1 Finite-impulse response filter design

These notes describe some basic techniques for designing real causal finite impulse response (FIR) filters. The goal is to design a causal, FIR filter whose impulse response $h[n]$ is zero for $n < 0$ or $n > M$, for a total impulse response length of $M + 1$. We will make three assumptions to simplify the design. First, we will assume $h[n]$ is real. Second, we will assume M is an even number, so the filter has an odd length. Third, we will assume that the filter is symmetric around $M/2$. It is possible to design complex causal FIR filters with even length, but it complicates the bookkeeping without changing the underlying ideas. Restricting this introduction to the straightforward case of real symmetric odd-length FIR filters allows us to minimize the distracting bookkeeping and focus on the important ideas.

Section 2.4 of the textbook described how nonrecursive linear, constant-coefficient difference equations produce FIR filters, but does not describe how to find the coefficients b_ℓ for a desired filter. Finding the filter coefficients b_ℓ is equivalent to finding the impulse response $h[n]$. One way to see this is to compare the linear, constant-coefficient difference equation with the convolution sum. The Mth-order nonrecursive linear, constant-coefficient difference equation describes $y[n]$ in terms of a sum of scaled and shifted version of the input $x[n]$ using the coefficients b_ℓ:

$$y[n] = \sum_{\ell=0}^{M} b_\ell x[n-\ell]. \tag{1}$$

The convolution sum also describes the output at a sum of scaled and shifted versions of the input, but uses the impulse response $h[n]$ to scale the terms. When the impulse response is a causal FIR filter as specified above, the convolution sum simplifies to

$$y[n] = \sum_{\ell=0}^{M} h[\ell] x[n-\ell]. \tag{2}$$

For the output $y[n]$ to be the same in Eqs. (1) and (2), the coefficients b_ℓ must be the same as the impulse response, i.e., $h[\ell] = b_\ell$. The same result can be obtained by setting the input $x[n]$ to an impulse $\delta[n]$ in Eq. (1), and remembering that the output is then $y[n] = h[n]$. Either approach demonstrates that the linear, constant-coefficient difference equation coefficients and the impulse response are the same, but neither approach tells us how to find $h[n]$ for a desired frequency response $H_{id}(e^{j\omega})$.

Filter design usually focuses on the frequency response magnitude $|H(e^{j\omega})|$. A natural starting point for FIR filter design is to find the impulse response $h_{id}[n]$ for the desired ideal frequency response $H_{id}(e^{j\omega})$. For example, for an ideal lowpass filter (LPF) with cutoff ω_c as shown in Figure 6.10(b), Eq. (6.20) gives the ideal impulse response

$$h_{id}[n] = \frac{\sin(\omega_c n)}{\pi n}. \tag{3}$$

This impulse response has two problems from the point of causal FIR filter design. First, it is not causal. Second, it is infinitely long, so no amount of delay can make it causal.

Any practical FIR filter will differ from the ideal filter in two important respects. First, the response will not be exactly constant in the passband or stopband, but will instead have ripples above and below the desired frequency response magnitude. Second, a practical filter requires some frequency width to transition from the passband to the stopband, rather than switching instantaneously as the ideal filter in Figure 6.10(b) does. Recognizing these limitations, FIR filters are designed to satisfy a set of specifications on the frequency response magnitude $|H(e^{j\omega})|$ such as those shown in Figure 1 for a LPF.

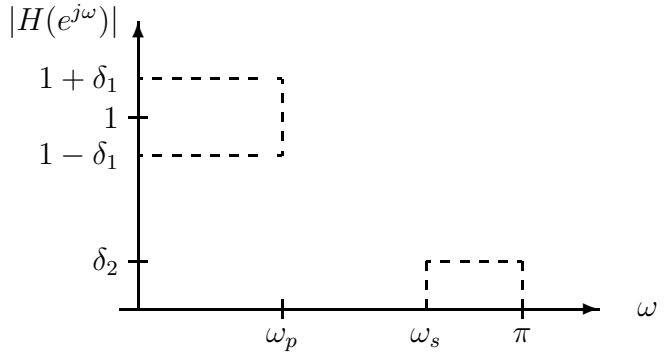

Figure 1: Specifications for the frequency response magnitude of a discrete-time lowpass filter

The region for $\omega < \omega_p$ is the passband, and the region for $\omega > \omega_s$ is the stopband. The region between the passband and stopband, $\omega_p \leq \omega \leq \omega_s$ is the transition band, and the width of this region $\Delta\omega = \omega_s - \omega_p$ is the transition band width. Within the passband, the frequency response magnitude must be within δ_1 of the ideal value of 1, and so δ_1 is the passband ripple. Similarly, δ_2 is the stopband ripple, since the frequency response magnitude of the filter must be less than δ_2 within the stopband. The ripples are sometimes specified in terms of dB of attenuation, which is computed as $-20\log_{10}\delta$. For example, if the passband is constrained to fall within $0.99 \leq |H(e^{j\omega})| \leq 1.01$, then $\delta_1 = 0.01$ and we say that the passband has $-20\log_{10} 0.01 = 40$ dB of attenuation. Since we are restricting our attention to filters with a real impulse response $h[n]$, the magnitude is even, and the specification on $|H(e^{j\omega})|$ is assumed to be symmetric about $\omega = 0$.

