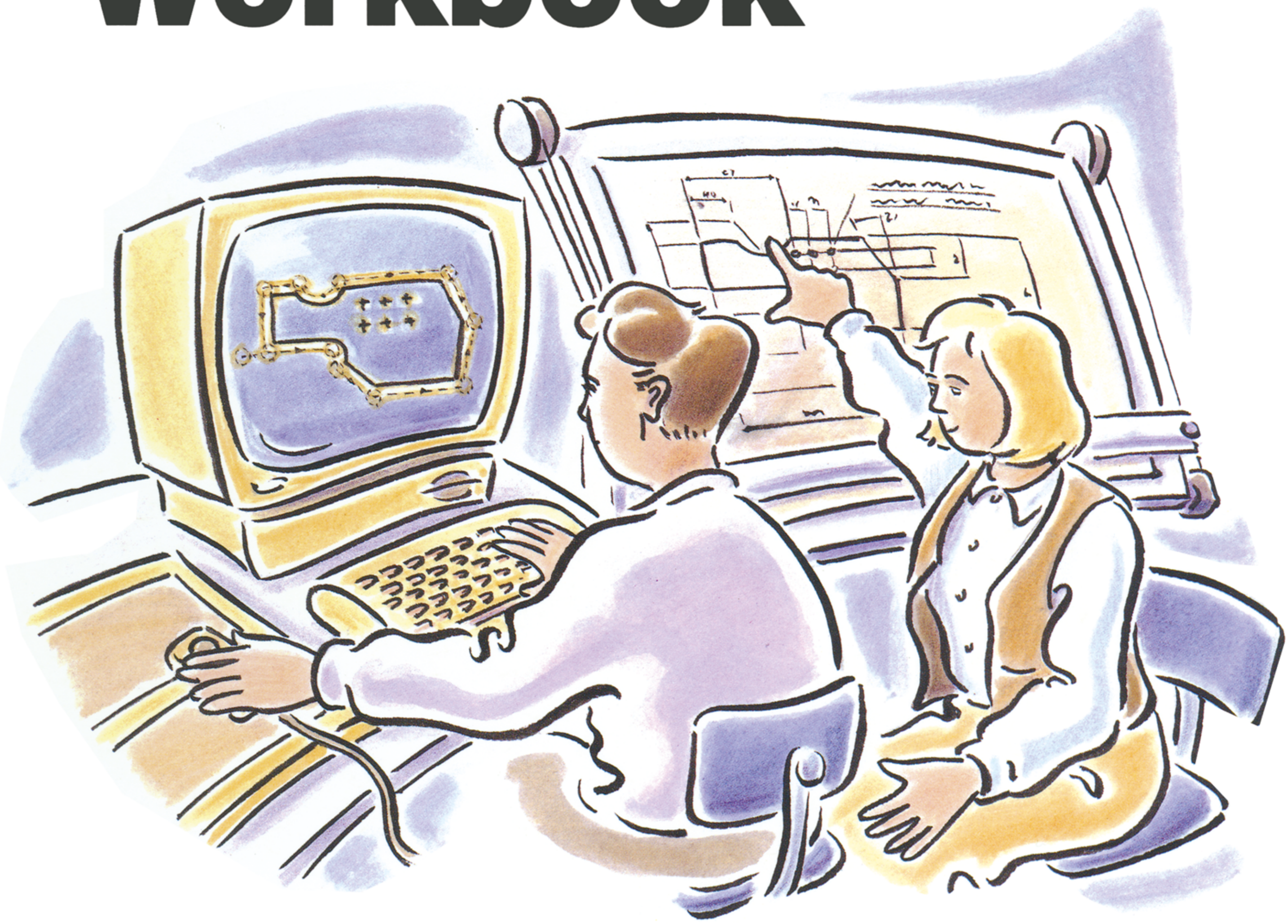
City and Guilds/Macmillan Education

CNC Part Programming Workbook



Bernard Hodges

Consultant Editor City and Guilds/Macmillan CAE series:
Peter Riley

CNC
Part Programming
Workbook

City and Guilds Co-publishing Series

City and Guilds of London Institute has a long history of providing assessments and certification to those who have undertaken education and training in a wide variety of technical subjects or occupational areas. Its business is essentially to provide an assurance that pre-determined standards have been met. That activity has grown in importance over the past few years as government and national bodies strive to create the right conditions for the steady growth of a skilled and flexible workforce.

Both teachers and learners need materials to support them as they work towards the attainment of qualifications, and City and Guilds is pleased to be working with several distinguished publishers towards meeting that need. It has been closely involved in planning, author selection and text appraisal, although the opinions expressed in the publications are those of the individual authors and are not necessarily those of the Institute.

City and Guilds is fully committed to the projects listed below and is pleased to commend them to teaching staff, students and their advisers.

Carolyn Andrew and others, *Business Administration Level I* and *Business Administration Level II*, John Murray
Chris Cook, *Assessor Workbook*, Macmillan
David Minton, *Teaching Skills in Further and Adult Education*, Macmillan
Graham Morris and Lesley Reveler, *Retail Certificate Workbook* (Levels 1 and 2), Macmillan
Peter Riley (consultant editor), *Computer-aided Engineering*, and associated Workbooks: *CNC Setting and Operation*; *CNC Part Programming*; *Computer-aided Draughting*; *Robot Technology*; *Programmable Logic Control*, Macmillan
Barbara Wilson, *Information Technology: the Basics*, Macmillan
Caroline Wilkinson, *Information Technology in the Office*, Macmillan

CNC Part Programming Workbook

Bernard Hodges

Department of Engineering
Bournemouth and Poole College of Further Education

Consultant Editor: *Peter Riley*
Formerly Head of Department of Engineering Technology
Blackpool and The Fylde College

150th YEAR
M
MACMILLAN

City and Guilds
C+G

First published 1994 by
THE MACMILLAN PRESS LTD
Houndmills, Basingstoke, Hampshire RG21 2XS
and London
Companies and representatives
throughout the world

ISBN 978-0-333-56506-3 ISBN 978-1-349-12683-5 (eBook)
DOI 10.1007/978-1-349-12683-5

A catalogue record for this book is available
from the British Library

10 9 8 7 6 5 4 3 2 1
03 02 01 00 99 98 97 96 95 94

Acknowledgement
Thanks are due to Mazak Yamazaki Machinery UK Ltd for the photographs on pages 2 and 22.

Contents

Introduction

CNC (computer numerical control) and CAD/CAM (integrated computer-aided draughting and manufacture) are essential elements in many industrial processes. The assignments in this workbook will provide a broad range of practical experiences which represent a valuable foundation to the successful application of this technology.

The subject is a diverse one, which frequently involves separate elements of a manufacturing process. Because of this, practical activities have been carefully selected to create a comprehensive yet cohesive programme of study which closely follows the CNC part programming syllabus of the City and Guilds Computer-aided Engineering 230 series.

This workbook will also prove to be a useful aid to those studying standard modules including CNC, which are available through the Business and Technology Education Council (BTEC).

The approach in all the assignments concerns the practical application of CNC. This ranges from program planning and writing to editing and proving. It is a multi-disciplined technology which demands a multi-skilled engineer for effective application. The workbook will be particularly suitable for:

- recently trained engineers wishing to advance to more specialised work using new technology
- mature, skilled and experienced engineers who need to update and enhance their traditional skills with CNC programming and CAD/CAM system applications
- service and maintenance personnel who wish to broaden their skills and knowledge base in response to multi- and inter-disciplinary developments in CNC technology
- technical trainers and teachers who are seeking to acquire new technology skills in response to changing course demands.

The result of rapid technological advances in industry is that there has never been a more exciting time to be an engineer. This workbook has been written for those concerned with sharing and exploiting the benefits that may be derived from this new technology and its associated working methods.

City and Guilds/Macmillan publishing for computer-aided engineering

This workbook is one of a series of City and Guilds/Macmillan books which together give complete and up-to-date coverage of computer-aided engineering. A core text, or source book (*Computer-aided Engineering*), gives basic information on all the main topic areas (basic CNC; CNC setting and operation; CNC part programming; CNC advanced part programming; basic CAD/CAM; computer-aided draughting; advanced CAD; basic robotics; robot technology; programmable logic controllers; more advanced programmable logic controllers). It has tasks structured in to the text to encourage active learning.

Workbooks cover five main topics: CNC setting and operation; CNC part programming; computer-aided draughting; robot technology; programmable logic controllers. Each workbook includes all the operational information and guidance needed for completion of the practical assignments and tasks.

The books complement each other but can be used independently of each other. Peter Riley (formerly Head of Department of Engineering Technology, Blackpool and The Fylde College) is Consultant Editor of the series.

General note

The purpose of this workbook is to help you to learn how to work from engineering drawings of components which are to be machined and to produce part programs which incorporate the various commands and functions of a CNC system.

Most of the exercises in the workbook are practical and you will need to have access to a CNC system. This could be available either at a college or at your workplace.

Most CNC systems operate on similar principles, but the command syntax for positional data, tool calling and machining sequences varies. There are no definitive answers to these practical exercises. The way in which a programmer approaches a job depends on the jigs, fixtures and stock tooling that are available, as well as on the type of CNC machine tool that is to machine the part and on its programming functions. Many companies have a set procedure for their part programming, which may have been formulated over many years. This is particularly the case in the use of canned cycles for operations such as drilling holes, defining the tool change positions, methods of 'blocking out' (the rough cutting of the billet prior to finish machining) and the start and termination sequences of a CNC program.

Safety note

CNC machine tools move very fast in order to achieve the high production rates for which they are designed. It is therefore very important that, before using such a machine, you are completely familiar with the program and the machine slide/tool movements for the component that is to be machined. You should be given adequate supervision at all times. If you are at all in doubt, consult someone who is qualified to give advice.

An operator using a CNC system

How to use this book

Each learning assignment in this workbook has a similar structure, to make its use as straightforward as possible. Information and guidance that is needed for the completion of the practical work is included with each assignment.

You will be able to identify the following parts of the text:

- Background information introducing the topic at the beginning of each assignment.
- Other relevant knowledge given under the heading 'Additional information'.
- In the sections 'Useful observations' you will find points which will help you in becoming familiar with the process and in exploring ways in which it can be used.
- The practical 'Tasks' are presented in a logical sequence so that they can be accomplished safely and successfully. In many cases 'Additional tasks' are included to reinforce and enhance the basic practical work.
- If there is information of particular interest concerning the practical tasks, you will find this under the heading 'A point to note' or 'Points to note'.

All the diagrams and illustrations which are needed for each assignment are given at the appropriate point in the text.

You are recommended to obtain a folder in which to keep work which you have completed. This will serve as a record of your achievements and may be useful for future reference.

Learning Assignment 1

Machine axes and datums

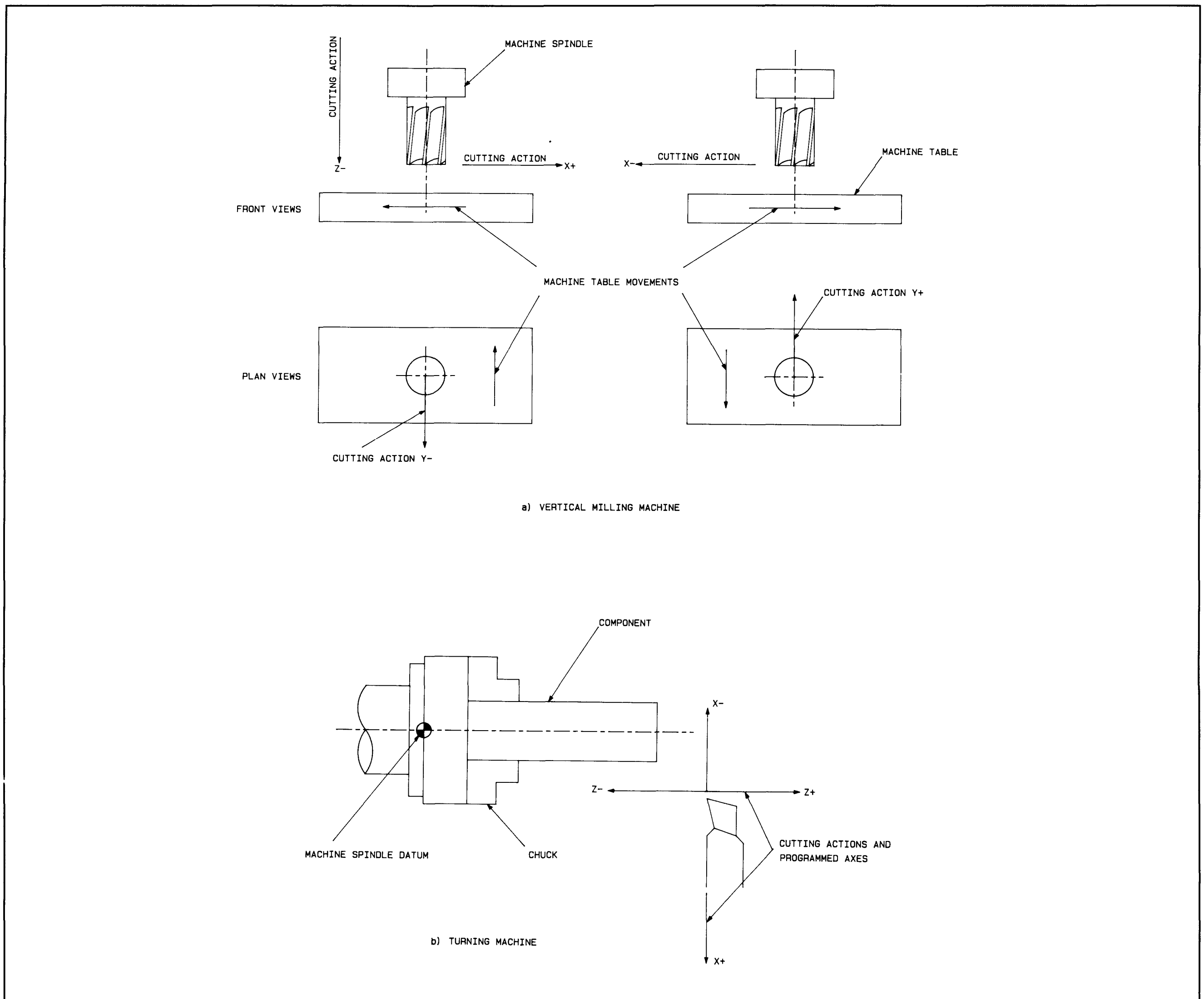
Axial definitions represent the linear movements of machine slides, which may be assigned X, Y and Z depending upon the type of machine. When using vertical milling machines, X and Y represent horizontal movements of the table, whereas in turning the tool movements are X and Z. In either case, precisely controlled linear movements are achieved.

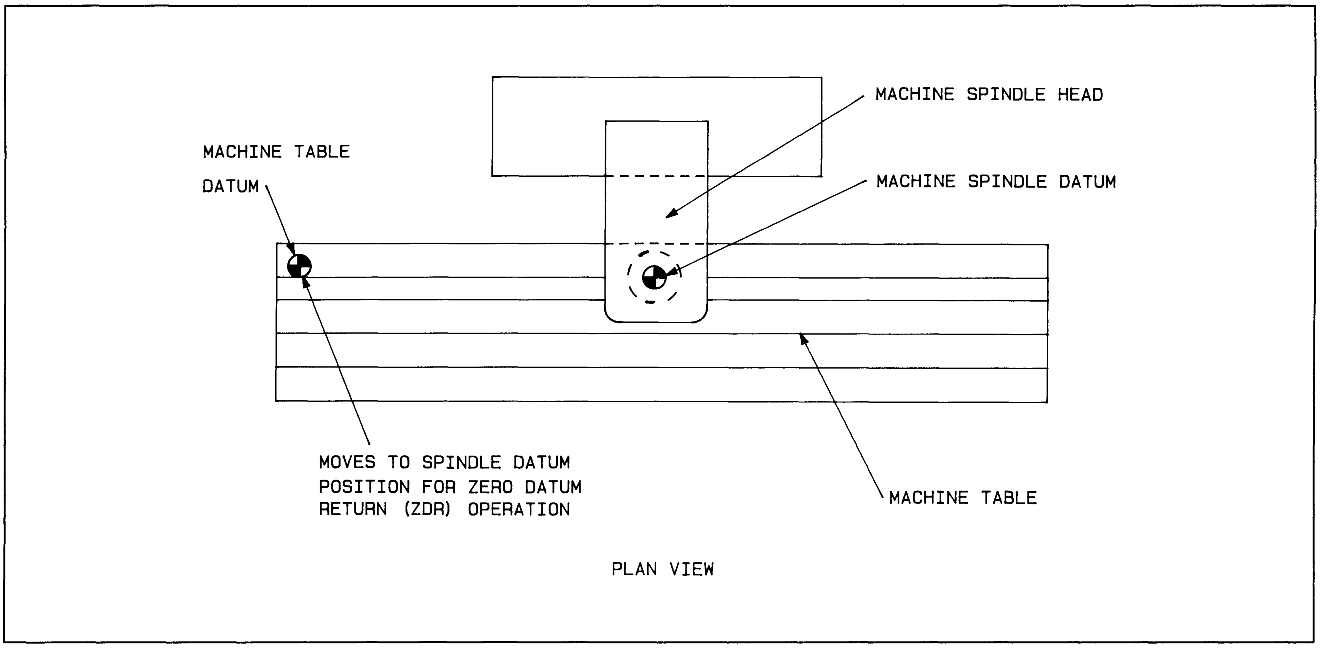
When programming, it helps if you relate these movements to the direction which the cutter travels relative to the component. Consider the vertical milling machine shown in Figure 1.1a. For the component to move from right to left (viewed from the front of the machine), the programmed axis movement is X+. To cut away from the front of the table the axis movement is Y+, and for the cutter to enter the work, the axis movement is Z–.

Only two axes are generally used in turning. The tool cross slide movement towards the centre is the X– axis. To move the tool towards the headstock the axis movement is Z–; thus movement of the tool away from the chuck is always Z+, which is a safety feature (see Figure 1.1b).

Rotational movements for ancillary equipment such as rotary tables

▼ **Figure 1.1** Definitions of machine axes



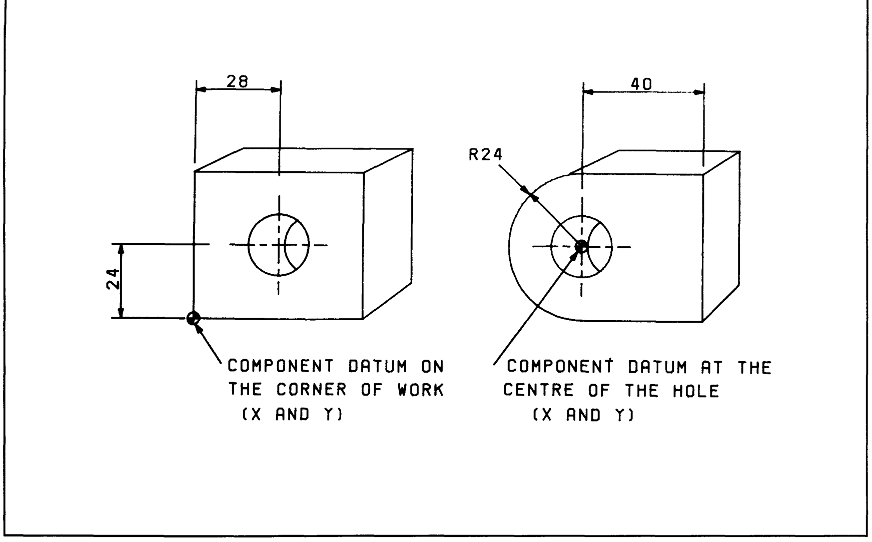


▲ **Figure 1.2** Vertical milling machine table datum

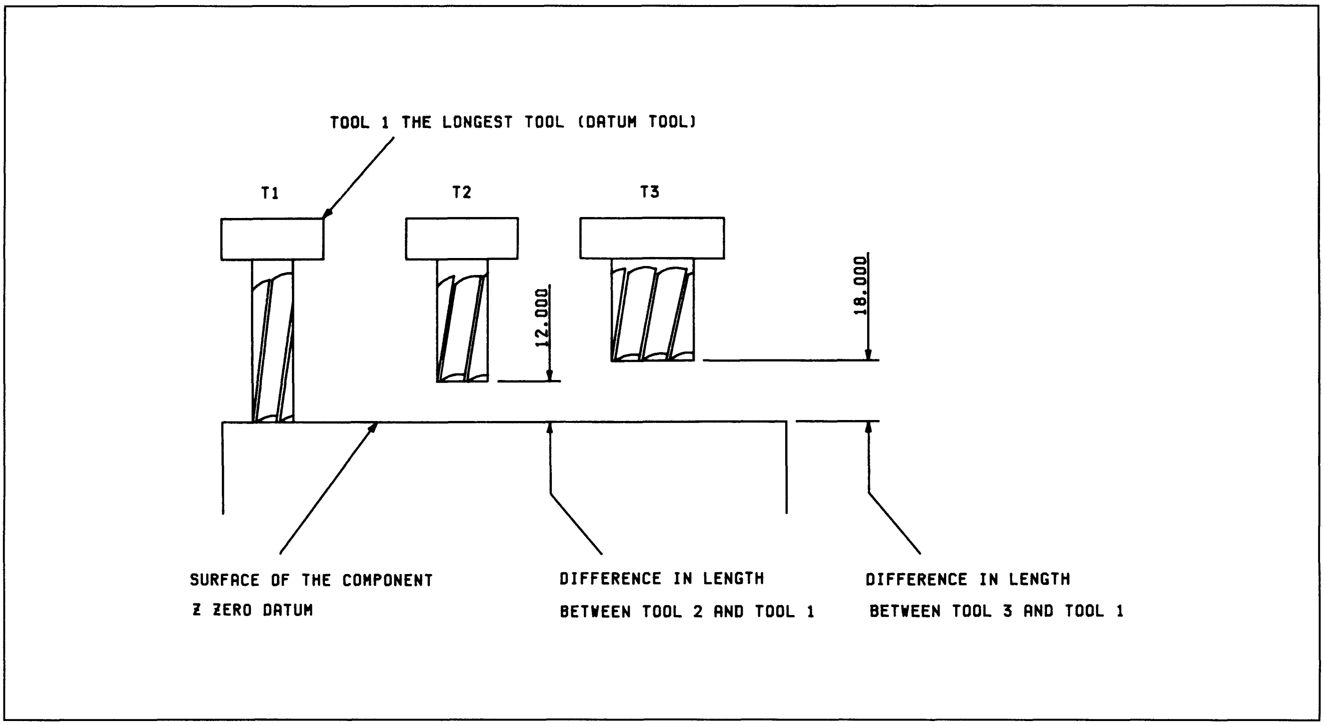
may be programmed. These axes are generally denoted by the letters A, B and C, with clockwise movement designated as positive.

The following datums are used in CNC machining:

- **Machine datums**, which are determined by the machine manufacturers. The spindle datum for a turning machine is shown in Figure 1.1b. On a vertical milling machine (Figure 1.2), the X and Y values are set at zero when the table datum is positioned directly underneath the spindle datum.
- The **program** or **component datum**, which is set by the programmer when writing the CNC program; this could be the corner of the component, the centre of a hole or the centre of the right-hand face in turning (Figure 1.3).
- The **tool datum**, which is required because all tools, whether used for milling or turning, have different projected lengths when held in the tool holder. The **datum tool** is frequently the longest tool to be used in a machining cycle and all other tool projections used are measured and compared to this (Figure 1.4).



