[MUSIC PLAYING]

ALAN OPPENHEIM: Well, last time we had discussed, in a very general way, the field of digital signal processing, and hopefully one of the things that you were convinced of was the importance of this set of techniques in a general sense. What we'd like to begin on now is a discussion of some of the details of digital signal processing. And in particular, in this lecture, I would like to introduce the class of discrete time signals, and also the class of discrete time systems.

Well, last time, I reminded you that I will be assuming a familiarity with continuous time signals and systems. And what you'll see as we go through this lecture-- and in fact, as we go through all of these lectures-- is a very strong similarity between the results that you're familiar with in the continuous time case, and results that will develop for the discrete time case.

Obviously, of course, there will also be some differences. And these differences are important, and they will be important to focus on. So there will be a great deal of similarity, also a few differences. And these few differences are very important.

In the discrete time domain, we're concerned with processing signals that are sequences. That is, these signals are functions of an integer variable, which we'll call n. Typically, will depict sequences graphically, as I've illustrated here on this first viewgraph, where I've denoted a general sequence-- which I call x of n-- a function of the energy variable n, and the sequence values, I've represented by a bar graph with the height of each bar corresponding to the sequence value.

So we have, then, for example, at n equals 0, a line of height x of 0 at n equals 1 a line of height x of 1, et cetera. Of course, as I've drawn this horizontal axis, I've drawn it as a continuous line, but it's important to keep in mind that the sequences only are defined-- or only make sense-- for integer values of the argument.

If n is an integer, then x of n makes sense. If n is not an integer, it's not just that x of n is 0 or something like that, it's simply that x of n is not defined. In other words, x of n makes sense, or is defined for n an integer, and does not make sense, or is not defined, if n is not an integer.

So this, then, is a graphical representation of a general sequence. And just as in the continuous time case, we have a number of important basic sequences that we would like to focus on.

The first of these that I'd like to introduce is the unit sample, or impulse sequence, which we'll denote by delta of n. The unit sample sequence is a sequence whose value is unity at n equals 0, and it's equal to 0 otherwise. That is delta of n equals 1 at n equals 0, and delta of n is equal to 0 for n not equal to 0.

The unit sample sequence plays the same role in the discrete time case that the unit impulse time function plays in the continuous time case. Let me just mention, however, that the unit sample sequence is very easily defined. There's no issue here about the definition. In contrast with the usual mathematical problems with a continuous time impulse-- which is infinitely big at the origin, and 0 every place else, but has some area, et cetera. So there's a very precise definition here of the unit sample, or unit impulse sequence, and it will play a role similar to the unit impulse time function in the continuous time case.

Another important basic sequence is the unit step sequence, which I'm denoting by u of n. And the unit step sequence is equal to unity for n greater than or equal to 0, and is equal to 0 for n less than 0. So the unit step sequence, then, is 0 for n less than 0, it's unity four n greater than or equal to 0. This shouldn't be, incidentally, less than or equal to 0, this should be n less than 0.

And it is-- again, does not have the difficulties in definition that a continuous time unit step normally has. It's the change in value from n equals minus 1 to n equals 0 is precisely and easily defined. Here the unit step is 0, here the unit step is 1. There's no issue, as there is in the continuous time case, about discontinuity.

Just as in the continuous time case, there is a simple relationship between the unit sample, or a unit impulse, and the unit step. In a discrete time case, there is likewise a simple relationship between the unit sample sequence, and the unit step sequence. In particular, the unit sample sequence can be obtained from the unit step sequence by constructing the first difference. The unit sample sequence, delta of n, is equal to u of n, the unit step, minus the unit step delayed by 1 sample.

And we can see that that obviously is the case, if we look at the unit step sequence, u of n. Here we have the unit step sequence delayed by 1-- so this occurs at n equals 0, this is n equals 1. And clearly, the difference between those two will generate a sample at n equals 0, 0, of course, for n less than 0, and 0, likewise, for n greater than 0, because these sequence values will be canceled out by those sequence values.

So the unit sample sequence is the unit step minus the delayed unit step, or the unit sample sequence is equal to the first difference of the unit step sequence.

Similarly, we can obtain the unit step sequence from the unit sample sequence by constructing a running sum. In particular, the unit step sequence is equal to a running sum of the unit sample sequence. In other words, we're constructing here a sum from k equals minus infinity to the independent variable n of the sequence, delta of k.

Here we have the sequence, delta of k. If n is less than 0, then we're adding up sequence values from minus infinity to some negative value of k. And the only values that we collect when we do that are values equal to 0. So for n less than 0, we see that we obtain 0 for this sum. If n is greater than 0, as I've indicated here, then the sum, as we collect sequence values from minus infinity up to n, we collect 1 non-0 sequence value which is equal to 1. Consequently for n greater than 0, we obtain, in this running sum, u of n equal to 1.

