

EN 253: Matlab Homework #2

Douglas R. Lanman

3 November 2005

1 Introduction

As discussed in the assignment handout, the goal of this writeup is to present the design of a bandpass FIR filter using the three methods discussed in class: (1) the window method, (2) the frequency sampling method, and (3) optimal equiripple design.

To begin our analysis, we must specify the filter passband, stopband, and attenuation parameters [See Mitra, p. 490, Fig. 9.1]. First, note that the first stopband is given by $\omega \in (0, \omega_{s_1})$, the passband is given by $\omega \in (\omega_{p_1}, \omega_{p_2})$, and the second stopband is given by $\omega \in (\omega_{s_2}, \pi)$ (in radial frequency units). The radial frequencies are as follows

$$\begin{aligned}\omega_{s_1} &= 2\pi \left(\frac{1,768 \text{ Hz}}{44,100 \text{ Hz}} \right) \approx 0.08018141\pi, & \omega_{s_2} &= 2\pi \left(\frac{5,967 \text{ Hz}}{44,100 \text{ Hz}} \right) \approx 0.2706122\pi \\ \omega_{p_1} &= 2\pi \left(\frac{2,431 \text{ Hz}}{44,100 \text{ Hz}} \right) \approx 0.1102494\pi, & \omega_{p_2} &= 2\pi \left(\frac{4,862 \text{ Hz}}{44,100 \text{ Hz}} \right) \approx 0.2204989\pi\end{aligned}$$

Note that there are two asymmetric transition bands given by

$$\Delta\omega_1 = \omega_{p_1} - \omega_{s_1} \approx 0.03006803\pi \quad (1)$$

$$\Delta\omega_2 = \omega_{s_2} - \omega_{p_2} \approx 0.05011338\pi \quad (2)$$

To complete the filter specification, we must define $\{\delta_s, \delta_p\}$ the stopband and passband peak ripple values, respectively. Recall that the problem statement gives $\alpha_s = 60$ dB and $\alpha_p = 0.5$ dB (this was confirmed by the TA). Note that the passband and stopband peak ripple values can be converted to linear scale, from decibels, using the following expressions [See Mitra, p. 490, (9.3,9.4)].

$$\alpha_p = -20 \log_{10}(1 - \delta_p) \text{ dB}$$

$$\alpha_s = -20 \log_{10}(\delta_s) \text{ dB}$$

Substituting for $\{\alpha_s, \alpha_p\}$ we obtain

$$\delta_p = 1 - 10^{\frac{-0.5}{20}} \approx 0.05593912 \quad (3)$$

$$\delta_s = 10^{\frac{-60}{20}} = 0.001 \quad (4)$$

2 Filter Length Prediction

In this section, I will provide a prediction of the length of the optimal FIR filter using three formulas: (1) Kaiser's, (2) Hermann's, and (3) Mintzer and Liu's. As we will see, these formulas provide similar, but not identical, estimates that can be used to guide the design

process.

To begin, let's consider the familiar Kaiser's formula presented in class [See Mitra, p. 524, (10.3)].

$$N_{Kaiser} \cong \frac{-20 \log_{10}(\sqrt{\delta_p \delta_s}) - 13}{14.6(\Delta\omega/2\pi)}$$

As discussed in Mitra (p. 526), the transition width $\Delta\omega$ should correspond to the smallest transition width for asymmetric bandpass filters. As a result, the first transition band ($\Delta\omega_1$) will be used for $\Delta\omega$. Using Equations (1,3,4) we find

$$\boxed{N_{Kaiser} \cong 135}$$

Alternative filter order prediction formulas were briefly discussed in class, including Hermann's and Mintzer and Liu's. Let's first consider Hermann's formula, which is outlined in Mitra (p. 525).

$$\begin{aligned} N_{Hermann} &\cong \frac{D_\infty(\delta_p, \delta_s)}{(\Delta\omega/2\pi)} - F(\delta_p, \delta_s)(\Delta\omega/2\pi) \\ D_\infty(\delta_p, \delta_s) &= \log_{10} \delta_s [a_1(\log_{10} \delta_p)^2 + a_2(\log_{10} \delta_p) + a_3] \\ &\quad - [a_4(\log_{10} \delta_p)^2 + a_5(\log_{10} \delta_p) + a_6] \\ F(\delta_p, \delta_s) &= b_1 + b_2[\log_{10} \delta_p - \log_{10} \delta_s] \end{aligned}$$

The coefficients are given by the following values

$$\begin{aligned} a_1 &= 0.005309, a_2 = 0.07114, a_3 = -0.4761 \\ a_4 &= 0.00266, a_5 = 0.5941, a_6 = 0.4278 \\ b_1 &= 11.01217, b_2 = 0.51244 \end{aligned}$$

Using the smaller transition width $\Delta\omega_1$ and Equations(1,3,4) we find

$$\boxed{N_{Hermann} \cong 132}$$

Finally, let's consider the Mintzer and Liu prediction formula [See F. Mintzer and B. Liu, "An Estimate of the Order of an Optimal FIR Band-pass Digital Filter", 1978]. As discussed in class, this formula is quite similar to Hermann's and is given by the following expressions

$$\begin{aligned} N_{Mintzer} &\cong \frac{C_\infty(\delta_p, \delta_s)}{(\Delta\omega/2\pi)} + G(\delta_p, \delta_s)(\Delta\omega/2\pi) \\ C_\infty(\delta_p, \delta_s) &= \log_{10} \delta_s [a_1(\log_{10} \delta_p)^2 + a_2(\log_{10} \delta_p) + a_3] \\ &\quad + [a_4(\log_{10} \delta_p)^2 + a_5(\log_{10} \delta_p) + a_6] \\ G(\delta_p, \delta_s) &= -14.6 \log_{10} \left(\frac{\delta_p}{\delta_s} \right) - 16.9 \end{aligned}$$

where the coefficients are given by

$$a_1 = 0.01201, a_2 = 0.09664, a_3 = -0.51325$$

$$a_4 = 0.00203, a_5 = -0.5705, a_6 = -0.44314$$

Once again, using the smaller transition width $\Delta\omega_1$ and Equations (1,3,4) we find

$$N_{Mintzer} \cong 141$$

Notice that the three filter order prediction formulas give similar results. Also recall that the FIR filter length is equal to the order plus one. The predicted optimal FIR filter lengths are tabulated below.

