Matlab Project: FIR filter design

EE-125

In this project you will use three methods of FIR filter design discussed in class: windowing, frequency-sampling, and Parks-McClellan equiripple filter design. In the first part of the project, you will write your own code to design filters using the frequency-sampling approach, and will explore the effects of the transition band. In the second part of the project, you will use built-in MATLAB tools to design equiripple and window-based filters, and will compare filter lengths needed to meet the specification.

The problems below ask you to plot the frequency response of the filters you create. When you do so, be sure to plot the amplitude and phase as a function of continuous-time frequency (i.e., in Hz) to verify the filter specifications (also in Hz) are satisfied. See 'help freqz' for how to do this easily.Also, plot filter responses in power in dB.

Filter design by frequency sampling

Your goal is to design a filter given the following specification and inputs:

- Sampling frequency = 5000 Hz
- Filter length L = 23
- Lowpass response: 1 for $f < 750$ Hz, 0 else

As a note, several options approaches are possible when designing filters using frequency-sampling. An efficient (but less general) approach is to compute analytic formulas for the $h[n]$ based on specific cases (filter length is symmetric or antisymmetric, etc.), which is the approach taken in section 10.2.3 of the textbook. A more straightforward approach is to directly specify the desired frequency response $H(\omega)$on an evenly spaced grid, then use the matlab 'ifft' command to calculate $h[n]$. This is the approach we will use here.

In doing so we will use a key Fourier transform property: the transform of a real signal is conjugate symmetric. Thus as long as we specific a conjugate-symmetric $H(\omega)$, the IFFT should give us a real-valued filter $h[n]$ (though numerical effects means there may be a small imaginary component, which we will remove).

Design your filter using the following steps. In your report, document these steps using code, any comments you find useful, and the plots noted below. As always, clearly label all plots.

- Set up a vector of frequencies f that is of length L. To avoid confusion, remember the Matlab convention used in fft and ifft: the first frequency in the array should be 0 Hz, and the L samples should cover the region $0 - f_s$, remembering that the region $f_s/2$ to f_s will be identical to the negative frequencies.
- Set up a vector H_{mag} (also length L) with the desired frequency response. Plot f vs. H_{mag} to ensure it is as expected. You may want to explore use of *fftshift* to get a plot with DC in the middle of the frequency axis. The number of frequencies with non-zero magnitude should be symmetric around DC.
- Create a corresponding vector $Hphase$ with the desired linear phase, so that the filter will be delayed to be causal. Make sure that the phase is zero at $f = 0$ Hz. Also, make sure that it is conjugate-symmetric; this can be a little tricky. Next, create the complex-valued frequency response $H = H_{mag}. * Hphase$.
- Now, use the 'ifft' command to create the filter:

h = ifft(H);

Examine the values of h. Any imaginary part of h should be very small; if not you should revisit your $Hphase$ calculation. Remove any small imaginary part using the command $h = real(h)$;. If you did things correctly, $h(n)$ should look like a sinc function.

- Finally, plot the magnitude and phase response of the filter. Does it match your expectations? Include a plot which clearly shows the response.

Congratulations! You have now designed a digital FIR filter from scratch.

Effect of transition band

(Note: This section assumes that you were able to implement the filter above. If you weren't able to (it can be tricky) use Matlab's 'fir2' command to implement a filter and answer the questions below.)

When designing FIR filters using the frequency-sampling method, we do not have the ability to directly control the amount of ripple in the filter output. This means we must carefully review and check the filter response. In the example above, the specified filter response had an extremely sharp transition (from 1 to 0 in a single frequency bin). This can lead to ripple and 'ringing' phenomena similar to the Gibb's phenomena that we studied earlier. We can help reduce ripple by introducing a slower transition in a 'transition band' between the passband and stopband, and by adjusting the speed of transition.

Explore this issue further as follows:

- Carefully review the ripple in the filter you designed above. This 'ripple' corresponds to error in your filter design. Make a zoomed plot showing the response in the passband, and measure and label the maximum error in the passband. At what frequency is this error the largest? IMPORTANT: In this context, 'error' means deviation from the desired response, which is 1 in the passband.

- Repeat the step below to find the max error in the stopband. Show a zoomed plot with stopband response, and list the max error and its frequency in your report.

- Create a new magnitude response H_{mag1} that has a 1-sample transition band (i.e., one frequency bin with response 0.5 that separates the pass and stop bands). Calculate the new filter coefficients $h2[n]$. Make a plot which has both your original $h(n)$ (no transition) and $h2(n)$ on the sample plot. Repeat the previous two steps to plot and characterize the passband and stopband errors, and comment on what you see.

- In the previous step, you used 0.5 for the transition value. In this step, vary the value of the transition value from 0.1 to 0.9 in steps of 0.1. For each, tabulate the max error in the passband and in the stopband. Summarize the results in a table (max errors vs. transition value). Discuss: what is the best choice if you are mainly interested in passband performance? in stop-band performance?

- Repeat the step above, but with a magnitude response H_{mag2} that has a 2-sample transition band (linearly decreasing magnitude). How much does this affect your measured error?

Filter design by Parks-McClellan

Your goal is to design a filter given the following specification and inputs:

- Sampling frequency = 1000 Hz
- Lowpass response, cutoff 100 Hz
- Allowable passband distortion (ripple) is $\pm 2\%$, i.e. $\delta_1 = 0.02$

- Above 175 Hz, the filter must have an attenuation of at least 40 dB, i.e. $20\log_{10}(\delta_2) = -40$.

Use the Matlab Parks-McClellan filter implementation (function 'firpmord' and 'firpm') to design the minimum-length filter that meets the specification. Besides 'firpmord', you can use the discussion in section 10.2.7 of the textbook to guide you in your search for the filter length M. Plot the magnitude and phase response of the filter you have designed, and show plots verifying it meets the design specifications. Be sure to look at passband and stopband ripple. Did you need to adjust the output of firpmord?

Filter design by windowing

Here, your goal is to meet the filter specifications that were listed for the Parks-McClellan design, but using FIR design by windowing.

1. For the filter length M you found for the equiripple filter, design a Hamming FIR window (remember the function 'fir1'). Note that 'fir1' specifies response in terms of the cutoff frequency - explain how you choose this value based on the filter specifications above. Plot the frequency response of the filter you have designed. Does it meet the filter design guidelines for pass- and stop-band ripple? If not, increase the filter length until it meets the desired characteristics.

2. Using a rectangular (boxcar) window, do the same - what filter length do we need to meet the specs?

3. Estimate the percent savings in run time that you would expect for the equiripple filter compared to the Hamming or boxcar.

Hint: It may seem strange that you could ever meet the filter specs for 40 dB suppression using the rectangular window, since its first sidelobe is only -13 dB. However, what lets you find a solution in this case is that there is a transition band. Because the sidelobe levels continue to decay away from the mainlobe, you should be able to find a design 'squeezes' many sidelobes into the transition band, so that the sidelobes sufficiently that the filter specs are met.

This points out part of why FIR filter design by windowing is widely used: with enough adjustments, you can use a relatively simple design technique to find workable filters. Question 3) above should help point out disadvantages of the approach.