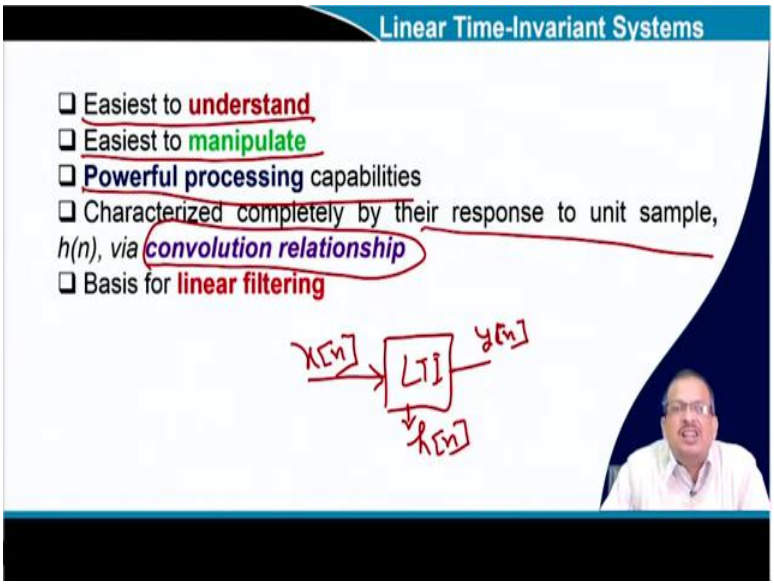
Signal Processing Techniques and its Applications
Prof. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 09
D.T.S (L.T.I System)

Ok. So, in the last class, we discussed the characteristics of discrete systems, and we analyzed that there is a discrete system, a time-invariant discrete system, a time-variant discrete system, a stability of the discrete system, a causal system, and a non-causal system. So, today we will discuss LTI systems and linear time invariant systems. Why this LTI system?

(Refer Slide Time: 00:57)



So, linear time-invariant system; the system will be linear and time-invariant.

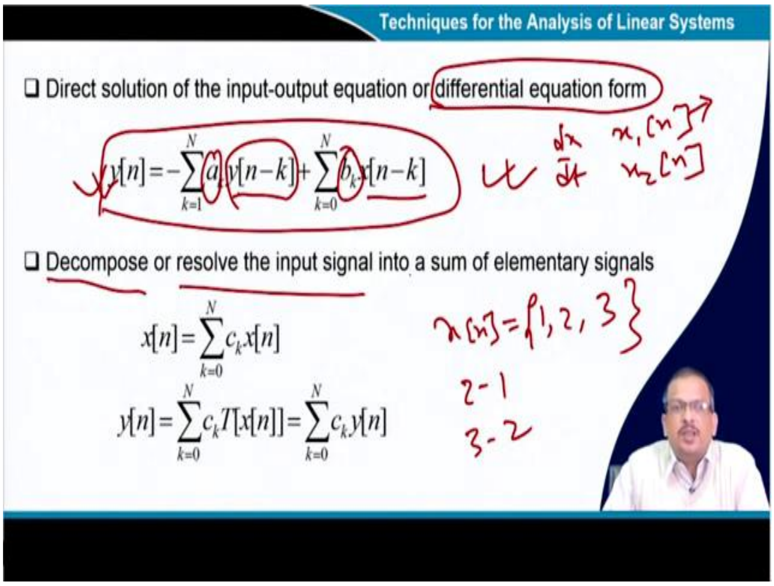
(Refer Slide Time: 01:02)

Why do we choose the linear time-invariant system? Because the linear time-invariant system is easy to understand because it follows the superpositions principle. So, it is easy to understand is one of the factor. Second, it is easy to manipulate. I can manipulate the system very easily and have powerful processing capabilities.

So, and so, are characterized completely by their response to the unit impulse. So, a system can be characterized by that unit impulse response, which is called a convolution relationship. We will discuss why it is a convolution relationship. So, we will only talk about the LTI system, the linear time invariant system. So, as you know, what is that? Suppose I want the LTI system, ok.

So, if I apply an input x[n] because it is a discrete system. So, the signal is discrete, and the output is also in discrete y[n]. Now, you know that the transfer function of the system is defined by h[n], which is called impulse response or discrete impulse response of the system, and the system property is linear and time-invariant.

