The computer will be programmed to select the appropriate spindle speed and feed rate from an input of information relating to these factors. To assist the input of information there will be a material file and surface roughness file within the computer memory, as shown in Figure 5.2.

Cutting tools available will also be numerically identified. A simple question-and-answer routine will extract from the computer memory all the necessary data to give the correct cutting conditions; the programmer then enters the axes calculations or parameters to finish the program. An example of a question-and-answer routine is as follows:

CRT question	**Operator's keyed response**
Material?	5 input
Material diameter?	2.0 input
Surface code?	4 input
Tool number?	8 input

The preceding illustrates just a small part of the total data input necessary to make a component, and, having established basic cutting conditions, the programmer would proceed to feed in additional data relating to slide movement, etc. However, slide movement data input can be reduced to a question-and-answer routine, even when the movements are complex, such as when machining a radius or cutting a screw thread.

Consider the production of a thread on a work diameter that has just been produced by the preceding data entry. The required data input may be restricted to the following questions:

Thread root diameter?
Lead?
Number of starts?
Start location?
Finish location?

From this the control will determine the number of passes necessary, the depth of cut taken by each pass, and the feed rate needed to produce the required pitch. Even the spindle speed may vary automatically to allow for roughing and finishing cuts.

MATERIAL STOCK FILE	
CODE	MATERIAL
1	MILD STEEL
2	MED. CARBON STEEL
3	STAINLESS STEEL
4	CAST IRON
5	ALUMINUM

SURFACE ROUGHNESS FILE	
CODE	Microinches
1	100
2	50
3	25
4	12.5
5	6.3

Figure 5.2 *Material file and surface roughness file.*

Routines such as the one described are commonly known as "canned cycles" and are not restricted to conversational MDI but may also form part of other programming systems. The use of such routines is described in more detail in Chapter 6.

Data entered in response to questions can be recorded, usually on magnetic tape, for future use. Some advanced conversational MDI systems incorporate the use of computer graphics (see "Graphical Numerical Control," Chapter 8).

Perforated Tape Input

Not so long ago numerical control was generally referred to as tape control, an indication of the important part this input medium has played in the development of the technology. The expression is not quite so common as it was, but perforated tape is still widely used.

The basis of tape control is the transfer of coded information contained on a perforated tape to the machine control unit via a tape reader.

The standard tape width is 1 in./25 mm. Originally only paper tape was used, and it is still the most popular material, a factor very much in its favor being its low cost. It is available in rolls or precisely folded in a concertina or fanlike arrangement (Figure 5.3). The rolls are most commonly used, but the folded paper is possibly easier to store.

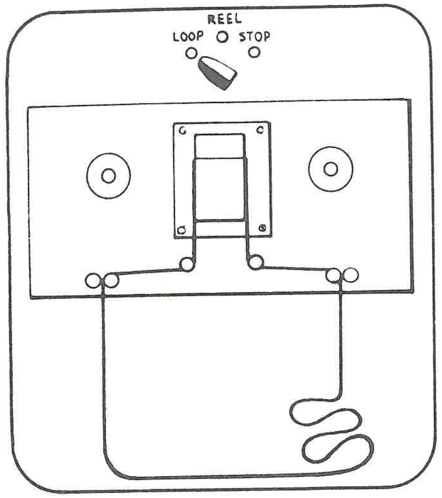
One of the problems with paper tape is that the sprocket drive holes that are used to carry the tape through some tape readers tend to wear or even tear. Also, a tape can easily be damaged by contact with oil, which is always likely in a shop atmosphere. This has led to the introduction of other materials or combinations of materials; but, while these tapes are more durable, they are usually more expensive.

Examples of tape materials other than paper include polyester film, paper–polyester–paper laminates, polyester–aluminum-foil–polyester laminates, metallized polyester, and aluminum laminate or Mylar. Some of these tapes can cause excessive punch wear and the adhesives used to produce some laminates have also presented punching problems.

The choice of tape is also affected by the tape reader being used. The prime function of the reader is to detect the presence and position of the perforations in the tape. Tape readers may be fiber optic or photoelectrical. Various tape readers are shown in Figure 5.4.

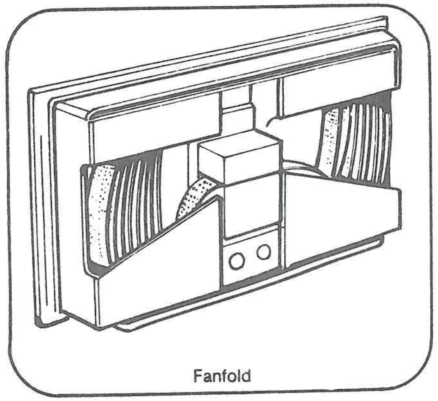
Fiber optic and photoelectric readers have light either passing through the perforations or being reflected off a reflector positioned behind the tape. Having detected a perforation, the reader converts this information into an electrical signal that is transmitted to the control unit. Light source readers using the reflective principle will require a tape with a nonreflective surface finish or color, while the direct light readers require tapes that are not translucent.

The qualities required of tapes used in photoelectric tape readers have resulted in a variety of colors being used. Some tapes are dual colored, which helps to reduce the possibility of reverse loading in the readers.



Loop

The tape is joined at its ends to produce a continuous loop.

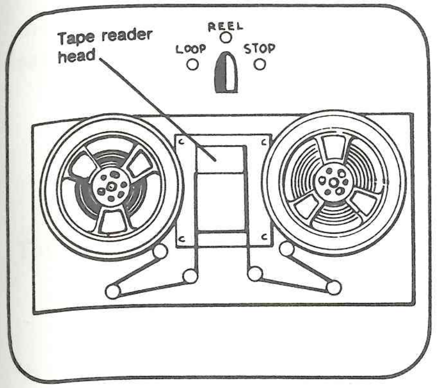


Fanfold or random length

The tape is drawn across the reader and passed from one compartment to the other.

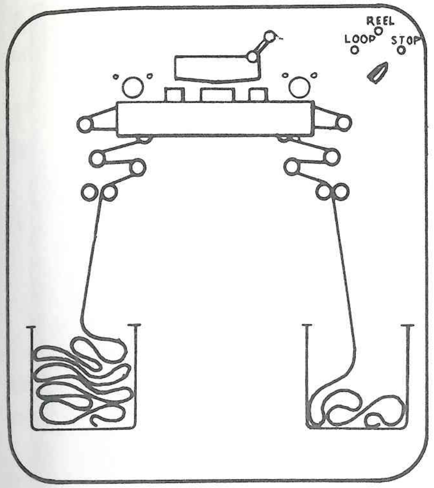
Figure 5.3 *CNC tape configurations.*

Reverse loading is also prevented by the offset of the sprocket holes when used, placing the edge of the tape closest to the sprocket holes to the back of the reader. In some cases the feed direction is indicated by arrows printed on the tape. In addition to this, when the tape is severed from the main roll after punching, the leading end will be pointed and the trailing end will have a corresponding recess.



Reel

The tape is wound from one reel to another across the tape reader head.



Tape Tumble Box

The tape is left loose at both ends and is allowed to fall freely into a tumble box. Although this was an original configuration, it is rarely used today.

Figure 5.3 *(Continued)*

The advent of the computer as an integral part of control systems has minimized the strength and wear problems associated with plain paper tapes. On older control units the tape is run through the reader each time a component is machined. For short programs it can be spliced to form a continuous loop, thus eliminating the need for rewinding. For longer programs the tape is wound from reel to reel. Now it is common practice to use the tape more as a storage medium, feeding the data it contains into the control unit computer by just one pass through the tape reader. The computer retains the data in its memory and this facilitates data retrieval as and when required. When data are transferred from tape to the computer memory, the tape can be removed from the reader to avoid contamination.

Tape Standards and the Binary Code When a tape reader detects a perforation, the transmission of an electrical signal to the control unit results. The simplest way an electrical signal can be meaningful is by its presence or absence, creating an on–off effect. The detection of a perforation registers an "on" signal, and this signal will be given further meaning by the position of the perforation that caused it, as will be explained subsequently.

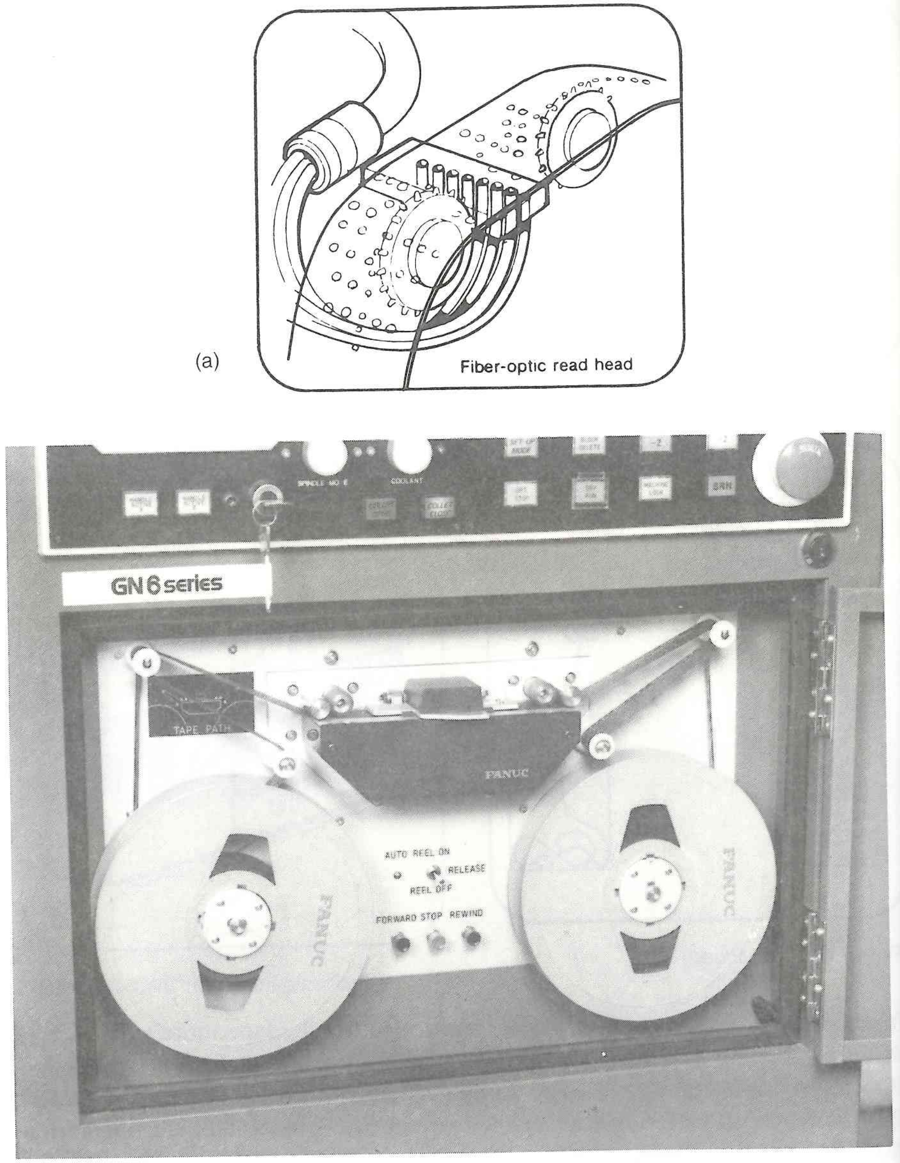


Figure 5.4 (a) *Fiber-optic and* (b) *photoelectric tape reader.*

Whatever the type of tape reader, the switching effect is achieved by using binary arithmetic, a system that has 2 as a base and can convey numerical values in terms of 1 and 0: "on" and "off."

To understand the binary system of numbers it is helpful to look more closely at the familiar decimal system, which uses 10 as a base. In this system numerical values are constructed from multiples of units, tens, hundreds, thousands, etc.

Unit	$1 = 10^0$
Ten	$10 = 10^1$
Hundred	$100 = 10^2$
Thousand	$1000 = 10^3$

Thus the number 2345 is made up as follows:

Thousand	**Hundred**	**Ten**	**Unit**
(10^3) 2	(10^2) 3	(10^1) 4	(10^0) 5

This is a convenient, well-understood method of expressing numbers, but unfortunately it does not readily relate to the requirements of electrical switching control.

Now consider the application of the binary system using two as the base—2^0, 2^1, 2^2, 2^3, and so on—and then relate this to certain decimal values as follows:

Let $2^0 = 1$

Then $2^1 = 2$
Then $2^2 = 4$
Then $2^3 = 8$ and so on

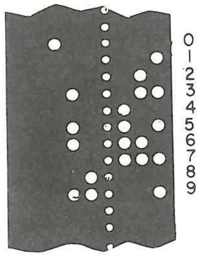
By using this small range of binary values any decimal digit can be expressed as shown in Table 5.1.

Now if the holes in the perforated control tape are arranged in columns or tracks corresponding to the binary values indicated in Table 5.1 it is possible

Table 5.1 Decimal digits expressed in binary.

Decimal Digit	**Binary equivalent**				**Composition**
	2^3	**2^2**	**2^1**	**2^0**	
1				1	$2^0 = 1$
2			1	0	$2^1 + 0 = 2$
3			1	1	$2^1 + 2^0 = 3$
4		1	0	0	$2^2 + 0 + 0 = 4$
5		1	0	1	$2^2 + 0 + 2^0 = 5$
6		1	1	0	$2^2 + 2^1 + 0 = 6$
7		1	1	1	$2^2 + 2^1 + 2^0 = 7$
8	1	0	0	0	$2^3 + 0 + 0 + 0 = 8$
9	1	0	0	1	$2^3 + 0 + 0 + 2^0 = 9$

to express the required decimal values by making perforations in the appropriate places:



The tape shown above has, eight tracks or vertical columns where a hole might be punched. Five columns are all that is required to express numbers, columns one through four indicate binary values and six indicates a zero. However, the numbers used in numerical control need an identity; for example, a slide movement not only has a dimensional value but the axis in which movement is required has to be defined. This definition is achieved by using letters, as explained in Chapter 1.

There are two tape standards in general use, the ISO (International Standards Organization or ASCII, American Standard Code for Information Interchange) and the EIA (Electrical Industries Association); the latter was developed in the United States of America and gained wide acceptance before the introduction of the ISO standards. The two tapes identify letters in different ways, but both use the binary coded decimal system for numbers. The following description is applicable only to the ISO standard.

The 26 letters of the alphabet are identified numerically from 1 to 26. We have seen that the digits 1 to 9 can be expressed using four binary columns. To include the numbers 10 to 26 requires a fifth column, a fifth track in the tape, so that the decimal value can be expressed *in one row* of punched holes as shown in Table 5.2.

Table 5.2 Letters of the alphabet expressed numerically.

Letter	Decimal Digit	Binary equivalent					Composition
		2^4	2^3	2^2	2^1	2^0	
J	10		1	0	1	0	$2^3 + 0 + 2^1 + 0 = 10$
K	11		1	0	1	1	$2^3 + 0 + 2^1 + 2^0 = 11$
L	12		1	1	0	0	$2^3 + 2^2 + 0 + 0 = 12$
M	13		1	1	0	1	$2^3 + 2^2 + 0 + 2^0 = 13$
N	14		1	1	1	0	$2^3 + 2^2 + 2^1 + 0 = 14$
O	15		1	1	1	1	$2^3 + 2^2 + 2^1 + 2^0 = 15$
P	16	1	0	0	0	0	$2^4 + 0 + 0 + 0 + 0 = 16$
Q	17	1	0	0	0	1	$2^4 + 0 + 0 + 0 + 2^0 = 17$
R	18	1	0	0	1	0	$2^4 + 0 + 0 + 2^1 + 0 = 18$
S	19	1	0	0	1	1	$2^4 + 0 + 0 + 2^1 + 2^0 = 19$

There is, of course, a conflict as far as the first nine decimal digits are concerned. Does the value 7, for example, indicate a numerical value or the seventh letter, G? This is clarified by increasing the number of tracks in the tape from five to seven. Digits are indicated by additional holes being punched in both tracks five and six, while letters are indicated by holes punched in track 7.

The control system will require other characters as well as numbers and letters. For instance, a minus (−) sign may be necessary to indicate the direction of slide movement. These additional symbols have been allocated combinations of punched holes not used otherwise. Thus all the data required for CNC part programming purposes can be expressed via a seven-bit code. The term "bit," incidentally, is derived from BInary digiT.

Finally, to check the accuracy of the tape punching and tape reading, there is an eighth track referred to as a parity track. The ISO standard requires that each row contain an even number of holes. If the required character is expressed by an odd number of holes, an extra hole will be punched in track eight. If the required character is expressed by an even number of holes, there will be no extra hole in track eight. This system is referred to as "even parity." The EIA system also uses an eighth track, but as an end-of-block code. The fifth track is a parity punch to give odd parity in EIA-244-D standards. The newer EIA-358-B standard requires even parity and is becoming widely used in conjunction with computer applications. A visual check that each line of the tape contains an even number of holes (ISO) or an odd number of holes (EIA) is one method of ascertaining that there are no errors as a result of the equipment malfunctioning.

By reference to Figure 5.5 the reader can see the variations between the two tape standards. Most modern control systems will accept either standard EIA or ISO. (EIA standards 244 and 358 can be found in Appendix A.)

Tape Format Each horizontal row of holes in the tape is termed "character." Each set of characters is termed a "word." Each set of words is termed a "block." This is illustrated in Figure 5.6.

Blocks are identified by the letter N or O followed by three or four digits. A block can contain information on the type of slide movement required, length of slide movement, rate of slide movement, spindle speed, tool identity, etc., and will terminate with an "end of block" character.

The order in which words are entered in a block may be fixed or variable. The fixed block format requires each block to have the correct number of entries and they must appear in a set sequence. This means that data have to be reentered in each block even if there has been no change from the previous block and even if the numerical value of the entry is zero. On some systems each word has to be separated by the *tab* function, this type of format being referred to as *tab sequential*.

Much more commonly used is the variable block format in which words can be entered in any order. Their meaning or function is determined by the letter preceding the data, a system referred to as *word address*. Data that remain

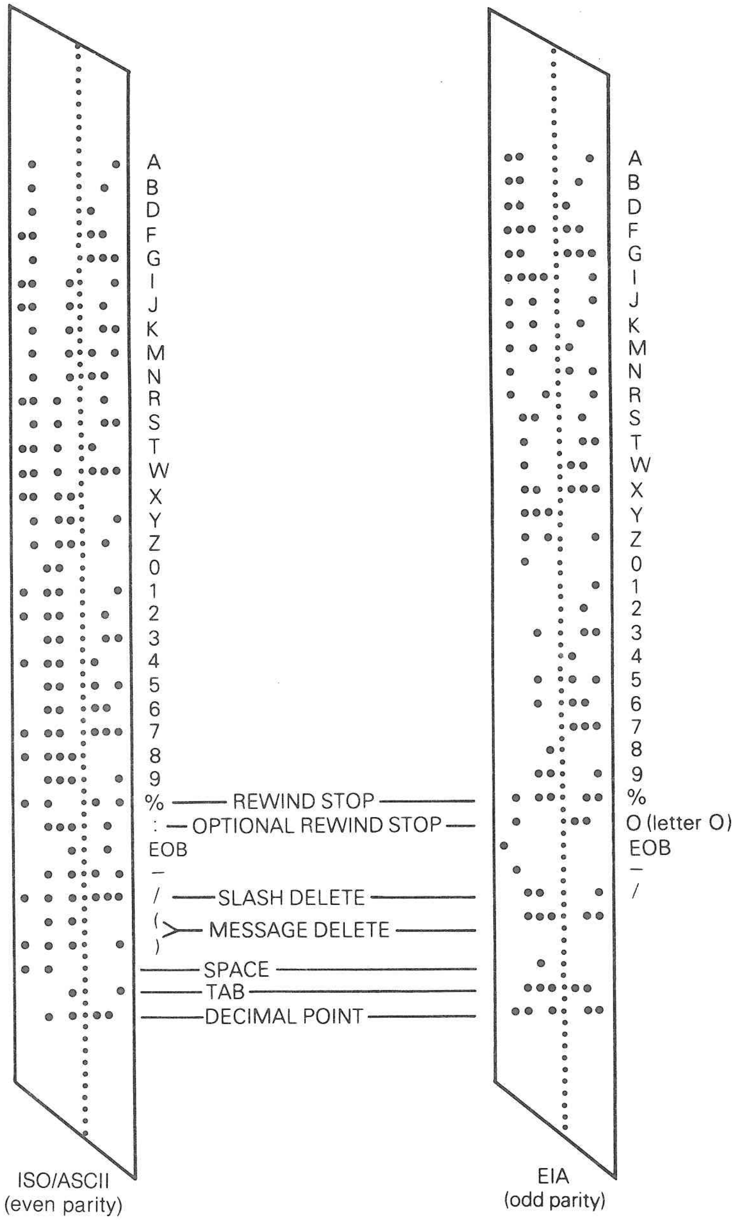


Figure 5.5 *Standard tape codes.*

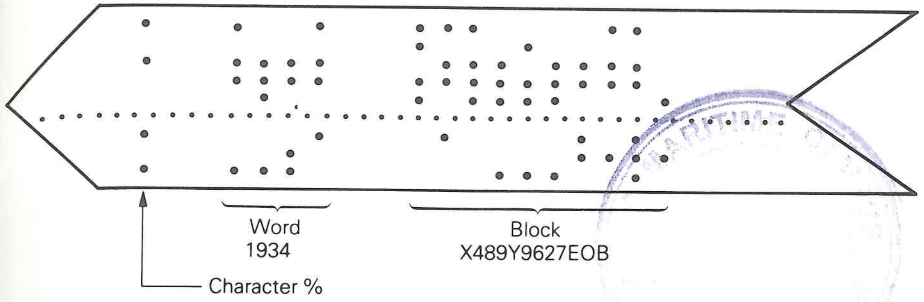


Figure 5.6 *Perforated tape format.*

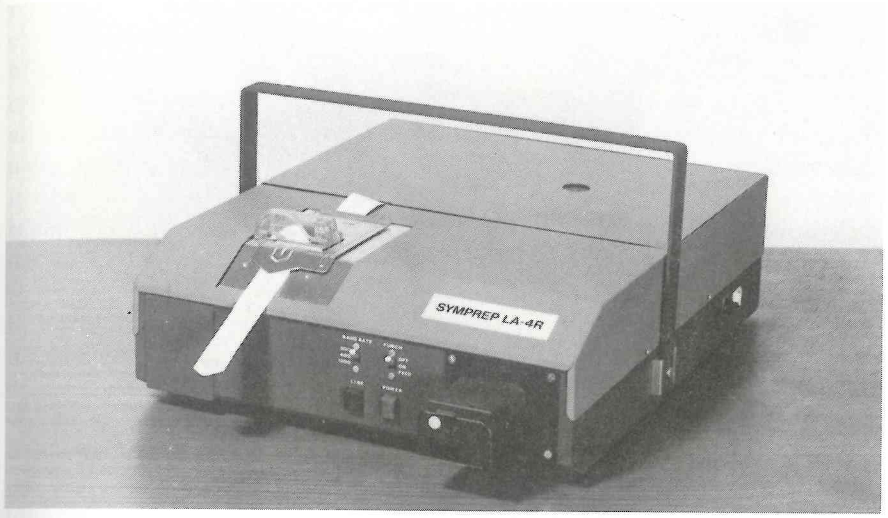


Figure 5.7 *Automatic tape punch for interfacing with computer.*

unchanged in following blocks need not be reentered, and this leads to more rapid programming and a considerable reduction in the resulting tape length.

Production of Punched Tape Punched tapes may be produced either from a teletypewriter or by an automatic tape punch interfaced (connected) with a computer. An automatic tape punch is shown in Figure 5.7.

The teletypewriter is similar in many ways to an electric typewriter. It has an alphanumeric (letters and numbers) keyboard, and the prewritten program is typed in the normal way either onto conventional teletype paper or onto a blank program sheet. A teletypewriter is shown in Figure 5.8.

Attached to the side of the teletypewriter is the punching device. The blank tape feeds automatically from a roll into the punching head and, as each character is typed on the keyboard, a row of holes is punched in the tape.

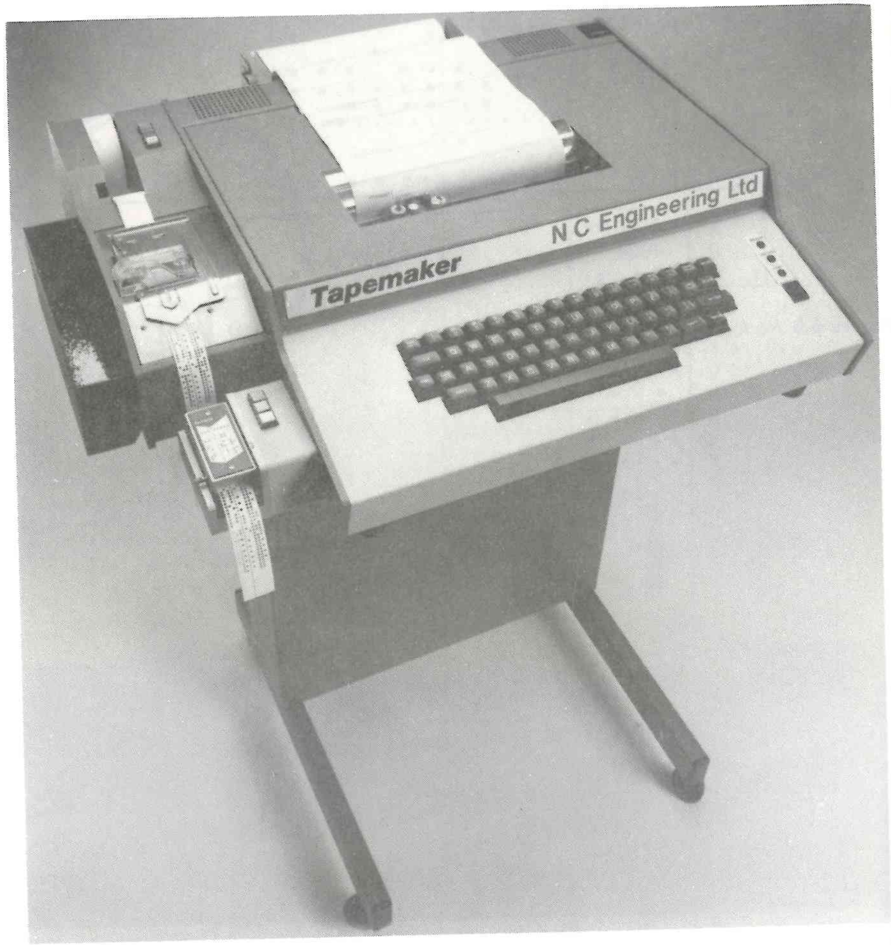


Figure 5.8 *Teletypewriter used for production of perforated tape.*

Additional copies of the tape or of the typewritten program can be made automatically.

The facility to produce additional copies can be used for tape correcting. The original tape is run through the machine, and while it is running through, a replica of the tape and a printout are being made. At the point where the correction is to be made the automatic process is stopped. The new data are then typed into the program in the normal way. When the correction is completed the original tape is inched forward to the point where the original entry is still valid and the automatic reproduction process is restarted.

With computer-linked tape preparation facilities, editing is somewhat simpler. The listed part program appears on the display screen, where it can readily be examined, and any necessary alterations can be made via the keyboard.

When the programmer is satisfied that all is correct, the tape punch is activated to produce a complete new tape. Additional copies of the tape can readily be made and an interfaced printer will provide copies for filing for future reference.

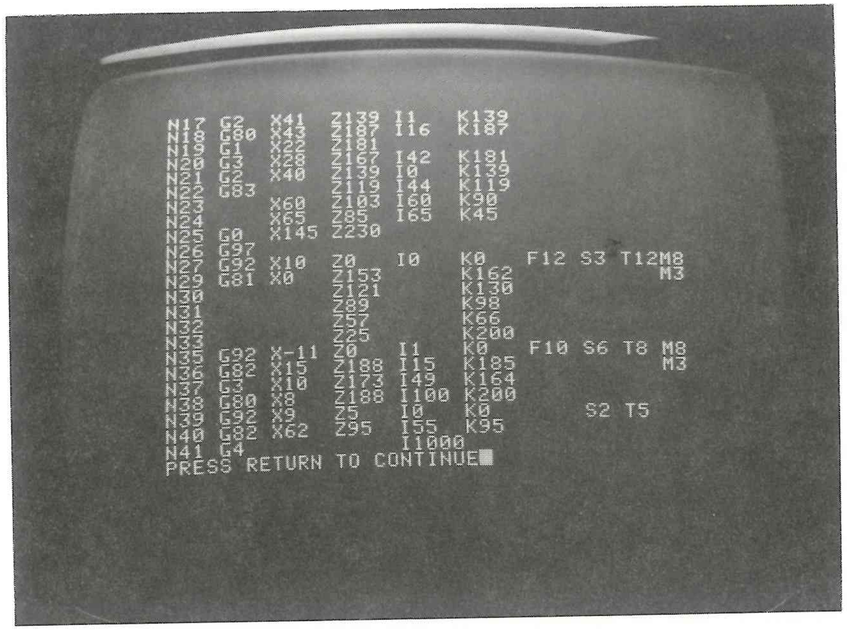
Tape Proving Before a punched tape can be used for machining, it should be "proved," that is, checked that the desired machine movements will be achieved. This can be done on the machine tool, although the wisdom of wasting valuable machining time in nonproductive testing is questionable.

