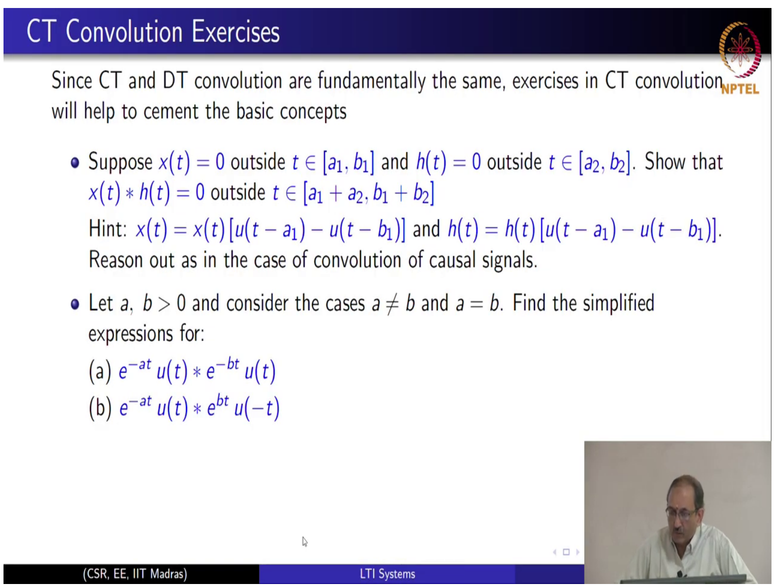
Digital Signal Processing

Prof. C.S. Ramalingam
Department Electrical Engineering
Indian Institute of Technology, Madras

Lecture 15:
LTI Systems (2)
Properties of Convolution: commutativity, associativity, distributivity
System interconnections
Role of impulse response in an LTI system

(Refer Slide Time: 00:17)

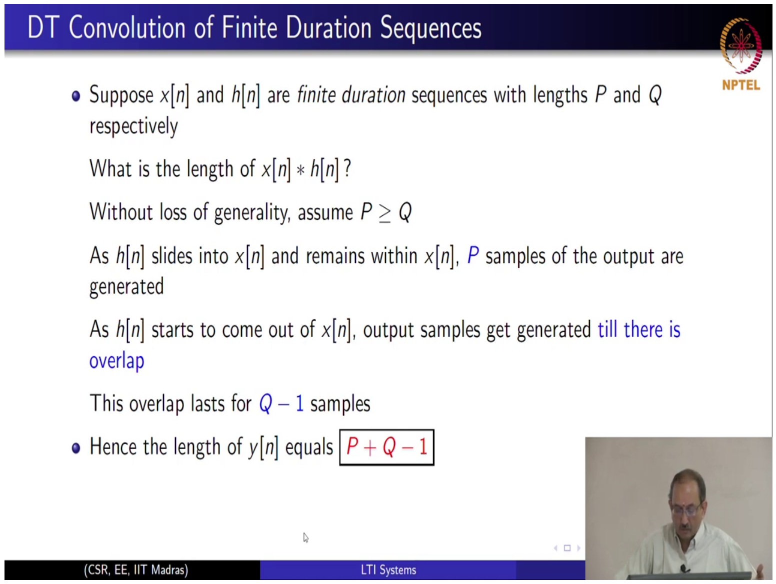


Let us continue from where we left last time. So, we being going through a very quick fire review of a systems and their properties and we also focused on LTI Systems. All of this is review for you because what happens in discrete-times is a close parallel of what was happening in continuous-time.

And we focused on convolution which is very important aspect of LTI systems and since an arbitrary signal can be expressed as a linear combination of scaled and delayed impulses. If you know the impulse response you can find the response to an arbitrary input and the output in such cases is given by the convolution of the input and the impulse response.

And these are some of the exercises in continuous-time that you need to be comfortable with. In particular convolution of a causal sequence and anti causal sequence, you need to be able to, when you are doing algebraic calculation, you should be able to fix the limits and get the output for all values of shifts. So, that is very important.

Refer Slide Time: 01:33)

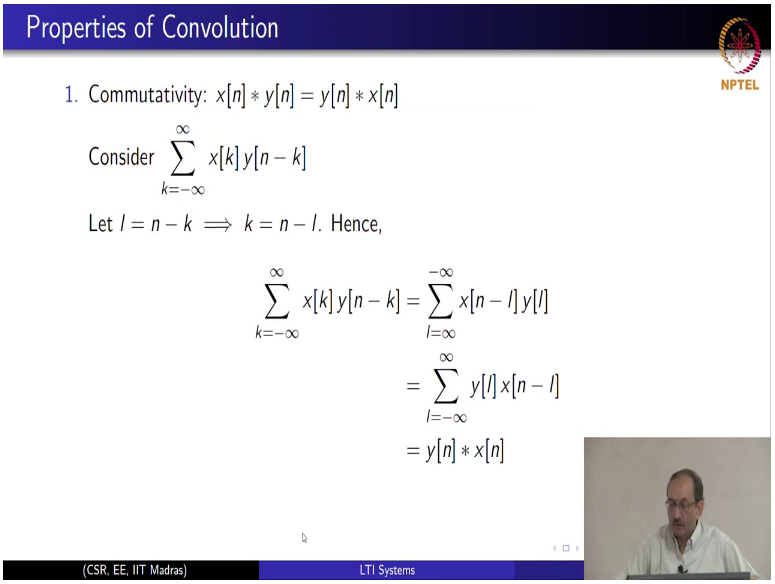


So, let us continue with this. If there are two sequences, $x[n]$ and $h[n]$ which are of finite duration, then if their lengths are P and Q; then the length of the convolution of these two sequences is easily determined. And without loss of generality you can assume $P \geq Q$. As $h[n]$ slides into $x[n]$ and remain within $x[n]$, then P samples of the output are generated; so that is easy to see.