Prediction Formula	Length
Kaiser	136
Hermann	133
Mintzer and Liu	142

3 Window Method

In this section, I will summarize my design of the required FIR filter using the window method. First, analytical results will be presented. Afterward, *Matlab* plots will be used to analyze the performance of the FIR filter and confirm that it meets all design criteria.

3.1 Ideal Impulse Response

The ideal infinite discrete-time filter sequence can be obtained by taking the inverse DTFT of the desired filter $H_D(e^{j\omega})$. For arbitrary lower and upper cutoff frequencies $\{\omega_{c_1}, \omega_{c_2}\}$ we can define the desired frequency response as follows

$$H_D(e^{j\omega}) = \begin{cases} 0, & 0 \leq |\omega| < \omega_{c_1} \\ 1, & \omega_{c_1} \leq |\omega| \leq \omega_{c_2} \\ 0, & \omega_{c_2} < |\omega| \leq \pi \end{cases}$$

The desired impulse response $h_d[n]$ has the following solution [c.f. Mitra, p. 528, (10.17)]. First, let's begin by writing the impulse response as the inverse DTFT of the frequency response.

$$h_d[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_D(e^{j\omega}) e^{j\omega n} d\omega$$

Notice that the desired frequency response is only non-zero and equal to unity within the passband, so we have

$$h_d[n] = \frac{1}{2\pi} \int_{-\omega_{c_2}}^{-\omega_{c_1}} e^{j\omega n} d\omega + \frac{1}{2\pi} \int_{\omega_{c_1}}^{\omega_{c_2}} e^{j\omega n} d\omega$$

For $n \neq 0$, we obtain the following result

$$h_d[n] = \frac{1}{(2\pi n)j} e^{j\omega n} \Big|_{\omega=-\omega_{c2}}^{\omega=-\omega_{c1}} + \frac{1}{(2\pi n)j} e^{j\omega n} \Big|_{\omega=\omega_{c1}}^{\omega=\omega_{c2}}$$

$$h_d[n] = \frac{1}{n\pi} \left[\frac{e^{-j\omega_{c1}n} - e^{-j\omega_{c2}n}}{2j} \right] + \frac{1}{n\pi} \left[\frac{e^{j\omega_{c2}n} - e^{j\omega_{c1}n}}{2j} \right]$$

Rearranging terms, we can express the solution in the form of sinusoids

$$h_d[n] = \frac{1}{n\pi} \left[\frac{e^{j\omega_{c2}n} - e^{-j\omega_{c2}n}}{2j} \right] - \frac{1}{n\pi} \left[\frac{e^{j\omega_{c1}n} - e^{-j\omega_{c1}n}}{2j} \right]$$

$$h_d[n] = \frac{\sin(\omega_{c2}n)}{n\pi} - \frac{\sin(\omega_{c1}n)}{n\pi}, \quad n \neq 0$$

We can obtain the solution for $n = 0$ by considering the limiting behavior (and applying L'Hospital's rule).

$$\lim_{n \rightarrow 0} \frac{\sin(\omega_{c2}n)}{n\pi} - \frac{\sin(\omega_{c1}n)}{n\pi} = \lim_{n \rightarrow 0} \frac{\omega_{c2} \cos(\omega_{c2}n)}{\pi} - \frac{\omega_{c1} \cos(\omega_{c1}n)}{\pi} = \frac{\omega_{c2} - \omega_{c1}}{\pi}$$

In conclusion, the closed-form expression for the ideal infinite discrete-time filter sequence is

$$h_d[n] = \begin{cases} \frac{\sin(\omega_{c2}n)}{n\pi} - \frac{\sin(\omega_{c1}n)}{n\pi}, & n \neq 0 \\ \frac{\omega_{c2} - \omega_{c1}}{\pi}, & n = 0 \end{cases}$$

To proceed with our design, we must now select *sensible* cutoffs for the ideal filter used in this method. As discussed in the problem statement, we do not necessarily want to select the endpoints of the passband as the cutoff frequencies (i.e., 2,431 Hz and 4,862 Hz); such a choice would necessitate a larger filter due to the narrowing of the passband. We will use the standard choice of the halfway points in the transition bands as our cutoff frequencies. As a result, we find

$$f_{c1} = \frac{1,768 \text{ Hz} + 2,431 \text{ Hz}}{2} = 2,099.5 \text{ Hz}$$

$$f_{c2} = \frac{4,862 \text{ Hz} + 5,967 \text{ Hz}}{2} = 5,414.5 \text{ Hz}$$

where f_{c1} is the lower cutoff frequency and f_{c2} is the upper cutoff frequency for the ideal filter. In radial frequency units, we conclude

$$\omega_{c1} = \frac{\omega_{s1} + \omega_{p1}}{2} = 0.09521542\pi$$

$$\omega_{c1} = \frac{\omega_{p2} + \omega_{s2}}{2} = 0.2455556\pi$$

3.2 Filter Length and Window Selection

In order to create a finite length filter we must truncate the ideal, infinite-length impulse response $h_d[n]$. As previously discussed, the filter length prediction formulas were used to estimate the minimum length for an FIR filter meeting our design criteria. As we will see in the following sections, however, the windowing method does not achieve the optimal length predicted by these formulas. Instead, the “windowing method” filter tends to be longer than the predicted length by at least a factor of two – indicating the fundamental limitation of the windowing method for filter design.

If we simply truncate the ideal sequence without applying a window, then we are implicitly assuming a rectangular window. As can be seen in [Mitra, p. 534, Figure 10.7 and Table 10.2] there are a variety of window functions available. In general, there are two classes of windows: fixed and parametric. In this problem we will consider both classes.