On basic numerically controlled machines, that is, those not computer controlled, testing facilities are limited to a "dry run," where all machine feedrates are changed to rapid transverse. Owing to the inherent danger present here from the flip of a switch, no part is placed in the machine and tooling may also be removed. It should be noted that a dry run only checks for command format errors, and after machine setup, the program should be cycled through in a block by block or single cycle mode to check for physical errors. On computer controlled machines the dry run can be complemented by a "test run," that is, all axis and spindle movements are inhibited, but the visual display is continually updated as demanded by the program in real time, that is, the actual time it would take to machine the component. Errors in the program, like no spindle speed stated would be indicated by an appropriate error message appearing on the CRT screen.

To avoid incurring the nonproductive downtime referred to, other test facilities remote from the machine may be more appropriate; for example, a plotter may be used. A plotter is in effect an automatic drawing device. The profile of the cutter path is traced out by the machine according to the data supplied via the tape. The result is, of course, a "flat" view; depth, that is, the third axis, is achieved by using colors and different views if available. An interfaced printer may be used to provide a copy of the program and a drawing of the tool path.

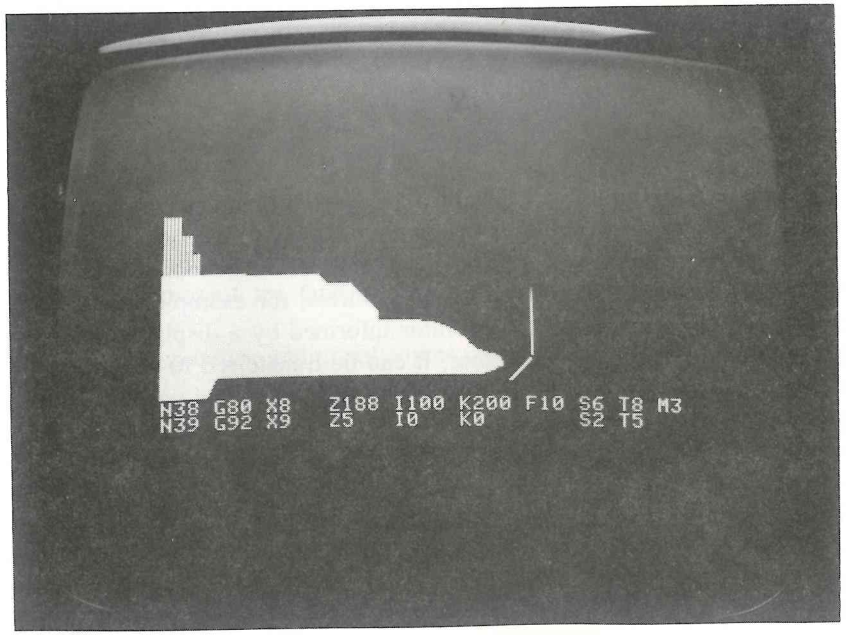
The tape-proving facilities referred to are largely redundant when computer graphics are used as part of the programming preparation process. The pre-prepared program is fed into the computer via keyboard, floppy disk, or tape, the entry appearing on the CRT. Incorrect entries, for example, an unrealistic feed rate, can be stalled and the operator informed by a displayed error message. When the program is complete, it can be transferred to storage and the computer graphics are then used to simulate a test run. The correct blank size appears and, using animated tool movements, is "machined" according to the program requirements. The use of a computer for program proving is illustrated in Figure 5.9. A computer printout of a program, together with a graphical representation of the component, is shown in Figure 5.10. It should be noted that graphic types of prove-out do not eliminate the need for a slowed down single step mode run through the program after each setup.

When the programmer is satisfied that the entry is correct, a tape can be produced very rapidly at the touch of a key via an automatic tape punch in-



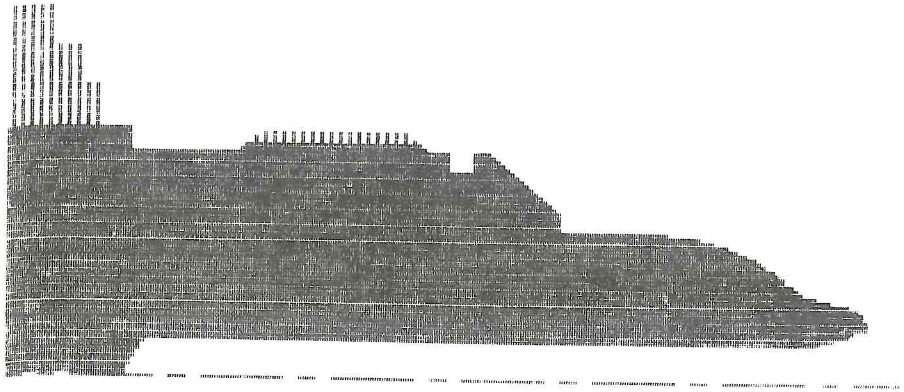
```
N17 G2  X41  Z139 I1   K139
N18 G80 X43  Z187 I16  K187
N19 G1  X22  Z181
N20 G3  X28  Z167 I42  K181
N21 G2  X40  Z139 I0   K139
N22 G83      Z119 I44  K119
N23     X60  Z103 I60  K90
N24     X65  Z85  I65  K45
N25 G0  X145 Z230
N26 G97
N27 G92 X10  Z0   I0   K0   F12 S3 T12M8
N29 G81 X0   Z153      K162          M3
N30          Z121      K130
N31          Z89       K98
N32          Z57       K66
N33          Z25       K200
N35 G92 X-11 Z0   I1   K0   F10 S6 T8 M8
N36 G82 X15  Z188 I15  K185          M3
N37 G3  X10  Z173 I49  K164
N38 G80 X8   Z188 I100 K200
N39 G92 X9   Z5   I0   K0       S2 T5
N40 G82 X62  Z95  I55  K95
N41 G4            I1000
PRESS RETURN TO CONTINUE█
```

Figure 5.9 (a) *Use of computer for program proving: program listing. (Metric output example for unidentified machine tool.)*



```
N38 G80 X8  Z188 I100 K200 F10 S6 T8 M3
N39 G92 X9  Z5   I0   K0       S2 T5
```

Figure 5.9 (b) *Use of computer for program proving: graphical simulation.*



```
Optional stop out.          Single step out.


        Time Taken = 11.93 min.
        =======================


PROGRAM USED TO GENERATE WORKPIECE
==================================
```

Figure 5.10 (a) *Computer printout and graphical representation of component in metric output form for unidentified machine tool.*

Figure 5.10 (b) Program used to generate workpiece.

```
N1   G96                                        S1000
N2   G95
N3   G92  X10  Z0    I0    K1    F99   S9  T1        M8
N4   G82  X61  Z188  I61   K91                       M3
N5   G81  X68  Z84   I68   K188                      M6
N6   G84  X56  Z188  I56   K109                      M3
N7        X51        I51   K114
N8        X46        I46   K119
N9   G0   X41
N10  G83       Z119  I45
N11  G81  X63  Z101  I63   K188
N12  G84  X36  Z188  I36   K159
N13       X31        I31   K165
N14       X27        I27   K170
N15  G82  X23        I23   K181
N16  G3   X29  Z167  I41
```

```
N17   G2    X41    Z139   I1      K139
N18   G80   X43    Z187   I16     K187
N19   G1    X22    Z181
N20   G3    X28    Z167   I42     K181
N21   G2    X40    Z139   I0      K139
N22   G83          Z119   I44     K119
N23         X60    Z103   I60     K90
N24         X65    Z85    I65     K45
N25   G0    X145   Z230
N26   G97
N27   G92   X10    Z0     I0      K0     F12   S3   T12   M8
N29   G81   X0     Z153           K162                    M3
N30                Z121           K130
N31                Z89            K98
N32                Z57            K66
N33                Z25            K200
N35   G92   X-11   Z0     I1      K0     F10   S6   T8    M8
N36   G82   X15    Z188   I15     K185                    M3
N37   G3    X10    Z173   I49     K164
N38   G80   X8     Z188   I100    K200
N39   G92   X9     Z5     I0      K0           S2   T5
N40   G82   X62    Z95    I55     K95
N41   G4                  I1000
N42   G80   X68           I68     K40
N43   G81   X60    Z40
N44   G85   X68    Z36    I60     K36
N45                Z31            K31
N46                Z27            K27
N47   G0           Z53
N48   G1    X60    Z45
N49   G80   X68    Z30    I100    K200
N50   G92   X5     Z3     I0      K1     F99   S5   T6
N51   G0    X66    Z100
N52   G34   X64    Z37            K50
N53         X63
N54         X62
N55   G0    X100   Z200                                   M2
```

terfaced with the computer. Similarly, an interfaced printer will produce a printout.

Magnetic Tape Data Input

Magnetic tape, in the form of cassettes, is a widely used means of transmitting data (Figure 5.11). The advantages claimed for it are:

(a) easier handling;
(b) more rapidly produced and read;
(c) the program can be erased and the tape re-used;
(d) simpler editing;
(e) more storage space than a paper tape of the same length;
(f) more durable than paper tapes.

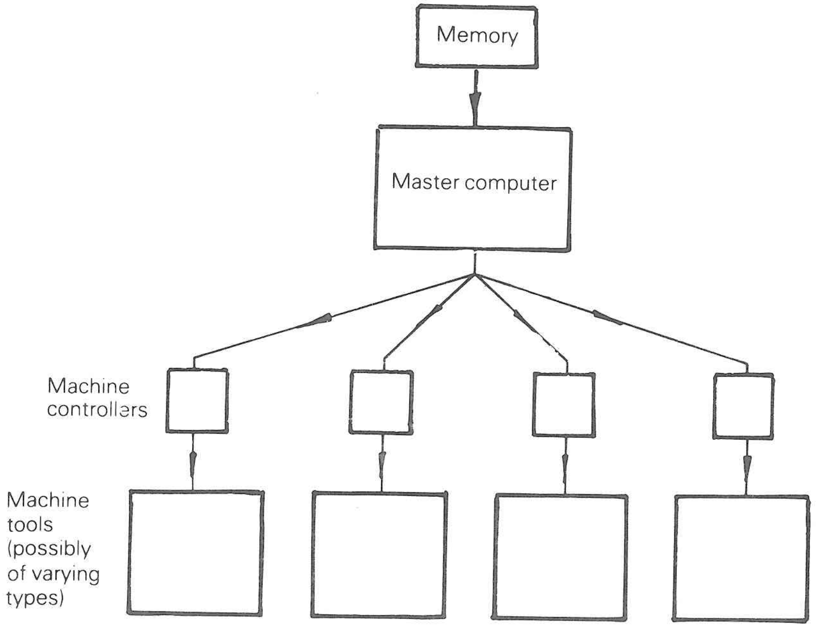


Figure 5.11 *Data input from master computer: direct numerical control (DNC).*

Early applications of magnetic tape involved a recording being made as machining of the first component was being carried out "manually" from the control console, a form of manual data input. A major drawback of this system was that the final program was only as rapid as the human reactions producing it and, with all the dial setting and switching involved, it was inevitably rather slow compared with modern techniques.

It was the advent of computer-controlled machining and computer-aided program proving that resulted in a more general application of magnetic tape. To record a part program in this way involves entering a program by MDI either at the machine control unit or through a computer keyboard. After the program has been entered it can be listed, edited and proved by using computer graphics, as discussed earlier. Finally, the program is recorded in much the same way as a tape recording of a piece of music can be made from a record or the radio.

Magnetic tape recordings do have the disadvantage of not being visible without the use of special CRT screens, printers, or plotters. Their other major disadvantage is that the magnetic information can be easily scrambled or destroyed if placed near magnetic fields or machine tools.

Data Input via Portable Electronic Storage Unit

There are problems associated with the use of paper and magnetic tape in what may be a dirty, and therefore potentially damaging, environment such as may be encountered in a machine shop. These problems may be eliminated by the use of a portable electronic storage unit. The data are first transferred into the unit away from the machine shop. The unit is then carried to the machine, connected to the machine control unit, and the data are then transferred. Similarly data already in the machine control unit may be transferred into the portable unit for downloading and storage elsewhere. Data transfer is very rapid. The capacity of the portable units is such that a number of programs can be accommodated at any one time.

Magnetic Disk Input via an Interfaced Computer

Providing the distance is not too great, it is possible to directly cable-link a microcomputer to a machine tool or, alternatively, to have a computer mounted on a trolley that can be brought alongside the machine and temporarily interfaced. It is then possible to transfer data stored on a magnetic diskette, commonly referred to as a "floppy disk," into the computer and hence into the machine control unit. Data already in the control unit can also be extracted and recorded. Newer CNC machine tool controls are now also appearing with portable or built in disk drives.

Disks have the same disadvantage as magnetic tape in that the program is not visible but, as with tapes, a program may be transferred into the computer for visual display, and printing and plotting facilities may be included.

The rate at which data can be transferred or retrieved using a disk is much faster than when using a tape and, size for size, the storage capacity of the disk is much greater.

Master Computer Data Input

An extension of the concept described previously of linking a computer to a machine tool is when a computer, usually a mini or mainframe, is permanently linked to a series of machines. Prepared programs stored in the memory of the master computer are then transferred to the microcomputer of the control unit of the selected machine tool as and when required. The concept is illustrated in Figure 5.11 and is referred to as direct numerical control (DNC).

Buffer Storage

Most modern control systems have a buffer storage, which is a capability to hold data extracted from the computer memory in an intermediate position. As one block of information is being processed the next is ready for instant transmission, the object being to speed the rate at which data are processed so that there is a minimum loss of machining time. Also the elimination of a dwell between blocks avoids marking the machined surface.

QUESTIONS

1 When would it be economically inadvisable to enter data manually into a machine control unit?

2 What is conversational manual input and what are the advantages and disadvantages of entering a machining program by this method?

3 Name three types of materials, or combinations of materials, used for perforated tape and state the advantages of each.

4 Name three types of tape readers. Which of these is most commonly used?

5 How is the reverse loading of perforated tape into a tape reader prevented?

6 Why is the binary system of numbers used to indicate the meaning of data input on a perforated tape?

7 What is the origin of the expression "eight bit" as applied to perforated tape?

8 Name the two tape standards in general use and explain how, by a visual check, you could identify them.

9 Explain what is meant by "character," "word," and "block" as applied to perforated tape format.

10 What is the difference between tab sequential and variable block tape format?

11 Describe two ways in which a tape can be proved away from the machine tool.

12 What is the difference between a dry run and a test run when checking data input?

13 What are the advantages of magnetic tape as a data storage medium?

14 What are the advantages of the floppy disk as a data storage edium?

15 What is a buffer storage and why is it necessary?

6

TERMS AND DEFINITIONS ASSOCIATED WITH PART PROGRAMMING AND MACHINE CONTROL

PART PROGRAMMING

The expression "part programming" causes some confusion, since "part" is often thought to mean something that is incomplete. In numerical control terms a part program is, in fact, a complete program. The word "part" means component.

PREPARATORY FUNCTIONS

Preparatory functions are used to inform the machine control unit of the facilities required for the machining that is to be carried out. For example, the control unit will need to know if the axis movements stated dimensionally in the program are to be made in inch or metric units, and whether the spindle is to rotate in a clockwise or counterclockwise direction.

The way in which machine controllers are provided with such information depends on the type of control unit. On conversational MDI systems, it may simply involve pressing the appropriate button on the control panel. For systems using the word address programming method, the various preparatory functions were originally standardized (ANSI/EIA RS274-D:1979; BS 3635:1972), each function being identified by the address letter G followed by two digits. Thus preparatory functions came to be referred to generally as "G codes." The Standard has been adopted and is widely used, although variations in the allocation of special G codes will be encountered.

The preparatory functions, as they appear in the Standard, are shown in Table 6.1. The codes used for any particular control system will depend on the machine type and the sophistication of the system and, although a complete list such as the original standard is rather extensive, it should be appreciated that the number of codes included in any one system will be considerably fewer in number.

Table 6.1 Preparatory functions codes (M = modal).

Code Number	Function	Modal[a]
G00	Rapid positioning, point to point	(M)
G01	Linear positioning at controlled feed rate	(M)
G02	Circular interpolation CW—two dimensional	(M)
G03	Circular interpolation CCW—two dimensional	(M)
G04	Dwell for programmed duration	
G05	Unassigned EIA code may be used as hold. Cancelled by operator	
G06	Parabolic interpolation	(M)
G07	Unassigned EIA code reserved for future standarization	
G08	Programmed slide acceleration	
G09	Programmed slide deceleration	
G10 G11 G12	Unassigned EIA code sometimes used for machine lock and unlock devices	
G13–G16	Axis selection	(M)
G17	*XY* plane selection	(M)
G18	*ZX* plane selection	(M)
G19	*YZ* plane selection	(M)
G20	Unassigned EIA code	
G21 G22 G23	Unassigned EIA code sometimes used for nonstop blended interpolation movements	
G24	Unassigned EIA code	
G25–G29	Permanently unassigned. Available for individual use	
G30 G31 G32	Unassigned EIA code	
G33	Thread cutting, constant lead	(M)
G34	Thread cutting, increasing lead	(M)
G35	Thread cutting, decreasing lead	(M)
G36–G39	Permanently unassigned. Available for individual use	
G40	Cutter compensation/offset, cancel	(M)
G41	Cutter compensation, left	(M)
G42	Cutter compensation, right	(M)
G43	Cutter offset inside corner	(M)
G44	Cutter offset outside corner	(M)
G45 G46 G47 G48 G49	Unassigned EIA code	
G50	Reserved for adaptive control	
G51	Cutter compensation +/0	
G52	Cutter compensation −/0	
G53	Linear shift cancel	(M)
G54	Linear shift *X*	(M)

Table 6.1 *(Continued)*

Code Number	Function	Modal[a]
G55	Linear shift *Y*	(M)
G56	Linear shift *Z*	(M)
G57	Linear shift *XY*	(M)
G58	Linear shift *XZ*	(M)
G59	Linear shift *YZ*	(M)
G60–G69	Unassigned EIA codes	
G70	Inch programming	(M)
G71	Metric programming	(M)
G72	Circular interpolation—CW (three dimensional)	(M)
G73	Circular interpolation—CCW (three dimensional)	(M)
G74	Cancel multiquadrant circular interpolation	(M)
G75	Multiquadrant circular interpolation	(M)
G76–G79	Unassigned EIA code	
G80	Fixed cycle cancel	(M)
G81	Fixed cycle 1	(M)
G82	Fixed cycle 2	(M)
G83	Fixed cycle 3	(M)
G84	Fixed cycle 4	(M)
G85	Fixed cycle 5	(M)
G86	Fixed cycle 6	(M)
G87	Fixed cycle 7	(M)
G88	Fixed cycle 8	(M)
G89	Fixed cycle 9	(M)
G90	Absolute dimension input	(M)
G91	Incremental dimension input	(M)
G92	Preload registers	
G93	Inverse time feedrate (*V*/*D*)	(M)
G94	Inches (millimeters) per minute feedrate	(M)
G95	Inches (millimeters) per revolution feedrate	(M)
G96	Constant surface speed, feet (meters) per minute	(M)
G97	Revolutions per minute	(M)
G98 G99	Unassigned EIA code	

[a] Function retained until cancelled or superceded by subsequent command of same letter.

Many preparatory functions are modal, that is, they stay in operation until changed or cancelled.

MISCELLANEOUS FUNCTIONS

In addition to preparatory functions there are a number of other functions that are required from time to time throughout the machining program. For example, coolant may be required while metal cutting is actually under way but will need to be turned off during a tool-changing sequence. Operations such as this are called "miscellaneous functions."

Conversational MDI control systems will, as with preparatory functions, have their own particular way of initiating miscellaneous functions, but for word address systems the EIA standards have been adopted except for special options on particular machine tools. The functions are referred to as "M functions" and are identified by the address letter M followed by two digits.

The original standardized miscellaneous functions are listed in Table 6.2. The functions available will vary from one control system to another, the number available being fewer than the complete list.

POSITIONING CONTROL

The basis of numerically controlled machining is the programmed movement of the machine slides to predetermined positions. This positioning is described in three ways:

(a) point-to-point
(b) line motion or linear interpolation
(c) contouring or circular interpolation.

Point-to-Point Positioning

Point-to-point positioning involves programming instructions that only identify the next position required. The position may be reached by movement in one or more axes. When more than one axis is involved, the movements are not coordinated with each other, even though they may occur simultaneously. The rate of movement is usually, although not necessarily, the maximum for the machine.

Figure 6.1 shows a component the machining of which would involve point-to-point positioning, the holes being drilled in the sequence A to D. Note that it is the positioning prior to drilling that is point-to-point, not the drilling operation itself.

Line Motion Control

Line motion control is also referred to as linear interpolation. The programmed movement results from instructions that specify the next required position and also the feed rate to be used to reach that position. This type of positioning would be involved in machining the slot in the component shown in Figure 6.2, the cutter moving in relation to the workpiece from point A to point B. Although a continuous cutter path appears to be the result, two distinct slide movements are involved, each slide movement being independent of the other.

Linear interpolation was initially defined as slide movement at programmed

Table 6.2 Miscellaneous functions codes.

Code number	Function	Function Starts Relative To Commanded Motion In Its Block: With	After Completion	Modal[a]
M00	Program stop		X	
M01	Optional stop		X	
M02	End of program		X	
M03	Spindle on CW	X		X
M04	Spindle on CCW	X		X
M05	Spindle off		X	X
M06	Tool change			
M07	Coolant 2 on	X		X
M08	Coolant 1 on	X		X
M09	Coolant off		X	X
M10	Clamp			X
M11	Unclamp			X
M12	Synchronization code		X	X
M13	Spindle on CW, coolant on	X		X
M14	Spindle on CCW, coolant on	X		X
M15	Motion in the positive direction	X		
M16	Motion in the negative direction	X		
M17 M18	Unassigned EIA code. Reserved for future standardization			
M19	Oriented spindle stop			X
M20–M29	Permanently unassigned. Available for individual use			
M30	End of tape/data		X	X
M31	Interlock bypass	X		X
M32–M35	Unassigned EIA code			
M36 M37 M38 M39	Permanently unassigned. Available for individual use			
M40–M46	Gear changes if used otherwise unassigned	X		X
M47	Return to program start			
M48	Cancel M49	X		X
M49	Feed/speed bypass override	X		X
M50–M57	Unassigned EIA code			
M58	Cancel M59	X		X
M59	Bypass constant surface speed updating	X		X
M60–M89	Unassigned EIA code			
M90–M99	Reserved for user			

[a] Function retained until cancelled or superceded by subsequent command of same letter.

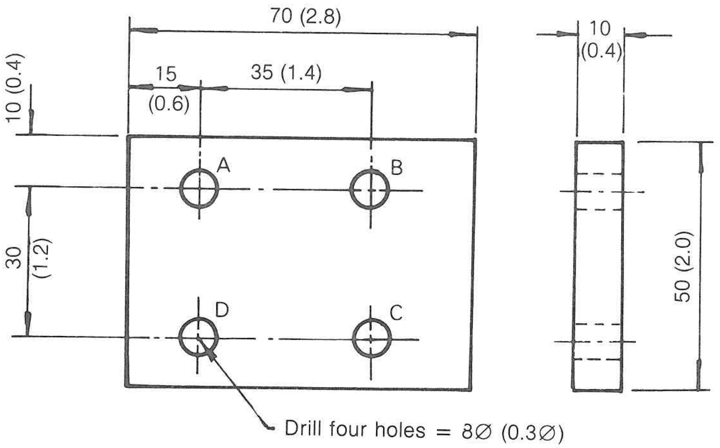


Figure 6.1 *Component detail involving point-to-point positioning. (Inch units are given in parentheses.)*

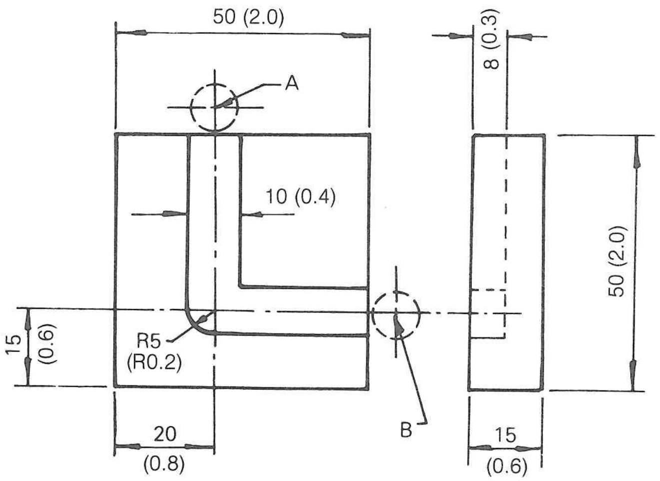


Figure 6.2 *Component detail involving line motion control. (Inch units are given in parentheses.)*

feed rates parallel to the machine axes. More recently it has also been used to describe linear movement when two, or sometimes three, slides are moving at the same time at programmed feed rates, a facility not available on earlier control systems. When two slides are moving simultaneously, an angular tool path results, and when three slides are involved, the result would be as indicated in Figure 6.3.

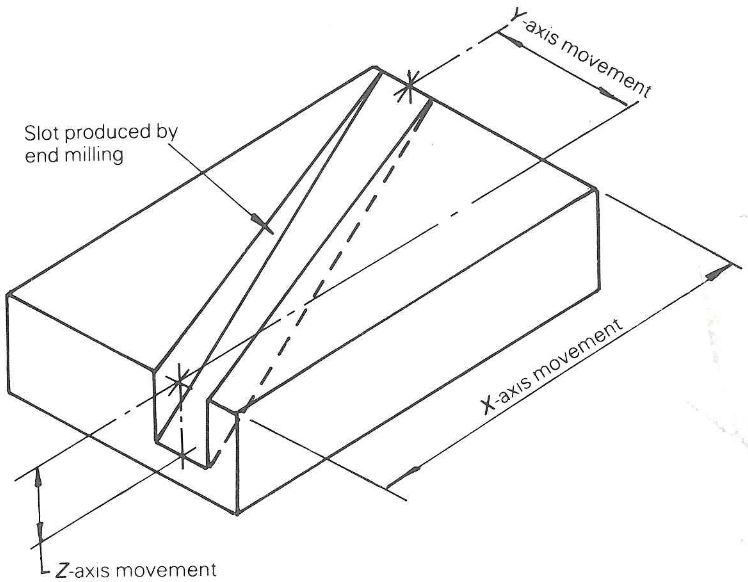


Figure 6.3 *Resulting tool path when three slides move simultaneously.*

Contouring

Contouring also involves two or more controlled slide movements resulting from program data that specify the next position required and the required feed rates to reach that position, so there is some overlap between linear interpolation and contouring. However, contouring can also be much more complex, involving combinations of angular movement and curves with one feature moving without interruption in the cutting process into another. This type of movement gives rise to the expression continuous path machining, which is often used to describe contouring.

Machining of the elliptical profile shown in Figure 6.4 would involve continuous path movement. Likewise, the radii shown on the components in Figure 6.5 would be produced in a similar manner. The elliptical shape is not readily defined in numerical terms, and to produce the necessary cutter path would present an interesting, although not insurmountable, problem to the part programmer unless the control system was specially equipped with a canned cycle to deal with such a situation. On the other hand, the two radii shown in Figure 6.5 are an everyday occurrence and most control systems can readily accommodate the production of a radius, or a combination of radii. Such a facility is referred to as circular interpolation.

Circular arcs may be programmed in the *XY*, *XZ*, and *YZ* planes. In exceptional cases three axes may be involved, resulting, in effect, in a helical tool path.

