EE 224 Signals and Systems I
Lab 06: FIR Filters

In this lab you will experiment with FIR filters, in both the time domain and frequency domain.

1 Concept of FIR filter

1.1 Time-domain (20 minutes)

An FIR filter is a discrete LTI system whose impulse response has only a finite number of non-zero taps. The following equation defines an FIR filter that is causal:

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

We can see that the filter output y[n] is simply the weighted sum of the shifted versions of the input. This summation is called convolution. At each time instant n, y[n] is a linear combination of the input at the current and other times. The signal h[k] is the impulse response of the filter. The filter in (1) is causal because y[n] does not depend on future inputs.

In Matlab, signals are represented as vectors. So a vector [x1, x2, x3, x4] can be used to represent a signal with 4 values. If the time for the first sample is zero, then it represents the following signal:

x[0]=x1, x[1]=x2, x[2]=x3, x[3]=x4

If the time for the first sample is for n = –2, then it represents the following signal:

x[-2]=x1, x[-1]=x2, x[0]=x3, x[1]=x4

The programmer should know what exactly each vector represents in the code. In this Lab, we assume the the first sample corresponds to n = 0.

The following Matlab function, *filterfir*, performs the FIR filtering as given in (1).

```
function y=filterfir(h, x)
% FILTERFIR Finite impulse response (FIR) filtering function
%
% Y=FILTERFIR(H, X) returns the FIR filter result of X by the LTI system
% whose impulse response is H. It is assumed that both H and X represent
% signals that start at time zero. If X is a row vector, then Y will be
% a row vector. If X is a column vector, Y will be column vector.
%
% EXAMPLE:
% h=[0 2];
% x=[2 4 1];
% y=filterfir(h, x)

lenh=length(h);
lenx=length(x);
leny=lenh+lenx-1;

% Make an empty Y vector in the same direction as X
if size(x, 1)==lenx
y=zeros(leny, 1);
```

```
else
y=zeros(1, leny);
end

% Do the filtering operation
for k=1:lenh
y(k:k+lenx-1)=y(k:k+lenx-1)+h(k)*x;
end
return; % End of the function
```

Exercise 1. Explain in your own words how *filterfir* works.

Exercise 2. Define the impulse response of a 10-point averager as

```
avg10=1/10*ones(1,10)
```

and generate a random signal of length 100 by

```
x=rand(1, 100)
```

Filter the the signal *x* and plot both *x* and its filtered version *y* in the same plot:

```
stem(x); hold on; plot(y, '-or');
```

Explain what you see in your lab report.

The Matlab command *rand* generates random numbers uniformly distributed between 0 and 1.

1.2 Frequency domain (20 minutes)

The frequency response of an LTI system with impulse response h[n] is given by

$$H(\omega) = \sum_{n=-\infty}^{\infty} h[n]e^{-j\omega n}. \qquad (2)$$

For the FIR filter in (1), this becomes

$$H(\omega) = \sum_{n=0}^{M} h[n]e^{-j\omega n}. \qquad (3)$$

The following Matlab code computes the frequency response for a given FIR filter of the form (1) and plots the magnitude and phase responses.

```
function freqr(h)
% FREQR Frequency Response of FIR filter
%
% FREQR(H) plots the magnitude and phase response of an FIR filter
% whose impulse response is H. It is assumed that H represents
% a signal that starts at time zero.
%
% h=[1 2 1];
% freqr(h)

N=200; % number of frequencies to evaluate

% begin calculation
omega=-pi:2*pi/N:pi;
FF=exp(-j*omega'*(0:length(h)-1));
```

```
H=FF*h(:); % The frequency response
mH=abs(H); % The magnitude response
pH=angle(H); % The phase response
% end calculation

% plot the results
subplot(2, 1, 1); plot(omega, mH);
xlabel('\omega'); ylabel('lH(\omega)l'); title('magnitude response');
h=gca; grid on;
set(h, 'XLim', [-pi, pi]); % set x axis limit
set(h, 'XTick', [-pi: 0.25*pi: pi]); % X axis tick positions
set(h, 'FontName', 'symbol'); % prepare to type tick label
set(h, 'XTickLabel', '-pll-0.5pll0ll0.5pllp');% tick label

subplot(2, 1, 2); plot(omega, pH);
xlabel('\omega'); ylabel('\angle H(\omega)'); title('phase response');
h=gca; grid on;
set(h, 'XLim', [-pi, pi]); % set x axis limit
set(h, 'XTick', [-pi: 0.25*pi: pi]); % X axis tick positions
set(h, 'FontName', 'symbol'); % prepare to type tick label
set(h, 'XTickLabel', '-pll-0.5pll0ll0.5pllp');% tick label
return; % End of the function
```

Exercise 3. Explain how the code works between % begin calculation and % end calculation in your own words.

