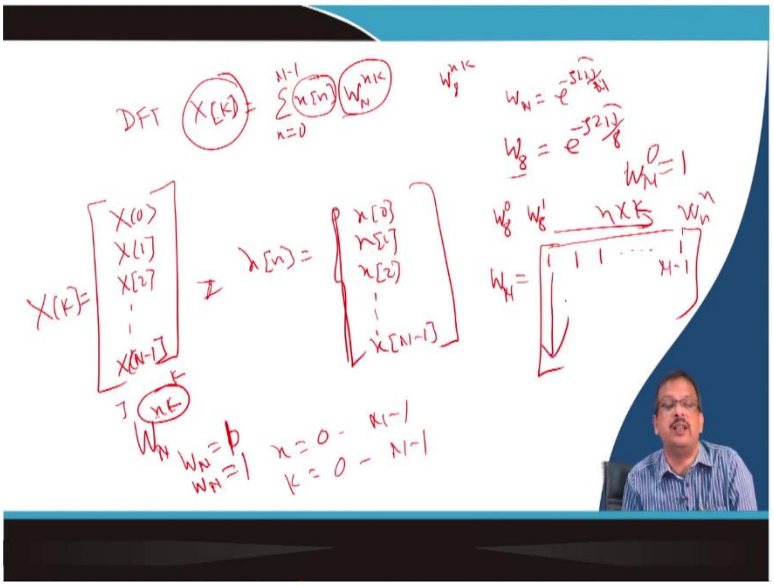
Signal Processing Techniques and Its Applications
Prof. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 19
Discrete Fourier Transform Linear Transform View (Continued)

(Refer Slide Time: 00:40)



So, let us see the DFT as a linear transform, the linear transform linear transformation. So, DFT or IDFT. So, what is DFT?

$$X[k] = \sum_{n=0}^{N-1} x[n]W_N^{kn}$$

where WN is equal to $e^{-j2\pi/N}$. So, I can say that if it is N point, let us say 8 points. So, I can say W 8 equals $e^{-j2\pi/8}$, ok or not.

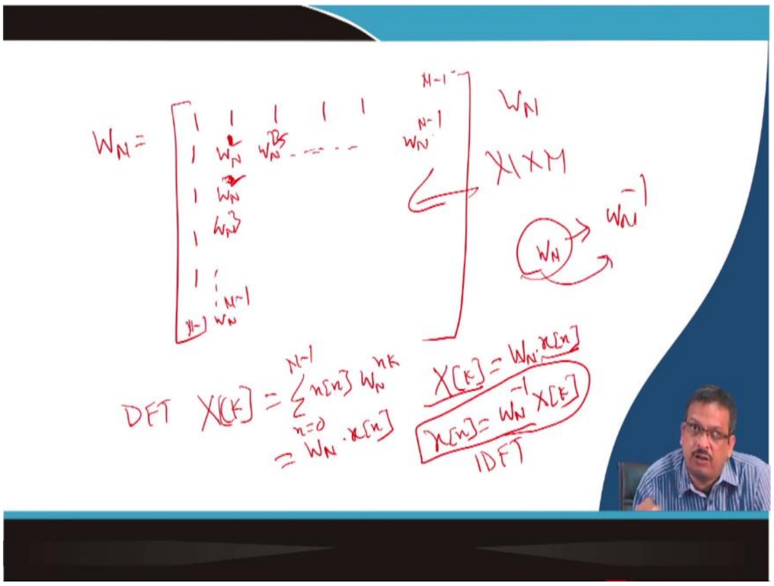
So, I can calculate W 8 easily, and then I can calculate instead of calculating directly here. I can calculate W_8^0 W_8^1 because it is W8. So, if the 8 point DFT, then W_8^{nk} or if I generally if I write. So, it is nothing but a matrix, WN is nothing but a matrix ok, x[n] is also an N matrix and X[k] is also an N matrix. So, let us say what the X[k] matrix is; this is nothing but an X[0], X[1], X[2] dot, dot, dot, dot, dot x[n-1].

What is x[n]? x[n], x[n] is also an signal. So, I can say x[n], this is X[k], x[n] is equal to nothing but a x[0] matrix form, x[1] x[2] dot, dot, dot x[n-1] ok. Now, what is the WN matrix? So, WN is nothing, but I can say WN 0 when I say k equals 0; if I put k equals 0, WN to the power 0 is nothing but a 1.

