapters 7 and 8 describe the actual parts machined and also describe experimental models of a fluidic element produced by this technique. Chapter 9 gives a brief account on the performance tests of the fluidic elements.

Chapter 10 concludes the thesis with discussions and reiterates some of the primary objectives.

CHAPTER 2

DESCRIPTION OF THE MOOG HYDRA-POINT MACHINING CENTRE

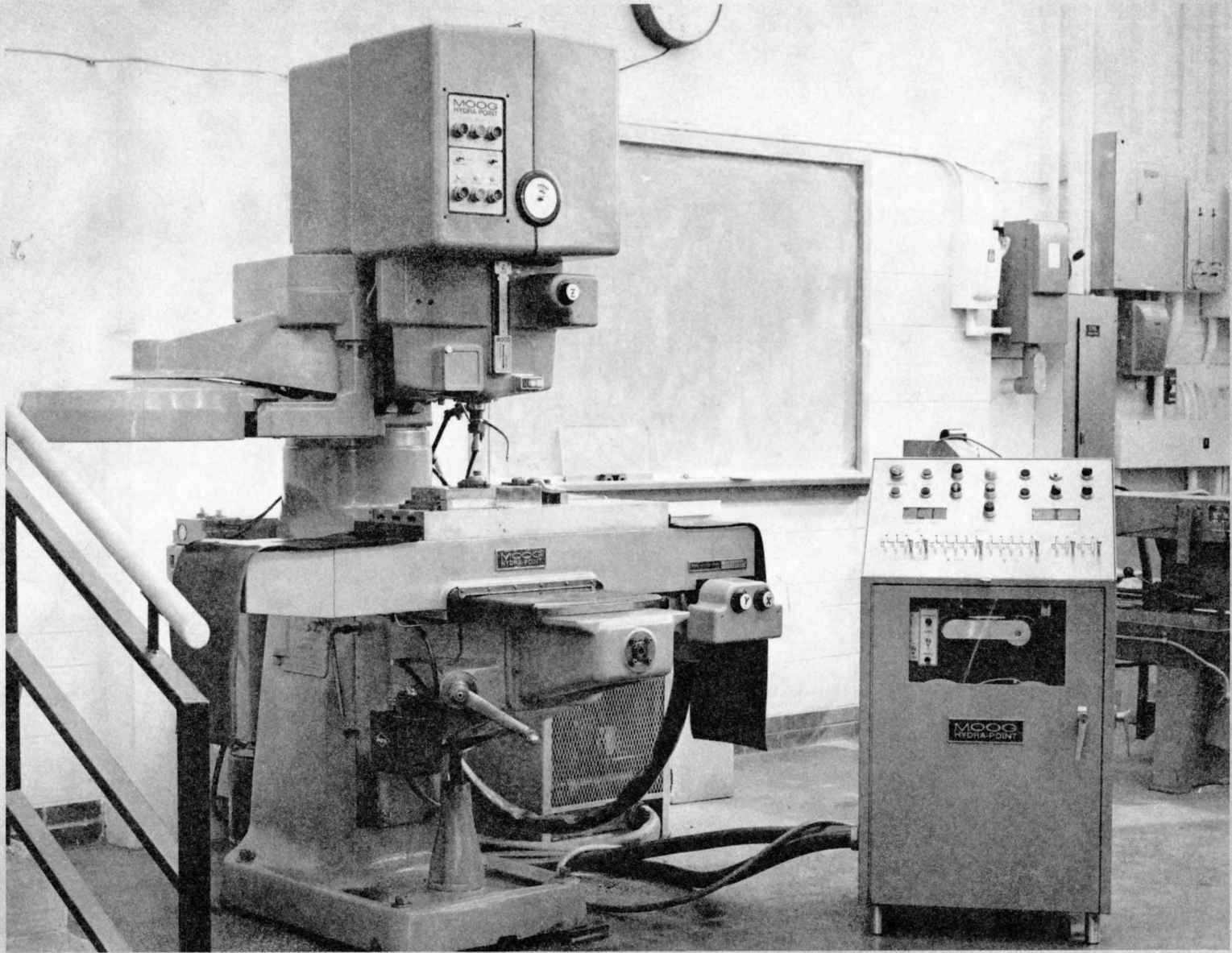
HARDWARE SYSTEM

The Moog Hydra-Point model 83-1000 MC is a tape-controlled, 3-axis, point-to-point positioning, vertical spindle machine which is capable of rapid positioning of a workpiece for successive machining operations. The basic Hydra-Point Machining Centre consists of a Control Console, a Hydraulic Power Supply, a Bridgeport Model HP milling machine, and a Moog Automatic Tool Changer as shown in Figure 2.1.

The Bridgeport machine is specially designed and manufactured for the numerical control system which includes pneumatic sensing, logic and output elements. Actuation of the table and saddle is achieved by hydraulic cylinders. The spindle is driven by a 1.5 HP, 1730 RPM motor. The maximum X-Y travel is 20" by 10" with a spindle travel of 5".

The Control Console consists of a control panel, a tape reading and a tape punching system and relays for operating the pneumatic and hydraulic valves. The control panel comprises a manual dial assembly and the necessary controls and indicators to energize and operate the system.

FIGURE 2.1



The tape reader, located on the front, consists of a reading head, a take-up reel and a rewinding reel with the appropriate control levers and switches.

The pneumatic tape punch is located on the right vertical side of the Console which is actuated manually with the control dial knobs on the control panel.

The Power Supply consists of a hydraulic pump, a double-ended hydraulic pump motor, a radiator, a fan, a reservoir, a relief valve, a supply filter and a bypass filter. The hydraulic fluid is presurized to 1500 psig at the manifold valve. The supply filter contains 20 micron and the bypass filter 5 micron elements.

The Automatic Tool Changer is a 24-tool carousel which provides random or sequential tool selection using the proper Miscellaneous Function and Tool Function Codes in the tape.

The control specifications of the machine are:

Absolute positioning accuracy:	$\pm .001''$
Precision:	$\pm .0003''$
Resolution:	$.001''$
X-Y Travel:	20" by 10"
Positioning speed:	300"/min.
X-Y Feed Rates:	1 to 40"/min $\pm 20\%$
Z Feed Rates:	1 to 64"/min. $\pm 20\%$
Zero Adjustment:	$\pm .02''$

SOFTWARE SYSTEM

The input information to the machine is by holes punched in a standard tape (paper or otherwise). The tape specifications are 1.000" \pm .003" wide by .004" \pm .0003" thick. It is divided into 9 equally-spaced tracks across its width. The centres of the tracks are the locations of the punched holes which are .072" \pm .001" \pm .002" dia. each and each row of holes are separated by 0.1" from another, with cumulative error of \pm .009" in 6".

The first three and the 5th tracks are used to designate binary values of 2^0 , 2^1 , 2^2 , 2^3 respectively while the 4th track is occupied by smaller guide holes of .046" \pm .002" \pm .001" dia. each. See Figure 2.2 for a sample of the tape.

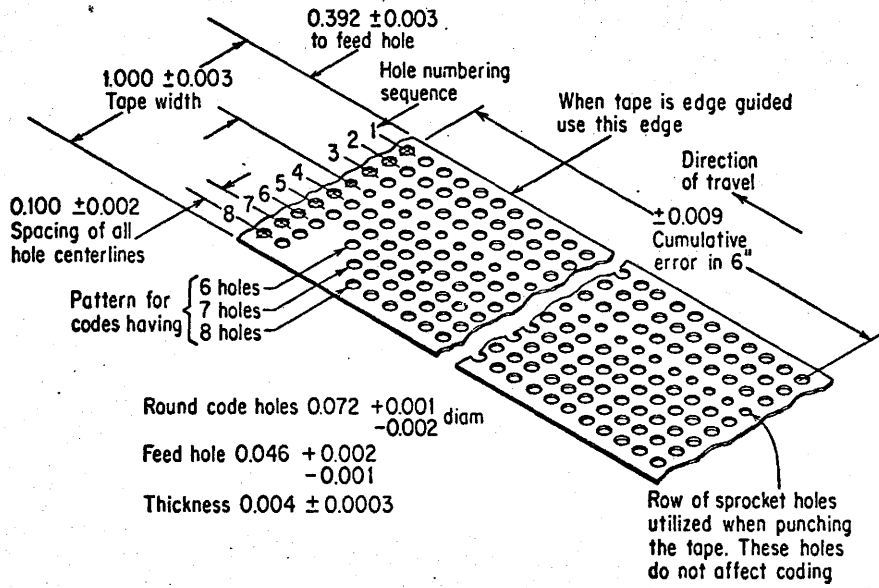


FIGURE 2.2

A maximum of 4 bits of binary digits are used because the control logic uses the Binary-Coded-Decimal numbering system so that any decimal number from 0 to 9 can be represented by the combination of binary digits in one row of the track. For example, the value 5 is represented by $2^0 + 2^2$ or the presence of a hole punched in track one and track three. The value 8 is a hole in track five. The absence of holes in a row of tracks represents a zero value. Only one decimal digit can be coded in each row.

The Moog Hydra-Point uses a unique programming format which always requires a constant number of decimal digits to specify a given block of information; in other words, it uses a "fixed block format". All information is represented by numbers only and conforms to the EIA (Electronic Industries Association) Standards of coding which are basically Binary-Coded-Decimal as described before.

The length of each block is 20 digits which is divided as follows (Special Functions to be described later):

The first 3 digits: used as sequence number which are not operational.

The 4th digit: for the Preparatory Function code.

The next 5 digits: for X dimension with 3 figures after decimal (or for Feed Point of Z-axis).

The following 5 digits: for X dimension with 3 figures after decimal (or for Final Depth of Z-axis).

(Note: the decimal point is not coded since it is always at a fixed location).

The 15th digit: reserved digit which is always zero.

The 16th and 17th digits: for Tool Function code.

The 18th and 19th digits: for Miscellaneous Functions code.

The 20th digit: always an End-of-Block code or Carriage Return code in the EIA Standards which is a single hole occupying the 9th track (or the 8th bit) in the tape.

The particular numbers used for the Preparatory Functions and Miscellaneous Functions are shown in Table 2.3. The Tool Function Code is shown in Table 2.4. A sample tape with a block of information is shown in Figure 2.5.

PREPARATORY FUNCTIONS

- 0 — POSITION ONLY
- 1 — TAP-FEED CONTROL
- 4 — PECK
- 5 — DRILL
- 6 — MILL
- 7 — BORE
- 8 — TAP-PITCH CONTROL
- 9 — READ Z

MISCELLANEOUS FUNCTIONS

- 00 — PROGRAM STOP
- 02 — END OF PROGRAM
- 06 — TOOL CHANGE (MANUAL)
- 07 — FLOOD COOLANT ON
- 08 — MIST COOLANT ON
- 09 — COOLANT OFF
- 80 — NO CHANGE
- 81 — PARTIAL RETRACT ON
- 82 — PARTIAL RETRACT OFF
- 83 — INDEXING - WORK PIECE
- 84 — "X" FEED RATE
- 85 — "Y" FEED RATE
- 86 — AUTOMATIC TOOL CHANGE
- 88 — SPINDLE SPEED CHANGE
- 89 — "Z" FEED RATE

TABLE 2.3

TOOL FUNCT. CODE	X, Y AXIS FEED IN/MIN	Z AXIS FEED IN/MIN	SPINDLE SPEED R. P. M.	TOOL FUNCT. CODE	SPINDLE SPEED R. P. M.
01	1.0	1.0	65	21	525
02	1.9	2.1	75	22	625
03	2.7	3.6	91	23	760
04	4.0	5.3	105	24	875
05	5.5	8.1	120	25	1015
06	7.2	10.3	140	26	1140
07	9.1	13.3	155	27	1300
08	11.2	16.4	175	28	1450
09	13.3	20	195	29	1610
10	16.0	25	215	30	1800
11	18.6	28	240	31	2000
12	21.5	32	270	32	2225
13	24.7	37	300	33	2470
14	27.9	42	330	34	2740
15	31.5	48	375	35	3100
16	35.6	53	415	36	3450
17	39.7	59	465	37	3830
18	44.0	64	495	38	4090

TABLE 2.4

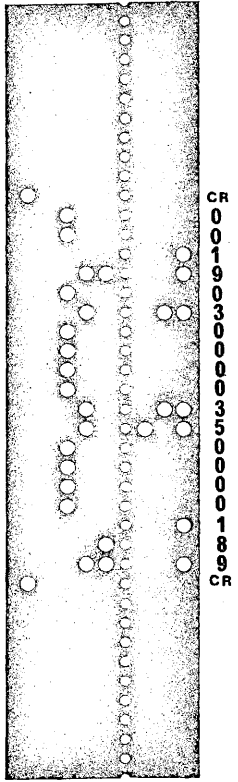


FIGURE 2.5

Unlike the other N/C systems which use numerals as well as letters to specify certain instructions, the Moog Hydra-Point utilizes only the first 4 binary bits plus the 8th bit. In the machine tape reader head, all other tracks except the first 4 and the 8th tracks are masked off. In so doing, it will accept either the parity or non-parity EIA codes. The parity EIA codes always have odd number (odd parity check) of holes in every row.

CHAPTER 3APPLICATION OF THE POINT-TO-POINT MACHINE FOR CONTOUR
CUTTING

Since the Moog Hydra-Point is basically a point-to-point machine, it requires only little effort in programming for positioning cutting. For example, if one wishes to drill 4 holes at points A,B,C,D, with dimensions 1.000, 1.000; 1.000, 2.000; 4.000, 2.000; 4.000, 1.000 inches respectively, the program for the Machine will look like the following:

```

001 0 00000 00000 0 35 88 ↓
002 9 03000 03500 0 01 89 ↓
003 5 01000 01000 0 01 89 ↓
004 5 01000 02000 0 01 89 ↓
005 5 04000 02000 0 01 89 ↓
006 5 04000 01000 0 01 89 ↓

```

(Spaces between numbers are used for clarity only)

Each row of numbers plus an End-of-Block (or Carriage Return), represented by a fullstop and a return arrow sign, comprises a block of 20 digits information. The first 2 rows of information given in the example, instruct the Machine to operate at the selected spindle

speed (in this case 3100 RPM), and read in the Z Feed Point (3.000 inches) and Final Z depth (3.500 inches). Rows number 003, 004, 005 and 006 command the Machine to drill the four holes A, B, C, D at the given locations with the given Z depth. Of course the proper tool size has to be determined beforehand and other operational instructions such as Coolant ON, Automatic Tool Change, can be inserted before the program as given in the example.

These numbers will then be punched in a paper tape using a Flexowriter (which produces parity EIA codes) in an unbroken string.

Similarly, the milling operation can be done by using the Preparatory Function code 6 (in column 4) with the final X-Y location and the appropriate X or Y Feed Rates. Other operations such as Boring, Pecking, Tapping, can also be performed using the proper Preparatory Function codes.

The machine Control System reads in the numbers punched in the tape and is able to interpret and perform the corresponding operations as required by the programmer. Normally a few blocks of information are sufficient to cause the machine to perform quite a number of operations

in rapid successions and with great precisions if they are restricted to macroscopic positioning operations.

The attempt to utilize the Machine to perform contour cutting (mainly milling) requires a totally different dimension of programming considerations.

Basically, the Machine is only position sensitive, i.e., it will obey the positioning instruction to go from point A to point B through the use of X or/and Y Feed Rates which are normally restricted to a number of discrete steps (18 in this machine), and are only $\pm 20\%$ accurate of the stated values. However, the positioning accuracy is fairly high: i.e., within $\pm .0003"$ of the programmed point.

Milling along the X or Y axis will have no problem since it will take on a selected Feed Rate (which will not be critical if the spindle speed is matched), and will arrive at the designated locations accurately. However, to mill along a slope (not to mention a curve), does present some difficulties.

It is possible through the tape commands to combine the X and Y Feed Rates simultaneously to produce

a slope, but the combined Feed Rates do not necessarily guarantee that the travel of the table will be along the correct path nor arrive at the desired point as dictated by the programmer.

Take a simple example. If one wishes to mill an accurate 30° slope with respect to the X-axis, one would have to use a X/Y feedrate ratio of $\sqrt{3}/1$ or 1.732. In examining through the Feed Rate Table, the closest ratio is 1.9/1.0 and 1.9 and the next one is 4.0/2.7 or 1.48. If one chooses the higher ratio, 1.9, the resultant path the Machine will follow can be shown in Figure 3.1. Alternately, if the low X/Y ratio is chosen, the actual cutting path will take on a different slope as shown in the same Figure. Therefore, there is no way of milling an accurate 30° slope with a single instruction.

Furthermore, the actual path will vary slightly from the predicted one due to inaccuracies in feed rate control. In theory, a 45° slope can be produced at any feed rate by making the X and Y feeds equal, however, the high tolerance in the feed rate control would result in a maximum of 40% deviation from the intended angle.

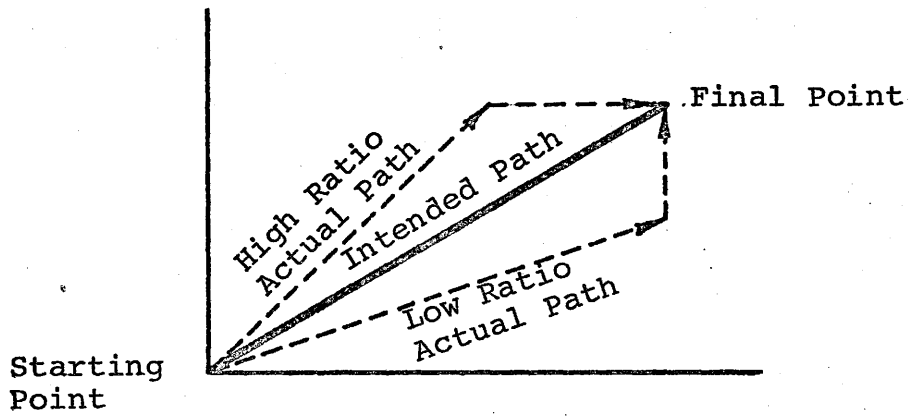


FIGURE 3.1

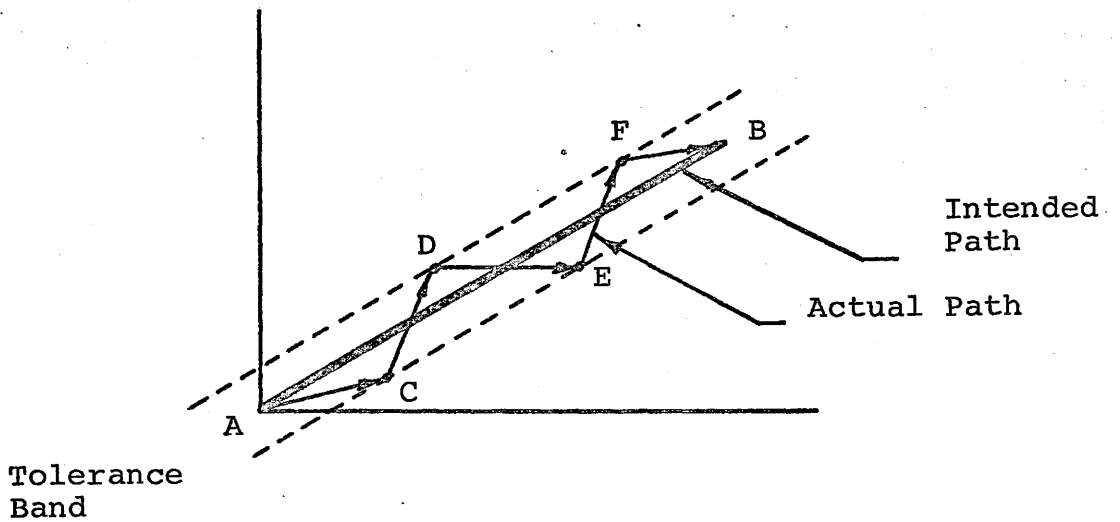


FIGURE 3.2

Nevertheless, it does not imply, from the preceding discussions, that one will not be able to obtain a slope of any value nor even a curved path, if one allows a certain band of tolerance for that intended path. Take the following example.

