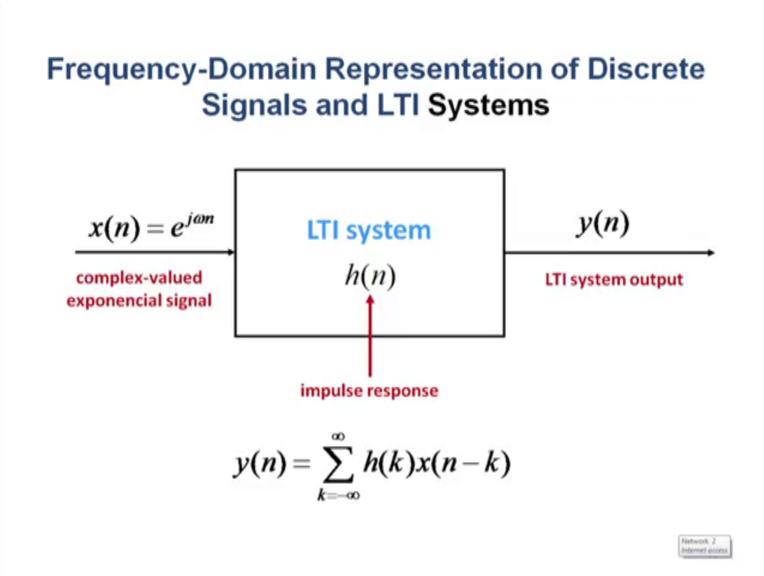
Digital Speech Processing
Prof. S. K. Das Mandal
Centre for Educational Technology
Indian Institute of Technology, Kharagpur

Lecture – 04
Review of DSP Concepts (Contd.)

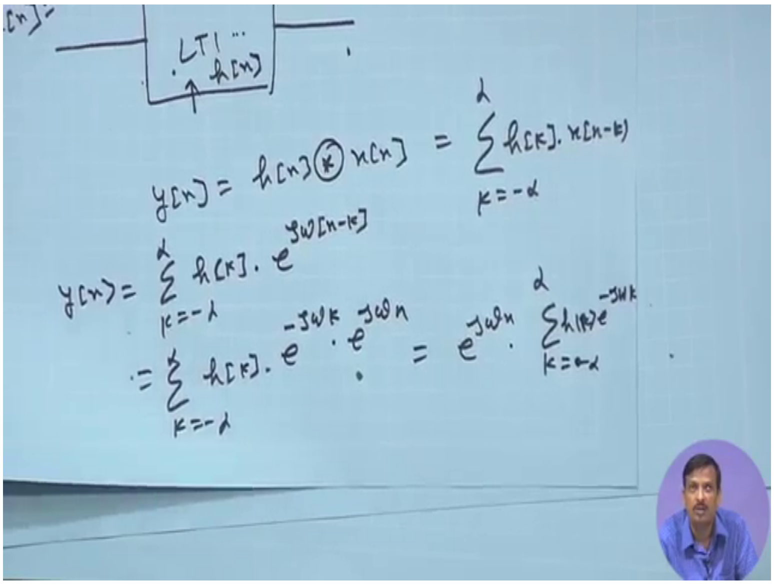
So we are discussing about that LTI systems and their implementation.

(Refer Slide Time: 00:22)



Now, come to the frequency domain representation of the discrete signals and a and LTI systems ok.

(Refer Slide Time: 00:33)

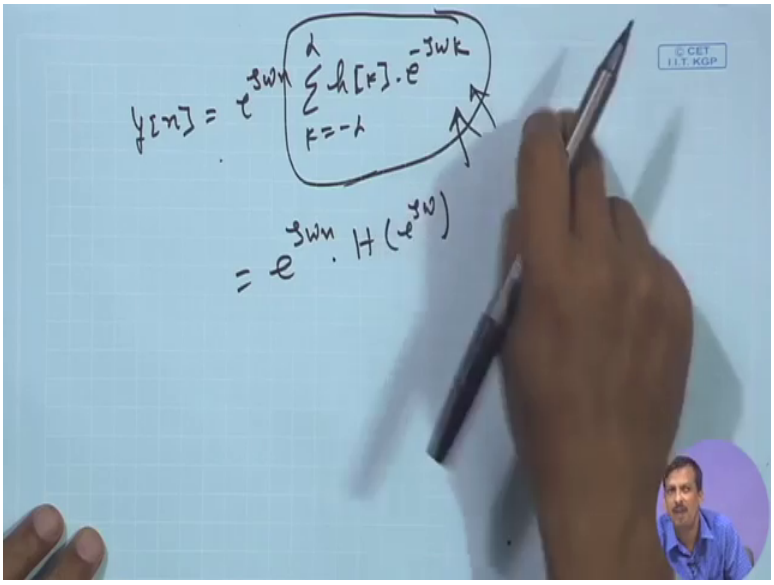


So, what is there? Suppose I have an LTI system in here, LTI system which is h n, h n let us. Now I want to provide the input and I get an output y n, if I provide a input x n. Now I want to know frequency response of this h n. So, mathematically I can do many things, but so what is what is y n y n is nothing but a h n convolve with x of n, which is nothing but a k equal to minus infinity to infinity h k into x of n minus k x of n minus k, you are or not. Now let us consider I want to find out the frequency response.

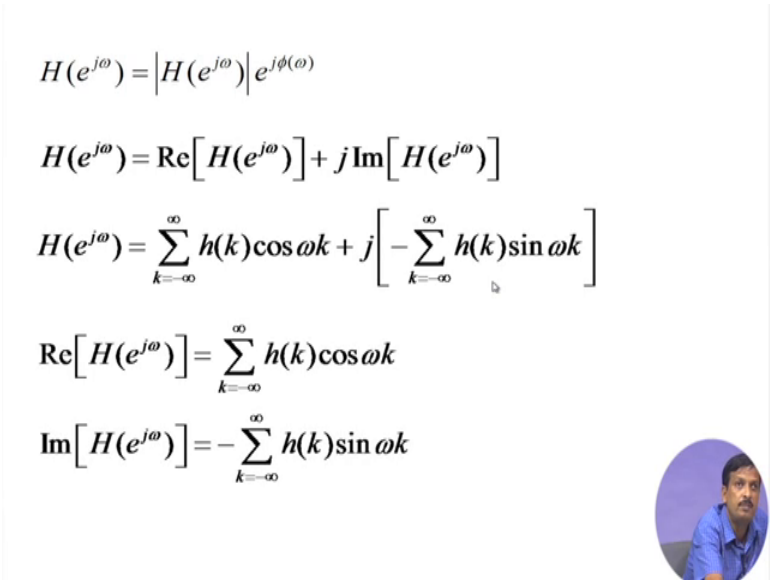
So, if I x n at the system, with an individual frequency each and every frequency let us say 1 hertz, 2 hertz, 3 hertz, 4 hertz, 5 hertz, 6 hertz, 7 hertz and then I try to find out the output, for 1 hertz 2 hertz 3 hertz then I get the frequency response of the system. So, instead of x n let us say I input the single sinusoidal. So, it is e to the power j omega. N j omega n I can instead of So, I apply a e to the power j omega n; while omega is distinct let us for 1 hertz. Then I change the omega for 2 hertz. Then I change the omega for 3 hertz. So, I can do that way. And n is the number of sample.

So, I apply a e to the power j omega n, what should be the output? Then y n is nothing but a k equal to minus infinity to infinity h k let us h k into x of n minus k is replaced by e to the power j omega. So, e to the power j omega n minus k. So, which is nothing but a k equal to minus infinity to infinity h k e to the power minus j omega k into e to the power j omega n or not. So, if I write down e to the power j omega n is nothing but a k equal to because this is not (Refer Time: 03:12) k. So, e to the power j omega n outside minus infinity to infinity, I can write down h k into e to the power minus j omega k. So, what is e to the power minus j omega k? What is e to the power h k?

(Refer Slide Time: 03:40)



(Refer Slide Time: 04:50)



So, I can say the y n is nothing but a e to the power j omega n at where k equal to minus infinity to infinity h k into e to the power minus j omega k. So, this term is nothing but the frequency response of the systems, frequency response of the systems. So, I can write down this term as a e to the power j omega n h of capital h, frequency plain e to the power j omega. Frequency response of that LTI system.

So, if you see e to the power j omega is the frequency response of the said LTI systems. Now I can say if I see this equation if you see this equation what is there? This is a complex number. So, there has a magnitude and there has a phase. So, I can say h e to the power j omega has an complex. So, it has it has an phase or you can say the magnitude, which is nothing but a h e to the power j omega mod and e to the power j phi omega is the phase. So, I can say h is the real part and imaginary part. Again this is complex number property. So, cos omega j sin omega k. So, I can write real part is nothing but a cos part, and imaginary part is nothing but a sin part, minus h k sin omega k.

(Refer Slide Time: 05:31)

Magnitude response:

$$|H(e^{j\omega})| = \sqrt{\mathrm{Re}\left[H(e^{j\omega})\right]^2 + \mathrm{Im}\left[H(e^{j\omega})\right]^2}$$

Phase response:

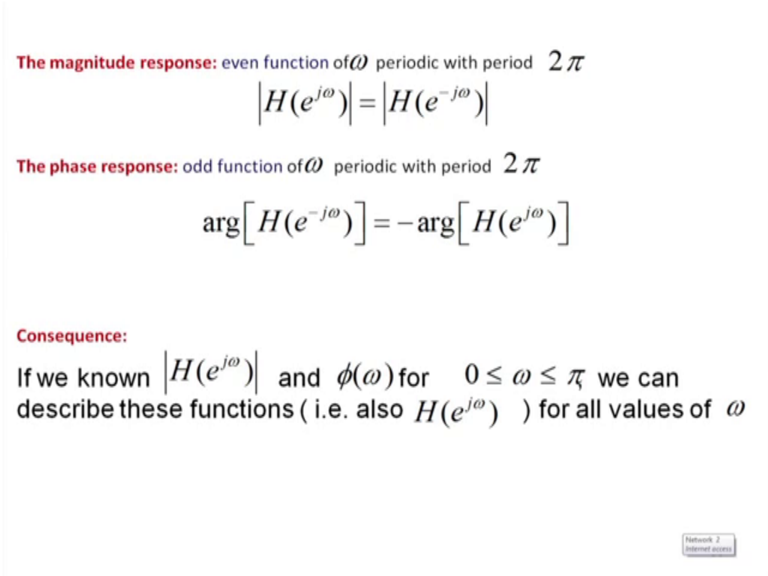
$$\phi(\omega) = \arg\left[H(e^{j\omega})\right] = arctg\,\frac{\mathrm{Im}\left[H(e^{j\omega})\right]}{\mathrm{Re}\left[H(e^{j\omega})\right]}$$

Group delay function:

$$\tau(\omega) = -\frac{d\phi(\omega)}{d\omega}$$

So, magnitude response is nothing but a complex number, real square plus imaginary square phase is nothing but a tan inverse or arc tan inverse imaginary by real part. And group delay function is nothing but a tau omega is equal to d phi omega by d omega. Those things will be used frequently in signal processing of speed signal processing.

