

## MITOCW | 10. Linear time-invariant (LTI) systems

The following content is provided under a Creative Commons license. Your support will help MIT OpenCourseWare continue to offer high-quality educational resources for free. To make a donation or view additional materials from hundreds of MIT courses, visit MIT OpenCourseWare at [ocw.mit.edu](http://ocw.mit.edu).

**PROFESSOR:** Thank you for coming out here in the rain and the day before a quiz, but this is stuff we need to know. So I'm going to be talking about a powerful class of models for communication channels. We've already seen the kind of setup that we're talking about.

And so what we're looking to do is model a channel between this point  $x_n$ , I think my pointer is-- OK, there we go. Between  $x_n$  and  $y_n$  out there, so what we refer to as the baseband channel. So we've got  $x_n$  coming in, various things being done to it, and then  $y_n$  coming out. And we refer to this as the baseband channel.

So what's happening in here is things like-- let's see-- D to A conversion, Digital to Analog. And then there is the modulation. And then there is the physical channel. I may not have left enough space in this box. But here is the demodulation and whatever filtering, demodulation and filtering that happens in there. So there is distortion-- oh, I'm sorry. I forgot the A to D, didn't I? So let's stick that back in here. We're doing all our demodulation and filtering in discrete time, so we have an A to D converter here, and then D mod and filtering.

And there is various places here that you can get distortion and noise. So for instance, the physical channel is a source of noise. But the discrete time operations as well, the computational pieces can also introduce noise. You could have numerical noise, because you're rounding off numbers, and so on.

So there are various places that noise can originate. And there are various places that distortion of the signal can originate. So in the filtering process, for instance, or the channel process, you can get phenomena that will take what started out as a straight edge here and cause it to now get a little bit spread out and not so clean at the edge. OK, so that's what we refer to as a distortion. So there is all sorts of things in here that can account for that.

Now, when we say baseband channel, we're actually trying to distinguish it from the channel that you see after the modulation. So once you've modulated, you typically move things to some other frequency range. And so the actual transmission across the physical channel happens in some other frequency range. And so the word "baseband" here is used to distinguish the channel that we're talking about from that channel.

So this is what we're going to be focusing on. And then will later come back to talking about the modulation and demodulation pieces. So last time, I introduced a way to represent such models just as systems with an input and an output.

One thing I made a point of saying was that when we look at a figure like this-- here is a system. We've got some input sequence that's actually going in and maps to some output sequence. So I use this notation with a dot there to indicate the entire time function.

So I've got some entire time function here that goes through the system and gets mapped to some entire time function there. And I'm not telling you the details of how that mapping happens yet, but this is my abstract picture. Now, in many places you'll see people writing-- and again, I said this last time. But I want to remind you, you'll see them labeling  $x_n$  going into the system and  $y_n$  coming out.

And when you see that, you've got to think that what you're looking at is just a snapshot at time  $n$ . So this picture is what you get in a snapshot at time  $n$ , whereas this picture is the picture that refers to actually mapping the input signal, the entire input signal to the output signal.

OK, so these are two different ways of representing things. In this system, I'm not taking the value of time  $n$  and producing a value at time  $n$ . I typically will need to look at lots of values of the input to figure out any particular value of the output.

All right, I did mention briefly the notion of causality. And we'll come back to that later. But the rough notion is that-- or a good enough notion is that the system is called causal if the response at any time depends only on present and past inputs and not on future inputs. That's easy enough.

And then there were-- we were going to specialize, actually, to the case of linear and time-invariant systems. And so I want to first introduce the notion of time invariance. Time invariance says basically that, if you shift the input by a certain amount, then the output gets just shifted by the same amount. But the same input-output pair works as before. So what you're really trying to get at is a time-invariant system is one where the laws by which you compose the values of the input to get the output don't change with time.

So let's see. Let me give you an example here. Suppose I had a system whose input and output were related in this fashion. Would that, do you think, be a time-invariant system or a time-varying system?

I seem to have functions of time in here. Does that make it a time-varying system? Or is it perhaps time invariant? Yeah?

**AUDIENCE:** Time invariant, because of the law [INAUDIBLE].

**PROFESSOR:** OK, so time invariant, because the law by which you're composing things to get the output doesn't depend on time. So the point is that these coefficients are constant. So because these are constant, what you have is actually a time-invariant system.