Figure 3.2 shows a desired slope to be machined from A to B. The Machine is not capable of producing such a slope using a single instruction but if a certain error band or deviation can be tolerated, the selection of certain combined feed rates of X and Y can cause the Machine table to travel within the tolerance band and stop over at the intermediate points C, D, E and F before arriving at point B. The knowledge of the coordinates of the points C, D, E, and F has to be known beforehand so that these boundary values can be programmed into the Machine together with the combined X-Y Feed Rates. This is called the Piece-wise Linear Technique which can be extended to produce curvature of any desired functions within the capability of the Machine.

If the tolerance band is quite narrow, say ± 0.005 ", a slope of 5 inches long will require as many as a hundred or even more blocks of information depending on the right choice of combined X-Y Feed Rates. The tasks

of determining these values and the intermediate coordinates will not be feasible if done manually, but, a computer program can be written to select and determine these values and assemble them in the correct block sequence format. This method has been adopted elsewhere (reference 18) and applied to this Moog machine.

A second method, known as the Step-wise Linear Technique, for producing non-standard paths, is possibly a more convenient and flexible method which will be discussed in depth in the following sections.

Consider the same slope used in the previous section with the same tolerance band as shown in Figure 3.3. It is possible to instruct the Machine to travel from point A along the Y-axis until it reaches the upper boundary defined by the tolerance limit, and stop at point C. From C it is instructed to change direction and travel along the X-axis until it hits point D at the lower boundary of the line AB. The same process will repeat itself to points E, F, G and H until it finally arrives at point B which is the desired location. Almost an equal number of X-steps as well as Y-steps will result in traversing from A to B.

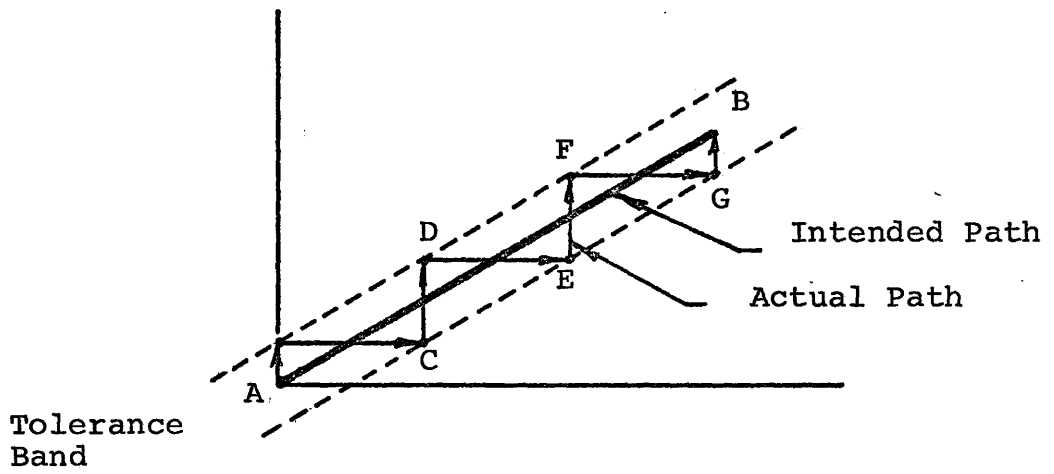


FIGURE 3.3

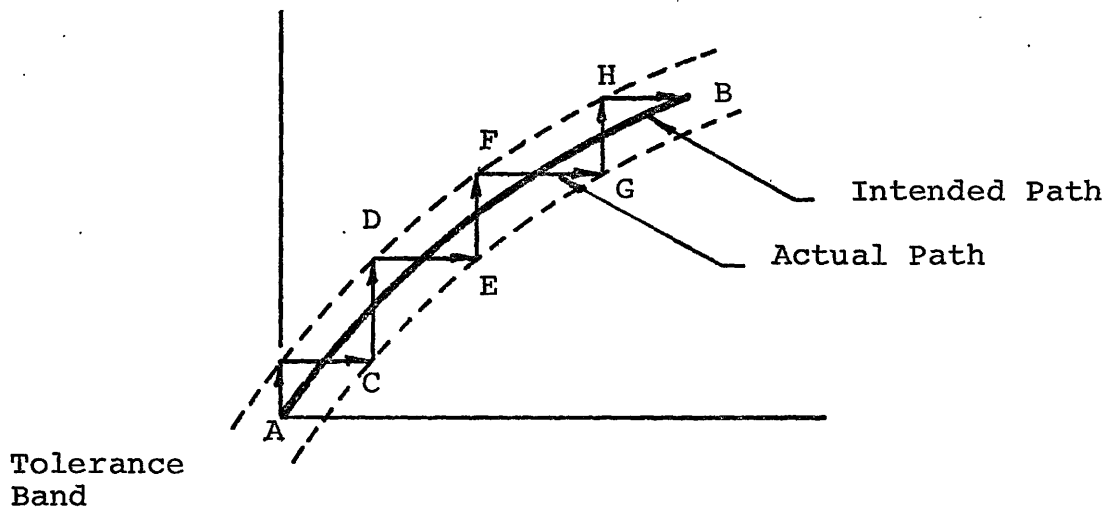


FIGURE 3.4

In the case of a curved path, the same procedure can be applied. Take Curve AB as shown in Figure 3.4. If AB can be defined mathematically, the Step-wise Linear Technique can also be applied without any difficulties.

The upper and lower boundary points C, D, E, F, etc of Curve AB can be determined accurately because they are geometrical locations and not velocity vector points. The cutting paths along the Y and X axes, therefore, are absolutely independent of the X-Y Feed Rates or their combinations. This is consistent with the prime design factors of the Machine which does not intend that the X-Y Feed Rates should influence the cutting paths and the final positions. This is an important principle in determining the appropriate interpolation procedure for a particular machine. In those machines in which the feed rates are subject to some variation due to external factors or where the ratios between X and Y rates are not precisely controlled, the actual cutting path in a combined feed rate motion is uncertain.

Of the two methods of interpolation described above, the author chose the latter because it is completely unrestrained in feed rate selections and much simpler to

program. The freedom of choice of the X-Y Feed Rates offers wider scope of programmable cutting speed for different metals from Aluminum to Stainless Steel and even for non-metals like Plexiglas and plastics. The choice of tool size will also be wider since a smaller feed rate will accommodate a smaller cutter which is particular important for narrow slot cutting.

Using the first method of combined X-Y Feed Rates, the size of tools; depth of cut and types of metal are restricted by the highest X or Y Feed Rate component which must be used to produce the desired slope. The unpredictable feed rate tolerance may also result in the cutter travelling outside of the prescribed error band before reaching the intermediate points.

One advantage of using the Piece-wise Linear Technique (1st Method) is the result of fewer blocks of information when compared with that required by the Step-wise Linear Technique (2nd Method) for the same segment of the curve. However, when the lowest Feed Rate (1"/min.) must be used in certain cases of machining, the Piece-wise Linear will approach the limiting case identical to the Step-wise Linear Technique.

Comparing the complexity of computer programming to produce the desired intermediate coordinate points for the two methods, the latter proves to be more superior since it only requires the definition of the intended paths mathematically. The tolerance error band will then be a constant value added to or subtracted from the X or Y variables along the curve.

Starting from an arbitrary starting point (such as A in Figure 3.4), the Y distance travelled can be determined by the iteration method in the Y direction until the new Y value and the current X value satisfy the upper limiting curve equation at C. This X-Y coordinate at C will serve as a new starting point with a current Y value for the iteration in the X direction until they fit the solution of the lower limiting curve equation. This stepping process is continued until the final point B is reached. A Flow Chart, depicting the programming procedure for the Step-wise Linear Technique, is shown in Figure 3.5.

In comparison, the Piece-wise Linear programming will involve the pairing of the X-Y Feed Rates to produce the intermediate slopes in addition to the iteration procedure for the intermediate points.

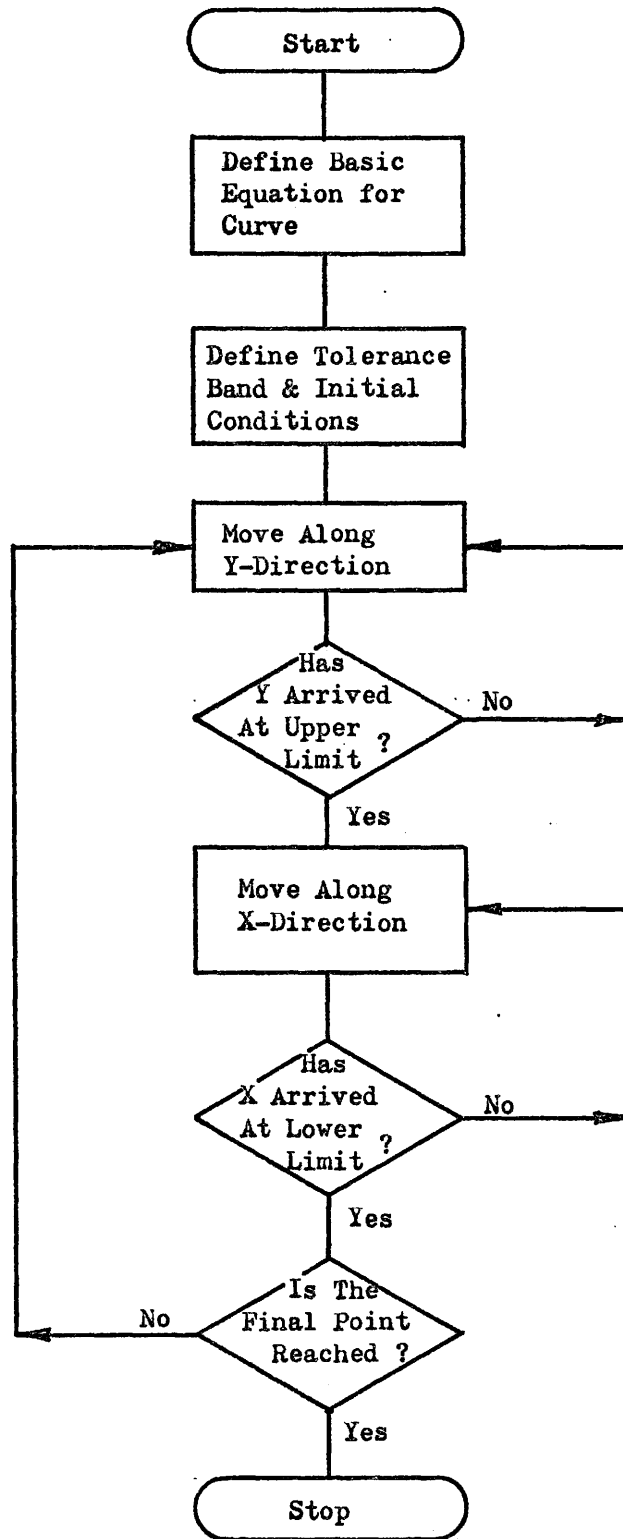


FIGURE 3.5

Since the resolution of the Machine is .001", the smallest tolerance in either X or Y direction is .001". However, the number of steps required to produce even a short curve with a .001" tolerance would be great. A reasonable tolerance band is $\pm .005$ " about the mean path so that the number of blocks of information can be kept to a reasonable length.

The actual applications of the Step-wise Linear Technique to the Moog Hydra-Point for contour cutting will be dealt with in depth following a Chapter describing the usage of a small, versatile computer, the PDP-8/L and a chapter dealing with the programming of the PDP-8/L for the translation of ASCII into EIA codes which is a necessary step before the Moog Hydra-Point can accept any input information.

CHAPTER 4SIMPLIFIED DESCRIPTION OF THE PDP-8/L COMPUTER SYSTEM

The PDP-8/L computer is different from the larger computers, such as the IBM 7040 or the CDC 6400, in several aspects. First, the PDP-8/L is fairly small in size with only about 4000 words of storing space. Second, the PDP-8/L is designed to be accessible logically from external manual switches as well as coded information. It is some kind of a poorman's portable computer rather than the general public's accounting machine. Third, the method of information input and output is very simple through the keyboard or a low or high speed paper tape system. Fourth, any kind of original program can be written (within the capacity of the computer) to execute another program. In contrast, the larger machines require the use of their preprogrammed languages.

This Chapter attempts to describe some basic structures of the PDP-8/L computer; it does not intend to convey a complete picture of the design and the usage of the computer. Rather, it tries to point out the highlights of the computer's functions and its fundamental building blocks; to outline its basic coded language and the importance of the complete access to the control and memory logic.

The physical PDP-8/L comprises the following basic units: Input, Control, Storage, Arithmetic, and Output. (See Block Diagram in Figure 4.1).

A) Input Unit

The Input Unit is generally a keyboard with a tape reader. Additional devices include high speed tape reader, magnetic tape reader, plugboard, etc. These input devices serve the functions of transmitting the input data, either prepared in tapes or manually typed, into the computer's Control unit.

B) Control Unit

The Control Unit is a preprogrammed piece of hardware which accepts the incoming data from the Input Devices, interprets them and carries out the proper executions according to the input instructions. The Unit is in form of a solid-state switching, counting, channelling mechanism which will respond to the proper input codes.

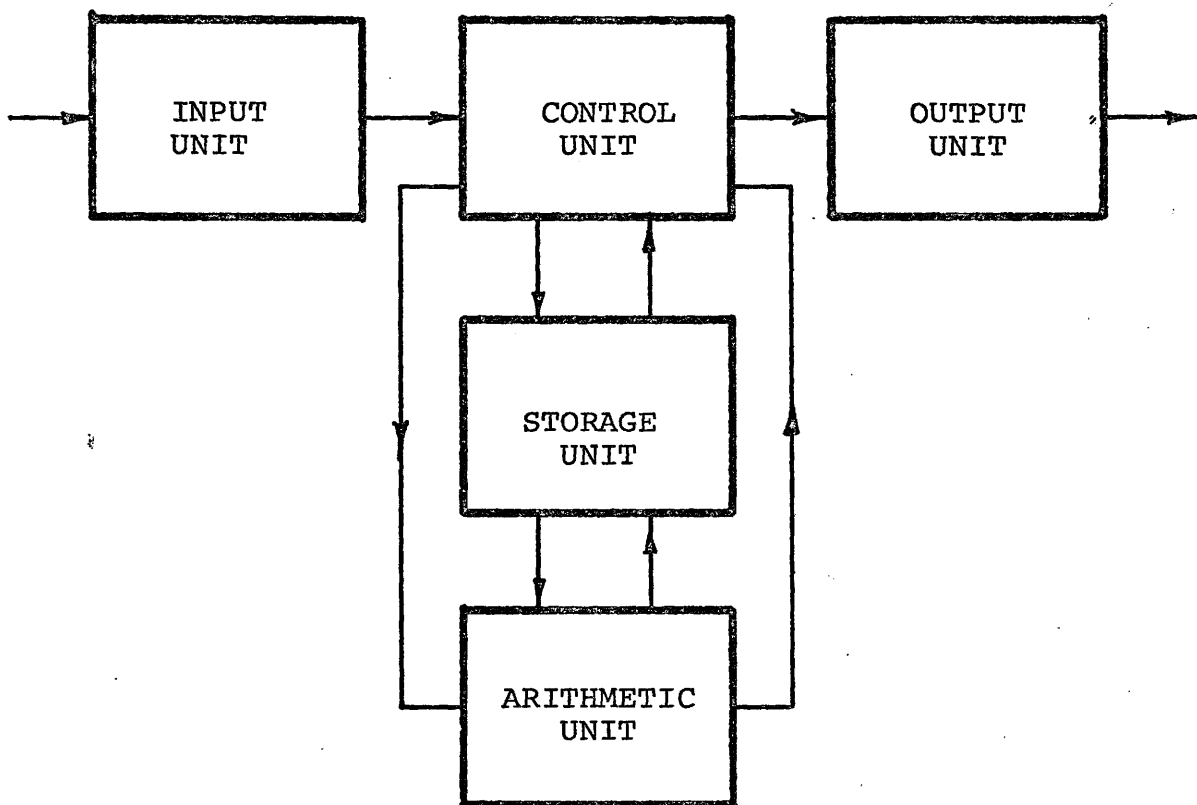


FIGURE 4.1

C) Storage Unit

The Storage Unit is a bank of memory core which is capable of holding or storing data directed by the Control Unit to a specific location in the memory bank. The PDP-8/L has 4096 physical "words" of memory spaces which are each 12 bits (binary digits) in length. Each word location has a unique address which is directly accessible by the Control Unit for either data depositing or retrieving purposes.

The memory bank may store control command codes or constant data or instruction to perform certain tasks until they are needed.

D) Arithmetic Unit

The Arithmetic Unit is a device which performs the actual computation and arithmetic operations. The Unit is referred to as an Accumulator and is a solid-state electronic logic device which carries out its functions by mainly performing a counting (addition or subtraction) of input pulses and storing them as partial sums during operation. All arithmetic operations are performed in the Accumulator.

E) Output Unit

The Output Unit is generally a keyboard typewriter (or teleprinter) with a tape punch which will respond to the output signal from the Control Unit. When the code for printing a character is received by the teleprinter, it identifies the code and causes the proper key to activate and at the same time the code in binary value will be punched in the paper tape to produce a permanent record of the output data. All the output data can be punched in the tape but not typed out. Information like Carriage Return, Line Feed, Bell, Leader/Trailer signals will cause the proper operation of the output mechanism. However, these codes when received, will be punched in the tape.

Other output devices include the high speed tape punch, magnetic tape, oscilloscope or even an analog voltage device whose output voltage can be used directly to regulate an external control mechanism. These output devices are not needed for our purpose here.

