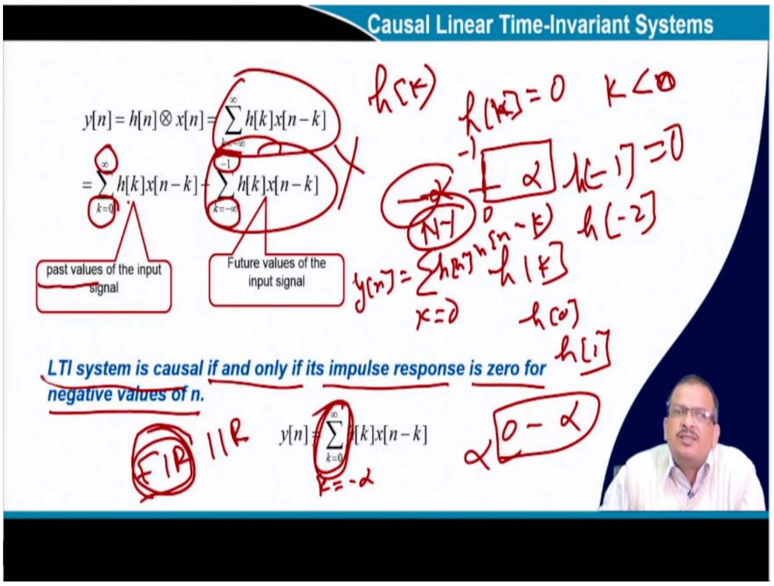
Signal Processing Techniques and its Applications

Prof. Shyamal Kumar Das Mandal
Advanced Technology Development Centre
Indian Institute of Technology, Kharagpur

Lecture - 10
Linear Time-Invariant Systems (Continued)

So, we are discussing that the output of an LTI system is nothing but a convolution of input along with the system's impulse response.

(Refer Slide Time: 00:38)



Now, we will discuss some linear time in variant systems, like causal, stability, etc. So, I said causal linear time-invariant system. So, how do I write the expression of the convolution? So, as you know, what is the convolution? k equal to minus infinity to minus infinity to infinity.

Now, what is the causality definition? An LTI system is causal if and only if its impulse response is 0 for negative values of n. That means h[k] or h[n] or h[k] will be equal to 0 if k is less than 0, a negative value of k. Any[n]egative value of k, h should be 0, and then only I can say the system is causal.

Now, if I see that I have an infinite sum, negative infinity to positive infinity, I can say[n]egative infinity to positive infinity, there is a0. So, I can consider 0 to infinity and minus infinity to minus 1. So, I just break up this summation: k equals 0 to infinity and k equals minus infinity to minus 1, which is nothing but a summation of minus infinity to plus infinity.

So, this signal depends on the (Refer Time: 02:18). So, I can say that here, k starts from 0. So, h[k], h[0], h[1]; so, those have a value. But the k starts from negative, h[-1], h[-2]. So, I know the definition of causality showed me that h[-1] value must be 0. So, I can say this term should not be present if it is a causal system. So, when I say that I am in the causal LTI system, I may write k equal to minus infinity to infinity instead of k equal to 0 to infinity.

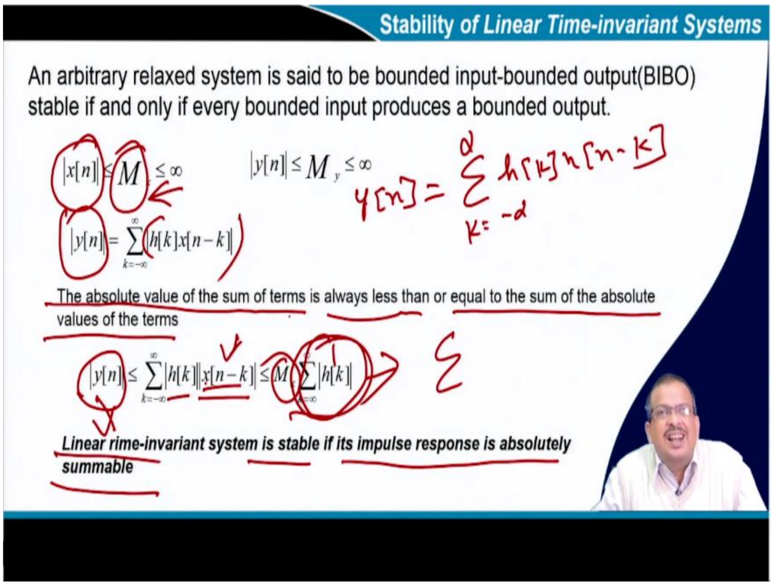
So, if I say my system is causal, 0 to infinite, what is that 0 to infinite, what is it called? It is called an order of the system. So, you heard about FIR and IIR; FIR means Finite Impulse Response, and IIR means Infinite Impulse Response.

