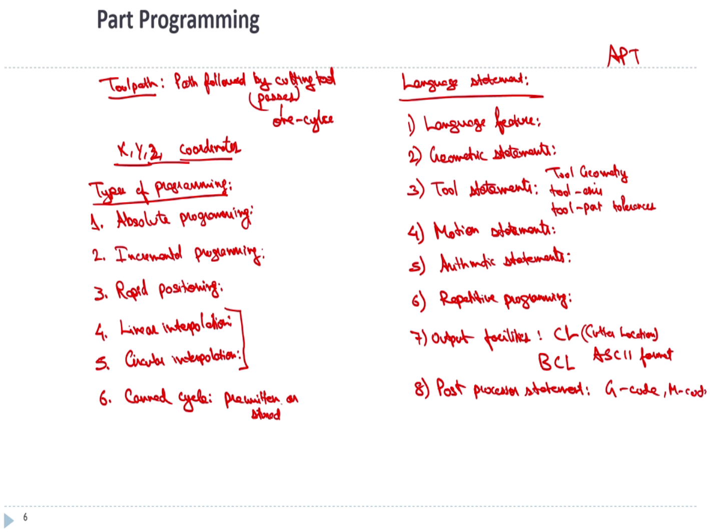
Computer Aided Decision Systems - Industrial Practices using Big Analytics
Professor Deepu Philip
Department of Industrial & Management Engineering
Professor Amandeep Singh
Imagineering Laboratory
Indian Institute of Technology, Kanpur
Lecture 38
DSS in CAM (Part 2 of 2)

(Refer Slide Time: 0:19)



Welcome, let us continue what we were discussing in the last lecture, the Part Programming. So, I will try to discuss the Tool Path Generation, Tool Path Verification, the different programming features that we have and then we will also see G and M-code in this lecture. So, first is Tool Path, in NC programming whenever the program starts to write a program they use their knowledge to plan a Part Program, in which they require Tool Path with proper cutting parameters is set. What is a Tool Path? It is a part that the cutting tool must follow. So, it follows this path and then comes back to its home position. That is the starting position this path is usually repeated more than once, when it is repeated more than once it is called a pass. For instance, in one go, if I suppose I need to cut this material down by 5 mm but I cannot do it in one go, I will do 1 mm in one pass, coming to home, position another 1 mm in the second pass. Five passes of 1 mm each would be taken.

In general, this would not be exactly five passes, it could be the five passes of varying depth of cut, that would be in a way it could be from 1.5 mm, then 1.4 mm, then 1.2 mm, 1.1 mm, then the last path might be something 0.5 mm because here we also trying to get better surface finish. So, the tool follows different motion paths and it does it in different passes, so that it completes the overall machine that is required, so pass is one cycle. One cycle that the machine is trying to take to finally finish the path.

Once the path is planned the NC programmers utilize their knowledge to generate details of the path which includes the X,Y, Z coordinates of the necessary points of the path. And the syntax of the programming language that we are trying to use, the post processing of the program and using the program to produce the actual path. NC packages generally provide the programmers with two libraries that they can use to define various tools with proper tool geometric parameters. For instance, a programmer can define a drill or a cutting mill once the tool is defined the Tool Path can be generated using the tool and path geometry. Nowadays, the CAM software such as Power Wheel or Mastercam, they provide the complete simulation how the tool is moving and they provide also the information on what is the time it is taken to the overall machining.

Additive Manufacturing also helps us to get the time and also cost that it would take to produce this path, depending upon the time it is taking. It will also calculate depending while we have the input of the material that is taken into account, so costing is also taken there.