Figure 4.2 shows a picture of the PDP-8/L computer with an Input-Output Teletype Unit.

THE PROGRAMMING FUNDAMENTALS OF THE PDP-8/L COMPUTER

The Control Unit is preprogrammed in such a way that it will only recognize numbers, or more precisely binary numbers, i.e. the on-off state of binary digits. The maximum storage capacity of the PDP-8/L is 4096 words or $111\ 111\ 111\ 111_2$ in binary and 7777_8 in octal locations. Each word is 12 bits long or 12 individual on-off positions in series. Theoretically therefore, there are 4096 ways of combining the binary numbers to instruct the Control Unit. However, the number used is considerably less.

The programming of the PDP-8/L requires the knowledge of the proper coded instructions each in 12 bits of binary or 4 bits octal code, the latter is more convenient and is generally used.

The standard set of instructions consists of 8 basic commands. Six of these are called Memory Reference Instructions (MRI) each of which is made up of an octal instruction code and 3 octal Operand codes.



FIGURE 4.2

These Operand codes specify the proper address of the location to which the instruction refers. All the core memory locations are only accessible by using the proper MRI codes which give the desired address in which the content of data is stored.

The six MRIs, with their mnemonic and octal equivalents, are listed below:

<u>INSTRUCTION</u>	<u>MNEMONIC</u>	<u>OCTAL VALUE</u>
Logical AND	AND	0 ___
Two's Complement Add	TAD	1 ___
Increment and Skip if Zero	ISZ	2 ___
Deposit and Clear Accumulator	DCA	3 ___
Jump to Subroutine	JMS	4 ___
Jump	JMP	5 ___

For example, an instruction which begins with an octal number 3 and followed by three more octal numbers like 277_8 or 3277_8 instructs the computer to take whatever is present in the Accumulator (of course the value is in 4 octal digits) and deposit it in the address specified by the number 277_8 or the 77th line of the current page of memory (each page has 200_8 lines), and then clear or

destroy the content of the Accumulator. Without going into great details, it suffices here to mention that the addressing of the current page or from one page to another is done by using different values of the second octal digit. In this way it is possible to address all 4096 locations using only the last 3 octal digits.

One of the two remaining instructions consists of two sub-groups of Microinstructions which allow the programmer to manipulate and/or test the data that have been computed through the required operations. These instructions do not have any location reference and are simple commands for executing certain operations. All these instructions begin with the octal value 7.

Quite a large number of Microinstructions are possible because of the availability of all the digits. Some important Microinstructions are listed below:

<u>INSTRUCTION</u>	<u>MNEMONIC</u>	<u>OCTAL VALUE</u>
Clear the Accumulator	CLA	7200
Complement the Accumulator	CMA	7040
Rotate Acc. and Link right Once	RAR	7010
Increment the Acc.	IAC	7001
Skip on Zero Acc.	SZA	7440
Halt the Program	HLT	7402

The purpose of presenting the Mnemonic as well as the Octal equivalent of the instructions will be clear when we consider the use of the Mnemonic Codes alone in writing programs in the later Chapters.

The last group of instructions which begin with code 6_ _ _ 8, is mainly for directing input-output operations. All information transmission between the input-computer and computer-output is done through the use of code 6. Some important commands are listed below:

<u>INSTRUCTION</u>	<u>MNEMONIC</u>	<u>OCTAL VALUE</u>
Skip if Keyboard/Reader flag 1	KSF	6031
Clear ACC, read Keyboard buffer and clear Keyboard Flag	KRB	6036
Skip if Printer/Punch Flag = 1	TSF	6041
Load Printer/Punch buffer, Print and Clear Flag	TLS	6046

A complete list of instructions can be found in Reference 9.

With the above basic knowledge and a deeper understanding of the hardware operating mechanism, a programmer will be able to communicate with the computer and utilize its high speed capability to perform the necessary calculation, manipulation and testing of the

desired information fed into the computer and to retrieve the results by any convenient output means.

The main purpose for acquiring the knowledge of the internal structure of the PDP-8/L computer system is to provide a powerful programming ability for the programmer who will not be restricted to the specific confines of a preprogrammed language usage such as Fortran, for computation or solving of his particular problems. The basic understanding of the PDP-8/L allows the programmer to play with the computer logic itself, even to manipulate the state of the binary bits in the memory banks as well as in the arithmetic binary counter unit (or the Accumulator). In so doing, the programmer has complete access to the internal switching operations of the computer and has the freedom to represent his output in any combination of the 12 bits of binary digits through the versatile input and output instructions.

This brief description of the PDP-8/L usage leads to the application of the computer to convert the ASCII Standards, represented by certain binary codings, to the EIA Standards, represented by different binary codings. It is to be treated in the next Chapter.

CHAPTER 5AN ORIGINAL PROGRAM FOR THE TRANSLATION OF ASCII INTO
EIA USING THE PDP-8/L COMPUTER

Almost all N/C machines to-date use a standard code devised by the EIA (Electronic Industries Association) as shown in Figure 5.1. The coding is generally done by punching holes in a paper tape of 1" wide as described in Chapter 2. The machines receive their information by "feeling" the presence or absence of the holes in the tape.

The Moog Hydra-Point employs only the numeric portion of the code since all its input commands and dimension specifications are represented by decimal numbers. The Machine is designed to accept only fixed-block information, i.e. the codes for each control function must always be present and must be located in the same position of each block which has a fixed length of 19 decimal numbers plus an End-of-Block code. This type of coding is generally done by the Binary-Coded-Decimal Method where each decimal number is represented by its binary equivalent which occupies the first four position of the binary digits.

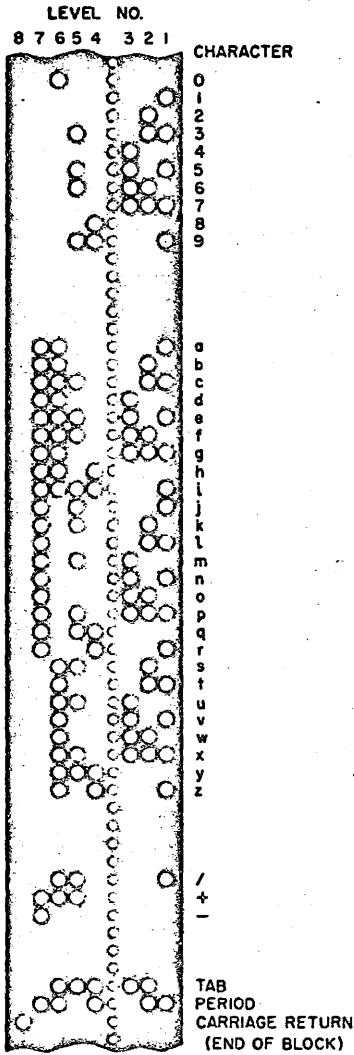


FIGURE 5.1

In contrast to the EIA codes used in N/C machines, the Teletype and other general purpose computer equipment utilize another standard code called the ASCII (U.S.A. Standard Code for Information Interchange) which is also basically binary but with additional digits.

From Chapter 3 it was seen that in order to utilize the Moog Hydra-Point for contour cutting, many intermediate points along and within a given error band must be specified. To obtain these points, a general purpose computer, with Fortran compiler facility, will be most useful. The use of the Fortran language provides the versatility of defining the mathematical model in simple expressions and the iteration method provides the required accuracy. Also, the output format can be designed to conform with the fixed block machine format.

A problem exists here when the program for the N/C machine is prepared with the aid of a general purpose computer. The desired output information will be in ASCII while the N/C machine accepts only EIA codes. Therefore, it was felt necessary to provide a means of automatically translating the ASCII into EIA codes.

As part of this thesis, the exercise of ASCII to EIA Conversion was carried out ab initio by the author. The logic and steps of conversion with the aid of the PDP-8/L computer are presented in the following sections.

Three basic programs were written, which were:

- (1) A simple conversion of ASCII into EIA for all numerals plus the Carriage Return code. All other characters will also be converted, but with meaningless output codes.
- (2) Conversion of ASCII into EIA plus the necessary parity code for the use of the Flexowriter printout. Other extraneous characters still present.
- (3) Conversion of only the numerals plus the Carriage Return and Space with the proper parity code added and ignoring all extraneous characters (without punching or advancing of tape). This program is applicable to both parity and non-parity ASCII.

Program (1) was a straight-forward masking operation of the incoming ASCII codes which produced the pure binary equivalent of the EIA (for numerals only). However, in so doing, other characters like Rubout, Linefeed, Space etc. would also be converted into some new characters which were not necessarily the EIA equivalents. This program was merely an exercise to deposit and retrieve the intended information and to carry out certain logical operations.

Program (2) was more refined in the sequence of conversion which can be seen from the Flow Chart in Figure 5.2 Briefly, the program calls for the acceptance of a character from the keyboard or tape, performs a masking operation, seeks out and identifies its value and gives it a new code in the EIA Standards, and punches the new character in the output tape. Other non-identifiable characters will pass through in the masked or reduced form and punched out in a shortened code not necessarily the EIA equivalent. The conversion process operates continuously until there is no more incoming characters. This program presents a problem that will require the converted tape to run through the Flexowriter once again to blank out the extraneous characters in order to produce the properly spaced blocks of information.

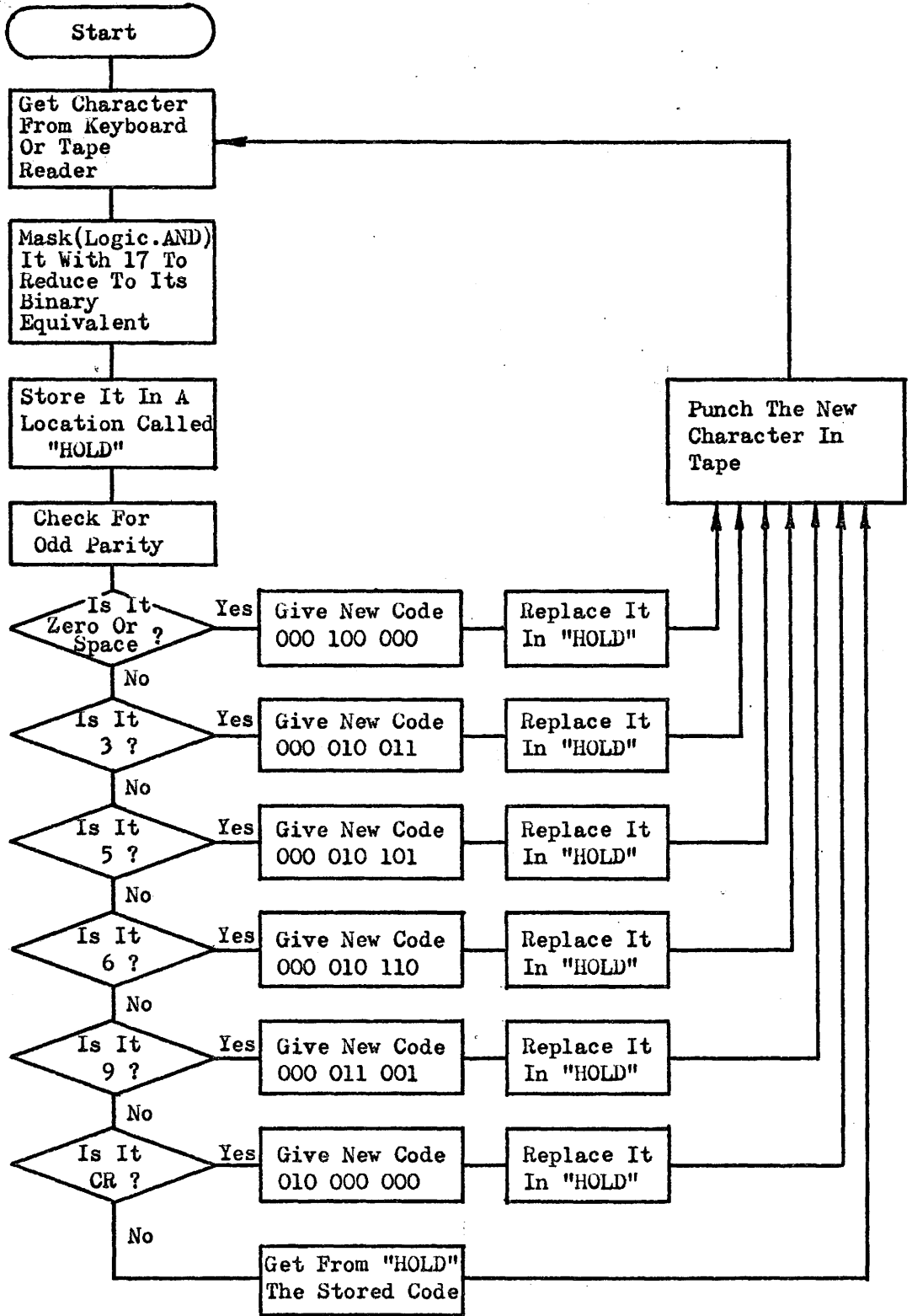


FIGURE 5.2

This requirement is almost mandatory in most available library Programs for the conversion of ASCII into EIA.

It was deemed necessary that a third program should be written to accommodate the extraneous characters and produce a tape directly usable by the Machine without the need of passing through the Flexowriter again to get a clean tape. This brought to the consideration of Program (3) whose Flow Chart is shown in Figure 5.3, and it was chosen as the final Conversion Program, detailed description of which will be given in the following sections.

Referring to the Flow Chart for Program (3) in Figure 5.3, this Program will basically accept one ASCII character from the Keyboard or Tape Reader and simultaneously (in microseconds) punch a new EIA code, for the same character, in the Output Tape. The cycle will repeat itself automatically as long as there are input data present.

The Program is divided into sections which comprise Subroutines:

1. A Subroutine for Accepting a character from the Input Device.

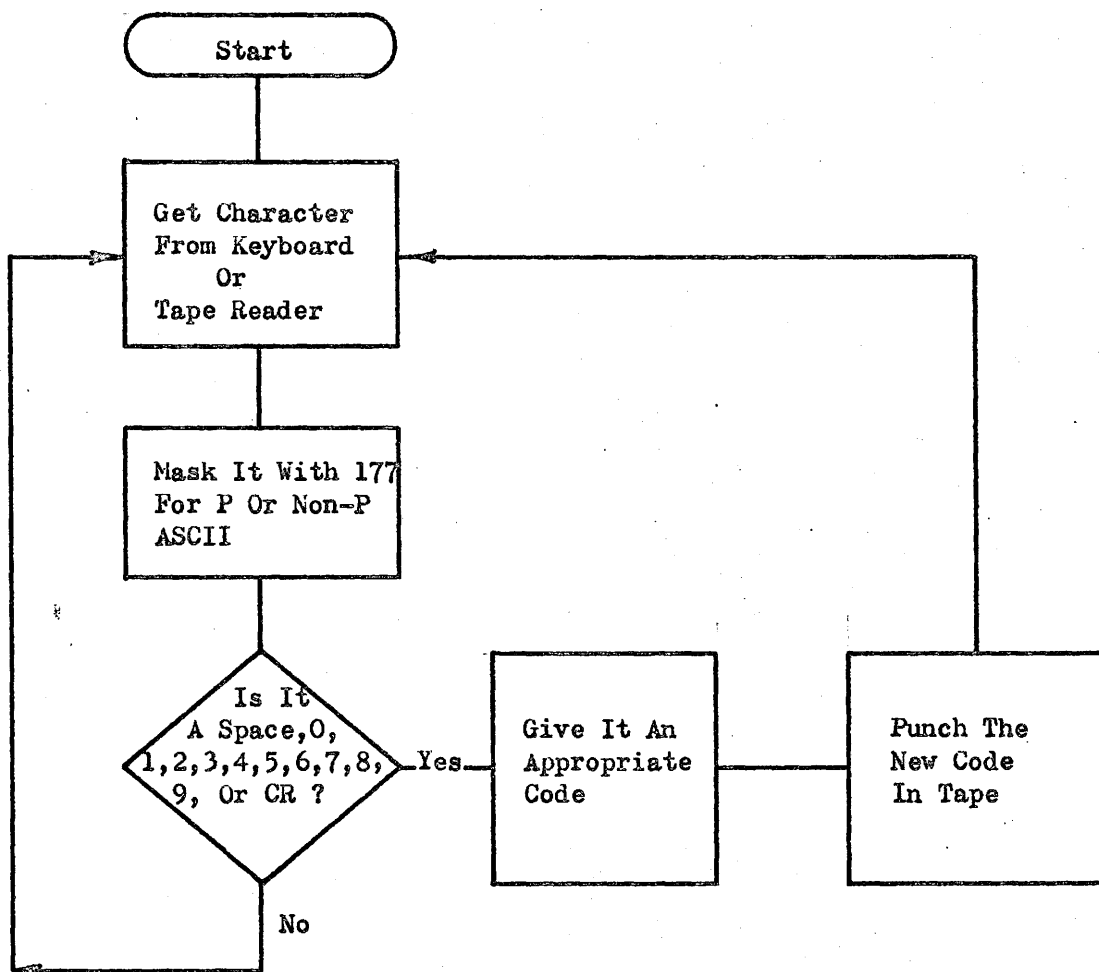


FIGURE 5.3

2. A Subroutine for Checking the incoming character and assigning it a new code.
3. A Subroutine for Punching the new character through the Output Device.

The Accepting and Punching Subroutines 1. and 3. contain standard PDP-8/L instructions (e.g. checking for Flags and activating the punching mechanism etc.) which will be shown in the actual conversion Program.

The Checking Subroutine 2. comprises the arithmetic operation of checking the incoming character in the following fashion.

When an incoming ASCII character is received, it will be masked by a logical AND operation with a constant 177_8 which effectively allows only the first seven binary bits to get through. This step is necessary to account for both the parity and non-parity ASCII.

After the masking operation, all the numerals which have the original values of 260_8 (0), 261_8 (1), 262_8 (2),, 270_8 (8), 271_8 (9) will become 60_8 (0), 71_8 (9) respectively (Brackets contain decimal equivalents). The Carriage Return and Space, which

have original values of 215_8 and 240_8 will be reduced to 15_8 and 40_8 respectively.

A further reduction step, which may not be necessary but produces a neater-looking operation, is to add the constant value of -60_8 to all characters. This essentially reduces the numerals to 0_8 , 1_8 , 2_8 , ..., 7_8 , 10_8 , 11_8 and the Carriage Return to -43_8 and Space to -20_8 .

A list of constants, which are the negative values of the reduced numerals plus the Carriage Return and Space, are previously specified by the programmer and input into the computer to be stored in specific locations. These values are -1_8 , -2_8 , ..., -7_8 , -10_8 , -11_8 , and 43_8 and 20_8 . All the numbers are in octal value when shown with a base 8.

The stored negative constants will serve the purpose of recognizing the unknown incoming character so that a correct new code can be assigned to it.

