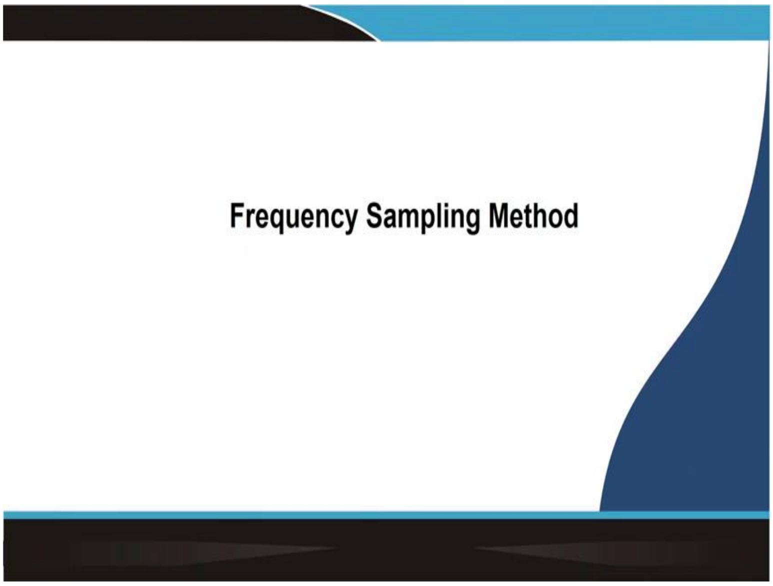
Signal Processing Techniques and Its Applications
Dr. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 39
Frequency Sampling Method

Ok, so, in the last class, we discussed that algebraic expression for calculating the h[n], which will be multiplied by the w[n], and then we get that in a combined filter response. And, then the signal will be convolved with that, and we will get the desired FIR filter implementation using windowing methods.

(Refer Slide Time: 00:44)

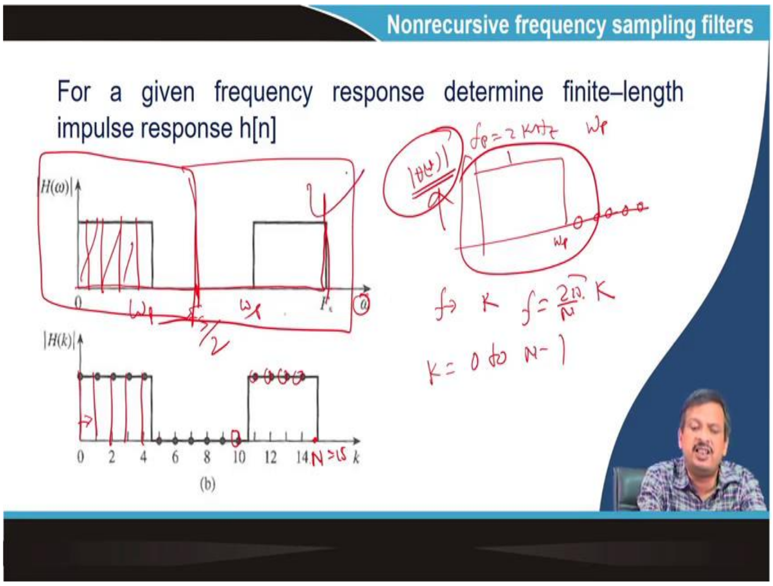


So, today, we talk about Frequency Sampling Methods. So, how do I implement an FIR filter using frequency sampling methods?

(Refer Slide Time: 00:57)

So, as you know, what is the frequency sampling method? So, let us say I said there are two methods of FIR filter are there in frequency sampling. One is called the Non-recursive FIR filter; another other is called a Recursive FIR filter. Let us first talk about the Non-recursive FIR filter.

(Refer Slide Time: 01:12)



So, what is that frequency sampling method? So, what is the frequency sampling method? It is nothing but the given specification of the filter, which is nothing but the spectrum of that filter. That means I have given design and low pass filter whose cut-off frequency lets us 2 kilohertz. So, I let us fp is equal to 2 kilohertz, and then I know the sampling frequency, and then I know ω p.

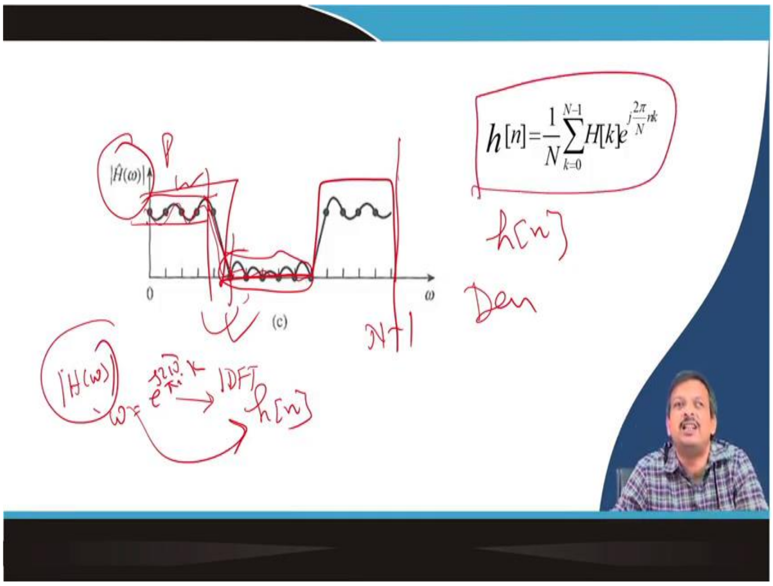
Once I know ωp, what is given to me? This one given to me is the magnitude response H, the ω mod magnitude response, which is given in specification ok. So, that is given in specification up to the ωp, this will be 1, and this will be 0 after ωp that was the specification is given. Now, if I do it in the actual magnitude response of this filter, it will look like this: 0 and then F s because the magnitude response is symmetry at F s by 2.

