STANFORD UNIVERSITY
DEPARTMENT of ELECTRICAL ENGINEERING

EE 102B Spring 2013
Lab #4: Filter Design of FIR Filters

Assigned: April 26, 2013
Due Date: May 3, 2013

Remember that you are bound by the Stanford University Honor Code. Your submitted work must be your own original work, not the result of a collaborative effort. If you have difficulties with any of the MATLAB programming, consult one of the course staff or a classmate, but do the suggested experiments and write up your answers on your own.

1 Warm-Up

1.1 Objective

The goal of this lab is to learn some methods for designing practical FIR filters in MATLAB. These filters will have a finite number of coefficients, and a frequency response that approximates an ideal frequency response shape.

1. *Windowing:* The concept of windowing is widely used in DSP when dealing with finite-length signals.

2. *Filter Specifications (aka "Specs"):* The quality of a designed filter is measured by how closely the actual response matches the desired ideal response. Often the desired match is set prior to the actual filter design step by drawing a tolerance region around the ideal filter shape. Then the minimum-order filter that fits inside the tolerance region is designed. When the frequency response fits inside the tolerance region, the design is said to "meet the specs."

3. *Design Methods:* Two very common approaches to FIR filter design are *windowing* and computer optimization. The `filterdesign` GUI in *SP-First* can do both.

1.2 Overview

There are many ways to approximate an ideal frequency response with a practical filter. For FIR filters the frequency response $H(e^{j\hat{\omega}})$ is a function of $\hat{\omega}$ that summarizes a LTI system's response to inputs such as complex exponentials and sinusoids. The MATLAB function `freqz.m` is used to compute samples of the frequency response in the *frequency domain*.[1] A filter design method produces filter coefficients $\{b_k\}$ for the time-domain implementation of the FIR filter as a difference equation

$$y[n] = b_0x[n] + b_1x[n-1] + b_2x[n-2] + ... + b_Mx[n-M]$$

so the `freqz` function is an essential step for making plots of the magnitude and phase of the designed frequency response and assessing how closely it matches the desired ideal response.

[1] If you are working at home and do not have the function `freqz.m`, there is a substitute available called `freekz.m`. You can find it in the *SP-First Toolbox*.

1.3 Frequency Response of FIR Filters

The general form of the frequency response for an M-th order FIR linear time-invariant system with filter coefficients $\{b_k\}$ is

$$H(e^{j\hat{\omega}}) = \sum_{k=0}^{M} b_k e^{-j\hat{\omega}k} = \sum_{k=0}^{M} h[k] e^{-j\hat{\omega}k} \tag{1}$$

In addition, recall that the impulse response $h[n]$ for the FIR filter has values equal to the filter coefficients. Also, note that M is the "order" of the filter, and the length of the impulse response of the filter is $M + 1$.

1.3.1 MATLAB Function for Frequency Response

MATLAB has a built-in function for computing the frequency response of a discrete-time LTI system. The following MATLAB statements show how to use `freqz` to compute and plot both the magnitude (absolute value) and the phase of the frequency response of an FIR filter whose impulse response is a truncated sinc function. The plots[2] are a function of $\hat{\omega}$ in the range $-\pi \leq \hat{\omega} \leq \pi$:

```
M = 22;           %-- Filter Order is even
nn = 0:M;         %-- "n index" vector
nsh = n - M/2;    %-- put main lobe in center of h[n]
wb = 0.32*pi;     %-- "Bandwidth" parameter of sinc
bb = sin(wb*nsh)./(pi*nsh);  %-- Filter Coefficients
bb(find(nsh==0)) = wb/pi;   %-- fix the divide by zero
ww = -pi:(pi/500):pi;   %-- omega hat frequency vector
H = freqz(bb, 1, ww);   %<--freekz(bb,1,ww) is an alternative
subplot(2,1,1);
plot(ww, abs(H)), grid on
subplot(2,1,2);
plot(ww, angle(H)), grid on
xlabel('Normalized Radian Frequency')
```

For FIR filters, the second argument of `freqz( _, 1, _ )` must always be equal to `1`. The frequency vector `ww` should cover an interval of length 2π for $\hat{\omega}$, and its spacing must be fine enough to give a smooth curve for $H(e^{j\hat{\omega}})$. The MATLAB variable for frequency response is capital `H` to be consistent with $H(e^{j\hat{\omega}})$.
Note: The filter coefficients $\{b_k\}$ are indexed from $k = 0$ to $k = M$ in (1), but since MATLAB's vector indexing starts at one, `bb(1)` in MATLAB $= b_0 = h[0]$ in (1).

