Manufacturing Systems Technology

Prof. Shantanu Bhattacharya

Department of Mechanical Engineering and

Department of Industrial and Production Engineering

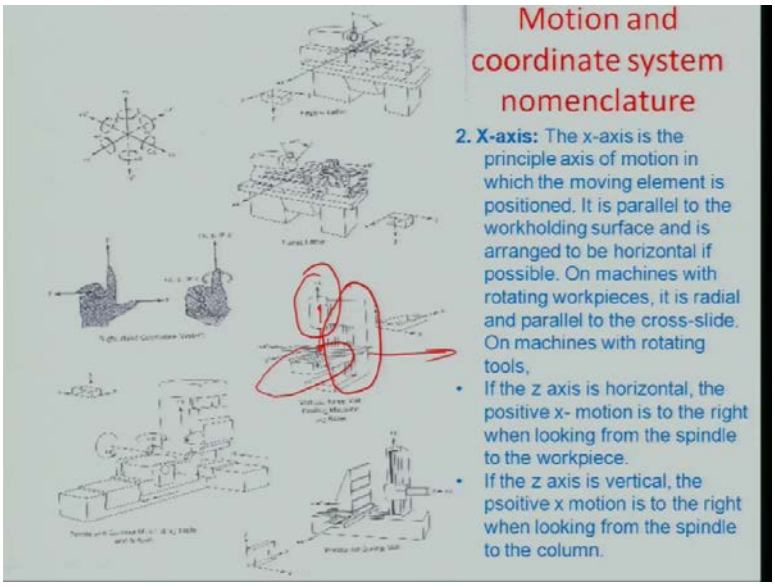
Indian Institute of Technology, Kanpur

Module – 06

Lecture - 33

Hello and welcome to this Manufacturing Systems Technology module 33.

(Refer Slide Time: 00:17)



So, we were talking about the different directions used in nomenclature for NC machining and I have just made you exercise, how to figure out the z motion in a for example, rotating work piece system or a rotating tool system. In that we had considered two operations, one is the face milling operation and another is normal lathe based, you know turning operation.

So, turning operation as you know the work piece, the cylindrical work piece moves and the spindle axis is parallel to that of the work piece axis. So, that is the cutting motion or the cutting power which is being applied by that spindle and always we have to remember that the z axis of a machine is that spindle and that is how internationally this has been accepted as a norm based on which all the NC controller have been designed.

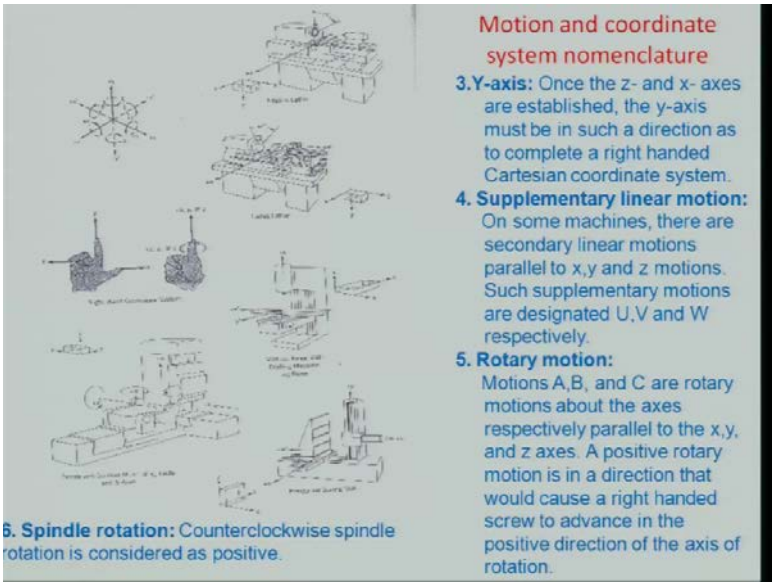
So, if you are looking into a face milling operation, what do you exactly get? For example, this right here shows you a detailed schematic of a face milling operation. So, you have a rotating tool somewhere here for example. So, you have some kind of a axis of this tool, which can come down you know or go up in this manner, there is a column side which is this particular side here right away from the spindle towards the body of the machine and then there is; obviously, a work piece stage, which is some kind of a slide you know which can operate in this particular direction.

