Week 04	• Explain Programming sequence and format – • Absolute and Incremental System • Explain G codes and M codes • Explain Linear interpolation and Circular Interpolation

Steps for preparing a part program:

(a) Determine the startup procedure, which includes the extraction of dimensional data from part drawings and data regarding surface quality requirements on the machined component.

(b) Select the tool and determine the tool offset.

(c) Set up the zero position for the workpiece.

(d) Select the speed and rotation of the spindle.

(e) Set up the tool motions according to the profile required.

(f) Return the cutting tool to the reference point after completion of work.

(g) End the program by stopping the spindle and coolant.

Methods of preparing part programming can be of two types:

(a) Manual part programming, and ***(b) Computer aided part programming.***

Manual part programming: The programmer first prepares the program manuscript in a standard format.

Manuscripts are typed with a device known as flexo writer, which is also used to type the program instructions. After the program is typed, the punched tape is prepared on the flexo writer. Complex shaped components require tedious calculations. This type of programming is carried out for simple machining parts produced on point-to-point machine tool.

To be able to create a part program manually, need the following information:

(a) Knowledge about various manufacturing processes and machines.

(b) Sequence of operations to be performed for a given component.

(c) Knowledge of selection of the cutting parameters.

(d) Editing the part program according to the design changes.

(e) Knowledge about the codes and functions used in part programs.

Types of CNC codes

Preparatory function (G – codes): The preparatory function instructs the machine tool to get prepared for the operation to follow. It is preset function associated with the movement of machine axis as per geometry of the finished work piece. Preparatory function is represented by two digits preceded by word 'G' i.e. G00, G01, G90 etc.

Example: code G00 is used for rapid traverse.

Miscellaneous codes (M – codes): The M code is used to specify certain miscellaneous or auxiliary functions such as spindle start rotation, spindle stop, coolant on/off etc. The miscellaneous functions are those functions which do not relate the dimensional movement of the machine. Example: M03 Spindle rotation clockwise

Modal and Non-modal codes: In principle, all codes are either modal or non-modal. **Modal code** stays in effect until cancelled by another code in the same group. The control remembers modal codes. This gives the programmer an opportunity to save programming time.

Non-modal code stays in effect only for the block in which it is programmed. Afterwards, its function is turned off automatically. For instance G04 is a non-modal code to program a dwell. The control does not memorize the non-modal code, so it is called as one shot codes.

Some of important codes are given as under with their functions:

G-Codes (Preparatory Functions)		M-Codes (Miscellaneous Functions)	
Code	**Function**	**Code**	**Function**
G00	Rapid positioning	M02	Program end
G01	Linear interpolation	M03	Spindle rotation on-clockwise
G02	Circular interpolation clockwise (CW)	M04	Spindle rotation on-counterclockwise
G03	Circular interpolation counterclockwise(CCW)	M05	Spindle stop
G04	Dwell for specific time	M06	Tool change
G17	Selection of X-Y plane	M08	Coolant on
G18	Selection of Z-X plane	M09	Coolant off
G19	Selection of Y-Z plane	M30	Program stop, reset to start
G33	Thread cutting	M98	Transfer to subprogram(call subroutine)
G40	Cutter compensation cancel	M99	End of subprogram
G41	Cutter compensation left		
G42	Cutter compensation right		
G54	Settable zero offset		
G70	Inch Dimensioning(G20 can also be used)		
G71	Metric Dimensioning(mm)(G21 can also be used)		
G90	Absolute programming	G80	Canned cycle cancel
G91	Incremental programming	G81	Drill canned cycle
G94	Feed rate mm/min.(Mill)	G94	Feed rate mm/min.(Lathe)(must use with canned cycle
G95	Feed rate mm/rev..(Mill)	G95	Feed rate mm/rev.(Lathe))(must use with canned cycle

Programming Sequence -

Programming Sequence: CNC part programming involves a series of coded instructions that direct the operations of a CNC machine. This sequence typically includes:

- Program Start and Initialization: Setting up initial conditions like coordinate system, units, and safety measures.
- Tool Selection and Spindle Control: Specifying which tool to use and controlling spindle speed and direction.
- Positioning and Motion Commands: Moving the tool to desired locations using linear or circular interpolation.
- Machining Operations: Executing specific machining tasks like drilling, milling, or turning.
- Tool Changes (if necessary): Swapping tools during the machining process.
- Program End: Returning the machine to a safe state and terminating the program.

Programming format:

Tape Programming Format

Both EIA and ISO use three types of formats for compiling of NC data into suitable blocks of information with slight difference.

Word Address Format

This type of tape format uses alphabets called address, identifying the function of numerical data followed. This format is used by most of the NC machines, also called variable block format. A typical instruction block will be as below :

N20 G00 X1.200 Y.100 F325 S1000 T03 M09 <EOB>

or N20 G00 X1.200 Y.100 F325 S1000 T03 M09;

The MCU uses this alphabet for addressing a memory location in it.