Again by inspecting the plot in Mitra (p. 534), we see that only the Hann and Blackman windows meet our design criterion of 60 dB attenuation in the stopband. As a result, we will restrict our analysis to these two fixed window functions. Although these windows meet the required stopband attenuation goal, a sufficient number of samples must be used to ensure that the main lobe does not extend further than the smallest transition band (i.e., $\Delta\omega_{c1}$). Using the formulas from [Mitra, p. 535, Table 10.2] we obtain the following predictions of the main lobe widths

$$\text{Hann Window: } \Delta_{ML} = \frac{4f_s}{L}$$

$$\text{Blackman Window: } \Delta_{ML} = \frac{6f_s}{L}$$

where $f_s = 44,100$ Hz (the sampling frequency) and L is the length of the FIR filter. Note that half the main lobe width must be less than or equal to the effective transition width: $\frac{1}{2}\Delta_{ML} = \Delta f = f_{c1} - f_{s1} = 2,099.5$ Hz - $1,768$ Hz = 331.5 Hz. Solving for L , the required filter length we find

$$\text{Hann Window: } L \approx \left\lceil \frac{2f_s}{\Delta f} \right\rceil = 267$$

$$\text{Blackman Window: } L \approx \left\lceil \frac{3f_s}{\Delta f} \right\rceil = 400$$

This is only a rough estimate of the required filter order, but indicates that, using a fixed window, it will be difficult to achieve the predicted optimal filter length. This is due to the fact that fixed windows do not allow dynamic optimization of the transition width and sidelobe attenuation. For such functionality we will have to consider parametric windows. For this writeup, I will restrict my analysis to the Kaiser window (although my *Matlab* code also supports Dolph-Chebyshev windows).

The Kaiser window function is defined as follows [See Mitra, p. 538, (10.39)]

$$w[n] = \frac{I_0\left\{\beta\sqrt{1 - (n/M)^2}\right\}}{I_0(\beta)}, \quad -M \leq n \leq M$$

where β is the parametric parameter and $I_0(u)$ is the modified zeroth-order Bessel function. Similar to our rough analysis for fixed window lengths, Kaiser provides the following formulas to predict the necessary window length and β for a given design

$$\beta = \begin{cases} 0.1102(\alpha_s - 8.7), & \text{for } \alpha_s > 50 \\ 0.5842(\alpha_s - 21)^{0.4} + 0.07886(\alpha_s - 21), & \text{for } 21 \leq \alpha_s \leq 50 \\ 0, & \text{for } \alpha_s < 21 \end{cases}$$

$$N = \frac{\alpha_s - 8}{2.285\Delta\omega}$$

Substituting for the values of $\Delta\omega_1$ and α_s , we obtain the following results

$$N = 241, \beta \approx 5.65326$$

3.3 Frequency Domain Analysis of Filter Design

As discussed in the previous section, we are going to evaluate the Hann, Blackman, and Kaiser windows. In order to demonstrate that our filter meets the design criterion, I will evaluate the frequency response magnitude of the truncated, windowed impulse response. In this manner, we can graphically confirm that the frequency response of our truncated, windowed impulse response achieves the required attenuation and transition width criteria.

As was demonstrated in Matlab Homework #1, we can use a finite-length DFT to approximate a DTFT [See Mitra, pp. 240-241]. In order to achieve a greater frequency resolution than simply taking the N-point DFT, we will use the standard method of zero-padding the impulse response sequence to obtain a length M sequence ($M > N$). As a result, for an arbitrary window $w[n]$, we obtain the following frequency response (via the M-point DFT)

$$X(e^{j\omega_k}) = \sum_{n=0}^{M-1} h_e[n] e^{-j2\pi kn/M}$$

where the truncated, zero-padded impulse response is given by

$$h_e[n] = \begin{cases} w[n] \cdot h_d[n], & 0 \leq n < N - 1 \\ 0, & N \leq n \leq M - 1 \end{cases}$$

The attached *Matlab* program `Problem_1.m` allows the user to select from various window functions and filter lengths and plots both the impulse response and the frequency response magnitude for the resulting FIR filter [See attached source code]. Both the impulse responses and frequency response magnitudes are evaluated for the Hann, Blackman, and Kaiser windows. *Please note that all figures are reproduced in the Appendix at full-page resolution.*

Using `Problem_1.m`, I began my filter design by choosing a filter length L and window type `wType` [See lines 31 and 35]. For the Hann window, I initially choose a length-267 filter (as predicted in Section 3.2). While this filter was nearly acceptable, it technically did not meet the attenuation criterion of 60 dB at the beginning of the stopband (it did meet the passband requirements). Using the frequency response magnitude plot, I progressively increased the filter length until the design criteria were satisfied everywhere. Since I

