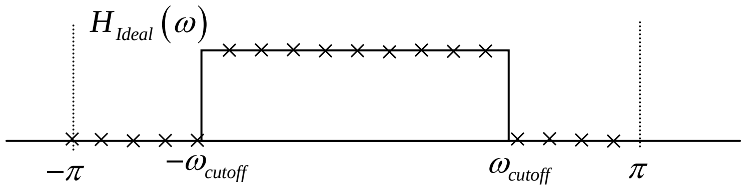
ECE 246 446 –Fall 2014 Computer Project 2 Nov 10, 2014
Due (in Class) Dec 11, 2014

Theory:

The objective of filter design is often to approximate an ideal frequency response. An ideal low pass filter, for example, has a response of the form (the solid line)



One way to generate an FIR filter of a specified length with a frequency response that approximates an ideal response is to choose the filter weights (i.e. the values of the impulse response) so that the frequency response $H(\omega)$ of the FIR filter matches the ideal frequency response at a set of sample frequencies (illustrated by the $\times's$ in the figure).

An FIR filter $h[n]$ of length $2M+1$ that has non-zero terms $h[-M],\ldots,h[0],\ldots,h[M]$ can generally be chosen so that

$$\begin{aligned} H(\omega_1) &= H_{Ideal}(\omega_1) \\ H(\omega_2) &= H_{Ideal}(\omega_2) \\ &\vdots \\ H(\omega_{2M+1}) &= H_{Ideal}(\omega_{2M+1}) \end{aligned} \qquad (*)$$

for any set of $2M+1$ frequencies $\omega_1,\ldots,\omega_{2N+1}$ because these assignments form a system of $2M+1$ equations with $2M+1$ unknowns $h[-M],\ldots,h[0],\ldots,h[M]$.

Problems:

A). The design criteria for ideal filters can be relaxed by allowing gaps between pass bands and stop bands. In that case, an ideal low pass filter is specified by two cutoff frequencies rather than one: an upper frequency ω_p for the pass band and a lower frequency ω_s for the stop band that is greater than or equal to ω_p.

Write a Matlab function with the name and signature

```
function h=  FIR(wp,ws,Np,Ns,delta)
```

that returns an FIR filter $h[-M],\ldots,h[M]$ of length $2M+1$ $(=N_p+N_s)$ with a frequency response that:

1. Matches the frequency response of an ideal low pass filter at N_p frequencies that uniformly sample the pass band $(-\omega_p,\omega_p)$ (including the endpoints).

2. Matches the frequency response of an ideal low pass filter at N_s frequencies that uniformly sample the stop band $(\omega_s,2\pi-\omega_s)$ (including the endpoints).

Notes 1. The impulse response of the filter is found by solving the system of equations (*)

2. To insure that the FIR filter is symmetric about 0 the N_p and N_q arguments must have an odd sum

3. The `delta` argument in the `FIR()` function call is not used until part B)

Next, use the `FIR()` function to compute the filter weights:

```
h0=FIR(pi/4,pi/2.5,6,9,0);
```

This should result in an FIR filter with 15 non-zero coefficients.

Center these filter weights in an array of length 256 that is otherwise 0. The center point of the filter should be located at element 129 of the array.

Plot the frequency response of the filter by taking the real part of the FFT of the 256 point array that contains the filter weights (the imaginary part should be 0). The $\omega=0$ frequency should be located at the center of the horizontal axis.

B). One way to improve the approximation to the ideal low pass filter that was obtained in part A) is to adjust the location of the frequencies where the approximate and ideal responses are equated. The four frequencies that have the greatest influence on the shape of the frequency response are the two bounding frequencies at the ends of the pass band: $\pm\omega_p$, and the two bounding frequencies at the ends of the stop band: $\pm\omega_s$.

Revise the code for the FIR function developed in part A by adjusting the location of the four bounding frequencies as follows:

$$\begin{aligned}&\omega_p\to\omega_p-\delta \qquad -\omega_p\to-\omega_p+\delta\\&\omega_s\to\omega_s+\delta \qquad -\omega_s\to-\omega_s-\delta \quad (equiv: 2\pi-\omega_s\to 2\pi-\omega_s-\delta)\end{aligned}$$

where the value of δ is specified by the **`delta`** argument in the function call.

Then use the revised **`FIR()`** function to compute the filter weights

```
h1=FIR(pi/4,pi/2.5,6,9,.025*pi);
h2=FIR(pi/4,pi/2.5,6,9,.050*pi);
h3=FIR(pi/4,pi/2.5,6,9,.075*pi);
```

Next, compute 256 point frequency responses of these filters in the same way that the frequency response of **`h0`** was computed in part A). Plot all four frequency responses in a single plot using a different color for each curve. Make sure to include a legend that labels the 4 curves.

What conclusions can you draw from these results ?

Note: The Parks-McClellan Algorithm is a well known filter design method that systematically adjusts the locations of the frequencies where the filter response matches the ideal response to obtain the best possible approximation.

C). A number of standard windows have the general form

$$W[m] = \begin{cases} a - b\cos\left(\dfrac{\pi m}{M+1}\right) + c\cos\left(\dfrac{2\pi m}{M+1}\right) & |n| \le M \\ 0 & otherwise \end{cases}.$$

More specifically,

name	*a*	*b*	*c*
boxcar	1	0	0
Hanning	.5	−.5	0
Hamming	.54	−.46	0
Blackman	.42	−.5	.08

Write a Matlab function with the name and signature

```
function h=  Window(M,a,b,c)
```

that returns the filter weights $h[-M],\ldots,h[M]$ for the given parameters.

Next, compute the filter weights

```
w_box_car   =  Window(7,1,0,0);
w_Hanning   =  Window(7,.5,-.5,0);
w_Hamming   =  Window(7,.54,-.46,0);
w_Blackman  =  Window(7,.42,-.5,.08);
```

The length of each of these filters should be 15. Generate a Cartesian plot of all 4 window functions using a different color for each window. Make sure to include a legend that labels the 4 curves.

Center each window in an array of length 256 that is otherwise 0. The center point of the window should be located at element 129 of the array.

These 0-padded windows may be used to form impulse responses that approximate an ideal low pass filter with $\omega_s = \omega_p = \left(\pi/4 + \pi/2.5\right)/2$, as follows:

1. Generate an array of 256 frequency samples of the ideal frequency response.
2. Take the inverse FFT of this response to obtain an impulse response of length 256
3. Multiply the 256 term impulse response by one of the windows to limit the number of terms in the impulse response to the number of terms in the window.

Compute the frequency responses of all 4 of the impulse responses that result from applying this procedure to the 4 windows. In addition, compute the frequency response of the filter

```
hFIR=FIR(pi/4,pi/2.5,6,9,.098*pi);
```

Generate a Cartesian plot of the frequency responses for the filters obtained from all 4 windows and also the frequency response of the filter **`hFIR.`** Use a different color for each response and make sure to include a legend that labels the 5 curves.

D). How do the frequency responses obtained from the four windows compare with each other, and also with the FIR filter ?