

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:**

Throughout these lectures, we have been considering the analysis and the implementation of discrete time linear shift invariant systems, or what we've been referring to as digital filters. In this lecture, I would like to begin the discussion of the design of digital filters. Now the rationale for digital filters is exactly the same as it is for analog filters. In particular, we have taken advantage several times of the important property of linear shift in variance systems that complex exponentials are eigenfunctions of this class of systems. In other words, then with a complex exponential input, the output is a complex exponential of the same complex frequency multiplied by the system function $H(j\omega)$. Or equivalently, with a sinusoidal input, the output is likewise sinusoidal with the same frequency but with a change in amplitude and with a change in phase.

Now what this property of linear shift invariant systems permits us to consider is the separation of signals that have been added when these signals occupy different frequency bands. Specifically, we can consider a signal composed of the sum of two signals, one of which we want, the other of which we don't want, or we would like to suppress. And if those two signals are in different frequency bands-- in other words, their spectra occupy different segments of the frequency axis, then we can choose the system function for the linear shift invariant system so that it is unity in the band of frequencies that we want to keep and at 0 in the band of frequencies that we want to reject. That of course, is exactly the same notion as we have in continuous time or analog systems or filters.

So for example, if we had a signal, which was the sum of two components, one of which was low frequency, and that we wanted to keep, and the other of which was high frequency and that we wanted to suppress, then we can consider extracting the wanted signal with a system whose frequency response ideally is unity at the low frequency end and at zero at the high frequency end, or after some cutoff frequency, which I've denoted by ω_P .

So this then is one type of filter and one class of filtering problems, generally, what we would refer to as frequency selective filters. We've illustrated that here with the example of a low-pass filter. But of course there are other kinds of frequency selective filters-- for example, high-pass, band-pass, band-stop, multiple band filters, et cetera. Furthermore, the filtering problem, both digital and analog, is of course, not restricted to just simply frequency selective filters. We can think of problems-- we can talk about a wide variety of problems, in fact-- where the frequency characteristics that are desired are not piecewise constant as they are in the frequency selective filter case, but might be non-piecewise constant such as, for example, with the case of a differentiator, in which case the desired frequency characteristic is in fact, linear with frequency, or in the inverse filtering problem where the desired frequency characteristic is in general of a somewhat arbitrary shape. And so in this more general sense, the filtering problem can be thought of in terms of the implementation of a linear shift invariant system with a specified frequency response characteristic that we wish to achieve.

To carry out the discussion, all the discussion that will be carrying out today and in fact throughout most of these lectures on digital filter design, I'll relate primarily to frequency selective filters and in fact, to low-pass filters, although it's important to keep in mind that the digital filter problem, the digital filter design problem, is not restricted to just that class of filters.

Well, for the case of an ideal low-pass filter, or a low-pass filter, this is the desired frequency characteristic. And so what's the problem? Well, we have seen in the last several lectures that for the implementation of digital filters, it's particularly convenient to have a filter or a system whose system function is representable as a rational function of z , or equivalently whose input-output characteristic can be related by means of a linear constant coefficient difference equation.

A linear constant coefficient difference equation, or a rational system function, can't achieve exactly an ideal low-pass filter characteristic as we have here, and in general can't achieve any ideal frequency selective characteristic where the transition from one band to another is as abrupt as we would like it to be ideally. So in fact, what's required is to approximate this ideal frequency characteristic, and the approximation of the ideal frequency characteristic, and the design of a rational transfer function to implement that approximation is what the digital filter design problem is about.

So whereas we talk ideally about a filter characteristic, as I've indicated here, in fact, we have to allow some deviation from this ideal characteristic so that we would permit as an example a

frequency response, rather than being unity in the passband, to deviate from unity by some amount. And let's designate that by say, δ_P , denoting a passband deviation. So that in fact where we might specify is a filter which in the passband, in the band of frequencies that we wish to pass, rather than being exactly unity, is constrained in the interval between unity and $1 - \delta_P$.

