INTRODUCTION TO CAM

Computer-aided manufacturing (CAM) is the use of computer-based software tools that assist engineers and machinists in manufacturing or prototyping product components. Its primary purpose is to create a faster production process and components with more precise dimensions and material consistency, which in some cases, uses only the required amount of raw material (thus minimizing waste), while simultaneously reducing energy consumption. CAM is a programming tool that makes it possible to manufacture physical models using computer-aided design (CAD) programs. CAM creates real life versions of components designed within a software package

CNC Technology:

Numerical Control (NC) is a software-based machine tool control technique developed at Massachusetts Institute of Technology (MIT) in early 1960s. It has now evolved into a mature technology known as Computer Numerical Control (CNC). Although major applications of CNC even today continue to be in machining, it finds applications in other processes such as sheet metal working, non-traditional machining and inspection. Robots and Rapid Prototyping machines are also CNC controlled. In fact, any process that can be visualized as a sequence of motions and switching functions can be controlled by CNC. These motions and switching functions are input in the form of alphanumeric instructions. CNC is the basis of flexible automation which helps industries cut down time-to-market and enables launch of even low volume products. Unlimited muscle power, unmanned operation, independent axes coordinated through software, simplified generic tooling even for the most complex jobs and accurate construction are some of the salient features of CNC.

CNC Machining:

Automats and Special Purpose Machines (SPMs) require special cams/ templates and clutch settings for each part. Manufacture of these cams/ templates is costly and slow. Furthermore, changing over from one part to the other on these machines also consumes considerable time. The high cost and long time of these hard automated machines to produce parts can be justified only in mass production. With the advent of fast, rigid and accurate CNC machines and sophisticated CAM packages, generation of NC programs and change over from one product to the other are easy and fast as it does not require any mechanical change. These in conjunction with advanced cutting tools have made High Speed Cutting (HSC) of hard materials a reality. Therefore, CNC machining has become a standard means to produce dies and molds; tool makers today require EDM only for producing inaccessible regions, sharp corners, tiny features and desired surface quality. Intricate aerospace parts are realized through 5 axis CNC machining. Internet technology in a global village enables designing in one place, NC programming and verification in another place and actual machining in yet another place.

Advantages of CNC

- Flexibility
- Accuracy
- Speed
- Simplified fixturing and generic cutting tools
- Storage of machining skill in NC programs
- Less skilled operators will do
- Less fatigue to the operators

G-codes:

G-Code, or preparatory code or function, are functions in the Numerical control programming language. The G-codes are the codes that position the tool and do the actual work, as opposed to M-codes, that manages the machine; T for tool-related codes. S and F are tool-Speed and tool-Feed, and finally D-codes for tool compensation. The programming language of Numerical Control (NC) is sometimes informally called G-code. But in actuality, G-codes are only a part of the NC-programming language that controls NC and CNC machine tools. A basic list of `G' operation codes is given below. This direct motion of the tool.

1. G00 - Rapid move (not cutting)
2. G01 - Linear move
3. G02 - Clockwise circular motion
4. G03 - Counterclockwise circular motion
5. G04 - Dwell
6. G05 - Pause (for operator intervention)
7. G08 - Acceleration
8. G09 - Deceleration
9. G17 - x-y plane for circular interpolation
10. G18 - z-x plane for circular interpolation
11. G19 - y-z plane for circular interpolation
12. G20 - turning cycle or inch data specification
13. G21 - thread cutting cycle or metric data specification
14. G24 - face turning cycle
15. G25 - wait for input #1 to go low (Prolight Mill)
16. G26 - wait for input #1 to go high (Prolight Mill)
17. G28 - return to reference point
18. G29 - return from reference point
19. G31 - Stop on input (INROB1 is high) (Prolight Mill)
20. G33-35 - thread cutting functions (Emco Lathe)
21. G35 - wait for input #2 to go low (Prolight Mill)
22. G36 - wait for input #2 to go high (Prolight Mill)
23. G40 - cutter compensation cancel
24. G41 - cutter compensation to the left
25. G42 - cutter compensation to the right
26. G43 - tool length compensation, positive
27. G44 - tool length compensation, negative

28. G50 - Preset position
29. G70 - set inch based units or finishing cycle
30. G71 - set metric units or stock removal
31. G72 - indicate finishing cycle (EMCO Lathe)
32. G72 - 3D circular interpolation clockwise (Prolight Mill)
33. G73 - turning cycle contour (EMCO Lathe)
34. G73 - 3D circular interpolation counter clockwise (Prolight Mill)
35. G74 - facing cycle contour (Emco Lathe)
36. G74.1 - disable 360 deg arcs (Prolight Mill)
37. G75 - pattern repeating (Emco Lathe)
38. G75.1 - enable 360 degree arcs (Prolight Mill)
39. G76 - deep hole drilling, cut cycle in z-axis
40. G77 - cut-in cycle in x-axis
41. G78 - multiple threading cycle
42. G80 - fixed cycle cancel
43. G81-89 - fixed cycles specified by machine tool manufacturers
44. G81 - drilling cycle (Prolight Mill)
45. G82 - straight drilling cycle with dwell (Prolight Mill)
46. G83 - drilling cycle (EMCO Lathe)
47. G83 - peck drilling cycle (Prolight Mill)
48. G84 - taping cycle (EMCO Lathe)
49. G85 - reaming cycle (EMCO Lathe)
50. G85 - boring cycle (Prolight mill)
51. G86 - boring with spindle off and dwell cycle (Prolight Mill)
52. G89 - boring cycle with dwell (Prolight Mill)
53. G90 - absolute dimension program
54. G91 - incremental dimensions
55. G92 - Spindle speed limit
56. G93 - Coordinate system setting
57. G94 - Feed rate in ipm (EMCO Lathe)
58. G95 - Feed rate in ipr (EMCO Lathe)
59. G96 - Surface cutting speed (EMCO Lathe)
60. G97 - Rotational speed rpm (EMCO Lathe)
61. G98 - withdraw the tool to the starting point or feed per minute
62. G99 - withdraw the tool to a safe plane or feed per revolution
63. G101 - Spline interpolation (Prolight Mill)

M-Codes:

M-Codes control machine functions and these include,

1. M00 - program stop
2. M01 - optional stop using stop button
3. M02 - end of program
4. M03 - spindle on CW
5. M04 - spindle on CCW
6. M05 - spindle off
7. M06 - tool change
8. M07 - flood with coolant
9. M08 - mist with coolant
10. M08 - turn on accessory #1 (120VAC outlet) (Prolight Mill)
11. M09 - coolant off
12. M09 - turn off accessory #1 (120VAC outlet) (Prolight Mill)

13. M10 - turn on accessory #2 (120VAC outlet) (Prolight Mill)
14. M11 - turn off accessory #2 (120VAC outlet) (Prolight Mill) or tool change
15. M17 - subroutine end
16. M20 - tailstock back (EMCO Lathe)
17. M20 - Chain to next program (Prolight Mill)
18. M21 - tailstock forward (EMCO Lathe)
19. M22 - Write current position to data file (Prolight Mill)
20. M25 - open chuck (EMCO Lathe)
21. M25 - set output #1 off (Prolight Mill)
22. M26 - close chuck (EMCO Lathe)
23. M26 - set output #1 on (Prolight Mill)
24. M30 - end of tape (rewind)
25. M35 - set output #2 off (Prolight Mill)
26. M36 - set output #2 on (Prolight Mill)
27. M38 - put stepper motors on low power standby (Prolight Mill)
28. M47 - restart a program continuously, or a fixed number of times (Prolight Mill)
29. M71 - puff blowing on (EMCO Lathe)
30. M72 - puff blowing off (EMCO Lathe)
31. M96 - compensate for rounded external curves
32. M97 - compensate for sharp external curves
33. M98 - subprogram call
34. M99 - return from subprogram, jump instruction
35. M101 - move x-axis home (Prolight Mill)
36. M102 - move y-axis home (Prolight Mill)
37. M103 - move z-axis home (Prolight Mill)

CNC PROGRAMMING:

The coordinates are almost exclusively Cartesian and the origin is on the work piece. For a lathe, the infeed/radial axis is the x-axis, the carriage/length axis is the z-axis. There is no need for a y-axis because the tool moves in a plane through the rotational center of the work. Coordinates on the work piece shown below are relative to the work.

CNC lathes are rapidly replacing the older production lathes (multispindle, etc) due to their ease of setting and operation. They are designed to use modern carbide tooling and fully utilize modern processes. The part may be designed and the tool paths programmed by the CAD/CAM process, and the resulting file uploaded to the machine, and once set and trailed the machine will continue to turn out parts under the occasional supervision of an operator. The machine is controlled electronically via a computer menu style interface; the program may be modified and displayed at the machine, along with a simulated view of the process. The setter/operator needs a high level of skill to perform the process, however the knowledge base is broader compared to the older production machines where intimate knowledge of each machine was considered essential. These machines are often set and operated by the same person, where the operator will supervise a small number of machines (cell).

INTRODUCTION

Linear Interpolation

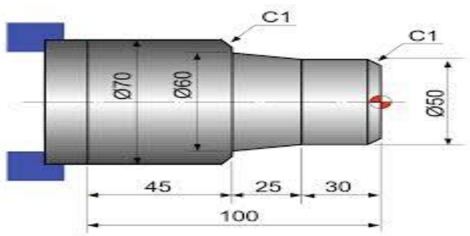
Linear interpolation is used in part programming to make a straight cutting motion from the cutter start position to its end position. It always uses the shortest distance a cutting tool path can take

G01 Linear Interpolation Motion (Group 01)

F - Feed rate
* **B** - B-axis motion command
* **C** - C-Axis motion command
* **U** - X-axis incremental motion command
* **W** - Z-axis incremental motion command
* **X** - X-axis absolute motion command
* **Y** - Y-axis absolute motion command
* **Z** - Z-axis absolute motion command
* **A** - Optional angle of movement (used with only one of *X*, *Z*, *U*, *W*)
* **I** - X-axis chamfering from Z to X (the sign does not matter, only for 90 degree turns)
* **K** - Z-axis chamfering from X to Z (the sign does not matter, only for 90 degree turns)

* **,C** - Distance from center of intersection where the chamfer begins (the sign does not matter, can chamfer non-90 degree lines)
* **,R / R** - Radius of the fillet or arc (the sign does not matter)

This G code provides for straight line (linear) motion from point to point. Motion can occur in 1 or more axes. You can command a *G01* with 3 or more axes All axes will start and finish motion at the same time. The speed of all axes is controlled so that the feed rate specified is achieved along the actual path. The C-Axis may also be commanded and this will provide a helical (spiral) motion. A C-Axis feed rate is dependent on the C-Axis diameter setting (Setting 102) to create a helical motion.
The *F* address (feedrate) command is modal and may be specified in a previous block. Only the axes specified are moved.



N020 G01 X25 Z-30 F250

N020 IS BLOCK NUMBER 20, G01 IS LINEAR INTERPOLATION, X25 IS X-COORDINATE,Z-30 IS Z-COORDINATE(When The Tool Moves towords Work Piece Negative Sign, Away From Work Piece +Ve Sign), F250 is feed rate 250 mm/min

Interpolation methods in CNC machine
Linear interpolation methods and Circular interpolation methods

Circular interpolation

The method of circular contouring is called circular interpolation. It is commonly used in profiling on CNC vertical and horizontal machining centers, as well as on lathes and many other CNC machines, such as simple milling machines, routers, burners, water jet and laser profilers, wire EDM, and others

G02 CW / G03 CCW Circular Interpolation Motion (Group 01)

F - Feedrate
* **I** - Distance along X Axis to center of circle
* **J** - Distance along Y Axis to center of circle
* **K** - Distance along Z Axis to center of circle
* **R** - Radius of circle
* **X** - X-Axis motion command
* **Y** - Y-Axis motion command
* **Z** - Z-Axis motion command
* **A** - A-Axis motion command

*indicates optional

I,**J** and **K** is the preferred method to program a radius. **R** is suitable for general radii.

These G codes are used to specify circular motion. Two axes are necessary to complete circular motion and the correct plane, *G17*-*G19*, must be used. There are two methods of commanding a *G02* or *G03*, the first is using the **I**, **J**, **K** addresses and the second is using the **R** address.

Using I, J, K addresses

I, *J* and *K* address are used to locate the arc center in relation to the start point. In other words, the *I*, *J*, *K* addresses are the distances from the starting point to the center of the circle. Only the *I*, *J*, or *K* specific to the selected plane are allowed (*G17* uses *IJ*, *G18* uses *IK* and *G19* uses *JK*). The *X*, *Y*, and *Z* commands specify the end point of the arc. If the *X*, *Y*, and *Z* location for the selected plane is not specified, the endpoint of the arc is the same as the starting point for that axis.

To cut a full circle the *I*, *J*, *K* addresses must be used; using an R address will not work. To cut a full circle, do not specify an ending point (*X*, *Y*, and *Z*); program *I*, *J*, or *K* to define the center of the circle. For example:

G02 I3.0 J4.0 (Assumes G17; XY plane) ;

Using the R address

The *R*-value defines the distance from the starting point to the center of the circle. Use a positive *R*-value for radii of 180 or less, and a negative *R*-value for radii more than 180

Interpolation

Circular Interpolation commands (G02, G03 G Codes) are used to move a tool along a circular arc.

G02 – Circular Interpolation Clockwise.