1.4 Windowing

The concept of windowing is widely used in signal processing. The basic idea is ***to extract a finite section*** of a very long signal $x[n]$ ***via multiplication*** by a finite-length sequence $w[n]$; i.e., $w[n]x[n]$. For example, consider the simplest window function, which is the L-point *rectangular window* defined as

$$w_r[n] = \begin{cases} 1 & 0 \leq n \leq L-1 \\ 0 & \text{elsewhere} \end{cases} \tag{2}$$

The important idea is that the product $w_r[n]x[n + n_0]$ will extract L values from the signal $x[n]$ starting

[2] If the output of the `freqz` function is not assigned, then plots are generated automatically; however, the magnitude is given in decibels which is a logarithmic scale. For linear magnitude plots a separate call to `plot` is necessary.

at $n = n_0$. Thus the following are equivalent

$$w_r[n]x[n+n_0] = \begin{cases} 0 & n < 0 \\ \cancelto{1}{w_r}[n]x[n+n_0] & 0 \le n \le L-1 \\ 0 & n \ge L \end{cases} \tag{3}$$

The name *window* comes from the idea that we can only "see" L values of the signal $x[n+n_0]$ within the window interval when we "look" through the window. Multiplying by $w[n]$ is looking through the window. When we change n_0, the signal shifts, and we see a different length-L section of the signal.

All the nonzero values of the window function do not have to be ones as in (2), but they should be positive. For example, the L-point *Hamming window* is defined as

$$w_m[n] = \begin{cases} 0.54 - 0.46\cos(2\pi n/(L-1)) & 0 \le n \le L-1 \\ 0 & \text{elsewhere} \end{cases} \tag{4}$$

The MATLAB function `hamming(L)` will generate a vector with values given by (4). A stem plot of the Hamming window would show that the values are larger in the middle and taper off near the ends.

(a) Make a stem plot of the Hamming window for $L = 23$, over the index range $0 \le n \le L-1$.

(b) Determine the maximum value of the window and the index location of the maximum. Also, determine the values of the window at $n = 0$, $n = 11$, and $n = 22$.

(c) The Hamming window is said to have *even symmetry.* What does this mean in this case?

1.5 Ideal Filters and Practical Filters

Ideal Filters are given by their frequency response, consisting of *perfect* passbands and stopbands. The ideal filters cannot be FIR filters because there is no finite set of filter coefficients whose DTFT will be the ideal frequency response. The impulse response of the ideal lowpass filter (LPF) is an infinitely long sinc function as shown by the following DTFT pair:

$$h_{\text{ILPF}}[n] = \frac{\sin(\hat{\omega}_c n)}{\pi n} \quad \Longleftrightarrow \quad H_{\text{ILPF}}(e^{j\hat{\omega}}) = \begin{cases} 1 & |\hat{\omega}| \le \hat{\omega}_c \\ 0 & \hat{\omega}_c < |\hat{\omega}| \le \pi \end{cases} \tag{5}$$

where $\hat{\omega}_c$ is the normalized *cutoff frequency* of the ideal LPF, which separates the passband from the stopband.

In the `dltidemo` GUI, you can choose ideal lowpass filters (LPF), highpass filters (HPF) and bandpass filters (BPF) because the GUI does not use the impulse response. The ideal LPFs and HPFs have one parameter for the cutoff frequency. The ideal BPF has a parameter for center frequency which determines where the band is located; its bandwidth (in the `dltidemo` GUI) is always 0.4π. All the ideal filters have an additional parameter for the slope of the phase of $H(e^{j\hat{\omega}})$.

