University of Illinois at Urbana-Champaign
Department of Electrical and Computer Engineering

ECE 311: Digital Signal Processing Lab

LAB 6: **Finite-Length Impulse Response Filter Design**

Fall 2015

1 Objective

The objective of this lab is to help you design finite-length impulse response (FIR) filters.
Key Concepts:

- Windowing method
- Frequency sampling method
- Parks-McClellan method

2 Background

2.1 Review Generalized Linear Phase FIR Filters

Consider a filter with a real-valued, length N impulse response $\{h_n\}_{n=0}^{N-1}$ which has a DTFT of the form

$$H_d(\omega) = R(\omega)e^{j(\alpha - \omega M)} \tag{1}$$

where $R(\omega)$ is real-valued, α is a real-valued constant, and $M = \frac{N-1}{2}$. Note that

$$R(\omega) = \begin{cases} |R(\omega)| & \text{if } R(\omega) > 0 \\ e^{\pm j\pi}|R(\omega)| & \text{if } R(\omega) < 0. \end{cases}$$

Thus, a sign change in $R(\omega)$ results in a $\pm\pi$ jump in the phase of $H_d(\omega)$. Phase jumps of π may not be 'unwrapped' like jumps of 2π, but the filter $H_d(\omega)$ will still have generalized linear phase.

2.1.1 Type I Generalized Linear Phase

An FIR filter with real-valued impulse response $\{h_n\}_{n=0}^{N-1}$ as specified above has type I generalized linear phase if and only if it has even symmetry given by

$$h[n] = h[N-1-n] = \begin{cases} n = 0, 1, \ldots, N/2 - 1 & N \text{ even} \\ n = 0, 1, \ldots, (N-1)/2 & N \text{ odd.} \end{cases} \tag{2}$$

Considering Eq. 1, it may be proved that for type I generalized linear phase $\alpha = 0$ and $R(\omega)$ must be an even function (e.g. $R(\omega) = R(-\omega)$, a sum of cosine functions).

An example of type I generalized linear phase for an FIR filter is shown in Fig. 1.

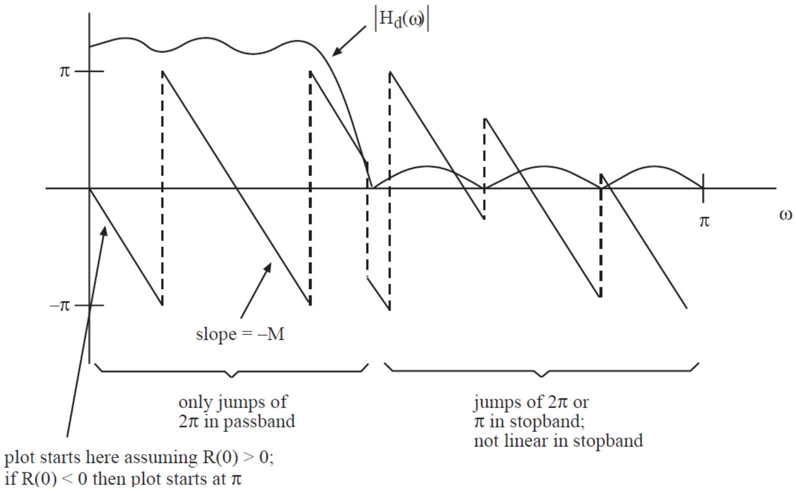


Figure 1

2.1.2 Type II Generalized Linear Phase

An FIR filter with real-valued impulse response $\{h_n\}_{n=0}^{N-1}$ has type II generalized linear phase if and only if it has odd symmetry given by

$$h[n] = -h[N-1-n] = \begin{cases} n = 0, 1, \ldots, N/2 - 1 & N \text{ even} \\ n = 0, 1, \ldots, (N-1)/2 & N \text{ odd} \end{cases} \tag{3}$$

Considering Eq. 1, it may be proved that for type II generalized linear phase $\alpha = \pm\frac{\pi}{2}$ and $R(\omega)$ must be an odd function (e.g. $R(\omega) = -R(-\omega)$, a sum of sine functions). If N is odd, $h[M]$ $(M = (N-1)/2)$ must be zero to satisfy the odd symmetry condition.

2.2 FIR Filter Design using Windowing Method

Let us begin with the ideal lowpass filter, shown below (left), with the desired frequency response $D(\omega)$. We might choose the filter coefficients to be the inverse DTFT of the ideal lowpass filter, $d[n] = \frac{\omega_c}{\pi}\text{sinc}[\omega_c n]$, as shown below (right).

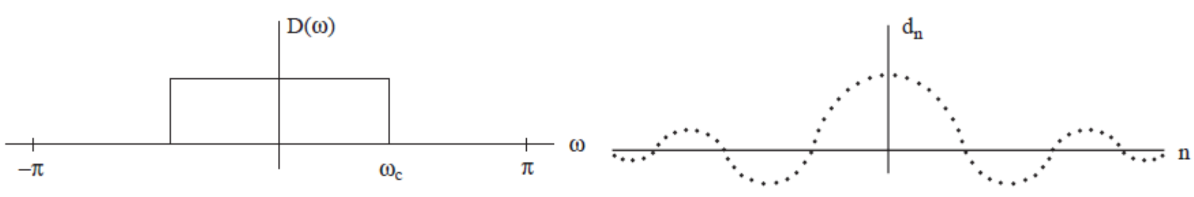


Figure 2

However, the sequence $d[n]$ is infinitely long and non-causal. To obtain an FIR filter, it is necessary to shift the sequence The shift of the impulse response adds linear phase to the complex frequency response. The truncation causes sidelobes in the magnitude response, known as Gibb's phenomenon. These sidelobes can be reduced by windowing the impulse response (in the time domain), at the cost of increasing the transition bandwidth in the magnitude frequency response.