When I restricted this sum with a finite number, instead of infinite, if I write y[n] is equal to k equal to 0 to N-1 h[k] into x[n-k]; that means the n-1 is a finite number or n is a finite number then I call the system has a finite impulse response, so I call FIR, Finite Impulse Response.

So, you have heard about FIR filters. So, if I consider the filter to be a finite impulse response system, then I call it an FIR filter. What is IIR? IIR means Infinite Impulse Response; that means h[k] has an infinite value, and k varies from 0 to infinity. If it is causal, then it is 0 to infinity. If it is non-causal, then it is minus infinity to infinity, understand.

So, there are two terms: causal, non-causal, FIR, and IIR. If it is a causal system, we reduce to minus infinity to infinity to 0 to infinity. If it is restricted to a finite number, I restricted to k equal to 0 to N-1, where n is the order of the system. So, n number of impulses I have taken. I have truncated that infinite impulse to n number of impulses called FIR, Finite Impulse Response, clear.

(Refer Slide Time: 05:25)



Now, I come to the stability of a linear LTI system. How do I calculate the stability of an LTI system? So, what is the definition of stability? I have already discussed this in this week's lecture. What is the definition? The definition is that if I apply a bounded input, the output must be bounded, and then I can say the system is stable.

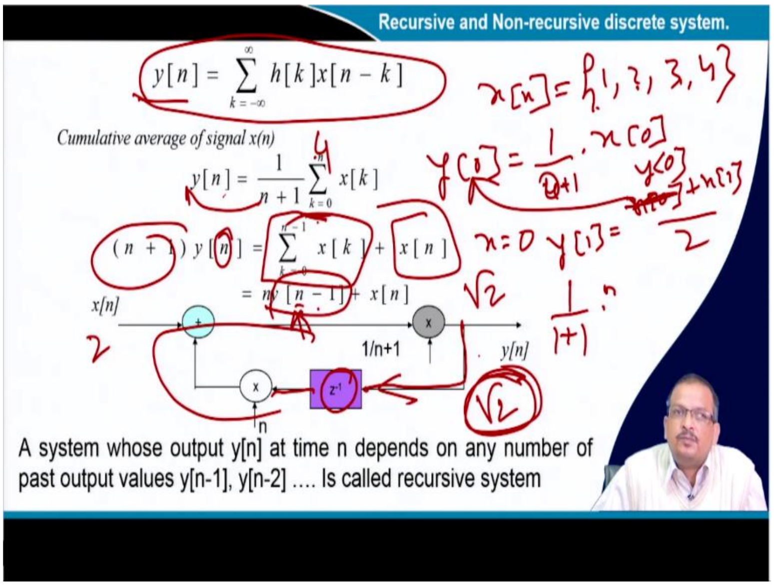
That means I apply a bounded input, and I get a bounded output. Only then can I say the system is stable. So, if I say the bounded input is mod of x equals M x, which is less than infinite, it is not infinite. Then, the output must be bounded. So, now, I take the system. What is y[n]? So, what is a system? y[n] is nothing but a convolution equation k equal to minus infinity to infinity h[k] x[n-k].

If I take the magnitude, what is the magnitude? When I take the mod. So, I take the mod on both sides. So, I take the mod on both sides. So, I can say this mod: the absolute value of the sum of the I term is always less than or equal to the sum of the absolute value of the term, which is standard mathematics.

So, now, I can say the mod of y[n] will be less than equal to the mod of h[k] into the mod of x[n-k]. The input is bounded, so a mod of x[n-k] is M x bounded input. Now, output will be bounded if this term is bounded. So, when this term will be, this term summation of impulse response must be bounded. So, the LTI system is stable if its impulse response is summable.

So, if this is bounded, then only the LTI system will be bounded. So, the LTI system will be bounded; if the impulse response is absolutely summable, then only I can say the system is stable. So, what do I do? I get that impulse response; I just add them; if the add value is infinite, that response add value becomes a finite value, and then I can say, ok, this system is stable. So, this is the stability condition of an LTI system.