G03 – Circular Interpolation Anti-Clockwise.

How to Program Circular Interpolation?

On CNC lathe machines with control Clockwise Circular Interpolation (Arc) can be programmed in following ways.

N10 G02 X Z R

or

N10 G02 X Z I K

Explanation

N10 G02 X Z R

X – Arc end-point in X-axis

Z – Arc end-point in Z-axis

R – Arc Radius.

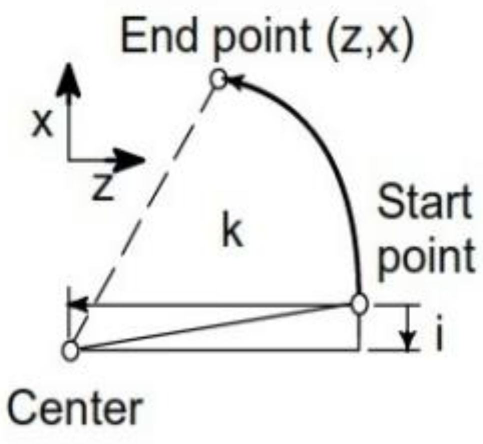
N10 G02 X Z I K

X-Arc end-point in X-axis.

Z – Arc end-point in Z-axis.

I – Distance from arc-start-point to arc-center-point in X-axis.

K – Distance from arc-start-point to arc-center-point in Z-axis.



Fanuc Circular Interpolation G02

EXP NO: 1 **DATE:**

STUDY OF "G" CODES AND "M" CODES

Aim: To study the "G" codes and "M" codes for the CNC lathe and milling machine.

Programming codes:

- Codes are model and do not have to be repeated in every sequence line.
- All dimensions are entered as decimals.

G –codes for turning machine:

MTAB INDIA PVT LTD

G- To define tool movement for preparatory function.

- G00 - Rapid move (not cutting)
- G01 - Linear move
- G02 - Clockwise circular motion
- G03 - Counterclockwise circular motion
- G04 - Dwell
- G17 - x-y plane
- G18 - z-x plane
- G19 - y-z plane
- G20 - inch data specification
- G21 - metric data specification
- G 28 - return to reference point (home position)
- G40 - cutter compensation cancel
- G50 - Preset position
- G70 - set inch based units or finishing cycle
- G71 - set metric units or stock removal or multiple turning
- G75 - pattern repeating (Emco Lathe) or multiple grooving
- G76 - deep hole drilling, cut cycle in z-axis or multiple threading
- G90 - absolute dimension program
- G91 - incremental dimensions
- G98 - withdraw the tool to the starting point or feed per minute

M –codes for turning machine:

- M00 - program stop
- M01 - optional stop using stop button
- M02 - end of program
- M03 - spindle on CW
- M04 - spindle on CCW
- M05 - spindle off
- M06 - tool change
- M10 - chuck open
- M11 - chuck close or tool change
- M 30 - program stop and restart

G –codes for milling machine:

MTAB INDIA PVT LTD

- G00 - Rapid move (not cutting)

- G01 - Linear move
- G02 - Clockwise circular motion
- G03 - Counterclockwise circular motion
- G04 - Dwell
- G17 - x-y plane
- G18 - z-x plane
- G19 - y-z plane
- G20 - inch data specification
- G21 - metric data specification
- G 28 - return to reference point (home position)
- G40 - cutter compensation cancel
- G50 - Preset position
- G68 – Rotation ON
- G69 – Rotation OFF
- G80 - fixed cycle cancel
- G83 - drilling or peak drilling cycle
- G90 - absolute dimension program
- G91 - incremental dimensions
- G94 - Feed rate in ipm
- G98 - withdraw the tool to the starting point or feed per minute
- G99 - withdraw the tool to a safe plane or feed per revolution
- G172- pocket frame mill
- G173- pocket outside face mill

M –codes for milling machine:

- M00 - program stop
- M01 - optional stop using stop button
- M02 - end of program
- M03 - spindle on CW
- M04 - spindle on CCW
- M05 - spindle off
- M06 - tool change
- M10 - chuck open
- M11 - chuck close or tool change
- M 30 - program stop and restart
- M 98 - sub program call
- M 99 - sub program exit

Result:

EXP NO: 2 **DATE:**

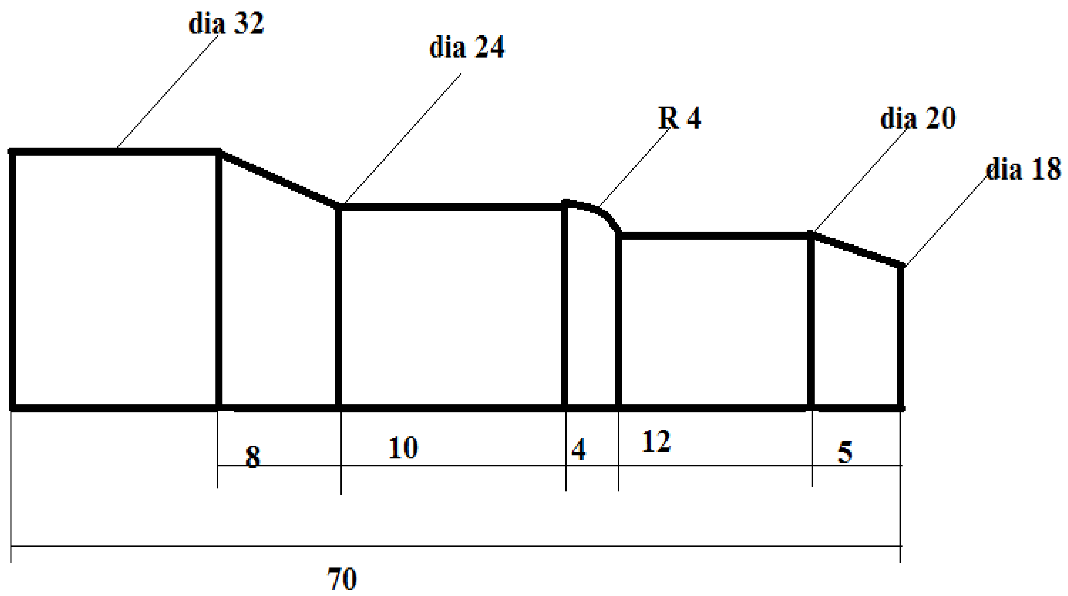
CNC LATHE – MULTIPLE TURNING OPERATION

Aim: To perform multiple turning operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece("cylinder");
G71G94;
G75X0Z0;
M06T2D1;
M03S1500;
M08;
G00X32Z2;

Go for counter turning-counter call-sub program-give the name-then back-stock removal-there is a window for stock removal appears-

PRG	B
SC	2.00
F	100.00
MACHINING	ROUGHING+FINISHING
FS	100.00 LONGITUDINAL OUT SIDE
D	0.300
UX	0.100
UZ	0.100
XD	32.00 abs
ZD	2.00 abs
RELIEF CUTS	YES
FR	100.00

ACCEPT

<u>Note:</u> the stock removal name is different what we given in the sub program name.

```
M09;
G75X0Z0;
MO5;
M30;
```

Then go subprogram in the part program it self-subprogram-new-the name is same as what we given in the countercall sub program name- then type the program.

```
G01X18 Z0;
G01 X20 Z-5;
G01Z-17;
G03X24Z-21CR=4;
G01Z-31;
G01X32 Z-39;
M17;
```

Then go for the main program-simulation-execute-auto-reset-start.

<u>Result:</u>

EXP NO: 3 **DATE:**

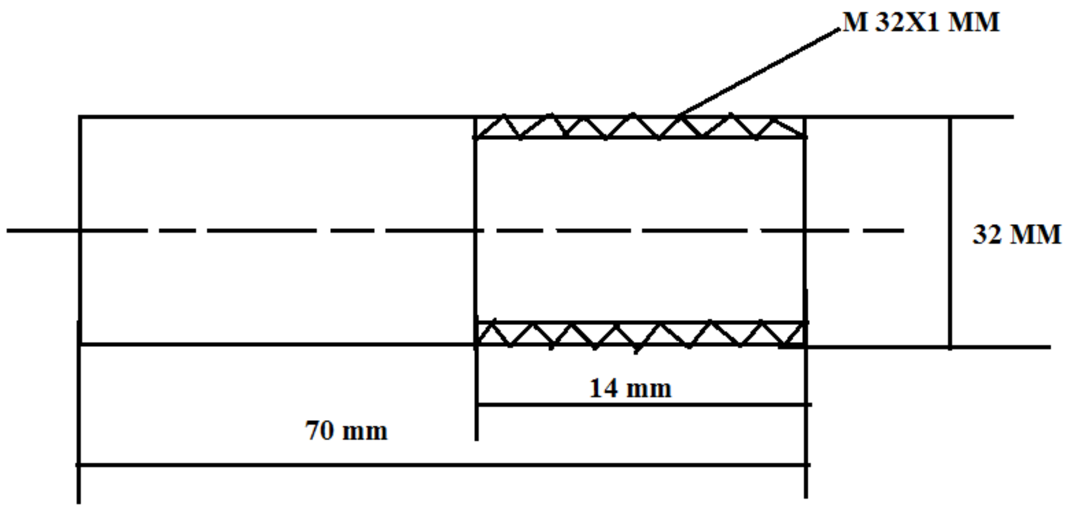
CNC LATHE – EXTERNAL THREADING OPERATION

Aim: To perform external thredaing operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece(“cylinder”);
G71G94;
G75X0Z0;
M06T04D1;
M03S500;
M08;
G00X32Z2;

Then go for turning in the panel board-and then threading table will be appears.

TABLE	ISO METRIC
SELECT	M 10
P	1.500 mm/rev
MACHINING	ROUGHING+FINISHING Linear External thread

X0	32.00
Z0	0.000
Z1	-14.00 abs
Lw	5.000
LR	0.000
H1	0.920
∞p	30.00
D1	0.100
U	0.010
NN	0
VR	0.500
MULTIPLE	NO
∞0	0.00

ACCEPT

M09;
G75X0Z0;
MO5;
M30;
Then go for the main program-simulation-execute-auto-reset-start.

Result:

EXP NO: 4 **DATE:**

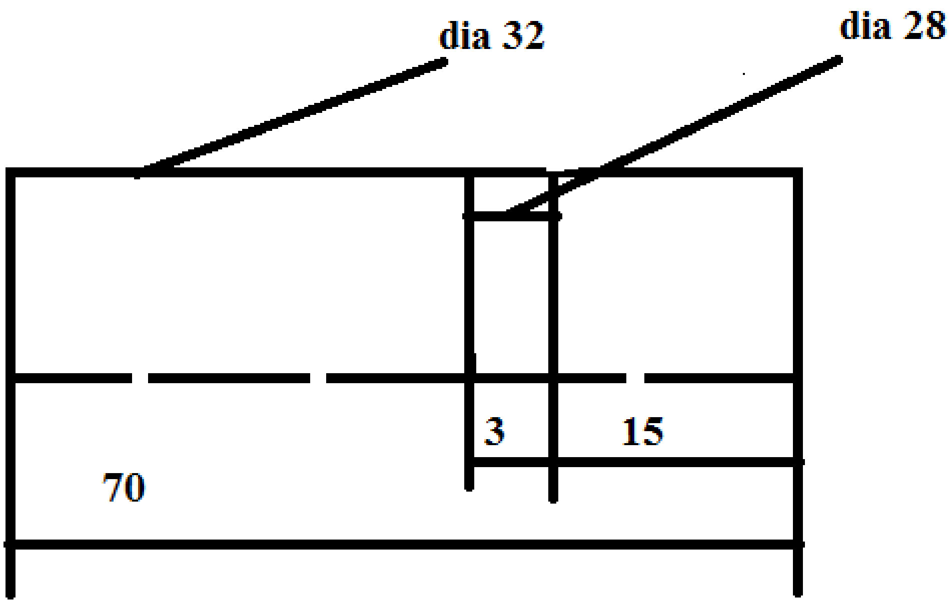
CNC LATHE – EXTERNAL GROOVING OPERATION

Aim: To perform external Grooving operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece("cylinder");
G71G94;
G75X0Z0;
M06T6D1;
M03S900;
M08;
G00X33Z5;
GO1 X32 Z-15F50;

Then go for turning in the panel board-Grooving-and then grooving table will be appears.

SC	5.00
F	50.00
MACHINING	ROUGHING+FINISHING
POS	Position of machining reference point
X0	32.00
Z0	-17.5
B1	3.00
T1	28.00
D	0.100
UX	0.050
UZ	0.050
N	1

ACCEPT

G01 Z5;
M09;
G75X0Z0;
MO5;
M30;
Then go for the main program-simulation-execute-auto-reset-start.