◀ **Figure 1.3** Component datums



▲ **Figure 1.4** Setting tool length offsets

USEFUL OBSERVATIONS

Manually drive the table or turret of a CNC machine, and observe the digital readout of the axis positions and the direction in which the axes are moving.

Select the offset facilities for the machine and note the display and the positive or negative direction signs for each tool offset axis.

Additional information

The usual way of programming is to designate movement relative to the program/component datum. This is known as **absolute movement,** since all movements are taken from the same fixed point. Alternatively, **incremental movement** is programmed from the present position to the next – that is, the previous position becomes zero.

Zero datum shifts

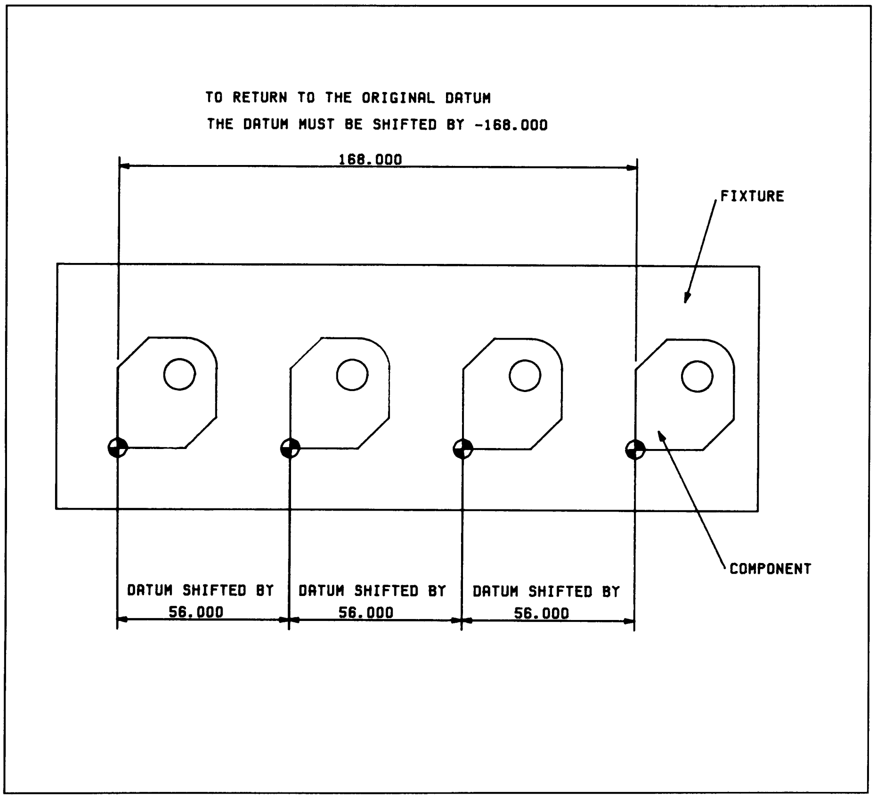
In both milling and turning, the program datum point often needs to be moved during the machining cycle. This could be when several identical components are positioned on the machine bed at equal distances apart (Figure 1.5). When the first part has been machined, the program is repeated by a datum shift command to the second component, and so on (by programming the appropriate G code, discussed in Learning Assignment 3, and the offset distance). Zero datum shift is also used in turning where part of the component is held inside the chuck, to reduce overhang. Once the first half has been machined an optional stop is programmed so that the operator can re-position the component to suit the zero shift offset. For instance, if a bar stock is 352 mm long, a 150 mm length could be machined with the program datum at the right-hand face. The component is then re-positioned so that a 50 mm length is held in the chuck, and the datum point is shifted by 150 mm to the left of the original datum point, so producing a turned part 300 mm long (Figure 1.6).

When turning several grooves, or machining a series of slots, a zero datum shift can be used: the groove or slot is programmed and the programmed moves are then shifted along or across the work by the relevant offset value. This has the obvious advantage that the length of the program may be greatly reduced.

TASKS

Task 1.1 Datum positions

Determine the X and Y (for a milling machine), or X and Z (for a turning machine), co-ordinate distances from the machine datum to the component datum.



◀ **Figure 1.5** Zero datum shifts

TASKS

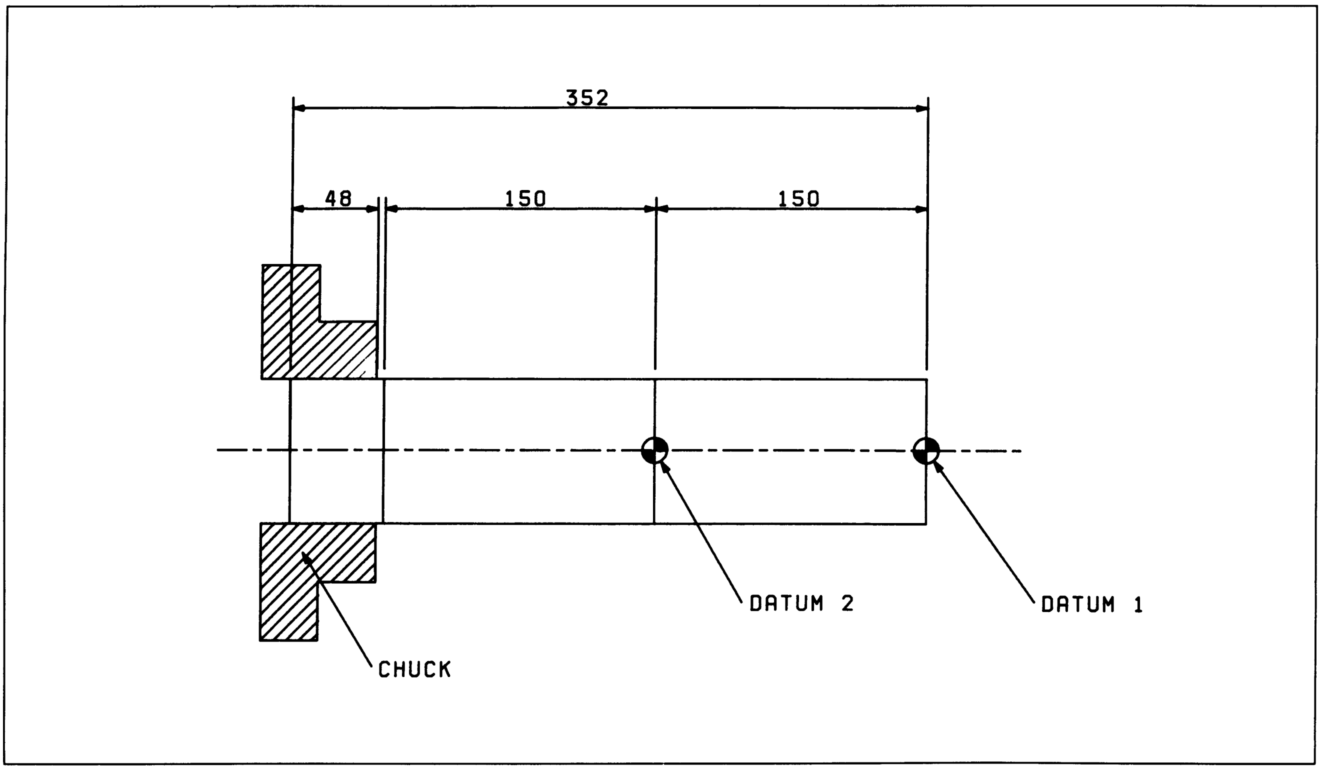
Task 1.2 Z datum position (milling)

On a milling machine, determine the distance from a component Z datum to the spindle datum (that is, the home position of the quill).

Task 1.3 Tool length offsets

Find the tool length offset data for a multi-tooled milling machine. Make a note of the longest tool (the datum tool), and record the difference in length of the other tools with respect to the datum tool.

▼ **Figure 1.6** Shifting the zero datum on a turned component

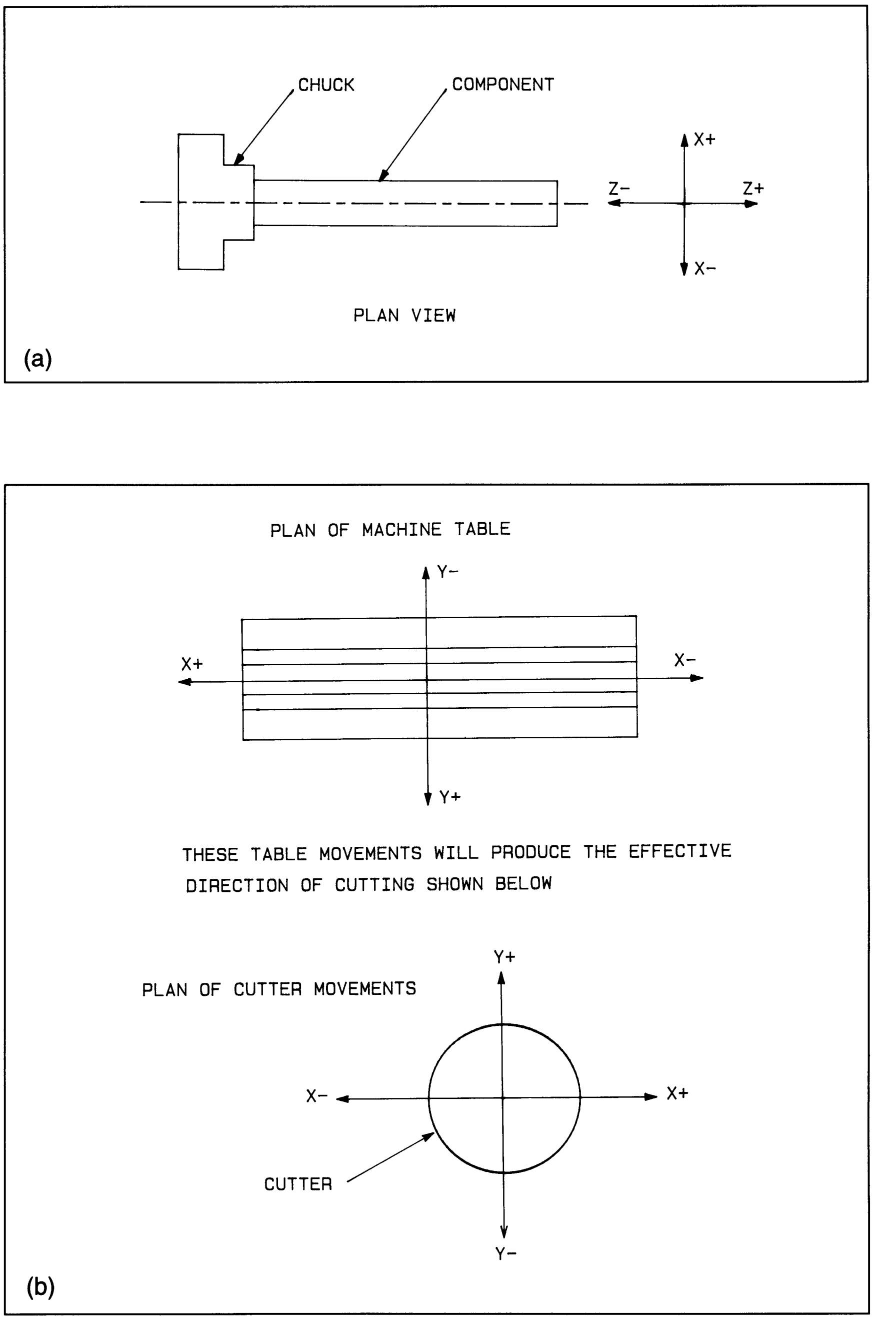


POINTS TO NOTE

When using a CNC turning machine, the directional signs of axial movements in the X and Z will be the same as those for the dimensions indicated on the workshop drawing. On a milling machine, the movement of the component rather than the tool in X and Y will be the reverse (Figure 1.7).

ADDITIONAL TASK

Locate a component on the table of a milling machine and set the longest tool at a precise distance from the top surface of the component in the Z axis by using a setting block of a known height to set a safe clearance plane. Note the reading on the digital output for the Z axis.



▶ **Figure 1.7** (a) Tool movements for a turning machine; (b) the table movements on a vertical milling machine

Learning Assignment 2

Setting up a component datum and the tooling

One of the tasks of the setter/operator is to position the component datum and the tools that are going to be used for machining a program. The programmer cannot always specify certain information such as the exact distance of the component datum from the machine datum, or the precise diameters or lengths of the tools to be used.

When the power on the CNC machine tool is first turned on, perhaps at the beginning of a shift, the tool has to be driven to the extremity of its programmable movement. The control system then sets the positional reading from the encoder on each axis (X, Y and Z) to the machine reference datum; thus the machine has 'datumed itself' and compensated for any positional variation due, for example, to thermal influences.

Setting up a component datum

Load a tool into a quill or turret and use the jog keys on the control to move near the component datum. This is usually some feature of the workpiece such as a corner or a hole or, in turning, the end of the bar (Figure 1.3, page 5). To get precise movement on to the component datum, use the handwheel facility. The distance of the actual position in X, Y (for milling) or X, Z (for turning), will be displayed as machine co-ordinates. Depending upon the type of control that the CNC machine has, these positional values may be entered as they are displayed (for example, X –355.002, Y –467.987) into the relative register as the component datum; alternatively, in the case of an absolute register, they are set to zero.

Moving the work datum point

In the example just described, the register containing the component datum X and Y co-ordinates could be selected by the G code G54. Several work co-ordinate systems are available on most CNC machines, and these can be selected by specifying one of the following G codes:

G54	Work co-ordinate system 1
G55	Work co-ordinate system 2
G56	Work co-ordinate system 3
G57	Work co-ordinate system 4
G58	Work co-ordinate system 5
G59	Work co-ordinate system 6

The co-ordinate values (X, Y and Z for milling, say) are entered into the appropriate register. For example, you would enter the X, Y and Z values into the register if you were using G54, into register 2 for G55, and so on. The work datum is moved if several identical workpieces are to be machined. They may be placed at random on a grid plate or pallet. When the first component has been machined, the work datum is moved by using one of the G50 codes, and the same program is then used to machine the next component. Moving the datum may be applicable where a machining sequence is to be repeated on a component – for example, repeating the drilling of holes on a pitch circle diameter (PCD) from one position to another.

Entering the tool length offsets

The setting of the tool length offsets for milling is begun by inserting the longest tool into the holder and touching it on to the datum surface of the component. The Z position value displayed is the distance from the end of the tool to the quill home position. For the datum tool this value is altered to zero and transferred into the register under the tool number. Thus if tool number 3 is the longest (datum) tool, H3 = 0.000.

This procedure is repeated for the other tools used in the machining of the component. A typical tool length offset register might read as follows:

H1 = –4.553
H2 = –7.354
H3 = –9.992
H4 = 0.000
H5 = –6.543

Clearly, in this case H4 is the datum tool.

USEFUL OBSERVATION

On a CNC machine, select the positional display which shows the position of the machine table (the X, Y, Z co-ordinates) as a relative value, as an absolute value, and as an actual distance from the machine table datum. Use the handwheel to move in increments of 10 mm in each direction and observe how these values change. You will notice that each positional display will change by 10 mm, even though their co-ordinates are different.

Additional information

Sometimes it may be difficult to determine which tool is the longest. If a tool is longer than the one chosen as the datum tool, its length will be displayed as a positive number (+0.3557, for instance). In some machines the control will move the quill up by 0.3557 when this tool is used; on other machines this tool would have to be set as the datum tool and all the other tools set accordingly.

Entering the tool radius

The radius or diameter of each tool has to be entered into the machine control, particularly if tool radius compensation has to be used (see Learning Assignment 6). In milling, tool diameters are accurately measured and the radius for each tool is entered into the register: for example, H17 = 4.995, H18 = 5.995. (These could be the radii for tools 1 and 2, but their values are stored in the registers H17 and H18.)

TASKS

Task 2.1 Setting tool length offsets on a milling machine

Hold a square piece of metal in the vice of a CNC milling machine. Select a drill and two slotdrills of any size and length. Load the longest tool into the quill, touch on to the surface of the metal block and set the tool length to zero (this is the datum tool). Transfer the value of 0.000 into the tool offset library. Repeat this procedure for the other two tools, ensuring that the values transferred to the offset library have negative signs.

Task 2.2 Setting tool length offsets on a turning machine

Hold a piece of round bar in the chuck of a CNC lathe and define the component datum position as being X0.000 on the centre line and Z0.000 at the end of the bar. Select three turning tools (for example, a roughing tool, a finishing tool and a parting-off tool) and position each in turn at the component datum. Enter their tool offset lengths into the offset library.

ADDITIONAL TASK

Take the block of metal that you used in Task 2.1. Program each corner as a different datum or work co-ordinate system. Enter the X, Y and Z co-ordinates for each corner in the G54, G55, G56 and G57 registers. Enter a program to machine one linear move (from X0.000 to X50.000, say) from each of the work datum points, with the Z value just above the work. The program could be similar to Program 2A, shown here.

```
N10 G54
N11 G0 Z5.000
N12 G0 X0.000 Y0.000
N13 G01 X50.000 Y0.000 F200
N14 X0.000 Y0.000
N15 G55
N16 X0.000 Y0.000
N17 X50.000 Y0.000
N18 X0.000 Y0.000
N19 G56 etc.
```

◀ **Program 2A**

Learning Assignment 3

Line format and program structure

A CNC program is a set of instructions that is read by the machine tool computer system to control machine slide movements, tool movements and additional miscellaneous functions. There are two types of CNC programming: **conversational format** and **word address format**. Most CNC machines use word address format, mainly for historical reasons – many of the first machines used this type of format.

Irrespective of the type of control system that is being used, all programs include the following types of information:

- **positional data**, with the directions of axial movements in X, Y and Z for milling machines, and X and Z for turning machines
- **command data** (in word address format, discussed below, these are expressed in terms of G codes), that is, instructions on how movements are to take place: for example, rapid traverse, linear or circular moves in absolute or incremental mode, data for executing canned cycles, and so on; Program 3A (which is in word address format, like the other programs in this Assignment, except where conversational format is specified) deals with positional and command data only
- **operational data** such as spindle speeds, feedrates and tooling commands; Program 3B includes positional, command and operational data
- **miscellaneous functions** (M codes), which do not affect the machine tool movement but execute important functions such as spindle stop, tool change, program stop or coolant on:off; Program 3C includes M codes as well as other types of data.

All these programs may be written manually and entered into the machine control unit manually (manual data input, MDI), that is, by typing in the program using the keys on the CNC controller or, alternatively, by using a CAD/CAM system.

In all programs the data is displayed in a line or block format, each program line or block being numbered and ending with an end-of-block character, often a semi-colon (;). You have to key in the end-of-block character when you are using manual data input, but this is not necessary

▶ **Program 3A**

```
N10 G90                  (N10 block number ten, absolute
                         programming)
N11 G01 X100.000 Y0.000  (move to a position of X 100 mm and
                         Y 0 mm)
N12 X200.000 Y100.000    (move at an angle to X 200 mm Y
                         100 mm)
N13 X300.000 Y100.000    (move in a straight line to
                         X 300.00 mm)
```

▶ **Program 3B**

```
N10 T5                         (tool number 5)
N11 G43 Z10.000 S2000          (spindle speed S of 2000
                               rev/min)
N12 G01 Z-20.000 F100          (feedrate F of 100 mm/sec)
N13 G01 X50.000 Y50.000 F250   (feedrate F of 250 mm/sec)
```

▶ **Program 3C**

```
N10 T1 M6              (quill home, spindle stop)
N11 G0 G90 S2000 M3    (spindle on clockwise)
N12 G43 Z10.000 M8     (coolant on)
```

```
N06 M9;                 (block number 6, coolant off)
N07 T1 M6;              (tool number one, spindle stop and
                        quill home)
N08 G0 G90 M3 S2000;    (absolute programming, spindle on
                        clockwise, spindle speed 2000 rpm)
N09 G43 Z10 H1 M8;      (tool length compensation applied
                        to the value set in H1 register,
                        rapid to 10 mm above component and
                        coolant on)
```

◀ **Program 3D**

when you are writing a program on a computer as it is automatically put on to the end of each line when it is transferred to the CNC control. Program 3D shows operational data, miscellaneous functions and end-of-block characters.

Conversational programming

This is usually carried out by the programmer on the machine controller with instructions and options displayed on a VDU screen. Various prompts are offered by the controller, and the programmer responds with appropriate inputs. A program block is thus generated and the programmer is then led on successively to each new line of the program. The prompts usually ask for information concerning dimensions, tool feedrates and speeds, the material to be cut, tool type and the designated surface finish. If erroneous information is entered (such as the wrong X and Y, or the wrong X and Z end position of an arc), the mistake may be displayed on the VDU screen. Sometimes, however, the error becomes apparent when you execute a program 'run through' (a facility on some machine controllers which enables the program to be proved without any movement of the machine table or tools).

Program 3E is a typical conversational program for a Heidenhain control system. Each line of program commences with a line number, and this is followed by commands which are entered by a conversational/menu system. Linear moves are designated by the letter L and circular moves using the letters CC and C for the circle centre and the circle end-point co-ordinates respectively, making the program easy to read. R0 means that no cutter compensation is to be applied, and RL indicates that cutter compensation is to be applied to the left of the workpiece profile.

Like all CNC programs, Program 3E has a clearly identifiable structure. Look at it carefully. You can see that the first three lines contain information regarding the tools required – the 'tool definitions', which state the difference in length of each tool and the radius.

```
1   TOOL DEF 1        L-10,050                      R+5,000
2   TOOL DEF 2        L0,000                        R+0,000
3   TOOL DEF 3        L13,250                       R+3,000
4   LBL1
5   TOOL CALL 0   Z                S1500
6   L                 Z+25,000                      R0  F9999  M05
7   L                 X-25,000  Y-40,000            R0  F9999  M
8   LBL 0
9   STOP                                                       M25
10  TOOL CALL 1   Z                S1500                       M
11  L                 X-15,000  Y-15,000            R0  F9999  M
12  L                 Z+1,000                       R0  F9999  M03
13  L                 Z-10,000                      R0  F100   M
14  L                 X+0,000   Y-15,000            RL  F200   M
15  L                 X+0,000   Y+20,000            RL  F200   M25
16  CC                X+20,000  Y+20,000
17  C                 X+40,000  Y+40,000  DR +      RL  F200   M
18  CALL LBL 1 REP
19  STOP                                                       M25
20  STOP                                                       M02
```

◀ **Program 3E**

Lines 6 and 7 give the co-ordinates in X, Y and Z of the tool change position. This will be a safe position where there is no danger of collisions.

Line 10 is the instruction for calling up the required tool (tool number 1 in this example).

Lines 11 to 17 give co-ordinate information for various linear and circular moves.

The end-of-program sequence is given in the last two lines of the program: the tool is brought to rest, and the code M02 signifies the end of the program.

Word address programming

This type of programming uses letter codes for the machine control to identify and execute the various types of instruction, such as positional data or miscellaneous functions. Up to 99 codes can be used for each letter and, although this is the most commonly used programming format, there is no international standard for their designation. This does have the advantage that manufacturers of machine controls can introduce codes for executing various functions particular to their own system. Most system manufacturers do, however, adopt the following G code identities:

G00	Rapid movement from one point to another
G01	Movement from one point to another with a feedrate applied
G02	Circular interpolation, clockwise
G03	Circular interpolation, counterclockwise
G04	Dwell, designated in seconds
G28	Machine table datum
G40	Cancel cutter compensation
G41	Cutter compensation to the left
G42	Cutter compensation to the right
G43	Apply tool length compensation
G49	Cancel tool length compensation
G80	Cancel a canned cycle
G81 to G89	Used for canned cycles (drilling, boring and tapping)
G90	Absolute dimension programming
G91	Incremental dimension programming.

