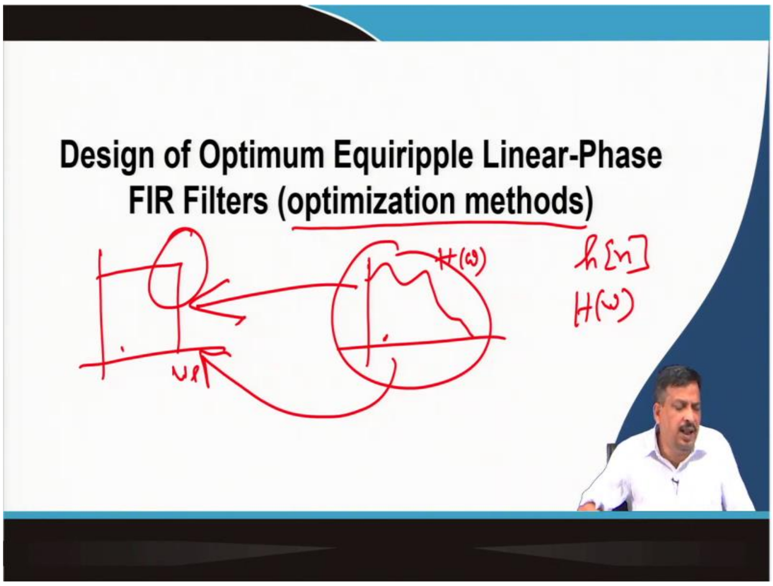
Signal Processing Techniques and its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 40
Design Optimum Equiripple Linear-Phase FIR Filters (optimization methods)

So, last week, we talked about the design of FIR filters, mainly using two methods: one is called Windowing Methods, and one is called Frequency Sampling Methods. So, in the last week, I discussed how to design an FIR filter using windowing methods and frequency sampling methods. So, this week, let us start with the design of FIR filter using approximation methods.

(Refer Slide Time: 00:50)

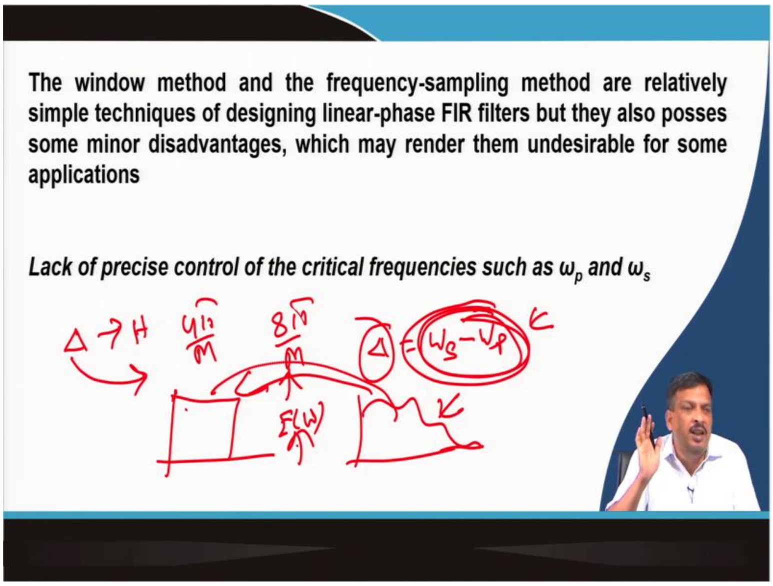


So, what is the name? The name is called Design of Optimum Equiripple Linear-Phase FIR Filter using optimization methods. What is the meaning? This is that let I have a given and desired frequency response; let us this is my desired frequency response, which is ω p, ok. So, I want to design an idle low-pass filter. And then, I have designed a filter. Let us say I have designed an h[n] I have designed an h[n], and I calculate H(ω). So, I get the design H(ω).

Now, what I want to know is how close my desired frequency is to the designed frequency response. So, are there errors between those two things? So, if I say this one minus this one, it is an error in my design. So, I have to optimize the error to get that as close as possible to the desired frequency response. So, that is the main motto of these methods.

So, what I said? Design of an optimum equiripple linear-phase FIR filter using optimization methods. So, why do we go for these methods?

(Refer Slide Time: 02:12)



Let us talk about how the window method and frequency sampling method are relatively simple techniques, as you say, simple techniques to design linear-phase FIR filters. But there is a certain limitation. What are those limitations? Now, if you see that if I use windowing methods, I know that the transition bandwidth δ depends on what type of window I use.

So, if I use a rectangular window, then the transition bandwidth is 4 π by a, and I think 4 π by M, but the present bandwidth is less. But what is there that in the case of the rectangular window, the problem is pass band, stop band ripple, and stopband ripple is very high? Now, if I want to reduce the ripple, then I have to go to the other window.

