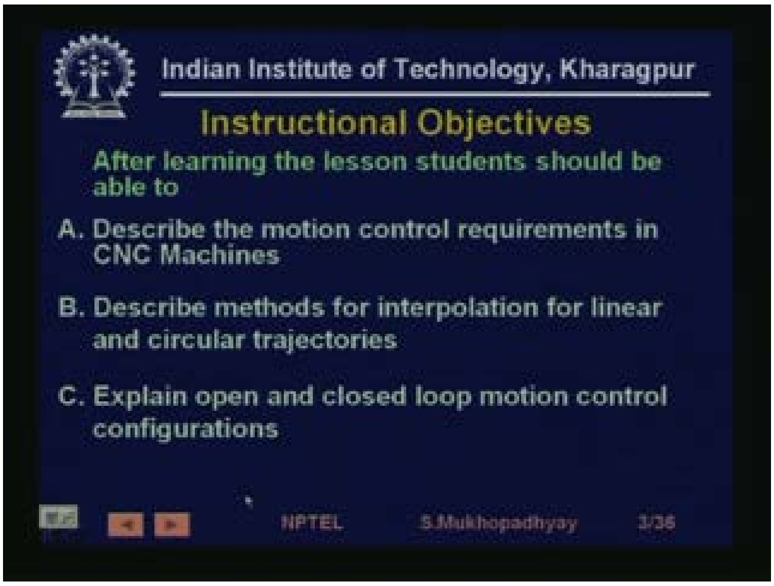
Industrial Automation & Control
Prof. S. Mukhopadhyay
Department of Electronics & Electrical Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 24
Contour Generation and Motion Control in CNC Machines

Welcome to lesson 24 of industrial automation and control, in the previous lesson, we have got a basic introduction of CNC machines, which are widely used in manufacturing, because of various advantages which we have seen. In this lesson, we will take a slightly closer look at how the CNC machine, actually creates the motion as we have seen that for metal cutting, the there has to be relative motion between the part, which may be lying on a tool, on a table or on a in a chalk in a lane and the tool.

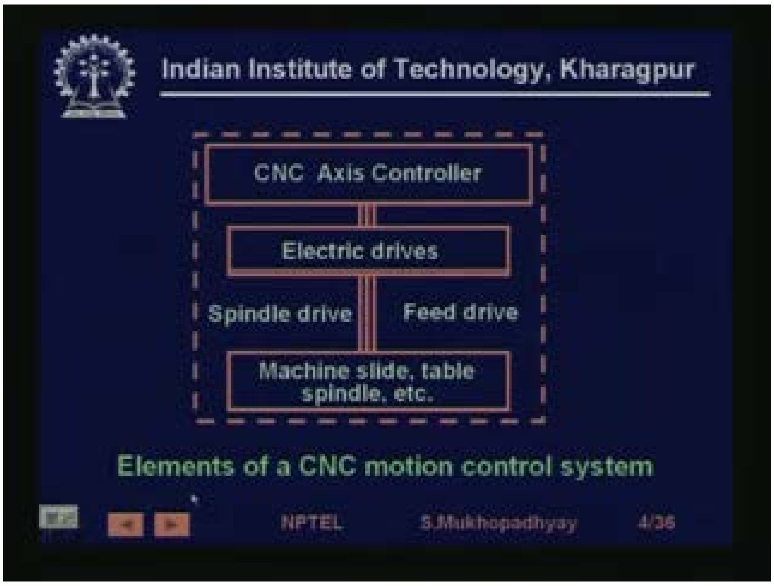
Now, in some cases as we have seen, the tool moves the tool undergoes rectilinear motion, while the job undergoes circular motion as in a lathe, while in the case of the milling the, job undergoes rectilinear motion and the cutter rotates. So, in this case, so we will see, basically we will see that how this, motions are generated, so that precise part dimensions can be created. So, this lesson is on contour generation and motion control in CNC machines.

(Refer Slide Time: 02:10)



Coming to instructional objectives, we will see the motion control requirements, the methods of interpolation for linear and circular trajectories, sometimes there are parabolic trajectories also we will not see that, and explain the open and close look motion control configurations, which are used in the CNC machine.

(Refer Slide Time: 02:37)

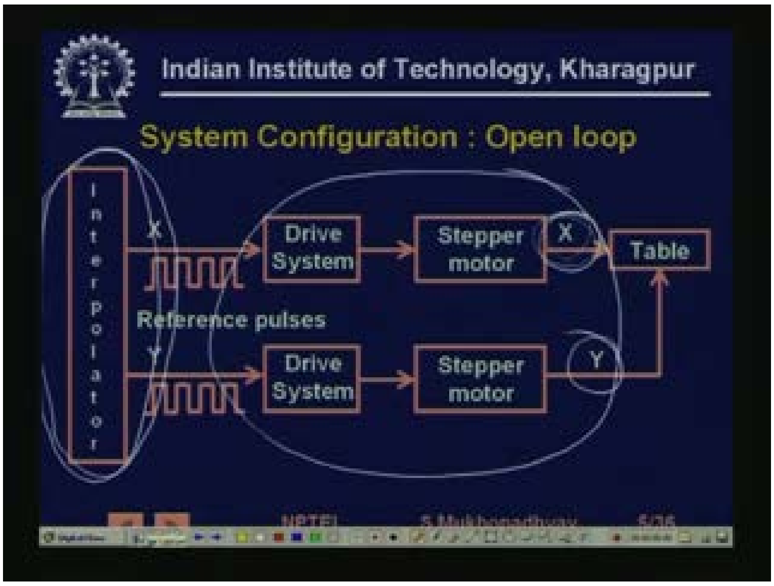


So, basic elements of a CNC motion control system are the, as far as the control is concerned, it is the CNC, what is called the axis controller, which is usually a microprocessor based board, which interfaces with electric drives. So, they contain the variable voltage, variable frequency power supplies, usually power electronic, which interface to the motors and these motors in turn interface mechanically, through gears and lead screws and another mechanisms, to the machine slide, the table the spindle etcetera.

And typically, we need to create to two kinds of motion, one is spindle drive, where that is the rotational motion, let us say for milling it, it is a cutter which rotates, so the spindle drive is on the cutter and the feed drive or the table drive, which is on the job in this case. The spindle drives problem is simpler in the sense that, typically once the a spindle speed is set, the problem is just to maintain the speed against loads, so it is a problem of speed control, the speed set point generally, once the material of the tool and the material of the job is fixed, it is the it generally remains fixed.

On the other hand, the feed drive or the table drive actually, has to create x y directional generally two dimensional, could be three dimensional also motion. So, that is complicated part, and we will look at that mainly, in fact almost solely.

(Refer Slide Time: 04:30)

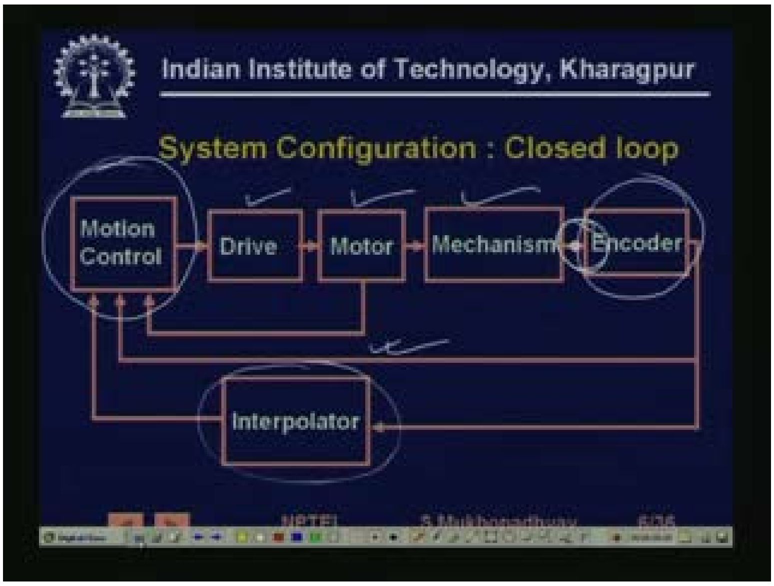


There are typically, there are two kinds of configurations which are used, for smaller systems, when with less critical requirements we use, open loop systems. So, you again you have a drive, you have motors, this is the table, that you can see, so this is the table in which, these are the two axis of motion, two axis X and Y, which create independent motions along the two axis of the table.

Now, the point is that, as we have explained previously also, that the purpose of a control system, is to is to achieve the or make the output track the set point, irrespective of disturbances, so that is the main job of the control. So, the controller assumes set point, so if you in the case, in the context of the CNC, if you ask the controller that let the table slide move along the X axis, at let us say 20 centimeter per minute. Then the job of the controller is to maintain that achieve that 20 centimeter per minute and then hold it.

But, what kind of motion is going to be created, what will be the velocity, along the X axis and the Y axis, these are the reference inputs to the control system, and they come from a different block called the interpolator. So, we are going to look at this interpolator, in some detail today, so these are the two things that, we are going to look at, so this is a control system and this is the interpolator, so the interpolator tells the CNC drive, as to where it wants to go, where it wants the table to go, and the and that control system achieves that, this is the way it works.

(Refer Slide Time: 06:53)



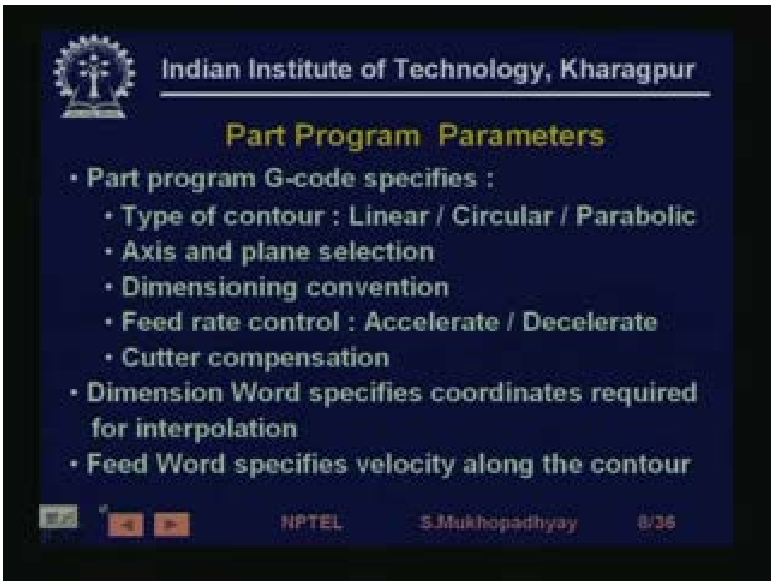
And, similar is the case for closed loop systems, the difference between an open loop and closed loop system, is that you might have noticed, that in the open loop system, there is no feedback from the table motion. So, the interpolator kind of assumes, that once it asks, the table to move along a certain direction, it will move, so typically, so that means, that it has to be guaranteed, and which now to guarantee that, you have to put certain kinds of restrictions on speed etcetera, because if you try to move it too fast.

