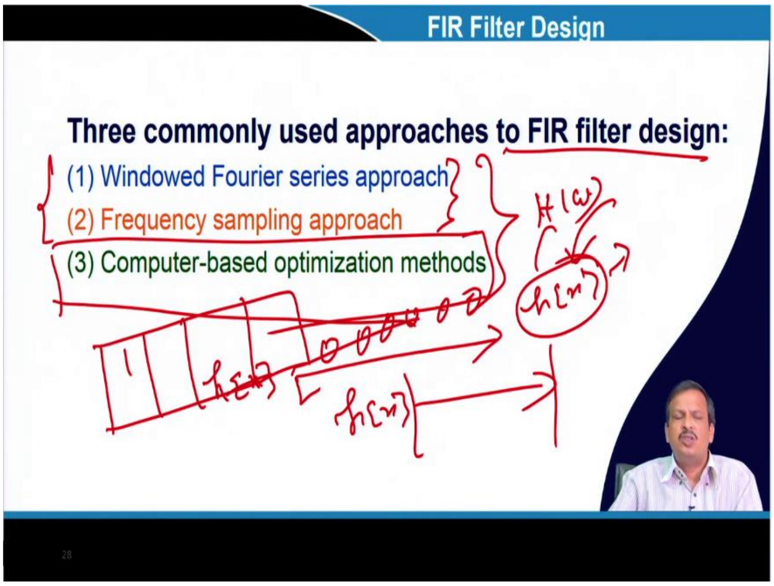
Signal Processing Techniques and Its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 38
FIR Filter Design

Ok, so, in the last class, we discussed the linear phase condition and all those things, ok? Now, we will talk about how to Design an FIR Filter or to design an FIR Filter.

(Refer Slide Time: 00:35)

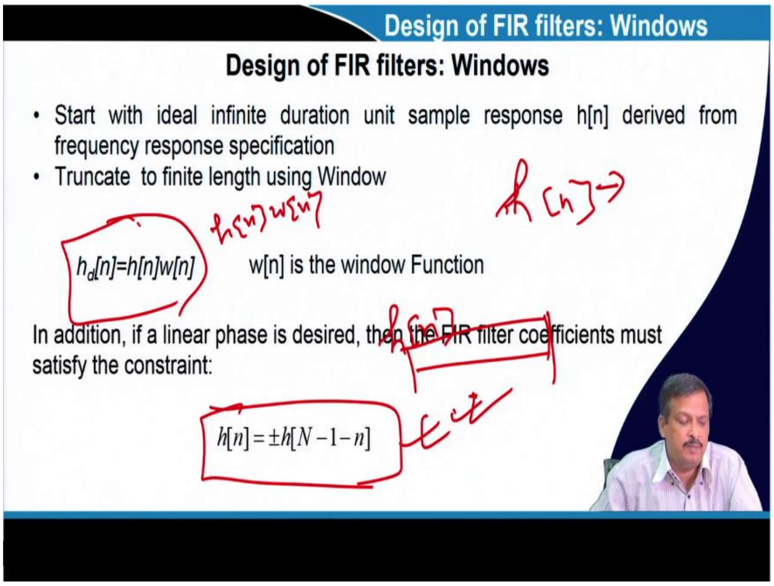
(Refer Slide Time: 00:36)



So, commonly, the FIR filter is designed using three approaches. Windowed Fourier series approach, frequency sampling approach and computer-based optimization methods. I will cover only these two approaches because this approach is in computer-based optimization.

So, what do we want to optimize? We want to optimize that filter response so that it matches the frequency characteristics of the given frequency characteristics of the filter. So, my discussion will be focused on this two method: the Window Fourier series approach and the frequency sampling approach.