Practical Filters are causal length-L FIR filters whose filter coefficients are chosen so that the resulting frequency response will closely approximate the desired frequency response of an ideal filter. The process of choosing the filter coefficients is called *filter design.* The practical FIR filters shown in `dltidemo` were designed using MATLAB's `fir1` function for digital filter design. The GUI offers length-15 LPFs and HPFs, and length-21 BPFs. The LPF and HPF designs depend on specifying one parameter for the cutoff frequency which lies midway between the non-ideal passband and stopband. The BPF requires two parameters: one for center frequency which determines where the passband is located, and other for the width of the passband.

In the `dltidemo` GUI, the default BPF cutoffs are $\pm 0.2\pi$ from the center frequency, so only the center frequency can be changed. These practical filters do not match ideal filters exactly, and this is readily apparent at the cutoff frequencies where the frequency response has a magnitude of 0.5.

1.5.1 Truncate the Ideal Impulse Response

One simple approach to designing a practical FIR filter is to truncate the impulse response of an ideal filter. This can be accomplished with a window function, so we usually say that the practical FIR filter has an impulse response that is a *windowed* version of the ideal impulse response. For example, we could make a length-23 FIR lowpass filter by taking the center portion of the sinc function:

$$h_1[n] = w_r[n] h_{\text{ILPF}}[n-11] = \begin{cases} \dfrac{\sin(\hat{\omega}_c(n-11))}{\pi(n-11)} & 0 \le n \le 22 \\ 0 & \text{elsewhere} \end{cases} \tag{6}$$

where $\hat{\omega}_c$ is the normalized cutoff frequency and $w_r[n]$ is a 23-point rectangular window. The sinc function must be time-shifted to put its peak in the middle of the window at $n = 11$, because we want the system defined by $h_1[n]$ in (6) to be causal. It is common that the ideal impulse response is time shifted by an amount equal to half the window length so that the resulting filter will have linear phase.

(a) For the time-windowed sinc function in (6), set $\hat{\omega}_c = 0.7\pi$. Then make a stem plot of the impulse response $h_1[n]$.

(b) Use MATLAB to determine (samples of) the DTFT of $h_1[n]$ and make a plot of the DTFT magnitude, $|H_1(e^{j\hat{\omega}})|$.

(c) We could experiment with different window functions. Change the window to the 23-point Hamming window, and define a new impulse response as

$$h_2[n] = w_{\text{HAMM}}[n] \left(\frac{\sin(\hat{\omega}_c(n-11))}{\pi(n-11)} \right)$$

Plot the impulse response.

(d) Use MATLAB to determine (samples of) the DTFT of $h_2[n]$ and make a plot of the DTFT magnitude, $|H_2(e^{j\hat{\omega}})|$. Explain why this "practical filter" is a better LPF than $H_1(e^{j\hat{\omega}})$ when evaluated as an approximation to the ideal LPF.

1.6 GUI for Filter Design

The *SP-First* GUI called `filterdesign` illustrates several filter design methods for LPF, BPF and HPF filter. The interface is shown in Fig. 1. Both FIR and IIR filters can be designed, but we will only be interested in the FIR case which would be selected with `FIR` button in the upper right. The default design method is the *Window Method* using a Hamming window. The window type can be selected from the drop-down list in the lower right. To specify the design it is necessary to set the order of the FIR filter and choose one or more cutoff frequencies; these parameters can be entered in the edit boxes. The specification of one or more cutoff frequencies (f_c) of the ideal filter must be entered using continuous-time frequency (in Hz), along with a sampling rate (f_s, also in Hz). In normalized frequency, the ideal cutoff frequency is $\hat{\omega}_c = 2\pi(f_c/f_s)$.

The plot initially shows the frequency response magnitude on a linear scale, with a frequency axis in Hz. Clicking on the y-axis label Magnitude will toggle the magnitude scale to a log scale in dB. Clicking

