

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.

PROFESSOR: In discussing the continuous-time and discrete-time Fourier transforms, we developed a number of important properties. Two particularly significant ones, as I mentioned at the time, are the modulation property and the convolution property.

Starting with the next lecture, the one after this one, we'll be developing and exploiting some of the consequences of the modulation property. In today's lecture though, I'd like to review and expand on the notion of filtering, which, as I had mentioned, flows more or less directly from the convolution property.

To begin, let me just quickly review what the convolution property is. Both for continuous-time and for discrete-time, the convolution property tells us that the Fourier transform of the convolution of two time functions is the product of the Fourier transforms.

Now, what this means in terms of linear time-invariant filters, since we know that in the time domain the output of a linear time-invariant filter is the convolution of the input and the impulse response, it says essentially then in the frequency domain that the Fourier transform of the output is the product the Fourier transform of the impulse response, namely the frequency response, and the Fourier transform of the input. So the output is described through that product.

Now, recall also that in developing the Fourier transform, I interpreted the Fourier transform as the complex amplitude of a decomposition of the signal in terms of a set of complex exponentials. And the frequency response or the convolution property, in effect, tells us how to modify the amplitudes of each of those complex exponentials as they go through the system.

Now, this led to the notion of filtering, where the basic concept was that since we

can modify the amplitudes of each of the complex exponential components separately, we can, for example, retain some of them and totally eliminate others. And this is the basic notion of filtering.

So we have, as you recall, first of all the notion in continuous-time of an ideal filter, for example, I illustrate here an ideal lowpass filter where we pass exactly frequency components in one band and reject totally frequency components in another band. The band being passed, of course, referred to as the passband, and the band rejected as the stopband.

I illustrated here a lowpass filter. We can, of course, reject the low frequencies and retain the high frequencies. And that then corresponds to an ideal highpass filter. Or we can just retain frequencies within a band. And so I show below what is referred to commonly as a bandpass filter.

Now, this is what the ideal filters looked like for continuous-time. For discrete-time, we have exactly the same situation. Namely, we have an ideal discrete-time lowpass filter, which passes exactly frequencies which are the low frequencies. Low frequencies, of course, being around 0, and because of the periodicity, also around 2π .

We show also an ideal highpass filter. And a highpass filter, as I indicated last time, passes frequencies around π . And finally, below that, I show an ideal bandpass filter passing frequencies someplace in the range between 0 and π . And recall also that the basic difference between continuous-time and discrete-time for these filters is that the discrete-time versions are, of course, periodic in frequency.

Now, let's look at these ideal filters, and in particular the ideal lowpass filter in the time domain. We have the frequency response of the ideal lowpass filter. And shown below it is the impulse response. So here is the frequency response and below it the impulse response of the ideal lowpass filter. And this, of course, is a $\sin x / x$ form of impulse response.

And recognize also or recall that since this frequency response is real-valued, the

impulse response, in other words, the inverse transform is an even function of time. And notice also, since I want to refer back to this, that the impulse response of an ideal lowpass filter, in fact, is non-causal. That follows, from among other things, from the fact that it's an even function.

But keep in mind, in fact, that a sine x over x function goes off to infinity in both directions. So the impulse response of the ideal lowpass filter is symmetric and continues to have tails off to plus and minus infinity.

Now, the situation is basically the same in the discrete-time case. Let's look at the frequency response and associated impulse response for an ideal discrete-time lowpass filter. So once again, here is the frequency response of the ideal lowpass filter. And below what I show the impulse response.

Again, it's a sine x over x type of impulse response. And again, we recognize that since in the frequency domain, this frequency response is real-valued. That means, as a consequence of the properties of the Fourier transform and inverse Fourier transform, that the impulse response is an even function in the time domain. And also, incidentally, the sine x over x function goes off to infinity, again, in both directions.

Now, we've talked about ideal filters in this discussion. And ideal filters all are, in fact, ideal in a certain sense. What they do ideally is they pass a certain band of frequencies exactly and they reject a band of frequencies exactly.

On the other hand, there are many filtering problems in which, generally, we don't have a sharp distinction between the frequencies we want to pass and the frequencies we want to reject. One example of this that's elaborated on in the text is the design of an automotive suspension system, which, in fact, is the design of a lowpass filter.

And basically what you want to do in a case like that is filter out or attenuate very rapid road variations and keep the lower variations in, of course, elevation of the highway or road. And what you can see intuitively is that there isn't really a very

sharp distinction or sharp cut-off between what you would logically call the low frequencies and what you would call the high frequencies.