(Refer Slide Time: 01:17)



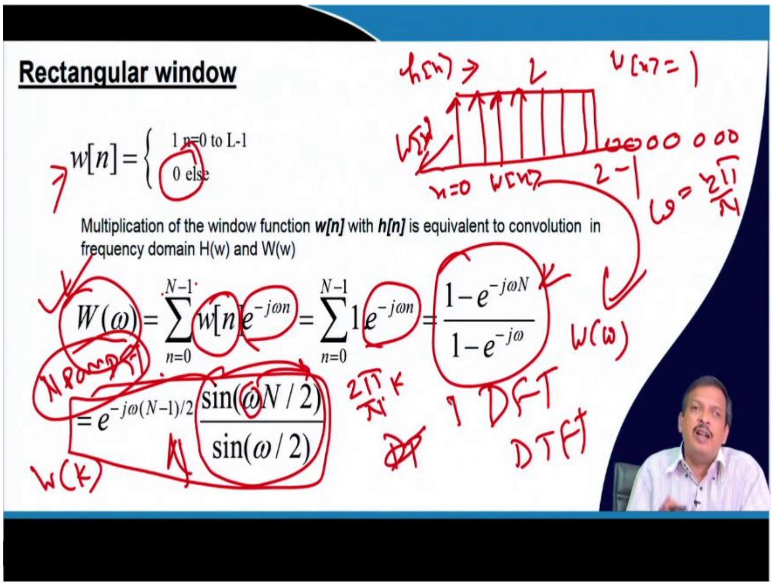
So, first, let us try the window Fourier series approach. So, why is it called the Windows Fourier series approach? Do not read the slide; first, only listen to the audio of what I am saying; just create thinking in your mind. So, what are you thinking? Let us say I do not show you the slide. So, what is the thinking? Thinking that I said windowed Fourier series approach; what is the meaning?

So, if I say that the filter response h[n], h[n] is an infinite series, suppose I know that H of ω; so, once I know the H of ω, how do I get h[n]? By taking the inverse Fourier transform only. So, if I take the H[ω] inverse Fourier transform, I get h[n], which is an infinite series, but I want to design an FIR filter. So, I cannot take an infinite series of h[n].

So, somehow, I have to truncate it. So, how do I cut that portion? So, that means I have to design a window that has a coefficient up to length, let us say, length L, which is the length of the filter and after that, the coefficients are 0. So, then, only I can say I have taken finite impulse into consideration, not infinite impulse. So that means, if I think that my filter impulse response h[n] is derived although it is infinite, I cut it to some length using a window function.

So, I once said a truncated length. So, I use a window function to truncate infinite impulse response in a finite range, and the second condition is that I have to have this one for linear phase design. So, this is my linear phase requirement, and this is my FIR requirement, that h[n] must be multiplied by a window function, which will be defined to cut that impulse response of a certain duration; the rest are 0. So, that is my window function, ok.

(Refer Slide Time: 04:14)



So, how do I design that filter? So, once I say design the filter, then I have to design and define a window. So, the role of the window is cut that h[n] in a finite duration. Let us see if the length of the window is L. So, L will quantify, or L is the order of the filter, which is also ok. Let us say rectangular window. What is the rectangular window? Rectangular window means up to L, the impulse response is 1.

So, this is the impulse response of w[n]; it is not frequency response, it is the impulse response. So, n is equal to 0 to L minus 1, all w[n] impulses are 1, else it is 0, after L by 1, L by L minus 1, it is 0. So, I define a window that has unit impulse response 1 for a duration L minus L, which is L equal to 0 to L minus 1, L square it is 0.