Similarly, we might permit the frequency characteristics, rather than requiring them to be exactly 0 in a stopband region. We might permit the frequency characteristic to deviate from 0 by some amount which we can denote as δ_S , which is the stopband deviation, in which case we would ask that the filter fall between 0 and δ_S , that is that the filter that we actually design be restricted to this frequency range. And finally recognizing that in reality we can't achieve a discontinuity in the frequency response as sharp as I've indicated here, we would permit also a transition region. That is, we would permit a region, after which the frequency characteristic must be below δ_S , and before which the frequency characteristic must be between 1 and $1 - \delta_P$, but allowing a region for it to make the transition from the passband to the stopband.

So these are the kinds of specifications on a filter. And a typical kind of characteristic then that might result is a filter perhaps that ripples between these limits 1 and $1 - \delta_P$ in the passband, and then makes the transition into a stopband region and perhaps wiggles around in the stopband region, sometimes perhaps touching the limits and sometimes not.

So what the digital filter design problem is then, is given a set of specifications on the filter, recognizing that we can implement an ideal filter, but in fact allowing some deviation from the ideal filter. The digital filter design problem is then to design a rational transfer function, which approximates in some sense, this ideal filter maintaining the specifications of passband and stopband deviation that we've imposed on the problem.

Now, there are several classes of design techniques that we can consider. The first class of design techniques, which in fact, we won't be talking much about, is the class that I've referred to as analytical design techniques. Basically what I mean by that are design techniques that permit the approximation of the ideal frequency characteristic.

The approximation is carried out through an analytical procedure and results in a closed form transfer function. That in fact is a very common class of techniques for continuous time or analog filter design. And that leads us to the second class of design techniques, namely the

class of design techniques that I refer to as mapping continuous time to discrete time.

Well, why in the world would we ever want a map from continuous time to discrete time? Well, the reason, very simply, is that a lot of work. And a lot of results have developed for the design of continuous time filters. And in fact, a lot of these procedures are analytical procedures. They lead to analytical, or they lead to closed form designs and their analytical procedures that can be used. Now, if it's possible to map those designs to digital designs, then in fact, that's something we should do simply to take advantage of the kinds of results that have been worked out previously in the design of analog filters.

An important point to keep in mind, however, is that we don't mean by this approximating an analog filter with a digital filter. That's not the objective, that's a point that we've stressed throughout these lectures. Our objective is not to approximate an analog filter with a digital filter. But if in fact, we can utilize analog designs in designing digital filters, then that's something we should do. And in fact, there are some procedures that allow us to do this that are very successful.

The third class of design techniques are what are referred to as algorithmic or computer-aided design techniques. And the basic idea here is that there are a variety of algorithmic procedures, iterative procedures that we can go through to carry out the digital filter design. And we'll see a brief discussion of a number of these in the next several lectures, both for infinite impulse response filters and for finite impulse response filters.

Now, in fact on the issue of infinite impulse response and finite impulse response, it turns out that it's convenient to separate the design issues for infinite impulse response and for finite impulse response filters. There are techniques for the finite impulse response case that are not applicable to the infinite impulse response case, and vice versa. And consequently, as we talk about digital filter design techniques, we'll be talking first about the design of infinite impulse response, or what we alternatively refer to as recursive digital filters, and then we'll talk separately about some design procedures for finite impulse response or non-recursive digital filters.

So for this lecture, and in fact for this lecture and the next lecture, and in fact for this and the next two lectures, we'll be focusing entirely on the design of infinite impulse response digital filters. And in the lecture following that, we'll consider some design procedures for finite impulse response filters.

All right, well, the first class of procedures, design techniques that I'd like to discuss, are ones related to mapping from continuous time to discrete time-- that is, taking analog filter design techniques and utilizing them for the design of digital filters. So the procedure basically is that we're mapping from continuous time to discrete time. We have a continuous time system function, $H_a(s)$, which is the Laplace transform of the impulse response, $h_a(t)$, the H standing for analog. And we'd like to map that to a digital filter transfer function, $H(z)$, or equivalently map the impulse response to a digital filter unit sample response, $h(n)$. And we'd like to do this, of course, in such a way that if this was a good filter in the analog domain, then this will be a good filter in the digital domain.