And as $h[n]$ starts to come out of $x[n]$, the output samples generated get generated till there is overlap. And the overlap lasts for $Q - 1$ samples and therefore, the total length is $P + Q - 1$, alright. And you get $P + Q - 1$ because when the overlap seizes, one shift means a shift by one sample. So, this is the difference between continuous-time and discrete-time, when the overlap seizes, the difference is just one point and one point does not contribute to the length as far as a continuous-time signals are concerned.

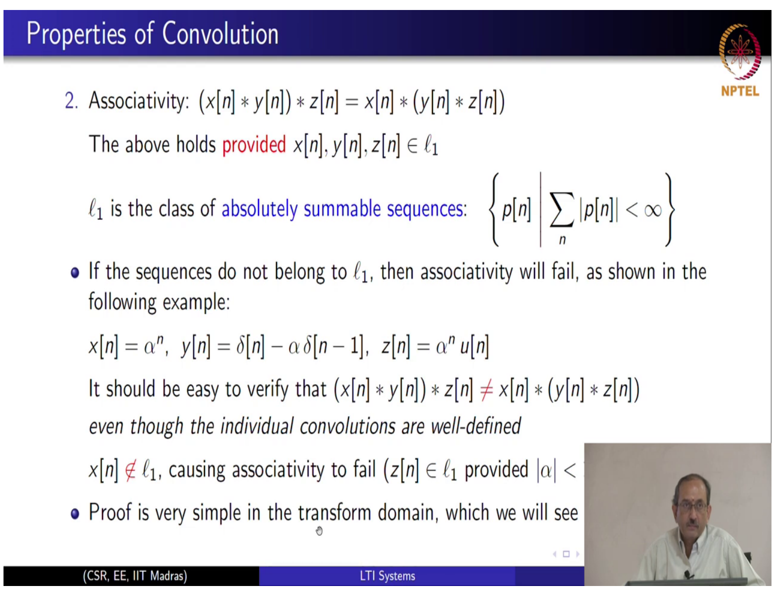
Therefore, if the duration of one pulse is t_1 and the duration of the other pulse is t_1; the overall duration of the continuous-time case is $t_1 + t_2$. Whereas, in the discrete-time case, it is not $P + Q$, but it is $P + Q - 1$ because when you seize to overlapp, difference is 1 sample.

(Refer Slide Time: 03:30)



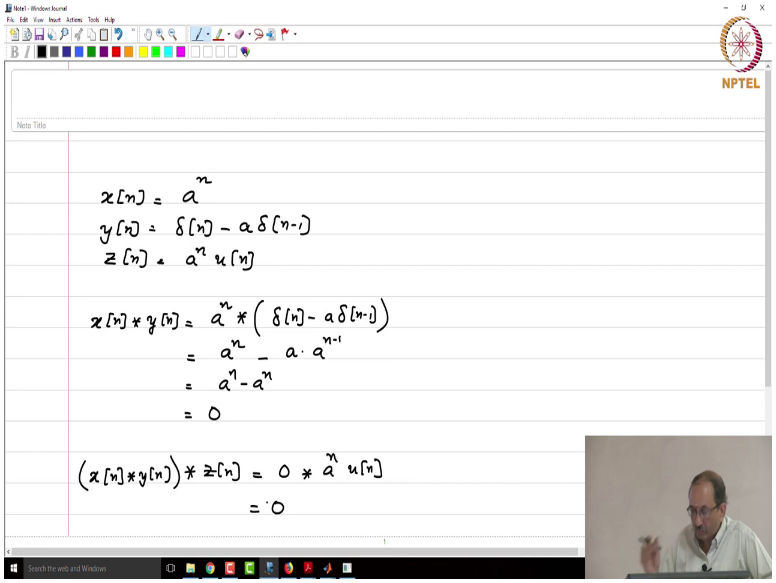
Now, let us look at properties of convolution that you must already be familiar with convolution is commutative. So, x convolved with y is a same as y convolved with x and this is easily seen. All you need to do is you need to make a change of variable $l = n - k$ and if you did that, just a couple of steps you will be able to see that, $x[n] * y[n]$ is the same as $y[n] * x[n]$; so, this is commutative.

(Refer Slide Time: 04:01)



The next property that we want to see is associativity of convolution. So, $(x[n] * y[n]) * z[n]$ is the same as $x[n] * (y[n] * z[n])$. And this is similar to the property in continuous-time and one way of seeing this is if you convolve in time; you multiply in the transform domain and multiplication is associative and hence associativity of convolution can easily be seen by looking at the corresponding behavior in the transform domain.

(Refer Slide Time: 04:54)

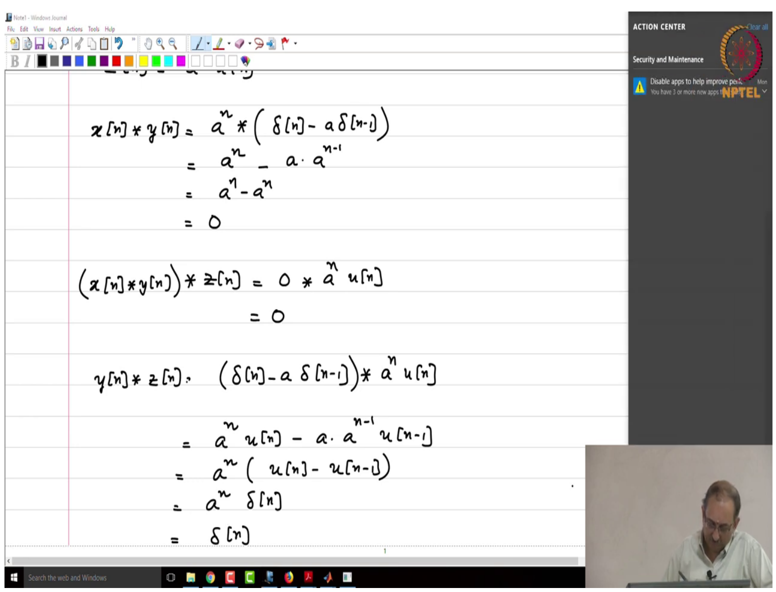


Now, let us examine this a little more. So, let us take $x[n]$ to be a^n; $y[n]$ is $\delta[n] - a\delta[n-1]$, $z[n]$ is $a^n u[n]$, so these are the three sequences.