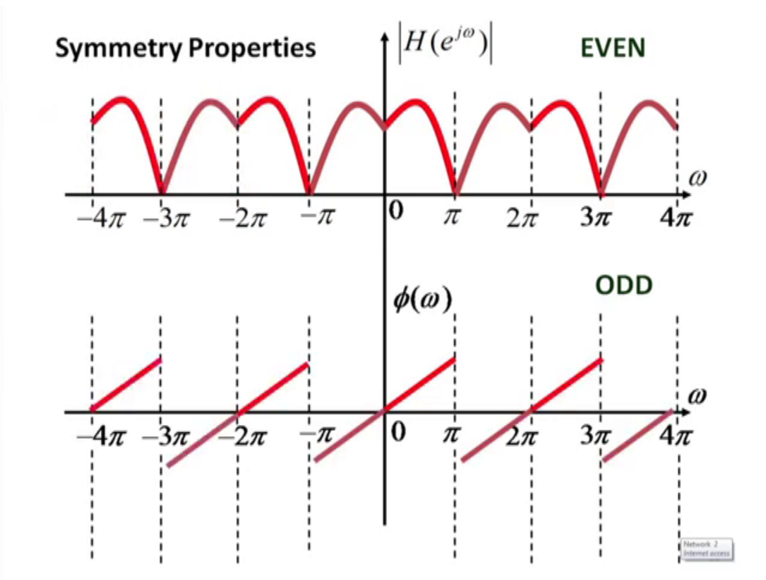
(Refer Slide Time: 06:04)



Then I am not going this property you can find out phase component is nothing but a odd function and magnitude response is nothing but a even function ok.

So, this is the LTI systems signals with the DSP concept will be used later on.

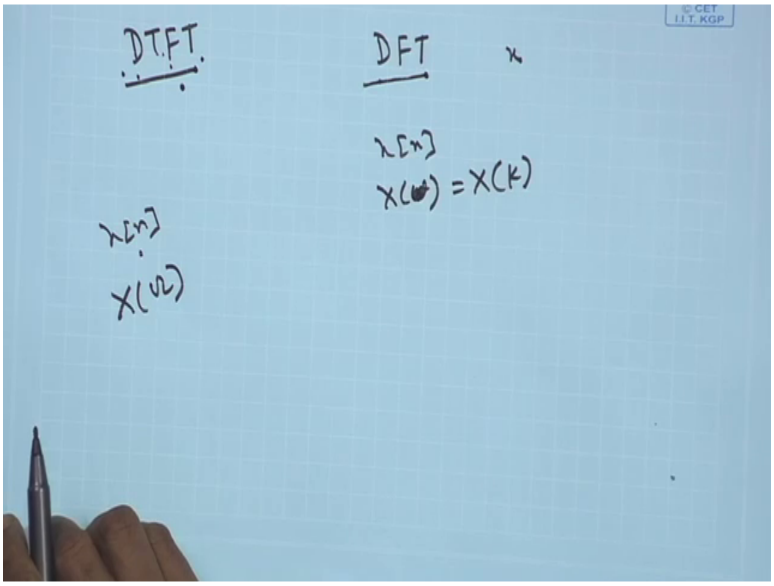
(Refer Slide Time: 06:23)



There is a another important DSP concept is there that is will be used I am not this explaining this curve, that is called discrete Fourier transform. I think many of you know that, one is called discrete time Fourier transform DTFT.

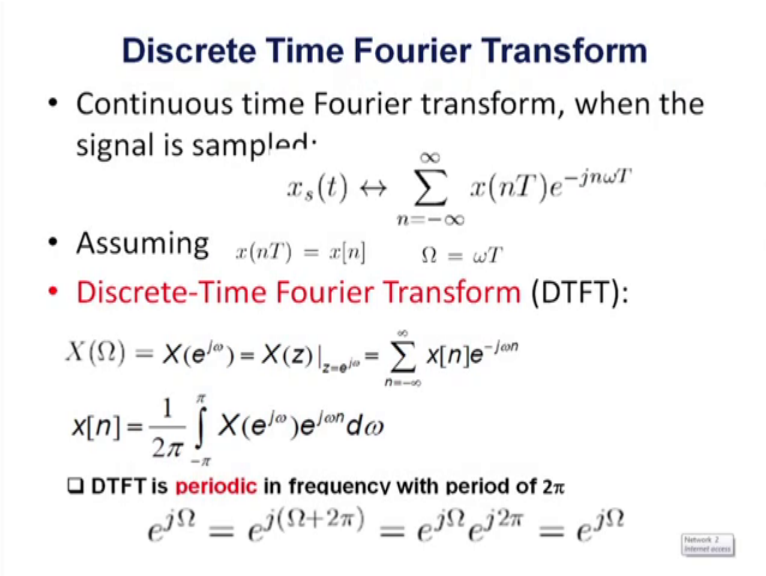
(Refer Slide Time: 06:33)

(Refer Slide Time: 06:39)



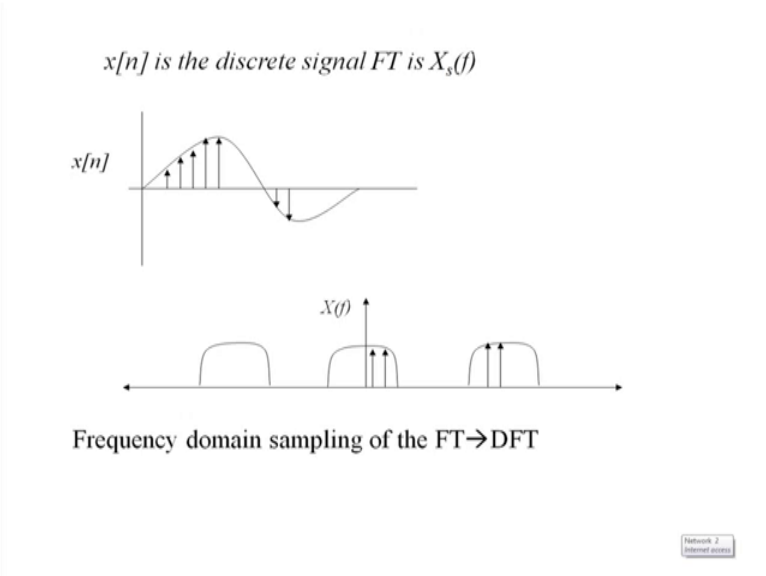
Discrete time Fourier transform, another is called DFT discrete Fourier transform. Here the time domain signal is discrete, but frequency domain it is continuous. Time domain signal is discrete, but frequency domain it is continuous. So, discrete time Fourier transform, time domain it is discrete, but here discrete Fourier transform, input is discrete and frequency domain also it is represented as a discrete ok.

(Refer Slide Time: 07:19)



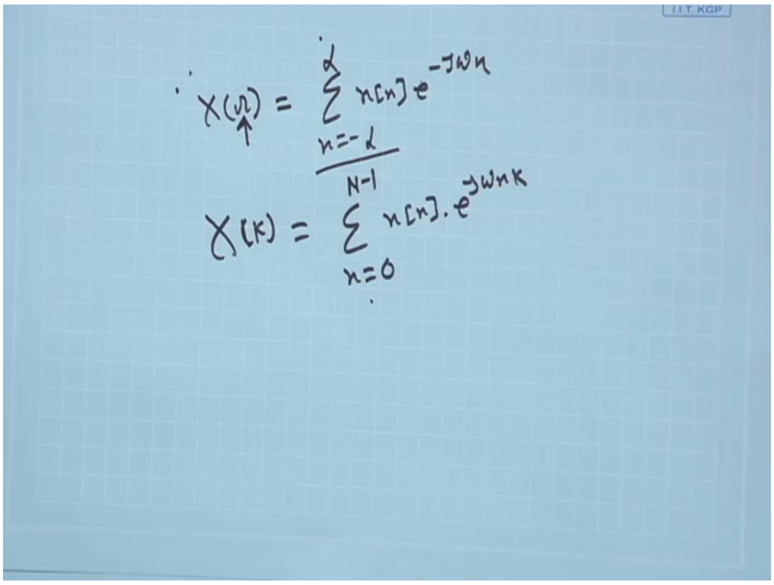
So, this is a DTFT and DFT. So, DTFT x n is in DTFT x n is digital, but what about I get x omega is continuous, but here x n is also digital and x omega is also digital. Some time you write it x k discrete frequency instead of omega. So, what is the concept behind this? I am not going this one this DFT, I will describe in the DFT.

(Refer Slide Time: 07:56)



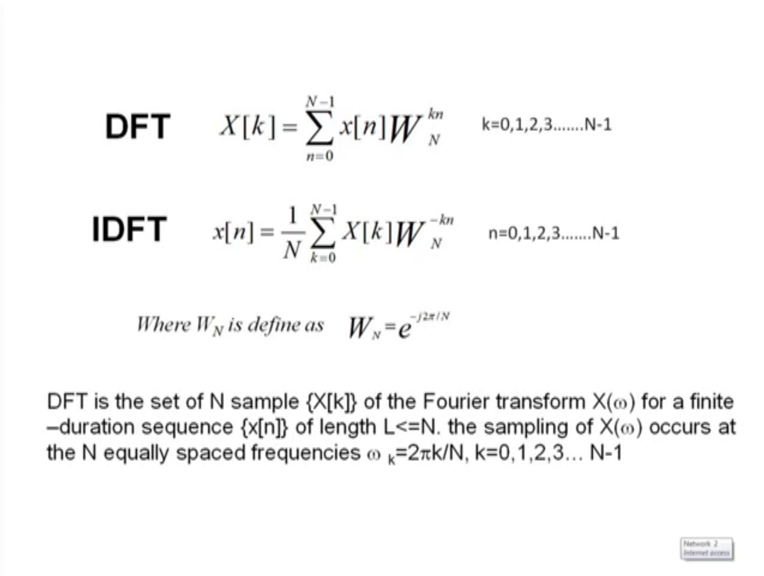
So, what I will do discrete Fourier transform. So, discrete time first you just know the equation of discrete time Fourier transform. So, I can say x of omega where the frequency is continuous is nothing but a n equal to minus infinity to infinity, x n e to the power minus j omega n ok.

(Refer Slide Time: 08:12)



Now, I said the frequency domain it also discrete, that is called DFT discrete time Fourier and discrete Fourier transform. So, it is not discrete time Fourier transform it is discrete Fourier transform. So, this omega is replaced by k, k is the discrete Fourier transform. So, n equal to then becomes 0 to n minus 1 x of n e to the power j omega n k.

