Manufacturing Systems Technology

Prof. Shantanu Bhattacharya

Department of Mechanical Engineering and

Department of Industrial and Production Engineering

Indian Institute of Technology, Kanpur

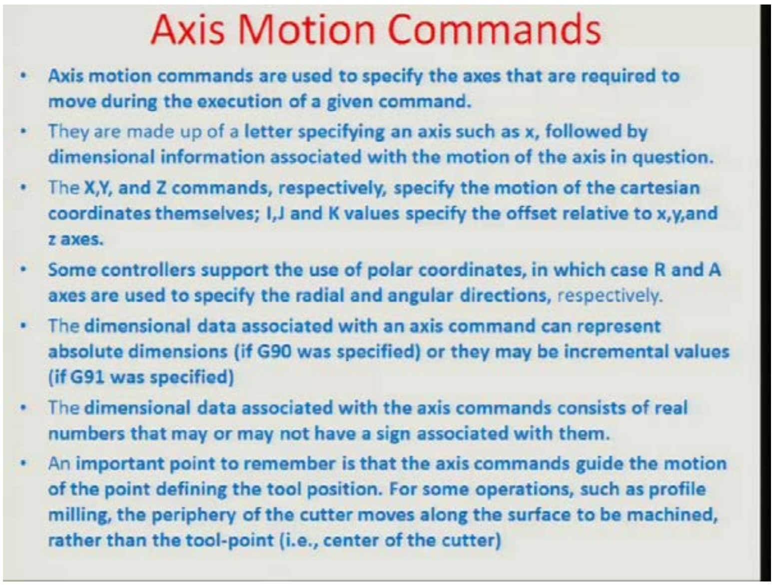
Module- 07

Lecture- 40

Hello and welcome to the Manufacturing Systems Technology module 40. Brief recap of what we did in the last lecture, we talked about the various Caned Cycles and there are certain G codes or preparatory codes which are available for representing the Caned Cycles including the basic operations like drilling, counter boring, so and so forth. We also illustrated some of the basic G codes needed for linear interpolation and rapid point to point positioning system, which otherwise is executed using you know, full speed of both the motors or both the axis.

So, today we will be seeing some other commands which are useful for example, list axis motion commands, and miscellaneous commands, and some things related to feed and speed and so discuss in great details about all these different commands. And then probably towards the end of this module, may be in the next module we will also do one program full program NC part program of a certain situation specific situation which has been provided.

(Refer Slide Time: 02:21)



So, let us look into Axis Motion Commands. So obviously, these are the set of commands which are used to specify the axis that are required to move during the execution of a given commands. So, there are two different possibilities here: one is the absolute positioning, and another is the incremental; and I think we had talked about this in quite detail manner earlier. And so the absolute in the incremental positioning is totally based on whether you basically reading with reference to the origin or you are going from one point that you have already reached to next available point in the array of motion. So, during the execution of the given command, it is basically the stepper motors which operate in the axis start to get traverse.

As you have seen earlier in almost all the n c programs such commands are given though the coordinate which is involved; for example, let say we say capital X 4 3 meaning there by that X is corresponding to a axis motion command and it majorly represents a certain X coordinate if you talking about absolute positioning with reference to the origin. So, in the next line if there is a certain other X which is there, it will automatically give you a relationship between how to move between the previous X and the next X. So, this is represented in more than two lines, the motion is represented in more than two lines such as system.