Now, let us find out what $x[n] * y[n]$ is. So, this is nothing, but a^n convolved with $\delta[n] - a\delta[n-1]$ and this is what? I think you need to push pen and paper rather than staring at the screen. So, this of course, a^n convolved with $\delta[n]$ is a^n, a^n convolved with $\delta[n-1]$ is a^{n-1}.

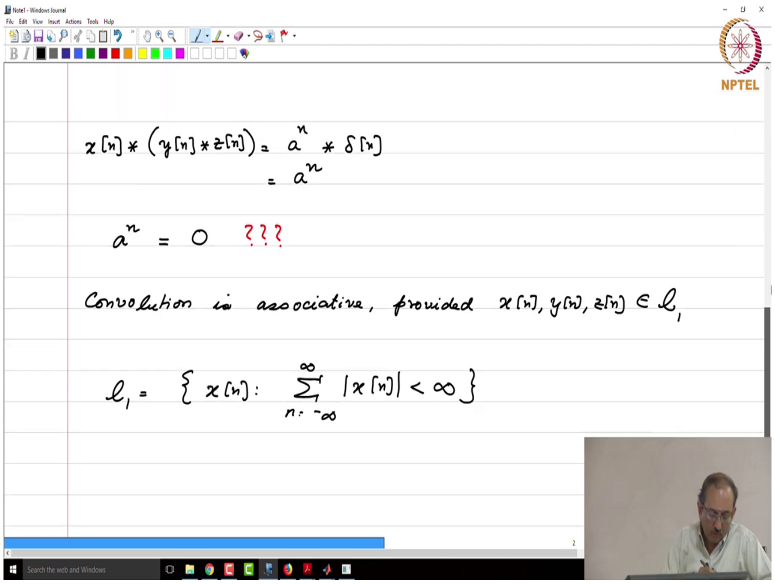
So, this is indeed $a^n - a^n$ and the answer is 0; now $x[n]$ convolved with $y[n]$, now we want to convolve this with $z[n]$. So, this is indeed 0 which is the result of the first convolution convolved with $a^n u[n]$ and this of course is 0.

(Refer Slide Time: 07:15)



Now, let us do the other convolution. So, you want $\delta[n]$ convolved with $z[n]$. So, this is $\delta[n] - a\delta[n-1]$ convolved with $a^n u[n]$. So, this is nothing but $a^n u[n] - a.a^{n-1}u[n-1]$. So, this is $a^n(u[n] - u[n-1])$; $u[n] - u[n-1]$ is $\delta[n]$ and $a^n\delta[n]$ is $\delta[n]$ itself because this picks out the sample at $n = 0$, correct.

(Refer Slide Time: 08:58)



So, now let us convolve $x[n]$ with the result of $y[n]$ convolved with $z[n]$; $x[n]$ of course is, a^n convolved with $\delta[n]$. So, this answer should be a^n right. Therefore, $a^n = 0$. Convolution is after all associative right, either a power n equal to 0 is true or convolution is not associative, right.

Student: Sir, in first part when we are convolving x and y. Yes.

Student: We are considering that a to be a constant, that is not a (Refer Time: 10:09).

Say that again. So, a^n is indeed a constant; so why is that a problem?

Student: But what is n a if the modular of $x[n]$ into $\delta[n-1]$ is equal to **(Refer Time: 10:26)**.

No. So, what is it in this step that fails based on what all objection you are raising.

Student: What sir?

No. So, you said a being a constant is the problem, correct?

Student: Yeah.

So, where is it that is causing these steps to fail?

Student: Suppose downward and when we (Refer Time: 10:53).

Yeah.

Student: And $x[n]$ into $\delta[n - n_0]$ is equal to.

$x[n - n_0]$ yeah.

Student: That will not apply.

Why? Why that will not apply? No why is that thing will not apply? That surely applies.

Student: (Refer Time: 11:21).

Right, unless I am not getting exactly what you are trying to point out.

Student: What if that one of those variables is not LTI?

Here, we are talking only with properties of convolution, that convolution happens to be something that LTI systems satisfies; that is a different reason. So, why do you want to bring LTI systems here? This is purely a property of convolution if someone did not know LTI systems he or she can still grappled with this particular example and then see something is not quite right.

So you being told convolution is associative, what you probably have not being told is that convolution is associative under certain conditions alright. So, convolution is associative provided $x[n]$, $y[n]$ and $z[n]$ belong to l_1; l_1 is the space of all absolutely summable sequences. So, l_1 is the set of all sequences such that

$$l_1 = \left\{ x[n];\ \sum_{-\infty}^{\infty} |x[n] < \infty| \right\}$$

So, this is the space of absolutely summable sequences; only if all the three sequences under consideration belong to this class, only then we can make the conclusion that convolution is associative; right here we have come up with an example where associativity is not valid and the reason is this condition is violated. So, which sequence violates not belonging to l_1?

Student: x.

$x[n]$?

Student: (Refer Time: 14:20).

