

ECE 2025 Fall 2003
Lab #8: Filter Design: Bandpass Filters

Date: 21–27 Oct 2003

You should read the Pre-Lab section of the lab and do all the exercises in the Pre-Lab section before your assigned lab time. You **MUST** complete the online Pre-Post-Lab exercise on Web-CT at the beginning of your scheduled lab session. You can use MATLAB and also consult your lab report or any notes you might have, but you cannot discuss the exercises with any other students. You will have approximately 20 minutes at the beginning of your lab session to complete the online Pre-Post-Lab exercise. The Pre-Post-Lab exercise for this lab includes some questions about concepts from the previous Lab report as well as questions on the Pre-Lab section of this lab.

The Warm-up section of each lab must be completed **during your assigned Lab time** and the steps marked *Instructor Verification* must also be signed off **during the lab time**. After completing the warm-up section, turn in the verification sheet to your TA.

It is only necessary to turn in Section 4 as this week’s lab report.

Forgeries and plagiarism are a violation of the honor code and will be referred to the Dean of Students for disciplinary action. You are allowed to discuss lab exercises with other students and you are allowed to consult old lab reports but the submitted work should be original and it should be your own work.

The lab report for this week will be an **Informal Lab Report**.

The report will **due the next time your lab meets: 28 Oct – 3 Nov**.

1 Introduction

The goal of this lab is to continue our study the response of FIR filters to inputs such as complex exponentials and sinusoids. In the experiments of this lab, you will use `firfilt()`, or `conv()`, to implement filters and `freqz()` to obtain the filter’s frequency response.¹ As a result, you should learn how to characterize a filter by knowing how it reacts to different frequency components in the input.

This lab also introduces the process of “filter design” and then uses that skill to design practical filters: *lowpass*, *highpass* and *bandpass* filters. Bandpass filters can be used to detect and extract information from sinusoidal signals, e.g., individual notes in a musical passage or tones in a touch-tone telephone dialer.

2 Pre-Lab

The filter design process can be understood by examining the *passband* and *stopband* of a lowpass filter.

2.1 Frequency Response of FIR Filters

The output or *response* of a filter for a complex sinusoid input, $e^{j\hat{\omega}n}$, depends on the frequency, $\hat{\omega}$. Often a filter is described solely by how it affects different input frequencies—this is called the *frequency response*.

The frequency response of a general FIR linear time-invariant system is

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

¹If you are working at home and do not have the function `freqz.m`, there is a substitute available called `freeskz.m`. You can get it from the ECE-2025 WebCT page.

For the process of filter design, it is important to recognize that the choice of filter coefficients $\{b_k\}$ determines the frequency response. Thus we will give a formula for the $\{b_k\}$ coefficients that will give a particular behavior for $H(e^{j\hat{\omega}})$.

2.1.1 MATLAB Frequency Response for Lowpass Filter

MATLAB has a built-in function for computing the frequency response of a discrete-time LTI system. For example, the filter coefficients LPF could be given by the formula:

$$h[n] = (0.42 - 0.5 \cos(2\pi n/M) + 0.08 \cos(4\pi n/M)) \left(\frac{\sin(\hat{\omega}_c(n - M/2))}{\pi(n - M/2)} \right) \quad \text{for } n = 0, 1, 2, \dots, M \quad (2)$$

where M is the filter order, which should be an even integer. The first term in $h[n]$ is called a Blackman window; the second one is a “sinc function.” The design parameter $\hat{\omega}_c$ is called the *cutoff frequency* of the filter because it determines the passband and stopband regions of the frequency response (below). 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 for one instance of the filter in (2) as a function of $\hat{\omega}$ in the range $-\pi \leq \hat{\omega} \leq \pi$:

```
M = 50;   nn = 0:M;
blackman = 0.42-0.5*cos(2*pi*nn/M)+0.08*cos(4*pi*nn/M); %-- Blackman window
wc = 0.3*pi;
sincwc = sin(wc*(nn-M/2))./(pi*(nn-M/2)); %-- sinc function
sincwc(M/2 + 1) = wc/pi; %-- fix divide by zero
bb = blackman.*sincwc; %-- Filter Coefficients
ww = -pi:(pi/100):pi; %-- omega hat
HH = freqz(bb, 1, ww); %---freakz.m is an alternative
subplot(2,1,1);
plot(ww, abs(HH))
subplot(2,1,2);
plot(ww, angle(HH))
xlabel('Normalized Radian Frequency')
```