Then, sometimes if you have stepper motor drive, we will see when we study stepper motors that, some pulses may be missed and things like that. So, this puts some restrictions, plus the whether the whether we also need to ensure that the, load does not become too much, so these are used for smaller machines. We otherwise the, you might give a command to the motor, but the motor may not be able to execute it, and you are since you are not sensing it, whether the motor has actually executed.

So, you will be assuming that it has executed that motion command, and in which case your dimensions are going to be wrong. So, I mean closed loop system, as will seeing that we there are this is the actual motion, this is the actual motion, that we want to create, in this case we have only shown one axis. So, this the motion sensing device or the encoder, sometimes you may use a resolver, so the encoder from the encoder, you get feedbacks, right.

So, feedbacks are used to, control the used in the control system, this, the control system this is the controller, this is the drive and this the motor, and this is the final mechanism, which we want to move. So, these encoder feedbacks, and these motors speed feedbacks etcetera are all used in the control system. And they are also used in the interpolator, because the overall cutting is actually a sequence of commands, which the interpolator gives, the interpolator has to, in a feedback system the interpolator has to know, when it is first set of command has been executed and it needs to issue new commands. So, there are lots of feedbacks and these systems obviously, do not make assumptions and therefore, they are generally more accurate, and they can work on higher loads.

(Refer Slide Time: 09:36)



So, no we will first, take a look at the I think, so you see that, what kind of part dimensions we want to cut, is actually specified in a part program by the engineer. So, we have seen the structure of part programs, part programs contents blocks, program blocks, in which there are kinds of codes which specify, that for a given set of operations, what are the parameters and what needs to what exactly needs to be done, what amount of feed, what kind of tool, which axis motions will take place, what kind of interpolations will happen, etcetera.

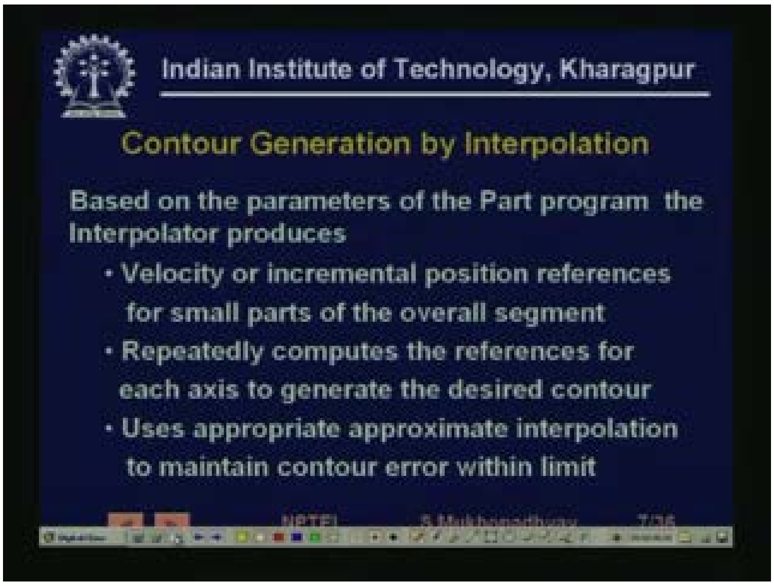
So, we have for example, there are some it is written in terms of codes, and there are some codes called G codes, which give it, give information to the interpolator as to what kind of contour it is, whether it is a, suppose you have specified two end points. Then, the question is, that whether the cutter will go, in a will make a linear interpolation about this, about this points and move like this or will it make a circular interpolation. Now, if it makes a circular interpolation, then it also has to supply additional parameters, that where, is the radius of this circle, because through, this two point a large number of circles can be drawn.

So, once you, so you provide this two end points and you provide the coordinates of the center of the circle, which it must follow, and then it will follow that circular interpolation. So, this types of contour are provided, then this interpolations are going to be created, in which axis or in which planes, so that is selected, various dimensioning conventions, whether they are absolute or relative as we have seen in the earlier lesson, have to be provided, feed rate controls have to be provided, that is at what rate, it will move along that trajectory, but will a move that what velocity.

So, that cutting speed, determines things like you know surface finish etcetera, so that when it needs to accelerate, when it needs to needs to decelerate etcetera. Then, there is something called cutter compensation, because you need to ensure that the, motion is that the, your dimension is actually created on the surface of the part, but not at the center of the tool, which is cutting.

So, there is some cutter compensation, which has to be done, which we are not dealing with. Several, various coordinates have to be specified and these are specified, in what is known as dimension word and the feed word specifies these feed rates, and their control, so at what feed rate. So, such information what I am trying to say is that, these information’s, which are input to the interpolator are provided in the part program, so now let us see, what the part program does with it.

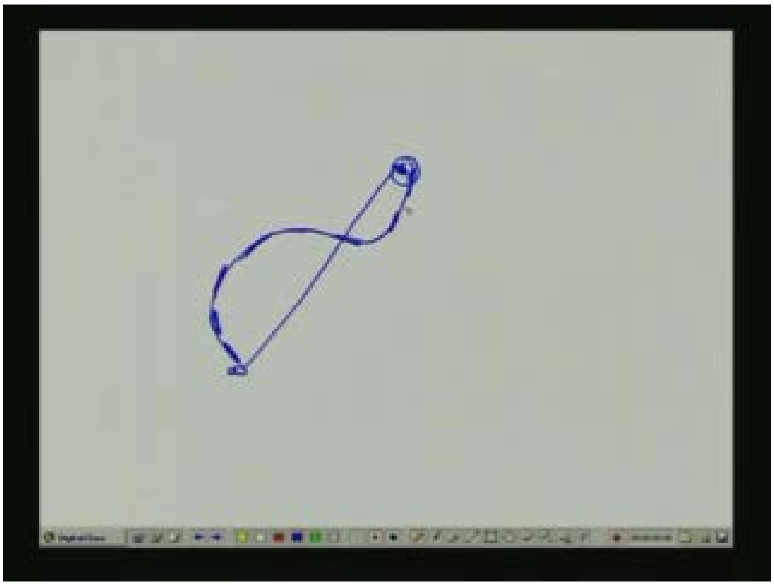
(Refer Slide Time: 12:33)



So, based on the parameters of the part program, the interpolator produces, velocity or incremental position reference, you see there are as we have seen that, there are two kinds of systems, one is point to point system, and anther is contouring system. So, in a point to point system, while that while the motion is taking place, right suppose you have a drill machine, so when the drill is moving from here to here, it is not cutting, while in a contouring system, while the table moves there is cutting.

So, for drills the final position, two positions are actually important, that will decide, where the hole is drilled, let us say on a plate. But, from this point to this point, can actually, go along any track and for saving time it generally takes the small, straight line track and moves at the fastest possible speed, because it is not a cutting at that time.

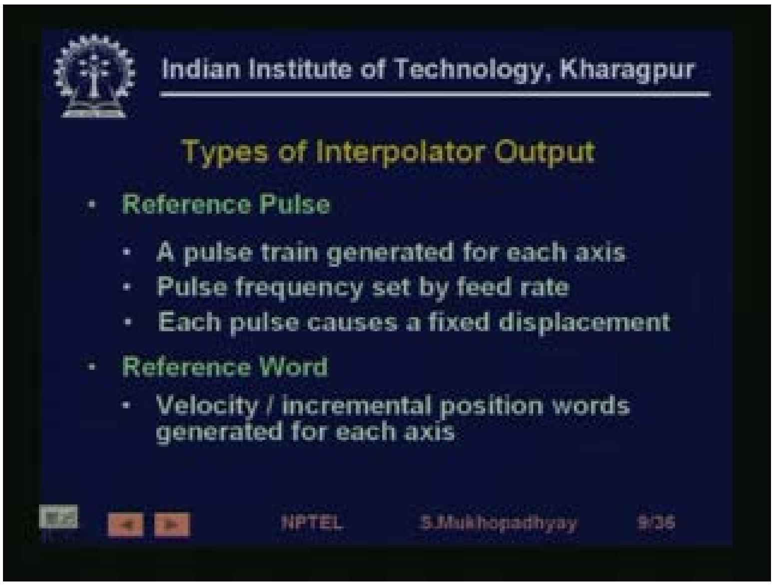
(Refer Slide Time: 13:39)



While in a contouring system, suppose you want to, it is important to understand, that suppose you want to go from this point, let me take a blue pen, suppose you want to go from this point to point. So, in a contouring system, whether we are going like this or whether we are going like this, it makes no difference, because it is the end points which are important. While in a rather, in a point to point system it is not important, but in a contouring system, it is very important, because you can see that, the contour between these two points are in these cases are completely different.

So, the contouring system actually, continuously along the trajectory it has to maintain velocities, the write VX and VY values, so that the desired contour is actually generated. So, velocities are incremental position references for small parts of the overall segment are to be generated, and these are to be generated repeatedly to… And then send as references for each axis, that is XYZ axis drives to generate the desired contour. And we have to generate them, we have to compute them, re compute them, repeatedly and so often, that the contour error is left within a certain minimum, certain maximum rather,, so this is the job of the interpolator.

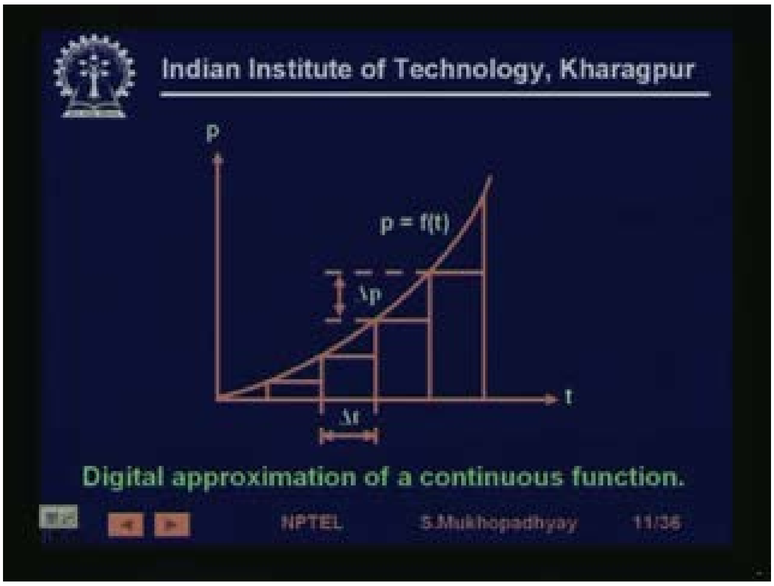
(Refer Slide Time: 15:06)



So, with this, now how does the interpolator give command to the drive or the control system, so there are two systems, one is called a reference pulse system, where pulses are provided at sudden frequencies to the motion controller, and each pulse indicates that a fix displacement along, that axis should be made. So, small pulse, each pulse will make a small displacement, and these pulses, are send to the different axis and the pulse frequency, at what frequency we are going to send it, depends on the feed rate.