Result:

EXP NO: 05 **DATE:**

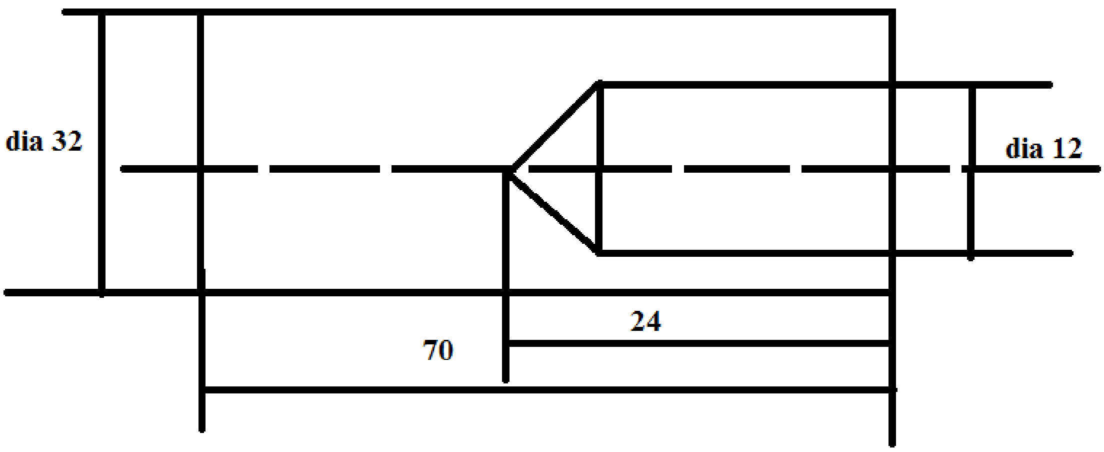
CNC LATHE – DRILLING OPERATION

Aim: To perform Drilling operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece("cylinder");
G71G94;
G75X0Z0;
M06T7D1;
M03S1000;
M08;
G00X0Z5F50;

Then go for DRILLING in the panel board-Deep hole drilling-and then Drilling table will be appears.

PL	G17(XY)
RP	1.00
SC	5.00
Z0	0.00
Z1	-24 abs
D	-0.500 abs
FD1	70.00%
DF	50.00%
V1	0.300
Lead distance	Automatically
DTB	0.000S
DT	0.000S
DTB	0.000S

ACCEPT
THEN TYPE
MCALL:

G00 Z10;
M09;
G75X0Z0;
MO5;
M30;
Then go for the main program-simulation-execute-auto-reset-start.

Result:

EXP NO: 06 **DATE:**

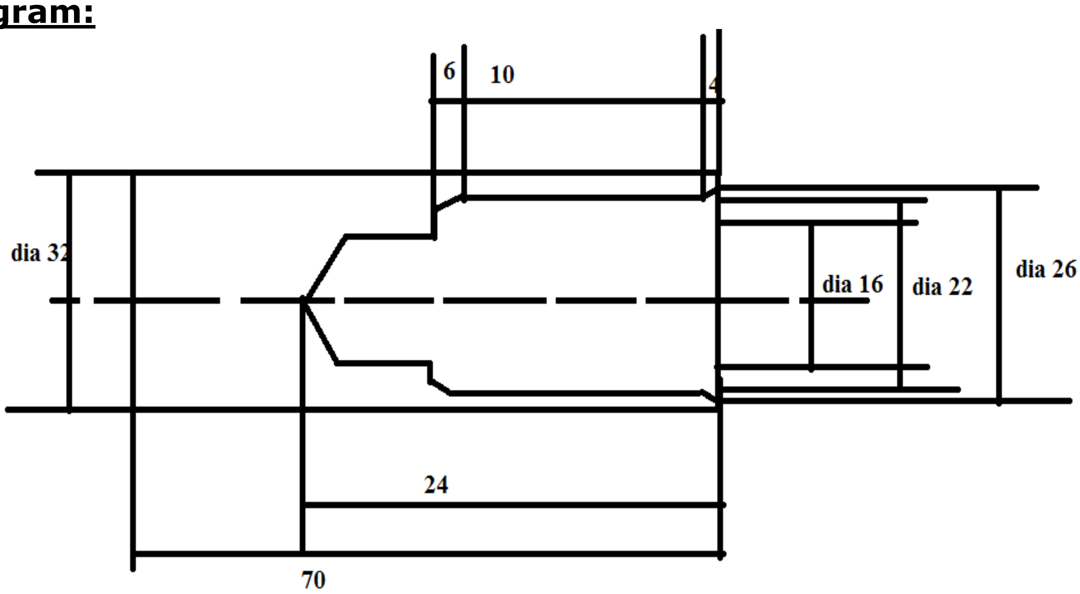
CNC LATHE – BORING OR INTERNAL TURNING OPERATION

Aim: To perform internal turning operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece(“cylinder”);
G71G94;
G75X0Z0;
M06T3D1;
M03S1500;
M08;
G00X12Z5F100;

Go for counter turning-counter call-sub program-give the name-then back-stock removal-there is a window for stock removal appears-

PRG	G
SC	5.00
F	100.00
MACHINING	ROUGHING+FINISHING
FS	100.00 LONGITUDINAL INSIDE
RP	0.500abs
D	0.200
UX	0.100
UZ	0.100
XD	12.00 abs
ZD	5.00 abs
RELIEF CUTS	YES
FR	100.00

ACCEPT

Note: the stock removal name is different what we given in the sub program name.
M09;
G75X0Z0;
MO5;
M30;
Then go subprogram in the part program it self-subprogram-new-the name is same as what we given in the countercall sub program name-then type the program.
G01X26 Z0;
G02 X22 Z-4CR=4;
G01Z-14;
G01X16Z-20;
G01X12;
M17;
Then go for the main program-simulation-execute-auto-reset-start.

Result:

EXP NO: 07 **DATE:**

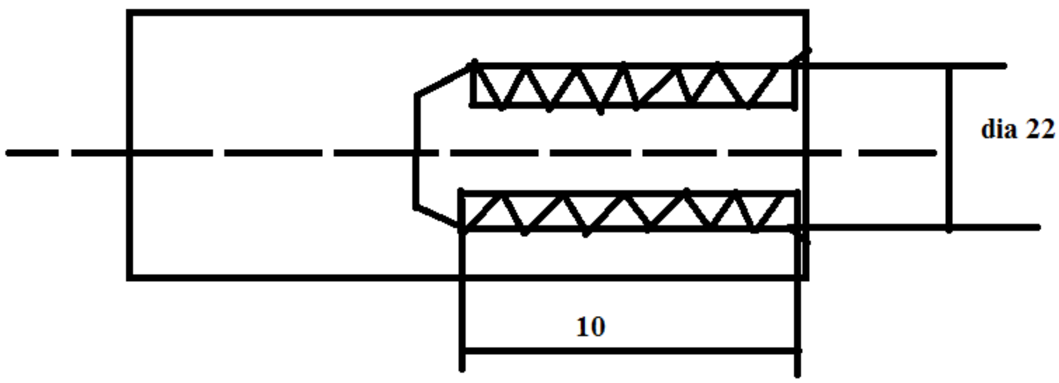
CNC LATHE – INTERNAL THREADING OPERATION

Aim: To perform Internal threading operation by using CNC programming codes.

Apparatus required:

- CNC machine
- Personal computer
- CNC software
- Cutting tool
- Work piece

Program:



Define the work piece
Work piece("cylinder");
G71G94;
G75X0Z0;
M06T01D1;
M03S500;
M08;
G00X22Z2;

Then go for turning in the panel board-and then threading table will be appears.

TABLE	ISO METRIC
SELECT	M 10
P	1.500 mm/rev
MACHINING	ROUGHING+FINISHING Linear Internal thread
X0	22.00

Z0	0.000
Z1	-10 abs
Lw	5.000
LR	0.000
H1	0.920
∞p	30.00
D1	0.100
U	0.010
NN	0
VR	0.500
MULTIPLE	NO
∞0	0.00

ACCEPT

M09;
G75X0Z0;
MO5;
M30;
Then go for the main program-simulation-execute-auto-reset-start.

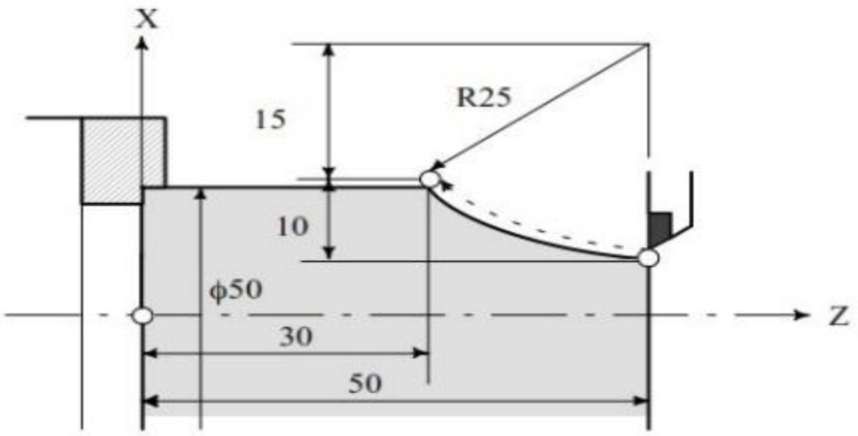
Result:

EXP NO: 08 **DATE:**

Circular Interpolation G02 G Code, Diameter Programming

Aim: To perform diameter programming by using Circular Interpolation commands (G02, G03 G Codes) are used to move a tool along a circular arc.

Apparatus : The following circular interpolation program is programmed in multiple different ways. CNC programmers/machinists can use G02/G03 X Z R method or G02/G03 X Z I K method if control allows.



Fanuc Circular Interpolation G02 G Code Example
(Diameter programming)
G02 X50 Z30 I25 F0.3
G02 U20 W-20 I25 F0.3
G02 X50 Z30 R25 F0.3
G02 U20 W-20 R25 F0.3

Result:

INTRODUCTION

Fanuc G72.2 Linear Copy (Figure Copy Function CNC Mill)

Fanuc G72.2 Linear Copy

Using G72.2 Linear Copy G-code a figure specified by a subprogram can be repeatedly produced with Linear movement.

Programming

G72.2 P... L... I... J...

Parameters

Parameter	Description
P	Subprogram number
L	Number of times the operation is repeated
I	Shift along X-axis
J	Shift along Y-axis

G-Code Data

Modal/Non-Modal	G-Code Group
Non-Modal	00

Programming Notes

Notes

In the G72.2 block, addresses other than P, L, I and J are ignored.

P, I and J must always be specified.

If L is not specified, the figure is copied once.

For shifts (I, J), specify increments.The n-th geometric shift is equal to the specified shift times (n – 1).

First block of the subprogram

Always specify a move command in the first block of a subprogram that performs a linear copy.

If the first block contains only the program number such as O00001234; and does not have a move command, movement may stop at the start point of the figure made by the n-th (n = 1,2, 3, …) copying.

Example of an incorrect program

_ O00001234 ;

G00 G90 X100.0 Y200.0 ;

____;

____;

M99 ;

Example of a correct program

O00001000 G00 G90 X100.0 Y200.0 ;

____;

____;

M99 ;

Limitation

_Specifying two or more commands to copy a figure

G72.2 cannot be specified more than once in a subprogram for making a linear copy (If this is attempted, alarm PS0901will occur).
In a subprogram that specifies linear copy, however, rotational copy (G72.1) can be specified. Similarly, in a subprogram that specifies rotational copy, linear copy can be specified.

Commands that must not be specified

Within a program that performs a linear copy, the following must not be specified:

_Command for changing the selected plane (G17 to G19)
_Command for specifying polar coordinates (G16)
_Reference position return command(G28)
_Axis switching
_Coordinate system rotation (G68)
_scaling (G51)
_programmable mirror image (G51.1)

Single block

Single-block stops are not performed in a block with G721.1 or G72.2.

G2 Circular Interpolation CW

G2 X... Y... I... J...

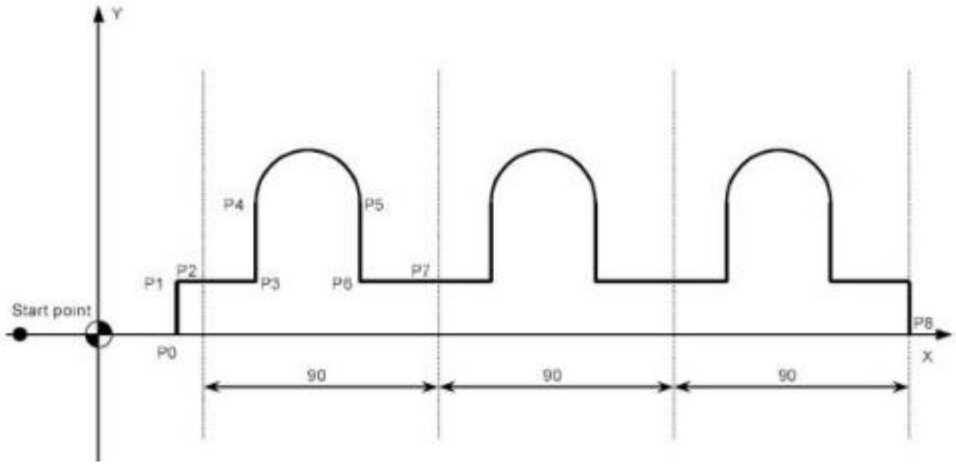
Parameter	**Desscription**
X	Coordinates of the arc end point in x-axis.
Y	Coordinates of the arc end point in y-axis.
I	Distance from arc start-point to arc-center-point in Xaxis.
J	Distance from arc start-point to arc-center-point in Yaxis

EXP NO: 09 **DATE:**

Linear Interpolation G72.2 Linear copy Programming

Aim: To perform Linear interpolation G72.2 Linear copy programming by using Linear Interpolation parameters (P,L,I,J) and using G-code commands

Apparatus : The following Linear interpolation G72.2 Linear copy program is programmed in multiple different ways. CNC programmers/machinists can use G-codes & M-codes, Parameters(P,L,I,J) if control allows.