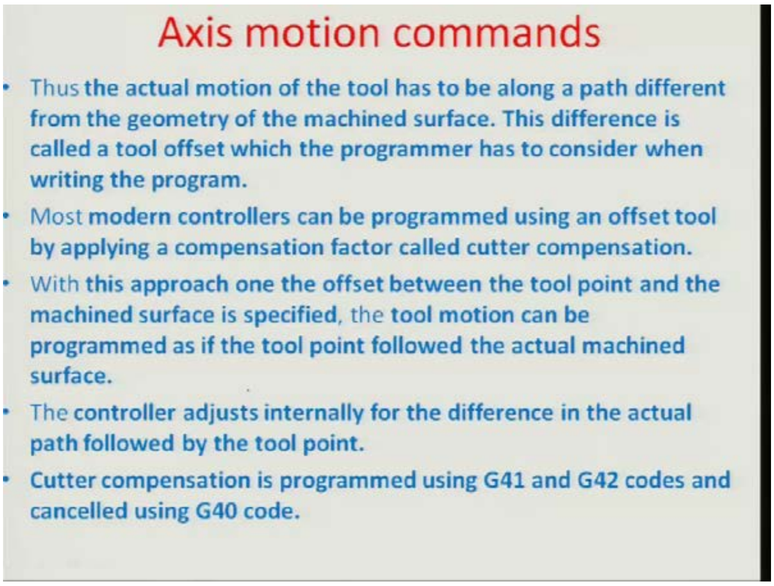
So, the letter specifying X followed by a dimensional information, is associated with the motion of the axis in question and the XYZ command respectively specify the motion of the Cartesian coordinates themselves. And I think had illustrated in great details about the values I, J, K, earlier; were I, J, K, really are the representation of the offset that is there are from the central radius. So, that it can give you bases of the beginning position of an arc, you know which the circular interpolation can happen in either in the clock wise or counter clock wise manner. So, the I, J, and K also sort of values which specify the offset relative to the X, Y, and Z axis. So, you know some controllers are really designed for the polar system of coordinates, were you will have a radial and angular direction.

So, for that this subscript that uses R and A, indicating radially outward and inward and then angularly whatever is the direction; obviously, this same methodology has to be followed regarding the positive and the negative directions as help. Has been discussed earlier in case of clockwise or count clockwise rotation concerned with the particular axis of motion. The dimensional data associated with the axis commands of course, comes from as illustrated earlier some of the preparatory functions and this case it is typically G90 or G91 which gives you in sense of absolute of the incremental dimensioning and we have really maintained again and again that these two are completely different systems of part programming. And one is more convenient for the operator to read and just you know as a function of the or the coordinates that the drawing would have try to just mention about those coordinates and let the controller do the remaining job of calculation in between.

So, the dimensional data associated with the axis command contains mostly real numbers. Real numbers meaning you can have up to decimal places; obviously, the decimal places are represented in terms of the two subscript numbers which are associated with that. For examples if I say plus minus X 4 3, it means it is a real number which can go from 0000.000 to 9999.999. So, I think we are at length discussed about this when we were talking about the controller code or the controller specification which is given by the machine manufacture.

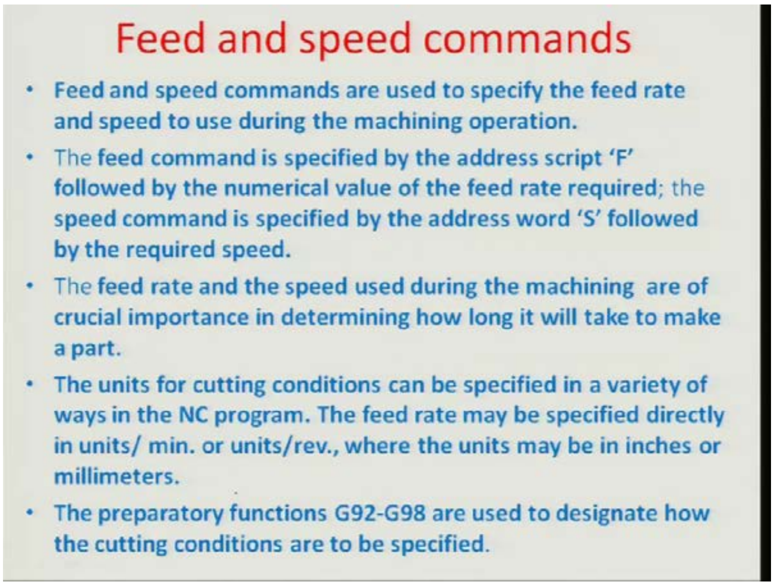
So, an important point to remember is that the axis commands guide the motion of the point defining the tool position, and for some operations such as let us say profile milling you know the problem happens because, now the periphery of the cutter moves along the surface that has to be machined, rather than the tool point. And the tool point in this cases sort of you know the center of the cutter. So, I would say that in such situations when there is circular cutter being used and there is difference between the actual tool point which is more closer to the you know the machined work piece and the radial center of tool. We used ah particular preparatory function which is known as the cutter compensation, which is sometimes given is the part of the controller capability. So, if you are able to just map the tooling center in the way that moves and you have already given the data for the compensation, the cutter compensation value automatically it would describe the tool point based on the program.