So to get the output at any time, you're taking  $1/3$  of the output of the previous time plus twice the input of the present time. And that prescription holds along the entire time axis. So the actual value of  $n$  doesn't matter.

But if I had here some function of  $n$ , if, instead of  $1/3$ , I had something like  $1/3$  to the  $n$ , now I've got a time-varying system, because the law by which I combine things actually depends on my position along the time axis. So this would be time invariant. This would be not. So that's what this is trying to get at. Easy enough.

The other notion was that of linearity. By the way, if you read the chapter, you'll see some other examples that will help you hone your intuition for what's time invariant and what's not. For linearity, the basic idea was that you can superpose inputs and find the corresponding responses by superposition.

So if you've got the results of two experiments, the input in one experiment and the output, the input in a second experiment and the output, and then you take a new experiment in which the input is a linear combination of the previous two ones, the response will be the same linear combination of the previous two responses. So that's the basic idea here.

So linearity means that superposition works. And so this is another feature that we'll use. And for this example on top, do you think it's linear or not?

So what you really-- the way to think about it is, suppose I had an experiment A in which my output was  $y$ , in which I fed in  $x_A$ , some time signal, and I got a response, some time signal. So what that means is that this is true. This is what it means to say that this is an input-output pair in experiment A.

And now in experiment B, similarly, I have  $y_B$  satisfying this equation. So the subscript here just means experiment A, experiment B. So this is an experiment A and experiment B.

So now the question you want to ask yourself is, is it true that, if I defined a new input  $x_n$  to be, let's say-- what notation did I use there? Well, I didn't want an A and a B, did I? OK, if you ignore the notation on my slides, let's say that this is an  $\alpha x_A$  plus  $\beta x_B$ .

OK, so here is a new experiment in which I'm going to use an input that's a linear combination of the previous two inputs with some arbitrary weights  $\alpha$  and  $\beta$ . And the question then is, is the corresponding combination of the outputs in the previous experiment, so  $\alpha y_A$  plus  $\beta y_B$ , does this  $x$  and  $y$  end pair satisfy the same equation? OK, so what we want to check now is, is it true-- well, is it true that the  $x_n$  here, the  $y_n$  here will satisfy the equation on top?

And you can see very quickly that it will. And the reason is that  $y_n$  here is expressed as a linear function of the  $y_{n-1}$  and  $x_{n-1}$ . So when you substitute these in, you'll find that  $x_n$  defined this way and  $y_n$  defined this way will actually satisfy that equation. So this is what superposition requires you to test. So if it's true for every possible pair of experiments here and every pair of weights  $\alpha$  and  $\beta$  that the superposition satisfies the equations governing the system, then what you have is a linear system.

What about if I had to change this to  $1/3$  to the power  $n$ ? So I had a time-varying expression of this type. So I have a time-varying system, do you think this system would still be linear? So if you work through it, you'll see, for the same reason, that superposition still works.

So if I had  $1/3$  to the  $n$  there instead of  $1/3$ , I get a time-varying system, but I could still superimpose solutions. It would be a linear time-varying system. Now, we don't want to spend too much time teasing all these apart, because what we'll be focused on is linear and time-invariant systems. And you'll actually come quickly to recognize them.

OK, I defined last time also a pair of special signals which you've seen before, the unit sample signal which has the value 1 just at one point and the unit step signal. So let me just sketch them out for you here. So the unit sample, this is a signal  $\delta[n]$  which is an entire signal. It's not just the number 1 at time 0. It's the entire signal. That's the unit sample function.

There is another notation that's also sometimes used, which is  $\delta[n]$ . So this notation is a little bit more evocative of a function, whereas here, you are often tempted to think of it as a number. This says, what I'm looking at is a function. It's a unit sample function. And the 1 is at the value 0.

So if you had  $\delta[n-3]$ , that would be this function shifted from 0 to 1, 2, 3. So the 1, value 1 would sit at time 3. Another notation for that would have been this. Sometimes that notation is useful also in making sense of expressions that you're looking at.

OK, so this was the unit sample function. And then the unit step function steps up from 0 to 1 at time 0. That's the unit step function.

And we also talked about the response to these two inputs. So you see them up there. And now my question is, if a unit sample signal at the input produces the unit sample response  $h_n$  at the output and  $u_n$  produces the step response, and if what you have is an LTI system in here-- so it's the same LTI system that we're talking about-- can you actually relate the two?