For example, if the incoming character is a 3, which has a value of 263_8 in ASCII, after masking and reduction it will become 3_8 . A screening operation will follow which allows the stored constants to be

added to 3_8 one by one. Upon the summation with -3_8 a zero value will result which will be detected by the logic set up and the incoming character is immediately recognized. A new code with the value of 23_8 , uniquely defined for 3 in the EIA Standards, will be given to it and the computer is then instructed to punch the new code in the output tape. Note that the keyboard may not type the new character because the typing mechanism may not recognize the EIA code to cause any action.

The EIA values for the ASCII equivalents are also constants previously stored in the computer.

All incoming characters are screened through by the negative constants and the appropriate new values given to the proper characters. If any of the incoming character do not fall into the group listed above (all numerals plus the Carriage Return and Space), which will normally occur in tapes prepared from a Fortran program with the presence of extraneous characters like Linefeed, Rubout; no zero value in the screening and reduction process will result. The computer is instructed to ignore them and go to fetch another character until a zero resultant occurs. This skipping process should not cause the paper tape to advance a blank space because fixed length of 20 digits or 20 consecutive rows

of holes for each block must be kept absolutely rigid throughout the complete machine program. The Machine will not respond to improper instructions due to unequal block length or dislocated coded information.

With the knowledge of the basic programming principles of the PDP-8/L computer as described in Chapter 4, it is not necessary to remember the exact numeric codes for the specific instructions. Rather, it is the mnemonic codes which play a more important role in the programming. The mnemonic names are mainly abbreviations of the desired instructions. A programmer can write his programs entirely in mnemonic (or symbolic) codes and make use of a specially prepared assembler Program called PAL (Program Assembly Language) which is a powerful translator which converts the acceptable symbols into their equivalent numeric codes and will produce a numeric binary object tape. This binary tape represents the actual program if one is patient enough to look up every number for his instructions.

The PAL not only will convert symbols into binary numbers but will diagnose and trouble-shoot the original symbolic program so that the programmer can make the necessary corrections.

The Program for ASCII to EIA conversion as outlined above and depicted in Figure 5.3, was written and a printout was obtained using the PAL as shown in Appendix B.

The first column of numbers are the address of the locations of the memory core in which the data are intended to be kept. The second column of numbers represents the content of the memory core at the particular address. The contents can be instructions, commands, or constant values. All numbers are in octal system.

Figure 5.4 shows a typical block of information before and after translation.

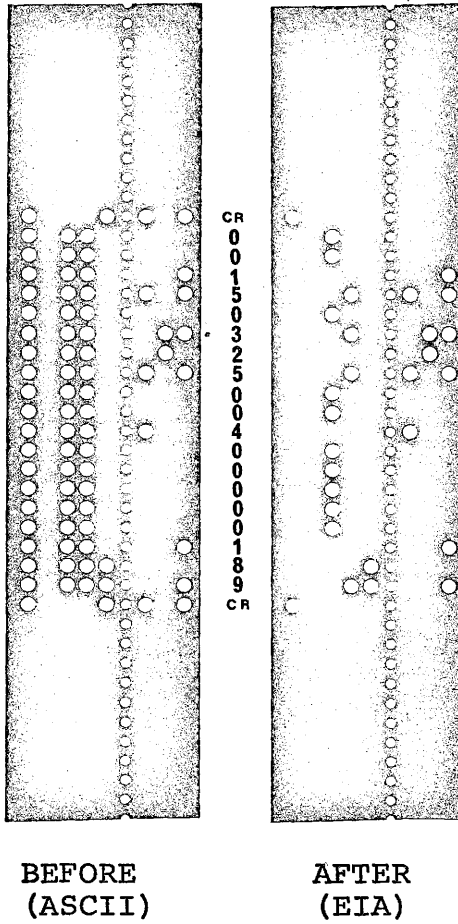


FIGURE 5.4

CHAPTER 6PROGRAMS FOR DERIVATION OF COORDINATES OF SIMPLE CURVES

The Moog Hydra-Point is a point-to-point machine which is basically used for positioning cutting application. The attempt to use it for continuous contour cutting requires the specification of intermediate coordinates along the desired path within the allowable error bands. To obtain these intermediate points, it is best to employ the service of a computer since in some instances the number of points involved can be quite substantial.

The Fortran language's method of iteration is particularly suitable for interpolating the intermediate points of our Step-wise Linear curves. Almost any curve which can be defined mathematically can be interpolated using Fortran.

The basic program involves the definition of the desired curve, either from experimental data or from design, with a suitable mathematical model. Next an error band is specified as an allowable tolerance which essentially defines an upper and a lower boundary to the desired curve.

It was described in Chapter 3 that the continuous contour will actually be composed of many right-angled steps within the ascribed tolerance band. Therefore, the program will also be called to derive small alternating increments in the X and Y directions within the error band. The points which lie on the boundaries of the error band will be taken as the intermediate solutions along the desired curve.

The remaining step is to output the solutions in the form of a punched tape. Unfortunately the main computer CDC 6400 at McMaster University is not equipped with a tape punching output device since almost all computed results are presented in printed sheet or stored on magnetic tapes. Very few attempts have been made to utilize the output results directly for any physical application. Anthropomorphically, it may be said to write a lot, but not actually to do anything.

An alternative Time Sharing terminal in the Mechanical Engineering Department, has a Teletype input-output console, which can produce an output tape in ASCII code and which can be tied in with the General Electric's central computer in Toronto. The process of input and output is, however, rather slow.

TEST PROGRAMSCIRCLE

As a first exercise for the machine code Conversion Program as well as for the Moog Hydra-Point, a test program for machining a circle was selected. The simple expression $R = \sqrt{x^2 + y^2}$, where R is the Radius of the circle minus half the Radius of the cutting tool, and x and y are the corresponding coordinates, represents the equation of a circle.

The method and steps for obtaining the circle coordinates are illustrated in the Flow Chart as shown in Figure 6.1. A printout of the actual program can be seen in Appendix C. Note that the output Format calls for printing, without extra spaces, the Sequence No. in 3 digits; the Preparatory Function in 1 digit; the X coordinate in 5 digits; the Y coordinate in 5 digits; the Reserve plus the Tool function in 3 digits; and the Miscellaneous Function in 2 digits. It will produce a string of 19 unbroken integers in series. Spaces within the string of integers will be treated as zeros upon performing the Translation operation. The 20th digit will be translated automatically when the Carriage Return code is encountered.

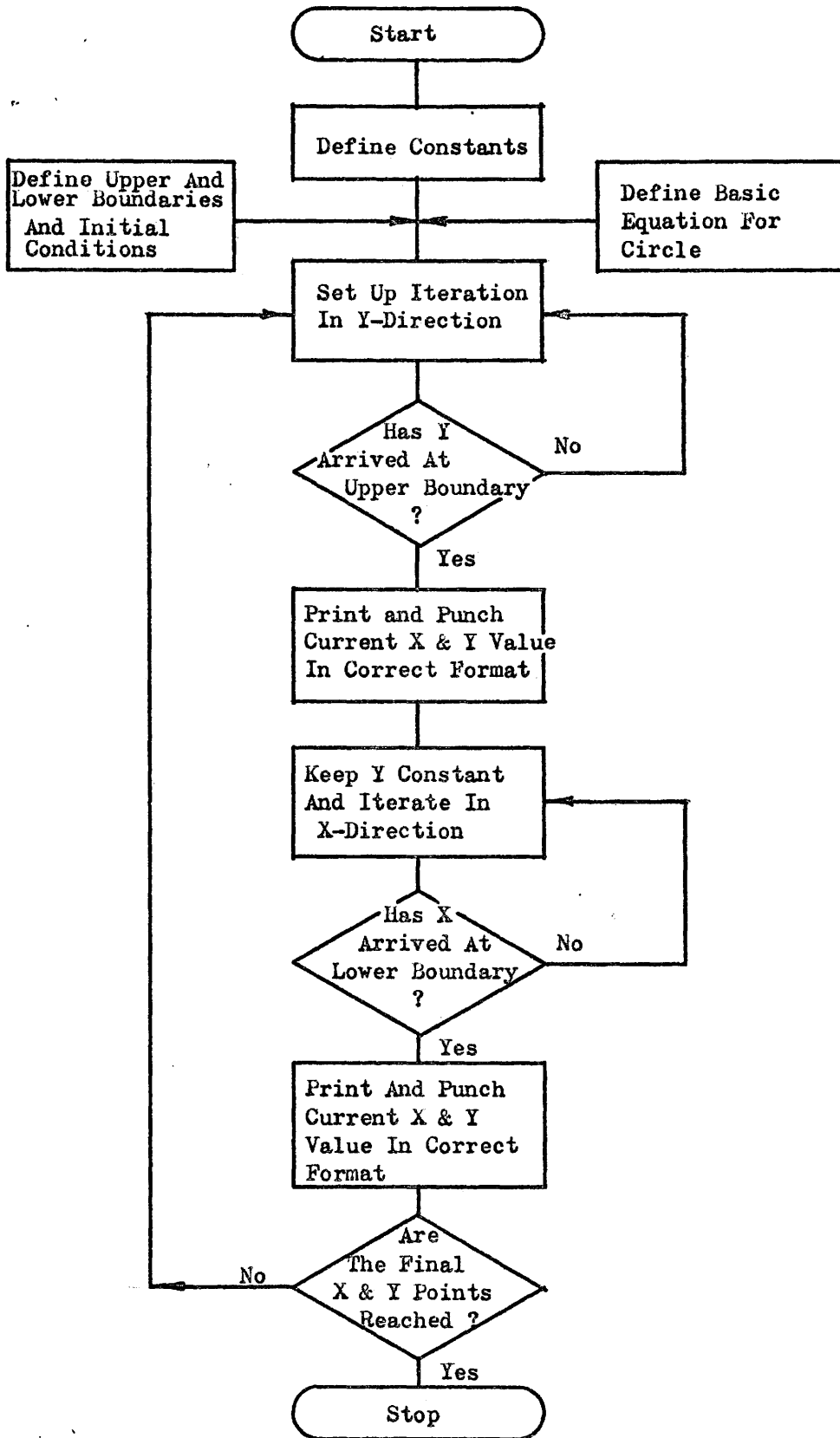


FIGURE 6.1

THE ELLIPSE

The second program is an extension of the first program for machining circular paths to include elliptical contours. This program serves two purposes:

1. It deals with a more practical requirement (unlike the circle which can always be bored, drilled or trepanned out).
2. It has been applied directly to the machining of a set of elliptical dies which were made for studying the bulging effects of different sheet metals (in another area of research of Production Engineering by other graduate students).

Basically, the ellipse is defined mathematically by its major and minor semi-axes with relationship to the x and y coordinates. The equation

$$x^2/a^2 + y^2/b^2 = 1$$

defines the ellipse where a is the semi-major axis and b is the semi-minor axis, x and y are the variables.

The procedure of obtaining the intermediate coordinates for the ellipse is similar to that of the circle. The error band is defined by solving the equations with a new $A = a \pm E$, and a new $B = b \pm E$ where E is the allowable tolerance. The solutions of these two equations will essentially give the coordinates of two ellipses, separated by a value of E , inside and outside the original ellipse.

With the defined inner and outer tolerance limits, the flow diagram for determining the intermediate step-wise coordinates will almost be identical to that of the circle when the appropriate constants and equations are used.

The actual program for the ellipse can be seen from the printout shown in Appendix D. Note that the physical dimensions and tolerance limits of the ellipse can be easily altered by simply changing the constant values such as the semi major and minor axes and the tolerance level in the beginning of the Program.

A THIRD DEGREE POLYNOMIAL

The programming of well-established geometrical contours such as the circles and ellipses is relatively simple since many of their intermediate points can readily be determined even by inspection. In practice, curves of unknown geometry are more likely to be encountered. This will necessitate the expansion of particular programs into more general applications.

To illustrate the programming versatility, a polynomial of third degree is chosen and shown as below:

$$y = Ax(x-B)(x-C)$$

where A, B, and C are constants.

When plotted, this polynomial will produce a curve which will pass through the X-axis three times when x assumes the value of 0, B or C.

This simple equation will serve, as a start, to illustrate methods for defining curves of more complex nature. When experimental or design curves,

whose original curvatures are not defined mathematically, are given; they can always be reduced to some mathematical functions by the Least Square Method or other classical interpolation techniques.

The determination of the X-Y steps within the error band for a general curve is much more involved than for simple geometrical curves. Once the equation for the general curve is defined, it is also required that the maxima and minima, within the range of interest, be known in order to make the proper iteration direction change about the turning points of the curve.

To accomplish this, the equation is first solved on the computer at discrete intervals and the maxima and minima are obtained. Next, the Step-wise Linear Technique is applied, with the proper error band defined, to produce the desired intermediate coordinates. These two programs can be combined effectively to interpolate polynomials of higher degrees.

The polynomial equation given above is assumed to be the curve for the cutter to follow. It actually defines the centre path of the cutter. The part surface,

however, will differ slightly mathematically depending on the size of the cutter used. This method is acceptable and is applied when the diameter of the cutter is much smaller than the curvature of the required curve. The equations describing the cutting surface and the cutter path will not in general have the same form.

On the other hand, if the cutter size is quite comparable to the curvature, the exact cutter path must be established. The following gives an analytical method for defining the exact cutter path.

Figure 6.2 shows a required machined surface defined by

$$y = f(x)$$

with given cutter diameter = $2C$.

The first criterion is that the tangents on both the required machined curve and cutter curve are mutually orthogonal at all points and separated by a constant distance C . From this relationship, two equations can be written:

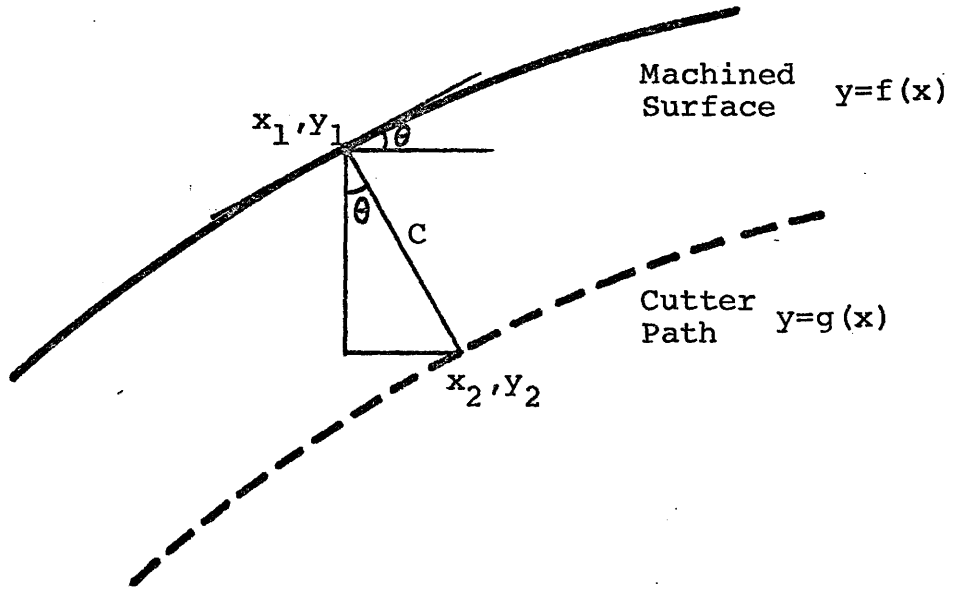


FIGURE 6.2

$$(y_2 - y_1)^2 + (x_2 - x_1)^2 = C^2 \dots\dots\dots (1)$$

$$\frac{y_2 - y_1}{C} = \cos \theta \dots\dots\dots (2)$$

since $\tan \theta = \frac{dy}{dx}$ equation (2) becomes

$$y_2 - y_1 = C \cos \tan^{-1} \frac{dy}{dx}$$

and substituting into (1), we get

$$(C \cos \tan^{-1} \frac{dy}{dx})^2 + (x_2 - x_1)^2 = C^2 \dots\dots (3)$$

Given x_1 and y_1 on curve $y = f(x)$, x_2 and y_2 can be obtained to define $y = g(x)$ for the cutter path since at both points $\frac{dy}{dx}$ are equal.

This feature could be included in a refined programming arrangement, but has not been done in this particular work.

CHAPTER 7MACHINING OF TEST PARTSTHE CIRCLE

The program for the circle was written with the following parameters:

Circle Diameter:	4"
Slot Cutter Size:	$\frac{1}{2}$ "
Cutter Center Diameter:	3.75"
Tolerance:	$\pm .005$ "

These dimensions were incorporated into the program to produce a tape for machining a circular slot $\frac{1}{2}$ " wide by 0.25" deep in an Aluminum plate.

The tape contained about 670 blocks of information for the complete circle with equal numbers of alternating X and Y milling instructions within the prescribed tolerance band.

The total time required to output the ASCII tape from the computer was 30 minutes and an additional 30 minutes were required to translate the ASCII tape into the EIA tape, using a low speed tape punch.

The actual machining time taken by the Moog for one pass was 36 minutes. It was done using the slowest Feed Rate of 1"/min. in both X and Y direction. If a higher Feed Rate is used, the actual machining time may be reduced slightly. Here the predominant factor is the time required to read in the information from the tape which normally takes two seconds for each block.

A photograph of the machined sample is shown in Figure 7.1. It can be seen from the untouched sample that due to the nature of the programming logic, the area around the 45° angles about the X-Y axes exhibit excellent finish because of closer steps of machining while at the apexes of the X-Y axes, larger machining steps are quite visible.

Theoretically, the maximum error anywhere along the path should be .010" between the trough and the peak of the steps. However, due to the use of large cutter and the compactness of steps in some areas, the actual error expected will be less than the prescribed value.