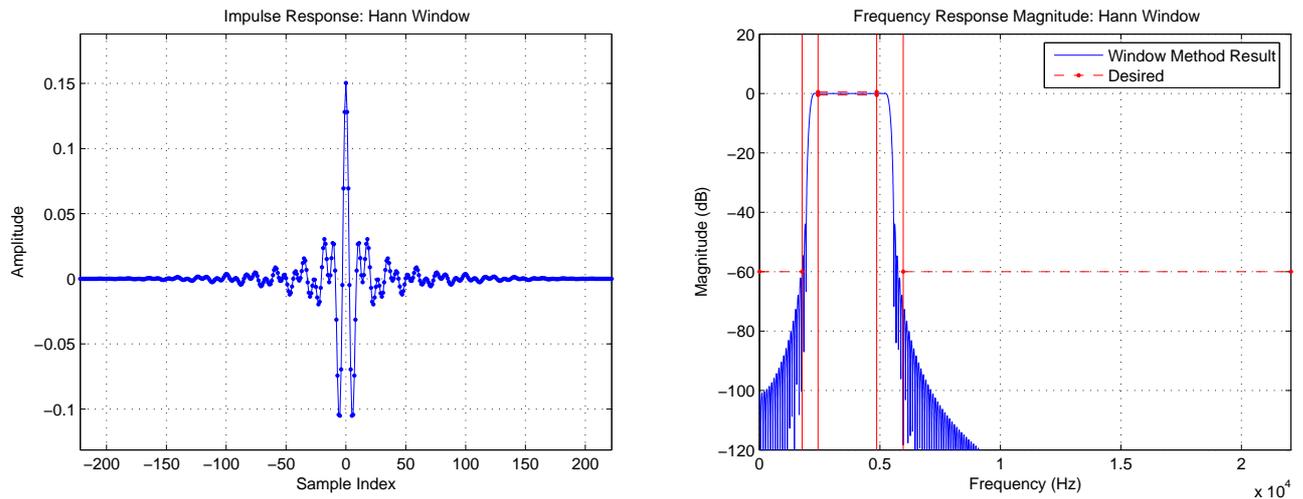


Figure 1: Results for a 445-point Hann Window.

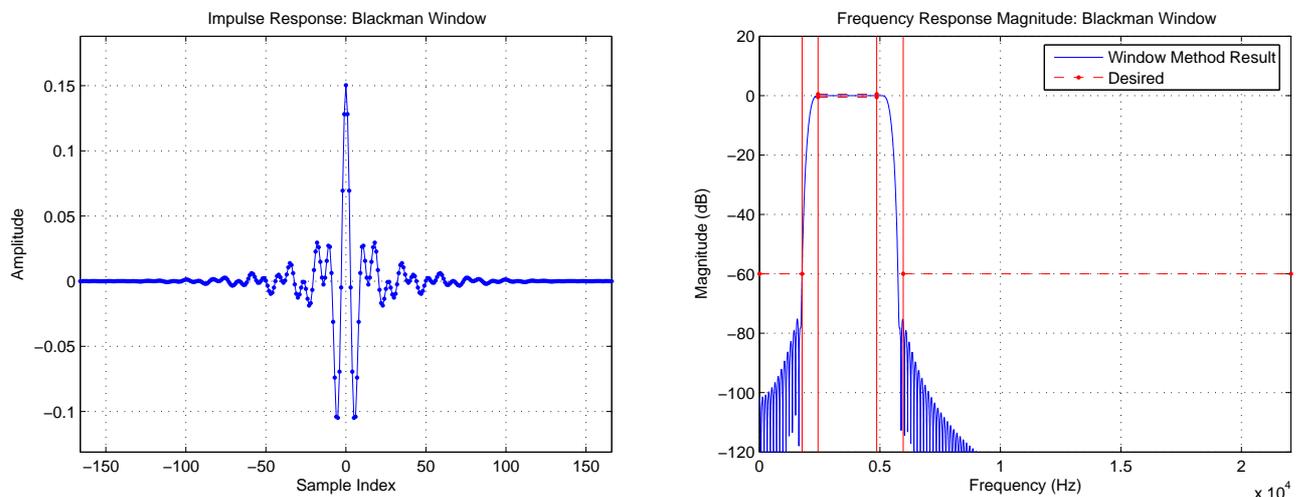


Figure 2: Results for a 333-point Blackman Window.

am interested in designing a Type I FIR filter (i.e., even-symmetric and real-valued) I only considered odd values of the filter length. In conclusion, I found that the Hann window required a length-445 filter to meet all design goals. The Hann-windowed impulse response and corresponding frequency response magnitude for a length-445 filter are shown in Figure 1.

A similar design process was undertaken for the Blackman window. Using an initial length of 400 (as predicted in Section 3.2), I found that this initial estimate was too conservative. From visual inspection, I found that a length-333 window was the minimum allowed by the design criteria. The Blackman-windowed impulse response and frequency response magnitude for a length-333 window are shown in Figure 2.

Not surprisingly, I found that the parametric Kaiser window achieved the shortest-length filter meeting the design criterion. Beginning with the initial length of 241 predicted in

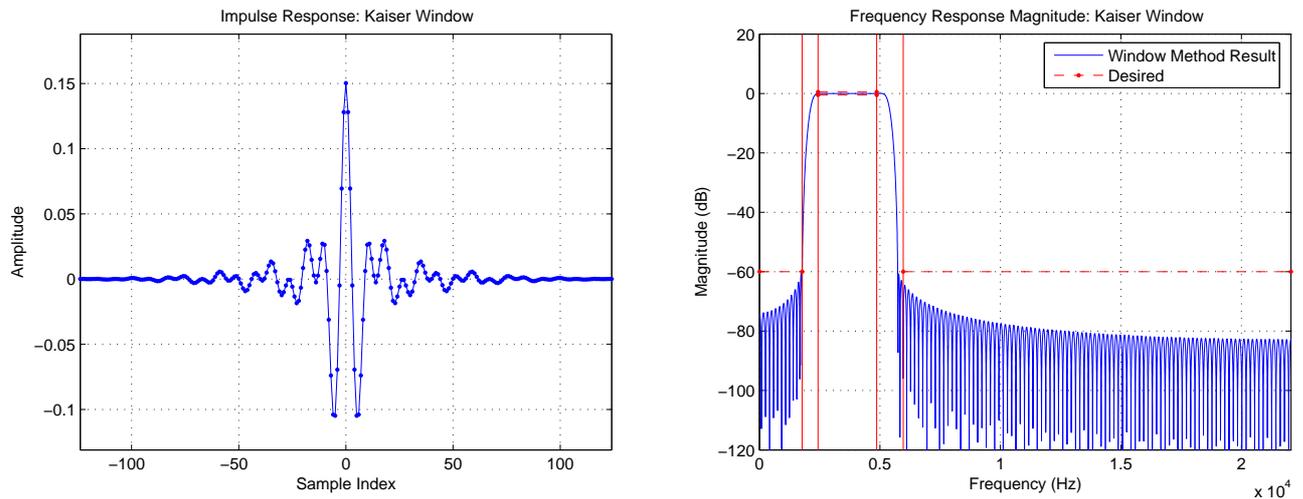


Figure 3: Results for a 249-point Kaiser Window.