(Refer Slide Time: 08:29)



Now, there is a recursive and non-recursive discrete system. We have already learned about the recursive algorithm and recursive system. So, what are recursive and non-recursive discrete systems? So, I have an h[n], which is the output of a system, convolution sum. So, a cumulative average of a signal; suppose I have a signal x[n]. I want to calculate the cumulative average.

So, suppose x[n] is equal to {1, 2, 3, 4}, how do I calculate the cumulative average? So, what is the average? Sum divided by the number, ok. So, I can say the first sum is nothing but k equal to 0 to let us say this is {1, 2, 3, 4}, so n equal to 4. So, I can say

$$y[0] = \frac{1}{0+1}\sum_{k=0}^{n} x[0]$$

So, here, n is equal to 0. Then, what is y[1]?

$$y[1] = \frac{x[0] + x[1]}{2}$$

So 1 by[1] plus 1 into x[0] plus x[1].

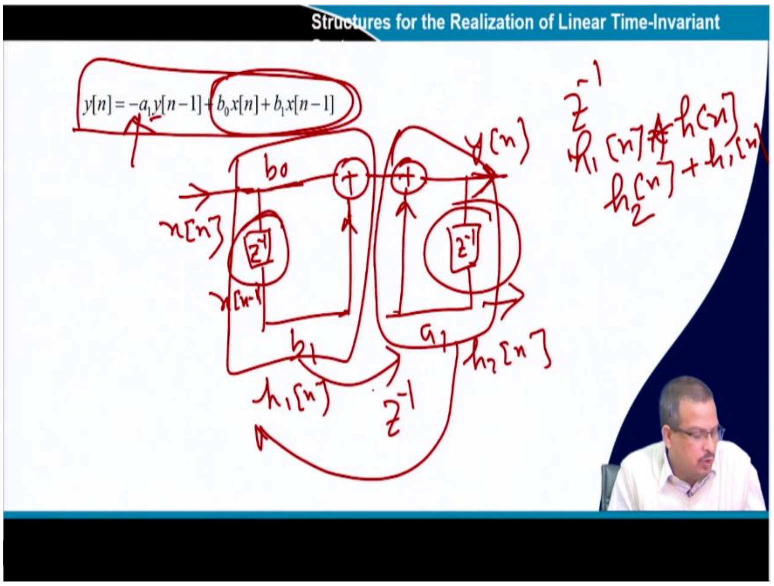
Now, what is x[0] if you see instead of? x[0] is nothing but a y[0]. So, instead of x[0], I can write down y[0+1]. So, for the cumulative sum, if I know the previous output, I just add the present input, and I can get that cumulative sum. That is why I wrote this. I multiply with this, and I can say the present cumulative sum at n equal at n is nothing but the n-1 sum plus current input, n-1 sum plus current input.

So, once I say that n-y, n-1 means previous output, that means there is some sort of recursion, and n-1 means one sample delay. So, z^{-1}, a recursion is required. So, then I call the system recursive. So, the cumulative sum is a recursive system.

Now, if I told you, can you calculate √2 using a recursive algorithm or √5 using a recursive algorithm? Can you do that? So, x[n] is equal to 2; I want the system to be √2. So, I want the recursive system, which is nothing but a calculation of the √2 using a recursive system.

First, you try, and then you watch the next video. Do not watch that; I will provide the solution in the next lecture. So, before that, first, up to this, you stop here and try to do that in a recursive system, you to calculate √2. And if you can do it, do not go for that lecture. So, I will give you the solution in the next lecture, but I will emphasize, or I will encourage you to do it in a recursive system, ok?

(Refer Slide Time: 13:13)



Then, the realization of the LTI system. This is very important. Suppose I have this system; how can I realize this system using a computer? How do I realize this system? So, I know this is y[n], ok, and this is x[n]. So, if you see that the x[n] will be multiplied by b0 and the one sample delay z^{-1}, x[n-1] will be multiplied by b1 and added up. So, I get up to this. This part I have done.

