Digital Filtering

Rick Aster and Brian Borchers

October 17, 2008

Digital Filtering

We next turn to the (very broad) topic of how to manipulate a sampled signal to alter the amplitude and/or phase of different frequency components. There is an incredible amount of literature on this subject, and we will only be able to scratch the surface here. There are many reasons for wanting to filter a signal including:

1. Noise rejection/Signal enhancement
2. Remove an instrument response from the signal
3. Differentiate or integrate
4. Change the sampling rate
5. Artistic effects in audio and video.

Some particular types of filters that we will look at include **low pass**, **high pass**, and **band pass** filters which cut off portions of the frequency spectrum while allowing other frequencies to pass through the filter. In implementing these filters we will want to achieve the desired frequency response while otherwise distorting the signal as little as possible.

We will concentrate our analysis on filters which are themselves linear time invariant systems. This will enable us to apply all of the techniques that we have previously developed for LTI systems to analyze the performance of the filter. However, it is also possible to construct more complicated nonlinear filters which are not LTI systems.

In analyzing filters we will again encounter the concept of **stability**. A linear filter is stable if the corresponding LTI is stable (i.e., has an impulse response that eventually goes to zero). If we try to use an unstable filter to process real data, we would likely find that small amounts of noise in the input build up in the output to intolerable levels.

In some cases a filter is designed to process the signal in **real time** with little or no delay, while in other cases we must receive the entire signal before we can begin processing it. There is generally a preference for linear filters which are causal, since these filters can be implemented in real time. If an acausal filter needs to look ahead only a few samples, then we can implement the filter in real time by simply inserting a buffer before the filter and allowing the filter to delay its output by a few samples.

In practice, much of the human effort that goes into designing and implementing digital filters is about tradeoffs between the desired frequency and phase response of the filter and difficulty and cost of the implementation of the filter.

Filtering by Direct Manipulation of the FFT

One very simple approach to filtering a sampled signal is to compute its FFT and then manipulate the individual components of the FFT to achieve a desired frequency or phase response. This approach gives us complete freedom to control the frequency and phase response of the filter.

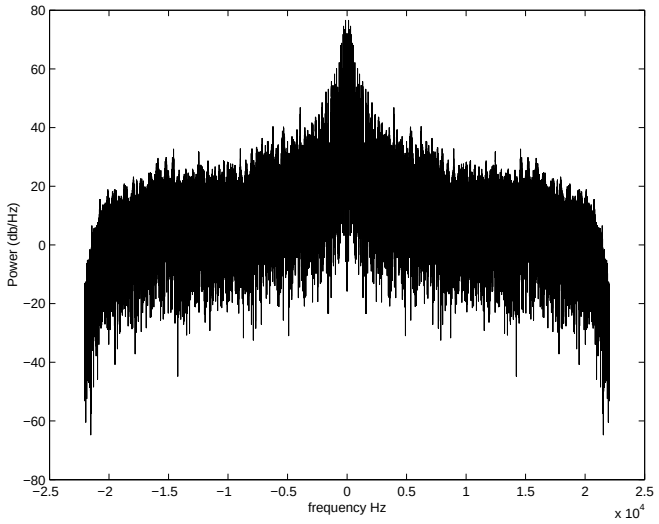

Figure 1: Spectrum of the original signal.

There are two significant costs associated with implementing a filter in this fashion. The first problem is that computing the FFT of a signal can (depending on the sampling rate and time duration of the signal) be very computationally intensive. For very long signals, computing the FFT of the entire signal may not even be practical. The second issue is that since we must have the entire signal in hand before we can begin filtering, we cannot use this method for real time filtering. It is also easy to introduce acausal effects that may not be realistic or desirable using simple FFT modification.

For example, consider a 20 second long recording of some guitar music. (The audio clip and MATLAB codes for this example will be made available on the class web site.) The audio is digitized according to the consumer audio CD standard at a sample rate of 44.1 Khz, with 16 bits per sample. For this example, we've combined the left and right channels into one mono channel. Thus the 20 seconds of audio requires $20 \times 44100 \times 16$ bits, or 1.76 megabytes of storage. We'll store the samples in MATLAB as 8 byte double precision numbers, which expands the storage requirements by a factor of four to about 8 megabytes. This is fairly large, but still well within the memory size limits of our computers.

At the sampling rate of 44.1 Khz, the Nyquist frequency is 22.05 Khz. Is a practical matter, most of us can't hear (and the speakers in our classroom can't reproduce) much above about 15 Khz. Before sampling, this signal was passed through an analog anti-aliasing filter that eliminated all frequencies above 22.05 Khz. Thus the sampling rate has been chosen to effectively reproduce all of the frequencies in the original music while avoiding aliasing problems.

We read the signal into MATLAB and compute its FFT. Since there are 882,000 real values in the original signal, the FFT also has 882,000 complex components. Since there are 882,000 frequency components over a frequency range of 0 to 44,100 Hz, each component of the FFT represents a frequency range of 0.05 Hz. Figure 1 shows a plot of the absolute values of the FFT versus frequency. The vertical aix represents power. We have used a dB scale. The horizontal access represents frequency in Hz. For convenience, we have used the MATLAB command **fftshift** to rearrange the entries in the FFT so that 0 Hz is at the center of the spectrum.

Now, suppose that we want to low pass filter the signal, eliminating all frequency components above 2 Khz. We do this by simply setting to 0 those elements of the FFT that correspond to frequencies above 2 Khz. Figure 2 shows a plot of the revised spectrum. At all frequencies below 2 Khz, we've left the spectrum alone, while at all frequencies above 2 Khz, we've zeroed out the entries in the FFT.

We can play the filtered signal, and hear that it sounds much like the original recording, but somewhat "dull." The guitar notes are at frequencies between about 400 Hz and 1 Khz. However, as a

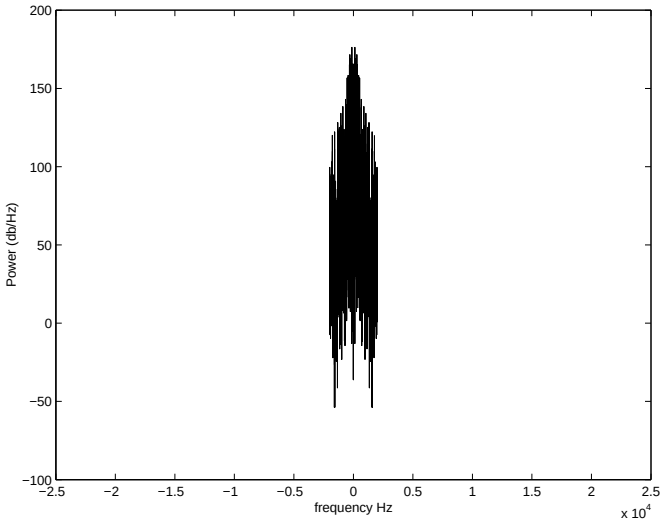

Figure 2: Spectrum after low pass filtering.

guitar string plays a note, the string also vibrates at multiples of the base frequency. These **harmonics** are what give the guitar its particular tone. By filtering out the harmonics, we've effectively dulled the tone of the guitar.

We can also try filtering out the fundamental frequencies and just listen to the higher harmonics. Figure 3 shows the spectrum after filtering out everything below 1 Khz. When you listen to the playback of the filtered signal, you'll still be able to hear the original music, because the harmonics still carry the tune.

Phase Shifts

So far, we've only adjusted the amplitude of various frequency components in the FFT. A filter which doesn't change the phases of any of the components of the FFT is called a **zero–phase filter**. Hidden within the phase of the complex numbers in the FFT is the information about when the various notes appear in the signal. Adjusting the phases of the FFT components can do some interesting things to the signal.

Continuing our example, we'll set the phase of each complex number in the FFT to 0 by taking the absolute value of each component. When played back, the resulting signal bears little resemblance to the original signal. The original frequencies are all present, but the order of the notes has disappeared completely.

In general, filtering that effects phase can cause distortion in the signal that will make it virtually unrecognizable. However, there is one important type of phase adjustment that can be tolerated and is fact useful in many contexts. We will consider adjusting the phase of each frequency component of the FFT by an amount proportional to its frequency.

That is, if the original signal contains a frequency component of the form

$$\phi(t) = Ae^{2\pi ift} \tag{1}$$

then we will adjust this to

$$\hat{\phi}(t) = Ae^{2\pi ift+icf} \tag{2}$$

where c is some constant of proportionality. What does this do to the signal? The signal is shifted by cf radians, or $cf/(2\pi)$ cycles. Since the time length of each cycle is $1/f$, $\hat{\phi}(t)$ is $\phi(t)$ shifted in time by

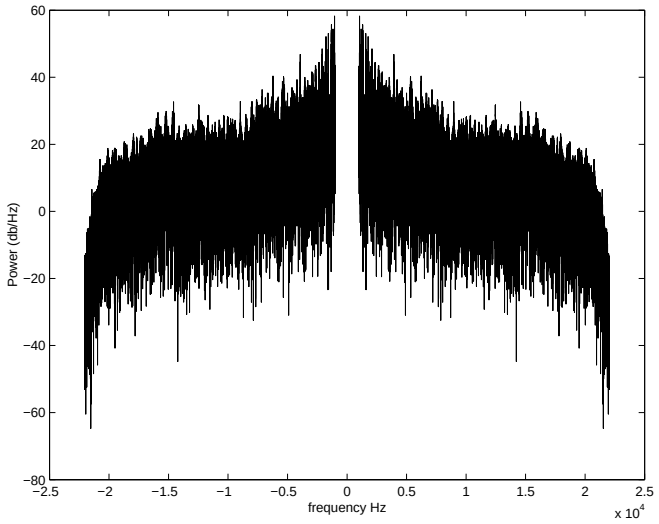

Figure 3: Spectrum after high pass filtering.

$(cf/(2\pi))(1/f) = c/(2\pi)$. Notice that this time shift is independent of f! Thus if we apply a phase shift of cf at each frequency, then we'll get a consistent time shift of $c/(2\pi)$.

A filter which shifts each phase in the FFT by an amount proportional to its frequency is called a **linear phase filter**. The time shift introduced by a linear phase filter can sometimes be a nuisance. However, there is a clever technique for correcting for the time shift. We apply a linear phase filter to our signal, then time reverse the filtered signal and apply the same filter a second time, and finally time reverse the twice filtered signal. This has the effect of shifting the signal forward and backward in time by the same amount. It also effectively squares the frequency response of the filter. This technique is implemented in the MATLAB command **filtfilt**.