Well, of course in designing digital filters by implementing and mapping of the type that we're talking about, there are some obvious restrictions that we would like to impose, one of which is that we would like the behavior of $H_a(s)$, the analog system function, on the $j\omega$ axis to map to a corresponding behavior on the unit circle. The point, of course, is that an analog system function or an analog filter, an analog frequency response, when we talk about the frequency response, we're looking in the s -plane on the $j\omega$ axis. And for the digital filter, it is the behavior of the system function on the unit circle that dictates what the frequency response is like. If the analog filter has a good frequency response-- in other words, it has a good behavior on the $j\omega$ axis-- that's what we would like to have mapped over to a good behavior in the z -plane on the unit circle.

So this then is the first condition that we would like to impose. And a second condition that it's reasonable to impose is that a stable analog system function mapped to a stable digital system function. In other words, we'd like to be confident that our design procedure is such that if we had a good analog filter and it was stable, that when we were all done, we would end up with a good digital filter that was also stable. So these, then, are two of the conditions that are basic to any design procedure that maps from continuous time to discrete time.

Well, there are a number of procedures that are available. One of the first that tends to come to mind when you think of mapping a continuous time filter to a discrete time filter is to approximate in some sense, let's say the differential equation for the analog filter, by simply replacing the derivatives by differences. So the first method that I'd like to talk about is a method which corresponds to going from the analog filter to the digital filter by mapping differentials in the analog domain to differences in the digital domain. And let me tell you in advance that what we'll see, in fact, is that although this is intuitively one of the first methods

that tends to come to mind, that in fact, it is not a particularly good method in terms of the basic guidelines that we've set down.

But let's go through the method anyway. Here we have an analog system function, H sub a of S . And the corresponding differential equation, the filter then is described in terms of a linear combination of derivatives of the output equal to a linear combination of derivatives of the input.

Now we want to convert this in some way to a difference equation. And so we can consider replacing y sub a of t by y of n , y of n , of course is the output of the digital filter, in such a way that the derivative, a first derivative sampled at t equal to n times T gets replaced by the first difference of the output of the digital filter. Where the first difference I'm defining as the first forward difference. That is, y of n plus 1 minus y of n divided by T . So the idea here is basically to say, well look, I can generate a difference equation from this differential equation by replacing the derivatives by differences-- in the particular case I'm talking about, by forward differences.

And what will result is a difference equation, and consequently what also results is a digital filter. Incidentally, I'm talking about this here for the case of forward differences in the text to see the contrast. We talk about a similar procedure, but with the use of backward differences rather than forward differences.

All right, well let's see, in fact, what happens then. We want to replace a first derivative by a first forward difference. More generally, what we would want to replace a k -th derivative by is the k -th forward difference, where the k -th forward difference is defined iteratively. In other words, the k -th forward difference of y of n is the first forward difference of the k minus first forward difference of y of n . Simply the k -th forward difference to correspond to the k -th derivative is the first difference implemented over and over and over again, k times.

So what results then by making this substitution, is that the differential equation that we had, which was a linear combination of the derivatives of y of t equal to a linear combination of the derivatives of x of t , that's replaced by the same linear combination of the k -th difference of y of n equal to the corresponding linear combination of the k -th differences of x of n . And this then is a difference equation, because of the fact that each of these differences, each of these four differences, or the k -th forward difference, involves differences of y of n . That is why even plus 1 minus y of n So this is a difference equation derived from the differential equation by

replacing the k-th derivative by the k-th forward difference.

Well let's see what this means in terms of the mapping from the s-plane to the z-plane. First of all, let me remind you that in the continuous time domain, the Laplace transform of the derivative of a time function is equal to s times the Laplace transform of the time function. That is, the operation of differentiation in the Laplace transform domain gets carried over to a multiplication by s.

Similarly, we can look at the z transform of the first difference, the z transform of the first difference, $y[n+1] - y[n]$ divided by T, simply by applying the properties of z transforms, results in $z - 1$ divided by T times the z transform of $y[n]$. So the operation of taking the first difference corresponds to multiplying the z transform by $z - 1$ divided by T.