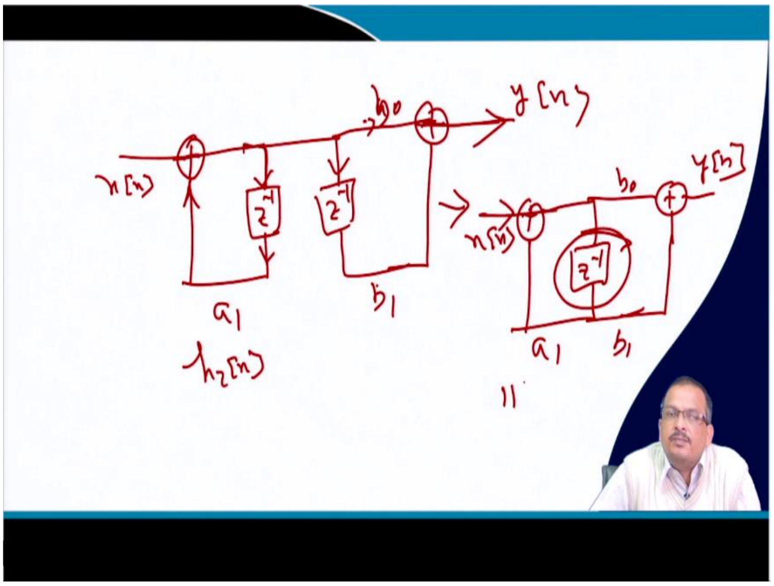
Now, this will be added up y[n-1]. So, this is my y. So, if I put z^{-1} here and a1 here, that will be added up. So, that gives me the y[n]. So, that is the realization of the system. Understand or not. So, that is the realization of the system. So, that is the signal flow diagram of this system.

Now, what do you observe? You observe two delays: the input signal delay and the output signal delay. So, what do you mean by z^{-1}? z^{-1} means one sample delay. So, I have to memorise the past input and past output. So, if I have to memorize both, then I require a two-memory location.

Now, can I simplify it to a single memory location, a single z^{-1}? So, when I require a two-memory location, it is called one structure implementation, structure one structure implementation. It is called a structure relation when I require only a single memory. So, now, suppose I want to do that. How do I do that? So, what is the diagram? This is my diagram.

So, since the h[1] convolved with h1[n] convolved with h2[n], and h2[n] convolved with or h1[n] plus h2[n] is nothing but h2[n] plus h1[n] is equal. So, let us say this is my h2[n] and my h1[n]. Now, since I know that whether I h2[n] plus h1[n] or h[2] h1[n] plus h2[n] both are equal, I can say that I can interchange this to here and this to here. So, how do I do that?

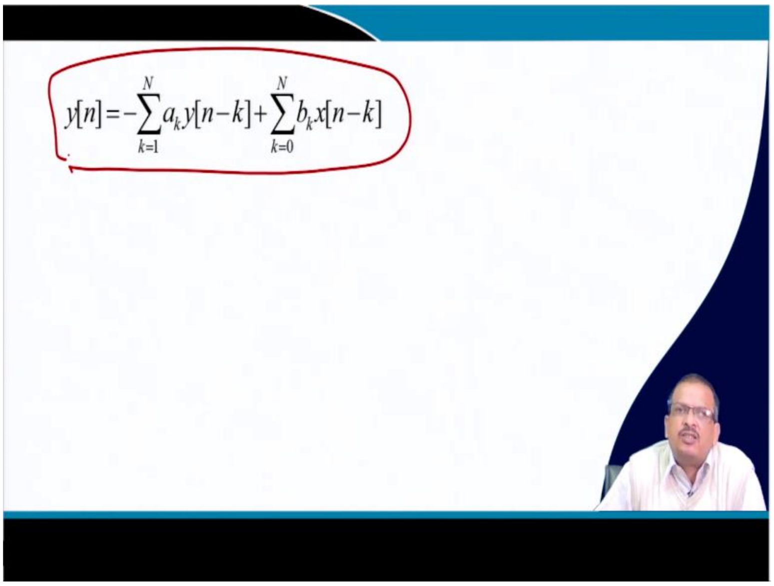
(Refer Slide Time: 16:31)



So, I have an x[n], and I will interchange z^{-1} and a1 and add up. So, this is my h2[n], plus this is my y[n]. What do I do? h^{-1}. This is nothing but a b0; this is nothing but a b1 and added up. So, this portion I have used in here and this portion I have used in here, so I interchange. So, I get this one.

