Lab 4
FIR Filter Design

September 14, 2012

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

1 Prelab

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

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

   where $\omega_o = \pi/4$

2. *h[n]* has infinite number of terms. In order to implement the filter you will need to limit the number of terms used. Write a MATLAB code to generate 101 terms of *h[n]* for $n = -50, \ldots, 50$. and 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:

`Sample File 1 coef.h`

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

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. 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.

2 Lab

1. Create a new project.

2. Create a DSP/BIOS file and add the necessary files to the project.

3. 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 $x[n]$.

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

5. 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.

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

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