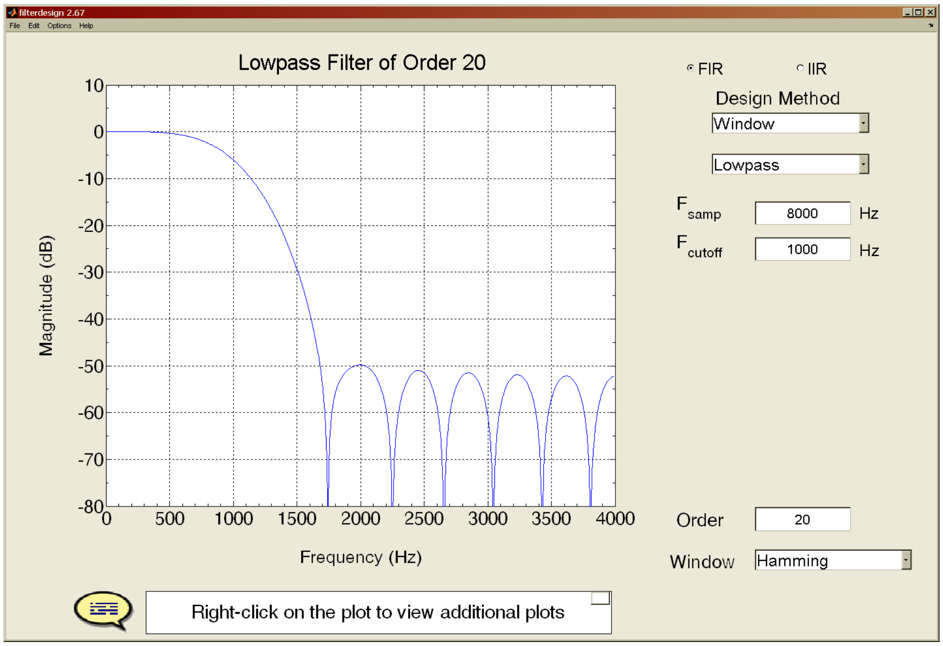


Figure 1: Interface for the `filterdesign` GUI. When the Filter Choice is set to `FIR`, many different window types can be selected, including the Hamming window and the Rectangular window (i.e., no tapering at the ends).

on the word $\boxed{\text{Frequency}}$ will toggle the frequency axis to normalized frequency $\hat{\omega}$, and also let you enter the ideal cutoff frequency using $\hat{\omega}_c$. Recall that the normalized ideal cutoff frequency is $\hat{\omega}_c = 2\pi(f_c/f_s)$. The plotting region can also show the phase response of $H(e^{j\hat{\omega}})$, or the impulse response of the filter $h[n]$. Right click on the plot region to get a menu.

The filter coefficients can be "exported" from the GUI by using the menu `File->Export Coeffs`. To have some filters for comparison, redo the designs from the Section 1.5.1, and export the filter coefficients to the workspace under unique names. Then you can make your own plot of the frequency response in MATLAB using the `freekz` function (or `freqz`) followed by a `plot` command.

The $\boxed{\text{Options}}$ menu provides zooming and a grid via `Options->Zoom` and `Options->Grid`.

1.7 Design Filters with the `filterdesign` GUI

For practice, use the `filterdesign` GUI to design two lowpass FIR filters with order $M = 22$ (or length $L = 23$). Use a cutoff frequency of $\hat{\omega}_c = 0.32\pi = 2\pi(1600/10000)$. Create one using a Hamming window, the other with a Rectangular window which should give a result like Fig. 2. Right click on the plot to see options for displaying the impulse response either windowed or unwindowed. The unwindowed version just displays the truncated sinc function, i.e., rectangular windowing.

1.7.1 Passband Defined for the Frequency Response

Frequency-selective digital filters, e.g., LPFs, BPFs and HPFs, have a frequency response magnitude that is close to one in some frequency regions, and close to zero in others. For example, the plot in Fig. 2 is a lowpass filter whose magnitude is close to one when $0 \leq \hat{\omega} < 0.883$. This region where the magnitude is close to one is called the ***passband*** of the filter. It will be useful to have a precise definition of the passband edges, so that the passband width can be measured and we can compare different filters.