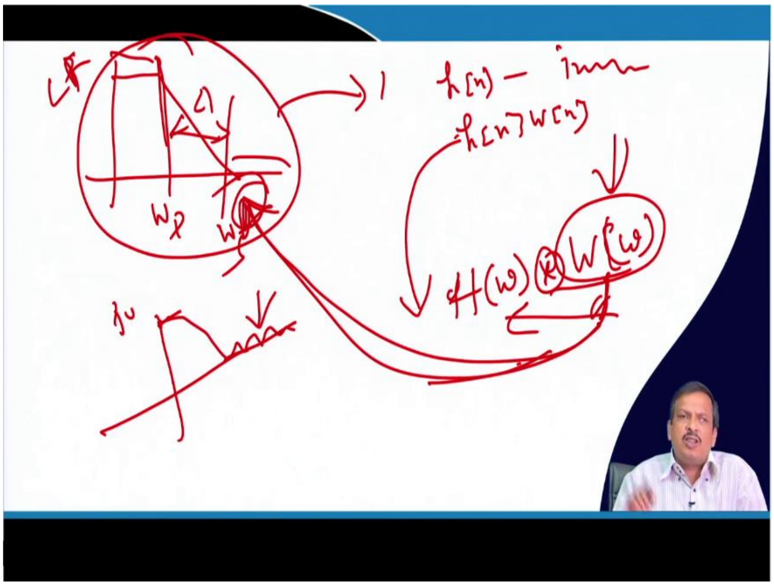
Now, if this is my impulse response small w[n], then what is the frequency response? W[ω] is nothing but a discrete Fourier transform of w[n]. So, w[n] is a sequence, and I have to apply N point DFT: Discrete Fourier Transform.

Let us say apply N point DFT, so once I apply N point DFT. So, n equal to 0 to N minus 1 signal $e^{-j\omega n}$. So, which is called discrete, this is called Fourier transform, but the signal is discrete, but the frequency domain is not discrete. So, it is not DFT. It is DTFT; when I say DFT instead of ω, I will write tell me instead of ω 2π by N.

N is the number of discrete points. So, when I say DFT, then this W k becomes the same thing, only this ω will be replaced by 2π by N into k, nothing else. So, let us do it discrete Fourier transform n equal to 0 to N minus 1 $e^{-j\omega n}$. So, if I simplify it, the summation of an infinite series becomes like this. So, I know sin theta is equal to e to the power j theta minus divided by 2. So, I applied that principle, and I got this one.

So, this is the frequency domain representation of this window; if you see, this is nothing but a sin function. So, there is a window frequency response. So, I have a frequency response of the window.

(Refer Slide Time: 07:45)



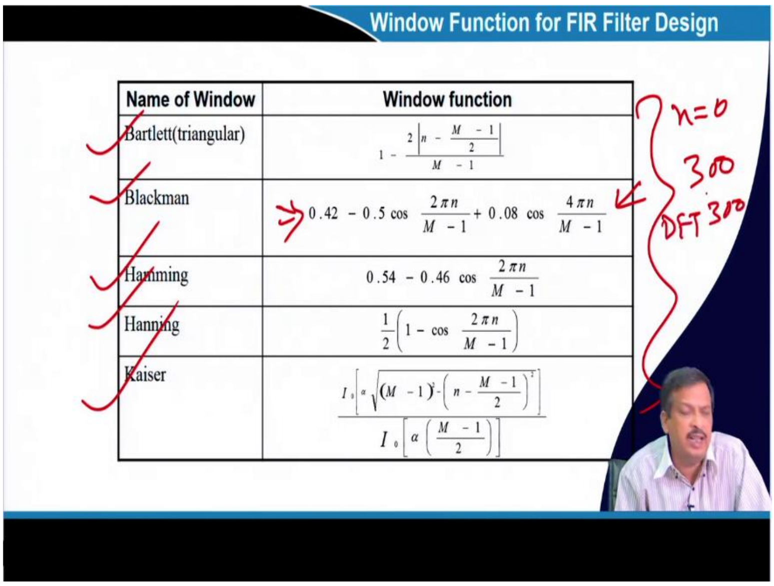
So, when I say I want to design an FIR filter whose specification is given to me, let us say somebody said to me that this is my specification, that this is my transition bandwidth, this is the cut-off frequency, low pass filter. Low pass filter with a cut-off frequency ω P and ω S is the stop band edge frequency, and transition frequency is δ, ripples are given, all are given.

So, this is the frequency response to the filter that I want. So, when I make it inverse Fourier transform, I get h[n], which is infinite in length; I make it finite by multiplying with the window function w[n]. Now, if I think on the reverse side, what is the frequency response of this combined portion? It is nothing but a convolution of H[ω] and window ω, time domain multiplication frequency domain convolution.

