

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.

[MUSIC PLAYING]

**ALAN
OPPENHEIM:** In the last several lectures, we've considered a number of procedures for designing digital filters for the class of infinite impulse response filters. And the two primary design techniques that we considered, in fact, the two useful design techniques that we developed, were the techniques which we refer to as impulse invariance, and also the bilinear transformation.

You recall, hopefully, that both of these techniques are techniques that correspond to mapping a continuous time or analog filter to a digital filter. What I'd like to do in this lecture is consider an example of the design of a digital low-pass filter using each of these techniques, using impulse invariance and the bilinear transformation. And hopefully this will give you a better feeling for how both of these techniques work.

The class of analog filters that I would like to use are the class of so-called analog Butterworth filters, which are a useful class of lowpass analog and digital filters, and also a relatively simple class. To first of all, define the class of analog filters that we're considering, they are the class of analog Butterworth filters. And this class of filters is defined by the analog system function H sub a of $j\omega$ magnitude squared of the form 1 over 1 plus $j\omega$ divided by $j\omega$ sub c to the $2N$.

Now, there are two parameters in this expression. One is what's referred to often as the cutoff frequency, ω sub c . And the second is the order of the filter, namely N . Let me remind you that we're looking at here an expression for the squared magnitude function. And so we have a $2N$ here. But in fact, N is what corresponds to the order of the filter.

Well, if we look at this frequency response characteristic, we observe that at ω equal to 0 , the gain or the magnitude squared is equal to unity. At ω equal to ω sub c , the j 's of course cancel out. At ω equal to ω sub c , this term then is equal to 1 , and we have for the magnitude squared function, a half. Or for the magnitude function, which is what we're looking at here, 1 divided by the square root of 2 .

So at ω equal to ω sub c , an analog Butterworth filter has a magnitude which is down by 1 over the square root of 2 . And the other important characteristic is that the

frequency response is a monotonic function of capital Ω . So that as capital Ω increases from 0 to infinity, the magnitude of the frequency response is a monotonic function.

Well, we observed that the cutoff frequency, the parameter ω_c , adjusts in a sense where we might divide the frequency characteristic into passband, transition band, and stopband. We have another parameter, which is the parameter N , the order of the filter. And basically the effect that the parameter N has is to affect the shape of this frequency response in the sense that if capital N is larger, the frequency response tends to be flatter longer, so that for a larger N than what I've drawn with the black curve, the frequency response might look more like this and then drop off sharper. Whereas if capital N is smaller, then it will be less flat in the passband and drop off more slowly.

So in fact, the way that I've drawn it here, this is the behavior of the frequency response curve as the parameter capital N increases. So, the higher the order of the filter, the sharper the drop from what we could consider as a passband region into a stopband region. And obviously then, the sharper the filter we want, the higher order filter we would design.

Now, to look at the pole 0 pattern or the 0 pattern really of the Butterworth filter, we can convert this expression into an expression relating the magnitude squared function expressed in terms of the Laplace transform variable s . And so this expression corresponds to the product $H(s) \times H(s^*)$, evaluated at $s = j\omega$, in which case the $j\omega$ is what corresponds to s , and so $H(s) \times H(s^*)$ is of the form $1 / (1 + s^2 / \omega_c^2)^N$.

Well, what we recognize is that this expression has poles, if we solve this equation. This expression has poles at of course the roots of the denominator polynomial or at $s = \pm j\omega_c$. That's simply solving the equation with this denominator polynomial equal to 0. Well, we can see more specifically where these poles lie in the s plane by recognizing that these are the $2N$ roots of minus 1. And then we have a j , which corresponds to $e^{j\pi/2}$. So we can rewrite this expression for the poles as $s = \pm j\omega_c e^{j\pi/2}$. That's one of the roots of minus 1, plus $k \times 2\pi$ where k is an integer.

This generates the roots of minus 1. We need to multiply by j , and we can do that by multiplying by $e^{j\pi/2}$, which is equal to $j \times \omega_c$, which is the cutoff frequency parameter for the Butterworth filter. So this then specifies where the pole locations

are for the Butterworth filter. For example, if we consider the case, let's say with n equal to 3, k let's say equal to 0. That says then that we have a pole at e to the $j\pi$ divided by 6, e to the $j\pi$ over 6, times e to the $j\pi$ over 2. In other words, we have a pole in the s plane at an angle of π over 6 plus π over 2, which corresponds to this location.

