

ECE 2025 Spring 2000
Lab #7: Frequency Response & Nulling Filters

Date: 28 Feb.–2 March 1999

This is *the official* Lab #7 description.

The Warm-up section of each lab must be completed in Lab and the steps marked *Instructor Verification* must also be signed off **during the lab time**.

The lab report for this lab will be **INFORMAL**: discuss your results from section 3. Staple the **Instructor Verification** sheet to the end of your lab report.

The report will **due on Wednesday, 15 March, for the Mon/Wed. sections and on Thursday, 16-March, for the Tuesday/Thursday sections.**

1 Introduction

The goal of this lab is to study the response of FIR filters to inputs such as complex exponentials and sinusoids. In addition, we will use FIR filters to study properties such as linearity and time-invariance. 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 a practical application where sinusoidal signals are used to transmit information: a touch-tone dialer. Bandpass FIR filters can be used to extract the information encoded in the waveforms.

1.1 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

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

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

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

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

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

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

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

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

²The 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.

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 describes the phase change of the complex sinusoid and the magnitude describes the gain applied to the complex sinusoid.

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

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

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 $-\pi \leq \hat{\omega} \leq \pi$:

```
bb = [0.5, 0.5];           %-- Filter Coefficients
ww = -pi:(pi/100):pi;     %-- omega hat
H = freqz(bb, 1, ww);     %<--freakz.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 1.³ 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.

1.2 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 $\hat{\omega} + 2\pi$.

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

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 $\mathcal{H}(\hat{\omega})$ only needs to extend over the interval $-\pi \leq \hat{\omega} \leq \pi$.

2 Warm-up

2.1 Frequency Response of the Four-Point Averager

In Chapter 6 we examined filters that average input samples over a certain interval. These filters are called “running average” filters or “averagers” and they have the following form for the L -point averager:

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

³If 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.

- (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:

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

Instructor Verification (separate page)

- (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 $H(e^{j\hat{\omega}})$. The filter coefficient vector for the 4-point averager is defined via:

$$\text{bb} = 1/4 * \text{ones}(1, 4);$$

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)?

Instructor Verification (separate page)

2.2 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 do lots of tests at once. In the following example, `find` extracts all the numbers that “round” to 3:

$$\text{xx} = 1.4:0.33:5, \text{ jkl} = \text{find}(\text{round}(\text{xx})==3), \text{ xx}(\text{jkl})$$

The argument of the `find` function can be any logical expression. Notice that `find` returns a list of indices where the logical condition is true. See `help` on `relop` for information.

Now, suppose that you have a frequency response:

$$\text{ww} = -\text{pi}:(\text{pi}/100):\text{pi}; \text{ HH} = \text{freqz}((1/11)*\text{ones}(1,11), 1, \text{ww});$$

Use the `find` command to determine the indices where `HH` is zero, and 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 of `HH` is less than some rather small number, e.g., 1×10^{-6} . Explain your answer to the TA by also plotting the magnitude of `HH` which is the 11-point averager discussed in Chapter 6 (and in the lecture).

Instructor Verification (separate page)

3 Lab Exercises

3.1 Nulling Filters for Interference Rejection

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

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

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

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

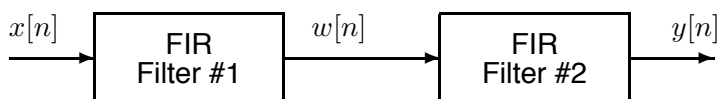


Figure 1: Cascade of two FIR nulling filters.

- We can remove more than one sinusoid by connecting FIR nulling filters “in cascade” as shown in Fig. 1. Design a filtering system that consists of the cascade of two FIR nulling filters that will eliminate the following frequencies: $\hat{\omega} = 0.3\pi$, and $\hat{\omega} = 0.4\pi$. For this part, derive the filter coefficients of both nulling filters.
- Plot the magnitude and phase of the frequency response $\mathcal{H}(\hat{\omega})$ of the overall cascaded system. Notice that the value of $\mathcal{H}(\hat{\omega})$ 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 $\mathcal{H}(0) = 1$. 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 $\frac{1}{2}\mathcal{H}(\hat{\omega})$.
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.

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

$$x[n] = 10 + 10 \cos(0.1\pi n) + 20 \cos(0.3\pi n - \pi/3) + 15 \cos(0.4\pi n - \pi/4)$$

Make the input signal 150 samples long over the range $0 \leq n \leq 149$.

- Use `firfilt` (or `conv`) to filter the signal $x[n]$ through the filters designed in parts (a) and (b). Show the MATLAB code that you wrote to implement the cascade of two FIR filters.
- Make a plot of the output signal—show the first 40 points. Determine (by hand) the exact mathematical formula (magnitude, phase and frequency) for the output signal for $n \geq 5$. Show that the MATLAB plot of the output signal matches this mathematical formula for most of the first 40 points.
- Explain why the output signal is different for the first few points. 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.

3.2 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 a sinusoidal interference, and also assess how much the desired signal is distorted by the nulling process.

- (a) Load the file `speech7` which contains one signal, `x2c100`, which is the sum of a speech signal plus a sinusoid at 100 hertz. The utterance is the same one found in `lab6dat`: “Oak is strong and also gives shade.” The sampling rate of this signal is 8000 Hz. Listen to this signal to verify that the interference is so strong that the speech is barely recognizable.
- (b) Design an FIR nulling filter to remove the sinusoid completely. This can be Gives the values of the filter coefficients.
- (c) Plot the frequency response of the nulling filter designed in the previous part.
- (d) Process the corrupted signal, `x2c100`, through the nulling filter. Listen to the result and assess how successful the processing was. How well was the interference removed? Was the speech signal corrupted in the process?

Note: the MATLAB function `soundsc()` takes its scaling from the largest value in the output signal. Since the FIR filter has a “start-up” region for $n = 0, 1, \dots, M - 1$, the output scaling might be dominated by these values. Therefore, you should ignore these values by working with the output signal $y[n]$ for $n \geq M$, i.e., use the colon notation in MATLAB to get `yy(start:end)` where `start` is some index after the “start-up” region.⁴ In addition, the same sort of “edge effect” happens at the end of the signal when the input signal is finite length, so you should remove the last M points from the output signal.

- (e) 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.

⁴The 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
Spring-2000
INSTRUCTOR VERIFICATION PAGE

Staple this page to the end of your Lab Report.

Name: _____ Date of Lab: _____

Part 2.1 Mathematical derivation of the frequency response of a 4-point averager:

Verified: _____ Date/Time: _____

Part 2.1 Plot the frequency response of a 4-point averager—magnitude and phase. Check the range of frequencies ($\hat{\omega}$) for the plot.

Verified: _____ Date/Time: _____

Part 2.2: Using the MATLAB `find` function to locate frequency response features (for the 11-pt. averager) automatically. Where do you expect to find the zeros of this frequency response?

Verified: _____ Date/Time: _____