And $z[n]$ and $|a| \geq 1$. So, when $|a| \geq 1$; $z[n]$ does not belong to l_1. Even when $|a|$ is strictly less than 1. For that particular case, $x[n]$ is not an l_1 sequence. Therefore, when you are told that convolution is associative, implicitly in that statement is the assumption that all these three sequences belong to the space of l_1. So, here is an example where that condition fails.

Student: Why that?

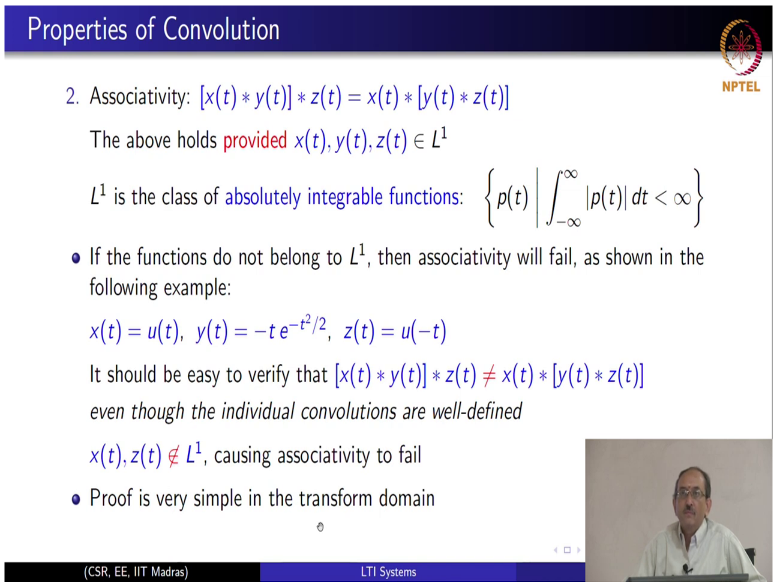
Ok, to proove that will take us outside the scope of this course alright, hand book of Fourier theorems, Campini, take a look at that book; I think its Cambridge University press. So, all the theorems that you study in both CTFT and DTFT, more rigor you will find in math books. So, the above holds provided that each of these three sequences belong; belongs to the space of l_1 and l_1 is the class of absolutely summable sequences. So, if the sequences do not belong to l_1, then associativity will fail. So, this is exactly the sequence that I worked out the details of just now.

So, as we saw each of these individual convolutions exists there are no problems, but associativity does not hold. And proof of associativity is very simple in the transform domain. And it is simple in the transform domain because the transform you assume it exists. And a^n which is the everlasting exponential clearly does not have a transform; the only case where it does poses a transform is when a equals? When?

Student: (Refer Time: 16:50).

Oh Yeah, very good when a is $e^{j\omega}$ then transform exists, the transform is impulsive. But anyway, these sequences require special treatment. So, whatever Fourier transforms that you have learnt so far; it is without rigor. So, if you study Fourier transforms from the maths people; they will talk about the space of l_1; L_1 and L_1 functions which are the class of absolutely integrable functions and the class of square integrable functions; absolutely square integrable functions. So, they divide the Fourier transform into these two sub sections and study them more rigorously right. So, as far as we are concerned, transform always exists. We do not care, we will leave the question of existence to those people.

(Refer Slide Time: 17:56)

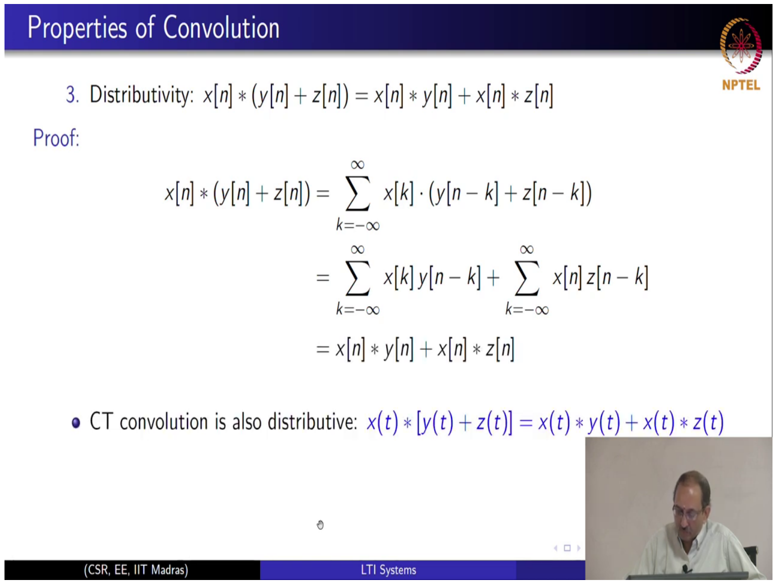


So, this the counter part two; the continuous-time signals again we assume associativity provided these belong to the space of L^1. So, this is the space of absolutely integrable functions and here is an example where associativity will fail in the continuous-time case.

Student: Depends on (Refer Time: 19:11) know the (Refer Time: 19:14).