Returning to our original example, suppose that we multiply each component of the FFT by e^{i15f}. This effectively adds $15f$ to the phase angle of each component of the FFT. For example, at f=22000 Hz, the phase is shifted by $\phi = 330000$ radians, which is 52,521 cycles, or 2.39 seconds. Similarly, at 100 Hz, the phase is shifted by $\phi = 1500$ radians, or 238.7 cycles, which is also 2.39 seconds. We then invert the FFT to recover the filtered signal.

Note that the direction of this phase shift is backward in time. That is, at time $t = 0$, we hear what was originally in the signal at $t = 2.39$ seconds. What do you expect to hear during the last 2.39 seconds of the playback? Remember that the FFT assumes that the entire signal is periodic.

Finally, we'll consider another common and often harmless phase shift. Suppose that the phase of each component of the FFT is adjusted by π. This is equivalent to multiplying the component of the FFT by $e^{i\pi}$, which is just -1! Since the FFT is a linear transformation of the original signal, we can easily compute the effect of this phase shift on the original signal. The inverse FFT of minus one times the FFT of the original signal is minus one times the inverse FFT of the FFT of the original signal, or just minus the original signal.

For many purposes, $\phi(t)$ and $-\phi(t)$ are indistinguishable signals. In our audio example, this phase phase shift makes no discernible difference, because your hearing system effectively analyzes the amplitudes of different frequency components and not their phases.

Finite Impulse Response Filtering

By *finite impulse response* or *FIR* filters, we refer to linear operators which have finite duration impulse responses. Such filters can be easily implemented by simply convolving the input signal with the impulse response. Since the impulse response is typically very short (perhaps as few as 10 samples), this convolution can be done directly without using the FFT.

Finite impulse response filters are invariably stable because they have no recursive components (internal feedback.) Once the input goes to 0, the output will return to zero within a period of time determined by the length of the impulse response. In the following, M will be the length of the filter sequence, N will be the length of the input sequence, n will be used as a time index, and k will be used as a frequency index.

A common and easy to understand example is the symmetric, M-point (M odd) *running mean* filter, which is defined as

$$w_n = \frac{1}{M}\Pi_M = \begin{cases} 1/M & \text{for } |n| \le (M-1)/2 \\ 0 & \text{for } |n| > (M-1)/2 \end{cases} . \tag{3}$$

The M filter impulse response values $w_0, w_1, ..., w_{M-1}$, are often referred to in this context as *weights* . Convolution of an arbitrary sequence, y_n, with this particular w_n results in a sequence with frequency characteristics (according to the convolution theorem)

$$Z_k = Y_k \cdot \mathrm{DFT}[w_n] = Y_k \cdot \frac{1}{M} \sum_{n=-(M-1)/2}^{(M-1)/2} e^{-\imath 2\pi kn/N} \tag{4}$$

Recall from our lecture notes on Fourier Theory in Discrete Time (equations (59) and (60)) that

$$\sum_{n=-M}^{M} e^{-\imath 2\pi fn} = \frac{\sin(N\pi f)}{\sin(\pi f)} \tag{5}$$

where $N = 2M + 1$. Thus

$$Z_k = Y_k \cdot \frac{1}{M} \frac{\sin(M\pi k/N)}{\sin(\pi k/N)} . \tag{6}$$

The net result is a low-pass filter with a Dirichlet kernel frequency response function. The DFT of w_n is real or *zero phase* because w_n is real and symmetric about zero.

Note that although this low pass filter has a zero phase contribution, it is also acausal and thus can only be implemented on a pre-recorded signal. This is easily gotten around with the implementation of a pre-event memory in the recording system.

As we have already seen, linear phase response is a desirable property of a filter. All M-point, real-valued FIR filters with symmetric weights have this property, as we can see by expressing the frequency response as

$$W_k = \sum_{n=0}^{M-1} w_n e^{-\imath 2\pi kn/N} = e^{-\imath \pi k(M-1)/N} \sum_{n=-(M-1)/2}^{(M-1)/2} w_n e^{-\imath 2\pi kn/N} \tag{7}$$

$$= e^{-\imath \pi k(M-1)/N} \left(2 \sum_{n=1}^{(M-1)/2} w_n \cos(2\pi kn/N) + w_0 \right) = P(k) \cdot A(k) . \tag{8}$$

The phase factor $P(k)$ is complex with magnitude one, so it only adjusts the phase. Furthermore, the phase adjustment is a linear function of k. Meanwhile, the amplitude factor $A(k)$ is real, so it only changes the relative amplitude at different frequencies.

The MATLAB command **conv** can be used to convolve the filter sequence w with the input sequence x. One problem with this is that the convolution will lengthen the sequence by $M - 1$ samples. This is because the response of the filter continues after the end of the input signal. If these samples are unwanted, you can simply truncate the filtered signal. e.g.

```
>> y=conv(x,w);
>> y=y(1:N);
```

An alternative is to use the MATLAB command **filter**. This command is designed for more complicated IIR filters which are specified by two vectors (discussed later in these notes). However, it can be used with a FIR filter by specifying the filter weights as the first argument, and "[1]" as the second argument. e.g.

```
>> y=filter(w,[1],x);
```

Now suppose we have some desired continuous frequency filter characteristic, $\Omega(f)$, and we wish to construct an FIR realization, specified by the N weights, w_n. As our realization is discrete, we let $\Omega(f)$ be periodic in f, and apply the inverse Fourier transform of the desired response on the Nyquist interval to obtain

$$w_n = \int_{-f_s/2}^{f_s/2} \Omega(f) e^{\imath 2\pi n f} \, df \ . \tag{9}$$

However, the resulting sequence (9) may have an infinite number of nonzero w_n. Consider, for example, the perfect low pass filter, with a desired cutoff frequency of f_s/α, defined by

$$\Omega(f) = \Pi(\alpha f/2) \tag{10}$$

where we've taken $f_s = 1$. The inverse Fourier transform (9) then gives

$$w_n = 2\int_0^{1/\alpha} \cos(2\pi f n) \, df = \frac{2}{\alpha}\mathrm{sinc}(2n/\alpha) \tag{11}$$

which has an infinite number of nonzero w_n.

The sinc function decays as n^{-1}. What happens if we simply truncate the series to M terms, bounded by $\pm(M-1)/2$? In this case we are multiplying the ideal frequency response by the boxcar function, which has the same DFT as the running mean filter (without the $1/M$ normalization).

$$\sum_{n=-(M-1)/2}^{(M-1)/2} (1) \cdot e^{-\imath 2\pi k n/N} = \frac{\sin(M\pi k/N)}{\sin(\pi k/N)} \equiv D(M,N,k) \tag{12}$$

which is the familiar Dirichlet kernel. The frequency response of our truncated realization is thus the convolution of the desired response with the Fourier transform of the above discrete boxcar function weighting. This particular realization is thus not especially desirable because the Dirichlet kernel is a very oscillatory function which doesn't fall off particularly rapidly with frequency. The result is the introduction of large side lobes to the frequency response of this filter realization. We can reduce this problem by applying less abrupt truncation and/or by taking N to be as large as possible. This brings us back once again to the issue of *windowing*, which arose in different contexts in our earlier discussion of power spectral estimation.

Although usually not an optimal way to design filters, windowing the infinite sequence defined by (9) provides a simple way of obtaining useful closed forms for FIR filter weights. Some examples of windowed realizations of ideal low-pass filters for $\alpha = 4$ (filter corner at 1/2 of the Nyquist) are shown in Figures 4, 5, 6.

Because of the unique correspondence between an N-length sequence and its N DFT coefficients, an N-length FIR filter can be uniquely specified by N DFT coefficients. Another design method for obtaining FIR filter weights, called *frequency sampling*, is thus to specify frequency characteristics at up to N desired frequencies and then take the IDFT, rather than the inverse continuous FFT, as we did in (9). This gets around the problem of truncating an infinite number of weights, as the IDFT produces

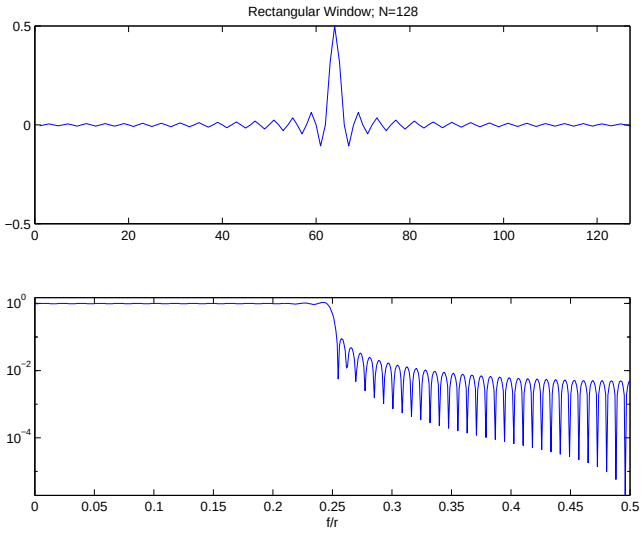

Figure 4: FIR weights and response for a 128-point rectangular window FIR realization of a low pass filter with a desired cutoff frequency of $f = r/4$.

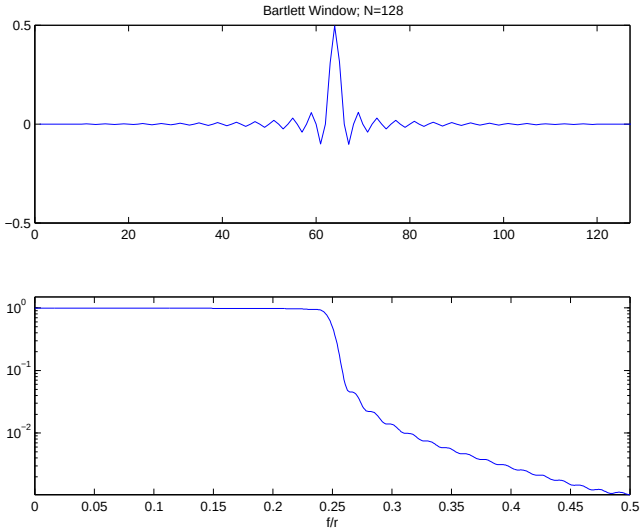

Figure 5: FIR weights and response for a 128-point Bartlett window FIR realization of a low pass filter with a desired cutoff frequency of $fr/4$. Note the reduction in ripple near the transition band relative to the simple truncation series (Figure 4).