So, I can say the upper matrix is 1 1 1 dot dot 1. How much? WN n minus 1 time 0. Similarly, this side also I get. So, on this side, I can get WN nk ok understand, WN to the power n k, you can say there is a nk index. If I say this is j this is k, so, k index and n index.

So, if the n index is 0 when I say the n index is 0, then also WN is equal to 0. When the k index is WN equal to 1, when is k n equal to 0, k equal to 0, then also it is 1. So, I have a matrix which is a direction n, n cross k. So, I can say. So, all n also varies up to 0 to N minus 1 and k also varies from 0 to N minus 1. So, this side also I have an N minus 1 element and this side also I have an n minus element.

(Refer Slide Time: 04:38)

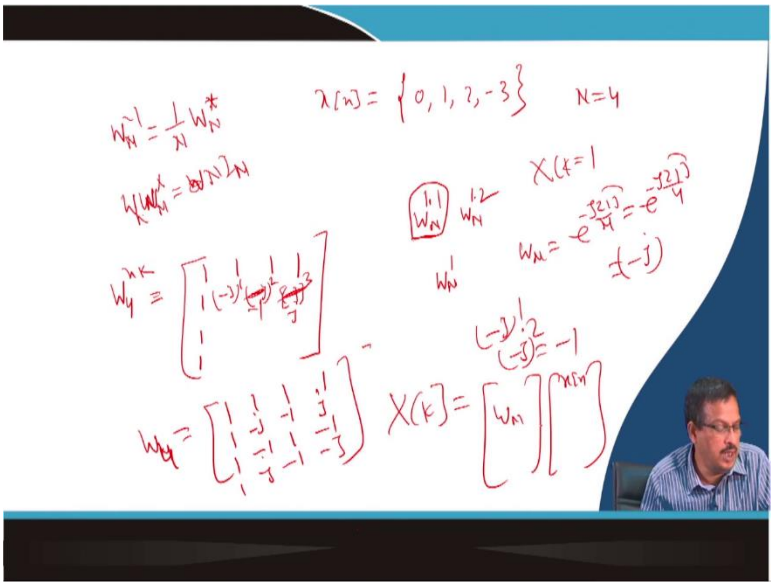


So, if I want to write down the WN matrix, on this side, I have a 1 1 1 1 1 N minus 1 number of elements, and this side's first column will be 1 N minus 1 number of 1. The second column will be WN square, WN cube dot, dot, dot, and dot WN^{N-1}, ok? Is that ok or not? Similarly, WN square if it is WN square, sorry W first is WN then square then N minus 1, similarly WN W N square WN cube dot, dot, dot, dot WN N minus 1.

So, I have an N cross N matrix of WN, which varies from 0 to N minus 1. Now, I can say that DFT(Discrete Fourier Transform) is nothing but an X[k], which is equal to the summation of n equal to 0 to N minus 1 x[n] WN n k. So, I can say it is nothing but a x[n] multiplied by I can say WN multiply by x[n]. So, I can say X[k] is nothing but a WN multiplied by x[n].

So, WN, is this matrix coefficient matrix okay? If that is, let us say I have a WN that exists as an^{-1} transform. So, I can say x[n] equals WN^{-1} X[k], linear transformation. So, if the WN matrix exists, its^{-1} transformation, then I can say x[n] is nothing but an X[k] divided by WN, which is nothing but a matrix^{-1}. So, this is nothing but the IDFT. But this WN^{-1} is not exactly the^{-1} of this WN.

(Refer Slide Time: 07:36)



What is WN^{-1}? Basically, the WN^{-1} can be written as 1 by N WN conjugate, WN conjugate. So, that is why WN into WN conjugate, if I say it is equal to W, or I can say this is nothing but an N into I N, identity matrix N into I N ok. Let us see an example, and then you understand how this kind of thing will be used for matrix algebra can be used for calculation.