So, at F s by 2, it will be symmetry, ok, or not. So, that is why I get this one is ωp and this one is also ω p. So, this side and this side both side I will get this is the magnitude response of the filter, and this one is the symmetry property. So, I use F s by 2 part is sufficient, this part is sufficient, but in theory, I get that entire part ok. So, this has given me what I can do now.

So, this axis is ω, or I can say this axis is f, which is normalized discrete frequency; this axis can be represented by k, and the relationship is f equal to 2 π by N in into k. So, this axis I sampled. So, I take a sample, let us say take I take N number of sample, N number of sample let here it is 15 number of sample N equal to 15. So, if I take N number of samples, then the resolution is 2 π by N into k.

So, that I know, that becomes k. So, k varies from 0 to N minus 1. So, I take those samples. So, instead of taking that continuous frequency response, I sampled the frequency response, which is why it is called the frequency sampling method. So, I took the finite length of frequency sampling N, N equal to 15, maybe N equal to 64, and N equal to 320. So, once I take it and then once I take the inverse Fourier transform and design the filter, then if we draw the frequency response of the filter, it will look like this.

(Refer Slide Time: 04:14)

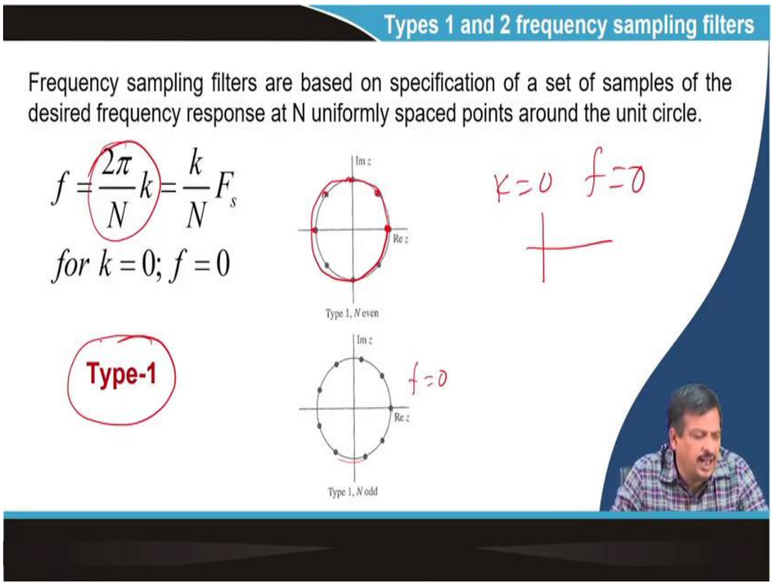


So, that introduces some ripple due to the finiteness, some ripple in the pass band, some ripple in the stop band, and some transition bandwidth. So, this is the symmetry property. So, the signal is time domain all. So, this is the design filter's frequency response and magnitude response. So, this is the design filter, and this is the given filter. This is given by the user, and I take the frequency sampling. I design the filter, and once I draw the frequency response, I will get this one ok.

So, now, the frequency sampling method, I can take it due to the finiteness of that impulse response, the filter response, instead of this sharp cut-off, which changed to this kind of thing. So, how do I design this one? So, how do I derive this h n? The method is very simple: first, from the given specification, we draw the mod of H[ω] and then sample the ω, $e^{-j2\pi/Nk}$. So, $\omega = e^{j2\pi/Nk}$.

So, that I sample k is 0, 1, 2, 3, 4. So 0, 1, 2, 3, 4 up to N minus 1 ok, then from there I compute h[n]. So, h[n] is nothing but an inverse discrete Fourier transform of H[ω]. I get h[n], which is nothing but this one.

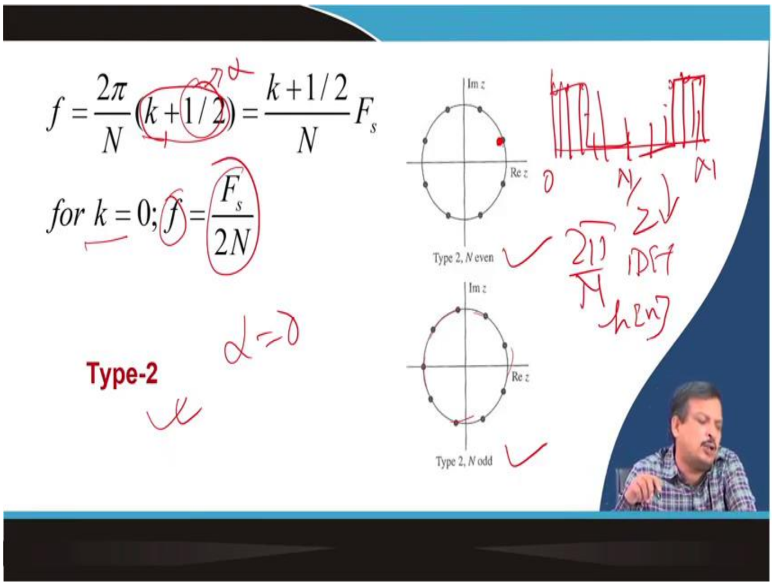
(Refer Slide Time: 06:21)



So, this is the simple method. Now, what is the problem here? The problem here is one: This type of sampling can happen in two types; let us sample it. So, f is equal to twice π by N into k. Now, if the k is equal to 0, then f is equal to also 0. So, I start from 0; so, the unit circle is the unit circle. So, I take the sample at the 0th location, then uniformly sample, including a starting from the 0th location.