Now, in the Part Programming we have different programming steps, steps or the principles that it uses, there are different kinds of Part Programs. I would say the type of programming:

1. Absolute Programming- Absolute Programming means first the origin is defined, absolute origin what is my present coordinate, from the absolute origin what is my future coordinate, from the absolute origin what is the next value. See, each time X1, Y1, Z1 to X2, Y2, Z2 values would be taken from the original. So here, in the Absolute Programs for each axis of motion the origin is selected, the intersection of the X,Y and Z axis of the motion defines the machine origin which must be coincident with the Part Program origin. Once the origin is selected all locations are defined with respect to the original as I just mentioned.
2. Incremental Programming- Incremental Programming means it is not necessary to select the origin now from the current point to the next point. Now, the next point becomes my origin. From there that goes to the next point each time we are putting an increment of the specific dimension that we are using. So, here, the immediate advantage is that the programmed motion directly matches the actual motion of the tool because it starts from the last position of the cutter or the tool.

3. Rapid Positioning- Rapid Positioning means it is a point to point machining positioning or Rapid Positioning system where we increase the machine speed so that we enhance the machine productivity. It is majorly used in drilling and boring. For instance, drilling hole has happened at one place at a specific speed and it comes out, rapidly it goes to the other hole, it does the other hole, rapidly it comes to the third hole it does this hole, if this rapid motion is not there machine tool will go so slow and this is just a completely non-productive time that is there. So, to cater this Rapid Positioning is a very important step that is always put as an input in a machine program.
4. Linear Interpolation- Linear Interpolation means the programming () G-code helps us to produce straight line motions by allowing the corresponding machine slides to move at different speeds independent of each other. For instance, in a three axis machine the slide in X direction, in Y direction, in Z directions can move at different speeds that helps us to have an interpolation that is in a linear motion only. Majorly, in a two axis machine it would always be linear, in a three axis machinery it could also be linear but those are independent.
5. Circular Interpolation- It is the counterpart. So, to achieve this motion the relative positions and velocities of two machine slides are initiated and maintained on a constantly changing basis but starting and stopping at the same time. Special code is used in a numerical control program to indicate Circular Interpolation, so, both the linear and central interpolations are used to guide tools along linear and circular paths respectively.
6. Canned Cycle- Canned Cycle means it is a specific set of cycle pre-written or stored cycle because it is pre-written or stored it is why is known as Canned Cycle, where standard subroutines or procedures are followed which are taken from the library of the machine tool. The examples include the drilling, tapping, boring, turning, threading in which the fixed or variable cycles could be taken. The fixed cycles are there which are not adaptable to a specific user needs, the variable cycles are there that could shoot to a particular application that a person or the programmer is trying to work on.

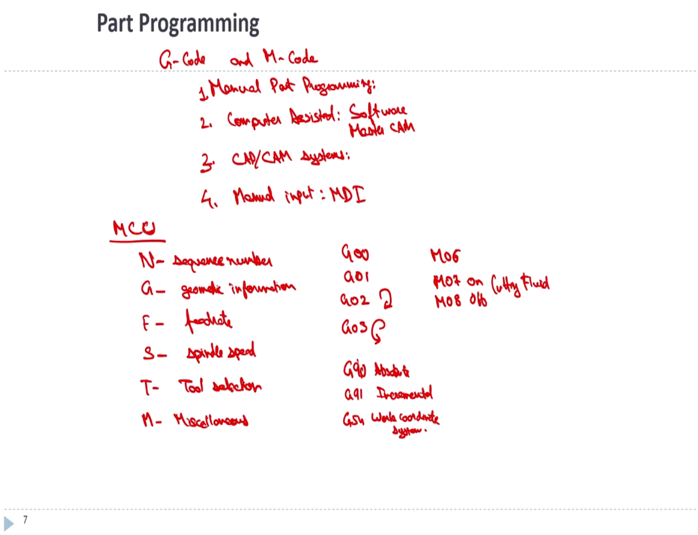
Now, we are different from the NC programming languages. Now, in a CNC Language Statement what are the features?

