Digital Signal Processing & Its Applications

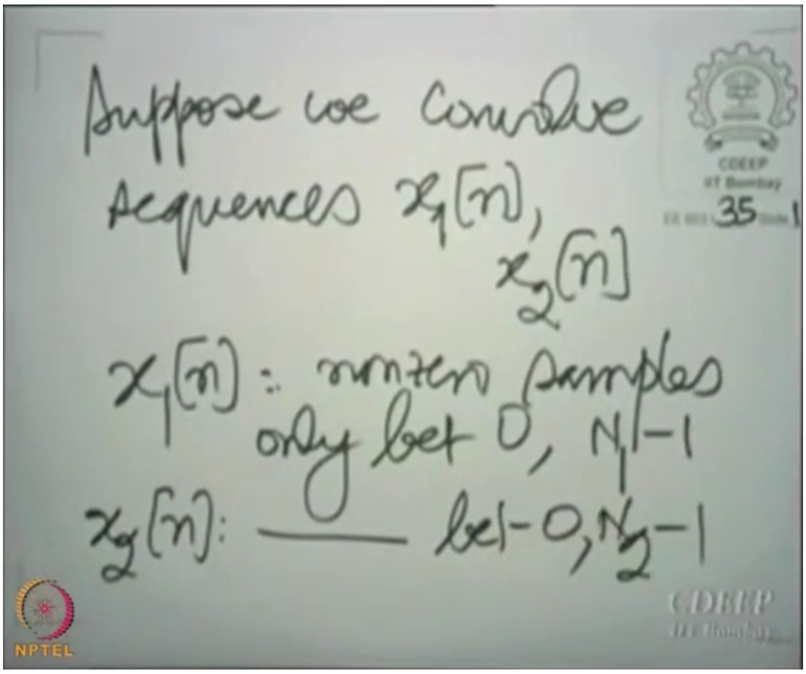
Professor Vikram M. Gadre
Department of Electrical Engineering,
Indian Institute of Technology, Bombay

Lecture No. 35 b

Introduction to the Circular Convolution

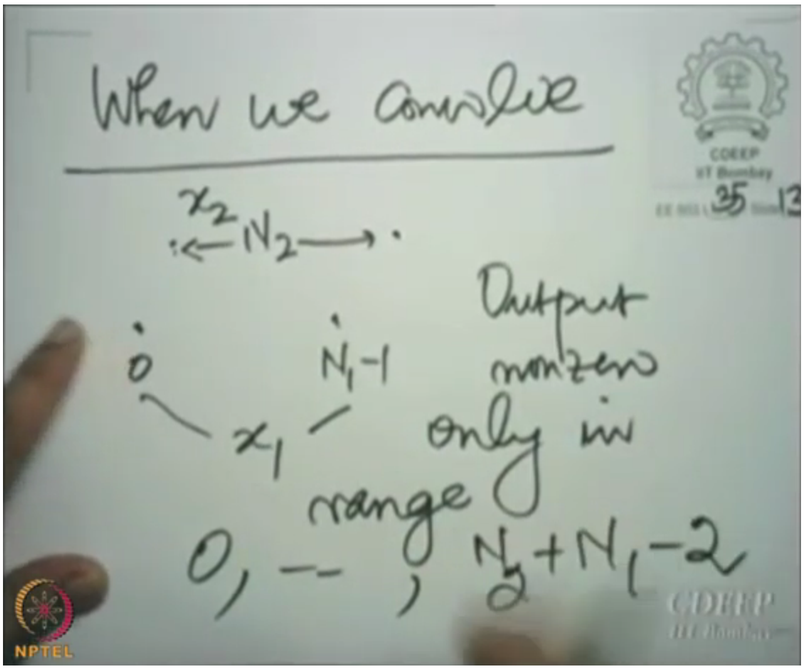
Now let us carry this argument to convolution.

(Refer Slide Time: 00:27)



So suppose we take the convolution of two sequences, $x_1[n]$, which without a loss of generality has non-zero samples only between 0 and $N_1 - 1$, and $x_2[n]$, which has non-zero samples only between 0 and $N_2 - 1$. In other words the so-called "length" of the sequence $x_1[n]$ is N_1 and the so-called length of the sequence $x_2[n]$ is N_2. Now the question is what happens when you convolve these, what length would it have? Let us see.

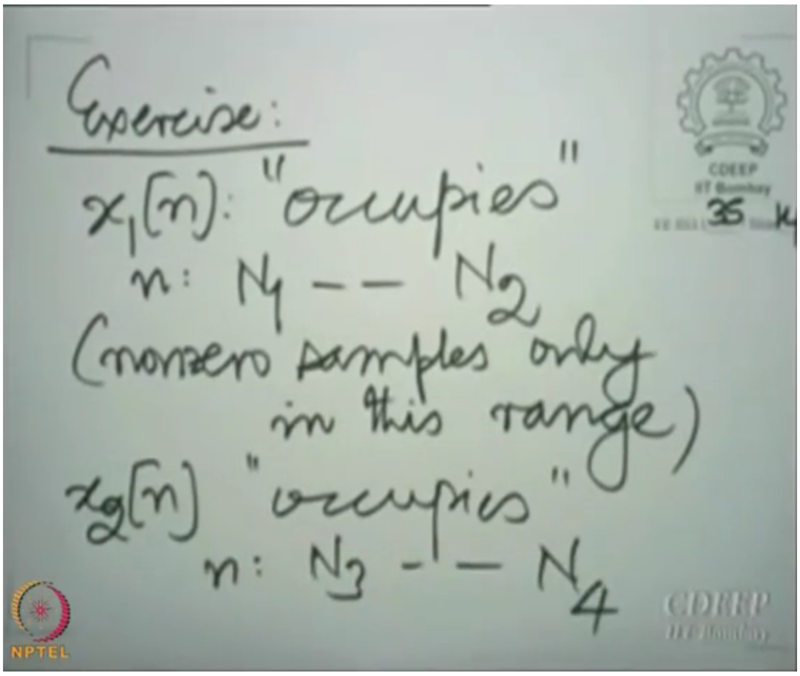
(Refer Slide Time: 01:38)



When we convolve the two sequences, we would have 0 to $N_1 - 1$ samples of x_1, and the sequence x_2 is moved around on this to find the result of the convolution. So $x_2$has N_2 samples which move, and the movement begins with the sample $x_2[0]$ (at the rightmost end of the sequence due to the flipping of the sequence) aligned with the sample $x_1[0]$ (the leftmost sample in the sequence). You see the output starts only when the sample $x_2[0]$ reaches $x_1[0]$ and the output ends when the sample $x_2[N_2 - 1]$ reaches the sample $x_1[N_1 - 1]$.