(Refer Slide Time: 07:21)



So, the actual motion; obviously, would then be different, quite different then the motion of the tool center. And that is how you can actually program mostly cutter compensation. So, it is programmed using the code G41, G42 and then; obviously, it is the sort of a modal command which once programmed would remain in the system. Every time it would get executed unless you cancel it. So, the cancellation command for that cutter compensation is the code G40. So, these are the three preparatory codes or G codes which are used indicate the cutter compensation part in the program, part program.

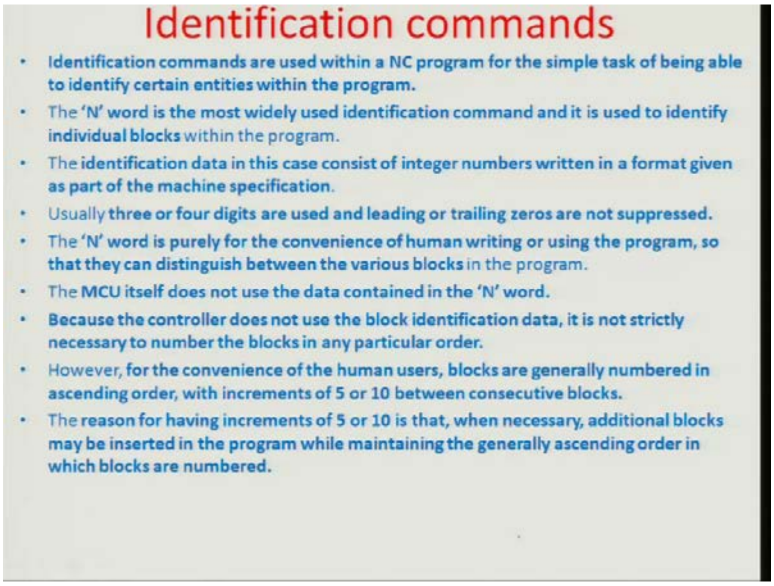
(Refer Slide Time: 08:03)



So, that is all about axis motion commands; obviously, now you are concerned with how the tool moves with respect to the work piece. Once you have given from where to where it should move, it should also be able to describe what rate should be move and so therefore, feed and speed commands come into picture. As the name probably suggest you know feed and speed commands are mostly used to specify the feed rate in the speed used during the machining operation. And the feed command is specify by the address script F, you can see mostly in all the controllers specification the letter F with they with plus minus again and with also. And with numerical value like let say you can define as F plus minus 40. So, again it means that what are the number of digits up to which you can actually programmed. So it can go from 0000 to 9999, different feed values that is what the capability of the controller would them be.

So, similarly when speed command is specified in a similar manner, only thing is the letter S is used in instead of F. And the feed rate in the speed used during machining there of critical importance and that is towards how much time it will take from machining the particular part. So, I think we already know about these terms feed rates or speed, normally feed rate is specified in terms of units per minute or units per revolution; where the units may be described in different systems like the FPS system inches or the metric system millimeters so on and do forth. So, feed rate; obviously, can be either in terms of per unit time or per unit revolution of the work piece. It is very handy as per unit revolution when we talking about a turning center for example. And just need to mention that the preparatory functions G92 to G98 are generally used to designate how the cutting conditions are to be specified. Whether it is a inch per or unit per revolution or unit per minute rate at which the feed has to be specified so and so forth. So, that is about feed and speed commands. We also have in detail mention the identification commands which are really the address for a given block this start of a given block.