These G codes may be either **modal** or **non-modal**. A modal G code stays in operation until it is cancelled; for example, when a G01 is programmed, all movement will be linear with a designated feedrate until the code is cancelled with a G0 or G02/G03. A non-modal G code is executed only within the block in which it is programmed and is not applicable to the following blocks. A dwell command is an example of a non-modal code.

Some typical M codes or miscellaneous functions (which, as previously stated, refer to non-movement data) are given below:

M00	Program stop
M01	Optional stop
M02	End of program
M03	Spindle on clockwise
M04	Spindle on counterclockwise
M05	Spindle off
M06	Tool change
M08	Coolant on
M09	Coolant off
M13	Spindle on clockwise and coolant on
M30	End of tape.

Program 3F is for a milling machine using a Fanuc control and is in word address format. Look at it carefully. The first character in the program is the percentage sign, %; this character indicates the start of a program and must be there for the control to accept the data. The second line is the program number, 0123.

The first word of a block, N0 on the third line, is the block number and this serves to identify the block. Each block must have a distinctive block number. The second word in the block is the 'preparatory' function. This word provides information for the control system regarding tool or component movements, the interpretation of co-ordinates used and so on.

In Program 3F block N0 is a safety routine: previously programmed cutter compensation and canned cycles (see pages 24 and 34) are cancelled and the machine moves to the tool change position.

Blocks N3 to N6 relate to functions executed at the tool change position: change to tool number 1, coolant on, spindle on clockwise.

The following nine blocks contain movement information/toolpaths for machining a particular profile, and blocks N16 and N17 are end-of-program commands.

The data in each block can be represented either in a **fixed block**, which contains all data even if some of it is the same as in the previous block, or in a **variable block** (which is the more common), in which the data is not repeated from the previous block and can be entered in a random order. It is important that the programmer is familiar with the data format of the control and the way in which it is displayed in each block. This varies from one control system to another and will be shown in the manufacturer's manual. A typical block might look like this:

N05 G02 X+043 Y+043 Z+043 F03 S04 M02

although in many systems the leading zeros can be omitted. In this example,

N05 – for the block number 5:
maximum number of digits available for block numbers = 4

G02 – for the preparatory function 2:
maximum number of digits available for G codes = 2

X+043 – dimension along the X axis (note that a word can have a sign):
maximum number of digits available for dimensions = 4 in front of the decimal point, 3 behind it.

The same format is used for movement in the Y and Z axes:

F03 – for the feedrate 3:
maximum number of digits available for feedrate = 3

S04 – for the spindle speed 4:
maximum number of digits available for spindle speed = 4

M02 – for miscellaneous function 2:
maximum number of digits available for M codes = 2

▼ **Program 3F**

```
%
O123
N0      G28 G90 G40 G80.
N1      G21
N2      G0 X-40.000 Y-40.000
N3      T1 M6
N4      G0 G90 M3 S2500
N5      G43 Z50 H1 M8
N6      (TOOL DIA 12.000
        SLOTDRILL)
N7      G0 X-10.000 Y-20.000
N8      G01 Z-8.000 F60
N9      G41
N10     G01 X0.000 Y0.000
        F250
N11     X0.000 Y50.000
N12     G02 X20.000 Y70.000
        R20.000
N13     G01 X90.000 Y70.000
N14     G40
N15     X-40.000 Y-40.000
N16     G28 Z0.000 M5
N17     M30
%
```

Absolute and incremental programming

In **absolute dimension programming**, the end-point of the tool movement is determined by its specified co-ordinates in the chosen axial plane (X and Y, or X and Z). These can be positive or negative, depending on the direction of movement from the program datum point. Programming in absolute dimensions is established by using the preparatory function G90, and acts modally. For example, if a linear tool movement is required starting from the program datum (X0.000 Y0.000) and moving to point P1 with the co-ordinates X = 10, Y = 10 and to a point P2 with the co-ordinates X = 20, Y = 10, the program would look like Program 3G.

▼ **Program 3G**

```
N10 G90
N11 G01 X10.000 Y10.000 F100
N12 X20.000
```

In **incremental programming**, the end-position of the tool movement is determined by the distances (measured along the axes) of the present tool position relative to the required next position. The co-ordinates can be positive or negative, depending on the direction of movement from the present tool position. Incremental programming is established by the preparatory function G91, which like G90 acts modally. For the same movement as in the absolute programming example, and again starting from the program datum, the program would look like Program 3H.

▼ **Program 3H**

```
N10 G91
N11 G01 X10.000 Y10.000 F100
N12 X10.000 Y0.000
```

USEFUL OBSERVATIONS

When programming in word address format it is not necessary to repeat 'modal' preparatory functions on each block of data. For instance, once a G01 has been programmed for controlled linear movement, the G01 can be omitted from all subsequent blocks where linear movement is required. But once G01 has been cancelled (with a G02, for example), the G01 needs to be re-programmed when linear movement needs to be restored.

When X, Y or Z co-ordinates are the same in the preceding block they need not be re-programmed. You can see an example of this is in Program 3G, where the Y value of 10.000 is the same in block 12 as in block 11.

TASKS

Task 3.1 Data format

Find out the program format of the CNC machine that you are using, by consulting the manufacturer's manual. Study the preparatory codes and miscellaneous functions that are used. Also make a note of the data format and how it is displayed. Make sure that you are thoroughly familiar with the format.

Task 3.2 Machining a slot

A slot is to be machined on a vertical CNC milling machine. The slot is to be 100 mm long, 12 mm wide and 5 mm deep. The start position of the slot is X50 and Y50 from the component datum and it is to be machined using a 12 mm diameter slotdrill.

Construct a program to machine the slot and identify the program structure and the types of data used – for example, tool change position, end of program, linear move with feedrate.

Task 3.3 Task 3.2 in incremental mode

Repeat Task 3.2 for machining the slot using incremental moves.

ADDITIONAL TASK

Three horizontal slots are to be machined in a square piece of metal. Choose values for the length, width and depth of the slots. The slots are to be directly below each other and the component datum is at the top left corner. Write the program to include incremental and absolute moves such that all X and Z moves are absolute and all Y moves are incremental.

Learning Assignment 4

Input and edit function

When you have entered a program into the control and executed a proving run, you may find you need to edit the program. You may need to change the feedrates or the tool change positions, to correct errors in the positional data or to alter the values for the tool diameters and lengths. Within the program, you may need to add or delete a block or blocks, or to insert other programs into the main program in the form of sub-programs or macro routines.

Additional information

Many different CNC machine controls are used in industry, and the editing facility and functions vary from one machine to another. All CNC control systems normally allow the programmer or operator to execute the editing functions described above, but some have additional features too. These may enable you to search for a particular command, feedrate or block number. They may also include a 'find and replace' facility for the global editing of spindle speeds or feedrates, and refinements for moving blocks of data from one position in the program to another.

USEFUL OBSERVATION

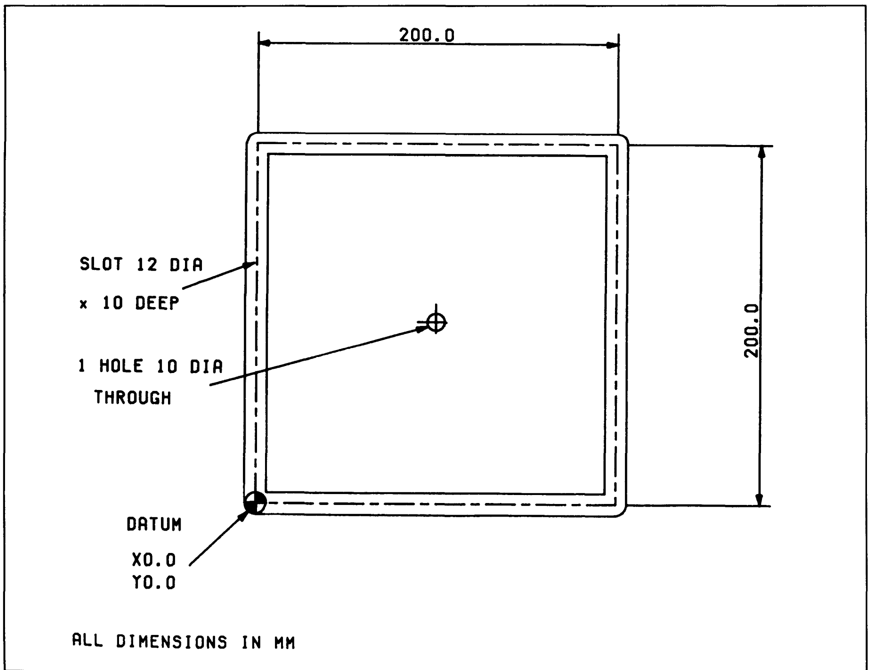
Examine the editing facility on a CNC controller and observe how the program is displayed. Make a note of how many lines of data are shown on the screen for editing. Notice how the positional data is given: for example, either leading or trailing zeros may be used (thus a length of 20 mm can be expressed as 020000, 020., 20.00 or 20.), and notice too the use of decimal points. Look at how the blocks are numbered (for instance, N001, N002, N5, N10), and what end-of-block character is used.

TASKS

Task 4.1 Inputting a program

Program 4A (overleaf) was written for a vertical milling machine with a Fanuc control system. If you are using a different system, carefully amend the program accordingly. Refer to the manual or seek qualified advice if you are not sure how to do this.

Input the program using manual data input. The program will machine a square slot with a centre line 200 mm × 200 mm, and drill a hole at the centre of the square (Figure 4.1). The program datum is at the bottom left-hand corner of the square slot. Choose a suitable slotdrill and twist drill, and save the program when you have completed the input.



◀ **Figure 4.1** Machining a square slot

```
%
O123
N01 G91 G40 G80 G0 Z0.000
N02 G21
N03 G90
N04 M9
N05 G0 X-50.000 Y-50.000
N06 G28 G49 Z50.000 M5
N07 T1 M6
N08 G0 G90 M3 S2000
N09 G43 Z25.000 H1 M8
N10 (TOOL DIA12.000
    SLOTDRILL)
N11 G0 X0.000 Y0.000
N12 G01 Z-10.000 F30
N13 X0.000 Y200.000 F150
N14 X200.000 Y200.000
N15 X200.000 Y0.000
N16 X0.000 Y0.000
N17 G0 Z20.000
N18 X-50.000 Y-50.000
N19 M9
N20 G28 G49 Z50.000 M5
N21 T2 M6
N22 G0 G90 M3 S2000
N23 G43 Z50.000 H2 M8
N24 (TOOL DIA10.000 drill)
N25 G0 X100.000 Y100.000
N26 G83 G98 Z-12.000 R2.000
Q4.000 F50
N27 G80
N28 G0 X-50.000 Y-50.000
N29 M9
N30 G28 G91 Z0.000 M5
N31 M30
N32 %
```

▲ **Program 4A**

POINTS TO NOTE

Programs can be proved out after editing by using a backplot facility in CAM software or, if this is not available, by running the machine tool in single block mode (in this mode, movement of the machine bed stops at the end of each block of the program). When doing this you may wish to reduce the feedrates by using the feed-override control (%). This is not advisable if you are machining metal, however, as various tooling problems can arise if you do not adhere to the recommended feedrates.

TASKS

In a Fanuc programming system for a milling machine the difference between the length of each tool compared to the datum tool is stored, and these differences are displayed as H values. For example, if tool number 1 is the datum tool (that is, the tool length offset is equal to zero), then H1 = 0.000. If tool number 2 is 3 mm shorter than the datum tool, H2 = –3.000. When tool number 2 is used and tool length compensation is applied, the quill will move down 3 mm to compensate for the difference in length.

In this task a canned cycle (see page 34) is used to drill a hole – a G83 code. In this cycle R refers to the retract height of the drill and Q is the specified distance for each drill peck taken from the retract height.

Edit the program to change the feedrates and to change the positional data for the hole from X100.0 Y100.0 to X130.0 and Y120.0. Save the edited program.

Re-load the original program. Now move the drilling operation to the beginning of the program, that is, insert block numbers N24 to N28 after block number N9.

ADDITIONAL TASK

Change the drill size and alter the size of the square profile to 250 mm × 250 mm. Prove out the new program, save it and obtain a hard copy print-out. A machine proving-out material may be used, such as high-density polyurethane foam, machining wax, or soft wood, which will not damage the tools if there is a mistake in the program.

Manually writing a CNC program using the text editor of a CAM system

As explained previously, a CNC program can be written manually and loaded into the machine control by manual data input (MDI). Writing a program directly into a CAM system has several advantages.

1 The text editor may have more editing facilities than the machine control, such as merging CNC programs, moving blocks of data from one address in a program to another, global changes to spindle speeds and feedrates, search facilities for strings of data and so forth.
2 From the editor it may be possible to produce a 'backplot' of the cutter paths which can be visualised on a screen or using a plotter, before the program is loaded into the CNC machine control. This is a useful way of checking a manually written program for errors.
3 A CNC program may be downloaded direct to the machine control, so eliminating the use of paper tape. A hard copy print-out can be obtained, and this can be used for operator documentation or kept for future reference and stored as a permanent record.

Task 4.2 Identifying features in a text editor of a CAM system

Examine the manual data input facilities on a CAM system and identify the purpose of the following functions:

- save file
- load file
- delete file
- merge file
- copy block
- move block.

Task 4.3 Inputting a program for a square profile

Program 4B is a CNC program for machining a square profile and machining one hole, using a Fanuc, or alternative, CNC control. Copy Program 4B into the text editor of your CAM system and, if possible, run the tool simulation by using the 'backplot' facility.

ADDITIONAL TASK

This task will help you to become familiar with features of the text editor. Move the programming blocks for machining the hole (blocks N25 to N27) to the beginning of the program, so that the profile machining is the second operation. Change the horizontal feedrate for the profile to 250 mm/min and all vertical feedrates to 75 mm/min. Save the program, giving it a name of your choice, and get a print-out. Also save that part of the program that concerns the machining of the hole as a separate program. Recall this program and produce a 'hard copy' for reference.

```
%
O1230
N01 G28 G91 G40 G80 G0 Z0.
N02 G28 X0. Y0.
N03 G21
N04 G92 X0. Y0. Z0.
N05 G90
N06 G0 X-50. Y-50.
N07 T1 M6
N08 G0 G90 M3 S2000
N09 G43 Z25. H1 M8
N10 (SLOT DRILL, TOOL DIA 12.)
N11 G0 X-13.500 Y-19.900
N12 G41 H17
N13 G01 Z-10. F50
N14 G01 X-10. Y0. F300
N15 G01 X-10. Y100.
N16 G02 X0. Y110. R10
N17 G01 X100. Y110.
N18 G02 X110. Y100. R10.
N19 G01 X110. Y0.
N20 G02 X100. Y-10. R10.
N21 G01 X0. Y-10.
N22 G02 X-10. Y0. R10.
N24 G40
N25 G0 X20. Y20.
N26 G83 Z-12. R2. Q3. F50
N27 G80 Z5.
N28 G0 X-50. Y-50.
N29 M9
N30 G28 G91 Z0. M5
N31 G28 X0. Y0.
N33 %
```

◀ **Program 4B**

USEFUL OBSERVATION

Find out how to load the text editor in the CAD/CAM package available to you. The way in which this is done varies from one CAM system to another, so if you have any difficulties get advice from someone who is familiar with the software.

POINTS TO NOTE

When changing the feedrates use the find/replace function, as this is quicker than manually making each change separately.

When editing to produce a separate program for machining the hole, remember to include the program blocks for the end of the program (blocks N28 to N33) and re-number the program blocks accordingly. Use the re-sequence function if this is available.

Learning Assignment 5

Work holding and tooling

CNC machine tools frequently operate at greater feedrates and speeds than those used in conventional machining, so larger forces may be applied to the component and therefore to the clamping/holding devices. As the programmer, it is part of your task to decide what type of work-holding device is required for a particular operation. The following conditions must be met:

1 When clamped, the component must not be able to move in any direction, either linearly or rotationally. It is essential that the tool cutting forces are exerted against a part or parts of the work-holding device that cannot be moved (Figure 5.1). This can be difficult to achieve when machining complex components of an irregular shape, such as castings, and you may need to use more than one holding device for all machining operations to be executed.
2 The component must not 'chatter' or deflect as it is being machined; this would produce a poor surface finish, dimensional inaccuracy and possibly tool damage. This may be overcome by using steadies for turning or fixture inserts when machining thin-walled components.
3 The component must be positively located and precisely positioned within the work-holding device. The part program datum point (that is, the point from which the co-ordinates of toolpath are programmed), is generally the same as the component datum point; so if the component is not positioned accurately the required component shape may not be achieved. A grid plate with a holding device attached can provide positive location and position in several directions.
4 In order to avoid collisions, the toolpaths must not coincide with any part of the work-holding device or clamps. When proving out the program, you may find you need to insert moves in the program to ensure this condition is met.
5 The work-holding device must provide adequate location and support to achieve the geometric tolerances specified on the component drawing.
6 All information relating to the work-holding device(s) used and the setting of datums must be documented for the machine setter/operator.

TASKS

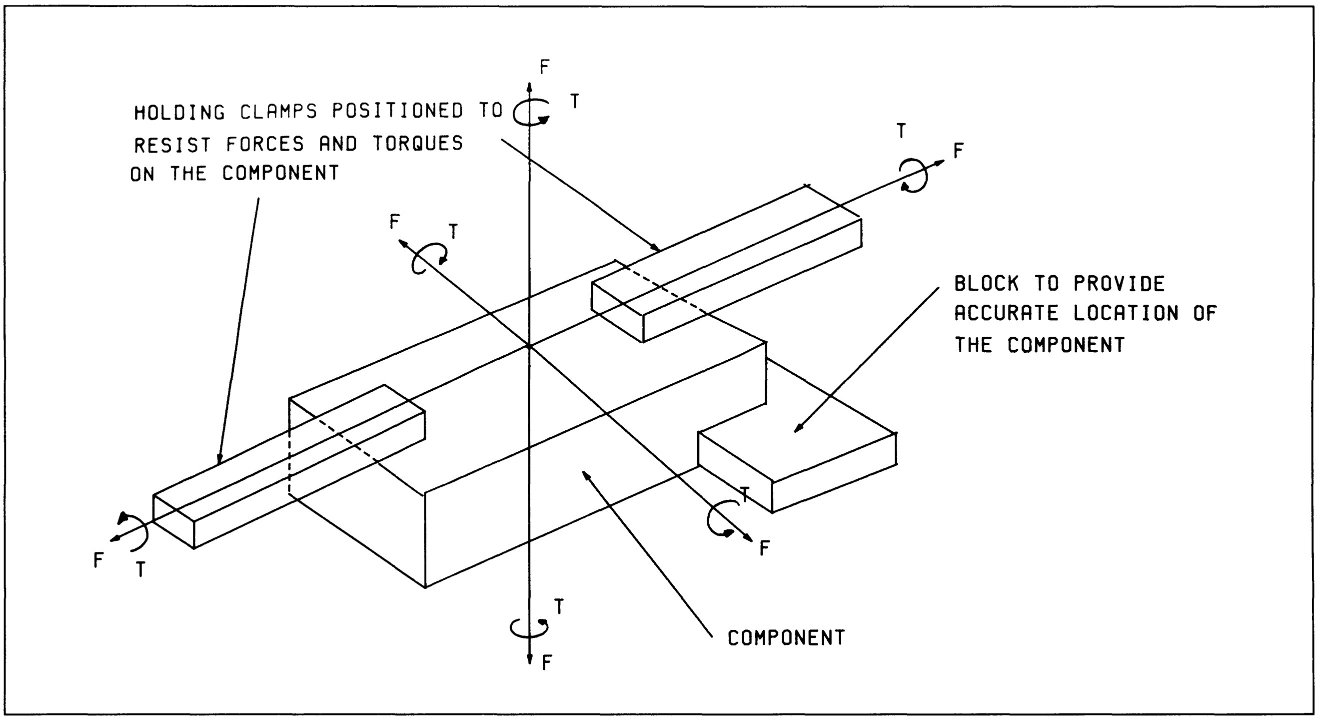
Task 5.1 Identifying work-holding devices

On a CNC machine of your choice, examine the work-holding devices available and comment on:

- the method of locating/fixing the device to the machine table, slideway or spindle
- how the component is located in the device to ensure that the component datum is always in the same position
- the method(s) used to secure the component in the work-holding device
- any safety or 'fail safe' features that are built into the work-holding devices.

Tooling

In order to machine components to the tolerances specified on the part



▲ **Figure 5.1** Forces and torques being exerted on the component due to the cutting action of the machine tool

drawings, at a high production rate and at minimal cost, you must have a good knowledge of the tooling system for the machine that is being programmed. You therefore need to be familiar with the range of tools available so that you can select the appropriate tool(s) for the machining process. You will need to consider:

- the type of tool holder and how it is held in the tool post or milling machine quill (Figure 5.2)
- the shape, size and type of tool – for example, a 10 mm dia endmill, a 14 mm dia slotdrill, a 5 mm parting-off tool or a right-hand roughing tool
- the method of holding the tool tip insert(s) within the holder and details of the tool tip insert being used, in particular the material from which the insert is made and the coating material, its geometrical configuration, its physical size and the manufacturer's recommended cutting feeds and spindle speeds for the material being machined

You will also need to be familiar with the tool classification and/or coding system used within your company. (Most companies try to standardise on their tooling to reduce overheads and to aim at performing similar tasks with the same tools.) Tools can be classified and coded numerically, so that a particular tool can be identified by its code numbers. For example, a lathe tool could be identified by numbers indicating the serial number for the tool holder, the tool nose radius, the insert material, any setting dimensions if it is a preset tool, the tool shape and the type and shape of the tool insert (an example is shown in Figure 5.3).