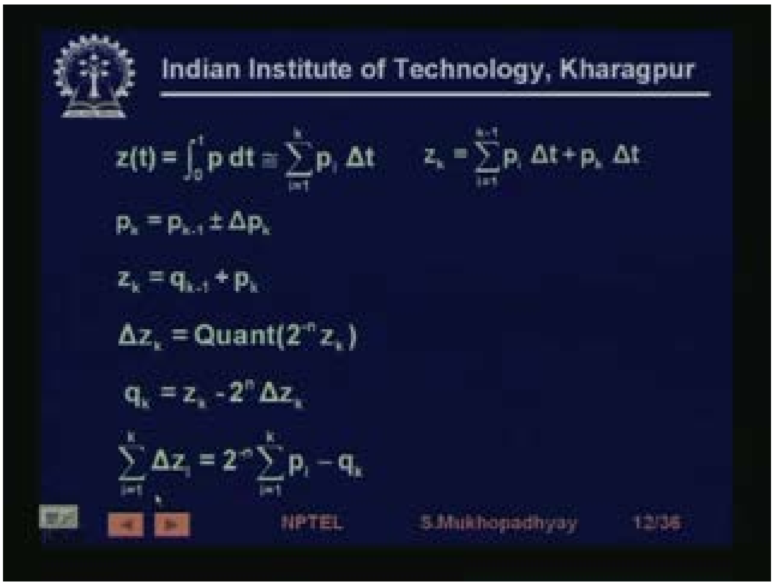
So, you are giving single bit, so the interface is a single bit, which is either 1 or 0, on the other hand, you have a reference word system, where you actually, send the hole coordinate to the, to the motion controller. And, say that, either achieve this velocity or achieve this position in X and Y, so in one case you are sending a number, multi bit number, in another case, you are sending a pulse and the sort of the some of the pulses which actually indicate the, motion that you want to create.

(Refer Slide Time: 16:25)



So, let us first, take the case of a linear interpolation, that is the simplest and see how a reference pulse linear interpolation can be done. So, first of all, we have to understand, that we are going to send pulses at discreet intervals, so if you want to create a straight line, then a straight line is nothing but, an integration of a constant. So, if we can send, if we can have, if we can make an integrator, and frequently is add a word, add a constant number to it, then that integrator value will actually rise along a line. Now, this integrator value, can stand for the coordinate motion, now so while doing that, now only thing is that, we cannot, we cannot send we do not want to send multi bit number, in the case of a reference pulse system, you want to send single bits, so how do we do that.

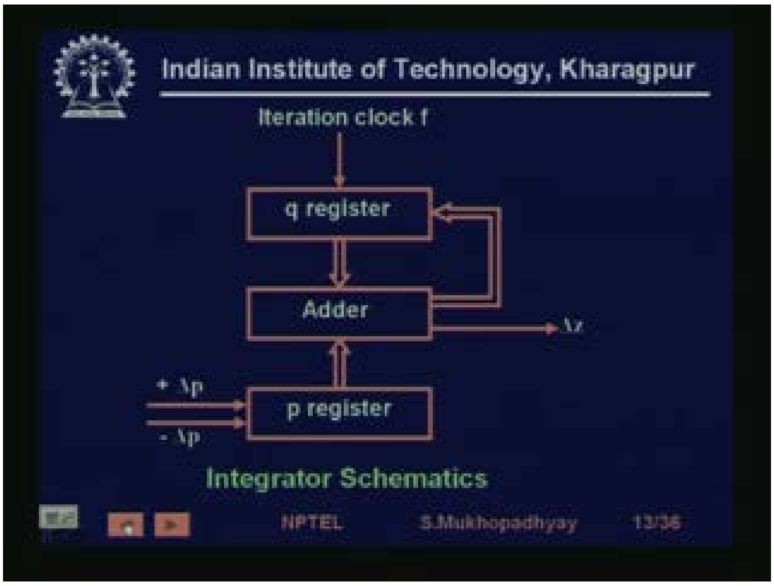
(Refer Slide Time: 17:32)



So, our simple strategy is that, we keep on adding a fixed increment, and whenever, this sum exceeds a certain number, then we send a command to the motion control system, that now, make this motion. So, it is not that, every time we are going to send a multi bit number, but when the sum of these numbers, so we go on adding a number, to an integrator, and when this in, and this increment becomes larger than a certain quantity, then we give one command, to the to the machine that you, now move make one motion along this axis.

So, mathematically speaking, this is very simple, that we want to integrate a certain number P, we want to integrate a certain number P, so discrete approximation if we make, then we can write that Z is sigma., so this is my previous value of Z, we can I can say that, this is Z k minus 1. So, I keep adding, this P k into delta t, to Z k minus 1 to get Z, it is a standard rectangular integration. Now, so we compute like this, now what we are going to add, that we can also change, in the case of linear interpolation it need not be changed, and it can be, because we every time we are adding the same constant P. But, if we want to have some other kinds of interpolation, like we have, like if you want to have circular interpolation or we want to have exponential interpolation, then we need to change this P, so we actually have this, fixed length resistor q.

(Refer Slide Time: 19:18)



Let us, I think we should see the picture first and, the so you see this is the picture, so you see, this there is this q resistor which is finite this is a fix length resistor, pretty small length. So, what we do is, with and this is my p resistor, which gets added every time and by using these two plus and minus inputs, I can increment the value of p or decrement the value of p, in the case of linear interpolation, it need not be exercised p is constant.

So, we keep adding, this p every time a clock comes, we keep adding this p to q, so it will happen, that there will be an overflow after some time, because q is a finite resistor, so we have, we are adding a number to it is after sometime, the number will be such, so large, that it can be held in q. So, at that point, there will occur, what is known as overflow, now every time an overflow occurs, a pulse command is actually goes to the right system, that now an overflow has occurred, so now move.

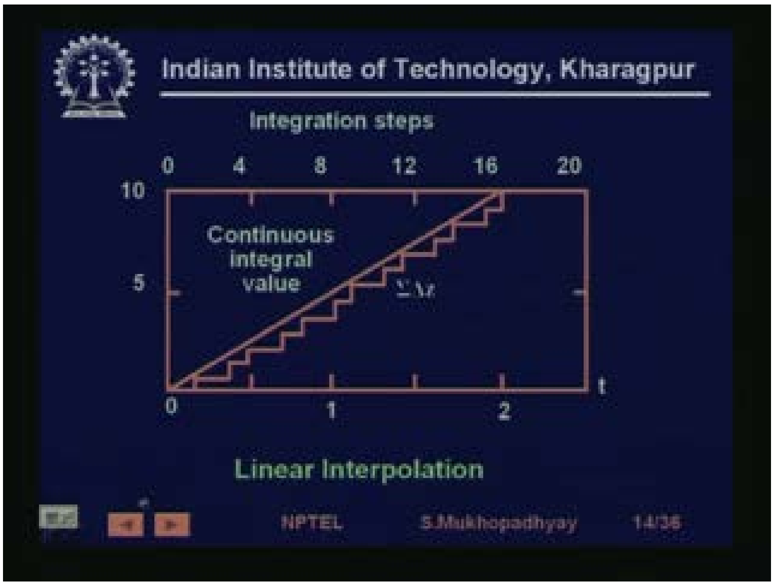
And, this and after and this overflow is actually it, so when does overflow occurs, when suppose, you have an n bit resistor overflow occurs, when the when the number becomes two to the power n, 0 to 2 the power minus 1, 0 to 2 the power n minus 1 it can hold. The moment the number becomes 2 to the power n, there will be an overflow or more, so now we take this delta Z out and whatever, is remaining, after taking 2 to the power n out, goes back to q resistor and again p gets added a bit.

So, every time there is on overflow in the addition, we send one motion reference pulse, to the motion control system, this is this is the principle. So, now we can look at, so what we are doing is, every time we are adding, we are getting to get Z k, we are adding P k with the earlier value of the q resistor, and then we are quantizing 2 to the power minus n into Z k. So, whenever Z k becomes greater than 2 to the power n, this quantization result will produce a 1.

So, this 1 will now go as a reference pulse to the motion control system and will produce a motion. And, what is now remaining in the q resistor, this has been sent to the motion control system, so whatever is remaining from Z k is goes to q k and again p k gets that, so this is the way it gets done, and we can easily see, that the overall integral, which gives at k, that k is actually the sum of, Z k is actually the sum of p i's that is the true value of Z k.

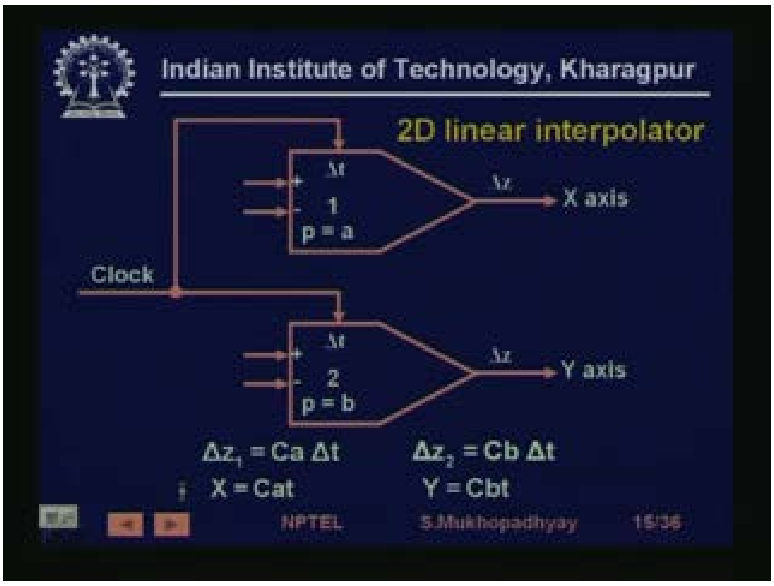
But, we are, what are we actually sending to the control system, we are sending delta Z i, but delta Z i are actually related to delta p i, because this is nothing but, this minus q k, because every time, the rest of the value is stored in q k. So whatever, is remaining is stored in q k, so actually this number, which is being sent to the motion control system and this number, which is a true integral, can only defer, by maximum by the length of q k. And so the difference is always bounded and so as the real integral rises, this stepped integral delta Z s, will also rise and the and the error between the real integral, and the stepped integral, will never exceed a certain finite bound.

(Refer Slide Time: 22:56)



So, that is what is being ensured, so you see, this is what is happening, that this is, so I do not know why this is happening. So, you see that, this the real integral and this is the sigma delta Z, and it follows that.

(Refer Slide Time: 23:27)



Now, we are talking of two dimensional interpolations, so far we are talking of integrating along one axis, now actually we have two axis, suppose we have a table, we have to move the table in this plane and the milling cutter, let us say, is rotating here. So, as the part moves this way, this way, that way the cutter will cut and create a, will cut the part, so we need to create two dimensional motion in this example, so how do you do, two dimensional motion, so you actually have two of, two such integrator obviously.

