Digital Signal Processing

Prof. S. C. Dutta Roy

Department of Electrical Engineering

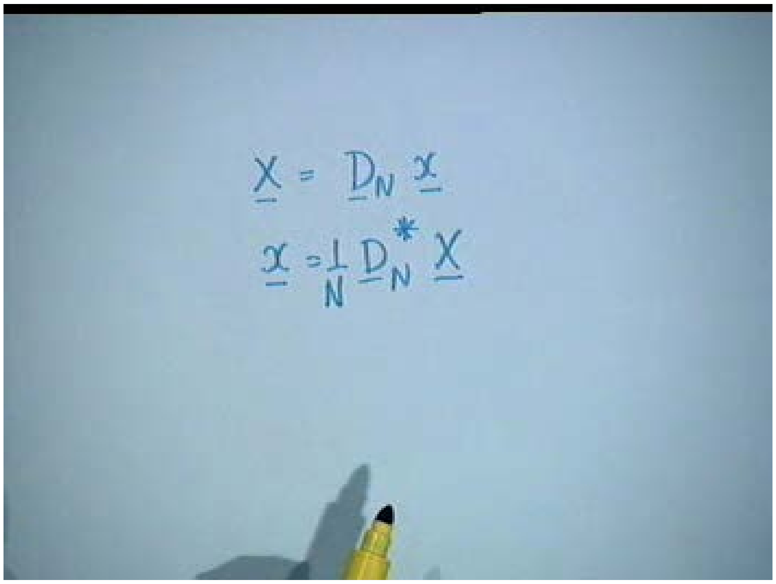
Indian Institute of Technology, Delhi

Lecture - 10

Discrete Fourier Transform (D F T Cont.)

This is the 10th lecture on Digital Signal Processing and we continue our discussion on DFT. In the last lecture we discussed the symmetry relationships for Fourier transform, and introduced the concept behind DFT. What is the concept? There are N number of pieces of information in the time domain, and therefore, N number of pieces of information in the frequency domain should be adequate for any signal. The latter are usually obtained by sampling the Fourier transform at N number of points at uniform intervals. We also stated the formulas for DFT and IDFT and emphasized that one follows from the other. The two are not independent of each other because DFT as well as FT are one to one transformations. We sketched the proof of IDFT and we gave examples of delta n, delta (n – m) and 2cosine (r(pi)n/N). Then we said that DFT is useful and important because of FFT, which itself is not a new transformation; it is an algorithm for computation of DFT. Then we showed how from the N samples in the frequency domain, i.e. X(k), k = 0 to N – 1, one can find the total spectrum $X(e^{jomega})$ which is a process of interpolation. The formula does not look very nice but it can be programmed and then you can rapidly compute the in between samples of DFT. We also made a matrix representation of DFT and IDFT.

(Refer Slide Time: 03:00 – 04:21)

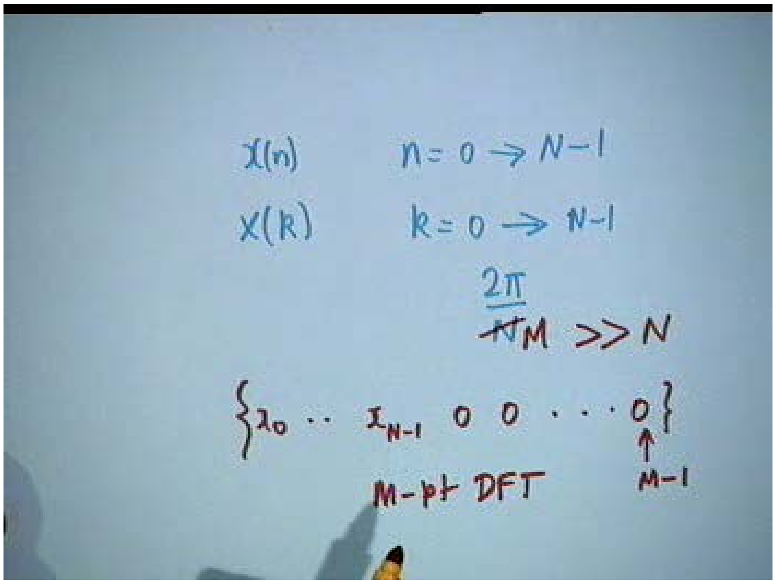


$$\underline{X} = \underline{D}_N \underline{x}$$

$$\underline{x} = \frac{1}{N} \underline{D}_N^{*} \underline{X}$$

We showed that $\underline{X}$ vector is simply $\underline{D}_N$ which is an N × N square matrix, times $\underline{x}$ vector. The entries in $\underline{D}_N$ are as follows. The first row is all one, the first column is all one, and then for the rest, you have powers of $W_{n.}$ We also showed that the inverse of DFT can be written as (1/N) $\underline{D}_N * \underline{X}$. We shall demonstrate these matrix representations today and find out how these are useful. One point that I wish to emphasize here is that if X(k), k = 0 to N – 1, are given and you want a denser representation, i.e. you want the in between samples, then you can use the interpolation formula; you can also do it differently, using DFT only.

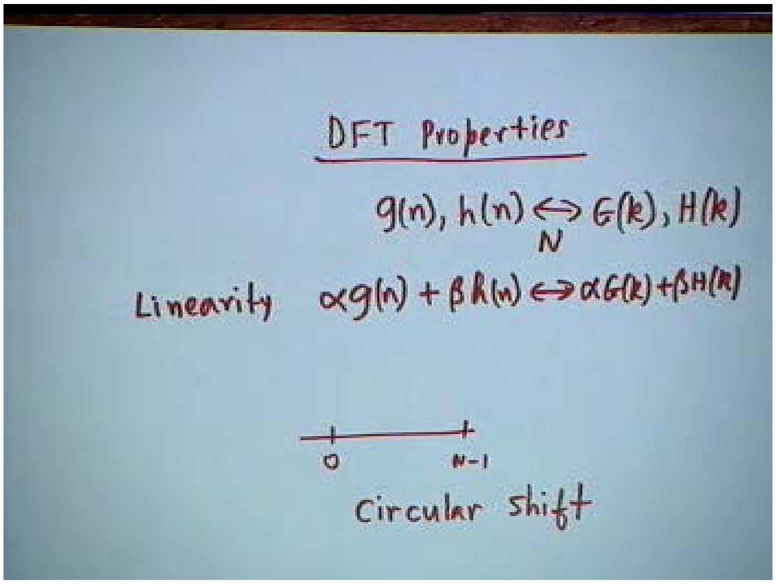
(Refer Slide Time: 04:43 – 07:09)