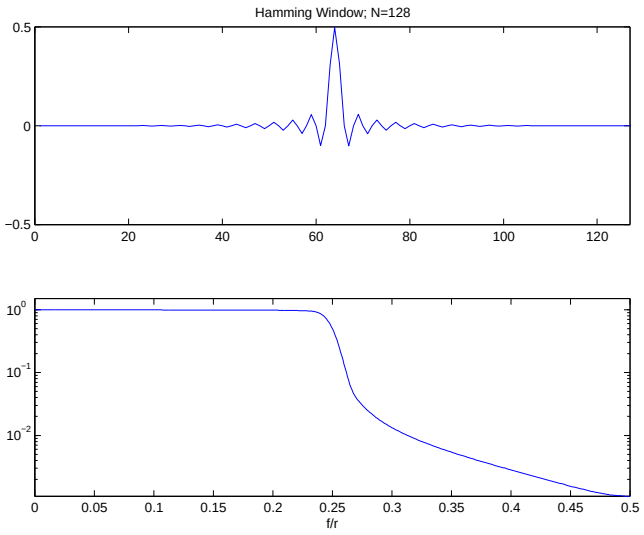

Figure 6: FIR weights and response for a 128-point Hamming window FIR realization of a low pass filter with a desired cutoff frequency of $r/4$. Note the reduction in ripple near the transition band relative to the simple truncation series (Figure 4) and the Bartlett window (Figure 5). The tradeoff for smoother response and better rejection outside of the desired passband is to have a more gradual transition.

exactly N weights. For example, the perfect low pass filter realization, where the passband is defined from $k = -(M-1)/2$ to $k = (M-1)/2$ becomes

$$w_n = \frac{1}{N} \sum_{k=-(M-1)/2}^{(M-1)/2} (1) \cdot e^{\imath 2\pi nk/N} = \frac{1}{N} \frac{\sin(\pi nM/N)}{\sin(\pi n/N)} = \frac{1}{N} D(M, N, n) \ . \tag{13}$$

Convolution of an input series of length N with (13) is identical to simply taking the DFT of the input series, setting the frequency components for $|k| > M$ equal to zero, and then inverse transforming the modified k-series back to the n domain via the IDFT.

In practice, we determine the desired filter length M, then pick N so that a filter of length M covers all of the frequencies for which we want a nonzero response. Once the filter sequence is computed, we can apply the filter to a sequence of arbitrary length by convolving the filter sequence with the input sequence.

An issue with this type of filtering is that we have only defined the frequency response at N points. what happens to the frequency response at frequencies that are not constrained?

The frequency response of the sequence w_n is given by (12). Taking a unit sampling interval (so that f is normalized to the Nyquist frequency) gives (when the Hermitian terms are collapsed into a cosine function)

$$W(f) = \frac{2}{N} \sum_{n=1}^{N/2-1} \left(\frac{\sin(\pi nM/N)}{\sin(\pi n/N)} \cos(2\pi nf) \right) + \frac{M}{N} + \frac{1}{N} \cos(2\pi nf) \tag{14}$$

where the last two terms are for $n = 0$ and $n = N/2$, respectively. (14) is plotted as a function of normalized frequency in Figures 7 and 8 for $N = 128$, $M = 31$ and for $N = 512$, $M = 127$.

We see that the frequency response oscillates wildly between the frequency sample points, even though it dutifully follows the ideal low pass specification exactly at the prescribed frequencies. The largest overshoots occur near the transition band. This type of behavior at the intermediate frequencies

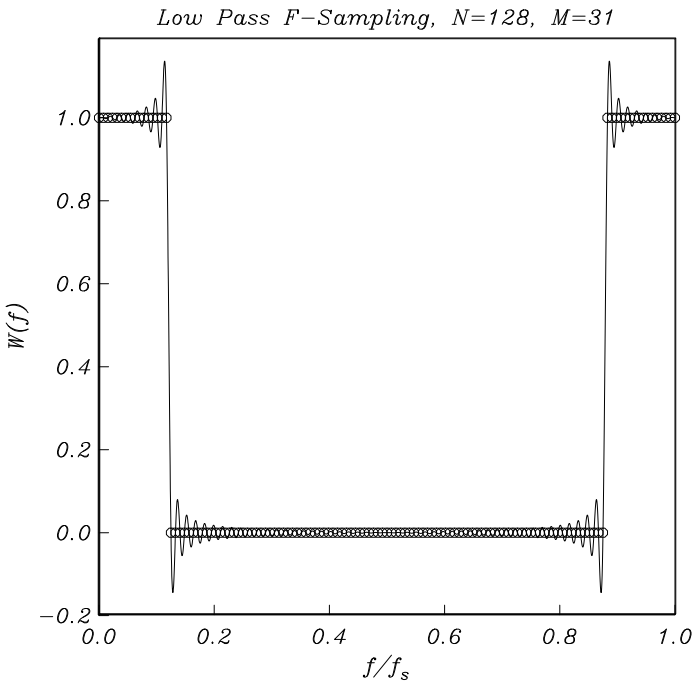

Figure 7: Frequency sampling frequency response in attempting to realize an ideal low pass filter; N=128

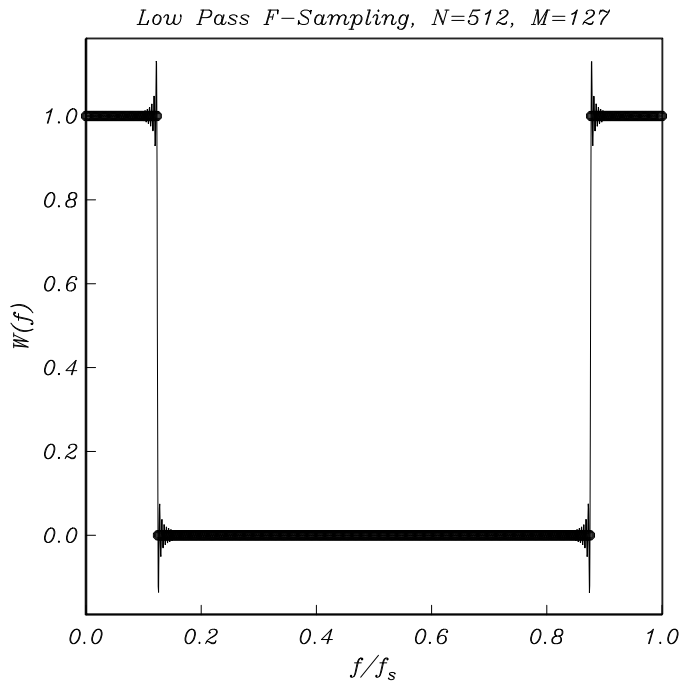

Figure 8: Another frequency sampling frequency response in attempting to realize an ideal low pass filter; N=512. Note that increasing the number of frequency specifications does not reduce the amplitude of the undesirable response ripple.

is called the *Gibbs phenomenon* and Figures 7 and 8 show that it has the unfortunate property that the percent overshoot does not decrease as N increases, although the width of the ripples does decrease as we squeeze them by stubbornly specifying more and more frequencies in our frequency sampling procedure.

If frequency sampling is really equivalent to direct manipulation of the FFT, then why didn't we notice any problems when we directly manipulated the FFT of the 20 second audio clip? In that case, the FFT had 882,000 frequencies, so the equivalent FIR filter would consist of a sequence of over 80,000 weights. Thus the ripple was confined to extremely narrow frequency bands near the cutoff at 2 Khz.

It turns out that one can in fact design much better behaved (smaller ripple) filters by using more sophisticated design methods. Although we won't get into in these notes, one very popular approach is the use of the Remez exchange algorithm to design FIR filters with specified maximum and minimum amplitudes in each of several frequency bands. Typically, a quite small filter (say 15 points) can adequately match the desired frequency response with very little ripple. The MATLAB command **firpm** can be used to implement this approach to designing a FIR filter.

More compact filter representations are possible if we allow *recursive* elements in our filters, where a component of the output is mixed in with the input. In addition, there are systems which have impulse responses with non-zero values at $t = \infty$ (e.g., integrators) which cannot be expressed at all in finite length FIR series. To fully appreciate this and to get a more general outlook on discrete realizations of continuous idealizations, we need to introduce some additional transforms which are closely related to the Fourier transform.

The Laplace Transform

The *One-Sided Laplace transform* is a generalized Fourier transform which explicitly allows for complex frequency, $s = \sigma + \imath\omega$, where σ and ω are real

$$\Phi(s) \equiv L[\phi(t)] = \int_0^\infty \phi(t) e^{-st}\, dt\ . \tag{15}$$

The convergence of the integral is very much an issue. Assuming that s is a positive real number or is complex with a positive real part, the function e^{-st} will go to 0 as t goes to infinity. For the integral to converge, $\phi(t)$ must not grow too quickly as t goes to infinity. If $|\phi(t)| \leq Ke^{bt}$, for some real constants K and b, and $\mathrm{Re}(s) > b$, then the integral will converge.

Note that an alternative *Two-Sided Laplace Transform* is used by some authors. In the two–sided Laplace transform, the integral is evaluated from minus infinity to plus infinity instead of from 0 to plus infinity. The two sided Laplace transform of $H(t)\phi(t)$ is precisely the one sided transform of $\phi(t)$.

If we make the substitution $s = 2\pi\imath f = \imath\omega$, we get

$$L[\phi(t)] = \int_0^\infty \phi(t) e^{-st}\, dt = \int_{-\infty}^\infty H(t)\phi(t) e^{-2\pi\imath f}\, dt = F[H(t)\phi(t)]\ . \tag{16}$$

The the Laplace transform of $\phi(t)$ is equivalent to the Fourier transform of $H(t)\phi(t)$. An alternative way to look at this is to say that as long as our signals are zero before time $t = 0$, the Fourier transform and Laplace transform are equivalent. This equivalence will be used frequently. In practice, we will often assume that signals begin after time $t = 0$, so that multiplying by $H(t)$ isn't necessary. Because of this relationship between the Laplace transform and the Fourier transform, many properties of the Laplace transform can be proved by using the already known properties of the Fourier transform.

For example, consider the action of a linear time invariant system on a signal $x(t)$, which we'll assume is zero for all t before $t = 0$. Let $\phi(t)$ be the impulse response of the system, and let $\Phi(f)$ be the Fourier transform of the impulse response. Assume further that the system is causal so that the output, $y(t)$, is zero before time $t = 0$. We know from our work with the Fourier transform that the Fourier transform of the output is $Y(f) = X(f)\Phi(f)$. Using our substitution $s = 2\pi\imath f$, we get that $Y(s) = X(s)\Phi(s)$. Here we've abused notation slightly by using $Y(s)$ for the Laplace transform of $y(t)$

and $Y(f)$ for the Fourier transform of $y(t)$. As long as all of the functions involved are zero before time 0, this works beautifully.