Once I use the hamming window, hamming window, I know the transition bandwidth increases from 8 π by M. So, in the same order why, my transition bandwidth is larger. So, what is the problem with windowing methods? That lack of precise control over the critical frequency ωp pass band edge frequency, ωs the pass the stop band edge frequency.

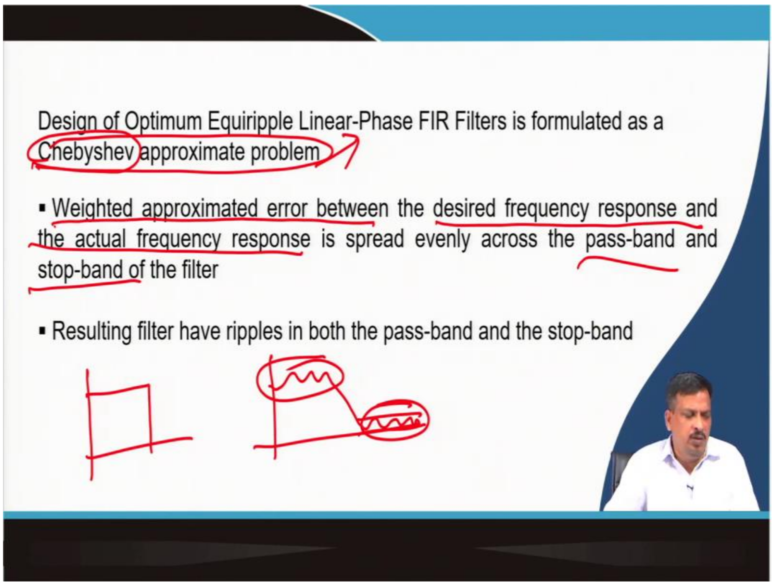
So, the transition bandwidth δ is nothing but a, you know, stop band edge frequency minus pass band edge frequency. So, in this part, I have a lack of control that depends on the window, if I want to reduce the ripple, the transition bandwidth increases; if I want to increase the ripple, the transition bandwidth decreases, and that is ok. However, there is a lack of control over the windowing method.

Similarly, the frequency sampling method also means that I have a sample with a frequency of either type 1 or type 2 in both cases. I have designed using sampling frequency sampling methods, but these frequency responses may not be as controlled as there is no control over this transition bandwidth. So, that is the main drawback in the design of FIR filters using window methods and frequency sampling methods.

Now, I want a method that has the desired control over transition bandwidth. So, what do I want? I want to design an FIR filter using methods that can give me control over the transition bandwidth and the ripple in the pass band and stop band. That is why I said I want to design an FIR filter, which is an equiripple FIR filter linear-phase FIR filter, using optimization methods.

So, optimization is nothing but a graph of; let us say that suppose I want this filter, now I design this filter. Now, I know this is my design part, and this is my desired part; I know the error. Now, I say I want to minimize the error so that my design response is as close as possible to the desired response. That I want. So, that is nothing but an approximation.

(Refer Slide Time: 05:41)



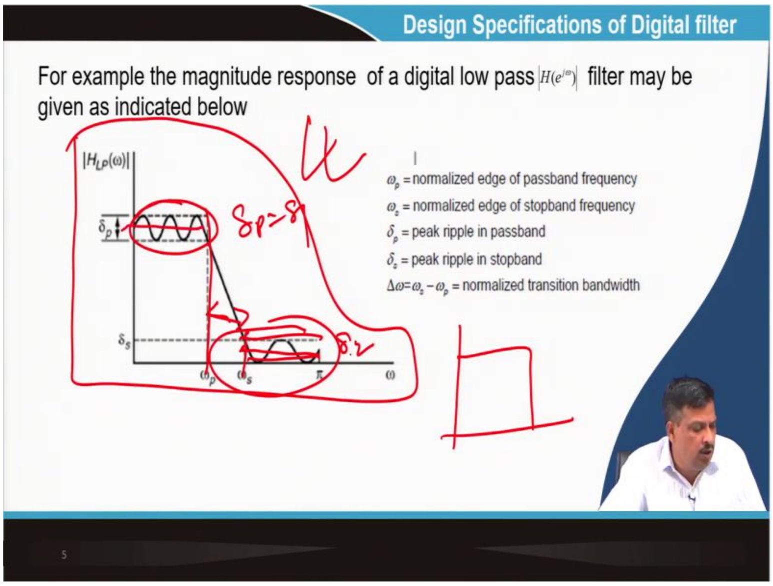
So, that approximation is done using the Chebyshev approximation problem. That is used in mathematics. Do you know the Chebyshev approximation problem in mathematics? We have studied it to optimize that distribution approximation. So, I have a desired distribution; I have a given distribution, and then I change the parameter so that it is close to the desired one. So, that approximation is called the Chebyshev approximation problem. The same method is used to design this kind of filter.

