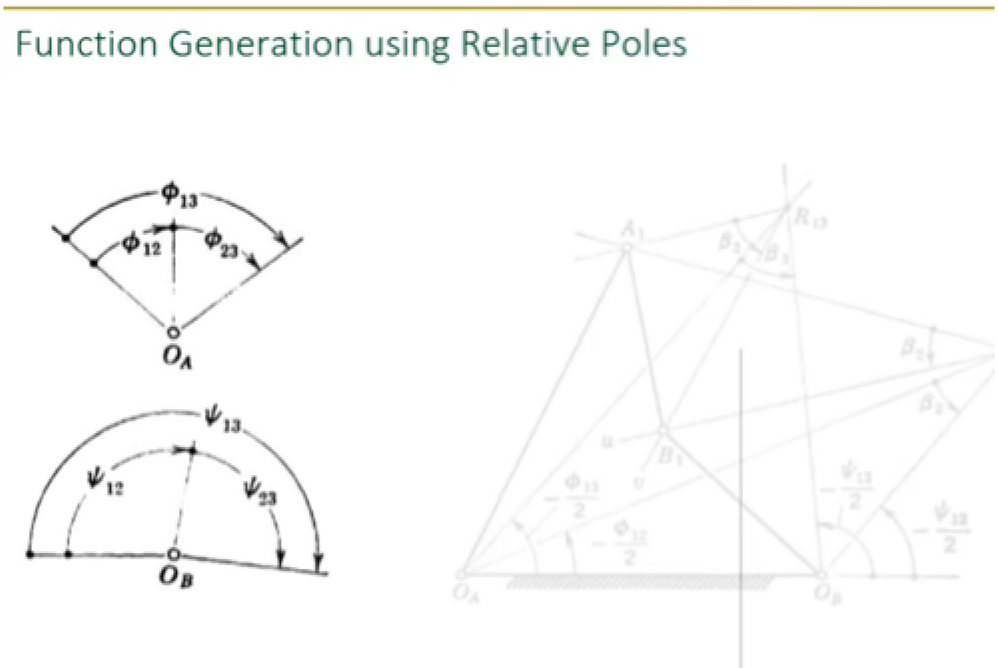
Lecture-11

Theory of mechanisms

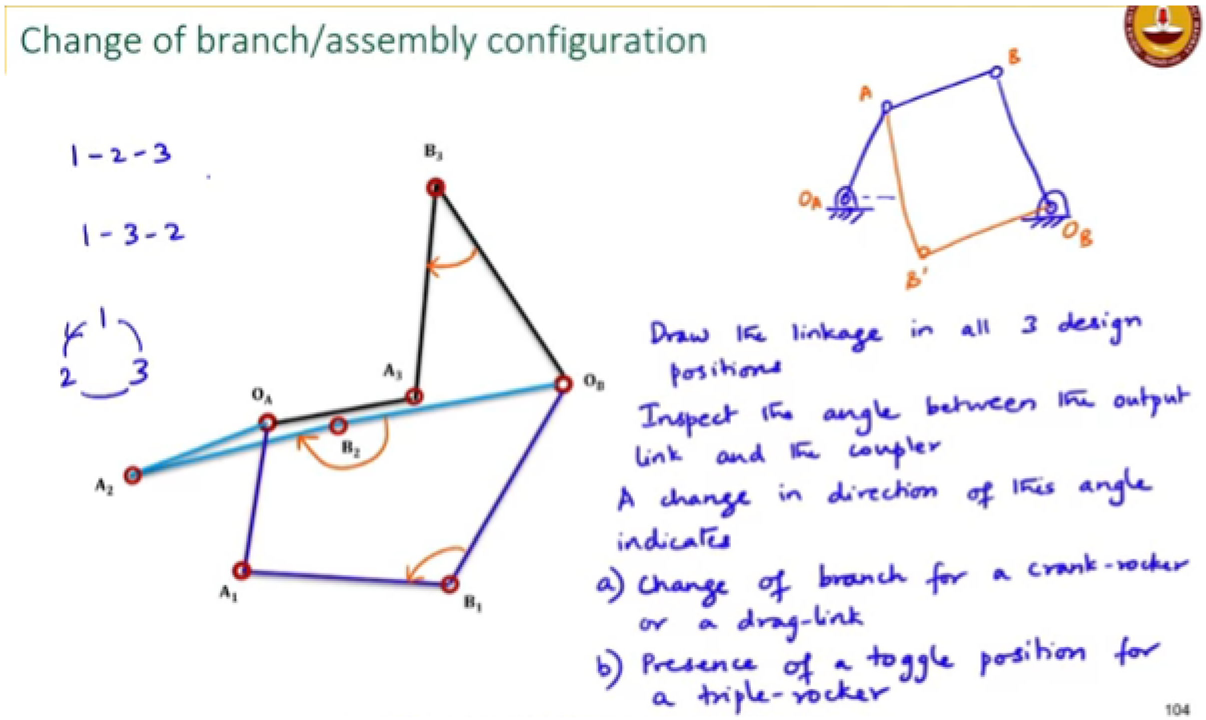
Structural error and Chebyshev spacing

Refer slide time: (0:13)



So, yesterday we looked at function Generation, using relative poles, both the fo position and three position now, generation and that kind of concludes our investigation of the graphical techniques, well come back to some of them when we do couple of curves in more detail because, we did not really focus a lot on path generation but, that's a separate topic, that we will deal with a little bit later, okay. But, graphical synthesis for motion generation, extreme positions, dead center positions, function generation and path generation, for like specific points, these are the things that we have looked at, and also slider-crank synthesis, to some extent for two positions, generation. So, what you will notice? With all these graphical techniques, is that one they are very simple. So, they can give you some quick solutions they are also very intuitive. So, they give you a very good picture of how, how to go about it because, it is all based on geometry, which is very convenient it's easy to understand and easy to apply as well so, in many cases a graphical solution may be suitable for your application, one of the things that we saw, with these methods, is in all of them what we essentially did? Was once, you do the synthesis what you can guarantee? is that the linkage can be assembled in those specific positions, that you designed for it that's the only guarantee you have, you really don't know what's going to happen? In between the positions, and in many cases, it may happen that your design may not be suitable because, of something that happens in between the precision points that you chose, the specific positions that you chose for your synthesis. So, usually you know apart from you know if it is a crank if you want a crank-rocker you do the Grashof check. But, apart from that for other links, linkages that you design it is always a good idea to either make a model, either you know, on the computer or a physical model and see, whether it is able to traverse all those positions smoothly, whether it's able to go from one position to the other.

Refer slide time: (3: 20)