(Refer Slide Time: 02:42)



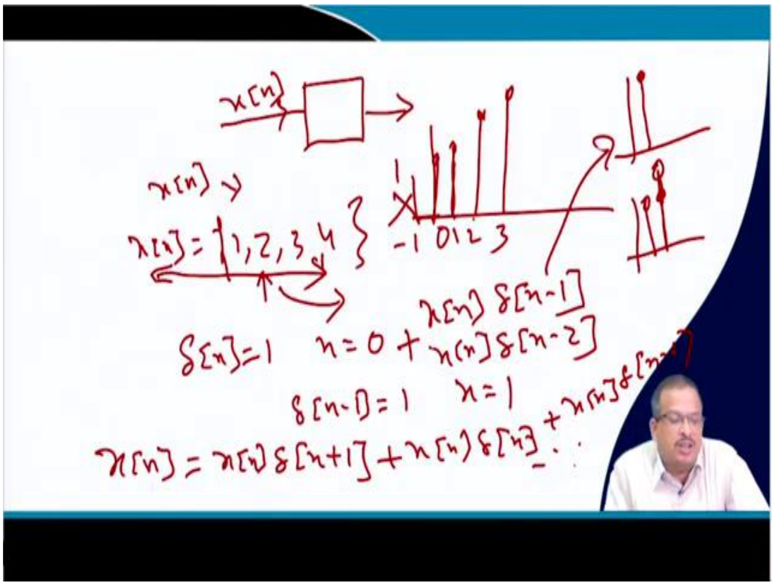
Now, what we want to know, as you know all of you know, is that I know that a discrete-time or because you can say that an LTI system can be analyzed using two methods; one is called a direct discrete solution of the input-output equation or differential equation form.

So, a discrete system is nothing but a form of a differential equation. You know that differentiation between two signals when it is a continuous signal, we call dx/dt, but when a discrete system; suppose I have an x1[n] sequence and I have another sequence called x2[n], the differentiation means or if x1[n] is a sequence the first order differentiation means the difference between the sample.

So, suppose I have an x[n] is equal to {1, 2, 3} then I said first order differentiation equation may be the 2 minus 1 and 3 minus 2 like that difference. So, I can express the output as a form of differential equation that depends on the previous output and present and past input.

So, I can write down that where ak and bk are the constant coefficients. This is one of the approaches which is called differential equation form. Other approaches de-composite or resolve the input signal into a sum of elementary signals.

(Refer Slide Time: 04:26)



Suppose let us say I have a system and I apply a signal x[n]. So, as I know x[n] is a discrete signal. So, I can say x[n] consists of several signals. Several signals means suppose x; so, let us see an example x[n] look like this: {1, 2, 3, 4}. Let us say this is the 0th sample. So, I can say this is nothing but a 0. This is minus 1 means 1, 1 is 2, the value is 2, 2 is value 3, then the value is 4. So, this is 1, 2, 3.

Now, I can say that instead of writing x[n] I want to write in δ using δ function. So, as I discussed in that previous class, the δ function. So, I know that $\delta[n]$ is equal to 1 when n is equal to 0. So, I said that x[n] consists of several δ signals. Let us say I said that x[n] multiplied by $\delta[n-1]$; that means $\delta[n-1]$ is equal to 1 when n is equal to 1.

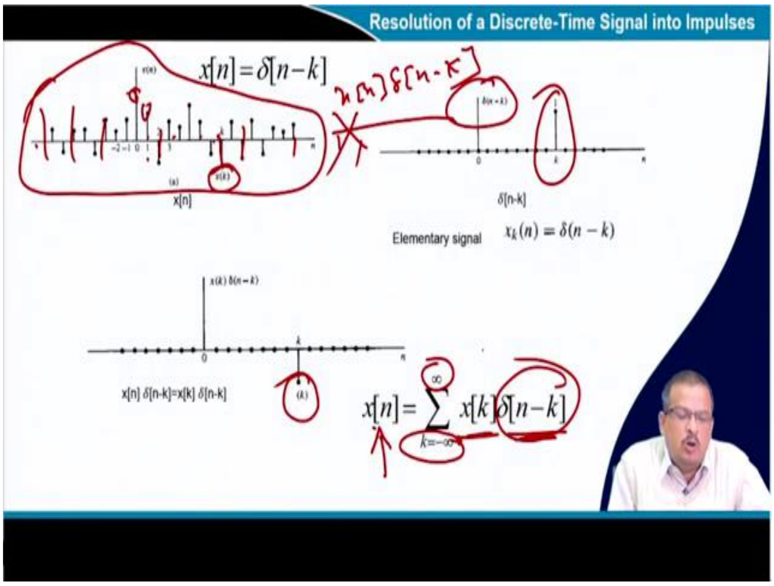
So, I can say that the output of that product will be only the sample number 1. So, this is in one elementary signal. Then I can say x[n] multiplied by $\delta[n-2]$. So, that signal is nothing, but a second sample will be the 1 multiplied by 1. So, the second sample I will get. Now, if I take the sum, so, I take the sum of δ x[n] plus $\delta[n-1]$ plus x[n] into $\delta[n-2]$, I get that signal.

Now if I take the sum of all those things, so, I can say

$$x[n] = x[n] * \delta[n+1] + x[n] * \delta[n] + x[n] * \delta[n-1].$$

So, dot dot dot up to 3. So, I can decompose it as an input signal in terms of the δ function.