So the question is, can you relate the unit sample response and the step response? Do I need to give you both if I have an LTI system? Or does it suffice to give you one?

So here is one way to think of that. This, by the way, is the same LTI system. Maybe I should indicate that more explicitly by, let's say, it's a specific system, system zero. And with the same system, I'm trying to deduce the results of another experiment.

So if we're thinking superposition, can you tell me how to write the unit sample function as a linear combination of unit step functions, maybe delayed unit step functions, scaled unit step functions? Yeah?

**AUDIENCE:** [INAUDIBLE]

**PROFESSOR:** OK, would it be-- you said  $u_n$  minus  $u_{n-1}$  plus 1? Or is it  $u_n$  minus 1?  $u_{n-1}$  minus 1? So what we're saying is, take this unit step and then subtract from it a unit step delayed by 1.

OK, so here is  $u_{n-1}$ . If we took the unit step and subtracted from it a delayed unit step, delayed by 1, the result will be just that value 1 at time 0 will survive. Everything else will cancel out. Is that what you had in mind? OK.

So if  $\delta_n$  can be written as that linear combination of unit steps, can you tell me how to write  $h_n$  in terms of unit step responses? We're talking about an LTI system. I took that out, but we're still talking about an LTI system here. Somebody who hasn't spoken maybe? Yeah?

**AUDIENCE:** It should just be  $x_{n-1}$  of  $u_{n-1}$ .

**PROFESSOR:** Yeah, OK. So superposition says that, if you've got an input that's a linear combination of inputs for which you know the results of the experiment, then the corresponding output is the same linear combination of the outputs for that experiment. So this is going to be  $s_{n-1}$  of  $u_{n-1}$ . So you can actually deduce the unit sample response, given the unit step response for an LTI system.

So let's see. We've used linearity. Have we used time invariance? We used linearity because we said, here is an experiment in which the input is a linear combination of inputs that we know the responses to. Where have we invoked time invariance? Anyone?

The superposition idea was part of the definition of linearity. Because a system is linear, if the input is a superposition of two inputs for which you know the response, then the output is the corresponding superposition of the responses. That seems like I've only used superposition there. Have I actually used time invariance as well? Yeah? Sorry?

**AUDIENCE:** [INAUDIBLE]

**PROFESSOR:** I've used it in concluding that, if I put in  $u_{n-1}$ , the response is  $s_{n-1}$ . So I've used time invariance as well as linearity here to come up with this statement. OK, good.

So this is what I have on the slide. And you've figured it all out already. We've arrived at this equation. Now, if I want to turn it around and write  $s_n$  in terms of the unit sample response, I can do that as well, except this is analogous to integrating a differential equation.

What we have is a difference equation here. And when you come to integrate, well, in discrete time, what you do is summation instead of integration. You need to assume an initial condition of some kind.

And so it turns out if you assume that, way back in the past, the value of the step response was 0, then you can actually go from this description to a description the other way, relating the step response to the unit sample response. OK, so if I have a causal system, for instance, so the causal system, it's got no response until the input hits it. So when I put a unit step in, I'm not going to get a response until time 0. And so I know at minus infinity, the step response was 0. And I can move forward from there.

OK, so you can actually relate the step response to the unit sample response the other way as well here, where the summation is from minus infinity to the  $n$  that you're interested in. We'll be dealing right through with causal systems. If there are any deviations from that, we'll point them out. But basically, we'll be dealing with causal systems.

OK, so let's-- this is an identity we'll be wanting to play with a bit. So let me put it up here. So the step response, let's say, for a causal system is going to be summation from  $k$  equals minus infinity to  $n$  of  $h[k]$ . So I take all the values of the unit sample response up to the present time and sum them together to get the value of the step response at the present time.

OK, so let's look at an example here. Here is the unit sample response of a particular LTI system. Is this a causal system? So this is the response to a unit sample.

So the input was 0 everywhere except for a value of 1 here. And you see that the response actually happens subsequent to that input. So if the response starts at time 0 or later for an input that started at time 0 or later, then what you are looking at is a causal system here, certainly in the case of a unit sample response.

OK, so what's the step response going to look like, then? Anyone want to say in words? Yeah?

**AUDIENCE:** [INAUDIBLE]

**PROFESSOR:** OK, so the step response, if we're evaluating it at times over here, we're summing all the values of  $h[k]$  from minus infinity up to the present time. So the step response is 0 here, is 0 here, is 0 here. And then at time 3, the step response jumps to 1, and from then on stays at 1.