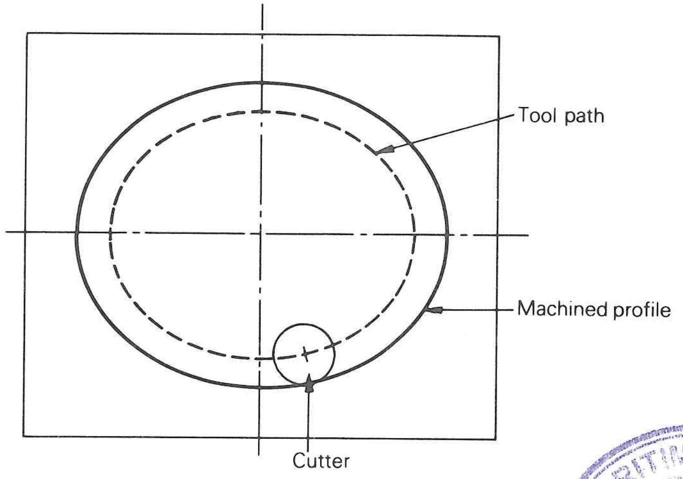


Figure 6.4 *Component profile produced by contouring.*

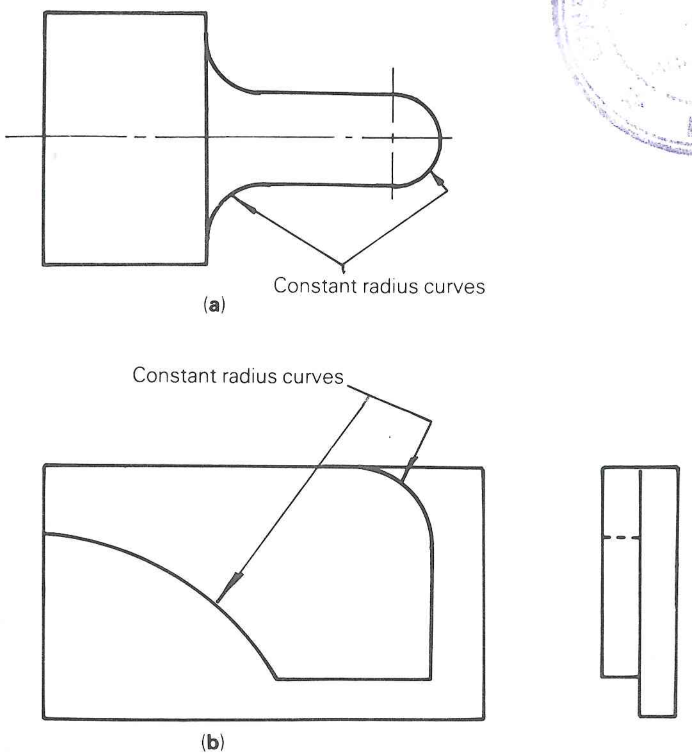


Figure 6.5 *Components with radial features requiring circular interpolation:* (a) *turned component and* (b) *milled profile.*

PROGRAMMING POSITIONAL MOVES

In practice the three types of positioning referred to previously are rarely isolated. The production of the majority of components will involve a combination of the techniques. However, it will be necessary to clearly identify in the part program the type of positioning required at each stage of the machining process.

Manual data input systems will vary from one control system to another. For example, a widely used training machine specifies all linear movement as linear interpolation and differentiates by linking the movement to an appropriate feed rate. The program entry is reduced to pressing a linear interpolation key followed by the dimensional detail and the feed rate. Similarly, a radius is simply defined by pressing a circular interpolation key, followed by a data entry of the dimensional value of the target position, the radius, and the direction of rotation as either clockwise or counterclockwise.

Control systems using the recommendations contained in EIA RS-274-D or BS 3635:1972 will specify the type of positioning involved by using the appropriate preparatory function or G code, the common ones being as follows:

G00	Point-to-point
G01	Linear interpolation
G02	Circular interpolation clockwise
G03	Circular interpolation counterclockwise

Having defined the type of positioning in this way the instruction is completed by including dimensional details of the move together with the feed rate for G01, G02, and G03. G00 moves are usually made at the maximum slide traverse rate for the machine.

DIMENSIONAL DEFINITIONS OF SLIDE MOVEMENT

In Chapter 1 it was explained that the axes in which slide movement can take place are designated by a letter and either a plus (+) or minus (−) sign to indicate the direction of movement. Unfortunately, these designated slide movements, owing to the different design configurations of machine tools, do not always coincide with the movement of the tool in relation to the work, and as a result this can cause some confusion when slide movements are being determined. In the case of a turning center with a conventional tool post there is no problem, since the slide movement and the tool movement in relation to the work are identical. But on a vertical machining center, for example, to achieve a positive (+) movement of the tool in relation to the work, the table, not the cutter, has to move, and this movement is in the opposite direction. Since a move in the wrong direction, especially at a rapid feed rate, could have disastrous results, this fact should be clearly understood.

A sound technique when determining slide movements is to program the tool movement in relation to the work. In other words, on all types of machines, imagine it is the tool moving and not, as is sometimes the case, the workpiece. To do this it is necessary to redefine some, but not all, of the machine movements. A simple diagram such as the one alongside the components shown in Figures 6.6 and 6.9 is usually very helpful.

Once the direction of movement has been established it will need to be dimensionally defined. There are two methods used, and they are referred to as:

(a) absolute;
(b) incremental.

Figure 6.6 shows the profile of a component to be machined on a turning center using the machine spindle center line and the face of the workpiece as datums in the *X* and *Z* axes respectively. Assume the sequence of machining is to commence with the 1.4 in. (35 mm) diameter, followed by the 1.2 in. (30 mm) diameter and finishing with the 1 in. (25 mm) diameter.

To machine the profile using absolute dimensions, it is necessary to relate all the slide movements to a preestablished datum. The movements required in absolute terms are indicated in Figure 6.7. Note that all position commands are the actual distance that the tool tip is from the datum point.

Incremental positioning involves relating the slide movement to the final position of the previous move. The slide movements, expressed in incremental terms, which would be necessary to machine the profile are indicated in Figure 6.8. Note position commands indicate the direction and the exact amount of slide motion required.

Note that each dimension in the *X* axis in Figure 6.7 is equal to the work

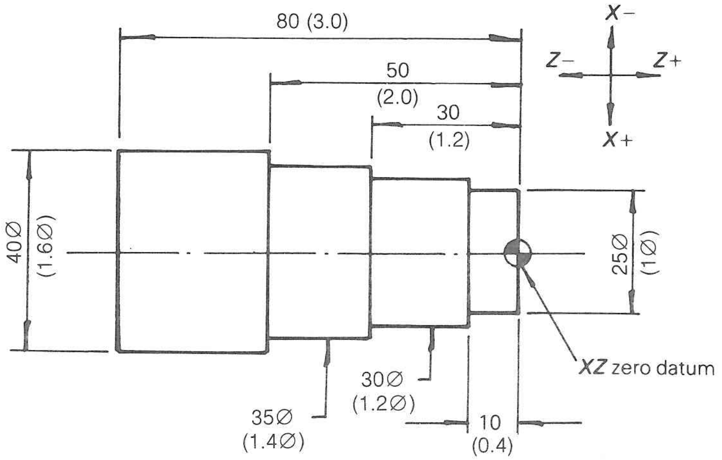
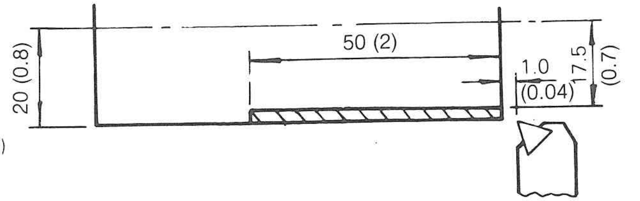


Figure 6.6 *Component detail. (Inch units are given in parentheses.)*

(a)

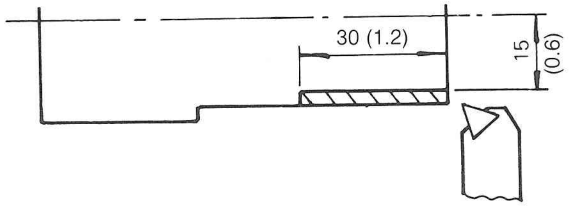


All cuts start 1 mm (0.04 in.) from face

Axis movements

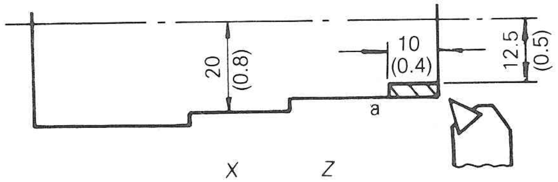
	X	Z
Feed in	17.5 (0.7)	1.0 (0.04)
Turn 35∅ (1.4∅)	17.5 (0.7)	−50.0 (−2)
Withdraw	18.0 (0.72)	−50.0 (−2)
Return	18.0 (0.72)	1.0 (0.04)

(b)



	X	Z
Feed in	15.0 (0.6)	1.0 (0.04)
Turn 30∅ (1.2∅)	15.0 (0.6)	−30.0 (−1.2)
Withdraw	15.5 (0.62)	−30.0 (−1.2)
Return	15.5 (0.62)	1.0 (0.04)

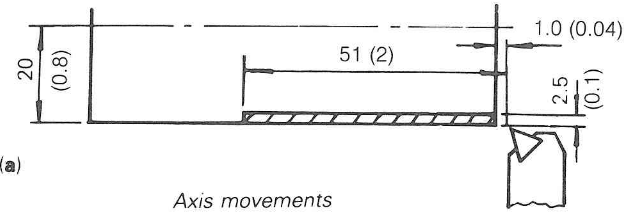
(c)



	X	Z
Feed in	12.5 (0.5)	1.0 (0.04)
Turn 25∅ (1∅)	12.5 (0.5)	−10.0 (−0.4)
Withdraw	20.0 (0.8)	−10.0 (−0.4)
Return	20.0 (0.8)	1.0 (0.04)

Figure 6.7 *Turning using absolute positioning. (Inch units are given in parentheses.)*

(a)

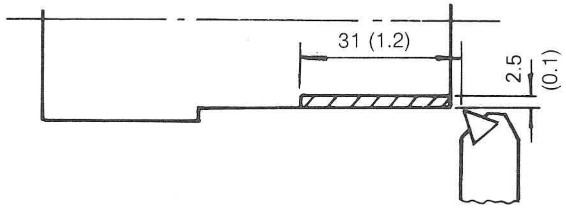


All cuts start 1 mm (0.04 in.) from face

Axis movements

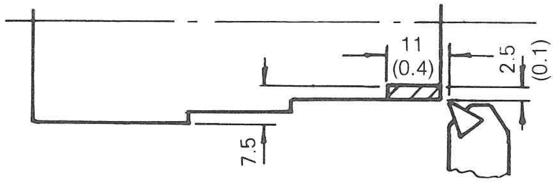
	X	Z
Feed in	−2.5 (−0.1)	0
Turn 35∅ (1.4∅)	0	−51.0 (−2)
Withdraw	0.5 (0.02)	0
Return	0	51.0 (2)

(b)



	X	Z
Feed in	−3.0 (−0.12)	0
Turn 30∅ (1.2∅)	0	−31.0 (−1.2)
Withdraw	−0.5 (0.02)	0
Return	0	31.0 (−1.2)

(c)



	X	Z
Feed in	−3.0 (−0.12)	0
Turn 25∅ (1∅)	0	−11.0 (−0.4)
Withdraw	7.5 (0.3)	0
Return	0	11.0 (0.4)

Figure 6.8 *Turning using incremental positioning. (Inch units are given in parentheses.)*

radius. When turning, some control systems will require dimensions in the *X* axis to be stated as a diameter, other machines may allow the programmer to select radius or diameter programming.

Figure 6.9 shows a component that is to be milled in the sequence A to C on a vertical machining center using datums as indicated. Assume that the movement in the *Z* axis to give a slot depth of 0.4 in (10 mm) has already been made. The necessary slide movements in the *X* and *Y* axes in absolute and incremental terms are indicated in Figures 6.10 and 6.11, respectively.

On the more sophisticated control systems, it is possible to use absolute and incremental dimensional definition within the same program, the distinction being achieved by using the G91 preparatory function code when the switch from absolute (G90) to incremental (G91) is to be made.

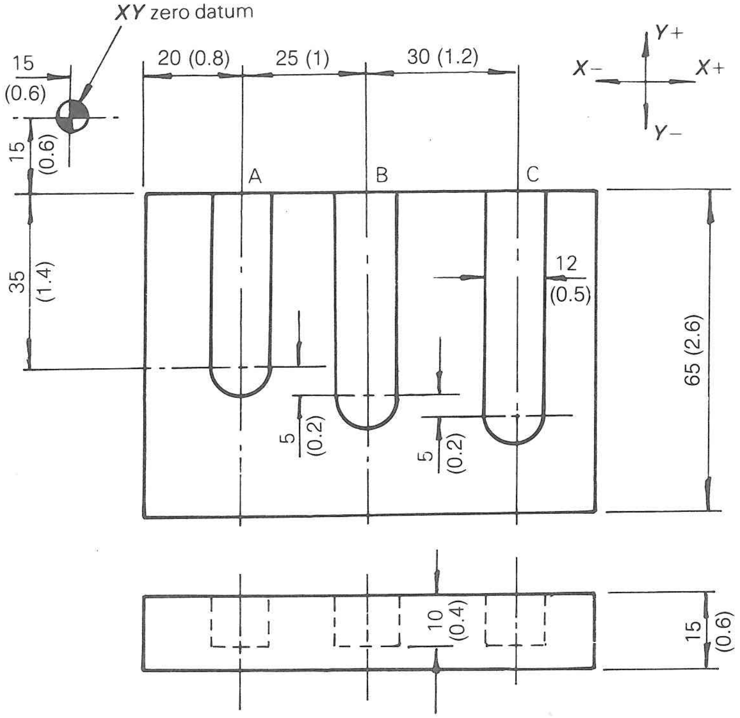


Figure 6.9 *Component detail. (Inch units given in parentheses.)*

CIRCULAR INTERPOLATION

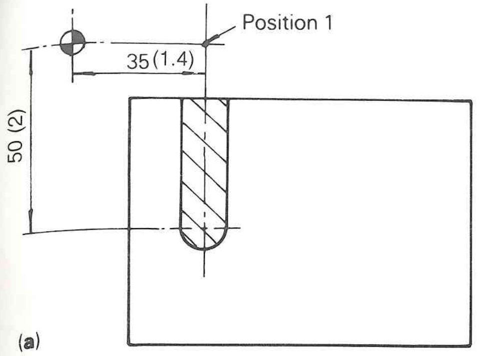
It was stated earlier that circular arc programming, particularly on conversational data input systems, has been reduced to simply dimensionally defining the target position, the radius, and the direction in which movement is to take place. On control systems using the word address format, it is rather more complex and there are slight variations in approach. Two of these variations will be considered later.

Common to all systems used to program circular movement is the need to determine whether the relative tool travel is in a clockwise (CW) or counterclockwise (CCW) direction. The following approach is usually helpful.

1. For milling operations look along the machine spindle toward the surface being machined.
2. For turning operations look on to the top face of the cutting tool. (For inverted tooling this involves looking at the tool from below.)

The standard G codes for circular interpolation are G02 (CW) and G03 (CCW). However, not all systems adopt this recommendation and there is at least one widely used system in which they are reversed, that is, G02 is CCW and G03

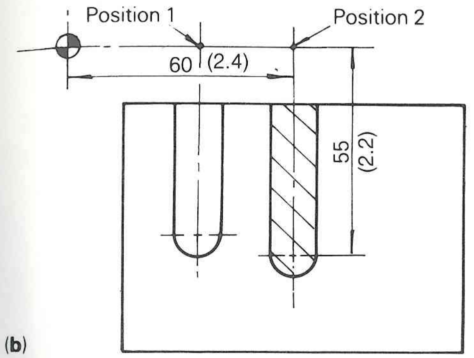
Figure 6.10 *Milling using absolute positioning. (Inch units are given in parentheses.)*



(a)

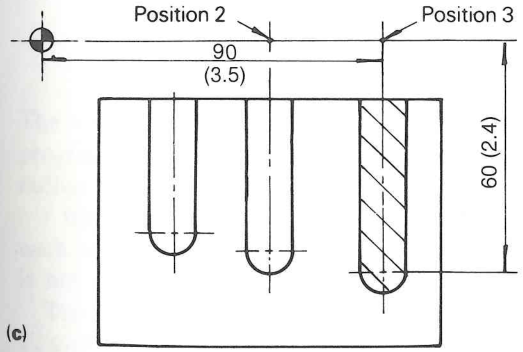
Axis movements

	X	*Y*
Move from datum to position 1	35.00 (1.4)	0
Mill to length	35.00 (1.4)	−50.00 (−2.00)
Return to position 1	35.00 (1.4)	0



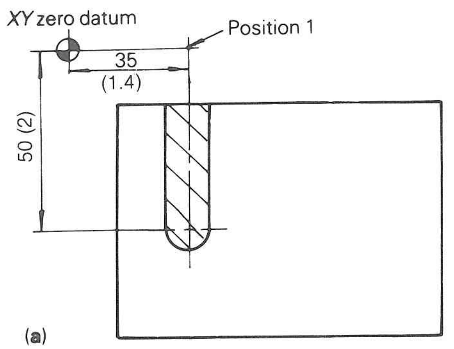
(b)

	X	*Y*
Move from position 1 to position 2	60.00 (2.4)	0
Mill to length	60.00 (2.4)	−55.00 (−2.2)
Return to position 2	60.00 (2.4)	0



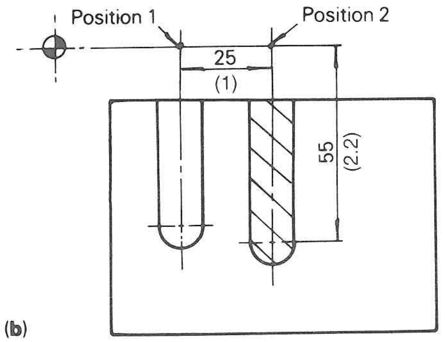
(c)

	X	*Y*
Move from position 2 to position 3	90.00 (3.5)	0
Mill to length	90.00 (3.5)	−60.00 (−2.4)
Return to position 3	90.00 (3.5)	0
Return to datum	0	0

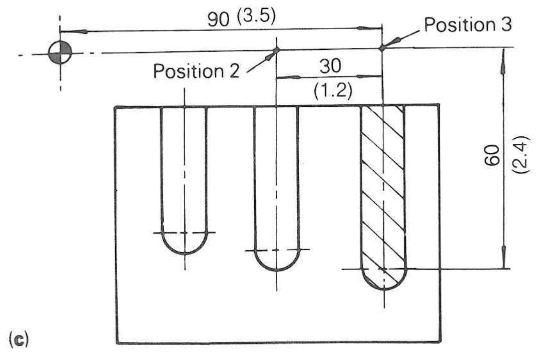


Axis movements

	X	Y
Move from datum to position 1	35.00 (1.4)	0
Mill to length	0	−50.00 (−2.00)
Return to position 1	0	50.00 (2.00)



	X	Y
Move from position 1 to position 2	25.00 (1.00)	0
Mill to length	0	−55.00 (−2.2)
Return to position 2	0	55.00 (2.2)



	X	Y
Move from position 2 to position 3	30.00 (1.2)	0
Mill to length	0	−60.00 (−2.4)
Return to position 3	0	60.00 (2.4)
Return to datum	−90.00 (−3.5)	0

Figure 6.11 *Milling using incremental positioning. (Inch units are given in parentheses.)*

is CW. (In this case it is advisable to refer to the machine tool programming manual.)

The three variations in arc programming referred to above are as follows. Note: That machines will normally not have all three methods of circular arc programming.

Method 1

Assuming that the last programmed move brought the cutting tool to the start point, the arc is defined in the following manner:

1. The finish or target point of the arc is dimensionally defined in relation to the start point using the appropriate combination of *X*, *Y*, and *Z* dimensional values stated in absolute or incremental terms.
2. The center of the arc is dimensionally defined in relation to the start point using *I*, *J*, and *K* values measured along the corresponding *X*, *Y*, and *Z* axes respectively.

Thus the arc shown in Figure 6.12 would be programmed as follows. In absolute terms using diameter programming:

		X	Z	I	K
Inch	G02	1.6	2.0	0	0.8
Metric	G02	40	50	0	20

In incremental terms:

		X	Z	I	K
Inch	G02	0.8	−0.8	0	0.8
Metric	G02	20	−20	0	20

The variation in the *X* values in these two examples is because the absolute program assumes that *X* values are programmed as a diameter rather than a radius.

I has no value because the center and start point of the arc are in line with each other in relationship to the *X* axis. In practice, when a value is zero, it is not entered in the program.

The *I*, *J*, and *K* values are always positive, with *I* related to *X*, *J* related to *Y*, and *K* related to *Z*.

Complete circles and semicircles are programmed as a series of 90° quadrants in many cases. Thus a complete circle would require four lines of program entry. New pieces of equipment can now complete full circles in one line of program entry.

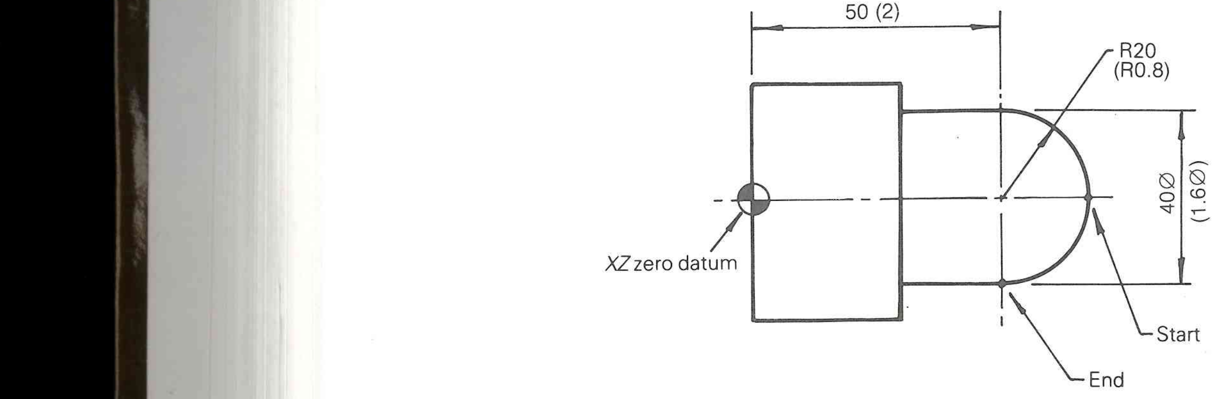


Figure 6.12 *Turned component detail involving arc programming. (Inch units are given in parentheses.)*

Figure 6.13 shows the program for a milled profile. The cutter radius has been ignored.

In absolute terms:

Inch	G	*X*	*Y*	*I*	*J*
	03	2	−1.2	1.2	0
	02	3.5	−1.2	0	1.6
Metric	G	*X*	*Y*	*I*	*J*
	03	50	−30	30	0
	02	90	−70	0	40

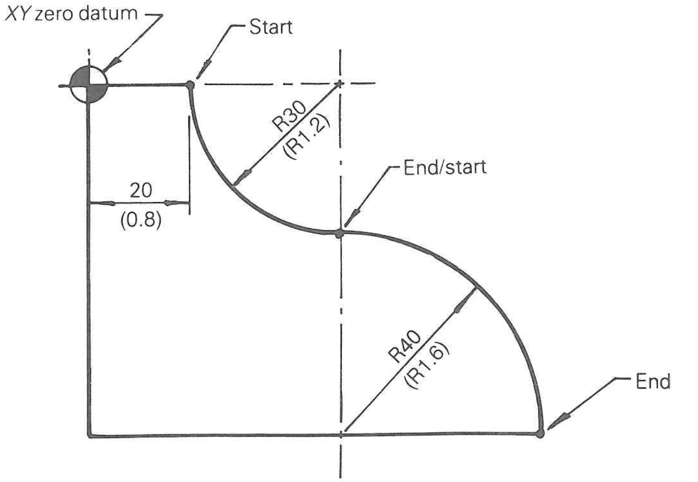


Figure 6.13 *Milled component involving arc programming. (Inch units are given in parentheses.)*

In incremental terms:

Inch	G	*X*	*Y*	*I*	*J*
	03	1.2	−1.2	1.2	0
	02	1.6	−1.6	0	1.6
Metric	G	*X*	*Y*	*I*	*J*
	03	30	−30	30	0
	02	40	−40	0	40

There are often situations where the start and/or stop points do not coincide with an X, Y, or Z axis, and it is then necessary to make a series of calculations. Such a situation is shown in Fig. 6.14. Dimensional values for X, Y, I, and J have to be determined. The necessary trigonometry is indicated in Fig. 6.15.

From A to B the magnitude of the X move is

Inch $1 \times \cos 30^\circ - 1 \times \cos 75^\circ = 0.866 - 0.259 = 0.607$
Metric $25.00 \cos 30^\circ - 25.00 \cos 75^\circ = 21.65 - 6.47 = 15.18$

From A to B the magnitude of the Y move is

Inch $1 \times \sin 75^\circ - 1 \times \sin 30^\circ = 0.966 - 0.500 = 0.466$
Metric $25.00 \sin 75^\circ - 25 \sin 30^\circ = 24.15 - 12.50 = 11.65$

The magnitude of the I dimension in the X axis is

Inch $1 \times \cos 75^\circ = 0.259$
Metric $25 \cos 75^\circ = 6.47$

The magnitude of J in the Y axis is

Inch $1 \times \sin 75^\circ = 0.966$
Metric $25.00 \sin 75^\circ = 24.15$

Once the dimensions have been incorporated, they are incorporated in the program as before.

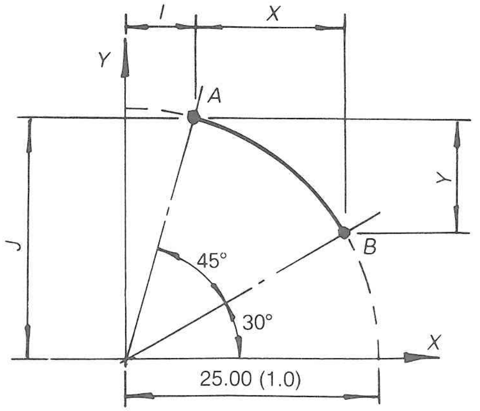


Figure 6.14 *Partial arc programming. (Inch units are given in parentheses.)*

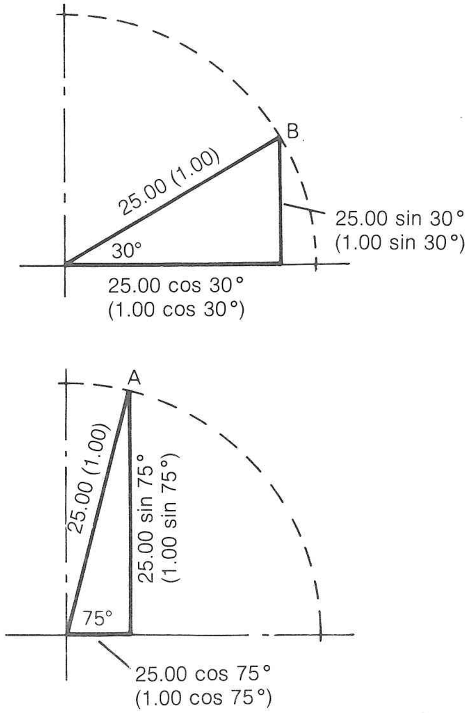


Figure 6.15 *Trigonometry required to program a partial arc. (Inch units are given in parentheses.)*

Method 2

The second method of arc programming varies from the one previously described in the way in which the arc center is defined. As in the previous method, it will be assumed that the cutting tool has arrived at the start point of the curve. To continue, the following data are required.

1. The finish or target point of the arc is dimensionally defined in relation to the start point using the appropriate combination of *X*, *Y*, and *Z* values stated in absolute or incremental terms.
2. The center of the arc is dimensionally defined in relation to the program datum using *I*, *J*, and *K* values measured along the corresponding *X*, *Y*, and *Z* axes respectively.

Using this method the arc shown in Fig. 6.12 would be programmed as follows.