To ascertain the actual performance of the Machine as well as the Programs, the machined circle

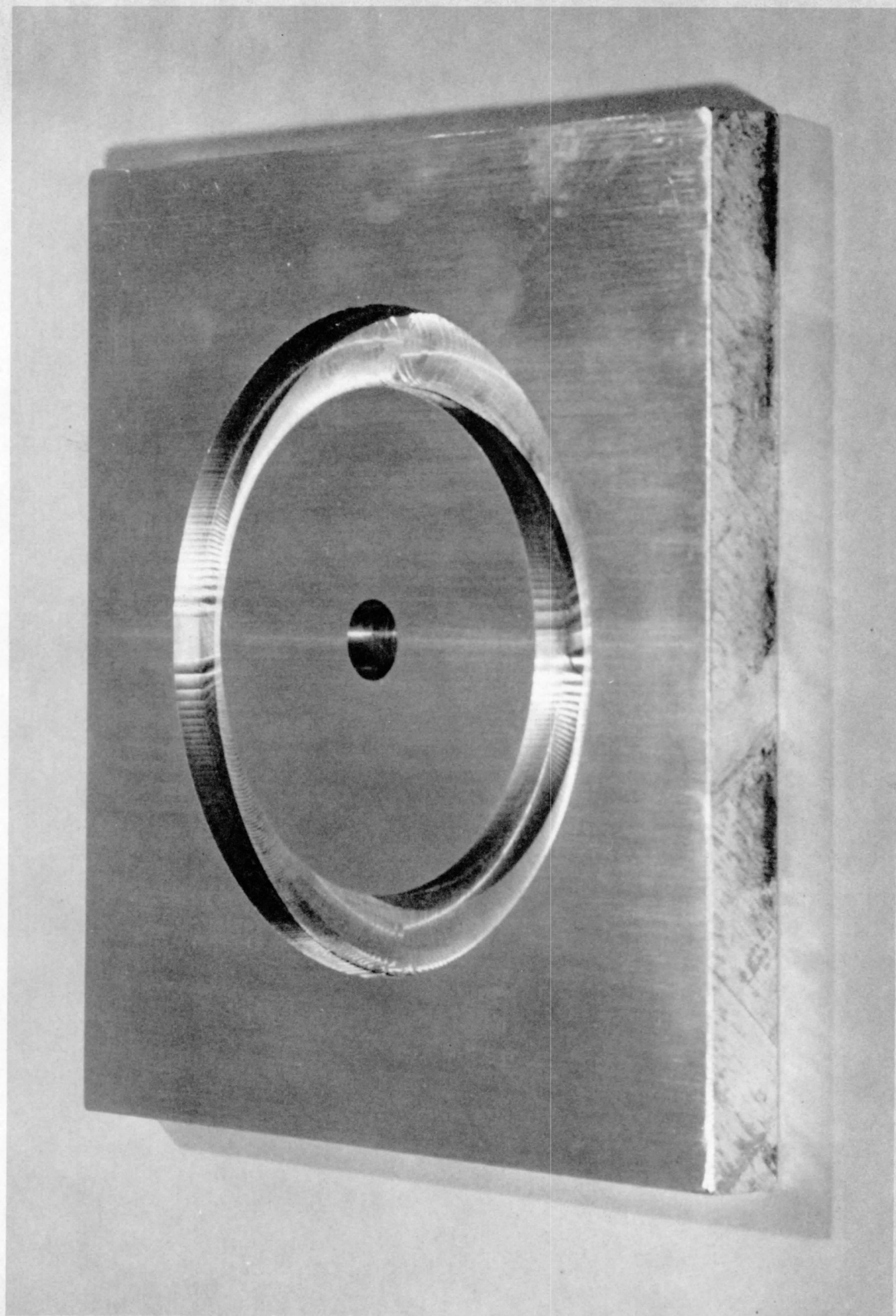


FIGURE 7.1

was subjected to a precision inspection of roundness by a Talyrond, a roundness measuring instrument.

The Talyrond is a well-known roundness testing instrument which carries a stylus about a rotating spindle. The tip of the stylus describes a true circle about the centre of the spindle. When the stylus is caused to contact the periphery of circular work, internal, external, conical or spherical, the departures from true roundness in the work will give rise to departures of the tip of the stylus from a true circular path. These departures are amplified electronically and recorded on a polar graph which is produced by electrical discharge on teledeltos paper.

Figure 7.2 shows the polar graphs of two identical circles machined by the same tape. Notice that the graph's X-Y axes are aligned with the machine X-Y axes to enable easier inspection. The sharp spikes which appear at the X-Y apexes indicate the occurrence of the maximum errors. With the radial scale set to .001"/small division, the maximum error noted is .009" from trough to peak, which verifies the expected result of the designed value. At the 45° areas, however, the surface roughness is about .001".

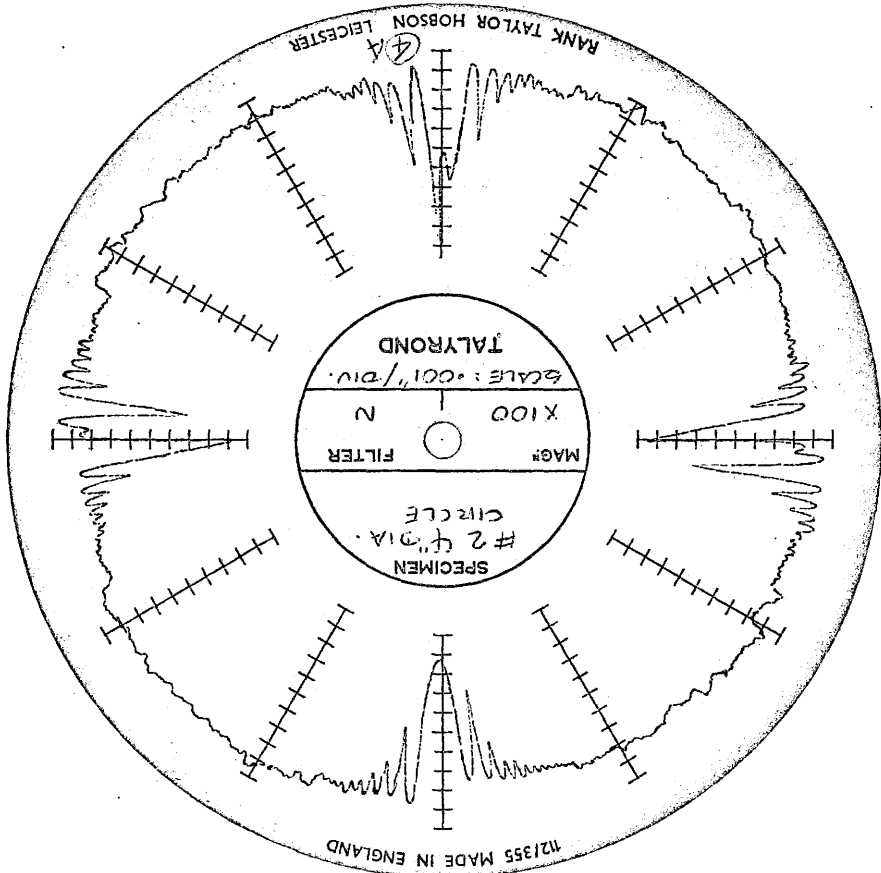
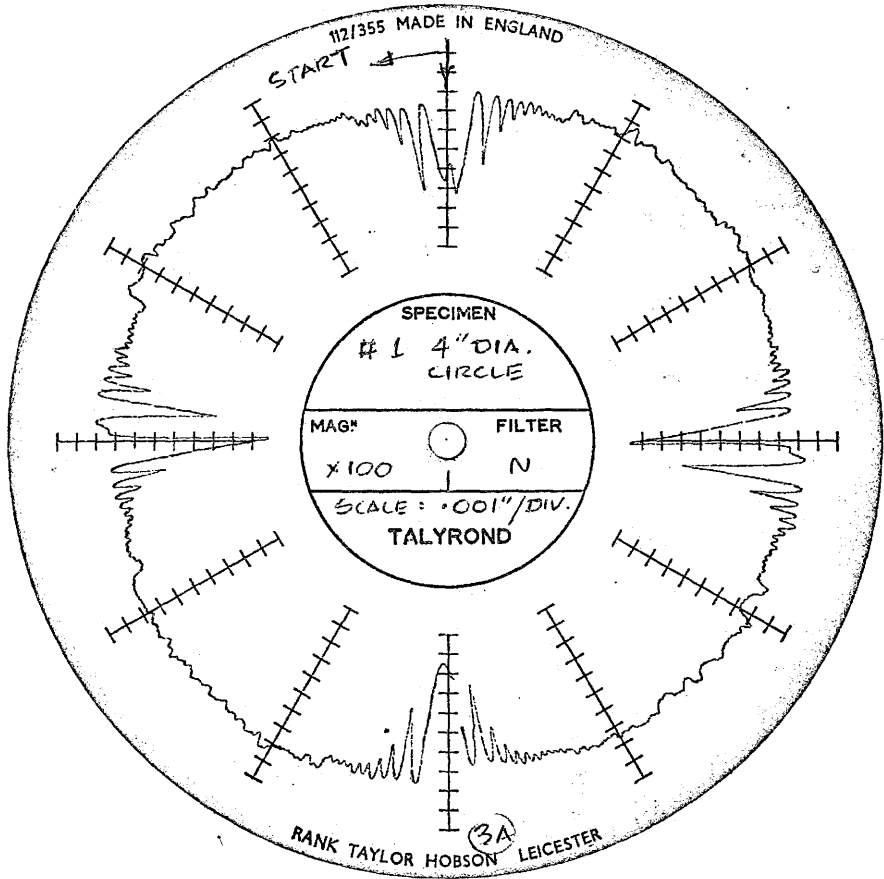


FIGURE 7.2

The shape of the spikes shown on the polar graphs should not be taken to represent the actual shape of the machined path, bearing in mind that the angular ratio of the instrument with respect to the workpiece is not amplified but retained at a ratio of 1:1. On the other hand, the radial amplitude is magnified 100 times. The centre of the circle is thus suppressed to enable plotting of different diameter circles on a standard-size polar graph. The result of this produces a peculiar effect on the shape of roughness which appears to be densely packed and highly accentuated.

Comparison of the two graphs reveals almost identical error paths in the two circles. The second circle was machined to verify the accuracy and the repeatability figures specified for the Machine.

The use of various size cutters will drastically influence the surface finish of the workpiece. As a rule, a larger cutter will result in smaller deviations from the mean path since the larger cutter area tends to smooth out sharp variations; but the maximum cutter size is limited by the size of the required slot to be

machined. In an actual part, the surface roughness can easily be reduced by conventional finishing methods.

THE ELLIPSE

To accommodate the Machine, the ellipse's constants were chosen to be:

A, the semi-major axis:	3"
B, the semi-minor axis:	2.25"
E, the tolerance level:	.005"

This gave a 6" by 4.5" ellipse with $\pm .005$ " error. See actual program printout in Appendix D.

The output tape contains about 1000 blocks of information which took 46 minutes to produce and another 45 minutes to translate. The machining time was 50 minutes per pass.

A photograph of the machined ellipse is shown in Figure 7.3. Once again the surface finish can be seen to have similar pattern as observed in the circle with larger machining steps occurring around the turning points of the X and Y axes.

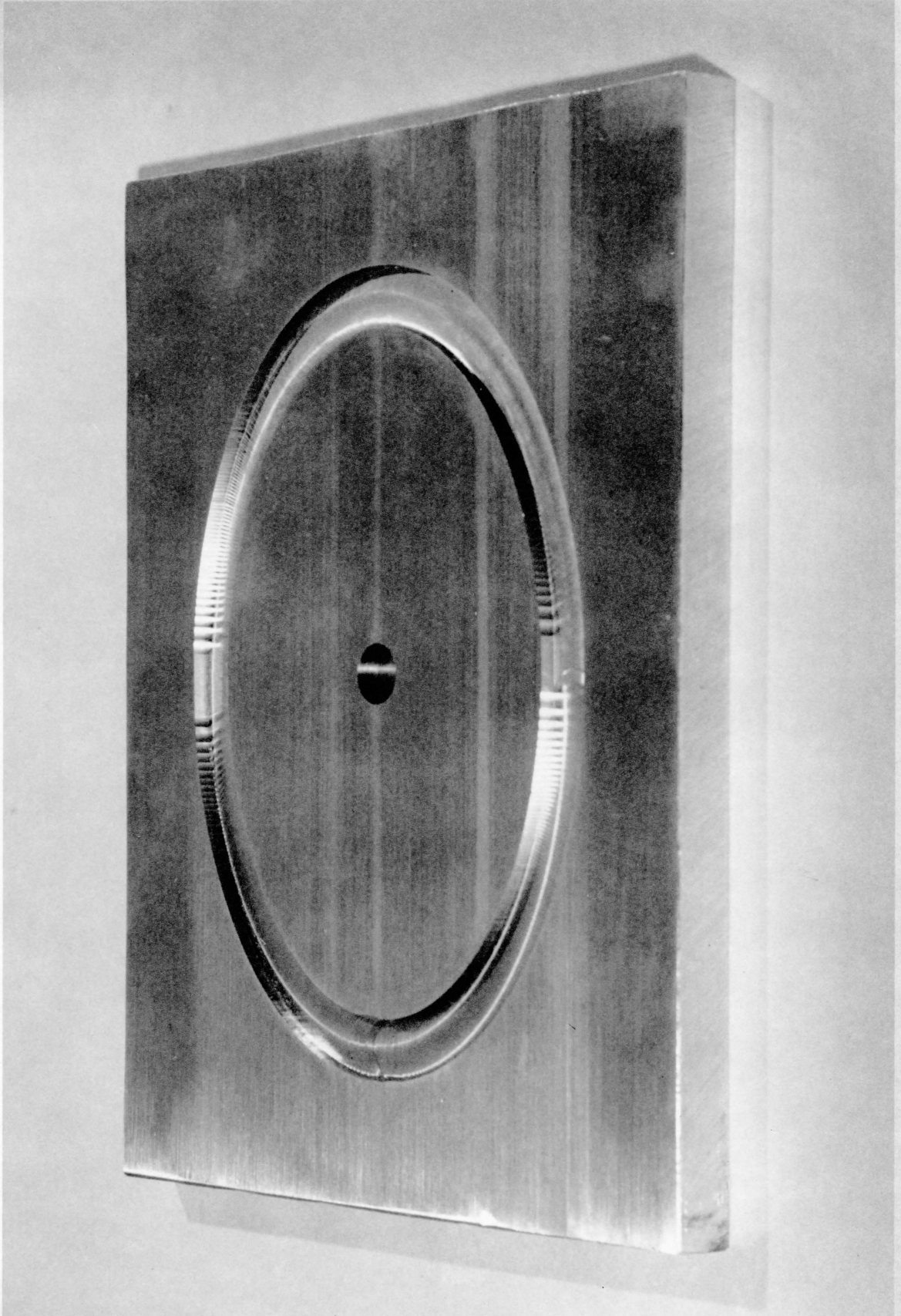


FIGURE 7.3

The ellipse program was eventually applied to fabricating a steel die for sheet metal bulging experiments. As shown in Figure 7.4, the die block measures approximately 12" x 8" x 2" with an elliptical opening of 6" x 3". Three programs were required to machine the complete configuration: one for the die opening as well as the clearance, one for the corner rounding machining, and the third one for the elliptical groove. The same basic program was used but with 3 different sets of major and minor axes.

THE THIRD DEGREE POLYNOMIAL

As an example of the utilization of the Machine to produce a more general curve mathematically defined, the equation for the polynomial given in the previous Chapter is utilized to produce a tape with the following constant values: (This equation is taken to be the centre of the cutter path)

$$A = 0.2''$$

$$B = 2.0''$$

$$C = 6.0''$$

$$\text{and error } E = 0.01''$$

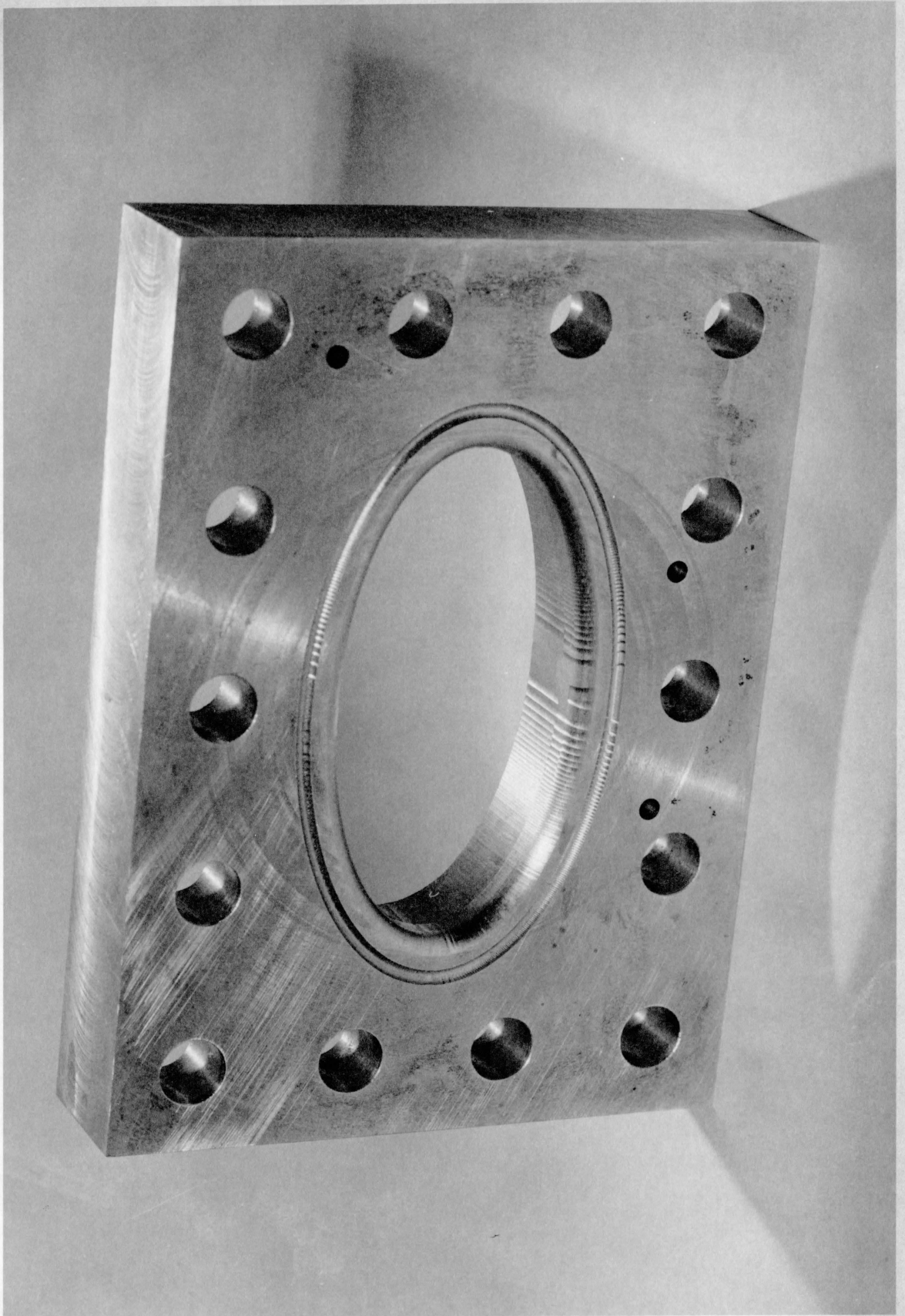


FIGURE 7.4

This will give the equation

$$Y = 0.2x (x-2.0) (x-6.0) \pm 0.01$$

with the three cross-over points selected at 0", 2" and 6" on the X-axis respectively, and a tolerance limit of ± 0.01 ".