Then we'll have poles distributed in angle around a circle in the s plane, distributed in an angle where the angular spacing is π over 3. The π over 3 arising from a consideration of $2\pi k$ divided by $2N$, where in this case, N is equal to 6. So these poles are distributed equally spaced in angle on a circle in the s plane. For N equal to 3, the angular spacing is π over 3. For k equal to 0 that first pole corresponds to this one, and then we continue around.

Now these are poles distributed on a circle, which tends to look like the z plane. But I stress that what we're talking about here is the s plane. In particular, we're talking about the analog frequency response design as the class of analog Butterworth filters, which we're then going to map to digital filters. So this is a circle in the s plane. It's often referred to as the Butterworth circle. And it has a radius which is equal to ω_c .

In designing an analog Butterworth filter then, the procedure is to appropriately choose the parameters ω_c and capital N . Those are the parameters of the analog Butterworth filter. And then from the poles distributed on the Butterworth circle, this gives us the pole locations for the product, $H_{\text{sub } a} \text{ of } s$ times $H_{\text{sub } a} \text{ of } -s$.

We can then simply associate with the transfer function $H_{\text{sub } a} \text{ of } s$. The poles in the left half plane, to generate a stable filter, and then the ones which are contributed by the factor $H_{\text{sub } a} \text{ of } -s$ are the corresponding image poles in the right half plane. The procedure then is to design a squared magnitude function. Talking about a Butterworth filter, we choose these parameters ω_c and N . That tells us where the poles are located on the Butterworth circle. From the pole locations for the squared magnitude function, then we can factor that into $H_{\text{sub } a} \text{ of } s$ times $H_{\text{sub } a} \text{ of } -s$, by choosing the poles that fall in the left half plane, to give us a causal, stable analog filter.

All right. Well, that is an analog filter. What we would like to talk about is the design of a digital filter. And the procedure that we're going to go through is to choose a set of specs for the digital filter, and then decide on what the corresponding specifications should be for the analog filter, depending on whether we're using impulse invariance or the bilinear transformation, design the analog filter, and then map the result to a digital filter using one of those two

techniques.

So let's choose a set of digital filter specifications for a digital lowpass filter with a passband which occurs for frequencies now in the digital domain, for frequencies below 0.2π , and a stopband region that occurs for frequencies greater than 0.3π , and the transition band then, of course, is the region in between those.

So, we would like to design a digital filter, which stays within some limits in this passband region, drops below some limit, δ_s in the stopband region. And we'll choose as the limits in the passband, unity for the upper limit and $1 - \delta_p$ for the lower limit. Where $1 - \delta_p$ might not and in fact it won't turn out to be $1/\sqrt{2}$, in other words, this cutoff frequency will not necessarily correspond to the parameter ω_c for the Butterworth filter.

All right. Well, so let's choose a set of specifications. The ones that I've chosen are the requirement that the frequency response characteristic be within 1db of unity in the passband. In other words, the requirement that $1 - \delta_p$ be greater than or equal to minus 1db. So this deviation must be less than 1db. And then the requirement that in the stopband the frequency response characteristic is below minus 15db. In other words, the stopband attenuation for frequencies greater than 0.3π is greater than minus 15db. It's further down than minus 15db. So δ_s is less than or equal to minus 15db.

The one other specification that we'll impose, or in fact, it's really a consequence that will result from using a Butterworth filter, is the fact that the frequency characteristic is monotonic. So we have the passband edge, the stopband edge, the passband ripple or deviation, and the stopband ripple or deviation. And we can, of course, rewrite these specifications in terms of requirements on the transfer function or frequency response, the magnitude of the frequency response.

The requirement that the deviation be less than 1db in the passband is the requirement then that $20 \log_{10}$ of the magnitude of the frequency response-- that's the deviation db-- be greater than or equal to minus 1, or that the magnitude be greater than or equal to $10^{-0.05}$. And of course, we wanted also to be less than unity, and that's an implied condition. So it must be less than unity, and it must be greater than or equal to $10^{-0.05}$ for frequencies from 0 to 0.2π .