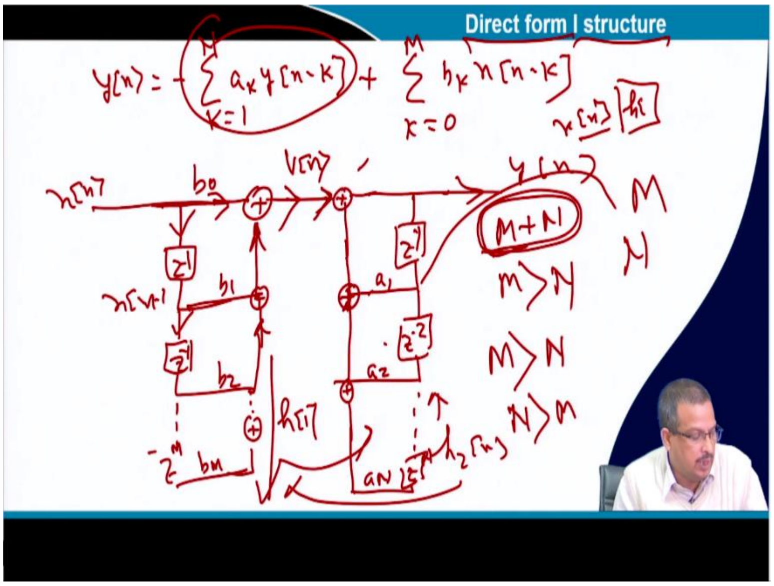
Now, I can see that the same signal is delayed by one sample. So, instead of that, I can easily say it is nothing but a z^{-1} multiplied by a1 added with x[n] and b0, b1 added with this one, I get y[n]. So, what is required? A single-element delay is required. So, that is called second structure implementation, understand.

(Refer Slide Time: 18:05)



Now, I come to the whole system. Now, suppose this is my system: k equal to 1 to N, k equal to 0 to N. So, how do I implement structure 1 and structure 2? So, direct structure 1, implement discrete structure 1 implementation or direct form 1 structural implementation or direct form 1, direct form 1 and direct form 2. So, direct form 1 structure and direct form 2 structure, what is direct form one structure?

(Refer Slide Time: 18:51)



So, if I want to draw the direct form of one structure, how do I draw that? So, here let us copy this one:

$$y[n] = -\sum_{k=1}^{N} ak * y[n-k] + \sum_{k=0}^{M} bk * x[n-k]$$

So, x[n] will be multiplied by b0 and added up. With what? b1. So, once I say b1, so this z^{-1}, I get x[n-1] here. So, this will be multiplied by b1 and added up. Similarly, b2, so again delay minus 1, b2, added up. So, dot dot dot z^{-M}, M sample delay. So, x[n] minus M I get, this will be multiplied by bM and added up.

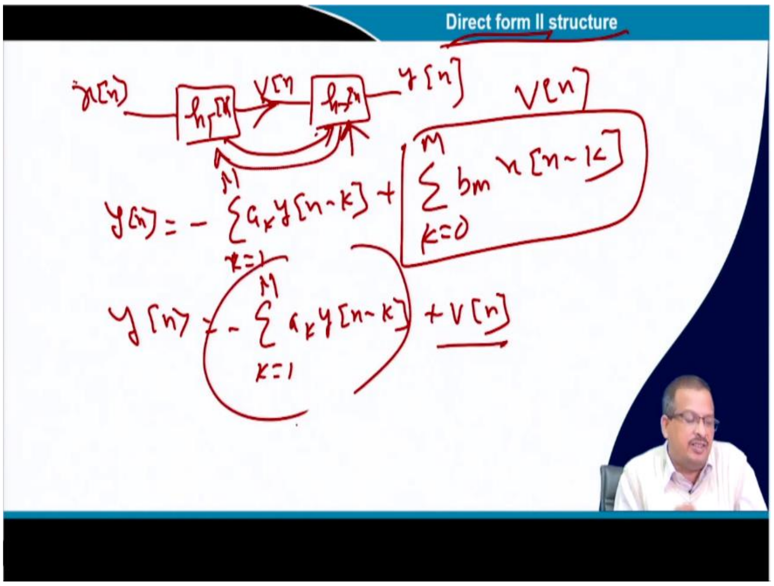
What is y[n]? I had y[n], ok. I have to delay by; so, you give that arrow unless you give the arrow, and you get 0 marks in that arrow. Now, what I have to implement this one? So, what is delayed by one sample, z^{-1}? Multiply by how much? It is nothing but a1 and added up. Similarly, z^{-2} is multiplied by a2, and added up. So, that way, I can go up to z^{-n} multiplied by a_n like that.

