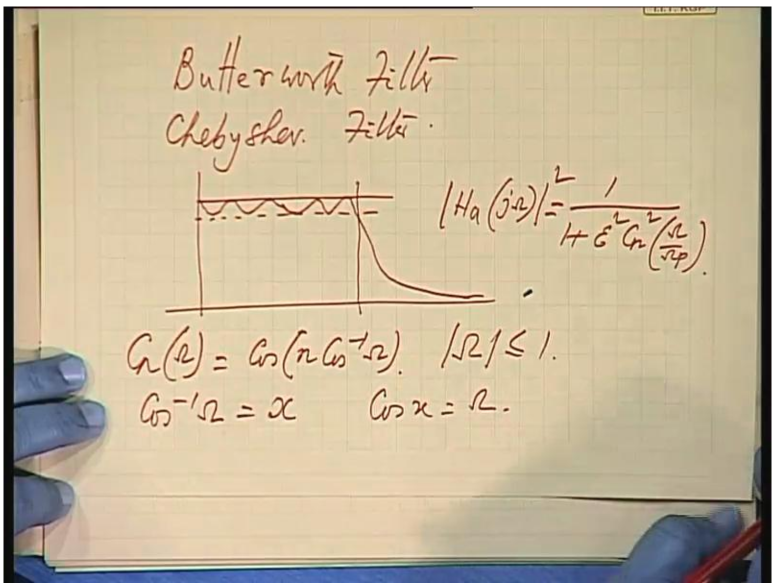
Digital Signal Processing
Prof. T.K. Basu
Department of Electrical Engineering
Indian Institute of Technology, Kharagpur

Lecture - 18
IIR Filters (Contd.)

We shall be continuing with the design of analogue filter and then will implement an IIR filter from an analogue design.

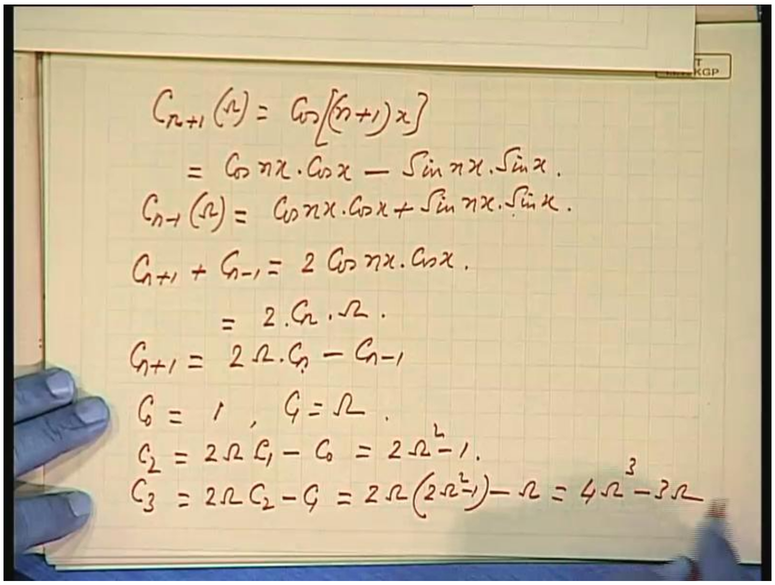
(Refer Slide Time: 00:53)



Last time we are discussing about Butterworth filter; today we shall be taking up Chebyshev filter, of the first kind. Now, Chebyshev filter functions are like this; that is in the pass band, you have a definite width of that we prove like this and then in the stop band it should go like this, okay. The function is given by the magnitude square equal to 1 by 1 plus epsilon square C n square omega by omega p, where C n is a Chebyshev function of nth order. We will just very briefly discuss about this function and then see how this makes our requirement for the filter.

Now, C n is defined as C n omega is cosine of n cos inverse omega, okay; omega less than equal to 1, all right. So, how much is cos inverse omega? Let us assume this as x, then cos of x is equal to omega, okay. Let us assume cos inverse omega is x, so cos of x is equal to omega. So, cos of n cosine inverse omega.

(Refer Slide Time: 02:49)



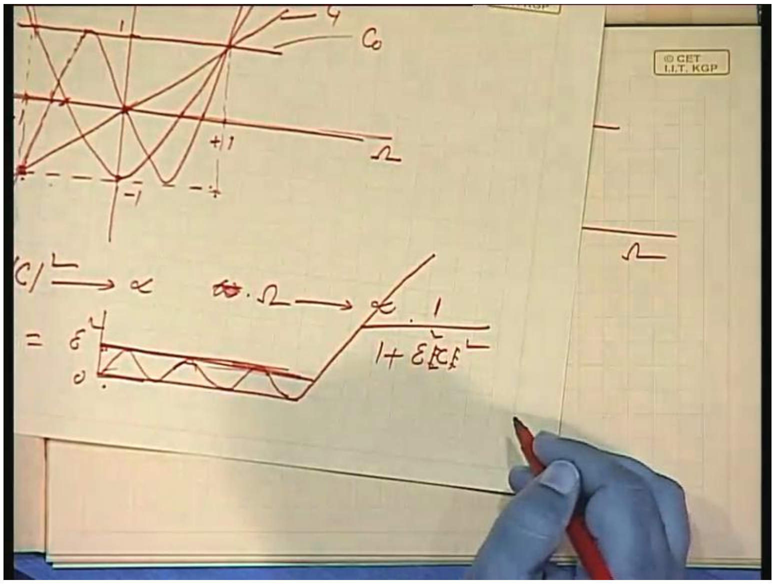
If I take henceforth, I will be writing just C n, C n plus 1 and so on; they are all functions of omega. So, what will be C n plus 1? C n plus 1 is cos of n plus 1 into x okay, which is cos n x, cos x minus sin n x, sin x. C n minus 1 x, C n minus 1 omega would be cos n x cos x plus sin n x sin x. If I add them together, then C n plus 1 plus C n minus 1; that will be twice cos of n x cos of x, do you agree?

So twice C n, cos n x is C n into cos of x omega, okay. So, C n plus 1 becomes twice omega C n minus C n minus 1, okay. So, this is a recursive relationship that you have got. In terms of C n and its previous value, we can generate a new polynomial C n plus 1; this is a polynomial in omega. Let us see, what will be C 0? C 0 is from definition; C 0 is cos of 0 into something, so cos of 0 is 1, so C 0 is 1.