So, if N is equal to even, I will get this one; if N is equal to odd, I will get this one, but start from f equal to 0, inside the unit circle ok. So this one is called type 1 frequency sampling filter type 1 frequency sampling filter. So, in type 1, the frequency sampling starts from 0.

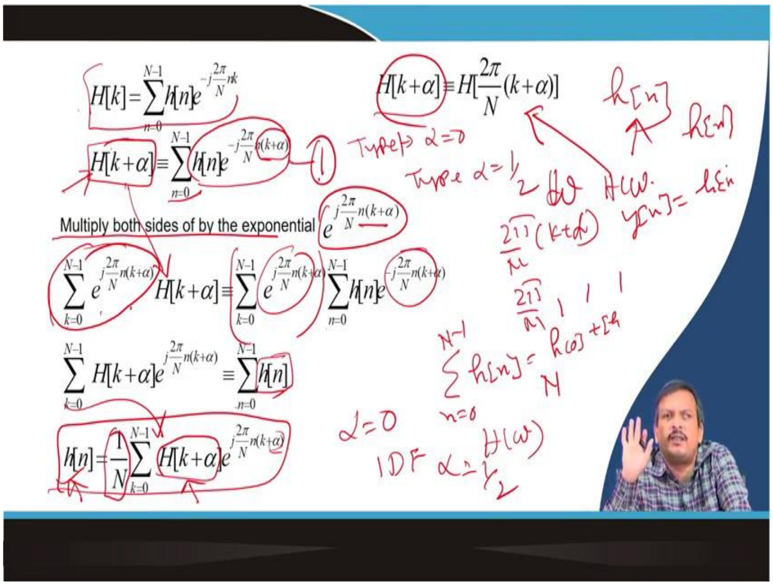
(Refer Slide Time: 07:25)



Then I can say that instead of 0, frequency sampling can start from some little bit of offset that is nothing but a k plus half I added; this is called type 2. So, if k is equal to 0, f is not 0, and f is F s by 2 N. So, it starts from here. So, this is for N equal to even, and this is for N equal to odd, and this is the unit circle. So, this is the type 2 sample position, and this one is the type 1 sample position.

So, when I say type 1, then I say k is k plus α. If I say it generalizes, then α is equal to 0. So, if this half is represented by α. So, if α is equal to 0, it is type 1; if α is equal to half, it is called type 2. So, I can implement a type 1. So, f equals to start from 0 f equal to 0 sampling at 0 from 0. So, the frequency is 0; that means, including the DC and without the DC, I can also implement that filter, which is F s by starting from F s by 2 N. So, type 1 implementation and type 2 implementation.

(Refer Slide Time: 08:49)



Let us talk about how to implement it. Let us say a generalized form of H[k]. So, what is H[k]? H[k] is nothing, but if h[n] is my impulse response, then H[k] is nothing but a frequency transform of h[n] or a Fourier transform of h[n]. So, if I generalize it, which is nothing but a k plus α, which includes type 1 and type 2, when I say type 1, then I say that α is equal to 0; in the case of type 2, I say α is equal to half ok.

So, I can say the frequency response is nothing but the inverse Fourier transform, but k is equal to k plus α, and k is replaced by k plus α. So, basically, H[k+α] is nothing but a h. What is the resolution? 2 π by N into k plus α that is the frequency, f or ω I can say this is H ω. So, the H[k+α] is equivalent to H[ω] which is sampled at 2 π by N, is it clear?

Now, let us see equation number 1. So, in equation number 1, if I multiply both sides with $e^{j\frac{2\pi}{N}(k+\alpha)n}$ and take the sum over k equal to 0 to N minus 1. So, I am writing this extra part. So, I am writing on this side, and I will write on this side also. So, I write on this side also, now if you see this one is equal to this one, then these two will cancel each other. So, this is nothing but a n equal to 0 to N minus 1 h[n].

So, what is there? n equal to 0 to N minus 1 h[n], which is nothing but a h 0 plus h 1 plus. So, if all are 1, it is nothing but a N. So, if it is N, then I can say h[n] is equal to 1 by N in this one. So, now, if the α is equal to 0, this is nothing but the simple inverse DFT, IDFT; if it is α not equal to 0, then I can also calculate.