So in particular, since it's monotonic, we'll impose that condition at the frequency point 0.2π ,

corresponding to the end of the stopband and the beginning of the transition band. The second condition that we impose is the condition that by the time we get to ω equal to 0.3π , we would like the frequency response to be down at least 15 db, and more if possible. But the condition is that the $20 \log$ to the base 10 of the magnitude at ω equal to 0.3π must be less than or equal to minus 15. So this says we must be further down than 15 db, or we can translate that into a condition on the magnitude, so that the magnitude of that frequency is less than or equal to 10 to the minus 0.75 .

Well, let's consider two-- we want to consider two design procedures. Let's first consider the impulse invariant design procedure, in which case we would like to convert these specifications on the digital filter to corresponding specifications on the analog filter. Now, we know that from discussions that we've had previously that if we have an analog frequency response and we apply impulse invariance to the impulse response, the digital frequency response consists of the analog frequency response scaled in frequency so that capital Ω is equal to little ω divided by capital T . And then that scaled replica of H sub a added to itself delayed by little ω equal to 2π , 4π , et cetera. That is multiples of 2π .

So, that means then that the digital filter is scaled repeated replicas of the analog filter, and so we would choose the analog filter to have similar specifications to the digital filter, but with the corresponding frequency parameters dictated by this transformation, capital Ω equal to little ω , divided by capital T . Well, that of course neglects aliasing, and in fact as we go through the impulse invariant design, we'll initially neglect aliasing. And a rationale for doing this or a reason why, in fact, it's not unreasonable to do that, will emerge as we go through the example. So for the time being, in fact, we're going to neglect aliasing, so that we'll think of the analog and digital frequency responses as simply related by this substitution of frequency variable, capital Ω equal to little ω divided by capital T .

Well, that leads us then to the analog frequency response which I've depicted here. We had a factor of 1 over capital T in the amplitude of this expression. So for the digital filter to result with an amplitude of unity at dc, we would scale the analog frequency response by capital T . So the important values then in capital Ω are the ones corresponding to little ω equal to 0.2π , or since capital Ω is equal to little ω divided by capital T . We have a passband edge occurring at capital Ω equal to 0.2π over capital T , and a stopband edge equal to 0.3π over capital T .

At this passband edge, according to the specifications on the digital filter that we want, we

would like the frequency response for frequencies below this-- In other words, in the passband-- to be between capital T and capital T times 10 to the minus 0.05. That's the guarantee that we have less than 1 db ripple in the passband. And for frequencies greater than 0.3π , we would like to be below capital T times 10 to the minus 0.75. That's the guarantee that we're down 15 db or more in the stopband.

The way I've drawn it here, I've implied that I will exactly meet the specifications at the passband edge, and exactly meet the specifications at the stopband edge. You can imagine, of course, that if I did that and then converted to a digital filter using impulse invariance that there would be an effect due to aliasing, which in fact would result in my not meeting exactly the specifications at the stopband edge. And that's a point that we'll deal with in a few minutes.

All right, well now we have our specifications. Let me incidentally point out that there's this parameter capital T that we're carrying around. And as I'll justify partly on the basis of this example, the parameter capital T is an irrelevant parameter. It's a parameter that will disappear. We'll carry it through with us for this one example. But for other examples and problems that you work through the study guide, you should have absorbed the fact that capital T is irrelevant, and in fact capital T can be chosen equal to unity. But let's carry it through in this example, and just see how it disappears.

All right. Well, so we want the analog Butterworth filter meaning these specifications. And we want it at $\omega = 0$ to have a gain magnitude squared function to be T squared, so that the magnitude is T. So here we have the form then of the analog Butterworth filter. And in order to meet these specifications at this frequency and this one, we then have two equations. One equation says that at capital $\Omega = 0.2\pi$ over capital T, we would like the value of the magnitude function to be T times 10 to the minus 0.05, which implies that this denominator should be equal to 10 to the 0.1.

Second of all, at the stopband frequency, 0.3π divided by capital T, the fact that we want the magnitude to be T times 10 to the minus 0.75 implies the magnitude squared is of course the square of that. And we want then the denominator to be equal to 10 to the 1.5. So we have two points that we're trying to nail down. We have two parameters, ω_c and N. From these two points that we're trying to nail down, we get two equations. And we can now solve these equations for our parameters, ω_c and capital N.