In absolute terms:

	G	*X*	*Z*	*I*	*K*
Inch	02	1.6	2.0	0	2.0
Metric	02	40	50	0	50

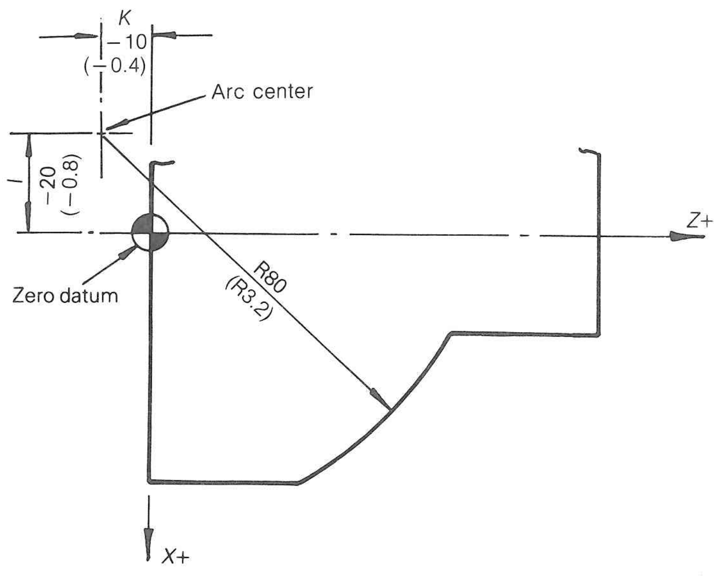


Figure 6.16 *Negative I and K values. (Inch units are given in parentheses.)*

In incremental terms:

	G	*X*	*Z*	*I*	*K*
Inch	02	0.8	−0.8	0	2.0
Metric	02	20	−20	0	50

Note that in this example it is *I* that has no value, since the center of the arc lies on the *X* datum and therefore *I* would be omitted from the program.

When the arc center is related to the program datum it is possible for the *I*, *J*, and *K* values to be a negative quantity, as illustrated in Figure 6.16.

The programming methods referred to above concern arcs of up to 90°. Some of the more modern control systems permit programming of arcs in excess of 90° in one data block, a facility referred to as 'multi-quadrant' programming or 360° circular interpolation.

Method 3

The third method of arc programming on some controls is to use absolute or incremental polar coordinates. It varies from the previous methods in that it does not use *I* and *J* values. With this method the circle center point has been defined previously with *X*, *Y*, or *Z* values. The arc is then programmed with a radius dimension and an angular amount of tool path from the circle center. A positive or negative angle will establish the direction of the cutter path:

1. The circle center is established with absolute or incremental dimensions.

2. The tool will have been moved to the arc start point.
3. The degrees of arc and radius are then programmed, with the sign (+ or −) on the degrees of arc establishing direction of cut.

Using the above terms the arc in Figure 6.12 would be programmed as follows:

In absolute terms:

Inch	CC X0 *Z*2	Define circle center
	G1 *X*0 *Z*2.8	Position cutter to starting point
	C Polar radius 0.8 polar angle 90°	Cut circle
Metric	CC X0 *Z*50	Define circle center
	G1 *X*0 *Z*70	Position cutter to start point
	C Polar radius 20 polar angle 90°	Cut circle

Positive angles denote clockwise motion; negative angles denote counterclockwise motion. Note: Most machines with polar coordinate circular interpolation capabilities are conversationally controlled.

In incremental terms:

Inch	CC *X*0 *Z*2	Circle center from datum
	G1 *X*0 *Z*2.8	Position tool to start from circle center
	C Polar radius 0.8 polar angle 90°	Circular movement
Metric	CC *X*0 *Z*50	Circle center from datum
	G1 *X*0 *Z*20	Position tool to start from circle center
	C Polar radius 20 polar angle 90°	Circular movement

RAMP

The starting and stopping of slide servo motors appear to be instantaneous. In fact there is, of course, a brief period of acceleration at the start of a move and a brief period of deceleration at the end of a move. This is shown graphically in Figure 6.17.

The period of acceleration is known as "ramp up" and the period of deceleration as "ramp down." The ramp is a carefully designed feature of the servo motor.

From a metal-cutting point of view, the quicker a slide attains its correct feed rate the better, and ideally this should be maintained throughout the cut. The ramp period therefore is kept as brief as possible, but consideration has to be given to ensuring that at the end of the movement there is no motor overrun or oscillation, both of which could affect the dimensional accuracy of the component.

For linear interpolation the ramp effect is rarely of concern, but for circular interpolation, and particularly where one curve runs into another, it is preferable that there is no speed variation of the servo motor, and thus of the feed rate of the slide, however small this might be. Any such variation would not only affect the metal-removal rate but may also affect the dimensional accuracy and surface finish of the component. Because of this, many control units are equipped with a *ramp inhibit* or *ramp suppression* facility, which means there is no slowing down or acceleration of the slide movement as one programmed movement leads into a second. G codes allocated to ramp are usually G08 and G09.

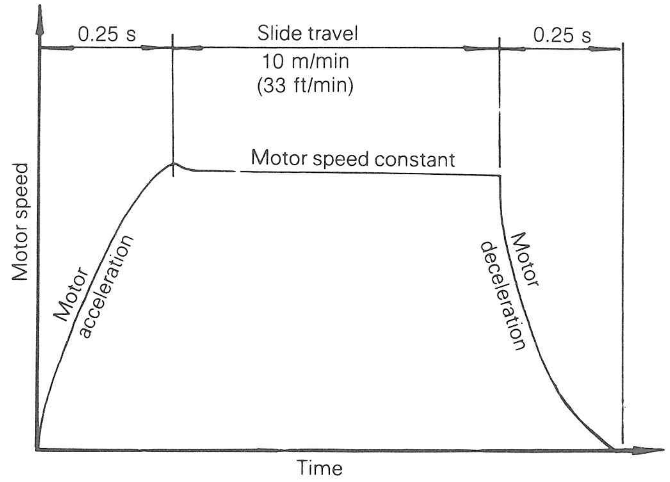


Figure 6.17 *Servo motor speed/feed rate relationship.*

REPETITIVE MACHINING SEQUENCES

There are a number of machining sequences that are commonly used when machining a variety of components. Other less common sequences may be repetitive, but only on one particular component. It is helpful, since it reduces the program length, if such a sequence can be programmed just once and given an identity so that it can be called back into the main program as and when required. Such sequences are referred to in a variety of ways, for example, as cycles, subroutines, loops, patterns, and macros. Although this can be slightly confusing, there are instances when one particular title appears to be more appropriate than the others. Various types of repeat machining sequences are discussed here.

Standardized Fixed Cycles

A number of the basic machining sequences, or cycles, commonly used were initially standardized (ANSI/EIA RS-274-D:1979; BS 3635:1972). The recommendations were commonly adopted and continue to be employed today. The machining cycles are identified by assigned G codes, and when they are incorporated into a control system, they are referred to as "fixed" or "canned" cycles. Perhaps the most commonly used fixed cycle is that of drilling a hole. Consider the hole shown in Figure 6.18(a). The sequence of machine movements involved in drilling the hole would be:

1. Position to hole location.
2. Lower the spindle at a programmed feed rate.
3. Lift the spindle rapidly to the start position.

Now consider the process of drilling the hole shown in Figure 6.18(b). The same sequence of spindle movements is necessary; the only variation is in the depth of travel. To program such a sequence of moves is quite simple, but if there were a large number of holes to be drilled, apart from the boredom of repeating the necessary data when writing the program, the program itself would be very long. In addition, the fewer data commands that have to be handled the less likely it is that errors will be made. By standardizing the sequence of moves the only additional data requirements are the new hole location, depth of cut, feed rate, and spindle speed. This information, with the appropriate G code, is entered only once. Each time the slide moves to bring the spindle to a new position in relation to the work another hole is drilled to the programmed depth.

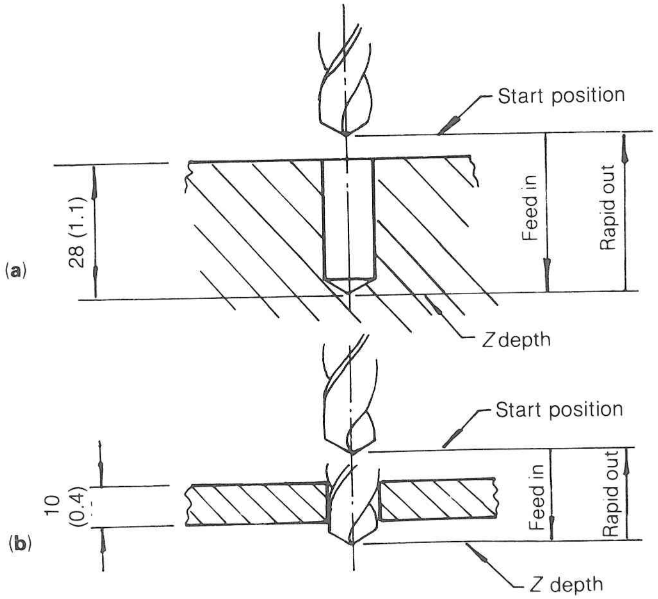


Figure 6.18 *Movements required to drill holes. (Inch units are given in parentheses.)*

Nonstandardized Fixed Cycles

It is often the case that manufacturers of machine control units wish to include in their systems cycles that are not necessarily widely applicable and therefore do not fit into the "standardized" category, but the inclusion of which considerably enhances their control system. The cycles they choose to include will depend on the machine type to which the control is to be fitted. Some of the more common cycles of this nature are discussed below.

Loops The term "loop" is particularly relevant when reducing raw material to size by making a series of roughing cuts. Consider the component shown in Figure 6.19, which is to be reduced from 50 mm (2 in.) to 26 mm (1 in.) diameter by a series of cuts each of 2 mm (0.08 in.) depth. Assuming that the starting point for the tool is as shown, the tool will first move in a distance of 2.5 mm (0.1 in.), thus taking a 2 mm (0.08 in.) depth of cut, travel along a length of 50 mm (2 in.), retract 0.5 mm (0.02 in.), and return to the *Z* datum, thereby completing the loop. It will then move in a distance of 2.5 mm (0.1 in.), feed along 50 mm (2 in.), retract 0.5 mm (0.02 in.), and return to the *Z* datum, and so on. The loop, including the feed rate, is programmed just once, but is repeated via the "loop count" command in the main program as many

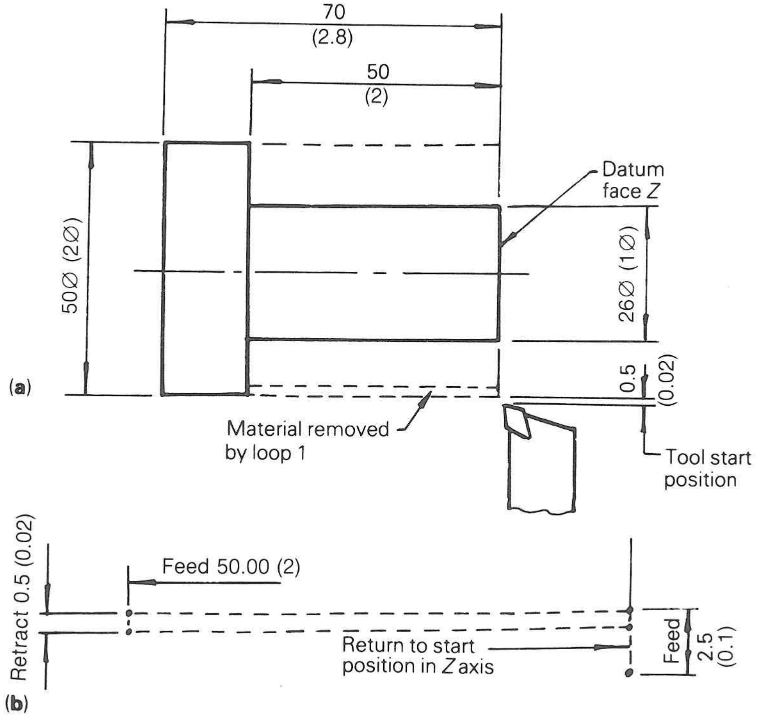


Figure 6.19 *Looping or roughing cycle:* (a) *component and* (b) *loop details, repeated six times. (Inch units are given in parentheses.)*

times as necessary to reduce the work to the required diameter. Note: some controls will do this with special "G" codes, while other controls will use special command codes, but the results are the same.

Face Milling Cycle Figure 6.20 shows details of a face milling cycle. After programming the appropriate G code, together with spindle speed and feed rate, the only other information required are the *X* and *Y* dimensions of the face to be milled. The control unit computer will determine the number of passes necessary and the appropriate cutter step-over to machine the face. The cutter diameter will be picked up automatically from previously entered information. This type of cycle is very commonly found on conversationally programmed controls.

Slot Milling Cycle Figure 6.21 illustrates a slot milling routine. As with face milling, the programmer has to state spindle speed, feed rate, and slot dimensions in the *X* and *Y* axes. The first pass made by the cutter passes through the middle of the slot and then returns to the start. Further passes are made until the correct depth is achieved, the number of passes necessary being determined by the axis increment depth programmed in the cycle. When the correct depth is reached, the cutter path is that of a series of cycles increasing in size with each pass. Some controls vary this process by cutting the entire slot at each depth except the finish pass. Again, as with the face milling, the computer will determine the step-over and the number of cycles necessary to machine the slot to size.

Pocket Milling Figure 6.22 illustrates the pocket milling cycle. This cycle starts at the center of the pocket, the cutter feeding in the *Z* axis to a programmed depth. There follows a series of cycles until the programmed *X* and *Y* dimensions are reached, the step-over of up to 80% of the cutter diameter will ensure that a flat surface is produced by providing overlap of passes. Some systems provide for a cycle that roughs out the main pocket and then machines to size with a small finishing cut. If the pocket depth is such that more than one increment in the *Z* axis is necessary, the slide movement returns the cutter to the center of the pocket and the cycle is repeated at the next depth.

Bolt Hole Circles The term "bolt hole circle" means that a number of holes are required equally spaced on a stated pitch circle diameter as illustrated in Figure 6.23. Given that the program has brought the cutter to the pole position,

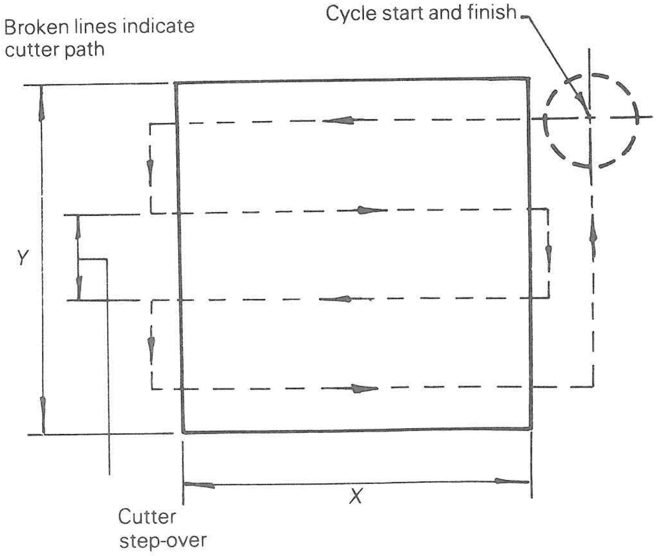


Figure 6.20 *Face milling cycle.*

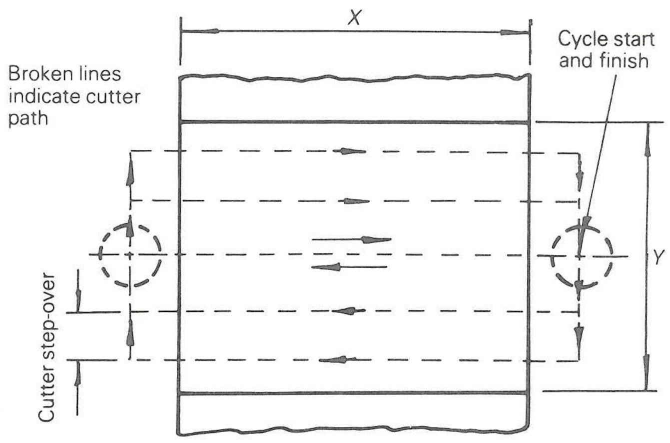


Figure 6.21 *Slot milling cycle.*

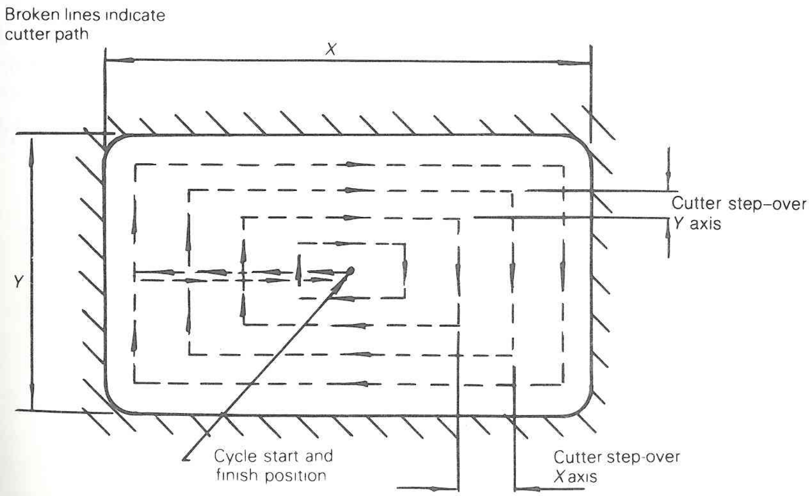


Figure 6.22 *Pocket milling cycle.*

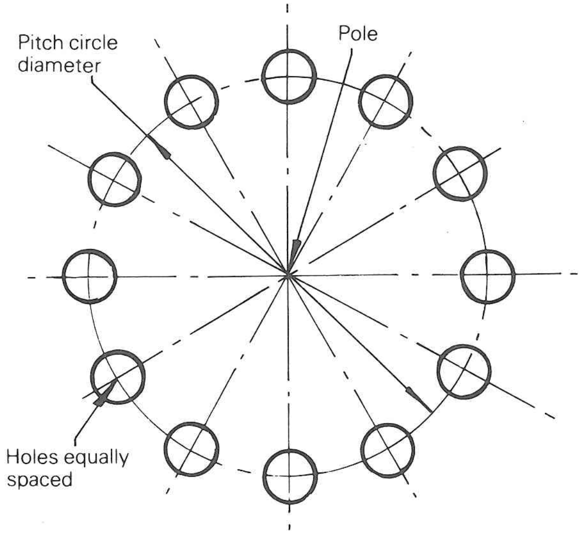


Figure 6.23 *Bolt hole circle.*

the other dimensional data required are the position of the first hole, the *Z* axis movement, the pitch diameter or radius, depending on the control system, and the number of holes required. The computer makes all the necessary calculations to convert the polar coordinates to linear coordinates and to move the slides accordingly.

A variation of this cycle will cater for just two or three holes positioned in an angular relationship to one another. An example is detailed in Figure 6.24. Again, the pole position is programmed and the cutter will be at this point when the cycle commences. The additional dimensional data that have to be supplied are the *Z* axis movement, the polar radius and the polar angle(s), and the number of holes required, the computer then converts this information to slide movement in the appropriate axes.

On some control systems it is possible to "rotate" more complex loop programmed features such as the example shown in Figure 6.25.

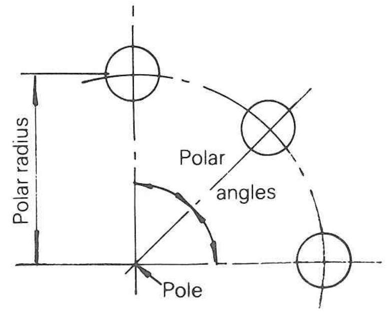


Figure 6.24 *Polar coordinates.*

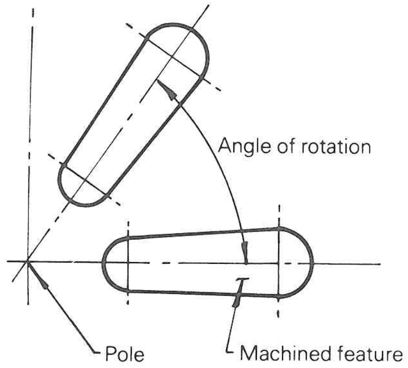


Figure 6.25 *Feature rotation.*

Cycles Devised by the Part Programmer

Cycles devised by the part programmer may be defined as follows. First, there are cycles that are devised specifically for one particular machining task. Second, there are those that may be used when machining a range of components.

Consider the component shown in Figure 6.26, which has a repetitive feature, namely, the recess. When writing a program for machining this particular component, the programmer would devise a cycle, in situations such as this being referred to as a "routine," for producing just one recess. Via an appropriate call the blocks of data defining the routine can be activated as and when required within the main machining program at new locations.

The construction of a routine may include subroutines also specifically constructed by the part programmer and may also utilize any fixed or canned cycles that are considered appropriate. The technique of programming cycles or routines within routines is referred to as "nesting" and is further described subsequently.

Assume the component shown in Figure 6.26 is quite large so that within each recess there were also a number of holes arranged in three groups, as

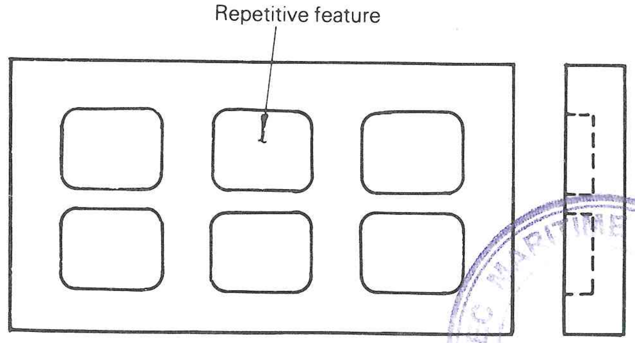


Figure 6.26 *Component with repetitive feature.*

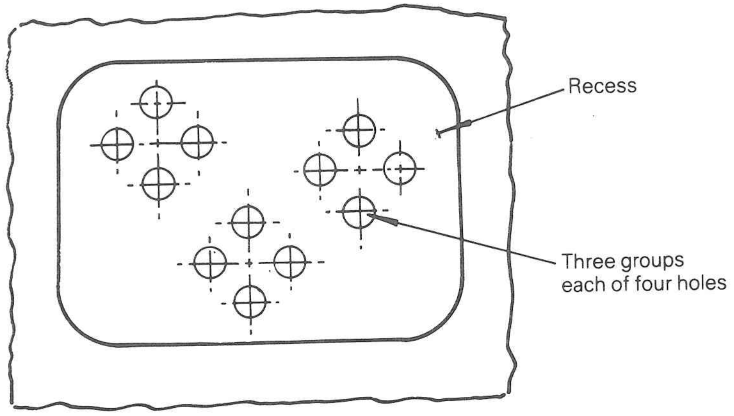


Figure 6.27 *Enlarged detail of component in Figure 6.26.*

shown in 6.27. The main routine would be the data necessary for the production of the recess, as explained above. The subroutine would be the data necessary to produce a group of four holes. The subroutine would be nested within the main routine and called into the main program on three occasions.

However, the production of the four holes is repetitive, and thus it is possible to program to produce just one hole, but to repeat the sequence four times. The complete sequence for producing the component is illustrated diagrammatically in Figure 6.28. On some control systems it is possible to program cycles within cycles as many as eight deep.

Programmer-devised cycles of the second type, to which reference was made above, are useful when a machined feature commonly occurs within the production schedule of a particular company, that is, a machined feature (possibly of unusual design) is required over a range of components. To accommodate this situation some control systems permit routines that are "user defined" to be prepared and "stored within the control system," so that they may be recalled and utilized as and when required as part of a more comprehensive machining program. A routine of this nature is also referred to as a "macro."

A macro may have fixed dimensions or it may have parametric variables, that is, the dimensions may be varied to produce different versions of the same basic feature or component. This technique is referred to as "parametric programming" and is described in more detail in Chapter 9.

MIRROR IMAGE

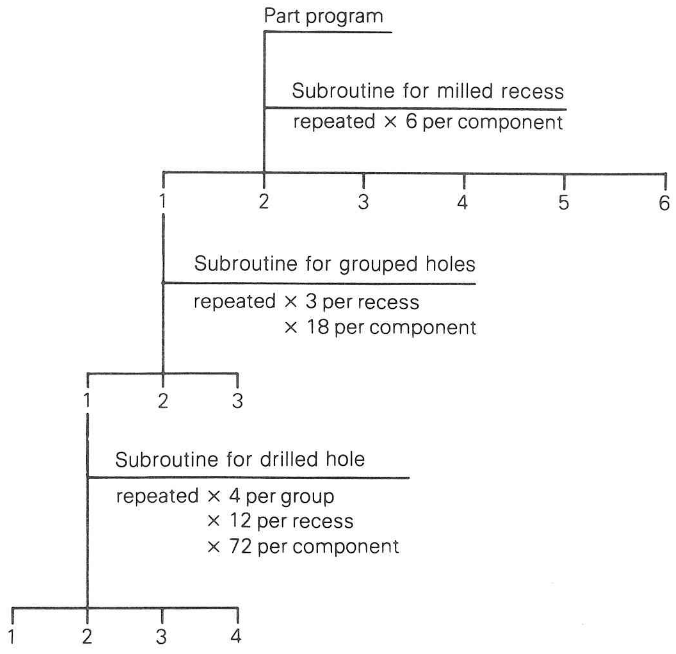


Figure 6.28 *'Nesting' three deep.*

A commonly occurring aspect of mechanical engineering design is the need for components, or features of components, that are dimensionally identical but geometrically opposite either in two axes or in one axis. By using the mirror-image facility such components or features can be machined from just one set of data. The component shown in Figure 6.29 has a feature that is mirrored in two axes. Note that, to produce the second profile, the positive incremental values become negative and the negative incremental values become positive. To produce a feature of the opposite hand, as shown in Figure 6.30, the direction of slide movement changes in one axis only.

SCALING

Another common requirement in mechanical engineering design is components with the same geometrical shape but varying dimensionally. Figure 6.31 illustrates two such components. When a control system is fitted with a scaling facility, it is possible to produce a range of components, varying in size, from one set of program data. The facility can also be used to produce geometrically identical features of components that may be required to be reproduced to different sizes.

SLASH DELETE

The slash or block delete facility enables part, or parts, of a program to be omitted. It is particularly useful when producing components that have slight

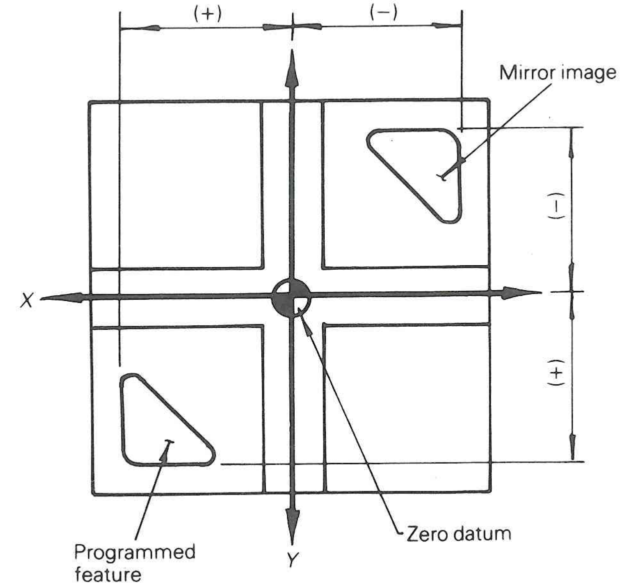


Figure 6.29 *Mirror image in two axes.*

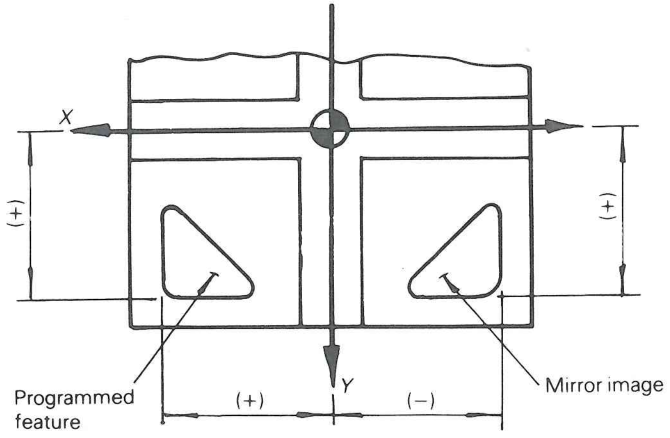


Figure 6.30 *Mirror image in one axis.*

dimensional variations. For example, a hole may be required in one version of a component but not in another, although all other details may be identical. The program data relating to the production of the hole are contained within the programmed symbols/, one at the start of each block concerned. An example is shown below. See manufacturers' programming manual for possible variations in format.