Suppose your x(n) exists from n = 0 to N – 1 and you wish to compute X(k) for k = 0 to N – 1. This is the usual DFT but suppose you want to compute X(k) not at 2pi/N intervals but smaller intervals; that is what we mean by interpolation. Suppose instead of 2pi/N intervals, we wish to compute at intervals of 2pi/M, where M is much greater than N. if N is 128, may be you wish to compute at 1024 points, that is M may be 1024. Then it will be very dense representation of the spectrum and the envelope shall represent, almost accurately, $X(e^{jomega})$.

What you can do is to append the necessary number of 0s to x(n) such that its length becomes M. You have $x_{0, …..}$ x_{N-1}, then you add 0, 0, 0 up to M – 1. Take this sequence instead of the original sequence; the complexity of computation does not increase thereby because 0 multiplied by anything is 0. But if you compute the M point DFT of this extended sequence then you shall get samples of X(k) at intervals of 2pi/M. This is an alternative method to using interpolation; you artificially extend the length of the sequence by adding 0s and then compute M point DFT.

(Refer Slide Time: 07:23 – 04:21)



DFT Properties

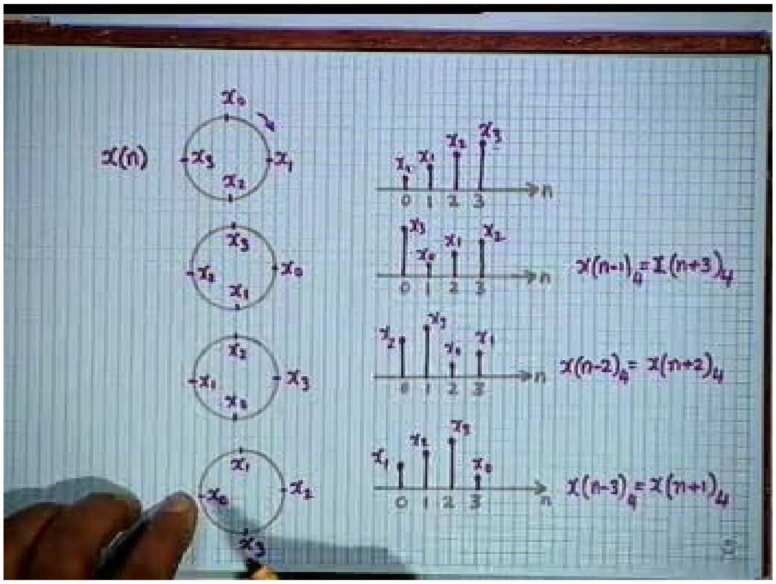
$g(n), h(n) \overset{}{\underset{N}{\Longleftrightarrow}} G(k), H(k)$

Linearity $\alpha g(n) + \beta h(n) \Longleftrightarrow \alpha G(k) + \beta H(k)$

Circular shift

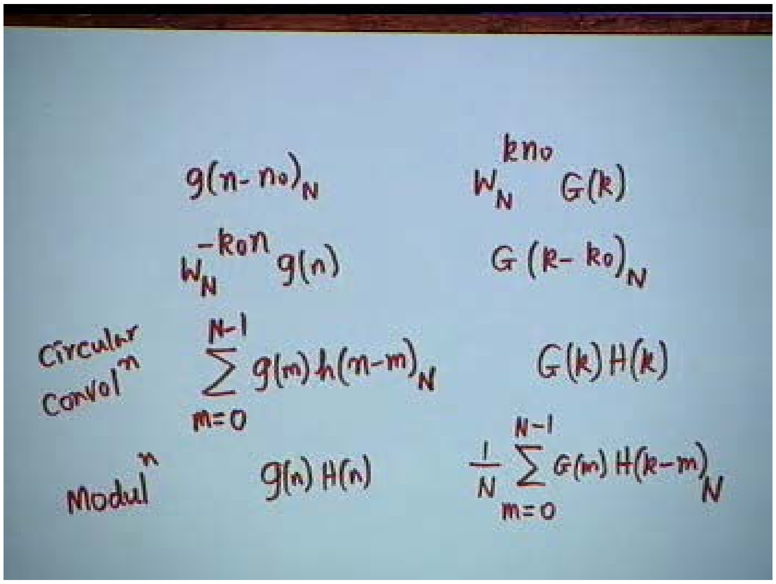
Next we go to DFT property. Once again, the proofs shall be left to you but I shall explain the steps and their significance while discussing the properties. Consider two sequences g(n) and h(n) whose N-point DFTs are G(k) and H(k), respectively. The first property, as is true about all transformations, is linearity: alpha g(n) + beta h(n) shall give you alpha G(k) + beta H(k) as the DFT. Now in ordinary Fourier transforms we showed that if the sequence is delayed by n_0 samples, then the transform is multiplied by $e^{-\text{ jno(omega)}}$. Here, if you have samples between 0 and N – 1 and you delay it by one or more samples, then you go out of the range of vision; therefore you have to define some other kind of delay or some other kind of shift, such that even after that shift the sequence remains in the range of vision. So we define a new concept, called circular shift. Let us understand how circular shift occurs. It is like a circle along which a group of people are sitting; when one person shifts to the right or to the left, then the previous person comes and occupies the vacant position. This is called circular shift. I shall explain this with the help of a figure. The figure that I have prepared is the following and I shall project it by parts.