Solving the two equations of Y will produce two identical curves about the X-axis separated by a distance of 2E. The program for deriving the intermediate coordinates was similar to that described for the circle and the ellipse.

The output of the program contains 1200 blocks which took 60 minutes to produce and another 50 minutes to translate. The machining time taken was 55 minutes.

A photograph of the machined path of the equation is shown in Figure 7.5 with the X-Y axes clearly indicated.

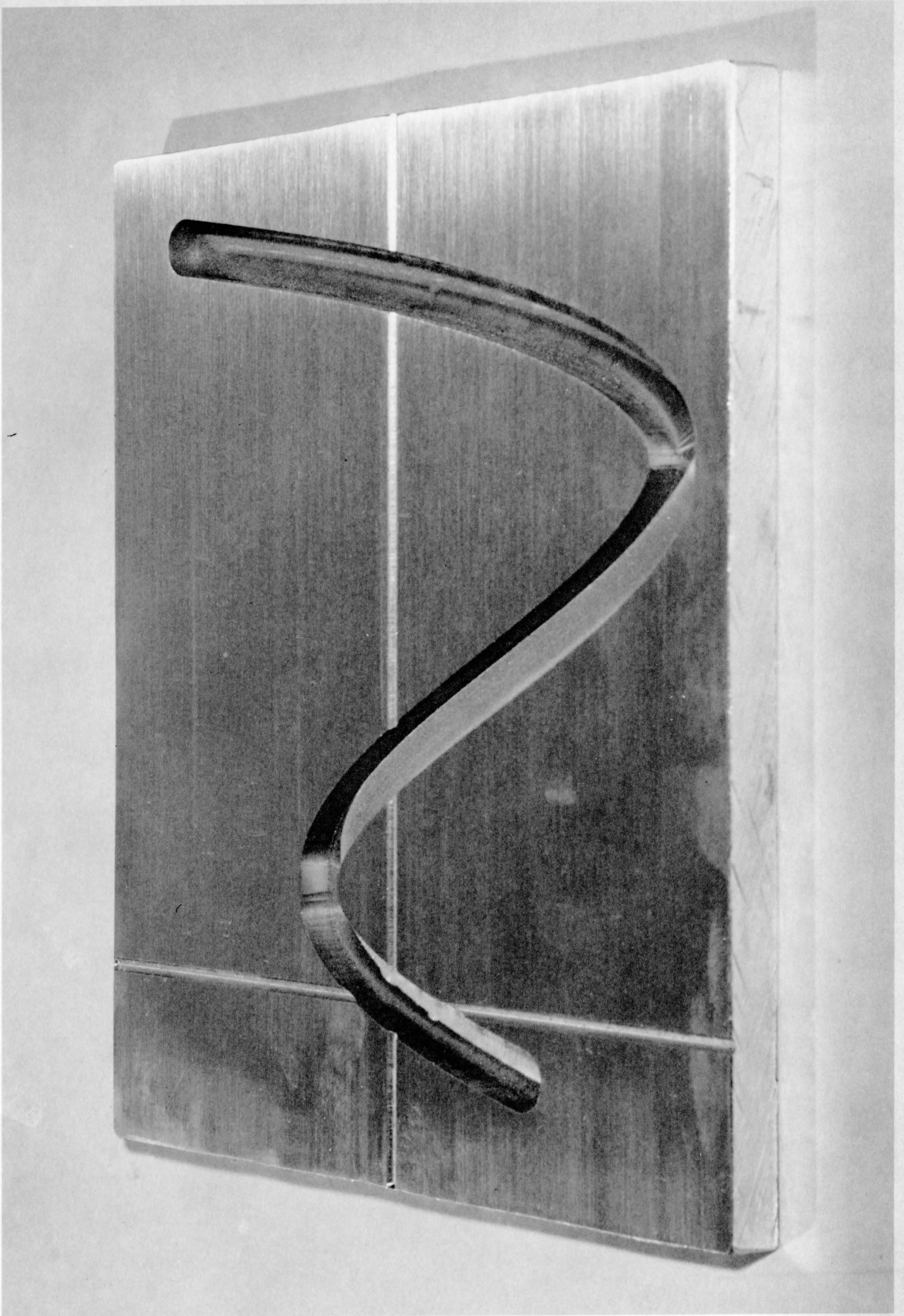


FIGURE 7.5

CHAPTER 8MACHINING OF LARGE SCALE MODELS OF A FLUID LOGIC ELEMENT

Chapters 6 and 7 have presented a detailed treatment of the extended utilization of the point-to-point Machine accompanied by tests as well as practical examples. The technique for such utilization was adequately defined. In summary, the basic procedure is first to establish the required mathematical model for the intended cutter path. Interpolation technique such as step-wise linear or piece-wise linear method can then be applied. Iteration for acquiring the intermediate cutter positions will be done by computer-aided programming.

In order to further demonstrate the machine's newly acquired ability, a subproject was proposed wherein two large scale models of a fluid logic element were to be fabricated. This subproject would serve two immediate purposes. First, it would enable the contouring technique to be applied to fabricating quite a complicated assembly of fluid flow channels. Second, the completed models would be used to demonstrate the behaviour of fluid logic control.

The fluidic element was chosen because the author has had several years experience in the design, manufacture and application of fluidics and the implementation of fluid-state technology in control systems. He also envisaged numerical control machining as a novel and practical approach to producing intricate and highly accurate models as well as templates for the manufacturing of dies or moulds used in the manufacture of fluidic devices.

The basic element selected was a fluidic bistable or flip-flop device. It was originally designed by the Fluidic Research Team at Aviation Electric Ltd. in Montreal. The author wishes to declare here that the use of similar design features of the particular device is purely for the purpose of academic feasibility study only.

Figure 8.1 shows an outline drawing of the bistable element with outstanding features labelled. Power channel, Control channels, Vents and Output channels. The actual element measures about 1" by 1" with a power nozzle of 0.020" by 0.040". The proposed model was ten times scale, to be machined in Plexiglas plates.

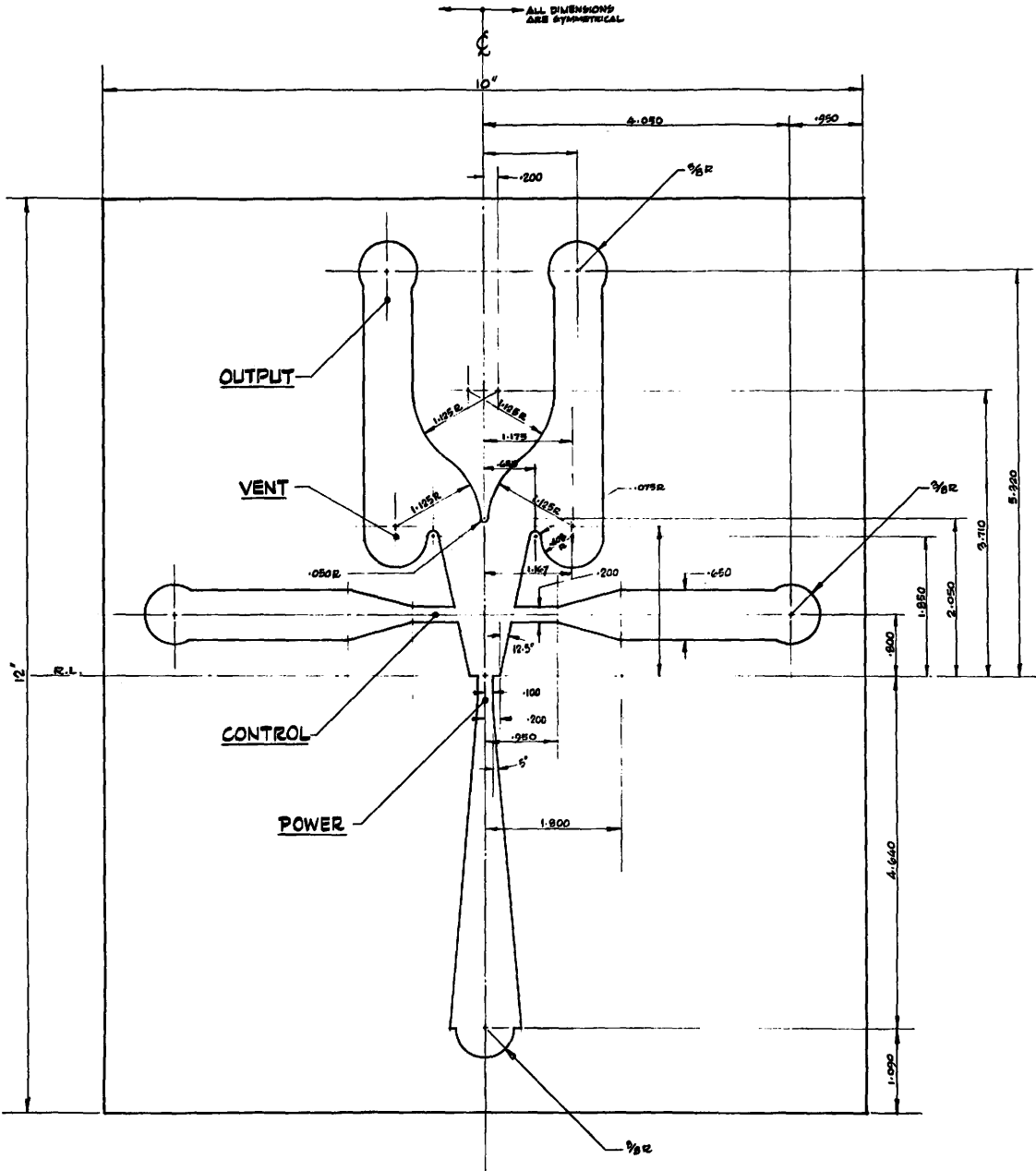


FIGURE 8.1

From the drawing, the outline was subdivided into different segments which could be easily defined mathematically. Altogether there were 4 linear slope segments and 9 circular segments which made up the main vertical design section including the power channel, the vents and the output channels.

It was also the intention of this subproject to demonstrate the flexibility of design changes associated with the particular element by devising alterable subprogram to do so. The design variation will be applied to place the horizontal control channels in two different locations: one at the power jet exit and the other at an established distance downstream of the power jet.

The Control channels are separately described by a subprogram which was composed of 4 linear slope interpolations and some straight milling instructions. This subprogram enables the placement of the Control channels anywhere along the power jet exit as desired by the designer.

The tapes for the main program were produced by linear and circular interpolations with appropriate parameters fed into the computer. Altogether there were 13 interpolation routines which contained a total of 2050 blocks of information.

The subprogram for the Control channels contains 600 blocks and the total time taken to machine a complete element was 3 hours.

Two models were made with the same main program (for the vertical branch) but one with the Control channels situated at the power jet exit and the other at 0.8" downstream of the power jet. The machining was done in one pass only with a 0.125" dia. slot cutter through a .375" thick Plexiglas. The finished plates are shown in Figure 8.2 and Figure 8.3 respectively.

Visual inspection of the machined surface finish revealed uneven roughness in both the sloped surface (step-wise interpolated) as well as the straight-milled surface. Unlike metal cutting, the Plexiglas has a tendency to become welded with the cutting chips unless extra special precautions are taken; e.g. small depth of cut, optimum spindle speed and feedrate selection, consistent coolant application etc.

Both surfaces were subjected to roughness measurement by a Talysurf, which is a surface roughness measuring instrument similar in principles to that of the Talyrond but applicable only to linear surface inspection. The results of the inspection are shown in Figure 8.4 and Figure 8.5.

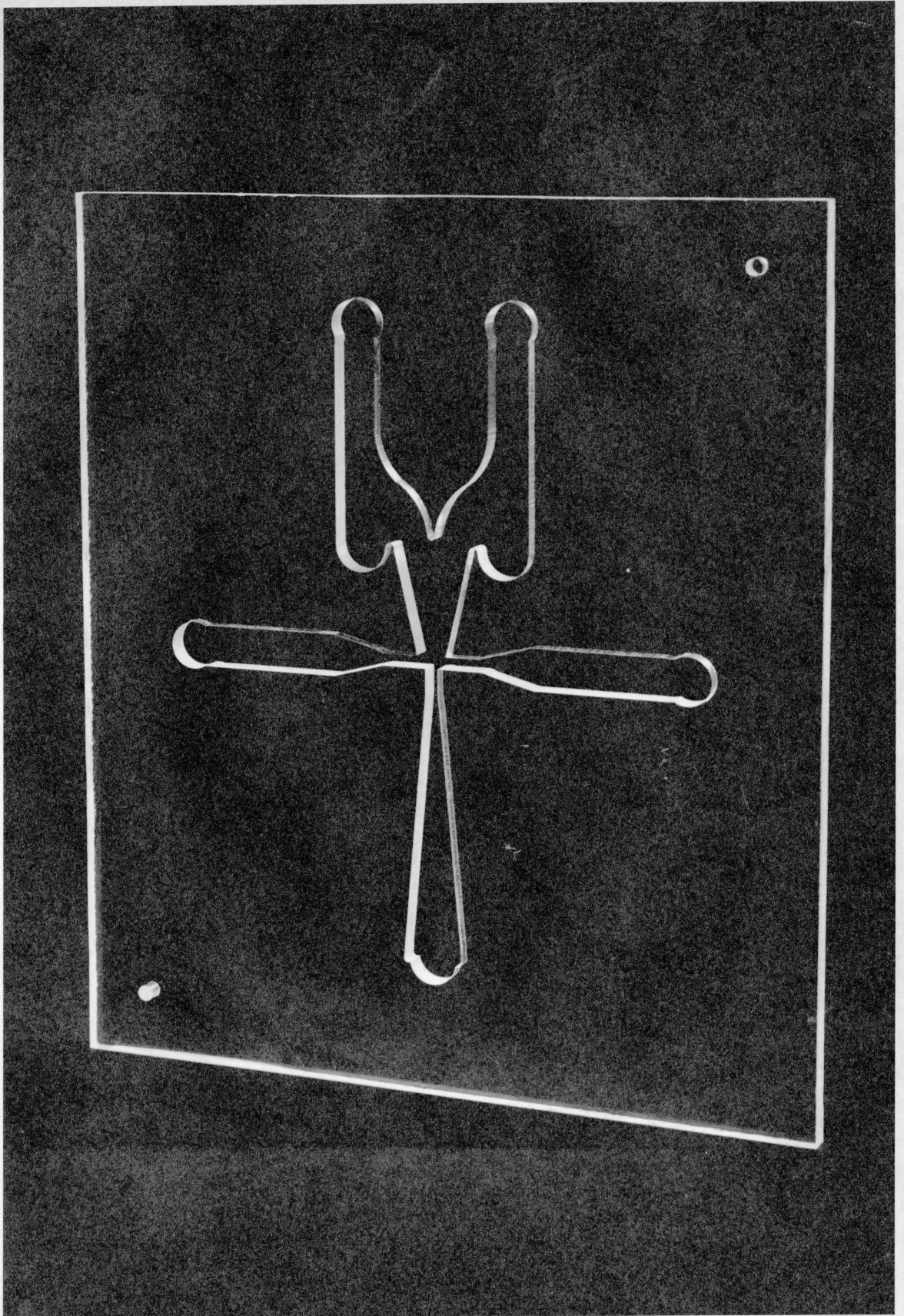


FIGURE 8.2

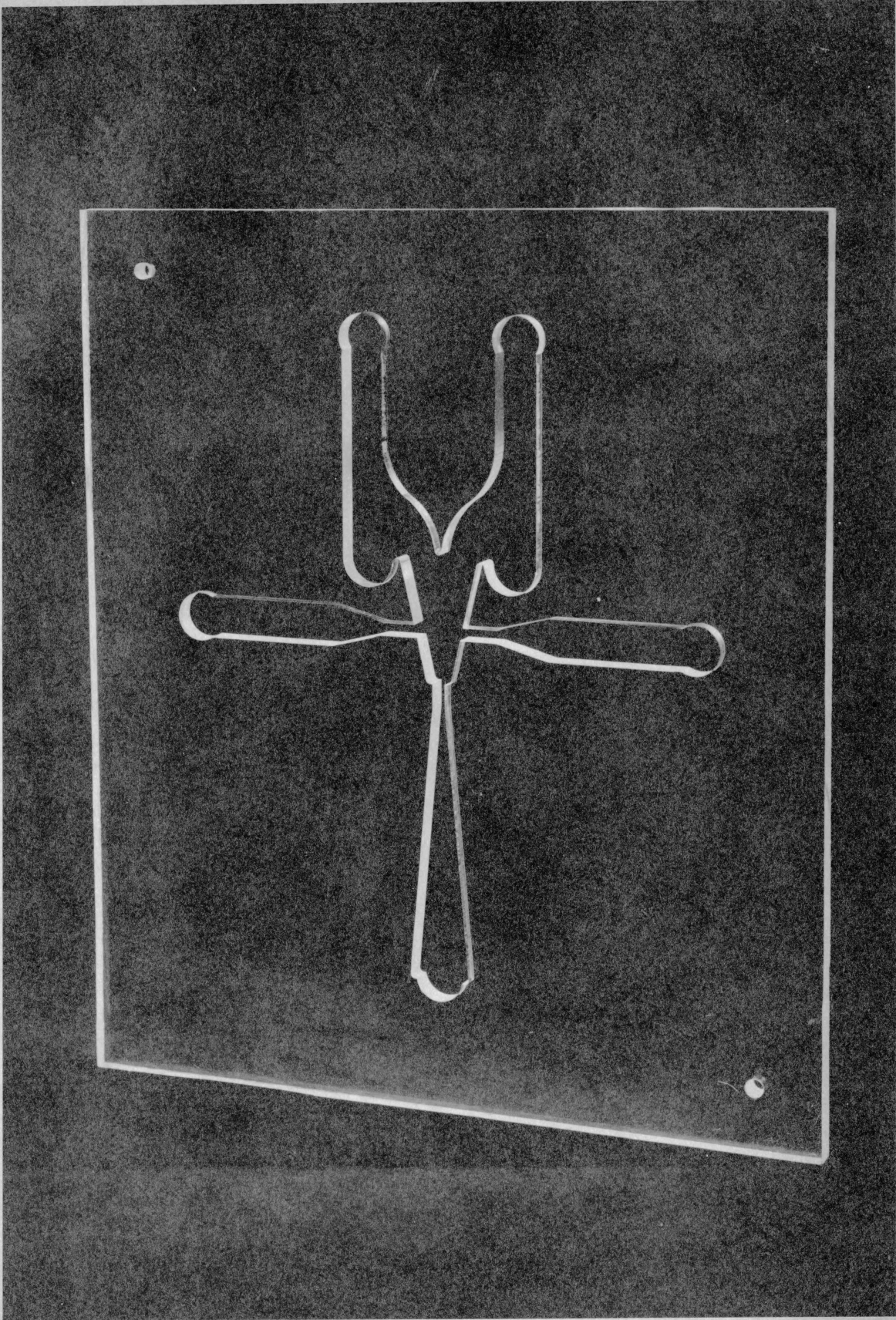
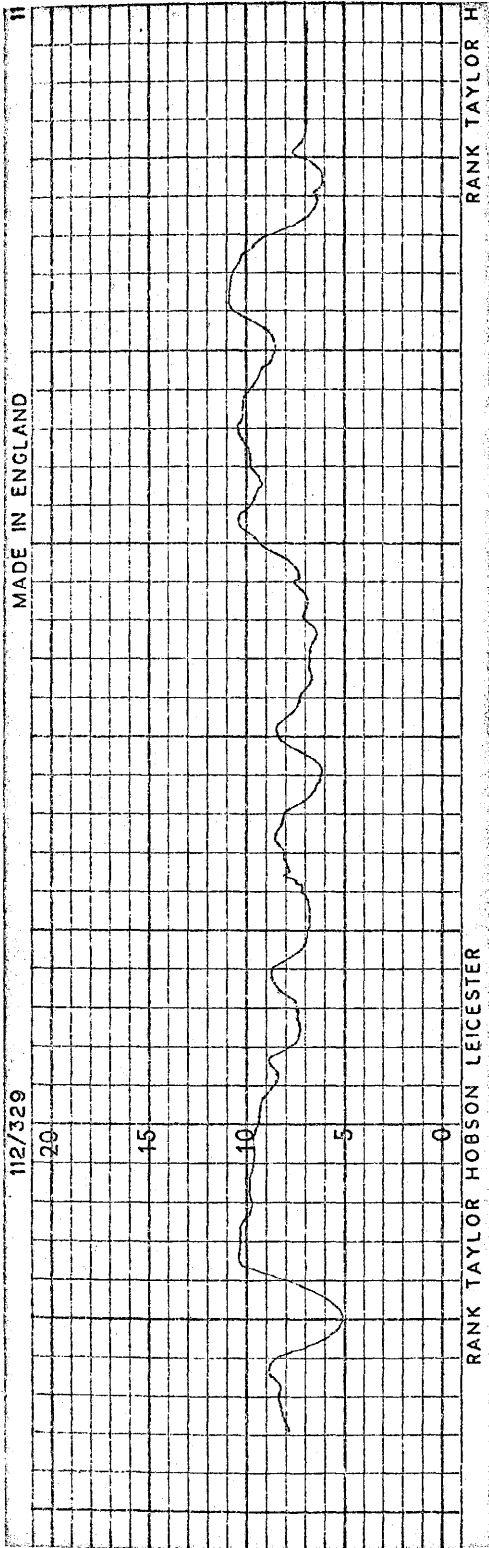