So, what exact frequency response do I get? I will get the convolution between the frequency response of my truncated filter and the window function frequency response. So, the window function plays an important role in defining how close I approximate this one because I have to approximate this one. But, since I use a window function for the FIR filter, this approximation will be disturbed. So, how much approximation close an approximation I can make depends on the window function.

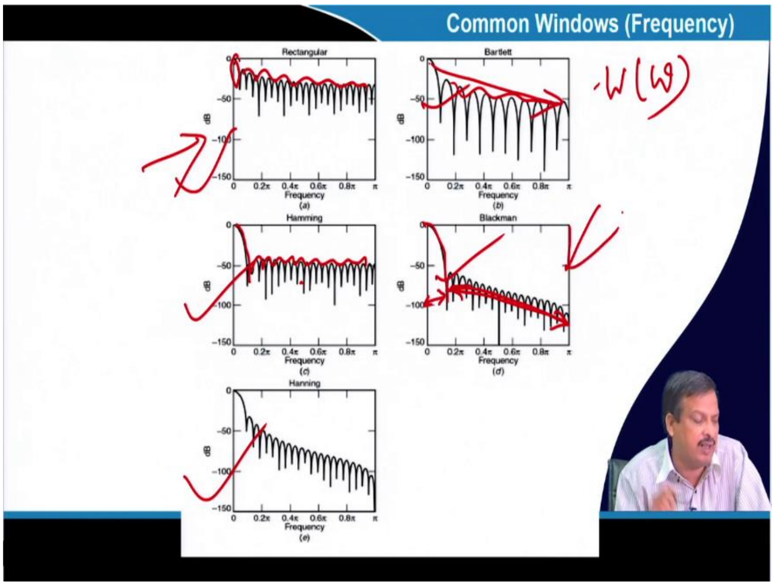
So, what is there? So, if I say window function, window function has a main lobe and a side lobe, but I discuss this during your discrete Fourier transform. So, does any window function have a main lobe and a side lobe? Is that okay? So, the main lobe is the if I want to design a filter. So, the main lobe must be here, and the side lobe could produce the ripple in the stop band, and the main lobe gives me the passband frequency. So, the window has a role in designing or characterising the filter frequency response.

(Refer Slide Time: 10:21)



So that is why people have designed different windows for different purposes. Rectangular window: The main lobe width is small, but the side lobe has a very wide ripple. So, there are a lot of windows: Blackman window, Hamming window, Hanning window, and Kaiser window. So, I have already discussed those window functions during my discrete Fourier transform discussion. So, there is a lot of STFT when I talk about the STFT; we have discussed a lot of this.

(Refer Slide Time: 10:54)

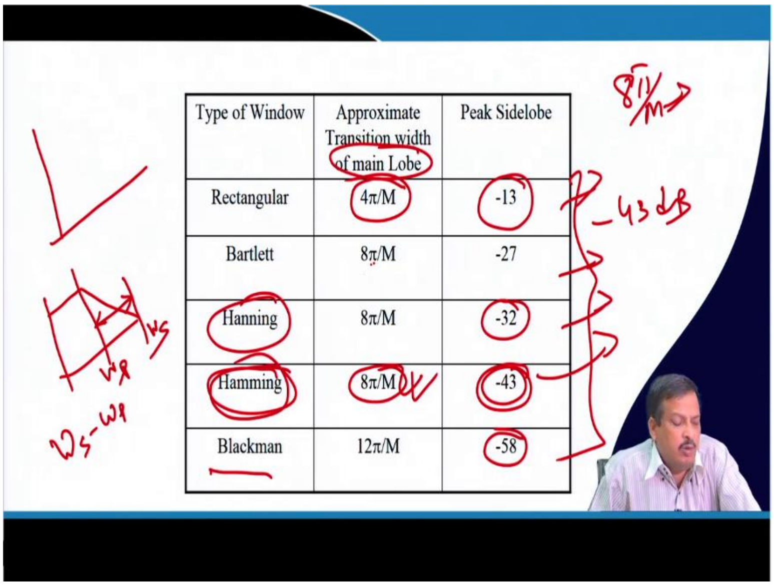


And every window function has its own frequency response. So, those are nothing but a W[ω]. So, we collect some samples of, let us, 300 samples of the windows using this function. How do you do the frequency response? Let us see this Blackman window.