1. It is the Language Feature itself. Language Feature means the programmer defines the variables and arrays of the program. For instance, in a C language itself we do define what are the variables, what are the arrays here. Also in the language we design what are the variables or what are the fixed parameters there, that means variables arrays and geometric entities are defined here, so we select accordingly. So, in a Part Programming language a programmer takes the geometry of the Part Program from an engineering drawing or from a CAD database to describe the path to be taken by the cutter. So, we have different kinds of languages available, CNC programming languages generally APT. APT (Automatically Programmed Tool) where the syntax is different from the very general NC programs that is also one of the languages that is used.
2. Now we have Geometric Statements. From Geometric Statements I mean the description of the curves or the surfaces so in a Computer Aided Design environment the geometric information required by these statements is generated interactively when the user selects the surface to be machined when using an CNC software. Geometric statements should precede the Motion Statements in a Part Program.
3. And, also we have with us the Tool Statements. Tool statement means the tool geometry, tool axis and tool pat tolerances as I discussed. If I have a work piece to be produced with a tolerance of maybe of 1.01 millimeter the machine should have a tolerance 10 times of it. So, machine tool tolerance interfaces are very important. 10 times of it means for 0.01, it should have a tolerance of 0.001, it should be more precise the resolution should be lower. So, Tool Statements are there
4. Then what we have with us is the Motion Statements. Motion statements means these guide the tool into its motion, they provide information for the type of the machining, we are going to do the PTP or continuous part, or we are going to have some direction of cutting, speed, feed rate and so forth. So, all these statements are part of the Motion Statements. All the statements come as a portion of Motion Statements.
5. Next, what we have is Arithmetic Statements, as the name speaks it is just addition, subtraction just as in the Python language we add or subtract something and we also can approach certain other functions such as we take square root, sign of something, these Arithmetic Statements would be taken.
6. Then, what we have is Repetitive Programming, statements that provide the looping of a system, branching of a system and coordinate transformations. So, all these

constitute the Repetitive Programs where we have a macro facility that enables the programmer to deal with Repetitive Programming more effectively.

7. Then, we have Output Facilities, so by Output Facilities I mean the Cutter Location. Where is it? CL is Cutter Location. The Part Program forms a CL data which is stored in a file which is also known as CL file it is usually stored in ASCII format. Also sometimes, it is known as Binary CL file or BCL. In an attempt to reduce the file size and to increase the speed of its post processing the Part Program can also be stored in a Binary Cutter Location system or CL file.
8. So, then we have with us at last is the Post Processor Statement which means the Cutter Location file or the Binary Cutter Location file is written in the programming language syntax. This must be post processed by a selected post processor, the post processor here is hardware dependent which is written for a specific controller, so most of the controller accepts the G and M-codes.

(Refer Slide Time: 15:30)



Talking about the G and M-code, this was the point we were trying to reach to understand what Part Programming is and how are this code able to make the machine to understand how are the machines moving, or how the tool should be moved, what should be the location, what tool should be selected, the different classes of it as I said, in different layers the information regarding the motion, the annotations, the tool to be selected should be put, so that is why we have G-code and M-code where the programming could be taken in different modes.

1. It could be maybe a Manual Part Programming. Manual Part Programming is just like you take an exam, you write the program in pen or you type the program exactly G0 G90 G54, then you start the first line of the program N1 and try to say this is the my position X00, Y00, Z00, when you try to make a program all manual that is Manual Part Programming. So, this is completely manual.
2. Second could be in Part Programming that is Computer Assisted. In a Computer Assisted Program whatever you are trying to do in a simple manual Part Programming you are trying to use a software right. This is generally a Master CAM Software.
3. Another advanced programming could be using CAD-CAM systems. Which means not only the basic Master CAM Program, one can use a complete CNC package to develop a program. This is similar to the second approach that is Computer Assisted Program but the benefit here is that the program uses the CAT database of the part directly and there is no need to translate it or import it into a CAD-CAM program through the IGES or STEP file.
4. Fourth method could be a Complete Manual Input that means the programmer uses the controller of the machine tool to input the data, it goes to the machine, type the program there at the machine itself and then they do it, they call it MDI (Manual Data Input).