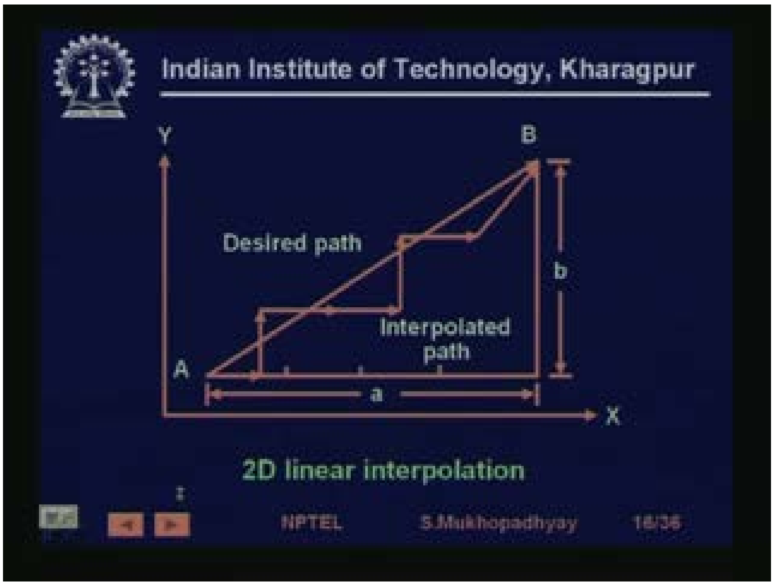
Now, the thing is that, they may have to be moved along different rates, like for example, if you are moving along the 45 degree, imagine the X Y coordinate imagine the X Y coordinates. So, if you are moving along this axis 45 degree, then the velocity along this axis and the velocity along this, axis has to be maintained, on the other hand, if we are moving along this axis, then there has to be more velocity along Y axis, and less along X axis.

They are related by the orientation of the line, that you want to cut, so obviously, and the slope is determined by what, determined by how frequently, the delta Z commands are given along that axis. So, it depends on, how frequently an overflow is going to occur, so now, when will overflow occur frequently, when the number p which being added, will be larger, so if we add, a high value of p, then it is possible that every, time you add it there is going to be an overflow, an on a average.

If we add a small number p, then it may happen that every 4 or 5 times you add p, you are going to get an overflow. So, in this way, the so by setting the value of p, the ratio of the velocities along the two axis can be controlled, and that is being shown here, so you have two integrators. So, this is one p and this the other p, so you in one case you have p equal to a, and in other case you have p equal to b, and you can find that, the x and y coordinates will, so the slope is going to be decided as b by a, so the slope of the line that it will cut, is going to be b by a.

So, this is the way, by the way these integrators previously in the era of ordinary NC machines, these things used to be realized by hardware, using you now, a detail electronic circuits, but now they are realized in software. So, as for as these diagrams are concerned, they are actually you know, like subroutine function blocks, which gets executed and this clocks are not the not the hardware clocks, which actually go to a hardware, but they can be you know like interrupts. So, every time the interrupt comes, then these routines get recomputed, and an output goes, so these diagrams has to be interpreted as like that.

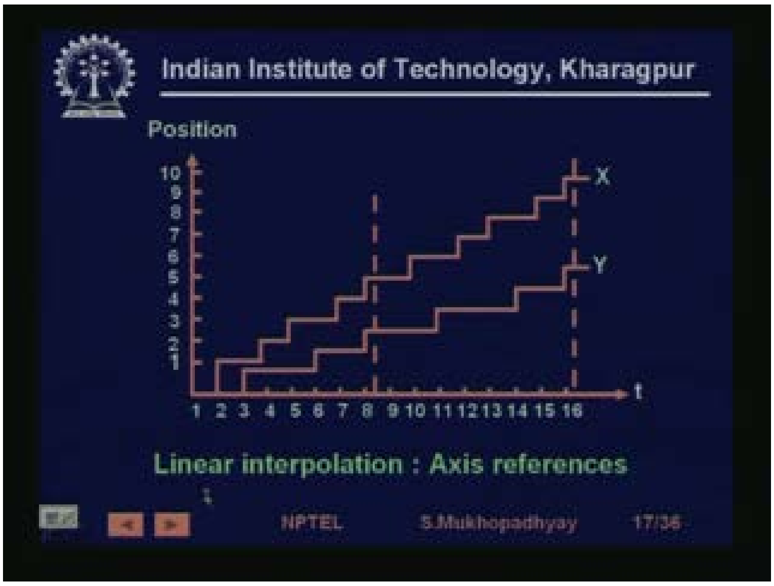
(Refer Slide Time: 26:36)



So, you see, that this is the case, where we want to have, let us say, this is b by a, so you see, that to be able to generate, now you see the contour is important, so it is not that, we can first give a amount of x pulses and then give b amount of y pulses. So, that so even then the final position of the tool is going, to go from A to B, but it must follow, this path as closely as possible, that is contour generation. So, you see, what is happening is in the first interval an x pulse is given, but no y pulse is given.

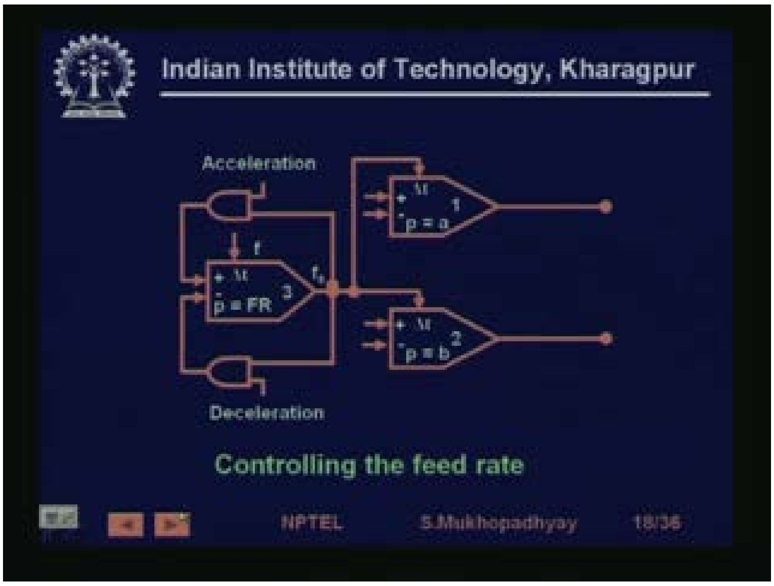
In the second interval y pulse is given and no x pulse is given, then two consecutive x pulses are given, y pulse and x pulse, and here both x and y pulses are given. So, it seems both x and y pulses are given, so the table actually moves along these direction, because both the motors are now rotating simultaneously. So, this has to be determined, so this actually is determined by this, automatically gets determined by this overflow frequencies, so whenever, you get delta z for x, you get a x pulse whenever, you get delta z for y, you get a y pulse and they may occur simultaneously also.

(Refer Slide Time: 28:04)



So, you see, this is this is a pattern, by which this the previous one was X Y plot, this an X and Y versus time plots. So, you can see that at various instance for example, you can see that during this instance the X and Y pulses are coming simultaneously, so this is the same thing.

(Refer Slide Time: 28:29)



Now, how do we control, that sometimes you know you need to control the feed rate, why you need to control the feed rate, because, also because a table with a part on it. See, sometimes you may like, that for example, let us say, that the classic example is a point to point systems. So, you may, you will like the motion to be created, that is you may like to move from point to point to be very, very fast, so you go up high speed and you give very fast feed rate, I mean motion rates, but then when you approach the final point, that you that is slowed down.

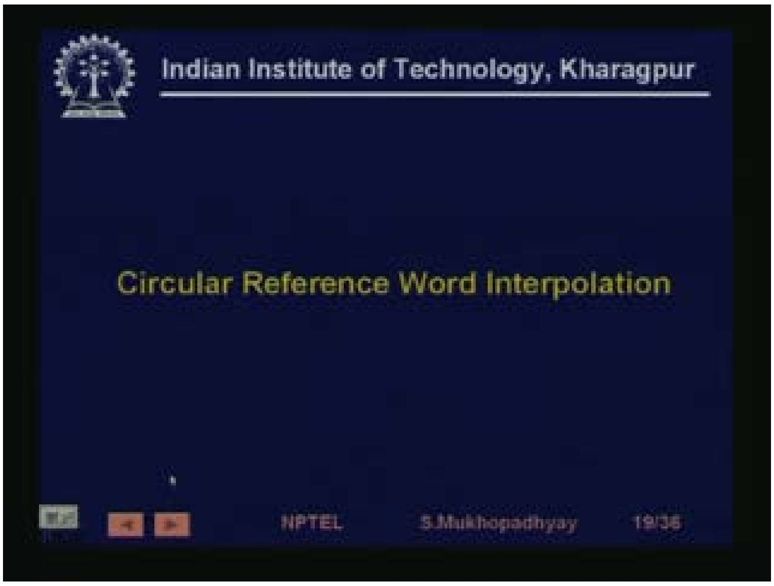
Similarly, you may also do for contour making machine, because if you are, if you are cutting the contour at a very high feed rate, then as you are in approaching the end point, if you do not slow down, then you are going to overshoot and you are going to spoil the part. So, to avoid doing that, we need deceleration and acceleration mechanisms, so this is one we are doing that, where what we are doing is you see, this is the clock, this is the clock, this is the clock and this is the clock.

So, the clock itself is being generated by an integrator, and so this is the feed rate, so this sets, this value of p sets the feed rate and these are actually enable inputs. So, when this is 1, then this pulse will feed, and when this is 1, this pulse will feed, when both of them are 0, these lines will be cut, and the value of p will be maintained. So, when you want to maintain, constant velocity simply set these to 0, so there will, you will get these pulses at constant rate, as clocks to this.

When we want to decelerate suppose, so you make this one, so now, what happens, is that we can actually, by analysis we can show that this is like approximately, it will be like exponentially, it will be like exponentially decaying the clock frequency. So, what happens is that every time this goes 1, every time you get a 1 here, that 1 will now propagate here, and it is connected to the negative input, so the feed rate will now be decremented by 1.

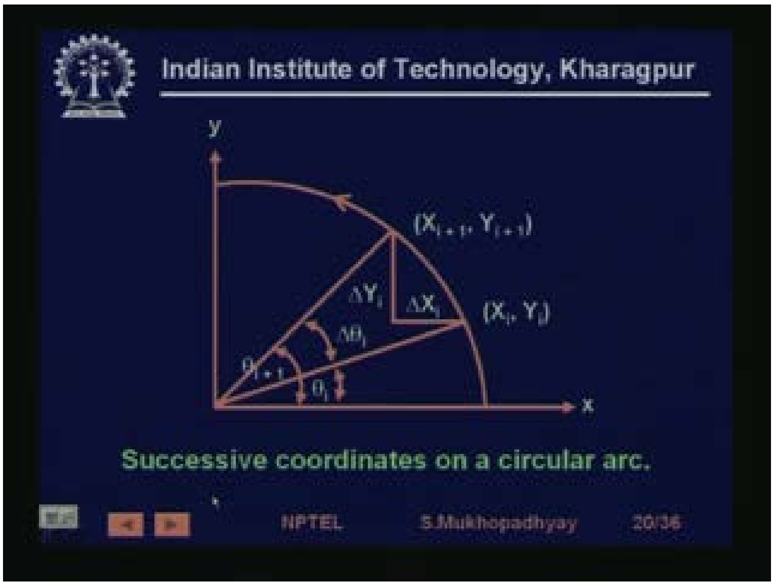
So, next time, a pulse is going to be generated slower again, when another pulse is generated, again this is going to be decremented. So, as pulses are generated, if this input is 1, then the feed rate comes down, and it can be shown that, it will, that will exponentially reduce the, so the pulse rate frequency will get, will come down in this fashion over time, so the in this way, you can one can control the feed rate.