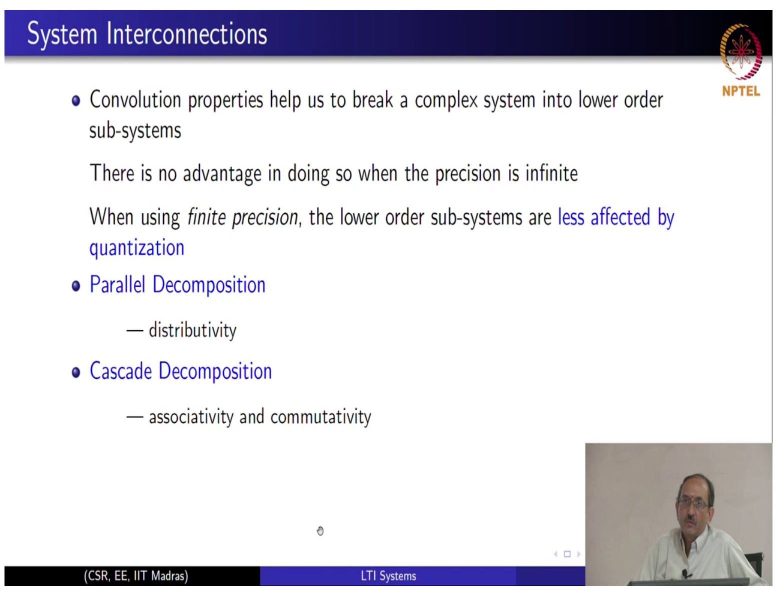
So, assume for real numbers. So, it follows from it is an axiom, the axioms that you assume for real numbers alright. So, the famous book Calculus by Spivak, if you are interested take a look at Spivak Calculus amazing book; so the first chapter talks about these axioms.

(Refer Slide Time: 19:51)



And then distributivity; $x[n] * (y[n] + z[n])$ is $x[n] * y[n] + x[n] * z[n]$. And this of course, follows from the distributivity of multiplication over addition for real numbers, this follows for real and complex numbers based on that this convolution property is established very easily, the CT convolution is also distributive.

(Refer Slide Time: 20:19)



Now, this associativity and this distributivity property of convolution is just not an ideal mathematical curiosity or property that follows; it really plays a very important role in practical applications. It plays a role in practical applications in that it helps us to break down complex systems into lower order subsystems. By complex systems, we mean systems with high order.

And you can break down such high order systems into combinations of lower order systems. And in practice typically, we will implement lower order systems in terms of first and second order. There is no advantage of doing this when the precision is infinite. But in all practical cases, the precision is finite. Whenever you are going to implement a system, you will be requiring say, multiplier elements and those multiplier elements cannot be of infinite precision.

In discrete-time systems, what you will do is these multiplier coefficients will have finite number of bits. When you have finite number of bits, you are introducing quantization error. When there is quantization error, then if you have a very high order system, if you want the quantized systems behavior to be not too different from the ideal infinite precision system, then you have to expend a large number of bits. Whereas, if you break it down into smaller subsystems the precision needed can be reduced; the number of bits needed can be reduced.

So, this is where this theoretical properties indeed very much used in practice because lower order systems it can be shown that they are less affected by quantization. And we carry out two kinds of decompositions; one is parallel decomposition. In parallel decomposition, the property that is used is distributivity; we will see that in a minute. And the other decomposition is cascade decomposition and here two properties are used both associativity and commutativity; these two properties are used, alright.

So, if you have a certain real number; a typically when you are implementing it in on a machine, we use binary representation. And even you are using binary representation; then you need to expend certain number of bits. And if the number that you want to represent requires an infinite number of binary digits; you do not have infinite number of digits to expend, so you have to represent that number by a finite number of bits.

So, this means your truncating the representation; so this is quantization. That is your actual real number having infinite precision when it represent, when it is representing as a sequence of binary digits, if the representation for exact representation is infinite which is not possible, if you truncate there, there you introduce quantization. And even if a number can be represented in a finite number of bits, if that representation requires a large number of bits; suppose if it requires 100 bits to represent it exactly; you do not have hundred bits in practice, right. So, typically if you want to guess what would be the number of bits that are used in practice?

Student: (Refer Time: 24:25).

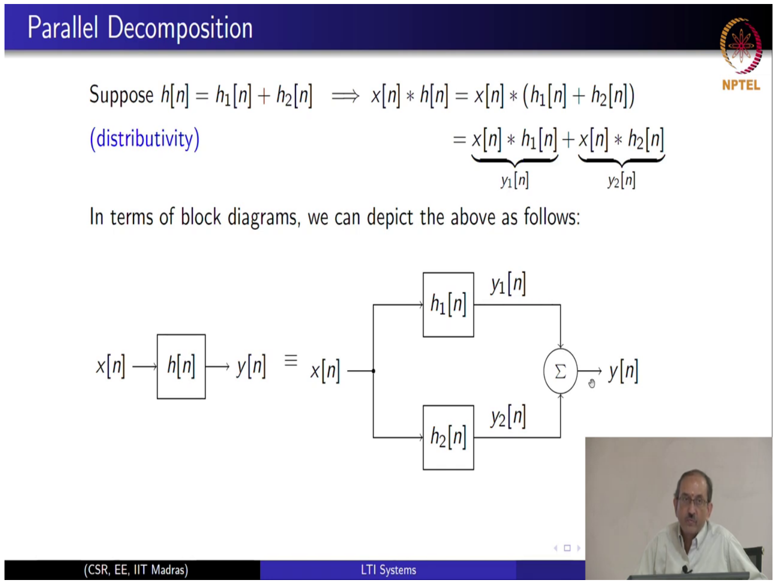
Say that again.

Student: 16.

Yeah, 16 is commonly used number of bits for representing numbers and now because hardware is becoming cheaper you can afford to spend more bits. For example, now for sound, typically what is the number of bits that I used to represent sound samples? What you do is you that is an analog waveform you sample and then you represent it as binary numbers. And the number of bits used in for sound these days, any guesses? 24 bits; these days 24 bits is the standard for representing sound.

So, that I hope answers the question of what precision and quantization they mean, ok. So, in practice higher order systems are typically broken down into first and second order systems. And this is where we come up with parallel and cascade decomposition and these properties of convolution are indeed used here.