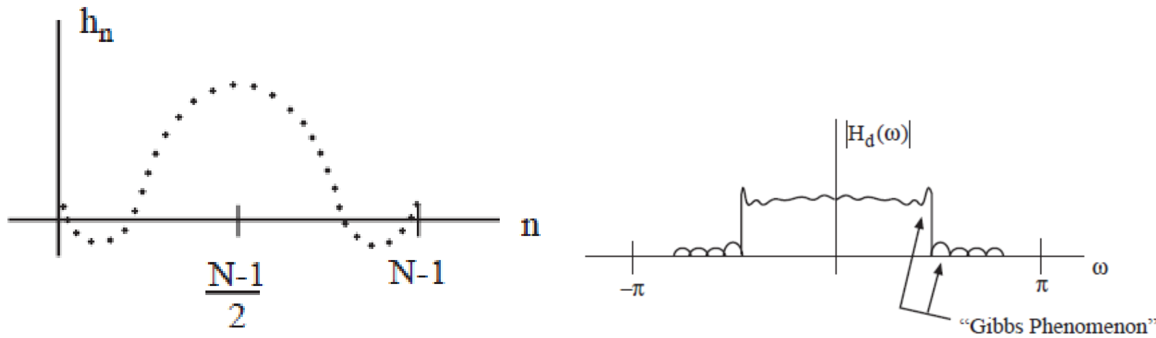


Figure 3

To design a generalized linear phase filter of coefficients $\{h_n\}_{n=0}^{N-1}$ such that $|H_d(\omega)|$ approximates the desired frequency response:

1. Let $G_d(\omega) = D(\omega)e^{-jM\omega}$, where $M = \frac{N-1}{2}$
2. Find $g[n] = \text{DTFT}^{-1}\{G_d(\omega)\}$
3. Let $h[n] = w[n]g[n]$, where $w[n]$ is the window function (e.g. Hamming window centered at $n = M$)

2.3 FIR Filter Design using Frequency-Sampling Method

As the name implies, this design method is based on sampling the ideal frequency response. Let $G_d(\omega)$ be the ideal frequency response, with linear phase. We then force $H_d(\omega)$ to agree with $G_d(\omega)$ at frequency-sampled points $\omega = \frac{2\pi m}{N}$ for $0 \leq m \leq N-1$. The impulse response is determined by the inverse DFT

$$h[n] = \frac{1}{N}\sum_{m=0}^{N-1} G_d(\tfrac{2\pi m}{N})e^{j\frac{2\pi mn}{N}} \quad 0 \leq n \leq N-1.$$

The ideal and frequency-sampled filters are shown in Fig. 4.

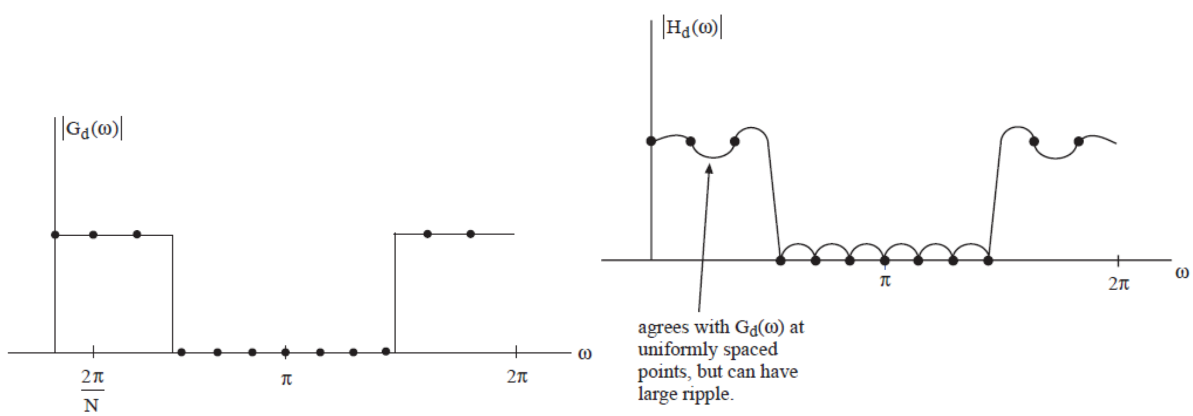


Figure 4

Note that the two frequency responses $H_d(\omega)$ and $G_d(\omega)$ are only identical for the sampled values. The implementable frequency response $H_d(\omega)$ can have significant ripples. One way to reduce the ripples is to allow for a variable transition sample in the transition band, whose amplitude can be chosen to utilize tradeoffs between ripple amplitudes and the transition bandwidth. Such a sample is represented with an 'x' in the figure below.