(Refer Slide Time: 31:20)



Now, we come to a circular interpolation case.

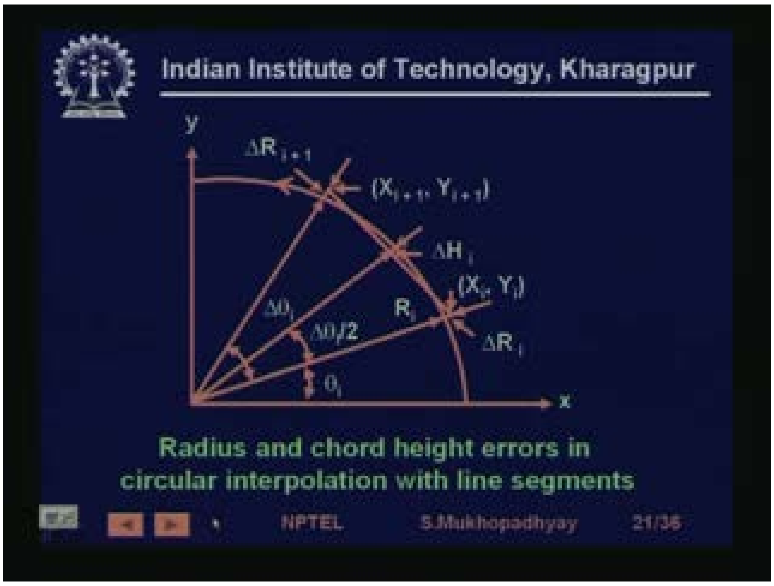
(Refer Slide Time: 31:26)



And, so what we see is that, so we want to, what we want to do, we want to move along this circle. So, suppose, so at some angle theta i, we are in this position and at an, after an angle delta theta I, we want to be in this position, now what is, one of the approaches is, is that you approximate the circle by a straight line, make this angle small, so you actually are traversing, very short straight line segments, to make the circle, and, if your segment is narrow enough, then you can almost achieve an arbitrary accuracy, very close accuracy.

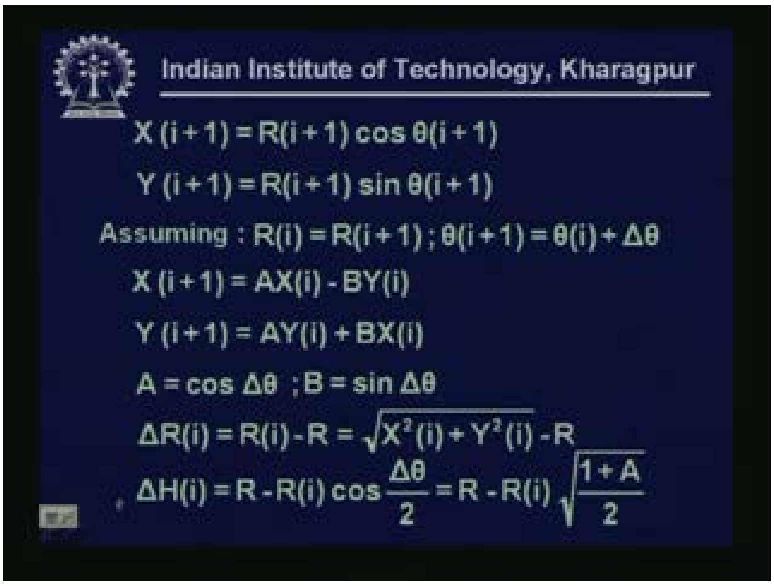
And in any case remember, that the accuracy of the machine cutting is actually, limited to what is known as the basic length unit of the machine that is the distance that it, moves to get you know one encoder feed, one encoder pulse, let us say. So, smaller movement then that cannot be, cannot be sense and therefore, cannot be I mean distinguished between, so therefore, there is a concept of the basic length unit, and we need to keep all I mean the aim is to keep the, dimension errors limited to that.

(Refer Slide Time: 32:57)



So, this is, what we are doing successive coordinates, so now, we want to we will see that, if we did a linear approximations. So, suppose we are here, slightly off from the circle and we are moving along this straight line to this point, so that we are going to incur two errors, I mean the error about the contour is actually, maximum there are the extreme are this is, these are the radius errors, delta R i's and delta R i plus 1, and this is known as a chord height error, which occurs in the middle, so we need to keep these things in check, so some elementary math's, it is the simple trigonometry, will tell you the following.

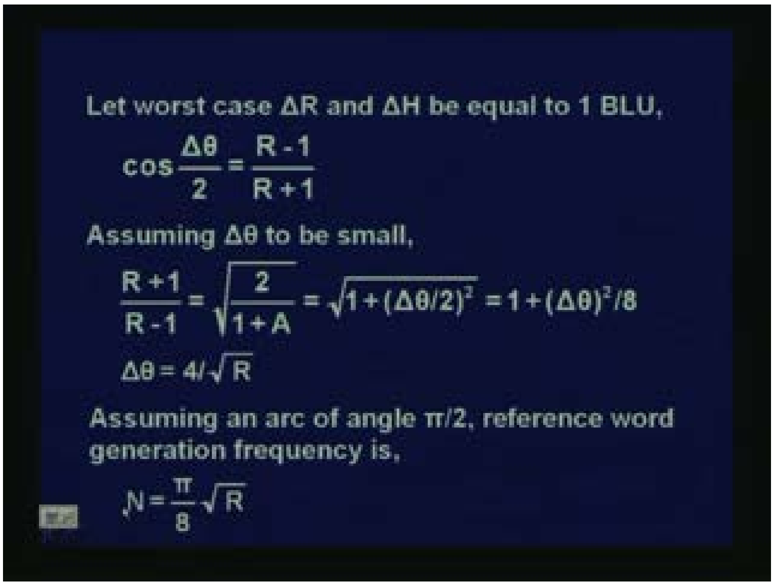
(Refer Slide Time: 33:43)



These are some simple relations which hold, for example, this X i plus 1, equal to R i plus 1 into cos theta i plus 1, Y i plus 1 equal to this. Suppose we assume, that we do such an interpolation strategy, such that R i is equal to R I plus 1, and it may be a little different from, that real R that we want, and suppose we assume that, these segments over, which we are generating trajectory is a by a moving by a constant angle delta theta. Then, we can easily show, that this becomes a linear relation, that is X i plus 1, is given like this, where a is cos del this is just, you know cos A plus cos B formula, so and B is this.

And, this is my height error, a radius error R i minus R, and this my chord height error, which is R minus R i cos delta theta by 2, if you see the triangle, it will be immediately clear. For example, so now cos delta theta by 2, is written in terms of you know cos 2 theta cos theta formula, so an A is cos delta theta, so we have this formula that, cos delta theta by 2 is actually this part.

(Refer Slide Time: 35:01)



So, now we can, now suppose, we want to have such a, we want to know for example, first of all, that if we want to restrict, this delta R and delta H to one basic length unit, then what is the obviously, as if we make this delta theta smaller and smaller, then this linear approximation will approach the circle. So, how often, do we need to re compute this lines commands and then give the corresponding command to the machine, this is the question, that we are following that, we trying to solve.

So, because, the smaller this angle is the faster we need to re compute this and well, it may give some you know it is a computational burden on the machine. So, we have you, know this is a standard formula, that cos delta theta by 2, is R this is simply geometry, now if we assume delta theta to be small, then we can you know assume, make such approximations that sine delta theta equal to delta theta. and 1 by 1 minus delta theta, 1 by 1 minus X equal to 1 plus X, and things like that.

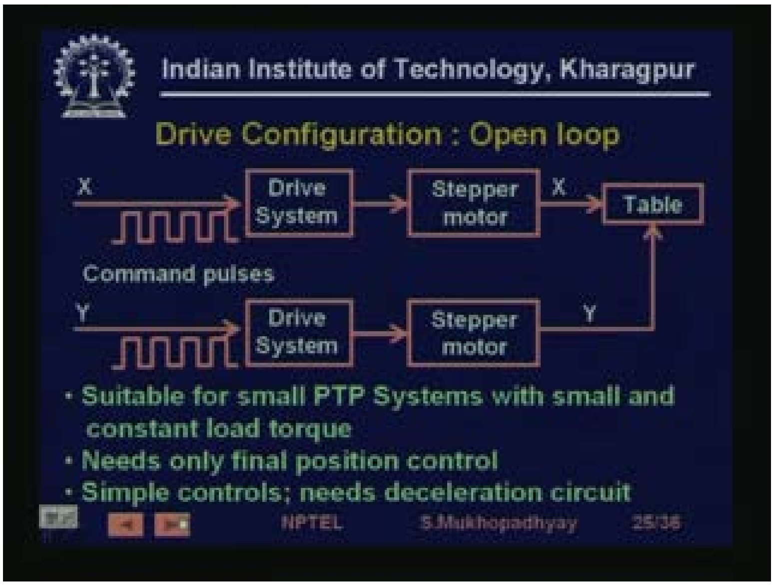
So, if you do such approximation, this is easy to derive that, this is your R plus 1 by R minus 1, so if you want to have, now so from there you can find, that what should be delta theta, such that, the radius errors are limited to one basic length units. So, it turns out, that it should be 4 by root R, and so how many times do you need to compute, normally these arcs are provided as, maximum arcs are provided as quarter of circles. So, if you have an arc of angle pi by 2, then the reference word generation frequency is pi by 8 into root R. So, this tells you, that how frequently, you should generate the new reference words to be able to restrict, errors within one basic length unit.

(Refer Slide Time: 37:06)

So, what we have seen so ((Refer Time: 37:10)), we have roughly seen interpolation, there are various other strategies. We have seen, how to generate linear interpolation, by very simple integration, we have seen how to generate, circular interpolation using linear interpolation seen that, by frequent enough generation of lines, we can keep errors in check, we have also seen how to control feed rates right.

So, for many of the parts these are you know, trajectories are formed in terms of circular and linear segments, so that is the way the interpolator, gives commands to the control system, now we are going to take a look at the control system itself.

(Refer Slide Time: 37:55)



So, again come coming back, we have two drive configurations and, so we have two drive configurations, one is open loop, so just pulses are coming to the drive system, these are typically applied to stepper motors. Because, in a stepper motor if you give one pulse, then the motor will rotate by, by a fixed angle, which is it is resolution, which depends on its number of poles, number of stags etcetera. But, it will, it moves from one stable position, you give it one pulse it will move by a small angle and then becomes steady in another stable position.