(Refer Slide Time: 07:24)

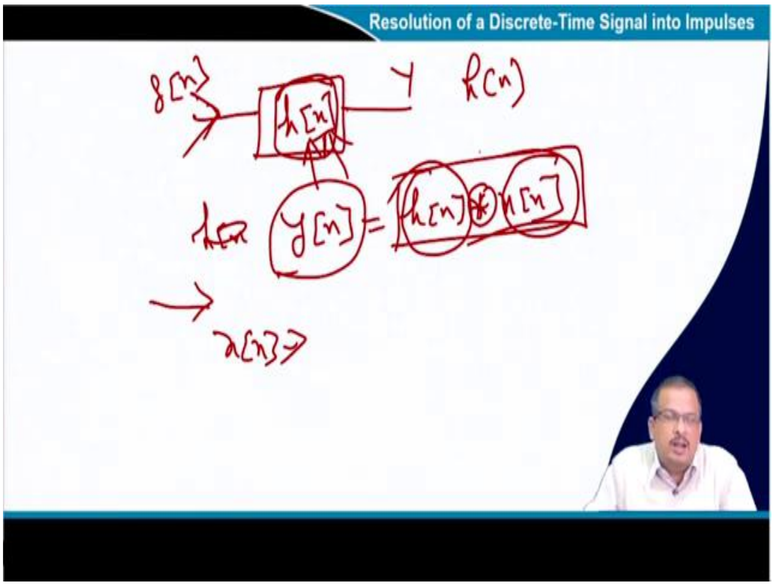


So, now I generalized it. How do I generalize? Let us say I have a signal. This is an x[n] signal. So, a lot of samples are there; the 0th sample is this value, 1 sample is this value, and the kth sample has a value of x[k]; the kth sample has a value of x[k]. Now, if I multiply that if I say that x[n] is multiplied with δ[n-k], so, if I represent like that that x[n] is equal to δ[n]. So, δ[n-k] is nothing but a 1 at n equal to k.

So, this is δ[n-k]. Now, if I multiply these two things, so, what will get? All the samples will multiply by[0], and the kth sample will multiply by 1. So, I get the kth sample value.

So, similarly, I can say k varies from minus infinity to infinity x[k] multiplied by δ[n-k]. So, I get all the samples. So, what am I doing? I am decomposing the signal x[n] in term of δ function. Now, you may ask how I analyze that h[n]. I have decomposed the x[n] then how do I analyze that h[n]?

(Refer Slide Time: 08:57)



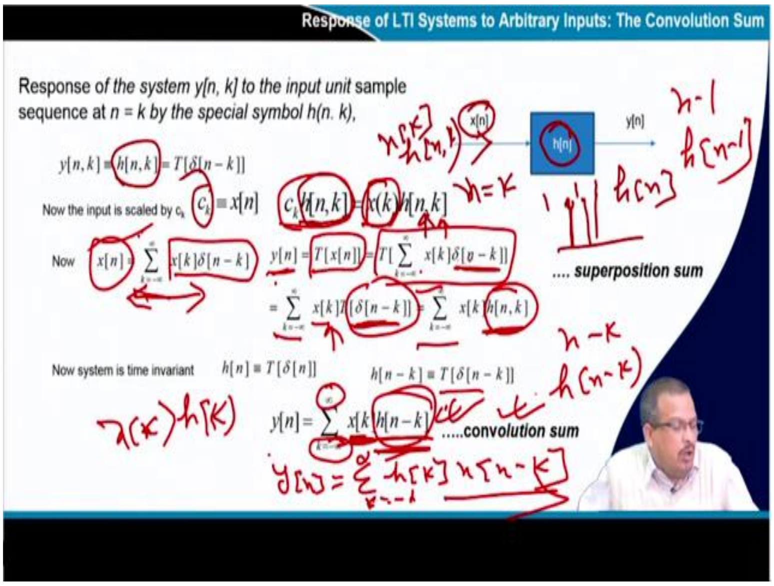
So, as you know, if a system h[n] is the impulse response of the system, what is impulse response? Impulse response means I apply an impulse function as an input, and I take the output. So, I apply δ function at the input and I take the output. So, if I take the all δ function output I get the response of h[n] as we discussed. So, first, I apply δ[0], then apply δ[1]. Maybe it applies δ [-1].

So, the index will change, and I get the impulse response of the system h[n]. As you know the output of the system whose impulse response is h[n] is nothing but a convolution h[n] convolved with input signal x[n]. Why it is convolved? Why[n]ot it is multiplication?

So, I have to prove that I want to show that the output of an LTI system whose impulse response is h[n] is nothing but a convolution between the impulse response and input signal that I know but have to prove it. How do I do that? To do that, I consider the input signal as nothing but a decomposition of a. I can decompose it at an elementary signal level, and I apply each of the elementary signals at the input, and I get the output.