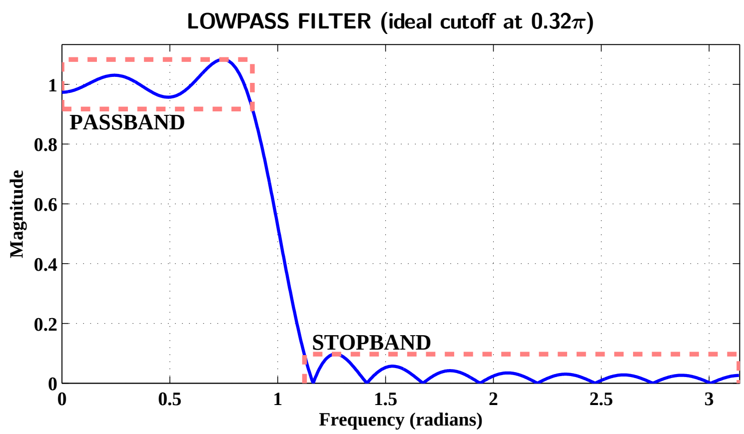


Figure 2: Passband and stopband defined for a typical lowpass filter. This particular filter is a length-23 FIR filter designed with a rectangular window and a sinc function with a cutoff frequency of $\hat{\omega}_c = 0.32\pi$. The passband and stopband ripples are defined to be 0.1, from which the passband and stopband edges can be measured. The approximate value of the passband edge is $\hat{\omega}_p = 0.281\pi \approx 0.883$; the stopband edge, $\hat{\omega}_s = 0.358\pi \approx 1.125$.

(a) From the plot of the magnitude response, e.g, in MATLAB or in the `filterdesign` GUI, it is possible to determine the set of frequencies where the magnitude is very close to one, as defined by $\big||H(e^{j\hat{\omega}})| - 1\big|$ being less than δ_p. This deviation from one is the maximum approximation error in the passband. It is called the *passband ripple*. A common choice for the passband ripple is between 0.01 and 0.1, i.e., 1% to 10%. The set of frequencies in a passband should be a region of the form $\hat{\omega}_1 \le \hat{\omega} \le \hat{\omega}_2$.

(b) For a lowpass filter, the passband region extends from $\hat{\omega} = 0$ to $\hat{\omega}_p$, where the parameter $\hat{\omega}_p$ is called the *passband edge*. For the two LPFs designed in Section 1.7 determine an *accurate estimate* of $\hat{\omega}_p$ assuming a passband ripple (δ_p) of 0.1 for the Rectangular window case, and $\delta_p = 0.01$ for the Hamming window case.[3] Compare these *actual passband edges* to the design parameter $\hat{\omega}_c$ which is called the *cutoff frequency*.
Note: There is often confusion that $\hat{\omega}_c$ and $\hat{\omega}_p$ are the same, but after doing a few examples it should become clear that is not the case.

1.7.2 Stopband Defined for the Frequency Response

When the frequency response (magnitude) of the digital filter is close to zero, we have the ***stopband*** region of the filter. In the lowpass filter example of Fig. 2, the magnitude is close to zero when the frequency $1.125 \le \hat{\omega} \le \pi$, i.e., high frequencies. When the frequency response of a LPF is plotted only for nonnegative frequencies, the stopband will be a region of the form $\hat{\omega}_s \le \hat{\omega} \le \pi$. The parameter $\hat{\omega}_s$ is called the *stopband edge*. We can make a precise measurement of the *stopband edge* as follows:

(a) For the lowpass filters from Section 1.7, zoom in on the plot of frequency response magnitude in the `filterdesign` GUI to measure the stopband edge, or use a dB magnitude plot. Then determine the set of frequencies where the magnitude is nearly zero, as defined by $|H(e^{j\hat{\omega}})|$ being less than $\delta_s = 0.1$ for the Rectangular window case, and less than $\delta_s = 0.01$ for the Hamming window design.

[3]The `filterdesign` GUI has a zoom capability (`Options->zoom`), and the grid can be turned on (`Options->grid`). Also, when the pointer is placed to hover over the frequency response the coordinates are read from the plot.

(b) Compare the values of $\hat{\omega}_s$ found in the previous part to the design parameter $\hat{\omega}_c$ (the cutoff frequency).

1.7.3 Transition Zone of the LPF

The difference between the stopband edge and the passband edge is called the *transition width* of the filter: $\Delta\hat{\omega} = \hat{\omega}_s - \hat{\omega}_p$. The smaller the transition width, the better the filter because it is closer to the *ideal filter* which has a transition width of zero.