Main program

```
O3000 ;
 N10 G90 G00 X-30. Y0 ;
 N20 X0 ;
N30 G01 G17 G41 X30. D01 F100 ; (P0)
N40 Y20. ; (P1)
N50 X40. ; (P2)
 N60 G72.2 P3100 L3 I90.0 J0 ;
N70 G90 X310. Y0 ; (P8)
N80 X0 ;
N90 G40 G00 X-30.0 ;
N100 M30 ;
```

Sub program

```
O3100 G91 G01 X20. ; (P3)
N100 Y30. ;
(P4) N200 G02 X40. I20. ;
(P5) N300 G01 Y-30. ;
(P6) N400 X30. ;
(P7) N500 M99 ;
```

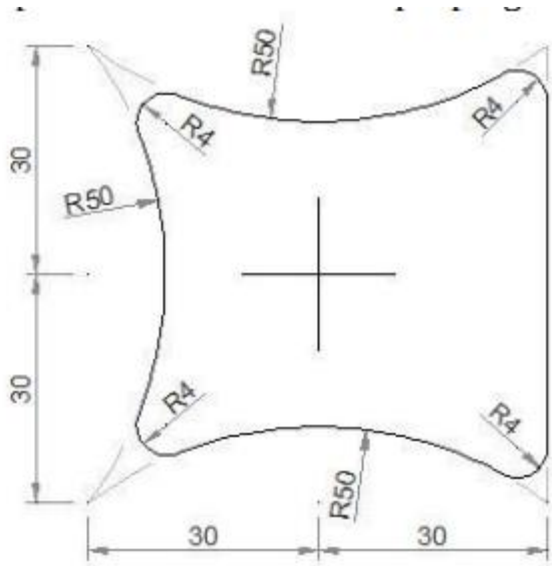
RESULT:

EXP NO: 10 **DATE:**

G02 G03 Example CNC Mill
G02 G03 Circular interpolation CNC mill program

Aim: To perform Circular interpolation G02,G03 CNC programming by using Circular Interpolation parameters (X,Y,I,J) and using G-code and M-code commands

Apparatus : The following Circular interpolation G02,G03 program is programmed in multiple different ways. CNC programmers/machinists can use G-codes & M-codes, Parameters(X,L,I,J) if control allows.



G02 G03 Example CNC Mill

CNC Part Program

G0 X30 Y-30 (P1)

G1 Y22.67 (P2)

G3 X24.07 Y26.18 R4 (P3)

G2 X-18.27 Y23.46 R50 (P4)

G3 X-23.46 Y18.27 R4(P5)

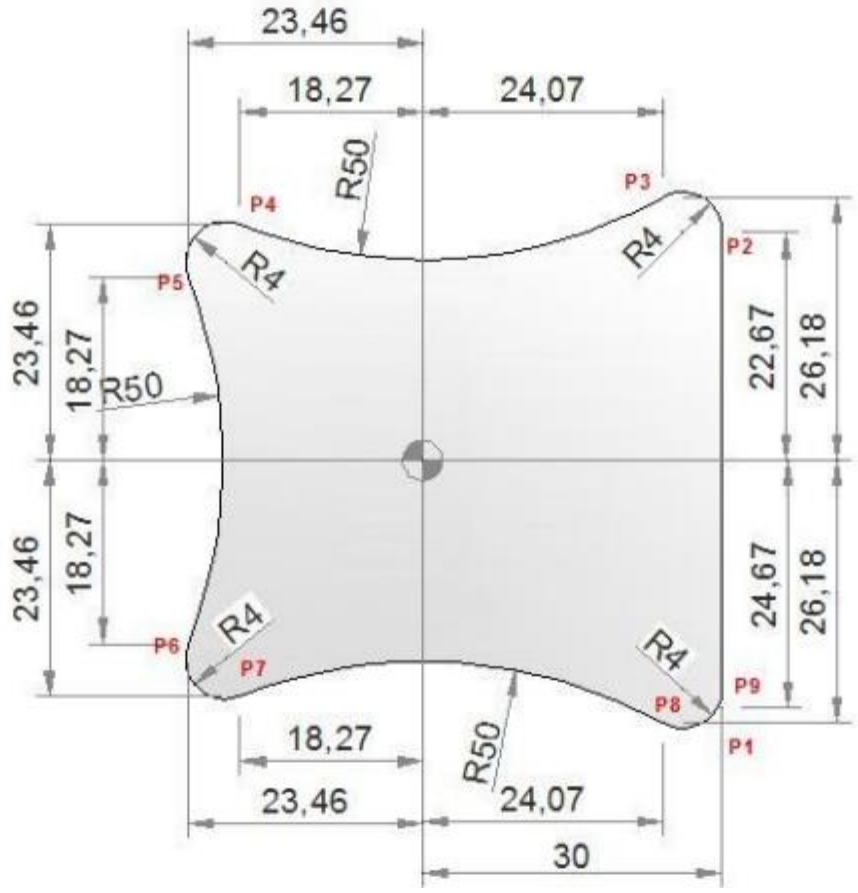
G2 X-23.46 Y-18.27 R50 (P6)

G3 X-18.27 Y-23.46 R4 (P7)

G2 X24.07 Y-26.18 R50 (P8)

G3 X30 Y-24.67 R4 (P9)

G1 X33



G M S T Codes Explanation

Code	Description
G00	Rapid traverse
G01	Linear interpolation
G02	Circular interpolation CW
G03	Circular interpolation CCW
G40	Cutter compensation cancel
G41	Tool nose radius compensation left
G43	Tool length compensation + direction
G49	Tool length compensation cancel
G90	Absolute command
G91	Increment command
M03	Spindle start forward CW
M06	Tool change
M30	End of program (Reset)
M98	Subprogram call
M99	End of subprogram
T	Tool
S	Speed
F	Feed

RESULT:

INTRODUCTION

G82 Drilling Cycle CNC Milling Example Program

G82 drilling cycle with dwell can be used for normal drilling where bottom of the hole need more accurate machining. Otherwise if you just want a drilling cycle without a dwell time at the bottom of the hole G81 drilling cycle is a big time saver. Following is a cnc programming example which illustrates the use of G82 Drill cycle.

PECK DRILLING

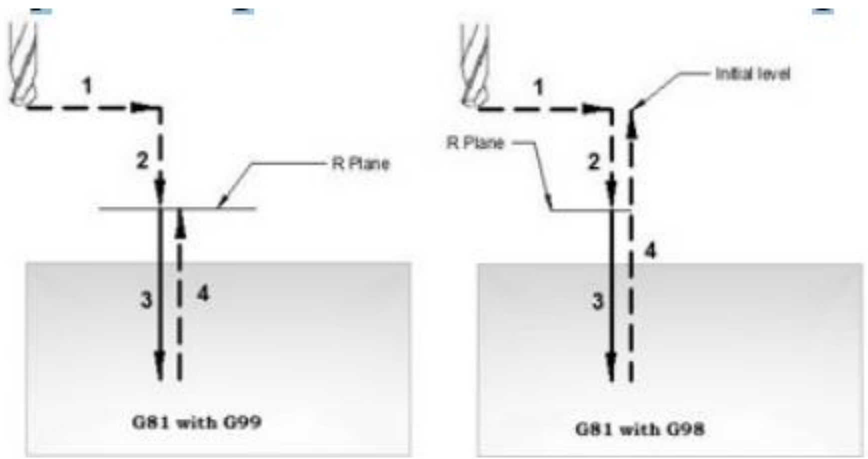
A complete cnc part-program which shows how G98 and G99 (canned cycle return level) work with G81 drilling cycle and G83 peck drilling cycle for drilling of a component which have different heights.

G98 G99 Summary

G98 and G99 are modal commands that change the way canned cycles (G81,G83 etc.) operate.

When G98 is active, the Z-axis will return to the start position (initial plane) when it completes an single operation.

When G99 is active, the Z-axis will be returned to the R point (plane) when the canned cycle completes a single hole. Then the machine will go to the next hole.

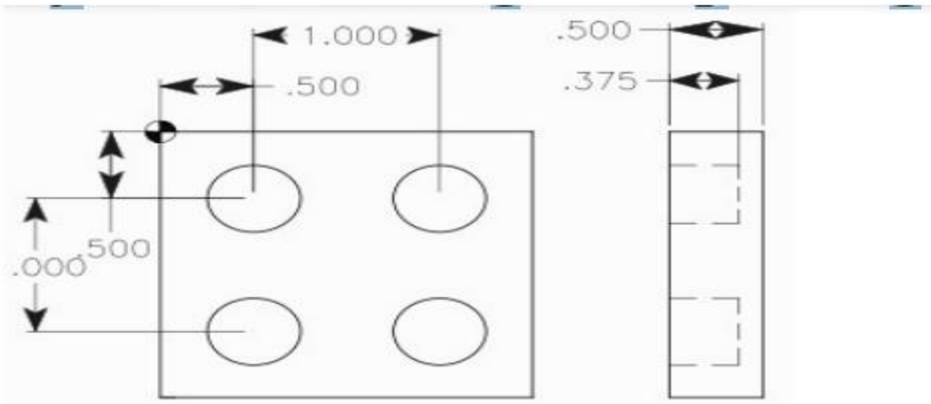


EXP NO: 11 **DATE:**

G82 Drilling Cycle CNC Milling Program

Aim: To perform G82 Drilling Cycle CNC Milling programming by using G-code,M-code, S,T and F commands

Apparatus : The following G82 Drilling Cycle CNC Milling program is programmed in multiple different ways. CNC programmers/machinists can use G-codes & M-codes, Parameters(P,J.X,L,I,J) if control allows.



G82 Drilling Canned Cycle with Dwell CNC Milling Example Program

O10076

N10 T11 M06

N20 G90 G54 G00 X0.5 Y-0.5

N30 S1200 M03

N40 G43 H11 Z1. M08

N50 G82 G99 Z-0.375 P1 R0.1 F7.5

N60 X1.5 N70 Y-1.5

N80 X0.5

N90 G80 G00 Z1. M09

N100 G53 G49 Z0. M05

N110 M30

Note N50 – CNC machines with Fanuc cnc control will use P1000 instead of P1 which is used for Haas CNC machines

RESULT:

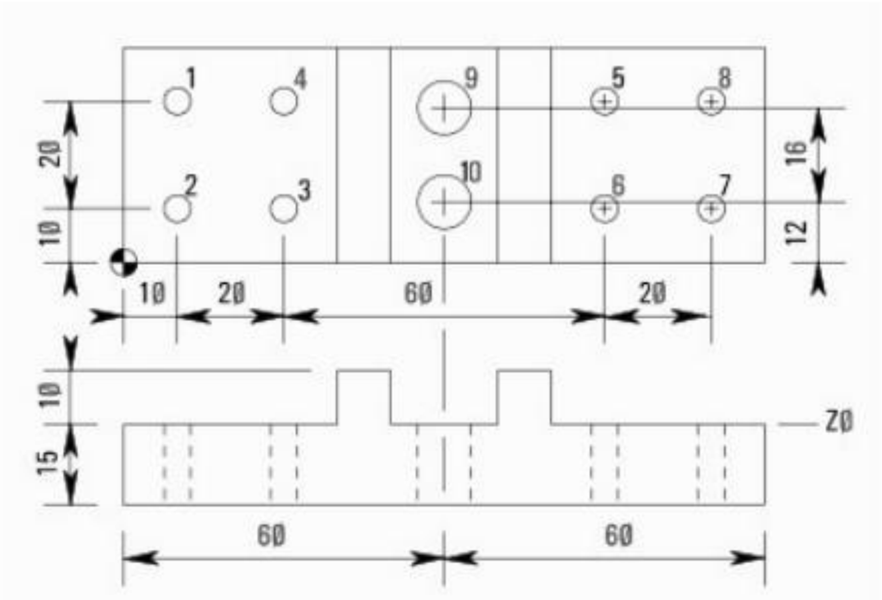
EXP NO: 12 **DATE:**

G81 Drilling Cycle G83 Peck Drilling with G98 G99 Example Program -Pocket Drilling (OR)
G81 drilling cycle working – G98 G99 return level CNC Part Program

---

Aim: To perform G 81 & G83 Peck Drilling Cycle CNC Milling programming by using G-code,M-code, S,T and F commands

Apparatus : The following G81 & G83 Peck Drilling Cycle CNC Milling program is programmed in multiple different ways. CNC programmers/machinists can use G-codes & M-codes, Parameters(P,J.X,L,I,J) if control allows.



G81 Drilling Cycle with G98 G99 G code Example Program
N10 M06 T01
N20 G90 G00 X10 Y30 Z12 S1000 M03
N30 G99 G81 X10 Y30 Z-17 R2 F75 (Hole 1)
N40 Y10 (Hole 2) N50 X30 (Hole 3)
N60 Y30 (Hole 4) N70 G98 X90 (Hole 5)
N80 G99 Y10 (Hole 6)
N90 X110(Hole 7)
N100 G98 Y30 (Hole 8)
N110 G91 G80 G28 X0 Y0 Z0 M05
N120 M06 T02
N130 G90 G00 X60 Y28 Z12 S750 M03
N140 G99 G83 X60 Y28 Z-17 Q6 R2 F60 (Hole 9)
N150 G98 Y12 (Hole 10)
N160 G91 G80 G28 X0 Y0 Z0 M05
N170 M30

Explanation

N10-Tool change (M06) to tool no.1

N20-Rapid traverse to X10 Y30 Z12, Spindle started clockwise (M03) with 1000rpm (S1000).