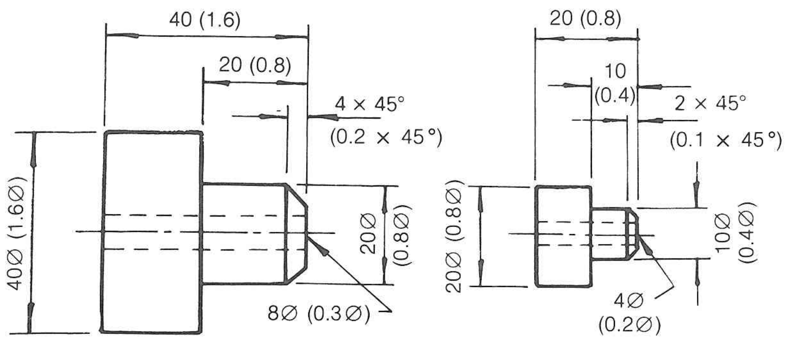


Figure 6.31 *Geometrically identical components suitable for production by scaling. (Inch units are given in parentheses.)*

```
/N05  G01    Z1000 F 150*
/N06  G00    Z-1000 *
```

To make a component *with* the hole, the operator need not take any action. To produce a component *without* the hole, the operator will have to activate the slash delete switch on the control console at the start of the program. When the slash is reached, the control unit will ignore the data that follow. On some systems, if the slash delete is not activated, the program will stop when the slash is reached and the operator then has to make a positive response either to activate the data or to delete them.

This facility is particularly useful when machining castings or forgings, where stock removal requirements may vary, the operator being given the option to include an extra cut or delete it as necessary.

JOG

The jog facility enables the machine operator to move the machine slides manually via the control console. This may need to be done for a variety of reasons, the most obvious one being when establishing datums at the initial setting of the machine. There are also two standard "G" code boring cycles that call for jog retracts after the cycle before returning to automatic operation. It may also be necessary to stop an automatic sequence and move the machine slides to facilitate work measurement, tool changing due to breakage, and so on. Whatever the reason it is desirable that the automatic program is restarted at the point at which it was interrupted, and most control systems have a *return from jog* facility that returns the machine slides to their original positions, this facility being activated manually via a button on the control console.

PROGRAM STOP

Stops in a machining sequence can be predetermined and included in the part program as a miscellaneous function (M00). Scheduled stops for measurement, tool changing (on manual tool change machines), etc., have to be notified to the machine operator so that he or she will be aware of his or her duties at this point.

Program stops can also be optional, that is, the sequence does not have to stop. Optional stops are also included in the part program as a miscellaneous function (M01) and the control will ignore the command unless the operator has previously activated a switch on the control console.

DATUMS

Machine Datum

The machine datum, also referred to as "zero datum" or simply as "zero," is a set position for the machine slides, having a numerical identity within the control system of zero. All slide movements are made in dimensional relationship to this datum as indicated earlier in the chapter, when absolute and incremental positioning moves were discussed.

On some machines the zero datum may be a permanent position that cannot be altered. On other machines a new zero is readily established by moving the slides so that the cutting tool is placed in the desired position in relation to the workpiece and then pressing the appropriate zero button on the control console. The facility to establish a datum in this manner is referred to as a floating zero or zero shift. The location of the original zero is not retained within the control memory.

A fixed machine datum may be helpful to the part programmer, especially when the programming is carried out remote from the machining facility, since the position can be taken into consideration when writing the program. It will be necessary, however, for the programmer to specify the exact location of the component in relation to the machine datum if the program is to achieve the desired results.

A floating zero affords greater flexibility when machine setting, since the work can be positioned anywhere within the range of slide movement and the zero established to suit. But this can be time-consuming and, if incorrectly carried out, may result in machining errors.

Program Datum

The program datum or zero is established by the part programmer when writing the part program, and the program will require all slide movements to be made in relation to that point.

In practice, the machine zero and the program zero are often synchronized by either accurately positioning the work or, when possible, resetting the machine zero. Any unavoidable variations between the two positions can be accommodated by using the zero offset facility, if available, as described below.

ZERO OFFSET

The zero offset facility enables a machine zero datum to be readily repositioned on a temporary basis. Once it has been repositioned, the slide movements that follow will be made in dimensional relationship to the new datum. It is particularly useful when the original machine datum does not coincide with the part program datum, a situation that can arise, for instance, when a part program has been prepared without regard to the normally fixed position of the machine datum and difficulties are encountered in positioning the workpiece to suit the part program. A simple example of this is when a part program for a turned component has been prepared using the forward face of the workpiece as a zero datum when the machine zero is, as is often the case, located at the back face of the chuck. Use of the zero offset facility will reestablish the zero at the work face. The zero offset facility also enables two or more components to be machined at one setting from the same part program. In Figure 6.32 component 1 would be machined with slide movements made in relation to datum 1. On completion of the machining sequence, the machine table would be caused to move, via the part program or by manual intervention at the keyboard to datum 2 where the offset feature would be activated to the predeter-

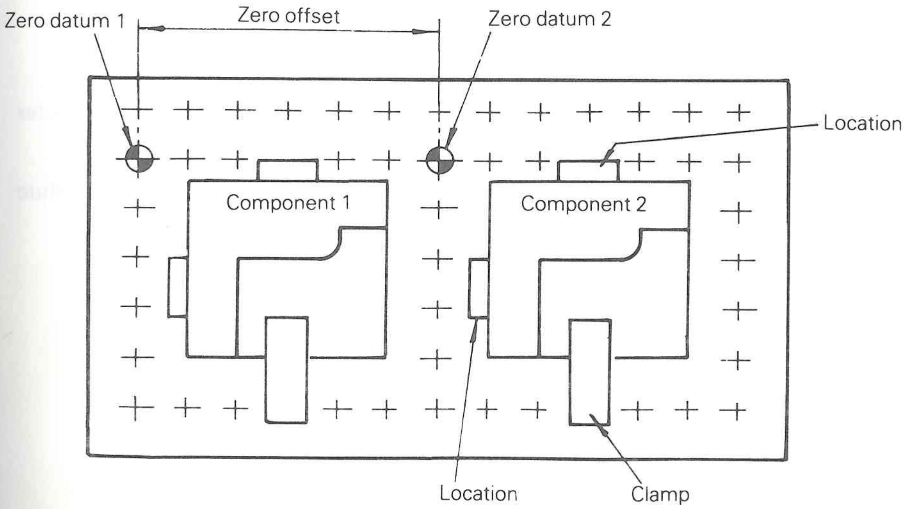


Figure 6.32 *The use of zero offset facility when machining components located on a grid plate.*

mined offset dimension and the machining sequence would be repeated for component 2, with all slide movements being made in relation to datum 2. The control would retain information regarding the location of the original datum, which remains a permanent feature of the part program, and the machine slide can be caused to return to that position.

The facility may also be used to machine identical features on components of different lengths. A simple example would be to cut a screw thread of particular dimensions on the ends of two bars the overall lengths of which are not the same. Similarly, the facility can also be used when turned workpieces are reversed for secondary operations and the second setting in relation to the Z axis zero differs from the first.

On the more sophisticated controls it is possible to establish a new zero, or zeros, at various stages throughout the program. All subsequent moves will be made in relation to the new datum, but these moves are not necessarily a repeat of the moves made before the new zero was established. This facility enables the features of complex or very long components to be machined by relating slide movement to more than one datum, thus simplifying programming and possibly reducing machining time by limiting the length of slide travel.

QUESTIONS

1 What is a preparatory function and how is it designated in word address programming?

2 What is a miscellaneous function and how is it designated in word address programming?

3 What is meant by the term "modal"?

4 Name and describe the three types of positioning control used on computer numerically controlled machine tools.

5 Explain, with the aid of a simple sketch, the difference between absolute and incremental dimension definition.

6 When are the letters *I*, *J*, and *K* used in a word address program?

7 What is meant by the term "ramp suppression"?

8 Describe what happens during a peck drill cycle.

9 What is a looping cycle and when is it used?

10 Describe what happens during a pocket milling cycle.

11 What is a bolt hole circle?

12 What is meant by the "rotation" of a machined feature?

13 With the aid of a simple sketch describe the effect of reproducing a machine feature using the mirror image programming facility in (a) two axes; (b) one axis.

14 What is the programming function that permits the production from one set of data, components geometrically identical but with proportional dimensional variations?

15 Explain the meaning of the term "nesting" as applied to machining cycles.

16 When is the block or slash delete facility likely to be used and how is it generally invoked?

17 What is the jog facility on a machine control system and when is it likely to be used?

18 Give two reasons for including an optional stop in a program.

19 Why is it necessary to inform a machine operator of the scheduled stops in a machining program?

20 With the aid of simple sketches describe the meaning of zero offset.

7

SPEEDS AND FEEDS FOR NUMERICALLY CONTROLLED MACHINING

CUTTING SPEEDS AND FEEDS

It is difficult to determine precise data for any metal-cutting operation without knowledge of the practicalities involved. For example, the condition of the machine, the power available, the rigidity of tooling and work-holding arrangements, the volume of metal to be removed, the surface finish required, and the type of coolant to be used are all factors that have to be considered when determining the appropriate speeds and feeds to be used. The programmer is obliged to make program entries that will be, if not perfect, then at least functional.

Should it prove that the programmed speed and feed are inappropriate, machine control units have manually operated override facilities to enable the operator to increase or decrease speeds and feeds as machining proceeds. If such action is found to be necessary, the programmer should be informed so that the part program can be modified accordingly.

The selection of appropriate speeds and feeds can be based on experience when the programmer possesses the necessary practical background. Experienced craftsmen often have an instinctive ability to recognize the correct speed and feed for any particular machining task, having long forgotten any theoretical basis they may have once been taught concerning such matters. But this approach is perhaps more suitable to conventional machining processes than it is to CNC machining.

Reference to data published by the manufacturers of the cutting tools to be used is more appropriate for numerically controlled machining, since the lack of operator involvement that is a feature of machining in this way makes it imperative that the optimum speeds and feeds are used. However, when reference is made to published data, the programmer should exercise caution. The figures quoted, while perfectly feasible when used under the correct conditions, can appear to be somewhat optimistic when applied to many machining situations. Examples of manufacturers' data relating to cutting speeds and feeds are included in Appendices B.

Whatever approach is used, it is essential that the programmer fully appreciates the capabilities of any machine for which he or she is preparing programs. For instance, it is pointless to program speeds and feeds that result in metal removal rates that are beyond the power capacity of the machine. Conversely, it is equally pointless to underuse the power available. If a programmer lacks essential knowledge of this nature, then liaison with those who have had practical experience of the machine should be a priority.

Unusual machining situations involving set-ups that may lack rigidity should be approached with care, and the programmed speeds and feeds should be initially on the low side. They can always be increased later in the light of experience gained at the machine when metal cutting has actually taken place.

Special attention should also be given to specific requirements regarding surface finish. Machinists are often obsessed with obtaining a "good" finish, but it should be remembered that there is no point in reducing metal removal rates to obtain a high-quality surface finish that is not necessary. A further point to remember is that a designer may specifically require a "rough" finish, so it is essential to work to the information contained on the drawing. The programmer should ensure that the setup person/operator responsible for the machining operation fully appreciates what the requirements are.

The intricate contouring capabilities of CNC machines can also present problems regarding surface finish. For example, feed rate that produces satisfactory results when machining a parallel turned surface may no longer produce acceptable results when the cutting tool changes direction to machine a tapered or radial surface. Similar situations present themselves during milling operations. When machining complex profiles that are subject to stringent surface finish requirements, the results obtained from programmed feeds and speeds, particularly feeds, should be monitored and modifications made if necessary.

Surface finish is, of course, affected by the condition of the cutting tool. Tool performance begins to deteriorate from the moment the tool is first used, and ultimately it not only affects the surface finish but may result in dimensional features not being maintained, unacceptable vibration, work deflection and eventually total failure of the tool. An unsuitable choice of cutting speeds and feeds may hasten this process, so the part programmer must give due consideration to tool life when making decisions in this respect.

It is possible to calculate the life of a cutting tool. The formula used for this purpose was derived by experimentation and, because of the many variables that exist between one machining situation and another, the results obtained from its application can only be used as a guide. The reader may reasonably ask why bother to make such a calculation if the result is only a guide, and it would be difficult to give a totally convincing answer. Nevertheless, it would be helpful to know when a tool was reaching the end of its life, so that replacement could be effected before total failure occurred. The automatic tool condition sensing devices being applied to the more complex CNC machining installations may provide the answer. In the meantime, how is the part programmer to decide what speeds and feeds to use to give an acceptable tool life, while at the same time achieving the basic objective of removing metal in the least possible time?

As stated earlier, reference can be made to the cutting tool manufacturers' literature. Their figures will have taken into account the fact that a reasonable tool life is required by the user of their products. But there is still the problem that local conditions may make their recommendations invalid. Yet again, it may be initially necessary to rely on past practical experience, and to be prepared to make modifications based on a reasoned appraisal of the situation when metal cutting is under way.

One further point relating to tool performance should be noted. The chip-breaking qualities of most carbide-tipped tools is directly related to speeds and feeds. If chip clearance becomes a problem, some modification of the cutting conditions may be appropriate.

SPINDLE SPEEDS

The program data controlling spindle speeds is expressed in one of two ways, the numerical value in both cases being preceded by the letter S. Thus a data entry of, say, S250 could indicate either a *constant surface cutting speed* of 250 m per minute or, alternatively, a *spindle speed* of 250 revolutions per minute (rev/min).

All machines have the facility to program a set spindle speed in revolutions per minute, while the alternative facility of programming a constant surface cutting speed is now commonly available on the majority of turning centers.

When both facilities are available, the machine controller differentiates between the two possibilities via a previous data entry that will establish the desired operating mode. In the case of word address programming this mode is established by a G code entry, commonly G96 (ft, or m per min) and G97 (rev/min). Feet or meters per minute will depend on whether the G70 inch programming or G71 metric programming code is active. In the case of conversational MDI, the mode is established by selection from the displayed options.

Consider the process of programming a constant surface cutting speed. The cutting speed is the rate at which the cutting tool passes over the workpiece material, or alternatively the rate at which the material is traveling as it passes the cutting tool. As indicated previously, it can be expressed in either meters or feet per minute.

Appropriate surface cutting speeds for use with cutting tools made of specific materials, when used to cut certain metals, have been determined by experiment. The figures give due regard to the maximum metal removal rates that can be obtained, while at the same time equating satisfactorily with other factors such as tool life, surface finish, and power consumption. These recommended cutting speeds are published by the manufacturers of cutting tools as a guide to users of their products (see Appendix B). As stated earlier, it may be necessary to modify these values to suit local conditions before making a program entry.

The advantage of programming a surface cutting speed as opposed to a set spindle speed in revolutions per minute is best appreciated by considering the simple operation of parting-off from bar stock during a turning operation.

During a parting-off operation the diameter of the work where metal cutting is actually taking place is steadily decreasing, and therefore the cutting efficiency is only maintained if the spindle speed increases at a corresponding rate. This steady increase, which maintains the most efficient metal cutting rate for that particular job material, is automatically achieved via the constant surface cutting speed programming facility.

The process of parting off is a convenient one to explain the value of the constant surface cutting speed programming. However, it is not a process where the use of such a facility is absolutely critical. The facility is more likely to be of value during the turning of complex profiles requiring a uniformly high standard of surface finish throughout the turned length.

In order to program a constant spindle speed in revolutions per minute, it is necessary to make a simple calculation that takes into consideration the recommended surface cutting speed referred to previously, and also the diameter of the workpiece in the case of turning operations, or the cutter in the case of milling operations. The relationship between these factors is expressed as follows.

$$\text{Spindle rev/min} = \frac{1000 \times \text{Cutting speed in m/min}}{\pi \times \text{Work or cutter diameter in mm}}$$

Multiplying the cutting speed by 1000 converts it from *meters* per minute to *millimeters* per minute, while multiplying the work or cutter diameter by π, that is, calculating the circumference, determines the relative linear travel per revolution in millimeters. Dividing the circumference in millimeters into the cutting speed in millimeters per minute determines the number of revolutions per minute required.

When inch programming, the relationship between spindle speed, work or cutter diameter and the required cutting speed is expressed as follows:

$$\text{Spindle speed in rev/min} = \frac{12 \times \text{Cutting speed in ft/min}}{\pi \times \text{Work or cutter diameter in inches}}$$

In this case multiplying the cutting speed by 12 and so converting it to in./min makes all the units in the equation compatible.

A number of cutting tool manufacturers distribute simple calculators, the use of which eliminates the need to make calculations. These devices will indicate the appropriate cutting tool material, selected from the manufacturer's range, that should be used when machining a certain material type, together with a recommended surface cutting speed for the type of operation (roughing or finishing) that is to be undertaken.

SURFACE CUTTING SPEEDS

The speeds in Table 7.1 are suitable for average metal-cutting conditions. In practice it may be possible to increase these speeds considerably for light finishing cuts or, conversely, to reduce them for roughing cuts. For more accurate speed and feed tables related to operation, material, and finish refer to *Machining Data Handbook* by the Machinability Data Center (Metcut Research Associates Inc.) or *Machinery's Handbook* by Oberg, Jones, and Horton (Industrial Press Inc.).

SPINDLE SPEEDS

Example: Determine the spindle speed required to turn a 2.5-in.-diameter piece of aluminum using a high-speed tool and a surface feet per minute (SFPM) cutting speed of 300 ft/min:

$$\text{Spindle speed, rev/min} = \frac{12 \times 300}{\pi \times 2.5} = 458$$

Example: Determine the spindle speed required to turn 50 mm diameter brass using a cemented carbide tool and a surface cutting speed of 180 m/min:

$$\text{Spindle speed, rev/min} = \frac{1000 \times 180}{\pi \times 50} = 1146$$

These spindle speeds will be correct for turning a 2.5 in. or 50 mm diameter bar. However, as the diameter decreases, as for instance during an end-facing operation, these spindle speeds are no longer valid or efficient. On many numerical control systems it is possible to program a surface cutting speed only, and the machine spindle speed will automatically vary within the range of the machine to compensate for changes in the work diameter, thus providing a constant cutting speed.

Table 7.1 Surface cutting speeds, m/min (ft/min)

	Part material			
Tool material	**Mild steel**	**Cast iron**	**Aluminum alloy**	**Brass**
Cemented carbide	170 (300)	100 (225)	250 (900)	180 (450)
High-speed steel	28 (90)	18 (75)	120 (300)	75 (150)

FEED RATES

The manufactuer's calculators referred to previously will also indicate an appropriate feed rate for the operation. The feed rate is the speed at which the cutter penetrates into the work material.

When programming data relating to feed rate, it can be expressed either as millimeters per minute (mm/min) or millimeters per revolution (mm/rev) of the machine spindle. With inch programming, the units will be inches per minute (in./min) or inches per revolution (in./rev).

The letter F is commonly used to denote the feed rate in a part program. Thus F25 could indicate a feed rate of 0.25 in. or mm/rev, while F80 could indicate a feed rate of 80 in. or mm/min, depending on the mode of expression being used. It is necessary to ensure that the data entered and the programming mode are compatible.

Although variations exist, the set-up data using the G codes common to a number or word address programming systems are as follows:

G94 feed/min
G95 feed/rev

In this situation the units to be used are established by the use of G70 for inch and G71 for metric.

Feed rates are published by cutting tool manufacturers in the same way as surface cutting speeds. Usually the rates are expressed as mm/rev or in./rev. To convert to mm/min or in./min involves making a simple calculation as follows:

$$\text{Feed mm/min} = \text{Feed mm/rev} \times \text{Spindle speed rev/min}$$

or,

$$\text{Feed in./min} = \text{Feed in./rev} \times \text{Spindle speed rev/min}$$

The manufacturers of milling cutters sometimes quote recommended feed rates in millimeters or inches per tooth, in which case it is necessary, prior to making the preceding calculation, to determine the feed per revolution of the cutter. This is achieved as follows:

$$\text{Feed/rev} = \text{Feed/tooth} \times \text{Number of cutter teeth}$$

Feed Rates for Turning

Cemented carbide tools are used extensviely for turning operations. It is common practice for the manufacturers to quote recommended feed rates in inches or millimeters per spindle revolution (in./rev. or mm/rev). Typical feed rates for different work materials are given in Table 7.2.

To determine the feed rate in mm/min (in./min):

$$\text{Feed mm/min (in./min)} = \text{Feed mm/rev (in./rev)} \times \text{Spindle speed (rev/min)}$$

Table 7.2 Typical feed rates for turning, mm/rev (in./rev)

Mild Steel	Cast Iron	Aluminum Alloy	Brass
0.25 (roughing 0.007–0.025) (finishing 0.002–0.007)	0.25 (roughing 0.007–0.025) (finishing 0.002–0.007)	0.3 (roughing 0.007–0.030) (finishing 0.002–0.007)	0.3 (roughing 0.007–0.030) (finishing 0.002–0.007)

When a constant surface cutting speed is programmed, that is, the spindle speed varies automatically to compensate for variations in the work diameter, the feed rate is programmed in in. or mm/rev to maintain a constant feed rate per spindle revolution. When the spindle speed is programmed at a constant rev/min, the feed rate can be entered either as in. or mm/rev or in. or mm/min, since both will result in a constant relationship between surface cutting speed and feed.

Feed Rates for Milling

The manufacturers of milling cutters state recommended feed rates as in. or mm/rev, in. or mm/min or in. or mm/tooth.

When feeds are quoted as in. or mm/rev or in. or mm/min, they usually refer to specific cutters in the manufacturer's range and cannot be generally applied. For instance, if two face mills both of the same diameter, but one having five carbide inserts and the other six, were used at the same spindle speed with a feed quoted per revolution, it would mean that the cutter with the fewest teeth would be subjected to a much higher volume of metal removal per tooth than the cutter with more teeth. So for general use feed rates quoted in in. or mm/tooth are more suitable. These data can then be used to determine the feed rate per revolution as follows:

Feed, mm/rev (in./rev) = Feed, mm/tooth (in./tooth) × Number of teeth

And from this formula,

Feed, mm/min (in./min) = Feed, mm/rev (in./rev) × Spindle speed (rev/min)

Typical feed rates are given in Table 7.3.

Feed Rates for Drilling

High-speed-steel drills are used extensively for producing smaller holes. Since small-diameter drills are liable to break, the feed rate is related to the drill size. Typical feed rates are given in Table 7.4.

Table 7.3 Typical feed rates for milling, mm/tooth (in./tooth).

	High-speed-steel cutters		Cemented carbide cutters	
Work Material	Face and Shell End Mills	End Mills	Face and Shell End Mills	End Mills[a]
Mild steel	0.25 (0.010)	0.15 (0.005)	0.30 (0.008–0.020)	0.18 (0.003–0.010)
Cast iron	0.30 (0.013)	0.18 (0.007)	0.50 (0.008–0.020)	0.21 (0.005–0.012)
Aluminum alloy	0.40 (0.022)	0.17 (0.011)	0.60 (0.005–0.020)	0.25 (0.005–0.020)
Brass	0.35 (0.014)	0.15 (0.007)	0.40 (0.005–0.020)	0.19 (0.003–0.012)

[a] As with high-speed-steel end mills, the design of some cemented carbide end mills provides for both plunge and side cutting.

Table 7.4 Typical feed rates for high-speed steel drills

Drill Size (mm)	2	4	6	8	10	12	14	16	18	20
Feed Rate (mm/rev)	0.05	0.10	0.12	0.15	0.18	0.21	0.24	0.26	0.28	0.30

Drill size (in.)	0.125	0.250	0.500	1.0	1.0
Feed rate (in./rev)	0.001–0.003	0.002–0.006	0.004–0.010	0.007–0.015	0.010–0.025

Cemented carbide drills, sometimes with tips brazed to a medium carbon steel shank, but more commonly as clamped inserts, are favored for larger holes. The feed rates for these drills compare with those used for carbide insert end mills. Carbide use in small holes requires a solid carbide tool, which is very expensive.

The feed rate for solid carbide drills can be determined by using tables for brazed tip as a starting point.

As with turning and milling, the feed in in. or mm/rev can be used to determine the feed in in. or mm/min, as follows:

Feed (in. or mm/min) = Feed (in. or mm/rev) × Spindle speed (rev/min)

FEED RATE AND SPINDLE SPEED OVERRIDE

On the majority of control systems there are facilities that enable the operator manually to change, via a setting dial, both the feed rate and the spindle speed. Selection is usually on a percentage basis, 0–150% being fairly typical.

Changes in speed and feed may be necessary for a number of reasons. For example, the operator may judge that the rate of metal removal can be safely increased and so may increase the feed rate. Similarly, the operator may judge that a prolonged tool life would result if the spindle speed were decreased, or the surface finish being obtained from the programmed feed rate may be unsatisfactory, and so on. Manual control of speeds and feeds is also a very helpful feature during machine setting and program proving.

Changes made manually do not affect the basic program, although if an operator decides that the programmed feed rate or spindle speed for any part of a program is unsatisfactory he or she should make the fact known so that a permanent change can be made.

QUESTIONS

1 Select a suitable surface cutting speed and calculate the spindle speed required to turn a mild steel component with a diameter of 55 mm, using a cemented carbide indexable insert cutting tool.

2 If the component in question 1 has a second diameter of 24 mm, what change in spindle speed will be necessary?

3 Compare the surface speeds for machining mild steel using high-speed steel and cemented carbide cutters and express the variation as a percentage.

4 What is meant by the term "constant surface cutting speed"? Quote a situation where it may be desirable during a machining operation.

5 Select a suitable feed rate in in./rev for finish turning free-cutting mild steel using a cemented carbide tool. Using these data, determine a suitable feed rate in in./min for turning a diameter of 3.5 in.

6 Select a suitable feed rate in mm/rev for finish turning cast iron using a cemented carbide tool. Using information found in previous tables, determine a suitable feed rate in mm/min if turning a diameter of 75 mm in the same cast iron material.

7 Select a suitable surface cutting speed and determine the spindle speed required to drill a 9/16 in. diameter hole in mild steel.

8 Select a suitable surface cutting speed and determine the spindle speed required to drill a 6 mm diameter hole in free-cutting brass.

9 Why is it that the feed rate in in. or mm/rev for drilling operations varies with the drill diameter?

10 Calculate the approximate feed rate in mm/min for milling a slot in brass with a high-speed steel cutter of 15 mm diameter.

11 Calculate the approximate feed rate in in./min for milling a slot in an aluminum part with a high-speed steel two flute cutter of 0.375 in. diameter.

12 Determine a suitable spindle speed for face milling aluminum alloy using a cemented carbide cartridge-type milling cutter of 100 mm diameter.

13 Determine a suitable spindle speed for face milling cast iron using a cemented carbide insert type milling cutter with a 4.5 in. diameter.

14 If the cutter in question had six cartridges, what would a suitable feed rate be in mm/rev?

15 If the cutter in question 13 had eight inserts, what would a suitable feed rate be in in./rev?

8

PART PROGRAMMING FOR COMPUTER NUMERICALLY CONTROLLED MACHINING

THE PART PROGRAM

The term part program is used to describe a set of instructions that, when entered into a machine control unit, will cause the machine to function in the manner necessary to produce a particular component or part. Manual part programming is the term used to describe the preparation of a part program without recourse to computing facilities to determine cutter paths, profile intersecting points, speeds and feeds, etc.

The program may be prepared manually and expressed in a coded language that is applicable to the machine controller being used. Alternatively, it may be written in another language or compiled by the use of computer graphics. The result is then post-processed, or translated, to suit the machine controller.

Included in the part program will be the necessary dimensional data relating to the features of the component itself, together with control data that will result in the machine making the slide movements required to produce the component. These data will be supplemented by instruction data that will activate and control the appropriate supporting functions.

Programs as entered into machine control units involve either of two programming concepts:

(a) word address
(b) conversational manual data input (MDI)

There are considerable variations between the two methods.

Whether a production scene incorporates total automation or merely one or two numerically controlled machines positioned among traditional machines, at the heart of successful numerical control is efficient competent part programming. The practical *skill* level requirement on the shop floor is, without doubt, in decline, but a high level of practical *knowledge* is essential if part programmers are to use costly equipment at their disposal to the best advantage. The selection of a correct sequence of operations, together with efficient cutting speeds and feeds, tooling and work holding, and the ability to express these requirements in the correct format are of paramount importance.

Unfortunately, programming methods differ and even when the basic approach is similar (for example, with word address), there are still variations and peculiarities, and conversational manual data input is very individual.