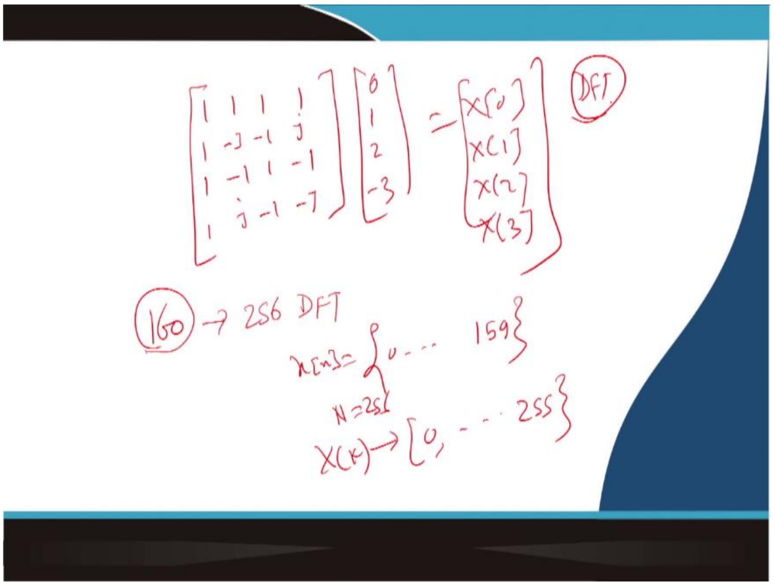
So, let us say I have a signal x[n]. Again, let us say I have a signal. Let us say 0, 1, 2, minus 3, 4 sample signal and N is equal to 4. So, I can say I have a W 4 n k. So, if I want to calculate the matrix, I would say 1 1 1 1 on this side, 1 1 1, okay? Now, what is WN n k? Let us say WN into n k, N equal to 1. So, what is it first say? k equal to 1 and n equal to 1.

So, k is equal to 1 n equal to 1 1 into 1, then I can say k is equal to 1 n equal to 2, so 1 into 2. So, what is WN 1 into 1? Nothing but a WN to the power 1. So, what is the value? I can say WN is equal to nothing but an $e^{-j2\pi/N}$, which is nothing but an $e^{-j2\pi/4}$ in this case.

So, I can say it is nothing but a minus j. So, I can say minus j to the power 1. So, here it will be minus j to the power 1, minus j to the power 1 into 2 and minus j to the power 1 into 3, ok. So, it is nothing, but I can simplify it minus j to the power 1 minus j minus j. The whole square is nothing but a minus 1. So, this will be replaced by minus 1, and this will be replaced by j.

So, the matrix is W 4 WN, or a W 4 matrix will come to 1 1 1 1 1 1 1. So, it is minus j minus 1 j. Then, if you calculate, it will come minus 1 1 and minus 1. Then, it will come to j minus 1 minus j. This is the W4 matrix. You can calculate; you can do it. Now, if I multiply this with. So, the X[k] matrix is nothing but a WN matrix multiplied by signal matrix x[n].

(Refer Slide Time: 11:32)



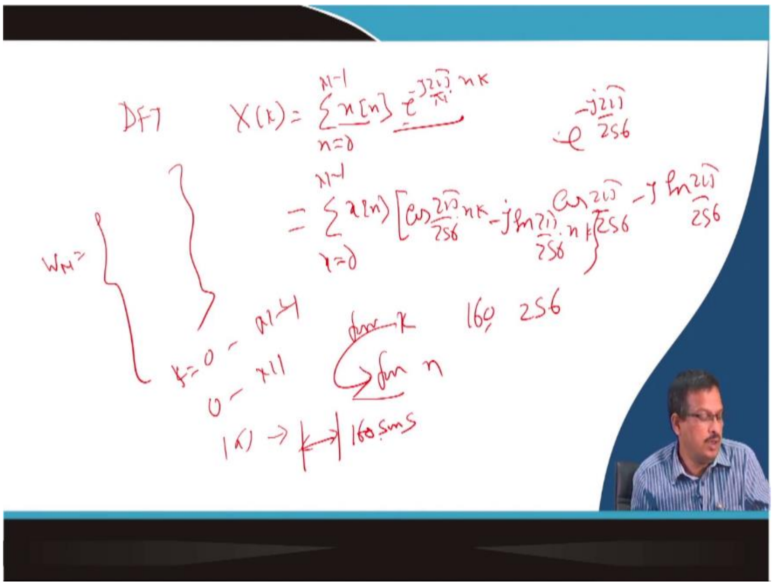
So, what is the signal matrix? So, I can say it is 1 1 1. So, I can say 1, 1, 1, 1, 1, 1, 1 and minus j, minus 1 j, minus j, minus 1 j, minus 1, 1, minus 1, minus 1, 1, minus 1, and j minus 1, minus j, j minus 1, minus j. This multiplied by X equals 0. So, X equal to 0 is 0, X equal to 1 is 1 2 and minus 3, 1 2 and minus 3. Now, you do the multiplication. So, you can get X[k].