(Refer Slide Time: 08:54)

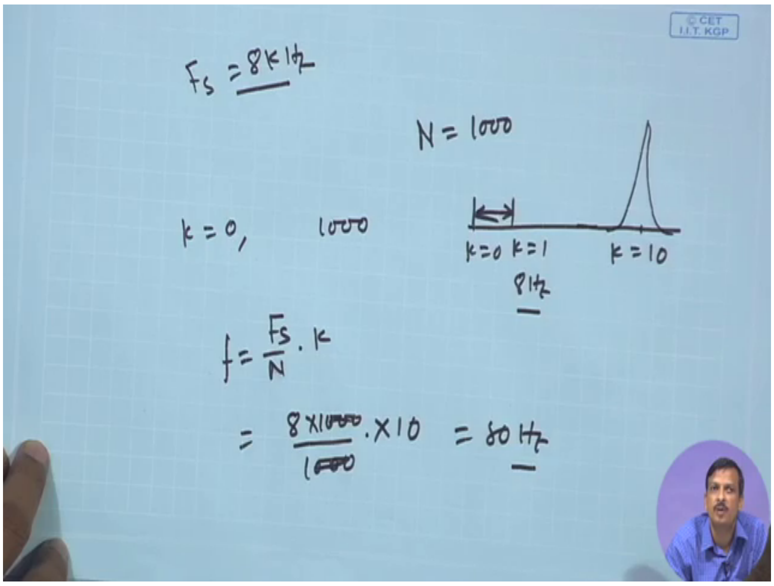


Why it is it is minus infinity to plus infinity, it is 0 to n minus 1 why? I said discrete frequency the frequency axis also discrete. And if I say that the frequency; that means, I am sampling the frequency scale also. So, suppose I have a frequency range this one, here to here is f, I want to discretize it. So, I am saying let us divided this length with a n number of discrete value.

So, I divided this n, n number of discrete value if it is n number of finite length discrete value. I can say then it is a the period is after n I am saying the signal is repeating itself. So, x k is a period of n capital n the number of sample I have taken I have the number of discrete sample the discrete sample I have made the frequency response. So, this is the if you explain it like this way. So, this is called. So, this is continuous frequency I discretized it with a number of discrete value. So, if it is discrete DFT then equation is x k is equal to n equal to 0 to n minus 1 x n e to the power j omega n k ok.

So, if it is that then this n is called DFT length of the DFT, length of the DFT.

(Refer Slide Time: 11:02)

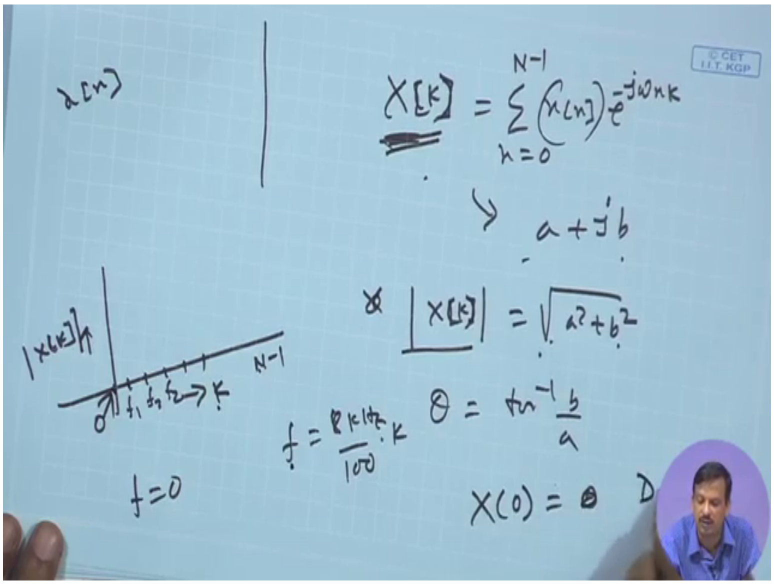


Now, suppose I have a signal with sampling frequency F s is 8 kilo hertz, sampling frequency is 8 kilo hertz. Now I said I take n is equal to let us 1000, 1000. I divided whole frequency range is the 8 kilo hertz, 8 kilo hertz frequency range. I divided in 1000 number of pieces equal amount of pieces. So, I can say each pieces. So, k is varies from 0 to 1000 0 1 2 3 4 6 4 5 6 8 up to 1000. So, I can say that whole 8 kilo hertz space is divided in 1000 pieces; that means, every sample k value, k equal to 0 k equal to 1 means it is separated by 8 100 hertz. So, the value distance between the 2 k value is 8 hertz this is called frequency resolution.

So, frequency resolution is nothing but a F s by N. So, suppose I want to find out the what is the frequency analogue frequency value at k equal to10. So, t is nothing but a into k is equal to f. So, at k equal to 10 in this case F s is 8 kilo hertz 8 into 1000 divided by 1000 into 10 for 80 hertz. So, my signal is contained 80 hertz component then the power at here will become here ok.

So, once I want to draw the draw a DFT, suppose I have drawn a DFT. So, I can say DFT once I done the DFT, I have a x k ok.

(Refer Slide Time: 13:06)

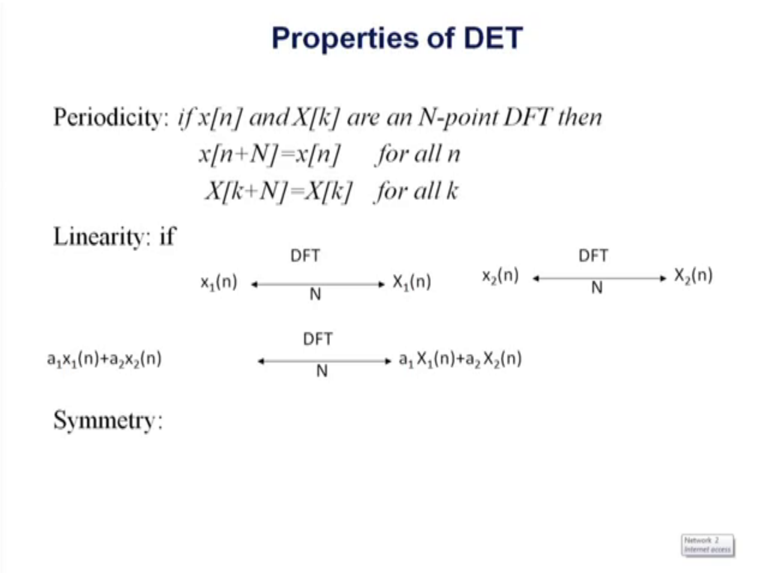


So, x k is nothing but a n equal to 0 to n minus 1 where n is the DFT length x of n input signal e to the power minus j omega n k. Now x k is a complex signal, s k is a complex signal. So, a complex signal has an amplitude and has an phase. So, I can write this signal is nothing but a cos and sin a plus j b, a plus j b. So, if it a plus j b then I can say x k has an magnitude this ones, which is defined like. So, discrete signal is third bracket always represented by a third bracket. Slide may be it is first bracket, but discrete signal is always represented by third bracket. So, mod of x k is nothing but a root over of a square plus b square, that is called magnitude response. So, what is a phase response, theta is nothing but a or arc of theta arc of x k x k theta is nothing but a tan inverse b by a, now suppose I told you I have a signal x n find out the magnitude frequency response, frequency magnitude response. So, find out the magnitude frequency response of x. So, I have to plot the mod of x k. So, after computing DFT I get a plus j b from. So, I compute the a square plus b square root over that is the amplitude if it is voltage then it is this has to be squared to make it power if it is power then it is root over of a square plus b square.

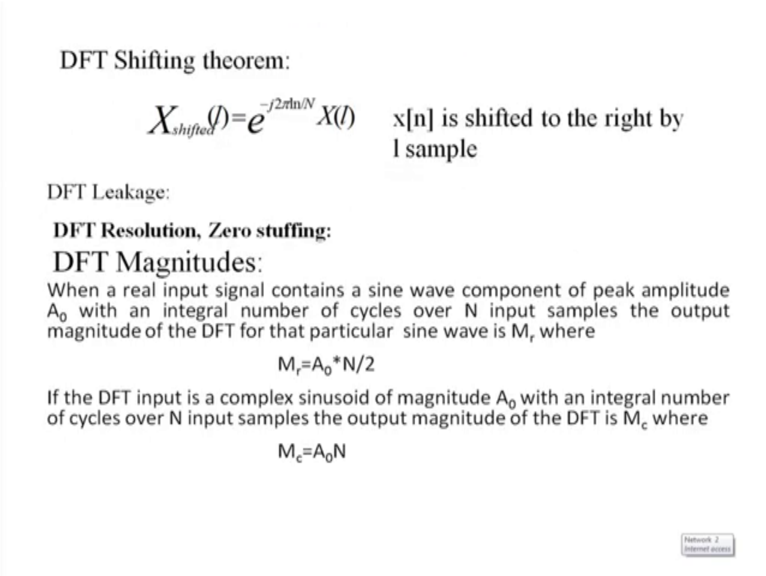
So now if I want to draw it this axis is k and this axis is the power mod of x k. So, k is nothing but the k is varies from 0 to n minus 1. Now I told you I do not want the discrete frequency k, can you make it with a axis of f instead of k can I replace this axis by f very easy because f is equal to nothing but a if this signal is sample this x n is sampled at this 8 kilo hertz and my n is equal to 1000, then I can say 8 k divided by 1000 into k I get the corresponding frequency.

So, instead of 0 I can add 0 this will be next frequency f 1 f 2 f 3 f 4 for different k value I get the analogue frequency value. Now if you see here f is equal to 0. So, x 0 is nothing but a frequency is equal to 0 that is nothing but a DC component of the signal. So, x 0 is a DC component of the signal. So, those properties you have to know frequency resolution DC component, how do you make the k to f conversion? Those things will be used in implementation of digital this digital speed processing. Now there is another property I have not describing this one again.

(Refer Slide Time: 16:43)

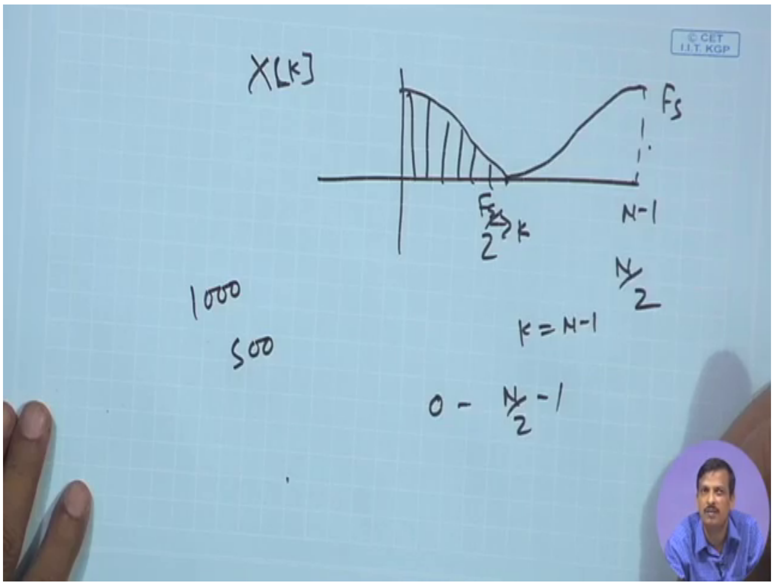


(Refer Slide Time: 16:44)



So, the symmetry property is there. So, if I say DFT is called symmetry d after DFT x k is symmetry s k is; obviously, the symmetry property ok.