So, I vary n from 0 to 300; let us see the 300 samples I am collecting using this window. And then what do I do? Let us make a DFT of 300 points, 300-point DFT. Then, if I plot it in a magnitude plot, only the magnitude part of the Fourier transform, I get this kind of plot.

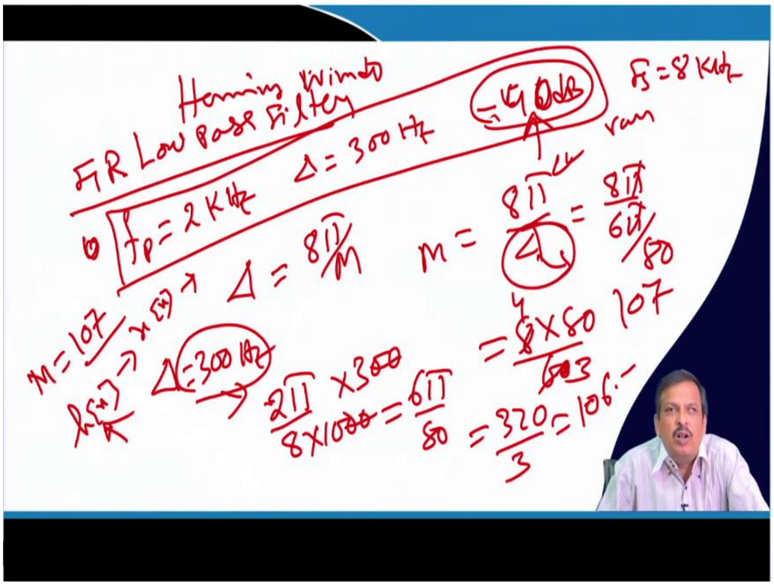
So, Hamming rectangular window, Hamming window, Hanning window, Bartlett window, Blackman window. So, if you see the Blackman window, the main lobe is wide, and the side lobe is also very much attenuated, but here, the main lobe is very narrow, and there is a lot of ripple in the side lobe. Here, the main lobe also has a lot of ripple in the side lobe, but attenuation in the side lobe is much higher. Here, if you see, the main lobe is wide, but the side lobe also has some attenuation is there.

(Refer Slide Time: 12:07)



So, depending on these characteristics, this is the relationship between the window function and peak side lobe and transition bandwidth in the main lobe. So, what is the relationship? So, if I say I want to design a filter, let us say I take an example.

(Refer Slide Time: 12:29)



So, I want to design a filter using a Hamming window. So, what is the if you see when I say the Hamming window, I know the peak side lobe is minus 43 dB. So, if I require a very much attenuation in the side lobe, then I can go for a Blackman window minus 58 dB. So, if the stop band attenuation is my important factor, then I can know which window I will use.

If I say no-stop band attenuation is not that important, then I can say I can use a rectangular window because it has less transition bandwidth. So, if you see this, it is the relationship of the transition bandwidth. What is the transition bandwidth? When you design the filter, the distance between the stop band frequency, edge stop band frequency, and passband frequency. So, ω S minus ω P is my transition bandwidth.

So, when I decrease the side lobe peak side lobe attenuation, then I can see my transition bandwidth is increasing because it is nothing but an 8 π by M, and M is the order of the filter. So, what happens if I choose very good attenuation in the side lobe? Then, my transition bandwidth will increase. Let us suppose somebody said to design an FIR filter, write it down in your pen and paper and then try to do that.

Pause the video here and write down the problem on paper. Then, try to solve it by yourself first, and then see the video and see what you have done wrong. Is that okay? Let us say I gave you a specification: design an FIR filter; I have to design an FIR filter whose FIR low pass filter, let us say, FIR low pass filter; design an FIR low pass filter whose cut-off frequency, let us say cut off frequency f p is equal to 2 kilohertz.

