Tuesday 2008-12-16 Dan Ellis <dpwe@ee.columbia.edu>

---

1. The question doesn't specify whether we are looking for an FIR or IIR filter, but ripples in pass band and monotonic in the stop band just screams Chebyshev I IIR.

   (a) To find the order, we need to use the design equation from slide 20 of Topic 8, meaning we need A, ϵ, Ω_p, and Ω_s. 1 dB of ripple means $10\log_{10}\frac{1}{1+\epsilon^2} = -1$ so $\epsilon = \sqrt{10^{0.1}-1} = 0.509$. Minimum stopband attenuation of 30 dB means $20\log_{10} A = 30$ so $A = 10^{3/2} = 31.6$. To find Ω_p and Ω_s, however, we will have to map the design band edges to normalized frequency (rad/samp), then perform our prewarping:

$$\begin{aligned}
\omega_p &= 2\pi\frac{1\,\text{kHz}}{20\,\text{kHz}} = \pi/10\,\text{rad/samp} \\
\omega_s &= 2\pi\frac{2\,\text{kHz}}{20\,\text{kHz}} = \pi/5\,\text{rad/samp} \\
\Omega_p &= \tan(\omega_p/2) = 0.158\,\text{rad/sec} \\
\Omega_s &= \tan(\omega_s/2) = 0.325\,\text{rad/sec}
\end{aligned} \tag{1}$$

   Hence the Chebyshev I filter order,

$$\begin{aligned}
N \geq & \frac{\cosh^{-1}\left(\frac{\sqrt{A^2-1}}{\epsilon}\right)}{\cosh^{-1}\left(\frac{\Omega_s}{\Omega_p}\right)} \\
= & \frac{\cosh^{-1}(62.1)}{\cosh^{-1}(2.05)} = 3.58
\end{aligned} \tag{2}$$

   Thus we must use order $N = 4$.

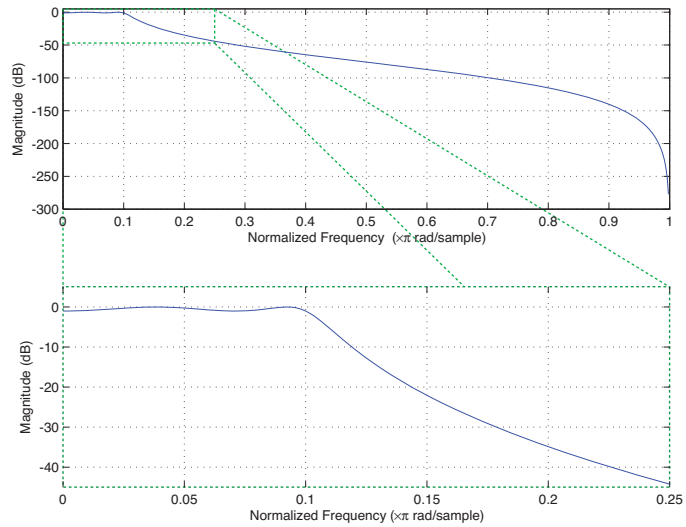
   A Chebyshev I filter will have only poles in the s-domain, so a 4th order filter will have 4 poles. On mapping to the z-domain via the bilinear transform, however, we will introduce zeros, since:

$$\begin{aligned}
G(s) &= \frac{K}{\prod_{i=1}^{N}(s-\lambda_i)} \\
G(z) &= G(s)\big|_{s=\frac{1-z^{-1}}{1+z^{-1}}} = \frac{K}{\prod_{i=1}^{N}(\frac{1-z^{-1}}{1+z^{-1}}-\lambda_i)} \\
&= \frac{K(1+z^{-1})^N}{\prod_{i=1}^{N}(1-z^{-1}-\lambda_i(1+z^{-1}))} \\
&= \frac{K(1+z^{-1})^N}{\prod_{i=1}^{N}((1-\lambda_i)-z^{-1}(1+\lambda_i))}
\end{aligned} \tag{3}$$

   which has N poles at $z = (1+\lambda_i)/(1-\lambda_i)$ (as expected), but also N zeros at $z = -1$, reflecting the fact that the gain of the discrete-time filter goes to zero at the Nyquist rate, which is the mapping of infinitely high frequency in the continuous domain.