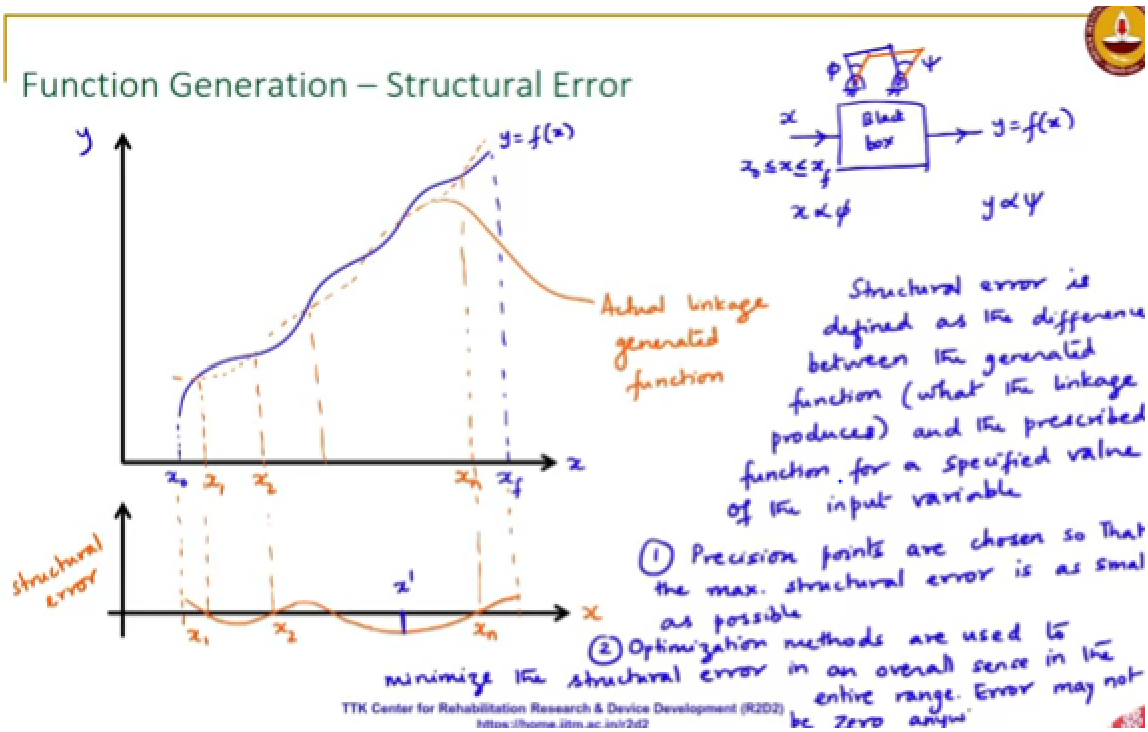


One, of the problems that can occur is what is called? A change of branch or assembly configuration, a four-bar can be assembled in two configurations, right you have if you have four links, okay. For the same input angle, I can assemble it like that or I can assemble it like this, okay. So, if I say this, is OA, AB this will b, A B dash, OB. So, these are the two possibilities and usually, once you assemble it in one configuration it will only move in that configuration, it will not be able to switch, from one configuration to the other, without you physically disassembling, the linkage and reconnecting it in the second configuration. The cases where it might happen is if you have a, a neutral, rash of neutral linkage, not necessarily a parallelogram but, a parallelogram linkage is a blush of neutral linkage. So, if you have the condition s plus L equal to P plus Q. then when it goes through a change point, it might actually change configurations, okay. But, in general, you will not be able to go from one configuration to the other, without physically disassembling the linkage. So, one of the problems that you are likely to encounter is, you may synthesize a linkage, okay. For say, three positions motion generation, okay .and then you find that you make the model you try to move it, it said hitting only two positions. But, you know that the third one also should be on that because it is lie in you know it is they, the moving pivots lie on circles right, what may happen? Is that two of those positions, may be on one configuration of the linkage, and the third position, and may be on another configuration.

So, from the synthesis, you could only guarantee that it will be assembled in those three positions but, whether it will be able to move .so, in this case it will not be able to move to one of the positions because, it lies on the other branch ,or other assembly configuration, of the linkage. So, which is a problem for you? so ,one way is to actually make a model and try to find whether it moves to all three positions, you can also look at so, the other thing you could do which is kind of easier to do when you ,are doing graphical synthesis, is to basically assemble the linkage in the three positions. so ,say I have designed, for motion generation a1, b1, a2, b2, a3, b3, okay. I wanted these three positions, okay .and I find easy enough a 1, a 2, a 3 should lie on a circle. So, I Find OA similarly b1, b2, b3 lie on a circle and I find OB, which is the center of that circle, when I do this I can assemble the linkage, you can assemble the linkage in those three so, you can basically draw the linkage in those three positions and then look at the angle between, the output link and the coupler. So, if you look at this angle. Okay. So, you look at the angle between OB, b and a in the three positions. So, here also I look at it, the angle is in this direction, okay. Now, in this position, if I draw that angle OB, b 2a, it has changed direction. So, that gives you an indication, that it's not going to be able to reach all three specified positions, in the same branch of the linkage, you would have so, that means it's not a suitable solution for your application. so ,with graphical thing you will just have to go back and try something else, maybe change one of the moving pivots, okay. so ,that would be odd yeah so, change one of the moving pivots so it would be that's one of the disadvantages, of the graphical process because, if you have to if you keep hitting the wrong solution, then it's going to be quite time-consuming, if you get a suitable solution quickly, you are lucky but, if not it can become, quite tedious to look for alternative solutions, okay.

