GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2025 Fall 2003
Lab #7: Frequency Response: Nulling Filters

Date: 7–20 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: 21–27 Oct**.

---

1 Introduction

The goal of this lab is to 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.

2 Pre-Lab

This lab also introduces a practical filter, the nulling filter. Nulling filters can be used to remove sinusoidal interference, e.g., jamming signals in a radar or communication system.

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*.

For example, the frequency response of the two-point averaging filter

$$y[n] = \tfrac{1}{2}x[n] + \tfrac{1}{2}x[n-1]$$

[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*, or get it from the ECE-2025 WebCT page.

can be found by using a general complex exponential as an input and observing the output or response.

$$x[n] = Ae^{j(\hat{\omega}n + \phi)} \tag{1}$$

$$y[n] = \tfrac{1}{2}Ae^{j(\hat{\omega}n + \phi)} + \tfrac{1}{2}Ae^{j(\hat{\omega}(n-1) + \phi)} \tag{2}$$

$$= Ae^{j(\hat{\omega}n + \phi)}\tfrac{1}{2}\left\{1 + e^{-j\hat{\omega}}\right\} \tag{3}$$

In (3) there are two terms, the original input, and a term that is a function of $\hat{\omega}$. This second term is the frequency response and it is commonly denoted by $H(e^{j\hat{\omega}})$.[2]

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \tfrac{1}{2}\left\{1 + e^{-j\hat{\omega}}\right\} \tag{4}$$

Once the frequency response, $H(e^{j\hat{\omega}})$, has been determined, the effect of the filter on any complex exponential may be determined by evaluating $H(e^{j\hat{\omega}})$ at the corresponding frequency. The output signal, $y[n]$, will be a complex exponential whose complex amplitude has a constant magnitude and phase. The phase of $H(e^{j\hat{\omega}})$ describes the phase change of the complex sinusoid and the magnitude of $H(e^{j\hat{\omega}})$ describes the gain applied to the complex sinusoid.

The frequency response of a general FIR linear time-invariant system with filter coefficients $\{b_k\}$ is

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \sum_{k=0}^{M} b_k e^{-j\hat{\omega}k} \tag{5}$$

2.1.1 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 a two-point averaging system as a function of $\hat{\omega}$ in the range $-\pi \le \hat{\omega} \le \pi$:

```
bb = [0.5, 0.5];        %-- Filter Coefficients
ww = -pi:(pi/100):pi;   %-- omega hat
H = freqz(bb, 1, ww);   %<--freekz.m is an alternative
subplot(2,1,1);
plot(ww, abs(H))
subplot(2,1,2);
plot(ww, angle(H))
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 $H(e^{j\hat{\omega}})$. Note: we will always use capital `H` for the frequency response.[3]

2.2 Periodicity of the Frequency Response

The frequency responses of discrete-time filters are *always* periodic with period equal to 2π. Explain why this is the case by stating a definition of the frequency response and then considering two input sinusoids whose frequencies are $\hat{\omega}$ and $\hat{\omega} + 2\pi$.

$$x_1[n] = e^{j\hat{\omega}n} \qquad \text{versus} \qquad x_2[n] = e^{j(\hat{\omega}+2\pi)n}$$

Consult Chapter 6 for a mathematical proof that the outputs from each of these signals will be identical (basically because $x_1[n]$ is equal to $x_2[n]$.) **The implication of periodicity is that a plot of $H(e^{j\hat{\omega}})$ only has to be made over the interval $-\pi \le \hat{\omega} \le \pi$.**

[2]The notation $H(e^{j\hat{\omega}})$ is used in place of $\mathcal{H}(\hat{\omega})$ for the frequency response because we will eventually connect this notation with the z-transform, $H(z)$, in Chapter 7.

[3]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.

2.3 Frequency Response of the Four-Point Averager

In Chapter 6 we examined filters that compute the average of input samples over an interval. These filters are called "running average" filters or "averagers" and they have the following form for the L-point averager:

$$y[n] = \frac{1}{L}\sum_{k=0}^{L-1} x[n-k] \tag{6}$$

(a) Use Euler's formula and complex number manipulations to show that the frequency response for the 4-point running average operator is given by:

$$H(e^{j\hat{\omega}}) = \mathcal{H}(\hat{\omega}) = \frac{2\cos(0.5\hat{\omega}) + 2\cos(1.5\hat{\omega})}{4}\, e^{-j1.5\hat{\omega}} \tag{7}$$

(b) Implement (7) directly in MATLAB. Use a vector that includes 400 samples between $-\pi$ and π for $\hat{\omega}$. Since the frequency response is a complex-valued quantity, use `abs()` and `angle()` to extract the magnitude and phase of the frequency response for plotting. Plotting the real and imaginary parts of $H(e^{j\hat{\omega}})$ is not very informative.

(c) In this part, use `freqz.m` in MATLAB to compute $H(e^{j\hat{\omega}})$ numerically (from the filter coefficients) and plot its magnitude and phase versus $\hat{\omega}$. Write the appropriate MATLAB code to plot both the magnitude and phase of $H(e^{j\hat{\omega}})$. Follow the example in Section 2.1.1. The filter coefficient vector for the 4-point averager is defined via:

```
bb = 1/4*ones(1,4);
```

Note: the function `freqz(bb,1,ww)` evaluates the frequency response for all frequencies in the vector `ww`. It uses the summation in (5), not the formula in (7). The filter coefficients are defined in the assignment to vector `bb`. How do your results compare with part (b)?
Note: the plots should not be identical, but you should be able to explain why they are equivalent.

2.4 The MATLAB FIND Function

Often signal processing functions are performed in order to extract information that can be used to make a decision. The decision process inevitably requires logical tests, which might be done with `if-then` constructs in MATLAB. However, MATLAB permits vectorization of such tests, and the `find` function is one way to determine which elements of a vector meet a certain logical criterion. In the following example, `find` extracts all the numbers that "round" to 3:

```
xx = 1.4:0.33:5, jkl = find(round(xx)==3), xx(jkl)
```

The argument of the `find` function can be any logical expression, and `find` returns a list of indices where that logical expression is true. See `help` on `relop` for information.
Now, suppose that you have a frequency response:

```
ww = -pi:(pi/500):pi; HH = freqz( 1/4*ones(1,4), 1, ww );
```

Use the `find` command to determine the indices where `HH` is zero, or very small. Then use those indices to display the list of frequencies where `HH` is zero. Since there might be round-off error in calculating `HH`, the logical test should be a test for those indices where the magnitude (absolute value in MATLAB) of `HH` is less than some rather small number, e.g., 1×10^{-6}. Compare your answer to the frequency response that you plotted for the four-point averager in Section 2.3.

3 Warm-up

The first objective of this warm-up is to use a MATLAB GUI to demonstrate nulling. 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`, is part of the *SP-First Toolbox*, or it can be downloaded from the web page: `http://users.ece.gatech.edu/mcclella/matlabGUIs/index.html`

3.1 LTI Frequency Response Demo

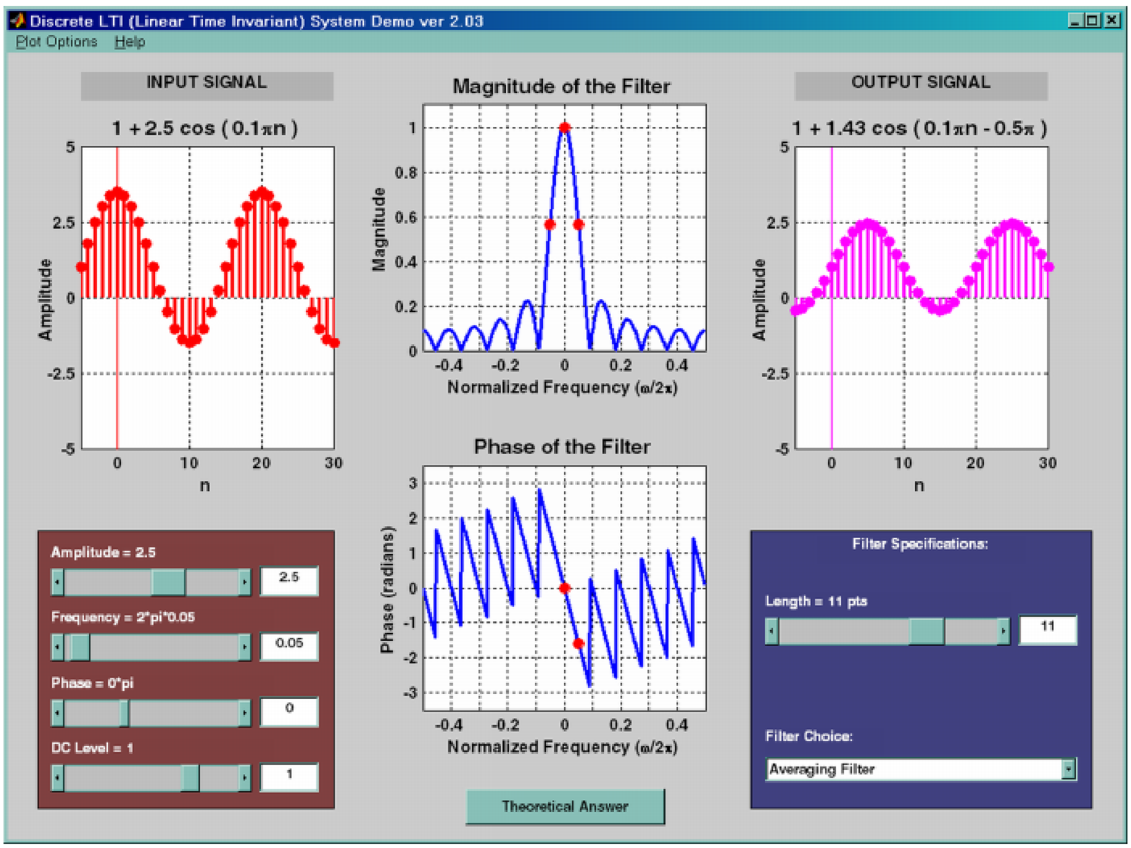

Figure 1: DLTI demo interface.

The `dltidemo` GUI 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:

(a) Set the input to $x[n] = 1.5\cos(0.1\pi(n-4))$

(b) Set the digital filter to be a 9-point averager.

(c) Determine the formula for the output signal and write it in the form: $y[n] = A\cos(\hat{\omega}_0(n - n_d))$.

(d) Using n_d for $y[n]$ and the fact that the input signal had a peak at $n = 4$, determine the amount of delay through the filter. In other words, how much has the peak of the cosine wave shifted?

Instructor Verification (separate page)

(e) Now, determine the length of the averaging filter so that the output will be zero, i.e., $y[n] = 0$. Use the GUI to show that you have the correct filter to zero the output. If the filter length is more than 15, you will have to enter the "Filter Specifications" with the `user Input` option.

(f) When the output is zero, the filter acts as a *Nulling Filter*, because it eliminates the input at $\hat{\omega} = 0.1\pi$. Which other frequencies $\hat{\omega}$ are also nulled? Find at least one.

Instructor Verification (separate page)

3.2 Cascading Two Systems

More complicated systems are often made up from simple building blocks. In Fig. 3, two FIR filters are shown connected "in cascade."

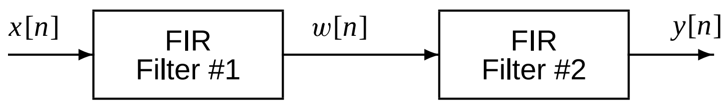

Figure 2: Cascade of two FIR filters.

Assume that the system in Fig. 3 is described by the two equations

$$w[n] = \sum_{\ell=0}^{M} \alpha^{\ell} x[n-\ell] \qquad \text{(FIR FILTER \#1)}$$

$$y[n] = w[n] + w[n-2] \qquad \text{(FIR FILTER \#2)}$$

(a) Use `freqz()` in MATLAB to get the frequency responses for the case where $\alpha = -1$ and $M = 8$. Plot the magnitude and phase of the frequency response for Filter #1, and also for Filter #2. Which one of these filters is a *highpass filter*?

(b) Filter #2 is a "nulling filter." Determine the frequency $\hat{\omega}$ that is removed by Filter #2.

(c) Plot the magnitude and phase of the frequency response of the overall cascaded system.

(d) Explain how the individual frequency responses in part(a) are combined to get the overall frequency response in part(b). Comment on the magnitude combinations as well as the phase combinations.

Instructor Verification (separate page)

3.3 Debugging

In the following MATLAB function, `myfilt.m`:

```
function bb = myfilt(alph,M)
% create filter coefficients, a little bit at a time
bb = 0;
for kk = 3:M
  bb = [-alph*bb, alph.^(M:-1:kk)];
end
```

show that you can use the MATLAB debugger to stop *during* the second iteration of the loop and plot the frequency response for the filter coefficients after the second iteration is complete. Suppose that the function is called from the command line via: `hh = myfilt(-0.98,8).`

Instructor Verification (separate page)

4 Lab Exercises

4.1 Nulling Filters for Interference Rejection

Nulling filters are filters that are able to completely eliminate some frequency component. If the "nulled" frequency is $\hat{\omega} = 0$ or $\hat{\omega} = \pi$, then a two-point FIR filter will do the nulling. The simplest possible general nulling filter can have as few as three coefficients. If $\hat{\omega}_\text{n}$ is the desired nulling frequency, then the following length-3 FIR filter

$$y[n] = x[n] - 2\cos(\hat{\omega}_\text{n})x[n-1] + x[n-2] \tag{8}$$

will have a zero in its frequency response at $\hat{\omega} = \pm\hat{\omega}_\text{n}$. For example, a filter designed to completely eliminate signals of the form $Ae^{\pm j0.5\pi n}$ would have the following coefficients because the input frequency is $\hat{\omega} = \pm 0.5\pi$.

$$b_0 = 1, \quad b_1 = -2\cos(0.5\pi) = 0, \quad b_2 = 1.$$

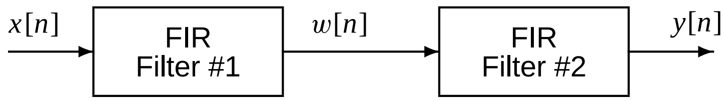

Figure 3: Cascade of two FIR nulling filters.

(a) We can remove more than one sinusoid by connecting FIR nulling filters "in cascade" as shown in Fig. 3. Design a filtering system that consists of the cascade of two FIR nulling filters that will eliminate the following frequencies: $\hat{\omega} = 0.15\pi$, and $\hat{\omega} = 0.35\pi$. For this part, derive the filter coefficients of both nulling filters.

(b) Plot the magnitude and phase of the frequency response $H(e^{j\hat{\omega}})$ of the overall cascaded system. Notice that the value of $H(e^{j\hat{\omega}})$ at $\hat{\omega} = 0$ is not equal to one. However, it is possible to scale the filter coefficients of one of the nulling filters to make $H(e^{j0}) = 1$. For example, if Filter #1 has coefficients $\{\beta_0,\ \beta_1,\ \beta_2\}$ and if we scale Filter #1 to have new coefficients $\{\frac{1}{2}\beta_0,\ \frac{1}{2}\beta_1,\ \frac{1}{2}\beta_2\}$, then the frequency response values would be half as big, i.e., the new frequency response would be $\frac{1}{2}H(e^{j\hat{\omega}})$.

Derive the scaling necessary to make the overall cascaded system have a frequency response that is one at DC. Determine the new values for the coefficients of Filter #1. Finally, plot the frequency response (magnitude and phase) of the cascaded system with scaling—this is the only plot you need to turn in for this part.

(c) Generate an input signal $x[n]$ that is the sum of three sinusoids plus a DC term:

$$x[n] = 20 + 100\cos(0.15\pi n) + 30\cos(0.25\pi n - \pi/3) + 50\cos(0.35\pi n)$$

Make the input signal 100 samples long over the range $0 \le n \le 99$.

(d) Use `firfilt` (or `conv`) to filter the signal $x[n]$ through the filters designed in part (b). Show the MATLAB code that you wrote to implement the cascade of two FIR filters.

(e) Make a plot of the output signal—show the first 41 points, i.e., $0 \le n \le 40$. Then, determine (by hand) the exact mathematical formula (magnitude, phase and frequency) for the output signal for $n \ge 5$. In a second plot, show that the MATLAB plot of the output signal matches this mathematical formula for $5 \le n \le 40$.

(f) The output signal will be different for the first few points because there is a "start-up" or "transient" region for the output. How many "start-up" points are found, and how is this number related to the lengths of the filters designed in part (a)? Hint: consider the length of a single FIR filter that is equivalent to the cascade of two length-3 FIRs.

4.2 Removing Hum from a Speech Signal