Now, also somewhat related to this is the fact that as we've seen in the time domain, these ideal filters have a very particular kind of character. For example, let's look back at the ideal lowpass filter. And we saw the impulse response. The impulse response is what we had shown here.

Let's now look at the step response of the discrete-time ideal lowpass filter. And notice the fact that it has a tail that oscillates. And when the step hits, in fact, it has an oscillatory behavior.

Now, exactly the same situation occurs in continuous-time. Let's look at the step response of the continuous-time ideal lowpass filter. And what we see is that when a step hits then, in fact, we get an oscillation.

And very often, that oscillation is something that's undesirable. For example, if you were designing an automotive suspension system and you hit a curve, which is a step input, in fact, you probably would not like to have the automobile oscillating, dying down in oscillation.

Now there's another very important point, which again, we can see either in continuous-time or discrete-time, which is that even if we want it to have an ideal filter, the ideal filter has another problem if we want to attempt to implement it in real time.

What's the problem? The problem is that since the impulse response is even and, in fact, has tails that go off to plus and minus infinity, it's non-causal. So if, in fact, we want to build a filter and the filter is restricted to operate in real time, then, in fact, we can't build an ideal filter.

So what that says is that, in practice, although ideal filters are nice to think about and perhaps relate to practical problems, more typically what we consider are nonideal filters and in the discrete-time case, a nonideal filter then we would have a characteristic somewhat like I've indicated here. Where instead of a very rapid

transition from passband to stopband, there would be a more gradual transition with a passband cutoff frequency and a stopband cutoff frequency. And perhaps also instead of having an exactly flat characteristic in the stopband in the passband, we would allow a certain amount of ripple.

We also have exactly the same situation in continuous-time, where here we'll just simply change our frequency axis to a continuous frequency axis instead of the discrete frequency axis. Again, we would think in terms of an allowable passband ripple, a transition from passband to stopband with a passband cutoff frequency and a stopband cutoff frequency.

So the notion here is that, again, ideal filters are ideal in some respects, not ideal in other respects. And for many practical problems, we may not want them. And even if we did want them, we may not be able to get them, perhaps because of this issue of causality.

Even if causality is not an issue, what happens in filter design and implementation, in fact, is that the sharper you attempt to make the cutoff, the more expensive, in some sense, the filter becomes, either in terms of components, in continuous-time, or in terms of computation in discrete-time. And so there are these whole variety of issues that really make it important to understand the notion nonideal filters.

Now, just to illustrate as an example, let me remind you of one example of what, in fact, is a nonideal lowpass filter. And we have looked previously at the associated differential equation.

Let me now, in fact, relate it to a circuit, and in particular an RC circuit, where the output could either be across the capacitor or the output can be across the resistor. So in effect, we have two systems here. We have a system, which is the system function from the voltage source input to the capacitor output, the system from the voltage source input to the resistor output.

And, in fact, just applying Kirchhoff's Voltage Law to this, we can relate those in a very straightforward way. It's very straightforward to verify that the system from

input to resistor output is simply the identity system with the capacitor output subtracted from it.

Now, we can write the differential equation for either of these systems and, as we talked about last time in the last several lectures, solve that equation using and exploiting the properties of the Fourier transform. And in fact, if we look at the differential equation relating the capacitor output to the voltage source input, we recognize that this is an example that, in effect, we've solved previously.

And so just working our way down, applying the Fourier transform to the differential equation and generating the system function by taking the ratio of the capacitor voltage or its Fourier transform to the Fourier transform of the source, we then have the system function associated with the system for which the output is the capacitor voltage. Or if we solve instead for the system function associated with the resistor output, we can simply subtract H_1 from unity. And the system function that we get in that case is the system function that I show here.

So we have, now, two system functions, one for the capacitor output, the other for the resistor output. And one, the first, corresponding to the capacitor output, in fact, if we plot it on a linear amplitude scale, looks like this. And as you can see, and as we saw last time, is an approximation to a lowpass filter. It is, in fact, and nonideal lowpass filter, whereas the resistor output is an approximation to a highpass filter, or in effect, a nonideal highpass filter.

So in one case, just comparing the two, we have a lowpass filter as the capacitor output associated with the capacitor output, and a highpass filter associated with the resistor output.

Let's just quickly look at that example now, looking on a Bode plot, instead of on the linear scale that we showed before. And recall incidentally, and be aware incidentally, of the fact that we can, of course, cascade several filters of this type and improve the characteristics.