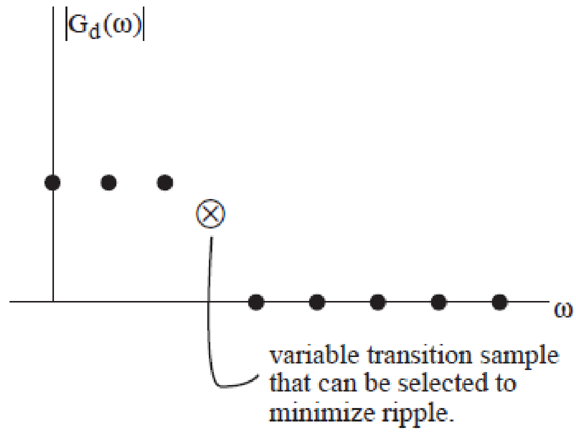
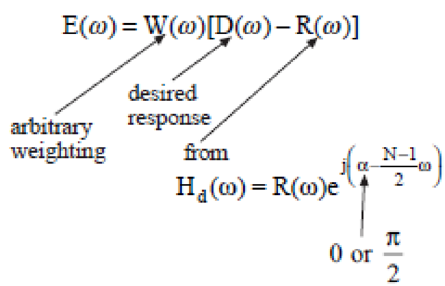


Figure 5

2.4 FIR Filters designed by the Parks-McClellan Method

The Parks-McClellan method minimizes the maximum deviations from the desired frequency response $D(\omega)$ using a weighted error metric



Let ω_p and ω_s be the passband and stopband cutoff frequencies, respectively. The Parks-McClellan algorithm finds the coefficients $\{h_n\}_{n=0}^{N-1}$ that minimize

$$\max|E(\omega)|,\ 0 \leq \omega \leq \omega_p,\ \omega_s \leq \omega \leq \pi.$$

The user specifies N, ω_p, ω_s, and the weighting factors W_p, and W_s. The designed filter has equiripple behavior with ripples δ_p and δ_s in the passband and stopband, respectively. The weighting factors can be chosen based on the relation $\delta_p W_p = \delta_s W_s$, and the order of the filter can be found empirically via

$$N \approx \frac{-10\log_{10}(\delta_p\delta_s)-13}{2.324(\omega_s-\omega_p)}\ .$$

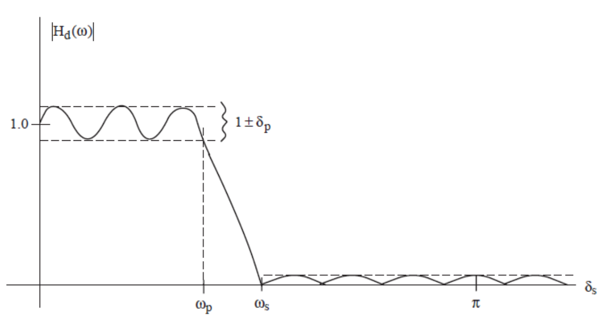


Figure 6

3 Lab Demo

3.1 FIR Filters designed by the Windowing Method

3.1.1 Rectangular Window vs Hamming Window

Design a lowpass filter of length $N = 21$ and cutoff frequency $\omega_c = \frac{\pi}{4}$. We first specify the frequency response of the filter, as the desired response with a linear phase added (based on the chosen filter length N)

$$\begin{aligned} G_d(\omega) &= D(\omega)e^{-j\frac{N-1}{2}\omega} \\ &= \begin{cases} e^{-j10\omega}, \ |\omega| \le \frac{\pi}{4} \\ 0, \ \frac{\pi}{4} \le |\omega| \le \pi \end{cases} \end{aligned}$$

Then, $g[n]$ is given by

$$\begin{aligned} g[n] &= \frac{1}{2\pi}\int_{-\frac{\pi}{4}}^{\frac{\pi}{4}} e^{-j10\omega}e^{j\omega n}d\omega \\ &= \frac{1}{4}\text{sinc}\Big(\frac{(n-10)\pi}{4}\Big). \end{aligned}$$

Finally, the FIR filter coefficients are given by $h[n] = w[n]g[n]$, where $w[n]$ is the selected window function. The impulse responses and frequency responses two filters designed by the windowing method (a rectangular window and a Hamming window, respectively), are shown below in Figure 7. The rectangular window has a narrower mainlobe but higher sidelobes while the Hamming window has a lower sidelobes but a wider mainlobe. These characteristics are reflected in the frequency responses below; the filter designed with a rectangular window has a sharp transition around $\omega = \pi/4$, whereas the filter designed with a Hamming window has a wider transition band. Also, as expected, the stopband ripple for the filter designed with a rectangular window is larger than the stopband ripple in the filter designed with a Hamming window.