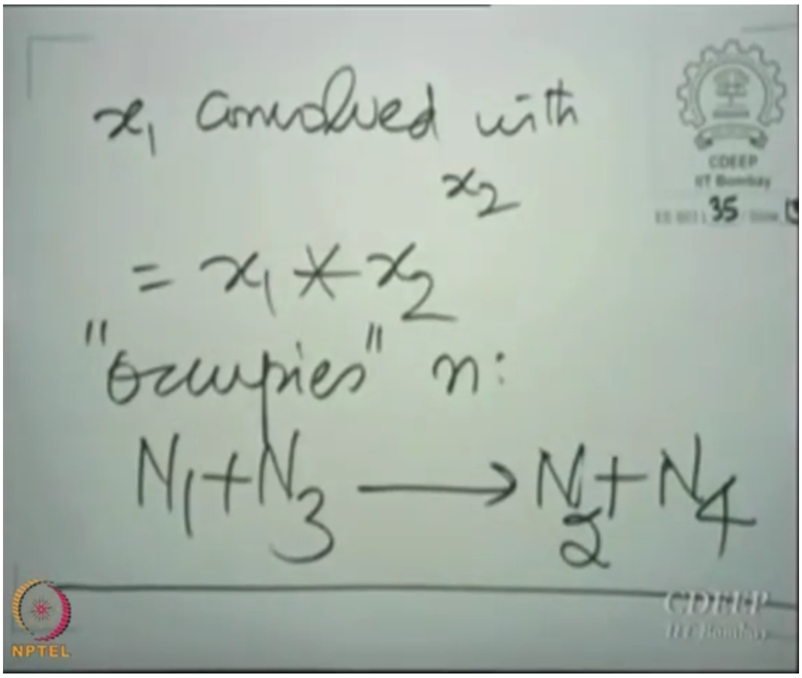
In other words, the output would be non-zero only in the range 0 to $N_1 + N_2 - 2$. You see, for the point $x_2[0]$ to reach the point $x_1[0]$, and for the sequence x_2 to move all the way upto the point when the point $x_2[N_2 - 1]$ reaches $x_1[N_1 - 1]$, you have essentially gone over $(N_1 - 1) + (N_2 - 1)$ samples, that is $N_1 + N_2 - 2$ samples. The movement has been from 0 to$N_1 + N_2 - 2$. In fact in general I leave this to you as an exercise.

(Refer Slide Time: 03:24)



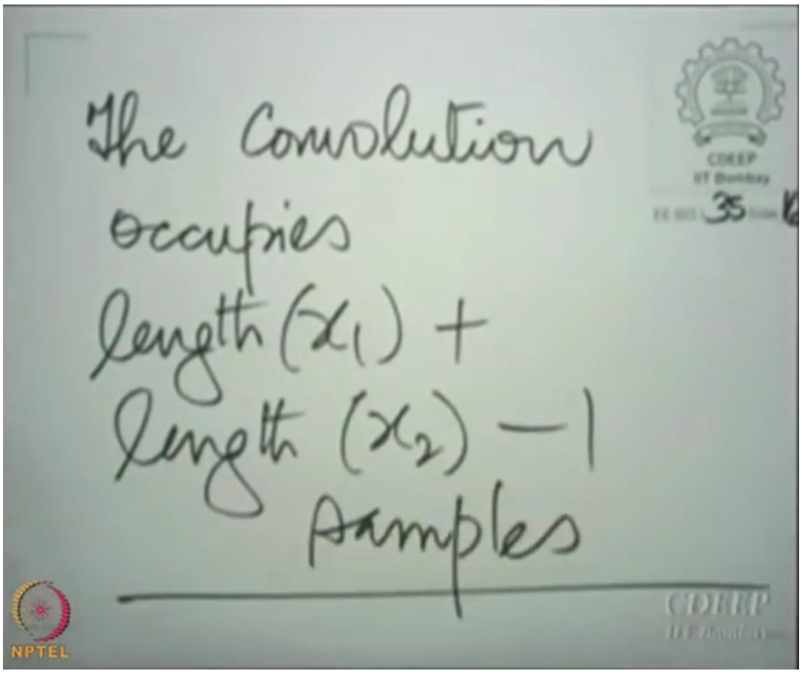
Suppose $x_1[n]$ occupies the range N_1 to N_2, meaning that the non-zero samples are only in this range, and similarly $x_2[n]$ occupies n going from N_3 to N_4. Here it should be noted that when we say occupies, or when we are saying that the non-zero samples are between these numbers we are not saying that all those samples need to be non-zero. We are saying that if there are any non-zero samples they are confined to between N_1 and N_2 for x_1 and between N_3 and N_4 for x_2. Some of the samples in between could also be 0.

(Refer Slide Time: 04:46)



Then if x_1 convolved with x_2, then the result $x_1 * x_2$ occupies $N_1 + N_3$ to $N_2 + N_4$. This is I will leave it to you as an exercise to prove. It is simple, just write down the convolution expression and you would see it in a fairly straightforward way, or you could do it graphically. In fact I would encourage you to use both approaches, the algebraic approach and the graphical approach as well and reaffirm the same result.