Tab Sequential Format

Here the alphabets are replaced by a Tab code, which is inserted between two words. The MCU reads the first Tab and stores the data in the first location then the second word is recognized by reading the record Tab. A typical Tab sequential instruction block will be as below :

>20 >00 >1.200 >.100 >325 >1000 >03 >09

Fixed Block Format

In fixed block format no letter address of Tab code are used and none of words can be omitted. The main advantage of this format is that the whole instruction block can be read at the same instant, instead of reading character by character. This format can only be used for positioning work only. A typical fixed block

instruction block will be as below:

20 00 1.200 .100 325 1000 03 09 <EOB>

3.5 Tape Programming Format

Standards were developed for the type and order of informations on punched tapes. Both EIA and ISO use three types of formats for compiling of NC data into suitable blocks of information with slight difference :

- Word address format
- Tab sequential format
- Fixed block format.

3.5.1 Word Address Format

This type of tape format *uses alphabets called address, identifying the function of numerical data followed.* Alphabatic characters such as N, G, XYZ, S, F, T, M have standard meanings. The MCU uses this alphabet for addressing a memory location in it. In this format, the block of information may be of any length and the words used need not be in any particular sequence because the letter address, will identify the corresponding word. To align and give space in the words a TAB key code is punched between words. If a word data is unchanged, then it need not be repeated in the successive block. This format is used by most of the NC machines, also called variable-block format sometimes. In this format, a typical instruction block will be as under :

N070 G81 X05764 Y04750 F475 S1000 T05 M08 <EOB>

In this block N, G, X, Y, F, S, T, M & EOB have their usual meanings as explained in details previously.

3.5.2 Tab Sequential Format

In Tab Sequential Format, the words are given in the sequential order as explained previously. But the *alphabets are replaced by a Tab Code which is inserted between two words.*

The MCU reads the first Tab and stores the data in the first location then the second word is recognised by reading the Record Tab. If the words are unchanged in next block then simply giving one more tab will indicate the word missing from the block. The end of block is indicated by 'EOB' code. This format is also called interchangable format sometimes. The system can be understood by the following examples.

— Suppose we have to give an instruction or block like :

N070 G81 X05764 Y04750 F475 T05 M08 <EOB>

then in Tab Sequential Format we have to give like :

>070 >81 >05764 >04750 >475 >05 >08

Similarly, if we need to give next instruction having same X coordinates then we give instruction as (only Y-axis and feed rate have been changed) :

>071	>	>	>05000	>435	>	>

3.5.3 Fixed Block Format

In fixed block format the information is given in a particular sequence and the block must contain a fixed number of characters. No letter address or Tab Code are used and none of words can be omitted. If any information remain the same as in previous block, even then the same is to be repeated in the next one also. In this format the first three digit refer to the sequence number, next two digit to preparatory function, next 5 × 3 digits for X, Y, Z co-ordinates, next four to speed, next three to feed and then next two for tool number and last two for miscellaneous function and one digit for EOB. Different machines have different fixed format widths. For example, an NC lathe may not have digits for Y axis. The two instructions given in previous example may be given as :

070	81	05764	04150	475	05	08	<EOB>

&

071	81	05764	05000	435	05	08	<EOB>

The main **advantage** of this format is that the whole instruction block can be read at the same instant, instead of reading character by character. Hence, it is best suitable for pneumatic, hole sensing, devices of tape reading. But at the same time, this format can only be used for positioning work only because contouring applications would require an extra ordinary length of the punched tape.

Absolute and Incremental System :

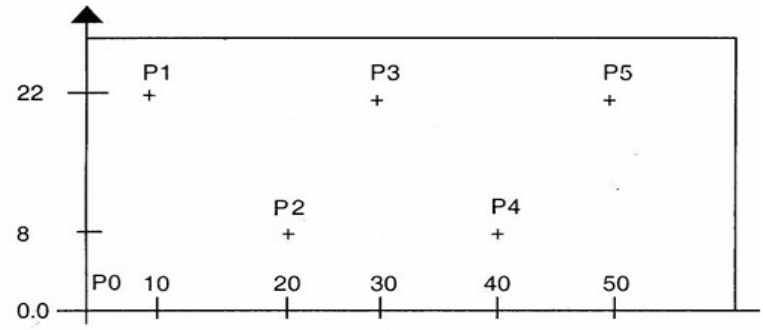
In CNC programming, the absolute and incremental coordinate systems define how the machine tool's position is specified. Absolute programming uses coordinates relative to a fixed zero point on the workpiece, while incremental programming uses coordinates relative to the tool's current position. Essentially, absolute coordinates are like giving directions from a landmark (zero point), while incremental coordinates are like giving directions from your current location.