(b) The magnitude response will simply be a low-pass response meeting the criteria. Here it is from Matlab:

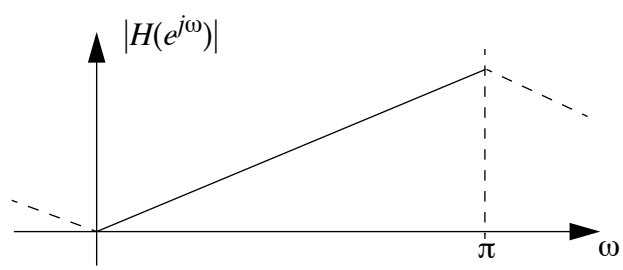
```
>> [b,a] = cheby1(4,1.0,0.1);
>> freqz(b,a)
```



Note the following features:

- There are ripples in the pass band and monotonic decline above the passband
- The ripples are 1dB down in the passband, and touch -1dB at $\omega = 0.1\pi$ (the passband edge).
- At $\omega = 0.2\pi$ (the stopband edge), the gain is significantly below the required -30dB, reflecting the margin between 'required' order of 3.58 and actual order $N = 4$.
- The gain heads for $-\infty$ dB at $\omega = \pi$.
- There are exactly two maxima in the passband ripples, corresponding to the nearest approach to the two positive-frequency poles. At $\omega = 0$, we are actually at a magnitude minimum (-1 dB) to obtain symmetry between positive and negative frequencies.

2. (a) From basic calculus, we know that $\frac{d}{dt}e^{j\Omega t} = j\Omega e^{j\Omega t}$. Thus we see that differentiating a complex sinusoid gives a complex sinusoid of the same frequency, but with its magnitude scaled by a constant proportional to the frequency Ω, and with a $\pi/2$ rad phase advance. Thus the ideal discrete differentiator also has a gain that increases linearly from 0 at $\omega = 0$ to a maximum at the Nyquist limit $\omega = \pi$ (and then, by the required symmetry and periodicity, ramps back down again), e.g.



(b) Because of the $\pi/2$ phase shift, an antisymmetric linear-phase FIR filter will exactly match the desired phase characteristics, modulo the fixed delay. Type 3 (odd-length) FIR filters have an

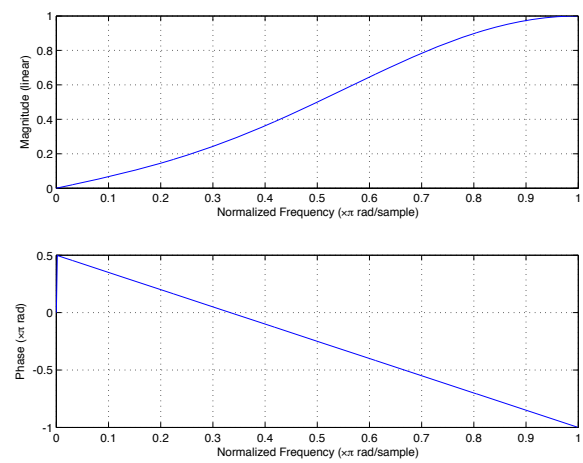
obligatory zero at $\omega = \pi$, so they wont be good here. That leaves Type 4, even-length antisymmetric filters. They have a non-integer (half-sample) fixed delay, which could be a problem, but weve been told to ignore delay here. By increasing the length of the filter, we can get increasingly accurate approximations to the magnitude response, up to whatever accuracy we desire.

(c) Because we want to use an even-length filter, we are going to choose a 3rd order (4 point) antisymmetric FIR filter. This has only 2 degrees of freedom, since $h[2] = -h[1]$ and $h[3] = -h[0]$. So all we have to do is solve for these two values in order to match our two spectral-magnitude samples. Slide 33 of Topic 6 shows how the frequency response is composed as the linear combination of sinusoidal basis functions, scaled by the impulse response coefficients. In our case, the magnitude response $\tilde{H}(\omega) = 2\left(h[1]\sin\frac{\omega}{2} + h[0]\sin\frac{3\omega}{2}\right)$, so we just need to choose the values we want for $\tilde{H}(\pi/2)$ and $\tilde{H}(\pi)$ and solve the linear equations. If we choose $\tilde{H}(\pi)$ as 1, then $\tilde{H}(\pi/2)$ should be 0.5, so we get:

$$\begin{aligned} 1 &= 2(h[1]\sin\frac{\pi}{2} + h[0]\sin\frac{3\pi}{2}) = 2h[1] - 2h[0] \\ 0.5 &= 2(h[1]\sin\frac{\pi}{4} + h[0]\sin\frac{3\pi}{4}) = \sqrt{2}h[1] - \sqrt{2}h[0] \end{aligned} \tag{4}$$