So in the continuous time domain, the derivative corresponded to multiplication by s. In the z transform domain, first differencing corresponded to multiplication by $z - 1$ divided by T. And it's straightforward to show that, in fact, the k-th forward difference corresponds to multiplying by $z - 1$ over T to the k. And we know that the k-th derivative corresponds to multiplication by s to the k.

So if you track that through, put that into the difference equation, and obtain the system function for the digital system, and compare it with the system function for the analog system, what you see resulting is that the operation of replacing differentials by differences corresponds to obtaining the transfer function of the digital filter from the transfer function of the analog filter by the substitution s equal to $z - 1$ divided by T. So it's basically a mapping then, S being replaced by $z - 1$ over T, or equivalently, z is equal to $1 + s$ times T. First point then is that this technique in fact corresponds to a mapping from the s-plane to the z-plane, with s replaced by a function of z.

Well, let's see what this mapping is, in fact, and decide whether it satisfies the objectives that we set down before. Here is a picture of the s-plane, and here's the $j\omega$ axis. Below we have the z-plane, the imaginary axis, real axis, and the unit circle. And the first question is, what does the $j\omega$ axis in the s-plane map to in the z-plane? Well, we substitute $s = j\omega$, so that z is equal to $1 + j\omega$ times T. Consequently, the $j\omega$ axis in the s-plane for this particular method maps to this line in the z-plane.

Well, is that good? No, it's not good, because it says that if we saw a good filter as we ran up

and down this axis, we would see a good filter as we run up and down this line. However, where we are interested in the filter characteristics ending up are on the unit circle, which in fact, are only close to this line when we're in the vicinity of z equal to 1.

So in fact, the first condition that we wanted isn't satisfied, namely the $j\omega$ axis does not get mapped to the unit circle. Furthermore, stable analog filters do not necessarily map to stable digital filters. For example, if we had a pole, let's say in the s -plane there, then it would fall in the z -plane, let's say there.

Well, that's all right. That's a stable pole, stable analog filter mapping to a stable digital filter. However, here's an even more stable analog filter. And if this pole is negative enough, then in fact, it falls outside the unit circle in the z -plane, and consequently the digital filter will be unstable.

That incidentally is a well-known property in numerical analysis of forward differences-- that is, the notion that forward differences are unstable. And this is one way of seeing that by interpreting it in terms of a mapping from the s -plane to the z -plane. We know incidentally that at least in an intuitive sense, it's reasonable to imagine that if we pick samples of a continuous time signal close enough together, that at least intuitively we ought to be able to approximate a derivative by a difference. And in fact, that fits in with the kinds of things that we've been seeing here in the following sense.

If we very highly oversample the signal, much higher than the Nyquist rate, then in fact, the portion of the z -plane where the signal spectrum falls is the portion of the z -plane around in the vicinity of z equal to 1. And you can see that in that case, that portion of the unit circle is pretty close to this line which was the mapping from the $j\omega$ axis into the z -plane. And so consequently, if we oversample rather extensively, then we can, in fact, map, more or less, our analog filter to a digital filter. But in fact, in the more usual case, the digital signal spectrum occupies the entire unit circle. And clearly away from the regions equals 1, the unit circle deviates rather dramatically from this vertical line.

All right, so this is a method then of filter design, which didn't satisfy either one of our initial guidelines. Namely, it didn't map the $j\omega$ axis to the unit circle. And also, it didn't map a stable analog filter to a stable digital filter.

Well, let's go on to a second method, which is better in both of those respects. And this is a method that is commonly referred to as the method of impulse invariance. And again, it's a

filter design method, which intuitively is very reasonable. The basic idea of impulse invariance is to convert an analog filter to a digital filter, simply by choosing the unit sample response of the digital filter to be equally spaced samples of the impulse response of the analog filter. So the unit sample response h of n is chosen to be equally spaced samples, samples spaced by T , equally spaced samples of the analog or continuous time filter.

Well, in at least one sense, we can see that this certainly leads to a good filter. And that is, if we had a continuous time analog filter that had good impulse response characteristics, then those good impulse response characteristics would be carried over to the digital filter unit sample response. For example, if we were interested in a filter with very low ringing or very short impulse response, then if we had an analog filter with those characteristics, then the digital filter would likewise have good impulse response characteristics.