Section 3.2, I found that a length-249 Kaiser window was the shortest odd-length filter that met all the design requirements. The results for the Kaiser window are shown in Figure 3. *Please consult the Appendix for high-resolution plots.*

In conclusion, we find that the window method does not achieve the predicted optimal filter lengths – even when a parametric window is used. This observation motivates the development of alternate filter design methodologies, such as the frequency sampling method or the optimal filter design – both of which will be considered in the following sections. The minimum filter lengths for each window type are tabulated below.

Window Type	Filter Length
Hann	445
Blackman	333
Kaiser	249

4 Frequency Sampling Method

In this section we will apply the frequency sampling method to design an FIR filter which meets all of the criteria discussed in Section 1. We will see that, initially, the frequency sampling method does not appear to be very effective in achieving short filters. The introduction of variable “transition” samples, however, will result in greatly-reduced filter lengths. As a result, we will discover that the frequency sampling method comes closer to approaching the predicted filter lengths we found in Section 2.

The frequency sampling method is a straightforward process: N equally-spaced samples are taken of the desired frequency response $H_D(e^{j\omega})$. The inverse DFT is applied to these

N points to yield a finite-length impulse response as follows [See Mitra, p. 236, (5.14)].

$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} H[k] e^{j2\pi kn/N}, \quad 0 \leq n \leq N-1$$

The values of $H[k]$ are given by uniformly sampling $H_D(e^{j\omega})$ as follows.

$$H[k] = H_D(e^{j\Delta\omega_N k}), \quad \text{where } \Delta\omega_N \equiv \frac{2\pi}{N}$$

The desired filter response is identical to that derived in Section 3.1.

$$H_D(e^{j\omega}) = \begin{cases} 0, & 0 \leq |\omega| < \omega_{c1} \\ 1, & \omega_{c1} \leq |\omega| \leq \omega_{c2} \\ 0, & \omega_{c2} < |\omega| \leq \pi \end{cases}$$

with the radial cutoff frequencies

$$\omega_{c1} = \frac{\omega_{s1} + \omega_{p1}}{2} = 0.09521542\pi$$

$$\omega_{c2} = \frac{\omega_{p2} + \omega_{s2}}{2} = 0.2455556\pi$$

The problem statement did not ask for an analytic solution for the inverse DFT, so I will numerically compute it using *Matlab's* `ifft.m`. As was done in Section 3, I will verify that the filter achieves the design criteria by computing the DFT of the resulting finite-length FIR filter and plotting the frequency response magnitude. I have included both the filter evaluation and plotting routines in `evalFilter.m` (attached at the end of this writeup).

4.1 Initial Results

As required by Problem 2, Part (a), I evaluated the N -point inverse DFT and obtained the finite-length impulse response. The M -point DFT was then used (with zero-padding) to evaluate whether or not the filter achieves the design criteria. Using a length-143 filter, as predicted by Minzter and Liu's formula clearly does not achieve the design specifications [See Figure 4 and attached high-resolution plots].

In order to meet the design criteria, I found that 19,929 points were required [See Figure 5]! This result is surprising, however it illuminates the central limitation of the frequency sampling method: a lack of tradeoff parameters. With the Kaiser window, we found that a parametric filter design process can optimize the tradeoff between the transition width and the stopband/passband attenuation levels. A "direct" application of the frequency sampling method does not allow one to perform such a tradeoff – resulting in very large filters to meet all design criteria simultaneously. In the following section we will consider a modification which will mitigate this problem.

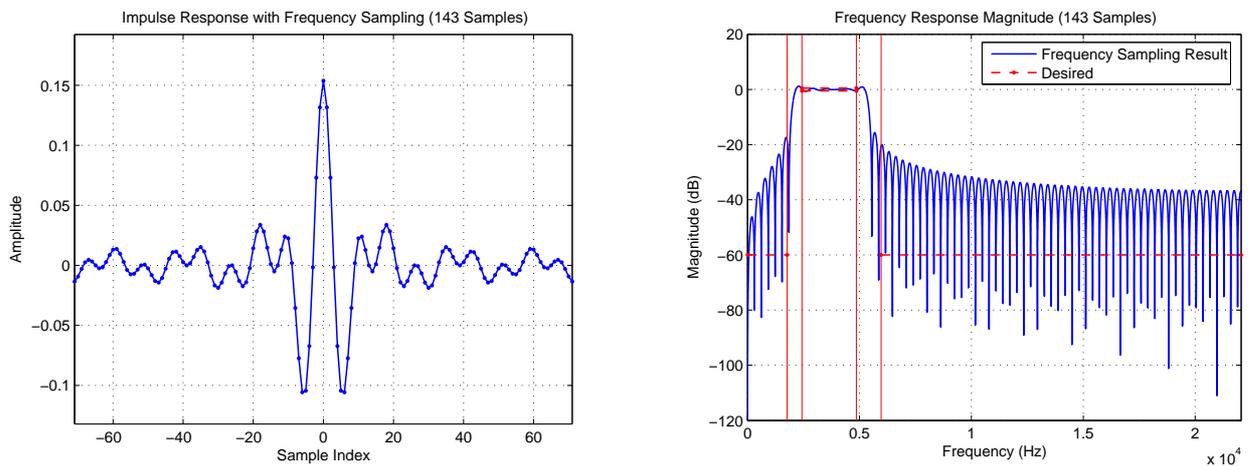


Figure 4: Frequency sampling results using 143 samples.

