MITOCW | watch?v=oJv4dsUID0Q

ALAN V. OPPENHEIM: During the last three lectures, we considered the design of infinite impulse response digital filters. The two main classes of design procedures that we discussed involved mapping continuous time designs to discrete time designs. That was the first class. And the second class was the use of algorithmic design procedures. In this lecture, I would like to discuss the design of finite impulse response digital filters.

And I remind you that as we've stressed in a number of the lectures, one of the very important aspects of finite impulse response digital filters is the fact that they can be implemented to have exactly linear phase. So finite impulse response filters, as opposed to infinite impulse response filters, are useful for problems where linear phase is an important constraint.

It turns out also-- and this isn't a point that we'll be touching on in these lectures-- but it turns out also that finite impulse response digital filters have some important advantages over infinite impulse response filters with regard to round off noise, that is, with regard to the effects of finite register length.

Now in general, the implementation of finite impulse response filters is considerably simpler for discrete time systems than it is for continuous time systems. Consequently, it turns out that there are not really available, or a wide variety of continuous time design procedures for finite impulse response filters as there were for infinite impulse response.

And consequently, the continuous time to discrete time mapping, as a design procedure, is not particularly applicable to the design of finite impulse response digital filters. So this includes, for example, the impulse and variant technique, and the use of the bilinear transformation. That is, those techniques are not particularly useful when we are talking about the design of finite impulse response filters.

As an additional interesting side point, if we were to begin with a finite impulse response analog or continuous time filter, and map it to a digital filter using the bilinear transformation, then in fact the resulting filter would not be a finite impulse response filter. The algorithmic procedures that we talked about in the previous lectures, of course, still do apply to the design of finite impulse response filters. And there also are some other algorithmic procedures that are discussed in the text, and that I will touch on just very briefly in today's lecture.

Now for the design of finite impulse response filters, there are basically three classes, or three types of design techniques that we'll be discussing. And in this lecture, what I intend to do is present the three of these, primarily with an eye toward giving you an appreciation for what the style of the design techniques is. That is, to give you an overview of these design techniques. But I won't dwell on many of the details.

So in the design of finite impulse response digital filters, then, we of course are talking about a unit sample response that is a finite length. And we'll choose the length, the interval over which the unit sample response is non-zero to be in the range from 0 to capital N minus 1. So the Z-transform of the finite impulse response filter is then given by the sum from 0 to capital N minus 1 of h of n, z to the minus n, which of course is a polynomial in z to the minus 1.

And we know that, for this class of systems, h of z has only zeros, except at z equal 0. That is, the poles of h of z can only fall at z equals 0, or z equals infinity. And I remind you also that for linear phase finite impulse response filters, the constraint on the unit sample response is that it basically is symmetrical about its midpoint. In other words, as we look at the unit sample response starting from the left hand end and starting from the right hand end, approaching toward the middle, then we see the same values. In other words, there's a symmetry that's required for linear phase.

The three basic design procedures that I'd like to talk about in this lecture, basically introduce in the lecture, are first of all, a design technique which is referred to as the window method. The second design technique that is commonly referred to as the frequency sampling method. And a third designed technique, which is an algorithmic design procedure which results in equiripple filters, which as I'll comment on again when we get to this point in the lecture, turn out to be important because they have some optimality properties. So these are the three basic design techniques, then, that I'd like to discuss in this lecture.

Well, to begin with, the window method is basically a method based on the following strategy. Let's suppose that we begin the design of the finite impulse response filter with a desired unit sample response, h sub d of n. For example, suppose that we wanted to design a low pass filter. We could begin with a desired unit sample response which was the unit sample response of an ideal low pass filter.

Now as you know, the unit sample response of an ideal low pass filter basically has a sin x over x type of envelope. In particular, the important characteristic is that it extends from minus infinity to plus infinity. So in fact, it doesn't correspond to a finite impulse response filter. But we could imagine obtaining the unit sample response of an FIR filter by perhaps truncating this infinitely long unit sample response so that we retain only the values in the interval from 0 to capital N minus 1.

Or more generally, we can imagine multiplying the desired unit sample response by a window, the window being 0 outside the interval from 0 to capital N minus 1. So the window method basically consists of beginning with a desired unit sample response, which in general will be infinitely long, or at least longer than the filter that we want to design. We apply to that a window, which is 0 outside the range 0 to capital N minus 1. And the result then is the unit sample response of the finite impulse response filter.

