

Different aspects of programming

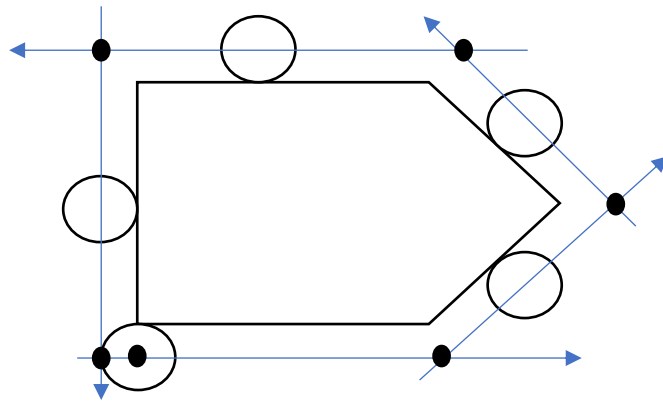
In CNC programming – we come across the following operations

Linear interpolation – already explained

Circular interpolation – already explained

Cutter radius or diameter compensation

When the cutter is cutting out the profile of a part (see fig, the material outside of the profile is to be cut) – we need to guide the cutter through its path. Here – a part in the shape of a 5-sided polygon is shown, with a circular cutter (an end milling cutter seen along its axis). The problem is that it is the



centre of the cutter that we program for. When we ask the cutter to reach a particular point – the centre of the cutter goes and sits at that point. Hence, if we simply give the coordinates of the points defining the corners of the part shown above, the cutter centre would travel through those points and part of the cutter would intrude inside the part profile and the job would be spoiled.

The other option is – the programmer could painstakingly find out the straight lines defining the locus of the centre of the cutter (shown in blue rays in the figure) and find the intersection points of such lines and pass the cutter through those intersection points. However, this would be computationally intensive for the programmer.

Hence, since a computer is anyway available – it is made use of by defining G-codes that would make the computer do the above calculations. The computer would simply have to be told which is the job geometry element to be cut (that is – whether it is side 1 or side 2 etc) and whether the cutter is to be shifted to the right of the movement or to the left.

So, if we want to shift to the right, we write G42 and if to the left, we write G41. We cancel the command by G40.

So we could write

```
T01 M06  
G00 G90 G54 X20 Y20  
G43 H04 Z20  
G42 D03 G01 X100 Y100  
X200
```

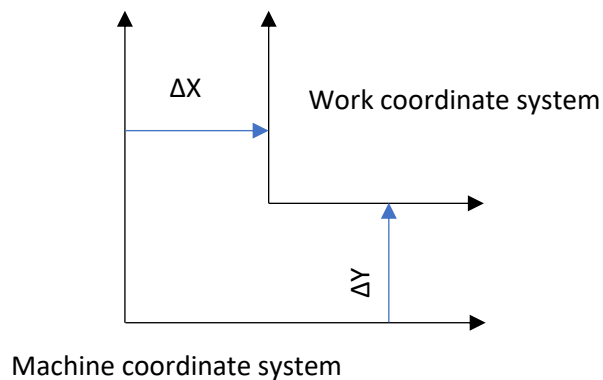
X250 Y150

This reads as follows : Load (M06) the cutter identified as 1 (T01) on to the machine spindle

Load the work coordinate system from memory and move the cutter to the location (20,20) in the work coordinate system. Which is the work coordinate system? It is a coordinate system which you are defining for your own benefit. It is generally different from the machine coordinate system. So – a word or two about coordinate systems used.

The machine coordinate system is a coordinate system which is fixed with the table. Its origin might be chosen as the lower left corner of the table and the Z=0 plane may be on the table surface or the highest point which the spindle nose can reach. This varies from machine to machine. If the table moves – we are moving the coordinate system with respect to the other fixed parts of the machine. Like for example – if the tool is fixed on the spindle, it does not move with the table. So we may say – the tool is moving in the opposite direction relative to the machine coordinate system. The machine coordinate system is the default system or it may be called by the G53 command.

For the ease of our own work – we generally select our own coordinate system for the work piece. Frequently, the work piece is placed on the machine and we select a point on the work piece to be the origin. It may be the origin of the coordinates in the drawing of the part. However, the axes of this coordinate system do not match with those of the machine coordinate system. So we measure the offsets and feed it into the machine memory against a work coordinate system name.



We can also include Z offsets into the work coordinate system so that the top surface of the work piece becomes the Z=0 plane, but some people prefer to pack all Z offsets into tool length offsets – which we will discuss just after this.

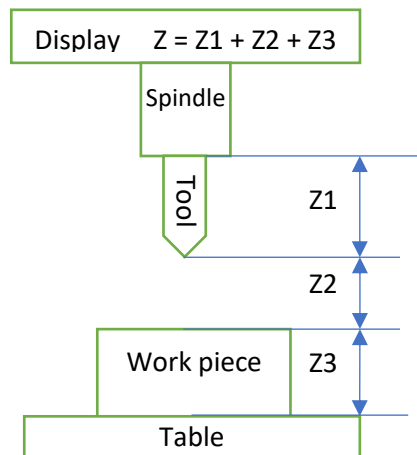
Different work coordinate systems can be defined and these are given names as G54, G55, G56, etc etc

Cutter length compensation or tool length offsets

As the tool / cutter is not an integral part of the machine, the location of the tool tip is not known to the machine. So, the operator can find out the protrusion of the tool tip from the plane of the spindle nose and put in that data into the machine memory. This data is called the tool length offset.

Tool length offset may or may not include work piece height. A convenient way of informing the machine about the tool length and work piece height is to manually touch the tool tip (while the tool

is mounted on the spindle) to the work piece surface and thereby find out the tool protrusion length and workpiece height and put that data into the machine memory against tool length offset.



So in this example – if the table surface is $Z = 0$ in the machine coordinate system, we have a number of options for handling the Z offsets.

We can name $Z3$ as offset in the work coordinate system.

Instead we can club the tool length offset and the work height into the tool length offset.

If we look at the display, say, when the tool is at the topmost point, $Z = Z1 + Z2 + Z3$

$Z1$ = tool protrusion beyond spindle nose

$Z3$ = Job height

If we now touch the tool tip to the job top surface, the reading of display would be

$Z = Z1 + Z3$.

This is put into the memory as tool offset, say in the memory address location H05. So, if there is any command for tool Z movement, say

T02 M06

G54.....(work coordinate system defined for X and Y but suppose Z is not defined, that is, $Z_{offset} = 0$)

G43 H04 Z05

G43 recalls the tool length offset as stored in H04 (which is $Z1 + Z3$) and the machine will move the tool to $Z = Z05 + Z1 + Z3$

This will locate the tool to a position 5 mm above the work surface

More later.....

Mirror imaging

Repeat loop

Subroutine call

Canned cycle

Cycle cancellation