For FIR filters, the second argument of `freqz(, 1, -)` must always be equal to 1.² 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 $H(e^{j\hat{\omega}})$.

2.2 Passband Defined for the Frequency Response

Certain types of digital filters have a frequency response (magnitude) that is either close to one, or close to zero. For example, the plot that would be made by the code in Section 2.1.1 is a lowpass filter whose magnitude is close to one when the frequency $\hat{\omega}$ is near zero. This region is called the *passband* of the filter. It will be useful to have a precise measurement of the passband width so that we can compare different filters.

- From the plot of the magnitude response in Section 2.1.1 determine the set of frequencies where $||H(e^{j\hat{\omega}})| - 1|$ is less than 0.0005. This should be a region of the form $-\hat{\omega}_p \leq \hat{\omega} \leq \hat{\omega}_p$. Determine $\hat{\omega}_p$ for the case where $M = 50$.
- Compare the value of $\hat{\omega}_p$ found in the previous part to the design parameter $\hat{\omega}_c$ in (2). They won't necessarily be equal.

²If 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.

2.3 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 Section 2.1.1, the magnitude is close to zero when the frequency $\hat{\omega}$ is near π (a high frequency). This region is called the *stopband* of the filter. We can make a precise measurement of the stopband edge as follows:

- From the plot of the magnitude response in Section 2.1.1 determine the set of frequencies where $|H(e^{j\hat{\omega}})|$ is less than 0.0005. This should be two regions: $\hat{\omega}_s \leq \hat{\omega} \leq \pi$ in positive frequencies and $-\pi \leq \hat{\omega} \leq -\hat{\omega}_s$ for negative frequencies. Determine $\hat{\omega}_s$ for the case where $M = 50$.
- Compare the value of $\hat{\omega}_s$ found in the previous part to the design parameter $\hat{\omega}_c$ in (2).

2.4 Linear Phase in the Frequency Response

The phase of the frequency response can be related to time delay. In the lowpass filter example of Section 2.1.1 the phase plot appears to be jagged, but it is actually linear.

- Determine the slope of the linear segments of the frequency response for the $M = 50$ filter in Section 2.1.1.
- Plot the impulse response of the digital filter in Section 2.1.1 for the range $n = 0, 1, \dots, M-1$. Then determine the symmetry point for $h[n]$, i.e., find the integer n_s such that $h[n_s + n] = h[n_s - n]$.
- Compare the value of the phase slope to the “symmetry point” of $h[n]$.

2.5 Use the MATLAB FIND Function

Remember that the `find` function returns a list of indices where a logical condition is true. See `help` on `relop` for information. For example, if you have the frequency response from Section 2.1.1 you could use the `find` command to determine the indices where `abs(HH)` is close to one, and then use those indices to display the list of frequencies in the passband.

3 Warm-up

The first objective of this warm-up is to use the MATLAB GUI `dltidemo` to demonstrate the frequency responses of lowpass, highpass and bandpass filters. If you are working in the ECE lab it is **NOT** necessary to install the GUI; otherwise, you must download the ZIP file and *install it into its own directory*. This demo, `dltidemo`, can be downloaded from the web page:

<http://users.ece.gatech.edu/mccllella/matlabGUIs/index.html>

3.1 LTI Frequency Response Demo

The `dltidemo` illustrates the “sinusoid-IN gives sinusoid-OUT” property of LTI systems. In this demo, you can change the amplitude, phase and frequency of an input sinusoid, $x[n]$, and you can change the digital filter that processes the signal. Then the GUI will show the output signal, $y[n]$, which is also a sinusoid (at the same frequency). Figure 1 shows the interface for the `dltidemo` GUI. It is possible to see the formula for the output signal, if you click on the `Theoretical Answer` button located at the bottom-middle part of the window. The digital filter can be changed by choosing different options in the `Filter Specifications` box in the lower right-hand corner.