Irrespective of the method that is used to create the NC Part Program, the instruction statement in the program must be in a basic code that is the Machine Control Unit. MCU helps us to understand the basic code and a conversion process that is done by the post processor is necessary when high level languages are used. So, existinG-code words that we use are:

1. N is for a ‘sequence number’
2. G is for a ‘preparatory word’ or maybe a ‘geometric information’
3. F is for the ‘feed drive', if I say F5 that means generally it is 5 mm per minute feed is given.
4. Then, S is for ‘spindle speed’ if I say S 1000 that means 1000 rotations per minute speed is given, this is spindle speed
5. T is for the ‘tool selection’ that means it specifies the tool from the tool library on the machine tool, what tool is to be taken

6. Then we have M that is for the ‘miscellaneous command’ but these make different layers to understand the different parts or different features of a program separately. Miscellaneous could be for a maybe a center lathe or maybe for a machining center, if I say M6 it is selecting the tool, then the miscellaneous operations have also different names.

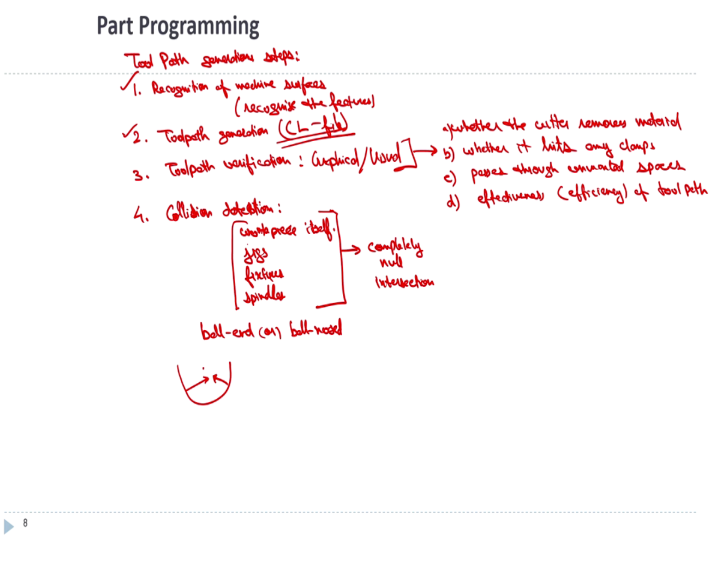
I could maybe define a few of them, maybe for example G00 is a program of rapid point to point movement, G01 could be linear motion between two points, G02 could be clockwise circular motion, G03 is a counter clockwise circular motion, and so on.

More important among these are:

1. G90, that is when we choose the absolute coordinates
2. G91 is incremental
3. G54 cannot be missed which means work coordinate system
4. M06 is tool change
5. M07
6. M08 is for turning on and off the cutting fluid.

This will make more sense when I come with an example at the end of this lecture taking geometry and try to understand how the Part Program is written.

(Refer Slide Time: 22:15)



Before that, I would like to discuss the Tool Path Generation and Tool Path Verification characteristics. Majorly, what we have discussed in this lecture and we will try to cover some other points.

1. First, we have discussed the steps in a Tool Path Generation is Recognition of Machine Surfaces that means we try to recognize the feature.
2. Next, what we have is the actual Tool Path Generation, once a Tool Path is generated we do not stop here because Tool Path Verification is also important. We need to verify whatever we have generated is correct or not.
3. Tool Path Verification that I will discuss and also,
4. Collision Detection: a very important factor, unless we detect the collision the whole system can be completely damaged because tools when they collide a positive space it could create a critical damage to maybe the tool could break, or maybe machine spindle could have a small flaw or small play in it. So, all of this could happen if the Collision Detection is now taken properly.

So, as I mentioned, nowadays, there are certain simulators or the simulations that you can see or visualize how the tool is moving, whether the tool is colliding with a workpiece itself, whether the tool is colliding with the machine component or so. So, this also ought to be taken care of.

Recognition of Machine Surfaces that we have discussed in detail and Tool Path Generation is the actual generation of the Tool Path where we try to store the path in the cutter location file. Next, we try to see whether the path should be verified. How does the path have to be verified? Now, in the CNC program the Tool Paths that are supposed to guide the tools during the actual machining includes a lot of coordinate values that are impossible to verify manually.