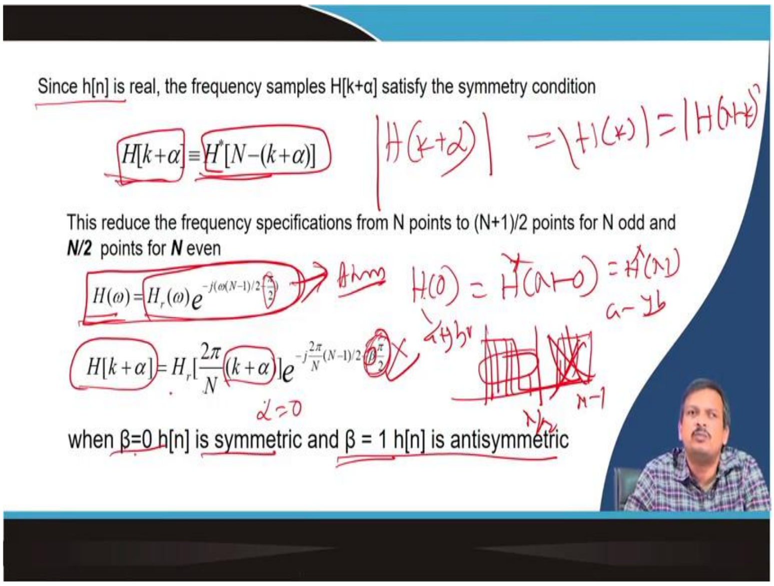
So, what is my intention? My intention is that I have given that $H[\omega]$ is given my given by the user, I frequency sample that $H[\omega]$ and from there, which is nothing but a $H[k+\alpha]$ depending on the implementation, I can say if it is type 1 the straight forward α is equal to 0 and I can calculate that $h[n]$ using inverse Fourier transform.

So, the idea is very simple: I have the specification given the magnitude, and the spectrum of the filter is given. What is spectrum? the y-axis is the magnitude, the x-axis is the frequency, and now this is symmetry. So, if I know that any of the discrete Fourier transforms, as you know, any discrete Fourier transform is symmetric, that this is 0, this is N. So, I know N by 2-point symmetry. So, if this is my filter response, this will also be repeated here. So, from there, I sampled it.

So, it is nothing but an N number of samples. So, it is 2 π by N is the sampling, the resolution of the then I take the number of samples. So, N number of points N number of samples I have collected, and once I have collected the N number of samples, I take the inverse discrete Fourier transform, and I get $h[n]$ in case of type 1. In the case of type 2, instead of inverse discrete Fourier transform, I am doing the same convolution, but with the α value, α is equal to half.

So, my ultimate objective is to find out $h[n]$; once I get $h[n]$, then I know $y[n]$ is nothing but a convolution between $h[n]$ and input signal $x[n]$. So, given the specification of the filter, I have to find out the frequency response of the filter and from the frequency response, I have to sample that frequency response, and I have to find out the impulse response of the finite impulse response of the filter. Once I get the impulse response, I just convolve it with the input signal, and I get that output of the filter.

(Refer Slide Time: 14:01)



So, if I do the mathematics, that is what I should do since h[n] is real, then I can say H[k+α] is also symmetry. So, which is in the complex domain H[k+α] equal to H star N minus k plus α, but when I said the mod domain, it is nothing but an H of. So, I can say I know H(k) is the mod of H(k) is equal to the mod of H[n-K]; that is, I know from the symmetry function.

In the case of a complex domain, it is nothing but a complex conjugate. So, you know if it is H 0, then I know it is nothing, but a H of complex conjugate of H 0 is H of minus 0 star. So, which is nothing but the H[n] star, so, if this is a + jb, this will be a - jb, which is the complex conjugate that I have given an assignment problem. also, this type ok. So, since that amplitude response or the magnitude response is symmetric,

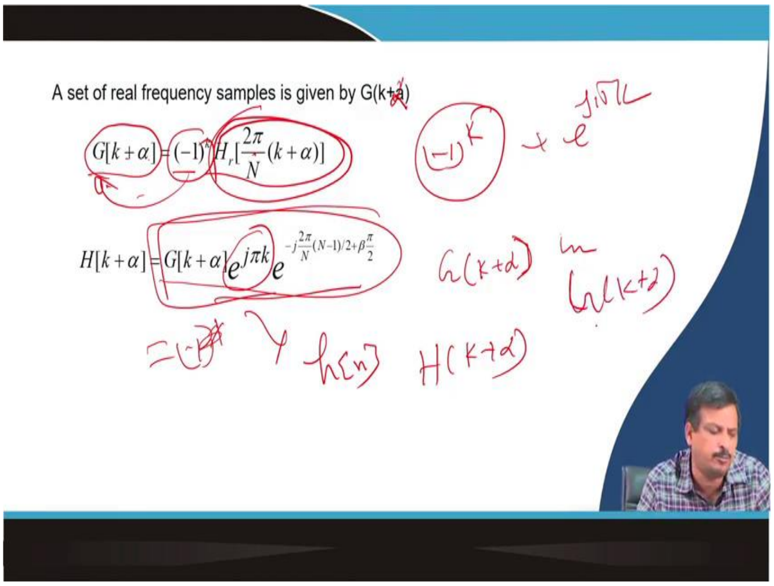
So, instead of, when I do the frequency sampling, you take the entire filter and N minus 1 number, and you collect the sample. So, instead of that, I can take only simple N by 2, which is sufficient; I do not require this portion. So, when I do the inverse DFT, I can simply take the N by 2 points, which is sufficient. So, N by 2 N may be an odd or N may be an even. So, in the case of N even, I can take N by 2; in the case of N odd, I just take N plus 1 by 2 points sufficient, ok?

So, how do I calculate math? If you see H[ω] is equal to this one, this one we have already said that symmetry and anti-symmetry properties of the filter, we have derived the linear phase condition if I took the linear phase condition, this is the generalized equation of the frequency response of the filter.

Now, I make it more generalized in the sampling domain, taking instead of π by 2, I take β π by 2. So, β is equal to 0, which means symmetric and β is equal to 1, which means anti-symmetric. So, this is the equation of an anti-symmetric frequency response.