(Refer Slide Time: 16:53)

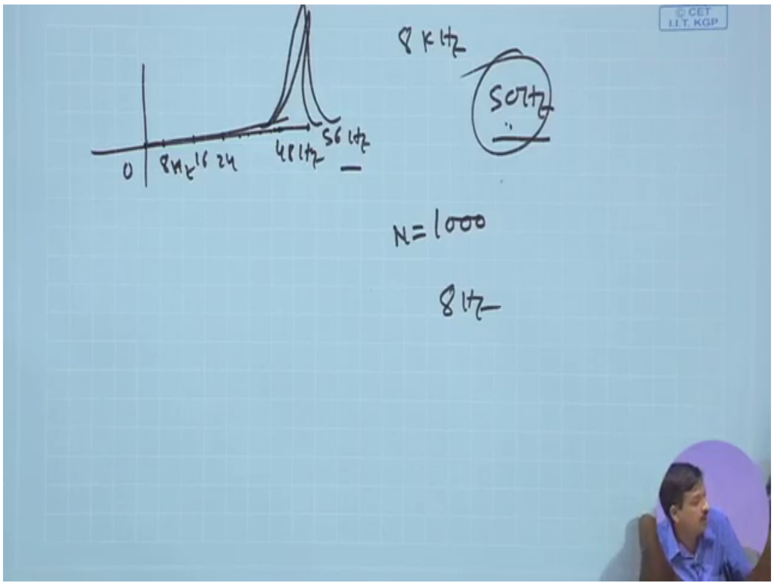


That means if I draw it here x k let us this is my k axis k. So, at here n minus 1, it is nothing but F s. At k equal to n minus 1 is nothing but a F s ok.

So, F s by 2 is my signal limitation. So, I can say it repeat the property this is the F s by 2 point or n by 2 point. So, 0 to n by N by 2 minus 1 give me the same this information and this information will be same information for base band signal. So, after analyse the frequency if I use 1000 point DFT 500 point property will be repeated in here in mirror in nature symmetric like.

So, I can take this ne only to find out the frequency response of the signal then DFT magnitude, when the real input signal contain a sin wave component a peak amplitude a 0 the integer number of cycle n, then the cycle of over a input sample then it is nothing but a M r equal to a 0 into. So, amplitude of the signal will be magnified by N by 2 if it is real valued signal if it is a complex signal it will be n number of. So, when you do the inverse DFT I should normalise the amplitude by if it is real input signal. So, it is n by 2 if it is a complex input signal then it will be M. And there is a DFT leakage all those things you can find in that DSP book DSP leakage means, it is 2 kind of things if you see.

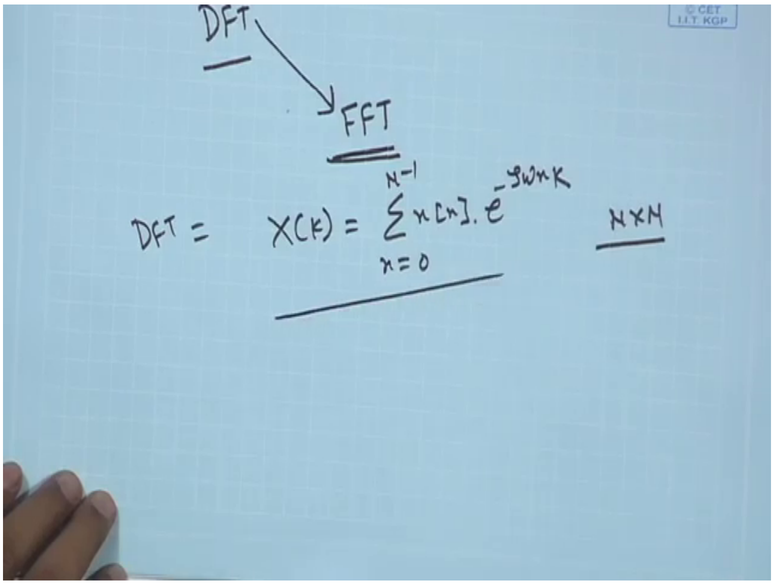
(Refer Slide Time: 18:51)



Suppose I have sampled the signal in 8 kilo hertz, and of a signal the let us the signal as 50 hertz. So, 50 hertz signal sinusoidal signal has sampled at 8 kilo hertz. So, if I take the 1000 n equal to 1000 sample, then the frequency resolution is 8 hertz. So, for every k I will get 0 then 8 hertz, then 16 hertz, then 24 hertz, then let us dot, dot, dot then 40, 48 hertz and then next one is 56 hertz.

So, my resolution does not support the 50 hertz. So, the power, but the signal has only power at 50 hertz. So, if I make the DFT. So, instead of looking like 50 hertz single spike it will be looked like this. Power will be distributed adjacent computation point. So, this will be look like this. So, this is the DSP computational leakage. Then there will be another effect windowing effect will also become, I will discuss later on when I design the filter. Now you know that implementation of DFT discrete Fourier transform is done based on fast Fourier transform algorithm.

(Refer Slide Time: 20:24)

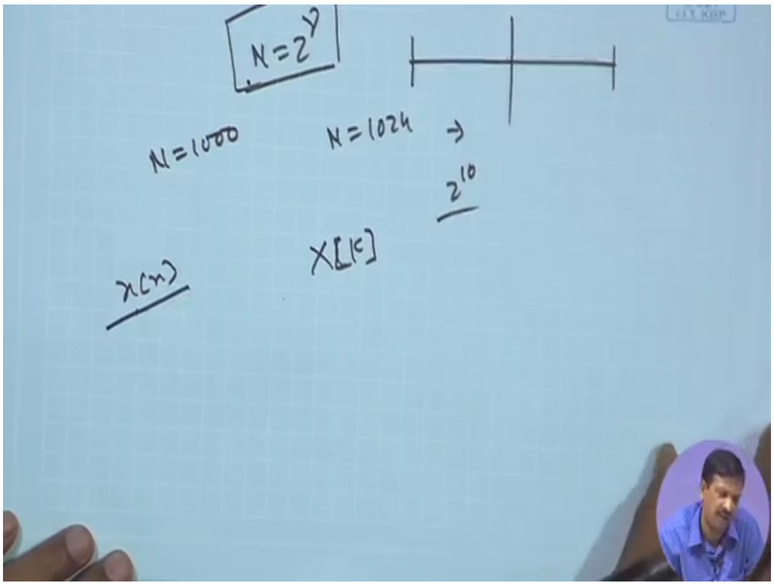


So, FFT is not the transform. FFT is an algorithm to implement DFT. That is why it is called fast Fourier transform algorithm, fast Fourier transform FFT. So, FFT is an implementation procedure for DFT. So, if I say that DFT is nothing but a x of k is nothing but a sum of n equal to 0 to n minus 1 x of n into e to the power minus j omega n k. So, how many number of complex multiplication is in here n cross n? N cross n complex multiplication is there.

Now I want to reduce it I want to reduce this complex multiplication. So, that it can compute very faster. So, how do can I do it that is why it is called fast Fourier transform. So, instead of n cross n can I use some other method which can be computationally which can reduce the computational cost of the DFT. One method is called fast Fourier transform.

I will discuss only the radix 2 algorithm other you can study from the DSP book. So, radix I will just discuss the philosophy this radix 2 algorithm is most commonly used algorithm. So, if I say I want to reduce the n cross n I asked from the computational background that suppose I have a search space if I make it binary search, then it is log 2 n so; that means, if I instead of computing using the whole signal, if I dissimulate the if I if I somewhere I can dissimulating the signal in some you can small chunk, and then if I compute is it help us yes.

(Refer Slide Time: 22:50)

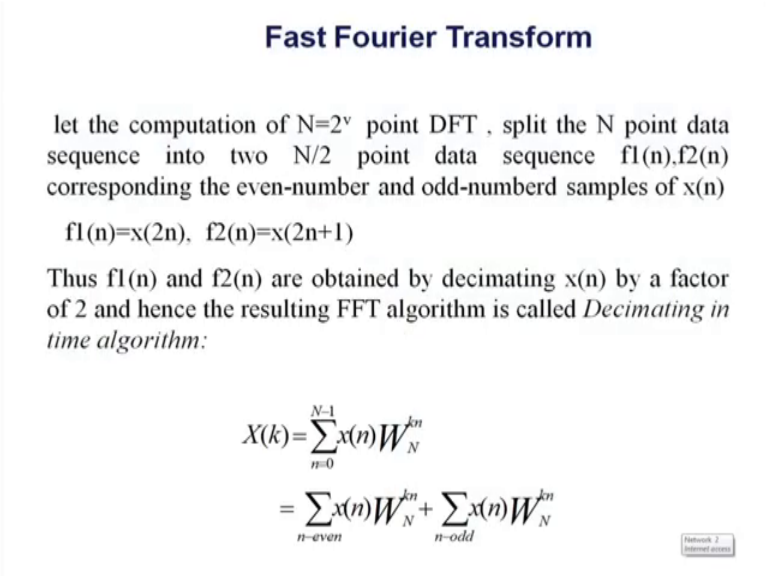


So, what they have done if FFT radix 2 algorithm, that if I take the length of the DFT is you can say that if the a n can be represented by 2 to the power something, the length of the DFT restriction is that I want to I want to divide the whole signal in binary space in 2 2 space of 2 part then 4 part 5 part like that. And so, like the binary search whole number divided middle first part second part. And then I use the first part then I use the second part like that way I want to do it.

So, what I want that length of the DFT must be expressed by this way 2 to the power something. So, I cannot take n equal to 1000 because it cannot be expressed in 2 to the power something away. So, if I take n is equal to 1 0 2 4 then I can say it is nothing but a 2 to the power 10. This is the restriction radix 2 algorithm that is why it is called radix 2 algorithm.

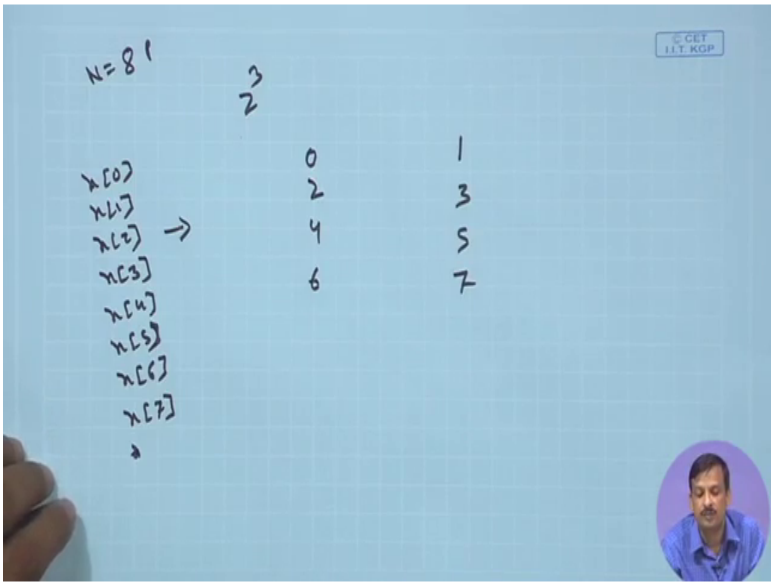
Then whole signal x n either can be broken in time domain or I can use whole x n and both the computational output this you know which is frequency domain output in small number of chunk. So, either x n can be divide or I can divide x k also. 2 space I can divide the signal. So, let us start with the disseminating time algorithm.