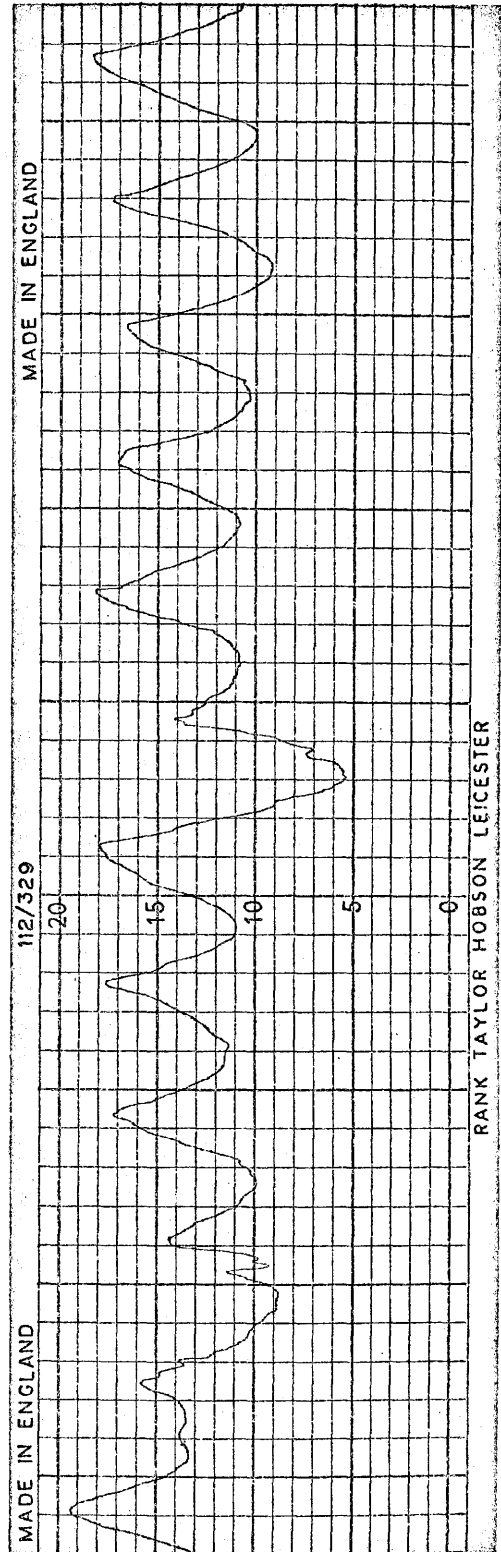
FIGURE 8.3



0.0002"/div. ←

FIGURE 8.4

→ 0.01"/div.



0.0002"/div. ←

FIGURE 8.5

→ 0.01"/div.

Figure 8.4 shows the surface finish on a typical section of the straight-milled surface with a maximum roughness of 0.0012". Figure 8.5 gives the finish on a step-wise linear interpolated slope with a maximum roughness of 0.0028" which is well within the designed tolerance error of ± 0.003 ". Notice the observable periodic waviness of the surface undoubtedly produced by the step-wise movements of the cutter.

Each of the machined plates formed the basic element cavity and with a back plate and a cover plate, was assembled into a working model. A brief description of the performance test of the fluidic models is presented in the next Chapter.

CHAPTER 9PERFORMANCE TESTS OF THE FLUIDIC BISTABLE MODELS

Since the primary objective of the thesis concerns with the extended utilization of the point-to-point numerical control machine tool as an aspect of the production engineering work, the principles of fluid flow phenomenon with respect to the fluid logic device will not be dealt with here. Complementary to Chapter 8, therefore, this Chapter will cover a brief static performance evaluation of the finished bistable models.

The basic characteristics required of a fluid logic element is the power consumption, switching pressure and output recovery pressure. Referring to the schematic diagram as shown in Figure 9.1, the power consumption to the element can be measured using a calibrated Flowrator and a pressure gauge with an air supply feeding the power port P. By gradually increasing the air supply and noting the corresponding flow and pressure values, a power curve, with variables P and Q can be plotted to give the power characteristic of the element.

The switching characteristic can be obtained by allowing the pressure supply to either control port to increase from zero at constant power supply pressures

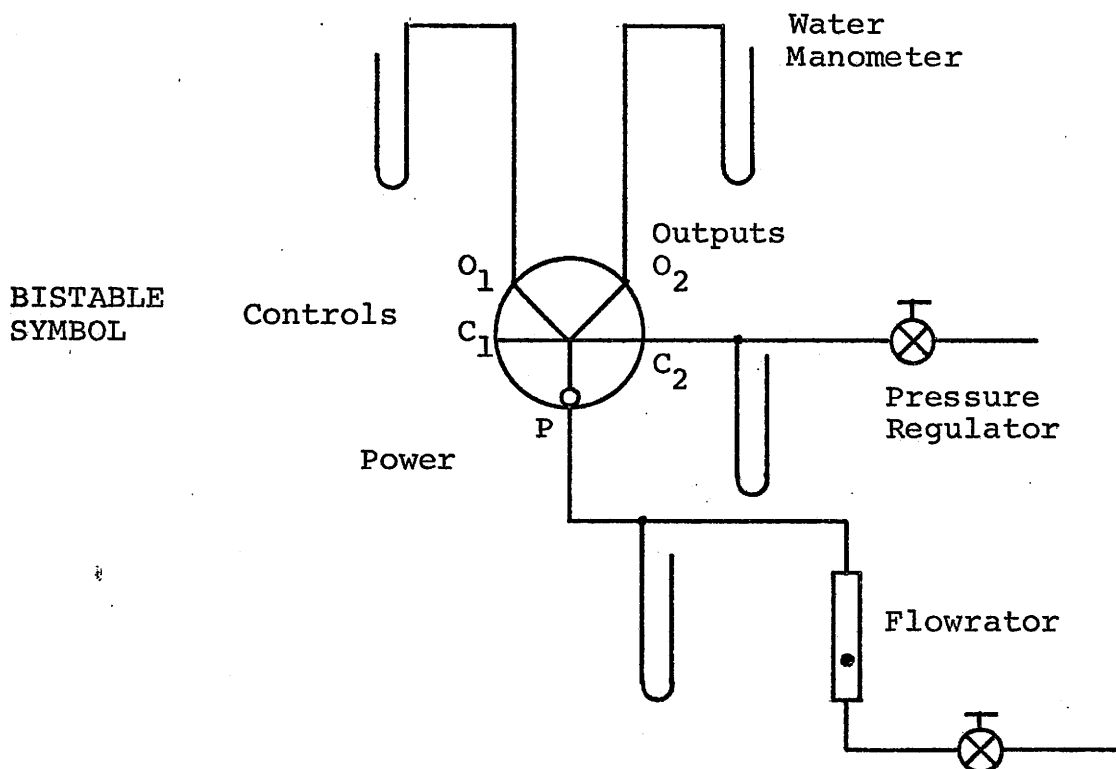


FIGURE 9.1

and noting the point when the output switches from one side to the other. At the same time, the output pressure levels are recorded to give the output pressure characteristic.

Figure 9.2 shows a composite graph of the three characteristics of the bistable element with the Control channels situated 0.8" downstream of the power jet. The design of this element results in a positive switching pressure requirement and a switching gain of about 6.

In contrast, the model with the Control channels situated at the power jet exit requires a negative switching pressure (suction) or even by blocking of the Control port. This design enables the element to be used in circuits where suction switching is mandatory to produce the desired function. The performance stability of this element, however, is not as good as that of the former design.

The characteristics shown from this test do not necessarily represent the optimal performance of the logic elements since they were primarily made for demonstration purposes only.

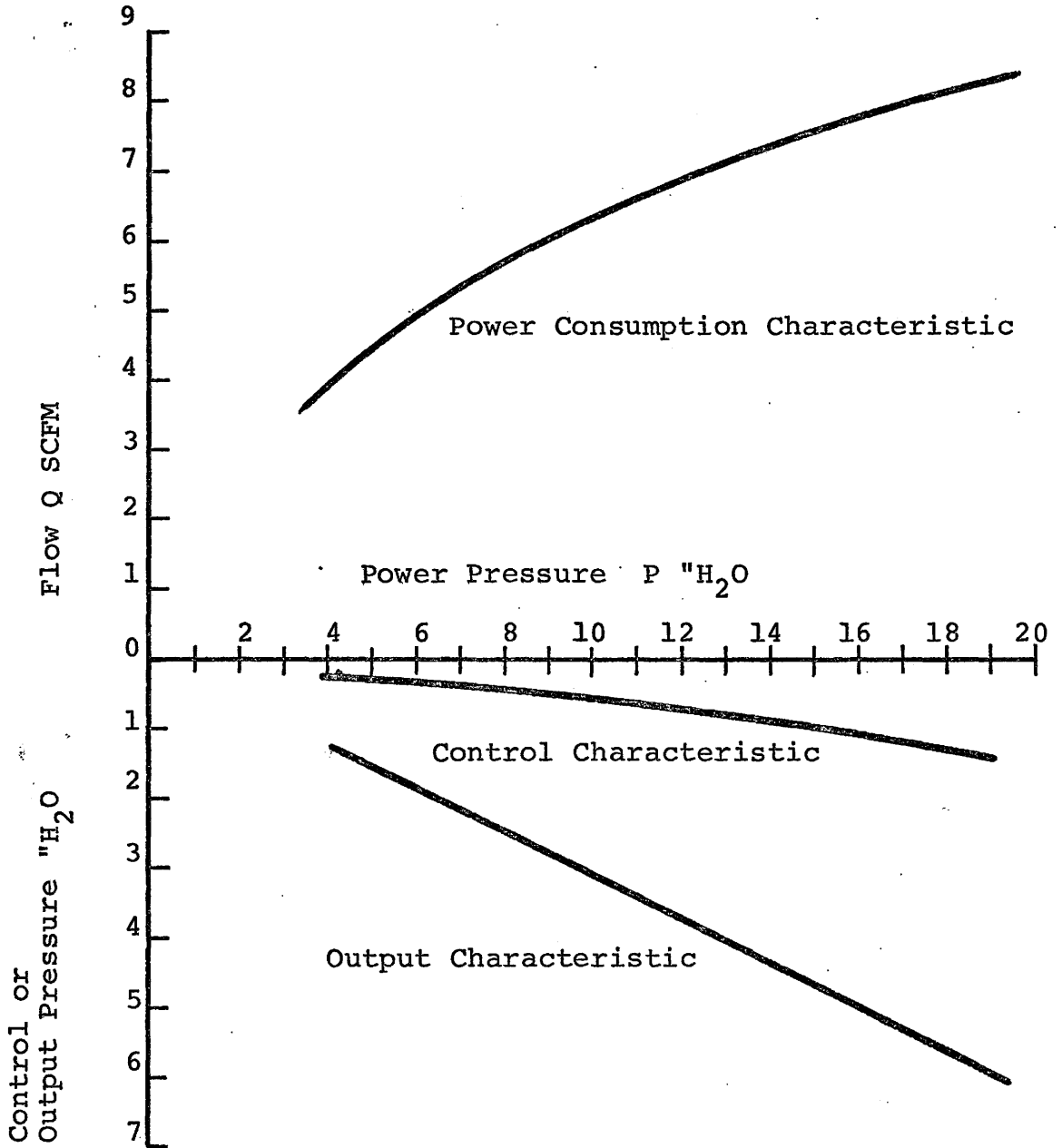


FIGURE 9.2

For more detailed performance characteristics of actual elements and their related applications, the following references are given:

Fluidic Elements Performance Characteristics

Aviation Electric Ltd., Montreal

Industrial Applications of Fluidics

Aviation Electric Ltd., Montreal

CHAPTER 10DISCUSSIONS AND CONCLUSIONS

In recent years, numerical control machine tools have been playing an ever-increasingly important role in the metal works industry. Both point-to-point and contouring machines have been used to their fullest extent as evidenced by their application in more intricate part design and higher accuracies of machines available.

The use of point-to-point machines for contouring work, however, has not been widely accepted yet. This approach may be both practical and economical if one considers its overall economics such as capital cost as compared to contouring machines, availability of computer facilities and programming personnel.

This thesis attempts to provide a guide to the latter part of the consideration by showing the computer-aided programming approach to broaden the use of a positioning machine and the results of such application.

Once the limited use of the point-to-point machine has been widened to include continuous contouring work as evidenced by the foregoing illustrations, it has

become much more versatile and valuable. One inherent disadvantage of the Moog machine, used in this work, is the slow response time associated with the all-pneumatic sensing and control systems.

Normal usage of the machine in macroscopic positioning work does not suffer from this shortcoming since the read-in time and transit time between defined locations are relatively insignificant. When microscopic positioning is required for the contouring application, however, a large number of blocks of information must be provided with each block of information giving command for a minute movement of the cutter between two closely defined points. The consequence is that significant amount of time is taken up to read in and interpret the data instead of performing the required work. An average of 2 seconds was observed to be required by the tape reader for each block of data.

When this was applied against the ellipse program as described in Chapter 7, it was found that out of the total time (50 minutes) required for machining the 6" by 4- $\frac{1}{2}$ " ellipse, about 33 minutes were taken up by the tape reader, leaving only 17 minutes for the actual machining work. The efficiency of the contouring work was therefore about 33%.

Nevertheless, the programming procedure developed here is applicable to all known point-to-point machines and would be more attractive on a machine having an electronic sensing and control package. Other procedures, e.g., the piece-wise linear technique, do not have such a wide applicability and would not be suitable for machines which do not have simultaneous coordinate feeds of different velocities.

A table of comparison, showing the programming and machining requirements of the Step-wise linear versus the piece-wise linear methods, is shown in Table 10.1.

To reiterate the major objectives of this thesis, the following is itemized:

1. Usefulness of a point-to-point machine could be greatly increased by limited contouring capability.
2. Development of general contouring procedure applicable to all known point-to-point machines.
3. Computer-aided programming was the essential ingredient.

TECHNIQUE	STEP-WISE LINEAR ±.005"			PIECE-WISE LINEAR ±.005"		
	NO. OF BLOCKS IN TAPE	MACHINING TIME/PASS	TOTAL MACHINING TIME	NO. OF BLOCKS IN TAPE	MACHINING TIME/PASS	TOTAL MACHINING TIME
4" DIA. CIRCLE IN ALUMINUM	670	30 min.	30 min.	200	9 min.	9 min.
6" x 4½" ELLIPSE IN ALUMINUM	1000	50 min.	50 min.	400	18 min.	18 min.
6" x 3" ELLIPTICAL STEEL DIE	800	40 min.	200 min. through 2" steel	350	16 min.	120 min.* through 2" steel
THIRD DEGREE POLYNOMIAL	1200	55 min.	55 min.	N.A.	N.A.	N.A.
FLUIDIC MODEL	2650	180 min.	180 min.	N.A.	N.A.	N.A.

* Approximate Value

TABLE 10.1

4. Design variations and interchange would be very simple by providing composite subroutine programs.
5. Manufacturing of a wide range of products with spacially related parts would be more economical using item 4.
6. Standardization of parts would be the result of item 5.

In general, the thesis stresses more on the method rather than the mechanics of contouring application of a point-to-point machine. The results of the application are reflected in several examples.

It would be commercially attractive to users with point-to-point machines to consider expanding their machines' capabilities or whose contouring requirement is only a fraction of their total machining work, by employing the knowhow as well as the software developed here.