FIR filters can be used to reject interfering signals that are sinusoidal. One situation where this might occur is in a tape recording of speech where the recorder is not adequately isolated from the power line signal which is a 60-Hz sinusoid. The recorded signal is actually the sum of two signals: the desired speech signal and a sinusoid, $A\cos(120\pi t + \phi)$. In this section, you will design an FIR nulling filter to remove interfering sinusoid, and also assess how much the desired signal is distorted by the nulling process.

(a) Load the file `speechbad` which contains one signal, `xxbad`, which is the sum of a speech signal plus very large amplitude sinusoids at 1555 Hz and 2222 Hz. The sinusoids start and stop during the utterance. The sampling rate of this signal is 8000 Hz, and the good speech signal was scaled so that its maximum value is one. Listen to this signal to verify that the interference is so strong that the speech is not recognizable.

(b) Design an FIR nulling filter to remove the sinusoids completely. This can be accomplished by finding the numerical values of the filter coefficients. Scale the filter coefficients so that the overall frequency response has a magnitude of 4.0 at $\hat{\omega} = \pi$, i.e., $|H(e^{j\pi})| = 4$.

(c) Plot the frequency response of the nulling filter designed in the previous part. Decide whether the frequency response is lowpass, highpass, bandreject, or bandpass.

(d) Process the corrupted signal, `xxbad`, through the nulling filter. Listen to the result and assess how successful the processing was. How well was the interference removed? Describe the artifacts that remain after the processing.
Note: Do not use the MATLAB function `soundsc()` to listen to the output, because it takes its scaling from the largest value in the output signal. Since the FIR filter has "start-up" and "ending" regions of length $M - 1$, the output scaling would be dominated by these values which happen to be very large in this case.[4] In addition, these startup and ending values occur in the middle of the processed speech, the only way to ignore these values is by using the ***unscaled function*** **`sound.m`** region. The original speech was scaled to have a maximum amplitude of one, and with the frequency response of the filter scaled to a maximum magnitude of 4.0, the speech output will have a maximum amplitude that is close to one. Therefore, it will not require scaling to be heard; but the transients of the FIR filter will be clipped by the `sound.m` function.

(e) Comment on whether or not the speech signal corrupted in the process. Use the frequency response (magnitude) to explain how the speech signal was corrupted. Your explanation will be qualitative, but should focus on whether the low frequency or high frequency regions of the speech were altered.

[4] The technical term for the "start-up" region is the *transient response* of the system, because the filter response passes through this region before getting into its steady-state response where the output is a pure sinusoid, or sum of sinusoids.

Lab #7
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(d) Use the `dltidemo` GUI to illustrate the operation of a 9-point averaging filter. Determine the amount of delay through the filter, and write your answer in the space below.

Verified:____________________ Date/Time:__________

Part 3.1(f) Use the `dltidemo` GUI to find a digital FIR filter that will null the input signal. Determine the filter length, and write your answer in the space below. Also determine which frequencies are nulled by the filter.

Verified:____________________ Date/Time:__________

Part 3.2 Plot the frequency response of the two filters in the cascade combination, and then explain how the magnitudes are combined and how the phases are combined to get the overall filter. Check the range of frequencies ($\hat{\omega}$) used for the plot.

Verified:____________________ Date/Time:__________

Part 3.3 Use the debugger to stop execution and plot the frequency response:

Verified:____________________ Date/Time:__________