Well, what is the effect of this in the frequency domain? We know that multiplication in the time domain corresponds to convolution in the frequency domain. And consequently, the frequency response of the finite impulse response filter that results, is the periodic convolution of the desired frequency response. That is, the Fourier transform of the desired unit sample response, that convolved with the Fourier transform of the window, w of n

For example, if we were talking about a rectangular window, and in a rectangular window, w of n is simply constant over the range 0 to capital N minus 1, the magnitude of the Fourier transform of the window has a form roughly as I've indicated here. This is drawn for the case, N equals 8. And it's the magnitude of h of e to the j omega-- I'm sorry, of w of e to the j omega. If we were to remove the magnitude sines, then every alternate lobe would be flipped over in sine. So in fact, this is a sin nx over sin x curve.

So for the particular case of a rectangular window, we would have not the ideal low pass filter that we began with, but the ideal lowpass filter frequency characteristic convolved with the Fourier transform of the window. So that's depicted here where we have the ideal low pass filter. Superimposed on top of it is the Fourier transform of, in this case, a rectangular window.

And then to obtain the resulting overall frequency response of the finite impulse response filter, that would be obtained by convolving these two together. And we can see that the result of that, very roughly, is to, in fact, at the discontinuity, the almost exact effect is to integrate the window frequency response, the frequency response of the window. And the resulting frequency response for the finite impulse response filter then, as opposed to being the ideal characteristic that we desired, in fact has some ripples in it.

Generally, then, it has some deviation in the pass band. It has some deviation in the stop band. And it has a transition region from pass band to stop band. And that's introduced because of the convolution of this window frequency response with the desired or ideal frequency response.

Now ideally, what would we like as the frequency response of the window? Well, ideally what we would like, of course, is we would like this function to be an impulse. Because if it's an impulse, then we convolve it with the desired frequency response, and the result will again be the desired frequency response. The overshoot that we have in the pass band and the stop band has arisen because of the side lobe structure of the frequency response of the window. And the transition region has arisen because, primarily because of the width of this central lobe.

So, in fact, if we want to keep the deviation in the pass band and stop band small, then that implies that we would like to keep the side lobe structure of the window small, the window frequency response small. And if we would like to keep the transition band narrow, that implies that we would like to keep this main lobe narrow. And as I think that your intuition should agree with by now, generally, an attempt to make this function narrower in frequency will have the general effect of making the window longer in time.

So that, in fact, as we try to make the time window, w of n, or equivalently, the order of the finite impulse response filter smaller, the tendency will be toward a widening transition band in the frequency response of the filter. Likewise, as we allow the order of the filter to get longer, we have more flexibility in the filter. And consequently, we're able to control better the transition band and the overshoot in both the pass band and in the stop band.

Well, in particular, again continuing with the example of a rectangular window, if we look at the frequency-- the unit sample response for the case of an ideal low pass filter multiplied by a 51 point rectangular window, then the unit sample response we obtain is as indicated here. Actually, I've marked this as the desired unit sample response. It's only equal to the desired unit sample response in the interval from 0 to 50. And the desired unit sample response, of course, continues to be non-zero outside that range. Whereas multiplying by a rectangular window truncates it to be 0 outside this range.

The resulting frequency response for this particular example I've indicated here. And as opposed to the previous frequency response that I depicted, this is shown on a logarithmic frequency scale. There, in fact, is some ripple in the pass band, although it's not evident on a logarithmic frequency scale. But generally, you can see the effect of a transition band, and then a side lobe structure in the stop band. And you can see also that the asymptotic behavior of this is-- in the vicinity for this particular example-- is in the vicinity of minus 20 dB attenuation in the stop band.

So this then is an example of a low pass filter that could be obtained by applying a rectangular window to the ideal low pass filter. There are other possible choices for windows. In particular, two other choices that I'd like to mention are what are referred to as the Hamming window and the Bartlett window. As opposed to the rectangular window, the Bartlett window is a triangular window, which has the effect basically of making the transition at the ends of the unit sample response less abrupt than they are in the case of a rectangular window.

The Hamming window is a window which corresponds essentially to a cosine bell sitting on a small pedestal. So for this particular example, again, the low pass filter, the ideal low pass filter windowed, we can compare the frequency response obtained with the Bartlett window to that obtained with the rectangular window, and also with the Hamming window. So let's take a look at that.

Here we have the frequency response of the Bartlett window. And you can see generally that the ripples aren't as severe in the stop band as they were in the case of the rectangular window. And in fact, we can compare these simply by referring back to the rectangular window frequency response that we had previously. So the rectangular window lies slightly under, or actually asymptotically touches the frequency response obtained from the Bartlett window. But for the Bartlett window, the ripples in the stop band aren't as severe as they are for the rectangular window.

Now to compare both of these to the Hamming window, we can put on top of this the frequency response for the Hamming window. And in this case, you can see that the frequency response obtained with the Hamming window applied to a low pass filter has considerably better stop band attenuation than either the rectangular or the Bartlett window. In particular, it comes down to about minus 30 or minus 32 or so dB.