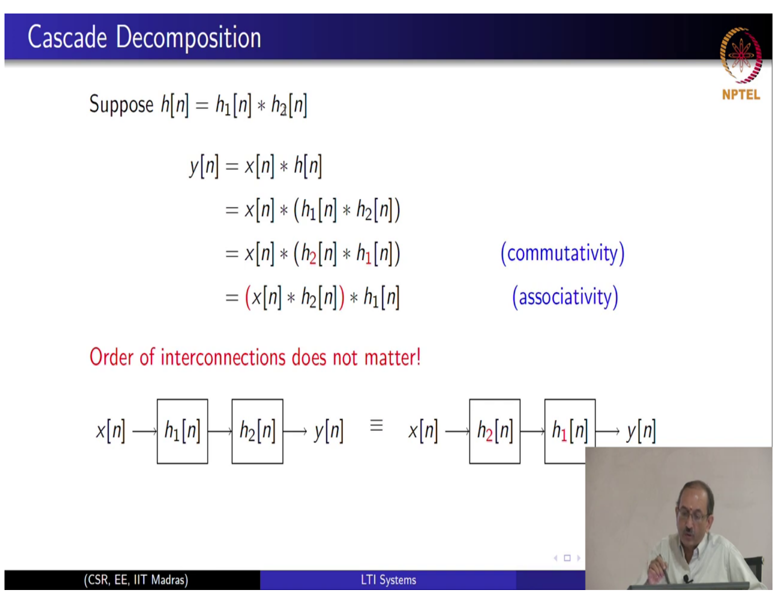
(Refer Slide Time: 25:41)



So, let us look at parallel decomposition; first suppose you have $h[n]$ if it is represented as $h1[n] + h2[n]$; so this is the higher order system. So, this is broken down into $h1[n] + h2[n]$ so that $x[n]$ convolved with $h[n]$ can be represented as $x[n]$ convolved with $h[n]$ in turn is $h1[n] + h2[n]$. And then we use distributivity and get y_1 and y_2. So, in terms of block diagram you have this original system as given here; it is broken into these two subsystems and you get $y[n]$.

So, again to reiterate you do not get any advantage out of this when the precision is infinite, but under finite precision these things help. So, when you want to implement digital filters; we do what is called parallel decomposition. So, we have a transfer function, we break it down into a partial fraction expansion and each of these terms is a smaller subsystem and each of these subsystems is implemented like this and then we add them up to get the final output. And each of these subsystems can afford to use a lower number of bits in terms of coefficients.

(Refer Slide Time: 27:10)

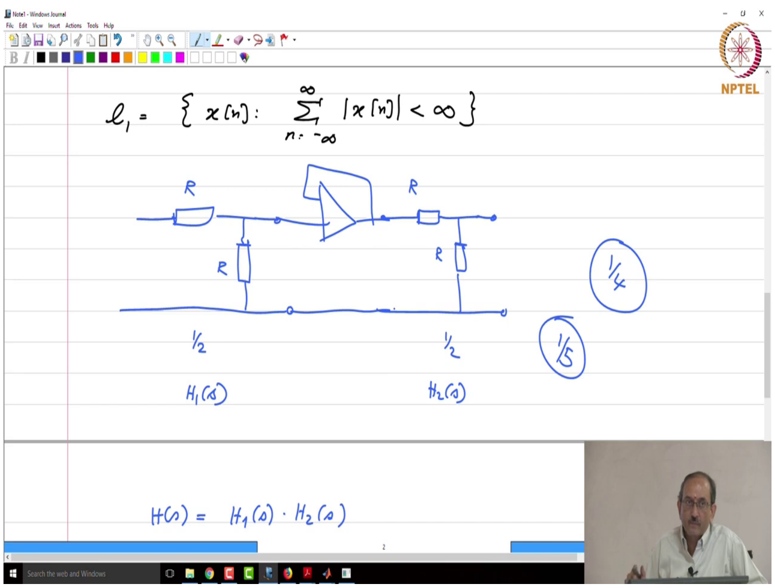


And the next is cascade decomposition. Suppose $h[n]$ is $h_1[n] * h_2[n]$, then you can replace $h[n]$ by $h_1[n] * h_2[n]$ and then you can interchange. So, where there you use commutativity and then again you are using associativity here.

So, what this tells you is order of interconnections does not matter. Again, if you have a high order system $h[n]$ and if you have broken it down into $h_1[n]$ and $h_2[n]$, there are under certain conditions where you want to interchange the order. Again, when you do a course on digital filters, you will realize for certain noise spectrum criterion, it would make sense to order the systems in a certain way and exactly opposite ordering would be what would be needed for some other criterion. So, all of these things are used in practice.

So, this again parallels what you have learned in continuous-time. So, if you have $H(s)$; you can break it down into $H_1(s)$ times $H_2(s)$ and then again exactly the same kind of ideas applied there. This one difference between cascading of some systems in continuous-time versus discrete-time; again this must be known to you or must have been pointed out to you.

(Refer Slide Time: 28:55)



So, in continuous time, if we had a system like this R and R; so the overall gain is 1/2 and if you cascade two such systems right. If you just cascade it right away the overall gain will be?

Student: (Refer Time: 29:26).

No, here is a system with gain 1/2; here is another system that is gain of 1/2. So, if you cascade these two systems; what do you expect the overall gain to be? Ok, this one easy answer. It will be? 1 by; so 1/4 is what you expect; what is the assumption under which the overall gain is 1/4?

So, this is let us call this as $H_1(s)$; this is $H_2(s)$ and the overall system $H(s)$ is $H_1(s)H_2(s)$. If you expect the gain to be 1/4; what is the assumption here? So, here. Right?

Student: Same as the above.