(Refer Slide Time: 05:44)

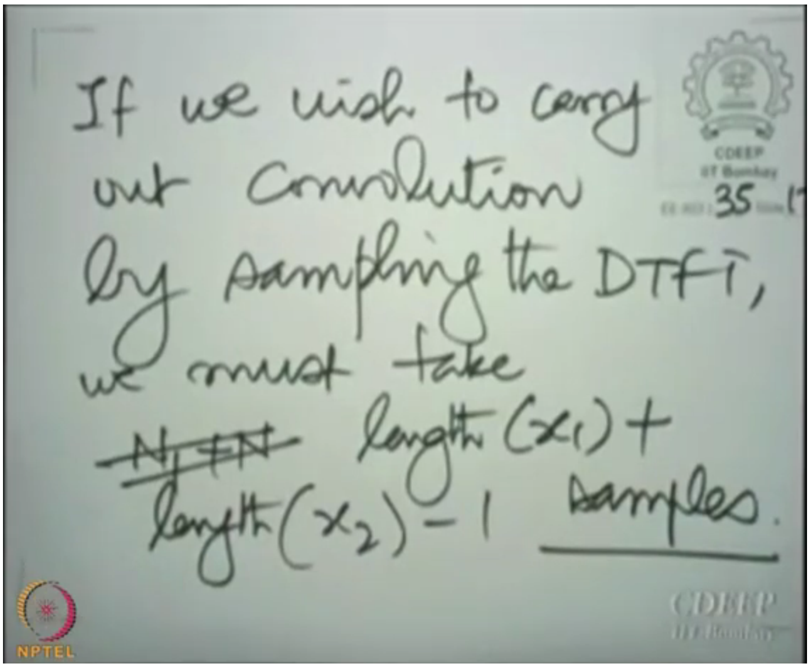


Therefore in a convolution of two sequences with N_1 and N_2 samples respectively, we have a resultant sequence with $N_1 + N_2 - 1$ samples. The convolution occupies a length of $length(x_1) + length(x_2) - 1$ samples.

And this gives us a hint as to how we should deal with convolution in the frequency domain if you want to sample. For the sequence $x_1[n]$ with N_1 samples, it is quite adequate to sample the frequency domain for x_1, i.e. $X_1(\omega)$ with only N_1 samples. And similarly if we wish to avoid time-domain aliasing when only x_2 is involved, it is sufficient to take capital N_2 samples. But to find the convolution of x_1 and x_2, you would want to multiply their discrete-time Fourier transforms and then take an inverse discrete-time Fourier transform.

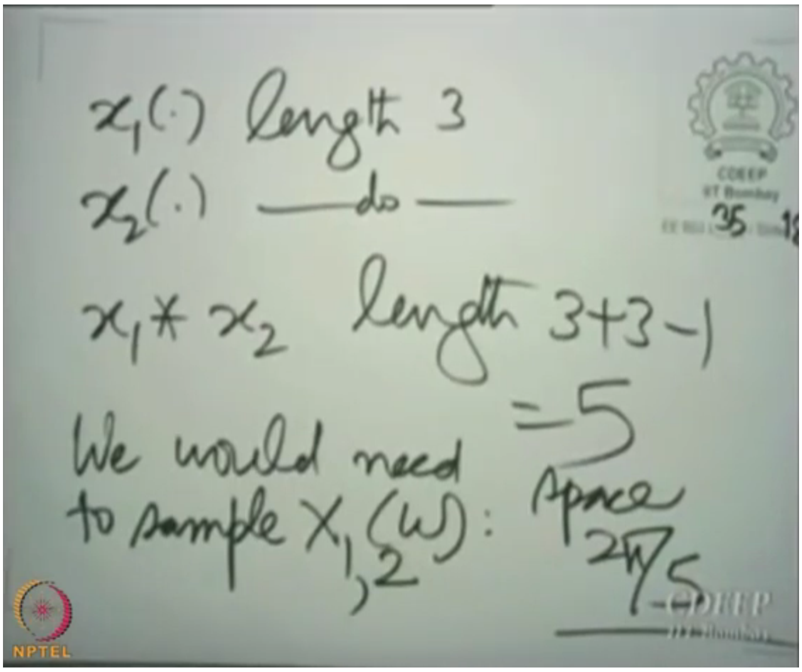
If you wish to use that route, then it is not adequate to sample with only N_1 samples or N_2 samples. You need to sample with $N_1 + N_2 - 1$ samples on all of them. That is very, very important. Let us make a note of that.

(Refer Slide Time: 07:25)



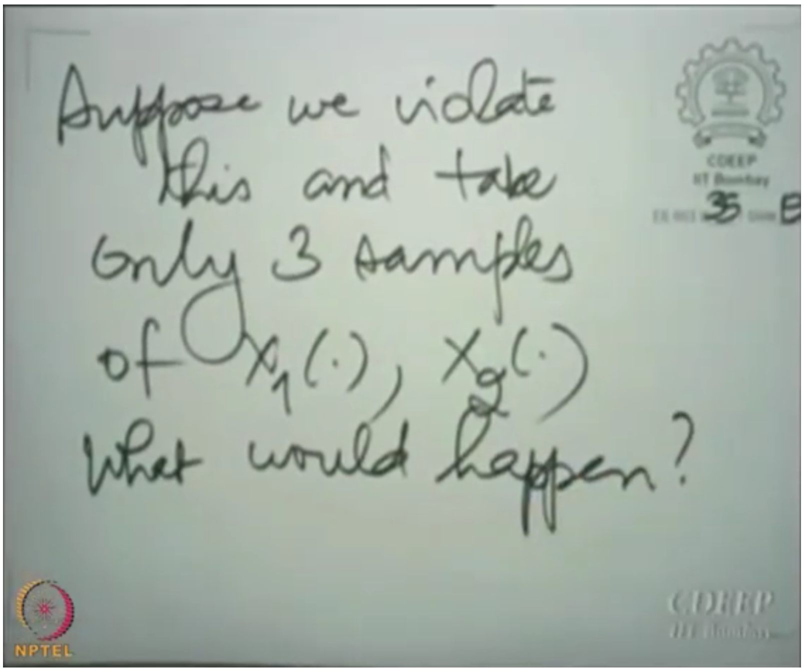
If we wish to carry out convolution by sampling the DTFT, we must take $length(x_1) + length(x_2) - 1$samples in the DTFT. And this is true for all the three, the DTFT of x_1, the DTFT of x_2 and the DTFT of $x_1 * x_2$. You need to work with these many samples.