So I have shown at the top a Bode plot of the system function associated with the

capacitor output. It's flat out to a frequency corresponding to 1 over the time constant, RC . And then it falls off at 10 dB per decade, a decade being a factor of 10.

Or if instead we look at the system function associated with the resistor output, that corresponds to a 10 dB per decade increase in frequency up to approximately the reciprocal of the time constant, and then approaching a flat characteristic after that.

And if we consider either one of these, looking back again at the lowpass filter, if we were to cascade several filters with this frequency response, then because we have things plotted on a Bode plot, the Bode plot for the cascade would simply be summing these. And so if we cascaded, for example, two stages instead of a roll-off at 10 dB per decade, it would roll off at 20 dB per decade.

Now, filters in this type, RC filters, perhaps several of them in cascade, are in fact very prevalent. And in fact, in an environment like this, where we're, in fact, doing recording, we see there are filters of that type that show up very commonly both in the audio and the video portion of the signal processing that's associated with making this set of tapes.

In fact, let's take a look in the control room. And what I'll be able to show you in the control room is the audio portion of the processing that's done and the kinds of filters, very much of the type we just talked about, that are associated with the signal processing that's done in preparing the audio for the tapes. So let's just take a walk into the control room and see what we see.

This is the control room that's used for camera switching. It's used for computer editing and also audio control. You can see the monitors, and these are used for the camera switching. And this is the computer editing console that's used for online and offline computer editing.

What I really want to demonstrate though, in the context of the lecture is the audio control panel, which contains, among other things, a variety of filters for high frequency, low frequencies, et cetera, basically equalization filters. And what we

have in the way of filtering is, first of all, what's referred to as a graphic equalizer, which consists of a set of bandpass filters, which I'll describe a little more carefully in a minute. And then also, an audio control panel, which is down here and which contains separate equalizer circuits for each of a whole set of channels and also lots of controls on them.

Well, let me begin in the demonstration by demonstrating a little bit of what the graphic equalizer does. Well, what we have is a set of bandpass filters. And what's indicated up here are the center frequencies of the filters, and then a slider switch for each one that lets us attenuate or amplify. And this is a dB scale.

So essentially, if you look across this bank of filters with the total output of the equalizer just being the sum of the outputs from each of these filters, interestingly the position of the slider switches as you move across here, in effect, shows you what the frequency response of the equalizer is. So you can change the overall shaping of the filter by moving the switches up and down.

Right now the equalizer is out. Let's put the equalizer into the circuit. And now I put in this filtering characteristic. And what I'd like to demonstrate is filtering with this, when we do things that are a little more dramatic than what would normally be done in a typical audio recording setting. And to do this, let's add to my voice some music to make it more interesting. Not that my voice isn't interesting as it is. But in any case, let's bring some music up.

[MUSIC PLAYING]

And now what I'll do is set the low frequencies flat. And let me take out the high frequencies above 800 cycles. And so now what we have, effectively, is a lowpass filter. And now with the lowpass filter, let me now bring the highs back up. And so I'm bringing up those bandpass filters.

And now let me cut out the lows. And you'll hear the lows disappearing and, in effect, keeping the highs in effectively crispens the sound, either my voice or the music. And finally, let me go back to 0 dB equalization on each of the filters. And

what I'll also do now is take the equalizer out of the circuit totally.

Now, let's take a look at the audio master control panel. And this panel has, of course, for each channel and, for example, the channel that we're working on, of a volume control. I can turn the volume down, and I can turn the volume up. And it also has, for this particular equalizer circuit, it has a set of three bandpass filters and knobs which let us either put in up to 12 dB gain or 12 dB attenuation in each of the bands, and also a selector switch that lets us select the center the band.

So let me just again demonstrate a little bit with this. And let's get a close up of this panel. So what we have, as I indicated, is three bandpass filters. And these knobs that I'm pointing to here are controls that allow us for each of the filters to put in up to 12 dB gain or 12 dB attenuation.

There are also with each of the filters a selector switch that lets us adjust the center frequency of the filter. Basically it's a two-position switch. There also, as you can see, is a button that let's us either put the equalization in or out.

Currently the equalization is out. Let's put the equalization in. We won't hear any effect from that, because the gain controls are all set at 0 dB. And I'll want to illustrate shortly the effect of these. But before I do, let me draw your attention to one other filter, which is this white switch. And this switch is a highpass filter that essentially cuts out frequencies below about 100 cycles.

So what it means is that if I put this switch in, everything is more or less flat above 100 cycles. And what that's used for, basically, is to eliminate perhaps 60 cycle noise, if that's present, or some low frequency hum or whatever. Well, we won't really demonstrate anything with that.