So, the way to check is to draw the linkage, what we did? Here was draw the linkage in all three design positions, inspect the angle between, a changes in direction. So, if it is a crank-rocker or a drag Link, it means there's a change of branch happening, if it is a grash of to if it's a triple rocker, then it indicates that there is a drive failure, that is there is a toggle position, it means there's probably a toggle position between, the two for a triple rocker, which is a non-Grashof. So, this is a quick way you can determine whether your solution is going to perform as indicated .again, the other problem could be how it reaches the positions, 1, 2 & 3 you did not said anything about, taking that into consideration while doing the graphical analysis. so it may go 1, 3 ,2 if it's a crank-rocker you can always start from a different position change the direction because, it can rotate. So, if I have if I'm supposed to reach 1, 2, 3 and instead it is doing 2, 3 or let's say ,1, 3, 2 ,okay. So, so all I have to do is, drive it and if it's a crank-rocker, I just need to drive it in the opposite direction, drive the crank in the opposite direction, if it is if it’s a crank-rocker right .so, in the crank rocket I can always start at some other position and still manage to get the 1, 2, 3 But, that's not the case with the linkage that's not a crank-rocker, ok. I won't have that flexibility to do that, in which case I may again have to go back and redesign, ok. So, these are some of the problems that we encounter because, of the fact that our synthesis is wholly based on specific positions, and not of the behavior in between those positions.

Refer slide time: (13:16)



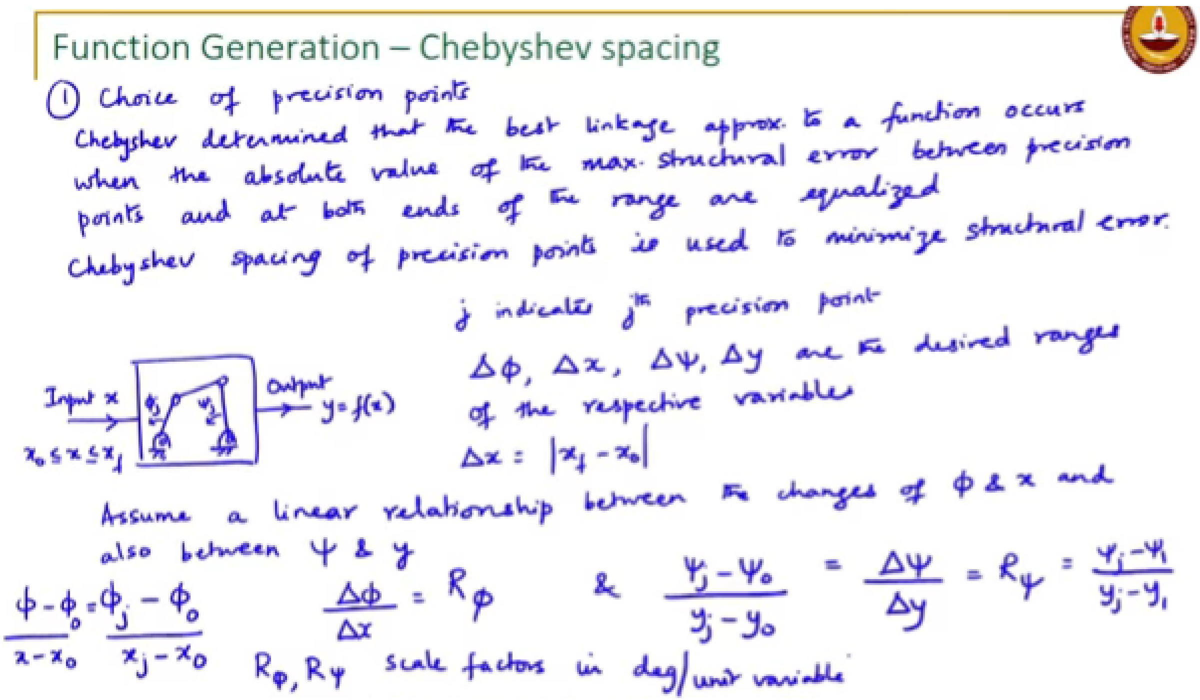
You, will actually find that even with the analytical methods, in many cases we will encounter, the same thing analytical methods, once you code it you have more flexibility to cycle through a number of solutions, as opposed to the graphical methods, okay .so, one of the first things we will look at is so, we saw especially for function generation right, we said say, I take the example of the valve opening. You, know you may want the valve to open in a specific relationship with respect to the input, okay. the input, may be uniform but, depending on the application you know I am, when I am at the later end of the scale, I am a back want ,the valve to open in a certain way in an earlier end of the scale I may want it .so, there may be a specific relationship. So, I may have all I am telling you are, okay. I have a black box, I have containing a linkage, I have a specific function, X and I want a specific output, and that may happen with this, kind of a linkage that's inside this box ,okay. This is what is happening? Phi and shi. Okay. So, I have a black box which contains a function generator, and I may want a specific relationship between, the input and the output, in a certain range, for the input variable, okay. If X varies from X naught to XF, in that particular range I want the output to bury in a certain manner. Now, if I use a linkage like this, okay. I may be, able to achieve that an exact Match.