(Refer Slide Time: 08:29)



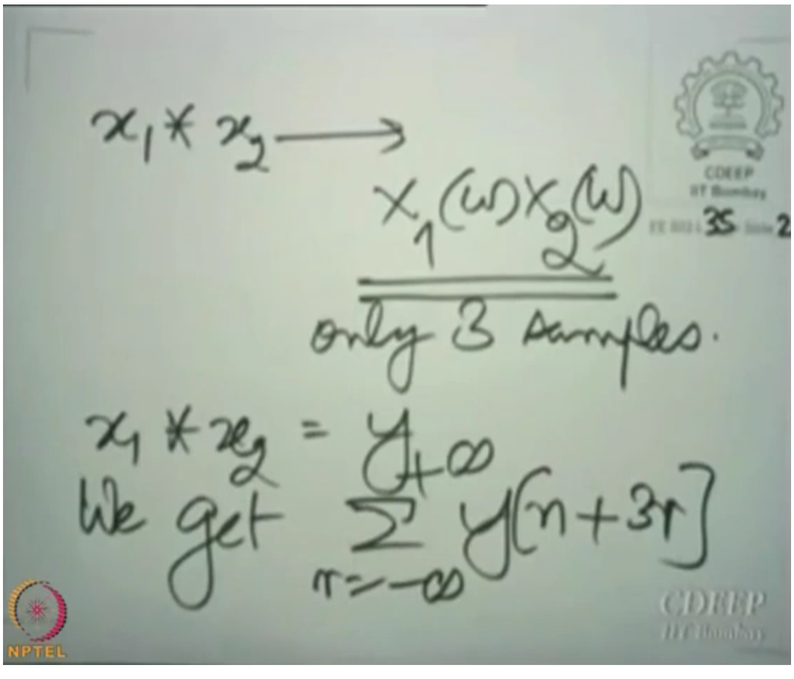
To take an example, suppose we had x_1 of length 3 and similarly for x_2, also of length 3, then $x_1 * x_2$ would be of length $3 + 3 - 1 = 5$. So we would need to sample $X_{1,2}(\omega)$ with a spacing of $\frac{2\pi}{5}$, not $\frac{2\pi}{3}$. We would need to take 5 samples of $X_1(\omega)$ and $X_2(\omega)$ and multiply the 5 samples, sample by sample. But if you do indeed take only 3 samples then what would happen?

(Refer Slide Time: 10:06)



Suppose we violate this, and take only 3 samples of X_1 and X_2, what would happen? That is you try and use only 3 samples.

(Refer Slide Time: 10:23)

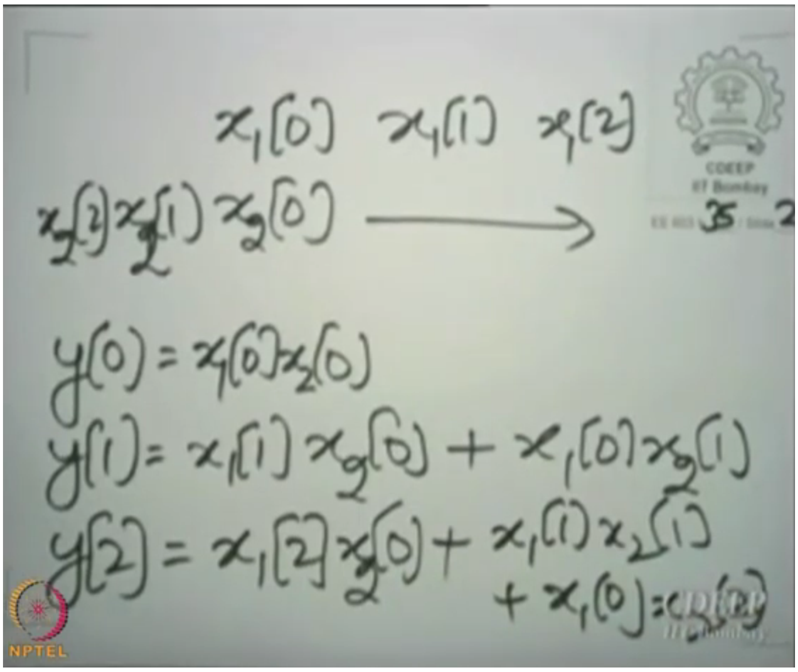


To convolve the two sequences, you would multiply their DTFTs. Let us call $x_1 * x_2[n]$ as $y[n]$. So if you have only 3 samples of the DTFTs, and if you wish to get back $y[n]$ what you are actually going to get is not y shifted by every multiple of 5 samples, but y shifted by every multiple of 3 samples.

So we are going to get:

$$\sum_{r=-\infty}^{+\infty} y[n + 3r]$$

(Refer Slide Time: 11:33)