Let's [? go ?] now with the equalization in, demonstrate the effect of boosting or attenuating the low and high frequencies. And again, I think to demonstrate this, it illustrates the point the best if we have a little background music. So maestro, if you can bring that up.

[MUSIC PLAYING]

And so now what I'm going to do is first boost the low frequencies. And that's what this potentiometer knob will do. So now, increasing the low frequency gain and, in fact, all the way up to 12 dB when I have the knob over as far as I've gone here. And so that has a very bassy sound. And in fact, we can make it even bassier by taking the high frequencies and attenuating those by 12 dB.

OK well, let's put some of the high frequencies back in. And now let's turn the low-frequency gain first back down to 0. And now we're back to flat equalization. And now I can turn the low frequency gain down so that I attenuate the low frequencies by much as 12 dB. And that's where we are now. And so this has, of course, a much crisper sound.

And to enhance the highs even more, I can, in addition to cutting out the lows, boost the highs by putting in, again, as much as 12 dB. OK well, let's turn down the music now and go back to no equalization by setting these knobs to 0 dB. And in fact, we can take the equalizer out. Well, that's a quick look at some real-world filters. Now let's stop having so much fun, and let's go back to the lecture.

OK well, that's a little behind-the-scenes look. What I'd like to do now is turn our attention to discrete-time filters. And as I've meant in previous lectures, there are basically two classes of discrete-time filters or discrete-time difference equations.

One class is referred to a non-recursive or moving average filter. And the basic idea with a moving average filter is something that perhaps you're somewhat familiar with intuitively.

Think of the notion of taking a data sequence, and let's suppose that what we wanted to do was apply some smoothing to the data sequence. We could, for example, think of taking adjacent points, averaging them together, and then moving that average along the data sequence. And what you can kind of see intuitively is that that would apply some smoothing.

So in fact, the difference equation, let's say, for three-point moving average would be the difference equation that I indicate here, just simply taking a data point and

the two data points adjacent to it and forming an average of those three. So if we thought of the processing involved, if we're forming an output sequence value, we would take three adjacent points and average them. That would give us the output add the associated time.

And then to compute the next output point, we would just simply slide this by one point, average these together, and that would give us the next output point. And we would continue along, just simply sliding and averaging to form the output data sequence.

Now, that's an example of what's commonly referred to a three-point moving average. In fact, we can generalize that notion in a number of ways. One way of generalizing the notion of a moving average from the three-point moving average, which I summarize again here, is to think of extending that to a larger number of points, and in fact applying weights to that as I indicated here, so that, in addition to just summing up the points and dividing by the number of points summed, we can, in fact, apply individual weights to the points so that it's what is often referred to as a weighting moving average.

And I show below one possible curve that might result, where these would be essentially the weights associated with this weighted moving average. And in fact, it's easy to verify that this indeed corresponds to the impulse response of the filter.

Well, just to cement this notion, let me show you an example or two. Here is an example of a five-point moving average. A five-point moving average would have an impulse response that just consists of a rectangle of length five. And if this is convolved with a data sequence, that would correspond to taking five adjacent points and, in effect, averaging them.

We've looked previously at the Fourier transform of this rectangular sequence. And the Fourier transform of that, in fact, is of the form of a $\sin(n\pi)/\sin(\pi x)$ curve. And as you can see, that is some approximation to a lowpass filter. And so this, again, is the impulse response and frequency response of a nonideal lowpass filter.

Now, there are a variety of algorithms that, in fact, tell you how to choose the weights associated with a weighted moving average to, in some sense, design better approximations and without going into the details of any of those algorithms.

Let me just show the result of choosing the weights for the design of a 251-point moving average filter, where the weights are chosen using an optimum algorithm to generate as sharp a cutoff as can possibly be generated. And so what I show here is the frequency response of the resulting filter on a logarithmic amplitude scale and a linear frequency scale. Notice that on this scale, the passband is very flat. Although here is an expanded view of it. And in fact, it has what's referred to as an equal-ripple characteristic.

And then here is the transition band. And here we have to stopband, which in fact is down somewhat more than 80 dB and, again, has what's referred to as an equal-ripple characteristic.

Now, the notion of a moving average for filtering is something that is very commonly used. I had shown last time actually the result of some filtering on a particular data sequence, the Dow Jones Industrial Average. And very often, in looking at various kinds of stock market publications, what you will see is the Dow Jones average shown in its raw form as a data sequence.