So, now if you cascade these two, you expect the gain to be 1/4, but you can just connect these two and you can find out what is V_0/V_i; we will find that the answer to be 1/5. And the reason is? So, it will not be 1/4; it is it actually be 1/5; so easy to see. So, if this not been pointed out in networking systems.

Student: Sir, when we connect the system we let the (Refer Time: 31:31) to the power (Refer Time: 31:32) currently not generated to open circuit.

Huh. Student: So, the; so let the current should not flow from this vertical R.

Ok, very good yeah. So, what you are assuming here is there is no, what is the term that is used for?

Student: (Refer Time: 31:50).

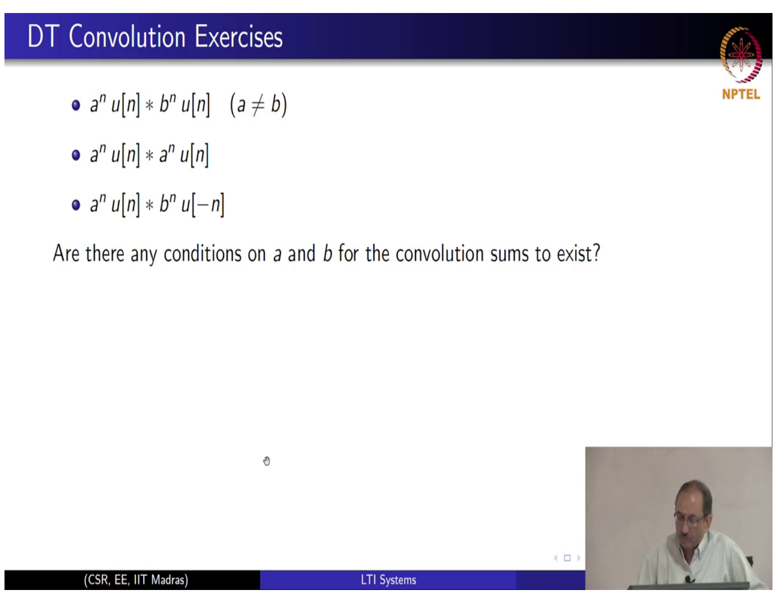
No. So, when you connect this; there should be no loading. So, when you connect this the individual currents that are flowing here are not the same when you connect them. When you connect this second circuit to the first one, the second circuit loads the first circuit. So, the circuit, this individual systems are no longer the same.

So, the condition under which this is true is; you need to connect these two by a voltage follower. So, you need to connect these to by a voltage follower; so that there is no loading; so that each of the systems, they are not changed when you connect them.

So, when you say you have I have $H_1(s)$ and $H_2(s)$; when you say I connect them and then the overall system trans function is indeed $H_1(s)H_2(s)$; the implicit assumption is loading. And when it comes to continuous-time systems, you need to worry about whether the loading is there or not. If loading is there, this multiplication of transform function is not valid.

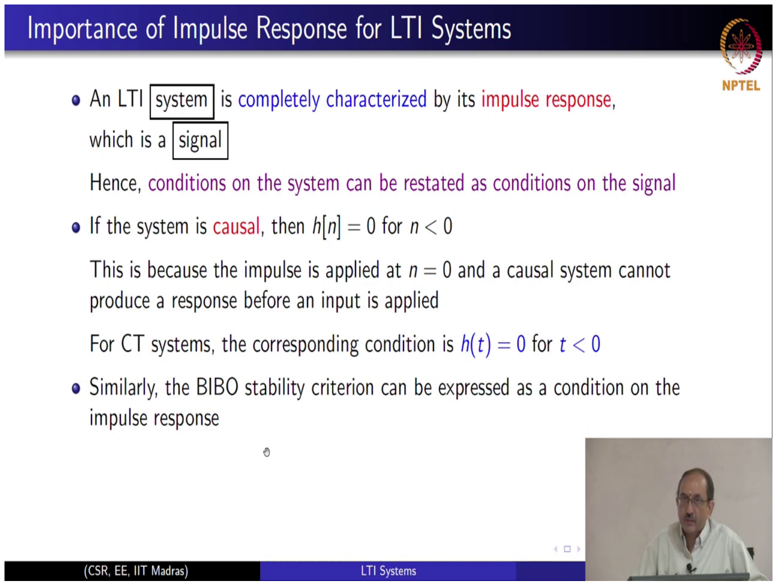
So, when you assume the transform functions are indeed multiplicative; the implicit assumption is no loading. When it comes to discrete-time, no such issues are present because here we are talking about the implementation of the computer. So, we are going to, these are all algorithms that are implemented on a computer; so, there is no question of loading in the discrete-time case. So, we can assume that if you have if $h_1[n]$ followed by $h_2[n]$, the transform domain, the corresponding transformer will be the z-transform; it will be $H_1(z)H_2(z)$ and the concept of loading does not apply to discrete-time systems.

(Refer Slide Time: 33:48)



And these are some of the counter parts of convolution exercises that you should be familiar with. Again you should be able to do the algebraic manipulation so that you get the correct answer. Again, because in the continuous-time case, the integration limits are $-\infty$ to ∞ and the discrete-time case; you have summation from minus infinity to plus infinity, you need to worry about these things existing, you need to worry about convergence of these things.

(Refer Slide Time: 34:34)



Now, let us focus on the LTI system and impulse response. So, the LTI system is completely characterized by impulse response. If you know the impulse response when LTI system, you know everything about the LTI system. And the impulse response is actually a signal whereas, the system is a system and everything is by a signal and that signal is special, it is the impulse response.

And conditions on the system can be restated as conditions on the signal, we will see examples of this. By the way, when you evaluate the impulse response of a system, what is that you are assuming about the system, when you are measuring the impulse response? This applies both continuous-time as well as discrete-time. So, based on your knowledge of continuous-time system and the impulse response concept that you have learnt, what is it that you assume about this system when you are.