So, so X would be say proportional to Phi, okay. The input angle Phi, and Y is proportional to Shi , okay. I can set it up so, that it now I can get the exact relationship between, x and y perhaps only at, certain specific points of the linkage movement, those will be the precision points where, exactly it matches what I need? Okay. in between, I may have error. So, if I have say, this is X, this is y, and say ,I am looking for a function ,that does this so, I have y equal to f of X ,which is the desired behavior, I am looking for, okay. And this is the desired behavior I am looking for say, from X naught to some XF, okay. Then my actual function generator, inside may actually give me something like, this. it may match exactly with what I need, at some n, precision points ,okay .so, there is no error ,at those precision points but everywhere else in between, I would have some error. So, if I look at my this, is called structural error. So, it it is, basically based on that's the characteristic of the linkage, that I am using, it's not something that I can rectify by say maybe you know changing the tolerances of manufacturing, okay. this is how, the linkage behaves, the relationship between, shi and Phi, is not going to match exactly the relationship between, y and X. and so, this is called the structural error, of the linkage, and it may be something like. So, here my actual. so, if I look at these points what will by error be, my error will be 0, at those points but everywhere else, I would have there would be a difference between, what my actual function? is generating versus, something like this. Okay? So, these at the precision points, I have zero error but, at the ends or in Between, I do have a non zero error, Okay. So, this orange one is the actual linkage generated function. So, structural error **is** defined, as the difference between, the generated function, which is what? The linkage actually produces, and the prescribed but specified function, for a specified value of the input variable. So, if I look at structural error at say, some X equal to you know some x value, okay? Some X dash, then this, would be my structural error, at that particular point. I would compute what the linkage generates, what I want the actual value of the function to be the difference between, them is the structural error, which I really can't do anything about unless I change the linkage itself, okay. Its it is not something, that can be rectified by more precise manufacturing or something like that. Okay? So, the desired characteristics, are satisfied only at specific points in their range, that you are looking at the range is X naught to X F. so, so how do we choose these typically you want to choose the precision points they, may be if you are looking at a range for the function, if you are only looking at so for instance in motion generation, if you're only looking at specific positions ,then those will be your precision points, okay .those will be the configurations you choose, in function generation this, this is more typical of function generation, where you are looking at the continuous behavior over, a range or in path generation, where you would have to look at the continuous behavior there over a specified range, there you want to pick those precision points, in such a way that you are minimizing the error ,over the entire range, okay.

So, you may want the precision points may be a design criterion because you want to hit certain specific spots, in which case you don't really care what happens? in between, those positions as long as the linkage is able to move, fun from one precision point to another that's all you are concerned about but, if you are looking at the behavior over a continuous range, then you're concerned about you, know how the error is going to be in between the points where the error is zero. So, there are different ways to approach it, one is you choose the precision points. So, that the maximum structural error is as small as possible, and the second approach is, you use optimization, the structural, structural error in an overall sense ,in the entire range in this kind of an approach ,in the second approach, the error may not be zero anywhere but ,the overall error will be minimized, you may get as close to the actual desired function, as possible but the error may not be zero .anywhere, in the precision point approach ,you have zero error at specified points. But ,the error at other places may be high, okay. So, error may not be zero.