In the Warm-up, you should perform the following steps with the `dltidemo` GUI:

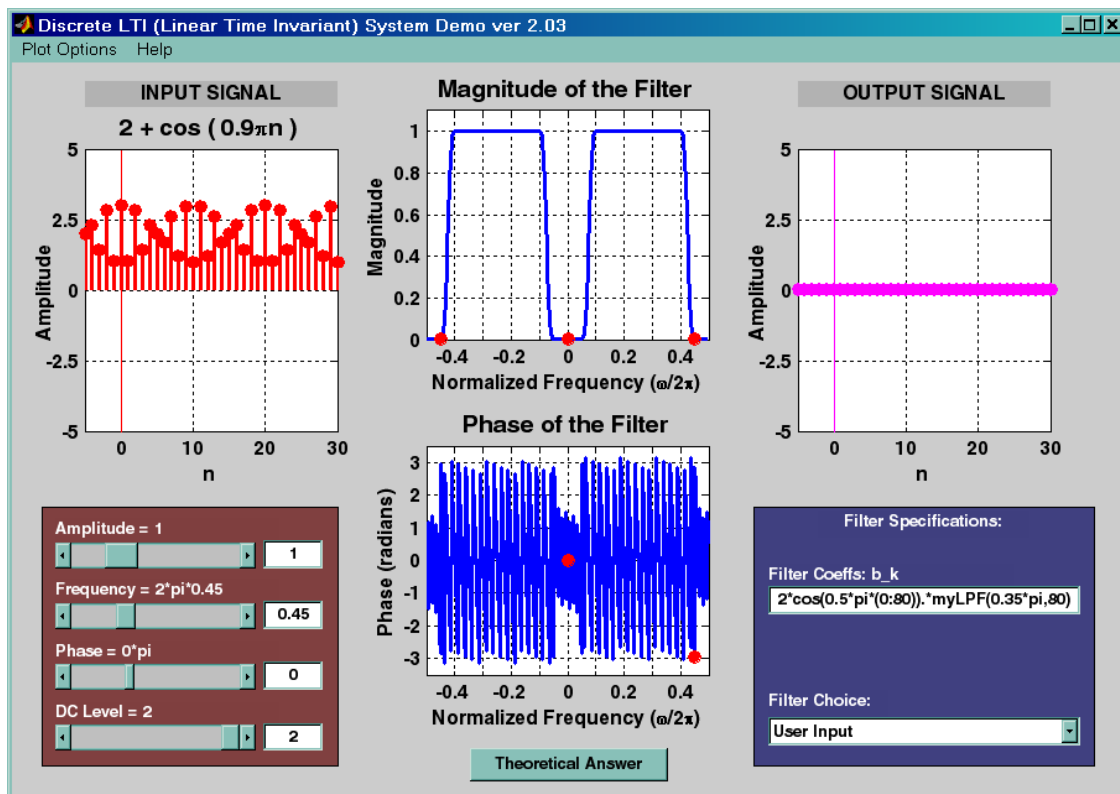


Figure 1: LTI demo interface. When the Filter Choice is set to User Input, MATLAB code can be entered in the text box for the filter coefficients.