Student: (Refer Time: 36:00).

Yeah, input is the impulse; so that is fine anything else? What is there anything that the system should be satisfying for you to?

Student: (Refer Time: 36:14).

Ok, yeah LTI system because only for such systems will the impulse response completely characterize the system. What is that you are assuming about the system in your when you say impulse response? Say you have continuous time case, you have an RLC circuit; what is it about the system you are assuming when you talk about the impulse response of a given RLC circuit?

Student: (Refer Time: 36:49).

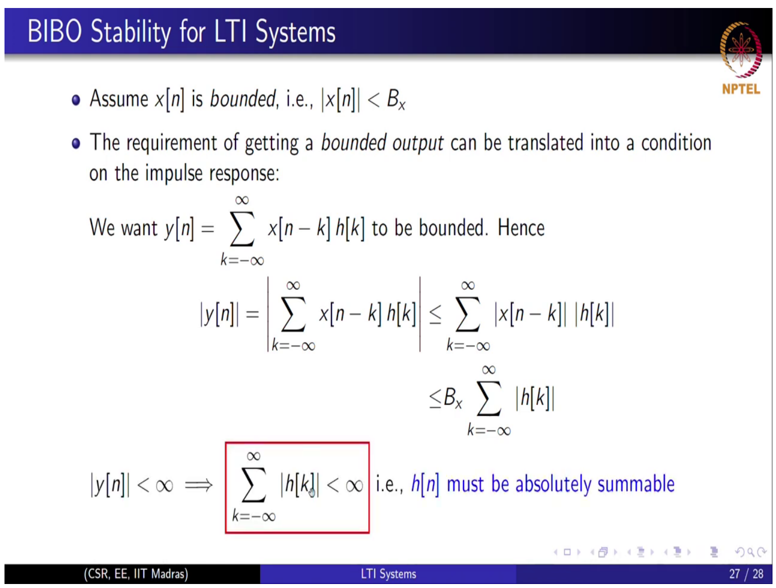
Very good, no initial conditions right. Similarly for a discrete-times system also, we are assuming all initial conditions are 0; why is it that this is needed?

Student: (Refer Time: 37:12).

Yes, this is not satisfied, the impulse response will not be unique. So, now, coming back to some of the conditions on the system being translated as conditions on the impulse response; what are the most important things is causality. So, if the system is causal, then $h[n]$ must be 0 for $n < 0$ and this is because the impulse is supplied at $n = 0$; the system is causal which means it cannot anticipate the impulse being applied and hence it cannot produce a response before the impulse is applied.

Therefore, the response necessarily has to begin at $n = 0$ and onwards. Therefore, for a causal system, the impulse response is always 0 for $n < 0$. So, condition on the system is now translated to a condition on the impulse response. So, for continuous-time systems, $h(t) = 0,\ t < 0$. Similarly, for stability the condition can be expressed as a condition on the impulse response.

(Refer Slide Time: 38:35)

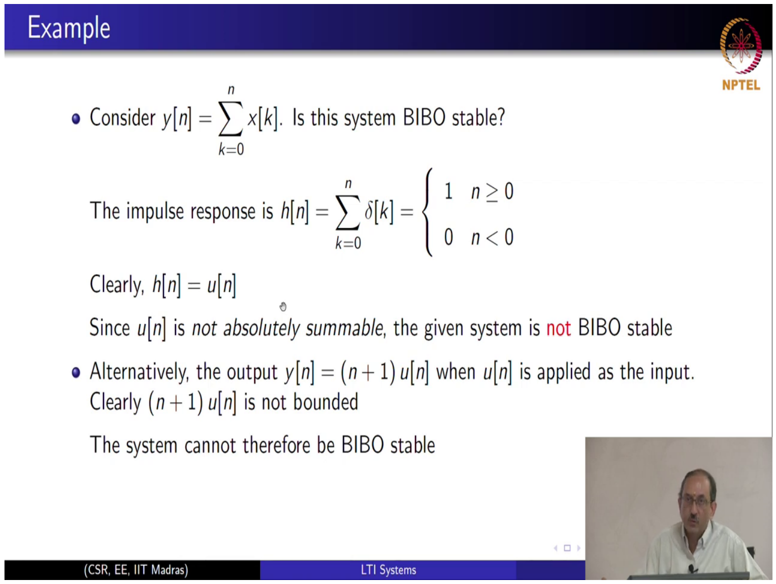


And, we assume that the input is bounded and this condition can be translated to a condition of the impulse response; this is no different from what you might have seen in continuous-time case.

So, we want $y[n]$ to be bounded and if you try to bound the output, then you see that the absolute value of the sum is always less than or equal to sum of the absolute values. And, so this is always less than or equal to B_x; you can always bound this upper bounded by the bound of the input and you get this. And if you want this to be finite, this leads to the condition that this has to be absolutely summable; only if this is true, is the system BIBO stable.

Therefore, condition on the system namely, bounded input must produce bounded output is translated as a condition on the impulse response namely it being absolutely summable. Counter part of this in the continuous-time case is impulse response must be absolutely integrable.

(Refer Slide Time: 39:40)



And here is an example that we have already seen. So, we saw that the running sum is not a BIBO stable because we applied a step, the output is a ramp which is not bounded. The impulse response of a such a system is nothing but $u[n]$ because, if you compute the running sum of the impulse its nothing, but the unit step and clearly the unit step is not absolutely summable.

So, this example that we saw, in terms of a bounded input $u[n]$ producing a ramp as the output and therefore, that is not bounded. It is also seen in terms of the impulse response is not being absolutely summable. In this case the impulse response is $u[n]$ therefore, this system is not BIBO stable.