So, here I think I had mentioned that if the tool is going away from the work piece, the work piece is placed in the bottom somewhere and the tool is rotating here and in the other side of the tool is the column. So, you can see the column through the tool and so basically the z axis of motion, the positive z axis of motion is as the tool goes away from work piece, so it go in this particular direction.

So, if supposing this goes away from the work piece on your right side, you can define as the positive x direction. And; obviously, if the positive x and positive z are defined, then going into the column or seeing into the column becomes the positive y directions. So, that is how you basically give the nomenclatures to this x and y and z axis of motions. So, when you are looking at a lathe; obviously, now you have a cross slide movement which is actually giving a radial feed as well as an axial feed, if it is parallel to the axis.

So, if the tool, supposing there is a rotating work piece and if the tool is coming towards the work piece, it is a negative z direction, it is a positive z direction when the tool goes clears of the work piece surface and when you are actually going into the... And so when you are sitting on the spindle towards a positive and facing towards the positive z direction on your right is the positive x direction and so; obviously, therefore, if you look at from where you are sitting downwards is going to be positive y direction. So, this is how you gage whether it is a rotating work piece or a rotating tool what would be the positives and negatives in the x, y and z directions.

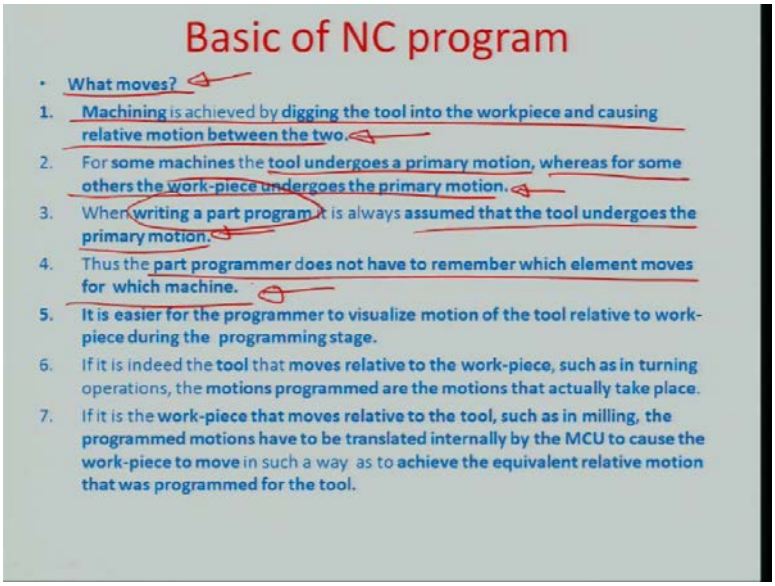
(Refer Slide Time: 03:16)



Once you do that then the question of rotary motions come rotary motion; obviously, if your finger you can do it with this right hand grippers. So, if the finger points out to the positive direction of a certain axis. For example, let say this is the direction of positive z axis in case of the milling tools as I suggested, then the curling fingers of your right hand would then give you the positive direction of rotation. So, in this particular case, if I look from our views, really the counter clock wise direction, which you are having for positive direction of rotation about this particular axis.

So, you now have three different axis directions and you have also three different rotation directions, which are involved in defining more or less all the complete relative motion of the tool with respect to the work piece. So, once you understand this whole concept of directions, then you can you are ready for programming and then all you need to do is to sort of organize the information that you are providing the controller. So, that every time it operates it can repeat those sequences and produce the same output as the result of it. So, let us look it to that aspect now.