How do you use it? We design and introduce a weighted approximation in error between the desired and actual frequency responses. So, I derived a weighted approximation error function and weighted approximation error between them, so what I want is that, ok, there may be an error between the desired one and the desired frequency and the actual frequency response. But that error should eventually be spread across the pass band and stop band.

So, when I say there is an error, and that error is eventually distributed over the pass band and stop band, I can say the pass band and stop band both the way there is a ripple is there. So, I may want this kind of frequency response, but I can lead down to this kind of frequency response. So, there will be an error that ripples in both the pass band and stop band.

So, I desire to use a weighted approximation error function, which will allow whatever kind of error I desire. So, instead of depending on the window and depending on the methods of frequency sampling, I can say I approx, this is an approximation problem, and I define an error function in which error is eventually evenly distributed among the pass band and stop band, and that error is I have control, I want to minimize that error. So, that is the acceptable design. So, how do I do that? That method.

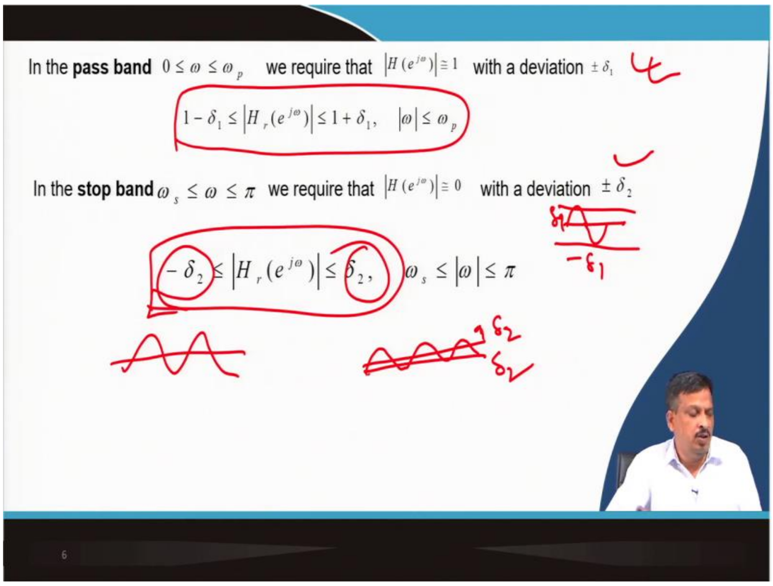
(Refer Slide Time: 08:10)



So, let us say again that you already know that. So, this is the given frequency response of a low-pass filter. So, you know this is called passband ripple. This is called stop band ripple. This is pass band edge frequency and stopband edge frequency, and this difference is called transition bandwidth. So, I have already known all those things. That is the same for all kinds of filter designs. So, this is the desired frequency response. The frequency response is provided per the specifications.

So, the ideal low pass filter is this one, but we cannot design the ideal one. So, people said, ok, I require a low pass filter whose stop band attenuation is 60 dB, pass band is this dB, and transition bandwidth is this one so that I know.

(Refer Slide Time: 09:05)

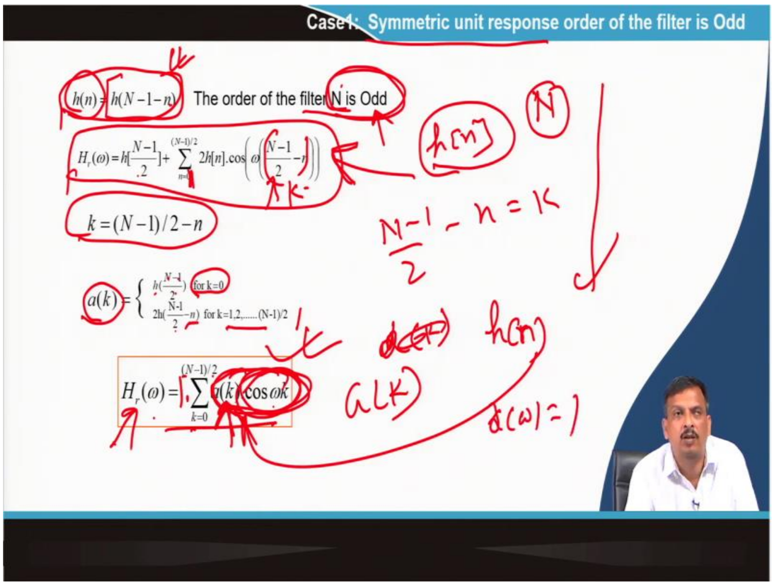


So, now, since the ripple is distributed evenly among this pass band and stop band, I can say that I required this frequency response in the pass band. The magnitude of the frequency response is one of the filters. δ1 is the deviation in the pass band. So, I can say this one is let us δp is equal to δ1.

And instead of δs, I said peak to peak δ2. So, here also δ1 is peak to peak, and here also δ is 2 is also a have a peak to peak. So, it is minus δ to plus δ. So, the stop band ripple will vary from one idea to another. So, one plus δ and one minus δ1. So, that is the stop band ripple variation.