(a) For the two lowpass filters from Section 1.7, determine the transition width.
Note: Comment on the statement, "when comparing equal-order FIR filters, the one with smaller transition width will have larger ripples."

(b) Design two new LPFs that have the same cutoff frequency, $\hat{\omega}_c = 0.32\pi$, but twice the order, i.e., $M = 44$. Repeat the measurement of $\hat{\omega}_p$, $\hat{\omega}_s$ and $\Delta\hat{\omega}$ for these two LPFs.

(c) Compare to the values of $\Delta\hat{\omega}$ from part (a). When the order doubles, describe what happens to the transition width.

1.7.4 Summary of Filter Specifications

The foregoing discussion of ripples, bandedges, and transition width can be summarized with the tolerance scheme shown in Fig. 3. The filter design process is to approximate the ideal frequency response very closely. Once we specify the desired ripples and bandedges, we can draw a template around the ideal frequency response. An acceptable filter design would be an FIR filter whose magnitude response lies entirely within the template. The length-23 FIR filter shown in Fig. 3 meets the specs, but if you designed a length-19 filter it would have a transition width that is greater than $\Delta\hat{\omega} = 0.08\pi$.

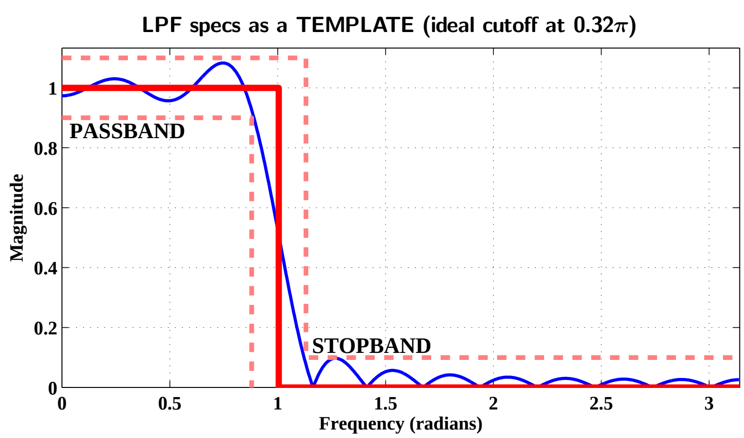


Figure 3: Tolerance scheme drawn around an ideal LPF with a cutoff frequency of $\hat{\omega}_c = 0.32\pi$. Dashed lines indicate the maximum allowable deviation from the ideal LPF. The template uses $\hat{\omega}_p = 0.28\pi$, $\hat{\omega}_s = 0.36\pi$, and $\delta_p = \delta_s = 0.1$. The actual FIR filter shown is the length-23 FIR filter from Fig. 2 which just barely meets these specs.

2 Lab Exercise

The objective of the lab exercise is to design FIR filters that can be lowpass, highpass or bandpass filters. The exercises will involve using the *SP-First* GUI `filterdesign` in which FIR filters are designed via the window method, or with a computer optimization technique.

2.1 Design Two Lowpass Filters

Design two lowpass FIR filters with $M = 30$ and $\hat{\omega}_c = 0.4\pi = 2\pi(2000/10000)$, one using a Hamming window, the other with a Rectangular window. For the measurement of passband and stopband edges, there are two approaches: use the `filterdesign` GUI and read numbers from the plot, zooming when necessary, or export the filter coefficients from the GUI and use MATLAB to make plots of the magnitude of the frequency response using `freekz` (or `freqz`) and `plot`. In MATLAB zooming would be more precise and reliable because the frequency sampling can be specified in the call to `freekz`.

(a) For the filter obtained with the rectangular window, determine an *accurate measurement* of the passband edge $(\hat{\omega}_p)$ assuming the passband ripple specification is $\delta_p = 0.1$, i.e., 1 ± 0.1.

(b) For the filter obtained with the rectangular window, determine an *accurate measurement* of the stopband edge $(\hat{\omega}_s)$ assuming the stopband ripple specification is $\delta_s = 0.1$.

(c) For the filter obtained with the Hamming window, determine an *accurate measurement* of the passband edge $(\hat{\omega}_p)$ assuming the passband ripple specification is $\delta_p = 0.01$, i.e., 1 ± 0.01.