Transition bandwidth is equal to 300 hertz. Ok, let us say I said the peak side lobe attenuation is minus, and I require minus. Let us say for 50, let us say 46 dB or let us say minus 40 dB. Let us say minus 40 dB; I require minus 40 dB. As I said, I have shown you a real specification of a filter, peak side lobe attenuation, peak side lobe attenuation, and peak pass band attenuation I have shown you. So, that is required minus 40 dB.

Now, if this is my requirement, then what window should I choose first? I said I require minus 40 dB attenuation in the peak side lobe, so if I see here, only this is 30 dB. So, I cannot use the Hanning window; I have to use the Hamming window. So, I use the Hamming window. So, in the Hamming window, I know the transition bandwidth is nothing but an 8 π by M, where M is the order of the filter. So, what should the order of the filter be? How do I determine? So, M is equal to 8 π by δ.

Now, if you see, this is in radian and if I cannot define this, it is in hertz. So, I have to make it radian. So, if my δ is equal to 300 hertz, so, if I say it is normalized discrete frequency, and let us say sampling frequency F s is equal to 8 kilohertz, then how do I have to convert 300 hertz into radian first? So, how do I convert? I know 8-kilo hertz means 2 pis; so, 2π divided by 8 kilohertz into 300 hertz.

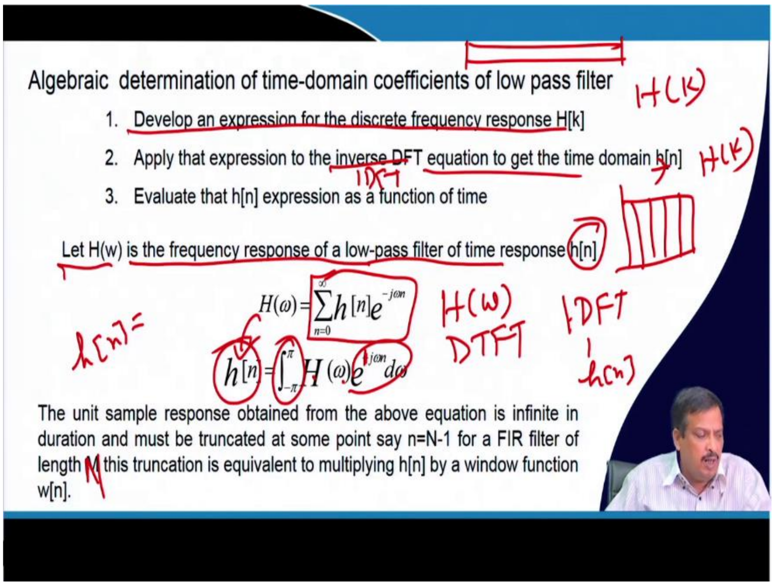
So, I can say 6 π by 80, so if I divide it by 8 π divided by 6 π by 80. So, π pi cancel, it is nothing, but 8 into 80 divided by 6. So, it is nothing but a, let us say, 3 4 320 divided by 3. So, I can say it is nothing but a 1, then 0; then I can say 6-point something. So, I can say let us 107. So, the order of the filter M is equal to 107.

Now, if I know h[n], then I can convolve with my input signal x[n], and I get the filtered output signal. So, my next step is how to find out h[n]. So, the first specification is to discrete frequency conversion relations between the transition bandwidth and the order of the filter. So, if my transition bandwidth is 300 hertz, my order of the filter using the Hamming window is nothing but 106.

Now, in the reverse way, somebody said the order of the filter is 200; what should be the transition bandwidth? You can easily calculate. So, you can calculate either the transition bandwidth or the order of the filter both ways.

So, those so, that from this relationship where I what I get? I get either M or I can calculate the transition bandwidth, and this relationship gives me which window I should choose, which window I should choose. So, I know which window I have to use; I know what order of the filter is ok; then I use the f p and F s to compute the h[n]. So, in the next step, I have to compute the filter coefficient. So, when I want to design a window Fourier transform, I have to compute the filter coefficient.