However we've been phrasing most of our discussion in terms of frequency response characteristics, we know from the discussion of sampling that we had a number of lectures ago that periodic sampling of this type results in a frequency response for the digital system, which is basically an aliased version of the frequency response for the analog system. That is, the digital frequency response is given by the sum that I've indicated here, which consists of scaled replicas of the Fourier transform of the analog filter, repeated over and over again with a spacing in little ω of 2π . Well, let's take a look at that in a little more detail.

Here we have the analog frequency response. And let's say that we're considering this for the case of a low-pass filter, which is what we've tended to focus on in this lecture. So that if we had an analog low-pass filter, cutoff frequency between minus ω_a and plus ω_a , then in forming the digital filter frequency response, we first want to form $h_a(j\omega_a/T)$, and then shift that in ω_a by integer multiples of 2π and add up those results.

So the first step is the scaling, basically replacing Ω by ω/T , in which case plotted as a function of ω , the cutoff frequency is then ω_a/T and minus ω_a/T , so that this step results in a linear scaling of the frequency axis so that Ω is replaced by ω/T . The second step then is to add together replicas of this analog frequency response separated in ω by integer multiples of 2π . And that's what I've indicated here. Of course, this periodically repeats over and over again, but I've sketched it just from minus 2π to plus 2π . We have the factor $1/T$ out in front for the amplitude.

But an important aspect of this is that we have in going from the analog frequency variable to the digital frequency variable, a linear scaling in the frequency axis. That doesn't show up particularly for a piecewise constant frequency characteristic as I have here. But generally, if I had some complicated shape, then when I convert to $h \text{ sub } a \text{ of } j \text{ omega over } T$, that basic shape will be preserved. And it will only be distorted by a linear scaling in the frequency axis. And that is an important advantage of the technique of impulse invariance.

Now there are some problems, one in particular. And this is the major problem. And that is that as I've drawn this example, we've considered an analog filter whose frequency response is strictly bandlimited. And so of course there is no aliasing that results.

But more generally, the kind of frequency characteristic that we would have would look perhaps like this, as an approximation, admittedly a somewhat crude approximation, to this ideal low-pass filter. In which case, as we add up the replicas of the scaled frequency response, then we in general will have interference or aliasing between different terms in this sum. So that in that case, while we start with an analog frequency characteristic that's desirable, the resulting digital frequency characteristic will be distorted somewhat because of the aliasing that results due to the separate terms that are added together in this sum. And one of the important aspects of implementing an impulse invariant filter design, then, is to in some way account for and hopefully minimize the effect of this aliasing.

All right, so with the impulse invariant method, one of the important advantages of impulse invariance is the fact that it tends to preserve good time domain characteristics. That is, the impulse response characteristics were preserved. And also, it provides a linear scaling from analog frequency to digital frequency so that if we had an analog filter with a very complicated frequency response, then we could be assured that the corresponding digital filter except for aliasing would have that same shape, but only with a linear scaling of the frequency axis.

It should be pointed out incidentally that impulse invariance doesn't imply step invariance, and you'll see this in the problems that you work through the study guide. Impulse invariance does not imply step invariance so that in fact, if it was the step response characteristics that we wanted to preserve, then the technique here isn't what we would use. We would use a modified technique, which would be not impulse invariance, but step invariance.

Well, let's finally look at what impulse invariance means in terms of a mapping from the s-plane to the z-plane. Let's look at it in a slightly different way than we've looked at it here. And

we can do that by expressing the analog system function in a partial fraction expansion, in terms of a sum of factors of this form. So we have residues A_k , poles s at $s = s_k$, and we're expressing the system function as a linear combination of terms of this form.

If there are multiple order poles, this expression is slightly more complicated. And what I'm about to say can be generalized to that case. But let's consider it here just for the case of simple poles, so that the system function in fact can be expanded out in the form that I have here.

All right, well, if this is the analog system function, then you should recall that the analog impulse response-- that is, the inverse Laplace transform of this-- is the sum of exponentials with amplitudes A_k , the same amplitudes as we have here, and complex exponentials $e^{s_k t}$. And this should be a t , not a T . That is time, t .