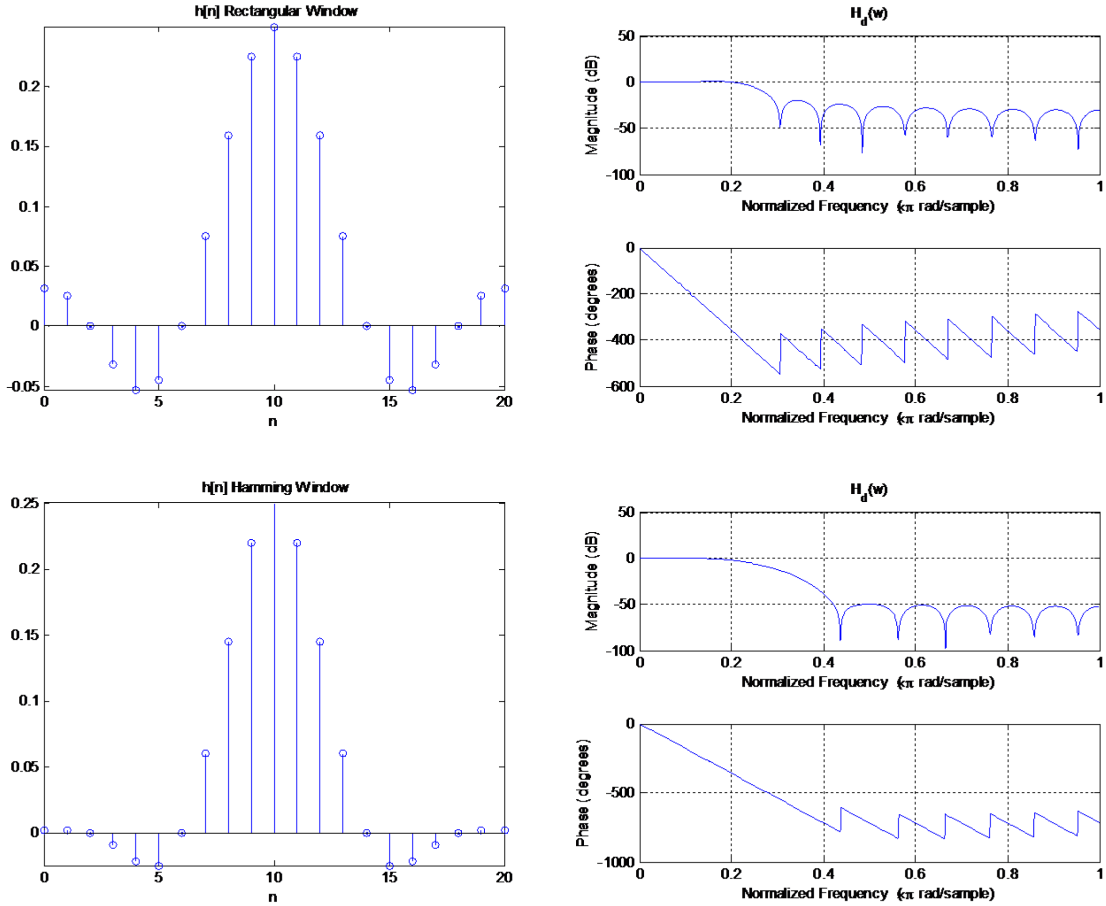


Figure 7: (top) FIR filter designed by the windowing method using a rectangular window, (bottom) FIR filter designed by the windowing method using a hamming window.

Remember, you can use many different other windows (triangular, hanning, kaiser, bartlett, blackman, etc.).

3.1.2 Increasing Window Length

To see the effect of window length, we now redesign the Hamming window filter with lengths $N = 21, 41, 61$. The corresponding impulse responses and frequency responses are shown in Fig. 8. As the window length increases, the transition bandwidth of the resulting filter decreases and the sidelobe height decreases.