This plays a role, this running sum plays a role, similar to the integral in the continuous time case. And so now we have a relationship between the unit sample sequence, and the unit step sequence.

There are other basic sequences that will play an important role, just as they do in the continuous time case. In particular, we have the real exponential sequence. The sequence x of n is equal to alpha to the n. I've depicted the exponential sequence here for the case in which alpha is positive but less than unity, so that as n increases, the exponential decreases.

So alpha to the n, if alpha is between 0 and 1, decreases with increasing n. If alpha is greater than 1, then the exponential sequence grows, of course, exponentially.

We have, also, the sinusoidal sequence. The general form of the sinusoidal sequence is x of n is equal to a cosine omega 0 n plus phi-- phi a phase angle, omega 0 we'll refer to as the frequency, a, of course, is the amplitude. I've illustrated a sinusoidal sequence here for a particular choice of omega 0 and phi, namely omega 0 equal to pi over 4, and phi equal minus pi over 8.

And in looking at this, we see that we get a periodic sequence that is, for this choice of parameters, this sequence is periodic. However, an important distinction between the continuous time case and the discrete time case, is that a discrete time sinusoidal signal is not necessarily periodic.

Particular, I've illustrated here another sinusoidal sequence with a different choice for omega 0 and phi. In this case, omega 0 equal to 3 pi over 7. With that choice of omega 0, these are the sequence values we obtain. And it should be obvious by examining this sequence that this sequence is no longer periodic, whereas this sequence was.

So the sinusoidal sequence may or may not be periodic. Of course, let me remind you that if n was allowed to vary continuously-- if it was not an integer variable-- x of n would be periodic. The periodicity is lost because n is now constrained to be an integer variable. And that is one of the important distinctions between continuous time signals and systems, and discrete time signals and systems.

Another point which I'll suggest now, although it's a point that I'll want to refer to in more detail in some of the future lectures, is that the sinusoidal sequence is only distinguishable as omega 0 runs from over the range 0 to 2 pi or minus pi to pi. If omega 0-- if we were to replace omega 0 by omega 0 plus 2 pi, then we would have in this argument omega 0 n plus 2 pi n. The 2 pi n, of course, would have no effect, and we'd end up with the same sequence.

So in fact as, we varied omega 0 between minus pi and plus pi, you will have seen all of the sinusoidal sequences with this amplitude and this phase that we can possibly generate.

One of the important of this set of basic signals is that they can be used to represent a more general class of signals, just as in the continuous time case, we use impulses, or we use complex exponentials, or we use sinusoids to represent very general signals, we can develop similar representations here.

Let me illustrate this with an example where I consider a general sequence here, x of n. The sequence values are x of 0, x of 1, x of 2, et cetera. And I mean to suggest here a very general sequence.

Now, I can decompose this sequence into a linear combination of weighted, delayed unit samples, unit sample sequences, by simply extracting individual sequence values out of x of n. For example, let's consider the sequence x of 0 times delta of n, as I've illustrated here. X of 0 times delta of n is equal to the original sequence x of n at n equals 0, and it's equal to 0 otherwise.

A second sequence, x of 1 times delta of n minus 1, is a unit sample sequence, delayed by 1 and having an amplitude equal to x of 1. So this sequence is equal to this one at n equals 1, and it's equal to 0, otherwise. The sum of these two, of course, is equal to this x of n at these two values, and equal to 0, otherwise.

Similarly, we can consider x of minus 1 times delta of n plus 1. That's a unit sample at n equals minus 1, with an amplitude of x of minus 1, which picks up this sequence value. x of minus 2 times delta of n plus 2 is a unit sample at n equals minus 2 multiplied by an amplitude x of minus 2, which picks up this sequence value, et cetera.

So I think you can see, then, that as we add these up, what we'll generate is this arbitrary sequence, x of n. In other words, we can construct x of n, an arbitrary sequence, as a linear combination of weighted delayed unit samples-- x of 0 times delta of n plus x of 1 times delta of n minus 1 plus x of minus 1 delta n plus 1, et cetera. Or, more generally, the sum from k equals minus infinity to plus infinity of x of k times delta of n minus k.

This, then, corresponds to a general representation of an arbitrary sequence in terms of weighted delayed unit samples. And it's a representation that we'll want to refer back to in a few minutes. It's not, as I've indicated, the only representation of arbitrary sequences-- we'll also be developing representations in terms of complex exponentials, or real exponentials, or in terms of sines and cosines.

OK, that is an introduction to the basic class of signals and the notion of sequences. What I'd like to do now is focus on the class of discrete time systems, and then refer back to some of these results to develop a general representation for a special class of discrete time systems.