If I told you how much memory is required, how much, and how much delay is required? M plus N. How many additions are there? How many multiplications? You can calculate those things. So, I require an M plus N number of delay, z^{-1} or M plus N or delay element. So, one delay element means one memory. I have to store it.

Now, if M is greater than N, M is greater than N, which means this structure will be longer than this structure. But total delay is required M plus N. Now, if I want to say that instead of that, either if it is M is greater than N, I have to implement with M number of delay, or if N is greater than M, I have to implement it using N number of delay. This is called direct from two structures.

So, how do I do that? This is h2[n], and this is h1[n], so I just reversed this one. I get that. So, if this signal is, let us say, this part's output is v[n], and v[n] is the input to the second system. So, this is; so, I can say I can say that x[n] applied to an h[n] h1[n], ok. I can do it here.

(Refer Slide Time: 22:44)



x[n] apply to a h1[n], output is v[n] applied to an h2[n] output is y[n]. Now, I said interchange the system, this system will go to here, and this system will come to here. So, if I know that

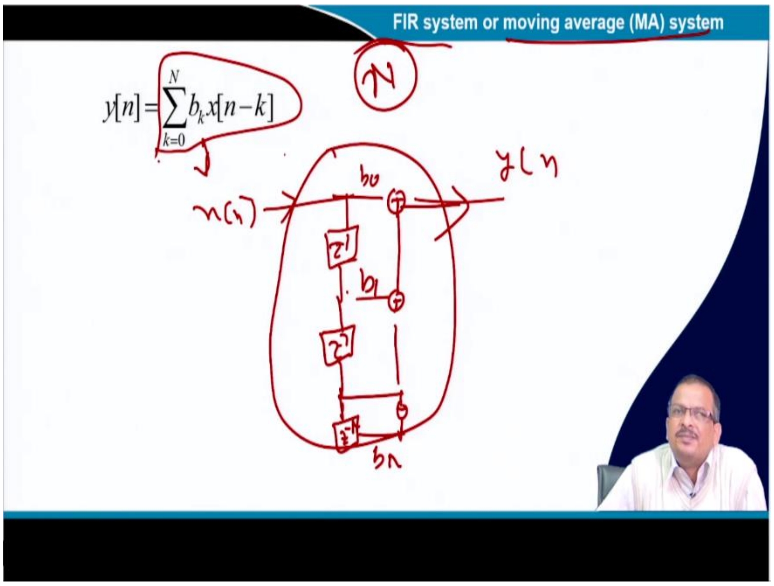
$$y[n] = -\sum_{k=1}^{N} ak * y[n-k] + \sum_{k=0}^{M} bk * x[n-k]$$

This is my system. So, I said this is v[n]. So, I can say

$$y[n] = -\sum_{k=1}^{N} ak * y[n-k] + v[n]$$

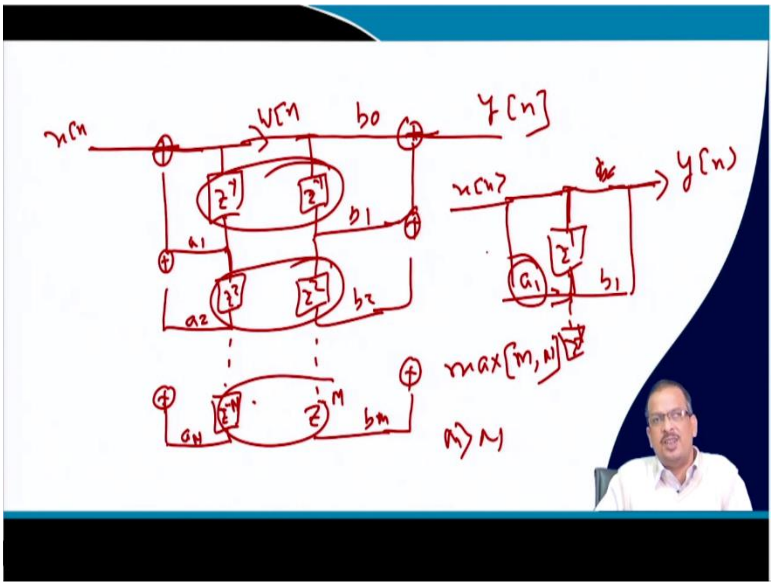
v[n] is the output here. So, v[n] is the input to an h2[n]. So, v[n] is the input to an h2[n]. So, this is my h2[n]. Now, I interchange.