(Refer Slide Time: 04:19)



So, what are the different basics of the NC programming system? So, if you look at it carefully. So, one thing which is important is what moves in a NC system. So, as you know machining is achieved by digging the tool into the work piece and causing relative motion between both the tool and the work piece. So, some of the machines; obviously, tool undergoes primary motion whereas, for some others primarily the work piece undergoes this same primary motion. So, it is always assumed that the tool under goes the primary motion when you are writing a part program.

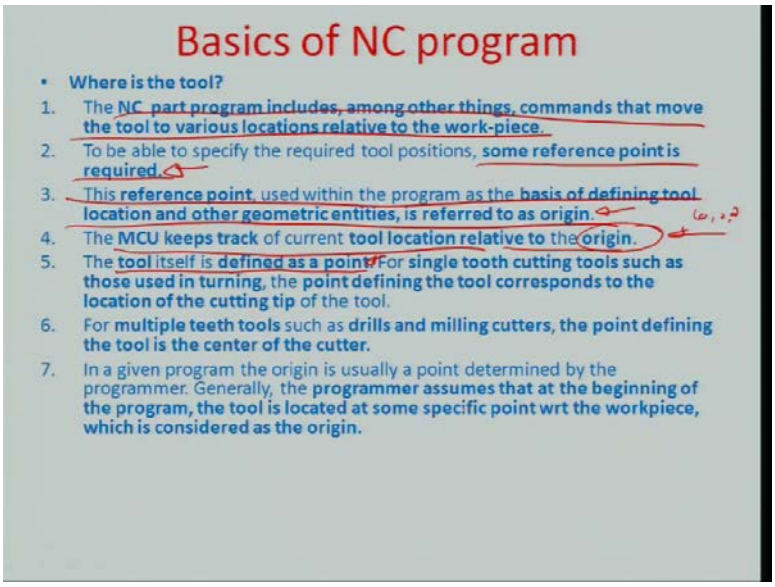
So, for a part programmer you are always sitting on the tool and not on the work piece that you have to remember and wherever the tool is writing over the work piece making it is own topology or own path is actually the path that you are trying to define as a programmer. And; obviously, if at a certain instance there is a lot of optimization, where the controller is able to do some, you know self assisted manipulations. It may feel at certain point of time, the work piece needs to be machined it is more is better in terms of time of machining, etcetera automatically the conversion will happen at the controller end, you need not worry about that.

But, as a programmer you must always give the tool path and you must ride on the tool for writing the program of the tool path with relation to the work piece in the question. So, the part programmer does not have to really remember which element moves, once you have programmed as the tool path, that is all what you need to really work on and

keep on modifying and wherever needed, there will be a choice given to the controller whether it is, you know feeling that moving the work piece would be optimum or tool would be optimum.

In some cases it may really move the work piece to make the situation time optimum etcetera. So, that is typically what you as a NC programmer have knowledge, that it is the tool, which is creating you or it is urging you to develop it is path programmer.

(Refer Slide Time: 06:21)



So, the NC part program includes among other things commands that move the tool to various locations are related to the work piece and so for that movement; obviously, there is the requirement of some reference point, which can be considered as a origin or as a start point to start the positioning with respect to. So, it is basically the origin of the orthogonal coordinate system, in which the part drawing has been layout in that particular manner.

So, this reference point used within the program solves as a basis of defining the tool location and other geometric entities and we referred to as a origin. And what the MCU really does it is to keep track of the current tool location relative to this point, this origin of the particular; you know three dimensional orthogonal coordinate system for example. So; obviously, because we are talking about a tooling, where there can be either a single point or a multi point engagement.