In the case of path generation, your structural error may be defined differently, function generation it is easy, you have one algebraic quantity, the angles so you can just do what they did now, in the case of path generation, you may actually have to define it as a vector between, the desired path you know. So, and the displacement vector between, where you actually reach versus there you want it to be, okay .so, it may be that kind of vector and if it is path generation with prescribed input timing, meaning at a specific point of the input I want to be at a certain path ,versus overall path you look at the normal distance, how far you are so, it would differ based on so, those the error computation will be more complex, than in the case of function generation where you can easily see that there's one quantity, there is an input and output quantity, algebraic quantities, that can be related motion generation ,you may have because you're looking at both path and orientation there ,in the case of motion generation, you have the path of a point plus the orientation of the rigid body. So, you may have to look at multiple structural error curves to analyze the error in those cases.

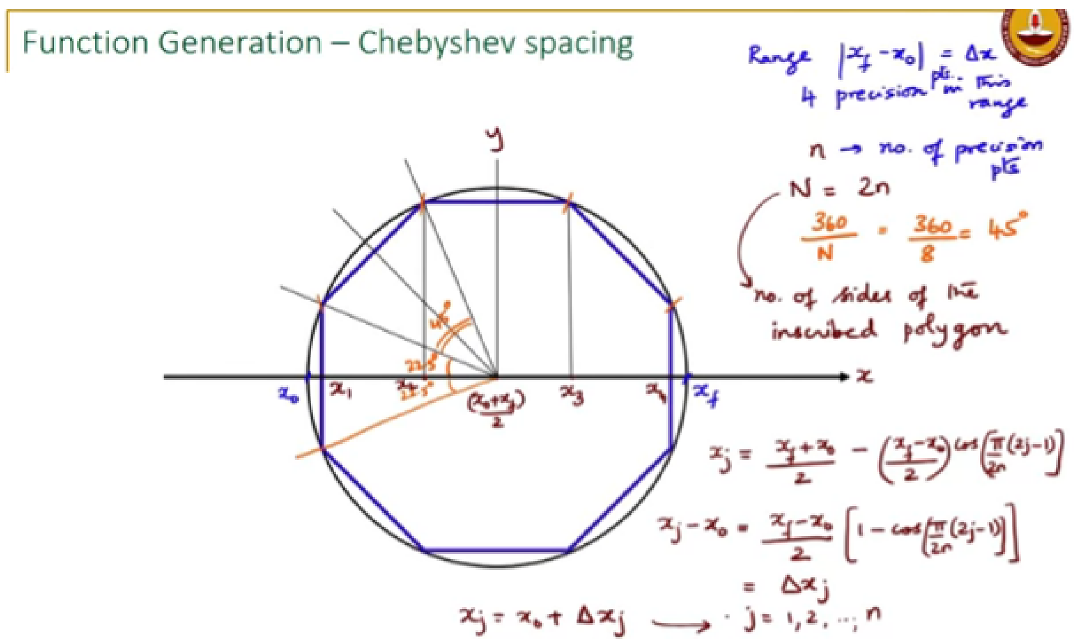
Refer slide time :(27:51)