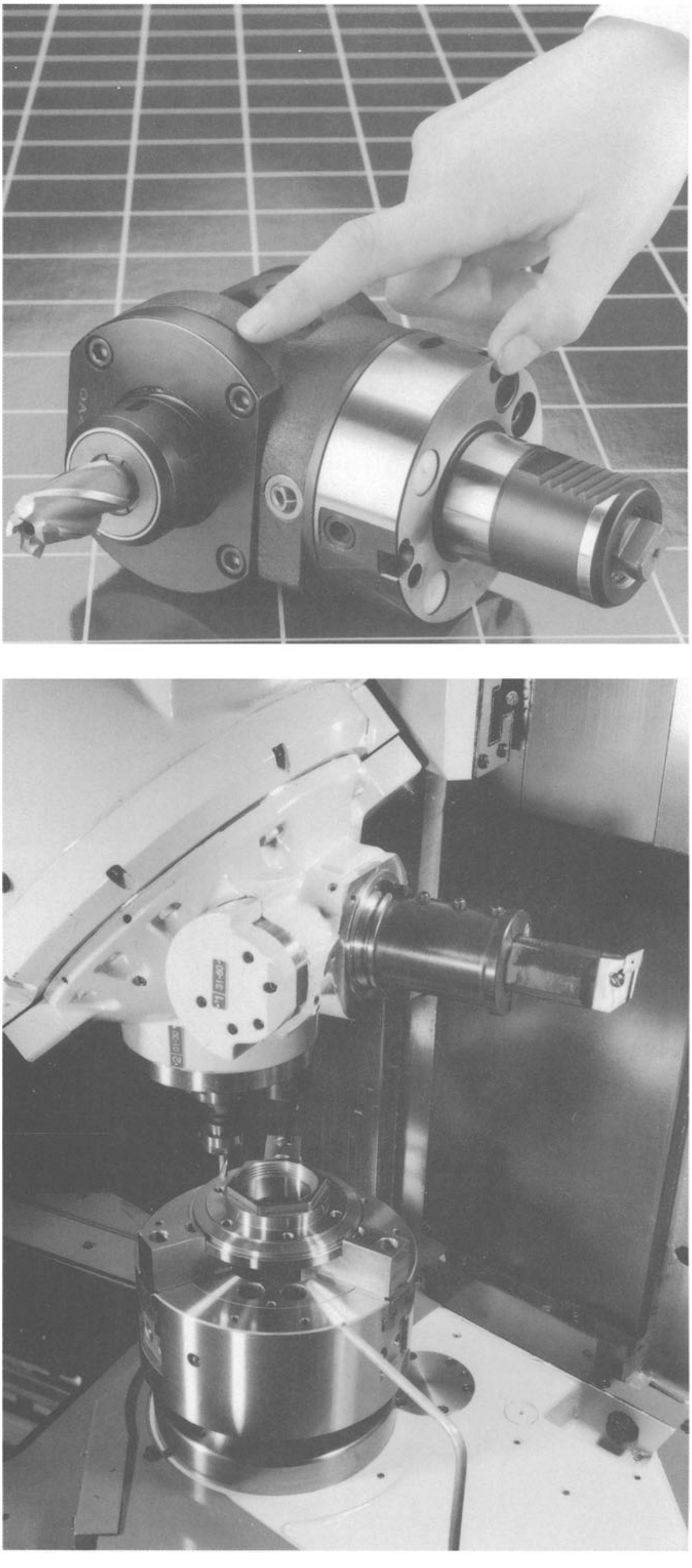
As programmer, you will be required to state the tool length offset number in the program for each tool. If, for instance, the tool length offset for tool number 4 is entered in the control tool offset system register number 7, the part program would show this as T0407.

You should document all information relating to the tooling specified, as part of the operator's documentation for machining the component.

Optimum cutting conditions

The choice of cutting feedrates and spindle speeds for machining a particular component depends very much on the experience of the programmer. The factors to be taken into consideration when determining the appropriate feedrates and speeds include the following:

▶ **Figure 5.2** (a) A tool holder; (b) the tool holder located in the turret



(a)

(b)

TITLE —— ROUGHING TOOL

CNC MACHINE —— T/006 (FANUCOT)

TOOL No	TOOL HOLDER No	NOSE RAD	INSERT No	INSERT CODE	TURRET HOME POSITION		
					A	B	C
T901	PDAJ 5312P65	0.8	DNMB16312	P30	X205.498 Z4.872	X193.760 Z103.765	X183.455 Z184.674

▲ **Figure 5.3** Tool classification/coding sheet

- the type and condition of the CNC machine tool (that is, the maximum power available and the age of the machine)
- the rigidity of the machine tool and the associated work-holding devices
- the size, shape and type of tooling that is being used
- the type of material that is to be machined, the amount of material that is to be removed and the machining process that is to be executed (for instance, whether it is drilling, tapping, rough stock removal or finish cutting)
- the machining tolerances and surface finish
- the type of coolant being used.

The data published by cutting tool manufacturers acts as a fairly good guide. It must be treated with some caution, however, since the figures quoted are often for ideal machining conditions, and may not be fully applicable to the machine tool, material or tooling 'set up' that is being used.

TASKS

Task 5.2 Work holding and tooling for a given component

Look at Figure 7.7 (page 32). Describe the tools that you would use for machining the component shown, and a method for holding the workpiece on the machine bed.

Task 5.3 Types of available tooling

For a CNC milling machine and a turning machine, investigate the types of tooling that are available for the programmer to use. In particular, look at the following:

- the type of tool storage, either on the machine or adjacent to it
- the methods used for locating and holding the tools in the machine tool
- the tool classification/identification system and the codes used for one milling cutter and one lathe tool
- how to determine the feedrate and spindle speeds for the above two tools when machining aluminium (6082TF) and mild steel (220M07) respectively (consult manufacturers' guide books for cutting tool materials)
- how the tool offsets are entered into the machine control, edited and displayed within the part program at the tool change position.

Learning Assignment 6

Toolpath calculations

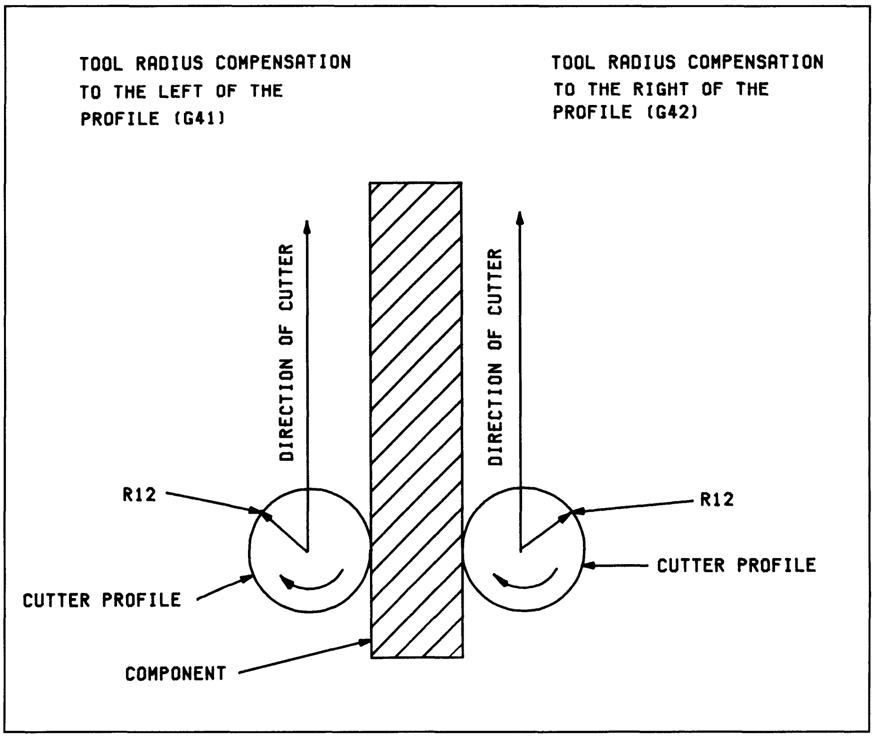
Before you can start to write a CNC part program manually, you may have to carry out various calculations to determine the tool centre line path in order to machine the required shape. These calculations may be needed to determine the intersect points of arcs and lines, arc centres and tangency points of lines on arcs or circles, and so on. If you are going to write programs manually, you will need a good knowledge of mathematics, trigonometry and geometry. These are not discussed in this workbook, since they are covered in the accompanying core book. If you feel you are on uncertain ground here, there are plenty of mathematical textbooks to which you can refer.

Cutter compensation

A valuable aid to the manual part programmer is the cutter tool radius compensation facility that is built into most CNC machine controllers. This allows you to write a program stating the X and Y, or X and Z, co-ordinates of the profile or other machining features, without having to calculate the centre line path of the tool (milling) or the centre line of the tool nose radius (turning).

To apply tool cutter compensation when programming in word address format, you need to call a G41 or G42 code, depending on whether the tool centre line is to be offset to the left or to the right of the programmed co-ordinates. The G41 code moves the tool to the left-hand side of the profile and a G42 to the right-hand side of the profile by the value of the cutter radius (Figure 6.1). Enter the measured tool radius and store it, with the relevant tool number (T02, for example), in the controller offset register.

It is good practice to apply cutter compensation some distance away from the actual workpiece, as this allows the tool to offset by its radius



▶ **Figure 6.1** Applying tool radius compensation before it makes contact with the workpiece. This also applies to cancelling tool cutter compensation by programming a G40 code – that is, you should move the tool away from the work before cancelling cutter compensation.

Radius offsets can be used if the finished machined component is oversize or undersize, perhaps because of the effects of tool wear. The effective size of the tool can be altered in the offset register to be smaller than the actual tool used. When re-machined, the component will then be that much smaller. Likewise, if a larger tool offset is entered the component will be larger.

Additional information

If you are not quite sure on which side of the component you should apply cutter radius compensation, imagine that you yourself are the cutting tool. As you approach the component profile, decide whether to step to the left or right to produce the required profile.

TASKS

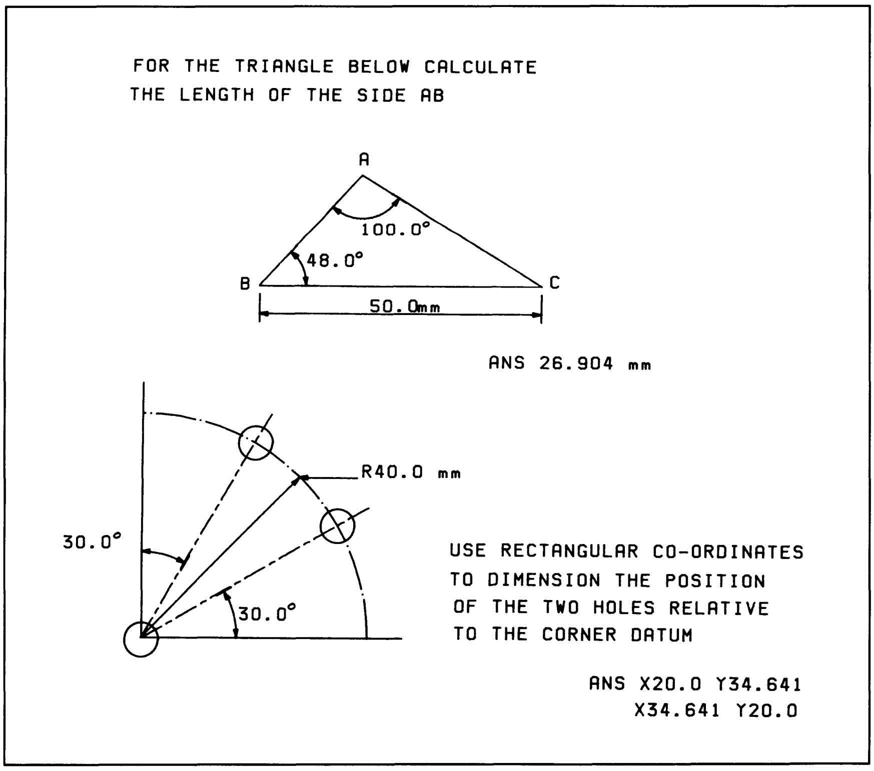
Task 6.1 Problem-solving using trigonometry

Answer the questions shown on Figures 6.2 and 6.3, which are exercises in solving problems using trigonometry and Pythagoras' theorem.

ADDITIONAL TASKS

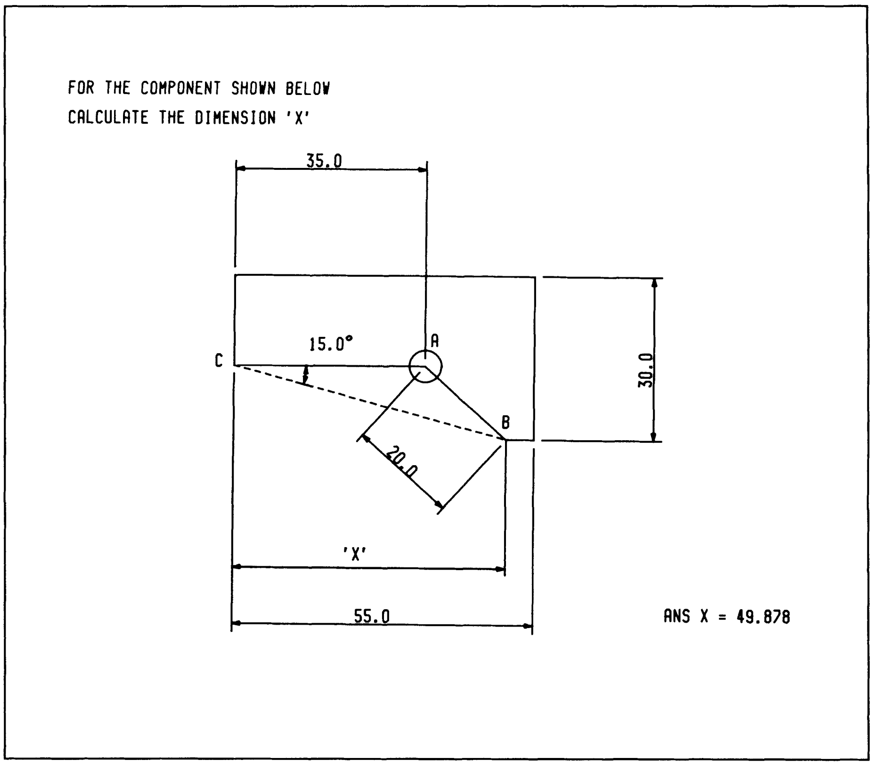
1 In which direction is cutter compensation applied when machining an outside profile and a pocket, with the spindle rotating clockwise? (*Hint:* when using CNC machines, it is good practice to 'climbmill', see page 46.)

2 The 'first off' of a batch of machined components has been inspected, and the inspector has found that the component profile was oversize by 0.1 mm. A 12 mm diameter endmill was used. How can the component be corrected without re-writing the part program?



◀ **Figure 6.2** Trigonometry exercises (Task 6.1)

▶ **Figure 6.3** Trigonometry exercise (Task 6.1)



Learning Assignment 7

Planning a CNC program

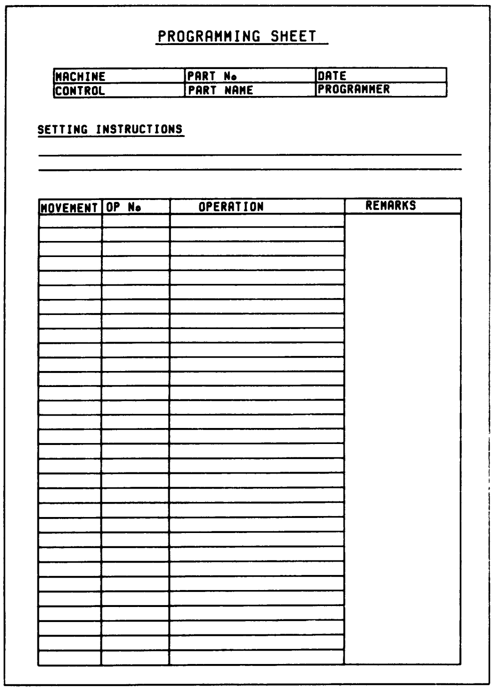
When you are beginning to construct a program there are many factors that you need to take into consideration, and a systematic approach is essential.

The first consideration is to plan the machining sequence, ensuring that all machining operations are carried out in a logical and economical way. At this stage you have to determine how the work is to be held: the clamping and mounting methods, the fixtures to be used and/or any machine accessories (such as rotary tables and/or steadies) that may be required. You also have to specify the tools and tool holders to be used, and any inspection tools or gauges that will be needed to check on the dimensional accuracy of the machined part. You will have to prepare a set of drawings and instructions for the machine setter and operator, to accompany the program.

For manual programming, you have to carry out all calculations for the co-ordinate points, the intersect points and such functions as feeds and speeds. This information is recorded on to a program sheet (an example of a blank form is shown in Figure 7.1), which also contains the co-ordinate moves, preparatory codes and miscellaneous functions, machine addresses and information for the operator.

You can enter the program into the CNC control by manual data input (MDI), or it can be downloaded from a computer, either directly or via paper tape or magnetic tape cassette.

Next, you need to enter the tooling offsets into the program. If you are using a manual tool change system, you should arrange all the tools in a tool storage rack according to the order in which they are going to be used in the program – for example, 15 mm dia slotdrill T1, 10 mm twist drill T2, 10 mm dia endmill T3. This reduces the time for tool


PROGRAMMING SHEET

MACHINE	PART No.	DATE
CONTROL	PART NAME	PROGRAMMER

SETTING INSTRUCTIONS

MOVEMENT	OP No.	OPERATION	REMARKS

◀ **Figure 7.1** A programming sheet

changing, since the correct tools are ready to hand in the required sequence.

The next stage is the 'proving out' of the program. You can carry this out in several different ways:

- you can machine the component in a suitable proving-out material (see page 18)
- on a milling machine, you can execute a 'dry run' by moving the machine table down to below the lowest programmed –Z value and running the program so that the tool is 'cutting' in air
- you can 'prove out' your program in metal by executing it in single block mode.

When the 'prove out' run is completed, inspect the part and make any necessary alterations to feeds/speeds, co-ordinate values, tool compensations and so on, using the machine tool editor.

Keep a record of the proven program in the form of a program listing, and store it on some medium (paper tape, floppy disc or magnetic tape) for future reference. This record should also include information about the set up – tools, fixtures and machining sequences – for any subsequent batches.

The planning stages of a CNC program can be summarised as follows:

- machining sequence – sequence of machining operations, fixtures, clamping, tools, tool holders, measuring tools, preliminary instructions and drawings
- calculations – co-ordinate points, feeds and speeds, depth of cuts etc.
- programming – preparatory codes, miscellaneous functions, co-ordinates
- input of data – MDI, punched tape, 'hard wiring' (downloading from a computer via a cable connection)
- tool setting – setting tool length offsets etc.
- proving – by 'dry run' etc., inspection of workpiece, editing the program where appropriate
- documentation – program listing, tool and fixture lists etc., storing.

Additional information

When planning the toolpath, give careful consideration to the tool change positions and the path that the tool takes to remove the metal. Avoid all unnecessary moves: although these moves may take only a few seconds, for a large batch they could result in a considerable waste of machine time. Make sure that automatic tool changing is executed at a safe position away from the work and clamps.

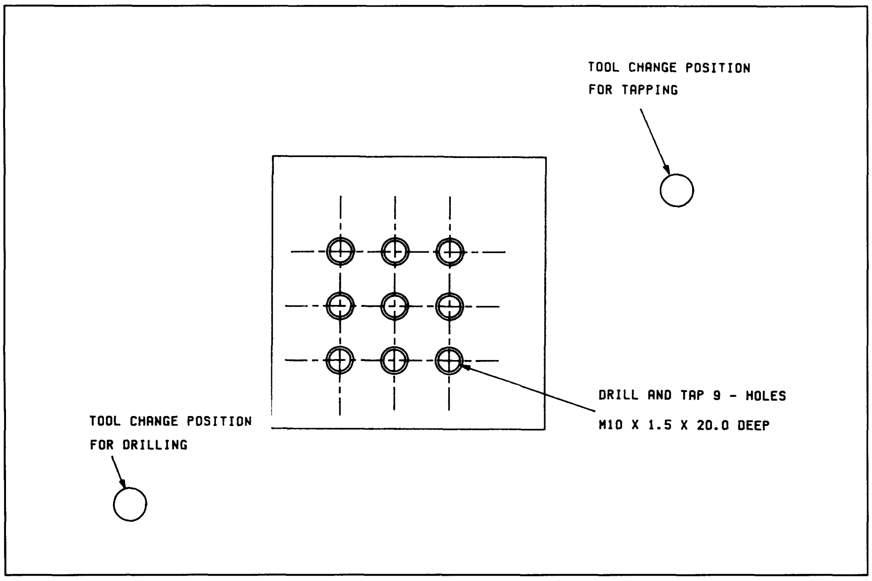
You can use various techniques throughout the program to save time. This is illustrated in Figure 7.2, where holes are to be drilled and tapped in a plate. The manual tool change position for collecting the drill is at the bottom left-hand side of the work and the tool change position for collecting the tap is on the right at the top. This eliminates an unnecessary move to the original tool change position to collect the tap.

To avoid damage to the tool and workpiece, check the toolpath carefully to ensure that there can be no collision with clamps, parts of a fixture or anything else on the machine bed during machining. You can verify this when you prove out the program. Check the application of tool cutter diameter compensation, to ensure it is made on the correct side of the workpiece.

You can incorporate optional stops into the program so that the operator may carry out an intermediate inspection of the workpiece, remove swarf, move or change clamps, or make any necessary adjustments to the tooling or fixturing.

USEFUL OBSERVATION

If you are using an existing program, identify the information given to the operator and the format that is used. Also note any optional stops in the program, and try to determine why they are included.



◀ **Figure 7.2** Drilling and tapping holes

Documentation

Documentation is an important part of planning and executing a CNC program because it enables the programmer to work in a logical manner through the part-programming sequence. It can also provide valuable information for the tool setter and/or operator. Moreover, the documentation can be referred to later, perhaps when a repeat order is received or when a similar component is to be machined.

CNC documentation varies from one engineering company to another, as a result of the company's policy together with its experience of programming and machining. In general, companies use documentation forms which indicate the CNC machine tool that is to be used – this could be the machine number or type and also the name of the CNC control fitted to it – together with the part name and part number of the component to be machined, the date when the program was written and the programmer's name. All this information would be included in the header for each sheet. A typical set of blank documentation sheets is shown in Figures 7.3 to 7.6.

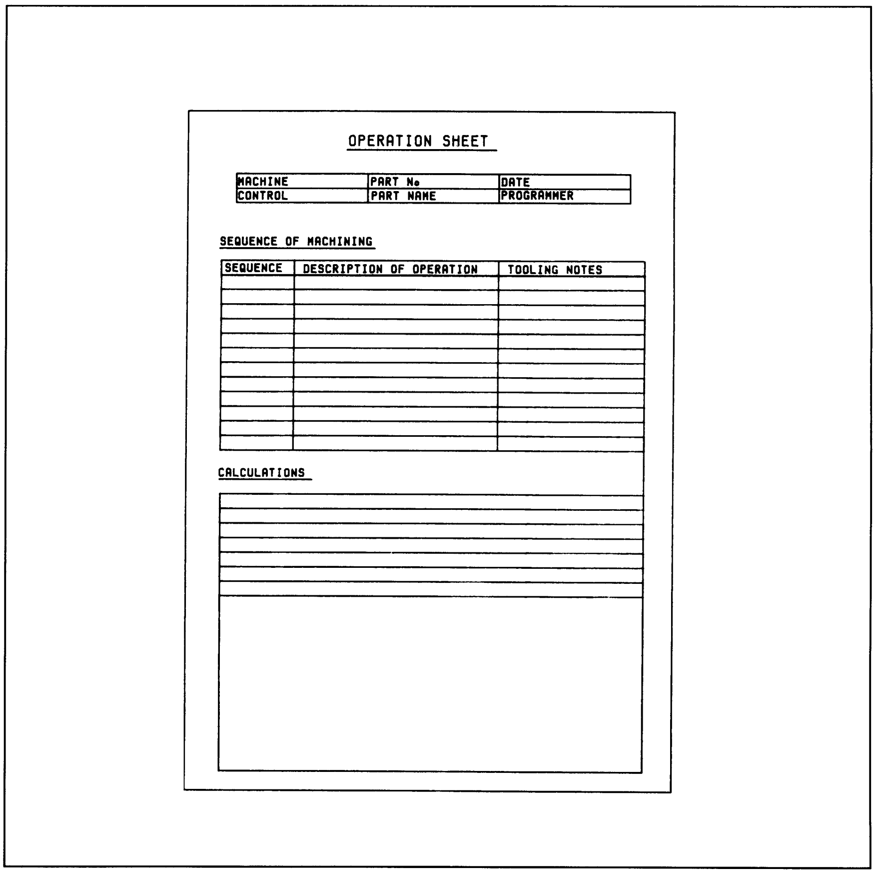
The **operation sheet** (Figure 7.3) shows the sequence that the component is to be machined, the description of each operation and tooling notes. For example, an entry might read: 'Operation 5 – rough turn 45 mm dia to 40 mm – use tool T10'.