First of all, let me begin with a general system, which is a discrete time system that has an input-- which is a sequence x of n. It has an output, which is a sequence y of n. And it has a system transformation, which I've denoted by T. Of course, there isn't much that you can do with a general system-- the difficulty with trying to describe a general system in general is that there are no properties that you can take advantage of.

So always, in characterizing systems, it's useful to specialize the class of systems-- that is, impose properties on the system which you can exploit to represent the system, or to implement the system, et cetera. The special class that we'll want to consider is the class of systems which are, first of all, linear. And second of all, shift invariant.

And this class corresponds, in the continuous time case, to the class of systems that we normally refer to as linear and time invariant. We'll tend to refer to these as linear and shift invariant, and for a shorthand notation, just express this as well LSI. When I refer to an LSI system, what I mean is a system that is linear and shift invariant.

Well, let's define these terms. First of all, linearity. Linearity states that if I excite the system with a sequence x 1 of n, and I get at the output a sequence y 1 of n, and if I excite with x 2 of n, and get at the output y 2 of n, then the condition of linearity is that the linear combination, a x 1 of n plus b x 2 of n, produces at the output a y1 of n plus b y 2 of n. That is, the response of the system to a sum of inputs, or a linear combination of inputs, is a linear combination of the corresponding outputs.

Now, by repeated application of just this statement, we can make a more general statement, which is that the sum of a sub k times x sub k of n produces, as an output, the sum of a sub k times y sub k of n. Linearity is stated here, of course, for two inputs, but we can obviously extend that to an arbitrary number of inputs so that the statement of linearity, as we'll refer to it, will generally be that a general linear combination of inputs produces, at the output, the same linear combination of the corresponding outputs.

So that's the condition of linearity. The second condition that we want to impose on our class of systems is the condition of shift invariance. What shift invariance says, simply, is that if x of n produces y of n, then x of n minus n 0 produces y of n minus n 0. It's basically a statement that the system doesn't particularly care what you call n equals 0. In other words, if we shift the input in n, we shift the output in n. It's exactly like the condition of time invariance in the continuous time case.

For example, if we excited the system with a unit sample, to get the unit sample response, h of n, then the response of the system, if the system is shift invariant, to a delayed unit sample is the delayed version of the unit sample response.

So this is shift invariance. We had the condition of linearity. These are two independent conditions which we'll now want to put together to develop a general representation for linear shift invariant systems.

Well, let me remind you from the last viewgraph that we had looked at, that we had a general representation for a sequence in terms of weighted delayed units samples. That is, previously we had developed a representation, x of n, of this form, which expresses an arbitrary sequence in terms of weighted delayed unit samples.

Well, this, then, says that x of n is expressed as a linear combination of basic inputs, and if we restrict ourselves to linear systems, then the output must correspond to the same linear combination of the corresponding outputs. Well, delta of n minus k into a linear shift invariant system will produce, at the output, by virtue of the property of shift invariance, the sequence h of n minus k.

And the linear combination of these delayed unit samples will produce at the output, by virtue of linearity, the same linear combination of the responses to delta of n minus k. So if we consider a linear shift invariant system, and because of this representation, we can say that the response due to an arbitrary input is equal to a linear combination of delayed unit sample responses. Where, in general, this is a sum from k equals minus infinity to plus infinity.

Well, this is the key result-- this is a statement that says that if we talk about linear shift invariant systems, then all that we need to know about the system to characterize it is its response to a unit sample. If we have its response to a unit sample, we can construct y of n in general-- that is, we can construct the output for an arbitrary input, x of k.

This is generally referred to as the convolution sum. In analogy with the convolution integral in the continuous time case, this is one way of writing the convolution sum. There is an alternative, which is interesting in that it suggests some important properties of linear shift invariant systems, and it's obtained simply by applying a substitution of variables to this expression.

In particular, suppose that we replace n minus k by the variable r, or equivalently, k is equal to n minus r. Then what we obtain is y of n is now equal to a sum-- and this is now a sum on r. For k, we have n minus r. So this is x of n minus r times h of n minus k, which is equal to r. So this h of r.

Well, it's a simple step to take, but in fact, what it says is that the system doesn't particularly care what you call the input to the system, and what you call the unit sample response of the system. Said another way, it says that convolution is commutative-- that is, if we represent the convolution of x of n with h of n with an asterisk, then, because of the fact that we were able to interchange the roles of x of n and h of n, that implies, essentially, that x of n convolved with h o n is the same thing as h of n convolved with x of n.

That implies, as I indicated, that if we had a system with an impulse response or unit sample response, h of n, and input x of n, and output y of n, that if I call this the input, and I call this the unit sample response-- as I've done here-- we obtain the same output. That is, h of n into a system with unit sample response, x of n, gives us at the output, y of n, also.