- Set the input to $x[n] = 2 + 1 \cos(0.9\pi n)$, which is a signal consisting of one low frequency component and one high frequency component.
- Set the digital filter to be the filter **Lowpass**, $L=15$. Observe the output and determine the exact mathematical formula for the output signal.
- Set the digital filter to be the filter **Highpass**, $L=15$. Observe the output and determine the exact mathematical formula for the output signal.
- Set the digital filter to be the filter **Bandpass**, $L=21$. Observe the output and determine the exact mathematical formula for the output signal.
- Use the frequency response, $H(e^{j\hat{\omega}})$, to explain why the outputs are different in these three cases.

Instructor Verification (separate page)

3.2 MATLAB Function for LPF Design

In this section you must write an M-file that will design a lowpass filter according to the formula given in (2). The M-file should return the filter coefficients given the order (M) and the frequency $\hat{\omega}_c$ as inputs.

- Use the following comments as a template for writing the M-file.

```

function hh = myLPF( wc, M )
%MYLPF   LPF design function
%
%   wc = design parameter giving the approximate location
%       of the passband and stopband edges
%   M = filter order
%   hh = impulse response of the FIR lowpass filter

```

- (b) Demonstrate that your M-file works by using it in the `dltidemo` GUI. There is a “User Input” option as one of the filter choices. You should be able to write a call to `myLPF` in the text box of the user input. Make a length-31 FIR LPF whose cutoff frequency is approximately 0.6π . Recall that the length of an FIR filter is $L = M + 1$.

Instructor Verification (separate page)

3.3 HPF and BPF Design from an LPF

It is possible to obtain the filter coefficients of a highpass filter (HPF), or a bandpass filter (BPF), by performing a simple modification of the filter coefficients from an LPF design. One technique involves multiplying by a cosine. Thus, if $h_L[n]$ is the impulse response of an LPF, the following impulse response:

$$h_B[n] = A h_L[n] \cos(\hat{\omega}_b n) \quad (3)$$

where $\hat{\omega}_b$ determines where the passband of the new filter will be located. For bandpass filters, the constant is $A = 2$, but for the special case of a highpass filter, we must use $A = 1$.

- (a) If we pick $\hat{\omega}_b = \pi$ in (3), the center of the passband moves from $\hat{\omega} = 0$ to $\hat{\omega} = \pi$, and the LPF becomes a highpass filter. Use the `dltidemo` GUI to demonstrate this behavior. In the text box for filter coefficients under the “User Input” option write one line of MATLAB code that calls `myLPF` and multiplies by an appropriate cosine. Make a length-41 FIR highpass filter (HPF) whose cutoff frequency is at approximately $\hat{\omega} = 2\pi(0.35)$. You will have to determine the appropriate cutoff frequency of the LPF in order to get 0.7π for the HPF.

Instructor Verification (separate page)

- (b) Repeat the operations in the previous part to get a length-41 bandpass filter (BPF) whose passband covers the region $0.4\pi \leq \hat{\omega} \leq 0.6\pi$. Explain how to choose $\hat{\omega}_b$ in this case.

4 Lab Exercises

The L -point averaging filter is a lowpass filter. Its passband width is controlled by the filter length L , being inversely proportional to L . In fact, you can use the GUI `dltidemo` to view the frequency response for different length averagers and measure the passband widths. The L -point averager has two shortcomings when a high-performance filter is needed: its passband is not flat and its stopband is not very close to zero. In the warm-up section, an alternative method of lowpass filter design was presented in which reasonably good filters can be obtained with relatively flat passbands and stopband that are within 0.05% of their desired values of one or zero.

4.1 Another Bandpass Filter Design Method

In this lab exercise, a systematic approach to highpass and bandpass filter design will be implemented based on the LPF design function that was written in the warm-up. For highpass filters, we base our approach on Fig. 2 which seems to suggest that a HPF is “one minus an LPF.” In a similar fashion, a bandpass filter can be created via “one minus the sum of an LPF and a HPF.”