N30-Drilling starts (G81) at X10 Y30 with cutting-feed (F75) drill will retract to R-plane after drilling operation.

N40-Next drilling position Y10 (as G99 is a modal g-code drill will keep on retracting to R-plane until G98 is given).

N50-Next drill at X30.

N60-Drill at Y30

N70-Drill at X90 & Retract to Initial-plane.

N80-Drill at Y10 & Retract to R-plane.

N90-Drill at X110

N100-Drill at Y30 & Retract to Initial-plane.

N110-Drilling cycle is cancelled (G80), return to reference point (G28) for tool change, stop spindle (M05).

N120-Tool change (M06) to tool number 2.

N130-Rapid traverse to X60 Y28 Z12, start spindle at 750rpm (S750) clockwise (M03).

N140-G83 Peck drilling starts at X60 Y28, drill depth is Z-17 and drill peck size is Q6, drilling feed is F60 N150-Next deep drill at Y12 (return to initial point).

N160-G83 Peck drilling cycle cancelled with G80, tool returned to reference point (G28), spindle stopped (M05). N170-Part-program end with return to program start (M30)

G & M Codes

Code	Description
T	Tool no. used.
M06	Tool change command.
G90	Absolute programming
G00	Rapid traverse
S	Cutter speed
M03	Cutter rotation Clockwise
M08	Coolant on.
G81	Fanuc drilling cycle.
G83	Fanuc peck drilling cycle.
G98	Return to initial point in canned cycle.
G99	Return to R point in canned cycle.
F	Cutting feed.
G80	Canned cycle cancel.
M09	Coolant off.
G28	Return to reference position.
G91	Incremental programming.
M05	Cutter rotation stop.
M30	CNC part-program end with return to program-start.

RESULT:

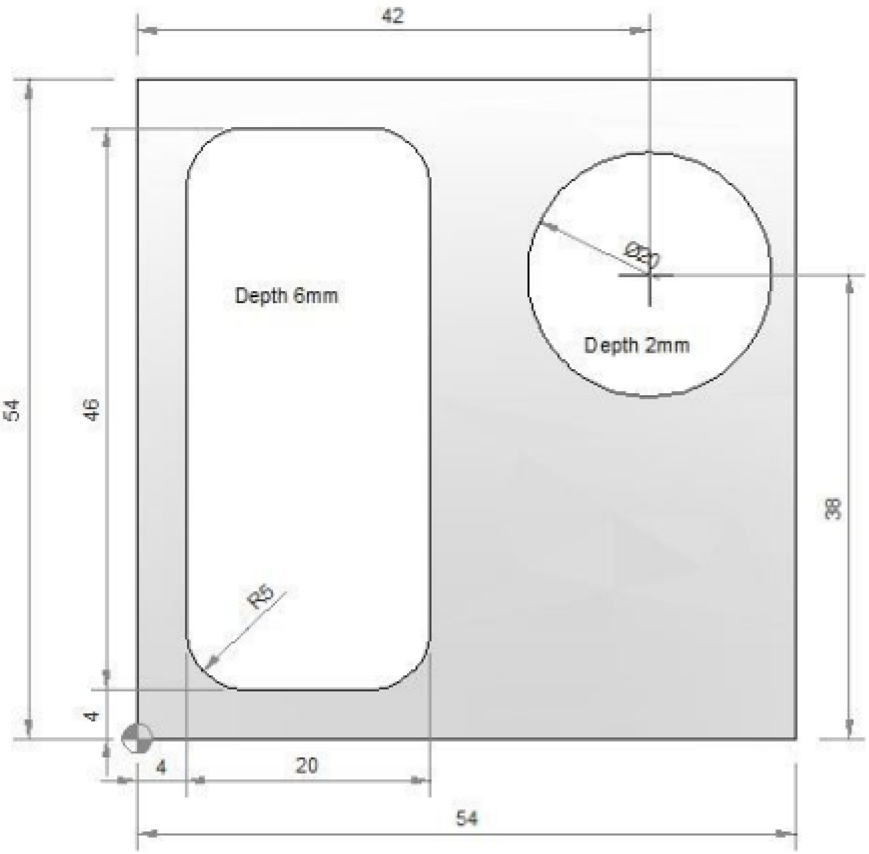
EXP NO: 13 **DATE:**

CNC Pocket Milling Program Example – Peck Milling

Aim: To perform CNC Pocket Milling G02,G03 CNC programming by using parameters (P,L,X,Y,I,J) and using G-codes, M-codes, S, T and F commands

Apparatus : The following Peck Milling program is programmed in multiple different ways. CNC programmers/machinists can use G-codes & M-codes, Parameters(P,L,X,Y,I,J) if control allows.

CNC milling program example which shows how a cnc program can be made to machine Pockets on a cnc mill. This program example uses Peck milling to cut material to machine a rectangular and one round pocket.



Main Program

Milling cutter diameter: 10mm

N05 G55
N10 M6 T2 H3 G43 M3
N15 S1000 F60
N20 G0 X9 Y9 Z1
N25 G1 Z0
N30 M98 P030035
N35 G0 Z1 G90
N40 X42 Y38

```
N45 G1 Z-2 F30
N50 X47 F300
N55 G3 X47 Y38 I-5 J0
 N60 G0 Z100 N65 G49
 N70 M30
```

Subprogram

```
O0035
N05 G1 Z-2 G91 F30
N10 X10 F100
N15 Y36
N20 X-10
N25 Y-36
N30 M99
```

Explanation Although this cnc mill program is self explanatory

M98 P030035

this code mean call Subprogram No. 0035 three times.

Code	Description
G00	Rapid traverse
G01	Linear interpolation
G02	Circular interpolation CW
G03	Circular interpolation CCW
G40	Cutter compensation cancel
G41	Tool nose radius compensation left
G43	Tool length compensation + direction
G49	Tool length compensation cancel
G90	Absolute command
G91	Increment command
M03	Spindle start forward CW
M06	Tool change
M30	End of program (Reset)
M98	Subprogram call
M99	End of subprogram
T	Tool
S	Speed
F	Feed

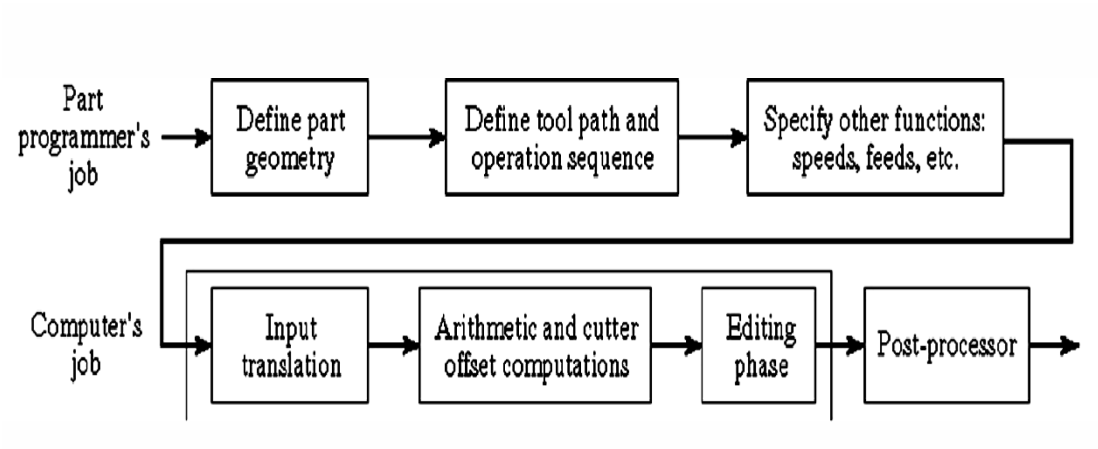
RESULT:

AUTOMATICALLY PROGRAMMED TOOLS (APT)

Computer assisted part programming
(APT, Automatically Programmed Tool)

-Manual part programming is time-consuming, tedious, and subject to human errors for complex jobs.
-Machining instructions are written in English-like statements that are translated by the computer into the low-level machine code of the MCU.
-It is used for more complex jobs.
-APT (Automatically Programmed Tool)
The various tasks in computer-assisted part programming are divided between;

The human part programmer, The computer.



Sequence of activities in computer-assisted part programming

Part Programmer's Job

Two main tasks of the programmer:
1- Define the part geometry
2- Specify the tool path and Operation Sequence

1- Define the part geometry

Underlying assumption: no matter how complex the part geometry, it is composed of basic geometric elements and mathematically defined surfaces
Geometry elements are sometimes defined only for use in specifying tool path
Examples of part geometry definitions:
P4 = POINT/35, 90,0
L1 = LINE/P1, P2
C1 = CIRCLE/CENTER, P8, RADIUS, 30.0

2- Specify the tool path and Operation Sequence

Tool path consists of a sequence of points or connected line and arc segments, using previously defined geometry elements

Point-to-Point command:

GOTO/P0

Continuous path command:

GOLFT/L2, TANTO, C1

Other Functions in Computer-Assisted Part Programming

-Specifying cutting speeds and feed rates

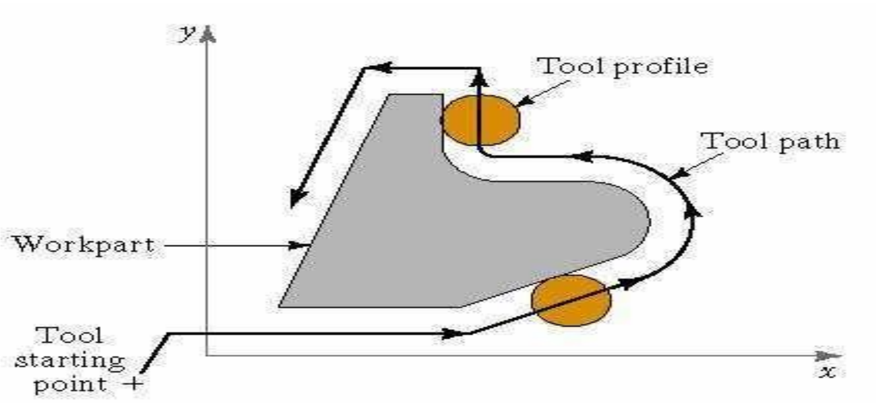
-Designating cutter size (for tool offset calculations)

-Specifying tolerances in circular interpolation

-Naming the program

-Identifying the machine tool

Cutter Offset



Cutter path must be offset from actual part outline by a distance equal to the cutter radius

Computer Tasks in Computer-Assisted Part Programming

1. Input translation – converts the coded instructions in the part program into computer- usable form
2. Arithmetic and cutter offset computations – performs the mathematical computations to define the part surface and generate the tool path, including cutter offset compensation (CLFILE)
3. Editing – provides readable data on cutter locations and machine tool operating commands (CLDATA)
4. Post processing – converts CLDATA into low-level code that can be interpreted by the MCU

There are four basic types of statements in the APT language:

1. Geometry statements, also called *definition statements*; are used to define the geometry elements that comprise the part.
2. Motion commands; are used to specify the tool path.
3. Postprocessor statements; control the machine tool operation, for example, to specify speeds and feeds, set tolerance values for circular interpolation, and actuate

other capabilities of the machine tool.

4. Auxiliary statements; a group of miscellaneous statements used to name the part program, insert comments in the program and accomplish similar functions.

- APT vocabulary words consist of six or fewer characters. The characters are almost always letters of the alphabet.

Geometry statements

The points, lines, and surfaces must be defined in the program prior to specifying the motion statements. The general form of an APT geometry statement is the following

SYMBOL = GEOMETRY TYPE/descriptive data

as an example;

P1 = POINT/20.0, 40.0, 60.0

major words (POINT) — minor words (20.0, 40.0, 60.0)

A symbol can be nay combination of six or fewer alphabetical and numerical characters, at leastone of which must be alphabetical. Also the symbol cannot be an APT vocabulary word. Some examples are presented in the following Table:

Examples of Permissible and Impermissible Symbols in APT Geometry Statements

Symbol	*Permissible or Not, and Why*
P1	Permissible
PZL	Permissible
ABCDEF	Permissible
PABCDEF	Not permissible, too many characters
123456	Not permissible, all numerical characters
POINT	Not permissible, APT vocabulary word
P1.5	Not permissible, only alphabetic and numerical characters are allowed

Points: Specification of a point can be accomplished by the following:

1) Designating its x, y, and z coordinates;

P1 = POINT/15.0, 10.0, 25.0

2) As the intersection of two intersecting lines;

P2 = POINT/INTOF, L1, L2

L1 and L2 are two previously defined lines.

Lines: A line in APT is considered to be of infinite length in both directions. Specification of a line can be accomplished by the following:

1. Two points through which it passes;

   L1 = LINE/P3, P4

   P3 and P4 are two previously defined points.

2. Passes through point (P5) and parallel to another line (L3) that has been previously defined; L2 = LINE/P5, PARLEL, L3

Planes: In APT, a plane extends indefinitely. A plane can be defined by the following:

1) Three points through which it passes;

PL1 = PLANE/P1, P2, P3

P1, P2 and P3 must be non-collinear.