1.1 Basic FIR lowpass filter design

The most straightforward method for designing a causal FIR filter is the three step algorithm below.

1. Find the ideal impulse response $h_{id}[n]$ corresponding to the desired frequency response $H_{id}(e^{j\omega})$.

2. Truncate the ideal impulse response $h_{id}[n]$ symmetrically about $n = 0$, setting it to zero for $n < -L$ and $n > L$, where $L = M/2$. Let $h_{nc}[n]$ be this truncated impulse response, so

$$h_{nc}[n] = \begin{cases} h_{id}[n], & -L \leq n \leq L \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

3. Delay the noncausal impulse response $h_{nc}[n]$ by L samples to make a causal filter. The resulting filter is

$$\begin{aligned} h[n] &= h_{nc}[n-L] && (5) \\ &= \begin{cases} h_{id}[n-L], & 0 \le n \le M = 2L \\ 0, & n < 0 \text{ or } n > M. \end{cases} && (6) \end{aligned}$$

The resulting impulse response $h[n]$ is a causal, FIR filter which is symmetric about $n = L$. If L is relatively large, the samples of $h_{id}[n]$ set to zero in step 2 often have small amplitudes, and it is reasonable to expect that the frequency response magnitude $|H(e^{j\omega})|$ approximates the desired $|H_{id}(e^{j\omega})|$.

For an ideal lowpass filter with cutoff frequency ω_c, substituting Eq. (3) into Eqs. (4) and (6) yields the following

$$\begin{aligned} h_{nc}[n] &= \begin{cases} \frac{\sin(\omega_c n)}{\pi n} & -L \le n \le L \\ 0, & \text{otherwise} \end{cases} && (7) \\ h[n] &= \begin{cases} \frac{\sin(\omega_c (n-L))}{\pi (n-L)} & 0 \le n \le 2L \\ 0, & \text{otherwise} \end{cases} && (8) \end{aligned}$$

Figure 2 plots both $h_{nc}[n]$ and $h[n]$ for a LPF with $\omega_c = \pi/3$ and $L = 20$. Note that the only difference between the impulse responses is the time axis which is shifted due to the delay by L in step 3.

The delay of L samples to make $h[n]$ casual does not change the filter's magnitude and adds a linear phase term to the filter's phase. From the delay property of the Fourier transform

$$H(e^{j\omega}) = e^{-j\omega L} H_{nc}(e^{j\omega}).$$

The magnitude of $e^{-j\omega L}$ is 1, so $|H(e^{j\omega})| = |H_{nc}(e^{j\omega})|$. Delaying the impulse response to make it causal leaves the magnitude unchanged. The phase of the causal FIR filter can be understood using symmetry properties of the Fourier transform. Because $h_{nc}[n]$ is even symmetric, $H_{nc}(e^{j\omega})$ is real, and therefore the phase

$$\angle H_{nc}(e^{j\omega}) = \arctan\left(\frac{j\mathcal{I}m\{H_{nc}(e^{j\omega})\}}{\mathcal{R}e\{H_{nc}(e^{j\omega})\}}\right)$$

is zero, since $j\mathcal{I}m\{H_{nc}(e^{j\omega})\} = 0$. Consequently, the phase of the filter can be determined to be

$$\begin{aligned} \angle H(e^{j\omega}) &= \angle H_{nc}(e^{j\omega}) + \angle e^{-j\omega L} \\ &= 0 - \omega L = -\omega L. && (9) \end{aligned}$$

Systems with the frequency response phase in this form are said to have linear phase. Although we will not discuss phase responses in depth in this class, it is worth making a few brief remarks about its advantages. Linear phase is a desirable quantity in many applications. A system with linear phase has constant group delay (See Section 6.2.2 in the textbook for a discussion of group delay). The group delay describes how many