REFERENCES

1. Arthur D. Roberts and Richard C. Prentice:
"Programming for Numerical Control Machines"
McGraw-Hill, 1968
2. James J. Childs:
"Principles of Numerical Control"
Industrial Press Inc., 1965
3. Raymond E. Howe, Editor:
"Introduction to Numerical Control in
Manufacturing"
ASTME, 1969
4. William C. Leone:
"Production Automation and Numerical Control"
Ronald Press Company, 1967
5. Andrew D. Booth, Editor:
"Progress in Automation, Volume 1"
Butterworths, London, 1960
6. Katherine A. Rempp:
"Numerical Control, A Bibliography"
University of Michigan, 1968
7. Glenn G. Ertell:
"Numerical Control"
Wiley-Interscience, 1969
8. W.H.P. Leslie, Editor:
"Numerical Control Programming Languages"
North-Holland Publishing Company, 1970

9. Prepared by Software Writing Group:
 - "Introduction to Programming: PDP-8 Family Computers"
 - "Programming Languages: PDP-8 Family Computers"
 - "Small Computer Handbook"Digital Equipment Corporation, 1970

10. L. Miller:
 - "Engineering Dimensional Metrology"Edward Arnold Ltd. 1962

11. A.W. Judge:
 - "Engineering Precision Measurements"Chapman and Hall Ltd., 1957

12. M.V. Heideman:
 - "A Study of Some Aspects of Numerically Controlled Machine Tools"M. Eng. Thesis, McMaster University, 1968

13. A. Camp, ASTME 32nd Annual Meeting Collected Papers:
 - "Manual and Computer Programming Requirements for N/C Point-to-Point Machine Tools"Vol. 64, Book 1, Paper 573, 1964

14. J.J. Childs:
 - "Contouring Machining by Point-to-Point N.C."Machinery, Vol. 73 (December 1966), pp. 112-113

15. G. DeGroat:
 - "Contouring on Point-to-Point N.C."American Machinist, Vol. 110 (August 15, 1966), pp. 110-111

16. J. Sandford:
"Low-Cost N.C. Contouring"
American Machinist, Vol. 110 (April 25, 1966),
pp. 83-85

17. "Moog Hydra-Point Tape Control System"
Machinery, Vol. 67 (January 1961), pp. 147-149

18. C. Husemeyer
"Some Aspects of Costing and Contouring
Program for Numerically Controlled Machine
Tools"
Master of Engineering Thesis, 1971,
McMaster University, Hamilton, Ontario.

APPENDIX ATHE ADVANTAGES OF NUMERICAL CONTROL MACHINE TOOLS

It would be appropriate here to enumerate the different advantageous characteristics of the numerical control machine tools in comparison with their conventional counterparts.

Broadly speaking, the tangible benefits of the numerical control machine systems over the conventional ones include greater speed, accuracy, repeatability, ease and economy of process planning, savings in tooling, ease of setup, versatility of operation, economy in elimination of scrap and shortness of lead times. In summary, numerical control's greatest assets are its flexibility and ability to effect dramatic increases in productivity.

Catagorically, the advantages can still be further subdivided into the following groups, viz: Productivity, flexibility, quality and administrative economy.

PRODUCTIVITY

Increased productivity is a direct consequence of automation. With the utilization of numerical control machine tools, substantial savings and improvement in direct labour, set-up time, change-over time, actual machining time, operator's decision and efficiency can always be effected.

FLEXIBILITY

This is associated with the unique characteristics of the controlling medium. As it was pointed out before, almost all design features and commanding instructions are represented by coded numbers or letters to effect the desired operational functions. Therefore the main effort in making design changes on the work lies in the simple alteration of the coded symbols pertaining to the intended part. Normally this is done by providing a number of interchangeable subroutine programs incorporated into the main program.

The flexibility in design changes can affect savings in tooling cost; ease of fabrication of similar items and improving standardization; producing prototypes economically; and make mirror image of parts (left-hand or right-hand parts) with minimal effort.

QUALITY

The requirement for high quality in a product may be restricted by the skill of the operator if conventional means is employed. This is not necessarily true if numerical control machine tools are used. Certain machines are designed to give extremely high accuracies in some applications where they cannot be achieved by manual methods, and often performing the tasks in a much shorter time. This is attributed to the use of high speed electronic control systems whose sensing and reaction responses are unattainable in humans.

Furthermore, the ability of the numerical control machine tool to provide repeatability of the required accuracy is almost unmatched by other methods. This is characterized by their absolute reproducibility of machining movements in response to the same controlling medium under identical environmental conditions.

The quality of accuracy and repeatability leads to parts interchangeability and ultimately, high degree of standardization.

ADMINISTRATIVE ECONOMY

The savings in administrative expenses as a direct result of applying numerical controls utilization can be very significant by considering the following improvements.

The lead time from conception stage to finished product can be greatly reduced with the result of more up-to-date marketing advantage and improved competition.

Scrap in the production stage can almost be eliminated. Floor space and inventory of tool and material can be kept under the optimum operating conditions. Efficiency of machine and operator utilization can be highly attained.

Production planning can be more accurate and less costly. Machine loading, performance, scheduling can be predicted with greater precision.

To complete the picture, it will be useful to once again clarify some false notions concerning the use of numerical controls.

First, they are basically special purpose machine tools, either completely manufactured or conventional machines retrofitted with control packages, whose built-in high-speed feedback response enables them to perform highly complicated tasks accurately and automatically. Their sole functions are for effecting increased short-run productivity even down to a one part prototype. In fact, in some instances, design changes are simply a matter of changing the input tapes. They are NOT automatic mass production machines although tape controlled and even direct computer feedback controlled, but specially designed machine tools are being utilized very extensively in high volume production application.

Second, the access to a computer is not always needed as this was generally regarded to be the case. Since the majority of the current numerical control machine tools are of point-to-point type, they do not require the service of a computer in any phase of their work. The program sheet for these machines can be prepared directly from the design drawings together with the necessary data associated with the machining procedure. (It should be remembered, however, that the liberal use of computer-aided programming as described in the foregoing Chapters was made mandatory because of the necessity of converting a point-to-point into a continuous contour cutting machine).

PAL PRINTOUT OF THE EIA CONVERSION PROGRAM

/// EIA

*200

```
0200 6046 GO, TLS
0201 6032 START, KCC
0202 4214 JMS ACCPT
0203 0101 AND MASK
0204 1102 TAD M60
0205 3100 DCA HOLD
0206 4227 JMS CHK
0207 7440 SZA
0210 5201 JMP START
0211 1100 TAD HOLD
0212 4221 JMS TYPE
0213 5201 JMP START
0214 0000 ACCPT, 0
0215 6031 KSF
0216 5215 JMP .-1
0217 6036 KRB
0220 5614 JMP I ACCPT
0221 0000 TYPE, 0
0222 6041 TSF
0223 5222 JMP .-1
0224 6046 TLS
0225 7300 CLA CLL
0226 5621 JMP I TYPE
0227 0000 CHK, 0
0230 1100 TAD HOLD
0231 7440 SZA
0232 5236 JMP .+4
0233 1103 TAD P0
0234 3100 DCA HOLD
0235 5627 JMP I CHK
0236 1116 TAD M1
0237 7640 SZA CLA
0240 5244 JMP .+4
0241 1104 TAD P1
0242 3100 DCA HOLD
0243 5627 JMP I CHK
0244 1100 TAD HOLD
0245 1117 TAD M2
0246 7640 SZA CLA
0247 5253 JMP .+4
0250 1105 TAD P2
0251 3100 DCA HOLD
0252 5627 JMP I CHK
0253 1100 TAD HOLD
0254 1120 TAD M3
0255 7640 SZA CLA .
```

0256	5262	JMP .+4
0257	1106	TAD P3
0260	3100	DCA HOLD
0261	5627	JMP I CHK
0262	1100	TAD HOLD
0263	1121	TAD M4
0264	7640	SZA CLA
0265	5271	JMP .+4
0266	1107	TAD P4
0267	3100	DCA HOLD
0270	5627	JMP I CHK
0271	1100	TAD HOLD
0272	1122	TAD M5
0273	7640	SZA CLA
0274	5300	JMP .+4
0275	1110	TAD P5
0276	3100	DCA HOLD
0277	5627	JMP I CHK
0300	1100	TAD HOLD
0301	1123	TAD M6
0302	7640	SZA CLA
0303	5307	JMP .+4
0304	1111	TAD P6
0305	3100	DCA HOLD
0306	5627	JMP I CHK
0307	1100	TAD HOLD
0310	1124	TAD M7
0311	7640	SZA CLA
0312	5316	JMP .+4
0313	1112	TAD P7
0314	3100	DCA HOLD
0315	5627	JMP I CHK
0316	1100	TAD HOLD
0317	1125	TAD M8
0320	7640	SZA CLA
0321	5325	JMP .+4
0322	1113	TAD P8
0323	3100	DCA HOLD
0324	5627	JMP I CHK
0325	1100	TAD HOLD
0326	1126	TAD M9
0327	7640	SZA CLA
0330	5334	JMP .+4
0331	1114	TAD P9
0332	3100	DCA HOLD
0333	5627	JMP I CHK
0334	1100	TAD HOLD
0335	1127	TAD MSP
0336	7640	SZA CLA
0337	5343	JMP .+4
0340	1103	TAD P0
0341	3100	DCA HOLD

Q342	5627	JMP I CHK
Q343	1100	TAD HOLD
Q344	1130	TAD MCR
Q345	7440	SZA
Q346	5627	JMP I CHK
Q347	1115	TAD PCR
Q350	3100	DCA HOLD
Q351	5627	JMP I CHK

/DEFINITIONS:

*100

Q100	0000	HOLD, 0
Q101	0177	MASK, 177
Q102	7720	M60, -60
Q103	0040	P0, 40
Q104	0001	P1, 1
Q105	0002	P2, 2
Q106	0023	P3, 23
Q107	0004	P4, 4
Q110	0025	P5, 25
Q111	0026	P6, 26
Q112	0007	P7, 7
Q113	0010	P8, 10
Q114	0031	P9, 31
Q115	0200	PCR, 200
Q116	7777	M1, -1
Q117	7776	M2, -2
Q120	7775	M3, -3
Q121	7774	M4, -4
Q122	7773	M5, -5
Q123	7772	M6, -6
Q124	7771	M7, -7
Q125	7770	M8, -10
Q126	7767	M9, -11
Q127	0020	MSP, 20
Q130	0043	MCR, 43

APPENDIX CFORTRAN CIRCLE PROGRAM

100PROGRAM OF 4 IN CIRCLE COORDINATES (1/2 IN CUTTER 10 THOU TOLERANCE)

```

105 A1=1019.
110 A2=0.
115 A3=0.
120 A4=2.
125 B1=2000.
130 B2=0.
135 B3=0.
140 B4=186.
145 C1=3000.
150 C2=0.
155 C3=0.
160 C4=8.
165 D1=4000.
170 D2=0.
175 D3=0.
180 D4=3188.
185 E1=5002.
190 E2=2500.
195 E3=40000.
200 E4=80.
205 G1=6903.
210 G2=0.
215 G3=35000.
220 G4=189.
225 PRINT 10,A1,A2,A3,A4
230 PRINT 10,B1,B2,B3,B4
235 PRINT 10,C1,C2,C3,C4
240 PRINT 10,D1,D2,D3,D4
245 PRINT 10,E1,E2,E3,E4
250 PRINT 10,G1,G2,G3,G4
255 10 FORMAT(1X,I5,I4,I5,I4)
260 AS=7. /STARTING SEQUENCE NO.
265 AP=6. /MILLING CODE
270 AT=1. /FEEDRATE=1 IN/MIN.
275 AMX=84. /X-FEED
280 AMY=85. /Y-FEED
285 AY=0. /INITIAL Y VALUE
290 AX=1750. /INITIAL X VALUE
295 RAD=1750. /RADIUS OF CIRCLE
300 20 AY=AY+1.
305 AR=SQRT(AX**2+AY**2)
310 ARL=1755. /OUTER LIMIT
315 IF(ARL-AR) 25,25,20
320 25 AXX=4000.-AX
325 AYY=4000.+AY

```

```

330 PRINT 30,AS,AP,AXX,AYY,AT,AMY
335 30 FORMAT(I3,I1,I5,I5,I3,I2)
340 40 AX=AX-1.
345 IF(AX-0.) 45,45,41
350 41 AR=SQRT(AX**2+AY**2)
355 ARS=1745. /INNER LIMIT
360 IF(AR-ARS) 42,42,40
365 42 AXX=4000.-AX
370 AYY=4000.+AY
375 AS=AS+1.
380 PRINT 30,AS,AP,AXX,AYY,AT,AMX
385 AS=AS+1.
390 IF(RAD-AY) 45,45,20
395 45 CONTINUE
400 50 AX=AX+1.
405 AR=SQRT(AX**2+AY**2)
410 IF(ARL-AR) 55,55,50
415 55 AXX=4000.+AX
420 AYY=4000.+AY
425 AS=AS+1.
430 PRINT 30,AS,AP,AXX,AYY,AT,AMX
435 60 AY=AY-1.
440 IF(AY-0.) 65,65,61
445 61 AR=SQRT(AX**2+AY**2)
450 IF(AR-ARS) 62,62,60
455 62 AXX=4000.+AX
460 AYY=4000.+AY
465 AS=AS+1.
470 PRINT 30,AS,AP,AXX,AYY,AT,AMY
475 IF(RAD-AX) 65,65,50
480 65 CONTINUE
485 70 AY=AY+1.
490 AR=SQRT(AX**2+AY**2)
495 IF(ARL-AR) 75,75,70
500 75 AXX=4000.+AX
505 AYY=4000.-AY
510 AS=AS+1.
515 PRINT 30,AS,AP,AXX,AYY,AT,AMY
520 80 AX=AX-1.
525 IF(AX-0.) 85,85,81
530 81 AR=SQRT(AX**2+AY**2)
535 IF(AR-ARS) 82,82,80
540 82 AXX=4000.+AX
545 AYY=4000.-AY
550 AS=AS+1.
555 PRINT 30,AS,AP,AXX,AYY,AT,AMX
560 IF(RAD-AY) 85,85,70
565 85 CONTINUE
570 90 AX=AX+1.
575 AR=SQRT(AX**2+AY**2)
580 IF(ARL-AR) 95,95,90

```

```
585 95 AXX=4000.-AX
590 AYY=4000.-AY
595 AS=AS+1.
600 PRINT 30,AS,AP,AXX,AYY,AT,AMX
605 100 AY=AY-1.
610 IF(AY-0.) 105,105,101
615 101 AR=SQRT(AX**2+AY**2)
620 IF(AR-ARS) 102,102,100
625 102 AXX=4000.-AX
630 AYY=4000.-AY
635 AS=AS+1.
640 PRINT 30,AS,AP,AXX,AYY,AT,AMY
645 IF(RAD-AX) 105,105,90
650 105 STOP
655 END
```

APPENDIX DFORTRAN ELLIPSE PROGRAM

```

100C  PROGRAM OF 6 BY 4-1/2 ELLIPSE COORDINATES (10 THOU TOLERANCE)

105  A1=1019.
110  A2=0.
115  A3=0.
120  A4=2.
125  B1=2000.
130  B2=0.
135  B3=0.
140  B4=186.
145  C1=3000.
150  C2=0.
155  C3=0.
160  C4=8.
165  D1=4000.
170  D2=0.
175  D3=0.
180  D4=3188.
185  E1=5001.
190  E2=0.
195  E3=40000.
200  E4=80.
205  G1=6903.
210  G2=0.
215  G3=35000.
220  G4=189.
225  PRINT 10,A1,A2,A3,A4
230  PRINT 10,B1,B2,B3,B4
235  PRINT 10,C1,C2,C3,C4
240  PRINT 10,D1,D2,D3,D4
245  PRINT 10,E1,E2,E3,E4
250  PRINT 10,G1,G2,G3,G4
255  10 FORMAT(IX,I5,I4,I5,I4)
260  AS=6.           /STARTING SEQUENCE NO.
265  AP=6.           /MILLING CODE
270  AT=5.           /FEEDRATE= 5 IN/MIN.
275  AMX=84.         /X-FEED
280  AMY=85.         /Y-FEED
285  CX=4000.        /CENTRE-X
290  CY=4000.        /CENTRE-Y
295  AC=3000.        /SEMI MAJOR AXIS
300  BC=2250.        /SEMI MINOR AXIS
305  AA=3005.        /OUTER MAJOR LIMIT
310  BB=2255.        /OUTER MINOR LIMIT
315  A=2995.         /INNER MAJOR LIMIT
320  B=2245.         /INNER MINOR LIMIT

```

```

325 AX=3000. /INITIAL X VALUE
330 AY=0. /INITIAL Y VALUE
335 20 AY=AY+1.
340 ONE=AX**2/AA**2+AY**2/BB**2
345 IF(ONE-1.)20,25,25
350 25 AXX=CX-AX
355 AYY=CY+AY
360 AS=AS+1.
365 PRINT 30,AS,AP,AXX,AYY,AT,AMY
370 30 FORMAT(I3,I1,I5,I5,I3,I2)
375 40 AX=AX-1.
380 IF(AX-0.)45,45,41
385 41 ONE=AX**2/A**2+AY**2/B**2
390 IF(ONE-1.)42,42,40
395 42 AXX=CX-AX
400 AYY=CY+AY
405 AS=AS+1.
410 PRINT 30,AS,AP,AXX,AYY,AT,AMX
415 IF(BC-AY)45,45,20
420 45 CONTINUE
425 50 AX=AX+1.
430 ONE=AX**2/AA**2+AY**2/BB**2
435 IF(ONE-1.)50,55,55
440 55 AXX=CX+AX
445 AYY=CY+AY
450 AS=AS+1.
455 PRINT 30,AS,AP,AXX,AYY,AT,AMX
460 60 AY=AY-1.
465 IF(AY-0.)65,65,61
470 61 ONE=AX**2/A**2+AY**2/B**2
475 IF(ONE-1.)62,62,60
480 62 AXX=CX+AX
485 AYY=CY+AY
490 AS=AS+1.
495 PRINT 30,AS,AP,AXX,AYY,AT,AMY
500 IF(AC-AX)65,65,50
505 65 CONTINUE
510 70 AY=AY+1.
515 ONE=AX**2/AA**2+AY**2/BB**2
520 IF(ONE-1.)70,75,75
525 75 AXX=CX+AX
530 AYY=CY-AY
535 AS=AS+1.
540 PRINT 30,AS,AP,AXX,AYY,AT,AMY
545 80 AX=AX-1.
550 IF(AX-0.)85,85,81
555 81 ONE=AX**2/A**2+AY**2/B**2
560 IF(ONE-1.)82,82,80
565 82 AXX=CX+AX

```

```
570 AYY=CY-AY
575 AS=AS+1.
580 PRINT 30,AS,AP,AXX,AYY,AT,AMX
585 IF(BC-AY)85,85,70
590 85 CONTINUE
595 90 AX=AX+1.
600 ONE=AX**2/AA**2+AY**2/BB**2
605 IF(ONE-1.)90,95,95
610 95 AXX=CX-AX
615 AYY=CY-AY
620 AS=AS+1.
625 PRINT 30,AS,AP,AXX,AYY,AT,AMX
630 100 AY=AY-1.
635 IF(AY-0.)105,105,101
640 101 ONE=AX**2/A**2+AY**2/B**2
645 IF(ONE-1.)102,102,100
650 102 AXX=CX-AX
655 AYY=CY-AY
660 AS=AS+1.
665 PRINT 30,AS,AP,AXX,AYY,AT,AMY
670 IF(AC-AX)105,105,90
675 105 STOP
680 END
```