EE 422G - Signals and Systems Laboratory

Lab 3 FIR Filters

Written by
Kevin D. Donohue
Department of Electrical and Computer Engineering
University of Kentucky
Lexington, KY 40506
September 26, 2010

Objectives:

- Use filter design and analysis tools to create FIR filters based on general filter specifications.
- Create a simulation with Simulink.

1. Background

Digital filters are used in a wide variety of signal processing applications, such as spectrum analysis, digital image processing, and pattern recognition. Digital filters eliminate a number of problems associated with their classical analog counterparts and thus are often used in place of analog filters. The most common digital filters belong to the class of discrete-time LTI (linear time invariant) systems, which are characterized by the properties of causality, recursibility, and stability. They can be characterized in the time domain by the unit-impulse response and in the transform domain by the transfer function. A unit-impulse response sequence of a causal LTI system can be either finite or infinite in duration. This property determines their classification as either a finite impulse response (FIR) or an infinite impulse response (IIR) systems. To illustrate this, consider the most general case of a discrete time LTI system with the input sequence denoted by $x(kT)$ and the resulting output sequence $y(kT)$ given by:

$$y(kT) = \sum_{m=0}^{M-1} b_m x((k-m)T) - \sum_{n=1}^{N-1} a_n y((k-n)T) \qquad (1)$$

The corresponding transfer function in the Z-domain is given by:

$$\hat{H}(z) = \frac{\hat{Y}(z)}{\hat{X}(z)} = \frac{\sum_{m=0}^{M-1} b_m z^{-m}}{1 + \sum_{n=1}^{N-1} a_n z^{-n}} \qquad (2)$$

If at least one denominator coefficient a_n is not zero, then system is recursive (its current output depends on previous output values), and as a result its impulse response is of infinite duration (IIR system). If all denominator coefficients are zero (polynomial of order 0), the corresponding system is non-recursive (FIR system), and its impulse response is of finite duration. The transfer function of Eq. (2) in this case becomes a polynomial of finite order M-1:

$$\hat{H}(z) = \frac{\hat{Y}(z)}{\hat{X}(z)} = \sum_{m=0}^{M-1} b_m z^{-m} \qquad (2)$$

The corresponding FIR difference equation in time domain is:

$$y(kT) = \sum_{m=0}^{M-1} b_m x((k-m)T) \qquad (3)$$

As with analog filter design, the general shape of the frequency response is the main criteria in discrete filter design. Recall the frequency response for continuous-time systems was obtained by substituting evaluating the transfer function on the $j\omega$ axis, similarly for the discrete case the transfer function in z is evaluated over the unit circle. In this case z is substituted with $z = \exp(j\omega/\omega_s)$ where ω_s is the sampling frequency in radians per second. Therefore, the frequency response of an FIR filter is given by:

$$\hat{H}(z)\Big|_{z=\exp\left[j\frac{\omega}{\omega_s}\right]} = \sum_{m=0}^{M-1} b_m \exp\left[-j\frac{m\omega}{\omega_s}\right] \qquad (4)$$

Note that even though the time domain is discrete, the frequency response is continuous (defined for all ω); however, it is periodic with period ω_s due to the periodic behavior of the complex exponential and consistent with the concept of aliasing.

The design of digital filters involves determining the filter order (M) and computing the values of the coefficients (b_i's in the above equations) to achieve the desired filter response. The desired response can be specified in the frequency domain in terms of the magnitude response and/or the phase response. It can also be specified in terms the impulse response. Once filter coefficients are computed, the filter performance must be analyzed to determine the filter meets specification.

In this lab you will design FIR filters using 2 popular methods – impulse response windowing and the Parks-McClellan algorithm. See help files in Matlab for *fir1()* and *firpm*() for a more detailed explanation of the 2 methods and algorithms used in each of these approaches. For the analysis of the filter, see help on transfer function evaluations tools like *fft()* (this takes the DFT of the signal) and *freqz()* (this implements the frequency response computation of Eq. (4)). There are 2 useful scripts posted on the class web site as well, which are *winlook* and *firlook* that show examples of using these functions.

2. Pre-Laboratory Assignment

1. Sketch a rectangular function with height 1 and width A seconds that is symmetrically distributed about 0. Sketch its Fourier transform (sinc function) and label the axis to identify the width of its mainlobe (distance between the first null points closest to 0 on the frequency axis), height of the mainlobe, and height of the first sidelobe (absolute value of the peak between first and second null points on either the positive or negative frequency axis).

2. For the tapering window functions listed below, write a script to plot each window function on the same graph (use different line styles for each window function). Create another graph and plot its DFT magnitude on a linear scale, and finally create another graph and plot its DFT magnitude on a dB scale. Comment on how the general window

shape (steepness of taper) affects the spectral magnitude (impact on width of mainlobe and height of sidelobes)

a) Boxcar
b) Triangular
c) Hamming

3. The Kiaser window is also very popular because the degree of the taper can be adjusted parametrically with parameter. Repeat number 2 for a Kaiser window of length 128 and with values of 2, 4, and 8. (see help *kaiser* in Matlab).

4. Become familiar with the template scripts *winlook.m* and *firlook.m*, posted on the course web site. Read through the comments so you know how they relate to the laboratory exercises. There is nothing the hand in on the prelab for this exercise.

3. Laboratory Assignment

(Assume sampling frequency of 44.1kHz, unless otherwise specified)

1. Design a 127th order linear-phase FIR low-pass filter with a cut-off at 6kHz using the windowing method (*fir1*). Design a filter using each of the following windows:

a. rectangular (`boxcar`)
b. triangular (`triang`)
c. Hamming (`hamming`)

On a single graph, plot the all the impulse responses together, and on another graph plot all the (frequency) magnitude responses together. In individual graphs plot the pole-zero locations of the 3 filters. **In the discussion section, compare the impact of the window shape on the filter characteristics (impulse response, Spectral magnitude, and zero placements).** Refer to the window plots of the prelab (in discussion use language such as mainlobe width, sidelobe, taper …) and make a general statement about the impact of window characteristics on the filter characteristics.

2. Repeat Exercise 1 for a 127th order linear-phase FIR low-pass filter using a Kaiser window with β = 2, 4, and 8 and pass-band cut off of 6kHz. Plot the impulse response, magnitude response, and zero-pole locations. **In the discussion section compare the characteristics of the magnitude response with each other and with the filters from the first exercise. Discuss how the trade-off between transition bandwidth and ripple varies with β .**

3. Repeat Exercise 2 for a 31st order linear-phase FIR low-pass filter. Plot the impulse response, magnitude response, and zero-pole locations. **In the discussion section compare the characteristics of the magnitude response with those from the previous exercise. Discuss the impact of filter order.**

4. Design an optimal 31-order low-pass FIR filter using the Parks-McClellan algorithm (*firpm*). Use a pass-band cutoff of 5.5kHz and a stop-band cutoff of 6.5kHz. Plot the