There is a slight price paid for this. And the price is, as you notice, that the transition band is slightly wider than it is for either of the other two windows. And in general, this is a type of trade off that arises again and again in the design of-- actually, always in the design of digital or analog filters. In particular, in using windows, there is always this trade off between improved stop band attenuation and a narrower-- or wider, rather-- transition band. As you are willing to allow the transition band to become wider, you can generally obtain some advantage in terms of stop band attenuation.

All right. So this is a quick view into the use of windows. In the text, in fact, there are quite a large number of windows that are discussed, besides the rectangular, the Bartlett, and the Hamming window. And similar frequency responses for filters designed using a variety of these windows is presented in the text.

The next design procedure that I would like to discuss is the design procedure that is referred to as the frequency sampling design procedure. And basically, the idea behind that procedure is to argue that if we have a desired frequency response, which I've indicated by the continuous line-- a desired frequency response-- then we could think of designing a digital filter by specifying samples of the desired frequency response at equally spaced points along the frequency axis.

Now we know that for a finite impulse response digital filter-- in general, for any finite length sequence-- we can represent it in terms of samples of its Fourier transform. That, in fact, is the discrete Fourier transform. So basically what we're saying is that we can begin with a desired frequency response and specify the discrete Fourier transform of the finite impulse response filter by specifying equally spaced samples of the desired frequency response. So we have then that the actual frequency response will correspond to capital N samples, equally spaced in frequency, of the desired frequency response characteristic.

Now one could say that, or one could inquire as to whether, in fact, if we do that, what we're able to implement is the desired digital filter frequency response. In other words, we know that we can implement a finite impulse response filter by using only the values specified in the discrete Fourier transform. And here we've specified a discrete Fourier transform that's unity in the pass band and it's zero in the stop band.

And one can ask, well, in fact, is that an ideal digital low pass filter? Well, the answer to that is no. And the reason is that it is important, very important to keep track of the fact that just because we specify the frequency response at equally spaced points along the frequency axis doesn't mean that, in between those points, the frequency characteristic has to be particularly well-behaved.

That is, the Fourier transform of the unit sample response is a continuous function of frequency. We're specifying samples of that continuous function of frequency. And there is implied an interpolation in between those samples. Well, in fact, if we were to look at a unit sample response whose discrete Fourier transform magnitude is unity in the pass band and zero in the stop band, one possible choice, in fact, is the unit sample response and frequency response that I've indicated here.

This is the unit sample response. This is the corresponding frequency response. The discrete Fourier transform of h of n, in fact, is exactly unity in the pass band and zero in the stop band. But the interpolation in between those points is not unity in the pass band and zero in the stop band. In fact, by simply putting the previous view graph on top of this one, we can see that this frequency response characteristic indeed goes through the specified frequency points.

But it is not particularly well constrained in between those points. So, in fact, for one example, if we chose this as the unit sample response corresponding to this set of frequency samples, the resulting frequency response of the finite impulse response low pass filter would be as we've indicated with this continuous curve.

Well, we know that there are other unit sample responses whose discrete Fourier transforms are exactly the same. The magnitude of the discrete Fourier transform is exactly the same as this one. In particular, we know that if we were to circularly rotate h of n, circularly shift h of n, the magnitude of the discrete Fourier transform would remain exactly the same.

So in fact, another unit sample response that would correspond to a frequency response, which likewise goes through the same frequency samples, is the one that we have here. This unit sample response is simply the previous one, circularly shifted by approximately n over 2, or n minus 1 over 2. And in this case, the frequency response characteristic is even worse than it was in the previous example.

So here we have a frequency response that has very wild, large fluctuations in the pass band, and even in the stop band, but still goes through the same frequency samples that we had specified previously. And we can see that by, again, putting these two on top of each other. And you can see, in this case again, that this frequency characteristic goes through the specified frequency samples. But it has a very wild behavior in between.

All right. So the frequency sampling method, then, is basically a method that consists of specifying samples of the frequency response, and then accepting the interpolation in between those samples that is dictated by the form of the unit sample response. And I stress that one has to be careful in the phase that's applied to this, or equivalently, the circular shift that's applied to the unit sample response, since as we've seen in the last several view graphs, that very seriously affects the interpolation in between these frequency points.

Furthermore, an additional point has been made, and that is that it is not-- even though we can implement a finite impulse response filter using the discrete Fourier transform, specifying the discrete Fourier transform to be unity in the pass band and zero in the stop band does not mean that we've implemented an ideal low pass filter.

All right. Well, we had-- this, of course, is not a unit sample response that we would be particularly anxious to use. Clearly, the behavior in the pass band and stop band is very severe, and in fact, too severe. So we could at least consider as somewhat reasonable the unit sample response and frequency response that we had previously, in this case, relatively well-behaved. That is, it ripples somewhat in the pass band and in the stop band. But now we could inquire as to whether there is some way of improving the pass band ripple and the stop band ripple.