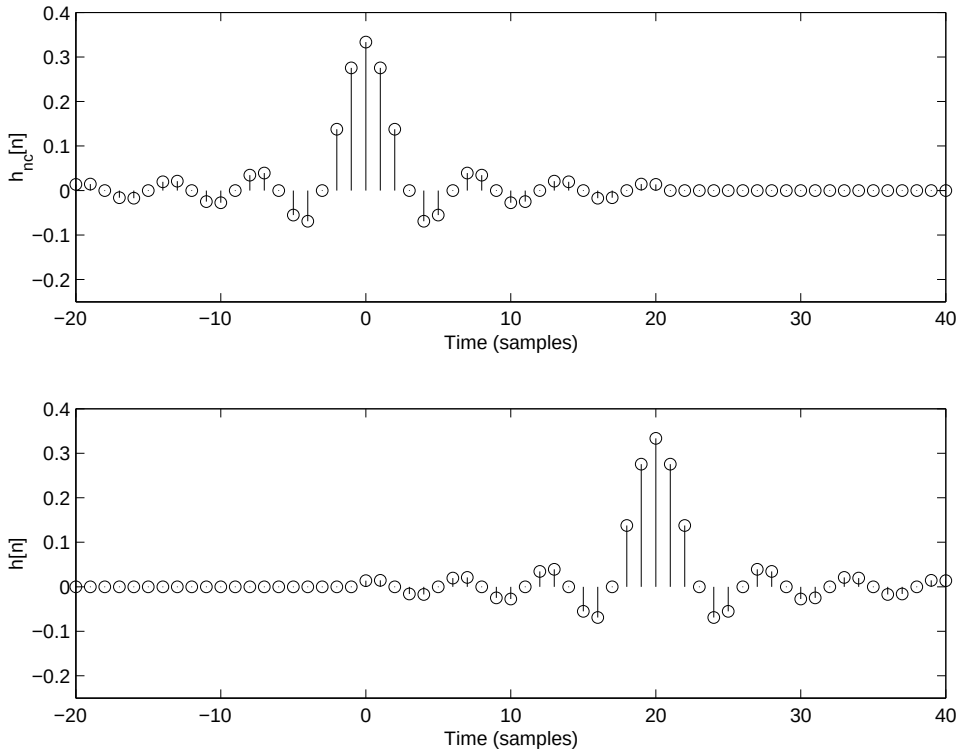

Figure 2: Noncausal and causal FIR lowpass filters with $L = 20$ and $\omega_c = \pi/3$

samples of delay each frequency experiences while passing through the system. Linear phase systems avoid a kind of distortion known as dispersion. Dispersion occurs when the different frequencies in the signal experience different delays through the filter, and thus the frequency components of the output signal are significantly misaligned in time compared to the frequency components of the input signal. In many speech and audio applications, dispersion must be severe before it is noticeable. In other applications, such as images, even modest amounts of dispersion are quite visible. Linear phase, or constant group delay, systems, have the same delay for all frequencies, and thus avoid dispersion. A linear phase filter preserves the relative time shifts between the frequency components in the input and output signals.

A natural question to ask is how well the truncated and delayed LPF performs. What are the passband and stopband ripples, and how wide is the transition band? Matlab can be used to compute the frequency response of the system and answer these questions. The following Matlab commands define the impulse response $h[n]$ for the example in Figure 2, and then use `freqz` to compute the frequency response at 8192 points on the interval $0 \leq \omega \leq \pi$.

```
L = 20;
n = 0:(2*L);
h = (1/3)*sinc((1/3)*(n-L));
[H,omega] = freqz(h,1,8192);
```

The Matlab `sinc(x)` function returns the values $sin(\pi x)/(\pi x)$. Multiplying the output of `sinc` by 1/3 cancels the extra 1/3 in the denominator of the `sinc` function that is not in Eq. (8). The input arguments to `freqz` specifying the linear, constant-coefficient difference equation are simply the impulse response $h[n]$ and 1. We demonstrated earlier that the linear, constant-coefficient difference equation coefficients b_ℓ for the input $x[n-\ell]$ terms are the values of $h[n]$, and since there are no recursion terms in $y[n]$, the other argument to `freqz` is simply `a = 1`. Figure 3 plots the resulting frequency response magnitude $|H(e^{j\omega})|$. The filter is basically a good LPF, but has a fair amount of ripple in the passband and stopband. Figure 4 zooms in on the passband and stopbands to show the peak ripples. This detailed view reveals that the peak ripple in both bands is about 0.09, or 21 dB of attenuation. In fact, all filters designed this way will have roughly the same ripple, regardless of the length M. Changing the length may slightly vary the peak ripple, but it will always be about 0.09. The detailed plots reveal another common feature of all filters designed with this technique. The maximum, or peak, passband ripple is roughly the same as the maximum stopband ripple.