So, Tool Path Verification has to be graphical or visual, the softwares has to simulate them, actual machining processes could be taken. So, by displaying the cutting tool moving its Tool Path relative to the stock and jigs and fixtures that could be taken shaded images, different colors could be taken there. This enables the CNC programs to spot any potential errors in the NC programs.

These animated or simulated Tool Path generated by displaying the tool position and orientation using the CNC program data at various points of Tool Path creating frames storing the frames for playback have certain advantages.

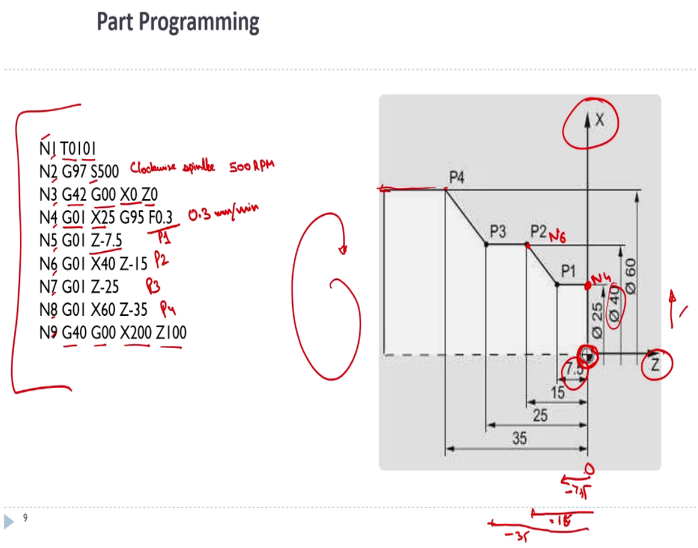
- We can check whether the cutter removes material from the stock or not.
- Next, we can see whether the cutter hits, if it hits any clamps or so.
- Also, in the verification we can see whether the cutter passes through the floor or side of the pocket or through a rib. It is to pass through unwanted spaces,
- Then also we can see the effectiveness or maybe efficiency of Tool Path. Other advantages may be the rapid turnaround or a program development, rapid training for the potential CNC programs without dentures that how the tools would go, how it would look like that is the visualization of that, then we have graphics display that when the Tool Path is generated, it can be verified here.

Now, an important factor here is Collision Detection. Collision Detection is concerned with finding out if the cutting tool and its assembly, that is a tool shank or maybe the tool collet or So, collides with the components of the manufacturing environment, that is it could collide with the workpiece itself, right? Or, it could collide with the jigs, fixtures, spindles or, any of the components in the overall work environment that you have. This intersection should be completely null. Generally, we use tools which are ball-end or ball-nosed.

So, in a ball-end or ball-nosed tool, generally what happens, the directions of the tool, for example, this is a tool movement is taken to the normal of the direction of its movement. So, the calculation of the cutter offsets is achieved by finding the directions normal to the surface of the tool. So, once we have determined the directions of the normals, then the tool offset by the radius of the cutter along with the normal vectors over the surface could be taken into account.

So, before even calculating these cutter offsets that could be a suspected collision in the future. So, we have to generate a Tool Path that is to mill a surface parametrically cutter location points are generated using the surface definition and machining parameters that are located in the RCL file. So, the ball and cutter tip is directed to move in straight line segments to each point. The fewer the points the more rough the resulting machine surfaces, however large point files should be avoided to minimize storage computation and milling time.

(Refer Slide Time: 30:13)



So, now let us have a quick look on a G-code program where we try to understand how the G-code is written when I say G-code it is G and M-code both and we try to look at that how do the machining or the program is made, this is the first Part Program where the G-code and M-codes are just ready, so this is a job on a lathe machine where circular machining is happening.

So, the job rotates in a clockwise fashion and this is the machining that is happening so when we say about the code the G and M-codes have different functions as I said different layers are there. For different operations, the annotations and everything are put in different parts.