The **work holding sheet** (Figure 7.4) contains information relating to the fixture or clamping device to be used. The fixture identification number is usually stated, and this sheet can include a sketch of the set up together with any relevant remarks or instructions such as the datum positions.

The **tool setting sheet** (Figure 7.5) gives a description of the tool to be used for each operation, the tool number and its identification code. It can also include details of the tool tip insert, the grade of the insert and the nose radius, together with the recommended spindle speed and the feedrates to use. A typical entry could be 'Operation 6 – Tool number 3 – Slotdrill 10 mm dia/TSD10/782 – Tip insert SF30/P25 – Nose radius 0.02 mm – Spindle speed 1000 rev/min – Feed per tooth 0.2 mm – Feedrate 150 mm/min'. It should also include information relating to the tool length, for use in setting up tool length compensation on the machine tool.

The **programming sheet** (Figure 7.6) lists the CNC part program, such as preparatory codes, miscellaneous functions and other programming codes. Sometimes tool movement can be shown in the form of arrows; for example, a horizontal arrow can represent tool movement from left to right, and a circular arrow clockwise movement. Included in this sheet would be relevant remarks for the machine tool setter/operator, such as 'Operation 3 – load casting in fixture'.

▶ **Figure 7.3** An operation sheet



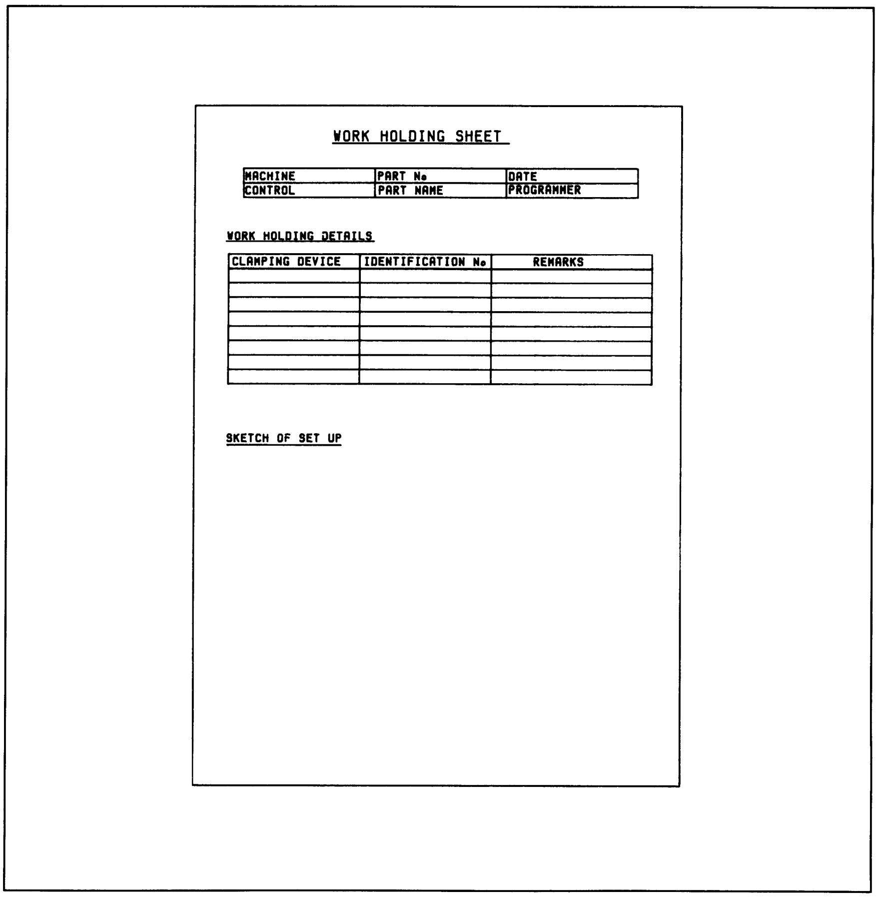
OPERATION SHEET

MACHINE	PART No	DATE
CONTROL	PART NAME	PROGRAMMER

SEQUENCE OF MACHINING

SEQUENCE	DESCRIPTION OF OPERATION	TOOLING NOTES

CALCULATIONS



WORK HOLDING SHEET

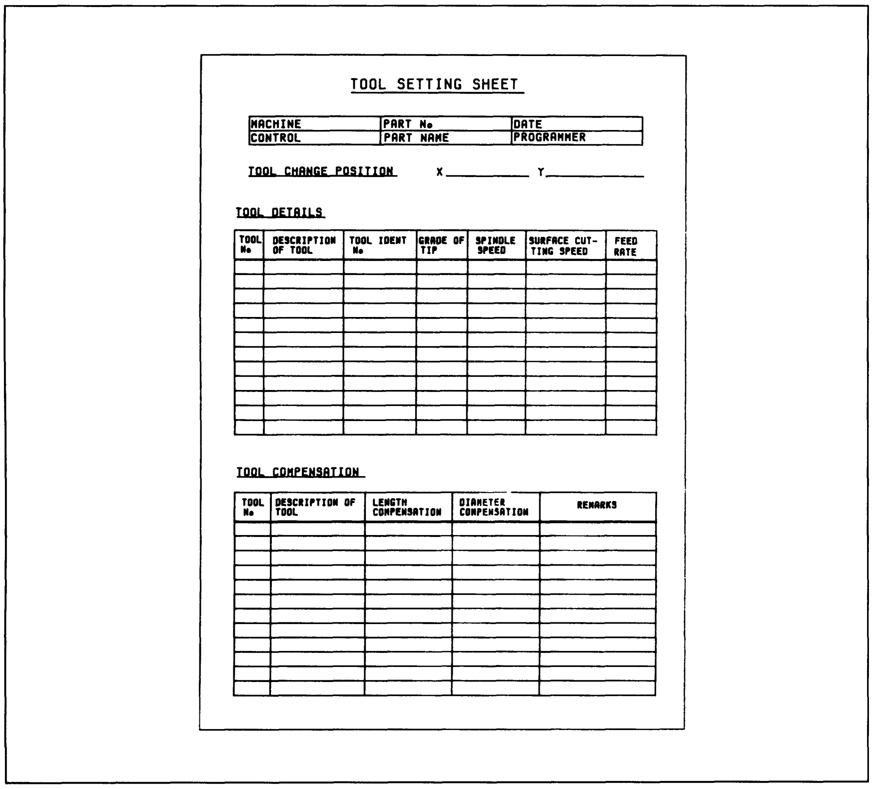
MACHINE	PART No	DATE
CONTROL	PART NAME	PROGRAMMER

WORK HOLDING DETAILS

CLAMPING DEVICE	IDENTIFICATION No	REMARKS

SKETCH OF SET UP

▶ **Figure 7.4** A work holding sheet



TOOL SETTING SHEET

MACHINE	PART No	DATE
CONTROL	PART NAME	PROGRAMMER

TOOL CHANGE POSITION X________ Y________

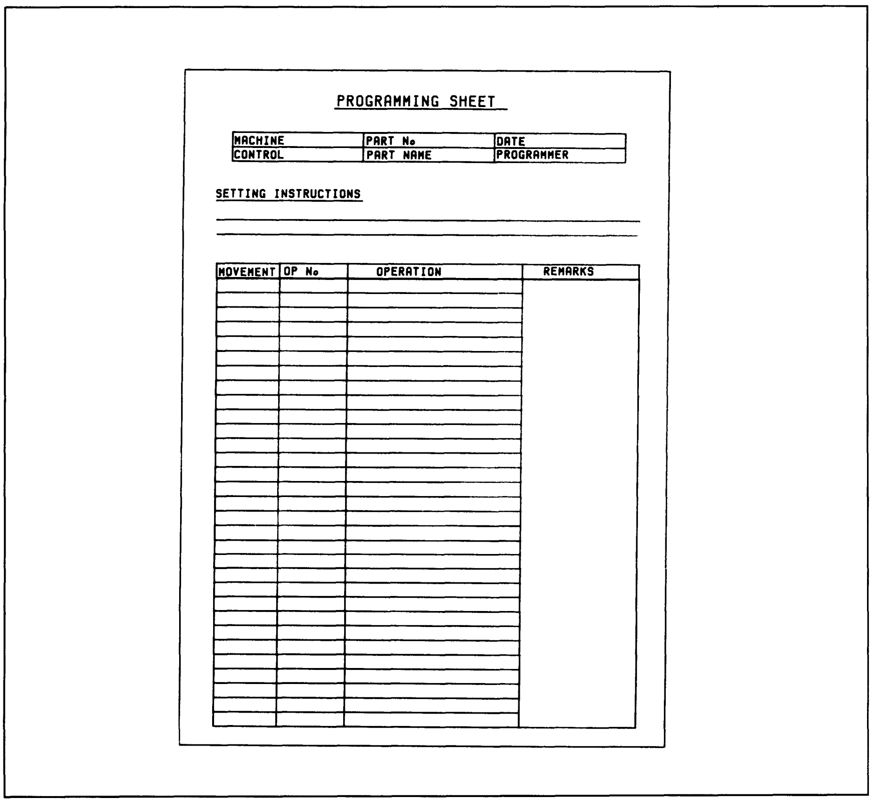
TOOL DETAILS

TOOL No	DESCRIPTION OF TOOL	TOOL IDENT No	GRADE OF TIP	SPINDLE SPEED	SURFACE CUT-TING SPEED	FEED RATE

TOOL COMPENSATION

TOOL No	DESCRIPTION OF TOOL	LENGTH COMPENSATION	DIAMETER COMPENSATION	REMARKS

◀ **Figure 7.5** A tool setting sheet



PROGRAMMING SHEET

MACHINE	PART No	DATE
CONTROL	PART NAME	PROGRAMMER

SETTING INSTRUCTIONS

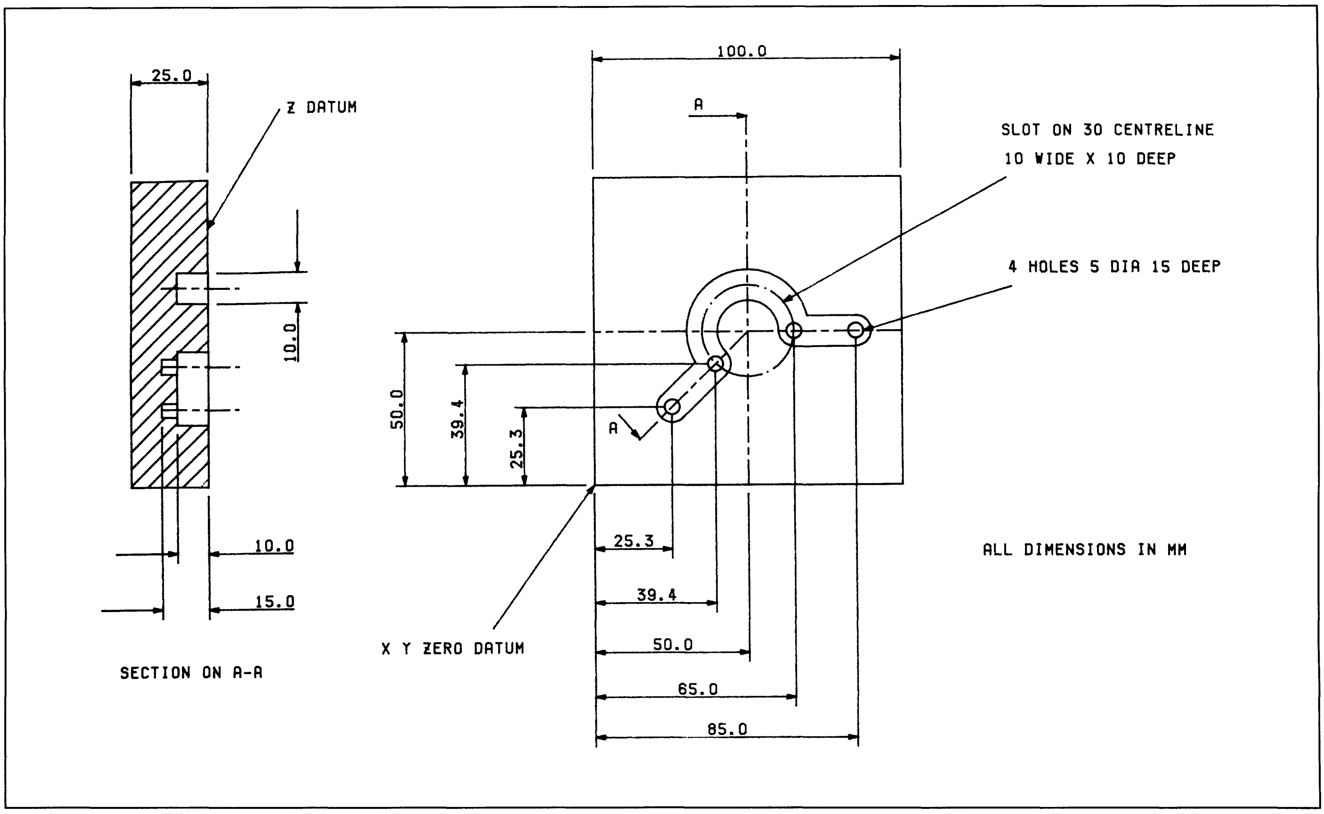
MOVEMENT	OP No	OPERATION	REMARKS

◀ **Figure 7.6** A programming sheet

TASKS

Task 7.1 Writing a program to include messages

Write a program to machine the 10 mm groove and drill the four holes in the guide plate shown in Figure 7.7. Include in your program relevant messages for the operator and use the program sheets that you have constructed for listing the tools, sequence of operations, work-holding devices and so on.



▲ **Figure 7.7** Guide plate (Task 7.1)

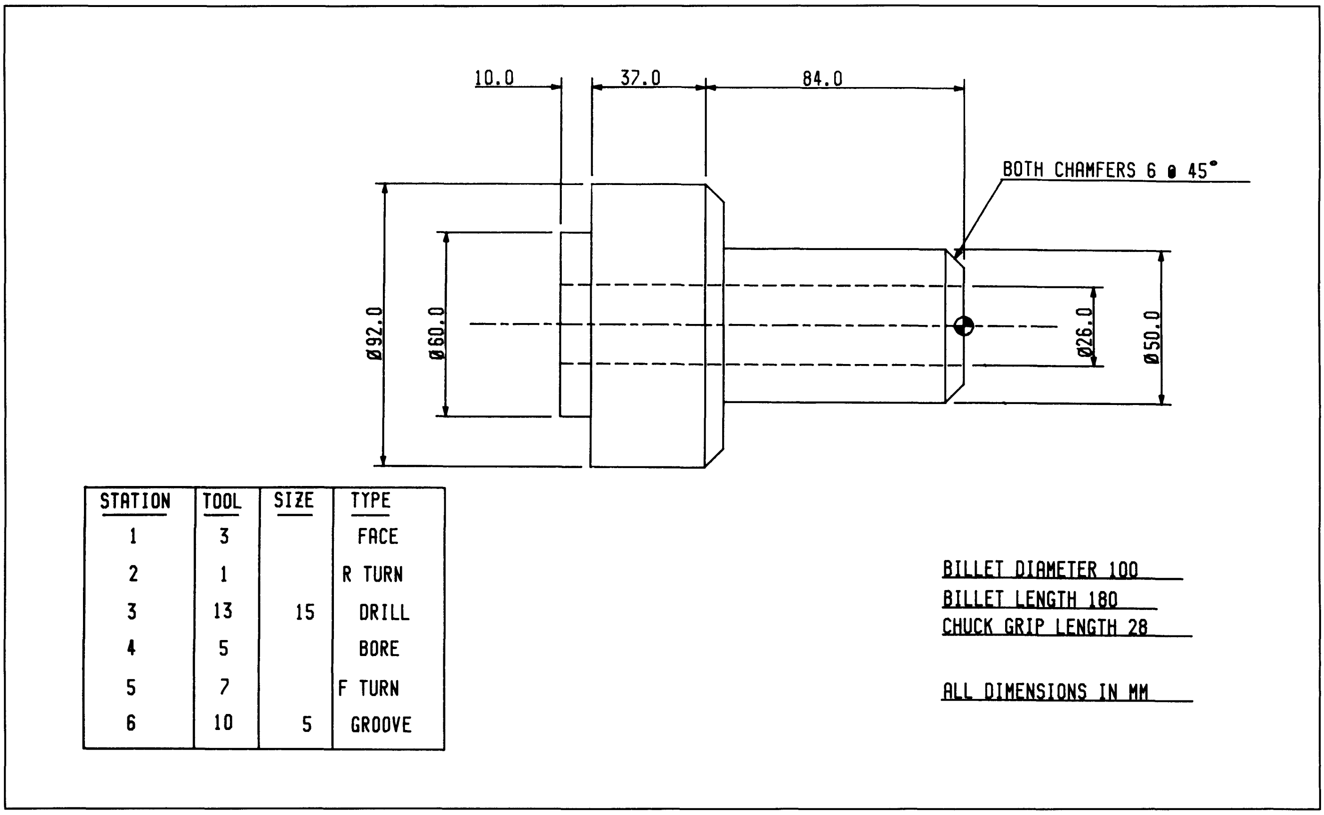
TASKS

Task 7.2 Documentation

For a simple turned component of your choice, requiring one tool for machining, design your own documentation that will include an operation sheet, work holding sheet and tool setting details.

Repeat this exercise for the turned component shown in Figure 7.8. Note that a tooling list is shown on the drawing.

▼ **Figure 7.8** A turned component (Task 7.2)



STATION	TOOL	SIZE	TYPE
1	3		FACE
2	1		R TURN
3	13	15	DRILL
4	5		BORE
5	7		F TURN
6	10	5	GROOVE

```
%
O2000
(PROGRAM DATE - 21/01/94)
(CUSTOMER NAME - AEROSPACE
FACTORS INC)
(DRAWING NUMBER - ETHW
27754/6-E)
(PART NAME - HINGED PLATE)
(MATERIAL - ALUMINIUM ALLOY)

N01 G28 G91 G40 G80 G0 Z0.
N02 G28 X0. Y0.
N03 G21
N04 G92 X0. Y0. Z0.
N05 G90
N06 M9
N07 G28 G49 Z50. M5
N08 T2 M6
N09 (HOLD IN VICE TO DRILL
AND REAM HOLES)
N10 G0 G90 M3 S2500
N11 G43 Z25. H2 M8
N12 (TOOL DIA 10.000 DRILL)
N13 G83 G98 Z-12.000 R2.000
Q3.000 F50
N14 X138.200 Y108.500
N15 X203.300 Y117.100
N16 X174.900 Y83.000
N17 G80
N18 G0 X39.300 Y26.300
N19 M9
N20 G28 G49 Z50. M5
N21 T3 M6
N22 G0 G90 M3 S360
N23 G43 Z50 H3 M8
N24 (TOOL DIA 10.000 REAMER)
N25 G81 G98 Z-12.000 R2.000
F30
N26 X138.200 Y108.500
N27 X203.300 Y117.100
N28 X174.900 Y83.000
N29 G80
N30 G0 X39.300 Y26.300
N31 M9
N32 G28 G49 Z50. M5
N33 T3 M6
N34 G0 G90 M3 S2500
N35 G43 Z50 H3 M8
N36 (TOOL DIA 16.000 SLOT-
DRILL)
N37 (HOLD IN FIXTURE
FX3665/6, LOCATE DOWEL PINS
IN HOLES)
N38 G0 X89.000 Y64.600
N39 G01 Z-10.000 F50
N40 G41 H19
N41 G01 X95.466 Y93.079 F250
N42 G01 X122.259 Y136.509
N43 G02 X137.065 Y145.050
R18.000
N44 G01 X234.626 Y147.831
N45 G02 X248.513 Y117.792
R18.000
N46 G01 X194.400 Y57.760
N47 G02 X177.592 Y52.145
R18.000
N48 G01 X107.312 Y65.966
N49 G02 X95.466 Y93.079
R18.000
N50 G0 X88.000 Y97.200
N51 G0 Z20.000
N52 G40
N53 G0 X39.300 Y26.300
N54 M9
N55 G28 G91 Z0. M5
N56 G28 X0. Y0.
N57 M30
N58 %
```

◀ **Program 7A**

POINTS TO NOTE

Problems often arise in machining because the cutter compensation has been applied or cancelled in the wrong place, or when the approach to the work, or the leaving of it, is executed in an unacceptable way.

It is important that the machine operator is supplied with all information regarding the machining sequence, the tooling arrangements and details of the fixtures and clamping that are to be used for the part program. The most effective way of giving this information is in the form of a set-up sheet together with a print-out of the CNC machine program. This should include instructions within the program where these are to be executed. A typical set-up sheet is shown in Figure 7.9.

TASKS

Task 7.3 Identifying tooling commands

In Program 7A, identify and bracket each of the tooling operations as a self-contained sequence. Note these sequences, together with the instructions for the operator.

```
SET-UP SHEET

TOOL REQUIREMENT FOR JOB - ETHW 27754/6-E
DATE: 26-01-1994            TIME: 13:19:29
=============================================================
UNITS = MM
TOOL NO. 02 OFFSET 00
            NAME = DRILL      MATERIAL = ALUMINIUM (SELECTED
                              SPEED/FEED = S2500 F250)

TOOL NO. 03 OFFSET 00
            NAME = REAMER     MATERIAL = STEEL
                              (SELECTED SPEED/FEED = S260 F120)

TOOL NO. 01 OFFSET 00
            NAME = SLOTDRILL MATERIAL = ALUMINIUM
                              (SELECTED SPEED/FEED = S2500 F250)
=============================================================
```

◀ **Figure 7.9** A set-up sheet

Canned cycles/macros

All CNC machine tools have, as part of the programming facility within the controller, the feature of executing machine **canned cycles** and/or sub-programs or **macro** routines. These are built into the controller's computer software and make it easy to program routine machining sequences. Such routines are drilling, tapping, spot-facing and area clearance for square pockets, circular pockets and boring cycles. The difference between the two is that a canned cycle is inherent within the control system and usually cannot be changed or re-programmed, whereas a macro is a separate program generated by the programmer and inserted within the main program or, in the case of some controls, stored apart from the main program text.

There are several advantages to the use of these 'in-built' features:

- they reduce programming time for the programmer
- they reduce the length of the program, which is useful when the program length is near to, or greater than, the capacity of the control memory
- they allow the programmer to write special routines for executing particular sequences which may be used frequently in their field of manufacture, such as holes on a PCD, special deep drilling cycles or common pocket clearance programs which are to be repeated on a range of different components.

Canned cycles

As explained, these routines are for executing different types of drilling, thread cutting and area clearance machining. In word address format they take the form of a preparatory function (such as G83), followed by co-ordinate information, retract height, pecking distance/number, dwell value, feedrate and so on. A typical canned cycle is shown below:

N25 G83 G99 X100.000 Y150.000 Z–12.000 Q3.0 R2.000 F300

where G83 refers to a peck drilling cycle
G99 instructs that the cycle commences at the retract height of the cycle call
Z = the depth to which the hole has to be drilled
Q = the number of pecks to be executed
R = the retract height which the drill moves between each hole
F = the programmed feedrate.