Pass band ripple variation instead of δs, I said pass band has a ripple, then I said in the middle there is a line, so there is an δ2 plus and δ2 minus. So, δ2 plus and δ2 minus that is a ripple condition. Instead of 0 to δs, I said minus δ2 to δ minus δ2 to plus δ2. So, that is the deviation in the stop band, and that is the deviation in the passband, ok. So, that I have known.

(Refer Slide Time: 10:36)



Now, I know that there are 4 methods for the design of linear-phase FIR filters. So, in the case of case 1, which is a symmetric unit response, an order N is odd. So, the order of the filter N is odd, and the impulse response is symmetric. So, that is the requirement for the linear phase. So, this is my linear-phase requirement plus-minus. So, I know this is a requirement. So, if it is symmetric, then I know h[n] is equal to h(N-1-n). That is my symmetric requirement.

And now small capital N is odd; that means the order of the filter is odd, ok. Now, I know if the order is odd already, we have derived for linear-phase condition what is the frequency response; that we have already known that that is my (Refer Time: 11:37). So, Hr is the design if h[n] is my impulse response, then this will be my Hr for my design frequency response. So, Hr is nothing but a frequency response of the design filter.

So, if h[n] is my impulse response, which is symmetric, and the order of N is odd, then this will be my frequency response. Now, in this equation, if I pick, make k equal to this one. So, this part is equal to k, then I can say that I can define

$$a(k)=\begin{cases} h\left(\frac{N-1}{2}\right), & \text{for } k=0 \\ 2h\left(\frac{N-1}{2}-n\right), & \text{for } k=1,2,\ldots,\frac{N-1}{2} \end{cases}$$

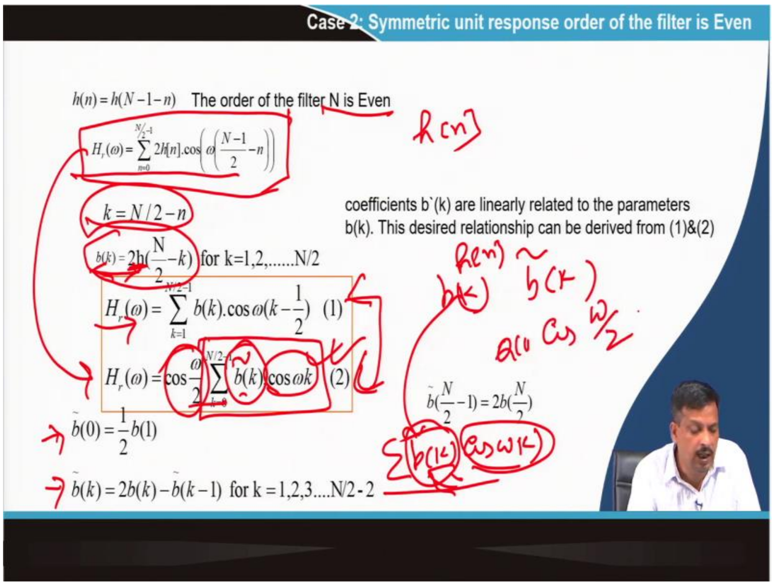
Then, I can say:

$$H_r(\omega) = \sum_{k=0}^{(N-1)/2} a(k)\cos(\omega k)$$

So, instead of summation size N, I write summation size k, ok. So, k is equal to 0 to N minus 1 by 2 α(k) cos ωk. So, α(k) is the filter coefficient or a k; let us say this is a(k), not α a. So, a(k) is the filter coefficient. So, I can say the frequency response is nothing but a convolution between the filter coefficient and cos ωk. Cos ωk is nothing but a for k equal to 1, 2; 0, 1, 2, 3, 4, so for every k, I get an ω. Understand or not.

So, it is nothing but a convolution between the filter coefficient α(k) and cos ωk. Cos ωk is nothing but define the different frequency. So, this is for symmetric conditions and if N is odd.

(Refer Slide Time: 13:58)