2) Passes through point (P2) and parallel to another plane (PL1) that has been previously defined;

PL2 = PLANE/P2, PARLEL, PL1

Circles: In APT, a circle is considered to be a cylindrical surface that is perpendicular to the *x-y* plane and extends to infinity in the *z*-direction. A circle can be defined by the following:

3) Its center and radius;
   C1 = CIRCLE/CENTER, P1, RADIUS, 25.0
4) Three points through which it passes;
   C2 = CIRCLE/P4, P5, P6
   The three points must not be collinear.

Certain ground rules must be obeyed when formulating APT geometry statements. Following are four important APT rules:

1. Coordinate data must be specified in the order *x*, then *y*, then *z*, because the statement

   P1 = POINT/20.5, 40.0, 60.0

   is interpreted to mean $x = 20.5$ mm, $y = 40.0$ mm, and $z = 60.0$ mm.

2. Any symbols used as descriptive data must have been previously defined; for example, in the statement

   P2 = POINT/INTOF, L1, L2

   the two lines L1 and L2 must have been previously defined. In setting up the list of geometry statements, the APT programmer must be sure to define symbols before using them in subsequent statements.

3. A symbol can be used to define only one geometry element. The same symbol cannot be used to define two different elements. For example, the following statements would be incorrect if they were included in the same program:

P1 = POINT/20, 40, 60

P1 = POINT/30, 50, 70

4. Only one symbol can be used to define any given element. For example, the following two statements in the same part program would be incorrect:

P1 = POINT/20, 40, 60

P2 = POINT/20, 40, 60

Motion Commands

All APT motion statements follow a common format, just as geometry statements have their ownformat. The general form of an APT motion command is:

MOTION COMMAND/descriptive data

as an example;

GOTO/P1

At the beginning of the sequence of motion statements, the tool must be given a starting point. This is likely to be the target point, the location where the operator has positioned the tool at the start of the job. The part programmer keys into this starting position with the following statement:

FROM/PTARG

Where FROM is an APT vocabulary word indicating that this is the initial point from which all others will be referenced; and PTARG is the symbol assigned to the starting point.

Another way to make this statement is the following:

FROM/-20.0, -20.0, 0

- The FROM statement occurs only at the start of the motion sequence

It is appropriate to distinguish between point-to-point motions and contouring motions.

Point-to-point motions

There are two commands; GOTO and GODLTA.

* The GOTO statement instructs the tool to go to a particular point location specified in the descriptive data. Two examples are:

GOTO/P2

GOTO/25.0, 40.0, 0

* The GODLTA command specifies an incremental move for the tool. To illustrate, the following statement instruct the tool to move from its present position by a distance of 50 mm in *x*-direction, 120 mm in *y*-direction, and 40 mm in *z*-direction;

GODLTA/50.0, 120.0, 40.0

* The GODLTA statement is useful in drilling and related machining operations. The tool can be directed to go to a given hole location; then the GODLTA command can be used to drillthe hole, as in the following sequence;

GOTO/P2 GODLTA/0, 0, -50.0

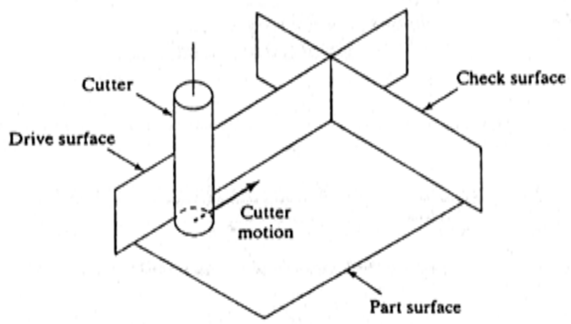
GODLTA/0, 0, 50.0

Contouring motions

These are more complicated than PTP commands are because the tool's position must becontinuously controlled throughout the move.

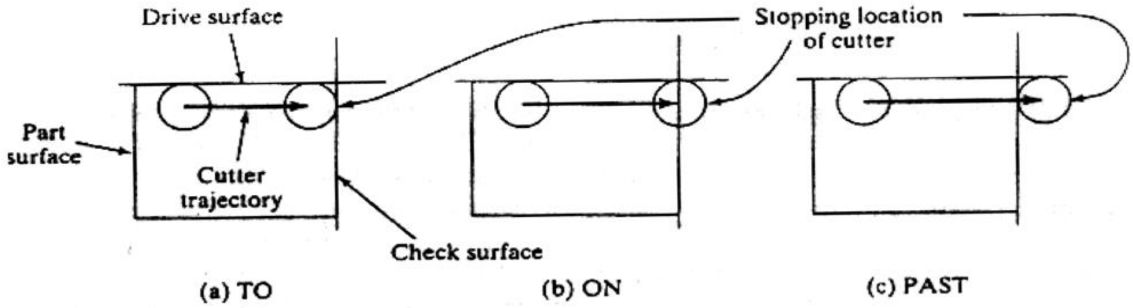
- The tool is directed along two intersecting surfaces until it reaches a third surface, as shownin the following Figure;

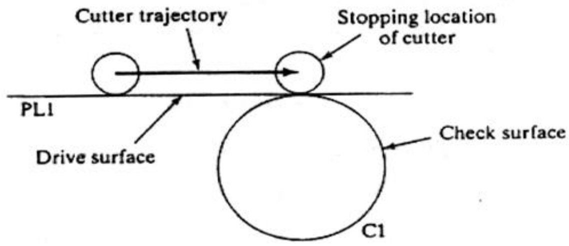
1. *Drive surface*; this is the surface that guides the side of the cutter. It is pictured as a plane in our Figure.
2. *Part surface*; this is the surface, again pictured as a plane, on which the bottom or nose of the tool is guided.
3. *Check surface*; this is the surface that stops the forward motion of the tool in the execution of the current command. One might say that this surface "checks" the advance of the tool.



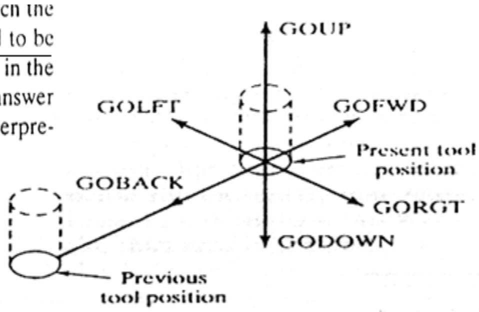
Three surfaces in APT contouring motions that guide the cutting tool.

There are several ways in which the check surface can be used. This is determined by using any of four APT modifier words in the descriptive data of the motion statement. The four modifier words are TO, ON, PAST, and TANTO. As depicted in Figure , the word TO positions the leading edge of the tool in contact with the check surface; ON positions the center of the tool on the check surface; and PAST puts the tool beyond the check surface, so that its trailing edge is in contact with the check surface. The fourth modifier word TANTO is used when the drive surface is tangent to a circular check surface, as in Figure TANTO moves the cutting tool to the point of tangency with the circular surface.





the part programmer must keep in mind the direction from which the tool is coming in the preceding motion command. The programmer must pretend to be riding on top of the tool, as if driving a car. After the tool reaches the check surface in the preceding move, does the next move involve a right turn or left turn or what? The answer to this question is determined by one of the following six motion words, whose interpretations are illustrated in Figure



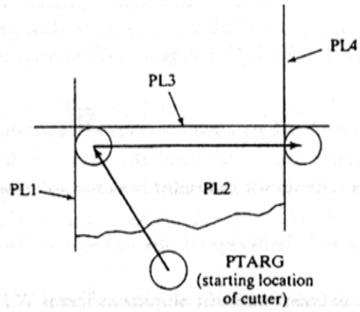
- GOLFT commands the tool to make a left turn relative to the last move.
- GORGT commands the tool to make a right turn relative to the last move.
- GOFWD commands the tool to move forward relative to the last move.
- GOBACK commands the tool to reverse direction relative to the last move.
- GOUP commands the tool to move upward relative to the last move.
- GODOWN commands the tool to move down relative to the last move.

Initialization of APT contouring motion sequence:

With reference to the Figure, the sequence takes the following form:

FROM/PTARG GO/TO, PL1,
TO, PL2, TO, PL3

- The three surfaces included in the GO statement must be specified in the order; (1) drive surface, (2) part surface, and (3) check surface.



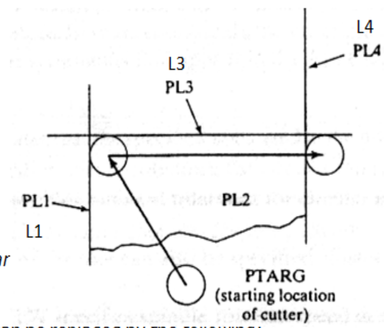
- Note that GO/TO is not the same as the GOTO command. GOTO is used only for PTP motions. The GO/ command is used to initialize a sequence of contouring motions and may take alternative forms such as GO/ON, GO/TO, or GO/PAST.

It is not necessary to redefine the part surface in every motion command after it has been initially defined as long as it remains the same in subsequent commands;

GORGT/PL3, PAST, PL4

* In engineering drawing, the sides of the part appear as lines, although they are three-dimensional surfaces on the physical part. In cases like this, it is more convenient for the programmer to define the part profile in terms of lines and circles rather than planes and cylinders.

* APT language system allows this because in APT, *lines are treated as planes* and *circles ar* perpendicular to the *x*-*y* plane.



Hence, the planes around the part outline can be replaced by lines (L1, L3, and L4). The commandscan be replaced by the following;

FROM/PTARG GO/TO, L1, TO,
PL2, TO, L3
GORGT/L3, PAST, L4

- Plane PL2 has not been converted to a line. As the "part surface" in the motion statement, it mustmaintain its status as a plane parallel to the *x*- and *y*-axes.

Postprocessor and Auxiliary statements

Postprocessor statements control the operation of the machine tool and play a supporting role in generating the tool path. Such statements are used to define *cutter size*, *specify speeds and feeds*, *turn coolant flow on and off*, and *control other features of the m/c tool*. The general form of the postprocessor statement is:

POSTPROCESSOR COMMAND/descriptive data

In some commands, the descriptive data is omitted. Some examples of the postprocessor statements are the following:

- UNITS/MM indicates that the specified units used in the program are INCHES or MM.
- INTOL/0.02 specifies inward tolerance for circular interpolation.
- OUTTOL/0.02 specifies outward tolerance for circular interpolation.
- CUTTER/20.0 defines cutter diameter for tool path offset calculations; the length and other dimensions of the tool can also be specified, if necessary, for three-dimensional machining.
- SPINDL/1000, CLW specifies spindle rotation speed in revolutions per minute. Either CLW (clockwise) or CCLW (counterclockwise) can be specified.

- SPINDL/OFF stops spindle rotation.
- FEDRAT/40, IPM specifies feed rate in millimeters per minute or inches per minute. Minor words IPM or IPR are used to indicate whether the feed rate is units per minute or units per revolution of the cutter, where the units are specified as inches or millimeters in a preceding UNITS statement.
- RAPID engages rapid traverse (high feed rate) for next move(s).
- COOLNT/FLOOD turns cutting fluid on.
- LOADTL/01 used with automatic toolchangers to identify which cutting tool should be loaded into the spindle.
- DELAY/30 temporarily stops the machine tool for a period specified in seconds.

Auxiliary statements are used to identify the part program, specify which postprocessor to use, insert remarks into the program, and so on. Some examples are following:

- PARTNO is the first statement in an APT program, used to identify the program; for example,

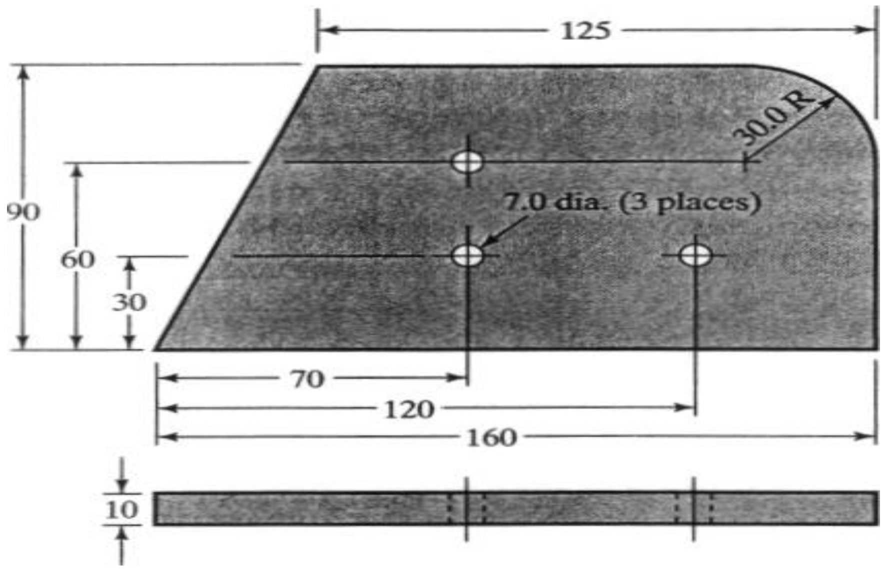
  PARTNO SAMPLE PART NUMBER ONE

- MACHIN/ permits the part programmer to specify the postprocessor, which in effect specifies the machine tool.
- CLPRNT stands for "cutter location print," which is used to print out the cutter location sequence.
- REMARK is used to insert explanatory comments into the program that are not interpreted or processed by the APT processor.