Well, if we do that, and it's a little tedious to do it, let me point out incidentally since we're trying

to keep track of this capital T, that capital T drops down into the numerator and so, in fact, we can solve for capital N and $\omega_c T$. In other words, we can just move this down here algebraically. And so we have then instead of the parameter ω_c , we have the parameter $\omega_c T$. If you grind through these two equations, what results is a value for N equal to 5.8858 and probably some other digits. And $\omega_c T$ equal to 0.70474. And we could imagine that at this point we're done, except that the parameter N corresponding to the order of the filter, of course, has to be an integer.

So, what that means is that we can't use the value of N as I've indicated here. We need an integer value. Also if we were to round this down or truncate it down to a choice of N equal to 5, then since the order is lower you can imagine that the filter characteristics won't be as good. In other words, we're not going to meet these specifications. So the other choice then is to use instead of this value of N, use a value of N equal to 6. In other words, round up to the next integer value. And if we do that then these two equations will no longer be satisfied. We're now using a higher order filter than these equations would dictate. Of course, we're using an integer order, because that's what we need. We're using a sixth order filter. And what that means is that if we choose N equal to 6, we can satisfy this equation, or we can satisfy this equation, but we can't satisfy both of them.

Well, which one should we try to meet and which one should we exceed? Well, we know that with impulse invariance there is the issue of aliasing. And so in fact, in impulse invariant design it will often happen that although we design exactly to meet the stopband specifications, the stopband specifications won't be met because of aliasing. Consequently, given the flexibility that we have, due to the fact that we can actually implement a higher order filter than these equations dictate, to compensate for aliasing we can exceed the stopband specifications and meet exactly the passband specifications. So if we substitute for N, N equals 6 into this equation that then requires that $\omega_c T$ be such that we exactly meet the passband specifications, in which case this equation won't be met. In fact, what will happen at the stopband edge is that the stopband specifications will be exceeded.

If we do this, then we have for the equation that we solve for $\omega_c T$ from, the equation which specified the passband edge, which is where we're going to meet the specs. And what results then is for $\omega_c T$, this value, rather than the previous value that had been obtained from the two equations. Well, from these parameters then, we have that ω_c is 0.7032 divided by capital T. We can substitute these parameters into the form for the

squared magnitude function, $H_{\text{a}}(s)$ times $H_{\text{a}}(-s)$, and actually it's this raised to the $2N$, which is the 2 times sixth power, which is the 12th power. So this then is the form for the analog squared magnitude function.

Now notice that the parameter capital T still hasn't disappeared. We still have it. We have it there, and we have it here. But notice that when we substitute in for ω_c , this number divided by capital T if you track through that algebra, capital T reappears in this expression as a multiplier on s . That is we had 0.7032 divided by capital T . That gets reflected back up into this numerator, so that we have s times capital T . Well, let's see what that means.

Suppose that capital T were equal to 1. And we got an analog impulse response, $H_{\text{a}}(T)$, or the system function $H_{\text{a}}(s)$. If instead we reinserted capital T and considered $H_{\text{a}}(s)$ times capital T multiplied by capital T that impulse response would be related to this one by simply a time scaling. In other words, this one would be replaced by $H_{\text{a}}(t)$ divided by capital T . Well, let's look at this and ask what happens when we sample it at samples spaced by capital T .

So, we look at this impulse response with T replaced by n times capital T , as we've specified here. And we observe that no matter what we pick capital T to be that parameter disappears. Because $t = n$ times capital T gives us samples $H_{\text{a}}(n)$, which are exactly the same samples that we would get no matter how we chose the parameter capital T , or equivalently whether or not we chose the parameter capital T equal to unity.

I think that it requires a little reflection to absorb this point. It's basically tied to the fact that the frequency response is specified for the digital filter, not for the analog filter. And specifying the frequency response for the digital filter, in effect, specifies capital Ω times capital T . Well, I leave it to you to think about this a little. But the important point is that, in fact this parameter T , if we specify the filter in terms of digital specifications, this parameter capital T is really irrelevant.