So, if I multiply with the β, the β is equal to 0, which means it is symmetric. So, this portion will not be there and β equal to 1s means it is an anti-symmetric filter, and Hr(ω) is nothing but a sample at 2 π by N k plus α. So, α is equal to 0, which means type 1 and α equal to 1s, which means type 2, α equal to half, which means type 2.

(Refer Slide Time: 17:01)



So, if I derive that equation, let us set the real frequency sample as given G k plus a. So, the real frequency sample is given G plus not a; it is α. So, insert symbol problem; so, α.

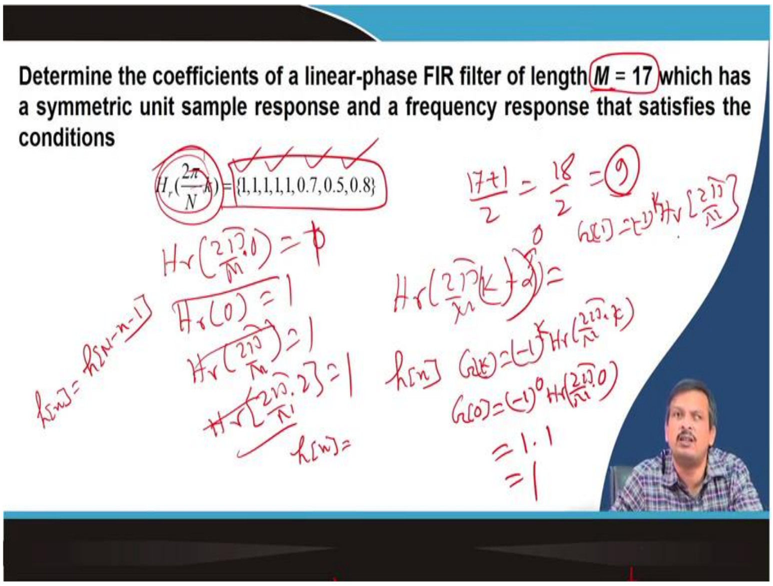
So, I can say
$$G[k+\alpha] = (-1)^k H_1\left[-\frac{2\pi}{N}(k+\alpha)\right]$$

Now, minus 1 to the power k, what is the meaning?

$$H[k+\alpha] = G[k+\alpha]e^{j\pi k}e^{-\frac{j2\pi(N-1)}{N}(k+\alpha)-j\pi\frac{N-1}{2}}$$

So, I get this one. So, what will be given? So, G k plus α is nothing but a given specification of the frequency response of the filter.

(Refer Slide Time: 18:03)

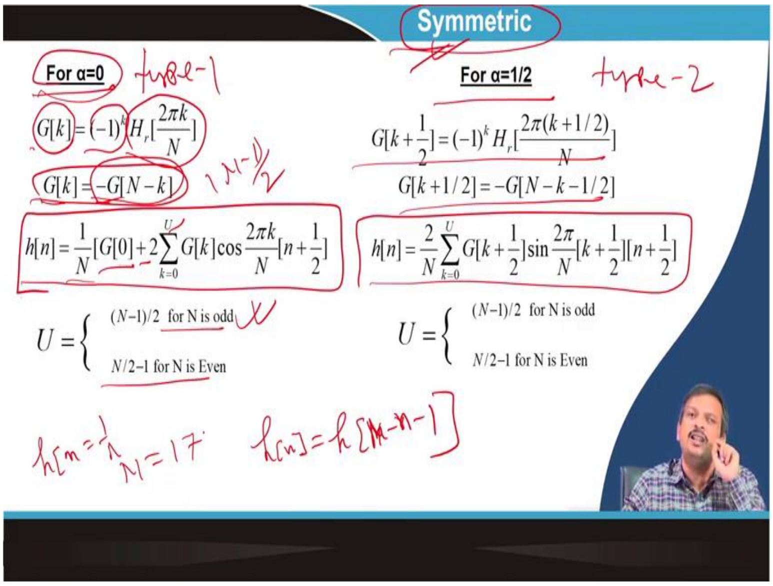


Let us give an example. Let us say I took your first example. Then, let us see the given specification is given this one: H r 2 π by N into k is equal to; so, k is equal to 0 means. So, H r 2 π by N 0 is equal to 1; that means H r 0 is 1 1 and H r 2 π by N is 1 H r 2 π by N into 2 is equal to 1.

So, if you see if I say M, the order of the filter is 17. So, 17 is the odd; so, 17 plus 1 divided by 2. So, I require 18 by 2. So, I can say 9 impulse frequency responses are required and 9 samples are required. So, those are the 9 samples given; the rest of the 9 I can generate. I know it is nothing but a complex conjugate. So, if it is real, it is the same. So, the next portion will be the same.

So, once I know this, I can implement this type 1 and type 2 filter. How? So, I have to so, G k. So, H this one I know. So, G k is nothing but a minus 1 to the power k, or I can say this one is minus 1 e to the power j. So, I can say that this is given. So, I can multiply this one, and I will get this one, okay? Then, I have to calculate h[n].