(Refer Slide Time: 10:18 – 14:20)



Look at this diagram, I have taken a very simple sequence of length 4, i.e. x_0, x_1, x_2, x_3 which are of arbitrary amplitudes. Look at this circle from which the name circular shift comes. I draw a circle and put them at equal intervals, along the circumference. The shift we are talking of is the clockwise shift, i.e. shift to the right. Now if I shift this sequence by one sample then x_0 will come in the position of x_1, x_1 will come to the position of x_2 and so on. This is what we mean by circular shift. The successive diagrams will show delay by one sample at each step. Now let us go to the Mathematics. In $x(n – 1)$, if $n = 0$, it becomes $x(– 1)$. There is no $x(– 1)$, so you take $(– 1)$ modulo 4, i.e. add 4 to – 1, then $x(4 – 1)$ is $x(3)$ so $x(3)$ comes at the position earlier occupied by x_0. And then the other 3 samples each shift by one position to the right. So in terms of circular shift, x_0 goes to this position 1 and x_3 is driven out of position 3 to occupy the vacant position. This is what we mean by circular shift. Now let us see a second example: $x(n – 2)_4$ is $x(n – 1)_4$ delayed by one sample and is shown in the next figure; similarly for $x(n – 3)_4$. Our range of vision remains from 0 to 3; otherwise DFT does not make sense: DFT freezes the range of vision from 0 to $N – 1$. Finally $x(n – 3)_4 = x(n + 1)_4$. x_0, x_1, x_2 and x_3 go in a merry-go-round fashion in circular shift or modular shift. The argument of x has to be taken modulo 4, and then your range of vision shall remain frozen between 0 and 3.

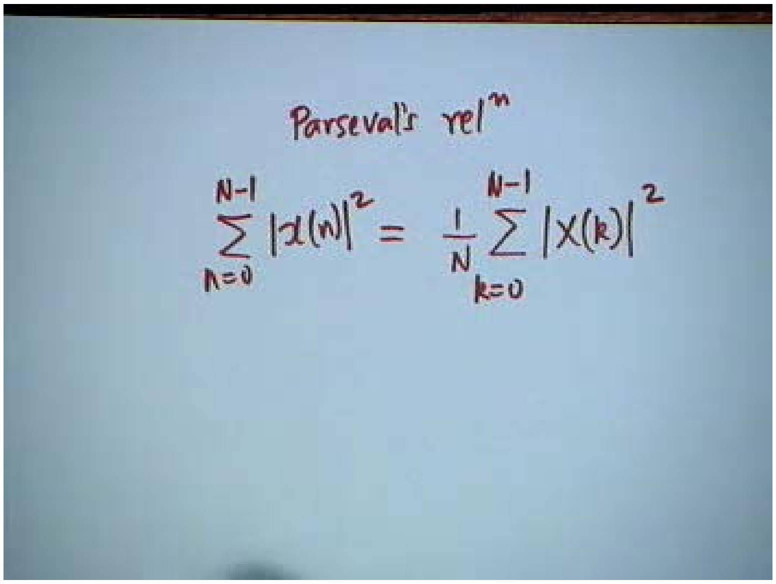
(Refer Slide Time: 14:29 – 18:02)



	$g(n-n_0)_N$	$W_N^{kn_0}\ G(k)$
	$W_N^{-k_0n}\ g(n)$	$G(k-k_0)_N$
Circular Convol^n	$\sum_{m=0}^{N-1} g(m)h(n-m)_N$	$G(k)H(k)$
Modul^n	$g(n)\ H(n)$	$\frac{1}{N}\sum_{m=0}^{N-1} G(m)\ H(k-m)_N$

Let us make a circular time shift by n_0 samples. What are the possible values of n_0? n_0 must remain within the range of vision. The DFT of $x(n - n_0)_N$ shall be W_N^{kn0} G(k). If you have a circular frequency shift, that is you have $W_N^{- k0n}$ g(n) then the DFT simply becomes $G(k - k_0)$; it follows from the definition. Then there is convolution. You know that in linear convolution, if you take the convolution of N samples, with another sequence which also has N samples, then the total length of the linear convolution shall be 2N – 1. In DFT applications, your range of vision has to be restricted to 0 to N – 1, you cannot go beyond that, and therefore you have to define what is called circular convolution, sometimes also called periodic convolution. Periodic convolution is summation $[g(m)h(n - m)_{N}]_{,}$ where the argument n – m had to be taken with modulo N. Then you make sure that the range of vision is restricted between 0 and N – 1. The shift has to be a circular shift, and m goes from 0 to N – 1. Then the DFT shall be simply the product of G(k) and H(k). So circular convolution gives rise to a multiplication in the frequency domain, exactly like Laplace transform or Fourier transform. Next comes the modulation property i.e. we consider the sequence g(n)h(n). In Fourier transform, it was integration but here it shall be a summation. The inverse DFT is also a summation unlike the previous case where it was integration. The Fourier transform of g(n)h(n) is (1/N) summation $[G(m)H(k - m)_N$ where m goes from 0 to N – 1. Clearly this is a convolution but a circular convolution. What it imposes on the mechanics of convolution is what we shall illustrate with a few interesting diagrams.

(Refer Slide Time: 18:10 – 14:20)

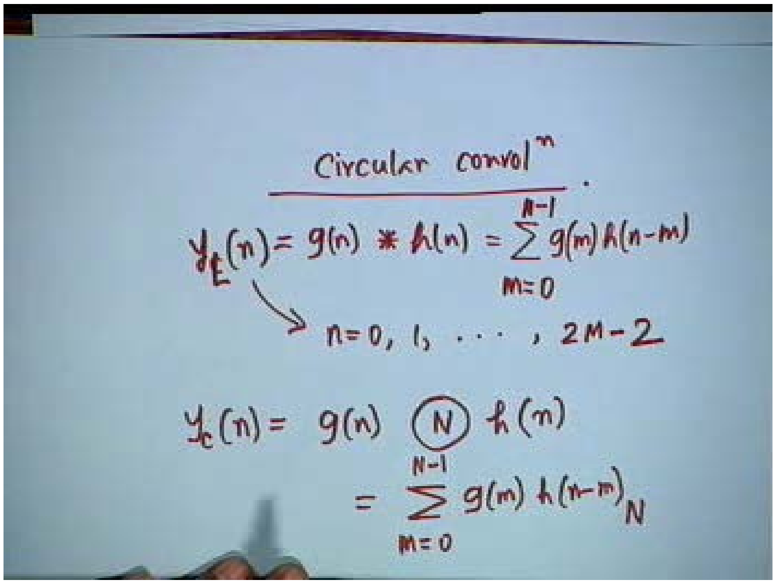


Parseval's rel^n

$$\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$$

Finally the Parseval's relation for energy is also valid here. That is, the energy in the time domain: given by the summation magnitude of $(x(n))^2$, where n = 0 to N – 1 is exactly equal to (1/N) summation magnitude of $X(k)^2$ where k goes from 0 to N – 1. The energy in the time domain is the same as the energy in the frequency domain. In Fourier transform this involved integration, here it is a summation. So the DFT makes life simple. You do not have to integrate; it is all summation. Mechanization in the computer is much easier in this case.