EXP NO: 14 **DATE:**

Mill the shown shape (APT LANGUAGE)

The drill will be operated at a feed of 0.05 mm/rev and a spindle speed of 1000 rev/min (corresponding to a surface speed of about 0.37 m/sec, which is slow for the aluminum work material). At the beginning of the job, the drill point will be positioned at a target point located at $x = 0$, $y = -50$, and $z = +10$ (axis units are millimeters). The program begins with the tool positioned at this target point.



Solution:

Cutter diameter data has been manually entered into offset register 05. At the beginning of the job, the cutter will be positioned so that its center tip is at a target point located at $x = 0$, $y = -50$, and $z = +10$. The program begins with the tool positioned at this location.

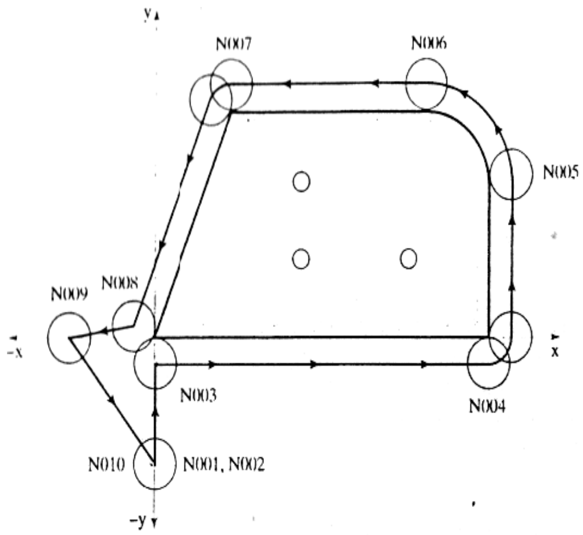
Feed = 50 mm/min.,

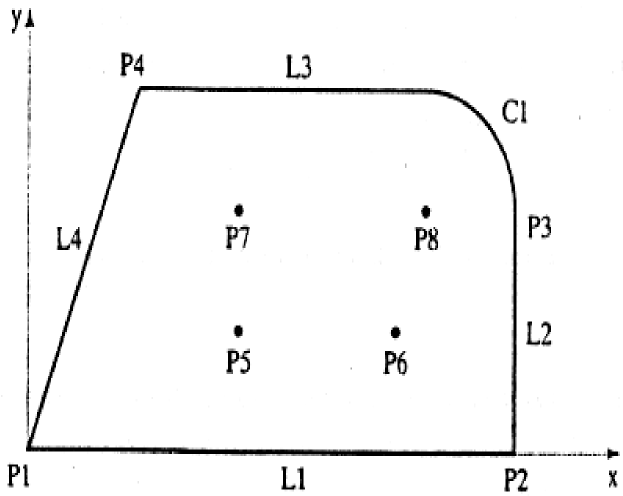
Speed = 1000 rev/min.,

Cutter diam. = 20 mm.

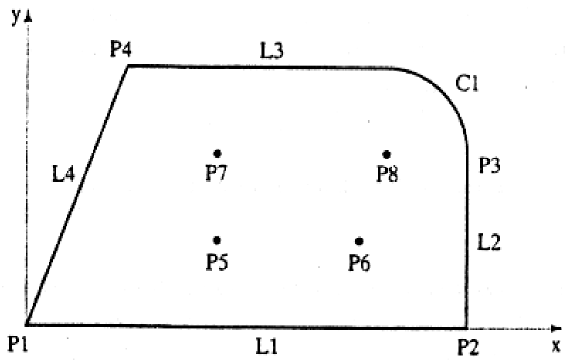
PARTNO SAMPLE PART MILLING OPERATION
MACHIN/MILLING,02
CLPRNT
UNITS/MM
CUTTER/20.0
REMARK Part geometry, Points and Lines are defined 25 mm below part top surface.
PTARG = POINT/0, -50.0, 10.0
P1 = POINT/0, 0, -25.0
P2 = POINT/160.0, 0, -25.0
P3 = POINT/160.0, 60.0, -25.0
P4 = POINT/35.0, 90.0, -25.0

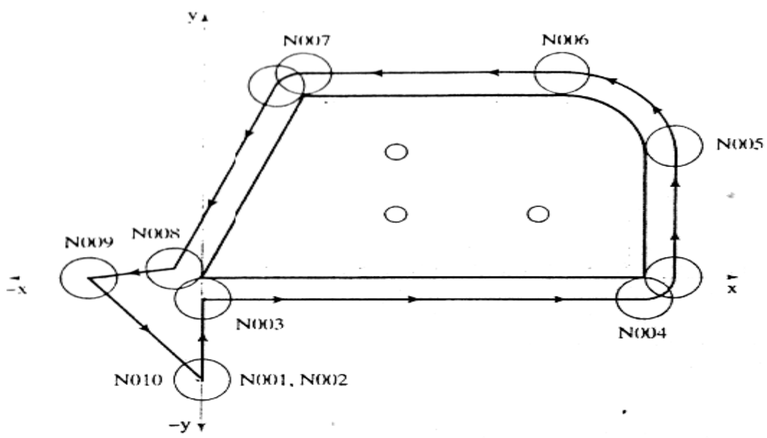
```
P8 = POINT/130.0, 60.0, -25.0
L1 = LINE/P1, P2
L2 = LINE/P2, P3
C1 = CIRCLE/CENTER, P8, RADIUS, 30.0
```






```
L3 = LINE/P4, LEFT, TANTO, C1
L4 = LINE/P4, P1
PL1 = PLANE/P1, P2, P4
REMARK Milling cutter motion statements.
 FROM/PTARG
SPINDL/1000, CLW FEEDRAT/50, IPM
GO/TO, L1, TO, PL1, ON, L4
GORGT/L1, PAST, L2
GOLFT/L2, TANTO, C1
GOFWD/C1, PAST, L3
GOFWD/L3, PAST, L4
GOLFT/L4, PAST, L1
RAPID
GOTO/PTARG
SPINDL/OFF
FINI
```





RESULT:

EXP NO: 15 **DATE:**

Mill the shown shape (APT LANGUAGE)

Example 15.1 Write an APT part programming to cut the profile as shown in Figure 15.17 with the cutter is 6.5 mm dia, cutting speed of 900 rpm and feed rate of 7.5 mm/min.

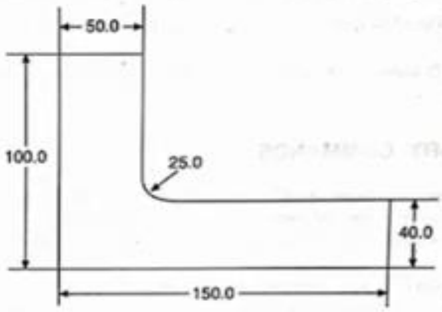


Fig. 15.17 Angular plate.

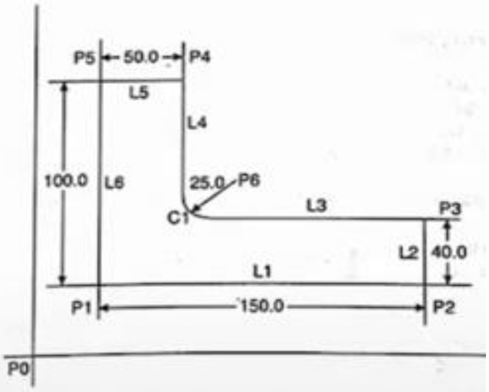


Fig. 15.18 Dimensioning in Cartesian coordinates.

Solution

The component is labelled to define the geometry as shown in Figure 15.18.

```
PARTNO/01
MACHIN/MILL1
INTOL/0.001
OUTTOL/0.001
CUTTER/6.5
SPINDL/900
FEDRAT/7.5
COOLNT/ON
P0 = POINT/-25.0, -25.0,0.0
P1 = POINT/0.0, 0.0, 0.0
P2 = POINT/150.0, 0.0, 0.0
P3 = POINT/150.0, 40.0, 0.0
P4 = POINT/50.0, 100.0, 0.0
P5 = POINT/0.0, 100.0, 0.0
P6 = POINT/75.0, 65.0,0.0
L1 = LINE/P1, P2
L2 = LINE/P2, P3
C1 = CIRCLE/CENTRE, P6, RADIUS, 25.0
L3 = LINE/P3, LEFT, TANTO, C1
L4 = LINE/P3, RIGHT, TANTO, C1
L5 = LINE/P4, P5
L1 = LINE/P5, P1
PL1 = PLANE/P1, P2, P5
FROM/P0
RAPID
GO/TO, L1, TO, PL1, TO, L6
GO/TO P1
GORGT/L1, PAST, L2
GORGT/L2, PAST, L3
GOLFT/L3, TANTO, C1
GOFWD/C1, PAST, L4
GOFWD/L4, PAST, L5
GOLFT/L5, PAST, L6
GOLFT/L6, PAST, L1
GOTO/P0
COOLNT/OFF
SPINDL/OFF
FINI
END
```

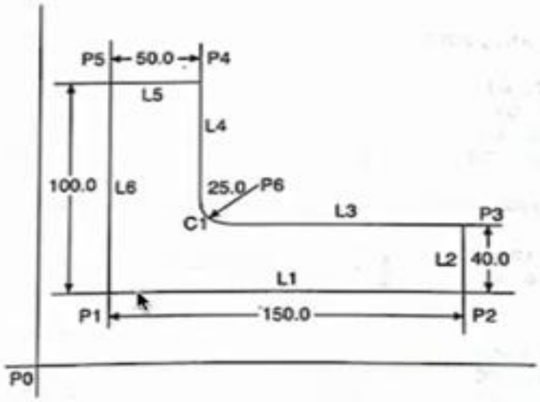


Fig. 15.18 Dimensioning in Cartesian coordinates.

RESULT:

EXP NO: 16 **DATE:**

APT part program to cut the profile in CNC Lathe

Write an APT part programming to cut the profile as shown in figure 1 with the cutter is 6.5 mm diameter, cutting speed of 900rpm and feed rate of 7.5mm/min

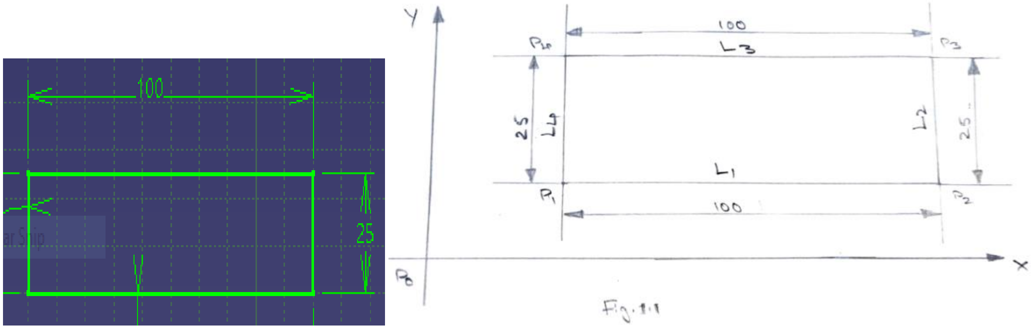


Fig.1

Solution :
The component is labeled to define the geometry as shown in figure.1.1

```
PART NO/01
INTOL/0.001
OUTTOL/0.001
CUTTER/6.5
SPINDL/900
FEDRAT/7.5
COOLANT/ON
P0=POINT/-25.0, -25.0, 0.0
P1=POINT/0.0, 0.0, 0.0
P2=POINT/100.0, 0.0, 0.0
P3=POINT/100.0,  25.0,  0.0
P4=POINT/100.0, 25.0, 100.0
L1=LINE/P1, P2
L2=LINE/P2, P3
L3=LINE/P3, P4
L4=LINE/P4’ P1
PL1=PLANE/P1, P2, P3, P4
FROM/P0
RAPID
GO/TO L1, TO, PL1, TO L4
GO/TO P1
GORGT/ L1, PAST, L2
GORGT/ L2, PAST, L3
GOLFT/L3, PAST, L4
GOLFT/L4, PAST, L1
GOTO/PO
COOLANT/OFF
SPINDLE/OFF
FINI
END
```

RESULT:

EXP NO: 17 **DATE:**

APT part program to cut the profile in CNC Lathe

Write an APT part programming to cut the profile as shown in figure 1 with the cutter is 6.5 mm diameter, cutting speed of 900rpm and feed rate of 7.5mm/min

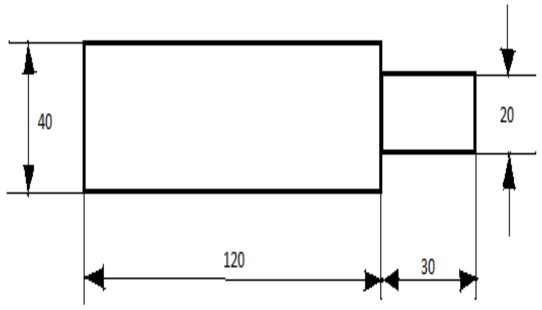


FIG.1

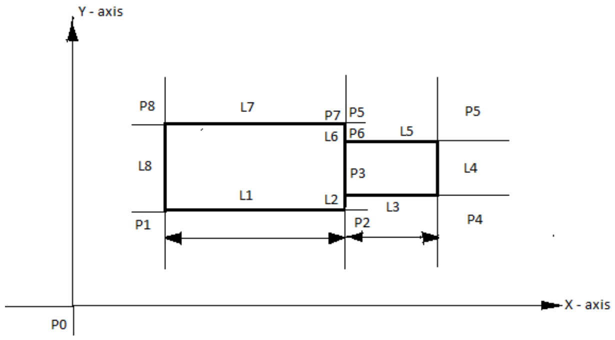


FIG-1.1

Solution :