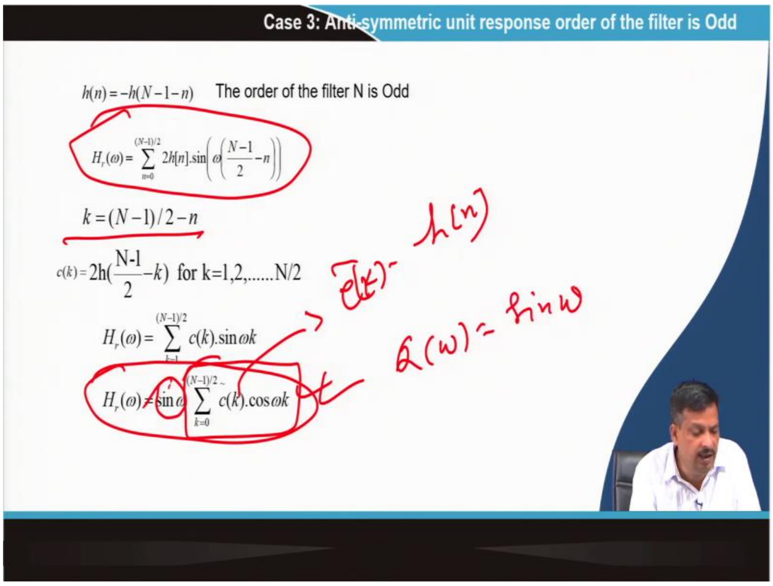
Now, let us say the same thing when the N is even; we have already derived this one; I think in my first 8th-week 1st lecture, you can see the linear-phase requirement; it may be a 2nd lecture. So, this is my frequency response if h[n] is my impulse response. Again, I said k is equal to this one, N by 2 minus n, then I can say b(k) is equal to this one. Let us say b(k) is equal to this one I define; then I can derive; I can rewrite this Hr(ω) as nothing but a cos(ω/2) b.

So, this is my first one, and this is my second one. The difference between the two equations is nothing but a relationship between the b(k) and b cap k. So, this is a b cap k, ok. So, the relations will come like this: if I 1 and 2 are equalized and do the relationship, you can derive this sequence relationship. So, now, you can see that Hr(ω) is written in one kind of format: b(k) into cos ωk.

So, in the case of the first one, it is 1 into this one. In the case of this one, this is convolution where this is the coefficient, which is related to the h n; b(k) is related to the h[n] because if you see the b(k), I define it in terms of h[n]. So, there is a linear relationship between the b(k) and h[n].

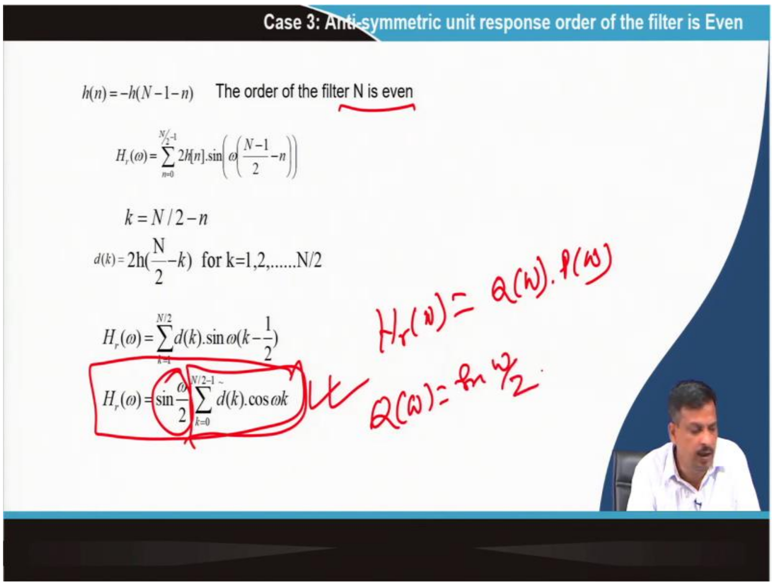
So, b(k) is related to h[n], and a(k) is related to h[n], so all h[n] basically define the filter coefficient. Here also, b(k) defines the filter coefficient and cos ωk; only one cos constant term is there, which is cos(ω/2) ; instead of 1 in the case of symmetric even, it is cos(ω/2).

(Refer Slide Time: 16:05)



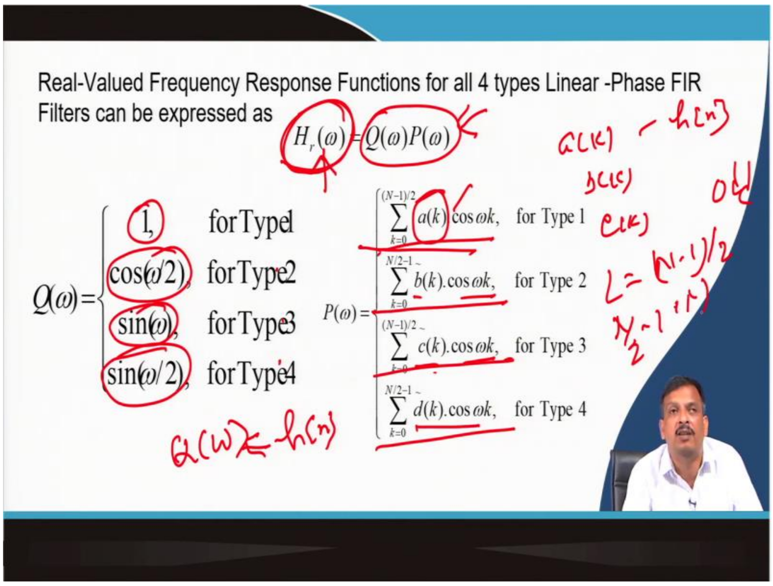
Similarly, for anti-symmetric, again, you take this one, and again, k is equal to this one. Again, the approximation is you can get the same kind of equation; this part is the same: c(k) is related to c cap k is related to h[n], linearly related to h[n]. Again, it has a multiplication factor sin ω, ok.