And the output will be since the system is linear, I can take them sum and I get that entire output. So, I de-composite x[n] in an elementary signal, and I apply each of the signals in the input. I collect the y[n], and I take the sum. So, if I take the sum, it will show that it is nothing but a convolution of the impulse response of the system and the signal. So, how do I show it? How do I prove it? Let us prove it.

(Refer Slide Time: 11:20)

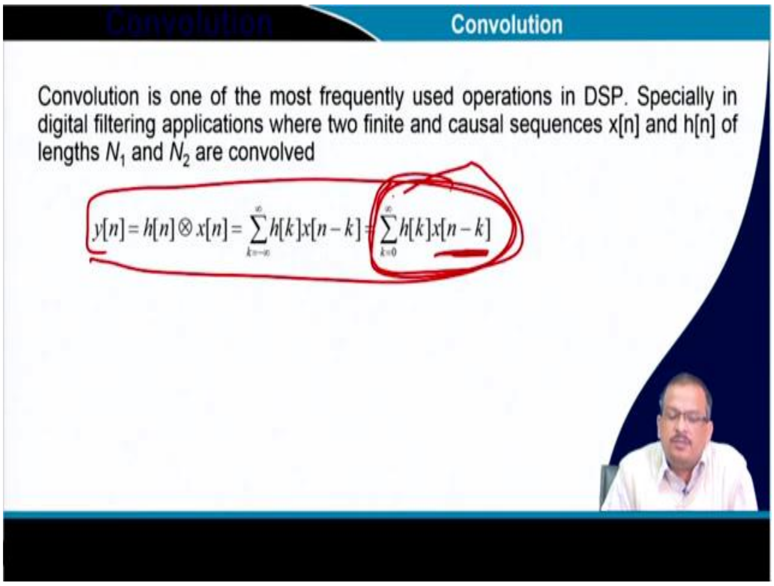


So, let us say that h[n, k] is the impulse response of the system. So, h[n] is the impulse response of the LTI system, x[n] is the input to the system, and y[n] is the output of the system. I have to prove that y[n] is nothing but a convolution between x[n] and h[n]. So, how do I do that?

So, what I said is that I de-composite x[n] in terms of the elementary signal. So, now, each of the sum, so, if I say that h[n]; so, what is h[n]? h[n] is nothing but a unit impulse response. So, the amplitude of the impulse is 1. Now, if I apply x[n], that means the amplitude of the impulse will be modified based on the x[n] or at that index. So, x[n] is nothing but a scaling function. So, instead of amplitude 1, the amplitude will depends on the value of x[n].

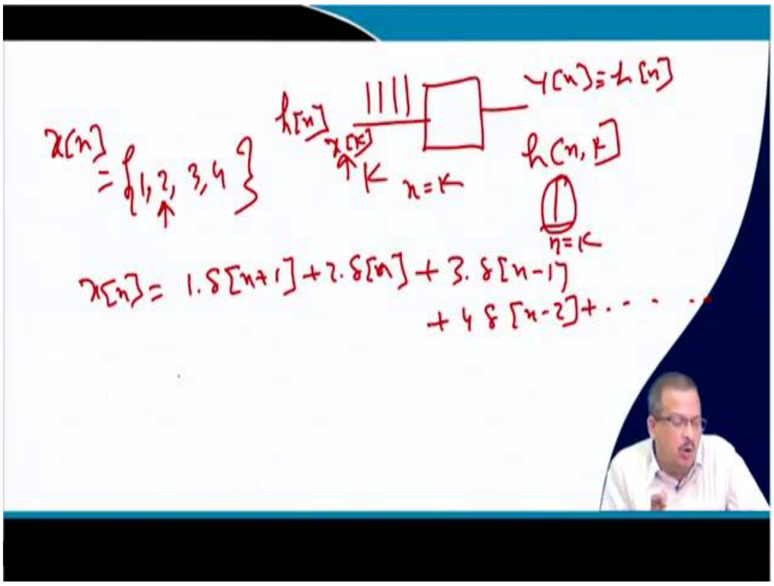
So, I can say that the unit impulse response, which is h[n, k], will be multiplied by suppose this is the unit impulse response as kept index when n is equal to k. So, that is nothing but a x[k] that will be multiplied with h[n, k] to give me the amplitude of that sample. Do you understand or not? So, what am I saying? I am saying h[n] is an impulse response of a system. So, let us say I took another slide so that I can show you.

(Refer Slide Time: 13:14)



Just a minute ok.

(Refer Slide Time: 13:22)



So, what I am saying is that h[n] is the impulse response of a system. So, what is this meaning? That meaning is that h[n] I have a system I apply a unit impulse, and I collect the output, and that y[n] is equal to h[n]. Now, I said if I say that kth instant when n is equal to k, that h[n, k] is the kth instant of the h[n]. So, the amplitude is 1 at n equal to k. Now, when I apply x[k] as an input, what is the meaning? The amplitude of the x[k] will modify the amplitude of the h[n, k], the kth position.