Thus the reader should appreciate that the ability to program with one control system, although there is much carryover, rarely means that knowledge can be used *in total* elsewhere. Specialist training is essential, and most machine-tool manufacturers respond to this by offering training courses as part of the overall package to customers buying their equipment. However, once the basic concepts involved in part programming are understood, the change from one system to another does not appear to be a major problem. Indeed, the variations encountered can be a source of much interest, while the mastery of yet another system can give considerable personal satisfaction.

PROCEDURE

Taking as a starting point the detail drawing of the component to be manufactured, the tasks that confront the part programmer may be listed as follows:

1. Select a machine capable of handling the required work.
2. Determine the machine process to be used.
3. Determine work holding and location techniques.
4. Determine tooling requirements and their identity.
5. Document, or otherwise record, instructions relating to work holding, work location, and tooling.
6. Calculate suitable cutting speeds and feed rates.
7. Calculate profile intersecting points, arc centers, etc.
8. Determine appropriate tool paths including the use of canned cycles and subroutines.
9. Prepare the part program.
10. Prove the part program and edit as necessary.
11. Record the part program for future use.

Although these stages have been given a separate identity, they are very much interrelated and cannot be treated in isolation. A diagrammatic impression of the approach to be adopted is given in Figure 8.1.

MACHINE SELECTION

In selecting the machine to be used the first consideration is the type of work that has to be carried out. The tolerance and surface finishes required on the part will determine the type of machine and process to be used. Even when the type of machine is established, its specifications will need to be reviewed to ensure that part accuracy can be maintained.

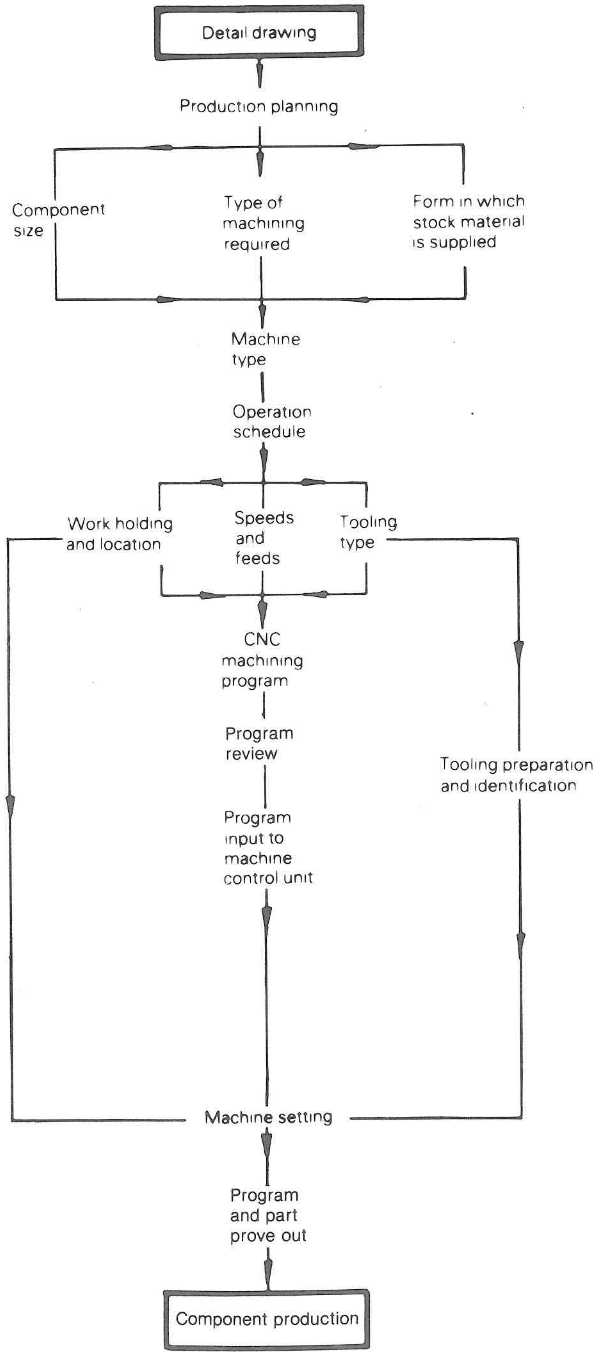


Figure 8.1 *Procedures associated with part programming.*

For relatively simple components the choice will be obvious and is likely to involve just one machine. On the other hand, more complex designs may require machining to be carried out on a second or perhaps third machine. It may be necessary to move from one machine to a second before returning to the original machine for further work, and so on.

Such transfers, and the stage at which they will take place, need to be determined clearly, since they will have a direct bearing on the preparation of appropriate machining programs.

Machine selection will also be influenced by component size, and the programmer must ensure that any machine used has the necessary physical capacity to accommodate the workpiece.

Decisions made at the component design stage relating to materials—whether the form of supply will be a casting or a solid bar, for instance—will also need to be considered, since this may have some bearing on work-holding and machine-loading arrangements.

PROCESSING OF MACHINING OPERATIONS

Having selected a machine capable of handling the required work, the next task confronting the part programmer is to decide on a suitable sequence of operations.

In order to do this effectively the programmer should ideally have a thorough understanding of the capabilities and operating procedures associated with the particular machine to be used, and adequate knowledge of the work-holding equipment and tooling that can be employed.

It is often the case that, giving due regard to safety requirements, a machining task can be tackled in more than one way with equally good results in terms of dimensional accuracy and surface finish. But the programmer must always bear in mind one objective is to complete the machining as quickly and efficiently as possible. There are two basic planning techniques that, when carefully considered, can make a significant contribution to achieving this objective.

The first is to carry out as much machining as possible at one work setting and to avoid unnecessary repositioning of the work, since this can be a very time-consuming business. The second is to carry out as much machining as possible with each cutting tool called, and to avoid unnecessary tool changing or indexing. The programmer should bear these points firmly in mind when listing the sequence of operations to be adopted.

The compilation of the process of operations to be used will not only be an aid to logical thinking throughout the rest of the part programming process, but it is also likely to be of value to the machine operator and may be required as a record for future reference. The more complex the component, the more vital the compilation of the process becomes.

It is likely that the operations process will form just part of the general documentation relating to a particular job, which will also contain information relating to work-holding, tooling, speeds, and feeds. The documentation relating to these aspects of part programming are discussed subsequently.

WORK-HOLDING AND LOCATION

The part programmer's responsibilities regarding work-holding and location are as follows:

(a) determine the work-holding device or devices to be used;
(b) determine if there will be a need to use supplementary support at any stage during a machining sequence;
(c) determine the means of ensuring accurate location of the workpiece prior to machining;
(d) document all matters relating to work setting that will have a direct effect on the validity of the part program and that will, therefore, be of importance to the machine set-up person.

Decisions made in relation to these factors are greatly influenced by component shape and size. Components of regular shape are usually accommodated in standard work-holding devices such as chucks, collets, and vises. Components of irregular shape often require special work-holding arrangements, and as a result demand extra attention from the programmer. He or she may find it necessary to include special slide movements in the program, solely to avoid collisions between the cutting tools and the clamping.

Similarly, the programmer will need to give special attention to components requiring supplementary support—the use of a center support or steady rest, for example—and may well have to include control of these features within the part program.

Multicomponent settings will also have a direct effect on the approach adopted when preparing the part program.

A special characteristic of CNC machining involving very high rates of metal removal is that considerble cutting forces may be exerted in a number of directions during the production of a single component, with very rapid change from one direction to another, possibly occurring without the safeguard of manual observation or intervention. This variation in cutting force direction means that the prime objective in work location, that of ensuring that the cutting forces are directed against an immovable feature in the work-holding arrangement, may not always be met when using standard equipment. For example, work held in a conventional machine vise is only positively located when the cutting forces are directed against the vise jaw. If the cutting force changes direction so that it is at 90° to the fixed jaw, there will be a frictional hold only, which is not foolproof.

When confronted with the problem of multidirectional cutting forces, the programmer should give full consideration to the alternative approaches available. Devices such as the grid plate will provide for positive location in several directions, but it may be necessary to use a specially devised fixture. A number of the project components included in Appendix C will require this approach.

It is possible that the work-holding equipment available is very limited in range, such as a machine vise. In this situation the programmer will have to make the best of arrangements such as the frictional hold described previously. For example, a reduction in metal removal rates will reduce the cutting forces exerted on the workpiece. Each problem encountered will require individual assessment, and the methods used to overcome the problem should be selected with reference to the high safety standards that are so essential in CNC machining.

Another factor that must be considered is that of geometric tolerances, as listed in Appendix D. When any of these are encountered on a part drawing, the programmer must ensure that the work-holding and location arrangements being used will enable them to be achieved. It is a further area of part programming that requires the programmer to be well versed in the practical side of CNC machining, and to have a full understanding of the capabilities and limitations of the work-holding devices that may be used.

In order that specified geometric requirements are satisfied, it may be necessary to adopt a special approach to work setting, or, as is more likely, work resetting before carrying out further operations. In such cases it is imperative that the part programmer indicates to the machine set-up person or operator his or her reasons for doing so. Such information is included in the general documentation relating to that particular workpiece.

The importance of positive location of the workpiece to absorb the forces exerted by the metal-cutting action has already been stressed. There is, however, another reason why the part programmer is concerned about precise location of the work. He or she will program the slide movements in relation to a datum that will be determined when the part program is prepared, and unless the part to be machined is precisely positioned in relation to that datum the intended machining features will not be achieved. Subsequent parts must also be positioned in exactly the same way to ensure uniformity of the product.

When establishing a program zero datum, the programmer will have to take into consideration the reference zero position that is an incorporated feature of the machine control system. The machine zero may or may not be in a fixed position. If it is fixed, it may be capable of being shifted on a temporary basis via the part program using the G92 preset code. It may be capable of being established anywhere within the operating range of the machine, or there may be limitations on repositioning. Whatever the circumstances, the programmer will need to understand them completely.

Consider first a control system that permits a machine zero to be established anywhere the programmer chooses. In this situation it may be considered that the correct programming approach is to establish a machine zero that will correspond with the chosen program zero. So for a component such as the one illustrated in Figure 8.2 the programmer selects the corner of the workpiece as zero for all programmed moves in the *X* and *Y* axes, and a 2 mm (0.1 in.) clearance between the top of the work and the *Z* axis zero. By selecting the upper left-hand corner of the part as program zero and machine zero, the programmer can use many part print dimensions in the part program. To ensure that there is correlation between the two zero positions the following machine setting approach will be necessary.

1. Set the corner of the vise jaw to zero in the *X* and *Y* axes (achieved by using a center locator or wiggler, or possibly an electronic probe).
2. Set the *Z* axis zero 2 mm (0.1 in.) above the work surface (achieved by touching on to a suitable worksetting block and calibrating the tool length offset accordingly).
3. Locate all workpieces using the corner of the fixed jaw of the vise as a reference position. (A plate attached to the vise jaw may be used to simplify this process).

The setting arrangement that accommodates the *X* and *Y* axes requirements is illustrated in Figure 8.3.

Consider now a situation involving a turned component such as that illustrated in Figure 8.4, and assume that the programmer has chosen to establish the face of the part as the *Z* datum zero and that the machine spindle center line is the *X* axis zero, as is normal. All that is required of the programmer is to ensure that the machine set-up person or operator is aware that the program datum is at the face of the work. The set-up person or operator will be required to establish the *Z* axis zero at the machine in the manner appropriate to that particular machine, and then ensure that all workpieces are all set to a measured overhang or to a stop as illustrated in Figure 8.5.

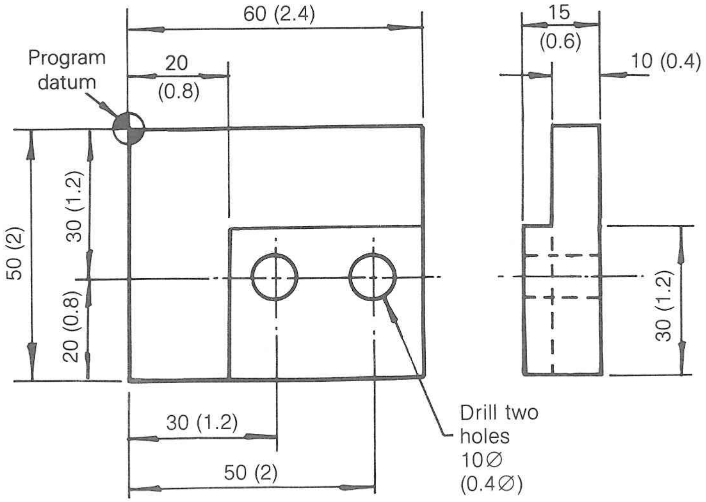


Figure 8.2 *Component detail. (Inch units are given in parentheses.)*

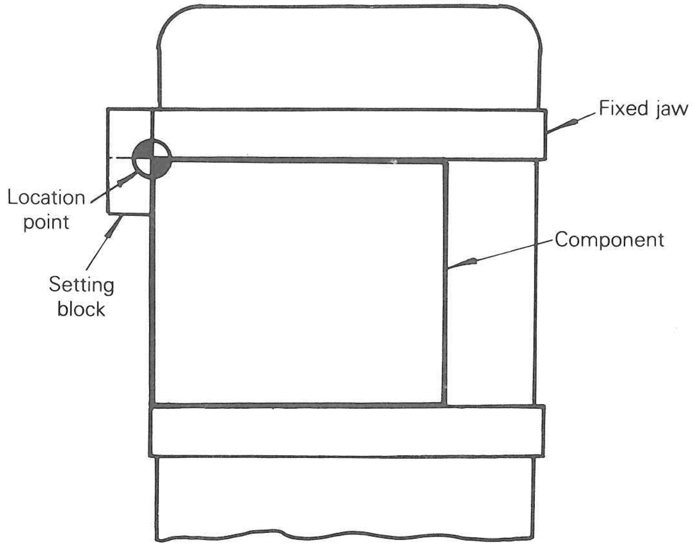
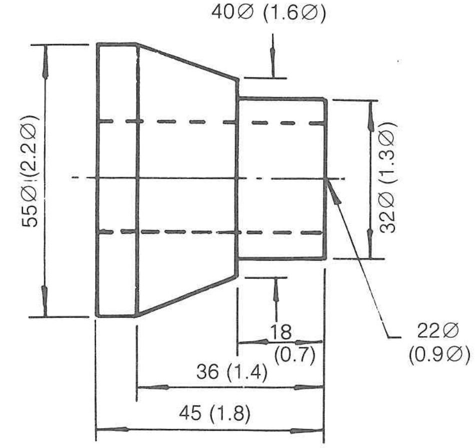


Figure 8.3 *Use of stop block on fixed jaw for component location.*



Material: medium carbon steel

Figure 8.4 *Component detail. (Inch units are given in parentheses.)*

It is often the case that turning centers have a set zero datum for the machine, usually at the back face of the chuck or a reference surface on the spindle nose.

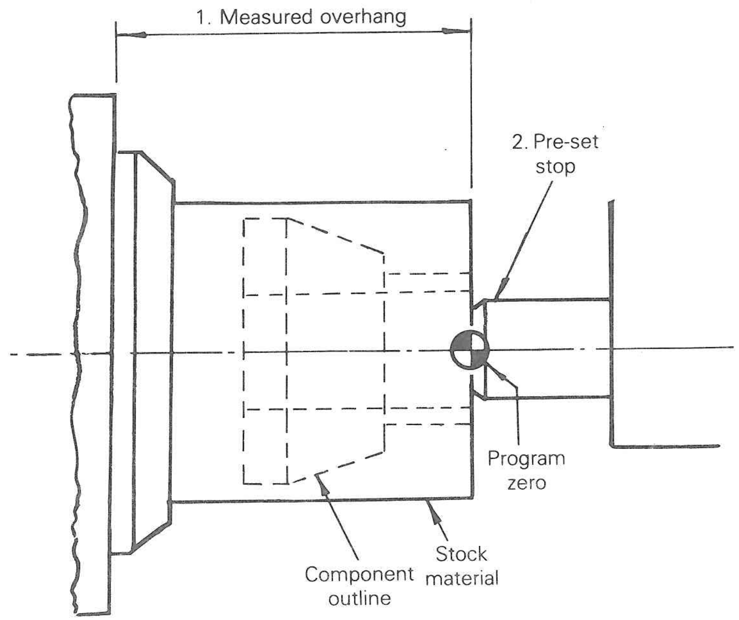


Figure 8.5 *Alternative work setting techniques to establish a datum for turned work.*

This type of zero cannot be changed but can be shifted on a temporary basis using the G92 preset axis code.

The programmer may choose to use the back face of the component as the program zero in the *Z* axis, a technique often applied when work is being produced from prepared billets. Work location is simple, and simply involves ensuring the material is firmly placed against the reference face. A further bonus is that all programmed slide movements will be positive.

To use the facility of repositioning the zero on a temporary basis—so that it corresponds to a program zero established at the workpiece face for instance—it will be necessary for the programmer to determine the amount of shift required to accommodate all the programmed movements in the *Z* axis. The dimensional value of the shift required, that is, the work overhang, must be documented. Eventually it will be entered into the program through the use of the G92 code followed by *X*, *Z*, and/or *Y* axis positional data as to what current slide positions should be. Some older machines may still use another method of establishing zero shifts through special offset tables, which can be activated by assigned codes like E, F, or H.

To ensure that the programmed machine movements achieve the desired effect, the work material has to be positioned accurately, either manually or automatically against stops, and this function is the responsibility of the machine set-up person or operator. The accuracy of this method of work setting can be improved if the overhang is slightly larger than the actual work requires, allowing a facing cut to be used early in the machining sequence to establish the new zero precisely.

It may be necessary to provide for more than one zero shift within the same turning program. A common situation is when the component length is such that, to ensure adequate support and to avoid chatter, part of the machining is carried out with a reduced overhang. After a programmed stop in the machining cycle, the operator repositions the work to suit the second zero position. Alternatively, the repositioning of the work may be achieved automatically through the program. This is particularly appropriate when a bar feed is utilized, the bar feeding to appropriate stops. The provision of a center support may also be a feature of such an arrangement. After the second zero shift all subsequent moves will be made in relation to that datum.

An example of a component which would involve two zero shifts during machining is shown in Figure 8.6. Because the diameter of the component is relatively small in proportion to its length, it would be advisable to use two settings and a center support for the second sequence of machining operations. The first setting involving the shift of the machine zero to the work face is illustrated in Figure 8.7(a), while the second setting requiring shifting the zero for a second time is shown in Figure 8.7(b).

The use of a second program zero is also applied to milling operations. An initial program zero is established and some machine movements will be made in relation to that datum. Then, via an appropriate program call the zero will be reestablished and all subsequent moves will be made in relation to the second datum.

One milling situation where the zero shift facility is particularly useful is when more than one component is to be machined at one setting, as illustrated in Figure 8.8. In this example a grid plate is used as a work-holding device. The advantage of the grid plate is that all the clamping and location points can be identified using a letter/number grid reference, like using a map reference

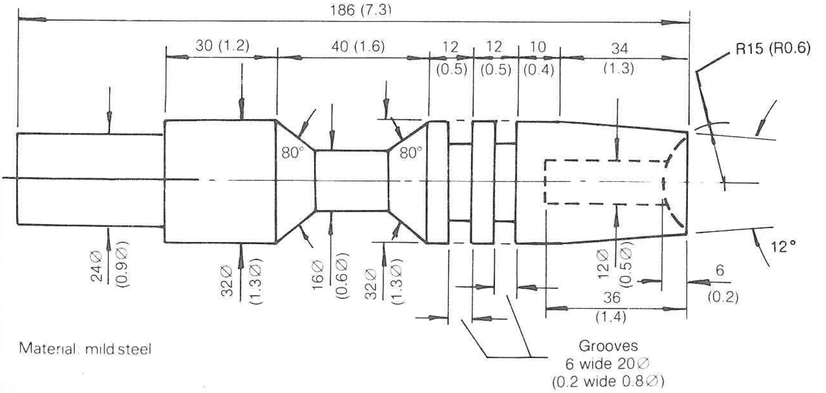


Figure 8.6 *Component detail. (Inch units are given in parentheses.)*

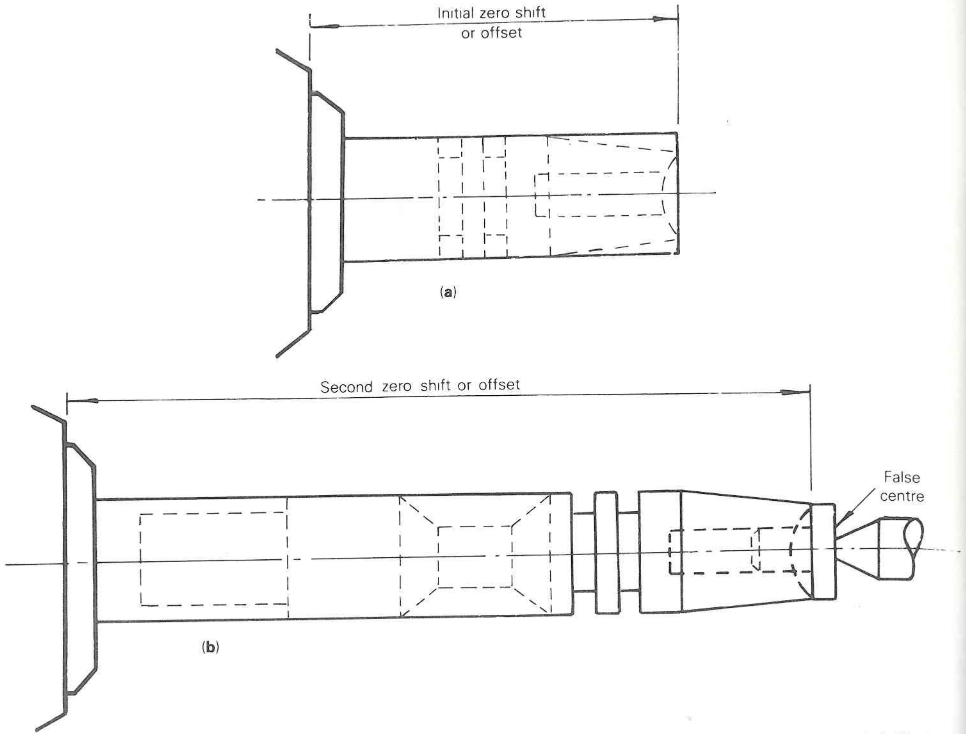


Figure 8.7 *The application of a second zero shift to accommodate work resetting* (a) *first work setting* (b) *second work setting.*

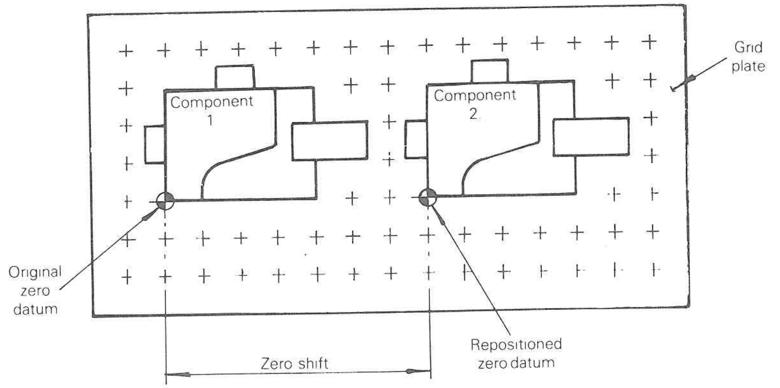


Figure 8.8 *Use of zero shift for multicomponent machining at one setting.*

to locate a particular town. Using this reference system the part programmer can instruct the machine set-up person/operator exactly where to position each component so that their location will correspond with the selected program zeros.

DOCUMENTATION ASSOCIATED WITH PART PROGRAMMING

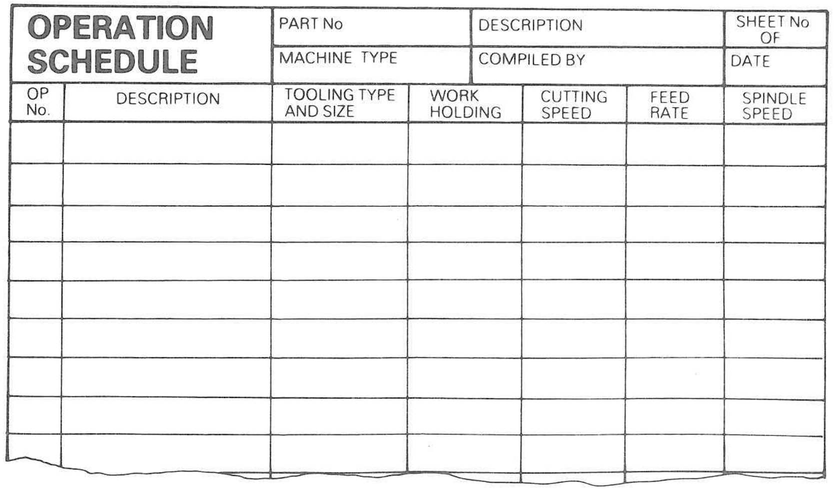
Before a part program can be compiled it is necessary to give some thought to the practical aspects of producing the component, and in most companies this is likely to involve the completion of an operation sheet. There is no standard operation sheet, and the format will vary from company to company. One which will meet the requirements of the exercises that follow is shown in Figure 8.9.

In addition to an operation sheet there is also the need for documentation relating to machine setting and tooling, because some of the decisions made during the operation planning stage, and which in turn are taken into account when writing the part program, are of direct concern to shopfloor personnel responsible for preparing the tooling and the machine. Again, there is no standard format for such documents. Each company will have its own procedure.

DOCUMENTATION RELATING TO MACHINE SETTING

Information regarding work-holding and location is of vital importance to the machine set-up person. He or she will also benefit from knowing the sequence of operations that has been adopted by the programmer. Also, it will be necessary to know the form in which the material to be machined is to be supplied. Ideally, all this information should be documented, not only as an aid to efficiency on the shop floor, but also to provide a record for future reference.

The documents used to convey this information will vary from company to company, and the precise way this information is disseminated is not of major



OPERATION SCHEDULE		PART No		DESCRIPTION		SHEET No OF
		MACHINE TYPE		COMPILED BY		DATE
OP No.	DESCRIPTION	TOOLING TYPE AND SIZE	WORK HOLDING	CUTTING SPEED	FEED RATE	SPINDLE SPEED

Figure 8.9 *Example of an operation schedule.*

importance. The important thing is that the shop floor personnel fully understand what is required. So how detailed does the information need to be? The answer depends on the complexity of the component and the machining operations involved.

Assume that the machine set-up person knows the sequence of machining operations involved and is to proceed with setting up the machine. Consider in the first instance work loading, holding, and location. What information is required?

A simple component that is to be turned in one set up from a prefaced billet could be accommodated with a few short notes as follows:

Material:	prepared billet, part number ****
Loading:	manual
Work-holding:	chuck type, fixture number ****
Location:	back face of chuck
Zero shift:	Z direction + or −, and value

The last item would indicate that a manual data entry shifting the Z axis zero from the spindle face to the workpiece face is required.

A more complex component requiring two settings, with the second operation requiring center support activated by an entry in the program, will require a little more detail and the information may be given as follows:

Material:	diameter and length of bar stock
Loading:	bar feed to programmed stops, bar stop number
Work-holding:	collet, with programmed center support for second setting
Zero shifts:	first setting, direction + or −, and value second setting, direction + or −, and value

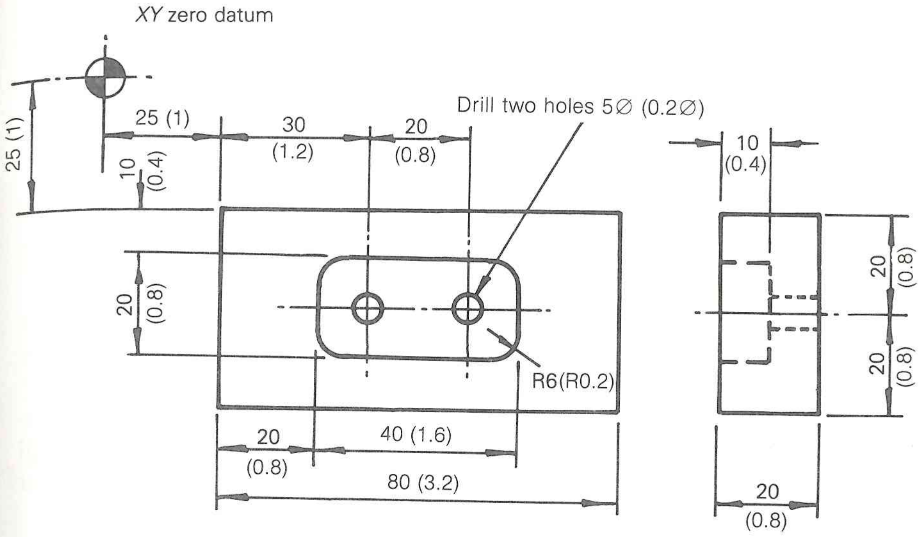
This information could be supplemented by two simple sketches showing the machining to be carried out at each setting.