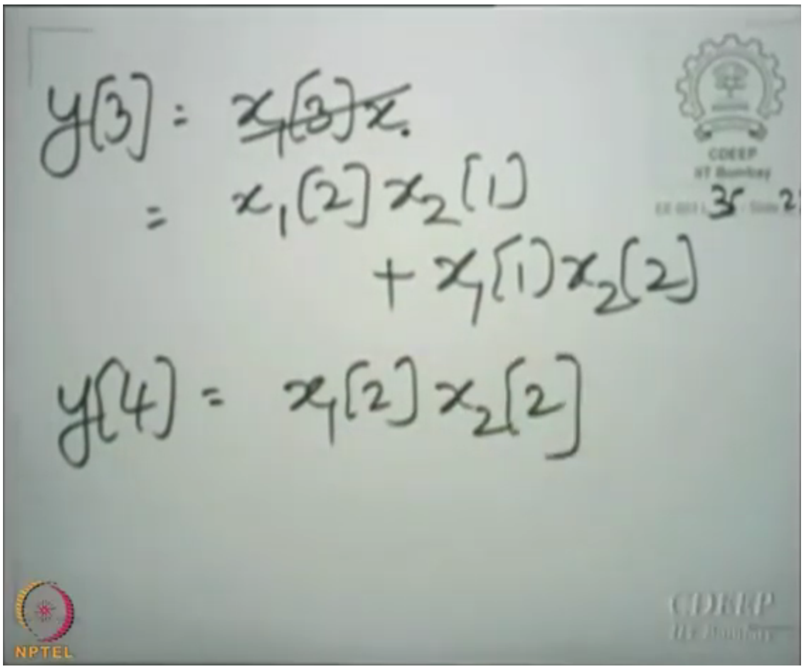
Now let us write down this convolution to understand it better. You see you have $x_1[0]$, $x_1[1]$ and $x_1[2]$, the 3 samples of x_1, which acts as the "platform", and you would have 3 samples of x_2 which move around this "platform", thus acting like the train. So the first three samples of the output are as follows:

$$y[0] = x_1[0]x_2[0]$$

$$y[1] = x_1[1]x_2[0] + x_1[0]x_2[1]$$

$$y[2] = x_1[2]x_2[0] + x_1[1]x_2[1] + x_1[0]x_2[2]$$

(Refer Slide Time: 12:51)

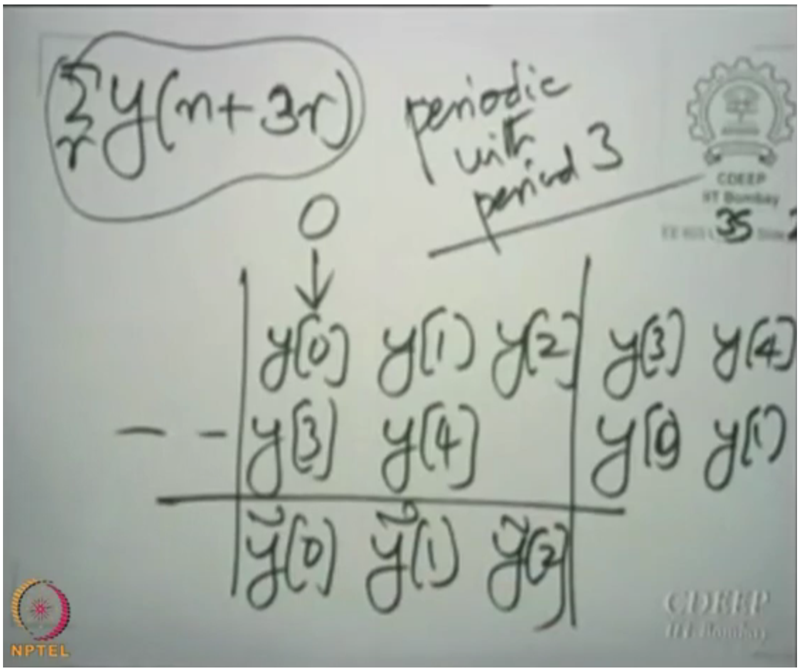


And as you continue, you can see that when you are calculating $y[3]$, then $x_2[0]$ has reached a zero sample. Hence the remaining samples can be calculated as follows:

$$y[3] \;=\; x_1[2]x_2[1] \;+\; x_1[1]x_2[2]$$

$$y[0] \;=\; x_1[2]x_2[2]$$

(Refer Slide Time: 14:01)



Now, when we carry out the time domain aliasing, what is going to happen? The sum $\sum_r y[n+3r]$ will look like this:

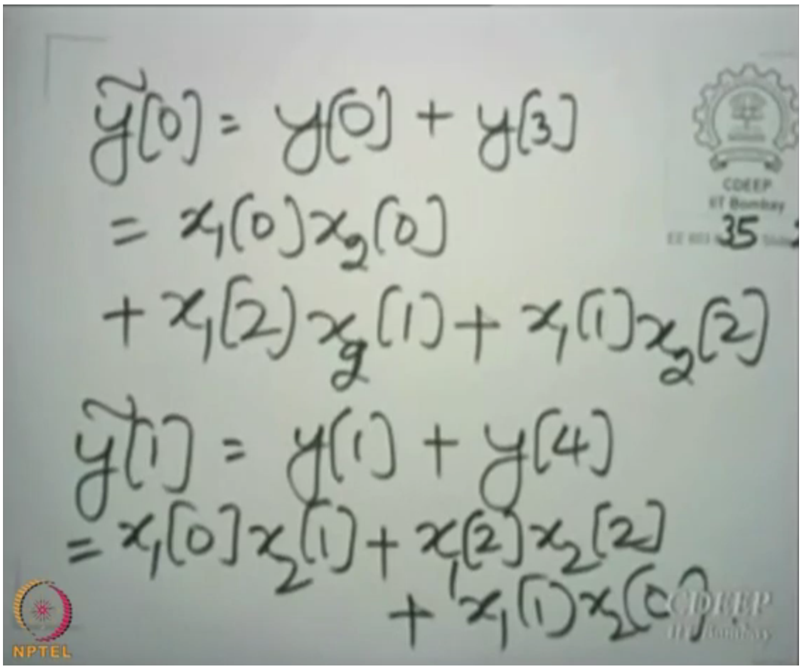
$$
\begin{array}{lllllll}
 & |\, y[0] & y[1] & y[2]\, | & y[3] & y[4] \\
.....\, |\, y[3] & y[4] & & |\, y[0] & y[1] & \\
 & |\, \tilde{y}[0] & \tilde{y}[1] & \tilde{y}[2]\, | & &
\end{array}
$$

At 0 you would have initially had just $y[0]$ and then starting from 0 $y[0]$, $y[1]$, $y[2]$, $y[3]$ and $y[4]$, but you are going to shift this by every multiple of 3 and add them. So $y[0]$ is also going to come aligned below $y[3]$ of the original copy, followed by $y[1]$ and so on. And this is also going to go backwards in steps of 3. So $y[4]$ would also come below $y[1]$.