So, I can say the input acts as a scaling factor. So, that is why I am writing that x[k] multiplied by h[n, k] is nothing; x[k] is nothing but a scaling factor of that impulse. Now say what is x[n], how the scaling factor is defined. So, I can say x[n] can be de-composited in terms of an elementary signal. So, what is that? I know that k is equal to minus infinity to infinity x[k] δ[n-k]. Again, I give an example so that you can understand. How do I do that?

Suppose I have a signal x[n] that is equal to 1, 2, 3, 4. Let us say this is my[0]th sample. So, what is the imperative in terms of δ representation? I know that the

$$x[n] = 1 * \delta[n+1] + 2 * \delta[n] + 3 * \delta[n-1] + 4 * \delta[n-2], \text{ and the rest are 0.}$$

So, I can say that if I generalize it, k varies from the index from minus infinity to infinity all the δ functions multiplied by the amplitude of x[n], x[n] that sample will be the signal x[n]. So, that is why I call de-composite the signal in terms of the elementary signal. Now, I said what is y[n]? y[n] is nothing but a when the signal x[n] passes through the system.

So, y[n] can be defined mathematically this way: the transform of x[n], x[n] will transform based on the system. Now, I put that x[n] in terms of a δ function. So, instead of x[n], I put k equal to 1 to infinity x[k] multiplied δ[n-k], which can give me that k equal to minus infinity to infinity x[k] and transform of because x[k] is a scaling factor.

So, the transform of δ[n-k] means nothing but the impulse response at the kth position. So, I can say k is equal to 0 to infinity. This is nothing but the h[n], k impulse response of n k. Now, the system is time-invariant. So, if it is time-invariant, I know that if I apply n minus k input, it will be h[n-k]. If I apply n minus 1 input it will be h[n-1].

So, I can say that y[n] is nothing but a k equal to minus infinity to infinity x[k] h[n-k]. This is if you see that it is not x[k] multiplied by h[k]. It is a multiplication and sum, but there is a sum shifting in h[k] which is called convolution sum. If you understand the meaning of convolution, what is convolution?

Your friends idea is convolved your idea. Suppose your friend is presenting an idea and you have a different idea, but after that presentation, you are convolved by your friend's idea. Then the self-convolution happens. What do you mean by physical convolution happening? That means your idea is modified as per your friend's idea.

So, that convolution happens, it is not multiplication. Given a physical example: suppose a system, let us say take a painting system; suppose I have a machine that can paint 6 centimetres of a pipe. If I want to colour the entire pipe, that means the characteristics of the paint machine have to be convolved with the pipe, the pipe is my input.

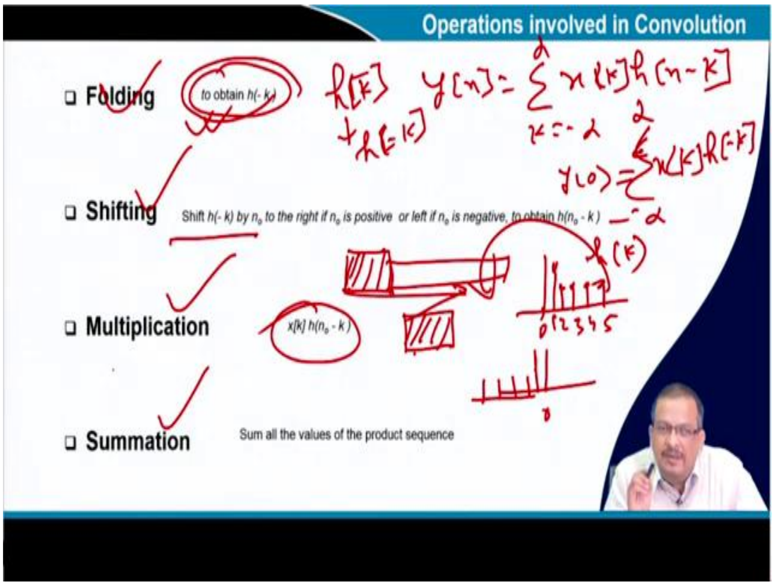
So, I have to slide the pipe over that h[k] or another way I can slide that system over the pipe. Both are possible. Here, I am sliding the pending system over the pipe x[n] input, but I can show that the convolution sum is reserved for associative property. So, instead of writing this, I can write y[n] as equal to k minus infinity to infinity h[k] multiplied by x[n-k]. Both are the same.

That means either painting machine can convolve the entire pipe. So, the painting machine is moving, or the pipe can move. Here, the painting machine is moving; here, the pipe is moving. This is a general idea. If you see, you know from your signals and system background that any output of a system is nothing but a convolution between the input and the impulse response of the system in the time domain.