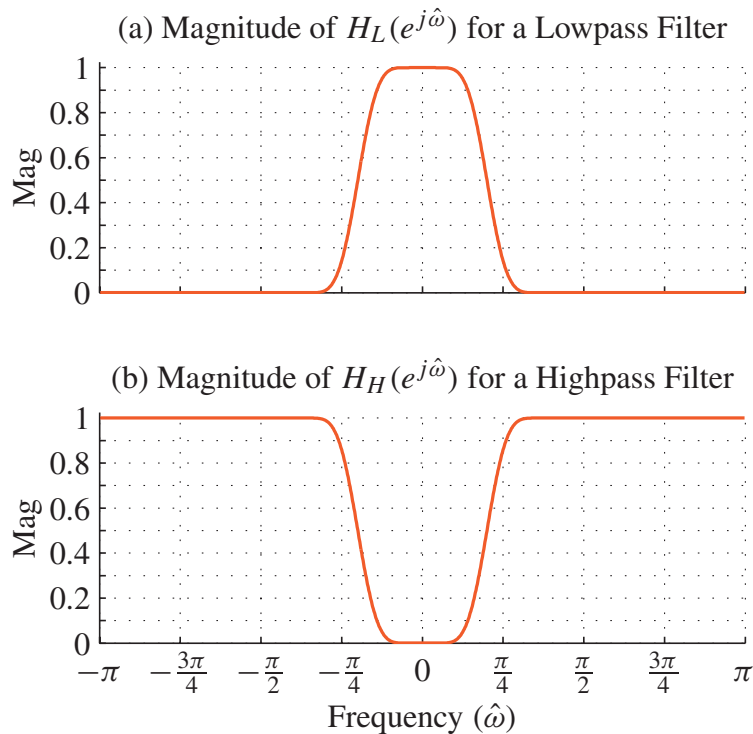


Figure 2: Frequency response magnitudes for (a) LPF and (b) HPF. Notice that $|H_H(e^{j\hat{\omega}})| = 1 - |H_L(e^{j\hat{\omega}})|$.

Note: the filter designs for this section cannot be done using the “multiply by cosine” method of Section 3.3 in the Warm-up.

4.1.1 Highpass Filter Design

For the frequency response of a HPF, we must define both the magnitude and the phase, so we write

$$H_H(e^{j\hat{\omega}}) = 1e^{-j\hat{\omega}n_d} - H_L(e^{j\hat{\omega}}) \quad (4)$$

where the time shift n_d must be chosen consistent with the linear phase of the LPF. In other words, we cannot directly implement “one minus an LPF.”

- Determine the value of n_d needed in (4), as a function of the filter order M . *Explain your answer.*
- Write an M-file called `myHPF(wc, M)` that will create an M^{th} order HPF with a cutoff frequency $\hat{\omega}_c$. This M-file should call the previously written M-file, `myLPF`.
- Plot the frequency response of a length-65 HPF whose cutoff frequency is $\hat{\omega}_c = 0.4\pi$. Determine the passband edge where the frequency response is within 0.05% of the passband value of one; and the stopband edge where the frequency response (magnitude) is less than 0.0005.

4.1.2 Bandpass Filter Design

For the frequency response of a BPF, we can write

$$H_B(e^{j\hat{\omega}}) = 1e^{-j\hat{\omega}n_d} - H_1(e^{j\hat{\omega}}) - H_2(e^{j\hat{\omega}}) \quad (5)$$

where $H_1(e^{j\hat{\omega}})$ is an LPF and $H_2(e^{j\hat{\omega}})$ is a HPF. Both filters must have the same order and the time shift n_d must be chosen consistent with the linear phase of the two filters.