All right. Well, so here is our analog design. And we can obtain $H_{\text{a}}(s)$ from this, according to the method that we talked about previously taking the left half plane poles of this expression. Converting those to a digital filter by using the transformation that we discussed in the previous two lectures, expanding, for example, in terms of residues and poles, and then applying the impulse invariant transformation. And if we do that, what results is a digital lowpass filter, which was the lowpass filter that we set out to design. And let me just show how

those specifications, or how that design comes out.

So this, for the example that we've been talking about, is a graph of the frequency response characteristics. Here is the magnitude as a function of digital frequency ω . And here is the same frequency response plotted in terms of db. And we see that we had in the specifications a passband edge at 0.2π , a stopband edge at 0.3π . At 0.2π we asked that the gain be down by 1db. And of course, it's hard to tell exactly that we're down 1db, but in fact we are. At ω equal to 0.3π , our specification was that we wanted to be down by 15 db.

In fact, we overcompensated, because of the fact that we had to round the order of the filter up to an integer value. And you can see, in fact, that if you look at this carefully, we exceed the specifications of 0.3π by a little bit. In other words, we're down by a little more than 15 db. In fact, for this particular filter you can imagine that the effective aliasing is, in fact, rather minimal. Because this frequency response, by the time we get out to π is essentially 0. Here we're down to minus 75 db. And that's, of course, the amount that would get aliased back in. So this is the resulting filter frequency characteristics for the impulse invariant design where we exceed slightly the stopband edge specifications.

Now, the next thing that I'd like to go onto is the corresponding design, the design or the corresponding Butterworth filter, now using the bilinear transformation rather than using impulse invariance. I remind you that the basic idea with the bilinear transformation, as we've discussed in the last lectures, is that we basically choose the analog specifications by pre-warping the critical frequencies for the digital filter through the nonlinear distortion curve corresponding to the bilinear transformation.

So, we would start here for the digital filter. Of course, here we're talking about a monotonic filter rather than one that ripples. But we have a passband edge at 0.2π , a stopband edge at 0.3π . We reflect those critical frequencies through the arctangent curve to obtain the corresponding critical frequencies for the analog Butterworth filter. When the analog Butterworth filter is designed, we then obtain the digital Butterworth filter by transforming this filter back through the bilinear transformation.

So, let's look at how this design works then. We have, as we've just reflected on that curve, we begin with the digital frequency specifications, reflect those through the bilinear warping curve to the analog specifications, and consequently the critical frequencies-- and let me emphasize first of all that just as with impulse invariance, you can show that the parameter capital T is an

irrelevant parameter. We could, in fact, carry it all the way through. And what would happen is that we would find that no matter what we chose capital T to be, the digital filter design that resulted would be exactly the same. So in fact, for this example I'll work through it simply with capital T equal to 1.

All right. With capital T equal to 1, we reflect these critical frequencies into this passband frequency edge and this stopband edge. And consequently, the analog filter-- this is the analog filter-- the analog filter that we design is one that meets these specifications. And consequently we have, again, the two equations that say that at the passband edge, we want the magnitude to be within 1db of unity. And at the stopband edge, we want the magnitude to be less than, down by more than 15 db. And the two equations that result are these, exactly the same as we had in the impulse invariant case, except that the frequencies are different.

So now substituting in for the form of the analog Butterworth filter into these expressions, and setting this up for equality in these expressions, what results then are the two equations which are identical to the two equations that we had for the impulse invariant design, except that the frequencies at which the passband and stopband edges of the analog Butterworth filter fall are different. They're different because of the warping of the bilinear curve, or the pre-warping that we have to apply.

So, these are the two equations. And if we solve these two equations for capital N, the resulting value for capital N is 5.3 and some digits. Notice incidentally that it's less than it was for the impulse invariant design. And in general that's a characteristic of the bilinear transformation. But so it's less, but it is not an integer. So the first thing we can ask again is, could we just simply truncate this down or round it down to the nearest integer value, in other words, N equals 5. But of course if we did that, then we wouldn't be able to pick omega sub c, so that these equations were satisfied or so that the specifications were met or exceeded.