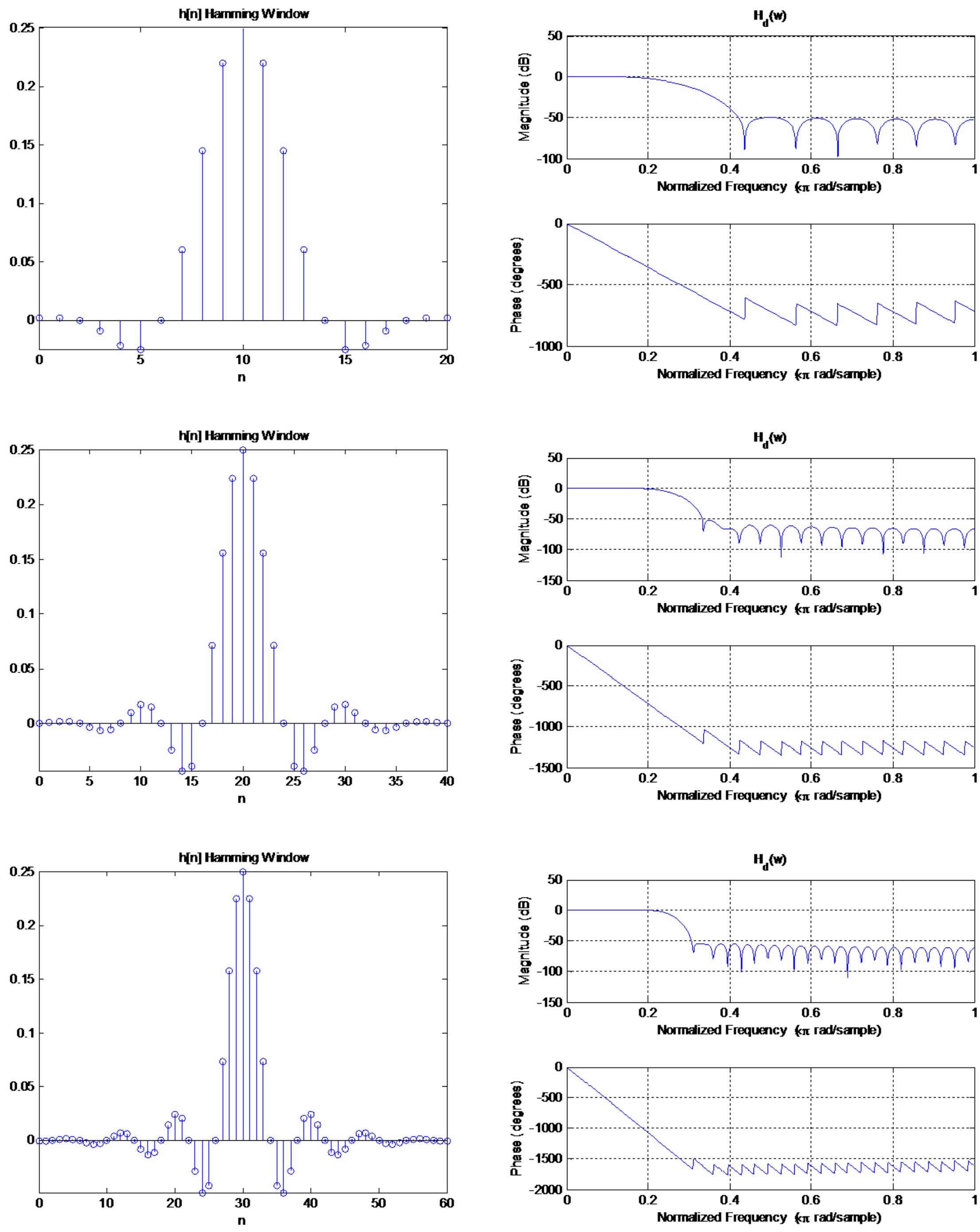


Figure 8: FIR filter designed by the windowing method using a hamming window (top) N=21, (middle) N=41, (bottom) N=61.

3.2 FIR Filters Designed by the Frequency Sampling Method

We will now design another lowpass filter of length $N = 21$ and cutoff frequency $\omega_c = \pi/4$, this time using the frequency sampling method. Because zero-phase filters are not realizable, we must add linear phase (based on the filter length) to the phase response, as follows:

$$G_d(\omega) = \begin{cases} e^{-j\frac{N-1}{2}\omega}, & 0 \le \omega \le \frac{\pi}{4} \\ 0, & \frac{\pi}{4} < \omega < \frac{7\pi}{4} \\ e^{-j\frac{N-1}{2}\omega}, & \frac{7\pi}{4} \le \omega < 2\pi \end{cases}$$

$$G_d(\tfrac{2\pi k}{N}) = \begin{cases} e^{-j\frac{\pi k(N-1)}{N}}, & 0 \le k \le 2 \\ 0, & 3 < k < 18 \\ e^{-j\frac{\pi k(N-1)}{N}}, & 19 \le k < 20 \end{cases}$$

We then take the inverse DFT of the samples to obtain the impulse response. The impulse response and frequency response of the designed filter are shown below.

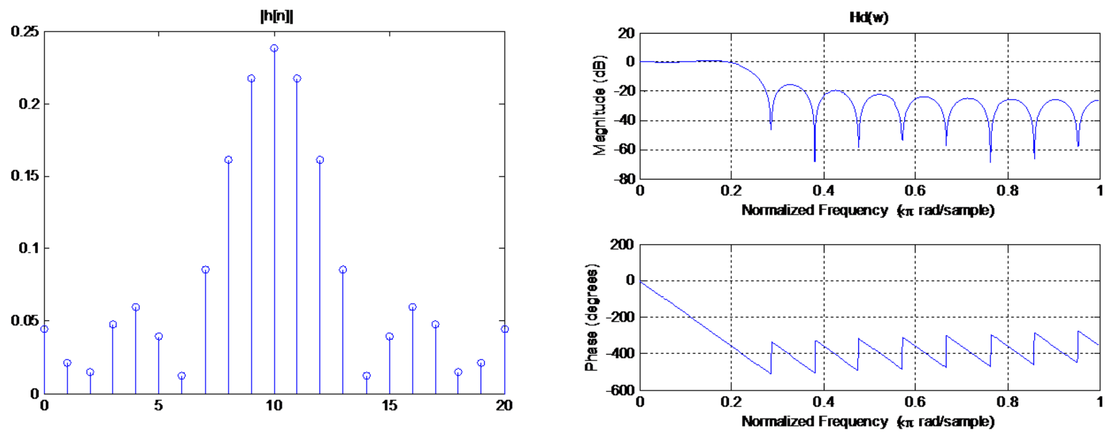


Figure 9

To reduce the stopband ripple, we can adjust one of the samples in the transition band at the cost of transition bandwidth

$$G_d(\tfrac{2\pi k}{N}) = \begin{cases} e^{-j\frac{\pi k(N-1)}{N}}, & 0 \le k \le 2 \\ ae^{-j\frac{3\pi(N-1)}{N}}, & k = 3 \\ 0, & 4 \le k \le 17 \\ ae^{-j\frac{18\pi(N-1)}{2N}}, & k = 18 \\ e^{-j\frac{\pi k(N-1)}{N}}, & 19 \le k \le 20 \end{cases}$$

Where $0 < a < 1$. The modified filter (with $a = 0.3$) is shown in Fig. 10, the stopband ripples have much lower amplitude than for the original design.