(Refer Slide Time: 19:45)

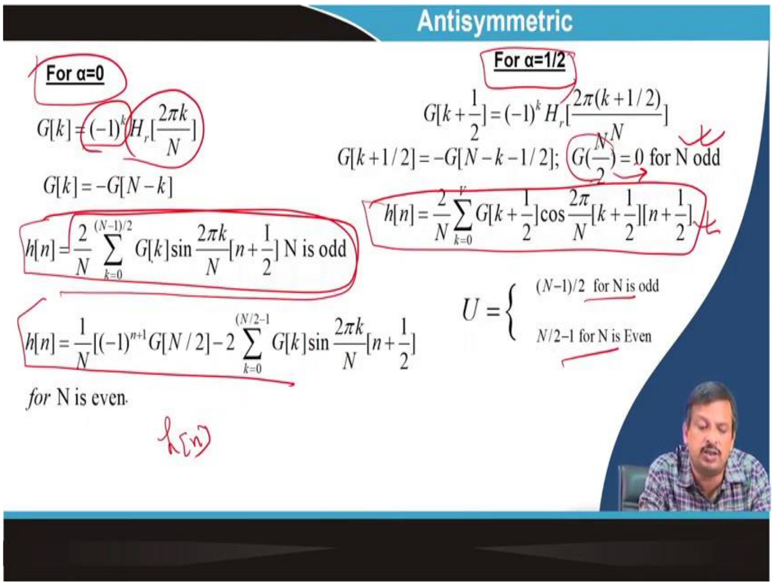


So, how do I calculate h[n]? So, there are two conditions: one is α is equal to 0, and one is α equal to half. This one is type 1, and this one is type 2 for symmetric conditions. So, symmetric means h[n] is equal to h[n] minus N minus n minus 1, which is the symmetric requirement.

So, if the symmetric requirement, I can say G k. So, α is equal to 0. So, it becomes G k minus 1 to the power k H r this one, and I can say G k is equal to minus G, and this is the symmetry. So, if you see that if H[k] is a symmetry, H[k+α] is the symmetry, then G k plus α also symmetry, the same frequency response. So, I can say G k is equal to minus G N minus k. Then I can divide that equation using the linear phase filtering methods which I described in, I think, lecture number 2.

So, if I put that equation, I get this is the impulse response of type 1 α equal to 0. So, U is equal to N minus 1 by 2 for N odd and equal to N by 2 minus 1 for N even. Similarly, for α equals half this one, and I can derive this equation from there. So, you can try to derive it in your home, try to derive this equation using mathematics, or you can directly use this equation to derive h[n] ok.

(Refer Slide Time: 21:31)



For the symmetric condition, then what is for the antisymmetric condition? If it is antisymmetric again, α equals 0 and α equals half. So, for α equal to 0, this is my equation for N odd, and this is my equation for N even. For α equal to half, this is my equation for N odd, and even for or this is the extra things I know for anti-symmetric N odd means G N by 2 is equal to 0. I am considering this one. I get this one for N odd and N even.

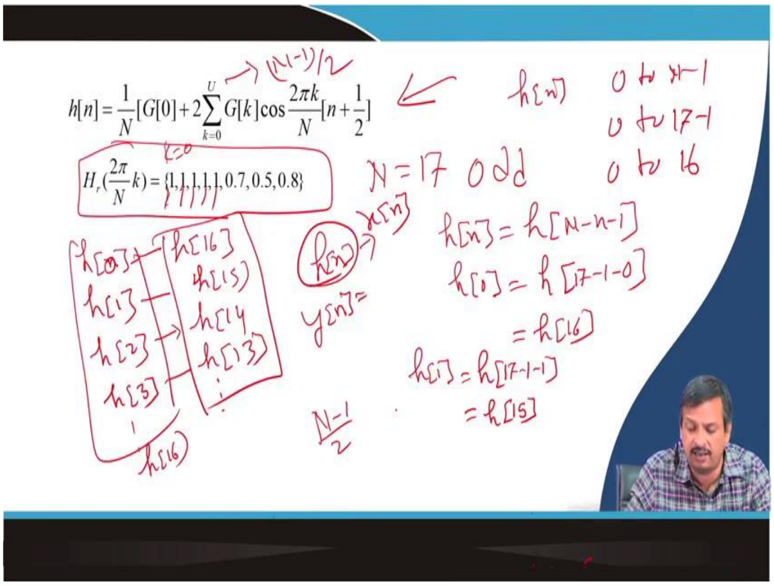
Now, let us see the problem is given like this. So, what is the problem? We determine the coefficient of linear phase FIR filter of length M is equal to 17, which is the symmetric unit impulse response; symmetric means h[n] is equal to h[n] minus n minus 1. So, I know this condition, which is condition number 1 symmetric condition. So, an α equal to 0. So, what is given? A equal to 0. So, Hr(2π /Nk) plus α, α is equal to 0. So, I can say this is nothing, but α I k is equal to is given.

So, this has been given to me, and now I have to find out h[n]. So, which condition will I apply? I will apply symmetric α equal to 0, and then I know h[n] is equal to 1 by N. So, I know how to calculate h[n]. So, I know that equation. So, here I put the equation. What is G k? G k is minus 1 to the power k into Hr(2πk /N). So, this is 2 π k by N. So, what is G k? G k is equal to minus 1 to the power k into Hr(2πN/k).

So, I can say k is equal to 0, G 0 is equal to minus 1 to the power 0 H r 2 π by N into 0, which is equal to 1. So, I can say this is 1 into 1 is equal to 1. What is G 1? I can say G 1 is equal to minus 1 to the power k, which means 1 into Hr(2π/N).

So, that is my G 1. So, I know G 0 and G 1, and then I can calculate h[n]. So, h[n], N is odd or even? N is what I gave. N is equal to 17. So, I use this equation: N is equal to odd. So, I can say U is nothing but N minus 1 divided by 2. So, I can say h[n], h[n] is equal to 1 by N G 0 plus. So, G 0 is nothing but a 1. So, I can calculate this one, where N equals 17.