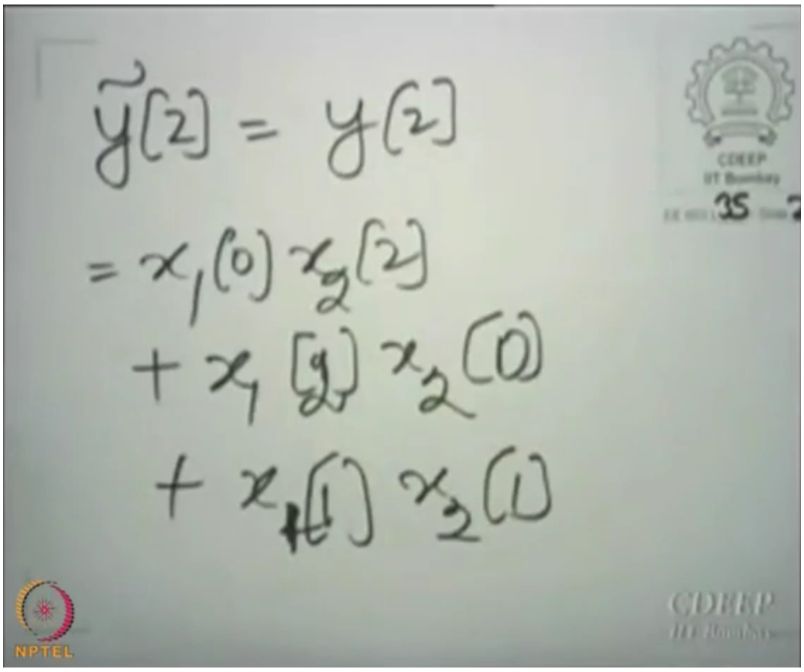
And as expected this sum is going to be periodic with period 3. The sequence obtained by taking the original sequence, shifting it by every multiple of 3 and adding the copies, is going to be periodic with period 3. That can be easily shown by putting $n+3$ in place of n in the sum, which gives you back the same sequence. So it is expected that this sequence is going to be periodic with period 3. Hence we only need to consider the principal period of 3 here.

Let us call the samples in this period $\tilde{y}[0]$, $\tilde{y}[1]$ and $\tilde{y}[2]$. We have an expression for each of these samples. Let us____write down these expressions.

(Refer Slide Time: 15:45)



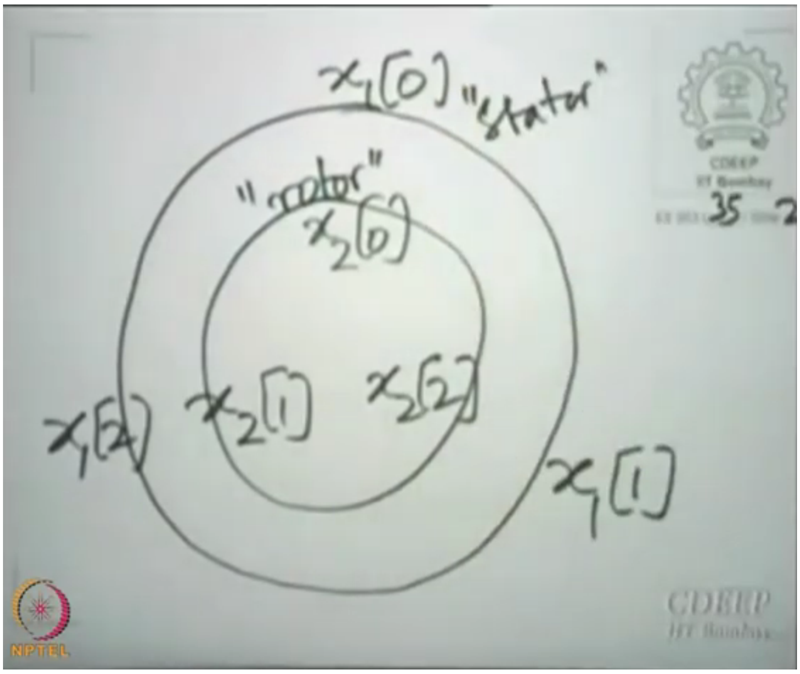
(Refer Slide Time: 17:54)



$$\tilde{y}[0] = y[0] + y[3]$$
$$= x_1[0]x_2[0] + x_1[2]x_2[1] + x_1[1]x_2[2]$$
$$\tilde{y}[1] = y[1] + y[4]$$
$$= x_1[0]x_2[1] + x_1[2]x_2[2] + x_1[1]x_2[0]$$
$$\tilde{y}[2] = y[2]$$
$$= x_1[0]x_2[2] + x_1[2]x_2[0] + x_1[1]x_2[1]$$

When writing each sample of $\tilde{y}$, we have taken care to write each product term in the sum in such a way that $x_1[0]$, $x_1[2]$ and $x_1[1]$ appear in the same order for all three samples of $\tilde{y}$. You will notice that it is different samples that are associated with these every time. In fact, to understand what is happening better, let us write down the samples not on a straight line but on the surface of a circle.

(Refer Slide Time: 19:15)



Let us fix the outer circle with the samples of x_1. So going in counterclockwise order, we have $x_1[0]$,$x_1[1]$ and$x_1[2]$. And we take an inner circle and put on it the samples of $x[2]$. Let us put the samples the way they associate for $\tilde{y}[0]$. So $x_1[0]$ associates with $x_2[0]$, so let us put $x_2[0]$ next to $x_1[0]$. Similarly $x_1[2]$ associates with $x_2[1]$ so let us put $x_2[1]$ next to $x_1[2]$. Finally, $x_1[1]$ associates with $x_2[2]$, so let us put $x_2[2]$ next to $x_1[1]$.