Another example of a canned cycle is G76, which is used on turning machines for machining a thread. This takes the form of:

G76 Z–25.000 X8.000 I0.25 P1.5000 Q0.750 F2.000

where G76 is the canned cycle call
Z = length of thread
X = depth of thread
I = thread radius
P = height of the thread
Q = depth of the first cut
F = lead for the thread.

Macros

The macro statement or sub-program can execute various machine moves and cycles, and can be executed any number of times within a main program. This is useful to the programmer, who thus needs to program a machining sequence only once, even though the sequence may be repeated several times within the program at different positions.

Program 8A is a typical program using a macro. In this program the macro is positioned towards the end of the main program, as macro number 2000. In fact, a macro can be positioned anywhere in the main program, since the control system scans the entire program when the macro is called up. It is, however, good practice to keep all the macros together either at the start or end of the program as this makes the part program easier for the operator and the programmer to understand.

In Program 8A, the macro is called up by using the miscellaneous function M98 followed by the macro number. The macro program is started with a colon (:), followed by the macro number, and terminated with a M99 function. A macro can be repeated by inserting the number of times that it needs to be executed in the line of the macro call; for instance, N39 M98 P32000 will execute the macro number 2000 three times.

▼ **Program 8A**

```
%
O1000
N01 G28 G90 G40 G80 G0 Z0.
N02 G28 X0. Y0.
N03 G21
N06 T1 M6
N07 G0 G90 X-23.0 Y0. M3
S3000
N08 G43 Z25.H1 M8
N09 G01 X169. F400
N10 Z-2.000 F50
N11 G01 X-23.
N12 G0 M9
N13 G28 G49 Z150. M5
N14 T3 M6
N15 G0 G90 Z144. Y0. M3
S3000
N16 G43 Z25. H3 M8
N17 M98 P2000
N18 X132.
N19 M98 P2000
N20 X120.
N21 G0 M9
N22 G28 G91 Z0. M5
N23 G28 X0.Y0.
N24 M30
:2000
N01 G0 Z-3.700
N02 G01 G41 G91 X0.000 Y-
22.038 F200 H17
N03 X10.511 Y0.000
N04 X-20.537 Y-32.537
N05 X21.026 Y-22.038
N06 G01 G40 X32.022 Y0.000
N07 G0 G90 Z2.000
N08 M99
%
```

Additional information

When using canned cycles, some milling controls use the Z depth value as being incremental from the last Z position called. This needs to be examined, since the previous Z position may be, say, 5 mm above the work surface. In such a case, 5 mm needs to be added on to the Z depth in the canned cycle. Generally the Z values are taken from the Z datum (the work surface), and have a positive sign for going into the metal. If a different datum is used, the canned cycle needs to be adjusted accordingly.

TASKS

Task 8.1 Identifying canned cycles

Examine the canned cycles that are present within the CNC machine with which you are familiar. Make a list of their functions and draw the tool movements that are associated with each one.

Task 8.2 Identifying macro programs on a CNC machine

Investigate, on a CNC machine tool with which you are familiar, how macro programs are written. Explain how a macro can be written as a separate program independent of the program with which it is to be used, i.e., having its own program number.

Task 8.3 Programming a cover plate

Figure 8.1 shows a drawing of a cover plate which has seven holes to be drilled and counterbored. Using drilling and counterboring canned cycles, write a program that will machine the holes. To reduce the program length write a macro for the hole positions, and use this macro when executing the drilling and counterboring canned cycles.

Task 8.4 Writing a macro program

A simple profile 20 mm deep is to be machined in two cuts 10 mm deep, using a 18 mm diameter slotdrill. Write a macro to rough-machine the profile, leaving 1 mm on the profile for the finishing cut. In the main program, lower the tool to –10 mm and call up the macro. On completion, lower the tool to –20 mm in the main program and repeat the macro. Change the tool for a 20 mm

A POINT TO NOTE

Macros are usually programmed in incremental format, as they can then be easily programmed and executed anywhere along the spindle axes or on the machine table. As moves from one macro position to another are easily programmed in absolute mode (i.e., relative to the program datum), it is advisable to end the macro program with a G90 so that the machine is in absolute format when re-entering the main program.

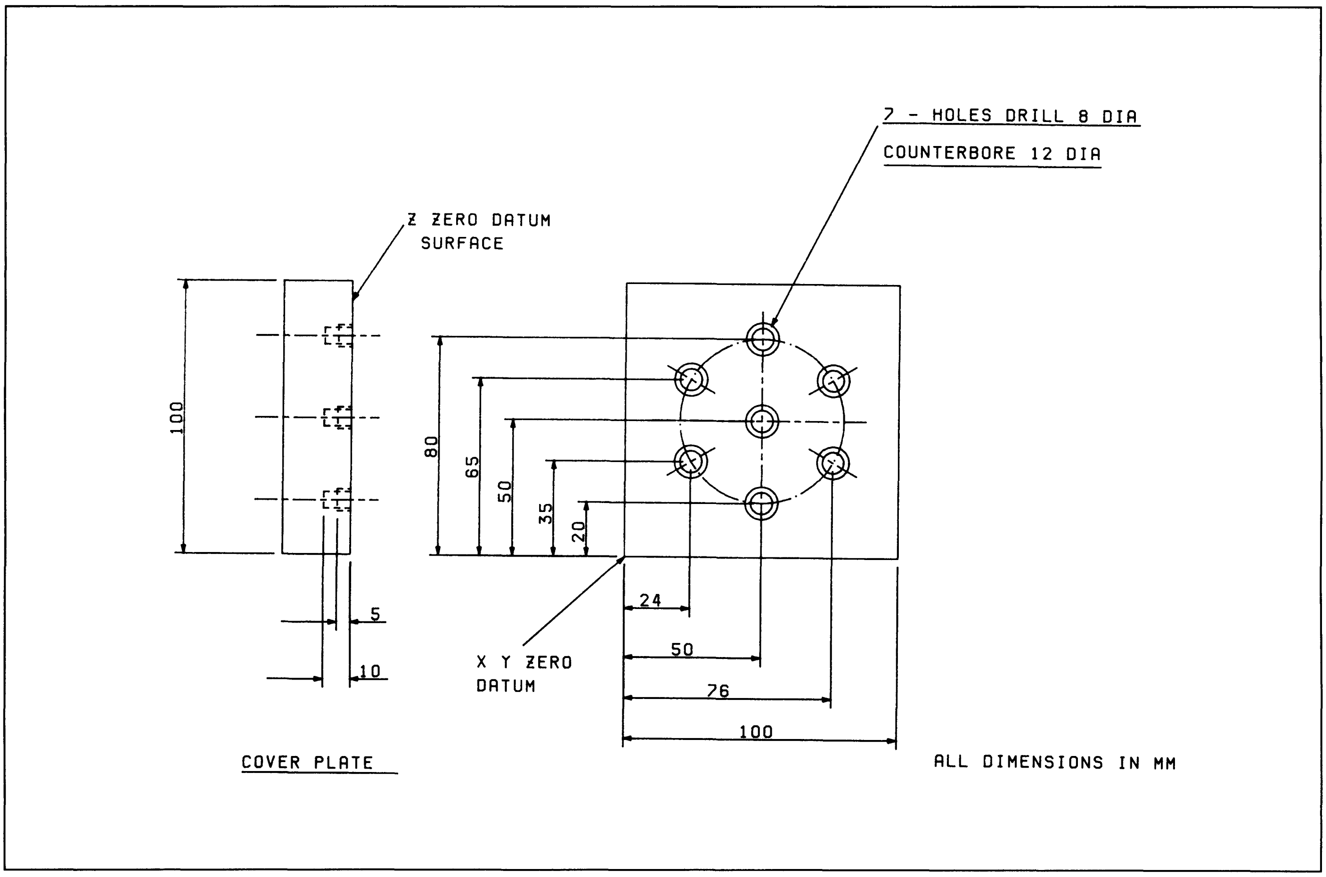
TASKS

diameter endmill and use the macro again for the finishing cut at –20 mm.

ADDITIONAL TASKS

Write a program which incorporates two types of canned cycle for hole generation (for example, for drilling and for boring).

Write a program which includes an area clearance cycle for machining a circular or square pocket.



▲ **Figure 8.1** Drilling and counterboring holes (Task 8.3)

Learning Assignment 9

Starting-up procedures for a CAD/CAM system

Most computer systems use similar hardware for data storage, such as floppy discs and a hard disc, with printers and/or plotters for output of hard copy. They may, however, use different ways to input information and data, such as moving a tracker ball or a mouse, or simply typing commands using the keyboard. Various facilities may also be available for transferring CNC programs to machine tools, such as a punched tape/reader or a magnetic tape cassette.

For operating CAD/CAM software the computer may be a stand-alone PC, a workstation or a network between computers and/or a mainframe machine. In all these arrangements, access may be gained to the CAM system by using passwords and/or 'pull down' menus.

When starting up on a CAD/CAM system, the drawing parameters may have to be set according to the size of the component, the component datum point, the grid size if required and the directory in which the data has to be stored.

There are also various settings for drawing functions: these could include the text height, the line thickness, colour and type (for example, chain or dashed), and the angle of text.

All CAD/CAM systems allow you to work on 'layers'; you can draw an outline on one layer, dimensions and text on another, construction lines on another. Using layers is an efficient way of organising the drawing information since layers can be 'turned off' or made invisible, thus allowing you to work on an uncluttered drawing.

TASKS

Task 9.1 Starting on a CAD/CAM system

Start up your CAD/CAM system and note the default values for the screen size, units, grid size, text height, snap value and the line type.

Figure 7.7 (page 32) shows a guide plate. Set up the following to draw the guide plate:

drawing size	A3	
units	mm	
number of decimal places	1	
scale	full size	
grid size	5 mm	
snap value	5 mm	
co-ordinate origin	X0.000	Y0.000

Set up one layer for the solid lines, another for the centre lines and a third for the dimension lines and text.

When your drawing is completed, save it.

A POINT TO NOTE

Some CAD systems require you to give the filename (drawing name) as part of the initial start-up procedure. Other systems prompt you for the drawing name when you save it for the first time.

Learning Assignment 10

Operating parameters for a CAM system

When you use a CAM system, you need to set up some additional parameters in the initial start-up procedure, over and above those required by a CAD system (discussed in Learning Assignment 9). These parameters relate to the ultimate generation of a CNC program to machine the drawn component and may include the following:

- the **data directory** – the directory for storing drawing profile and CNC data files; this allows data for different types of work, or for work for different customers, to be separated and stored in their own directories on disc
- the **machine directory** – the directory for storing post-processors, backplot files and tooling information; this allows data for different CNC machines to be stored in their own directory, such as Fanuc OM for milling, or Fanuc OT for turning
- the **tool library**, either milling tools or turning tools
- the **canned cycle call** – an option to use the canned cycle calls for the CNC machine tool set in the machine directory.

TASKS

Task 10.1 Starting on a CAM system

Select and step through the parameter settings in your CAD/CAM system and make a list of the settings suitable for the component shown in Figure 8.1 (page 36). It may be that your system will answer some or all of these questions because parameters may already be 'set up' as you progress through the system. Obviously, settings vary from one CAM system to another. Here, however, is a list of settings for Precision CAD/CAM software:

drawing size	A2
units	mm
number of decimal places	1
scale	1:1
co-ordinate origin	X0.000 Y0.000
number of decimal places	3
machine directory	Fanuc OM
data directory	Runcorn Engineering
tool library	milling tools
canned cycle calls	yes
importing drawing datas from a CAD system	DXF

Learning Assignment 11

Using a drawing package

The CAD front end of most CAD/CAM packages is usually a standard CAD system. Various graphical functions are used to draw circles, lines, arcs, points, tangents between circles, arcs through a point tangential to a line and so on. The main difference is in the way that you construct the geometry, since you only need to draw details like the profile shape or the hole positions. You do not have to incorporate features like hatching lines, dimensions or circles for holes into the drawing. You only have to specify the finished shape the tool has to produce, such as the profile shape, area clearance/pocket or points where holes have to be drilled, tapped or counterbored. Should you need to produce a fully dimensioned drawing to British Standard specifications, however, drawing features such as hatching and dimensioning are available on most CAD/CAM packages.

Additional information

Systems use a variety of different ways to access drawing commands. They may use a mouse or tracker ball for selecting from static or pull-down menus or icons, keyboard input, digitising pad or a combination of these methods.

Remember that line directions, and whether circles are drawn clockwise or counterclockwise, can be significant for the direction of the tool path generated by the CAM system. Also the mathematical data generated by the CAD system, like the line/arc start and finish positions, the intersect points of lines and arcs, and the number of decimal places, has to be in a similar format to that of the CAM system.

USEFUL OBSERVATION

Start up your CAD/CAM system and note the different ways in which geometry functions can be accessed to produce a drawing.

TASKS

Task 11.1 Functions in the root menu of a CAD system

Investigate the functions in the root menu of your CAD system, including the different ways in which points, arcs and lines can be drawn. (You can find information on this in the companion workbook *Computer-aided Draughting Workbook*, by Brian J. Townsend, published by City and Guilds/Macmillan.)

Task 11.2 Drawing exercises

Using your CAD system, construct the following drawing geometry:

1 A continuous line between the following points: P1 (X20 Y30), P2 (X50 Y45), P3 (X100 Y80), P4 (X100 Y–30), P5 (X–30 Y–30) and back to P1.
2 A line of 150 mm length at 35° from X0.0 Y0.0.
3 Three horizontal parallel lines 50 mm apart, each 100 mm long.
4 A circle passing through three points with co-ordinates X20 Y20, X35 Y35, X65 Y35.
5 A circle of 35 mm radius and a second circle of 50 mm radius such that the second circle touches the first circle at 0°.
6 Two circles of 80 mm diameter, centres 250 mm apart, with a horizontal tangent line passing between them.
7 A 10 mm chamfer on a corner at 45°.
8 Two vertical lines 30 mm apart, each 80 mm long. Join them with a horizontal line. Draw a 12 mm fillet radius at each corner.

9 A 40 mm fillet radius between two touching circles of 12 mm and 20 mm radius (both circle centres on the same horizontal centre line), so that the fillet radius intersects with the lowest quadrant of each circle.

10 A continuous curve (a spline, see Glossary) through the following points: P1 (X0 Y–5), P2 (X10 Y5), P3 (X20 Y–5), P4 (X30 Y5), P5 (X40 Y–5), P6 (X50 Y5).

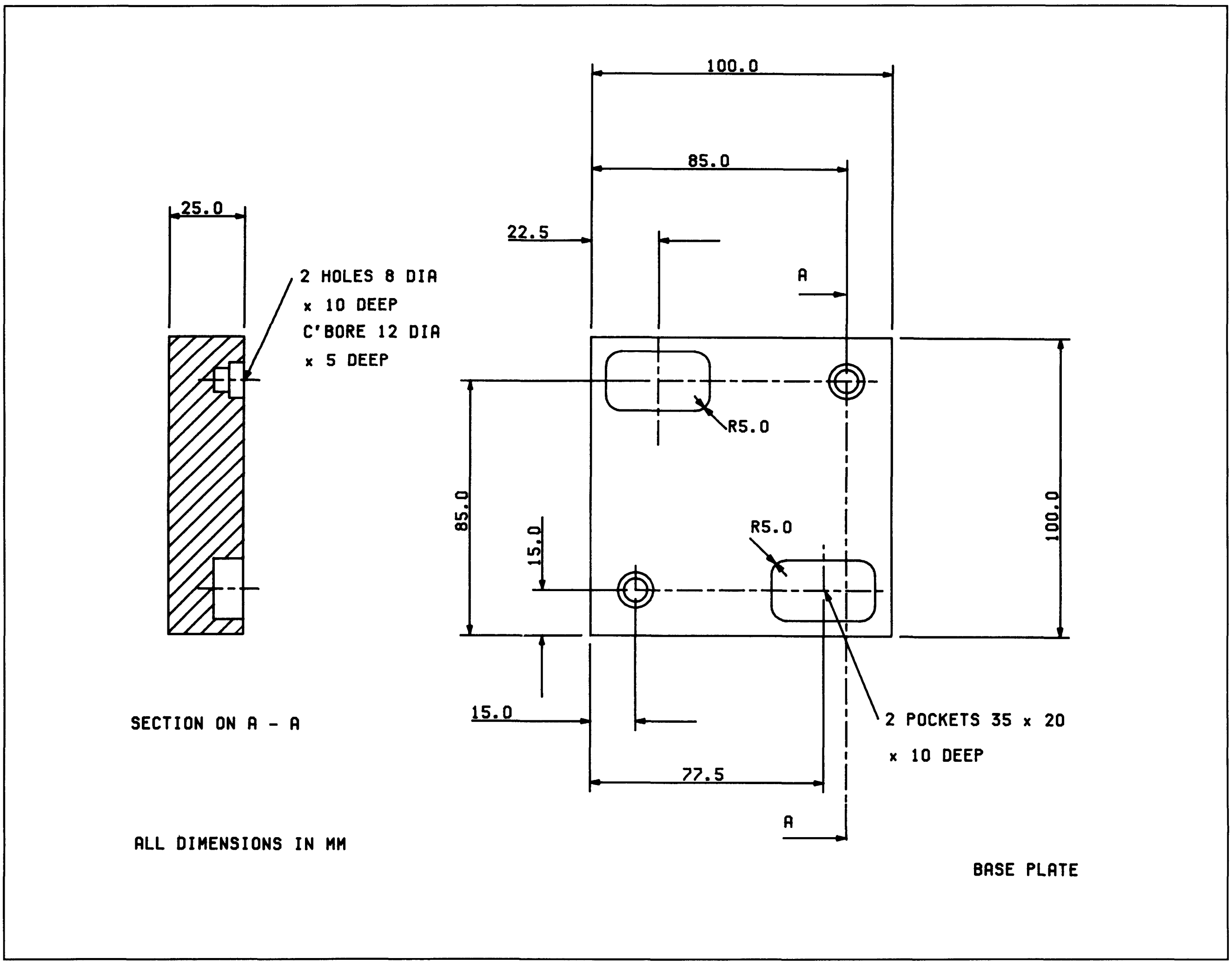
Task 11.3 Drawing a base plate

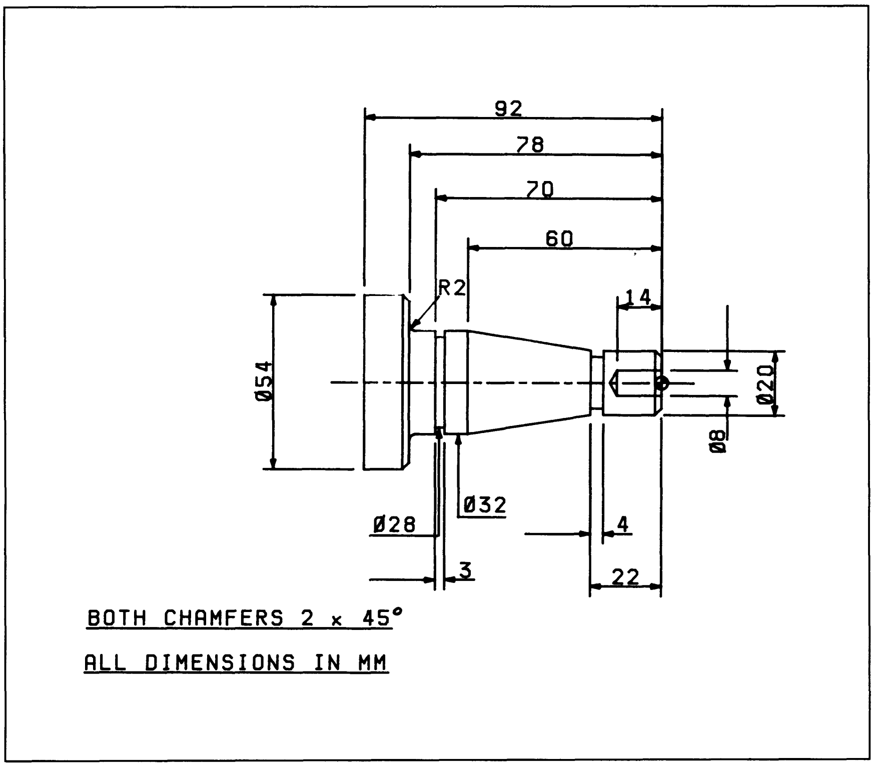
Figure 11.1 shows the drawing of a simple base plate with two holes drilled and counterbored, and two pockets. Reproduce the drawing to show both views (do not include the dimension lines and hatching). Make the origin point at the bottom left-hand corner of the base plate and include four major dimensions. When completed, erase one of the holes and one of the pockets and re-draw them using the move, copy and rotate commands from the existing hole and pocket.

ADDITIONAL TASK

A turned component is shown in Figure 11.2. Draw the component, including an end view. Note that the origin is at the right-hand end of the component, and that you need draw only the top or bottom half of the component in order to produce a CNC profile and a part program from the drawing data.

▼ **Figure 11.1** Drilling, counterboring and machining two pockets (Task 11.3)





◀ **Figure 11.2** Turned component

Using a CAM system to produce CNC part programs

The CAM section of the CAD/CAM package requires an input of geometrical data. This can be generated from the CAD front end of the package or imported from an external CAD system using IGES, DXF files or other suitable interfacing software. At this stage the important task is to determine the various toolpaths from the geometry of the drawing. To simplify this, delete or trim any construction lines and remove all dimension lines, hatching and so forth. Once you have defined the profile to be machined, select tools with appropriate feeds/speeds and program the cutter path around the desired profile. Store this information, and use it when required to produce a part program by post-processing into the relevant machine tool controller language.

In order to help you to become familiar with your CAM system, the following exercises have been designed to lead you through a gradual progression of CAM features. The project exercises at the end of this workbook incorporate most of the features that you will find in a commercial CAD/CAM system.