(Refer Slide Time: 16:34)



Now, again, if I say that N is even in the case of anti-symmetric, I can get the same things. Only the multiplication factor is sin(ω/2) and the rest are the same.

(Refer Slide Time: 16:59)



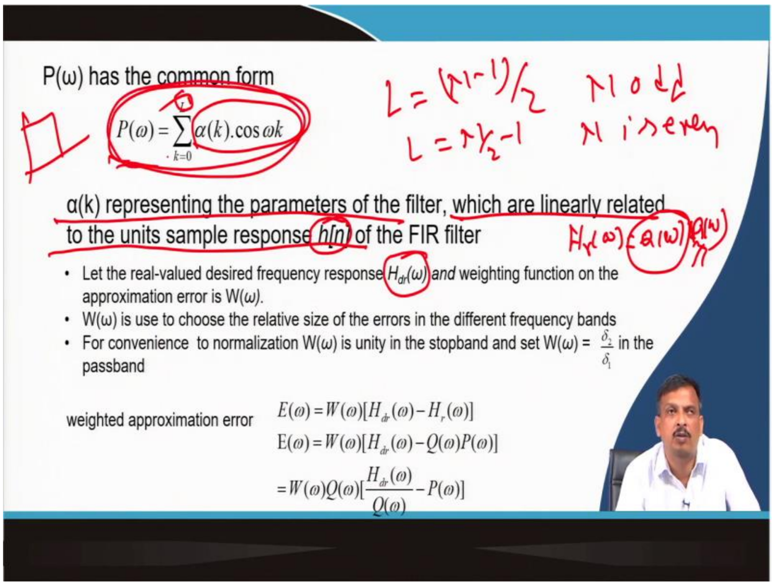
So, if I write all the 4, 4 equation, this equation, this equation, this equation and this equation in a summarized format, I can say Hr(ω); Hr(ω) is nothing but a I can say it is a Q(ω) multiplied by P(ω), where Q(ω) in this case Q(ω) is equal to sin(ω/2). In this case, Q(ω) is equal to sin(ω). In this case, Q(ω) is equal to cos, Q(ω) is equal to cos(ω/2) and in here it is nothing but a Q(ω) is equal to 1.

So, if I want to write down these 4 equations in a generalized form, which Hr(ω) is equal to Q(ω) and multiplied by P(ω). So, Q(ω) in the case of type 1 or case 1, is 1. In case of case 2, it is ω by 2. In case of case 3, it is sin(ω). In case of case 4, it is nothing but a sin(ω/2).

What is P(ω)? P(ω) type 1 k equal to 0 to this one, type 2 this one, type 3 this one, type 4 this one. Now, if you see Q(ω) does not depend on h[n]. because no component of h[n] is here; it is nothing, but a 1 is a constant. cos(ω/2) is nothing but a function of ω. sin(ω) is nothing but a function of ω. sin(ω/2), it is nothing but a function of ω.

But here, all are a(k), b(k), and c k, which are all related to h[n]. So, I can control Hr(ω) by controlling P(ω). How can I control P(ω)? My controlling of P(ω) depends on a(k), b(k), c(k) or d(k) and all are related to h[n]. So, if I find out the filter coefficient precisely and if I want to find out the filter coefficient precisely, such that I can design a perfect Hr(ω) which is as close as to desired frequency response.

(Refer Slide Time: 19:16)



So, how do you do that? So, the problem is, if you see the P(ω) in a common form, I can write down, let us say all are b(k) cos ωk, c(k) cos ωk Sorry, there is cos ωk d(k) cos ωk. I can write it down. So, all are the same kinds of things. So, I can say

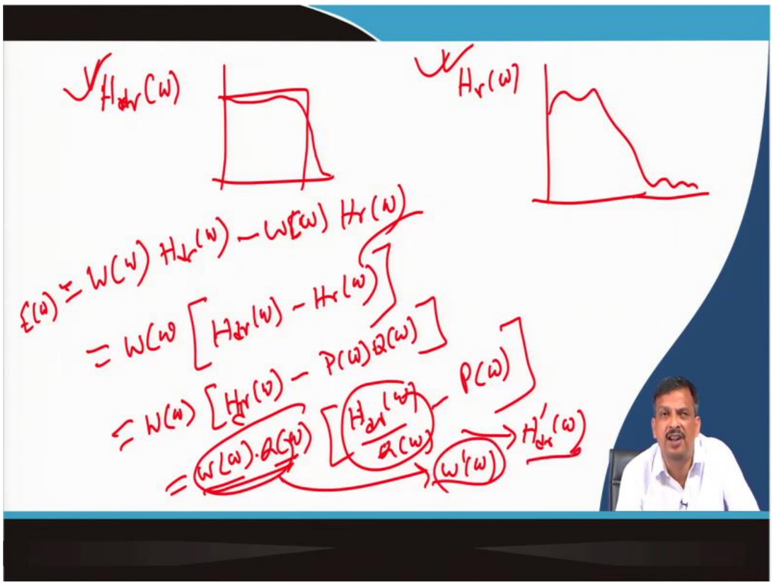
$$P(\omega) = \sum_{k=0}^{N} \alpha(k) \cdot \cos(\omega k)$$