C 1 is cos of 1 into cos of cos inverse of omega. So, omega itself okay. So, using these two I can generate C 2. So, C 2 will be twice omega into C 1 minus C 0 which is twice omega square minus 1, is that all right. Can you generate C 3 from there? 2 omega into C 2 minus C 1 which means 2 omega into 2 omega square minus 1 minus C 1, that is omega, okay.

So, 4 omega cube, correct me if I am wrong, minus three omega okay. You can generate a table of such polynomials by this recursive relationship, okay. So, I would request all of you try to do that for three, four more steps, try at home. Now, let us come back to some of the important features of these Chebyshev polynomials which will be very useful for us.

(Refer Slide Time: 06:14)



C 0 suppose, this is omega and this is C magnitude, okay all right. I will just write C n, what is C 0? 1. So, let this be 1. So, this is C 0 up to plus 1 and minus 1; we are considering up to plus 1 and minus 1 because cos x can be at the most plus 1 at the least minus 1 since the limit of omega is plus minus 1. So, we will study the behaviour of this function in this range for the time being.

Actually, when omega is greater than 1 it is given by cos, cos n cos inverse omega all right; hyperbolic functions and that will also generate, the same polynomial relationship okay. With a cos function, Hyperbolic functions also we will get the same recursive relationship, all right. So, it is immaterial whether we are restricting within plus minus 1 or beyond. We will see its behaviour within this range first.

So this is C 0, C 1 is omega; so it is a straight line like this, this is minus 1 okay. Then C 2 is 2 omega square minus 1. So, when omega equal to zero, it is minus 1 and when omega is equal to minus 1 or plus 1, it is 1. So, it is starts from here. It is a parabola going like this. So, this is C 1, this is C 2 okay. C three again 4 omega cube minus 3 omega when omega is minus 1, this is minus 1, it starts from here.

Then when omega equal to 0, it is 0 so it passes through this; somewhere in between it goes to positive plus 1. You can find out, at what value of omega it goes to plus 1, this is C 3 and so on. So, one thing you observe for all these functions, you can go on taking higher order polynomials. You will find the Chebyshev polynomial has this important characteristic within the range, minus 1 to plus 1; for x or for omega, the function C alternates between minus 1 and plus 1 all right, between minus 1 and plus 1.

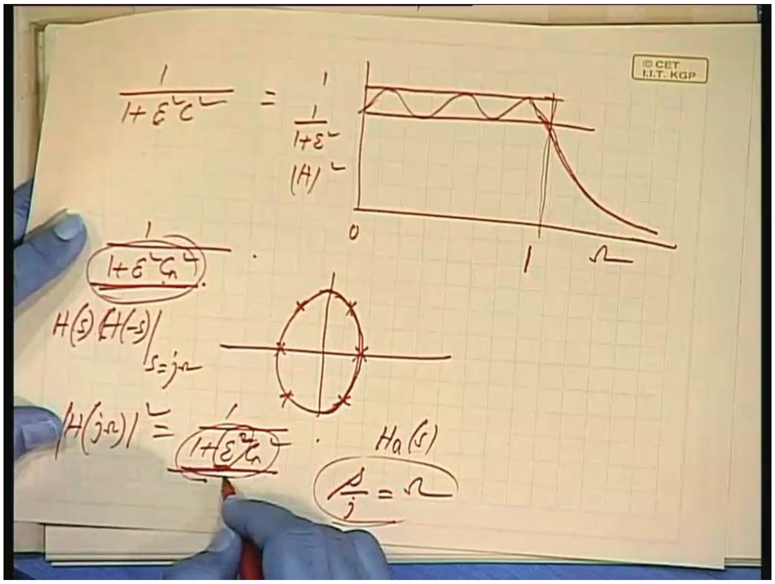
And then after that the magnitude goes up like this. The higher the order, faster will be the growth in magnitude; either it goes to the negative side or to the positive side. That means if I take C square, the magnitude will be tending to infinity, as omega as omega tends to infinity, okay. Now, let us see a function which is epsilon square, C square. Epsilon square C square will be, this range is scaled down, this range is scaled down between plus epsilon square and minus epsilon square when you take C square, is it not?

And since it is magnitude square, so it will be between 0 and 1; the least value it can have is 0, so square of this function multiplied by epsilon square will be alternating between 0 and epsilon square, do you all agree? Epsilon square, C square whether it is C 1, C 2, C 3, they will be restricted within this.

They can be, C 0 is the constant so it will be only this much. C 1, it may vary just once, the number of times it varies will be depending on the order; number of times it is alternating between minus 1 and plus 1, all right. See, if it is C 3 how many times it is crossing over? Say, once here, once here, once here, three times it is crossing this 0 level. If it is C 2, it is crossing once, twice, two times.

So, number of cross over will be depending on the order, okay. So, what will be 1 plus epsilon square magnitude C square? Once you square, you need not put the magnitude okay; one over this, what will it look like?

(Refer Slide Time: 12:05)



1 by 1 plus epsilon square, C square, what will it be like? Its minimum value is 0; maximum value is 1, plus epsilon, maximum value is epsilon square. So, it will be alternating between 1 and 1 by 1 plus epsilon square is it not; depending on the order, it will be alternating. This is 1 and after that the magnitude goes on increasing; either on this side or on that side, if you square it is will be always positive.

So, epsilon square, C square later on whatever be the order, will be going up like this, beyond 1. When omega is greater than 1, it will be shooting up; so there it will be falling gradually, this is a monotonic fall. There is no ripple here okay; ripples are only between 0 and 1. So, here also the ripples will be within 0 and 1.

So, this will be the nature of variation of this function, is that all right? So, we have got a Chebyshev polynomial like this. Now, what would be the roots of this; 1 by 1 plus epsilon square, C n square? This is given, you find out what will be the roots. Once you know the roots, you can use bi-linear transformation, substitute, okay. Today will also discuss another method; impulse invariance method.