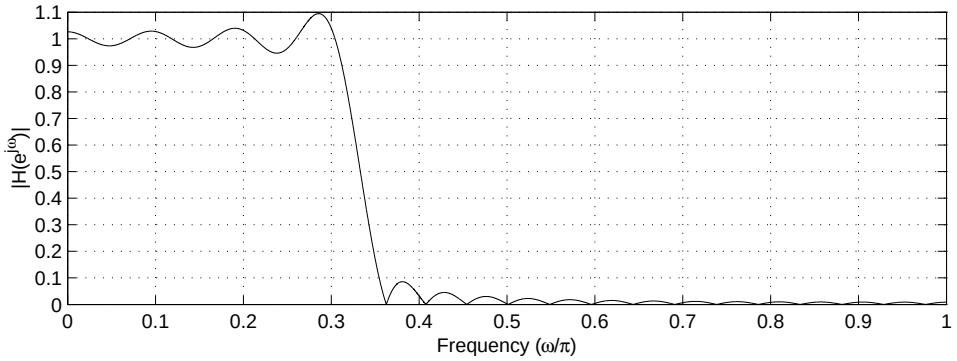

Figure 3: Frequency response magnitude $|H(e^{j\omega})|$ for 41-point FIR lowpass filter with $\omega_c = \pi/3$

For reference, the Matlab commands used to make Figure 3 are given below

```
plot(omega/pi,abs(H));
xlabel('Frequency (\omega/\pi)')
ylabel('|H(e^{j\omega})|')
axis([0 1 0 1.1])
set(gca,'ytick',[0:0.1:1.1]);
grid
```

The distance between the peak passband ripple and peak stopband ripple is roughly $4\pi/M$ for FIR filters designed with this technique. For $M = 40$, the distance between the peak ripples is about $\pi/10$. This corresponds well with the plot in Figure 3 where the peaks occur around 0.28π and 0.38π. The transition band distance $\Delta\omega$ is less than this peak ripple distance. The transition band width is the distance from the last frequency where $|H(e^{j\omega})|$ crosses $1 - \delta_1$ until the first frequency where $|H(e^{j\omega})|$ crosses δ_2. These crossings occur between the peak ripples. From the detailed plots of Figure 4, the

transition band width can be observed to be about $\Delta\omega = 0.04\pi$. In general, this width is also inversely proportional to the FIR filter length M, and is roughly $2\pi/M$. Increasing the length of the filter will make the transition band between the passband and stopband narrower, but has little effect on the peak passband and stopband ripples.

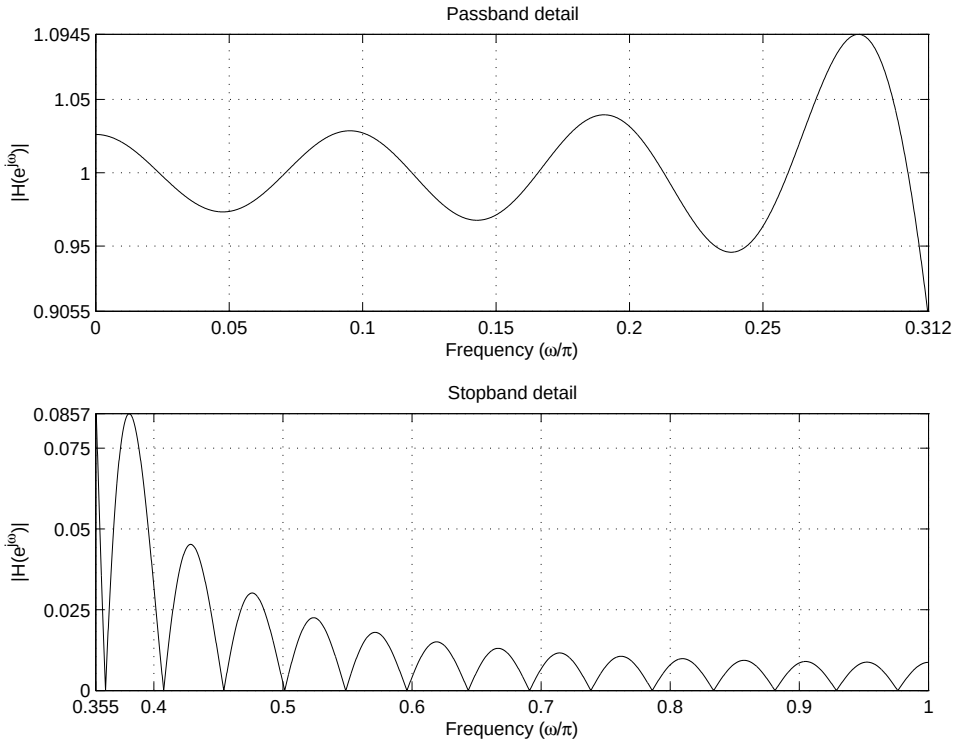

Figure 4: Detailed view of passband and stopband for FIR lowpass filter frequency response magnitude in Figure 3