(Refer Slide Time: 19:25 – 21:14)



Circular convol^{n}.

$$y_L(n) = g(n) * h(n) = \sum_{m=0}^{N-1} g(m)h(n-m)$$

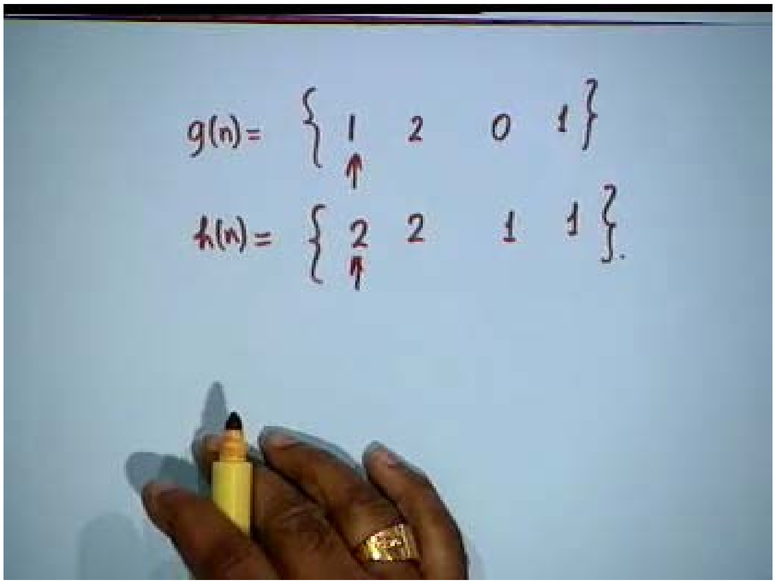
$$\searrow \quad n = 0, 1, \cdots, 2N-2$$

$$y_c(n) = g(n) \textcircled{N} h(n)$$

$$= \sum_{m=0}^{N-1} g(m)\, h(n-m)_N$$

Now let us understand what we mean by circular convolution. If you have two sequences g(n) and h(n) then the linear convolution $y_L(n)$, denoted by a *, is simply given by $y_L(n) = g(n)*h(n) =$ summation g(m)h(n – m), where m goes from 0 to N – 1 because both g(n) and h(n) are finite sequences of length N and $y_L(n)$ has to be computed from n = 0 to 2N – 2. The length has to be 2N – 1. This is linear convolution. The circular convolution, of g(n) and h(n) shall be denoted by $y_c(n)$. To emphasize that it is N point and circular convolution, instead of a star symbol, we simply include N within a circle. The expression is written like this: $y_c(n) = g(n)$ (N) h(n). By definition, this is equal to summation $[g(m)h(n – m)_N]$, where m goes from 0 to N – 1. To understand this a little more deeply, let us consider a simple example of two 4 point sequences (the number of points has to be the same in circular convolution but this is not necessary in linear convolution).

(Refer Slide Time: 21:37 – 22:20)

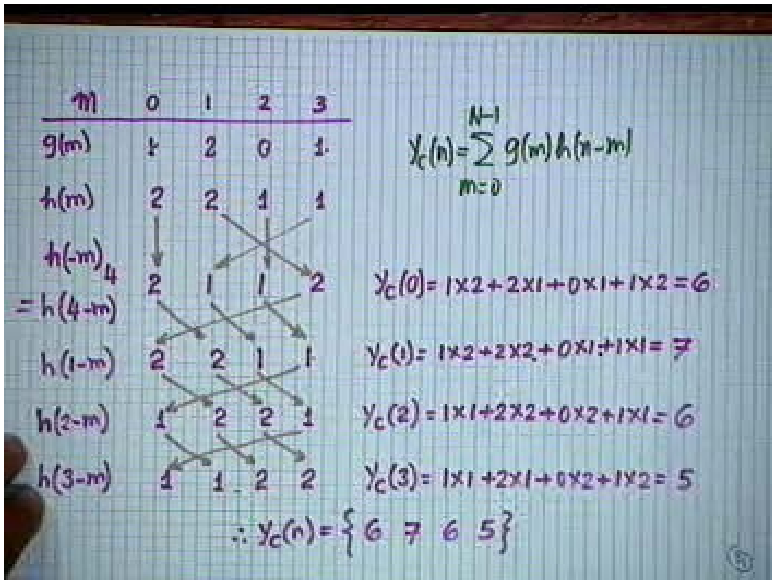


$$g(n) = \{\underset{\uparrow}{1} \quad 2 \quad 0 \quad 1\}$$

$$h(n) = \{\underset{\uparrow}{2} \quad 2 \quad 1 \quad 1\}.$$

The example that we take is g(n) = {1 2 0 1} consisting of four samples, the first sample being at n = 0, as indicated by an arrow. The sequence h(n) also has 4 points. The samples are: h(n) = {2 2 1 1}. Here also the arrow indicates h(0). The last sample must be at n = N – 1. Now let us see what is meant by circular convolution. We shall discuss several methods. You make your choice of the method which appears simple to you. Now look at this slide carefully. I project it in parts and steps.

