Lab6
FIR Filter Design

Dr. Aly El-Osery

November 8, 2017

In this lab you will learn how to design a FIR filters with different window techniques. You will learn more about the C6713 and its advantages and limitations.

1 Part 1

1. Find the impulse response, $h[n]$, of the following lowpass filter

$$H(e^{j\omega}) = \begin{cases} 1 & |\omega| < \omega_0 \\ 0 & \omega_0 < |\omega| < \pi \end{cases} \tag{1}$$

   where $\omega_0 = \pi/4$.

2. $h[n]$ has infinite number of terms. In order to be to implement the filter we will need to limit the number of terms used. Write a MATLAB code to generate 101 terms of $h[n]$ for $n = -50, \ldots, 50$. Print out a `stem` plot of the truncated impulse response $h[n]$.

3. Store the 101 terms in a text file. Here is a sample of how your file might look like, Program 1. Use MATLAB to automate this process. The extra lines are needed in order to be able to easily use the coefficients in our `C` code.

4. Plot the magnitude and phase responses of $H(e^{j\omega})$ vs. ω.

5. Change the number of taps and comment on the change in the magnitude response of $H(e^{j\omega})$.

6. Up to this point you have used a rectangular window to limit the number of terms of $h[n]$. Now, instead of using a rectangular window use a Hamming window. Print out a `stem` plot of the new coefficients.

7. Given the lowpass filter you have already designed, show how you can use a cos function to modify the filter coefficient to generate a bandpass filter

8. For the filter you designed in Part 7, plot the magnitude and phase responses of $H(e^{j\omega})$ vs. ω. with center frequency $\pi/2$.

9. Store the coefficients in a file.

Sample File 1 coef.h

```
#define N 101
float h[N]={
        0.0063661977,
        0.0045934506,
                    .
                    .
                    .
};
```

2 Part 2

1. Write a code to compute the following equation

$$y[n] = \sum_{k=0}^{N} h[k]x[n-k]$$

where $h[k]$ are the coefficients that you have generated in the prelab for the bandpass filter using a rectangular window and $x[n]$ are the data read from the left channel of the codec. In order to save memory create a circular buffer of length N to store $x[n]$.

2. Output the original signal from the left channel on the left channel without change and output $y[n]$ on the right channel.

3. Connect a function generator to the board input and an oscilloscope to the output. Notice the two channels on the oscilloscope as you are varying the frequency on the function generator.

4. Measure the frequency response (magnitude and phase) of the filter and compare it with the MATLAB results.

5. Now use the coefficients generated using the Hamming window. Measure the frequency response of the filter and compare it to the MATLAB results.

6. What is the limitation on the order of the FIR filter?