(d) For the filter obtained with the Hamming window, determine an *accurate measurement* of the stopband edge $(\hat{\omega}_s)$ assuming the stopband ripple specification is $\delta_s = 0.01$.

(e) *Question:* is the cutoff frequency half way between $(\hat{\omega}_p)$ and $(\hat{\omega}_s)$ for both filters?

2.2 Transition Zone of the LPF

The difference between the stopband edge and the passband edge is called the *transition width* of the filter: $\Delta\hat{\omega} = \hat{\omega}_s - \hat{\omega}_p$. The smaller the transition width, the better the filter because it is closer to the *ideal filter* which has a transition width of zero.

(a) For the two lowpass filters from Section 2.1, determine the transition width.

(b) *Comment:* "when comparing two M^{th} order filters, the one with a smaller transition width will have larger ripples." Verify that this is a true statement for the two filters.

(c) Design a new Hamming-window LPF that has the same cutoff frequency, $\hat{\omega}_c = 0.4\pi$, but twice the order, i.e., $M = 60$. Repeat the measurement of $\hat{\omega}_p$, $\hat{\omega}_s$ and $\Delta\hat{\omega}$ for this LPF.

(d) Compare the values of $\Delta\hat{\omega}$ from parts (a) and (c); when the order doubles, describe what happens to the transition width. Use this observation to explain that the following Hamming window design formula should be true

$$\Delta\hat{\omega} = \frac{C}{M} \tag{7}$$

and find the value of the constant C.

2.3 Design FIR Filter to Meet Given Specifications

Filter design for lowpass filters involves five parameters: two band edges, ripple heights in two bands, and the filter order. There is a sixth factor, which is the type of filter such as a Hamming windowed FIR filter. A typical design problem would be stated as follows: given the band edges and ripple heights, determine the *minimum order* filter that will meet the specs.

(a) Suppose that you are given $\hat{\omega}_p = 0.68\pi$, $\hat{\omega}_s = 0.72\pi$, $\delta_p = 0.01$, and $\delta_s = 0.01$. Make a sketch of an ideal filter and a template that looks like Fig. 3. Label everything carefully and completely.

(b) Use your Hamming window design formula (Eq. (7) from the previous section) to predict the filter length (L) that will be needed to meet the specs. Recall that $L = M + 1$.

(c) Design the Hamming-windowed FIR filter with the predicted order. Determine the correct value to use for the cutoff frequency. Make a copy of the frequency response plot using your computer's "screen shot" capability, and attach a printout of the figure to your report. Explain why the resulting frequency response does or does not meet the given specs.

2.4 Filter Design via Optimization

Many different methods have been developed for filter design via mathematical optimization. One of the widely used methods is `firpm` in MATLAB. For designing a LPF, it uses the following two step process:

1. Use the desired specifications for $\hat{\omega}_p$, $\hat{\omega}_s$, δ_p, and δ_s to estimate the filter order (M) that will be needed. This is done with the MATLAB function `firpmord`.

2. Use the outputs from `firpmord` as inputs to the function `firpm` to run the optimization and obtain the FIR filter coefficients that should meet the specs on δ_p and δ_s. In effect, the inputs to `firpm` are $\hat{\omega}_p$, $\hat{\omega}_s$, M, and the ratio δ_p/δ_s.

For the calling arguments of these functions, do `help firpmord` and `help firpm`.

(a) Suppose that you are given $\hat{\omega}_p = 0.68\pi$, $\hat{\omega}_s = 0.72\pi$, $\delta_p = 0.05$, and $\delta_s = 0.01$. Notice that the specs on δ_p and δ_s can be different, unlike the window method that always has $\delta_p = \delta_s$.
Carry out the two design steps above to get the filter order M and the filter coefficients $\{b_k\}$.

(b) Make a plot of the impulse response of the filter. Recall that the filter coefficients of the FIR filter are the values of the impulse response. Also, the DTFT of the impulse response is the frequency response of the FIR filter.

(c) Make a plot of the frequency response magnitude versus $\hat{\omega}$. Then check whether or not the ripple specs (δ_p, δ_s) have been met. The band edges should definitely be correct because $\hat{\omega}_p$ and $\hat{\omega}_s$ are inputs to `firpm`.