(Refer Slide Time: 22:55 – 27:00)



m	0	1	2	3
g(m)	1	2	0	1
h(m)	2	2	1	1
$h(-m)_4$ = h(4-m)	2	1	1	2
h(1-m)	2	2	1	1
h(2-m)	1	2	2	1
h(3-m)	1	1	2	2

$$y_c(n) = \sum_{m=0}^{N-1} g(m)h(n-m)$$

$$y_c(0) = 1\times2 + 2\times1 + 0\times1 + 1\times2 = 6$$

$$y_c(1) = 1\times2 + 2\times2 + 0\times1 + 1\times1 = 7$$

$$y_c(2) = 1\times1 + 2\times2 + 0\times2 + 1\times1 = 6$$

$$y_c(3) = 1\times1 + 2\times1 + 0\times2 + 1\times2 = 5$$

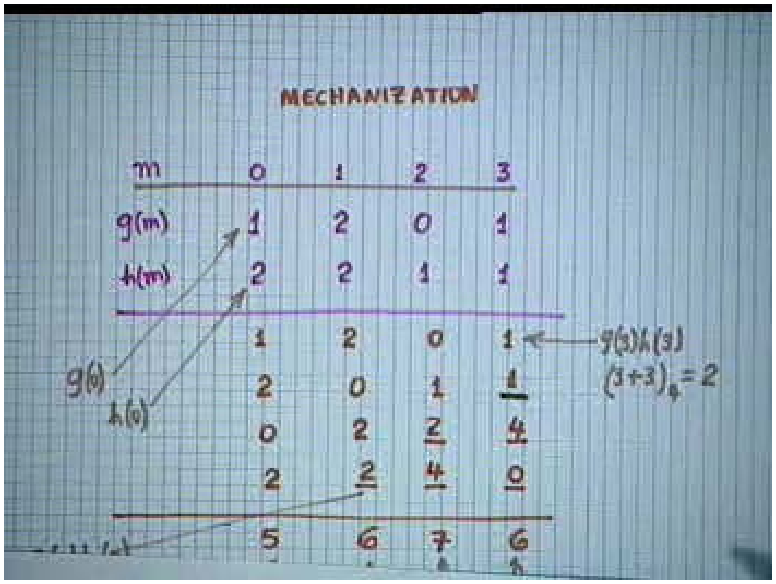
$$\therefore y_c(n) = \{6 \; 7 \; 6 \; 5\}$$

I have indicated m on the first line: 0, 1, 2, 3, g(m) in the second line: 1, 2, 0, 1 and h(m) in the third line: 2, 2, 1, 1. Now I have to find out h(– m), then shift it by one sample at each step. Now h(– m), is obtained by flipping back h(m) with m = 0 as pivot. In other words, we shall start from m = – 3; but we cannot do that, we must start from m = 0. So we take h(– m) modulo 4 which means that h(0) remains unchanged, but when you get m = – 1, the argument would change to m = 4 – 1, that is 3. This 1 in the position under m = 3 of h(m) comes here as indicated by an arrow. Then h(– 2) is same as h(2) so this one comes here directly and h(– 3) is the same as h(1) and therefore this comes here. I have taken modulo 4 so that my range of vision is 0 to 3. If you recall this summation, namely summation g(m) $h(n – m)_N$, m = 0 to N – 1, we shall have $y_c(0)$ by multiplying g(m) by h(– m) sample by sample and adding them. The sum is 6, which is $y_c(0)$. What happens next? For $y_c(1)$, I have to find h(1 – m), 1 – m means the whole sequence is shifted to the right by one sample and the last sample goes back to occupy the vacant position at n = 0. So 2 goes here, 1 goes here, 1 goes here and this 2 is running back, which I have indicated by this arrow. Since we have found out h(1 – m), you can now calculate $y_c(1)$. What you do is this: 1 multiplied by 2, 2 multiplied by 2, 0 multiplied by 1 and 1 multiplied 1, and the sum is 7. This process continues for another two steps. We have already found out two samples; another two samples have to be found. For h(2 – m), 2 comes here, 2 comes here, 1 comes here and this 1 goes back to the 0th position. I now multiply g(m) by h(2 – m) sample by sample: 1 times 1, 2 times 2 then 0 times 2 and 1 times 1, the sum is 6. Finally compute the last sample; once you do the first step, then the other steps are very simple. All we have to do is to repeat this arrow pattern. The value is 5. Finally, my $y_c(n)$ becomes {6, 7, 6, 5} with four samples. How do you mechanize this?

(Refer Slide Time: 27:27 – 27:45)

I gave you a trick for linear convolution. There should be mechanization here also. The trick is the following, let me explain in steps. What we do is the following, I have written m, g(m) and h(m) and I have identified which are g(0) and h(0), in order to be able to keep track of which sample will be what.

(Refer Slide Time: 28:32 – 31:56)



MECHANIZATION

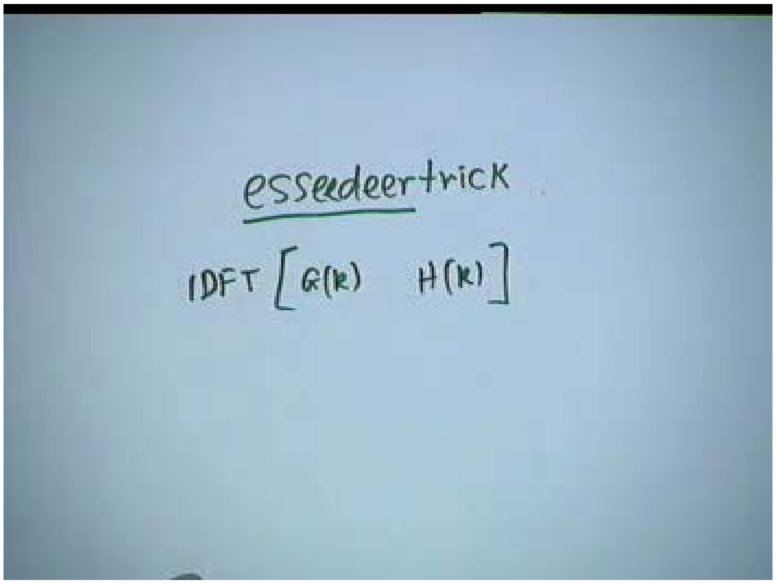
m	0	1	2	3
g(m)	1	2	0	1
h(m)	2	2	1	1
	1	2	0	1
	2	0	1	1
	0	2	2	4
	2	2	4	0
	5	6	7	6