So here, what we have, this N are the sequence numbers, sequence 1, 2, 3, 4, 5, 6, 7, 8 and 9, it is a 9 line code where T indicates my tool selection T0101. That means, this tool is selected. G97 means as I said it is the machine setting where spindle rotation is in clockwise direction, it is clockwise spindle rotation G97. And, S is for spindle speed, spindle speed is 500 rotations per minute.

Now, we have G42 which means that the P0 point where the two nodes radius composition is active, is selected, so X0, Z0. So, this is the point of my origin here on the figure so this is G00. And, I have set my tool at the point 00.

Now comes the first step where G01, G01 means linear motion only because the job is rotating, like suppose, this job is rotating and when the tool moves in a linear motion only, it will keep on removing the material, the job is rotating, the tool moves only in the linear fashion. So, that is why G01 linear motion is given to the tool and it comes here at X25 that means it goes 25 mm from here in the X direction, this is X direction given here. In the X direction the tool comes here now, I will put, this is my N4.

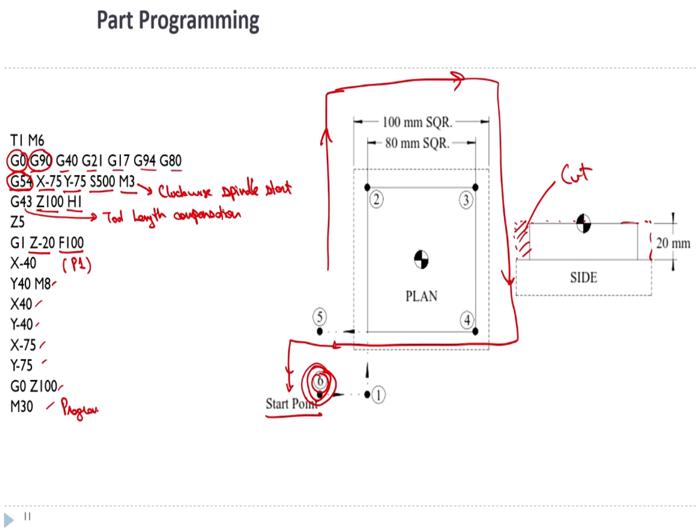
Now, since it has come to this position, now we says feed rate is 0.3 that means the feed is given as 0.3 millimeter per minute and with this feed, this is the feed rate of the tool, in this movement speed of the tool when it says speed is a spindle speed, the rotation speed, this is the feed of the tool, that is 0.3 millimeter per minute, it starts moving. And, when I say G95, it means the feed or the revolution in milling and drilling. It is a kind of a milling operation. The lathe or the turning operation is kind of a subset of a milling operation only, it starts moving.

Now, once the movement has started but it is there at one position only, now we will say, okay, in the Y direction you start moving at some minus of Y, whatever the distance is required. The distance is 7.5 millimeters, so, we say Z-7.5 or this dimension, this direction they have called it Z direction. So, next, it goes to N40 and this is the position where it reaches at N6, that is the 40 millimeter dia.

And, here X40 Z-15, this is another 15 is there from here itself, that means it is going from here 7.5, then it is going from here 15, what does this mean, this machine I would write it here -7.5-59-35, this means this is Absolute Programming, that is from the origin only it is moving towards the Z direction or the negative Z direction, this is Absolute Programming.

Now, here again finally, we go for G40, G00, X200, Z200 here the tool reaches here, at the point P4 it starts moving here, here we have reached the point P4, G60, Z-35, then we here we have at Z-25 we have reached the point P3, at Z-15 we had reached the point P2, at Z-7.5 we have reached the point P1 but this is how the G-code is a language that is read by a CNC program that tries to understand it.

(Refer Slide Time: 35:17)



Let us see another program here, this is a program in which simply starting from here this is my point number 6, it has written start point here already and I give the code G0, G90, G40, G21, G70, G94, G80, all these mean that we have given absolute coordinates, G90 means absolute coordinates and G00 that is we have a rapid movement also is activated here, all these codes have their own meanings. G54 is where I have selected the work coordinate system.