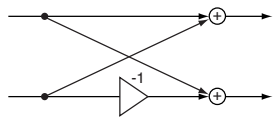
Thus, $h[0] = (\sqrt{2} - 2)/8$, $h[1] = (\sqrt{2} + 2)/8$. Heres the frequency response of the resulting 4 point filter:

```
>> h = [(sqrt(2)-2)/8 (sqrt(2)+2)/8];
>> h = [h,-fliplr(h)];
>> [H,W]=freqz(h);
>> subplot(211)
>> plot(W/pi, abs(H));
>> grid
>> subplot(212)
>> plot(W/pi, unwrap(angle(H))/pi);
>> grid
```



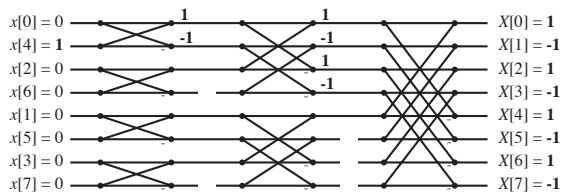
At such a low order, its not a very good approximation, but note that it does indeed show the intended gain values at $\omega = \pi/2$ and π, and that the phase is $+\pi/2$ at $\omega = 0$, and is a linear function of frequency with a slope of $(-3\pi/2)/\pi$, for a delay of 3/2 samples – i.e. its a constant delay away from being a pure 90 degree phase shift, regardless of the magnitude imperfections.

3. (a) The FFT flowgraph should be recognizable as a modified decimation-in-time structure, where the twiddle factors are factored out of the butterfly, which then only involves addition and subtraction. Thus for part (a) I was looking for the diagram below, with no mention of W_N.



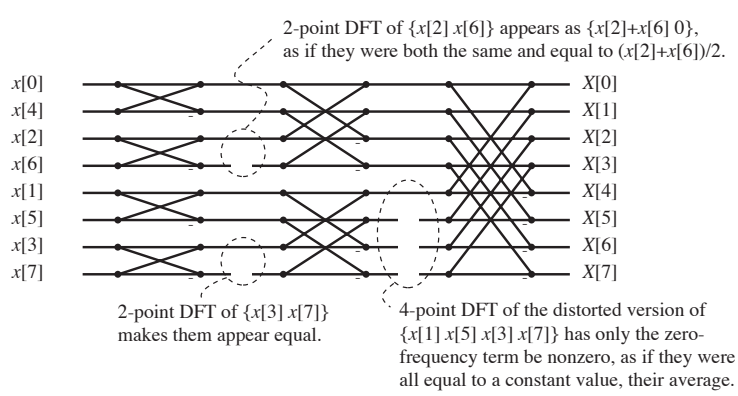
(b) Again by comparison with an 8-point modified DIT flowgraph, we can see that the indicated twiddle factors are $a = b = d = W_N^2$, $c = W_N^1$, and $e = W_N^3$. Since $N = 8$, $W_N = e^{-j2\pi/N} = e^{-j\pi/4} = \frac{1}{\sqrt{(2)}}(1-j)$, so $a, b, d = -j$, $c = (1-j)/\sqrt{2}$, and $e = (-1-j)/\sqrt{2}$.

(c) The input sequence $x[n] = \delta[n-4]$ i.e. only $x[4]$ is nonzero. Taking care to find it correctly on the bit-reversed- indexed input nodes, the flowgraph becomes as shown below:



The figure shows the negated inputs to the butterflies with a small "-", and shows the erroneously zeroed twiddle factors as broken lines. This outcome is of course the correct 8-point DFT of $\delta[n-4]$, my point being that with this test case, the bug would go undetected. (Because of the high structure and symmetry of the DFT, some bugs can give surprising and deceptive results, rather than obviously totally failing.)

(d) The intended approach to understanding what the buggy DFT actually does is shown below. The outputs of each 'stage' of the FFT are the smaller DFTs of subsets of the input points. Clearing some of these DFT values to zero is like 'filtering' these subsets to remove certain frequency components. Specifically, leaving only the zero-frequency (average) term for the three affected DFTs make the outputs correspond to apparent inputs with only a constant, average component (all points equal). Thus, the apparent input $x'[n]$ is given by: $x'[0] = x[0]$, $x'[4] = x[4]$, $x'[2] = x'[6] = (x[2] + x[6])/2$, $x'[1] = x'[5] = x'[3] = x'[7] = (x[1] + x[5] + x[3] + x[7])/4$.



4. This question is really about how to build fractional (subsample) delays. It tries to lead you towards the right approach.

   (a) This part is essentially trivial. We want to delay a signal sampled at 20 kHz by 1.25 ms i.e. by $20,000 \times 0.00125 = 20 \times 1.25 = 25$ samples. Since it is an integral number of samples, we simply use a delay line $H(z) = z^{-25}$, or $H(e^{j\omega}) = e^{-25j\omega}$. This pure delay has magnitude response $|H(e^{j\omega})| = 1$ i.e. constant for all ω, and a linear phase response $\theta(\omega) = -25\omega$, which is steep but otherwise simple. The impulse response is $h[n] = \delta[n - 25]$.