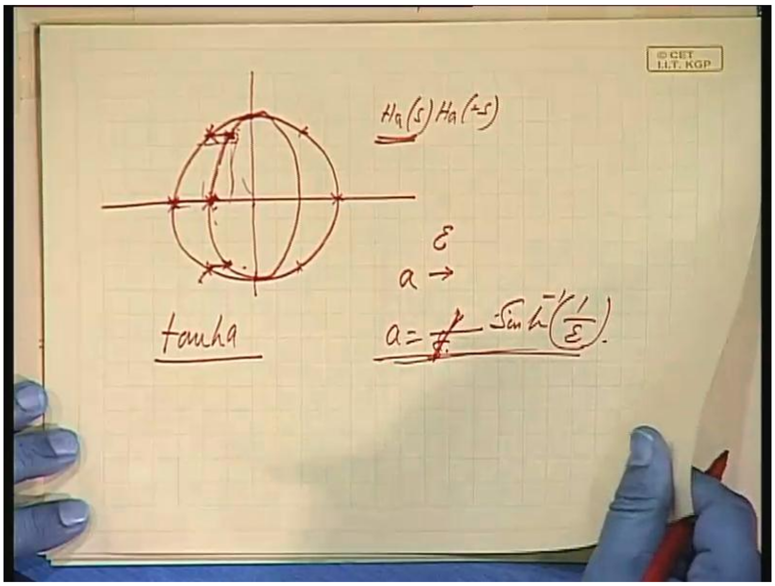
So, this design is very simple. Once you know the routes of this particular function; that is this is equal to, you write H s H minus s, s equal to j omega okay, is it not? H or G whatever you write, H j omega magnitude square is 1 by 1 plus epsilon square, C n square, okay. And we find, this is having this is H square; this function is having a characteristic like this. So, what will be the corresponding analogue filter function?

In case of Butterworth filter, what did you do? You substitute s by j is equal to omega, if you remember. And then you find out all the roots; they are all coming on the unit circle and then you take only left of plane roots again from the polynomial, that gives you H s. Similarly here, find out the roots of this. Take only the left of plane roots, there will be mirror images; right of plane roots and left of plane roots, they will come all in a symmetric position, there will be mirror images.

After this substitution, you calculate the roots. In this case, it will be an ellipse all right. So, they will come on ellipse. The roots may come like this. In case of Butterworth filter, there on a circle, okay. So there now, that shape of the ellipse will be depending on the factor epsilon, alike a Butterworth filter; in case of Chebyshev filter, you have to specify epsilon and depending on the value of epsilon the shape of these will also keep on changing, so the roots are not in unique position.

Third one, third order Butterworth filter will have unique position of the roots whereas third order Chebyshev filter; will have roots depending on the position of epsilon, value of epsilon, okay. So, every time you have to compute separately the roots. Now, that is quite labour some but anyway there are short cuts. Standard design books have given methods of computing it. The one of the simplest method is to find out, the roots of the corresponding Butterworth filter, okay.

(Refer Slide Time: 16:45)



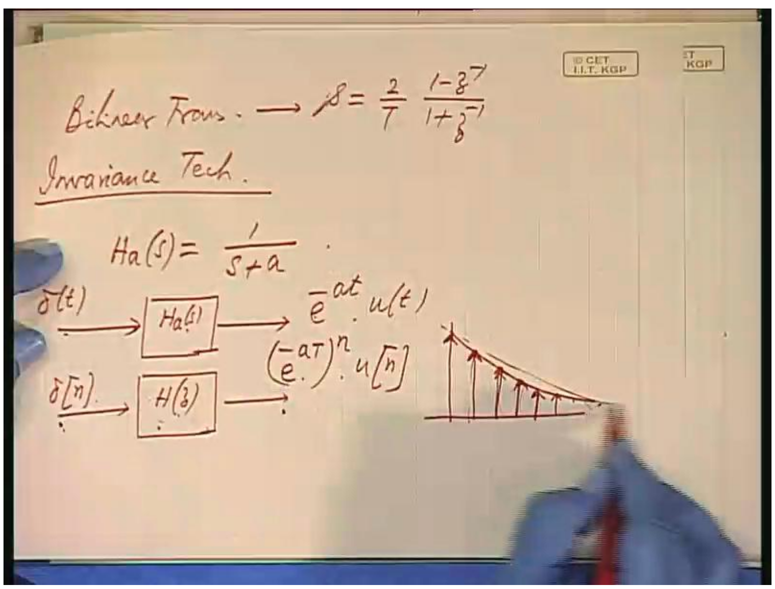
Sorry, suppose it is a third order Chebyshev filter, that we have got, all right. What will be the position of corresponding Butterworth filter? Then from the Butterworth filter; it is a 120 degrees, 128 degrees, 240 degrees, is it not? Because, there are six roots for third order Butterworth filters all right, corresponding to H a s into H a minus s.

You forget about the right half plane roots that corresponds to H a minus s. So, H a s is obtained by these three routes. There is a relationship, the ellipse that you get for the Chebyshev filter will be like this and the real parts shift by a factor and the imaginary parts remain intact, all right. You just squeeze the real parts then you know the locations of this okay, then it is very simple.

So given epsilon, you can calculate a factor a, which you can evaluate from epsilon; will not discuss in details the methods of computing these roots, they are very standard. So, it is basically tan hyperbolic a, which is a factor by which it should be squeezed and a, is something like 1 by sin sorry, sin hyperbolic 1 by epsilon something like this; there is a relationship while designing the filter will get into the details of this.

And if you take the real part, squeeze by this factor you get the real part of the Chebyshev filter. Imaginary parts you already know, so imaginary parts remain same. So, you can frame the polynomial and you can find out the corresponding Chebyshev filter, okay. Now, the method by which we are trying to go from analogue filter to a digital filter was, which was discussed last time; was bilinear transformation or in general Adams-Moulton transform. You can go for higher order transforms.

(Refer Slide Time: 19:34)



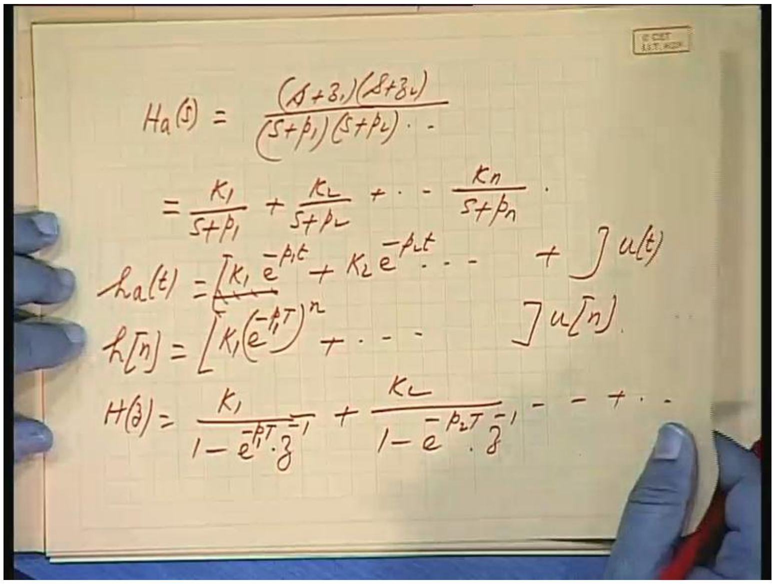
So, bilinear transform which gave us s equal to 2 by T 1 minus z inverse by 1 plus z inverse, okay. So, wherever there is an s, you replace by this; whether it is a Butterworth design or Chebyshev design, just a direct substitution of s with this. Now, let us discuss another simple method that is invariance technique. Now, invariance technique is a totally different approach.