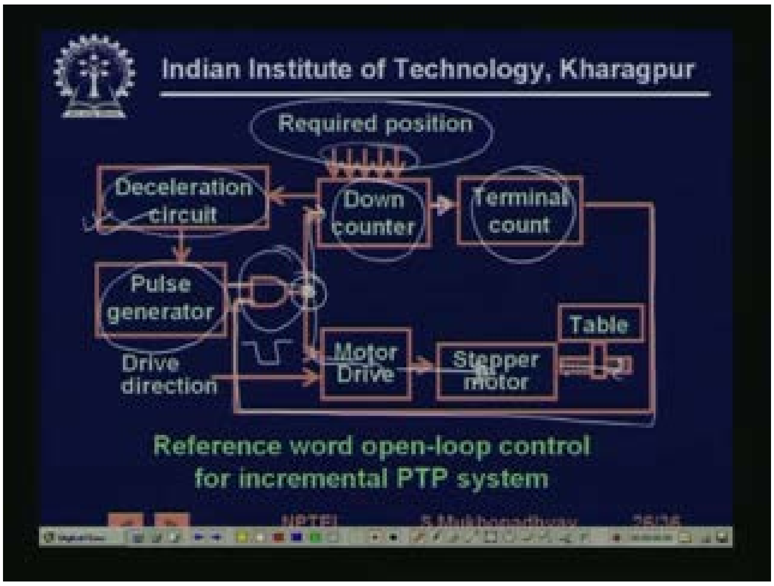
And, if you give it another pulse then it will move another time, sometimes if you want to increase the velocity, before it really comes to another stable position and stops we give a pulse. So, before it can, it stops, it gets that pulse and then moves further, but it is seen, that if you give these pulses too fast, then some of these pulses will be missed, by the motor, so the motor will not respond to them, as if that pulse is lost, so what we are that is the first thing, second thing is that, this whether the motor will, move or not that also depends on the load, now stepper motors are typically designed for smaller torque ratings.

So, first thing is that, they are useful for, they are useful for small machines, they have velocity restrictions and they need constant torque environments. Now, when a now the torque, on a machine is actually dependant on the feed rate, depth of cut and the materials etcetera. So therefore, they are typically, these kinds of ((Refer Time: 39:53)) are typical use of point to point system, because when normally because in a point to point system when the, let us say when the, drill is moving, it is not cutting, so the load is constant, it is also small.

And, another thing is that the trajectory is not important, only final, only the first and the final positions are important. So, you just, so you can move it, at a later will slightly smaller rates, so that, you ensure that, if you give a 5 pulses and it will move 5 distances, and it will not miss pulses. So, it is suitable for small point to point systems, with small and constant load torques and it needs only final position control, there is no contour required, because it is a point to point system.

And, main thing is that, they are simple, they are cheap electronic is simpler, you need deceleration circuits, because you need to bring down the velocity, otherwise because of inertia of the motor, it should not overshoot.

(Refer Slide Time: 41:02)



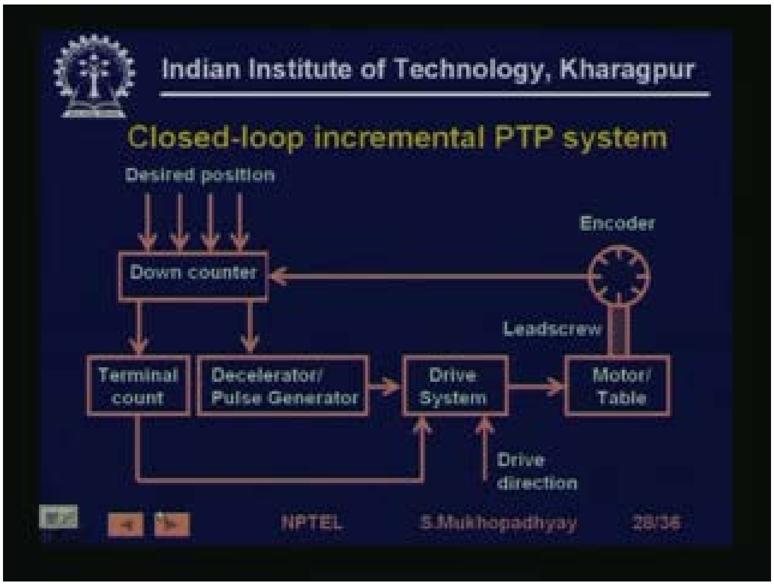
So, this is the typical circuit for a point to, so you see that, this is the reference word system, so we have the required position is actually, given by the interpolator, in terms of a digital word, that gets stored in a down counter. Now, this, from this down counter, it goes to a deceleration circuit, that is, if deceleration is required, actually what does the down counter, you see what is happening, is that through the, this is the pulse generator, which is giving the pulses, finite to the motor then the motor to the drive and the drive is giving to the motor, and then this table is moving.

Now, there is no feed backward, this is an open loop system, so when it gives, pulses to this it also gives, pulses to this, so the counter moves, so the counter counts down. So, at any time, the value held in the down counter, indicates how far it is from the final system, when the down counter value will become 0, at that time it has reached at the final position, so this deceleration circuit is actually activated, based on the value of the down counter.

So, if it is, close enough to the final position, then this deceleration circuit will decelerate these pulse rates, so that the machine slows down and finally, stops at the, reference position final reference position and does not do an overshoot. The moment the down counter becomes 0, this terminal count event has occurred and reached, so therefore, this pulse generator, pulse generator is nothing but, an oscillator, so that keeps on generating pulses, at the frequency which is controlled by deceleration circuit.

But, this pulses are enabled, by the terminal counts, so whenever, this terminal count is reached, then this N gate this is put to 0, and this N gate output will be 0, and no further pulses will go and the motor will stop. So, this is the way it operates, ((Refer Time: 42:58)) now if we have closed loop contrast to this there is a closed loop drives, which we have already seen that, there are various kinds of feedbacks.

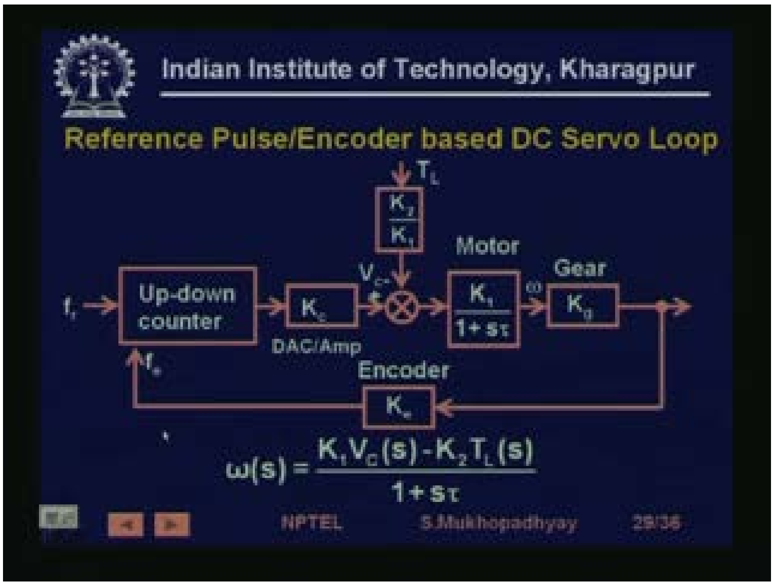
(Refer Slide Time: 43:12)



So, let us look take a look at closed loop systems, in closed loop systems, you can afford to, drive the system at a higher speed, because you are sensing the, because you are sensing, whether actual motion has been made by the, by the table, and you are decrementing your down counter, using that value. So, even if there is, even if the motor, misses one pulse, the down counter value is not going to be incremented and therefore, the right number of, motion creating pulses will finally, come to the motor.

So, you see, the main difference here, this is absolutely similar, the main difference here, is that the feedback is going from by through an encoder. So, this down counter is going to count, is going to be its values going to be decremented, only if the motor rotates and the table actually moves. Previously it was every time a pulse is given to the motor, the counter was being decremented, on the assumption that, if a pulse is given to the motor then the table will move, that assumption is not here. So, this is the basic difference, otherwise they are same and you can drive it, a little bit faster and with a little bit higher loads, if even if loads vary, and there is some problem, so at least the counter will not be decremented.

(Refer Slide Time: 44:37)



Now, we the other kinds other than stepper motor, is the other kinds of motors, which are used are DC servo motors, now AC servo motors are also used, but we are not going to look at, you know actually AC servo motors are from a control point of view, behave pretty much like DC servo motors. In fact, the whole of our electronics and machines, research wants tries to do that, so if you look at DC servo loop, we will get pretty good idea about, how this thing works.

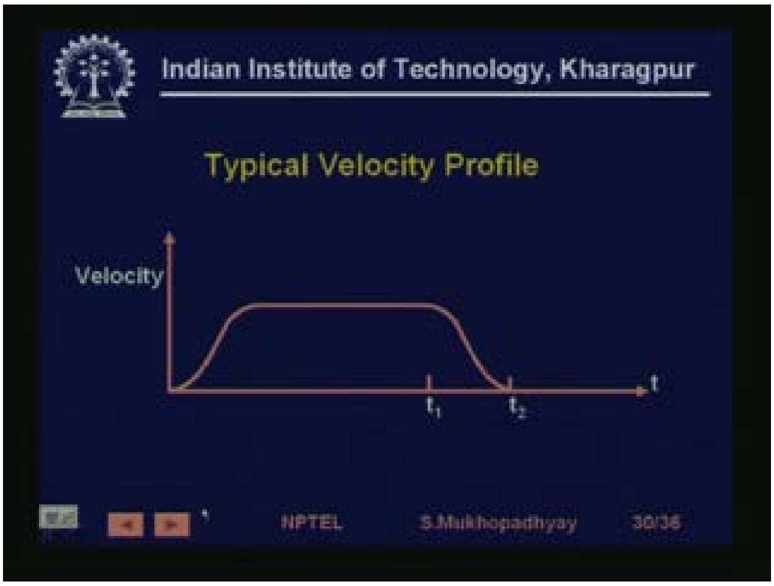
So, here, what we are doing is that we have a reference pulse system here, and we are driving a DC servo, so it is not a stepper motor case. So, what we are doing is that, this is the motor transfer function it turns out we can see that, when we see dc motors, that it turns out, that the speed, see we are trying to control the speed. So, basically in a motor if you, in a dc motor if you, change the armature voltage, then it will create, it will change the current flux is constant, so therefore, it will improve torque.

So, the motor will move and there is generally speed feedback, in such drives etcetera especially, so we try to control the speed. So, the speed transfer function and the speed will simply get integrated, to give the position, in fact, the encoder frequency, encoder every time it moves, it will give a pulse, so that if you measure the pulse frequencies, they are going to be proportional to the speeds. So, the speed is given, something like this, it is dependent on the control voltage, it is also dependent on the load torque.

And, if you, in neglect the mechanical time constant, then this is going to be the, this is going to be the, you will get a roughly a first order system. So, what is happening here, is that the reference pulses are coming into this up-down counter, this is an up-down counter, so every time a reference pulse comes, this counter is going to go up, and every time a the motor moves this is going to, go to the, this is a down terminal.