So X of, you get X[0], X[1], X[2], X[3] you get that. So, by a linear transformation, I can also calculate the discrete Fourier transform DFT; this is the calculation. So, how do you suppose you want to implement it? Forget about that hand calculation. Let us say I give you a signal whose length is equal to 160 samples, and I tell you, let us say, given a practical example, I told you to record an acoustic signal or audio signal and collect 160 samples and compute. Let us say 256-point DFT.

So, I collected 160 samples. So, my x[n] varies from 0 to 159 number of samples. I have to compute N equal to 256 point DFT to get X[k]. So, X[k] consists of 0 to 255 values. So, how do I do it? First, I have to write a DFT program.

So, what is the program? Very simple, so I know what is the program of the DFT program Let us say I take another slide: simple maths. Let us say without any simplification of complexity and other things, I just simply want to implement DFT in a C function, implement DFT in a C function.

(Refer Slide Time: 14:14)



So, I said to write a c algorithm for computing DFT Discrete Fourier Transform. So, you know.

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j\frac{2\pi}{N}nk}$$

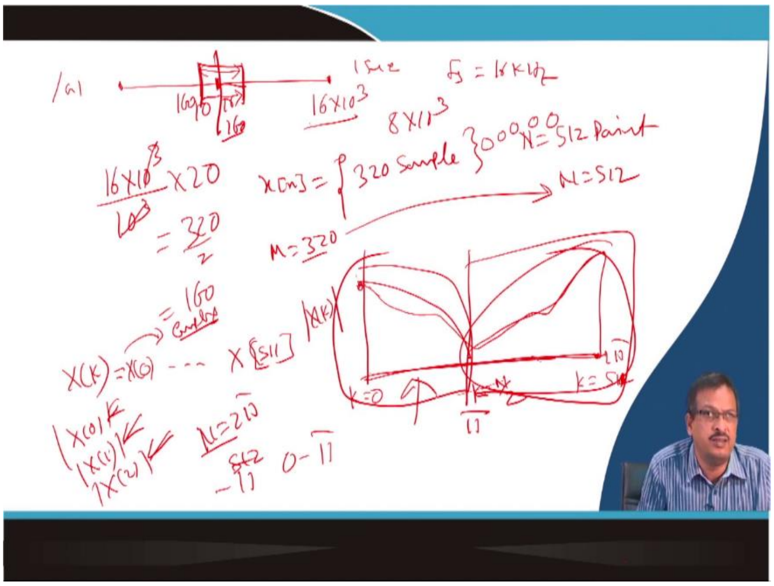
So, instead of every time, I calculate this one. So, if it is 256, I can calculate j2π by 256, that coefficient I can calculate separately, which is nothing but a cos 2π by 256 minus j sin 2π by 256.

So, I can calculate this WN matrix and store it in a file. Now, I simply multiply x[n] with those things and write a for loop, multiply that and add them. Or you can directly also do it. So, instead, mathematics is nothing but a n equal to 0 to n minus 1, x[n] into cos 2π by 256 into nk minus j sin 2π by 256 n k.

So, you have a very for loop, first k equal to first for loop k from 0 to n minus 1 second for loop will be calculated for every k, varying from 0 to n minus 1. So, I have a nested for loop, first for loop for k and second for loop for n. For every k, I have to calculate all n products and sums. So, I can easily write down that thing and the program.

So, this is the you can say that calculation of DFT then, that how do you. Now, if I say, if I ask you for 160 sample 256 point DFT, plot the magnitude spectra and draw the magnitude spectra for practical application. Let us say I said record your voice [FL], you create [FL], record the voice from that voice, and you take a window of the voice. I am not explaining that part; I will explain it later on SDFT.