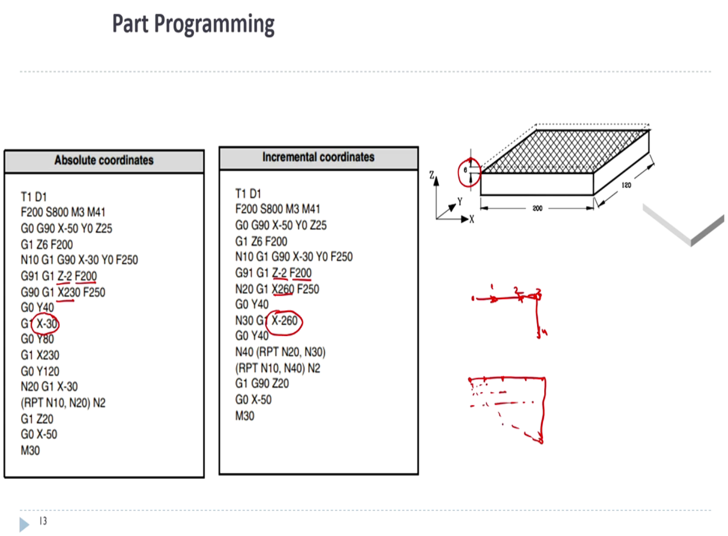
We talked about the coordinate systems that we select then we go to X-75, Y-75, speed is given 500 and M3, M3 means this is the M-code. M word means miscellaneous command where M3 means we say clockwise spindle start.

Now, we are here at this position. This is our 0-75 of X and -X75 of Y. This is my starting position here, at position 6. From here, I start G43 which means tool length compensation is given and Z100 and H1 from here the milling starts you can see it is a solid body. For instance, this is a solid body here, the tool starts moving its interface with the body and starts cutting the portion.

So, in this case this was a solid body and this portion is being cut. Now, this 20 mm is the depth that is being cut 20 mm, so, that is why Z is given -20 which means in this body the height of this body could be anything maybe 40 mm or so. It goes deep 20 mm and starts cutting it, this is mechanically that things are moving in a motion, electronically it is reading, this is what is a Decision Support System finally being handed by the CNC machines. Or, the language that is being read by a computer micro control machine. In which, now it says, Z minus 20 and feed rate 100, it goes to this level, here it goes at X minus 40, it goes to position 1 here, I would say P1 here, it is here and it starts cutting. Now, Y40 M8, X40, Y -40.

It keeps on moving in this direction, then this direction, then this direction. It comes back to position 5, then comes back to position 6. In the end this is how it goes. So, this is the code that is being run here and finally we say M30, it means that the program ends.

(Refer Slide Time: 38:29)

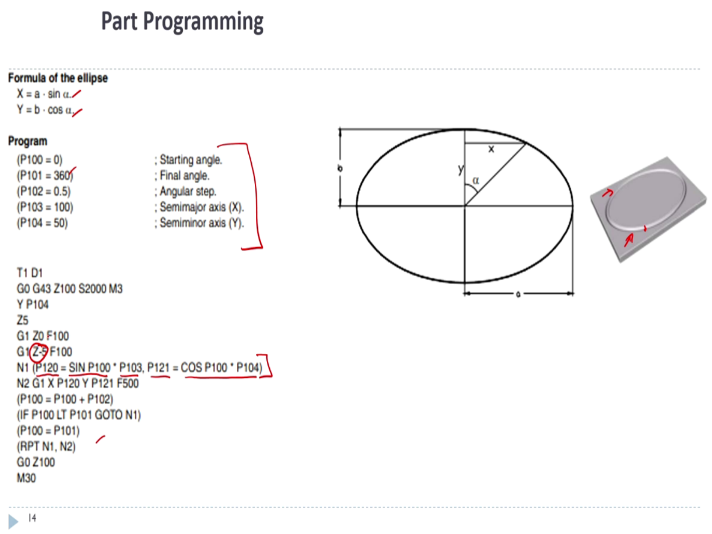


This is also one of the examples I have taken, in which this is Absolute Programming. This is a simple Absolute Program, for this component to be cut. So, this was the component which is also to be milled. This portion, 6 mm is to be milled. Throughout the component that means the overall width of the components to be reduced by 6 mm, so, this is an Absolute Program.