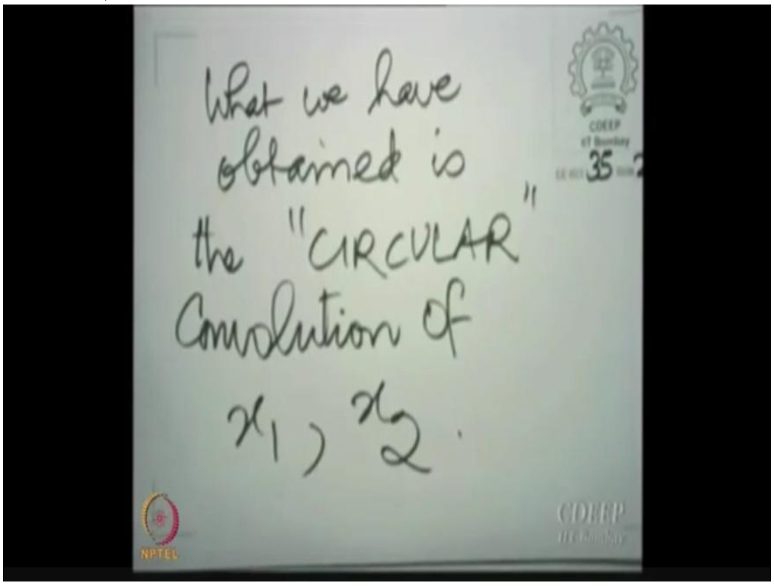
When computing the circular convolution, the inner circle remains fixed, so we call it the rotor and the outer circle which remains fixed, we call it the stator, taking a cue from the terminology for machines.

For $\tilde{y}[1]$, $x_1[0]$ now gets associated $x_2[1]$. So there is a movement (i.e. rotation of the rotor) of one step (in the clockwise direction). Here $x_1[0]$ gets associated with $x_2[1]$, $x_1[1]$ gets associated with $x_2[0]$ and $x_1[2]$ gets associated with $x_2[2]$.

When we come to $\tilde{y}[2]$, $x_1[0]$ is associated with $x_2[2]$. Here the rotor is rotated by 2 steps, and the associations are made after rotation of the rotor by 2 steps (in the clockwise direction).

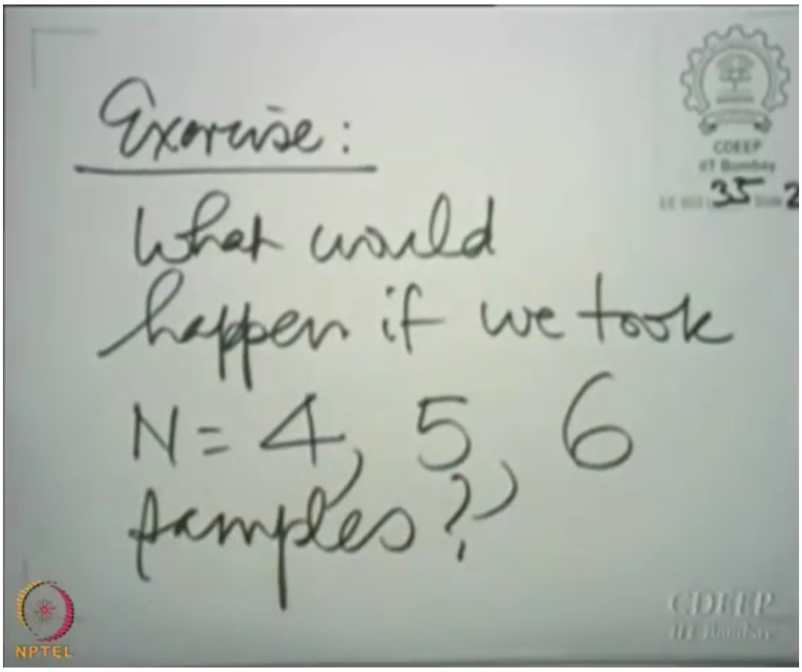
So it is as if we were convolving not on a straight line. It is as if we had a train and passengers in and out of the train not on a straight platform but on a circular platform. And therefore what we have got here is what we call circular convolution.

(Refer Slide Time: 22:17)



It is called the circular convolution because it is as if you were doing convolution with the sequences put on the surface of a circle, not on a straight line. And of course as expected circular convolution is bound to be periodic.

(Refer Slide Time: 23:18)

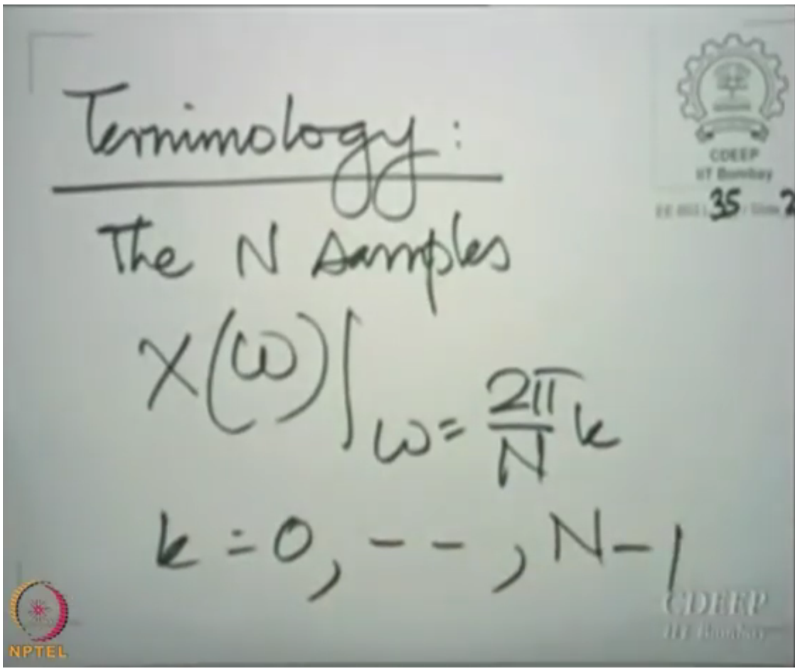


Now I leave it to you as an exercise to work out what happens in case you took 4 samples and not 5. So N equal to 4, 5, 6 samples, work them out for all the 3 cases.