(Refer Slide Time: 24:09)



So, when I interchange, I have to take another slide. So, when I do the inter change, how do I write it?

(Refer Slide Time: 24:19)



So, I said this is my x[n] and y[n]. So, the coefficient a, a_n will come to this side, so I can say it is nothing, but the z delay by z^{-1} and multiplied by a1 goes to add up here. Similarly, z^{-2} multiplied by a2, added up here.

So, dot dot z^{-n} multiplied by a_n, added up here. And this side is the same thing, z^{-1}. So, this is b0, this is b1, z^{-2}, this is b2, dot dot dot z^{-M} bm, added up. Now, instead of two delays, I can combine those things in a single delay.

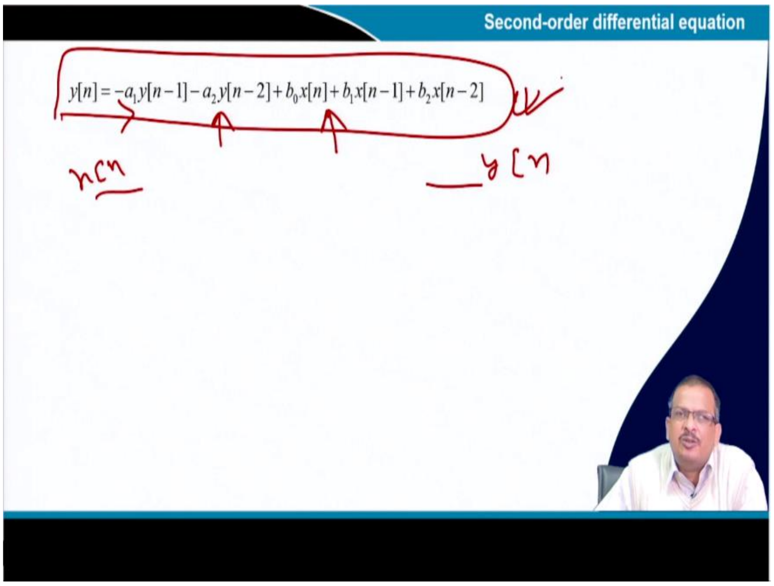
Let us say this is called w n. So, I can say x n will be applied here, w[n] will be come here, and delayed and wait a1 and this side will be b1, this side will be b0, I get y[n]. So, dot dot dot z^{-1}. So, if this M is greater than N, a number of delays requires an M number. If N is greater than M, the number of delays is N. Which one is maximum?

So, for structure 2 implementation, I require a number of delays, which is a max of M and N. Which one is the maximum? So, z^{-1} means the previous one, understand or not. So, suppose I want to implement in a computer structure using structure 2. In that case, I will store the signal, delayed signal of multi after multiplication of coefficient a1, and that factor will multiply with the b0 and b1 and go to the output. So this is called structure two implementations. Is it clear?

The next one is the FIR system or moving average system, FIR system or moving average system. Suppose what is the moving average system. It is nothing but the k equal to 0 to N x[n-k]. So, if I want to realize this system, so I know this is my y[n], this is my x[n], again without any remembering I can say z^{-1}, I get I have to multiply y.

So, when k is equal to 0, it will multiply by bz; when k is equal to 1, it will multiply by b1, all will be added up, z^{-2}, z^{-M} dot dot dot bN, added up, and I get the output. FIR system means finite impulse response system. So, N is a finite number. So, this is the implementation or moving average system. It is nothing but the moving average. Is it clear?

(Refer Slide Time: 28:24)



So, next class, I will talk about the second-order differential equation all those things. So, next, ok. Let us say I told you to draw this system's signal flow diagram using direct structure 2 or direct form 2 structure, direct form 2 structure implementation of this second order differential equation. Do it by yourself.

So, as you know, this will be x[n], and there will be a y[n]. I again know that either 1 or 2 you can do it. So, if it is two, first you draw the 1 and then interchange the interchange system, and you get that re-structure (Refer Time: 29:10). Yes, you do it, ok. So, do it in form, do it in your pen and paper and see whether you understand or not, ok?

Thank you.