Recall that the Fourier transform of the derivative of $f(t)$ is given by $F[f'(t)] = 2\pi\imath f F[f(t)]$. Using the equivalence of the Fourier and Laplace transforms for functions which are zero before time $t = 0$, we would get that $L[f'(t)] = sL[f(t)]$. This is almost, but not quite correct. The problem occurs because $f(0)$ might be nonzero. Using the definition of the Laplace transform and integration by parts, it's easy to show that $L[f'(t)] = sL[f(t)] - f(0)$. In general,

$$L[f^{(n)}(t)] = s^n L[f(t)] - s^{n-1} f(0) - \ldots - s f^{(n-2)}(0) - f^{n-1}(0) \ . \tag{17}$$

Next, we consider a linear time invariant system that is governed by a nth order linear differential equation with constant coefficients.

$$a_n \frac{d^n y}{dt^n} + \ldots + a_1 \frac{dy}{dt} + a_0 y = b_m \frac{d^m x}{dt^m} + \ldots + b_1 \frac{dx}{dt} + b_0 x \ . \tag{18}$$

Many (but by no means all) LTI's can be written in this form. If we assume that $y(0)$, $y'(0)$, ..., $y^{(n-1)}(0) = 0$, then by the rule for the Laplace transform of a derivative,

$$(a_n s^n + \ldots a_1 s + a_0)\, Y(s) = (b_m s^m + \ldots b_1 s + b_0)\, X(s). \tag{19}$$

This can be rewritten as

$$\frac{Y(s)}{X(s)} = \Phi(s) = \frac{\sum_{j=0}^{m} b_j s^j}{\sum_{k=0}^{n} a_k s^k} \ . \tag{20}$$

As in the Fourier transfer function definition, the m roots of the numerator of (20) are called zeros, because $\Phi(s)$ is zero there, and the n roots of the denominator are called poles, because $\Phi(s)$ is infinite there. If the coefficients, a_i and b_i in (18) are real, then the poles and zeros are either real or form complex conjugate pairs. Note that at a pole frequency, s_p, an output can occur even for zero input. As we have seen before, a stable system has all of its poles on the left hand side of the complex plane (i.e., $\mathrm{Re}(s_p) < 0$), so that the pole frequencies have negative real parts

Another qualitative point is that closely-spaced poles and zeros cancel and can be ignored unless we are very close to them. Indeed for large frequencies all poles and zeros will start to cancel in this manner, so that $\Phi(s)$ asymptotically approaches

$$G(s) = \frac{b_m}{a_n}(s)^{m-n} \tag{21}$$

which changes by some multiple of about 6 dB for every doubling in frequency (6 dB per octave).

The Inverse Laplace Transform

For $t \geq 0$, the inverse Laplace transform is given by

$$\phi(t) = \frac{1}{2\pi\imath} \int_{\gamma-\imath\infty}^{\gamma+\imath\infty} \Phi(s) e^{st}\ ds \tag{22}$$

where γ is selected to be large enough so that the integral converges. if $|\phi(t)| \leq K e^{bt}$ for some constants K and b, then any value of γ larger than b will suffice.

In practice, this integral is usually evaluated by the technique of *contour integration*, using a contour c which includes the line $\mathrm{Re}(s) = \gamma$ and a semicircular arc to the left. See figure 9. If the integral over the semicircular arc is 0 (because $\Phi(s)$ goes to zero fast enough as the radius increases), then we can replace the integral over the line with an integral around the entire contour

$$\phi(t) = \frac{1}{2\pi\imath} \int_{c} \Phi(s) e^{st}\ ds \ . \tag{23}$$

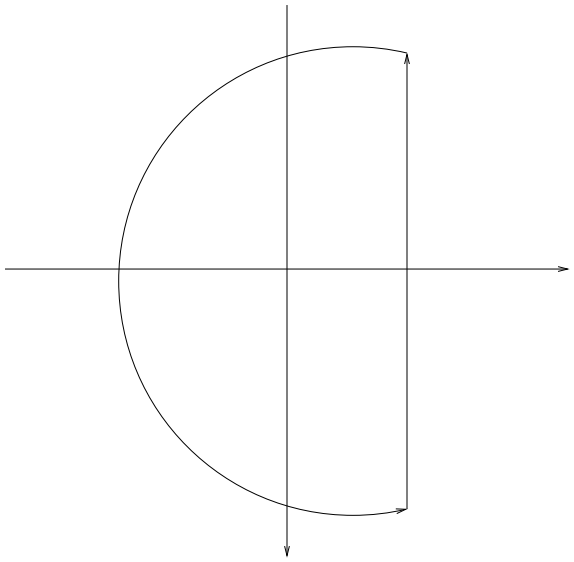

Figure 9: The contour used in computing the inverse Laplace transform.

Why bother with the contour integral? An important theorem of complex analysis states that if $f(z)$ has a finite number of poles, then the counter clockwise integral around a closed contour, which contains the poles of $f(z)$ can be evaluated by

$$\int_c f(z)dz = 2\pi\imath \sum_{\alpha=\text{poles of f(z)}} \text{residue}(\alpha) \tag{24}$$

where the residue at a pole $z = \alpha$ of order m is

$$\text{residue}(\alpha) = \frac{1}{(m-1)!} \lim_{z\to\alpha} \frac{d^{m-1}}{dz^{m-1}}(z-\alpha)^m f(z) \ . \tag{25}$$

Notice that the value of the contour integral depends only on the residues and the locations of the poles. Any contour which surrounds the same collection of poles will result in the same value for the integral! We can apply this formula to (23) to evaluate the inverse Laplace transform of $\Phi(s)$.

For example, suppose that our linear time invariant system is governed by the differential equation

$$2y'(t) + y(t) = x(t) \ . \tag{26}$$

We find that

$$\Phi(s) = \frac{1}{1+2s} \ . \tag{27}$$

In this case the system has a single pole of order 1 at s=-1/2. We will now find the impulse response by computing the inverse Laplace transform using (23) and contour integration. We can use the integration path from $-\imath\infty$ to $+\imath\infty$.

$$\phi(t) = \frac{1}{2\pi\imath} \int_{-\imath\infty}^{\imath\infty} \frac{1}{1+2s} e^{st} \ ds \ . \tag{28}$$

Our integrand goes to 0 very rapidly as our semicircular arc expands, so that in the limit, the integral over the semicircular arc is in fact 0. To show this, we use the substitution $s = Re^{\imath\theta}$, and take the limit as R goes to infinity

$$\int_{\pi/2}^{3\pi/2} \frac{1}{1+2Re^{\imath\theta}} e^{Re^{\imath\theta}t} Re^{i\theta} \ d\theta \ . \tag{29}$$

Table 1: Table of Laplace Transforms

$f(t)$	$F(s)$	where valid
1	$\frac{1}{s}$	$s > 0$
e^{at}	$\frac{1}{s-a}$	$s > a$
t^n	$\frac{n!}{s^{n+1}}$	$s > 0$
$\sin(at)$	$\frac{a}{s^2+a^2}$	$s > 0$
$\cos(at)$	$\frac{s}{s^2+a^2}$	$s > 0$
$e^{at}\sin(bt)$	$\frac{b}{(s-a)^2+b^2}$	$s > a$
$e^{at}\cos(bt)$	$\frac{s-a}{(s-a)^2+b^2}$	$s > a$
$H(t-c)$	$\frac{e^{-cs}}{s}$	$s > 0$
$\delta(t-c)$	e^{-cs}	
$f^{(n)}(t)$	$s^nF(s) - s^{n-1}f(0) - \ldots - f^{(n-1)}(0)$	
$e^{ct}f(t)$	$F(s-c)$	
$H(t-c)f(t-c)$	$e^{-cs}F(s)$	

In the limit as R goes to infinity, this integrand goes to 0 and the integral goes to 0, so it is safe in this case to replace (22) with (23). A very common mistake is to make the switch to the contour without checking that the integral over the semicircular arc is 0. In such cases, contour integration will give the wrong answer, so beware!

The residue at s=-1/2 is

$$\text{residue}(-1/2) = \lim_{z \to -1/2}(s+1/2)\frac{1}{1+2s}e^{st} = \frac{1}{2}e^{(-1/2)t} \tag{30}$$

The factors of $2\pi\imath$ in (23) and (24) cancel out, so

$$\phi(t) = \frac{1}{2}e^{(-1/2)t} \quad t \geq 0 \ . \tag{31}$$

Although any inverse Laplace transform can in theory be computed by this method, in practice it's usually easier to refer to a table of Laplace transforms or to use a symbolic computation package such as Maple to do the work. Table 1 gives some useful Laplace transforms.

The Chandler Wobble

As an example of a geophysical system transfer function with one complex pole in the s plane and a complex forcing and response, we next consider the *Chandler wobble* or *free nutation* response of the earth's spin axis, which changes due to some combination of mass shifts in the Earth due to oceanic or atmospheric circulation, glaciation, vegetation variations, snow or surface water accumulation, large earthquakes, mantle motions, core-mantle interactions, etc. Lately, it has been claimed that the most important processes, at least from 1985-1996, were atmospheric and oceanic processes, with the dominant mechanism being ocean-bottom pressure variations; Gross, 2000; GRL 27, p. 2329-2332). In a Cartesian grid is laid out at the north pole with an origin at the mean pole position (the axis of greatest moment of inertia), the spin axis at a given time can be specified as being at at (y_1, y_2) (Figure 10.)