(Refer Slide Time: 24:43)



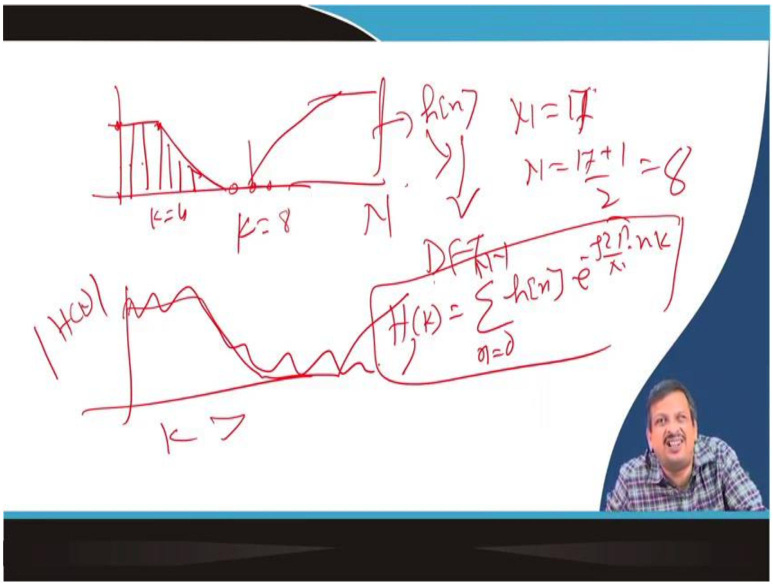
So, once I get h[n], how many h[n] is required? So, if I calculate h[n], let us take that equation this is given where N is equal to 17 odd, and I get this equation, I know U is equal to N minus 1 divided by 2. Now, I am calculating a h[n] using this equation. So, h[0], h[1], h[2], h[3]. So, how many? What is the length of h[n]? h[n] varies from 0 to N minus 1. So, it is nothing but a 0 to 17 minus 1; so, 0 to 16. So, index n up to dot dot h[16] , ok or not.

So, now I know h[n] is also symmetric. So, h[n] is equal to h[n] minus n minus 1. So, I know h of 0 is equal to h[17] minus 1 minus 0. So, which is nothing, but a h[16] . So, I do not have to compute h[16] , I know if I compute h of 0 which is equal to h[16] .

Similarly, h[1] is equal to h[17] minus 1 minus 1. So, which is nothing, but a h of minus 2 h[15] . So, h of 0 is equal to h[16] , h[1] is equal to h[15] , h[2] is equal to h[14] , h of 3 is equal to h[13] ; so, dot dot dot. So, I do not require to compute all N minus 1 number of h, I only have to compute N minus 1 by 2 number of h, is it clear?

So, I do not have to compute all the h n; I do not require it. So, I compute this h[n] and then repeat it on that side. I know this is a symmetric filter compute that h[n] is given. So, once I know the h[n], I can compute y[n]; y[n] is nothing but a convolution of h[n] with x[n]. If I want to know, after design, what is the frequency response of the filter? So, I have given that filter if you see k equal to 0 to 1, 0, 1, 2, 3, 4. So, for k equal to 4, I have given let us I take another slide here.

(Refer Slide Time: 27:33)

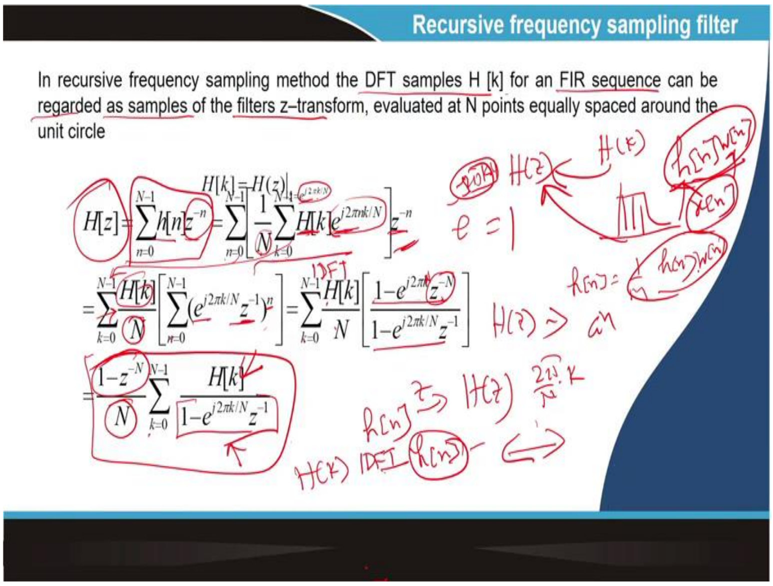


So, up to k equal to 4, it is 1. So, what kind of filter have I given? I have given this side is k. So, up to k equal to 0, it is 1, 1, 0, 1, 2, 3, 4 up to k equal to 4, it is 1, and then it is reduced, reduce, reduce like that kind of thing I have given, after that it is 0, after k equal to 8 N by 2. So, N by 2 means N plus 1 by 2. So, it is k equal to so, k equal if I say N equal to 17.

So, what is the symmetry? N is equal to 17 plus 1 divided by 2 is equal to 8. So, at k equal to 8, it will be symmetric; this is N. So, I can do that. So, I know that, and then I calculate h[n]. So, once I do the h[n], then if I take the discrete Fourier transform of h[n], if I compute H(k) using n equal to 0 to N minus 1 h[n] $e^{-j(2\pi/N)nk}$ using this formula, if I compute H[k].