So, n is a time domain because n is in the time index; this is discrete, but time index. So, this is the proof of why we call it convolution, ok? So, convolution is an important operation in DSP or signal processing, and convolution is always an important operation. So, you have to know how to compute a convolution or how to write and program for convolution.

If I told you to write a program for convolution, how would you develop the program? I have given an input and given an output. I have given a system, and given an input, you have to compute the output; it is nothing but a convolution. So, how do you calculate the convolution, or how do you write a program for convolution? As you can see from the math here, the pipe is running.

(Refer Slide Time: 21:47)



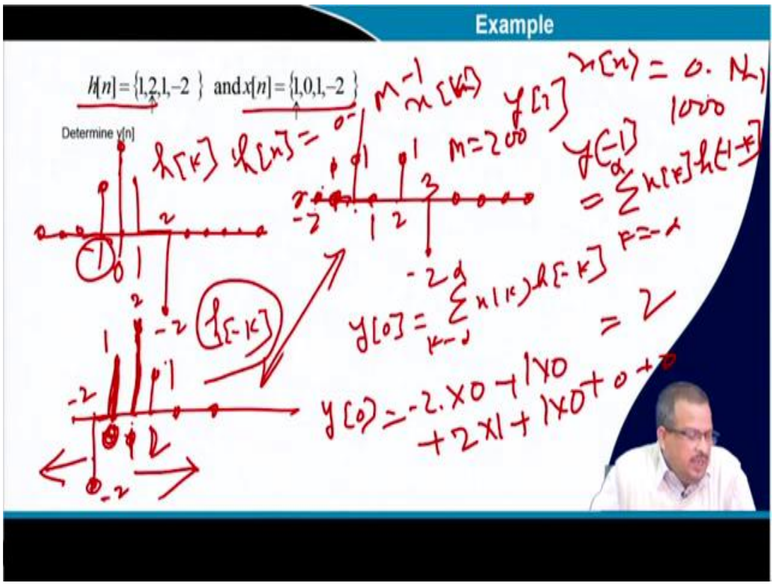
So, you will see that this required a 3 operation. First operation holding. If I say y[n] is equal to k equal to minus infinity to infinity x[k] h[n-k], suppose this one I want to implement. I the first operation I required. So, y[0] is nothing but a x[k] convolved with h[-k] sum minus infinity to infinity. So, the first operation for that convolution is folding. You must fold h[k] or h[n] or h[k] to get h[-k]. What is folding? What is the folding of a signal?

So, suppose I have an edge like this: let us see this is 0, this is 1, 2, 3, 4, 5 sample number, and this is the amplitude. What is folding? So, this is nothing but a, let us say, h[k]. If I say what is h[-k], so, the positive side will come to the negative side. So, I can say h[-k] is nothing but a 0 then 1, 2, 3, 4, 5. So, this side will come to this side, index will come to this side. So, this is as if it is folded on this side. So, this is called folding.

Then shifting. So, for y[0], the shifting, after folding, I have a signal x[n]. So, think about that: I have a paint machine 6-centimeter paint machine, and I have a pipe. So, I want to slide the paint machine over the pipe for uniform painting. So, what will you do? First, we will fold this system in the painting machine here fold in it and then slide it over the pipe. So, I shift it over the pipe. So, that is called shifting. Then, the paint will be imposed on the signal.

So, the paint is called multiplication and then takes the sum. So, folding, shifting, multiplication, and summation, you do not have to remember them by remembering facts of remembering you can analyse them. You can logically, you can see those are the operations required for convolution. Let us do a convolution manually, and then I will give a task to write a program for convolution.

(Refer Slide Time: 24:47)



So, let us say I have a system h[n] and I have a signal x[n]; the arrow indicates the 0th position sample. So, if I draw pictorially h[n], so, I know that here there is an amplitude value 1 at minus 1, amplitude value 2; let us say this is 2; 2 at 0 position, then I have an 1 again at 1 sample, then again minus 2 at the 2nd sample. So, this is minus 2 at the 2nd sample, and then all other samples are 0. So, this is my system, which is called h[k] let us say ok.

Now, what is the signal? I have a signal whose 0th value is 1 whose next value is 0. So, one value is 0, the second value is again 1, and the third value is minus 2; this is the third sample value. The rest are 0, and this side is also 0. So, if I want to, the first operation is folding. So, I have to calculate h[-k].

So, how do you calculate? I just fold it. So, I just folded it. So, this is the 0th value. So, this will be 2 and then minus 1 will be 1, and minus 2 will be 2. So, this is minus 2, this is minus 1, this is 0, and this is 1. This is the folded signal. Then, the folded signal will multiply with this is x[n] or x[k], I can say. So, multiply with x[k] and sum I get y[0].