If the forcing function, in this case, the migration of the Earth's principal axis of maximum rotational inertia due to mass movements, in the same coordinate system, is (x_1, x_2), the governing differential equations of motion are those of a body rotating slightly off from its maximum moment of inertia principal axis

$$\frac{\dot{y}_1}{\omega_c} + y_2 = x_2 \tag{32}$$

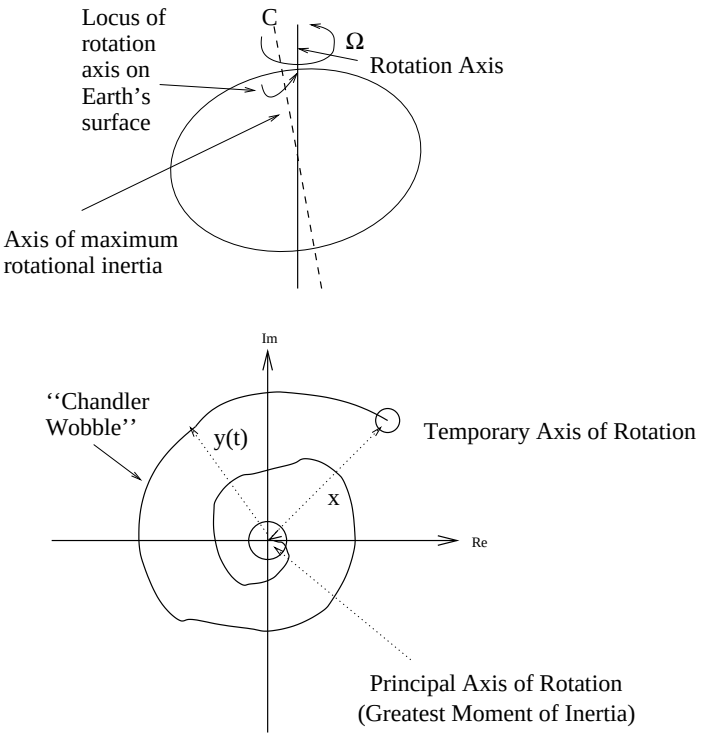

Figure 10: Geometry of the Chandler Wobble

.

$$\frac{-\dot{y}_2}{\omega_c} + y_1 = x_1 \tag{33}$$

where, for a rigid body,

$$\omega_c = \left(\frac{C-A}{C}\right)\Omega \tag{34}$$

where C and A are the polar and equatorial rotational moments of inertia and Ω is the spin rate. In the Earth, the components of the Chandler wobble have amplitudes of tens of meters, and are thus readily detectable using astronomical or other techniques. The ideal rigid body frequency (34) for a solid Earth is about 305 days, $(C-A)/C \approx 1/305.51$) but the observed decay constant is significantly longer (about 430 days) due to the Earth not being a perfectly elastic body.

We can jointly consider the two equations (32 and 33) by defining the complex quantities

$$x = x_1 + \imath x_2 \tag{35}$$

$$y = y_1 + \imath y_2 \tag{36}$$

to obtain

$$i\frac{\dot{y}}{\omega_c} + y = x \ . \tag{37}$$

Taking the Laplace transform of both sides gives

$$Y(s)\left(\frac{is}{\omega_c} + 1\right) = X(s) \tag{38}$$

so that the transfer function is

$$\frac{Y(s)}{X(s)} = \frac{\omega_c}{is + \omega_c} \tag{39}$$

which has a single pole at $s = \imath\omega_c$ (as $y(t)$ and $x(t)$ are complex valued, there is no complex conjugate pole at $s = -\imath\omega_c$ in this case). Physically, this means that the locus of the rotational axis on the earth's surface will indefinitely precess from west to east, once the system is excited. This asymmetry arises from the gyroscopic nature of the system. Dissipation in the earth (the principal cause or causes for the damping of the Chandler wobble are, again, controversial) can be accommodated by making ω_c complex

$$\omega_c = \frac{2\pi}{T_c}\left(1 + \frac{i}{2Q_c}\right) = \frac{\pi}{T_c}\left(2 + \frac{i}{Q_c}\right) \tag{40}$$

where Q_c is the quality factor (see Chapter 2) of the system and T_c is the natural frequency. The pole of the system response (39) then becomes

$$p = \frac{\pi}{T_c}\left(2\imath - \frac{1}{Q_c}\right) \tag{41}$$

which has a negative real part and hence describes a decaying sinusoidal motion. The impulse response is thus

$$\phi(t) = L^{-1}[Y(S)/X(s)] = \frac{1}{2\pi\imath}\int_c \frac{-i\omega_c}{s - \imath\omega_c} e^{st}\ ds \tag{42}$$

and may be found via contour integration and the residue theorem to be the complex sinusoid

$$= -i\omega_c e^{\imath\omega_c t} \tag{43}$$

where the phase of (43) signifies the phase relationship between the complex forcing and response functions, $x(t)$ and $y(t)$. In the problem of the Chandler wobble, the interesting physics are tied up in the measurement of T_c (which is around 430 days) and of the forcing function, $x(t)$. The wobble is continuously excited by mass movements in the solid Earth, oceans, and the atmosphere which change its

moments of inertia and averages about 0.14 seconds of arc (6.8×10^{-7} rad), which corresponds to a root mean square (rms) polar discrepancy of about 4.5 m).

It is worth noting that in some interesting situations, such as the excitation of the normal modes of the earth, we can examine the response and estimate the pole positions without worrying about the exact spectrum of the excitation function. This is because the excitation function is broad-band relative to our observational bandwidth and thus, on average, excites many frequencies.

The Z Transform

Just as the discrete Fourier transform as an alternative to the Fourier transform to analyze discretized signals, the *Z transform* is the discrete analog of the continuous Laplace transform.

Consider a complex variable z and define the *z transform* of a sequence x_n as

$$X(z) = Z[x_n] = \sum_{n=-\infty}^{\infty} x_n z^{-n} \ . \tag{44}$$

A warning: a few authors use z^n rather than z^{-n} in their z transform definitions. Again, this is mere convention, akin to choosing $e^{-\imath 2\pi ft}$ or $e^{\imath 2\pi ft}$ in the Fourier transform definitions, but can lead to misinterpretations. Also, some authors utilize a one-sided version of the z transform in which the sum runs from $n = 0$ to ∞. As a rule, always check to see what conventions a given author is using.

The general relationship between the z transform and the Fourier domain is shown in Figure 11. Multiplication by z^{-l} in the z domain is equivalent to a time delay (rightward shift) of l samples and multiplication by z^l is equivalent to a time advance (leftward shift) of l samples; the exponent of z in each term is a place holder to designate where a particular value fits into the time series. The time shift theorem for the z transform is thus

$$Z[x_{n-i}] = \sum_{n=-\infty}^{\infty} x_{n-i} z^{-n} = z^{-i} \sum_{n=-\infty}^{\infty} x_{n-i} z^{-(n-i)} = z^{-i} \sum_{m=-\infty}^{\infty} x_m z^{-m} \tag{45}$$

or

$$Z[x_{n-i}] = z^{-i} X(z) \ . \tag{46}$$

Finding closed-form expressions for the z transforms of common time sequences relies on the specific properties of each series, but as an example, consider an exponential series

$$x_n = \begin{cases} c^n & n \geq 0 \\ 0 & n < 0 \end{cases} \ . \tag{47}$$

In this case, we can use the standard procedure for collapsing geometric series to obtain

$$Z[x_n] = \sum_{n=0}^{\infty} c^n z^{-n} = \frac{1}{1 - cz^{-1}} = \frac{z}{z - c} \tag{48}$$

when $|z| > |c|$.

The case when $c = 1$ gives the z transform of the discrete unit step function

$$Z[H_n] = \frac{z}{z - 1} \ . \tag{49}$$

The convolution theorem relationship for the z transform is particularly easy to see. For a particular m, the terms in the product

$$W(z) = X(z)Y(z) = \sum_{m=-\infty}^{\infty} w_m z^{-m} \tag{50}$$

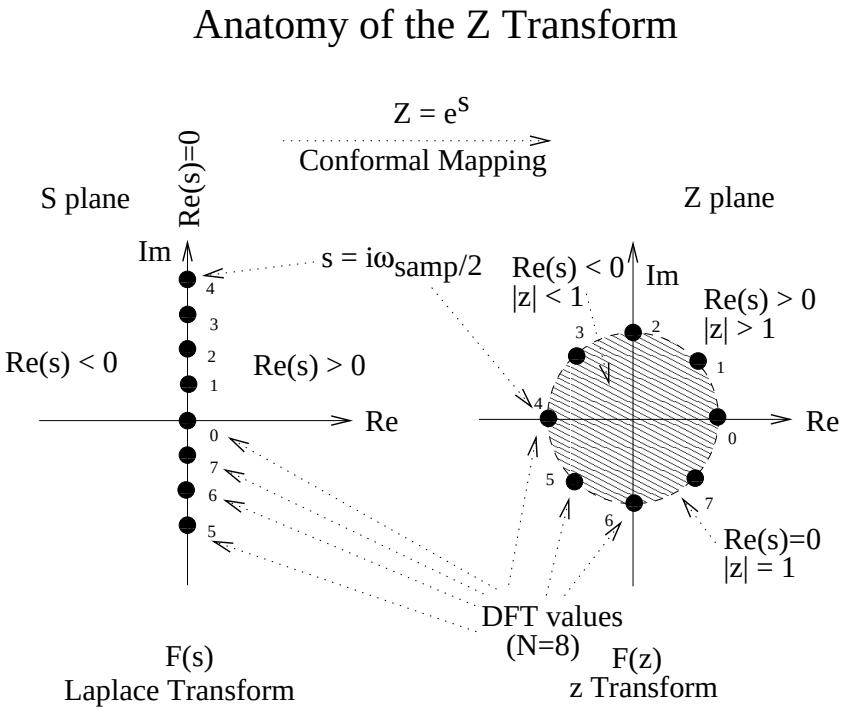

Figure 11: The Z Transform, and its relationship to the Fourier domain.

can be seen from polynomial multiplication of $X(z)$ and $Y(z)$ to be

$$x_n z^{-n} y_{m-n} z^{-(m-n)} \ . \tag{51}$$

It follows that

$$W_m = \sum_{n=-\infty}^{\infty} x_n y_{m-n} = \sum_{n=-\infty}^{\infty} y_n x_{m-n} \tag{52}$$

which is just the discrete (linear) convolution of x_n and y_n.

To evaluate the inverse z transform, we again make use of the technique of contour integration. By the residue theorem, the counterclockwise contour integration around a pole of degree $-k+1$ is

$$\frac{1}{2\pi\imath} \int_c \frac{dz}{z^{-k+1}} = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \ . \tag{53}$$

The inverse z transform is thus

$$x_n = \frac{1}{2\pi\imath} \int_c X(z) z^{n-1} \, dz \tag{54}$$

where c is a counterclockwise closed contour selected so that the integral will converge.