Then I do the same thing as I did in linear convolution. That is I multiply second and third rows sample by sample exactly like arithmetic multiplication, but without a carry. So 1 times 1 is 1, 1 times 0 is 0, 1 times 2 is 2 and 1 times 1 is 1; this is the first step. In the second step if you recall in arithmetic multiplication or linear convolution we have left this place vacant, we leave it vacant now also so we put a dash here and then go for the second one. That is 1 times 1 is 1 so this is 0, 1 times 2 is 2 then 1 times 1 is 1. We do not write the last sample to the left, it runs and occupies the vacant position we left in the rightmost column. In the next step, we leave two vacant places, then we multiply by 2; 2 times 1 is 2, 2 times 0 is 2 then 2 times 2 is 4. This 4 runs and occupies the first available vacant position. Then 2 times 1 goes to the next vacant position. You see that we have only created four samples in each step. In the last one, you leave three vacant spaces. Here 2 times 1 is 2 then 2 times 0 is 0, it comes here, 2 times 2 is 4 it comes here, 2 times 1 is 2 and it comes here. Now you add each column. Now there is a problem, we get {5 6 7 6} our result was {6 7 6 5}. You must identify which one is the 0th sample, then the rest will be clear.

You recall that this was g(0) and this is h(0). This column will give $y_c(0)$ because it contains g(0) h(0). The value is 6. Once you identify this one, then the identification problem is solved. You have $y_c(1)$ as the next right sample and so on. Suppose you have made a mistake in identifying this; how do you verify? Go back to the first sample; here is the first product. What is this? This is a multiplication of g(3) and h(3). The sum of the indices should be equal to the index of the output. Now 3 + 3 is 6 we do not have a 6th sample but 6 modulo 4 is 2 so this must be $y_c(2)$. Then you can verify whether you have done it correctly or not.

As I told you we can compute circular convolution by various other methods. One of the methods is that you find G(k) and H(k), multiply the two and then take IDFT of G(k) H(k).

(Refer Slide Time: 32:24 – 32:38)

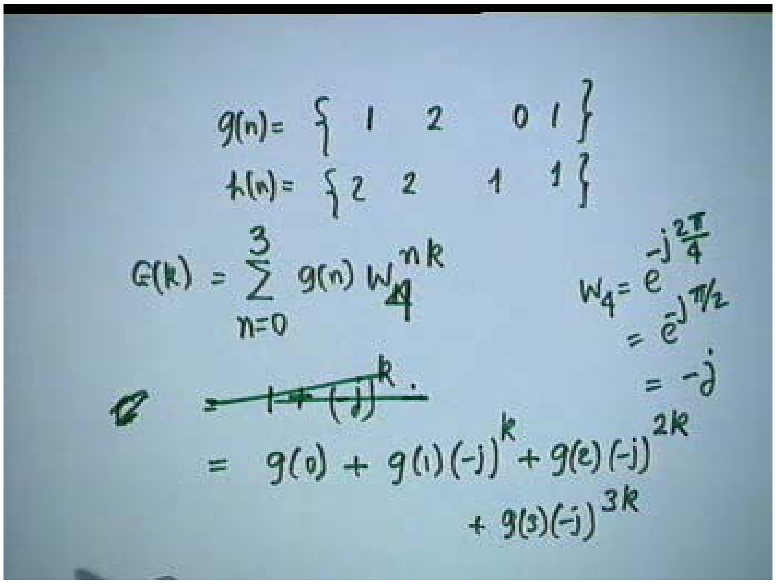


esseedeertrick

IDFT $[G(k) \quad H(k)]$

We will compute the same example by this method.

(Refer Slide Time: 32:48 - 35:10)



$g(n) = \{1 \quad 2 \quad 0 \quad 1\}$

$h(n) = \{2 \quad 2 \quad 1 \quad 1\}$

$$G(k) = \sum_{n=0}^{3} g(n) W_4^{nk}$$

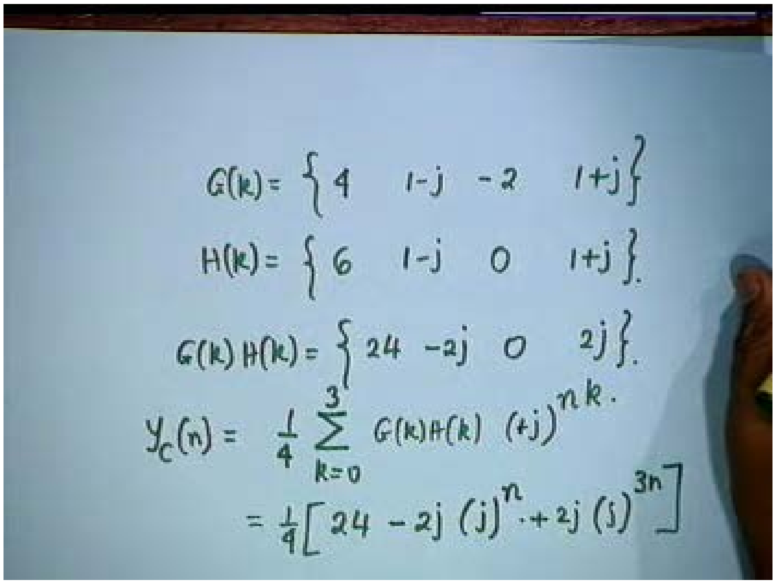
$$W_4 = e^{-j\frac{2\pi}{4}} = e^{-j\pi/2} = -j$$

$$\cancel{= 1 + (-j)^k .}$$

$$= g(0) + g(1)(-j)^k + g(2)(-j)^{2k} + g(3)(-j)^{3k}$$

Our sequences are g(n) = {1 2 0 1} and h(n) = {2 2 1 1}. The definition is that G(k) shall be summation $g(n)W_N^{nk}$, n goes from 0 to 3 and N is 4. Let us see what is W_4, $W_4 = e^{-j2pi/4} = -j$. I am calculating G(k) which is equal to g(0) multiplied by W_4^0 + g(1) multiplied by $-j^k$ + g(2) multiplied by $(-j)^{2k}$ + g(3) multiplied by $(-j)^{3k}$. Then you have to find out G(k) by putting the values of k. I leave this algebra to you.