So the step response is just that delayed step. And it kind of makes sense, because the kind of system we're talking about must be a delayed by 3 system here, because we put in a unit sample input, a unit sample function. And what came out, if you look at it, was actually  $\delta[n-3]$ .

It was the unit sample function delayed by 3 steps. Is the height-- the height is unchanged, right? The height of still 1. So this must be a delay by 3 system we're looking at. And sure, if we put in a unit step, we're getting-- or sorry, yeah, if we put in the unit step, we're going to get a response that's just the step delayed by 3. I'm going-- sorry, yeah?

**AUDIENCE:** Yeah, maybe I'm just a little confused here. But why is it from negative infinity to  $n$  and not like from  $n$  to positive infinity.

**PROFESSOR:** This is because of my assumption assuming  $s$  of minus infinity was 0. So I need to have a boundary condition from which I start inverting. So just to go back-- let me just go back a second here. Oh, where am I going?

OK, so we derived this first expression. If we want to turn it around, well,  $s_n$  is  $h_n$  plus  $s_n$  minus 1. And then I can solve for  $s_n$  minus 1. I can keep stepping backwards.

But at some point, I need an actual value so that I can close off that expression. And if you're talking about a causal system, then what you're guaranteed is that the step-- if you're talking about a causal linear system, because the all-zero input produces the all-zero output, and it's causal, you can actually deduce that the step response at time minus infinity must be 0. The input hasn't yet arrived. Therefore, the output must be 0.

**AUDIENCE:** So  $h$  of 5, does that mean [INAUDIBLE]?

**PROFESSOR:**  $h$  of 5 is just a number. It's not a function, right? If I write something like  $h$  of 5, it's just a number. So it means the value of the unit sample response at time 5.

OK, this takes a little getting used to, but let's do another example here. So here is another unit sample response. This is more complicated, though. I put in a unit sample. And what comes out is a response that-- well, it still starts at time 0.

So I'm talking about a causal system. Everything to the left is 0. And the signal takes a value 0.2 for some number of steps and then settles to 0.

So the question then is, what is the step response? So if you imagine that what you're doing to find the step response at any time is summing this from minus infinity up to that time, you will see that the step response is linearized like that. And then it settles out.

OK, so you can get one or the other. And there are other examples on the slides. I won't go through all of them. I'm going a little slow here, because we miss a recitation tomorrow. Recitations tomorrow are office hours, so I wanted to actually give you a few examples.

Here is a case where the unit sample response increases linearly and then stops. And so the unit step response actually starts to accelerate quadratically and then stops. This is the discrete time version of integration that we're looking at. Yeah?

**AUDIENCE:** [INAUDIBLE]

**PROFESSOR:** Oh, sorry, this thing?

**AUDIENCE:** Yeah.

**PROFESSOR:** Ah, OK, ignore that notation. First of all, it's bad notation. But these are figures that I got from somewhere.

If I was doing it from scratch, I wouldn't have put it in. But I'll explain it. That's the notation for convolution. I don't actually like that notation.

OK, examples of this type-- now, here is one important thing for you to get a feel for. Notice in all these examples, the unit sample response settles down to zero after some time. So you hit the system with a unit sample function at the input.

So you hit it with a value 1 at time 0 and nothing else. And it responds. And it response for a while and settles.

Now, that's not true for all systems, that they settle in finite time. A typical system might ring indefinitely, might respond indefinitely to a kick. Here, are all these examples are ones where the system has a transient and then settles down. And so what you expect to see in the step response is there is a transient.

And then it settles down. The difference is, in the step response, it settles to another value. It doesn't come back down to 0. So when this comes back down to zero, what this is settled up to is sort of the integral of this. It's the area under this, but we're talking about discrete time functions, not continuous time functions. So the value that it's settled to here is the area under this.

So the duration of a unit sample response gives you some feel for how long a transient lasts. So if you've got a channel and you hit it with an input, you know that the transient will last about as long as the unit sample response lasts. So the transient and the step response shows that clearly.

You can get more elaborate sorts of unit sample responses. Here is one that changes sign. And correspondingly, what you find with the step response is that it's not a monotonic increase to the final steady state.