FIR Filtering Example This example applies an FIR LPF to a signal consisting of two narrowband frequency pulses. Figure 5 plots the input signal, using Matlab's `plot` command, rather than `stem`. Strictly speaking, this is a discrete-time signal and should be displayed using `stem`. However, when the signal gets too long, we sometimes use `plot` instead because all of the circles created `stem` become unwieldy. Figure 6 shows the discrete-time Fourier transform magnitude $|X(e^{j\omega})|$ for this signal for the interval $-\pi \leq \omega \leq \pi$. The first pulse in $x[n]$ ($0 \leq n \leq 200$) oscillates more rapidly than the second pulse ($300 \leq n \leq 500$), so the first pulse has a higher frequency than the second pulse. The first pulse must correspond to the peaks at $\omega \approx \pm 0.5\pi$ in Figure 6, while the second pulse in $x[n]$ corresponds to the peaks at $\omega \approx \pm 0.1\pi$. Frequencies around 0.1π imply a period of roughly $2\pi/0.1\pi = 20$, which is also consistent with the period of the second pulse of $x[n]$ in Figure 5.

The input signal $x[n]$ is filtered using the causal FIR filter designed above. The Matlab command

```
y = filter(h,1,x);
```

implements the FIR filter. The filter could also be implemented with the Matlab `conv` command, but this would required computing a new index vector for `y` based on the start and end points of `h` and `x`. Figure 7 plots the resulting output signal. The filter removed the higher frequency pulse, but the lower frequency pulse passed through with its amplitude essentially unchanged. This results corresponds to what we would expect by comparing the input signal spectrum $|X(e^{j\omega})|$ in Figure 6 with the frequency response magnitude for the filter in Figure 3. A careful examination of Figure 7 reveals that the lower frequency pulse was delayed about 20 samples between the input and output signals. This delay of $L = M/2$ samples for all frequencies is a consistent feature for causal FIR filters designed using the technique above.

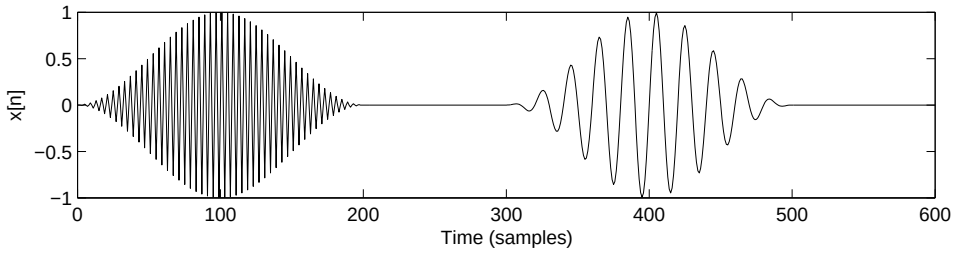

Figure 5: Input signal $x[n]$ with two narrowband pulses

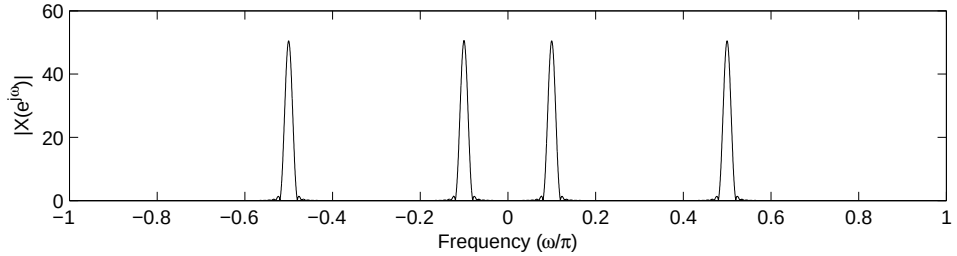

Figure 6: Input spectrum $|X(e^{j\omega})|$

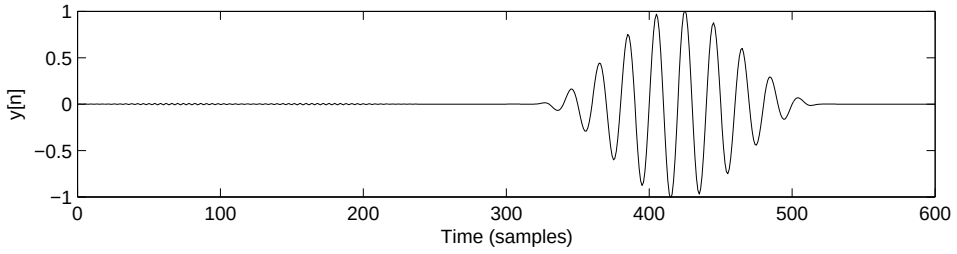

Figure 7: Output signal $y[n]$ with higher frequency pulse filtered out.