TASKS

Task 12.1 Machining a profile; drilling and counterboring holes

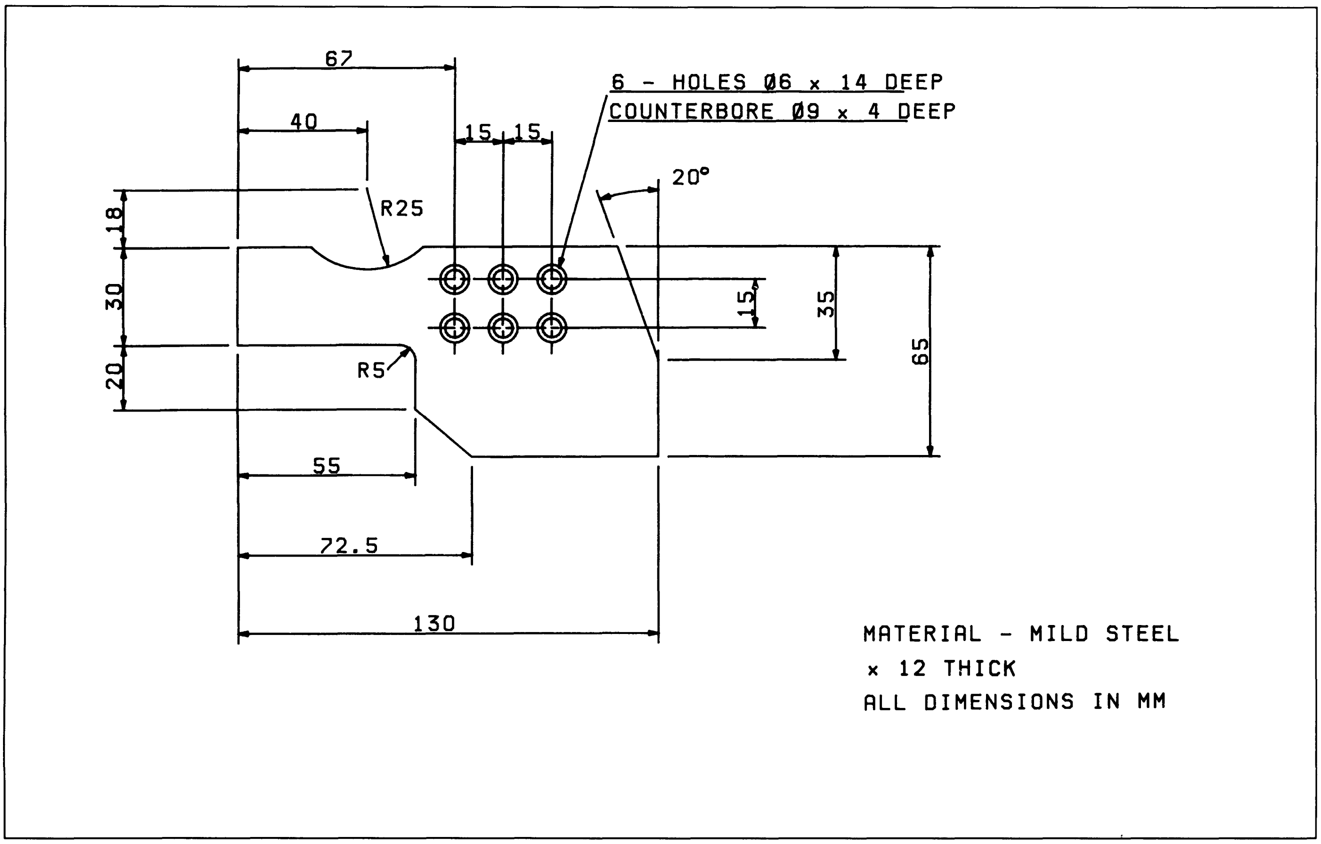
Figure 12.1 shows a flat plate that has to be profiled with the six holes drilled and counterbored. When machining this component it is to be held in a vice, and operation 1 consists of drilling and counterboring the holes. The component is then to be held in a fixture, the holes being used for location, and the profile machined as operation 2.

Use a CAD/CAM system to draw the profile and place points only where the holes are to be drilled. Note that in some CAM systems the machined profile is defined by moving the pointing device around the desired profile; others use a 'search for path' facility, and in some others the profile is defined manually by selecting lines, arcs and intersect points.

Select a suitable tool change position and use the tool library for tool selection, in this case a 6 mm diameter drill. Enter metric programming, absolute co-ordinates, feedrates, spindle speed, coolant on and clearance plane, and make a rapid move to a position above the first hole. Select a drilling canned cycle that you consider relevant and machine the six holes. Go to tool change, select the counterboring tool and the appropriate spindle speed, coolant on and feedrates, and counterbore the holes.

Operation 2 requires the profile to be machined. Bolt the component on to a fixture using two of the drilled holes. Select a 8 mm diameter slotdrill, spindle speed, coolant on and feedrates, and move to a position near the datum point. Plunge the cutter to depth and apply cutter radius compensation to the left-hand side of the profile. Repeat the toolpath for a finishing cut, lift the cutter above the work, cancel cutter compensation and return to the tool change position.

To verify that the machining sequence is correct, use the command from the menu to run a simulated toolpath and, if your system allows, view a simulation of the cutting paths in three dimensions. This has the advantage that Z values can be verified –



▲ **Figure 12.1** Machining a profile, drilling holes and counterboring (Task 12.1)

that is, that the tool has cleared the work when returning, for example, to the tool change position.

To view the data that has been generated by the software, go into the text editor. Familiarise yourself with the strings of instructions shown. An example is given below.

```
[Profile (BH1) data]
Units:mm        Scale: 1:1   Rad:8.000
Cutter start point X:–50.000  Y:–50.000
Operation :     MATERIAL [MATERIAL=ALUMINIUM]
Operation :     TCLIB [T=1 NAME=SLOTDRILL  T0=9
                ZC=50 RAD=8 MATERIAL=ALUMINIUM
                F=200  S=2000]
Cutter move point X:20.000  Y:-20.000
Operation :     ZFEED [ZE=100]
Operation :     COMP [COMP=1]
Approach        from    X:–20.000     Y:–20.000
                to      X:0.000       Y:0.000
Linear move     from    X:0.000       Y:0.000
                to      X:0.000       Y:30.000
Linear move     from    X:0.000       Y:30.000
                to      X:29.000      Y:30.000
Clockwise arc           X:29.000      Y:30.000
                        XC:40.000     YC:49.000       R:25.000
                to      X:49.000      Y:30.000
Linear move     from    X:49.000      Y:30.000
                to      X:123.667     Y:30.000
                                              etc.
```

This is a worthwhile exercise, since if you have made a mistake or need to alter something, it is relatively easy to change it now.

The last operation is to post-process your program into the machine tool language. The method of doing this depends on the CAD/CAM system available. If you have to name the post-processor at the beginning of the program, it will automatically output in that language. In some cases, however, the post-processor file can be chosen through a menu system at the end of the CAD/CAM sequence. Once the program has been post-processed, enter the text editor and familiarise yourself with the final output.

For your records, make a hard copy of the CNC program and the profile data (the toolpath co-ordinates) and a print-out of all the tool movements and cycles that you have used.

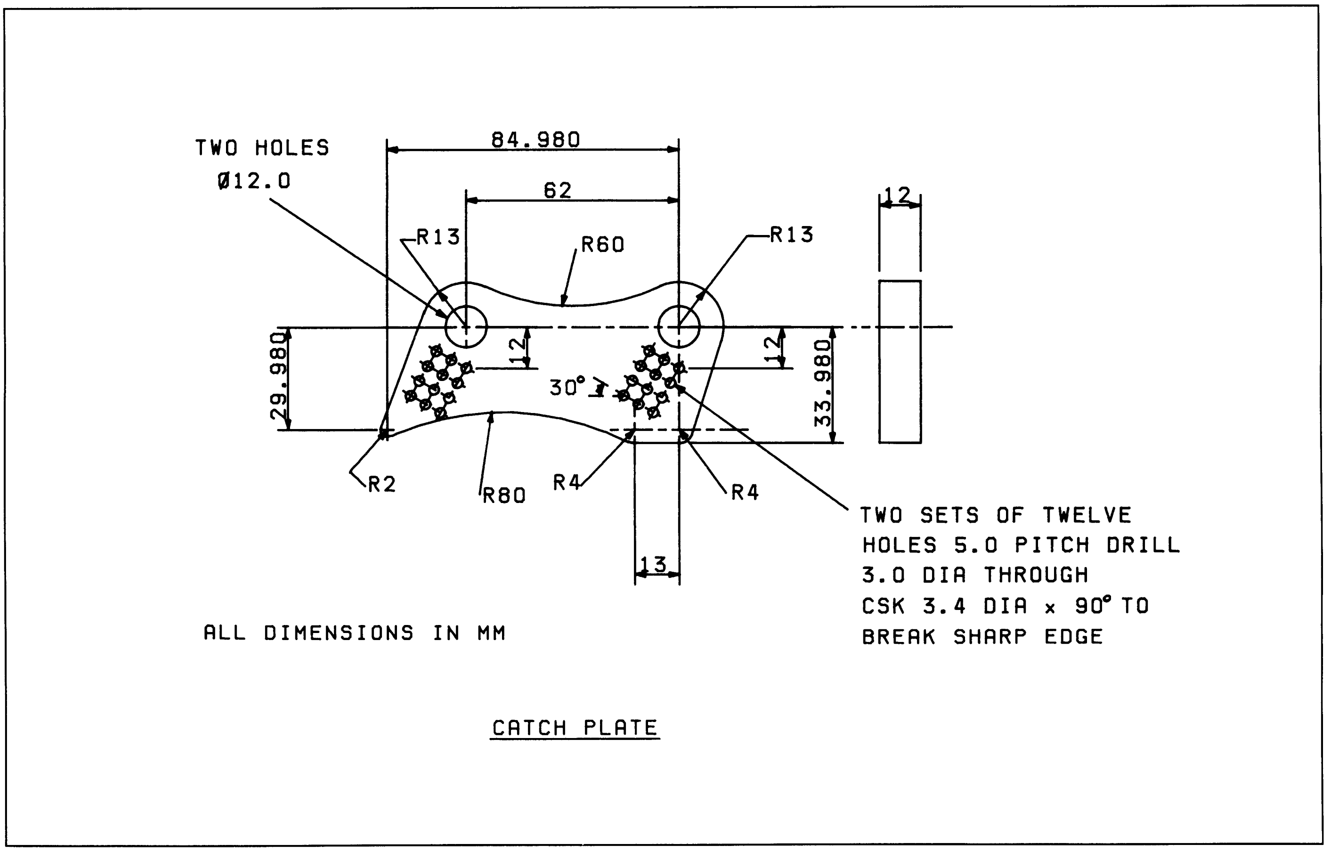
Task 12.2 Drilling holes in a grid

Repeat the procedure used in Task 12.1 for the component shown in Figure 12.2 which shows two sets of holes, each set forming a grid at 30° to the horizontal. Each set of holes is to be drilled 3 mm diameter and countersunk 3.4 mm diameter.

On a CAD/CAM system, use the commands for drilling holes in a grid. Generally the software will ask you to enter the position of the datum hole and the number of holes in the X direction, together with the pitch and angle. Repeat these commands for the holes in the Y direction. On entering this information you can call up an appropriate drilling canned cycle.

After drilling the 3 mm diameter holes, use the same drill to make pilot holes for the two 12 mm diameter holes.

▼ **Figure 12.2** Drilling holes in a grid (Task 12.1)



Learning Assignment 13

Machining pockets

The procedure for machining pockets of any shape, with or without 'islands', depends on the CAD/CAM software that you are using. The software may operate on a pull-down menu system which requires the name of the previously defined pocket shape and the name of each 'island' within the pocket. Or a pocket clearance routine within the software may be used to define the shape of the pocket, including 'islands'. In either case, the data entered is used to generate the cutter toolpaths.

TASKS

Task 13.1 Machining a profile and a circular pocket

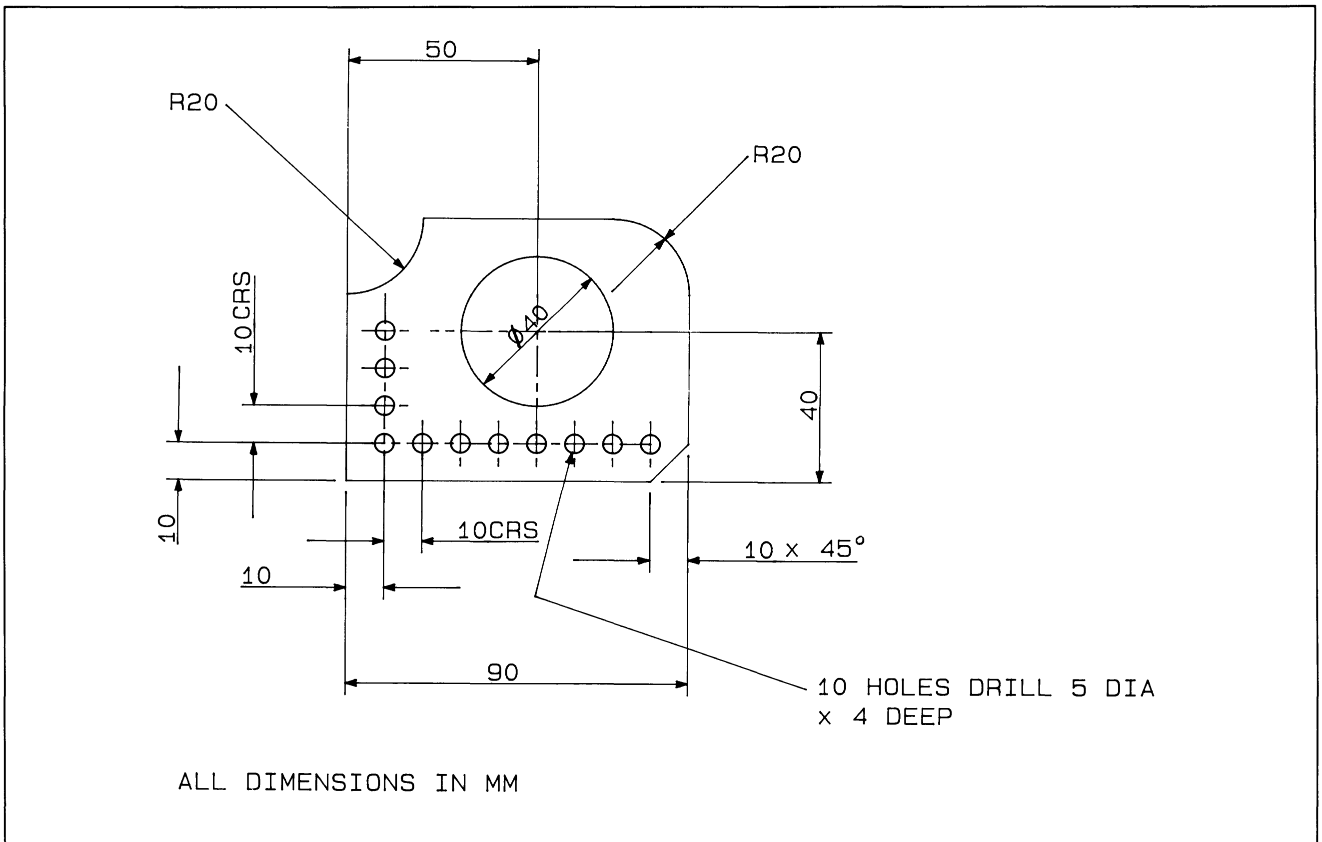
This exercise (Figure 13.1) involves machining a profile and drilling a series of holes, which you should be able to do if you have completed the previous exercises. The feature in this exercise is the machining of the 40 mm diameter circular pocket.

Draw the profile, the circular pocket and the hole positions. Produce the toolpaths for machining the outside profile and for drilling the holes.

Additional information

When you have defined the tool (perhaps a slotdrill of 10 mm diameter) and the feedrates, spindle speeds and so forth, call up the pocket routine from the menu system. This will ask you for information, which will probably include the start position where the tool is plunged into the

▼ **Figure 13.1** Machining a profile and drilling holes (Task 13.1)



work, the depth of the pocket, the value of pecking in the Z plane, the clearance plane, the machining angle and whether a finishing cut is required. All systems will ask you to identify the profile of the pocket and any 'islands' inside the pocket. (This is done by labelling or identifying predefined profile shapes – using, for example, Kurves as in the PEPS system – or defining these shapes by using a mouse or tracker ball.)

When you have completed the exercise, save to disc and produce hard copies of the CNC program, the toolpaths and so on as in the previous exercise.

Machining pockets using the 'freehand' milling command

Sometimes it is not practical to use a pocket routine, particularly if there are many 'islands' within the pocket. The reason is that, in some CAM systems, toolpath data is generated to lift the tool up from the work, move across the island at the clearance plane height, and plunge back down into the work. For some tasks, a large cutter may be needed to rough out the pocket, with a smaller cutter being used for the finishing cut. It is in cases like these that freehand milling is often used.

When the freehand milling command is used, the tool is shown on the screen and it can be dragged around the pocket, by using some pointing device (similar to the 'elastic band' command in a CAD system which 'stretches' lines or arcs). When the tool has been moved to a suitable position, the position is entered and the toolpath is shown as two parallel lines, the distance between them being equal to the diameter of the cutter.

TASKS

Task 13.2 Using the 'freehand' milling command

Using your CAD/CAM system, construct a square of side 200 mm. Add 30 mm fillet radii to the corners and draw a circle of 50 mm dia at the centre of the square. The square represents an outer pocket 10 mm deep and the circle an 'island' 10 mm high in the centre of the pocket. Use a 25 mm diameter slotdrill and the 'freehand' milling routine to rough out the pocket without lifting the tool above the island. If your system does not show the toolpath as the outside diameter of the tool (that is, if the tool centre line only is indicated), be careful that small areas of metal are not left unmachined. Change the tool to one of 10 mm diameter and complete the finishing cuts on the outer wall of the pocket and the island. Climbmill, that is, machine counterclockwise on the outer pocket profile and clockwise around the island.

Task 13.3 Machining a pocket with an island

Repeat this exercise for a pocket of depth 20 mm and an island height of 10 mm.

Machining slots

The component in this exercise, shown in Figure 14.1, has six slots 6 mm wide × 10 mm deep to be machined in addition to the outer profile. When drawing the component show the slots as single lines, 25 mm long (that is, the centre line of the slots will represent the line that the centre of the cutter will take). Select the tool change position and define the toolpath for machining the outside profile as in previous exercises, using a 12 mm dia cutter.

The six slots can be machined with a 6 mm slotdrill using either absolute or incremental programming. In absolute programming, the procedure would be as follows:

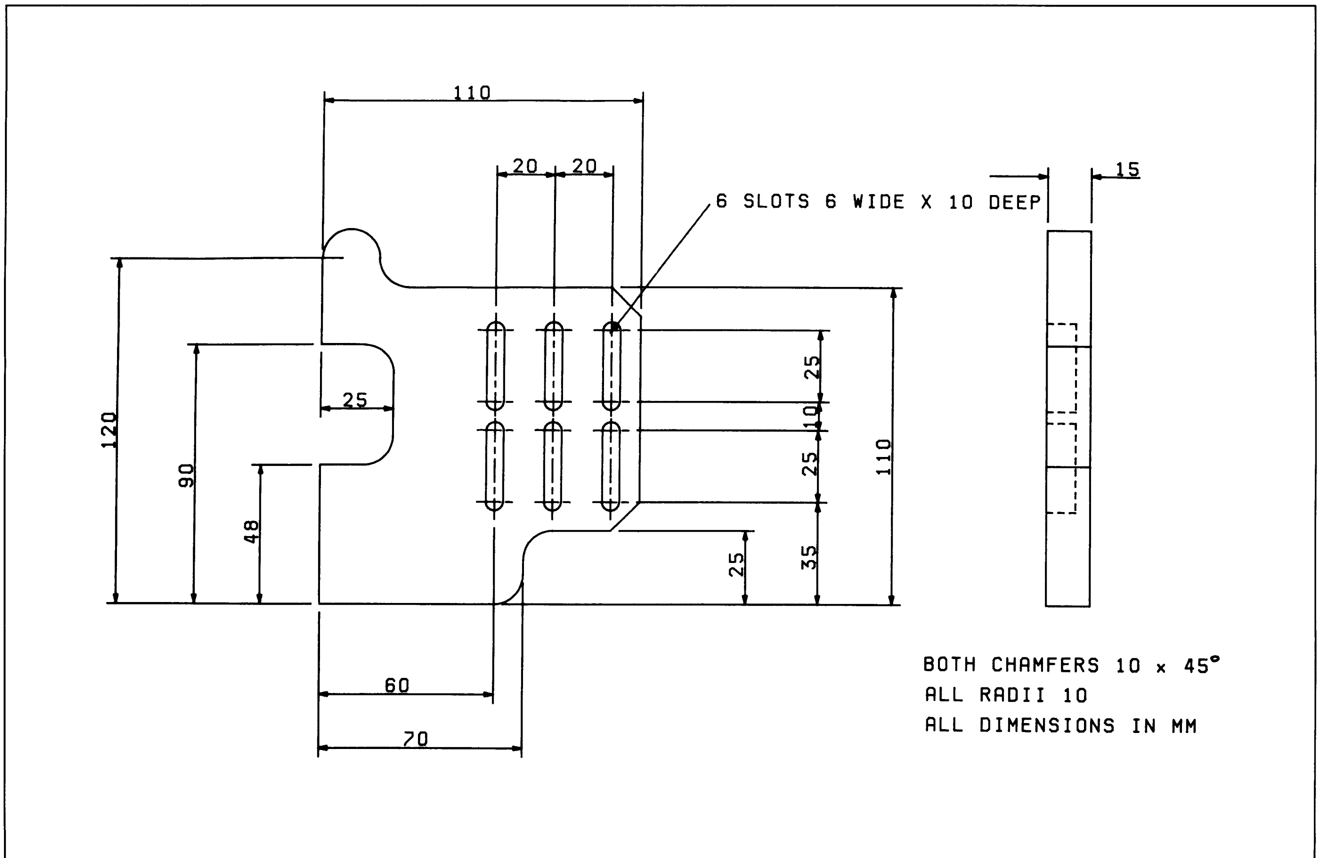
- take the cutter to X60 Y95 and plunge in to a depth of 10 mm, with a feedrate
- move the cutter to X60 Y70 and lift to the clearance plane
- move to X60 Y60 and plunge into the work 10 mm deep
- move to X60 Y35, lift to the clearance plane
- move to X80 Y35, plunge down to depth again ...

... and so on. This can be quite tedious, as each one of the slots has to be machined as a separate entity.

Since the dimensions for the slots are given incrementally, however, it is easier to program in this mode. Use the following steps:

- call up the incremental command within the CAM system
- enter the tool positions as X0 Y0, which is the beginning of the slot (X60 Y95 in absolute)
- plunge to depth, then move to Y–25

▼ **Figure 14.1** Machining a profile and slots



- lift and move to Y–10
- plunge to depth and move to Y–25
- lift and move to X80
- repeat the process, but this time see that the Y movements are positive.

Be careful with the Z movements as, in incremental programming, the last X, Y and Z positions become 0 for the next move, even though the Z position may be inside the work.

Additional information

A much quicker method of machining slots is to program using sub-programs or macro routines (page 34). Macro programs are available in most up-to-date CNC control systems and they enable the programmer to reduce the repeating or re-programming of similar machine moves.

Examples of operations where macro programming could be used would include the drilling of holes on a PCD, or the drilling and tapping of a number of holes in a plate. The hole positions can be stored as a macro which is called up if drilling or tapping cycles are to be executed.

Machining slots by using a macro

If a macro program is to be used in the example above, only one slot needs to be defined. Program the tool to move to X40 Y95 (20 mm to the left of the first slot) and to X3.000 (the clearance plane). Access macro program through the CAM system and give it a number, such as O123. Select G91 (or incremental mode) on your system and enter the following tool paths;

- at X0 enter into the work Z–13 with a vertical feedrate
- move to Y–25 with a horizontal feedrate
- rapid move to Z13 (3 mm above the work).

Then store these moves as a macro with a macroend command.

Although this defines the macro, a separate command needs to be executed in order to generate tool movement. A typical call would be macro call 123, repeat three times (P3123). This would execute the machining of the three horizontal slots. To move down to the three lower slots, use an absolute move of X40 Y60 to bring the cutter in a position to repeat the macro again three times.

You can see from this example that using the macro facility reduces not only the programming time but also the length of the CNC part program.