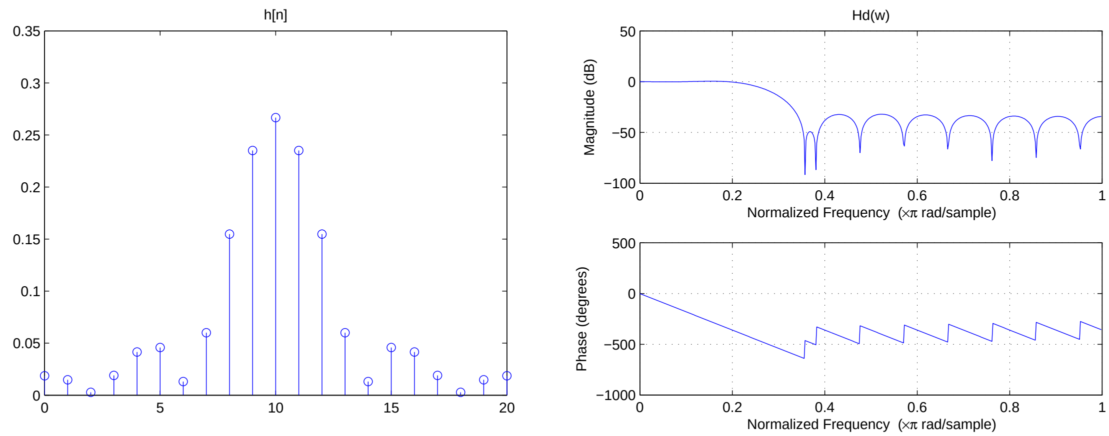


Figure 10

3.3 FIR Filter Designed by the Parks-McClellan Method

A lowpass filter is designed using the Parks-McClellan method. In this design. The impulse response and frequency response of the designed filter are shown below. In comparison, the impulse response and frequency response of a low-pass filter designed using the windowing method with a Kaiser window are also shown below. With about the same stopband ripple, the Parks-McClellan method achieves a much narrower transition band at the cost of passband ripple.

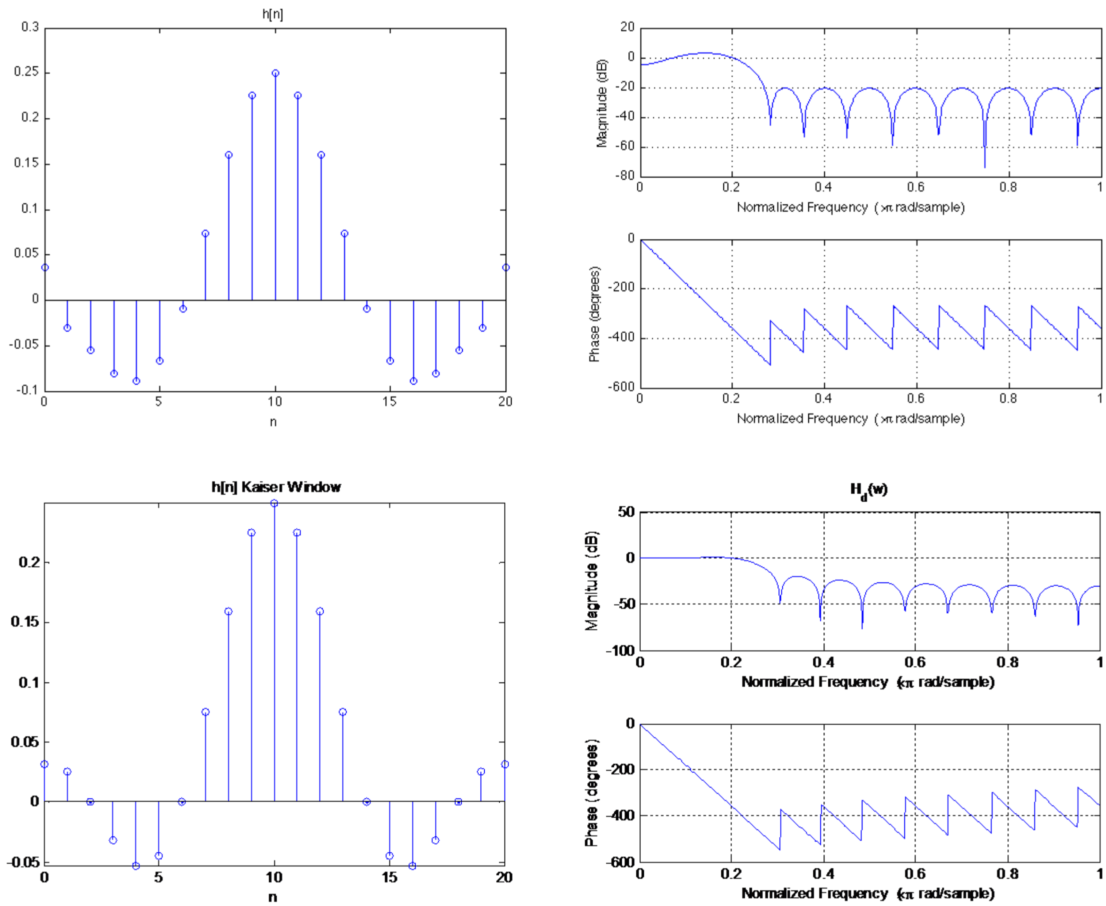


Figure 11: (top) Parks-McClellan method, (bottom) windowing method using a Kaiser window.