1.2 Windowed FIR lowpass filter design

We saw in the previous section that the peak passband and stopband ripples are largely insensitive to changes in the filter length. They may vary slightly as the filter length changes, but they will not be significantly increased or reduced within a wide range of filter lengths. The natural question that arises is how to design an FIR with smaller ripples. A straightforward way to do this is to generalize step 2 of the design algorithm to multiply the ideal impulse response $h_{id}[n]$ with a symmetric finite length window, rather than just truncating $h_{id}[n]$. Choosing the window $w[n] = 0$ for $n < -L$ and $n > L$ insures that the overall filter still has the desired length. The equations for the impulse responses now become

$$\begin{aligned} h_{nc}[n] &= w[n]h_{id}[n], \\ h[n] &= h_{nc}[n-L], \\ h[n] &= w[n-L]h_{id}[n-L]. \end{aligned}$$

In fact, the truncation algorithm presented in the previous section is equivalent to choosing the window to be

$$w[n] = \begin{cases} 1, & -L \le n \le L, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

This window is sometimes referred to as the rectangular window. Intuitively, abrupt transitions generate high frequencies. Therefore, smoothing the transition at both edges of the window should reduce the ripple height. There are many choices of windows which can be used to reduce the ripples in the filter frequency response. All of them also cause the transition band of the filter to be wider than the filter obtained by truncating $h_{id}[n]$. In order to reduce the ripples while keeping the transition bandwidth unchanged from the truncation approach, the FIR filter must be longer. Longer filters often imply higher costs or longer computation times, so this is a design tradeoff that must be evaluated carefully.

A common window used to design FIR filters is the Hamming window

$$w[n] = \begin{cases} 0.54 + 0.46\cos(\pi n/L), & -L \le n \le L, \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

Figure 8 plots a Hamming window $w[n]$ for $L = 20$. FIR filters designed using a Hamming window have peak ripples of roughly 0.0022, or about 53 dB of attenuation. The cost of this reduction in ripple is that the distance from peak passband to peak stopband ripple is twice as wide as a rectangular window at roughly $8\pi/M$. The transition band for an FIR filter designed with a Hamming window is about 3 times wider than an FIR filter designed with a rectangular window.

Frequency Responses of Hamming and Rectangular window FIR Filters This example contrasts the frequency response of the rectangular window filter designed in the previous section with the frequency response obtained using a Hamming window with the

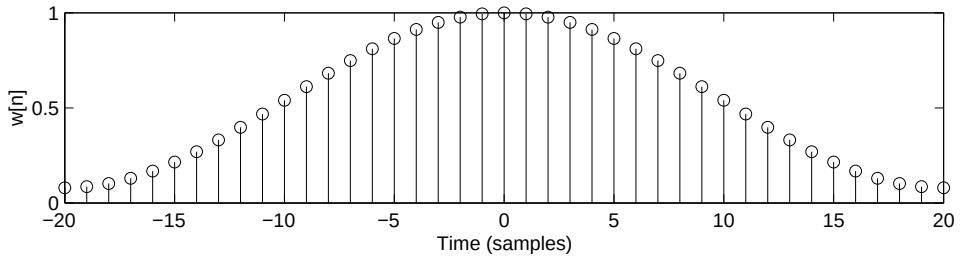

Figure 8: 41-point Hamming window $w[n]$

same length to truncate $h_{id}[n]$. Figure 9 plots the frequency responses for two FIR LPF filters with $M = 40$ and $\omega_c = \pi/3$. The solid line is the frequency response magnitude for the filter designed in the previous section using a rectangular window. The dashed line is filter obtained by applying the same length Hamming window to the same $h_{id}[n]$. The filter was designed in Matlab using

```
hham = fir1(40,1/3);
Hham = freqz(hham,1,8192);
```

By default, the `fir1(M,alpha)` command uses the Hamming window to design a causal FIR LPF with an impulse response from $0 \leq n \leq$ `M` and a cutoff frequency of `alpha`π. There are many other options to design other kinds of filters or use other windows. You are encouraged to read the `help` for this filter carefully. Both features of Hamming window FIR filters discussed above are clearly visible in Figure 9. The Hamming window filter has much smaller ripples (essentially invisible on this scale), and a wider transition band. The ripples for the Hamming window are more visible shown on a dB scale, as in Figure 10. The 53 dB of peak attenuation error is clearly visible in the stopband of Figure 10.

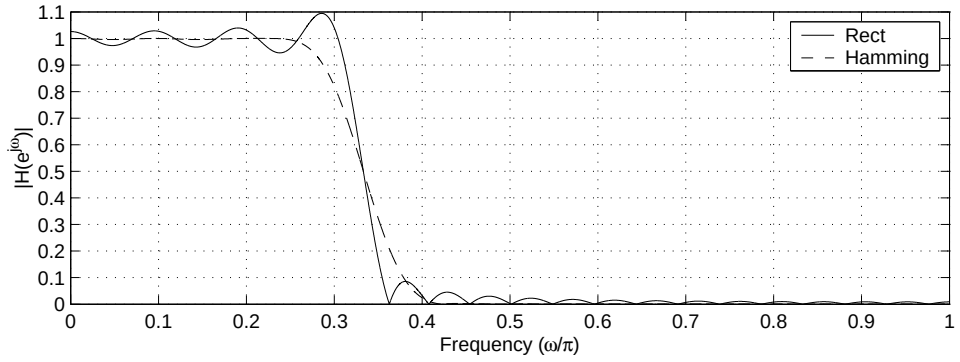

Figure 9: Comparison of frequency responses for 41-point FIR filters with $\omega_c = \pi/3$ designed using rectangular (solid) and Hamming (dashed) windows