(Refer Slide Time: 19:39)

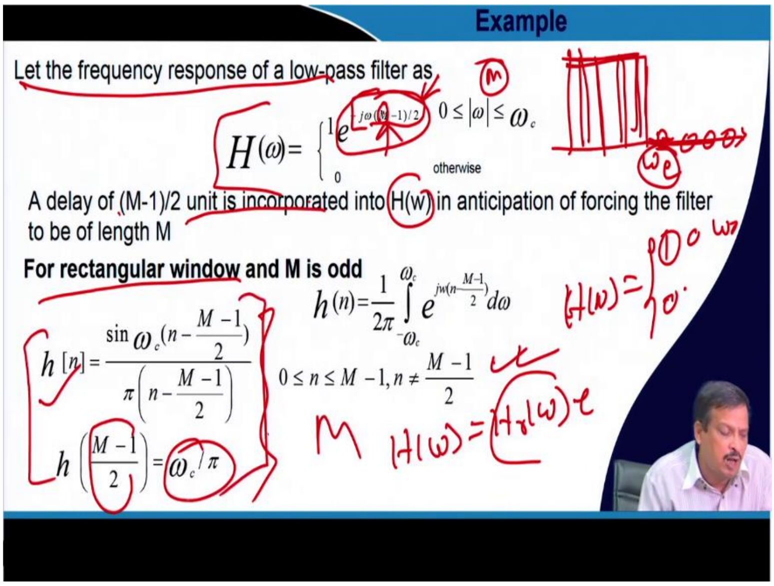


So, how do I compute the filter coefficient? How do I do that? So, develop an expression for the discrete frequency response of H k. So, given from the given specification, you have to develop the H k, this is my H k. So, k is discrete frequency, then apply inverse DFT or IDFT to get h[n]. So, once I know this one, it will be my HK. I can use IDFT to get h[n].

So, this is the procedure. I can use h[n] as a function, and I express it as a function of time. So, let us see H[ω] is the frequency response of a low pass filter response is h[n]. So, I have a low pass filter; I know that the low pass filter frequency response is H[ω]. So, H[ω] is nothing but this one. So, what is h n? It is nothing but an inverse Fourier transform.

So, if it is a Discrete Time Fourier Transform: DTFT then inverse 1 is minus π to π H of ω $e^{j\omega}$; this will be plus; plus d ω. But this one is infinite in duration. So, I have to truncate it N minus 1 point for a filter whose length is N, not M, N, which is equivalent to h[n] by a window function.

(Refer Slide Time: 21:34)

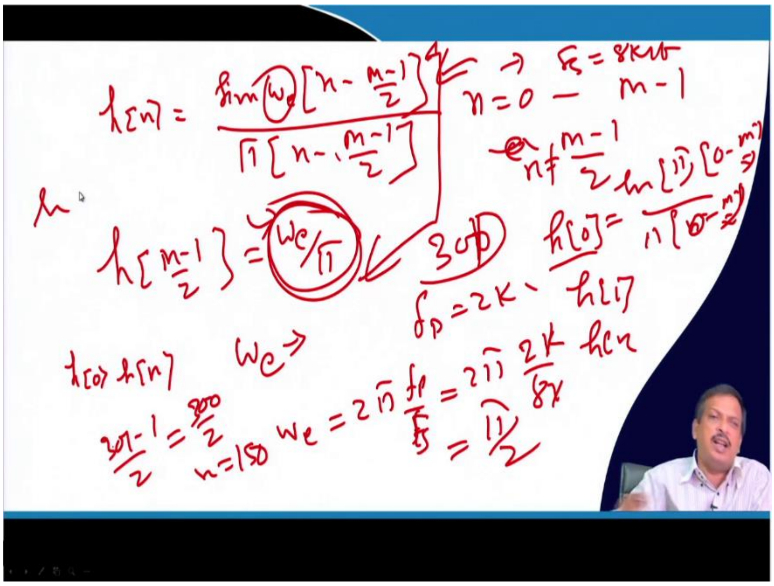


So, let us say this is my low-pass filter frequency response. What is required? Up to ω c up to ω c, this is 1, amplitude is 1; else, it is 0. This is my requirement; ω c is the cut-off frequency. Now, a delay M minus 1 by 2; so, M is the order, and M is the order of the filter M minus 1 by 2. So, M minus 1 by 2 unit is incorporated into H[ω]. So, why is this term incorporated? What is required?