And then very typically, you'll see also the result of a moving average, where the moving average might be on the order of day, or it might be on the order of months. The whole notion being to take some of the random high frequency fluctuations out of the average and show the low frequency, or trends, over some period of time.

So let's, in fact, go back to the Dow Jones average. And let me now show you what the result of filtering with a moving average filter would look like on the same Dow Jones industrial average sequence that I showed last time.

So once again, we have the Dow Jones average from 1927 to roughly 1932. At the top, we see the impulse response for the moving average. Again, I remind you on an expanded time scale, and what's shown here is the moving average with just one

point. So the output on the bottom trace is just simply identical to the input.

Now, let's increase the length of the moving average to two points. And we see that there is a small amount of smoothing, three points and just a little more smoothing, that gets inserted. Now a four-point moving average, and next the five-point moving average, and a six-point moving average next. And we see that the smoothing increases.

Now, let's increase the length of the moving average filter much more rapidly and watch how the output is more and more smooth in relation to the input. Again, I emphasize that the time scale for the impulse response is significantly expanded in relationship to the time scale for both the input and the output. And once again, through the magic of filtering, we've been able to eliminate the 1929 Stock Market Crash.

All right, so we've seen moving average filters, or what are sometimes referred to as non-recursive filters. And they are, as I stressed, a very important class of discrete-time filters.

Another very important class of discrete-time filters are what are referred to as recursive filters. Recursive filters are filters for which the difference equation has feedback from the output back into the input. In other words, the output depends not only on the input, but also on previous values of the output.

So for example, as I've stressed previously, a recursive difference equation has the general form that I indicate here, a linear combination of weighted outputs on the left-hand side and linear combination of weighted inputs on the right-hand side. And as we've talked about, we can solve this equation for the current output y at n in terms of current and past inputs and past outputs.

For example, just to interpret this, focus on the interpretation of this as a filter, let's look at a first order difference equation, which we've talked about and generated the solution to previously. So the first order difference equation would be as I indicated here. And imposing causality on this, so that we assume that we are running this as

a recursive forward in time, we can solve this for y of n in terms of x of n and y of n minus 1 weighted by the factor a . And I simply indicate the block diagram for this.

But what we want to examine now for this first order recursion is the frequency response and see its interpretation as a filter. Well in fact, again, the mathematics for this we've gone through in the last lecture. And so interpreting the first order difference equation as a system, what we're attempting to generate is the frequency response, which is the Fourier transform of the impulse response. And from the difference equation, we can, of course, solve for either one of those by using the properties, exploiting the properties, of Fourier transform.

Applying the Fourier transform to the difference equation, we will end up with the Fourier transform of the output equal to the Fourier transform of the input times this factor, which we know from the convolution property, in fact, is the frequency response of the system. So this is the frequency response. And of course, the inverse Fourier transform of that, which I indicate below, is the system impulse response.

So we have the frequency response obtained by applying the Fourier transform to the difference equation, the impulse response. And, as we did last time, we can look at that in terms of a frequency response characteristic. And recall that, depending on whether the factor a is positive or negative, we either get a lowpass filter or a highpass filter.

And if, in fact, we look at the frequency response for the factor a being positive, then we see that this is an approximation to a lowpass filter, whereas below it I show the frequency response for a negative. And there this corresponds to a highpass filter, because we're attenuating low frequencies and retaining the high frequencies.

And recall also that we illustrated this characteristic as a lowpass or highpass filter for the first order recursion by looking at how it worked as a filter in both cases when the input was the Dow Jones average. And indeed, we saw that it generated both lowpass and highpass filtering in the appropriate cases.

So for discrete-time, we have the two classes, moving average and recursive filters. And there are a variety of issues discussed in the text about why, in certain contexts, one might want to use one of the other. Basically, what happens is that for the moving average filter, for a given set of filter specifications, there are many more multiplications required than for a recursive filter. But there are, in certain contexts, some very important compensating benefits for the moving average filter.

Now, this concludes, pretty much, what I want to say in detail about filtering, the concept of filtering, in the set of lectures. This is only a very quick glimpse into a very important and very rich topic, and one, of course, that can be studied on its own in considerable amount of detail.

As the lectures go on, what we'll find is that the basic concept of filtering, both ideal and nonideal filtering, will be a very important part of what we do. And in particular, beginning with the next lecture, we'll turn to a discussion of modulation, exploiting the property of modulation as it relates to some practical problems. And what we'll find when we do that is that a very important part of that discussion and, in fact, a very important part of the use of modulation also just naturally incorporates the concept and properties of filtering. Thank you.