There is actually some oscillation before it settles down. But it's the same idea. You're computing the area under this, if you like, but the area goes positive, and then slightly negative, and so on-- well, positive and then less positive. And so that's what you're seeing up there.

Now, why do we talk about step responses so much? Well, it turns out that for a lot of what we do with signaling on communication channels, we're signaling with signals of this type, on-off-type signals, or plus-minus signals, the sort of square-wave-type signals or rectangular-wave signals. And these can be thought of as combinations of unit step functions. You may have seen this in recitation last time as well.

So you can take an input of this type and write it as a linear combination of unit step functions. A unit step function that has its step at 0, minus 1 that has its step at 4, plus 1 that has its step at 12, minus 1 that has its step at 24. So if you combine those, if you add up all of these, you're going to get that input.

So then it's back to this game again. If the input is a linear combination of unit steps scaled and delayed, then the response is going to be the same combination of unit steps. So that's what the response will look like.

So here is the step un gives rise to sn. Therefore, minus un minus 4 will give rise to minus s of n minus 4, and so on. So knowing the step response, you can actually say what the response of the channel is going to be.

All right, we've seen this visually too. I did an example last time where what went in was that square wave and what came out after we had done the demodulation of the filtering was a response that sort of had the features of what went in, but it was a little distorted. So you can see that what we're looking at here, for instance, is the character of the step response, because what went in at this point-- right now the input on the output are at rest. You've forgotten about what happened before.

And now the input jumps up. Well, the output doesn't jump up all the way immediately. It's got a little transient before it settles. So what you're looking at is really the step response of the channel, where the channel includes all these pieces.

It's everything including the filtering. In this particular example, if you go back and look at those slides, this was all entirely due to the local averaging that we were doing in the filtering here. But it does give you some kind of a distortion. OK, so the step response is important to figuring out the shape of the output of a channel.

Here is another example that has a more rounded kind of step response, but it's still the setup response we're looking at. So here is the input step. And here is the response to the step.

Again, there is this notation. And I've said, ignore this for now. We'll explain it shortly. OK, and once you've got the step response at the output, you're ready to start thinking about how you'll detect whether it was a 0 or a 1 that went in. So you might set a threshold, pick times at which you're going to sample. And then you come up with your call of what the input is, so 1, 0, 0, and so on.

So this seems all benign enough. But now what if you decide you want to get that information across the channel faster? So you want to signal faster? So what you're going to want to do is put that same information, the transition from 1 to 0 to 1, 1, 1, 0, 1, and so on, you want to squeeze that into a shorter length of time.

So suppose this is what you send over that same channel. Well, now you, again, are going to superpose the step responses. But what's happening now is you've gotten so ambitious with how fast you want to get the bits across that you're not giving the step response time to settle.

So over here, yes, there is time. The step response went up and settled because you had three 1's in a row over there. But now you're going down to 0 for one time instant and then jumping right back again. Well, here is the flipped over step response. And it doesn't have time to make it all the way down. It's jumped up again.

OK, so if you get very ambitious with your signaling to try and get more of the bits across, you're going to start seeing the limitations imposed by the channel. The channel can only respond so fast. And you can drive it faster. So it's important to have a feel for that as well.

So when the channel starts to respond like this, you become much more susceptible to noise. So for instance, if there was a noise spike at this point, you could well end up with a received sample that was above the threshold. And then you'd wrongly decode the 0 as a 1. So taking account of the channel characteristics is important when you're setting a signaling rate. You might want to get information across quickly, but you have to take account of the fact that the channel needs some time.

OK, so much for steps. We'll come back to that later. We can do the same kind of thing with unit samples. So here is-- and in fact, the rest of the lecture, we're going to be talking about making up a signal as a weighted combination of unit sample functions.

So take an arbitrary signal like this. Think of it as-- let's see, this starts with the value of something, 0.75, I guess, at time minus 2, and then a value of minus 0.5 at minus 1, and so on. So here is your input signal  $x_n$ . And I want to think of it as made up of a bunch of unit sample functions.



So what are the unit sample functions? Well, here is one that's centered at minus 2 but scaled by the value that the input signal has at time minus 2. Here is another one centered at minus 1, but scaled by the value that the input signal has at time minus 1, and so on.

So what I'm basically doing is decomposing the input into a weighted combination of unit samples. And you can always do that. And it looks a little magical when you put it into notation like this, but that's basically all that it's saying.