	Passband edge	Stopband edge	Passband ripple (dB)	Stopband ripple (dB)
Parks-McClellan	0.23π	0.27π	5	20
Kaiser	0.21π	0.29π	0.6	20

3.4 Applications of FIR Filters in Image Processing

In problem 5a of Lab 4, we tried deconvolution for a sound signal. Recall that we first modified the signal using an LSI filter, then undid the effects of the filter. Here let us try it in 2 dimensions using our new filter design techniques, specifically the frequency sampling method.

Given an image $I[p,q]$ and a convolution/blurring kernel $h[n]$, suppose the measured/blurred image $\hat{I}[p,q]$ is given by the following two dimensional convolution:

$$\hat{I}[p,q] = (I[p,q] \bigotimes_c h[n]) \bigotimes_r h[n] \ ,$$

where $\bigotimes_c$ and $\bigotimes_r$ represents a circular convolution along a column and row, respectively.

In the frequency domain, we have:

$$\hat{S}_d(\omega_x, \omega_y) = S_d(\omega_x, \omega_y) H_d(\omega_x) H_d(\omega_y),$$

where $\hat{S}_d(\omega_x, \omega_y)$, $S_d(\omega_x, \omega_y)$, and $H_d(\omega)$ are the DTFTs of $\hat{I}[p,q], I[p,q]$, and $h[n]$, respectively.

To recover the original image from the measured/blurred image, we can first design a deconvolution/deblurring filter, whose frequency response is given by

$$G_d(\omega) = \frac{1}{H_d} = \frac{H_d^*}{H_d^* H_d} = \frac{H_d^*(\omega)}{|H_d(\omega)|^2 + \epsilon} \tag{4}$$

where ϵ is a small positive scalar (e.g. 0.000001) to improve the numerical stability of the algorithm (avoid ‘Inf’ and ’NaN’ values).

The image can then be recovered as

$$I_{re}[p,q] = (\hat{I}[p,q] \bigotimes_c g[n]) \bigotimes_r g[n] \ ,$$

where $g[n]$ is the impulse response of the designed deconvolution/deblurring filter.

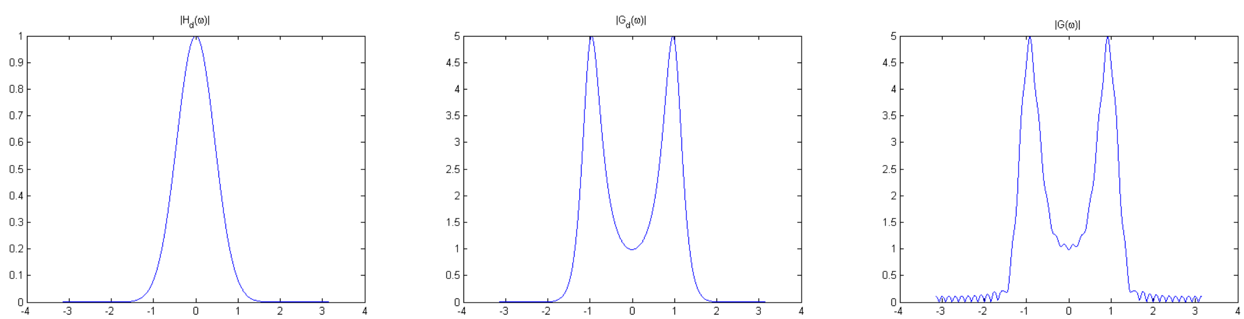
<u>Example:</u> Suppose the convolution/blurring kernel is given by

$$H_d(\omega) = e^{-\frac{\omega^2}{0.4} - j511\omega}. \tag{5}$$

Use the frequency sampling method to design an FIR filter of length $N = 41$ to approximate the frequency response of the deconvolution/deblurring filter

$$G_d(\omega) = \frac{H_d^*(\omega)}{|H_d(\omega)|^2 + 0.000001}.$$

The spectrum of the convolution/blurring kernel, the magnitude response of the ideal deconvolution/deblurring filter, and the magnitude response of the designed filter are shown below.



(a) $H_d(\omega)$ (b) $G_d(\omega)$ (c) Designed filter

Figure 12

The original image, the blurred image, and the de-blurred image are shown below. The number of the car’s license plate can now be clearly seen in the de-blurred image.

(a) Original image

(b) Blurred image

(c) De-blurred image

Figure 13

4 Homework 6 - Due 11/17/2015 at 5:00 PM

1. Design and analyze the following generalized linear phase FIR filters using the windowing method. For each part of the problem, you need to first derive the closed-form solution of $g[n]$ as in section 3.1 and apply the required window, then calculate and plot the impulse response, and the magnitude (in dB) and phase of the frequency response of the designed FIR filter. Find the passband ripple, stopband attenuation, passband edge frequency, and stopband edge frequency. Determine the type (I or II) of the designed filter.

- **(a)** Design a lowpass filter of length $N = 25$ with cutoff frequency $\omega_c = \pi/3$ using a rectangular window.
- **(b)** Design a lowpass filter of length $N = 25$ with cutoff frequency $\omega_c = \pi/3$ using a Hamming window.
- **(c)** Design a lowpass filter of length $N = 25$ with cutoff frequency $\omega_c = \pi/3$ using a Kaiser window. The Kaiser window can be generated by the MATLAB command `w=kaiser(N,beta)`. Choose the parameter `beta` such that the resulting filter has a stopband attenuation of 30dB. What value of `beta` did you use?
- **(d)** Repeat parts **(a)** through **(c)** using a length of $N = 50$. What do you observe when the length of the filter increases?
- **(e)** Using the modulation property, design a highpass filter with cutoff frequency $\omega_c = 2\pi/3$ based on the lowpass filter you designed in part **(b)**. Determine the type (I or II) of the designed filter.
- **(f)** Repeat part **(e)** with length $N = 50$. What do you observe when the length of the filter increases?