(Refer Slide Time: 24:17)



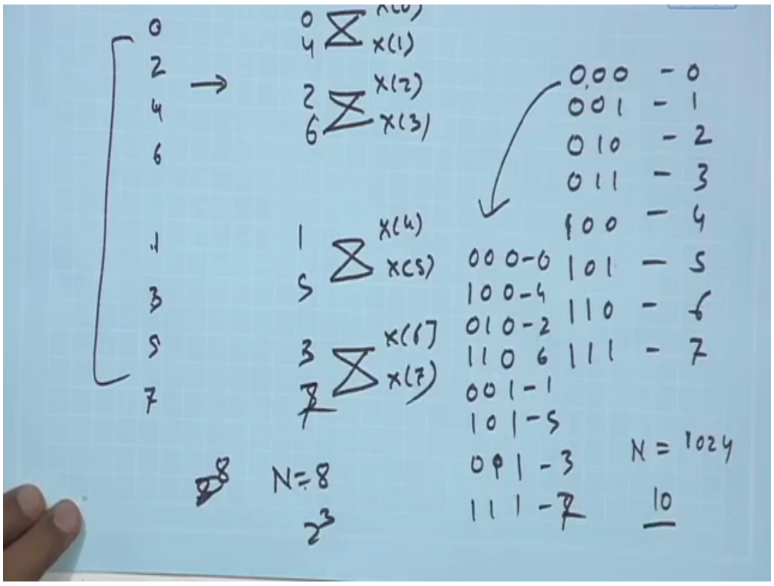
So, here what I do instead of computing whole DFT at a time I divide the input signal into 2 term, one is even term another is odd term. Then again this one is divided even and odd and even and odd until unless I reach 2 point, where I cannot divide one even and one odd. So, this algorithm is described in a book you can do it I just explain you by a let us take the 8 point DFT 8 point DFT n equal to 8.

(Refer Slide Time: 24:54)



So that means, it can be expressed 2 to the power 3. So, 3 stage I can divide it. So, first is let us the signal is x 0 x 1, x 2, x 3, x 4, x 5, x 6, x 7 and x 7. So, 8, 8 number of signal I have taken n equal to 8. I divide this signal space in 2 space. One is odd and even let us even. So, 0 2 4 6 1 3 5 7. So, I can say, let us start the signal instead of taking the signal 1 2 3 6.

(Refer Slide Time: 25:55)



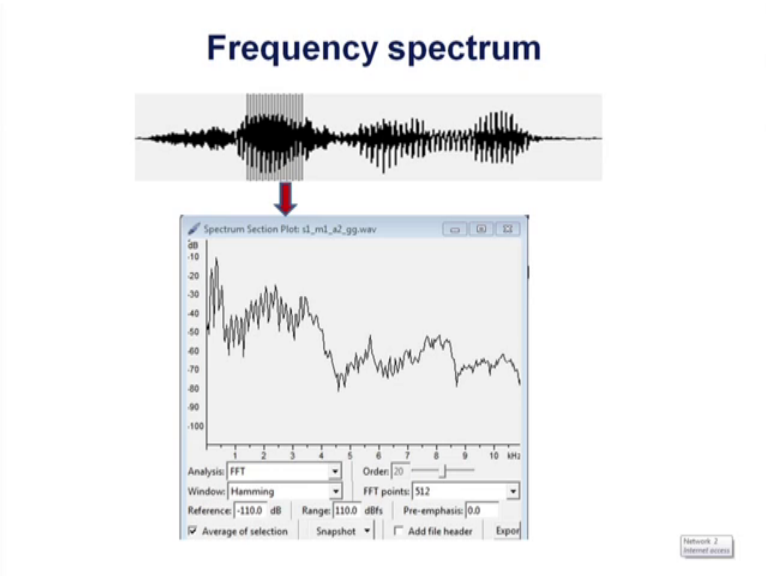
So, I take the upper part let us 0 2 4 6, and then I take 1 3 5 7. Then again this can be divided in 2 part, even 0 4 and odd 2 6 again it is even 1 5 odd 3 5.

So, once I get 2 signal I compute the DFT. This is called butterfly algorithm I have not detailed discussed the mathematics of butterfly algorithm. Now if you see I get the output x 0, x 1, x 2, x 3, x 4, x 5, x 6, x 7. Now you may say how do I divide this signal in odd and even if it is my signal size is 1 0 2 4. There is a trick if you know that the that trick I will discuss that is called bit reversible, that technique.

If you see 8 8 n equal to 8 then 0 0 0 my represent 0 0 0 1 represent 1 0 1 0 2 0 1 1 3 1 0 0 4 1 0 1 5 1 1 0 6 1 1 1 7. So, 2 to the power 8 sorry, n equal to 8. I required 3 bit, this 2 to the power 3 3 bit I required. So, I can see the 3 bit I can generate 7 number now to do this odd and even distribution. What I do? That excess the signal in bit reversible format. When you excess the signal the index term reverse the bit. If it is 10 bit reverses the bit. So, what I will do I reverse the bit. So, instead of 0 0 0 I write 0 0 0 because reversible and non reversible same 0 0 1 I write 1 0 0 0 1 0 I write 0 1 0 same 0 1 1 1 0 1 0 0 0 0 1 1 0 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1 1, if you see 0 4 2 6 1 5 3 3 and sorry, 1 5 3 7.

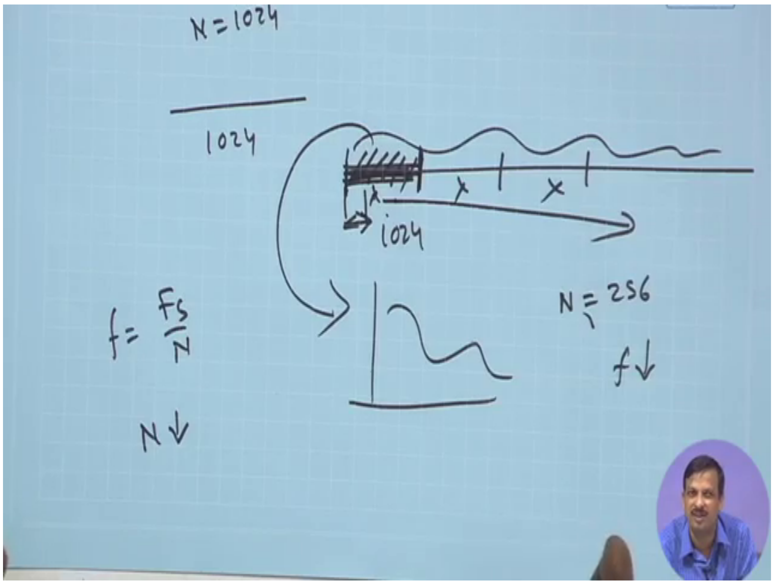
So, this arrangement is done automatically if I just access the signal the index of the signal it change in bit reversible mode. So, if it is n equal to let us 1 0 2 4. So, my I required 10 bit to represent the index. When I access the signal I will change the bit reverse I get it and then I do the calculation whatever is required for butterfly algorithm. So, that I have explained in DFT n I am not explaining details because these can be explained details in DSP class you know it.

(Refer Slide Time: 29:31)



Now, I come to the application side. So, once I done the or I use the DFT or frequency transform in speech, if you see suppose I have a time domain signal always showed you time domain signal and this is the this portion I have select this portion of the signal I have selected and take the frequency response of the signal. Now there is a point if I take n is equal to 1 0 2 4 point 1 0 2 4; that means, order of the DFT is 1 0 2 4. So, I take the signal 1 0 2 4 sample signal ok.

(Refer Slide Time: 30:14)



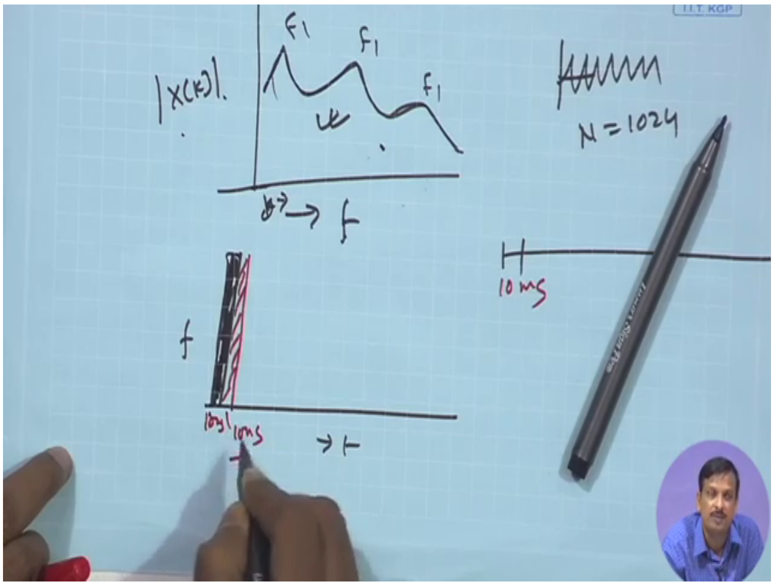
So, if I suppose I have a long signal which is stationary signal. So, whether I take this portion or I take this portion or I take this portion, this portion this portion or this portion, the signal property is remain same. In that case I do not have any problem.

Now, suppose the signal time is also changing. Along the time signal property it changing. So, if I say if I take this portion of the signal 1 0 2 4. So, whatever the frequency response, I will get frequency response I will get that is the average frequency response of whole signal. Means I am considering during this time the signal does not change it is property. So, what I am loosing? I am loosing the time domain resolution of the signal. Time domain resolution I am loosing.

Now if I want to increase the time domain resolution what I will do, instead of n equal to 1 0 2 4 let us I take n is equal to 256. Then my time domain resolution is increases. Because I my window size size is very less. So, I can get the better time resolution. Now what is happened? Since f is equal to frequency resolution is nothing but a F s by N. If N is decreasing the frequency resolution is also decreasing. So, once I increase the time domain resolution my frequency domain resolution is decreasing. So, if I take one signal I time domain resolution is loosing if I take small signal frequency resolution is loosing. So, this is the limitation of Fourier transform. That is why people are using more number of other number of transform like wavelength (Refer Time: 32:19) all kind of things is used ok.

So, let us forget about that things, suppose I told you record a voice or I can say record a signal and design an algorithm or design and compute or draw it is frequency response or spectrum of the signal I am not saying spectrogram. So, amplitude response representation is called spectrum. So, what is that it is nothing but a mod of x k versus frequency that drawing is called spectrum of the signal.