An implication of that is that linear shift invariant systems in cascade don't particularly care which order the systems are cascaded in. That is, if I have a system h 1 of n-- that is with unit sample response, h 1 of n-- and cascade with a system with unit sample response h 2 of n, and input x of n, the unit sample response of this overall system is h 1 of n convolved with h 2 of n. But since convolution is commutative, we could just as easily convolve h 2 with h 1, and that corresponds to a cascade of the system h 2 of n with the system h 1 of n.

So simply because of the fact that convolution is commutative, that implies that, for example, that linear shift invariant systems in cascade don't particularly care in which order their cascaded. There are lots of other properties of convolution and simple properties of linear shift invariant systems that are a direct consequence of convolution, and some of the properties that we've mentioned of convolution. And a number of these, we'll see in the lectures as we go along, and also a number of them you'll have a chance to wrestle with in the study guide.

OK, so this is a development of the convolution sum. One of the important aspects of the convolution sum is the steps involved in actually computing it-- that is, the manipulations involved in forming this sum. And as the last point that I'd like to make in this lecture, I'd like to illustrate, first with a viewgraph, and then with a movie, the computation, or the implementation, of the convolution sum.

So let's return to the viewgraph.

And I've indicated here, again, the convolution sum as-- we've derived it-- y of n is the sum of x of k h of n minus k. Now, in implementing the convolution sum, let me stress that it is for n, which is the independent variable for which we are computing the output-- that is, k is just simply a dummy variable inside this summation.

Well, I've illustrated here, for a specific example, a sequence, x of k, which is a constant from 0 to I guess n equals 10, equal to unity for n equals 0 to 10, and 0 otherwise. And I've indicated the sequence a little differently here than I had previously, since I want to distinguish between the x's, h's, and the y's. The sequence x of k, I'm denoting with little x's-- the sequence h, I'll denote note with little h's, and later on, the sequence y, we'll denote with little y's.

All right, so here we have the sequence a, that's this one. And plotted as a function of k. Here, I have the sequence, h of k, which I've chosen to be an exponential for k greater than 0, and 0 for k less than 0. But it's not h of k that we want in this sum. It's h of n minus k.

And what's n? Well, n is whatever value of the output sequence we're trying to compute. So if n was equal to 0, for example, we would want the sequence, h of 0 minus k. And that's this sequence.

It's h of k, flipped around in k, because of this minus sign, and not shifted one way or the other simply because we have a 0 here. So here's h of k, but that's not what you want. It's h of 0 minus k. And now to compute y of 0, we would multiply this sequence by this one, and compute the sum from minus infinity to plus infinity. That would give us the value, y of 0.

For a different value of n, we would have to look at h of n minus k, for whichever value of n we were computing this for. For example, if n is equal to minus 4, the sequence that we want is h of minus 4 minus k, and that is this sequence shifted to the left by 4. So to compute y of minus 4, we want this sequence multiplied by this one, and the sum computed from minus infinity to plus infinity.

And you can see, obviously, for that particular case-- that is n equal to minus 4-- the product of this and this is 0, and the sum will be equal to 0.

Well, let's look at this example a little more dynamically, with a movie that was prepared at Bell Telephone Laboratories by Dr. Ronald Shaefer.

OK, the convolution sum, then, we have is the sum of x of k, h of n minus k. And so what we would like to illustrate is the operation of evaluating this sum. On the top, we have x of k. On the bottom we have h of k-- he of k being an exponential. And now, we see h of minus k, namely h of k flipped over. As we shift h of k to the left, corresponding to n equals minus 1, then back to 0, and now to the right.

So we have h of 1 minus k. And then to 4 y of n, we want the product of h of minus k shifted with x of k, that product sum from minus infinity to plus infinity. So here, we see x of k times h of 1 minus k, 2 minus k, 3 minus k, et cetera. Those multiplied and then summed from minus infinity to plus infinity, we see during this portion that more and more values of h are engaged with non-0 values of x, and so y of n grows until we reach the point where values fall off the end of the non-0 values of x, where y of n decays.

So this, then, is an illustration of the linear convolution of x of k with h of k.

OK, this completes our introduction to discrete time signals and systems. There were a number of important points that we've made during this lecture. But the key result, and the result that it's important to develop some experience with, is the convolution sum.

Next time, we'll introduce some additional considerations, namely the considerations of stability and causality for discrete time systems. And we'll also discuss, briefly, the class of linear shift invariant systems that are represented by linear constant coefficient difference equations.

Finally, we'll try to tie some of this together and in particular, present what will be called the frequency response of the systems. And this will eventually lead into a discussion of Fourier transforms. Thank you.

[MUSIC PLAYING]