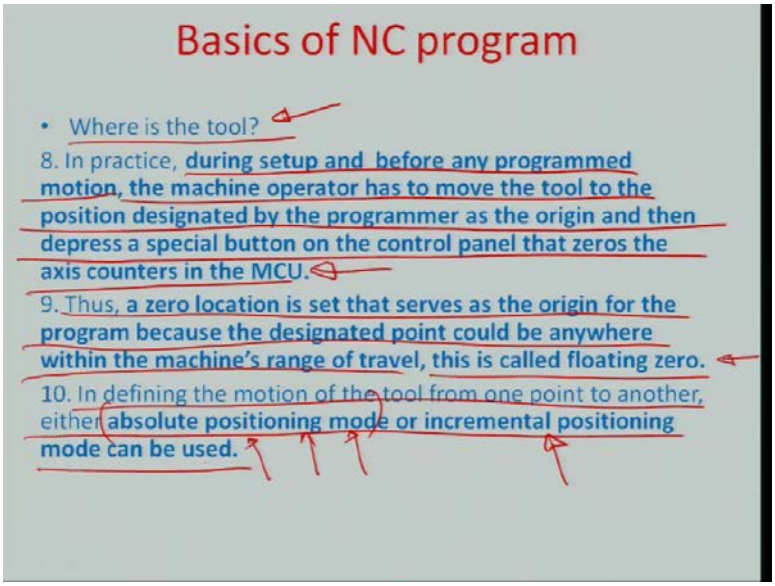
So, for a single point cutting tool it is almost always good that the point of engagement of the tool is used as a source for programming the path of this point of engagement along the various geometric, you know I mean sizes or shapes that it has to go through, to make the particular profile or topology on the work piece.

So, you using the engagement of the tool as a point, but in certain cases of multiple point cutting operations like milling etcetera, you have to assume that you as a programmer or sitting on the center of that particular tool and; obviously, you are compensated by creating a path which is parallel to the actual path of the point of contact by a distance amount equal to the radius. So, you are actually sitting at the center.

In some machine tools, there is always a tool compensation factor which comes in, where you can actually program the main tool path and forget about the, your path which you assume to be set at the center of the particular tool is always a one radius away from parallel to the tool path. So, this is always in some cases auto compensated, in some cases you are to do it yourself. So, in most of the programming that I will be teaching you through particular module we will be actually assuming ourselves to be in charge you know and we will be also assuming that we take care of the cuter compensation our self's rather than the controller.

So, there is no cuter composition effect as such in all our programming that we will show here. So, then the path that you have to really program is this the path that the center would take plus the radius which is there in the cuter and in that terms you will define all the coordinates which you are trying to plan on the work piece geometry.

(Refer Slide Time: 09:19)



So, in practice where really is the tool? So, during set up and before any program motion is executed the machine operator has to move the tool to a position designated by the programmer as the origin and then depresses special button on the control panel that zero's the axis counters to the MCU. So, this is also called zero setting of the particular tool, here you have to be careful that because of backlasher etcetera in some of the, you know motion controlled linkages there is always a tendency of the positing to deviates slightly and there is always offset or a zero error which comes into the system at this point of time.

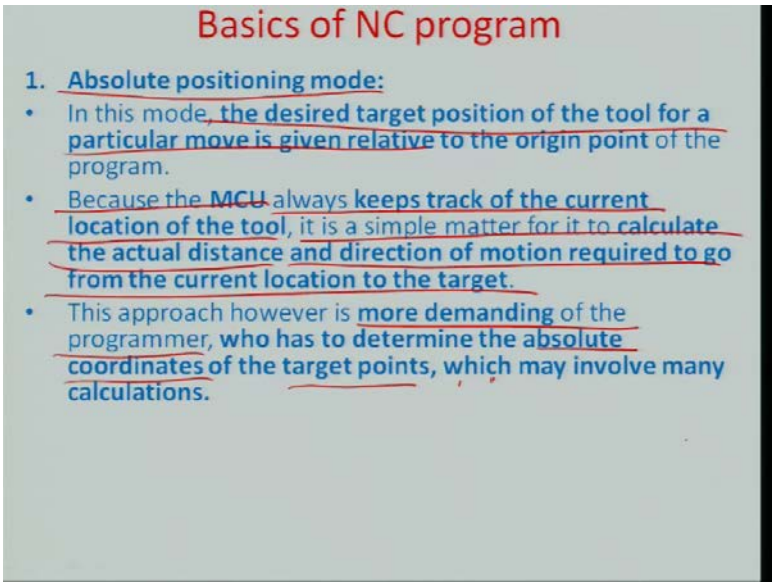
So, you have to be really careful as a programmer to take care of those small zero errors or offsets before proceeding ahead with the actual programming. So, there is a zero location is set that serves is an origin for the program, because the designated point to be anywhere within the machines range of travel and the therefore, this is also known as a floating zero sometimes.