(d) If the ripple specs are not met with the predicted order, then increase the order by one and try again. A higher order such as $M + 1$ or $M + 2$ should meet the specs.

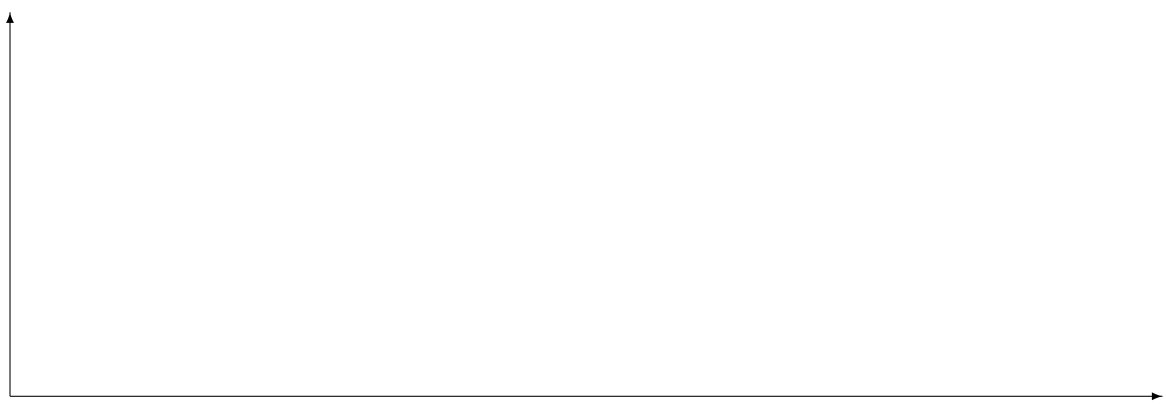
(e) The phase of this FIR filter will be linear phase. Determine the slope of the linear phase.

Lab #4
EE 102B   Spring-2013
LAB REPORT SUMMARY SHEET

Print this page, fill it out, and turn it in as part of your Lab #4 writeup.

Name: ______________________________ SUID: _______________ Date: ________

Part	Observations
2.1(a,b)	Passband and stopband edges of FIR filter designed with rectangular window. Measured values:
2.1(c,d)	Passband and stopband edges of FIR filter designed with Hamming window. Measured values:
2.2(a)	Transition widths of FIR filters designed with rectangular window and Hamming window.
2.2(c)	Passband and stopband edges, and transition width, of longer FIR filter designed with Hamming window. Measured values:
2.2(d)	Dependence of transition width on filter order for FIR filter designed with Hamming window. Find C in $\Delta\hat{\omega} = C/L$.
2.3(a)	Make a sketch of ideal filter including a tolerance template like Fig. 3. *Draw the sketch on the axes below.*
2.3(b)	Predict length of the FIR filter to be designed with the Hamming window, using $\Delta\hat{\omega} = C/L$. $L =$?
2.3(c)	Give the value of $\hat{\omega}_c$. Zoom in on passband stopband regions to verify. Attach a copy of a plot of the frequency response of filter.



Lab #4

EE 102B Spring-2013

LAB REPORT SUMMARY SHEET

Print this page, answer any questions and attach any requested plots.

Name: ____________________________ SUID: ______________ Date: _______

Part 2.4 Design an FIR filter via optimization with `firpm`.

(a) The specs are $\hat{\omega}_p = 0.68\pi$, $\hat{\omega}_s = 0.72\pi$, $\delta_p = 0.05$, and $\delta_s = 0.01$.
Carry out the two design steps to get the predicted filter order M and the filter coefficients $\{b_k\}$.

Note: If you had to increase the filter order to meet the the ripple specs (δ_p, δ_s), then give the value of M that was used, as well as the predicted value of M.

(b) Hand in a plot of the impulse response of the FIR filter.

(c,d) Hand in plot(s) of the frequency response magnitude versus $\hat{\omega}$. Verify that all the specs have been met; showing zoomed plots of the passband and stopband regions would be the best way to verify.

(e) The phase of this FIR filter will be linear phase. Determine the slope of the linear phase.