A simlar exercise can be carried out for workpieces involving milling. The exercise shown in Figure 8.10 could be produced on a "one part" basis or involve a multicomponent setting.

In the first instance the workpiece could be located using the corner of the fixed jaw of the vise as a reference point, a technique referred to on page 168. The instructions necessary to achieve this would be as follows:

Material:	prepared blank length × width × height
Work-holding:	machine vise, fixture number ****
Location:	left-corner of fixed jaw
Program datum:	X axis −25 mm (−1 in.) (axis-direction + or − value) Y axis 25 mm (1 in.) Z axis 2 mm (0.1 in.)

Again the information regarding the program datum may be more readily understood if the instructions include a sketch.



Material: aluminum alloy

Figure 8.10 *Component detail. (Inch units are given in parentheses.)*

A multicomponent setup involving the same component could involve the use of a grid plate. To convey the necessary set-up information, the programmer should be familiar with the grid plate and its associated locating and clamping devices. With such knowledge he or she may be able to give detailed instructions for the complete setting, using the grid references to position the various setting blocks, locating dowels, and clamps to be used in the operation. On the other hand, a competent set-up person could manage with the basic information included in Figure 8.11.

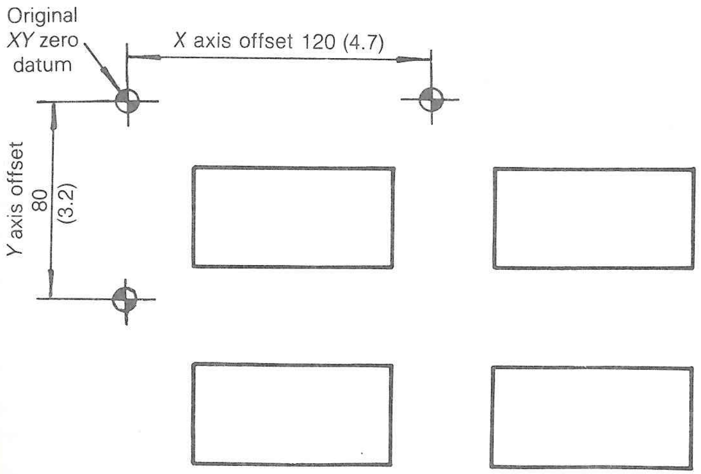


Figure 8.11 *Use of grid plate. (Inch units are given in parentheses.)*

TOOLING SELECTION AND IDENTIFICATION

The responsibilities of the part programmer concerning tooling are as follows:

(a) determine the appropriate tools to be used, including their shape and size and the material from which they will be made;
(b) allocate identification numbers to facilitate machine setting;
(c) allocate tool offset numbers;
(d) determine, when appropriate, the dimensonal value of the offsets;
(e) prepare appropriate documentation.

It is essential that a programmer is fully conversant with the tooling system for the machine involved, that is, the type of tooling that can be used and the way the tools can be located and held in position.

A major feature of CNC machining is the use of standard tooling. The intricate slide movements that are possible greatly minimize the need for special tooling, particularly form tools. In many ways the tooling requirements for CNC machining are less complex than for conventional machining.

Providing the programmer is conversant with the machine tooling system, the process of selecting tooling for a particular job is largely a case of selecting and utilizing standard items.

It is important that the correct tool material is used, particularly when using carbide inserts. Reference should be made to manufacturers' literature for guidance in this respect. Pages 44 through 46 give an indication of the type of information that is available.

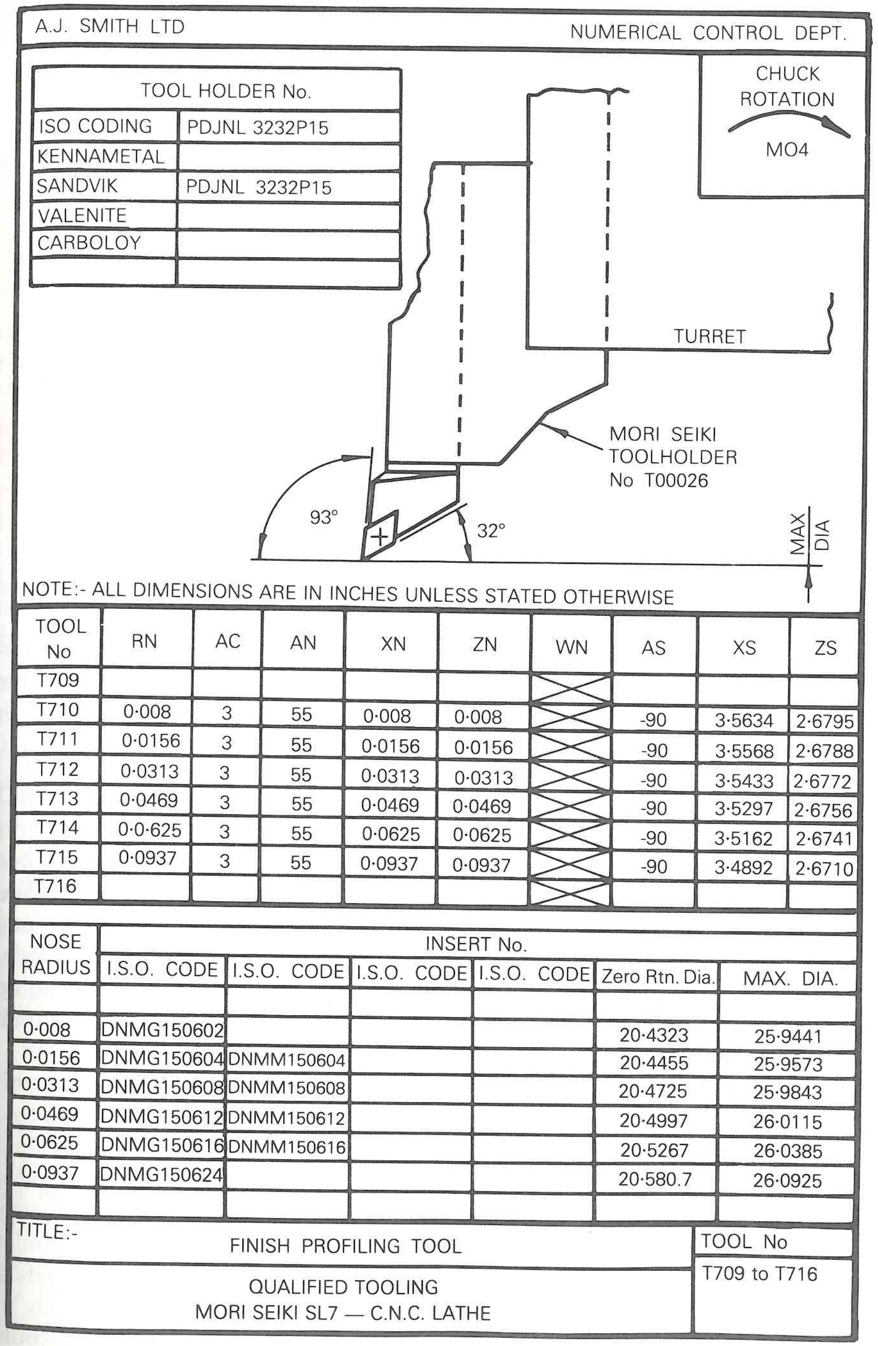
It is often the case that the tools available within a company for use on a particular machine will be further standardized with their details being documented. An example of a company-based tool standard is shown in Figure 8.12.

All tools are required to have a numerical identity within the part program. This identity, commonly the letter T followed by two digits, is allocated by the part programmer and will correspond with the numbered position the tool will occupy in the machine turret, magazine, or other storage facility. The position each tool will occupy is affected by factors which are discussed below.

Commonly used tools are often given an identity that is retained at all times, since this often eliminates the need to reset when jobs are changed. When this situation exists, it is essential that the part programmer knows exactly which tools are involved and their numerical identity.

TOOL STORAGE

With automatic tool changing facilities involving turrets, the positions for the tools in the turret are numbered. Thus a tool call of, say, T06 will cause the turret to index to position number six. The tool allocated the numerical identity 6 must be set in position six.



A.J. SMITH LTD NUMERICAL CONTROL DEPT.

TOOL HOLDER No.	
ISO CODING	PDJNL 3232P15
KENNAMETAL	
SANDVIK	PDJNL 3232P15
VALENITE	
CARBOLOY	

CHUCK ROTATION
MO4

NOTE:- ALL DIMENSIONS ARE IN INCHES UNLESS STATED OTHERWISE

TOOL No	RN	AC	AN	XN	ZN	WN	AS	XS	ZS
T709									
T710	0·008	3	55	0·008	0·008		-90	3·5634	2·6795
T711	0·0156	3	55	0·0156	0·0156		-90	3·5568	2·6788
T712	0·0313	3	55	0·0313	0·0313		-90	3·5433	2·6772
T713	0·0469	3	55	0·0469	0·0469		-90	3·5297	2·6756
T714	0·0·625	3	55	0·0625	0·0625		-90	3·5162	2·6741
T715	0·0937	3	55	0·0937	0·0937		-90	3·4892	2·6710
T716									

NOSE RADIUS	INSERT No.					
	I.S.O. CODE	I.S.O. CODE	I.S.O. CODE	I.S.O. CODE	Zero Rtn. Dia.	MAX. DIA.
0·008	DNMG150602				20·4323	25·9441
0·0156	DNMG150604	DNMM150604			20·4455	25·9573
0·0313	DNMG150608	DNMM150608			20·4725	25·9843
0·0469	DNMG150612	DNMM150612			20·4997	26·0115
0·0625	DNMG150616	DNMM150616			20·5267	26·0385
0·0937	DNMG150624				20·580.7	26·0925

TITLE:- FINISH PROFILING TOOL

QUALIFIED TOOLING
MORI SEIKI SL7 — C.N.C. LATHE

TOOL No
T709 to T716

Figure 8.12 *Company-devised tool standard. (All units are given in millimeters.)*

Similarly, tools changed by automatic handling devices will be housed in readiness in a tooling magazine. When a tool is called the magazine will index to bring the appropriate tooling station into a position where the tool located in that station can be accessed by the handling device. Clearly the correct tool must be in each numbered position if the programmed tool call is to bring the desired tool into the machining position.

Even when the tool change is a manual operation, effected by a programmed stop in the machining cycle, the process is assisted if the operator has a clear indication of the next tool to be used. It is usual, therefore, to number the tool storage positions or even the tools themselves. When the programmed break in machining occurs, the operator can refer to a document provided by the programmer to determine the next tool involved; on the more sophisticated control systems the tool may be indicated by a message displayed on the visual display unit of the control.

The programmer should give due thought to the positioning of the tools in relationship to each other in the turret or magazine. Most indexing arrangements involve rotation in one direction only, so to change, say, from T03 to T06 will require three indexing moves, two of which are time-consuming and unproductive. Therefore the objective should be to position the tools in the turret or magazine in the order in which they will be called into use, although this is not always possible in practice.

The problem of wasteful indexing time is considerably eased when the machine is equipped with the facility to index tooling by the shortest possible route. In other words, the turret or magazine will rotate either clockwise or counterclockwise depending on which tool is called.

TOOL CHANGING POSITION

The programmer should consider carefully the position the machine slides are to be in when a tool change is made. There is a tendency, particularly among students, to return the machine slides to a set position before making a change, a practice that may have its merits from a safety point of view early in training but which, like wasteful indexing moves, can add considerably to the total time taken to machine the part.

The objective must be to keep noncutting slide movement to a minimum. For example, on a vertical machining center it is often possible to effect a tool change immediately above the point at which the tool completes the required machining, the change being carried out after an appropriate *Z* up-movement of the machine spindle or head. This saves making a long and unnecessary journey to a set position such as the *XY* zero datum. On turning centers a similar time saving can be achieved by indexing as near to the workpiece as is safely possible. The programmer should always refer to the machinery manuals to check clearances necessary to allow tool indexing mechanisms and cutting tools to clear any obstructions.

REPLACEMENT TOOLING

For long production runs the programmer will need to give some thought to the provision of replacement tooling.

When tools need to be replaced it is possible for the set-up person to determine suitable offsets and make the necessary tool data entries as he or she would for the original tools, but this is time-consuming and interrupts production.

An alternative approach is to use replacement tooling which is identical to the original. Such identical tooling may be of two types, namely, "qualified" or "preset."

Qualified tooling is used on turning centers and has dimensions guaranteed by the manufacturer to within ±0.0005 in. or ±0.08 mm from up to three datum faces.

Preset tooling is precisely set to predetermined dimensions in the toolroom and is applied to turning tools and milling cutters.

The programmer may choose to recommend qualified or preset tools when compiling his or her tooling schedule, but if such tooling is prescribed, the programmer may need a feedback of information from the toolroom regarding the setting sizes. This information then becomes part of the overall programming and machine-setting package and should be documented for future reference.

TOOLING DOCUMENTATION

Documentation regarding tooling, as with machine setting instructions, may be simple or relatively complex. It depends largely on the size of the company and the degree of organization that exists.

The possibilities range from the situation where the machine set-up person has personal access to the range of tooling likely to be required, to situations where the tooling is prepared in a special-purpose tool room, issued to the set-up person as a package for that particular job, and on completion returned to the tool room for refurbishment and storage.

For each programmed tool the minimum information required on the shop floor is as follows:

(a) programmed identity—T01, T02, T03, etc.;
(b) tool type;
(c) holder type and size;
(d) insert type and size;
(e) overall dimensions (solid tools);
(f) projection of cutting tool from holder.

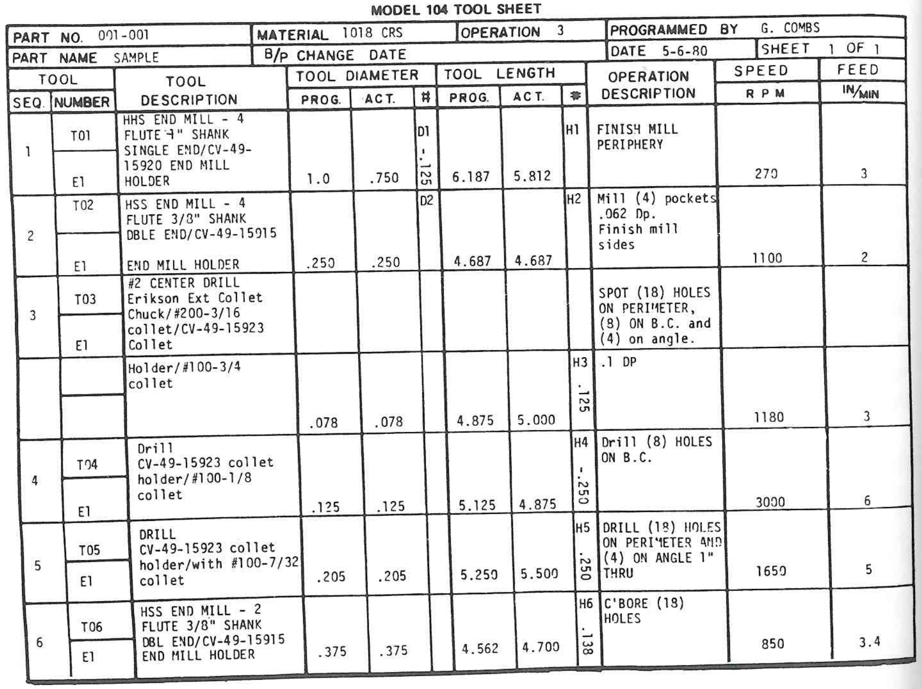
When presetting is involved, the tool design or program personnel usually determine the original preset dimensions. The sizes should ultimately be notified to the part programmer so that they may be recorded and included as part of the general documentation for that particular job. A well-organized tool preparation facility may well retain the data against their own job reference to facilitate the preparation of replacement tooling and to allow for the possibility of having to prepare identical tools at some future time. See Figures 8.13 and 8.14 for examples of tool data sheets.

When tooling offsets are being used to achieve a particular machining effect, as discussed on page 225, the value of the offsets must be included on the document.

It is often the situation that information regarding tooling, and sometimes information relating to machine setting, is included on the original part program form when one is used. Information documented in this way is of necessity rather brief, but in many cases is adequate.

Another practice widely adopted is to give tooling details alongside the tool call in the part program. Again, the information is brief but adequate for many situations.

The important thing is that the part programmer fully appreciates the needs of the people more directly concerned with the machining operation. There must be an efficient transfer of the relevant information. The means adopted



MODEL 104 TOOL SHEET

PART NO. 001-001
MATERIAL 1018 CRS
OPERATION 3
PROGRAMMED BY G. COMBS
PART NAME SAMPLE
B/P CHANGE DATE
DATE 5-6-80
SHEET 1 OF 1

TOOL		TOOL DESCRIPTION	TOOL DIAMETER			TOOL LENGTH			OPERATION DESCRIPTION	SPEED	FEED
SEQ.	NUMBER		PROG.	ACT.	#	PROG.	ACT.	#		R P M	IN/MIN
1	T01 E1	HHS END MILL - 4 FLUTE 1" SHANK SINGLE END/CV-49-15920 END MILL HOLDER	1.0	.750	D1 -.125	6.187	5.812	H1	FINISH MILL PERIPHERY	270	3
2	T02 E1	HSS END MILL - 4 FLUTE 3/8" SHANK DBLE END/CV-49-15915 END MILL HOLDER	.250	.250	D2	4.687	4.687	H2	Mill (4) pockets .062 Dp. Finish mill sides	1100	2
3	T03 E1	#2 CENTER DRILL Erikson Ext Collet Chuck/#200-3/16 collet/CV-49-15923 Collet							SPOT (18) HOLES ON PERIMETER, (8) ON B.C. and (4) on angle.		
		Holder/#100-3/4 collet	.078	.078		4.875	5.000	H3 .125	.1 DP	1180	3
4	T04 E1	Drill CV-49-15923 collet holder/#100-1/8 collet	.125	.125		5.125	4.875	H4 -.250	Drill (8) HOLES ON B.C.	3000	6
5	T05 E1	DRILL CV-49-15923 collet holder/with #100-7/32 collet	.205	.205		5.250	5.500	H5 .250	DRILL (18) HOLES ON PERIMETER AND (4) ON ANGLE 1" THRU	1650	5
6	T06 E1	HSS END MILL - 2 FLUTE 3/8" SHANK DBL END/CV-49-15915 END MILL HOLDER	.375	.375		4.562	4.700	H6 .138	C'BORE (18) HOLES	850	3.4

Figure 8.13 *Model 104 tool sheet. Ex-Cell-O Corp., Rockford Machine Tool Company, Rockford, IL*

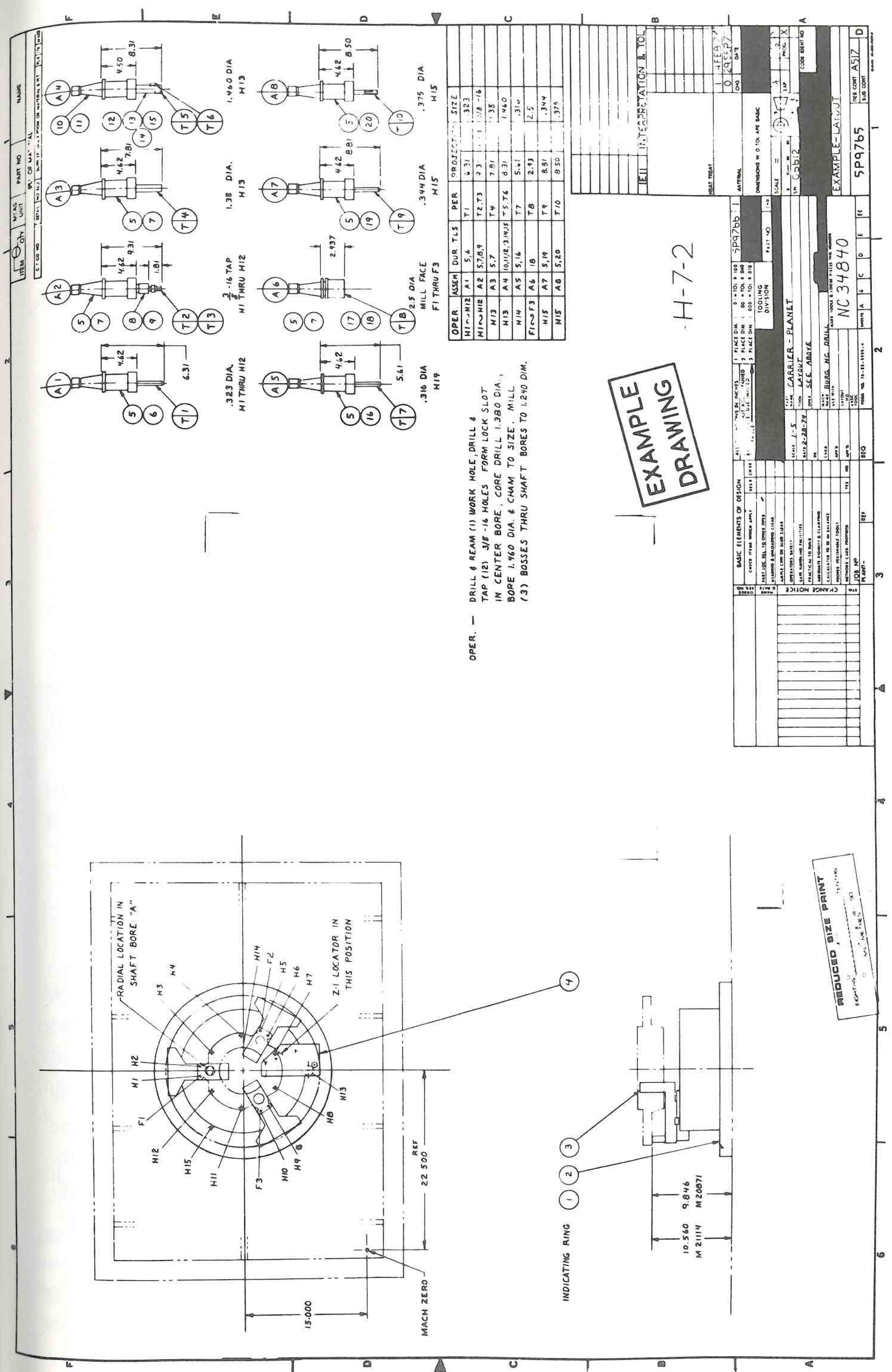


Figure 8.14 *Sample tool layout. Numerical Control Society, Glenview, IL*

to achieve this objective will vary, but the programmer should always remember that it is a very important aspect of his or her work.

PART PROGRAMMING PROCEDURE

The blocks of data entered in a part program are numbered N01, N02, N03, and so on. On completion of a machining program it is usually necessary to return to the beginning so that another component can be machined. The return to the program start position is usually achieved via a "rewind" or "return to start" command included at the end of the program.

With word address systems, this command is entered as a miscellaneous function designated M30, which has the effect of stopping all slide and spindle movement, turning off the coolant supply and rewinding the tape. When the tape has merely been used to transfer a program into the microcomputer memory, then it rewinds the program within the microcomputer. The stage at which this rewind must cease has to be identified, and this is achieved via a "rewind stop" program entry signified by the % sign. This is usually the first entry in a word address program.

With the start of the program established, the next three or four blocks of data will concern setting the machine controller so that it interprets subsequent data in the correct manner. These set-up entries include instructions relating to the following:

(a) *units,* which may be programmed in inch or metric;
(b) *slide movement,* which may be stated as incremental or absolute dimensional values;
(c) *speed,* which may be programmed as surface speed in feet/meters per minute or spindle speed in revolutions per minute;
(d) *feed,* which may be programmed as inches/millimeters per spindle revolution or inches/millimeters per minute.

Having established the basic set-up data, it may be helpful now to list in a general way the functions and machine movements necessary to produce the component. Consider the drawing for Exercise 1 (in Appendix C) and imagine that the machine is set with the spindle in its 'home' or 'base' datum position, that is, at a point some distance above the *XY* datum indicated on the drawing. Starting from this position, the part program must provide for the following:

1. Rapid linear movement to P1 in *X* and *Y*.
2. Rapid linear movement to a clearance position above *Z*0.

3. Spindle on clockwise direction.
4. Coolant on.
5. Feed linear movement to *Z* depth.
6. Rapid linear movement to clearance above *Z*0.
7. Rapid linear movement to P2.
8. Feed linear movement to *Z* depth.
9. Rapid linear movement to clearance above *Z*0.

. . . and so on.

These simple comments can, providing a space exists, be entered directly onto a program sheet or, if the program is being listed on plain paper, alongside each item of data, but it is probably a better plan to prepare a rough list in the first instance and then check carefully to ensure nothing has been overlooked. Relative codes and data can then be added to each statement.

Should it be found that, on completion of a program, omissions have inadvertently been made, the error can be rectified more easily if the block numbers are allocated in increments of five: N01, N05, N10, N15. It is then a simple matter to include additional blocks—N06, N07, N08, for instance—between N05 and N10.

If the program is being listed on a computer the blocks can be numbered consecutively, since any omission entered via the keyboard will automatically cause the existing blocks to renumber or, alternatively, renumbering can be easily effected. Many MDI control systems also have this facility.

A methodical approach to part programming is essential, and it is recommended that, even for a simple component, an operation schedule listing the tooling speeds and feeds to be used should be completed in the first instance.

WORD ADDRESS PROGRAMMING

Word address programming is largely based on an International Standards Organization (ISO) and Electronic Industries Association (EIA) code that require the program to be compiled using codes identified by letters, in particular G and M. Each code addresses, or directs, the item of data it precedes to perform a certain function within the control system.

The ISO and EIA Standards provided for 99 G codes and an identical number of M codes, each being exprssed by the address letter followed by two digits.

Not all the codes were allocated a specific function in the Standard and this gave the manufacturers of control systems the opportunity to introduce their own variations. There is, therefore, no standard word address machine programming language, although many of the recommendations made have been widely adopted.

The G codes, or preparatory functions, are used to set up the machine control unit modes of operation required for the machining that is to be carried out—whether movement is to be in a straight line/linear or radially/circular, for example. In general they relate to slide motion control. Examples of commonly used G codes are as follows:

G00	Rapid linear positioning, point to point
G01	Linear positioning at a controlled feed rate
G02	Circular interpolation, clockwise
G03	Circular interpolation, counter-clockwise
G04	Dwell for programmed duration
G33	Thread cutting, constant lead
G34	Thread cutting, increasing lead
G40	Cutter compensation, cancel
G41	Cutter compensation, left
G42	Cutter compensation, right
G70	Inch programming
G71	Metric programming
G80	Series associated with drilling, boring, tapping and reaming.

(For a complete list of G codes refer to Chapter 6.)

G codes may be "modal," that is, they remain active until cancelled. Alternatively they may be nonmodal, and are only operative for the block in which they are programmed.

The M codes, or miscellaneous functions, are used to establish requirements other than those related to slide movement. For example, they are used to activate spindle motion or to turn on a coolant supply. Examples of commonly used M codes are as follows:

M00	Program stop
M01	Optional stop
M02	End of program
M03	Spindle on clockwise
M04	Spindle on counter-clockwise
M05	Spindle off
M06	Tool change
M08	Coolant on
M09	Coolant off
M30	End of tape

(For a complete list of M functions refer to Chapter 6.)

As with G codes, some M functions are modal, remaining active until cancelled. M functions may also become active immediately upon reading of the block or after all block commands are completed. (Refer to machinery manuals to determine how various codes operate.)

In addition to the address letters G and M there is also common usage of S, F, and T to indicate speeds, feeds, and tooling. The letter N is always used to identify block numbers.

The distinction between word address and conversational programming is best appreciated by reference to the simple movements discussed earlier.

To program the linear movement of −39.786 mm or −1.6 in. in the *X* axis using the word address technique, it is first necessary to establish the operating mode required. This is done by including the appropriate G code, in this case G01. Thus the complete program entry for the required move will be:

Inch N260 G01 X-1.6
Metric N260 G01 X-39.786

Similarly, reconsider the 0.3 in. or 8 mm radial movement through an arc of 90°. Once again the mode of operation has to be established using the appropriate G code, which for circular movement in a clockwise direction is G02. It will also be necessary to define the target position in the appropriate axes and also the start of the arc in relation to the arc center using I, J, and K address letters that correspond to the *X*, *Y*, and *Z* axes respectively. A word address progrm entry to achieve this movement would read as follows:

Inch N350 X1.7 Z-3.0 K.3
Metric N350 G02 X43.765 Z-75.000 K8

There are variations in procedure even when word address programming such a common machining feature as a radius. On some control systems the arc center may have to be defined—still using the I, J, and K address letters—in relation to the program datum and not the start position.