(Refer Slide Time: 17:19)



So, you take a window of the voice and collect 160 samples, or I said, let us say practically I said to record a vowel [FL]. Let us say for 1 second with the sampling frequency Fs is equal to 16 kilohertz. So, how many samples in 1 second will be there? 16 into 10^3 sample will be there; let us take a 20 millisecond window in the middle. The 20-millisecond signal from the middle. So, you create a mark in the middle, 10 milliseconds this side, 10 milliseconds this side.

So, what is the sample index? So, 16 into 10^3 samples are included here. So, half of that means 8 into 10^3. So, I go to that sample, and then I say, take 20 milliseconds. So, how many samples will be there in 20 milliseconds? So, 20 milliseconds in 1 second, I have 16 into 10^3 samples. So, in 1 millisecond, that many samples in 20 milliseconds multiplied by 20.

So, 320 samples. So, 10 milliseconds divided by 2, 160 sample ok. So, I have 160 samples on this side and 160 samples I have. So, I have an x[n] with 320 samples, ok. Now, I said to compute DFT with a length of the DFT N equal to 512 points. So, how many 0s do I have to pay up? M is equal to 320, and N is equal to 512. So, I have to make 320; I have to make 512. So, that many 0s I have to add.

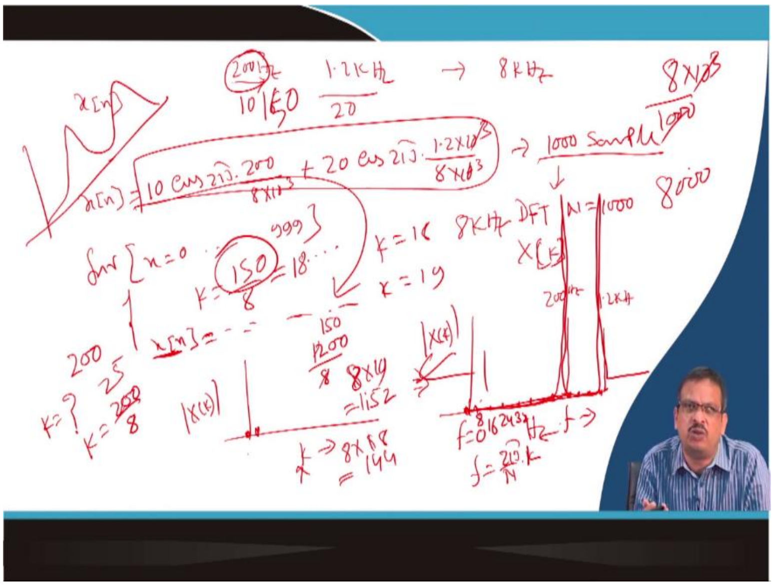
So, outside the window, my signal will be 0. Now, I write the C program and compute the DFT. So, once I compute the DFT, I get X[k]. So, X[k] has a value 0th sample. So, X[0] to X of 511. So, many X values will be there, and every X is a complex number. So, I have to calculate the mod of X 0 mod of x[1] mod of x[2]. I can calculate it easily if you calculate it. So, you get a sequence, which is a |X[k]|.

Now, I told you to plot the magnitude spectra. So, magnitude spectra mean the X-axis will be k, and the y-axis will be the |X[k]|. So, I have already computed different X k values, and I have taken the mod; I just plot that value. So, let us say you can get this kind of plot; let us say this kind of plot you get. Here, k is equal to 512, k is equal to 511 last point, and k is equal to 0. You can see you get a symmetrical plot at k equal to N by 2.

So, this portion will look the same as this portion; why? You have seen that N is equal to 2π. So, I have divided that 2π frequency in an N number of samples, N number of samples, which is 512. So, N by 2 is nothing but a π; I know the baseband signal frequency can vary from 0 to π or 0 to minus π; that is why I get 0 to π, and this represents 0 to minus π. You can fold this side, this side, so it is.

So, that is why this is 2π; you can plot and see it using MATLAB or a C program. You can plot it, calculate the value, go to the Excel sheet and plot that Excel, or you can use another plot function also; you can see this kind of plot you will get. Now, let us say in the ideal case, an ideal case if I say that ok.