You have the analogue filter function; say I will take a simple function s plus a, suppose it is first order filter, 1 by s plus a. If this is the filter, what is its impulse response? If I give a delta t input, it will be Laplace inverse of this, e to the power minus a t into u t that will be the output. So, we now take it in the discrete domain.

Given an impulse discrete function delta n, I should get a response, I should get a response which will be a discrete version of this that means; the analogue response is like this, e to the power minus a t u t. I should get with this model, a discrete version of this. What is this? This was H a s. What is that function which will give me this output? That means I want the output should be, e to the power minus a T to the power n u n, all right? This is e to the power minus a t into n.

So, this is known as impulse invariance technique that means; corresponding to an impulse input in the continuous domain, if the output is known then what would be the corresponding discretized output, corresponding to an impulse input? Evaluate that, and that is the filter which I am going to substitute for this, all right. So, there is impulse response in both the cases are matching. One is giving you only the discrete version of the continuous output. Now let us see, what should be this like.

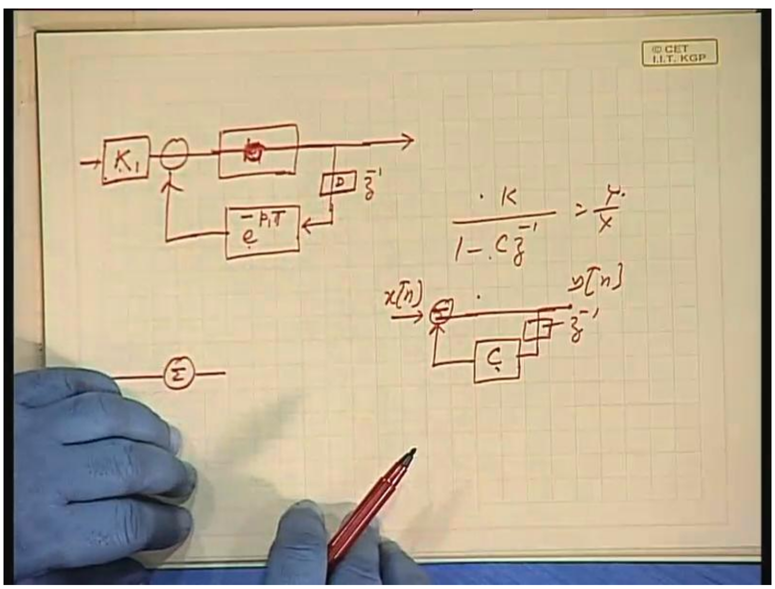
(Refer Slide Time: 22:38)



Suppose, I have an H a s equal to, these factors are 1 by s plus z 1 into s plus z 2; these are the zeros and these are the poles and so on. I can always write this as, some K 1 by s plus p 1 plus K 2 by s plus p 2 and so on; depending on the number of poles I have say, K n by s plus p n, okay. Then what will be the corresponding impulse response? It will be K 1 by sorry, e to the power minus p 1 t plus K 2, e to the power minus p 2 t and so on into u t, okay.

So, my h n will be the discrete version of these which will be K 1 into e to the power minus p 1 into capital T whole to the power n and so on into u n. So, what will be the corresponding H z? It will be K 1 c to the power n is, K 1 by 1 minus e to the power minus p T into z inverse, is it not? Where e to the power minus p T, p 1 T sorry, p 1 T; similarly K 2 by 1 minus e to the power minus p 2 T z inverse and so on, that will be the z transform. So, let us realise it in the hardware domain, this will be represented by simple first order IIR filter of this kind.

(Refer Slide Time: 24:44)

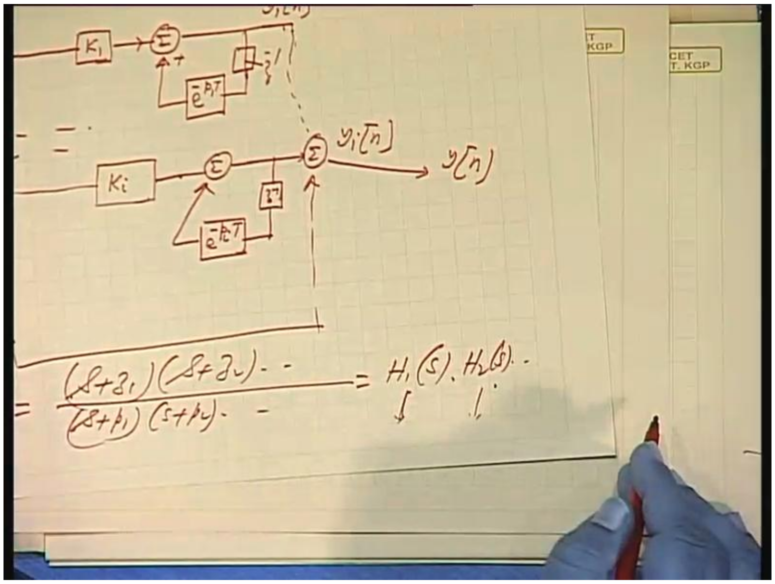


Suppose I have what should I have? Some gain K 1, okay. And then a feedback path e to the power minus P 1 T okay, and then I can put the delay here or I can put the delay here; as you please a delay which is corresponding to z to the power minus 1. So, this gets added with the, is this all right, is this all right? No, where should I put the K 1? K 1 is the multiplier okay, should it be this way? Put it as it is, is that all right?

Let us take 1 by, if it is 1 by 1 minus C z inverse, all right what is the corresponding structure? C, I can put a z inverse here or here, so whatever is the output y added okay. This is x and this is y all right. So, this gets added because this is Y by X. So, Y minus C z inverse, if it is brought to this side; so that is being added. So, this is the elementary structure.