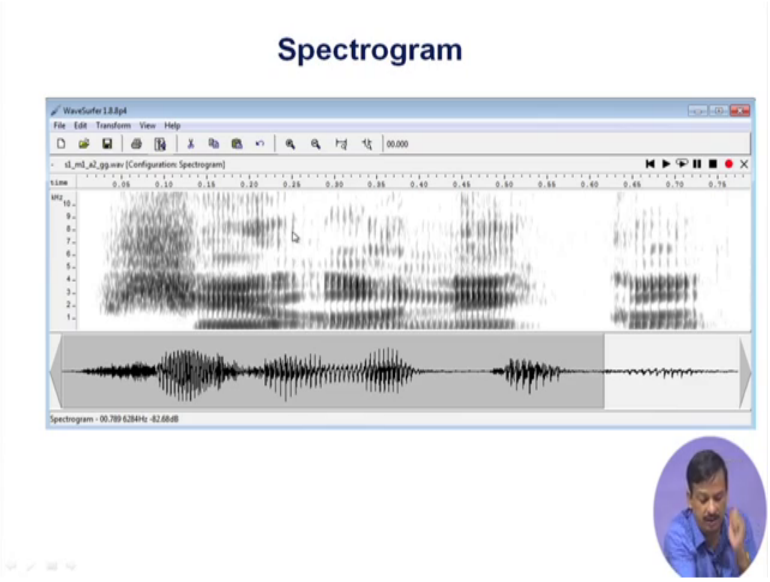
(Refer Slide Time: 33:00)



So, if I told you record the vowel and take the n equal to 1 0 2 4 point and draw the spectrum of the vowel. So, I will draw the frequency response of the vowel; that means, I can say I can draw it with respect to and I can convert the k with respect it then I get the frequency response of the signal. So, this can be draw in linear view or log view log view means if you see there is a 2 kind of view log view doing I will come later on. So, log view and linear view. So, there will be a log view I can take the output as a log. So, log spectrum a normal spectrum is a normal spectrum. So, then once I get the spectrum I get the formant frequency plot. So that means, I know the permanent frequency representation of the signal all details will come STFT when we discuss now this is called spectrum.

Now, when you draw, these kind of representation this is called spectrogram.

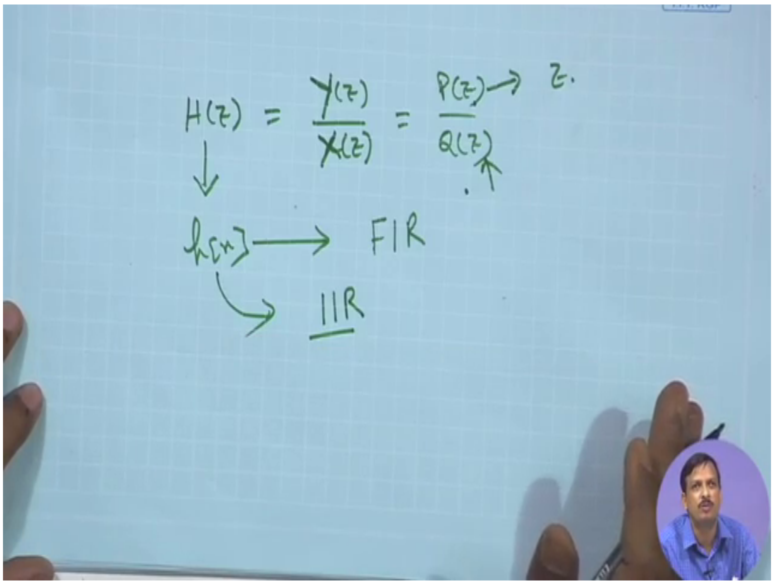
(Refer Slide Time: 34:19)



So, difference is that spectrum is frequency versus power means the power of each frequency plot. Spectrogram x axis is the time, y axis is the frequency and power is converted into the intensity or colour. So, it is the 3 d curve, but representation is that if you see there is a long line same line is coming why this line is coming. So, what I do? So, I want this axis is time, this axis is frequency and intensity as an amplitude. So, I have a long signal. So, I take so I take those one chunk of the signal from here, and find out this point, then I plot it here.

So, for this wholes chunk frequency representation will be same. Next chunk I will take and then I plot it here. So, if it is 10 millisecond window then this 10 millisecond will be same, this 10 millisecond will be same. That is why you see this kind of things are coming. So, this is called spectrogram and earlier one is called spectrum. So, next is the digital filter, this is the last portion of the DSP review. So, digital filter, if you say any forget about the filter any transfer function.

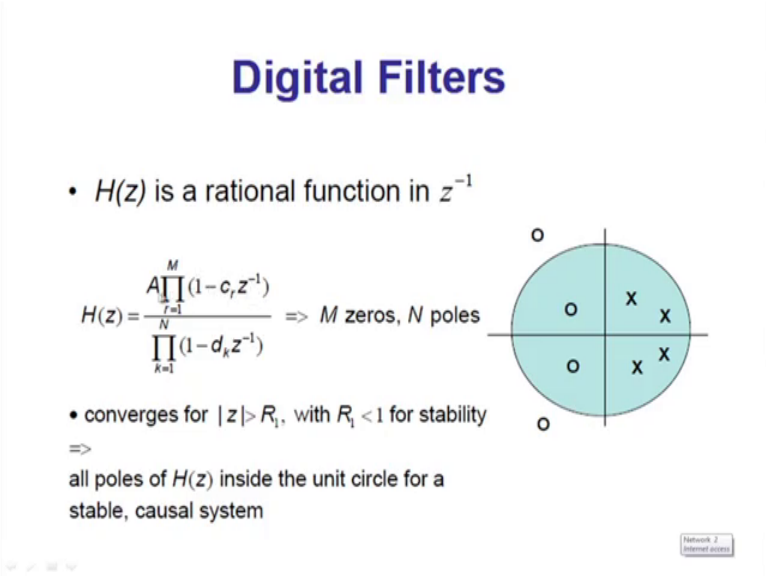
(Refer Slide Time: 36:06)



Let us I have a I have a transfer function let us h z, z transform you know. H z is nothing but a y output y z divided by input x z this is capital x capital y y z by x z is the transfer function h z. So, this H z is nothing but the some polynomial function. So, let us it is P z by Q z is nothing but a polynomial function. So, solution of P z provide me the 0 position and solution of Q z provide me the pole position. That is why H z is pole 0 filter, pole and 0 filter. That is why H z has a formant and 0 provide the anti formant or resonance anti resonance ok.

So, pole 0 filter. Any filter can be implemented using a pole 0 function. So, I can implement P z and I can implement Q z and I can implement h z. Now how it is look like this? If you see the mathematics.

(Refer Slide Time: 37:21)

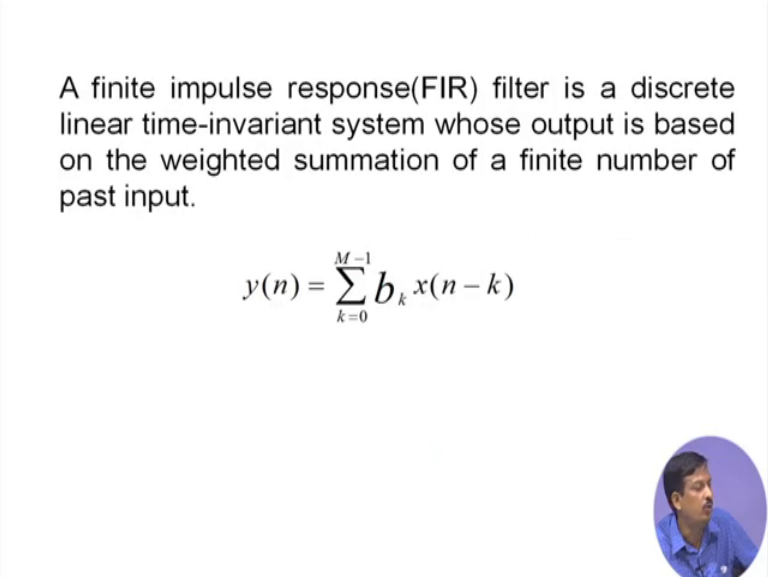


So, there is a number of pole n number of 0 if it is a order is n n number of 0 and n number of pole. So, every pole and 0 has a complex conjugate. So, that I will discuss later on that every pole has a complex conjugate if it is complex pole complex 0. So, it is at the real value H z is real then every pole on every 0 complex pole has it is conjugate value. So, if there is a pole in here, pole in here, there will be conjugate value pole in here. So, this is the unit circle in all poles a line in inside the unit circle then you can say there is a causal and stable filter now how do you implement filter in digital domain ok.

So, I can say H z has an impulse response H z has an impulse response or I can say H z in time domain it is nothing but h n. So, h n has an impulse response. It can be finite or it can be infinite non finite. So, if I say if it is infinite. So, if it can be finite or infinite. So, we say that it has a finite impulse response then we said the implementation is called FIR filter, finite impulse response filter. If I implement using infinite impulse response, and that is from minus infinity to plus infinity, then I call IIR filter. Infinite impulse response filter. So, one is finite impulse response, another implementation procedure is infinite impulse response ok.

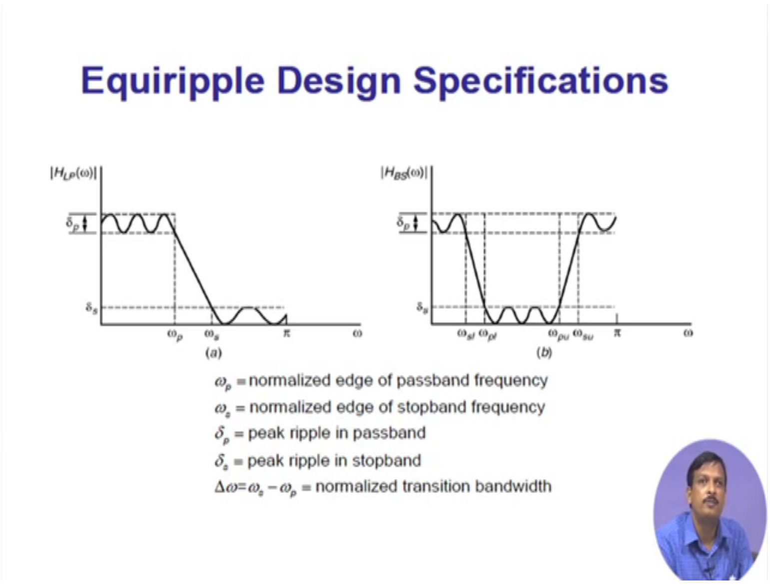
So, lest discuss about the FIR filter which is called finite impulse response filter ok.