The programming of radial movements using the word address method will be returned to later in the text.

A word address program that includes a number of codes in both inch and metric is listed below. The program relates to the component detailed in Figure 8.15, and is typical of its type. The comments written alongside the data should convey to the reader an impression of how, prior to programming, the machining of a component is first broken down into operations. It also shows how the necessary machine control data are presented. Later in the text further reference will be made to the program to illustrate specific programming techniques and features.

PROGRAMMING EXAMPLE

Figure 8.15 shows a simple turned component for which a part program is to be prepared using the following basic programming information. (Examples of more detailed programming specifications are given in Appendix C.)

PREPARATORY FUNCTIONS (G CODES)

G00	Rapid movement
G01	Linear interpolation—movement at a programmed feed rate
G02	Circular interpolation, clockwise
G03	Circular interpolation, counter clockwise
G40	Cancel tool nose radius compensation
G41	Tool nose radius compensation left
G42	Tool nose radius compensation right
G70	Inch units
G71	Metric units

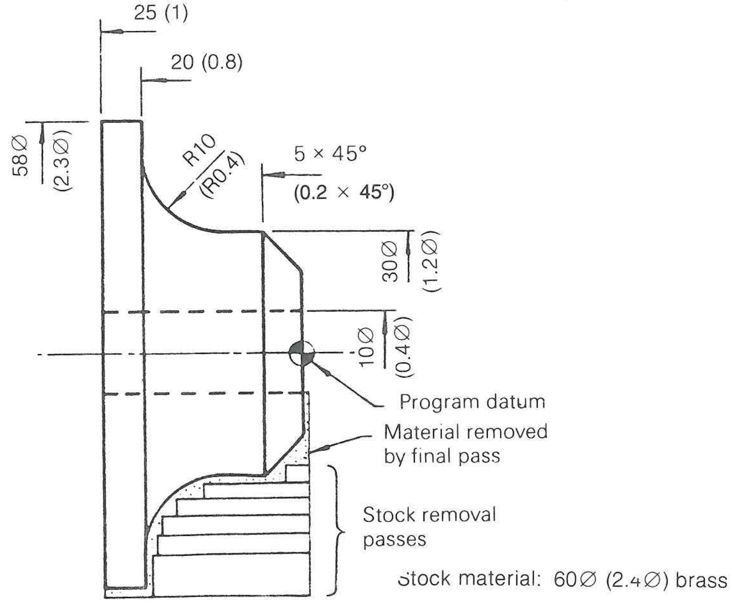


Figure 8.15 *Component detail. (Inch units are given in parentheses.)*

G90	Absolute positioning data	X axis values to be programmed as diameters
G91	Incremental positioning data	
G94	Feed (in. or mm)/min	
G95	Feed (in. or mm)/rev	
G96	Constant surface cutting speed	
G97	spindle speed rev/min	

MISCELLANEOUS FUNCTIONS (M CODES)

M00	Program stop
M01	Optional stop
M02	End program
M03	Spindle on clockwise
M04	Spindle on counterclockwise
M05	Spindle off
M06	Tool change
M08	Coolant on
M09	Coolant off
M30	End of program

OPERATION SCHEDULE

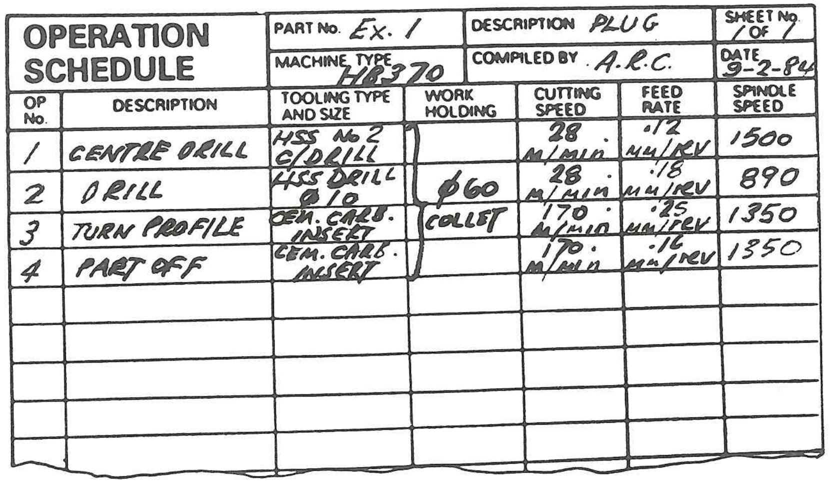
The first stage in the programming process is to prepare an operation schedule. An operation schedule for the component is shown in Figure 8.16, where only metric units are shown. The spindle speeds and feed rates have been determined by reference to the cutting data given in Chapter 7.

TOOLING INFORMATION

Although the component is a relatively simple one, it is still necessary to provide tooling information for the machine tool setter. This information is detailed on the form illustrated in Figure 8.17, where only metric units are shown.

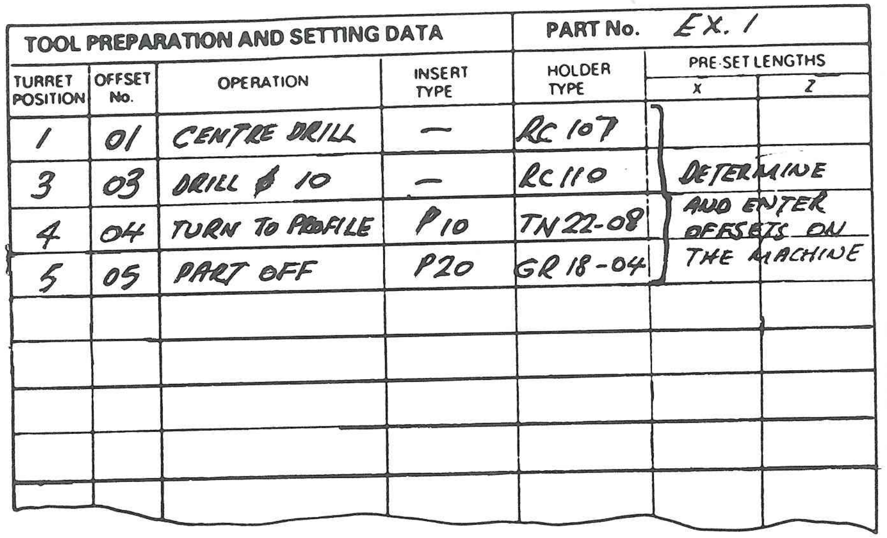
PROGRAM LISTING

Attention can now be given to listing the necessary programming data, together with appropriate remarks to ensure a logical approach is being adopted and to



OPERATION SCHEDULE		PART No. Ex. 1		DESCRIPTION PLUG		SHEET No. 1 OF 1
		MACHINE TYPE HB370		COMPILED BY A.R.C.		DATE 9-2-84
OP No.	DESCRIPTION	TOOLING TYPE AND SIZE	WORK HOLDING	CUTTING SPEED	FEED RATE	SPINDLE SPEED
1	CENTRE DRILL	HSS No 2 C/DRILL	∅60 COLLET	28 M/MIN	.12 MM/REV	1500
2	DRILL	HSS DRILL ∅10		28 M/MIN	.18 MM/REV	890
3	TURN PROFILE	CEM. CARB. INSERT		170 M/MIN	.25 MM/REV	1350
4	PART OFF	CEM. CARB. INSERT		170 M/MIN	.16 MM/REV	1350

Figure 8.16 *(Metric units)*



TOOL PREPARATION AND SETTING DATA				PART No. EX. 1		
TURRET POSITION	OFFSET No.	OPERATION	INSERT TYPE	HOLDER TYPE	PRE-SET LENGTHS	
					X	Z
1	01	CENTRE DRILL	—	RC 107	DETERMINE AND ENTER OFFSETS ON THE MACHINE	
3	03	DRILL ∅ 10	—	RC 110		
4	04	TURN TO PROFILE	P10	TN 22-08		
5	05	PART OFF	P20	GR 18-04		

Figure 8.17 *(Metric units)*

ensure that nothing is overlooked. The required program is listed below. (Note that, in this particular case, a programming form is not being used, but partially completed programming exercises involving the use of a form are given in Appendix C.)

PART PROGRAM (INCH)

```
      Data                           Remarks
N10   G70 G90                        Absolute inch
N15   G95 G97                        Feed inches/rev Spindle speed rev/min
N20   G92 X4.0 Z8.0                  Pre-set safe turret indexing position
N25   T0101 M06                      Tool change. Tool No. 1. Off-set No. 1
N30   S3000 M03                      Spindle on clockwise
N35   G00 X0 Z.1 M08                 Rapid to start position. Coolant on
N40   G01 Z-.3 F.003                 Center drill
N45   G00 Z.1                        Rapid retract
N50   X4.0 Z8.0                      Return to turret index position
N55   T0202 M06                      Tool change. Tool No. 2 Off-set No. 2
N60   S2380 M03                      Spindle Speed
N65   G00 X0 Z.1                     Rapid to start position
N70   G01 Z-1.2 F.015 M08            Drill through .4ø
N75   G00 Z.1                        Rapid retract
N80   X4.0 Z8.0                      Return to turret index position
N85   T0303 M06                      Tool change. Tool No. 3 Off-set No. 3
N90   S1430 M03                      Spindle speed
N95   G00 X1.97 Z.1                  Rapid to start position
N100  G01 Z-.768 F.020 M08
N105  G00 X2.05 Z-.688               Rapid retract to clear cut surface
N110  Z.1                            Second rough pass—start position
N115  X1.772
N120  G01 Z-.748
N125  G00 X1.85 Z-.669
N130  Z.1
N135  X1.575                         Third rough pass—start position
N140  G01 Z-.709
N145  G00 X1.654 Z-.63
N150  Z.1
N155  X1.417                         Fourth rough pass—start position
N160  G01 Z-.65
N165  G00 X1.496 Z-.57
N170  Z.1
N175  X1.26                          Fifth rough pass—start position
N180  G01 Z-.512
N185  G00 X1.339 Z-.433
N190  Z.1
N195  X.984                          Sixth rough pass—start position
N200  G01 Z-.079
N205  G00 X1.063 Z.1
```

Data		Remarks	
N210	G41	Cutter radius compensations	Finish machine profile
N215	X0	Rapid to *X* zero	
N220	G01 Z0	Rapid to *Z* zero	
N225	S1750 M03	Spindle speed and feed rate change	
N230	X1.0 F.007	Machine face to 1.0ø	
N235	X1.2 Z-.2	Machine chamfer	
N240	Z-.4	Linear move to radius start	
N245	G03 X2.0 Z.8 I.4	Circular interpolation	
N250	G01 X2.3	Linear move to 2.3ø	
N255	Z-1.1	Linear move to length	
N260	G00 X2.5	Lift from finished surface	
N265	G40	Cancel cutter radius compensation	
N270	X4.0 Z8.0	Return to turret index position	
N275	T0404 M06	Tool change. Tool No. 4 Parting Tool	
N280	S1430 F.007	Spindle speed and feed rate change	
N285	G00 X2.5 Z-1	Rapid to start	
N290	G01 X.08 M08	Part off leaving stock faced	
N295	G00 X4.0 Z8.0	Return to turret index position	
N300	G92 X0 Z0 M30	Program end. Spindle and coolant off	

Note: To simplify the program, neither tool nose radius or thickness were used.

PART PROGRAM (METRIC)

Data		Remarks	
N10	G71 G90	Absolute metric	
N15	G95 G97	Feed mm/rev Spindle speed rev/min	
N20	G92 X100 Z200	Pre-set safe turret indexing position	
N25	T0101 M06	Tool change. Tool No. 1. Off-set No. 1	
N30	S3000 M03	Spindle on clockwise.	
N35	G00 X0 Z2 M08	Rapid to start position, coolant on	
N40	G01 Z-8 F.1	Center drill	
N45	G00 Z2	Rapid retract	
N50	X100 Z200	Return to turret index position	
N55	T0202 M06	Tool change. Tool No. 2 Off-set No. 2	
N60	S2380 M03	Spindle speed	
N65	G00 X0 Z2	Rapid to start position	
N70	G01 Z-30 F.18 M08	Drill through 10ø	
N75	G00 Z2	Rapid retract	
N80	X100 Z200	Return to turret index position	
N85	T0303 M06	Tool change. Tool No. 3 Off-set No. 3	
N90	S1430 M03	Spindle speed	
N95	G00 X50 Z2	Rapid to start position	
N100	G01 Z-19.5 F.3 M08		
N105	G00 X52 Z-17.5	Rapid retract to clear cut surface	
N110	Z2		
N115	X45	Second rough pass—start position	
N120	G01 Z-19		
N125	G00 X47 Z-17		
N130	Z2		
N135	X40	Third rough pass—start position	
N140	G01 Z-18		
N145	G00 X42 Z-16		
N150	Z2		
N155	X36	Fourth rough pass—start position	
N160	G01 Z-16.5		
N165	G00 X38 Z-14.5		
N170	Z2		
N175	X32	Fifth rough pass—start position	
N180	G01 Z-13		
N185	G00 X34 Z-11		
N190	Z2		
N195	X25	Sixth rough pass—start position	
N200	G01 Z-2		
N205	G00 X27 Z2		
N210	G41	Cutter radius compensations	Finish machine profile
N215	X0	Rapid to *X* zero	
N220	G01 Z0	Rapid to *Z* zero	
N225	S1750 M03	Spindle speed	
N230	X20 F.15	Machine face to 20ø	
N235	X30 Z-5	Machine chamfer	
N240	Z-10	Linear move to radius start	
N245	G03 X50 Z-20 I10	Circular interpolation	
N250	G01 X58	Linear move to 58ø	
N255	Z-26	Linear move to length	
N260	G00 X64	Lift from finished surface	
N265	G40	Cancel cutter radius compensation	
N270	X100 Z200	Return to turret index position	
N275	T0404 M06	Tool change. Tool No. 4 parting tool	
N280	S1430 F.18	Spindle speed and feed rate change	
N285	G00 X64 Z-25	Rapid to start	
N290	G01 X-2	Part off leaving stock faced	
N295	G00 X100 Z200	Return to turret index position	
N300	G92 X0 Z0 M30	Program end. Spindle and coolant off	

Note: To simplify the program no tool nose radius or thickness was used.

DATA FORMAT

Data are written in blocks. The data within a block were once expressed in a fixed sequence with each block containing all data (even if they have not changed from the previous block), but now almost exclusively the commands appear in random order without the repetition of unchanged data but with each word being clearly identified by its address letter. The terminology used to describe these two methods is "fixed block" and "variable block word address," respectively. The following examples illustrate these formats.

Fixed Block Example

N	G	X	Y	Z	I	J	K	F	S	M	Remarks
0250	00	05000	09000	04000	00000	00000	00000	2000	0350	03	Rapid position X, Y, Z, and start spindle.
0300	01	08500	09000	04000	00000	00000	00000	0310	0350	08	Mill in X axis—turn coolant on
0350	00	08500	09000	04500	00000	00000	00000	2000	0350	09	Retract Z axis—turn coolant off

Note: When information is placed in machine format, no spaces occur between data words.

Variable Block Example (spaced out in a form)

N	G	X	Y	Z	I	J	K	F	S	M	Remarks
N0250	G00	X05000	Y09000	Z04000	—	—	—	F2000	S0350	M03	Rapid position X, Y, Z, and start spindle
N0300	G01	X08500	—	—	—	—	—	F0310		M08	Mill in X axis—turn coolant on
N0350	G00	—	—	Z04500	—	—	—	F2000		M09	Retract Z axis—turn coolant off

Variable Block Example (without form—using decimal point format)

N0250 G00 X5 Y9 Z4 F200 S350 M03	Rapid position X, Y, Z, and start spindle.
N0300 G01 X8.5 F31 M08	Mill in X axis—turn coolant on.
N0350 G00 Z4.5 F200 M09	Retract Z axis—turn coolant off.

It is necessary for the part programmer to be aware of the data format for the system being used, and also to be familiar with the classificaton of the data that dictates the way in which it may be presented within a block. For example, a programming manual could indicate that data must conform to the following classification:

N4, G2, X3/3, Y3/3, Z3/3, F4, S4, T2, M2 (METRIC)
N4, G2, X2/4, Y2/4, Z2/4, F4, S4, T2, M2 (INCH)

This classification indicates the following:

N4	The block sequence address letter N may be followed by up to four digits.
G2	The preparatory function address letter G may be followed by up to two digits.
X3/3, Y3/3, Z3/3 (Metric) X2/4, Y2/4, Z2/4 (Inch)	The axis identification letters X, Y, and Z may be followed by up to three digits in front of the decimal point, and up to three after in metric form. Identification letter may be followed by two digits in front of decimal point, and up to four after in inch format. (Dimensional values may be subject to other limitations as explained below.)
F4	The feed address letter F may be followed by up to four digits.
S4	The spindle speed or cutting speed address letter S may be followed by up to four digits.
T2	The tool address letter T may be followed by up to two digits.
M2	The miscellaneous function address letter M may be followed by up to two digits.

(Note: These formats will change from one machine control to another. For a complete list of your machine code formats refer to your machine tool manual.)

The description above has stated that up to so many digits may be used. Some systems require that leading zeros are included and some do not. Thus a linear slide movement at a programmed feed rate may be programmed as G01 or G1, depending on the system used.

Similarly, dimensional values may also have to be programmed according to certain rules. For instance, using a data classification of 3/3 it would be possible, depending on the requirements of the system, to program a value of 32 mm in a number of ways:

(a) 032000—all digits must be included but no decimal point.
(b) 32000—leading zeros are omitted, but no decimal point is required; trailing zeros must be included.
(c) 32.000—the decimal point and all trailing zeros are required.
(d) 32.—no leading or trailing zeros are required but the decimal point must be included.

(e) 32—whole numbers may be programmed without leading or trailing zeros and without a decimal point.

SLIDE MOVEMENTS

Both word address and conversational programming require definiton of the slide movements necessary to position the cutting tool correctly in relation to the work.

This positioning is described in three ways:

(a) point-to-point;
(b) linear interpolation;
(c) contouring/circular interpolation.

Point-to-point positioning involves programming instructions that identify only the next relative tool position required. The position may be reached by movement in one or more axes at a rate of travel that is generally, though not necessarily, the maximum for the machine. If metal cutting takes place during this type of motion, it must be in one axis or the cut path will not necessarily repeat cycle to cycle.

Figure 8.18 shows details of a component. To drill the holes in this component would require two-axis point-to-point positioning. Note that the positioning prior to drilling is clear of obstruction, therefore path is not important and the actual drilling is a single-axis move, so point to point can be used. Note that point-to-point positioning is not capable of machining angles or contours, because they require controlled movement of more that one axis.

Linear interpolation control requires programmed instructions that specify both the next position and the rate of travel, or feed rate, to be employed to reach that position; the resulting cutter path is a straight line. Metal cutting would normally take place during such a move. Linear interpolation allows the machining of straight lines at a feed rate using one or two axes of motion. Other more expensive machines allow linear interpolation in three axes simultaneously. Figure 8.19 illustrates examples of one- and two-axis linear interpolation moves.

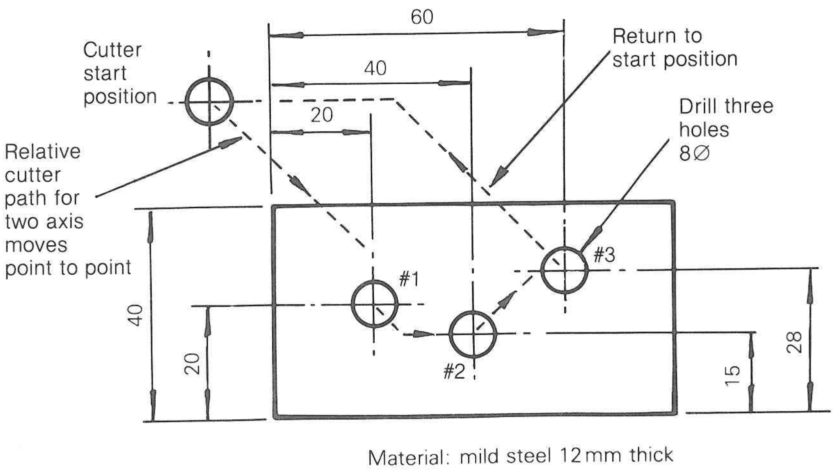


Figure 8.18 *Component detail requiring point-to-point positioning to drill holes.*

Contouring is used to describe movements involving at least two slides. The movements occur simultaneously and at a predetermined feed rate, and result in a continuous machining path which is not a straight line. An elliptical profile or a combination of arcs—the production of an arc being referred to as 'circular interpolation'—are good examples of contouring. Contouring will normally refer to irregular curves that must be machined using minute straight line segments to generate it. Contouring requires many data blocks and multiple axis movement capability. Circular interpolation, on the other hand, will produce uniform arc segments or circles with minimal programming owing to the machine control's ability to self-generate uniform arc data. The principle is illustrated in Figure 8.20.

DEFINITION OF THE AXES OF MOVEMENT

Whether conversational or word address programming is being used, the direction in which slide movement is to occur is defined by a letter, which for common machines is either X, Y, or Z for linear movement and B for rotary table movement where applicable. Axes definition codes appear together with

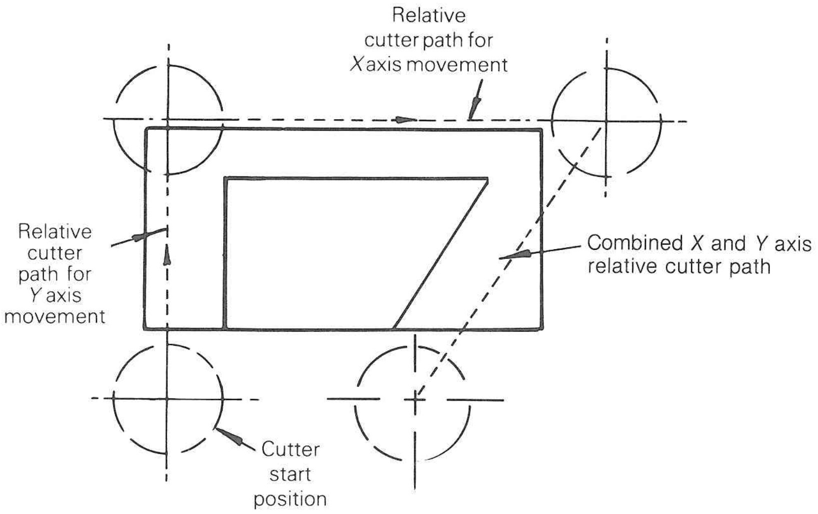


Figure 8.19 *Linear interpolation.*

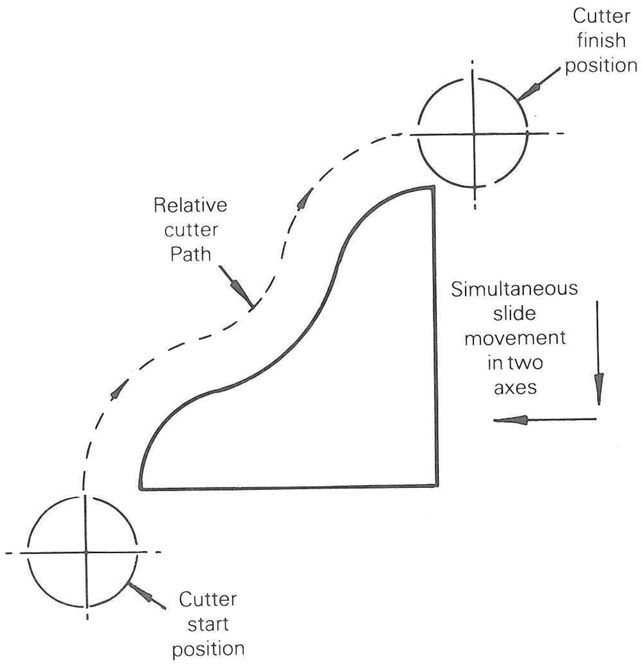


Figure 8.20 *Contouring.*

a positive (+) or negative (−) sign for determining direction. In practice the + sign is not actually entered, because if a sign is omitted, the control automatically assumes plus.

The definition of the axes of movement on common machine types, namely a turning center, a vertical machining center, and a horizontal machining center, are illustrated in Figure 8.21. Two points should be noted in relation to the illustrations. First, on a turning center having a rear-mounted tool post the plus (+) and minus (−) in the X axis would be reversed. Movement of the tool away from the spindle axis is always plus.

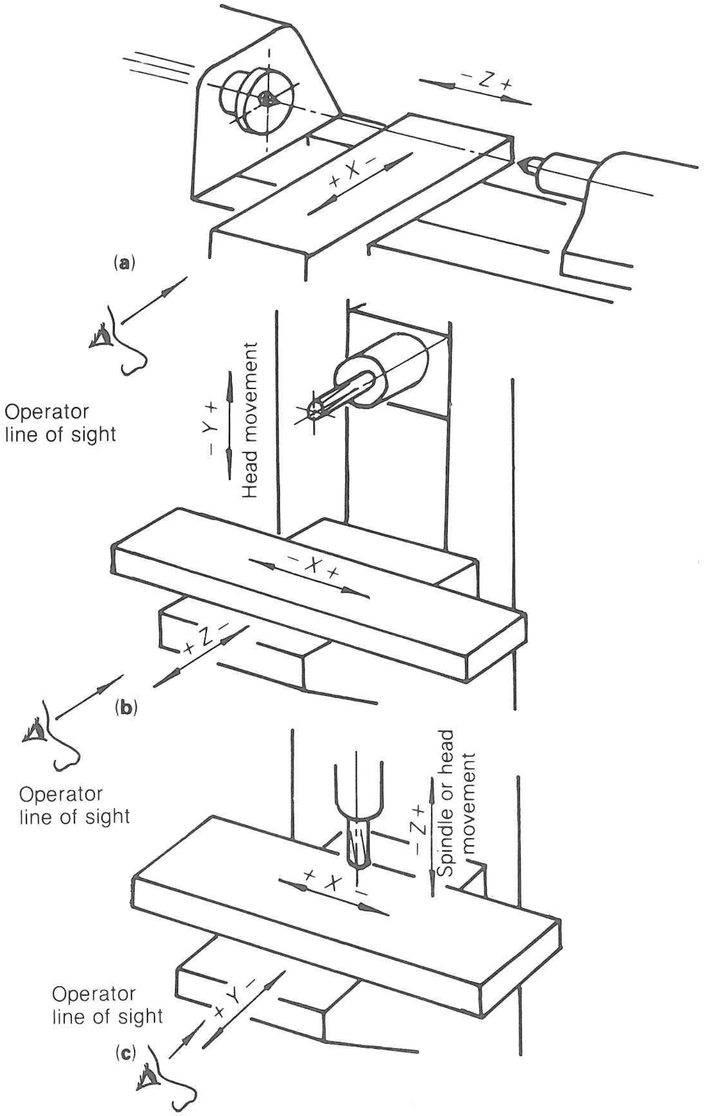


Figure 8.21 *Identification of slides and direction of the slide movement on common machine tools:* (**a**) *center lathe (turning center),* (**b**) *horizontal milling machine (horizontal machining center),* (**c**) *vertical milling machine (vertical machining center).*

Second, the axes definitions shown indicate the *machine* slide movements. In the case of a turning center these movements are identical to the tool movement in relation to the work. On milling machines, where it is the table and not the cutting tool which moves, this is not the case. For programming purposes, where it is easier to imagine that the tool is moving, it is necessary to redefine some movements. On a vertical machining center, for example, in order to achieve a tool movement in relation to the workpiece in the X positive or plus direction it is necessary to build a machine slide that movement in the X axis would cause physical table movement to the right under the machine spindle viewed from the front of the machine. In determining axes directions, the programmer can always consider that the part is viewed through the tool from the shank to the tip. Right-theoretical tool motion on the part is always positive, with left being negative. Tool motion viewed this way in the up direction on the part print is positive, with down being negative. Tool motion causing penetration of the tool into the work is negative, with retraction being positive. If rotary tables are involved, clockwise direction viewed looking into the table face from outside is positive, with counterclockwise being negative.