(Refer Slide Time: 35:24 – 37:41)



$$G(k) = \{4 \quad 1-j \quad -2 \quad 1+j\}$$

$$H(k) = \{6 \quad 1-j \quad 0 \quad 1+j\}.$$

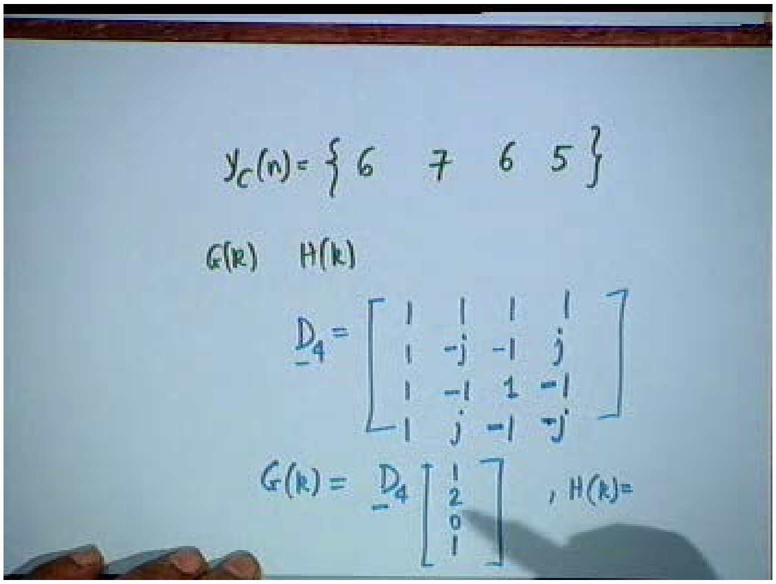
$$G(k)\,H(k) = \{24 \quad -2j \quad 0 \quad 2j\}.$$

$$y_c(n) = \frac{1}{4} \sum_{k=0}^{3} G(k)H(k)\,(+j)^{nk}.$$

$$= \frac{1}{4}\left[24 - 2j\,(j)^{n} + 2j\,(j)^{3n}\right]$$

Please verify whether my result is right or wrong. My result is G(k) = {4 1 – j – 2 1 + j}. In a similar manner I calculate H(k) = {6 1 – j 0 1 + j}. The product of G(k)h(k) shall be the multiplication of the corresponding samples, the final result being G(k) H(k) = {24 – 2j 0 2j}. Then $y_c(n)$ shall be IDFT of this. That is $y_c(n)$ = (1/4) summation $[G(k)H(k)\ (+\ j)]^{kn}$ where k = 0 to 3. Now you expand this, this will become: $y_c(n) = (1/4)[24 – 2j\ (j)^{n} + 2j\ (j)^{3n}]$. Now put n = 0, n = 1, n = 2 and n = 3 and compute this sequence.

(Refer Slide Time: 37:53 – 41:35)



$y_c(n) = \{6 \quad 7 \quad 6 \quad 5\}$

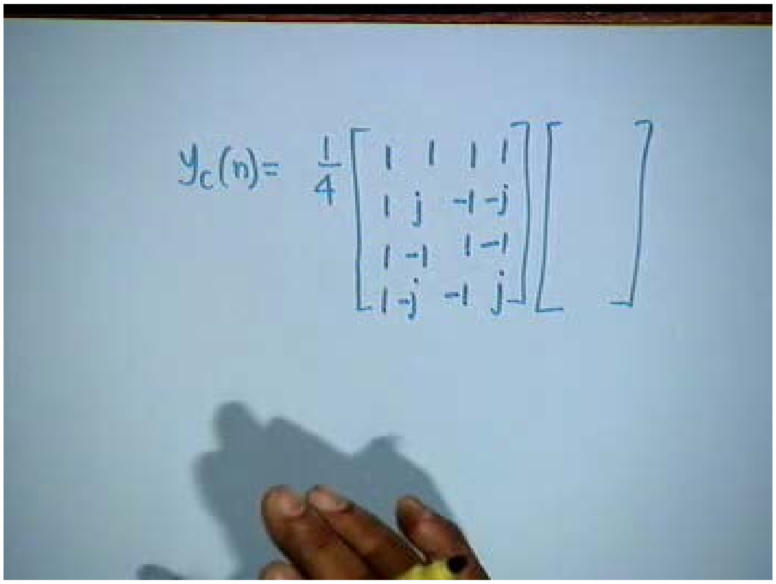
G(k) H(k)

$$\underline{D}_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -j & -1 & j \\ 1 & -1 & 1 & -1 \\ 1 & j & -1 & -j \end{bmatrix}$$

$$G(k) = \underline{D}_4 \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}, \ H(k) =$$

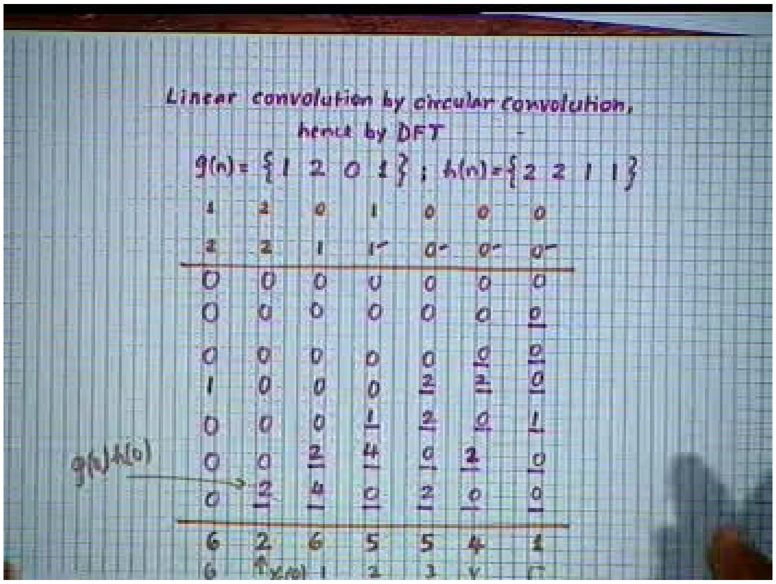
Final result that I get is: $y_c(n) = \{6\ 7\ 6\ 5\}$. I have given you three methods: One is graphical, one is a trick and the third is multiplication of G(k) by H(k) and taking inverse DFT. Now this is where the matrix representation comes to help. I find what D_4 is; D_4 will occur when finding G(k), finding H(k) and also in finding inverse DFT because G(k)H(k) is also of length 4. So one matrix D_4 is good enough. In D_4 matrix, the first row is 1 and the first column is also 1; then you have W_N^1 so $W_4 = -j$. The next one would be $(-j)^2 = -1$, then $(-j)^3 = +j$. You see, we cannot make a mistake here; it is very easy to do this, instead of summation form. Then the next row would have 1, $(-j)^2 = -1$, $W_N^4 = +1$, and finally, $W_N^6 = -1$. What was your third row? The third row is 1, $W_N^{2,}$ W_N^4, W_N^6 and so on because your k = 2. The next one that is in the third row, this will be W_N^3 which is j, then – 1 and then – j (this is W_N^3), then W_N^6 and finally, W_N^9 which is the product of these two. This you should be able to write without a thought. So we get G(k) = D_4 multiplied by the column matrix g(n) where elements are 1 2 0 1. Similarly you find H(k) with the same matrix D_4. Finally, you find out G(k)H(k).