(Refer Slide Time: 39:11)



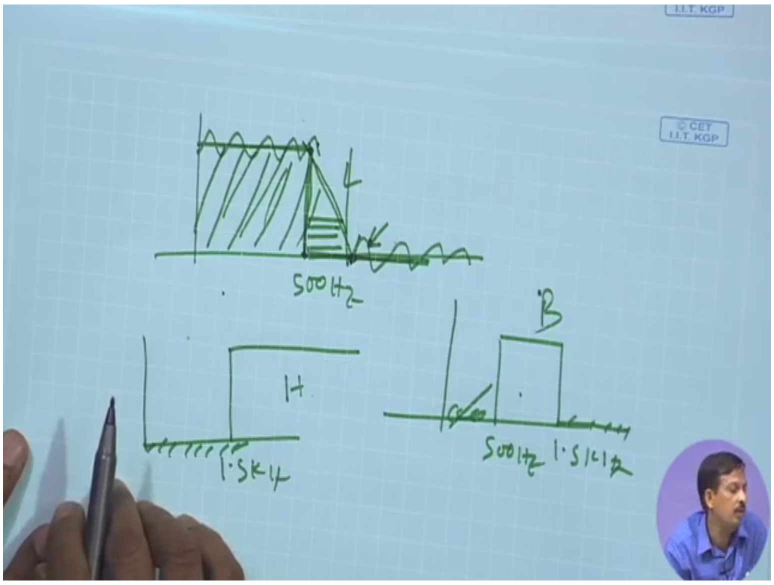
Now, what do you mean by a filter? That is very important. What do you mean by a filter?

(Refer Slide Time: 39:25)



So, suppose I filter means I something is unwanted I want to discard, that is filter. Now in the signal suppose I have a signal which has a frequency response let us all frequency are present. Now I want to cut down after 5 kilo hertz I want signal or system should not less signal should not produce any response; that means, that filter is there filter frequency response is there up to 500 kilo hertz or 5 5 let us 500 hertz, I want response should be like this, and after 500 hertz I do not want anything.

(Refer Slide Time: 40:00)

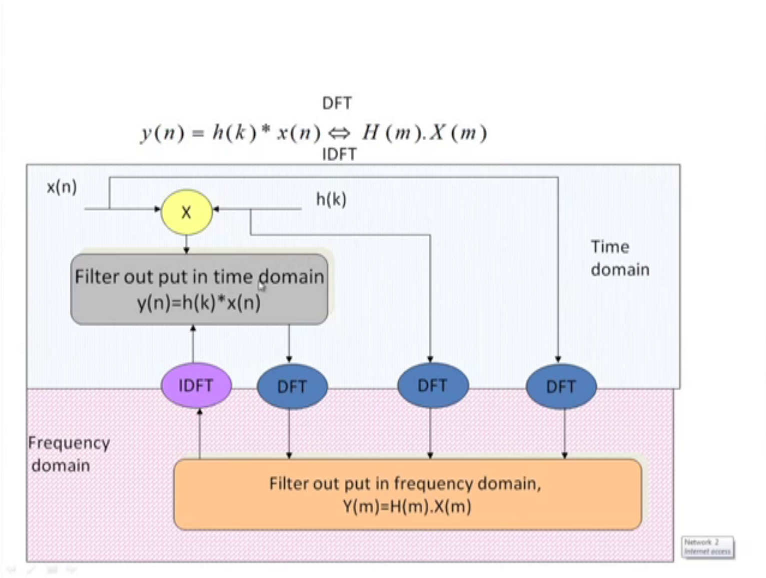


So that means, 0 to 500 hertz all frequency components should pass to the output, and after 500 hertz, none of the component should pass to the output, then it is called low pass filter. Because low frequency component is passed. High frequency component is attenuated to 0. So, there may be I may want this type, that frequency component between 500 hertz to 1.5 kilo hertz may pass. This portion also becomes 0 and this portion also becomes 0. So, that is my requirement, or I may say I this is called band pass filter. Or I may say I want a filter after 1.5 kilo hertz all frequency should pass, below 1.5 kilo hertz none of the frequency should be there. So, that should be 0, that is called high pass filter.

So, it is low pass filter, it is band pass filter it is high pass filter. That is my ideal filter frequency response I want, this is the ideal response, but if I implement it with a finite impulse response filter, then I instead of getting that single that this pass band with one, instead of this if you see this is flat one, here I can get a some kind of variation, that is called pass band ripple. Or instead of some cut off I can get a transit point that is called transitory response. And after that instead of 0 I can get some signal here also, that is called stop band ripple. Pass band ripple, stop band ripple, and filter condition transition bandwidth. So, where the amplitude is one to 0 is transitory. So, that transitory portion that frequency range is called transition bandwidth. So, if you say normalised when the peak ripple is pass band peak ripple in stop band and normalised transition bandwidth. So, this point to this point is called normalised transition bandwidth. So, this is pass band frequency, this is a stop band frequency, where I want to stop that filter. So, this bandwidth is called transition bandwidth ok.

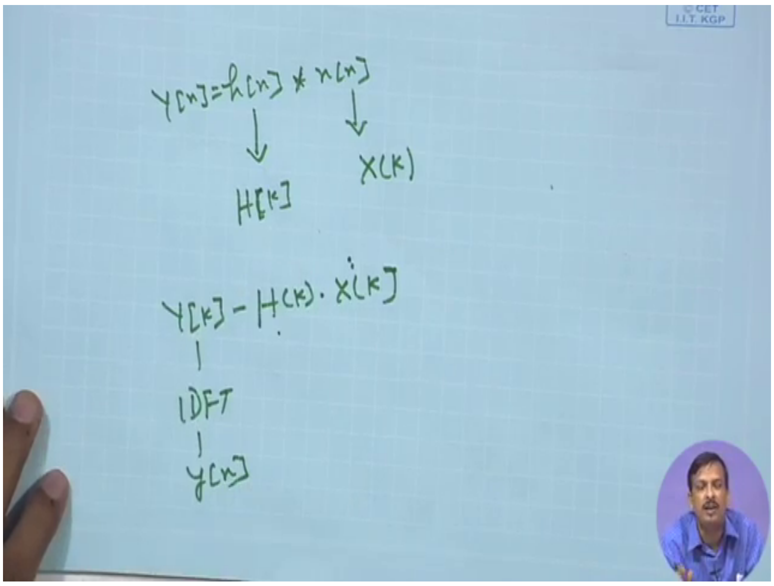
So, how do you implement the filter? FIR a too much outs.

(Refer Slide Time: 42:58)



So, I can say that plus I have a filter whose response is h n.

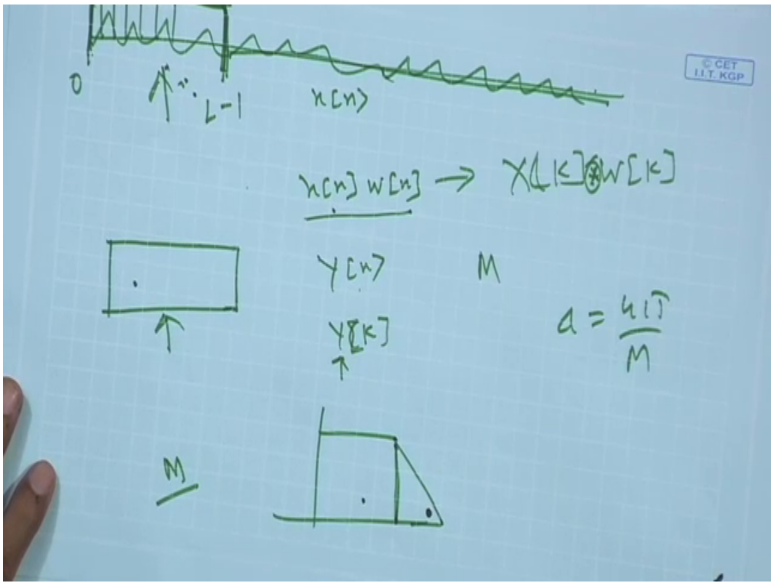
(Refer Slide Time: 43:04)



If I know the transfer function time domain transfer function of the filter, then I convolve the input signal I get the output of the filter. Or what I can do? I can take the frequency response of the filter which is h k, and I can take the frequency response of the input signal x k, and multiply h k into x k. I get y k, then I take the IDFT to get the y n. So, either I can implement in frequency domain, I can frequency domain transfer the h n h k, then I can get take the time domain signal in frequency domain, multiply both and get the output and take the IDFT get the input.

So, this is the 2 implementation either frequency domain implementation or time domain implementation. So, have you understand this either I can implement the filter in time domain or I can implement the filter in frequency domain. Now suppose I want to implement the filter in time domain or frequency or any domain when I want to implement, how to implement it? FIR filter implementation. So, suppose I have a long signal, long signal is there.

(Refer Slide Time: 44:32)



There is a long signal if I want take the whole signal. So, signal may be used signal. So, I do not want to take the whole signal at a time. Then what I have to do let us I have to cut the signal this portion, and apply the filter here then the take the next chunk next chunk next chunk again do that way.

Now, once I cut the filter what I am doing? Whether I take the DFT or filter, once I cut the cut the signal in a chunk; that means, if this is x n I am multiplying x n with a window function. Which is one within the interval let us it is l minus 1 0 to l minus 1. So, 0 to l minus 1 window function is a square amplitude is 1, after that it amplitude it 0, that is window function. So, if it is one if this type of window I am not multiplying the signal with any function, this is called rectangular window.

Now once I do the rectangular window then my frequency response of the output become y n, y n if I take the y k is also y k is also contain the frequency response of this window function. So, I am not getting the exact frequency response only the filter and input signal, but also it is combine with window function frequency response, that is why I get the different kinds of ripple. So, if you see different kind of window function, I define to reduce the different ripple stop band ripple and pass band ripple and different kind of window function is defined that is called blackman window hamming window hanning window kaiser window.

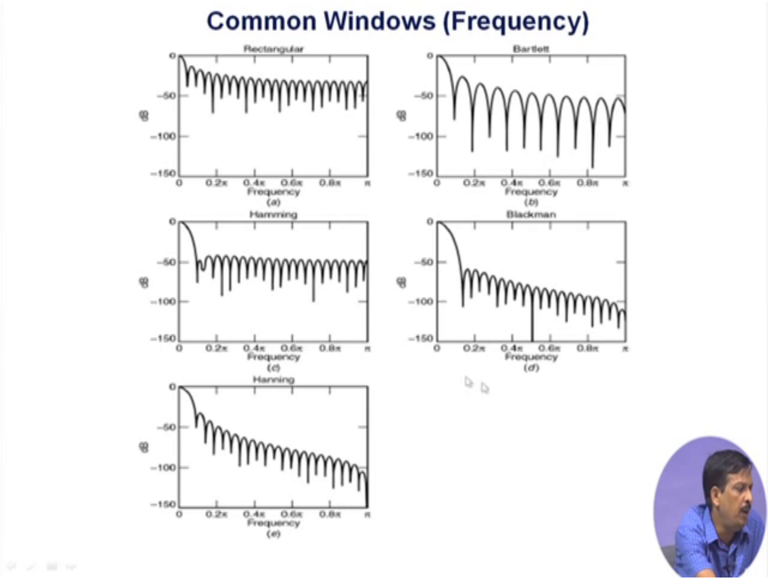
(Refer Slide Time: 46:24)

Window Function for FIR Filter Design

Name of Window	Window function
Bartlett(triangular)	$1-\frac{2\left\|n-\frac{M-1}{2}\right\|}{M-1}$
Blackman	$0.42-0.5\cos\frac{2\pi n}{M-1}+0.08\cos\frac{4\pi n}{M-1}$
Hamming	$0.54-0.46\cos\frac{2\pi n}{M-1}$
Hanning	$\frac{1}{2}\left(1-\cos\frac{2\pi n}{M-1}\right)$
Kaiser	$\frac{I_0\left[\alpha\sqrt{(M-1)^2-\left(n-\frac{M-1}{2}\right)^2}\right]}{I_0\left[\alpha\left(\frac{M-1}{2}\right)\right]}$

So, I will everybody has an window function. That is multiply with a input signal, and then convolve with a this Fourier transform function I get the output.