Absolute Coordinate System:

- In this system, every coordinate is defined with respect to a single, fixed point called the "workpiece zero point" or "program origin".
- To move the tool to a specific location, you specify the X, Y, and Z coordinates (and potentially others for rotary axes) relative to this fixed point.
- For example, if the zero point is at the corner of a part, and you want to move the tool to a point 10mm away in the X direction and 5mm away in the Y direction, you would use the coordinates (10, 5).
- The same coordinates would be used regardless of where the tool is currently located.

Incremental Coordinate System:

- In this system, coordinates are defined as offsets from the tool's current position.
- To move the tool, you specify how much it needs to move in each axis, not its absolute position.
- For example, if the tool is currently at (10, 5) and you want to move it 2mm to the right and 1mm up, you would use the coordinates (2, 1) in incremental mode.
- This means that even if the tool is at a different location, the same incremental coordinates will always move it by the same amount from its current position.



ABSOLUTE CO-ORDINATE SYSTEM

POINT	X CO-ORDINATE	Y CO-ORDINATE
PO	0	0
P1	10	22
P2	20	8
P3	30	22
P4	40	8
P5	50	22

INCREMENTAL COORDINATE SYSTEM

POINT	X CO-ORDINATE	YCO-ORDINATE
P0	0	0
P1	10	22
P2	10	-14
P3	10	14
P4	10	-14
P5	10	14

Example: Imagine a scenario where the tool starts at (0,0) and needs to move to P1, then to P2, P3, P4, and finally to P5. Then co-ordinate in both co-ordinate system is shown in above diagram.

Absolute co-ordinate system

- **In this system all the co-ordinates of point is measure from origin/datum.**
- **Checking of the programme is very easy in absolute system.**
- **If mistake is made in any value of dimension in particular block, it will affect that dimension only. There is no effect on remaining program.**
- **G90 code is used for the Absolute co-ordinate system.**
- **It is always preferred to write main program in absolute system**

Incremental co-ordinate system

- **In this system all the co-ordinates of point is measure from previous point.**
- **Checking of the programme is very difficult in incremental system.**
- **If mistake is made in any value of dimension in particular block, it will affect that entire program.**
- **G91 code is used for the incremental co-ordinate system.**
- **It is always preferred to write subroutines in Incremental system.**

Linear Interpolation:

Definition: Calculates a straight-line path between a starting and ending point.

Purpose: Used for creating straight lines, chamfers, and simple contours.

Implementation: Involves specifying the coordinates of the start and end points, and the CNC machine calculates the intermediate points to create the straight line.

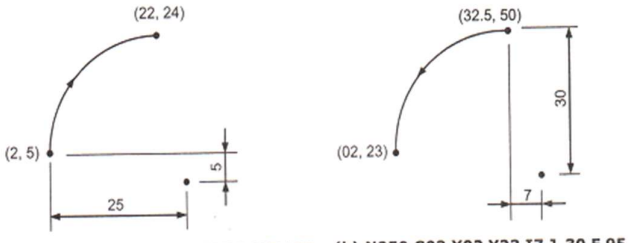
Accuracy: Generally accurate for linear movements, but may result in a faceted appearance when used for curves or arcs.

Circular Interpolation: If the tool is required to move along an arc or circular path, circular interpolation functions G02 or G03 are used to execute the instructions. G02 is used for clockwise interpolation and G03 is for counter clockwise interpolation.

Following information is required for writing an instructional block for circular interpolation:

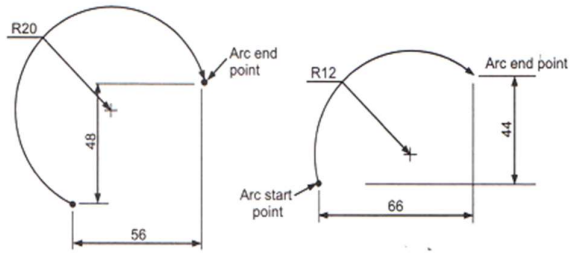
1. G-code depending on direction of interpolation (G02/G03),
2. Co-ordinates of the target point i.e. X, Y and Z co-ordinate for the end point of the arc,
3. Co-ordinates of the centre point of the circular arc which is designated by I, J and K addresses. I is the distance along the X-axis, J along the Y, and K along the Z. This parameter is defined as the vector from the starting point to the center of the arc.

When R is used, a complete 360 degree arc is not possible. X, Y, or Z is required to specify an endpoint different from the starting point. R is the distance from the starting point to the center of the circle. With a positive R, the control will generate a circular path of 180 degrees or less, but to generate a circular path of over 180 degrees, specify a negative R.



(a) N250 G02 X22 Y24 I25 J–5 F 125 (b) N250 G03 X02 Y23 I7 J–30 F 95

Fig. : Use of G02 or G03 Code with I and J Values



(a) N340 G02 X56 Y48 R20 F210 (b) N230 G03 X66 Y44 R–12 F135

Fig. : Use of G02 or G03 Code with R Values