Filtering with Rectangular and Hamming Window FIR filters This example demonstrates the advantage of using smaller ripples in some applications. Figure 11

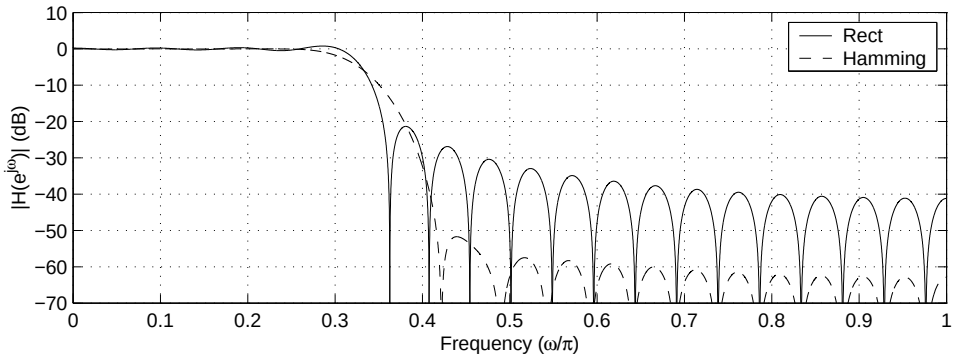

Figure 10: Frequency responses of Figure 9 plotted on a dB scale

compares the result of filtering the same input signal with two different LPFs with $M = 40$ and $\omega_c = \pi/3$. The first filter is designed with a rectangular window and the second is designed with a Hamming window. The input signal contains two frequency components. Unlike the earlier example with the two pulses, both frequencies are present all the time in this signal. The lower frequency component is at $\omega = \pi/5$ with amplitude 1 and the higher frequency ($\omega = 0.48\pi$) component has amplitude 100. The top panel of Figure 11 plots $x[n]$. The low frequency component is so far below the high frequency component that it is essentially invisible. In theory, the high frequency component is in the stopband of both filters, and both filters should remove the high frequency component, leaving only the low frequency component. In practice, the ripples of the rectangular window filter limits how much attenuation the high frequency component experiences. The middle panel in Figure 11 shows the output of the rectangular window filter. The high frequency component is much lower in amplitude in comparison to the input $x[n]$ in the top panel, but it still obscures the low frequency signal. The bottom panel in Figure 11 shows the output from the Hamming window filter. Here the low frequency signal is clearly visible and the high frequency signal is essentially absent.

1.3 Other FIR frequency selective filters

The sections above discussed lowpass filters, but not the other common frequency selective filters such as highpass filters and bandpass filters. The same 3 step algorithm applies for these filters. In order to find $h_{id}[n]$, it is often helpful to express the desired frequency response in terms of lowpass filter, then use Fourier transform properties to modify the prototype lowpass filter into the desired filter. For example, if we desire a highpass filter with cutoff $\omega_c = \pi/3$, we could observe that this filter's frequency response is

$$H_{id,hpf}(e^{j\omega}) = 1 - H_{id,lpf}(e^{j\omega}), \tag{12}$$

where $H_{id,lpf}(e^{j\omega})$ is the ideal lowpass filter in the previous sections. Taking the inverse Fourier transform, we find that our new ideal impulse response is