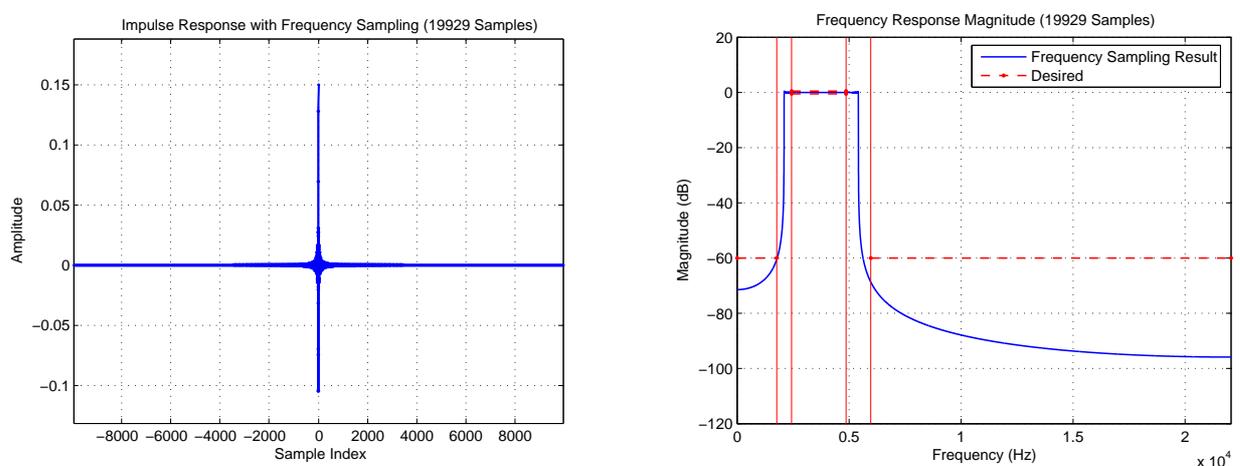


Figure 5: Frequency sampling results using 19,929 samples.

4.2 Optimizing the Transition Samples

As discussed in Problem 2, Part (b), we can reduce the required filter length by “relaxing” several samples in the transition band. In class, it was demonstrated that we could set a single sample in the transition band to $\frac{1}{2}$ to allow the frequency sampling method to have better tradeoffs between stopband/passband attenuation and transition width. This method is discussed, in depth, in [Rabiner, “Theory and Application of Digital Signal Processing”, pp. 108-123, 1975]. In this section I will evaluate the shortest allowable filters using one, two, and three variable transition samples with the frequency sampling method.

As discussed in Rabiner, the optimal values of the transition samples, for a given filter length, can be found using linear programming. Since this problem does not explicitly require implementing this optimal method, I will instead rely upon a non-linear optimization to iteratively refine the transition samples until they meet (or come as close as possible to) the design criteria. Using `evalFilter.m`, I wrote an additional optimization routine `Problem_2.m` to find the best-possible values of a fixed number of transition samples. Fun-

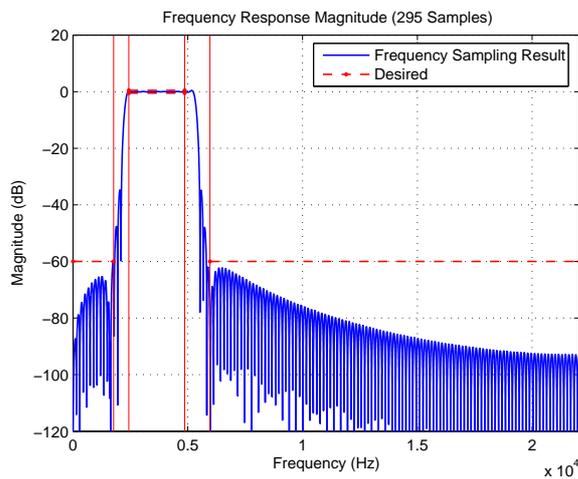


Figure 6: One variable transition sample.

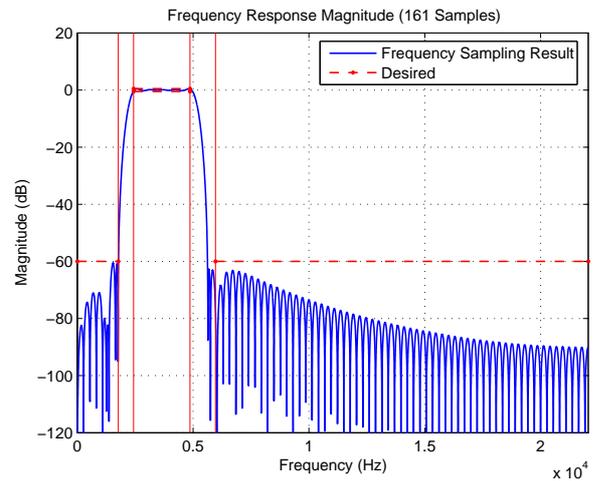


Figure 7: Two variable transition samples.

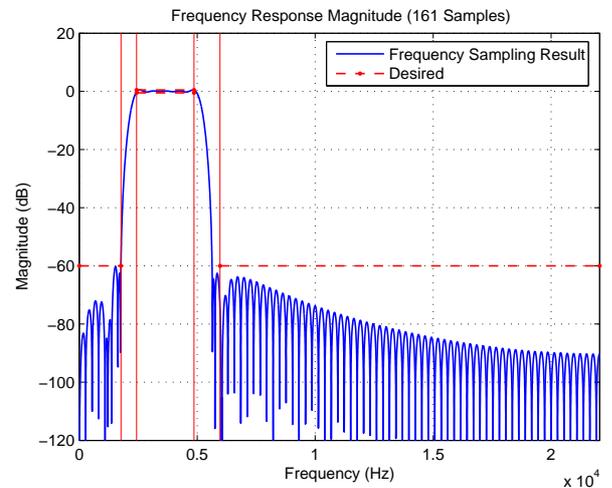
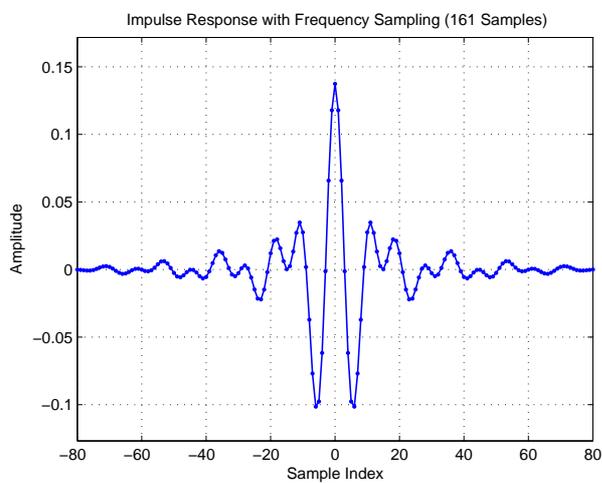