I require this to be all 1 and rest with. So, up to 0 to ω c; so, what will be H ω? Basically, I have to write H[ω] up to 1 0 0 to ω c; this will be 1 for every ω, and elsewhere, it will be 0. I incorporate another term, which is j ω M minus 1 by 2, and M is the order of the filter to make it a linear phase. So, forcing the filter to be length M I incorporated this one because I want the filter to be in a linear phase. So, I incorporate this part.

Then for a rectangular window where M is odd, let us say M is odd, rectangular window and M is odd, rectangular window M is odd. If you do that, then you can calculate h[n]; so, h[n] minus M is the order of the filter. So, M minus 1 by 2, it will be ω c by π and rest were. So, you know that H[ω] is nothing but a Hr(ω) into e to the power that part. So, Hr(ω) is this one for M odd, rectangular window means 1 1 up to that part and 0 afterwards. So, I can get this one. So, this is my h[n].

(Refer Slide Time: 24:04)



So, h[n] expression of h[n] expression of h[n] is equal to; so, I get

$$h[n] = \frac{\sin\left(\omega_c\left(n-\frac{M-1}{2}\right)\right)}{n\pi\left(n-\frac{M-1}{2}\right)}$$

So, except M n is not equal to M minus 1 by 2, this is this one, and when n is equal to M minus 1 by 2, this is nothing, but a ω c by π. Now, suppose I told you to write a program to compute the h 0, h 1, and h 2 to let us know the length of the filter is 300. 300 is the length of the filter. So, compute the 300 coefficient of h[n] and the 300 coefficient of h[n] when this equation is the filter coefficient expressed by this one.

What is ω c? Ω c is the cut-off frequency; let us see if the cut-off frequency is 2 kilohertz, then what is the ω c value? Let us use this filter design where the sampling frequency is 8 kilohertz. So, I know ω c is nothing but a 2π f p divided by F s. So, it is nothing but a 2π 2 kilohertz divided by 8 kilohertz. So, it is nothing but a π by 2. So, ω c is π by 2. So, I know the value of ω c; I know the value of ω c.

I can now produce h 0; so, h 0 n equal to 0 sin of ω c is nothing, but a π by 2, n equal to 0 minus M minus 1 by 2 divided by π into n minus n equal to 0 minus M minus 1 by 2, I get the h 0 value. Then I get the h 1 value, and then I get the h 2 value. Now, if it is ordered, is if it is odd. So, the order is 301, and let us say 301 is the odd. So, if it is M minus 1; so, 301 minus 1 divided by 2, which is nothing, but 300 divided by 2; that means n is equal to at n equal to 150, this will be only ω c by π.

So, π by 2 divided by pi; so, π by 2 divided by π means half. So that way, I can calculate h 0, h 1, and h 2, and I can get the filter coefficient. Once I get the filter coefficient and find out the filter coefficient, I just convolve it with my input signal to get the output. Is it clear?

Ok. So, this is the procedure for designing an FIR filter. First, you define your H[ω], and then you develop what is your expression of h[n]. Once you get it, then you can design the filter. Once you know that M is odd or n is even, what is the equation of that? I get that filter frequency response ok.

So, this is the filter FIR filter design using window Fourier methods. So, you can calculate and try many filter designs using MATLAB or write a C code for this filter design. Can I tell you to write a pass filter C code in C programming language or in MATLAB programming language? Do not use that filter function that is already installed in MATLAB; then, you should not go inside that filter.

So, if you use the toolbox only, then you define the frequency, and then the toolbox will design the filter for you. But, at least you know what are the input parameters you are giving and what the implication of those input parameters in the output, you know, once you know how to design an FIR filter.

Thank you.