(Refer Slide Time: 41:47 - 42:57)



Next step is to find out $y_c(n)$: for $y_c(n)$, you need D_4^*. For this, all you have to do is the following: in D_4 wherever – j occurs you put + j. The first is 1 1 1 1, then 1, + j, – 1, – j, then 1, + 1, – 1, – j, then 1, – 1, 1 – 1 and then 1 – j – 1 + j. This matrix is multiplied by the samples of G(k)H(k) put in a column matrix. You see, this computation is much easier to do, than writing in the summation form and keeping track of powers j. Once you are able to write $D_{4,}$ you are absolutely safe. The next point is, what is the use of circular convolution? The use is in computing linear convolution. You can compute linear convolution by using circular convolution. In circular convolution, if the number of samples is large, you do not compute like we did here; you may not be able to compute by hand. You have to take help of computer. What help? It is to use DFT. If the number of samples is large then you take the DFT of both the sequences, multiply them and find IDFT. But how do we carry out linear convolution? In linear convolution, as you know, the number of samples in individual sequences may not be the same. And the total number of samples in the output of linear convolution is $N_1 + N_2 – 1$. Therefore we apply the same trick that has been applied so far that is we pad 0s. I have enunciated this by means of a slide here which I will project gradually.

(Refer Slide Time: 44:41 - 47.01)



Linear convolution by circular convolution, hence by DFT

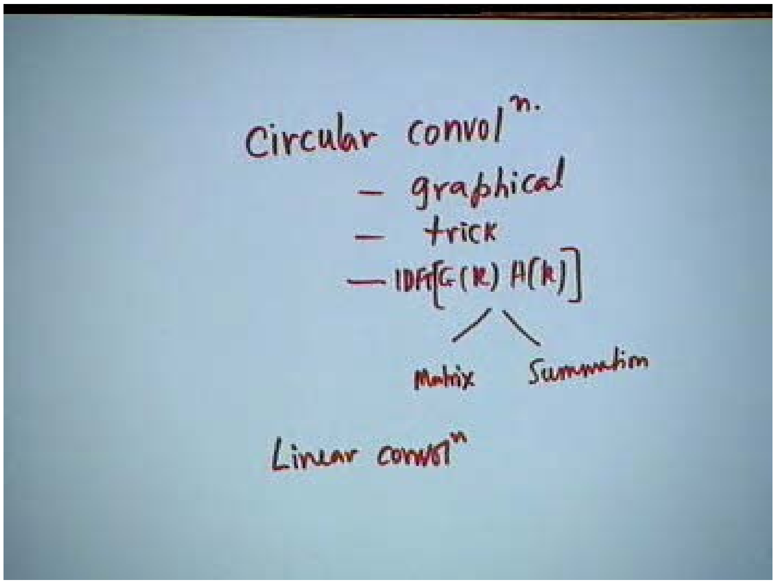
g(n) = {1 2 0 1} ; h(n) = {2 2 1 1}

1	2	0	1	0	0	0
2	2	1	1	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0
1	0	0	0	2	2	0
0	0	0	1	2	0	1
0	0	2	4	0	2	0
0	2	4	0	2	0	0
6	2	6	5	5	4	1
6	y(0)	1	2	3	4	5

g(0)h(0)

I said I have to compute linear convolution of the sequence g(n) = {1 2 0 1} with h(n) = {2 2 1 1} I know that my output shall go from n = 0 to 8 – 1 = 7. The length of the output shall be 7 therefore go from 0 to 6. The length of the output shall be 7 therefore we must have seven samples in both the input sequences. If you want to compute by circular convolution, number of samples should be the same. So I make both of them of length 7 by padding 0s. Then I compute the circular convolution by the trick I discussed earlier. If the number of samples is large then I shall use FFT. But if the samples are manageable like this, then you see how we can do it by mechanization. Here each sequence has 7 samples, so the output of circular convolution also has seven samples. My final result is here. And then to find out which one is my $y_c(0)$ I find out where g(0)h(0) occurs, addition of samples in this column shall give you $y_c(0)$. This will be followed by n = 1 2 3 4 5, and you come back to the extreme left sample, which must correspond to n = 6. So I can compute linear convolution by using circular convolution.

(Refer Slide Time: 47.28 - 49.06)



Circular convol$^{n.}$

- graphical
- trick
- IDFT[G(k) H(k)]

Matrix Summation

Linear convol^{n}

In summary we have found various methods for computing circular convolution. One is graphical, one is a trick that is basically a mechanization. One of the disadvantages of mechanization is that you lose track of the physical picture. But for an engineer once you have got the concept you should be able to mechanize it, time is important and therefore whatever works is good enough. In circular convolution, as you know, you have to identify which column shall give you the 0th sample and then everything else is clear. Then we said how to find circular convolution as IDFT [G(k)H(k)]. You can do this by two methods: matrix method, which appears to be easier, and the other one is to use the summation. There are 4 methods now. Finally we said that the linear convolution can also be computed by FFT if you padded sufficient number of 0s to the two input sequences which are to be convolved. The number of 0s has to be carefully chosen so that the length becomes equal to the expected length of the result. I think this is a correct point to stop, we shall continue in the next lecture.