Figure 8: Results using three variable transition samples.

damentally, this program uses *Matlab's* `fminsearch` to solve for the best values of either one, two, or three transition samples. My error function is given by the energy of the frequency response within the prohibited frequency domain (i.e., outside the required stopband and passband attenuation limits).

Following the method outlined in Rabiner, I used variable transition points that were located in the center of each transition band and were symmetric on either side of the passband. Using `Problem_2.m`, I tried decreasing the filter length until the frequency response magnitude satisfied all design criteria. For a single transition sample, I found that a length-295 filter met all design criteria [Figure 6]. For two transition samples, a length-161 filter was sufficient [Figure 7]. Finally, for three transition samples I again found that a length-161 filter was required [Figure 8]. The minimum filter lengths obtained using the frequency sampling method with variable transition samples are tabulated below. The “transition coefficients” column summarizes the values of the transition samples from low to high frequency in the lower transition band (recall they they are symmetric in the upper transition region). *High*

resolution plots are attached in the Appendix.

Transition Samples	Filter Length	Transition Values
Zero	19,929	NA
One	295	0.425
Two	161	0.0412, 0.461
Three	161	0.0450, 0.475, 1.01

5 Optimal Equiripple Method

5.1 Using FDATool

In this section I will apply the Parks-McClellan Algorithm to design an optimal equiripple filter that achieves the design criteria. As described in Problem 3, Part (a), *Matlab's* FDATool can be used to directly input the filter specifications. I have attached my input window in the Appendix. Note that the response type must be set as “Bandpass” and the frequency specifications and magnitude specifications are identical to those in the problem statement. It is also important to note that, in order to “design the optimal equiripple filter exactly for the specifications given”, we should use the “minimum order” mode in FDATool.

5.2 Design Analysis

Using the “minimum order” option, I find that a filter order of 132 is obtained by FDATool. This corresponds to a filter length of 133. In general, the filter order in FDATool is one less than the FIR filter length. The magnitude of the frequency response for the optical length-133 filter is shown in Figure 9.

It is important to note that, while acceptable, the “minimum order” filter provided by FDATool does not strictly satisfy our design criteria – its attenuation in the stopband and passband is insufficient. This can be clearly seen in Figure 9 and the attached plots in the Appendix. In order to strictly meet the design goals, we must use the “specify order” mode in FDATool.

It is important to note that the magnitude specification in the “specify order” mode is weight-based rather than the dB-scale attenuations used in the “minimum order” mode. As discussed in [Mitra, p. 542] we can determine the appropriate weights using the following equations.

$$W(\omega) = \begin{cases} 1, & \text{in the passband} \\ \delta_p/\delta_s \approx 55.939, & \text{in the stopbands} \end{cases}$$

Using these weights in the “specify order” mode allows us to design an optimal filter which exactly meets all design goals. Starting with the Kaiser filter order prediction of 135, I find that a filter order of 138 is required to satisfy all design goals. This corresponds to a length-139 filter! The impulse response and frequency response magnitudes for the optimal design are shown in Figure 10. *Please consult the Appendix for higher-resolution images and specific views of the stopbands and passbands.*

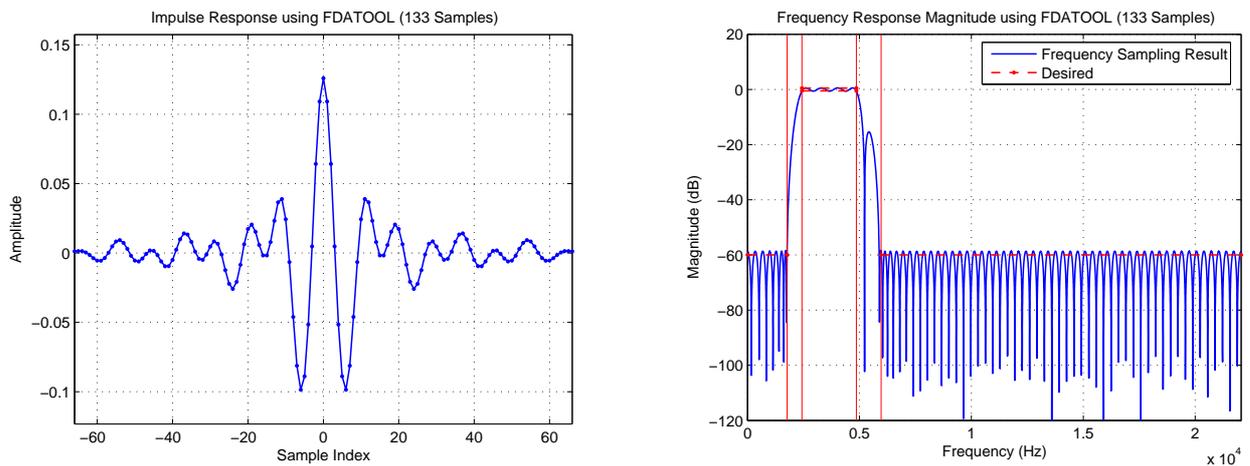


Figure 9: FDATool results using minimum order specification.