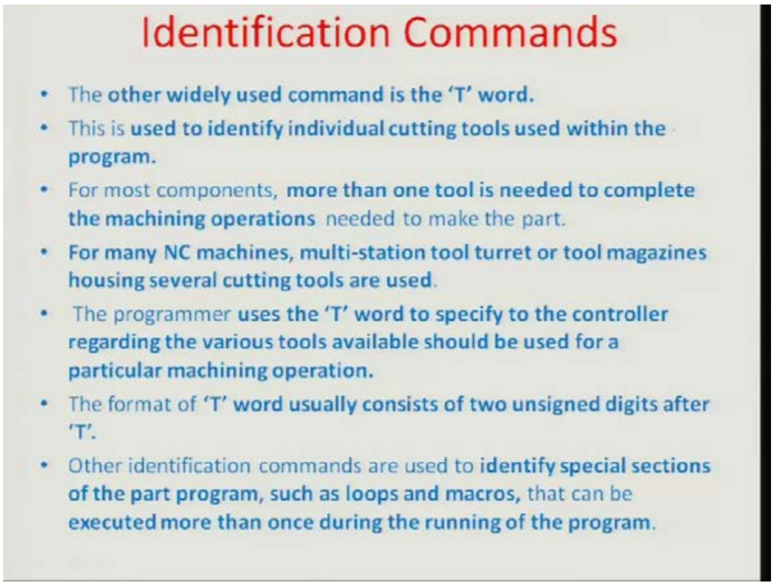
(Refer Slide Time: 10:52)



So, we generally address this with number N, and the N words mostly the most common identification identifier for an individual block. And although the MCU itself does not use the data contained in the M, but it is actually uses indexing tool mostly, so that you can see what comes in sequence first and comes in sequence later. So, normally I think I have illustrated this earlier, but again I want to retreat that for convenience of the human user it is a good practice that you live gaps in between when you talking about the particularly representing the N values.

For example, you can have one command block to have N05 and following block to have N10, so you are skipping 5 steps in between. And this is normally done in all the programs because if the controller has a capability of going from let us say for example, thousands steps. You always try to cover as much possible the in terms of the N numbers as so that later on any modification in the code etcetera that may be under taking can be easily implemented. So, if you have gaps between the first line and the second line in terms of the identifier numbers and those gaps can be filled on later on if you want to modify the code in between somewhere suitably, without really that we have into write whole code again. So, this is users convenience point of view that is various N numbers are illustrated on the different command lines and that is how identifications have been done.

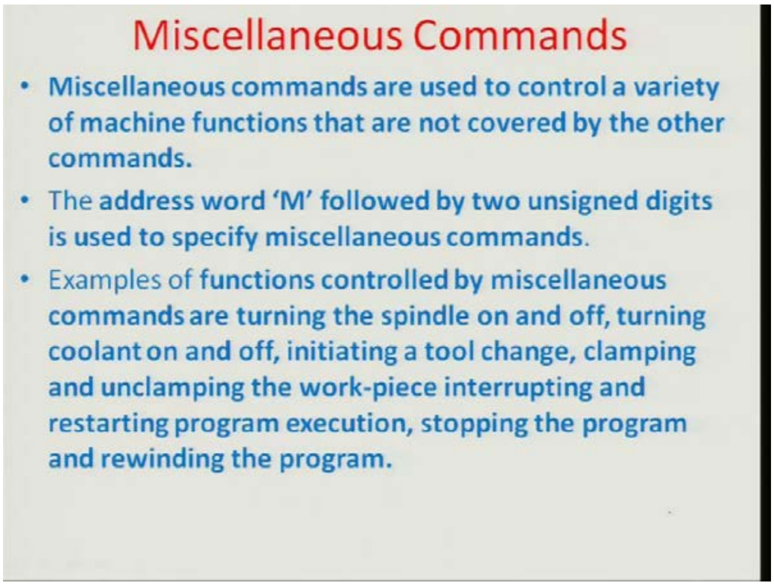
(Refer Slide Time: 12:21)



There are several other identified identification commands like let us say this word T here, which represent the individual cutting tool with the number. So obviously, if there is a tool magazine or turret which contains many tools of tool tips. There is always tendency of tool selection to happen within the, so the first logical step for the machine into be executed in a proper manner should the tool selection in that particular case. So, here for example, T01 would identify the first tool similarly if I am saying that the tool can go up to 2 0, T 2 0. So, it means that you can go from all the way from 00 to 99, so about 100 tools that you can actually be able to designate. So, T 2 0 would be then identifying the capability of that the controller can handle in terms of the number of tools. So, in the same manner you have described the tool selection process, and the other identification commands are used to identify special sections of the part program such as loops and macros. That can be executed more than once during the running of the program.