   (b) When the sampling rate drops to 10 kHz, we now wish for a delay that is half as many samples i.e. 12.5. This is no longer and integral number of samples, so a simple delay line won't work, although the ideal frequency response is still simply $H_b(e^{j\omega}) = e^{-12.5j\omega}$.

   There are several ways to approach this. One is to take the inverse DTFT of the ideal response. Another is to make the argument that we could upsample the data to 20 kHz, delay by 25 samples, then downsample by a factor of 2 back to 10 kHz (with an appropriate anti-aliasing filter). A third way is to recall that when looking at linear-phase FIR filters, we noticed that even-length (odd-order) "type II" filters had delays that were always midway between two sample points. Thus, one way to build this delay would be a 26-point, even length, symmetric FIR filter, which would span $h[0]$ to $h[25]$ and would have a delay of exactly 12.5 samples. We recall, however, that type II filters have an obligatory zero at $\omega = \pi$, so it will not be possible to get a completely flat magnitude response for all frequencies; in fact, we will end up with a finite-order Fourier approximation to a square wave.

   It turns out that all these approaches give essentially equivalent results. The upsample-downsample approach is perhaps conceptually most simple. Recall that simple interpolation of zeros to get twice as many zeros will place the original spectrum into the range $\omega = 0 \ldots \pi/2$, with an image of the negative frequency spectrum occupying the remainder of the 'principal' frequency range, $\omega = \pi/2 \ldots \pi$. To obtain just the spectrum of the original signal, this upsampled signal needs to be low-pass filtered at $\omega_c = \pi/2$. (This combination of interpolating zeros then low-pass filtering is simply conventional sample rate conversion.) We then apply the delay i.e. shift by 25 samples, then downsample back to 10 kHz by discarding every second sample. Because there is no energy above half the Nyquist rate (thanks to the low-pass filter), this decimation does not cause any aliasing, so we are done.

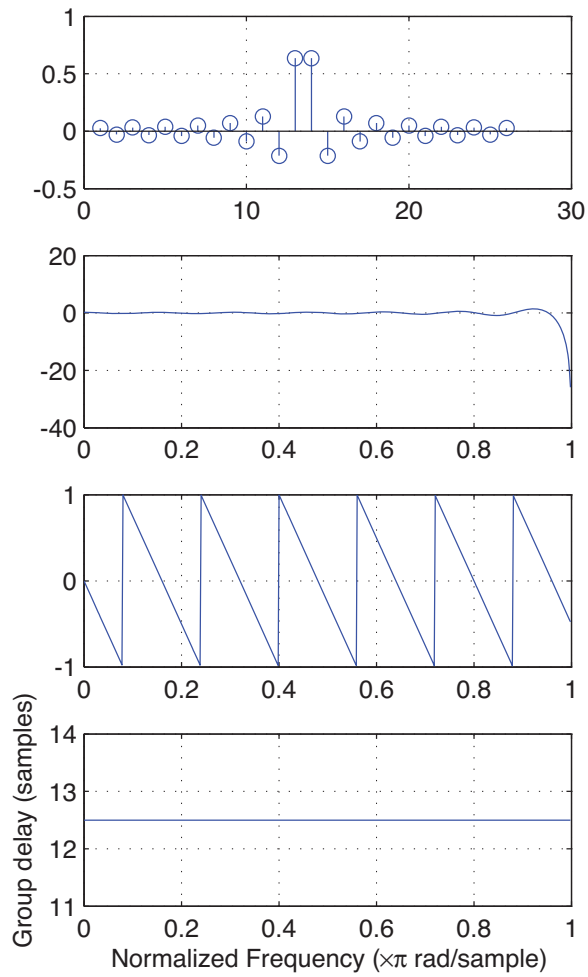
   By considering where each point in the final, decimated output comes from, you should be able to see that this sequence of upsampling, filtering, and downsampling is equivalent to taking every other sample of the low-pass filter impulse response and applying that (decimated) filter response to the original (10 kHz) signal.

   For an ideal brick-wall lowpass filter, the impulse response at 20 kHz is a sinc function: $h[n] = \sin(\frac{\pi}{2}n)/(\frac{\pi}{2}n)$. You can see that every even value of n other than $n = 0$ gives a zero value, since it falls on a zero crossing of the $\sin()$ numerator. Thus, if the delay was zero (or even), the effective filter at the lower sampling rate would be just a (shifted) impulse. However, if the delay is odd, the downsampled filter becomes all the nonzero, in-between values: this is the impulse response of an ideal, half-sample delay filter, the IDTFT of $e^{-j(r+0.5)\omega}$ where r is an integer.