- (a) Write an M-file called `myBPF(wc, M)` that will create an M^{th} order BPF with cutoff frequencies at $\hat{\omega}_1$ and $\hat{\omega}_2$. The first argument `wc` should be a two-element vector containing $\hat{\omega}_1$ (the lower cutoff) and $\hat{\omega}_2$ (the upper cutoff). This M-file should call the previously written M-files, `myLPF` and `myHPF`. Use $\hat{\omega}_1$ and $\hat{\omega}_2$ to determine the cutoff frequencies for the HPF and LPF in Eq. (5).
Note: the “cutoff frequency” is neither the passband edge nor the stopband edge; instead, it is somewhere in between. However, the “cutoff frequency” does control the location of the passband and stopband edges.
- (b) Generate a BPF whose passband covers the region from 0.2π to 0.4π ; and whose stopbands are the regions $[0, 0.16\pi]$ and $[0.44\pi, \pi]$. As before, the definition of the passband and stopband is that the magnitude response be within 0.05% of the true values of one and zero. This will require some trial and error to figure out the appropriate filter order to meet these specifications.
Note: The filter order M must be an even integer.
- (c) Plot the frequency response of the BPF designed in the previous part. On the magnitude plot (zoomed), mark the exact values for the passband edges and the stopband edges.
- (d) Plot the impulse response of the BPF designed in part (b).

4.2 Filtering a Speech Signal

FIR filters can be used to separate out frequency components of a signal. One common method for coding signals is to filter the signal into bands and then code the bands. This approach succeeds in audio coding because the human hearing system is sensitive to frequency. In this section, a speech signal will be filtered into four separate bands and then recombined.

- (a) For this processing we will use the “greasy wash water” speech signal, which is sampled at $f_s = 8000$ Hz. We will denote this speech signal as $s[n]$. It is contained in the zip file `Lab8greasy.zip`.
- (b) Design four filters: an LPF with a cutoff frequency of 1000 Hz; a BPF whose passband extends from a low cutoff frequency of 1000 Hz to a high cutoff of 2000 Hz; a second BPF from 2000 Hz to 3000 Hz; and a HPF whose cutoff frequency is 3000 Hz. Make the order of all the filters $M = 128$. Use the M-files developed in Sections 4.1.1 and 4.1.2. Show the frequency response of the filters; if possible, put them all on one graph, using different colors or line types to tell them apart.
Note: The passbands of the filters will not be exactly equal to the numbers given, but they will be close. On the other hand, the design method described above will allow the BPF to be created from the lowpass and highpass filters.
- (c) Filter the speech signal $s[n]$ through each of the four filters to produce four outputs: $y_L[n]$, $y_H[n]$, $y_{B_1}[n]$, and $y_{B_2}[n]$. Listen to the outputs and describe what you hear. Which one is the most intelligible or sounds the best? (Eliminate the start-up transient if it causes scaling problems in `soundsc`.)
- (d) Form a new signal as the sum of the four filtered signals.

$$z[n] = y_L[n] + y_H[n] + y_{B_1}[n] + y_{B_2}[n]$$

Listen to this signal and compare it to the original speech signal. Comment on what you hear.

- (e) Use the mathematical relationships among the four filters to write a formula for the output $z[n]$ in terms of the original speech signal $s[n]$. *Explain your answer.*
Note: the relationship is **not** $z[n] = s[n]$.

Lab #8

ECE-2025

Fall-2003

INSTRUCTOR VERIFICATION PAGE

For each verification, be prepared to explain your answer and respond to other related questions that the lab TA's or professors might ask. Turn this page in at the end of your lab period.

Name: _____

Date of Lab: _____

Part 3.1(b,c,d,e) Use the `dltdemo` GUI to demonstrate lowpass, highpass and bandpass filters. For each filter, write the expressions for the output $y[n]$ in the space below.

$$y_L[n] =$$

$$y_H[n] =$$

$$y_B[n] =$$

Verified: _____

Date/Time: _____

Part 3.2(b) Demonstrate that your M-file for LPF design works correctly by using it in the `dltdemo` GUI to obtain a length-31 LPF.

Verified: _____

Date/Time: _____

Part 3.3 (a) Exhibit a highpass filter obtained by multiplying a cosine times $h[n]$ from an LPF design. Use the `dltdemo` GUI to obtain a length-41 HPF.

Verified: _____

Date/Time: _____