So this then is the form of the impulse response, which corresponds to this system function. And now the impulse invariant method says, replace this by a unit sample response for the digital filter, which is obtained by replacing t by n times T . That is, obtaining the unit sample response of the digital filter by sampling the impulse response of the analog filter, in which case we get the sum of $A_k e^{s_k n T}$. And now time, t , is replaced by n times T .

A sampled unit step, a sampled analog unit step, results in a discrete time unit step. And so consequently, the unit sample response that we end up with is $A_k e^{s_k n T}$, the same coefficients as we had here, $e^{s_k n T}$ times the unit step.

Well, we can write that in a slightly different way. We can write that as the sum of A_k , and some factor, $e^{s_k n T}$ is what it is. Some factor raised to the n -th power times the unit step. And consequently we can write down by inspection the z transform or system function that corresponds to this unit sample response. In particular, it is $H(z) = \sum_k A_k e^{s_k T} z^{-n}$, that's simply by inspection of this expression.

All right. Well, so we started with an analog system function of this form, these coefficients in these poles. We ended up with a digital system function of this form, the same coefficients, and poles instead of at s_k , as they were in the analog case, at $e^{s_k T}$.

So this says then that the impulse invariant method maps a pole in the s -plane which was at s_k , maps it to a pole in the z -plane at $z = e^{s_k T}$. Well, does that mean

that impulse invariance is basically a method which maps s to z according to this transformation.

Well no, it doesn't do that. What it does is it maps the poles in the s -plane to poles in the z -plane according to this mapping. The other thing that it preserves are these coefficients, the coefficients $A_{sub k}$, that is the residues of the poles. So it maintains the residues of the poles. It maps the poles according to $z = e^{sT}$.

The 0 s, by the way, in general will not come out to have been mapped according to the same mappings $z = e^{sT}$. In other words, if we had a 0 at $s = s_0$, as we'll see actually in an example that we'll work next time, the 0 in the z -plane will not come out, will not come out at $z = e^{s_0 T}$. The poles are mapped according to that mapping, and the residues are preserved. And that is essentially what defines the impulse invariant mapping.

Well, we had our two initial guidelines. One is that we want the behavior on the $j\omega$ axis to map to the behavior in the digital domain on the unit circle. Does the impulse invariant method do that? Well, it does, except for the effect of aliasing. But basically, it is the analog frequency response that dictates what the digital frequency response is. And so in that sense, impulse invariance satisfies that objective. The second question is whether it maintains the ability of the filter. And the answer there also is that it does, so we can see that very easily.

In particular, if we have an analog pole with a real part $\sigma_{sub k}$, an imaginary part $\omega_{sub k}$, then the digital pole, the pole of the digital filter, maps to $z_{sub k} = e^{s_{sub k} T} = e^{\sigma_{sub k} T} e^{j\omega_{sub k} T}$. And the magnitude of this is equal to of course, the product of the magnitudes. The magnitude of this term is unity. So all that we care about is the magnitude of the $\sigma_{sub k} T$. Well, if the analog filter is stable, $\sigma_{sub k}$ is negative. If $\sigma_{sub k}$ is negative, then either the $\sigma_{sub k} T$ has a magnitude less than 1. So consequently, if the real part of $s_{sub k}$ is less than 0, negative, then the magnitude of $z_{sub k}$, the magnitude of the pole is less than 1.

So this in fact is a digital filter design technique that has a number of advantages. And in fact, it's a very useful design technique. It's one of the standard digital filter design techniques.

In the next lecture, I will briefly present an example of an impulse invariant design, just simply to remind you actually again of the fact that aliasing is an issue, number one, and number two, that it maps poles according to $z = e^{sT}$, but not 0 s. In fact, the 0 s end up someplace else. And in addition in that lecture, we'll talk about another design technique,

which is likewise one of the classical digital filter design methods, which is referred to as the bilinear transformation. And the lecture following that then I will show some examples of some digital filter designs, using both impulse invariance and the bilinear transformation to compare them. Thank you.