In defining the motion of the tool from one point to another, you follow different positing systems, one is the absolute positing, another is a incremental positing as the terminology itself indicates absolute positing probably means that with respect to the origin every time we go to a certain point on a tool geometry you should be able to read that particular coordinate with reference to the origin. And in the incremental stage all

what you should think of his that you have certain point at which you have position correctly.

What is the next increment that you should take to go to another point which is ahead of it? So, there is always a lot of computational saving on the part of the human being, but; obviously, your taxing the controller more, because every time it has to find out the relative position difference with respect to the origin you give a incremental command control. So, that in away is more computationally exhaustive process, so we will be actually mostly talking about the absolute positing mode, because that is where you can get to see what is the internal story behind, you know how this machine map or the relative map of the machine tool is read with respect to the work piece as a part of the program. So, let us look into little more detail about both the modes of programming.

(Refer Slide Time: 11:37)

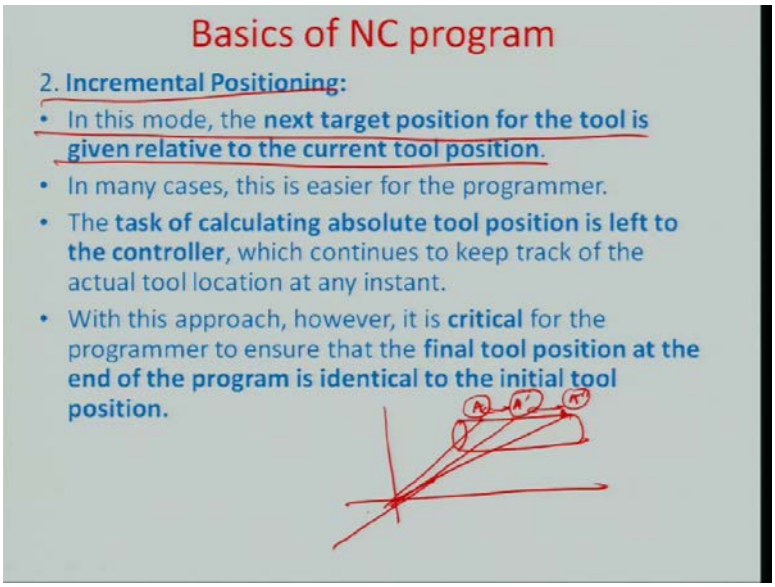


So, in the absolute positing mode the desired target position of the tool for a particular move is given relative to the origin point. So, if let say we wanted to make some kind of a cylinder, you know which is position somewhere here and there is some kind of coordinate data across the points a, a dash, a double prime these are always read with reference to the origin that is what actually it means when we are talking about absolute positing system. So, it is always reference read with respect to the origin.

So, because the MCU always keeps track of the current location of the tool it is simple matter for it to calculate the actual distance and direction of motion required to go from

the current location to the target. So; however, for a programming aspect; obviously, this is going to be more demanding of the programmer, because every time has to calculate the relative position himself you know with respect to the origin. So, that creates you know developing of this target coordinate points with the use of many involved and lengthy calculations on the part of the NC programmer.