So to make sense of this, think, for instance, of putting in an actual number here. So if I wanted  $x$  at time 3, well, I'm going to set  $n$  equals 3 on the right-hand side and evaluate the sum. Well, the only value that survives is the value for which  $k$  equals 3.

So I'll pull out  $x_3$ . So this kind of seems tautologous. But it's a way to represent a general input as a weighted combination of delayed unit samples.

OK, so if that was what went in, you're in the position now to tell me what comes out. So I'm talking about an LTI system. I'm talking about an LTI system. And the input  $x_n$  is a weighted combination with these weights of a bunch of unit sample functions.

Well, let's actually-- well, let me actually write this the other way as well. Another way to say this is, here is a time function going in. It's a weighted combination over all possible values of  $k$  of  $x_k$  times-- and this is my other notation, remember, for unit sample functions. So I'm saying this is a unit-- sorry, this should be  $\delta_{n-k}$ .

What should it be? Yeah, OK, so this is another way to write the same thing. We've chosen to write it this way. And actually, I find that simpler. But if you want to be reminded that what we're talking about here is an entire time function, then this is a notation that you might go to. Yeah?

**AUDIENCE:** Why is the first sum [INAUDIBLE]

**PROFESSOR:** OK, good question. Because I'm right now allowing my input to have values that extend from minus infinity to plus infinity. So I'm taking an arbitrary function.

If we're talking about an experiment in which the input starts at time 0, then we can actually simplify these. I'll show you that. OK, let me actually erase this, because I don't want to confuse you with that.

OK, so if this is what goes in-- it's a weighted combination of unit sample functions delayed-- what is it that must come out? OK, so what are we working with? We're working with the fact that, if  $\delta_{n-k}$  goes into our system, what comes out is  $x_k$ . So if it's a weighted combination of  $\delta_{n-k}$ 's that goes in, what's the response, given that this is an LTI system? Someone who hasn't spoken today, maybe? Do you want to try? yeah?

**AUDIENCE:** [INAUDIBLE]

**PROFESSOR:** The Same weighted combination of those responses-- so it's going to be summation over all  $k$ , the same weight. So it's going to be the  $x_k$ 's. But now here is the responses.

So what have we been able to do? We've been able to write down what the output looks like for an arbitrary input in terms of the unit sample response. If you give me the unit sample response for an LTI system, I can write down the general response, the response to a general input. And this is what we refer to as a convolution or a convolution sum. That's a convolution.

It may look mysterious. So let's actually do it. Let's do it step by step again. Here is our LTI system. If I put in a unit sample function-- OK, so this is the unit sample function going in-- I get some response.

And let's say this is 0. This is 1, 2, and so on. The response, we refer to as the unit sample response  $h_n$ . So what is this value? This is the value  $h_0$ . This is the value  $h_1$ . This is the value  $h_2$ , and so on.

OK, what if what goes in is actually the value  $x_0$  at this time and 0 everywhere else? What's the response in that case? So this is just a scaled version of the unit sample function.

Instead of 1 going in, I'm having  $x_0$  go in. So the response is going to be-- what do we have? The same response? Twice the response? What comes out?

**AUDIENCE:**  $x_0$  times that.

**PROFESSOR:** I didn't hear. Where did that come from? Yeah?

**AUDIENCE:**  $x_0$  times that.

**PROFESSOR:**  $x_0$  times that-- OK, so what we'll get is  $x_0$  times  $h_0$  coming out at the first time, and then  $x_0$  times  $h_1$  coming out of the second time, and then  $x_0$   $h_2$  at the next time. And if I keep going, I get  $x_0$ , let's say,  $h_n$  at this time, and so on. What happens if now it's not that, but it's some value  $x_1$  going in at time 1 and 0 everywhere else?

So this is starting-- this is centered at a time 1, not at time 0. And it's scaled by  $x_1$ . So what is it I'm going to see at the output? There was a hand somewhere there previously. Maybe you can answer now. Yeah?

**AUDIENCE:** The same graph translated 1 over and scaled by  $x_1$ .

**PROFESSOR:** Right, exactly. So what's going to happen is, nothing will happen here. I'll get  $x_1$   $h_0$ ,  $x_1$   $h_1$ , and so on,  $x_1$   $h$  of  $n$  minus 1. And it keeps going.

And you keep going here as well. You keep stringing in these. Each one of these will fire off a scale of the unit sample response, but delayed appropriately. And so at the next time, what you're going to get here is  $x_2$   $h$  of  $n$  minus 2. And it keeps going.