Now, it is a C 1, this constant which is changing e to the power minus P 1 T, e to the power minus P T 2 t and so on. And this constant K, I can put here separately. So, it is like this or you can put it at the end also as you please. So, there are such blocks and there to be added. So, what will be the next structure? Such blocks will come in parallel, is it not?

(Refer Slide Time: 27:29)



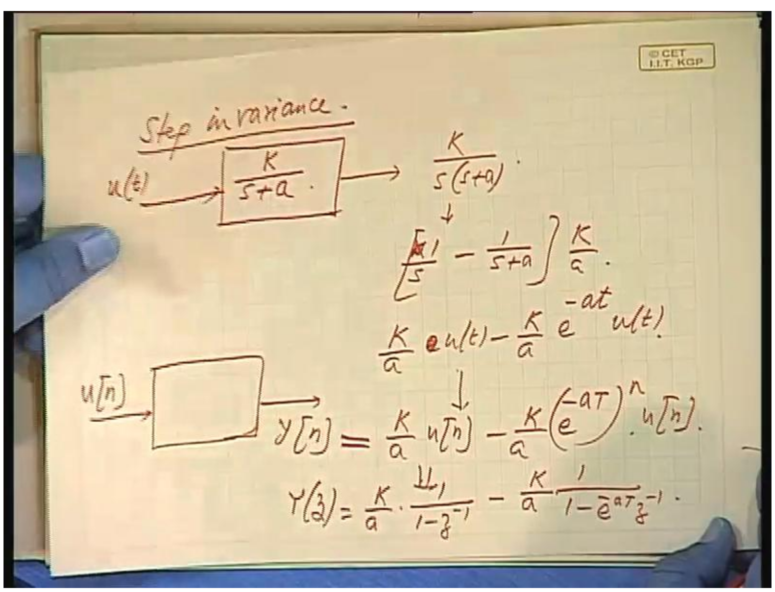
So it will be X n, we can have K 1 adder e to the power minus p 1 T, this will be say Y 1. Similarly, K I, I call it a general term; this is Y i and they are all added. There should be a delay here that is good, z to the power minus 1, e to the power minus p I T and like this. So, all all of them will be finally added here to give you the output Y n, is that okay?

I could have had also the factors in this mode, H z could have been obtained; say H a s, okay. Let us see whether it is possible. H a s is s plus z 1 into s plus z 2, one may suggest this s plus p 1 into s plus p 2 and so on. I can write as product of the first order factors s plus p 1, s plus p 2 in the denominators H 1 s, H 2 s and so on.

Can I write or realize H 1 s and H 2 s in cascade? Can I put them? Have you understood the question? Now, because each H 1 s, each first order factor I can realise by an equivalent digital element like this, like this; this is corresponding to a first order factor K 1 by s plus p 1 t, okay s plus p 1, okay. So, can I now put such blocks in the product form instead of summation, instead of summation like this? Suppose, I write in the form of products, is it possible to realise this? What you suggest?

Our condition was our condition was; we are replacing it assuming that we are giving an impulse input and that gives me an identical output in the discrete domain. Now, if you put them in cascade, what will be the output of the first set? It will not be an impulse anymore. So, you are not giving an impulse input and that condition is not satisfied. So, in the cascade form it will not be allowed, is that all right?

(Refer Slide Time: 31:31)

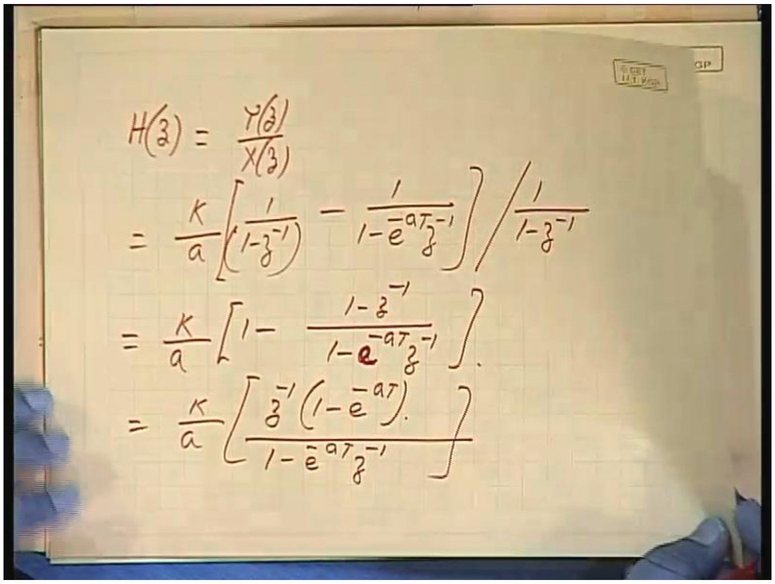


Now one may suggest, another invariance another invariance that is step invariance. You can have any of the standard signals, keep that as an invariance and then design a filter. Let us take once again a very simple factor, K by s plus a. Now, suppose I design it for an impulse instead of impulse for a step input, in the continuous domain if I give a step input, what is the corresponding output? K by s into s plus a, do you all agree?

So, what will be its time domain output? K by s, okay 1 by s minus 1 by s plus a into K by a. So, the output is K by a e to the sorry, u t minus K by a e to the power minus a t into u t, okay. So, what will be the discrete domain output? I want corresponding to step input u n, I should get an output y n which will be a discrete version of this, is that all right? So once I know, I will take z transform of this, z transform of this is one one by 1 minus z inverse, so divide the output by input that gives me the filter.

So let us see, what will be the discrete version of this. K by a u n minus K by a e to the power minus a T n u n, okay. So, what will be the corresponding z transform? That is y z, this is y n, is not? So, y z is equal K by a 1 by 1 minus z inverse; this is the z transform of step function u n, minus K by a e to the power minus a T into n, it is 1 minus e to the power minus a T z inverse, do you all agree?