Let us say here L only varies. L is equal to N minus 1 divided by 2 in case of odd and N by 2 minus 1 in case of N is even.

So, I can say only L will vary; L will be N minus 1 by 2 if N is odd. L will be N by 2 minus 1 if N is even. Otherwise, this is the same. Otherwise, all are the same; α (k) is a variable that is α(k), which may be a(k), which may be a b(k), which may be c(k), which may be d k. So, this α(k) represents the parameters of the filter, which are linearly related to the impulse response of the filter.

So, if I say Hr(ω), Hr(ω) is nothing but a combination of Q(ω) and P(ω). I have only control on P(ω), Q(ω) is a constant. So, how do I control P(ω)? By controlling h[n], which depends on α(k). So, α(k) is called the filter parameter. So, let us say the real value desired frequency response is Hdr(ω); that means I want this frequency response. So, I will take a slide here.

(Refer Slide Time: 21:22)



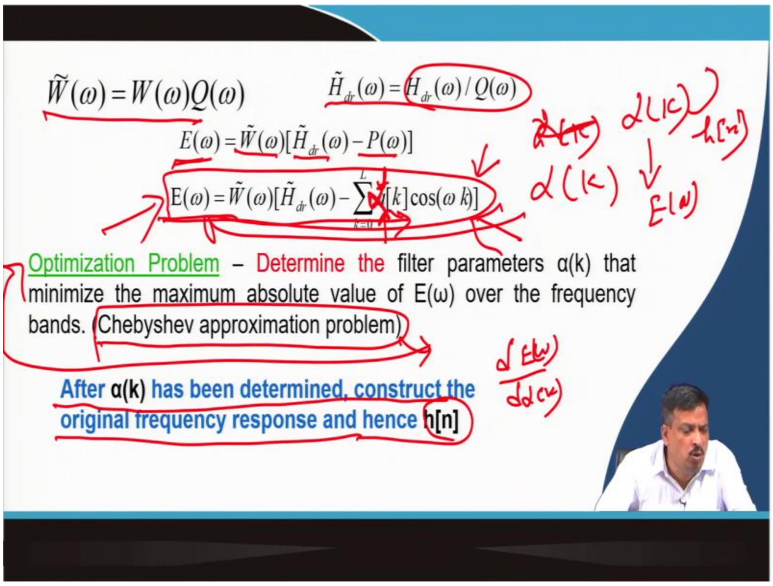
Let us say I want a frequency response, which is my desired frequency response, which is Hdr(ω), which is this one. Now, I design Hr(ω), which may be this one. Then, I want to compute the error between the desired one and the design one. Let us say I define a function W(ω), which relatively sizes the error in the different frequency bands. So, I can say I want to monitor the error by using W(ω). So, let us say I said, instead of this kind of error is sufficient for me.

So, I design a function, which is W(ω). So, W(ω) is the allowable error, I can say W(ω) Hdr(ω) is my new one. This I want, and I design W(ω) into Hr(ω). So, that is nothing but an E(ω), my error. So, an error is nothing but this one. So, this W(ω) is used to quantize the error.

So, I can say the error is nothing but a W(ω) common Hdr(ω) minus Hr(ω). Now, what is Hr(ω)? So, I can say W(ω), Hdr(ω) minus P(ω) and Q(ω), Hr(ω) is nothing but a Q(ω) into P(ω).

So, if I take Q(ω) outside, W(ω) into Q(ω) is equal to Hdr(ω), Hdr(ω) divided by Q(ω) minus P(ω). Now, if I say this one is nothing but a W dashed ω, and this one is nothing but an H dr dash ω. So, what I said? I said H dr dash ω is nothing but a W(ω) and Q ω. This is here.

(Refer Slide Time: 23:49)



And H dr nothing but a Hdr(ω) divided by Q(ω). So, I can say the error is nothing but, so this will be replaced by W dash ω, and this will be replaced by H dr dash ω. So, I can say the error is W cap ω, H dr cap ω minus P(ω). What is P(ω)? P(ω) already we have said this one is the P(ω) where α(k) is the control. So, I can say this is my equation of error. Now, if I told you, I want to minimize the error with respect to what.