(Refer Slide Time: 12:55)

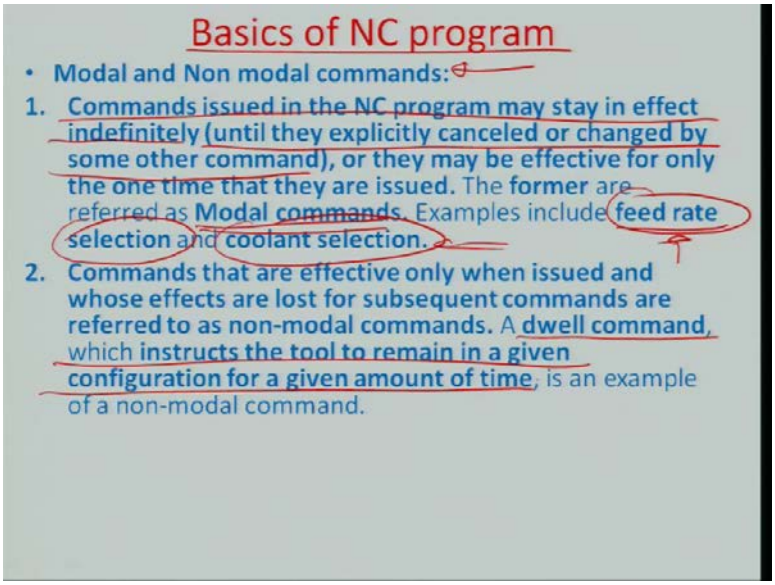


On the other hand incremental positioning is a mode where are the next target position for the tool is given relative to the current tool position. So, in the same illustration here all you need to know is what is the relative location change between a and a prime or a as matter of fact a double prime, another point in the particular tool. So, this incremental difference is what you are trying to give to the program and the program now is calculating how much this would mean in terms of with respect to the origin change in the coordinate system.

So, obviously, it is going to be more taxing on the controller now and lesser taxing on the human beings. So; obviously, it is a mix and match in real situation and real life situation; obviously, both the absolute as well as incremental programming or positioning would be used; however, in our sense here for example, in this particular course you would be limiting over self to absolute positing. So, that you can get a good feel of how the controller or what kind of exhaustion or exhaustive calculation steps are involved by the controller to back calculate from incremental mode the actually you will

absolute reference coordinates you know in a certain frame of origin in a certain reference frame, frame of reference.

(Refer Slide Time: 14:13)



The two other aspects of NC program which you need to know before finally, getting to start are the modal and non modal commands. So, you know there are commands issued in the NC program and that is how the languages has been developed that must stay in effect indefinitely, until there exclusively canceled or change by some other command. For example, let say if you have given a feed rate of a certain tool or maybe you have made a coolant selection unless and until you redefined another feed rate and other coolant selection option, like maybe you have put coolant on and the machining processes going on.

So, at the final step when the machining has completed you have to now put coolant off as selection option for the coolant to turn off. So, till and until the machine is going on the machine should ensure that there is the supply of the coolant. So, such commands are known as modal commands which actually once given would exist within the system, the others; obviously, are non modal commands where there is a onetime execution of the command followed by itself cancellation. So, there is no separate step needed to cancel it separately.

For example, if supposing we are talking about a dwell command that instruct the tool to remain in a given configuration for a given amount of time, the movement this whole

dwell option is executed, the command is executed the dwell counter is again reset back to zero and you have to again go to certain other location to again program a certain dwell time for the tool to be in that location for that time. So, every time you have given it after one execution round you have to reset the count at the into counter automatically gets reset back again and you have to again give a value to the counter to use put that particular command or particular counter in use.

So, these are the two main classifications of modal and non modal. So, in a nutshell what we have done is we have located, we have try to figure out what are the x, y, z directions and the rotations, we also try to figure out what are the absolute and incremental positioning bases and also the modal and non modal commands bases. Now, we go to the next step of the NC program which is actually how to start writing the program which will probably take it up in the next module.

Thank you.