   In practice we will have to truncate this decimated sinc function to create a realizable FIR filter. The truncation will result in some ringing (e.g. Gibb's phenomenon) around the cutoff frequency, which is the Nyquist rate for the 10 kHz system. In fact, the aliasing that occurs at the Nyquist frequency in the final downsampling results in the gain being zero at that frequency, introducing the "obligatory" zero of the type-II FIR filter – since the odd-indexed truncated

sinc function will indeed be a symmetric, even-length FIR filter (assuming we truncate it symmetrically). Below we see an example in Matlab; the important features were (a) linear phase response with slope -12.5; (b) near-flat magnitude response, but some ripples near Nyquist, and a zero at the Nyquist frequency itself; (c) impulse response is the maxima of a sinc function i.e. decays as $1/n$, points alternate in sign, but two equal-valued middle points.

```
>> n = -25:25;
>> hu = sin(pi/2*n)./(pi/2*n);
Warning:  Divide by zero.
>> hd = hu(1:2:end);
>> subplot(421)
>> stem(hd)
>> [HD,W] = freqz(hd);
>> subplot(423)
>> plot(W/pi,20*log10(abs(HD)))
>> subplot(425)
>> plot(W/pi,angle(HD)/pi)
>> subplot(427)
>> grpdelay(hd)
```



The final plot shows the group delay using Matlab's `grpdelay` function, which amounts to $-\frac{d}{d\omega}\angle\{H_b(e^{j\omega})\}$, and shows that the filter does indeed have a constant delay of 12.5 samples for all frequencies.

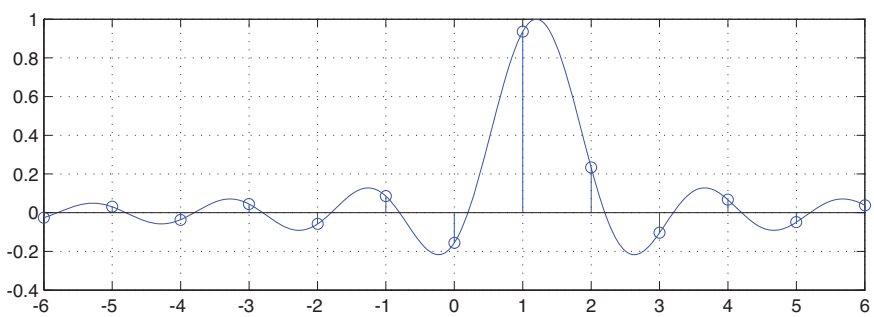
(c) To generalize this for an arbitrary delay which is not, in general, an integer number of samples, the upsample-filter-downsample approach will become impractical or impossible, so we should take the IDFT approach. The ideal system has frequency response $H_c(e^{j\omega}) = e^{-j\omega\tau}$, so its

magnitude is flat and its phase response is linear. Thus, the ideal filter impulse response is:

$$
\begin{aligned}
h_c[n] &= \mathcal{F}^{-1}\{e^{-j\omega\tau}\} \\
&= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{-j\omega\tau} e^{j\omega n} d\omega \\
&= \frac{1}{2\pi}\left[\frac{1}{j(n-\tau)} e^{j(n-\tau)\omega}\right]_{-\pi}^{\pi} \\
&= \frac{\sin \pi(n-\tau)}{\pi(n-\tau)}
\end{aligned}
\tag{5}
$$

Evaluating this for discrete values of n gives the (infinite extent) impulse response of the ideal filter, which can be truncated to give a practical implementation. These values are samples of the continuous $\mathrm{sinc}((n-\tau)\pi)$ function at integer values of n, as illustrated below, for $\tau = 1.2$:

```
>> tau = 1.2;
>> n = -6:.01:6;
>> sincpi = @(x) sin(pi*x)./(pi*x);
>> plot(n, sincpi(n-tau))
>> set(gca, 'XTick', -6:6)
>> grid
>> hold on
>> stem([-6:6],sincpi([-6:6]-tau))
>> hold off
```



As you can see, the actual values of the (truncated) FIR impluse response are values from the smooth sinc function, offset by the delay. When the delay is an integer number of samples, all the values other than the central one will be zero. If the delay is exactly midway between two samples (as in part (b)), the samples will all fall on (or very close to) the extrema of the sinc function, giving an even-length symmetric filter with no zero values.