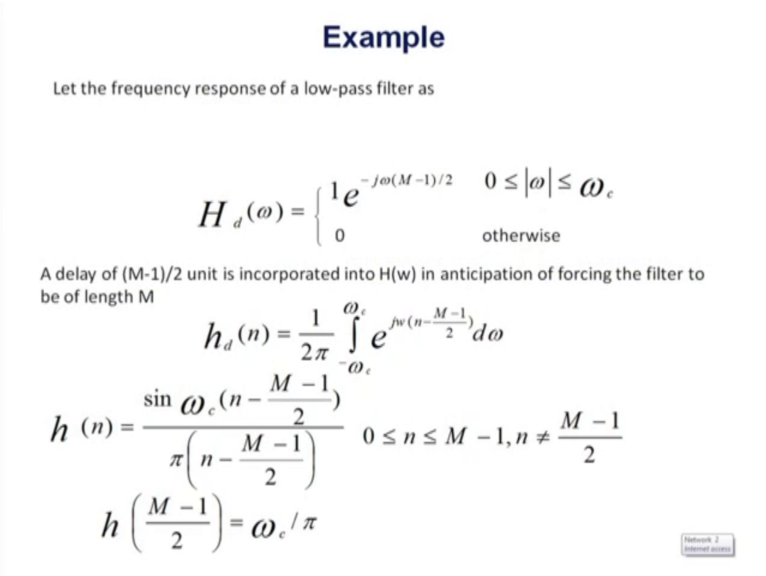
(Refer Slide Time: 46:52)



So, if I implement the filter this is the common window windows frequency response if it is frequency response of the rectangular window if you see like this. So, ripple is much more if I want to bartlett blackman window, frequency response is like this hamming window this hanning window this. So, most cases hanning window we are using ripple is very less.

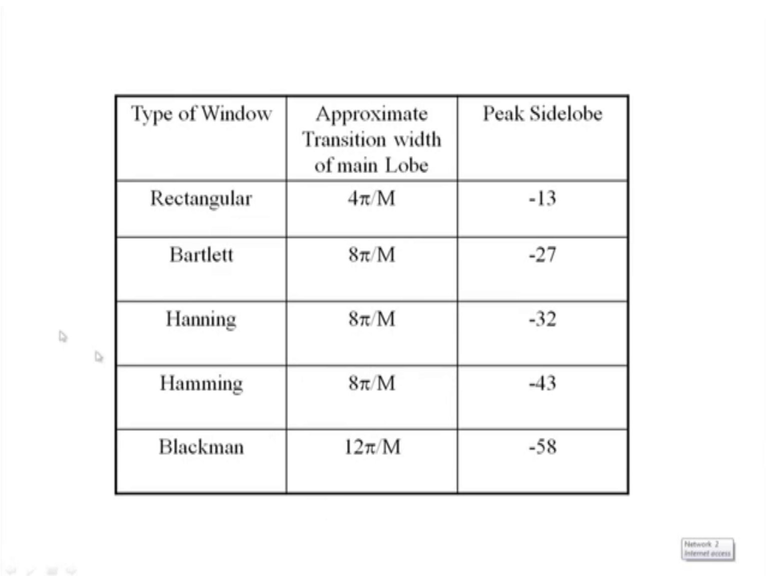
So, different window function has different kind of frequency response this frequency response will combine with a frequency response of the signal and pass to the filter. So, it is time domain multiplication; that means, window multiply with the signal in time domain. So, frequency domain it will be convolution. So, x k will be convolved with window function, and then multiply with frequency response of the filter ok.

(Refer Slide Time: 47:49)



So, this is happening. Details I will discuss when we discuss about the STFT short term frequency response short term Fourier transform. Now if you see the example implementation for a time domain signal, suppose I want to design a low pass filter cut off frequency is omega c. So, this is the frequency response of the filter. So, you see here implementation of filter, usually provide the frequency response of the filter. I have to design if I want to if I want to design in time domain.

(Refer Slide Time: 48:33)



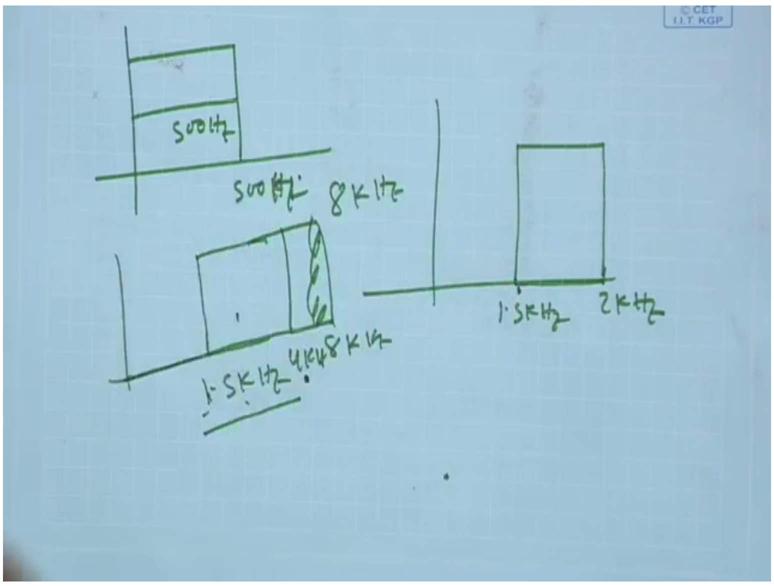
Type of Window	Approximate Transition width of main Lobe	Peak Sidelobe
Rectangular	4π/M	-13
Bartlett	8π/M	-27
Hanning	8π/M	-32
Hamming	8π/M	-43
Blackman	12π/M	-58

Then I have to derive the time domain representation or time domain transfer function or time domain impulse response of the filter and then convolved with a input signal. So, if I do that I will get this kind of response, and then I can implement it in time domain. So, if you see the book there is different. So, different window has different type of tension transition bandwidth and peak sidelobe. So, this is a ripple in peak sidelobe and this is the transition bandwidth.

So, if I rectangular filter the order of the filter is M then transition bandwidth delta is nothing but a 4 pi by M. So, M is the order of the filter if it a implement the filter in FIR way. So, I said finite impulse response, ideally an filter has an infinite impulse. So, I truncated the infinite impulse to a finite number, if it is M, then m is called order of the filter. Ideally if I want to design a low pass filter, this is infinite impulse response. Once I truncated it I cannot get this half cut off.

So, I get a transition. So, this transition bandwidth depends on the what kind of window I have used to truncate the Fourier transform function or truncate the signal. So, if I use the rectangular window, then it is 4 pi by m if it is hamming windoweight pi hanning window hamming windoweight pi by m hanning windoweight pi by m. So, depending on the requirement I should use which kind of window I should apply. Next one how to implement a band pass filter. So, suppose I implement a low pass filter up to 500 kilo hertz. Let us it is 500 hertz. So, bandwidth is 500 hertz.

(Refer Slide Time: 50:19)

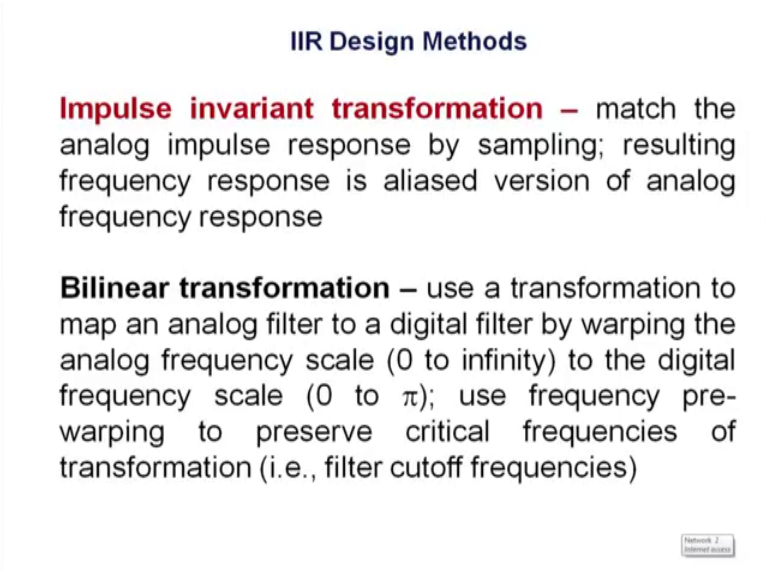


Now, I want to implement a band pass filter of 500 hertz bandwidth with a frequency between the 1.5 kilo hertz to 2 kilo hertz. So, what I will do? I will just shift this low pass filter to 1.5 kilo hertz frequency. So, that I get this one understand or not.

So, it is nothing but a frequency shifting of the low pass filter. So, any band pass filter I can implement in just implement the low pass filter of the desired bandwidth and shifted that frequency response where you want to implement the band pass filter. Similarly suppose I want to design a high pass filter of 1.5 kilo hertz. If the sampling frequency of the signal is 8 kilo hertz, then in frequency scale what is the maximum signal up to 8 kilo hertz I can get base band signal is 4 kilo hertz again. Even I can get the base band signal is 4 kilo hertz.

So, it is nothing but a band pass filter 1.5 kilo hertz to 4 kilo hertz if I design that filter and shifted at 1.5 kilo hertz I get the high pass filter. So, that way I can design the filter details filter design I have not going here because that you can that is prerequisite of this course that if you want to know the digital filter design digital signal processing you have to study. I just give you some concept of that you can say that this is overview of DSP. So, how to implement an IIR filter? There is a lot of methodology are there. By any I can take that derive that filter transfer function in x domain.

(Refer Slide Time: 52:14)



then go to the bilinear transformation to make it z domain, and once I get the z domain transfer function, I can use a discrete system to design that transfer function. So, that way you can implement.

So, this is the prerequisite part which I will be referred many times during the speech processing class. So, this is the overview of DSP. Please go through these slides, if it is there is a suppose you do not cover the DSP, then if you do not understand the particular point, then you read that DSP book (Refer Time: 52:53) or whatever digital signal processing book you can read. And also you can post that I am not understand this portion then I will help whatever I can do. Because if I just consider at the DSP then I am not I have to complete speech part. So, I am just truncated that just gist or summarise the DSP required DSP things in here, and some DSP also I require higher end DSP signal processing algorithm I will be used. And that time I will describe that things when I will use them.

So thank you.