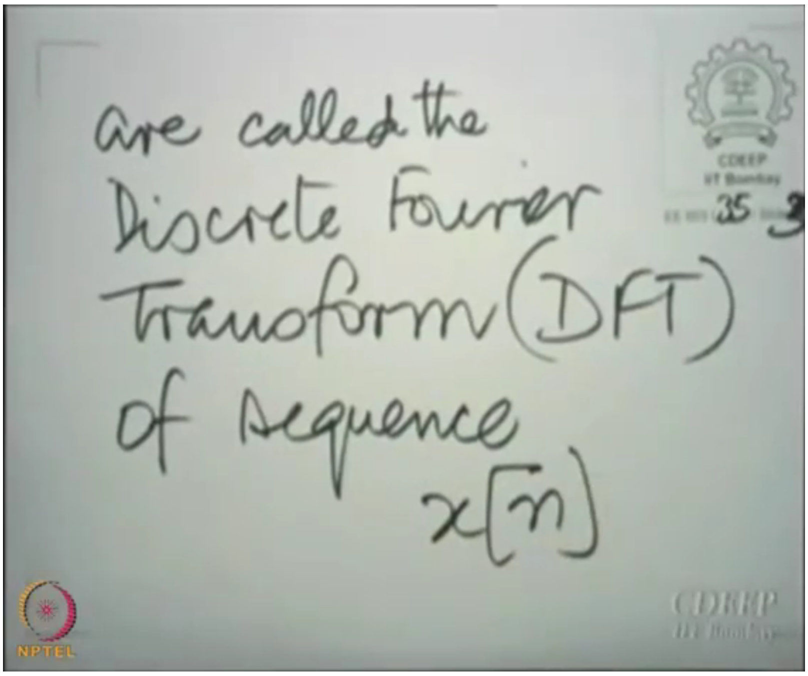
During our previous work, we took 4, 5 and 6 samples respectively of the sequences x_1 and x_2 and then worked in the frequency domain. Now in all this we are assuming that we have a way to go back after sampling, but we need to complete that one little step. How do we go back? After we have sampled the frequency axis how do we go back to the original time domain expression?

We can do that and obtain the correct sequence only if we have no time-domain aliasing. If you have time-domain aliasing, whatever we do is going to give us the aliased version of the sequence. Either way we must have a way of going back after sampling. To do that of course we can exploit or invoke the idea of vectors.

(Refer Slide Time: 24:46)



(Refer Slide Time: 25:31)



We must now put down some terminology. The N samples of $X(\omega)$, taken as:

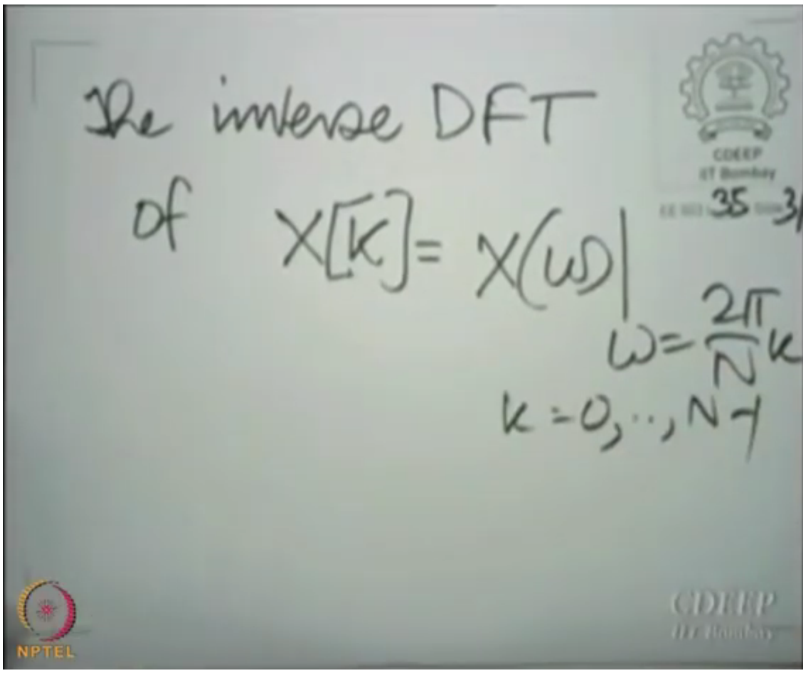
$$X(\omega)\big|_{\omega=\frac{2\pi}{N}k},\ k = 0,\ldots, N - 1,$$

these N samples are called the discrete Fourier transform, or DFT of the sequence $x[n]$. Now of course we should be more precise. We should call it the N-point discrete Fourier transform because we could have taken less or more samples too. Here we are assuming the sequence $x[n]$ has at most N samples.

You could possibly take a discrete Fourier transform as well, but then there will be time-domain aliasing, or you may be taking more samples than required. If you take more samples than required, there is no problem. So even if a sequence has only N non-zero samples, you could be very well taking more than N samples in the frequency domain.

And as you can see, you need to do it when you are trying to convolve two sequences. When you convolve two sequences you need to take more than the number of samples in either of the sequences. It is not unusual to do that.

(Refer Slide Time: 26:59)



Now if you have a discrete Fourier transform you need to have an inverse transform. We shall now use $X[k]$ to denote $X(\omega)|_{\omega=\frac{2\pi}{N}k}$, $k = 0,\dots, N - 1$. You have N such samples. So it is as if you had N dimensions along which you are trying to represent the original sequence.

So you have gone from N samples in one domain to N samples in a different domain, and each of these samples in the frequency domain corresponds to a vector, a vector created by a rotating complex number rotating with the corresponding frequency $\frac{2\pi}{N}k$.

So in the next lecture we shall see how we would reconstruct the original sequence or at least try to reconstruct the original sequence by using this idea of vectors. Not only that, we will also see how to do this efficiently. You see all this is useful if it is going to make our computation efficient. And therefore after having established that we can reconstruct we also need to see if we are going to get a computational advantage by discretization, which we shall also see in the next lecture. Thank you.