So in fact, we need to choose, we have to choose N equal to 6. We have to choose the next highest integer value. Choosing N equal to 6, we then have the choice of exactly satisfying this equation, or exactly satisfying this equation, or in fact not satisfying either one of them, but playing back and forth between the two. Well let's do it in a particular way. We can, for example, exactly meet the stopband specifications.

In other words, we can use N equals 6 in this equation, and solve for omega sub c, in which case the passband specifications will be exceeded. Recall that in the impulse invariant design

we did it the other way around. We met exactly the passband specifications, in which case the stopband specs were exceeded. But there we had a good reason for doing that. The reason that we did that is because we knew in advance that in any case the passband specs wouldn't be met exactly, because of the effect of aliasing. For the bilinear transformation, we don't have a problem with aliasing. So, in fact, we're free to play with this flexibility in a more general way.

All right. Well, in any case then if we meet the stopband specs, substituting N equals 6 in this equation, and solving for ω_c ; we end up with a value for ω_c , which is 0.76622. And so this then gives us the parameters ω_c and N for the analog Butterworth filter. We can then use those specifications, again to generate the pole pattern for the squared magnitude function $H_a(s)H_a(-s)$. Factor this by choosing the left half plane poles in s plane, and then mapping this function to a digital filter using the bilinear transformation.

And let's now look at how this filter comes out as compared with the digital Butterworth filter that we obtained by using impulse invariance. Well again, I have a display of the magnitude of the frequency response and also the gain in db. From what you may recall from the curve that I showed previously, this looks roughly the same. It's a monotonic filter and drops off. It's down by some amount at 0.2π , and the stopband is at 0.3π . In particular, according to the specifications, we wanted this to be down by no more than 1db at 0.2π .

Well, it's hard again to say that this is 1db or less. But if we look at 0.3π , we can see that this is in fact exactly down by 15 db. And that's the way that we designed it. We designed it to exactly meet the stopband cutoff and exceed the passband cutoff. And then the frequency response continues on from there. It's interesting, in fact, to compare the design that we got from the bilinear transformation with the design that we got previously with impulse invariance. And let me do that by superimposing the two of them. So that the one that I'm putting on now is the impulse invariant design, and let me just lay them on top of each other.

OK. You can see incidentally that looking at the frequency response on a magnitude plot there, in fact, are some differences. This line has gotten a little thicker. That is, the two don't exactly superimpose. But you can't really see the differences. Whereas you can when we look at this on a logarithmic plot. In particular, this is the frequency response that corresponded to the bilinear transformation. This is the frequency response that corresponded to impulse invariance. And this is, in general, a characteristic of the use of the bilinear transformation. It's a result essentially of the frequency distortion, because we've taken the frequency axis and

the continuous time frequency axis. As you run along it at a constant rate, it gets compressed more and more along the digital frequency axis.

That wasn't well said. But what I really mean to say is as we walk along here at a constant rate, the farther we walk, the faster we're running along the $j\omega$ axis. And that's why, in fact, the use of the bilinear transformation on a lowpass filter will give better attenuation at the higher frequencies than impulse invariance will. And I don't know if you can really see the fact that the passband specs were exceeded for the bilinear design. In fact, when you look at these view graphs in the study guide, I think you'll see that for impulse invariance the curve is slightly lower than it is for the bilinear transformation

OK well, look. This is a simple example that hopefully at least illustrates for you the basic procedure involved in carrying through a design of an infinite impulse response filter using impulse invariance and the bilinear transformation. There are other examples that are considered in the text. There is a discussion of digital Chebyshev filters, in other words digital designs beginning from an analog Chebyshev filter. And you'll also have an opportunity to carry through some designs in the study guide.

A point that I would like to mention briefly is that most of the discussion, in fact all of the discussion that we've gone through, has focused entirely on lowpass filters. In fact, the discussion generalizes in a fairly straightforward way to bandpass, bandstop, highpass, et cetera filters. In particular, as is discussed in some detail in the text, there are a class of frequency transformations that allow lowpass designs to get converted to bandstop, bandpass, highpass, et cetera designs.

So this concludes our discussion of design procedures for recursive or infinite impulse response digital filters. In the next lecture I'll discuss the design of finite impulse response filters, and in particular introduce a number of designs, one of which corresponds to a class of algorithmic designs. Thank you.

[MUSIC PLAYING]