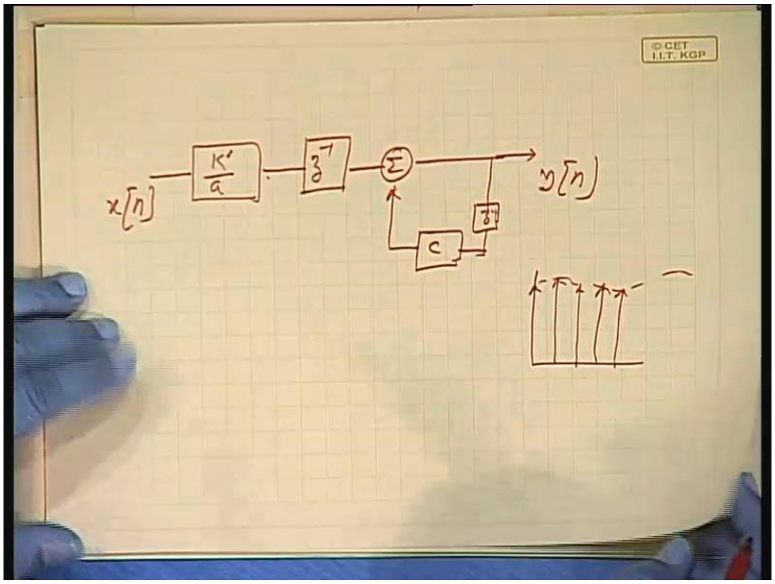
(Refer Slide Time: 34:02)



Therefore, H z corresponding H z is Y z by X z which is K by a 1 by 1 minus z inverse; K by a I can take out, minus 1 by 1 minus e to the power minus a T z inverse divided by 1 by 1 minus z inverse that is a x, is it not? X z is corresponding to u n. So, I have divided by X z, so that gives me the corresponding transfer function.

So, K by a this by this is 1, minus 1 by 1 minus z inverse by 1 minus e to the power minus a T z inverse, is it all right? K by a, so 1 will get cancelled minus e to the power minus; so z inverse into 1 minus e to the power minus a T, check if this is all right. So, this is the step invariance transfer function. So, once again we can take this in a simple block.

(Refer Slide Time: 35:41)



K by a then actually, I could have put K by a into 1 minus e to the power; let me call this whole thing as K dashed by a z inverse by 1 minus C z inverse, where K dashed is K by a into e to the power minus a T 1 minus e to the power minus a T and e to the power minus a T is C, okay. So, K dashed by a then I have a delay element. So, will put a delay element here, I can okay; z inverse then the standard one C and this, so this is the output.

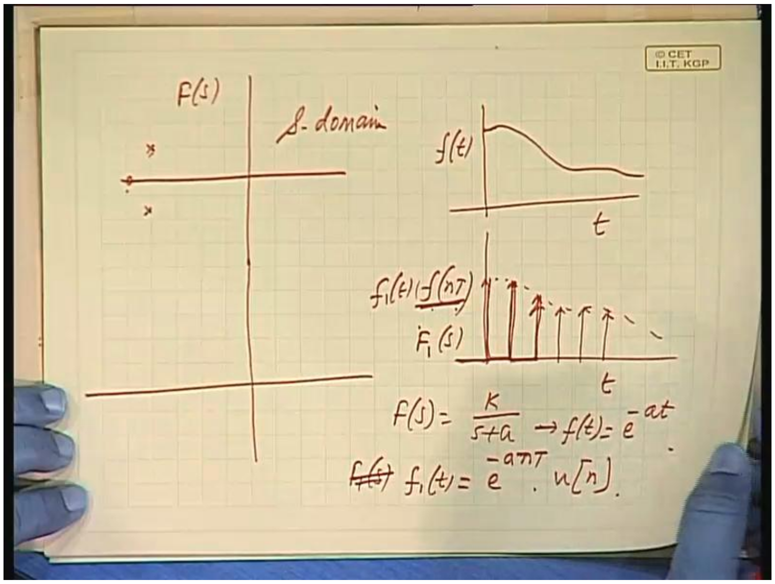
This output will match with the step response of the continuous domain all right. It will match with this, only it will be in the discrete form; whatever be the output in the continuous domain, its sample version will be appearing here. Now, we have seen the technique is similar but it will not generate the same transfer function; it does not generate the same transfer function, as we have got in case of impulse invariance.

So it is fixing, your fixing somewhere in a particular type of input. If you take the response corresponding to one standard input and try to finalise H z; that will not give you an an identical performance when the input has been changed to say, from step to impulse sorry, impulse to step. So, that is why these are known as specific invariance, that is either impulse invariance or step invariance; somebody may go for even ramp invariance, okay.

So, it is specific to a particular signal. These two are taken as standard ones that is step and impulse invariance. But another thing you observe, you have got two such delay elements all right; two such delay elements whereas in case of impulse invariance, you had one delay element. If you go for higher order invariance that is ramp and others then you will get more and more number of delay elements. Things will be, the single block will be quite complicated.

Now, this is all from one factor all right. So, K 1 by s plus p 1 plus K 2 by s plus p 2 and so on; if you have so many factors then for each one of them, you will have such blocks and all of them will be put in parallel, like the earlier case. And then they are to be added. So, you can see the complications will be more. So, this is all about the invariance technique. Now, what are the limitations? Say, let us consider the impulse invariance, what is the difficulty in this? What is the mapping of say, the poles and zeros in the s domain, in the s domain?

(Refer Slide Time: 39:39)

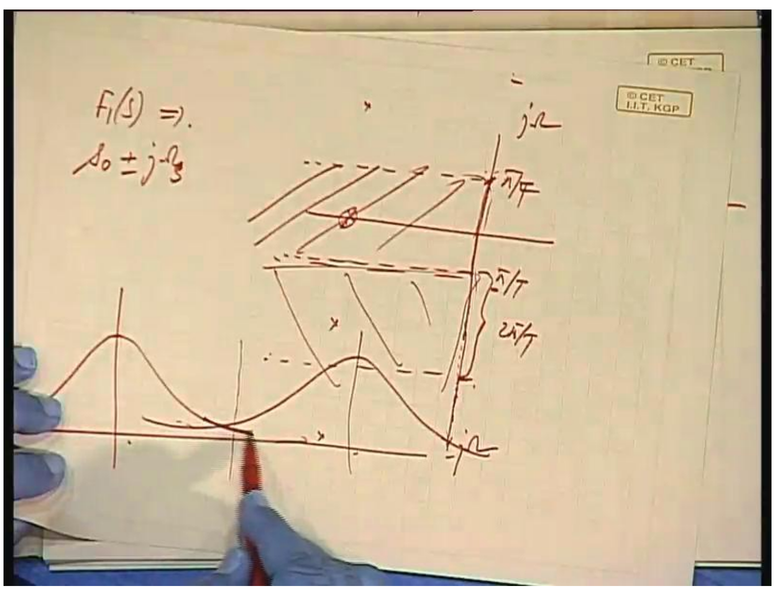


Suppose I have a set of poles and zeros all right, what would be in the, basically what you are doing; a continuous function all right which was like this, you are realizing by a discrete domain function like this. You are taking these values all right. Suppose, F s has roots like this then I call this corresponding to the discrete version; I call this as some F 1 s, the Laplace transform of this function is say F 1 s, what are the roots of F 1 s?