So, let's look at this first approach, which is the choice of precision points. So, this was proposed by Chebyshev, who said that? The best linkage approximation, to a function occurs when the absolute value, of the maximum structural error between, precision points, and at both ends of the range, are equalized So, this Chebyshev spacing, is used to minimize the structural error. so ,you choose the precision points in a specific way, to minimize the structural error and then of course you know, how to do the synthesis for a specified number of transition points, and this is actually like a first step approach, after that you will start taking smaller intervals and try to minimize the error. But, we will only look at this over the range that we are interested in we will look at, the spacing the precision point spacing. So, it's actually a nitrate of process that may have to be performed in order to minimize the error. So, you see here, that the first present point is not at the beginning of the range, and the last precision point is also not at the end of the range, okay. Because, according to Chabot Chebyshev spacing, the errors there is error at the ends of the range, which is equalized with the maximum structural error. Between, the between the points, okay. So, and for function generation, we usually measure the angles from the fur so, you can always take your Phi naught equal to zero, without any loss of generality because, in function generation you are only considered concerned with angular displacements, the once you design the linkage that relationship is maintained, regardless of whether you scale up or down the linkage or reorient the linkage, okay .so, they say, if you take I have my input X, in this range, and I have my output y equal to f of X. so, J indicates J th, precision point, okay .so, if I look at Delta, Delta Phi, Delta X, Delta shi, and Delta Y ,are the desired ranges, of the different variables. So, if I look at Delta X is, mod of XF minus X naught, typically what we will do? Is we will assume a linear relationship between X and Phi, similarly between y and shi. so, you have Phi J minus, any phi actually any phi, in the range, by X minus X naught will be equal to x naught, and you define Delta Phi, by Delta X as the scale factor, our Phi and shi J minus, shi naught, by

YJ minus y naught, is just a Phi by Delta Y. this, is defined as the scale factor our shi. Basically, you can shi J minus, shi I 1 by, Y J minus y1, to determine the value, or Y naught or shi naught. It’s a linear relationship. so ,any two points that you take you have the so, are Phi and are shi as are called the scale factors, in degree per unit variable. So, there is a geometrical method for the determination of the Chebyshev spacing.

Refer slide time: (37:02)



So, let's say I want I have a range XF minus X naught. Okay? And say, I want to determine 4 precision points, in this range. So, if I determine for precision points I will get three angular displacements for this, okay .okay ?so, the first thing we do is, we draw a circle, with diameter equal to the range, proportional to the range, okay .not necessarily equal it should be proportional to the range, then if I want 4 precision points, okay .say ,n is number of position points, then I have to inscribe an equilateral polygon, a regular polygon, equal to 2 n sites ,ok .so ,this is an octagon, I want 4 precision points. So, I draw an 8 sided polygon, regular polygons, such that two of its sides are vertical, okay. Here, I have to inscribe it in such a way that two of its sides are vertical .so, if I look at how would I do that? I know what is the? angle subtended by any of these. So, if I have 360 by n in this case would give me 45 degrees, okay. so ,I would this would be half of that so, I would draw and this angle would be 45, okay. So, I can determine what would be the side of the polygon by constructing this, where draw this, this will also be twenty two point five degrees, okay .so, I know the side of the polygon .so, I can construct typically, you only need to construct one half of the polygon, you don't need to construct the full polygon, okay. So, I can just construct, I know the length I just construct it like that, okay. Then, my precision points are these so, from the vertices of the polygon I drop, perpendiculars to the horizontal .so, I have this would be X 1, this would be X 2, this is X 3, and this is X 4 .so, I can read off I can measure from this construction, the values of X which would form my precision points. and then using a certain scale factor, I would translate it into values of phi because, my X is proportional to Phi ,ok .so, if I look at this ,this is obviously X naught ,plus X F by 2 ,this point .so, I can write in analytical form, I can write XJ as X F plus, X naught by 2 minus XF minus X naught by 2 cos of Phi by 2 n into 2 J minus 1. if you want you can remember this, or the graphical construction is easier, or you can remember it as XJ minus X naught equal to XF minus X naught by 2 into 1 minus cos Phi by 2 n 2j minus 1. so ,this is basically Delta XJ, the distance of a precision point from X naught, ok. Delta XJ. So, XJ equals X naught from the starting point, this distance X 1. So, would take the values J will be 1, 2, 2, n the precision points. So, I can write xj as X naught plus Delta XJ. I should if I say, n is the precision points ,let me use capital N as the number of for the polygon that is equal to 2 n. so, this is the number of sides of the inscribed polygon, typically if you are doing for three position points, because that's what you have learned synthesis? For, so far so, you would have a hexagon, okay. You would have a hexagon, and you can find the three precision points, of the using the hexagons, okay .and the central point will also be a precision point in that case, case of the exhibit. So, we will do an example in the next class, for function generation, using this spacing.