2. Design and analyze a lowpass FIR filter of length $N = 25$ with cutoff frequency $\omega_c = \pi/3$ using the frequency sampling method.

- **(a)** Write down the sampled frequency response $G_d(\omega)$ at $\omega = \frac{2\pi k}{N}$ for $0 \leq k \leq N-1$ with *zero-phase* ($G_d(\omega)$ is real).
- **(b)** Calculate and plot the impulse response, and the magnitude (in dB) and phase of the frequency response of the FIR filter from part **(a)**. Is the frequency response of the designed filter close to the desired one? Why or why not?
- **(c)** Write down the sampled frequency response $G_d(\omega)$ at $\omega = \frac{2\pi k}{N}$ for $0 \leq k \leq N-1$ with *generalized linear phase.*
- **(d)** Calculate and plot the impulse response and the magnitude (in dB) and phase of the frequency response of the FIR filter from part **(c)**. Find the passband ripple, stopband attenuation, passband edge frequency and stopband edge frequency.
- **(e)** Try to improve the FIR filter designed in part **(d)** by including a transition band in the frequency sampling design (as in lab demo 2). Use the same phase as in part **(d)** and let the magnitude for samples $k = 5$ and $k = 20$ become $|G_d(\frac{2\pi\times 5}{N})| = |G_d(\frac{2\pi\times 20}{N})| = a$, for some a such that $0 < a < 1$. What value of a leads to at least 35 dB attenuation in the stopband? Calculate and plot the impulse response, and the magnitude (in dB) and phase of the frequency response of the FIR filter. Find the passband ripple, stopband attenuation, passband edge frequency and stopband edge frequency.

3. Design FIR filters using the Parks-McClellan method. For each problem, calculate and plot the impulse response, and the magnitude (in dB) and phase of the frequency response of the designed filter. Find the passband ripple, stopband attenuation, passband edge frequency, and stopband edge frequency (i.e. `w =[1 1]`).

(a) Design a lowpass filter of length $N = 25$ with passband edge frequency 0.3π and stopband edge frequency 0.36π using the MATLAB command `firpm(n,f,a,w)` (you might need to look up the Matlab documentation for this function!). Let the weighting for the passband and stopband be equal.

(b) Increase the stopband attenuation of the filter in part (a) to at least 30 dB by increasing the transition bandwidth.

(c) Increase the stopband attenuation of the filter in part (a) to at least 30 dB by adjusting the weights.

(d) Increase the filter length to $N = 50$. Repeat part (a) and compare the results.

(e) Design a multi-band FIR filter of the minimum possible length while satisfying the following specifications: the first passband edge frequencies = $(0, 0.1\pi)$, the first stopband edge frequencies= $(0.15\pi, 0.3\pi)$, the second passband edge frequencies = $(0.4\pi, 0.7\pi)$, and the second stopband edge frequencies = $(0.75\pi, \pi)$. The attenuation is at least 30 dB in the first stopband and 40 dB in the second stopband. The passband ripple is at most 2 dB in the first passband and at most 0.5 dB in the second passband.

4. Two speech signals are stored in the file *speechsig.mat*. The vector `x` contains the original version of the speech signal. The vector `xnoise` contains the speech signal with some high-frequency noise added. Play both `x` and `xnoise` using the command `soundsc` to hear the difference between clean and corrupted speech signals.

(a) You were not given a sampling frequency F_s with the signal `.mat` file. A reasonable sampling frequency for an audio signal could be anywhere between 6000 Hz and 44100 Hz. By listening `soundsc(x,Fs)` find a reasonable value for F_s.

(b) Calculate and plot the magnitude of the DFT spectrum of each signal. Estimate the frequency range of the noise. What type of noise does this appear to be (e.g. added tones, pink noise (equal power per octave), white noise (gaussian noise))?

(c) Design a lowpass filter to clean the noise-corrupted speech signal, which works in this case because the signal and the noise are fairly well separated in frequency. Consider the following questions while designing the lowpass filter:

(i) What is the desired passband edge frequency of the filter?

(ii) What is the desired stopband edge frequency?

(iii) What is the desired stopband attenuation?

(iv) Which of the different design methods is most suitable to satisfy the above requirements?

(v) How can you design a filter satisfying the above requirements while keeping the order of the filter as low as possible?

Calculate and plot the impulse response, magnitude (in dB), and phase of the frequency response of the designed filter. Find the passband ripple, stopband attentuation, passband edge frequency, and stopband edge frequency.

(d) Process the noise-corrupted speech signal with the filter you designed. Calculate and plot the DFT spectrum of the filtered speech data. Play the filtered data and the uncorrupted data to hear the difference.

(e) Compare this result to a simple mean filter, which can diminish the effects of noise by averaging neighboring samples. You can design your mean filter `filter(b,a,x)` using `b=(1/N)*ones(1,N)` and `a=1`.