We know, or we anticipate, that there's a price that we would have to pay for that. Probably the price, and indeed it turns out that the price we have to pay is the price of widening the transition band. Well, how could we widen the transition band? We could imagine beginning with this particular design, taking as one possibility one of the stop band constrained points. And allow it to move into the transition band so that, in fact, we have a wider frequency range that we're permitting the transition from pass band to stop band to go to.

And let me illustrate that here, where now what we've done is to take one of the frequency samples in the stop band, move it up into the transition band. And as we do that, you can see for this particular example, the pass band ripple hasn't changed too much. But in fact, the stop band ripple has changed somewhat dramatically.

As we allow this transition point to move farther up into the transition band, basically widening the general pass band region, what we'll see is that the pass band ripple become somewhat smaller. And there, in fact, is a trade off that is introduced between the pass band ripple and the stop band ripple. And let me illustrate that with this next example, where in this case, we have moved the transition point up somewhat further.

So that whereas in the previous case the transition point was someplace down here, we've now moved it farther up into the transition region, closer to the pass band. And the pass band ripple has become correspondingly smaller. So this, in fact, is a design technique that can be used for the design of, generally, band frequency selective filters, where we first of all specify frequency samples.

In fact, this can be used, really, for the design of any digital filter characteristic with transition regions. And then, perhaps, [AUDIO OUT] reducing the ripple in the pass band and/or the stop band, or in the various bands, by allowing an additional point to be used to widen the transition band. And as we move this point up or down, then, we will reduce the pass band and the stop band ripple.

We can also, of course, consider moving not just one point into the transition region, but two points or three points or more points. Obviously, as we do that, we will widen the transition band further. And at the same time, that offers some additional flexibility, which permits us to reduce the pass band and stop band ripple. So this then is basically the strategy in the frequency sampling method of digital filter design.

The final method that I would like to touch on just very briefly is a method which I referred to at the beginning of this lecture as an equiripple design method. And basically, the notion here is that we've seen that there is a trade off between the transition region on the one hand, and pass band and stop band ripple on the other hand.

Well, one could inquire as to whether, in at least in some sense, we could design for ourselves an optimum class of filters. And indeed, you can. And in fact, it's possible to show that the optimum strategy in terms of minimizing pass band or stop band ripple, or in terms of minimizing a transition band, is to design the filter to have an equiripple frequency response characteristic. Well, one way of setting up the optimal condition-- and it turns out that there are a variety of ways, and these are discussed in quite a bit more detail in the text.

But for example, if we pick a pass band cutoff frequency and a stop band cutoff frequency. Furthermore, if we, of course, are going to allow some deviation in the pass band and some deviation in the stop band, if in addition, we fix the ratio of stop band to pass band ripple, and if we then inquire as to what sort of characteristic should result if delta 1, the pass band ripple, or equivalently, delta 2, the stop band ripple is minimized-- this should be delta 2-- so if delta 1 and delta 2 are minimized, it turns out it can be shown that that implies an equiripple characteristic in the pass band, and an equiripple characteristic in the stop band. That is, the ripples in the pass band should alternate between the upper and lower limits. And the ripples in a stop band should alternate between the upper and lower limits.

In the text, there are discussed several algorithmic procedures for actually designing equiripple filters. We won't go through those details in this lecture. Mainly, I want to emphasize the point that there are procedures that allow you to implement designs of this type. And that designs of this type are optimum. Let me just show you one example of an optimum equiripple finite impulse response filter that was designed when our computer budget was particularly big.

This is an example of a length 251 point finite impulse response digital filter, designed using one of the algorithms that is discussed in the text. And you can see that, indeed, the equiripple characteristics are evident in the pass band. This curve corresponds to the frequency characteristics plotted in dB. So the pass band ripple doesn't show up. So with that scale expanded, we can see the ripple. And then we have the ripple in the stop band. There should be 125 ripples all together.

And in fact, we could see if that was true. We could [AUDIO OUT] one, two, three. But let's not do that. Obviously, there's a very sharp transition band. And what turns out to be minus approximately 80-- approximately minus 85 dB attenuation in the stop band. So this is an example of a very high order equiripple optimum finite impulse response digital filter that was designed using one of the algorithms discussed in the text.

OK. Well, this, as I indicated at the beginning of the lecture, was intended primarily as a very brief overview of finite impulse response digital filter design. Hopefully, it gives you some appreciation for the style of the design techniques. There are, of course, a large number of details, some of which are discussed in the text and some of which aren't. But in any case, this, then, concludes our discussion of the design of infinite impulse response and finite impulse response digital filters.

In the next lecture, we will begin a new topic, specifically the topic involving the computation of the discrete Fourier transform. Thank you.