To discuss issues of convergence, we express the z transform as a function of complex z in a polar coordinate system $z = Re^{\imath 2\pi f}$, where R is a real number. The z transform is then

$$X(Re^{\imath 2\pi f}) = \sum_{n=-\infty}^{\infty} x_n \cdot (Re^{\imath 2\pi f})^{-n} = \sum_{n=-\infty}^{\infty} x_n R^{-n} e^{-\imath 2\pi f n} \tag{55}$$

for $R = 1$, z lies on the unit circle in the complex plane, and the z transform is equivalent to the Fourier series of the sequence x_n. The infinite series defined by the z transform (44) converges when

$$\sum_{n=-\infty}^{\infty} |x_n z^{-n}| = \sum_{n=-\infty}^{\infty} |x_n R^{-n}| < \infty \ . \tag{56}$$

As the z transform contains terms for both positive and negative n, the general situation is that the sequence converges in some annular region, where R is not so large that the negative n part of the sequence diverges, but not so small that the positive n part of the sequence diverges, i.e.,

$$R_{h-} < |z| < R_{h+} \tag{57}$$

where R_{h-} and R_{h+} designate the inner and outer radii of the annulus, respectively.

Given the inverse z transform (54) we can now examine what happens in the z domain when we multiply two time series together

$$w_n = x_n \cdot y_n \ . \tag{58}$$

Taking the z transform of both sides yields

$$W(z) = \sum_{n=-\infty}^{\infty} x_n y_n z^{-n} = \frac{1}{2\pi\imath} \sum_{n=-\infty}^{\infty} x_n \int_c Y(v) \left(\frac{v}{z}\right)^n \frac{dv}{v} \tag{59}$$

$$= \frac{1}{2\pi\imath} \int_c Y(v) \left\{ \sum_{n=-\infty}^{\infty} x_n \left(\frac{v}{z}\right)^n \right\} \frac{dv}{v} = \frac{1}{2\pi\imath} \int_c Y(v) X(z/v) \frac{dv}{v} \tag{60}$$

setting $z = Re^{\imath\phi}$ and letting the contour, c be the circular path $v = \rho e^{\imath\theta}$ gives

$$W(Re^{\imath\phi}) = \frac{1}{2\pi} \int_0^{2\pi} Y(\rho e^{\imath\theta}) X(\frac{R}{\rho} e^{\imath(\phi-\theta)}) \, d\theta \tag{61}$$

This is a generalized case of the convolution relationship for the Fourier transform. To see this, evaluate (61) on the unit circle, where $R = \rho = 1$, $\phi = 2\pi f$, and $\theta = 2\pi f'$ to obtain

$$W(f) = \int_0^1 Y(e^{2\pi\imath f'})X(e^{2\pi\imath(f-f')})\ df' \tag{62}$$

which is a circular convolution! In fact, the DFT of a sequence,

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\imath 2\pi nk/N} \tag{63}$$

is just the z transform evaluated at N equiangular points around the unit circle, i.e.,

$$X_k = X(z = e^{\imath 2\pi k/N}) \quad k = 0, 1, 2, ..., N-1\ . \tag{64}$$

How can we relate the discrete-time z transform to the continuous-time Laplace transform? How we do this is fundamental to designing discrete systems which mimic continuous ones.

Consider the Laplace transform of a sampled version of a continuous function, $x(t)$, which is assumed to be 0 before $t = 0$:

$$\int_0^\infty x(t)\mathrm{III}(t)e^{-st}\ dt = \int_0^\infty \sum_{n=0}^{\infty} x(n)\delta(t-n)e^{-st}\ dt \tag{65}$$

$$= \sum_{n=0}^{\infty} x(n) \int_0^\infty \delta(t-n)e^{-st}\ dt = \sum_{n=0}^{\infty} x(n)e^{-sn} = \sum_{n=0}^{\infty} x_n z^{-n}\ . \tag{66}$$

Notice that if we let $z = e^s$, then $z^{-n} = e^{-sn}$. Thus the mapping between z and s is simply $z = e^s$! The imaginary axis in the s-plane corresponds to the unit circle in the z-plane. Similarly, the right half s-plane maps outside of the z-plane unit circle and the left half of the s-plane maps inside of the z-plane unit circle. Note that this mapping is multivalued, with a periodicity of 2π in the s-plane imaginary dimension, i.e., all of the points $s = R(2\pi\imath f + 2\pi\imath m)$ map to the same point on the unit circle in the z plane, $z = e^{\imath R 2\pi f}$. This is a general picture of aliasing.

IIR filtering

We will now consider recursive filters which effectively take weighted averages of the input and previous output samples, and will utilize the z transform to analyze their responses. Many practical filters utilize feedback because it is provides significant efficiencies in representing filters. Most MATLAB filter design algorithms return IIR coefficients that are subsequently implemented with commands such as *filter or* filtfilt. Examples include *butter, cheby1, cheby2, ellip* and many others that seek to match some mathematically idealized analog response with a specified number of poles (for example, Butterworth filters are ones designed to have optimally flat frequency response in the passband (no ripple effects). as well as parametric filter design functions, such as *prony*, that will design coefficients corresponding to arbitrary desired response characteristics.

A general recursive digital filter equation, defined by its coefficents a_k and b_m can be written in the form

$$\sum_{k=0}^{K} a_k y_{n-k} = \sum_{m=0}^{M} b_m x_{n-m} \tag{67}$$

where y is the output sequence and x is the input sequence. This can be rewritten in terms of the current (n^{th}) output sample y_n as

$$y_n = \frac{\sum_{m=0}^{M} b_m x_{n-m} - \sum_{k=1}^{K} a_k y_{n-k}}{a_0}\ . \tag{68}$$

As equation (68) shows, recursive filters of this type, because the output always depends on prior inputs and outputs, are always causal. In this form, the filter is also trivial to program in the time domain. Given the filter coefficients b, a, and the input sequence x, the MATLAB command **filter** can be used to compute y.

We can compute the z transform of the impulse response of such a filter by multiplying equation (67) by z^{-n}, and summing up terms from $n = -\infty$ to ∞.

$$\sum_{n=-\infty}^{\infty} \sum_{k=0}^{K} a_k y_{n-k} z^{-n} = \sum_{n=-\infty}^{\infty} \sum_{m=0}^{M} b_m x_{n-m} z^{-n} \tag{69}$$

$$Y(z) \sum_{k=0}^{K} a_k z^{-k} = X(z) \sum_{m=0}^{M} b_m z^{-m}. \tag{70}$$

Thus

$$\Phi(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{m=0}^{M} b_m z^{-m}}{\sum_{k=0}^{K} a_k z^{-k}} \ . \tag{71}$$

Note the similarities between (67), (71) and (18), (20). This arises because delays map into powers of z^{-1} in the z transform, just as differentiation maps into powers of s in the Laplace transform. MATLAB provides a command *freqz* that calculates the N-point complex frequency response of such a filter, given the filter coefficients.

When is a recursive filter stable? To be stable, it must have that the impulse response sequence ϕ_n goes to 0 as n goes to infinity. Recall that if all of the poles of $\Phi(s)$ are in the left half plane (or have negative real parts), then an LTI is stable. Examining the conformal mapping ($z = e^s$) between the z and Laplace domains, we see that all of the poles of $\Phi(z)$ for a stable filter must correspondingly be contained within the unit circleh. Thus the stability condition for a recursive filter of the form (67) is that the poles of $\Phi(z)$ must lie within the unit circle.

The Impulse Invariance Method

Consider the simple continuous-time system defined by

$$\tau \frac{dy}{dt} + y = x \tag{72}$$

where τ is real. Solving for the transfer function using the Laplace transform yields

$$\frac{Y(s)}{X(s)} = \frac{1}{1 + \tau s} \tag{73}$$

which has a pole at $s = -1/\tau$ and is stable for $\tau > 0$. The frequency response is found by letting $s = 2\pi\imath f$

$$\frac{Y(f)}{X(f)} = \frac{1}{1 + \imath 2\pi\tau f} \tag{74}$$

which is 1 at zero frequency, and becomes smaller as f increases. This system is thus a low pass filter. The impulse response is

$$L^{-1}[Y(s)/X(s)] = \frac{1}{2\pi\imath} \int_c \frac{e^{st}\ ds}{1 + \tau s} = H(t)\tau^{-1} e^{st}|_{s=\tau^{-1}} = \frac{H(t)}{\tau} e^{-t/\tau} \ . \tag{75}$$

and the step response is thus

$$H(t) * H(t)\tau^{-1} e^{-t/\tau} = H(t)(1 - e^{-t/\tau}) \ . \tag{76}$$

This response is nonzero for all non-negative $t < \infty$, and thus cannot be modeled at large t with any FIR filter, unless we are willing to use an arbitrarily large number of filter terms. However, a very simple recursive filter can come much closer to mimicking the desired response.

In the *impulse invariance method*, we pick a recursive filter so that the impulse response of the digital filter matches the desired impulse response of the continuous filter.

Consider the step response of the discrete system defined by

$$y_n - \alpha y_{n-1} = x_n(1-\alpha) \tag{77}$$

For a step sequence input, we get

$$\begin{array}{c} y_0 = 1-\alpha \\ y_1 = \alpha(1-\alpha) + (1-\alpha) \\ y_2 = \alpha^2(1-\alpha) + \alpha(1-\alpha) + (1-\alpha) \end{array} \tag{78}$$

and so forth. In general,

$$y_n = (1-\alpha)\sum_{k=0}^{n} \alpha^k = 1 - \alpha^{n+1} = 1 - e^{(n+1)\ln(\alpha)} \tag{79}$$

which has the form of a sampled version of the desired continuous response (76). (77) is thus an IIR filter realization of (76).

To express this IIR filter in the z domain, recall the z^{-1} is the z transform of a one sample delay. We can thus map the x_n and y_n in (77) to the z domain by multiplying each term by z^{-n} and summing over all n

$$\sum_{n=-\infty}^{\infty} y_n z^{-n} - \alpha \sum_{n=-\infty}^{\infty} y_{n-1} z^{-n} = (1-\alpha) \sum_{n=-\infty}^{\infty} x_n z^{-n} \tag{80}$$

which can be factored as

$$Y(z)(1-\alpha z^{-1}) = (1-\alpha)X(z) \tag{81}$$

to obtain the z transfer function

$$\frac{Y(z)}{X(z)} = \frac{1-\alpha}{1-\alpha z^{-1}} \ . \tag{82}$$

To evaluate the frequency response of (82), we evaluate the transfer function on the unit circle, or at $z = e^{2\pi\imath f/f_s}$, where f_s is the sampling frequency, and the Nyquist interval is the range of frequencies is thus $-f_s/2 \le f \le f_s/2$

$$\Phi(z = e^{2\pi\imath f/f_s}) = \frac{1-\alpha}{1-\alpha e^{-\imath 2\pi f/f_s}} \ . \tag{83}$$

The corresponding frequency response of the continuous system is given by (74). Both the continuous and discrete response functions are plotted in Figure 13, using $\tau = 10$ and the corresponding value for α, $\alpha = 1 - 1/\tau$, so that values for the discrete and continuous time functions agree at $n = 0$ and $t = 0$, respectively.