The component is labeled to define the geometry as shown in figure.1.1

PART NO/01
INTOL/0.001
OUTTOL/0.001
CUTTER/6.5
SPINDL/900
FEDRAT/7.5
COOLANT/ON
P0=POINT/-25.0, -25.0, 0.0
P1=POINT/0.0, 0.0, 0.0
P2=POINT/120.0, 0.0, 0.0
P3=POINT/120.0, 10.0, 0.0
P4=POINT/120.0, 10.0, 30.0
P5=POINT/20.0, 10.0, 0.0
P6=POINT/10.0, 30.0, 0.0
P7=POINT/30.0, 10.0, 0.0
P8=POINT/120.0, 0.0, 30.0
L1=LINE/P1, P2
L2=LINE/P2, P3
L3=LINE/P3, P4
L4=LINE/P4’ P5
L5=LINE/P5’ P6
L6=LINE/P6’ P7
L7=LINE/P7’ P8
L8=LINE/P8’ P1
PL1=PLANE/P1, P2, P3, P4
FROM/P0
RAPID
GO/TO L1, TO, PL1, TO L8
GO/TO P1
GORGT/ L1, PAST, L2
GORGT/ L2, PAST, L3

GORGT/ L3, PAST, L4
GORGT/ L4, PAST, L5
GOLFT/ L5, PAST, L6
GOLFT/ L6, PAST, L7
GOLFT/ L7, PAST, L8
GOLFT/L4, PAST, L1
GOTO/PO
COOLANT/OFF
SPINDLE/OFF
FINI
END

RESULT:

5- & 6-Axis Articulated Robots

Five and six axis articulated robots are more adaptable and offer more flexibility than a SCARA. These robots also have outstanding speed and repeatability, compared to their SCARA counterparts. They are ideal for machine tending, knitting, packaging, inspection, and other applications where off-vertical material handling is needed. They can be mounted conventionally, upside down, or on the wall. ANSI and CE safety compliance allows global deployment.

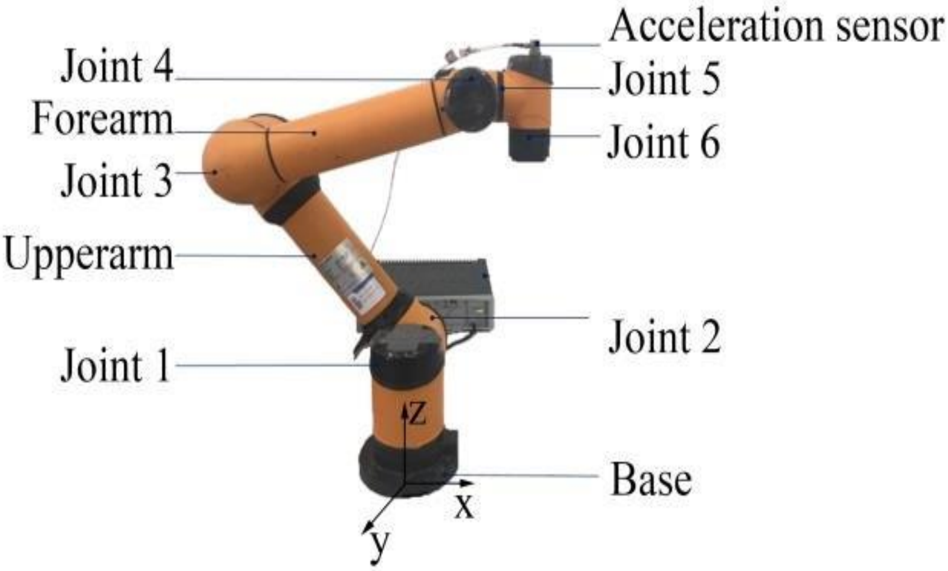
The HS-Series compact, four-axis SCARA robots offer the highest speed and repeatability in their class. Although these small industrial robots have a compact, space-saving design, they have a large load-handling capability, with a maximum payload of 5 kg. In addition, their large, 0.1 kgm⊃2; maximum allowable moment of inertia enables a wide range of end effectors and applications. All HS-Series robots are available in dust- and mistproof (IP65) or ISO 4 (class 10) clean room configurations.

6-Axis Articulated Robot Uses & Applications

- Arc welding.
- Material handling.
- Assembly.
- Part transfer.
- Pick and place,
- Packaging.
- Machine loading.
- Palletizing

5 axis machines uses

Using 5-axis machines lets you machine the workpiece from all sides — no manual rotation required. With 5-axis machining, you'll have higher yields, greater accuracy, and increased freedom of movement, as well as the ability to manufacture larger parts faster.



PICK AND PLACE ROBOTS

Picks a part Job piece from one location and moves it to another location.

- Part may be presented by mechanical feeding device or conveyor.
- For simple case, robot needs only 2 degree of freedom. One to lift and drop, other is to move between the pickup point and drop point.
- Has to track a moving pickup point or drop onto a moving conveyor. Either case requires sophisticated system inter-locks system.
- When different objects are handled, the robot must distinguish between them.
- To handle this issue, sensor system which executes the respective module must be used

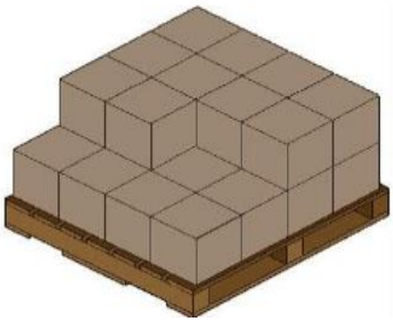
PALLETIZING AND RELATED OPERATIONS

Large amount of containers (cartons) are placed on a pallet (wooden platform) and then it is handled by fork-lift trucks or conveyors.

- These are very convenient method for shipping large quantity of products.

- Computer controlled robot, using high-level programming language is suitable for this operation.

Other Operations are:



Depalletizing Operations [Reversal of palletizing operations.

- Inserting parts into cartons / conveyor.
- Stacking and un-stacking operations.

The robot maybe called to load/unload different pallets differently like (which may): vary in size. Different products loaded onto pallets. Differences in numbers and combinations of cartons to different customers. Bar codes are used to solve the identification problem for depalletizing the optical reader system can be used. Differences in loading and unloading can be accomplished by loading the respective subordinate in to controller. Usually the palletizing operation is much complex then Depalletizing because of customer's orders (different boxes of different sizes has to be delivered to different customers.

Programming Languages

Different languages can be used for robot programming, and their purpose is to instruct the robot in how to perform these actions. Most robot languages implemented today are a combination of textual and teach-pendant programming. Some of the languages that have been developed are:

WAVE
VAL
AML
RAIL
MCL
TL- 10
IRL
PLAW
SINGLA
VAL II
VAL II

It is one of the most commonly used and easily learned languages.

- It is a computer-based control system and language designed for the industrial
robots at Unimation, Inc.

-The VAL II instructions are clear, concise, and generally self explanatory
- The language is easily learned.
-VAL II computes a continuous trajectory that permits complex motions to be executed quickly, with efficient use of system memory and reduction in overall system complexity.
-The VAL if system continuously generates robot commands and can simultaneously interact with a human operator, permitting on-line program generation and modification.
-A convenient feature of VAL If is the ability to use libraries of manipulation routines.ϖ Thus, complex operations can be easily and quickly programmed by combining predefined subtasks.

In the following, we describe the most commonly used VAL II commands.

MOVE P1	This causes the robot to move in joint interpolation motion from its present location to location P1.
MOVES P1	Here, the suffix S stands for straight-line interpolation motion.
MOVE P1 VIA P2	This command instructs the robot to move from its present location to P1, passing through location P2.
APPRO P1 10	This command instructs the robot to move near to the location P1 but offset from the location along the tool z-axis in the negative direction (above the part) by a distance of 10
DEPART 15	Similar to APPRO, this instructs the robot to depart by a specified distance (15 mm) from its present position. The APPRO and DEPART commands can be modified to use straight-line interpolation by adding the suffix S.

DEFINE PATH 1= PATH(P1,P2,P3,P5) MOVE PATH 1	The first command (DEFTNE) defines a path that consists of series of locations P1, P2, P3, and P5 (all previously defined). The second command (MOVE) instructs the robot to move through these points in joint interpolation. A MOVES command can be used to get straight-line interpolation
ABOVE & BELOW	These commands instruct the elbow of the robot to point up and down, respectively.
SPEED 50 IPS	This indicates that the speed of the end- effector during program execution should be 50 inch per second (in./s).
SPEED 75	This instructs the robot to operate at 75% of normal speed.
OPEN	Instructs end effector to open during the execution of the next motion.
CLOSE	Instructs the end-effector to close during the execution of the next motion.
OPENI	Causes the action to occur immediately.
CLOSEI	Causes the action to occur immediately

If a gripper is controlled using a servo-mechanism, the following commands may also be available.	
CLOSE 40 MM	The width of finger opening should be 40 mm.
CLOSE 3.0 LB	This causes 3 lb of gripping force to be applied against the part.
GRASP 10, 100	This statement causes the gripper to close immediately and checks whether the final opening is less than the specified amount of 10 mm. If it is, the program branches to statement 100 in the program
SIGNAL 4 ON	This allows the signal from output port 4 to be turned on at one point in the program and
SIGNAL 4 OFF	Turned off at another point in the program.
WAIT10 ON	This command makes the robot wait to get the signal on line 10 so that the device is on there.

Logarithmic, exponential, and similar functions:

The following relational and logical operators are also available.

EQ	Equal to
NE	Not equal to
GT	Greater than
GE	Greater than or equal to
LT	Less than
LE	Less than or equal to
AND	Logical AND operator
OR	Logical OR
NOT	Logical complement

IF (Logical expression) THEN
(Group of instructions)
ELSE
(Group of instructions)
END
	If the logical expression is true, the group of statements between THEN and ELSE is executed. If the logical expression is false, the group of statements between ELSE and END is executed. The program continues after the END statement.
	The group of instructions after the DO statement makes a logical set whose variable value would affect the logical expression with the UNTIL

DO (Group of instructions) UNTIL(Logical expression)	statement. After every execution of the group of instructions, the logical expression is valuated. If the result is false, the DO loop is executed again; if the result is true, the program continues.

TYPE "text" This statement displays the message given in the quotation marks. The statement is also used to display output information on the terminal.

PROMPT "text", INDEX This statement displays the message given in the quotation marks on the terminal. Then the system waits for the input value, which is to be assigned to the variable INDEX.

In most real-life problems, program sequence control is required. The following statements are used to control logic flow in the program.

GOTO 10This command causes an unconditional branch to statement 10.

SUBROUTINES can also be written and called in VAL II programs. Monitor mode commands are used for functions such as entering locations and systems supervision, data processing, and communications. Some of the commonly used monitor mode commands are as follows:

EDIT (Program name) This makes it possible to edit the existing program or to create a new program by the specified program name.

EXIT This command stores the program in controller memory and quits the edit mode.

STORE (Program name) This allows the program to be stored on a specified device.

READ (Program name) Reads a file from storage memory to robot controller.

LIST (Program name) Displays program on monitor.

PRINT (Program name) Provides hard copy.

DIRECTORY Provides a listing of the program names that are stored either in the controller memory or on the disk.

ERASE (Program name) Deletes the specified program from memory or storage.

EXECUTE (Program name) Makes the robot execute the specified program. It may be abbreviated as EX or EXEC.

ABORT Stops the robot motion during execution.

STOP The same as abort.

EXP NO: 18 **DATE:**

VAL II to command a PUMA robot

Develop a program in VAL II to command a PUMA robot to unload a cylindrical part of 10 mm diameter from machine 1 positioned at point P1 and load the part on machine 2 positioned at P2. The speed of robot motion is 40 in./s. However, because of safety precautions, the speed is reduced to 10 in./s while moving to a machine for an unloading or loading operation.

Solution

1. SIGNAL 5
2. SPEED 40 IPS
3. OPEN 100
4. APPRO PI, 50
5. SPEED 10 IPS
6. MOVE PI
7. GRASP 10, 100
8. DEPART P1, 50
9. SPEED 40 IPS
10. APPRO P2, 50
11. SPEED 10 IPS
12. MOVEP2
13. BELOW
14. OPENI 100
15. ABOVE
16. DEPART P2, 50
17. STOP

RESULT:

EXP NO: 19 **DATE:**

Pick and Place activity

Program for Pick and Place activity:

STATEMENT	**STATEMENT DESCRIPTION**
BRANCH PICK	The branch of program indicating part picks.
MOVE INTER	Move to an intermediate position chute.
WAIT 12	Wait for an incoming part to chute.
SIGNAL 5	Open gripper fingers (Sensor control)
MOVE PICK-UP	Move gripper and Pick-up the object.
SIGNAL 6	Close the gripper to grasp the object
MOVE INTER	Depart to intermediate position above chute.
END BRANCH	End of pick-up activity.
BRANCH PLACE	Start of placing activity
MOVE Z (-50)	Position part and gripper above the pallet
SIGNAL 5	Open gripper to release the part
MOVE Z (50)	Depart from the place point.
END BRANCH	

RESULT: Thus, we have studied how to perform Pick and Place operation.