And what if you're interested in the value at time  $n$ ? OK, so you look along here. And you've come to the value of time  $n$ . So it's going to be the sum of all of these, if your input is the sum of all of these, right?

If your input is the sum of all of these  $x$ 's, your response is going to be the sum of all of these. So what's the sum of all of these? Well,  $x_k$   $h$  of  $n$  minus  $k$ . That's all there is.

There is nothing-- there is no magic to this. It's just invoking linearity-- that's the scaling part of it-- and time invariance, which is the delaying part of it. It's as simple as that.

All right, so we'll be seeing this notation a lot. You probably recognize this kind of notation from the convolutional coder as well. And we don't want to keep writing these sums. So here is the notation that we use.

We say that  $x$  is convolved with  $h$ . And we are interested in the value at time  $n$ . So this operation of-- this summation here is referred to as a convolution, as I said.

I'm telling you what value of time I'm interested and the response at. That's the  $n$ . So that's what this notation is.

The  $k$  here is just a dummy index. We're summing over the  $k$ . It doesn't matter what I called it. I can call it  $j$ . I can call it  $l$ . It doesn't matter.

The important thing is this  $n$  here tells me at what time I'm looking for the response. And that's why that's the argument that I stick in here. All right, so all that's on the slides, but we've actually derived it ourselves here.

Now, again, some gripes about notation-- you'll find, if you look in most engineering textbooks, that this would be written  $x_n \star h_n$ . And I can't tell you how much I detest that notation. You'll never find it in a math book.

The problem here is that this  $n$  is being asked to do too many things. The  $n$  is supposed to suggest-- the  $x_n$  here is supposed to suggest we're interested in the whole time function. You would have been better off calling it  $x$  dot, but, OK, we're used to thinking of  $x_n$  as also denoting an entire time function.

This  $h$  is supposed-- the  $h$  of  $n$  is supposed to denote an entire time function. But the  $n$  is also supposed to tell you at what time you're interested in the response. So that index is just doing too much work.

And it ends up being confused and confusing notation. So when you're in your downstream classes from here, if you find an instructor using that, make sure that you give him or her grief and say you really can't make sense of that, because this is much cleaner notation. This is what conveys what's actually going on.

All right, I'm going to skip over a few things. I just want to suggest some properties here. And then we'll come back to more of this next time.

OK, so it turns out that convolution has nice properties. For instance, the order doesn't matter. You can write  $x \star h$  here, but it's the same as  $h \star x$ . And that just comes from making a change of variables in here.

If I call this  $m$ , then  $k$  is equal to  $n$  minus  $m$ . And I get something that looks different. But it's really the same thing. So this is the same as  $h \star x$ .

So convolution, you can interchange orders. There is some conditions on this, but we can talk about them later. You can associate them. You can group them arbitrarily.

And you can distribute convolution over additional functions. So all of this actually makes it-- this is very powerful, because it allows you to deal with combinations of systems. And I'll just give you one example. And then we'll quit.

So here is an example of the kind of thing you can do. Suppose you have an input going into one system, LTI, with a unit sample response  $h_1$ , and then the output of that going into a second system, LTI with unit sample response  $h_2$ , and then producing an overall output  $y_n$ . Well, so how do you get  $y$ ?

It's  $h_2$  convolved with  $w$ . I've dropped the argument  $n$  because I want to do this just for general values. But  $w$  itself is  $h_1$  convolved with  $x$ .

Now, I can group these any way I want. So I can, because convolution is associative, I can put those parentheses where I want. So this is equal to the expression at the end.

But that's the same result I'd get by putting this input into a single LTI system whose unit sample response was the convolution of the two individual ones. So I can start to collapse two systems into one equivalent LTI system. And that kind of thing ends up being powerful.

But I can also interchange orders. So from here, you can go to this, which then tells you that, for an LTI system, if you've got systems in cascade, if you've got two LTI systems in cascade, actually, the effect on the output is the same, whatever order the input-- the systems are connected in. You might ask yourself whether the same is true if this was linear but time varying. And you should hopefully find out that, in general, for linear but time-varying systems, you can't do this.

So really, linearity and time invariance is what it takes to be able to attain this. OK, let's leave it at this for now. And we'll pick up again-- well, you pick up some in problem set four and also next week.