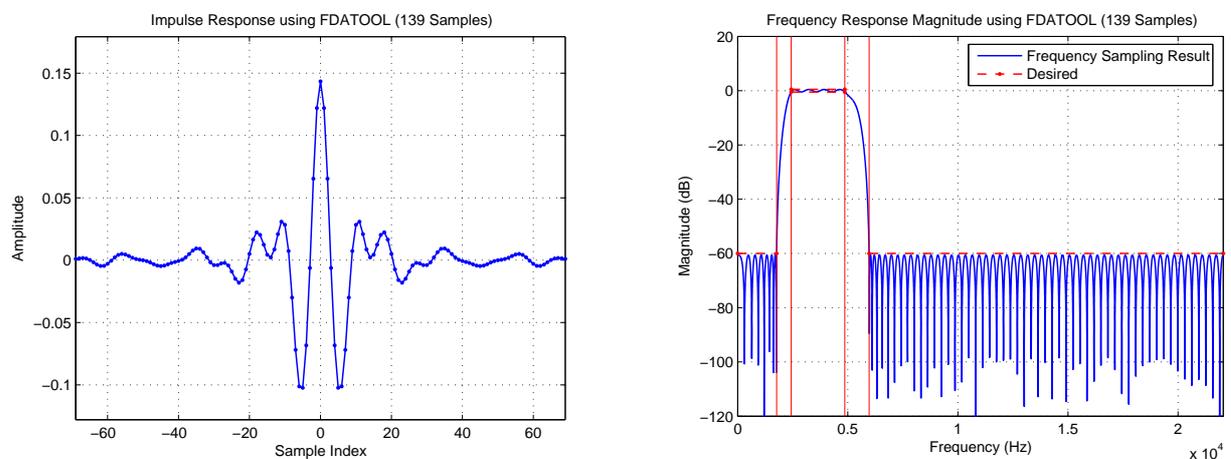


Figure 10: FDATool results using user-specified order of 138.

6 Conclusion

6.1 Comments on Design Methods

In this final section I will summarize my findings, and specifically answer Problem 3, Parts (c) and (d). In this writeup we have utilized three filter design methods: (1) the window method, (2) the frequency sampling method, and (3) optimal equiripple design. In Section 2 we used three filter order prediction formulas to estimate the minimum filter length. Those results are summarized below of convenience.

Prediction Formula	Length	Design Method	Length
Kaiser	136	Window Method	249
Hermann	133	Frequency Sampling	161
Mintzer and Liu	142	Optimal Equiripple	139

The experimentally-determined minimum filter lengths for each method are also shown. First, notice that the optimal equiripple method achieves the shortest filter which meets the

design goals and is within the range of values predicted. In fact, the optimal design is only 3 points longer than that predicted by Kaiser, 6 points longer than Hermann's result, and 3 points shorter than Minter and Liu's estimate. This indicates that the prediction formulas are very useful for gauging the length of an optimal design (although experimental verification will always be required).

In terms of the cost of implementation, the shortest filter is usually the best (all factors being equal) as it requires fewer "multiplication-additions" (i.e., MADDs) to evaluate. As a result, the length-139 filter obtained using `FDATOOL` is the best design.

Problem 3, Part (c) also asks us to evaluate the ease of design for each method. Overall, I believe that *Matlab's* `FDATOOL` is both the easiest to use for filter design and obtains the best results. While the window method and the frequency sampling method are conceptually simpler, they are more difficult to use. For the window method, we must obtain the ideal infinite impulse response which requires analytically evaluating the inverse DTFT integral [See 3.1]. For arbitrary filter designs this task may be difficult or analytically intractable. As a result, the window method mostly serves as a pedagogical tool.

The frequency sampling method is also conceptually simple and fairly straightforward to implement. Since the inverse DFT can be evaluated numerically, as was done in this writeup, it is easier to use than the windowing method. As was discovered in Section 4.2, variable transition samples must be introduced to allow the user to simultaneously optimize the transition width and the stopband/passband attenuation levels. In the most general case, this will require the iterative solution for the optimal transition values. In my implementation I used *Matlab's* `fminsearch` to solve for these parameters, however additional methods such as linear programming could be used. While transition samples improve the results, they increase the implementation complexity – the algorithm is now iterative. As a result, there are few benefits of the frequency sampling method over the optimal equiripple method (in terms of implementation complexity). Finally, the filters obtained with this method are significantly longer than the predicted or optimal values. For our design, a minimum length of 161 was required – 22 samples longer than the optimal equiripple result!

6.2 Would I Really Use This Filter?

This final section will address the limitations of the best filter I obtained: the length-139 optimal equiripple design shown in Figures 9 and 10. As previously discussed, "minimum order" mode in `FDATOOL` did not strictly satisfy the design criteria [See Figure 9], but was very close. If we decided to use this filter there are several limitations we'd have to address. First, the attenuation in the transition band is not smooth – there is a peak within the upper transition region. While we don't specify any strict requirements in the transition regions, we want them to be as smooth as possible – a condition not met by this design. As I have already discussed in Section 5.2, the "engineer's" solution to this problem is to simply increase the filter length until we achieve a smooth transition in each band. This is precisely what was done to obtain the length-139 filter shown in Figure 10. Note that this filter meets all design criteria and also exhibits smooth transition bands.

Since the length-139 filter shown in Figure 10 is my best result, let's briefly address some practical limitations it would have for real-world systems. As we have discussed in class, multirate filter techniques could be applied to reduce the number of operations required to evaluate this bandpass filter. These issues will be more fully explored in Homework #3, so I will only highlight this issue in this writeup.

In addition to saving operations using multirate filtering, we could also employ fixed-point arithmetic on a DSP processor to reduce computation requirements. First, note that all filters designed in this writeup have discrete samples but continuous amplitudes (i.e. they are discrete sequences). We have not yet discussed the limitations imposed by quantizing these filters to put them in a *digital* form appropriate for digital signal processors (DSPs). If we attempt to use these filters without quantizing them to discrete levels, then we will have to employ floating-point arithmetic to implement them in practice. An alternative to this is to quantize the filter ahead of time and employ fixed-point arithmetic for filter evaluation.