Have you got my point? What will be F 1 s like? If, F s you say K by s plus a, what is F 1 s? If it is K by s plus a, corresponding f t is e to the power minus a t. So, what is f 1 t, if I call it f 1 t, f n t as f 1 t; this is also a function whose values are like this 0 here, again this. So, f 1 t e to the power minus a n T into u n, all right.

So, what is a Laplace transform of this? What is the Laplace transform of this? And if you take the Laplace transform, I will leave it as an exercise in the tutorial class will discuss. If you take the Laplace transform of this, all of you please try this; Laplace transform of this then what are the roots of those for K by s plus a in the original time domain, the root was at minus a, is it not? So, if F s is having a root at minus a, what will be the roots of the discrete version that is Laplace transform of the discrete version, that is F 1 s?

(Refer Slide Time: 42:43)



We will find F 1 s will have, roots at regular intervals here then again; at regular intervals of 2 pi or if you talk in terms of pi or if you take this as minus omega j, j omega and plus j omega then at a regular intervals of that is; if there is a root at s 0 which is in this case minus a, it can be minus a plus j b also. Whatever roots you are getting here, that will be repeated at every interval of omega naught where omega naught is a sampling frequency, is that all right?

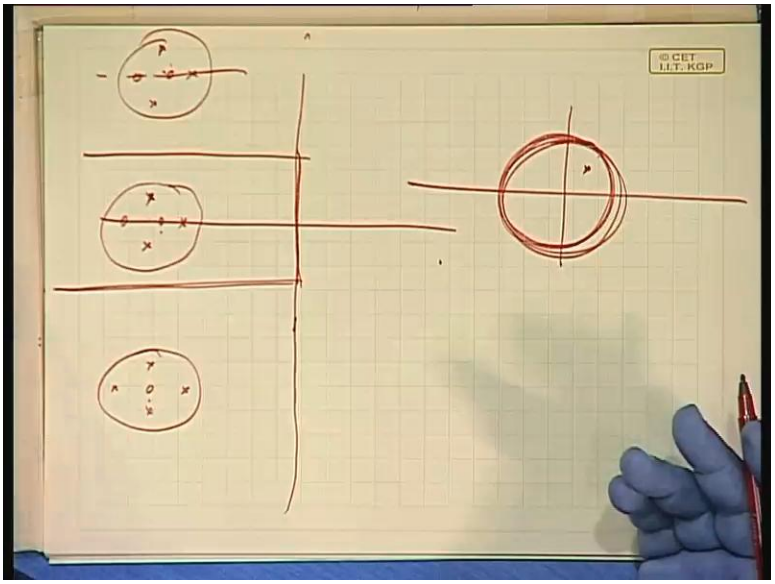
So, s naught plus minus plus minus j omega naught where omega naught or omega sample; I will write omega sample will be the location of other roots, that means there will be an infinite set of roots, all right. At that same location and as if they are mirror images all right, images about this is p i by T, this is p i by T plus at this minus p i by T; that corresponds to omega naught by 2 and omega plus naught omega by 2 and minus omega 1 by 2. Then this will be again 2 p i by T, this is a gap of 2 p i by T; there will be at 2 p i by T, again you will have the same roots. It is like this.

You go you walk along the imaginary axis; you imagine you are walking along the imaginary axis, all right. And to your left there is a pole, say at minus 5 on the real axis; on the real axis at minus a say at minus 5, you are observing a pole as you go past and your sampling frequency say, 1 kilohertz then again after 1 kilo hertz situation will be same; we will be observing again a pole to your left at distance of 5, you keep on observing that all right, you keep on observing that whether you go this way or go to the minus side, that is minus omega.

You will observe an infinite number of poles appear; keep on appearing after a frequency of the sampling frequency that is 1 kilo hertz. You remember if the frequency response of a signal, if the frequency spectrum of a signal is given like this and if we keep on sampling at a regular interval, then the frequency response for the sample signal is repeating after every 2 p i, is it not? It is precisely because of this, that means the number of poles or zeros will be coming will be repeating every 2 p i by T, okay. At every interval of 2 p i by T, we will have repetition of poles and zeros, will appear identical.

So, if I take a frequency, analogue frequency along this; so whatever I observing in this zone, in this strip, it will be repeated again in the strip, it will be repeated again like this, all right. So, it is like this.

(Refer Slide Time: 46:34)



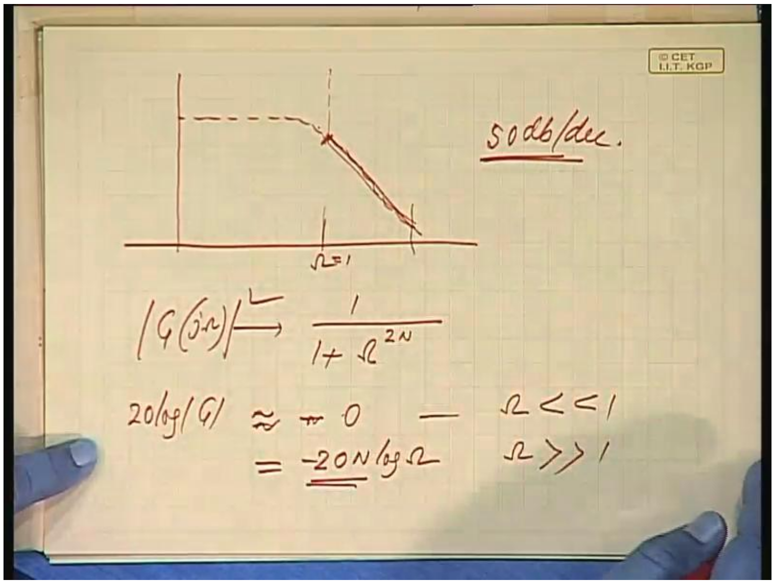
I have just discussed about a pole; it will be happening with the pole 0, the entire set. Suppose, I have two poles and one zero here or may be one or two more zeros here or a pole here; so in this the centre cluster will keep on repeating,…okay an infinite number of times. This will be the pole zeros poles and zeros of the sample function. If the original function f s is having this as the pole 0 cluster then the sample functions; if you take the Laplace transform of the sample function, sample functions poles and zeros will be such clusters, appearing an infinite number of times.