I will show you the parallel incremental coordinates or incremental program here. In this case you can see it starts, when this unit starts it goes to Z-2 and F200, it is starting from here only, and X from 230, it goes to here it is showing to 260. So, here in absolute it goes to the exact X-30 coordinate. Here, it is going minus 260 from the present position which was at 230 it goes to the it adds 260 or it subtracts 260 from 230, that is -30. Then, it goes there.

This is an increment from whatever position it is, it adds from here, it adds from here, it adds from here, it adds. It keeps on going from 1,2,3,4 points this is increment. Absolute it is, it always keeps on going from one position to another but everything is read from the origin only. From the origin only this dimension is read. From the origin only these coordinates are right, these are how increment and absolute coordinates could be chosen.

(Refer Slide Time: 39:56)

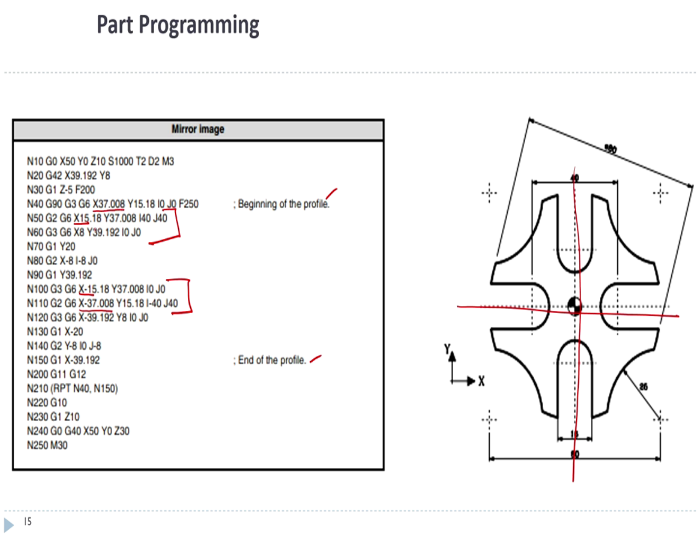


Yes, here we have = Asin alpha, Y = Bcos alpha, in the ellipse how do we draw an ellipse. Using that analogy only we have put this program. Here, N1 is the serial number in which P120 = sin P100 P103, P121 = cos P104, we have given the equation of the ellipse here. And, we have set all the programs in the beginning itself with the starting angle, finding the angle, what are these.

These are all kept here and we run this program to cut the ellipse on this plate, you can say here an ellipse is cut on a plate, in which small depth is given, depth given is minus 5 millimeters.

So, in general the dimensions given in the CNC program are in millimeters only. So, only we have to change this to the requirements that we have. Sometimes we have the FPS system or we have the MKS system. In the SI system it is put as millimeters in general. We can put in centimeters or so, whatever we like later.

(Refer Slide Time: 40:54)



So, this is a program where it is showing a mirror image. You can see the upper part of the program and the lower part of the program are almost the same. I can see this is the beginning of the profile, it is the end of the profile, it starts and begins. But it is a complete symmetric model along this and this axis. So, you can see these two lines and these two lines are almost similar programs with everything the same, only the difference is X15 is here X-15 is here.

Here, we have X-37.008, here we have X37.008. So, it is a replica of this program, this is how the mirror image of the program could be put, that means just in the C-coding or Python coding, when you have to repeat an array or so, you take one program copy it and paste it just change small variables or values there, you keep on repeating the programs, so this is how it goes.

So, these were a few examples of the Part Programming where you saw how G-codes are executed. This course could be seen in the simulation also as said. In the next lecture I would like to take 3D slicing. 3D slicing means when we have the file ready in the IGES or STEP format, we convert into STL format which is ((Stereolithic Ethical Format dot STL)) and we try to slice that for Additive Manufacturing that we will discuss in the next lecture. Thank you.