$$h_{id,hpf}[n] = \delta[n] - \frac{\sin \omega_c n}{\pi n}. \tag{13}$$

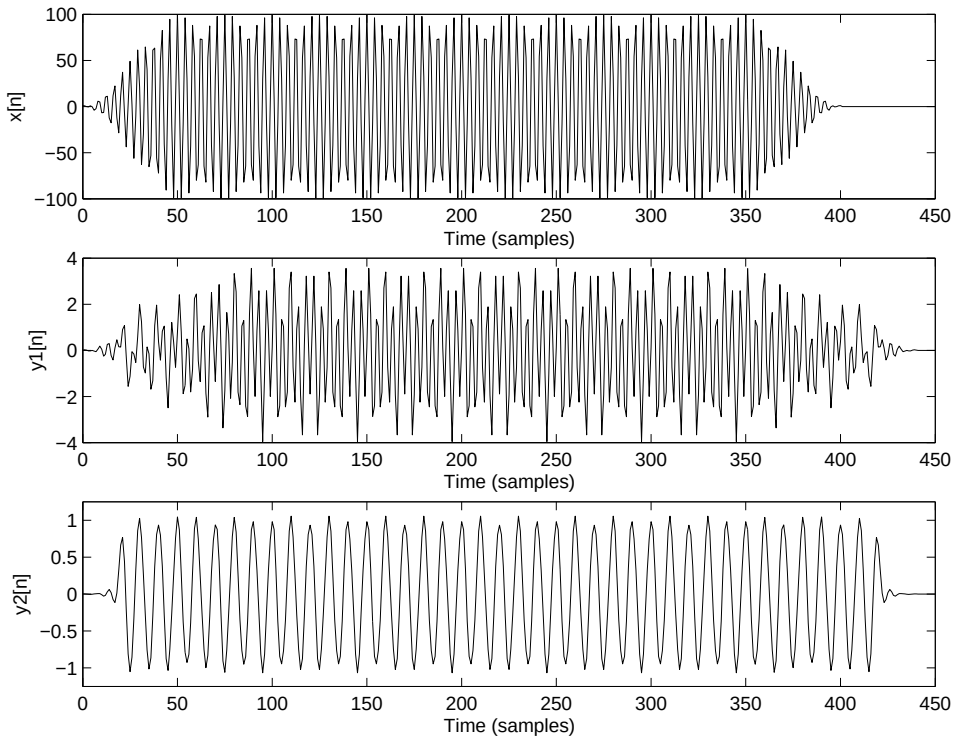

Figure 11: Comparison of rectangular vs. Hamming window FIR filters. The top panel is the input signal $x[n]$ with a loud high frequency signal and a quiet low frequency signal. The middle panel is the output from the rectangular window FIR filter. Although the high frequency signal is much lower amplitude in $x[n]$, it still obscures the low frequency signal. The bottom panel shows the output from the Hamming window filter in which the high frequency signal is so attenuated to be invisible, and the lower frequency ($\pi/5$) signal is clearly visible.

Once we have our ideal impulse response, steps 2 and 3 remain the same: window and delay. The following Matlab commands compute the impulse response for this FIR highpass filter using a rectangular window:

```
L = 20;
n = 0:(2*L);
h = [zeros(1,L) 1 zeros(1,L)] - (1/3)*sinc((1/3)*(n-L));
```

Note that the first term in the equation for `h` begins with `L` zeros to account for the delay by L in step 3. It is a common mistake to forget this delay and make a nonsymmetric $h[n]$. Figure 12 plots the impulse response $h[n]$ from these commands.

The frequency response magnitude in Figure 13 shows that we have in fact made the desired highpass filter. In fact, for this case the new filter has the same peak ripples and transition band width as the old filter. If we windowed the ideal response with a Hamming window, it would reduce the ripples but widen the transition band.

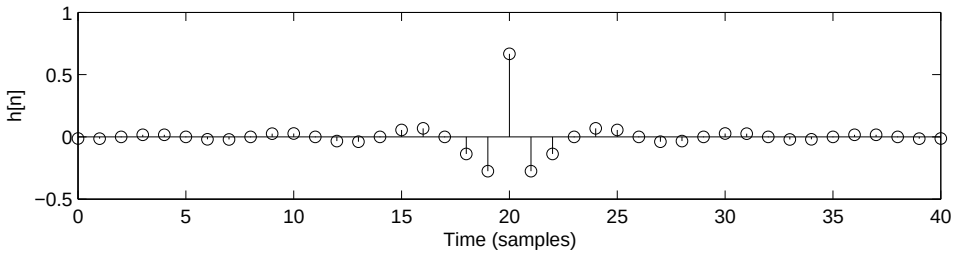

Figure 12: Highpass filter impulse response $h[n]$

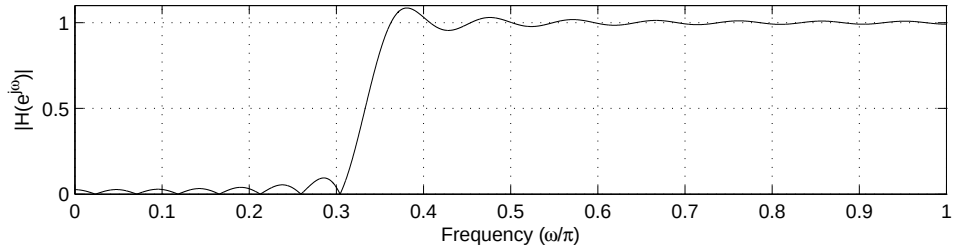

Figure 13: Highpass filter frequency response magnitude $|H(e^{j\omega})|$

Bandpass filters can be designed by first finding the impulse response for a LPF with the desired bandwidth, then using the modulation property to shift the passband in frequency to the desired location. Combining Fourier transform properties with some basic impulse responses allows you to design a wide range of FIR frequency selective filters.