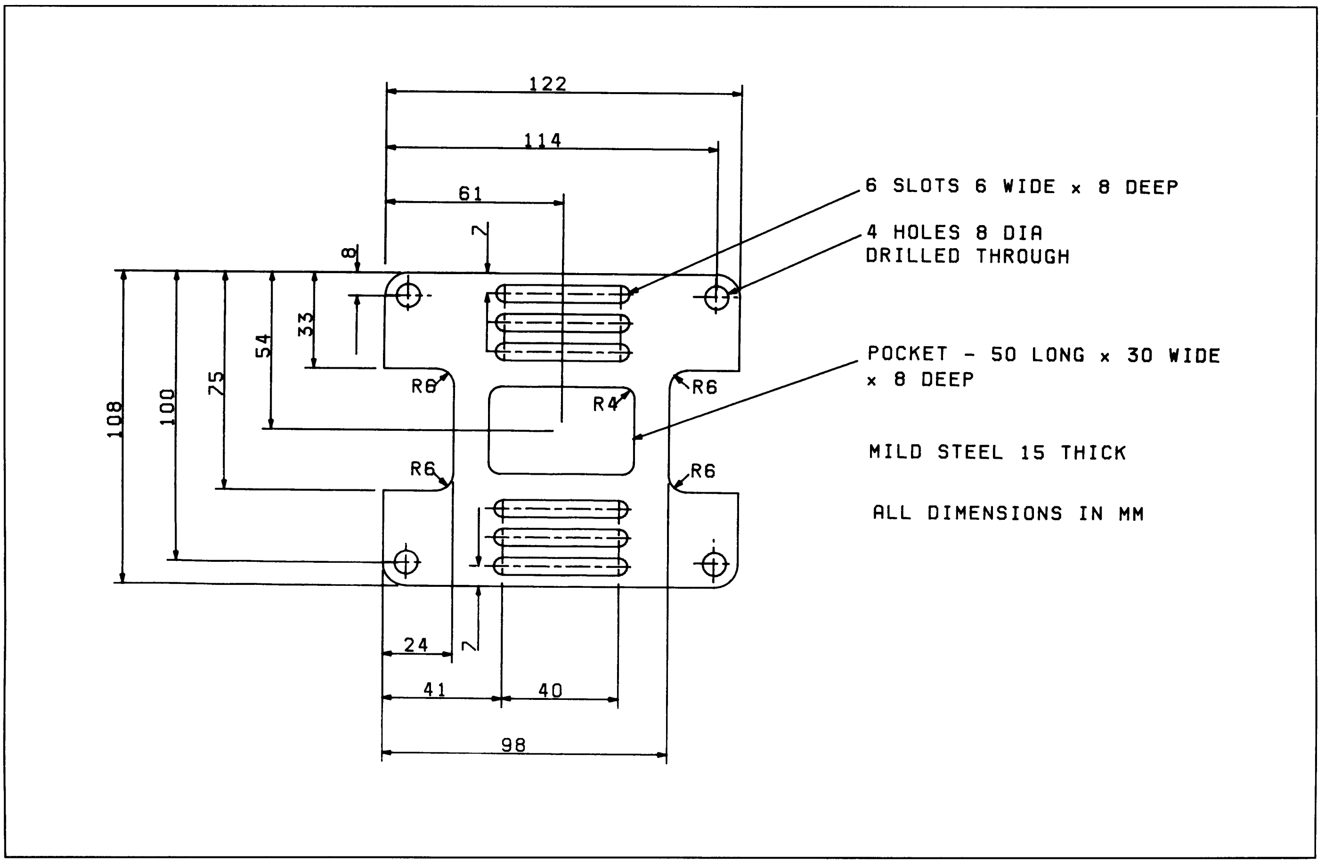
TASKS

Task 14.1 Machining slots using a CAM system

Program the slots in the component shown in Figure 14.1 using incremental programming. Then re-program using subroutines and macro programs. Produce a print-out of the CNC part program and the toolpaths.

ADDITIONAL TASK

Use the experience you have gained from the previous examples to produce a CNC program for machining the outer profile, the rectangular pocket and the six horizontal slots shown in Figure 14.2, using the macro method for machining the slots.



▲ **Figure 14.2** Machining slots using a macro

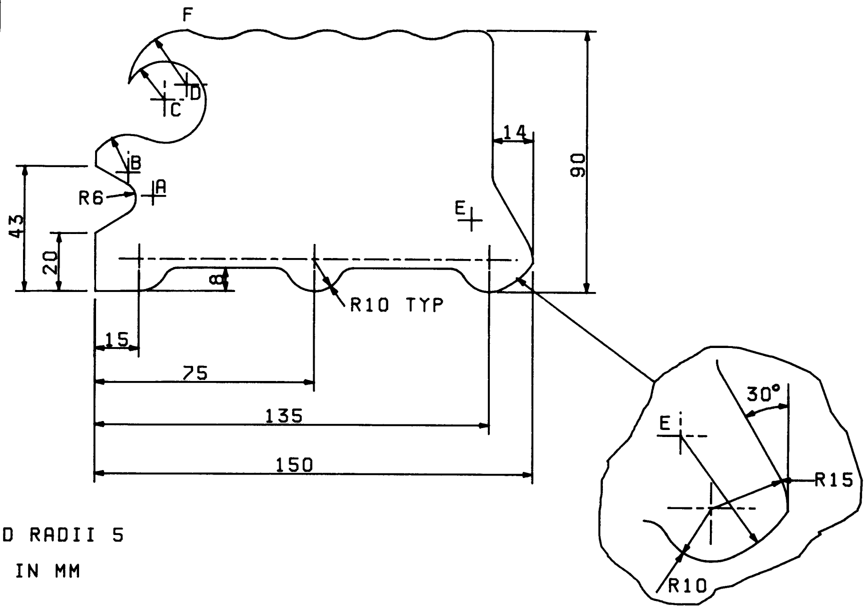
Learning Assignment 15

Complex profile with a spline

POSITION	X	Y
A	19.5	32
B	11.25	40
C	23.5	65
D	31.0	70
E	129	24
F	31.0	90

SPLINE POINTS FROM F

X	Y
12	-3
24	0
36	-3
48	0
60	-3
72	0
84	-3
96	0



ALL UNSPECIFIED RADII 5

ALL DIMENSIONS IN MM

▲ **Figure 15.1** Complex profile with a spline

The profile shown in Figure 15.1 is quite complex and for clarity's sake letters are used to designate certain points on the profile. Their X and Y co-ordinates are tabulated at the top left-hand corner of the drawing, thus avoiding a mass of dimension lines. A spline (a series of points through which a smooth curve is drawn) is shown along the top edge of the component. Consult your manual and research the commands that your CAD/CAM system uses for drawing splines.

TASKS

Task 15.1 Drawing a spline

To draw a spline using your CAD system, enter the following points:

(X20 Y7), (X34 Y14), (X46 Y25), (X58 Y10), (X70 Y20), (X86 Y40), (X100 Y70), (X115 Y98).

Use the appropriate CAD commands to join them with a spline.

Task 15.2 Machining a complex profile

In Figure 15.1, the radius at point C is 14 mm. You will have noticed that the radii at points B and D are missing. This is deliberate! When drawing the profile, find out which commands within the CAD/CAM system can be used to ascertain these values. (*Hint:* a line can be drawn between the two circle centres. Enquire to the length of the line using the system and subtract the given radius from it.)

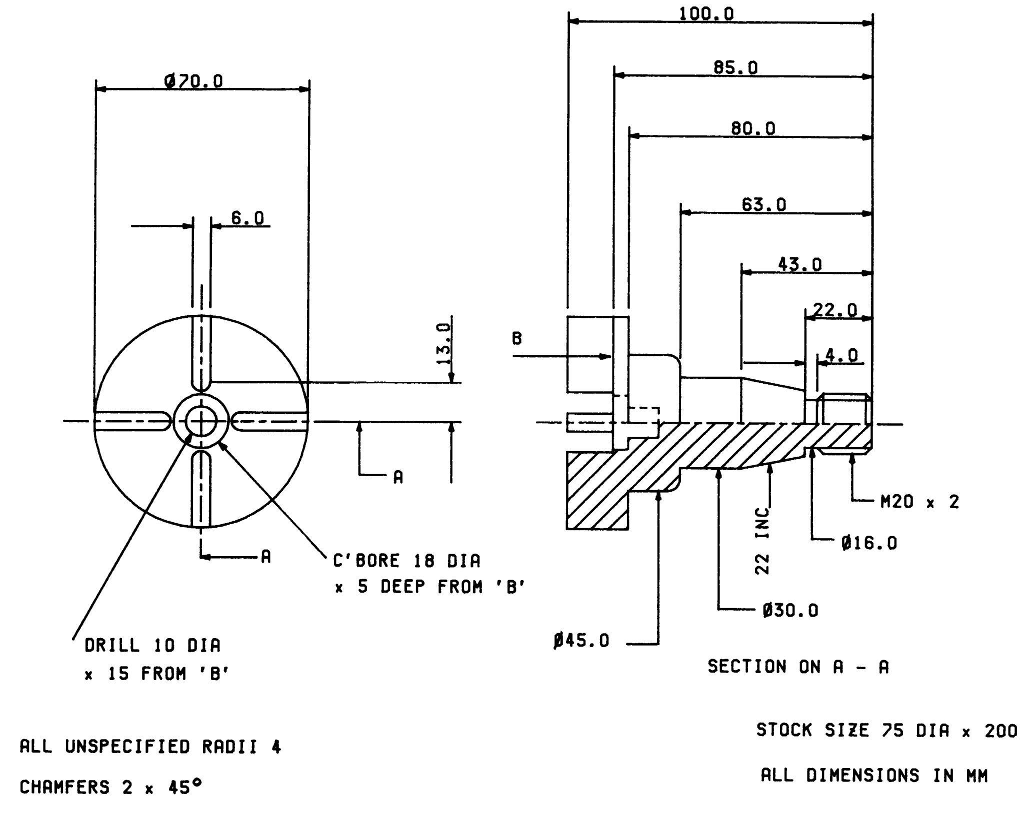
Draw the outer profile and, as for the previous exercises, produce print-outs of the toolpaths, the CNC part program and a complete tooling list.

Projects

If you have worked systematically through the assignments in this workbook, you should now be able to tackle the following milling and turning projects. It is suggested that for each project you should follow the procedure below.

1 Re-dimension the component drawing in absolute format, selecting the datum point that you consider most applicable.
2 Design a suitable fixture (or fixtures) and produce a fixture assembly drawing. Produce the engineering drawings for manufacturing purposes.
3 Produce drawings showing work-holding layouts for both turning and milling operations.
4 Prepare operation sheets showing the sequence of machining for both the milling and turning operations. These should include a comprehensive tooling list.
5 For each component, produce a drawing showing the intersection points with tabulated co-ordinates, dimension values and cutter paths.
6 Complete calculation sheets for points of intersection and the calculations for all feeds and speeds used.
7 List the part programs, including any relevant remarks alongside the appropriate blocks in the program. This applies to all the part programs, whether for turning, milling, drilling or tapping.

▼ **Figure P1** Project 1



8 Save the completed part programs on paper tape, or on a floppy or hard disc, for future reference.
9 Load the programs into a machine, prove out the programs and produce machined components.
10 Make a record of any editing changes or alterations within the programs, changes to the fixtures, toolpaths and so on.

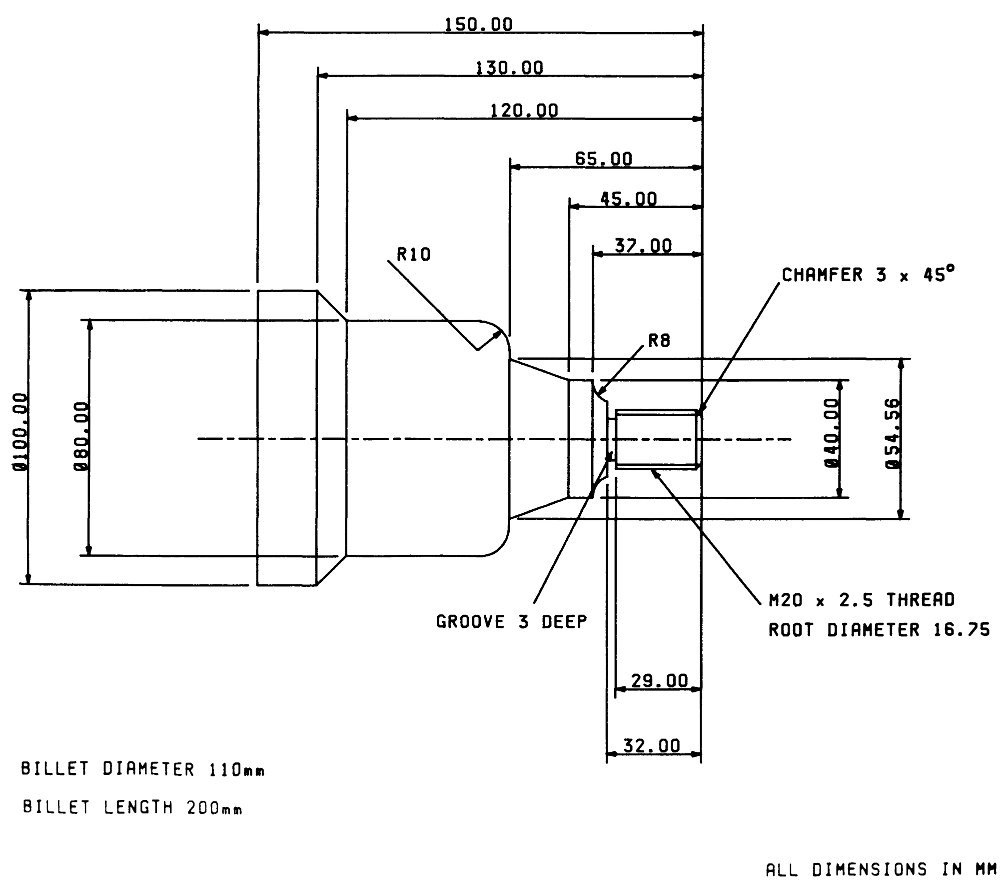
Project 1 Shaft with a thread and milled slots

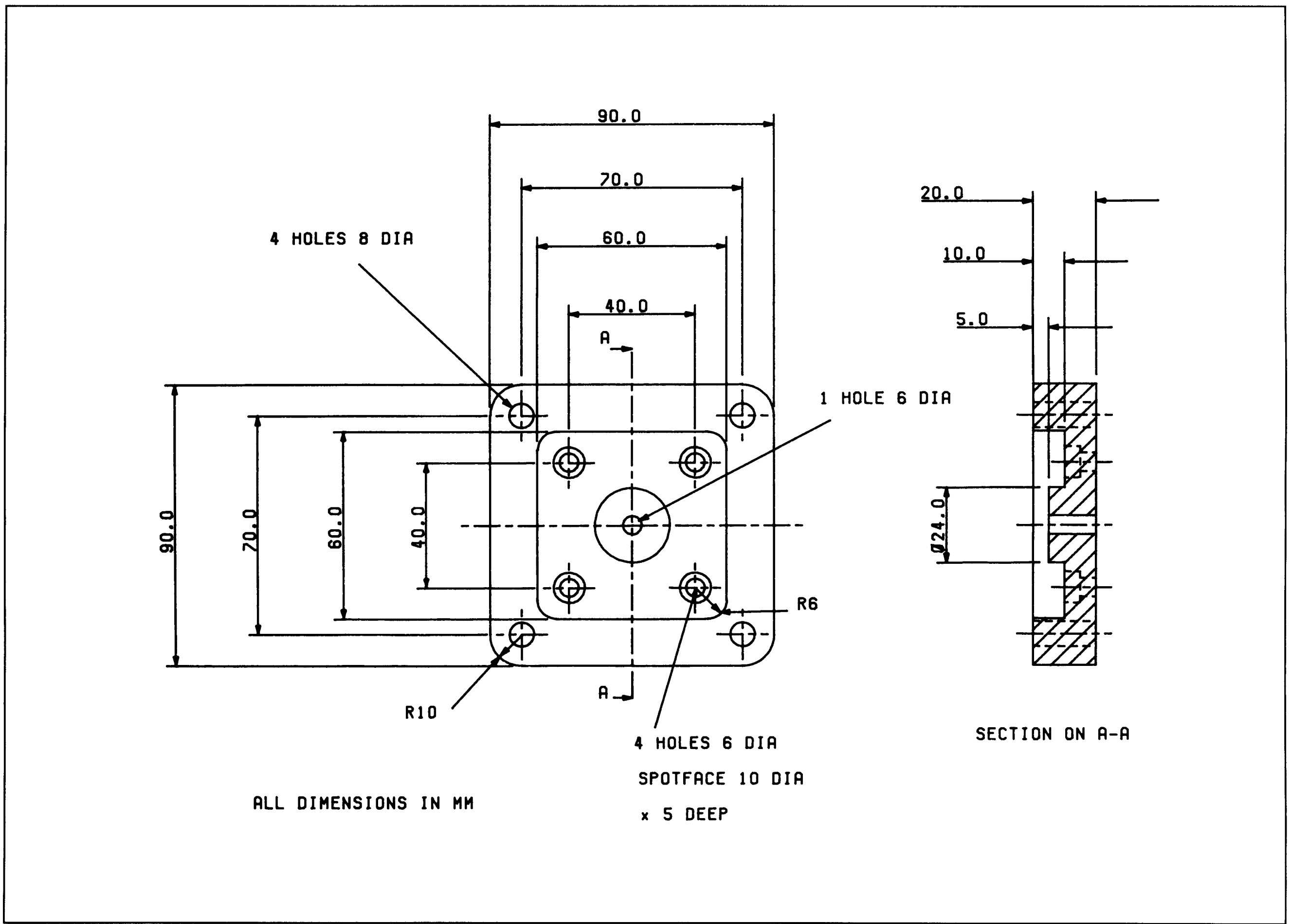
This project involves both milling and turning operations. The bar is turned down to the relevant dimensions and threaded. The slots and counterbored holes are then machined on a CNC milling machine. (*Hint:* When the workpiece is in the chuck, drill a pilot hole for the 10 mm diameter hole. This hole can then be used for setting up the work datum point for the milling operations.)

Project 2 A hub with a thread and taper

Rough-turn the diameters and taper, leaving 1 mm on for the finishing cut. Finish-turn the groove and thread, and machine the finishing cut in one pass of the cutting tool. (*Hints:* Check whether your CNC turning machine uses I and K or R for generating arcs. To ensure secure work holding, hold on the 80 mm diameter when parting off.)

▼ **Figure P2** Project 2





▲ **Project P3** Project 3

Project 3 Cover plate with machined pocket and 'island'

Hold the plate in a vice and drill the four 8 mm diameter holes. Bolt the workpiece in a fixture using the four holes for securing and location, and machine the pocket and profile. Drill and counterbore the four holes in the pocket. (*Hint:* When machining the pocket, use a pocket-milling cycle and machine 5 mm deep. Then machine the pocket and 'island' 10 mm deep.)

Projects using the CAD/CAM software

The manual program preparation projects can also be processed using the CAD/CAM system. Your documentation should include:

- printer listings of the geometry prior to producing the actual profile definition
- printer listings of the profile curve(s) with their respective numbers and names
- print-out of the drawing commands, toolpath commands, tooling and machine function calls and the canned cycles used
- a copy of the tooling 'set-up' sheets produced by the computer, together with the part program listing
- a print-out/screen dump showing the various stages of producing the profile and machining data, culminating in a three-dimensional image of the machine toolpaths and views shown in the plan and end elevations.

For your final assessment for CAD/CAM programming, produce the complete documentation and part program for Figure 14.2.

Glossary

Absolute programming Programming co-ordinates from a fixed datum

Backplotting The verification of a CNC program by displaying the toolpaths on either a plotter or a graphics display

Baud rate The rate of transmitting data, expressed in terms of bits per second

Block number A number identifying the position of a block of NC data, e.g. N006 G41 H24

CAM (computer-aided manufacture) The use of a computer to generate toolpath cutting data and hence CNC part programs

Canned cycle A single statement called from the CNC control to expedite a sequence of tool movements (such as a peck drilling cycle) that is repeated at specific co-ordinates until cancelled

Climbmill The milling cutter advances in the work in the same direction as the feed

CNC (computer numerical control) A microcomputer-controlled NC (numerical control) machine that controls the movements and operating functions of a tool by coded numerical data; has the facilities for storing programs, editing and in some cases displaying 3D toolpath graphics

Conversational format/programming A system that checks the validity of the program entered into the CNC controller and informs the operator/programmer if the next statement can be entered or if a mistake must be corrected

Co-ordinate Dimensional data which defines a location; it can be either absolute or relative

Cutter path The path taken by a cutter to generate a component of the desired shape

Cutter radius compensation On receipt of a G code, the cutter is moved to the left or right of the programmed co-ordinates by the length of the cutter radius or the tool nose radius

Cutting speed The surface speed of the component relative to the cutting edge of the tool

Datum A reference point from which all co-ordinates are measured; it can be set in the machine parameter settings (the machine table datum) or it can be movable (the component datum)

DXF Format for allowing CAD drawings to be exchanged to and from CAD systems or CAM systems

Feedrate or feed The travel of the tool per revolution of the workpiece in turning; the linear velocity (per minute) of the table or spindle in milling

Grid A network of equally spaced points forming squares, which is displayed on the computer screen; the start and end points of lines, centres of circles etc. can be snapped on to these points

Hardware Physical equipment; for a CAM system this could be a computer, plotter, printer, punched tape reader

IGES Initial Graphics Exchange Standard – a format for transferring drawings from one CAD system to another, or to a CAM system

Incremental programming Programming co-ordinates relative to the last position, that is, the present tool position is 0.

Machining cycle A series of moves with one or more cutting tools to produce a machined part

Macro A separate program for executing a series of machining moves that may be called up within the main program; a macro may be at the end of the main program or stored as a separate program within the machine control

MDI Manual data input of information into the CNC machine control

Mouse An input device for a computer; it defines the position of the cursor on the screen

Parameter A value, usually in bit form, that remains constant in the CNC controller until changed

Part program Specific and complete set of data and instructions to program a CNC machine tool to machine a part/component

Peck drilling In this technique the drill enters the work to the peck depth specified and then returns to a point at a specified distance above the work: this is repeated until the drill reaches the final depth of the hole (that is, the second peck distance is added on to the first peck distance, and so on); peck drilling enables the swarf to leave the hole

Pocket A term used in milling to describe a feature inside a profile that has a base, other than slots, blind holes etc.

Post-processor A complete program that converts the computer output data of a CAM system (the tool cutter path) into the CNC code for a particular machine tool

Quill The hollow spindle into which a tool holder is fitted on a milling machine

Register An area in the machine tool control that stores information and processes it in various ways

Single block mode When in this mode (for example, when proving a program) the CNC machine executes one program block at a time; to begin the next block the operator has to press the start button

Slotdrill A milling tool that can machine in the Z direction as well as in the X and Y directions

Software Computer programs for executing certain tasks, including creating a drawing, generating tool cutter paths, post-processing, DNC files to enable loading of NC programs from a computer to the machine control, etc.

Spline Either a smooth curve passing through a series of points, or a smooth curve that touches tangentially straight lines joining a series of points

Tool insert A carbide tip of various geometrical shapes, held in the tool holder

Tool length compensation Automatic compensation for the length difference between the tool to be used and datum tool; if the datum tool is the longer compensation is negative, and if it is shorter compensation is positive

Turret A tool-holding device used on turning machines, which rotates from one tool station to another

Tracker ball A ball which is rotated by an operator to move the cursor on the display screen

Word address format A program in which each word in a program block is identified by a letter, e.g. G90, M13

Index