So, this counter is going to go down, so this counter actually senses the in a sense a see the counter is integrating whenever, it getting this. And it is integrating it is coming down whenever it is getting this. So, actually this value of the counter, actually indicates a position error, so based on that you are giving a control voltage and the loop is working inside the motor, there is a speed feedback tachometer, we have assumed.

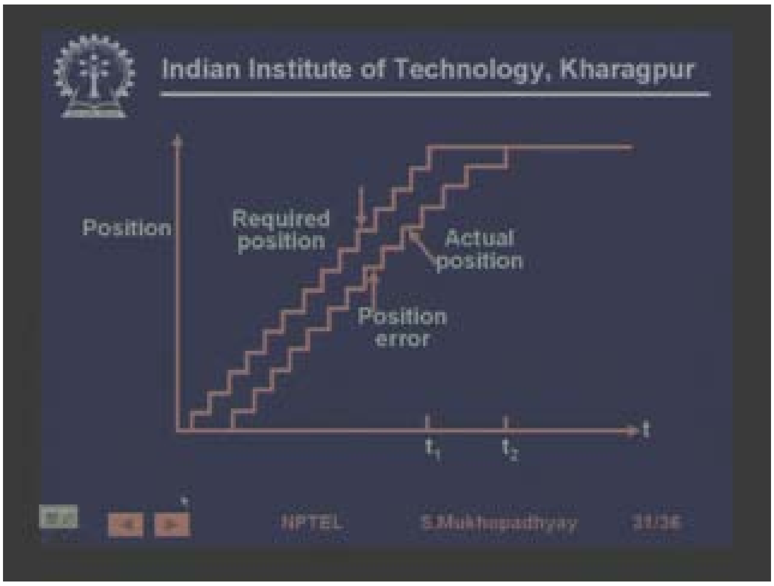
(Refer Slide Time: 47:26)



So, what kind of typical velocity profiles are going to get generated see initially, the initially the, lot of reference pulses will come and the motor will slowly start, it has an inertia it takes time to build velocity. So, the counter value will go up, as the counter value goes up, then the torque will also go up and the velocity will start to rise, see the velocity is increasing, that is there is positive acceleration. After sometime, as the velocity rises, the encoder pulses also go up, and then so the on an average the counter gets incremented equal number of times as it is decremented, and therefore, the counter value is maintained.

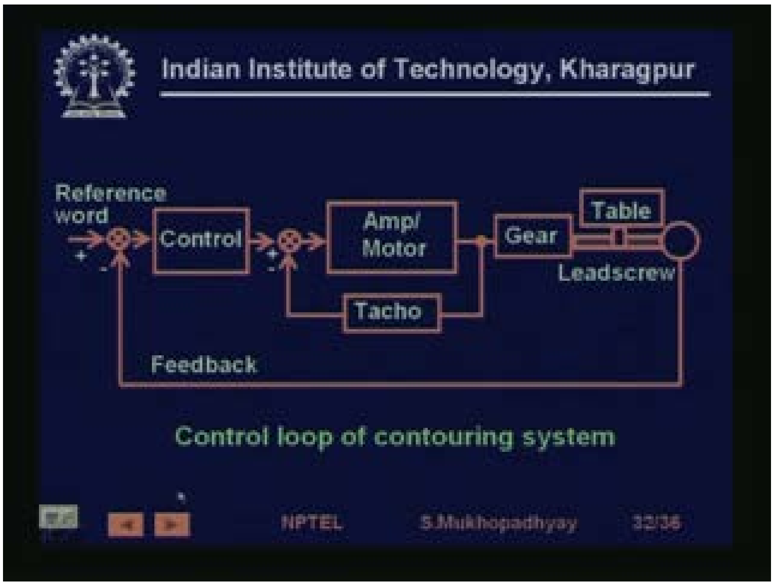
So, the velocity is maintained and then towards the end, what is going to happen is that the, this as the position error will reduce, so the reference pulses are will actually stop coming, they will their frequency will be reduced. So, now the, there will be more number of encoder pulses, so this error will, so the counter value will also come down, and then towards the end it will become zero, and then that time the motor will stop. So, there is a natural deceleration in the process, along with the fact that you can also put some you can slow down the reference pulses, using a deceleration circuit in the interpolator, so this is the typical velocity profile that may be followed.

(Refer Slide Time: 48:58)



So, you can see, that the position error actually change is like this that, if you just integrate the velocity profile, you will get this position error. It is shown in, shown discretely, so initially the required position is going up, because reference pulses are coming. For some time, the encoder pulses are coming slowly, so a position error builds up, that position error is maintained and then finally, that position error gets killed.

(Refer Slide Time: 49:26)

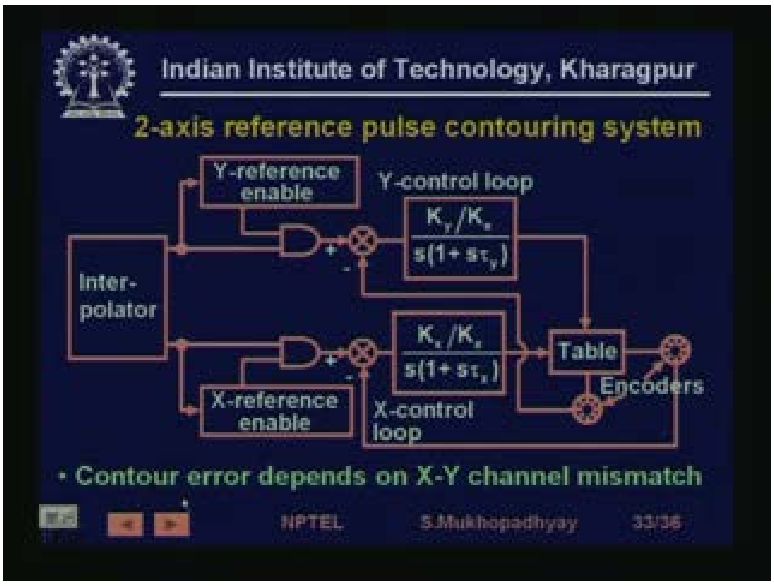


So, similarly you can have a reference word system, where which is a very conventional control system, where you simply use a, you give a simply give a position commands, it is a very conventional position feedback, using a, using this typical cascade control architecture. That is there is an, there is a loop inside a loop, so there is a tacho feedback for speed control and that is an actual feedback.

So, this is a very conventional control loop, which is used and remember that, this control loop designs are actually very important, because the transient response, of this drives are actually very important, especially when we at, when you are doing things like you know cutting corners. So, when you are cutting corners, then you can see that suppose, you are trying to cut a corner like this, so you are initially giving only Y axis commands, and then suddenly you are stopping the Y axis command and then you are starting to give X axis commands.

So, unless the transient response is good, then the trajectories are going to be, so the depending on the transient response the actual trajectory may be, different from the corner, that you get you at, so the transient response is a actually very important, for these systems.

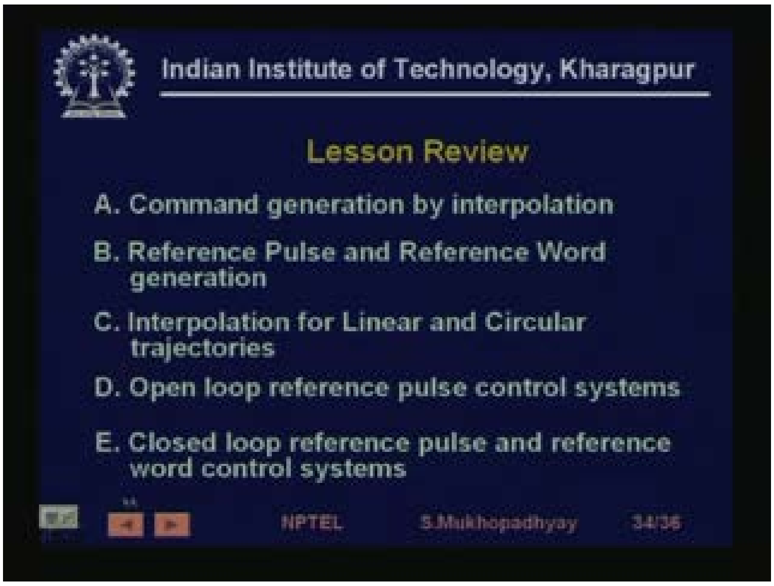
(Refer Slide Time: 50:42)



So, again with that was only one loop shown, and with respect to that, so now, you have two loops shown. So, this is the table, this is your Y axis control loop, and this is your X axis control loop, so you see the interpolator is giving reference words here, and the encoder is giving feedback, and rest of the control loop is just, shown by the transfer function. And there is an enable, so whenever the interpolator can enable these pulses, when it actually disables it, then these inputs go to 0.

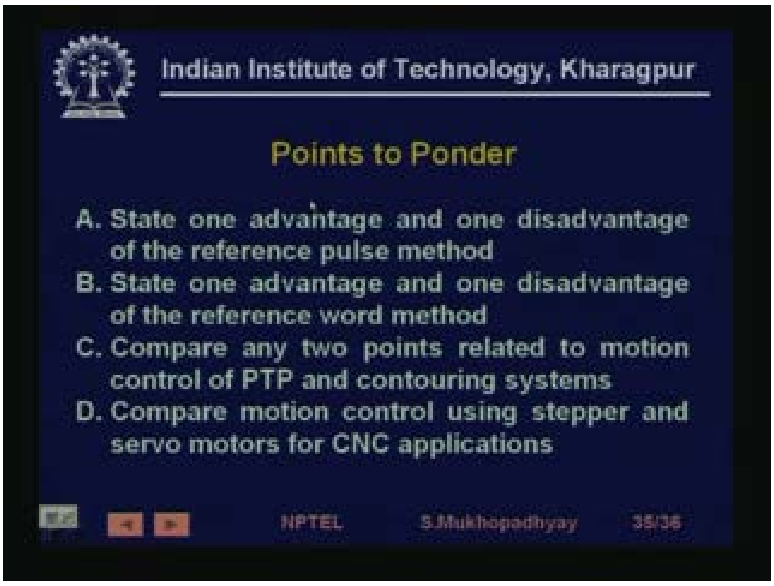
And then only encoder pulses will be coming and the counter values will go down, as we have discussed, so typically you have, so you can this is an architecture, for actually generating. So, basically it is a nothing but, a composition of the two axis drives, but we must remember that, the counter error, depends on X Y channel, this is very important, so that, this is one of the reasons, which why it, I mean N C machine controls are… So expensive is because that the channels have to be very carefully matched, because your interpolation will assume, equal response characteristics on the X and Y axis. And, if they are not matched, then they are going to be, then they are going to create contour errors so. Firstly, these loops have to be very high gain, and secondly, they have to be closely matched, so we are not going to the analysis of that.

(Refer Slide Time: 52:17)



So, we are coming so in this lessons, we have studied command generation by interpolation and reference pulse and reference word generation, we have seen how basic strategy for interpolation of linear and circular trajectories. And, we have looked at two different types of control, open loop reference pulse controls and closed loop reference pulse controls are typically used in open loop. Closed loop reference pulse and reference word control system.