So, if I go from here to here in the z domain, I will make one full circle, okay. Again from here to here, again another circle, I will keep on making circles okay that means; see this point this point a pole here is corresponding to this point then a pole here also will come here, okay. So, aliasing cannot be avoided, unless the frequencies the sampling is very very high. You cannot avoid aliasing effect in case of impulse invariance method, all right.

Aliasing effect means, this you cannot avoid it will be there okay; unless you do some pre pre-filtering okay, some low pass filtering of the signal these repetitions cannot be avoided and hence there will be always some aliasing. So, impulse invariance method has this limitation; there will be an aliasing effect which cannot be reduced and in case of bilinear transformation there is a warping effect, we have to take care of that by pre-warping the specifications.

And we will be in the next class; we shall be taking up of few examples with all these techniques. And will also take up Chebyshev approximations and Butterworth approximations what do they represent? How do you fix up the value of n?

(Refer Slide Time: 49:24)

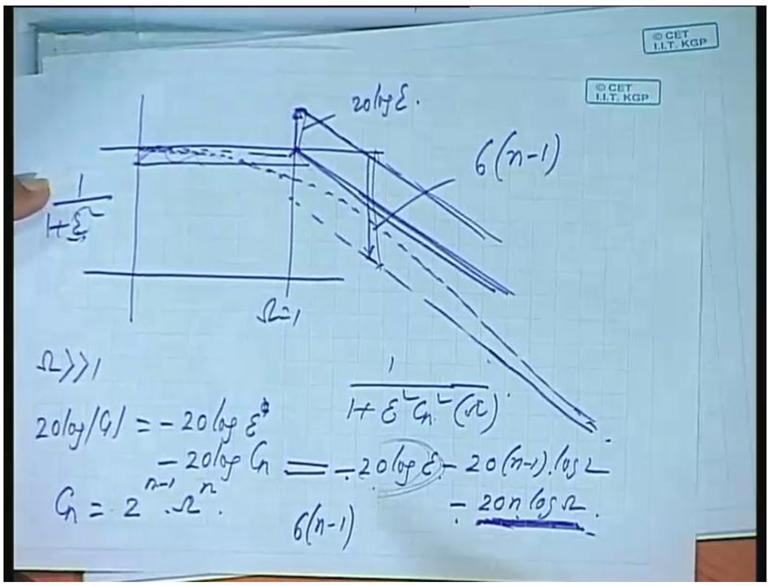


How do you fix up the value of n, the order of the filter? Now, I will just give you a very brief idea. G j omega okay the filter function, if it is given; the rate of fall is given. So, it is 1 plus says omega normalized, if I take the normalised value to the power 2 N. This is square; if I take log of this, so 20 log of G is log of this when omega is very very small, when omega is very very small this will be approximately 1. So, log of 1 that is 0.

That that will be approximately equal to 0 when omega is small. Omega is very large; it will be 20 N log omega, okay. If N is the order then 20 N log of omega, when omega is of course minus minus greater than 1; and in case of Butterworth filter, this is normalised that is whatever is a cut of frequency in terms of that, we are considering. So, if you know if the rate of fall is given approximately; suppose rate of fall is very close to say 50 d b per decade that should match with 20 N.

So, what is the integral value of N closes to this will be 3. So, if I choose N is equal to 3 then I will get at least 50 d b per decade, the rate may be higher. So, that is how you choose the order N in case of a Butterworth filter.

(Refer Slide Time: 51:34)



In case of Chebyshev filter, 1 by 1 plus epsilon square C n square omega, okay. So, when this factor is much larger when omega is much greater than 1; if you have seen the those curves Chebyshev curves, after omega equal to 1 the function C n will blow up, is not? So, at high high values of omega, this will be dominating so this can be neglected.

So, 20 log of G will be 20 minus 20 log of epsilon, epsilon minus 20 log of C n, okay. Now, let us see once again, what was C n? C n was a polynomial n omega, okay. What are, if you just look at C n, all the C n's for example; C 3 it is omega 3, 4, C 2, it was omega square 2. So, C is were taking omega greater than 1; in a polynomial, I will take only the highest term, it will be approximating to these value okay, highest order of the polynomial term.

So it will be, if it is 3, it is 2 to the power 2. So, it is C n will have 2 to the power n minus 1 and omega to the power n, is that all right? So, C n will be 2 to the power n minus 1 then omega to the power n, okay. So, 20 log of C n, this will be giving me minus 20 log of epsilon minus 20 log of this will give me n minus 1 log of 2 minus, how much is it? 20 n log of omega, is that all right?

So, okay let me take it here. This is omega equal to 1, so from this point onward I am drawing only the asymptotes, okay. So, it will be here close to 0, 20 log of epsilon; epsilon is a fraction, so log of a fraction will give me a negative quantity, negative and negative will give me positive. So, 20 log of epsilon though it is minus, but actually it is absolute value will be plus, is it not? Epsilon may be say, one fourth, one fifth so that will give me 20 log of epsilon.

And then 20 n log omega, 20 n log of omega, okay. So, that will give me a slope corresponding to 20 n d b per decade okay which we got for Butterworth filter. Suppose, this was 20 n d b per decade, 20 n d b per decade log of omega means, that gives me 20 log of omega means 20 d b per decade, so 20 n d b per decade. So, this is a slope add with that this quantity; so this loop has to be shifted up. If I have only this term added with this, so 20 n d b per decade fall minus, so it will be falling minus this in absolute term; this is a positive quantity, so it will be shifted up.

And then 20 into n minus 1 into log of 2, log of 2 is 0.3 0.3 into 20, that gives me 6. So, 6 into n minus 1; so from this point, this position again bring it down by 6 n minus 1 d b, okay. So, this will be the final asymptote, is that all right? So, the Chebyshev polynomial will be generating a function which should follow asymptotically this curve, all right? So from there you can find out n. Epsilon is formed from the ripple width, ripple width is 1 by 1 plus epsilon square that we have seen.

So, if the ripple width is given, you know epsilon; once in a epsilon, you know this 20 log of epsilon and then from the given characteristics you know, what should be the rate of fall okay. So, n can be selected from there. So, this is how we select the specifications in terms of n and epsilon in case of Chebyshev filter. So, we will stop here for today, we will take up some problems in the next class. Thank you very much.