So, if you see here, only I can minimize the error with respect to a(k) or α I can say α(k), not a cap, α(k) with respect to α(k). So, this is α(k). So, α(k) is the filter parameter with respect to which I want to minimize the E(ω). So, it is a minimization problem, so take the derivative. So, I want to find a set of α(k) that minimizes the maximum absolute value of E(ω). Minimize the E(ω) with respect to α(ω), α(k), is nothing but a taking derivative with dE(ω) dE(ω) divided by d α(k).

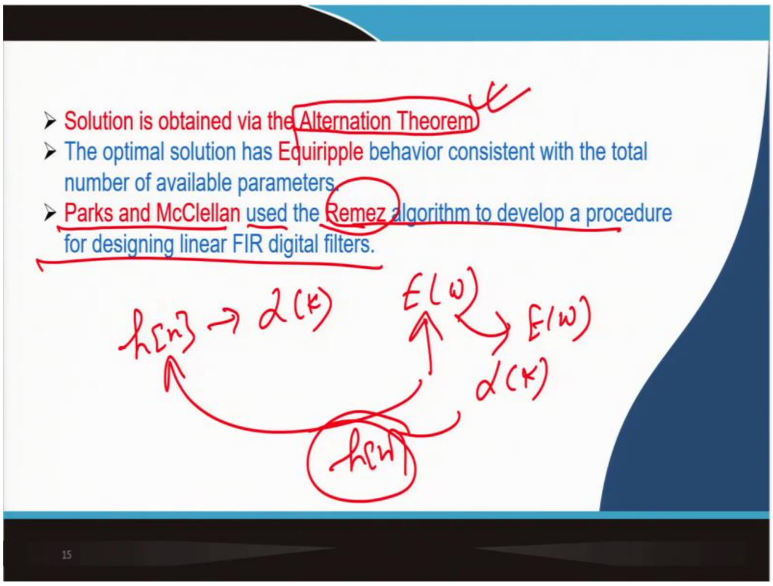
So, with respect to α(k), if I take the derivative and if I minimize this one. So, what do I want? I want to determine the desired α(k), which is related to the impulse response of the filter design filter so that E(w) becomes minimum. So, what is E(w)? It is the difference between the desired response and the design response.

So, that is, maximize the approximation problem, or I can say optimization problem, I want to optimize the error. So, that optimization is done using the Chebyshev approximation problem. So, that is a method by which I do this optimization. So, if you have a mathematical background, you can do that optimization, and you can do that. I am not covering it purposefully because if I cover it for a B.Tech engineer, it will be huge because it can take another 1-week course for this whole optimization problem description.

So, what is my job? Our job has to α(k) has been determined to construct the original frequency response and hence h[n]. So, once I minimize the E(w), for that set of α(k) for which the error is minima, from that α(k) value, if I say the h[n], I can design h[n]. So, α h[n] is given, α(k) is derived, calculate the error, minimize the error and modify that h[n].

So, once I do an approximation problem then I get the desired, once I go close to the desired frequency response, I can say that set of h[n] is my optimal set of impulse response which is close to the desired frequency response. So, that is the optimization problem.

(Refer Slide Time: 27:52)



So, what is the solution to this optimization problem? So, the solution is obtained via the alternation theorem. There is another theorem called the alternation theorem or alternation theorem. This alternation theorem is used to solve this. Now, there is one method Parks and McClellan used, the Remez algorithm, to develop the procedure for design. So, what is the procedure? I have to design a set of h[n], derive α(k), and find out the error; then, I want to minimize this error by changing this h[n].

Once I said this is optimum, E(w) is optimum, that set of α(k) for that set of h[n] is my desired impulse response of the filter. So, that approximation is made using an algorithm, which is the remaining algorithm. You can see that book, or you can go to any standard DSP book and get it. If you are interested, you can raise a question after covering the course. So, send an email, then I can take another class for this one, so that optimization problem, you know, ok.

So, that is the design method for FIR filters using optimization methods. So, if I summarize the FIR filter, you have to know the FIR filter finite impulse response filter, the property it may be linear symmetric or anti-symmetric if the order of the filter is even and odd. So, we know what kind of frequency response will be there for the linear-phase requirement. So, that is common for all kinds of design.

Then, I go for the design of 3 methods; one is called window methods, and I can design an FIR filter using window methods. So, using algebra, I can calculate h[n] multiplied by the W n and then convolve it with the input signal, and I get the desired output. Similarly, I can go for the frequency sampling methods; there is a type 1 sampling and type 2 sampling, and those equations are there. Using that method, I can also design the FIR filter for a given specification.

Then, I go for the optimization methods, which are computer-based design methods that can be optimization algorithms that can be done on a computer to find out a desired set of h[n] for which the difference between the desired frequency response and the design frequency response is minimal. So, that is the summary of the FIR filter.

So, from the next lecture, I will go for the IIR filter design, ok.

Thank you.