(Refer Slide Time: 52:47)



Before we close, we will look at some points to ponder, so let us say, state one advantage and one disadvantage of the reference pulse method. Obviously, reference pulse methods, will have they require less interfacing, only one line going into it, they can interface with, things like stepper motors. On the other hand, the disadvantages are also obvious, we have seen that, there is always some error and so it follows the curve from one side, then state one advantage and one disadvantage of reference word method.

Reference word method is the basic thing, that is, in fact, reference words methods are getting, more and more popular, than reference word method reference pulse methods because now hardware and ADCs and electronics, and software are so easy to, it is becoming much more easier, to program and to build. So, reference words methods are actually conventional control system, where you give a number or a signal as a reference signal, and the rest of the loops are designed.

And, compare any two points related to motion control of PTP and contouring system, so first thing is that PTP systems are working generally motion takes place at no load, constant load, and while in contouring systems motion takes place while cutting, so load can be different, Load can vary and load is generally higher. And, in PTP systems, there is no need to control, in velocities while it is traveling, because there is no contour generation only final position, need to be controlled.

While in contouring systems, the while the derive is moving, the machine is cutting, so every at, every point velocities are important, so obviously, contouring systems are much more complex, both interpolation and control. And, stepper and servo motors, again stepper motors have resolution problems, they cannot they only move by certain steps, they are very, they are their control circuits are simple, while servo motors have theoretically infinite resolution and, but their control circuits are complicated. So, this gives you some idea of the way, motion is generated in CNC machines.

Thank you very much.

Industrial Automation & Control

Prof. S. Mukhopadhyay

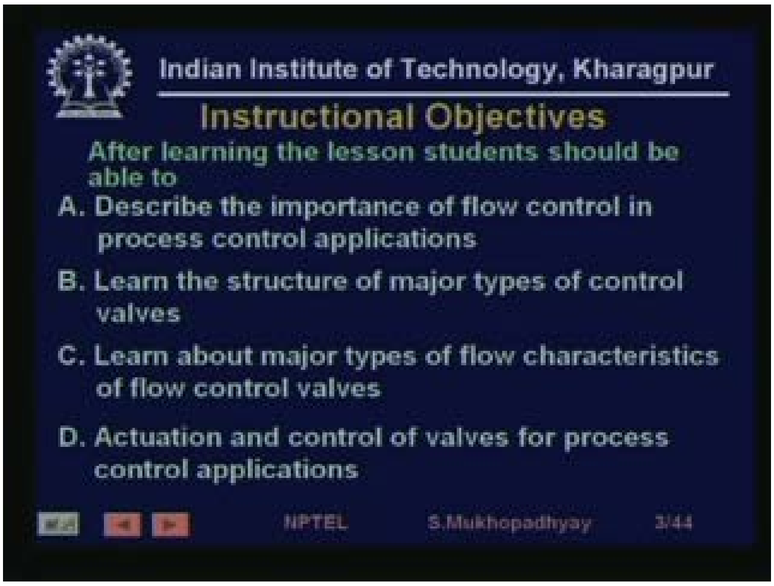
Department of Electrical Engineering

Indian Institute of Technology, Kharagpur

Lecture - 25

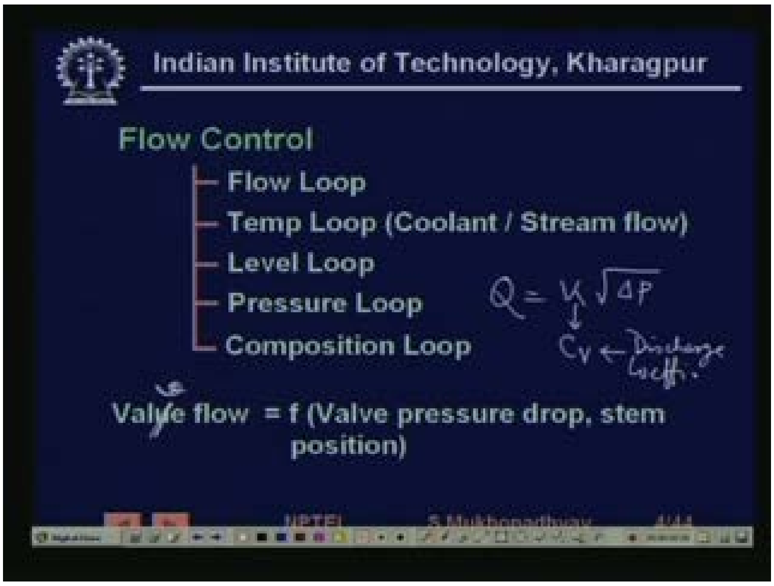
Flow Control Valves

(Refer Slide Time: 55:45)



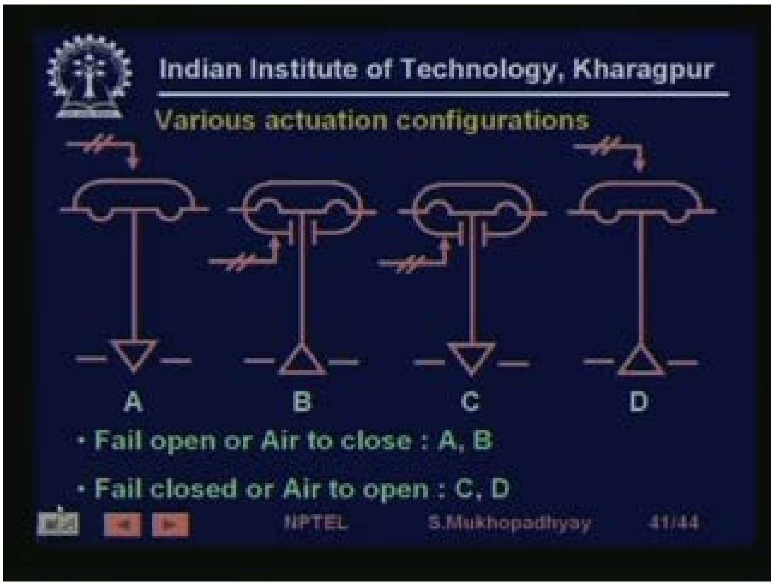
Welcome to lesson 25, on flow control valves of the course on industrial automation flow control valves are very important, so after learning the lesson, the student should be able to describe the importance of flow control valves, they are found everywhere in process industries. Learn the structure of major types of flow control valves, learn about the ((Refer Time: 56:10)) their flow characteristics, because that is very important in designing the applications. And finally, the how to actuate these valves and how to affect their characteristics, to achieve a certain characteristic of the process control loop, so these are the topics, that the student is expected to learn from this lesson.

(Refer Slide Time: 56:38)



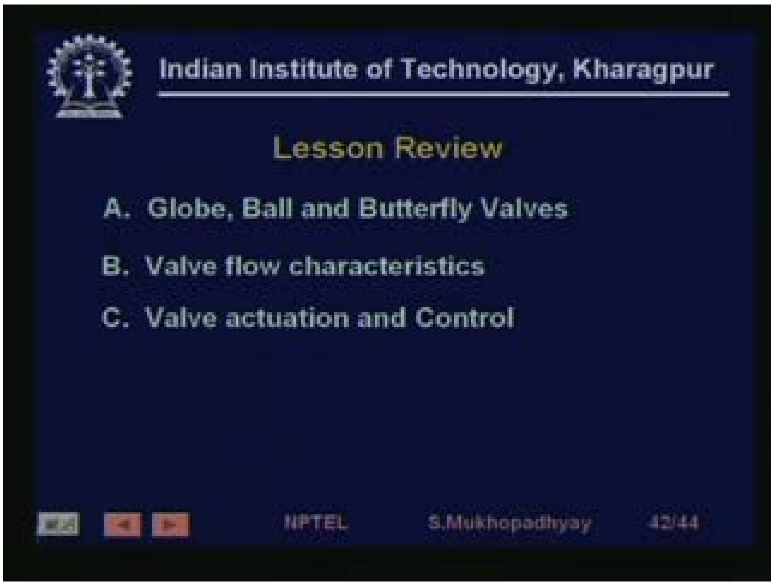
So, the first of all, let us have a look at, the importance of flow control, flow control is probably the most important control in a process control application, and as we shall see, during our process control module, that flow control loops, form a part of most type of control loops. For example they are parts of flow loops, where directly flow has to be controlled, flow is a final objective of control, they are parts of temperature loops, because temperature is generally controlled by controlling flow of either a coolant or and let us say, a steam for heating.

(Refer Slide Time: 57:20)



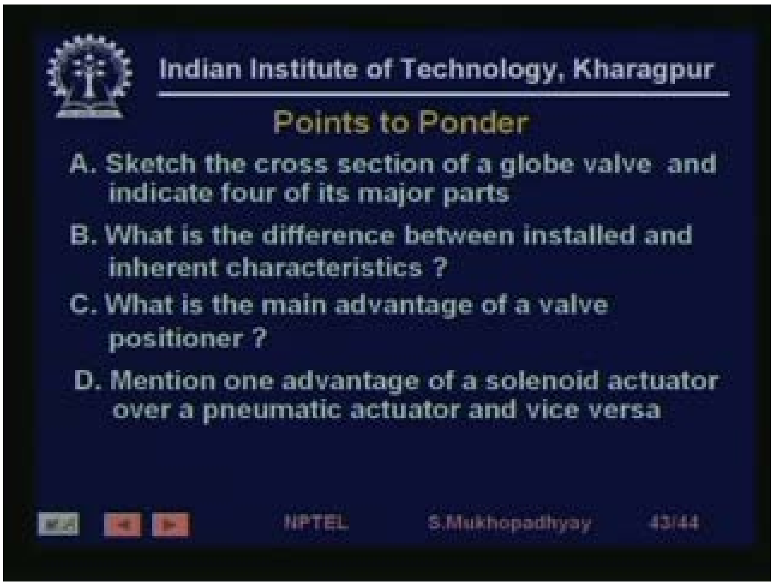
Similarly, there are airs to open valves, where if the air supply closer fails, then the valve will close. So, these, so you one has to choose a particular actuation configuration, to you know, avert industrial accidents under various kinds of failures, so that brings us to the end of this lecture.

(Refer Slide Time: 57:43)



So, as a matter of review, we have typically seen globe ball, ball and butterfly valves, we have seen various kinds of valve flow characteristics, both static and dynamic. And, we have seen, how valves are actuated and controlled, there is one aspect, which is treated in books, which is called valve sizing, that is for a given application determining, that the size of the valve. We are not talking about that because this essentially a process, design exercise and not does not concern auto machine control.

(Refer Slide Time: 58:18)



So, to end points to ponder, first is sketch the cross section of a globe valve and indicate four of its major parts, you should be able to do that. What is the difference between installed and inherent characteristic, this is extremely important and why this occurs, what is the main advantage of a valve positioned, why the one puts it. And finally, mention one advantage of a solenoid actuator, over a pneumatic actuator and one advantage of a pneumatic over a solenoid actuator, so that brings us to the end of the lecture.

Thank you very much.