2 Running the GUIs

2.1 The dconvdemo convolution GUI (10 minutes)

Download Matlab zip file containing *dconvdemo* and unzip it, preserving the directory structure (the defaultchoice). Perform the following steps with the *dconvdemo* GUI.

(a) Set the input to a finite-length pulse: x[n] = 2{u[n] – u[n – 10]}
(b) Set the filter impulse response to obtain a five-point averager.
(c) Use the GUI to produce the output signal. Verify with the TA.
(d) When you move the mouse pointer over the index "n" below the signal plot and do a click-hold, you will get a hand tool that allows you to move the n-pointer. By moving the pointer horizontally, you can observe the sliding window action of convolution. You can even move the index beyond the limits of the window and the plot will scroll over to align with "n."
(e) Set the filter's impulse response to a length-10 averager, i.e., h[n] = 0.1 {u[n] – u[n10]}. Use the GUI to produce the output signal.
(f) Set the filters impulse response to a shifted impulse, i.e., h[n] = δ[n–3]. Use the GUI to produce the output signal. Verify with the TA.
(g) Compare the outputs from parts (c), (e) and (f). Notice the different shapes (triangle, rectangle or trapezoid), the maximum values, and the different lengths of the outputs.

2.2 *dltidemo* frequency response GUI (10 minutes)

The *dltidemo* illustrates the sinusoid-IN gives sinusoid-OUT property of discrete-time LTI systems. In this demo, you can change the amplitude, phase and frequency of an input sinusoid, x[n], and you can change the digital filter that processes the signal. Then the GUI will show the output signal, y[n], which is also a sinusoid (at the same frequency). It is possible to see the formula for the output signal, if you click on the Theoretical Answer button located at the bottom-middle part of the window. The digital filter can be changed by choosing different options in the Filter Specifications box in the lower right-hand corner.

Perform the following steps with the *dltidemo* GUI:

(a) Set the input to $x[n] = 1.5 \cos(0.1\pi(n - 4))$

(b) Set the digital filter to be a 9-point averager.

(c) Determine the formula for the output signal and write it in the form:

$$y[n] = A \cos[\omega_0(n - nd)].$$

(d) Using n_d for y[n] and the fact that the input signal had a peak at n = 4, determine the amount of delay through the filter. In other words, how much has the peak of the cosine wave shifted?

3 Filter real signals

Load the datafiles in from a file called *filter_real_signals.mat* by using *load filter_real_signals.mat*. The data file contains two filters and three signals, stored as separate MATLAB variables:

- *x1:* a stair-step signal such as one might find in one sampled scan line from a TV test pattern image.
- *xtv:* an actual scan line from a digital image.
- *x2:* a speech waveform (“oak is strong”) sampled at fs = 8000 samples/second.
- *h1:* the coefficients for a FIR discrete-time filter of the form of (1).
- *h2*: coefficients for a second FIR filter.

After loading the data, use the *whos* function to verify that all five vectors are in your MATLAB workspace.

Exercise 4. (15 minutes) Write a function that does the follows: for a given signal, and two filters, generate a figure with 3 subplots. In the first subplot, plot the given signal. In the second, plot the filtered signal using the given filter number 1. In the third subplot, plot the filtered signal using given filter number 2. For the filtering, use the *filterfir* function provided.

Exercise 5. (15 minutes) Use the function you wrote, process each of the three signals *x1, x2, xtv* using the two filters *h1* and *h2*. Comment on what you observe.

Exercise 6. (15 minutes) Use the Matlab function *sound* and listen to the signal *x2* and its filtered versions, by *h1* and *h2*. Comment on what you hear.

Exercise 7. (15 minutes) For each of the two filters, plot both the magnitude response and phase response using the *freqr* function provided. Comment on the functions of the filters based on their frequency responses, and relate the magnitude responses to the time domain filter results in the previous exercise.