So, y[0] is nothing but a I know the summation of minus infinity to infinity x[k], k equal to into h[-k] because n is equal to 0. So, I multiply this one by minus 2 samples, and here, the minus 2 sample value is 0. So, I will multiply this one by this. So, I can say y[0] is equal to multiplying this with this.

So, minus 2 multiplied [0] amplitude also minus 2 plus minus 1 this is one sum. So, this is 1. So, 1 will be multiplied by here this one multiplied by this one. So, this one is 0 again. It is 0 plus 2. The 0th sample is 2, multiplied by the 0th sample is 1 plus the second sample is multiplied by 0 plus all other samples will multiply by 0. So, I can say this is nothing but a 2.

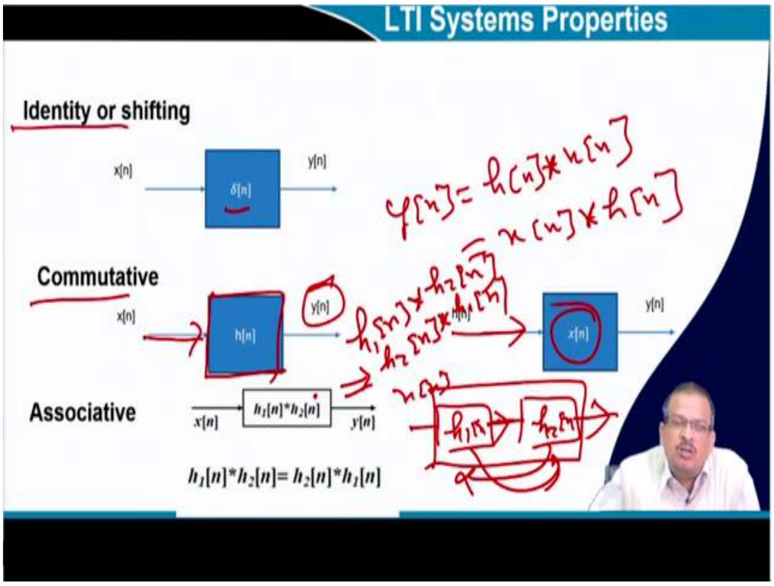
Now, when I want to calculate y 1, I just shift this signal by 1 sample. So, instead of 0 is here, the 0 will be here, the value of the 0th sample will be 1, the value of the 1 sample will be 2, and the value of the 2nd sample will be 1. Then again, multiply and sum. So, I shifted it and multiplied and sum shifted it and multiplied and sum.

Now, when I want to calculate since h[n] has a minus 1 value also, I may get y minus 1. When I get y minus 1 if you see k equal to minus infinity to infinity x[k] multiplied by h of minus 1 minus k; that means I required a right shift this side shift. So, I shift h[k] on this side and then take the product and sum. I get the y of minus 1, understand? So, I will take the shift on this side.

So, this is my h[-k] I take instead of taking the shift in this side, I will take the shift in this side by one sample, and then I take the product, and then I get the sum. Now, you think about what is the generalized formula or algorithm to write a convolution. So, if I am giving you the task to write an algorithm to compute the convolution of a nth element, let us say[n] equal to 1000 samples. You collect 1000 samples, and you have a frequency response of plus 2000 samples.

So, h[n] varies from 0 to m x[n] varies from 0 to n, and the number n minus 1 is m minus 1. So, this value of m is equal to 200, and the value of n is equal to 1000; write your C program or whatever algorithm for computing the convolution. So, this is convolution.

(Refer Slide Time: 30:32)



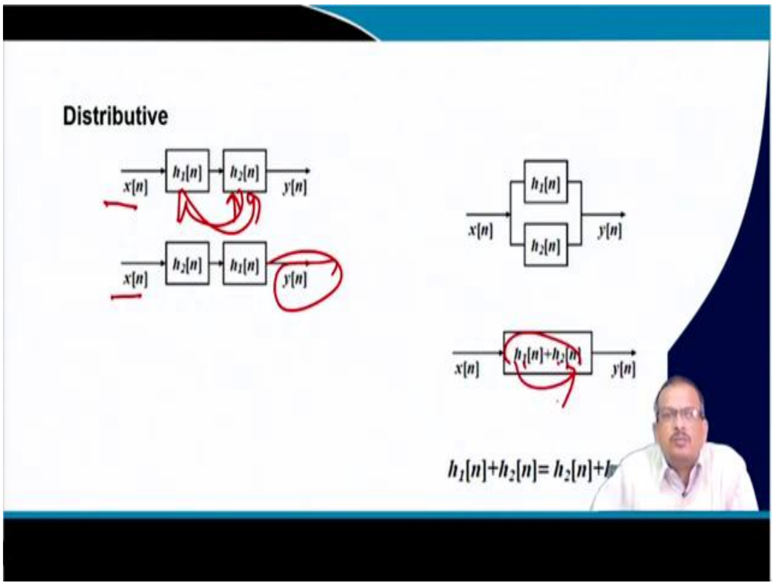
Now, what are the properties of convolution? You can say the properties of the LTI system because the output is nothing but a convolution. So, it is called identity or shifting is nothing but a δ function. If the system is δ function, I can get the identity signal or shifting can be done.