Then, if I plot mod of H[k] on this side and this side is k, I get back this same or not. I will not exactly get back the same because it will introduce some ripple because of the finiteness of the impulse response. So, I will get it like this: you can test it, take an example, do it, and test it. To reduce the complexity, you can take N, which is very small; then there is another method, which I said is a non-recursive implementation using frequency sampling.

(Refer Slide Time: 29:32)



Now, if I go for the recursive frequency sampling filter, in this recursive method, what we do is method the DFT sample H[k] for FIR sequence can be regarded as a sample of the filter z transform. So, what is given? So, let us have a filter z transform H(z) is given, or I can say that from given H(k), I want to find out what should be the possible H(z).

Once I know the H(z), I can easily implement it using discrete structure 1 or 2 methods. So, what I want to know for a given H[k] given frequency response, I sampled it, then from the sampled frequency response, I tried to find out H(z) that is my problem. So, what is H(z)?

H(z) is nothing but a z transform of the impulse response h[n]. So, if h[n] is the impulse response of the filter and that z transform is nothing, but if I take the z transform, I get H(z). So, $H[z] = \sum_{n=0}^{N-1} h[n] z^{-n}$. Now, what is h[n]? h[n] is nothing but an inverse Fourier transform of H(k), ok or not?

So, here, z is equal to what? j e to the power π by. So, what is the sampling frequency? 2 π by N into k is the sampling frequency in the frequency domain. So, it is nothing, but a 1 by N k equal to 0 to N minus 1 H(k) e to the power this one inverse Fourier transform, h[n] is the inverse Fourier transform in h[n] is the inverse, if I know H(k), then if I to IDFT Inverse Discrete Fourier Transform, I get h[n].

So, if this is the inverse IDFT of h[n] into z minus n, now if you see there is a two sum n equal to 0 to N minus 1 k equal to 0 to N minus 1. If you check the two sums, then N is constant. So, I take N capital N, which does not depend on small n. So, I can say that I can take this sum first, and I can capital N is outside, and H[k] does not also depend on N. So, I make it outside inside the sum of k, and then the sum of N depends on this one and this one. So, this one and this one I write whole to the power n.

So, a to the power n is nothing, but a 1 by a to the 1 minus a to the power n divided by this one ok. Now, if I see $e^{-j2\pi k}$, there will be I think there will be an N, N also here. So, 2 π k N is nothing, but a tell me, 1; integer all are integer. So, it is nothing but a 1 j cos theta minus plus j sin theta; so, all are 1. So, I can say it is 1. So, I can put this as 1. So, this is nothing but a 1 minus z^{-n}.

So, I can say 1 minus z^{-n} does not depend on k. So, I put it outside, N does not depend on k put it outside, then I said

$$H(z) = \sum_{k=0}^{N-1} H[k] \frac{1}{N} \sum_{n=0}^{N-1} \left(e^{j\frac{2\pi}{N}k} z^{-1}\right)^n$$

. Once I get my H(z), I can implement it using a discrete structure 1 or 2; it does not matter. So, this way, I can compute H(z).

So, this is called a recursive frequency sampling filter; one is non-recursive type 1, type 2, I can implement h n, here is recursive, is it clear? So this way, I can implement frequency sampling methods for the FIR filter. So, first, I will explain what I explained. Window methods using an algebraic function I have calculated h[n], I can multiply with w n, and then I can convolve with x[n] to get the filter output. So, I can write a program for h[n], then I can multiply w n with the h[n], and then I can convolve with x[n] to get the y n.

Similarly, in frequency sampling methods, whether I use non-recursive or recursive, I have to find out H(z) and realize H(z) using a discrete structure 1 or discrete structure 2. Now, if I go for non-recursive type 0 or type 1, I can implement type 0. I can implement type 1. Type 0 means α equal to 0, and type 1 is α equal to 0 and type 2 is α equal to half.

So, whatever I want, I can implement; I can kind of find out the h[n] once I get that h[n], I directly convolved with x[n]. So, windowing is not here; we are not using windowing in the frequency sampling method. But, in window methods, we calculate h[n] from that algebraic expression, then multiply by w n and then convolve it with x[n].

So, what I suggest is that you implement it for the windowing method and the frequency sampling method at least once so you understand how to implement the FIR filter. If we just write a MATLAB program FIR filter FIR and then give the parameter, you know what parameters to use. also, once you know the filter design, at least you know what parameters you have to provide for the FIR filter design, ok?

Next week, what we will do about the optimization method is let us say I have ripple is given, passband ripple is given, stopband ripple is given, and transition bandwidth is given. So, how do I design optimal methods, which I will cover in the next week? So, this week, what did I say? I have said the windowing methods, frequency sampling methods, and what are the filter specifications.

So, summarization is that this week we learnt how to specify a FIR filter or how to specify a filter, what is linear phase filter and how to implement the linear phase filter. Then I talked about the FIR filter. So, how do we implement an FIR filter using windowing methods and frequency sampling methods?

In windowing methods, I know that the stop band ripples depend on the window which window I should use and the transition bandwidth; if I want to increase or decrease the stop band ripple, then I have to increase the transition bandwidth.

So, optimization is based on which window I am using, then I can use the frequency sampling methods, and I can find out what the transition bandwidth is coming out and that frequency. Then, what do I want to know? What is the optimized value of order so that it will restricted with stop band ripple, pass band ripple and transition bandwidth? So, optimum FIR filter design, I will cover in the next week, ok.

Thank you.