The major discrepancy in the frequency domain is that a discrete system has a periodic frequency response, and so, for this filter, must return to a value of 1 at $f = f_s$, while the continuous system continues to approach zero response with increasing frequency at a rate of about 6 dB per octave.

The Bilinear Transformation

Consider writing the discrete y sequence at a sampling interval of Δ as

$$y(n\Delta) = \int_{\Delta(n-1)}^{n\Delta} \dot{y}(u)du + y(\Delta[n-1]) \ . \tag{84}$$

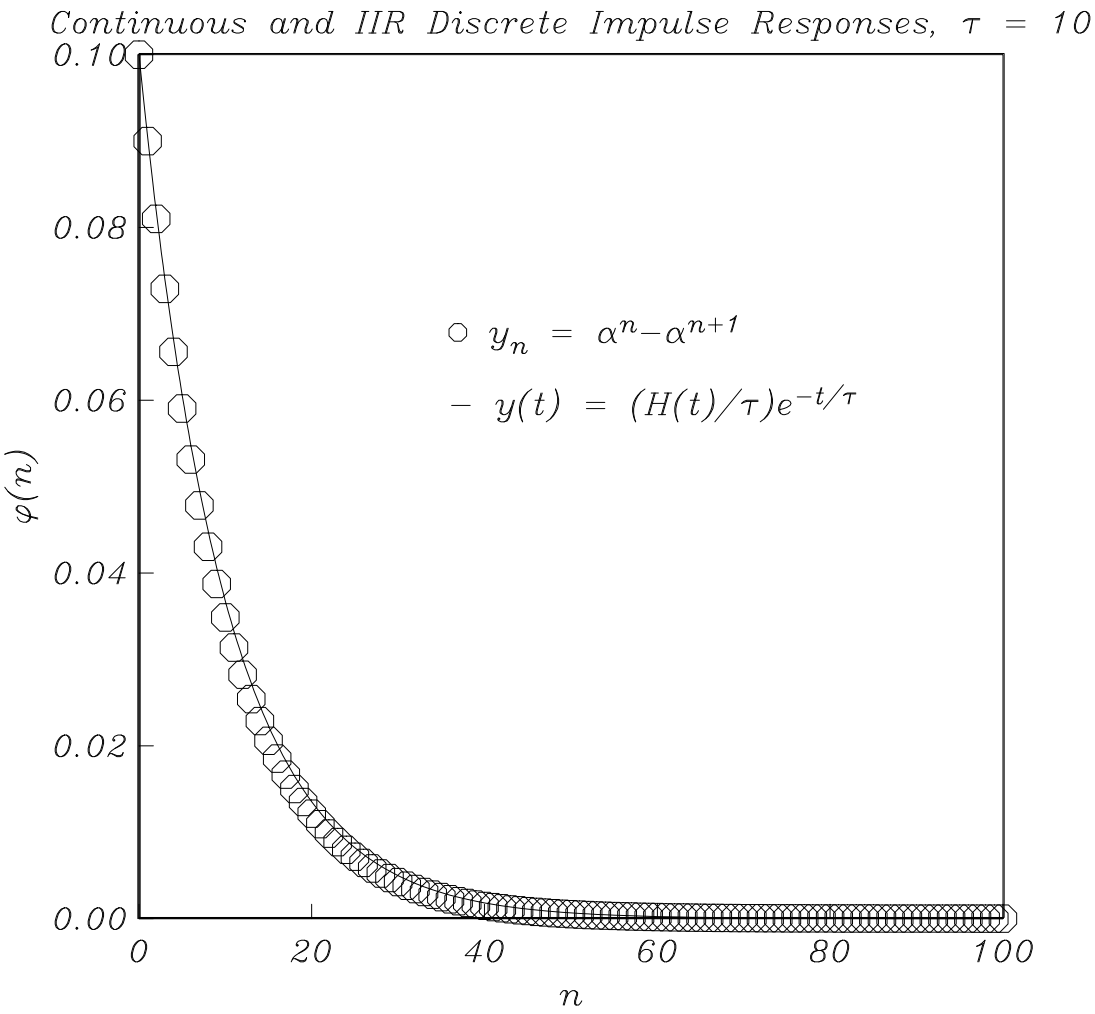

Figure 12: Impulse invariance discrete realization compared to a target continuous response in the time domain.

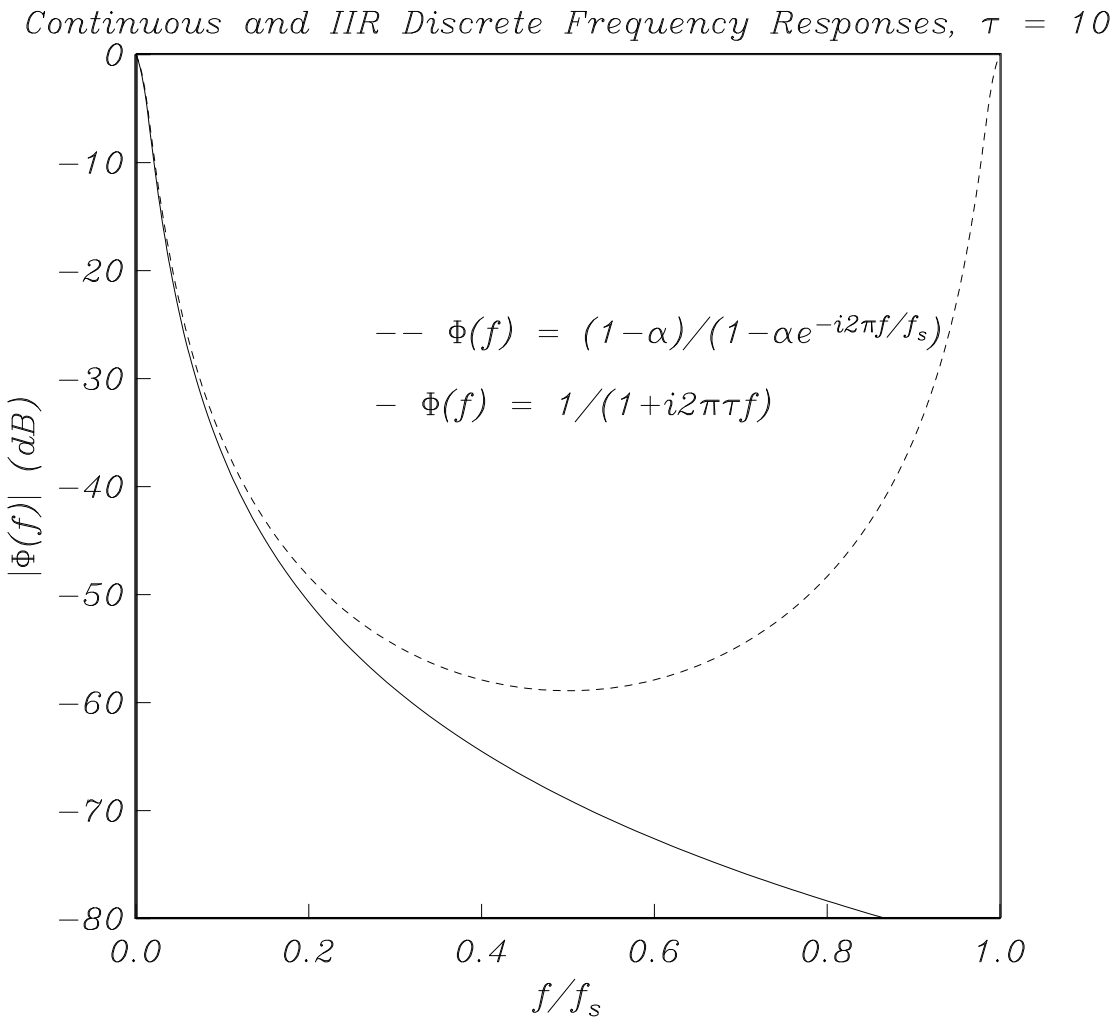

Figure 13: Impulse invariance discrete realization compared to a target continuous response in the frequency domain.

Approximating the integral in (84) by the trapezoidal rule then gives

$$y(n\Delta) \approx \frac{\Delta}{2}\left[\dot{y}(\Delta[n-1]) + \dot{y}(n\Delta)\right] + y(\Delta[n-1]) \ . \tag{85}$$

which has the discrete time counterpart

$$y_n = \frac{\Delta}{2}\left[\dot{y}_{n-1} + \dot{y}_n\right] + y_{n-1} \tag{86}$$

The original (rewritten) differential equation (72) is

$$\dot{y}_n = \frac{1}{\tau}(x_n - y_n) \tag{87}$$

or equivalently

$$\dot{y}_{n-1} = \frac{1}{\tau}(x_{n-1} - y_{n-1}) \ . \tag{88}$$

Thus, we can eliminate the time derivatives by evaluating $\dot{y} + \dot{y}_{n-1}$ from the sum of (87) and (88) and substituting the result into (86). This yields

$$y_n = \frac{\Delta}{2\tau}\left[x_{n-1} - y_{n-1} + x_n - y_n\right] + y_{n-1} \tag{89}$$

or

$$y_n(1 + \frac{\Delta}{2\tau}) - y_{n-1}(1 - \frac{\Delta}{2\tau}) = \frac{\Delta}{2\tau}(x_n + x_{n-1}) \ . \tag{90}$$

(88) has the z transform

$$\Phi(z) = \frac{Y(z)}{X(z)} = \frac{\frac{\Delta}{2\tau}(1 + z^{-1})}{(1 + \frac{\Delta}{2\tau}) - (1 - \frac{\Delta}{2\tau})z^{-1}} \tag{91}$$

$$= \frac{(1 + z^{-1})}{(\frac{2\tau}{\Delta} + 1) - (\frac{2\tau}{\Delta} - 1)z^{-1}} \tag{92}$$

$$= \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right)\left(\frac{1 - z^{-1}}{1 + z^{-1}}\right)} \ . \tag{93}$$

Evaluating the frequency response of (93) by taking $z = e^{2\pi\imath f/f_s}$, we get

$$\Phi(z = e^{i2\pi f/f_s}) = \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right)\left(\frac{1 - e^{-\imath 2\pi f/f_s}}{1 + e^{-\imath 2\pi f/f_s}}\right)} \tag{94}$$

$$= \frac{1}{1 + \left(\frac{2\tau}{\Delta}\right)\imath \tan \pi f/f_s} \ . \tag{95}$$

(95) is thus the response of the continuous system (73) with the substitution

$$s = \frac{2i}{\Delta} \tan \pi f/f_s \ . \tag{96}$$

(95) is plotted along with the continuous response in Figure 15).