Instead of δ[n], if I δ one minus 1 n minus 1, that means one sample shifting. Similarly commutative, if I apply x[n] to an h[n] impulse response of the system is h[n], then I get y[n]. Now, just vice versa. If I apply y[n] as an h[n] as an input and x[n] as a system, I get the same y[n], so that means y[n] is equal to h[n] convolved x[n].

So, that means either you shift the pipe or shift the paint machine both the way the pipe will be painted, and this is nothing but equal to x[n] convolved with h[n]. Both are equal. System level is also associative; that means, suppose I have two systems the I have I have a system h1[n], and I have another system h2[n].

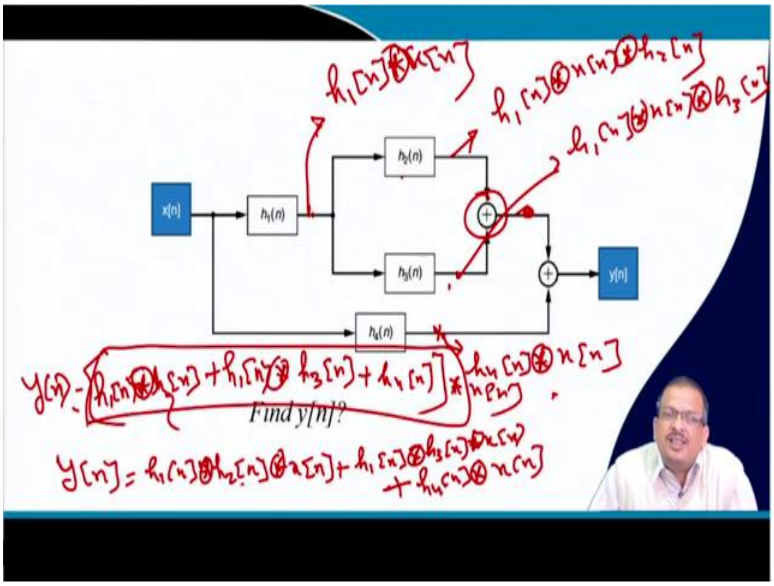
So, as you know, the output is x[n], and the c input is x[n]. So, output x[n] convolved with h1[n] and h1[n] again acts as an input, and I can finally output. So, I get that x[n] convolved with h1[n], then convolute h2[n]. Now, if I say those systems are one system, so, it is nothing but the h1[n] convolved with h2[n]. So, that is the same thing. I can also write h2[n] convolved with h 1. So, it does not matter whether this system is come first or this system is come first because it supports associative law.

(Refer Slide Time: 32:40)



Then distributed, x[n]. So, all are LTI systems; all are linear time-invariant systems. If I apply x[n], if there are two systems, h1[n] and h2[n], I can apply x[n]. I can say that they are distributed. They are interchanged, and the output will be the same. This is in convolution and then apply, here is distributed, I applied like this way whether it will be h1[n] plus h2[n] also this type h 2 and this type h 1.

(Refer Slide Time: 33:25)



So, once you know all those property suppose I give you this system. I told you to compute y[n]. What is the value expression of y[n]? So, what is there? What is the output here?

Here is nothing but a h1[n] convolved with x[n]. What is the output in here? Here is nothing but a h4[n] convolved with x[n]. So, this is an input two h2[n]. So, what is the output in here? It is nothing but the h1[n] convolved with x[n] convolved with h2[n].

Here is again h1[n] convolved with x[n] convolved with h3[n]. So, those two things will be added up. So, I can write down y[n] equals h1[n]. Let us say I write the signal at the end convolved with h2[n] convolved with x[n], plus I can say h1[n] convolved with h3[n] and convolved with x[n]. So, that is the output of here. Now, this will be added up plus h4[n] convolved with all convolution circle x[n].

So, if I take the x[n] as a common side, I can say y[n] is nothing but a h1[n] convolved with h2[n] ok plus sorry h2[n] this is h 2 plus h1[n] convolved with h3[n] plus h4[n] and whole with convolved with x[n], understand. So, any network I can give any system cascading.

So, I can say that I can find out the impulse response. If I told you the impulse response, so, if x[n] is a unit impulse, so, it is this expression is nothing but the impulse response of the entire equivalent system. So, as I give you the task, the task is what is the takeaway from this lecture. The takeaway from this lecture is how to write a program for convolution, and as you know, the output of a system, if the impulse response is given the output is nothing but a convolution with input ok.

Thank you.