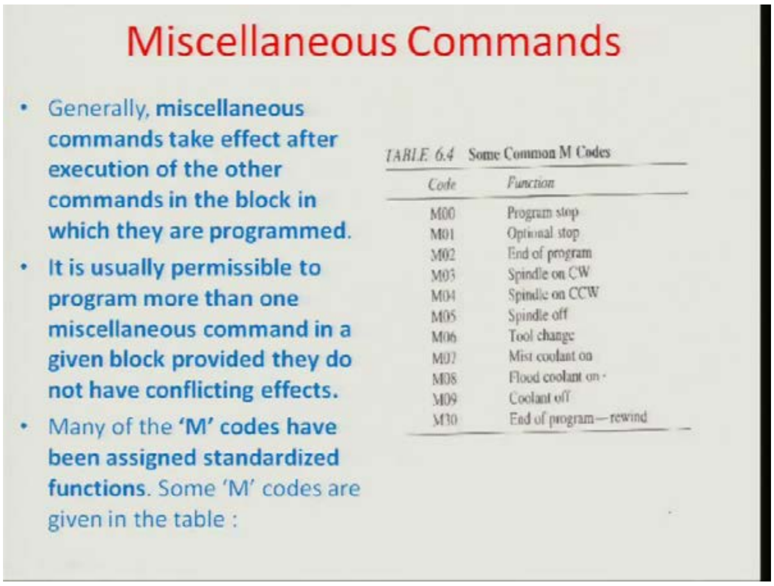
So, we then come to the last available type of command which is the miscellaneous command, is normally used to control variety of machine function that are not covered by the other commands done so for like preparatory commands, axis commands, feed speed commands, the identification commands so and so forth.

(Refer Slide Time: 14:17)



So, here as the name suggest the first letter M of the miscellaneous is basically used for executing this commands. I will just share with you list of such of commands which are available.

(Refer Slide Time: 14:18)



For example, some commonly used M codes could be how to stop a program M00, M01 is optional stop; 02 is the end of the program. You can also let the spindle turn clockwise or counter clockwise based on M03 or M04. You can have a spindle off on, you can have a tool change mist coolant on, flood coolant on all these different options given by the various miscellaneous commands. So, these are really the operations which are not been covered so far, in terms of the other functions that have been described earlier. And the generally the miscellaneous commands take effect after execution of the other commands in the block. So, they are last in the row and that is how they are also programmed. If you look at the controller specification also the miscellaneous really comes towards the last of the particular command line.

So usually permissible to produce more than one miscellaneous command in a given block as long as they do not have any conflicting interest for example, you cannot say coolant off and on together. So, that does not create any sound bases as long as there are no conflicting natures of these commands you can have multiple miscellaneous command in every command block. So, that is about it. So, that is sort of bringing us to the end of this particular module when we talking about the various programs.

The only other small thing which we need to mention are the sometimes special characters, like percentage sign etcetera which are done to you know just indicate the start of the program or the stop of the program. So, generally the percentage sign common to all controller is used for indicating the first line of the program. And so these special characters are just over and above what ever has been discussed so far in terms of the G commands the F S. The axis motion commands the identification commands in the miscellaneous commands. So, they are also serving some role the special characters in terms of startup program and stop up program.

Particularly when you are talking about ending the program you sometimes write end of block you know the letters EOB for example, illustrating that the program has a stop on that particular portion. So, this gives you holistic overview of all the modalities associated with the NC program, part program. Now we will do actually a part program in the next module and try to identify in a very nice and intuitive manner how you know the machine path can be planned based on the program.

Thank you.