Recalling that the continuous frequency response (74) is just (73), evaluated at $s = \imath 2\pi f$, we can see that the frequency mapping between (95) and 74) is just

$$2\pi f_c = \frac{2}{\Delta} \tan \pi f_d/f_s \tag{97}$$

where f_d is the digital frequency and f_c is the continuous frequency. The continuous system frequency response tends to zero as $f_c \to \infty$. The bilinear z transform frequency response, on the other hand tends to zero where

$$\frac{\pi f_d}{f_s} = \frac{\pi}{2}(2m+1) \tag{98}$$

or

$$f_d = \frac{f_s}{2}(2m+1) \tag{99}$$

where m is an integer, which is just at odd multiples of the Nyquist frequency, $f_N = f_s/2$. The bilinear z-transform substitution (96) thus maps the semi-infinite frequency interval of the continuous system $(-\infty,\ \infty)$ into the Nyquist interval $[-f_N,\ f_N]$. To obtain the digital transfer function, $\Phi_d(z)$, from a given analog filter transfer function, $\Phi_a(s)$, we simply substitute

$$s = \frac{2}{\Delta}\frac{1-z^{-1}}{1+z^{-1}}\ . \tag{100}$$

An alternative explanation of the bilinear transform approach is that if $z = e^s$, and

$$\hat{s} = 2\frac{1-\frac{1}{z}}{1+\frac{1}{z}} \tag{101}$$

then

$$\hat{s} = 2\frac{1-e^{-s}}{1+e^{-s}} = 2\frac{1-e^{-2\pi\imath f}}{1+e^{-2\pi\imath f}} = 2\imath\tan(\pi f)\ . \tag{102}$$

For small frequencies f, $\tan \pi f \approx \pi f$. Thus

$$\hat{s} \approx 2\pi\imath f\ . \tag{103}$$

That is, $\hat{s}$ is an approximation to s. By using $\hat{s}$ in place of s in the transfer function, we obtain a transfer function that can be expressed as a rational function of $1/z$. MATLAB has a function *bilinear* that implements this algorithm with an additional feature to exactly match the desired response up to some specified frequency.

Of course, we can never match the analog response with a digital system because of aliasing, but we can match some desirable characteristic of the analog system (e.g., ripple height, corner frequency, etc.) within the Nyquist interval. In general, we can do this far more compactly with an IIR filter, but as always, there is a price, in this case IIR filters will have more complicated (non-linear) phase characteristics than FIR filters. We can see this directly by noting that the z transform of an FIR filter is just a polynomial in z^{-1}, while the z transform of a recursive filter is a ratio of two polynomials (a rational function) of z^{-1}.

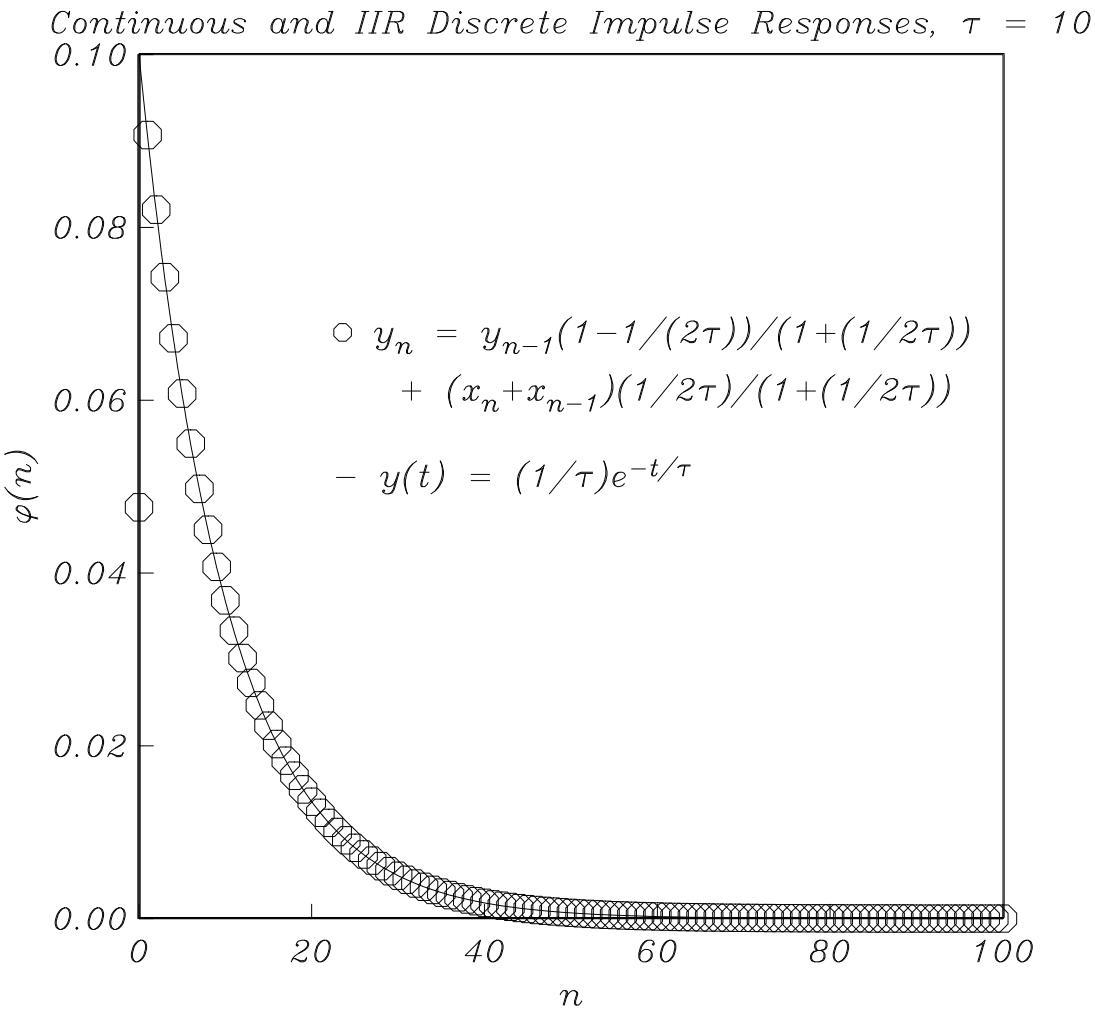

Figure 14: Bilinear z transform discrete realization response compared to a target continuous response in the time domain.

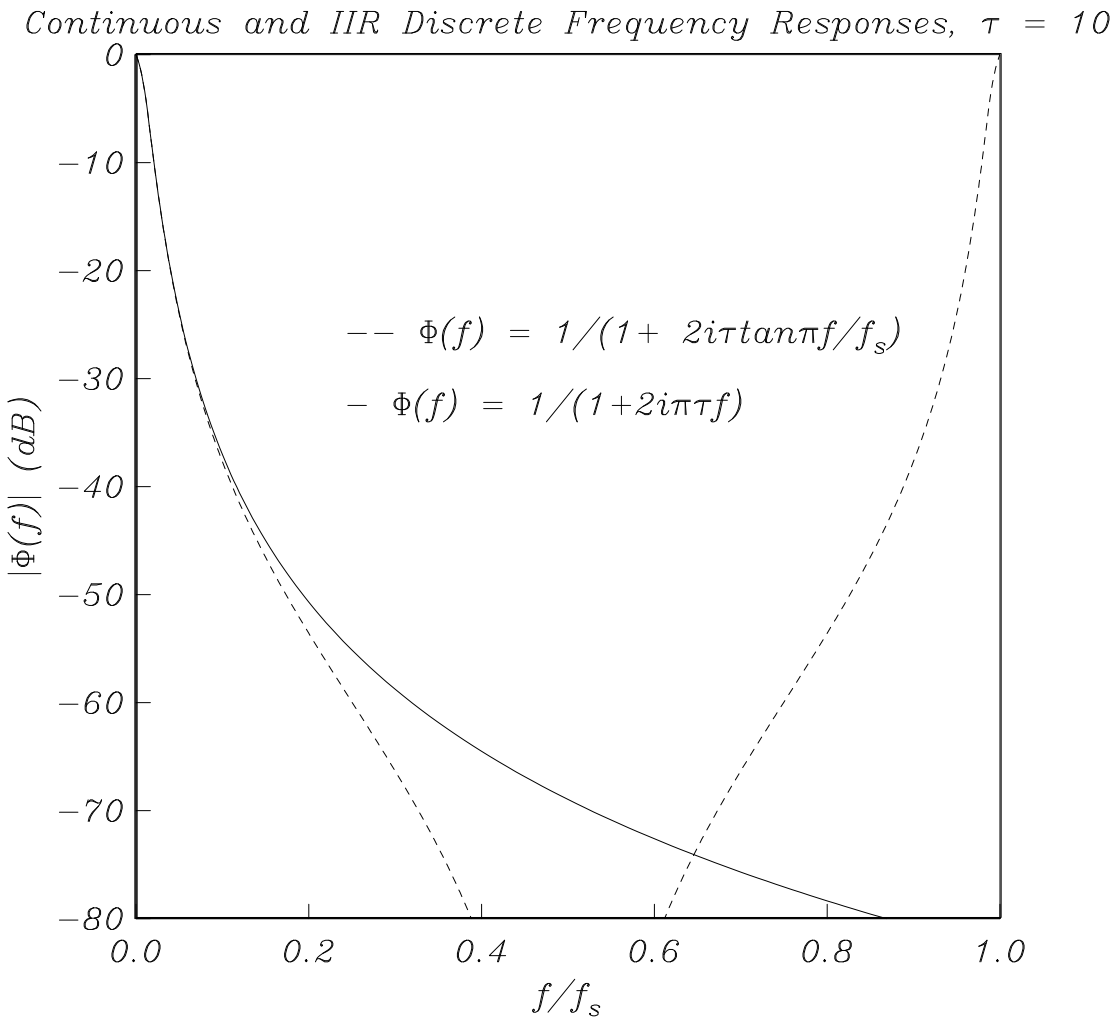

Figure 15: Bilinear z transform discrete realization response compared to a target continuous response in the frequency domain.