(Refer Slide Time: 22:30)



Let us take another slide and write down the problem to generate a signal x[n], which has a frequency component of 210 hertz and 1.2 kilohertz with a sampling frequency of 8 kilohertz. I said first, you write a program to generate a signal x[n], which has 2 components, 200 hertz with amplitude 10 and 1.2 kilohertz, let us say amplitude 20.

So, what is the equation? x[n] is equal to 10 cosine value, that is a cos function; cos 2π 200 hertz, F by F s 8 kilo hertz plus 20 cos 2π into 1.2 kilohertz divided by 8 kilohertz. So, this is your x[n]. So, generate 1000 samples. How do you generate? You write a for loop for n equal to 0 to 1000 minus 1. So, 999 ok n minus 1, x[n] compute this 1. You can write this function here directly, ok.

So, now x[n] has an array which contains the 1000 samples of this signal. Now, I said compute the DFT of length N equal to 1000; compute the DFT of length N equal to 1000. So, 0 padding is not required. I generate it. Now, you compute. So, what do I get? I get X[k], 1000 samples of X[k]. Now, I said, draw the magnitude spectra of X[k]. So, draw the magnitude spectra; so, I draw it. How do I draw it? I put it k on this axis, and this axis is |X[k]|.

Now, I told you instead of discrete frequency, draw it against another frequency scale, the hertz scale. So, I want to convert this to this axis is hertz, and this axis is $|X[k]|$. So, how do you convert? k equal to 0 means f equal to 0. So, this scale is f, k equal to 0 means f also 0 k equal to 1 means, f equal to how much? f equal to 2π by N into k. So, k is equal to 1. So, it is nothing but the 2π. What is the 2π? 2π means 8 kilohertz sampling frequency is 8 kilohertz. So, 8 divided by n equal to 1000.

So, the resolution is 8 hertz. So, I got 8 hertz, 16 hertz, 24 hertz, and 32 hertz, so I will get those points. Now, if I want to generate 200 hertz, what should the k value be? So, the resolution is 8 hertz, so k is equal to 200 divided by the resolution. So, I can say it is nothing but a 2 16 25. So, I get k at k equal to 25, I get 200-hertz component ok, then 1.2 kilohertz. So, 1.2 kilohertz divided by 8. So, I know 1 at k is equal to 150, so I get a 1.2 kilohertz component.

Now, suppose instead of 200 hertz, let us say 150 hertz, and then what is the value of k? k equal to 150 divided by 8. So, I can say 1 8 18 point something is coming. So, k is the coming fraction. So, either k is equal to 18 or k is equal to 19 because k cannot be a fraction. So, in f case either I get 8 into 18, which is equal to 144, or I get 8 into 19, 152.

So, either 152 hertz or 144 hertz; so, I cannot get exactly 150 hertz. So, as you know what they should look like, the ideal spectra should be a very high spike at a particular frequency and rest where it will be 0. But if you draw it, you do get a high spike. You get this kind of distribution, which is called DFT leakage. DFT leakage is caused by the power of the 150-hertz component being distributed nearby.

So, once since your frequency resolution is 8 hertz, your frequency resolution is 8 hertz. So, you can say that it is nothing but up to 8th hertz. The information is the same. So, you can get a single-function kind of response. So, what is the ideal frequency response instead of an idle to 150 hertz component?

At 150 hertz, it will be a very sharp spike. Otherwise, it will be 0, but once you draw it, you cannot get that idle 1. You can also get that energy distributed to nearby components. That is called DFT leakage. You do it and see it, and you fill in what DFT leakage is. Now, if you increase or decrease the resolution, you see that power is distributed among nearby lines.

So, if you increase the resolution, you see the peak is coming very sharp, which means leakage is reduced. Suppose I want to make the resolution 1 hertz in this case, then what should be the length of the DFT? 8000, which is computationally very complex. So, that is depending on your requirement, you have to choose the DFT length that I will come up with; that how it will make it computationally effective, which is called FFT Fast Fourier Transform.

So, for fast Fourier transform, we will explain how it is done and how it improves the speed of the calculation of the DFT, and then we will express the properties of DFT. So, different properties are there. So, 0 padding I have already mentioned, I have mentioned the DFT leakage, and I have mentioned the symmetry. So, mathematically, we see what symmetry property and circular convolution are, so those things will come up in the next lecture.

Thank you.