GEORGIA INSTITUTE OF TECHNOLOGY
SCHOOL of ELECTRICAL and COMPUTER ENGINEERING

ECE 2026 Spring 2018
Lab #8: Frequency Response of FIR Filters

Date: 12–15 Mar. 2018

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.

Verification: The In-Lab 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**. When you have completed a step that requires verification, simply raise your hand and demonstrate the step to the TA or instructor. After completing the warm-up section, turn in the verification sheet to your TA *before leaving the lab.*

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, but you cannot give or receive any written material or electronic files. In addition, you are not allowed to use or copy material from old lab reports from previous semesters. Your submitted work must be your own original work.

1 Introduction

The goal of this lab is to study the sinusoidal response of some simple FIR filters in MATLAB. This leads to a study of the frequency response function.

1. In the experiments of this lab, you will use the MATLAB GUI called `dltidemo` to the frequency response function for FIR filters. If you have installed the *SP-First* Toolbox, you will already have this demo on the `matlabpath`.

2. You will also study the ***Sinusoid-In Gives Sinusoid-Out*** property of LTI systems.

1.1 Overview

The goal of this lab is to study the frequency response. For FIR filters this is the response to inputs such as complex exponentials and sinusoids. Although you can use `firfilt()`, or `conv()`, to implement filters in the *time domain*, the function `freqz()` is used to characterize the filter in the *frequency domain* via its frequency response.[1] As a result, you will learn how to obtain the output when the input is a sum of sinusoids because the frequency response characterizes a filter by telling how the filter responds to different frequency components in the input.

1.2 Frequency Response of FIR Filters

The output or *response* of a filter for a complex sinusoid input, $e^{j\hat{\omega}n}$, is a complex exponential at the same frequency. The magnitude and phase of the output will be different, and that change depends on the frequency, $\hat{\omega}$. The dependence of these magnitude and phase changes versus frequency is called the *frequency response*. In effect, the filter is described by how it affects different input frequencies. 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*.

can be found by using a general complex exponential as an input $x[n]$ and observing the output or *response.*

$$\begin{aligned} x[n] &= Ae^{j(\hat{\omega}n+\varphi)} && (1)\\ y[n] &= \tfrac{1}{2}Ae^{j(\hat{\omega}n+\varphi)} + \tfrac{1}{2}Ae^{j(\hat{\omega}(n-1)+\varphi)} && (2)\\ &= Ae^{j(\hat{\omega}n+\varphi)}\left\{\tfrac{1}{2}+\tfrac{1}{2}e^{-j\hat{\omega}}\right\} && (3)\end{aligned}$$

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[2] by $H(e^{j\hat{\omega}})$.

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

The general form of the frequency response for an M-th order FIR linear time-invariant system with filter coefficients $\{b_k\}$ is

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

Once the frequency response, $H(e^{j\hat{\omega}})$, has been determined, the effect of the filter on any complex exponential input may be determined by evaluating $H(e^{j\hat{\omega}})$ at the corresponding input 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.

2 Pre-Lab

The goal of this lab is to learn about frequency response of FIR filters in MATLAB, and then study the response of FIR filters to various signals.

2.1 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 frequency vector
H = freqz(bb, 1, ww);   %<--freekz(bb,1,ww) is an alternative
subplot(2,1,1);
plot(ww, abs(H)), grid on
subplot(2,1,2);
plot(ww, angle(H)), grid on
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] 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`, `Hz`, 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.

2.2 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 using the definition of the frequency response (5) and then considering two input sinusoids whose frequencies are $\hat{\omega}$ and $\hat{\omega} + 2\pi$.

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

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

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 can be written as:

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

(b) Implement (7) directly in MATLAB. Use a vector that includes 400 samples covering the interval $[-\pi, \pi)$ for $\hat{\omega}$. Make plots of $C(\hat{\omega})$ and $\psi(\hat{\omega})$ versus $\hat{\omega}$. It is tempting to think that $C(\hat{\omega})$ is the magnitude of the frequency response, but $C(\hat{\omega})$ can go negative, so these are not plots of the magnitude and phase. You would have to use `abs()` and `angle()` to extract the magnitude $|H(e^{j\hat{\omega}})|$ and phase $\angle H(e^{j\hat{\omega}})$ of the frequency response for plotting.

(c) In this part, use `freqz.m` or `freekz.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. The filter coefficient vector for the 4-point averager is defined via:

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

Recall that 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 by converting the minus sign in the negative values of $C(\hat{\omega})$ to a phase of π radians, which then modifies the phase plot with jumps of $\pm\pi$ radians.

2.4 FIR Nulling Filters

A simple second-order FIR filter can be used to remove a sinusoid from an input signal. The general form for the filter coefficients of an FIR nulling filter is

$$b_0 = 1 \qquad b_1 = -2\cos(\hat{\omega}_{\text{NULL}}) \qquad b_2 = 1 \tag{8}$$

Nulling means that the frequency response will be zero at $\hat{\omega} = \hat{\omega}_{\text{NULL}}$. With $\hat{\omega}_{\text{NULL}} = 0.75\pi$, enter the filter coefficients in the `dltidemo` GUI to see the frequency response; verify that it is zero at $\hat{\omega} = 0.75\pi$. In addition, define the input to be $x[n] = 0.4 + 1.5\cos(0.75\pi n)$ and observe that the sinusoidal component is not present in the output.

2.5 Ideal Filters and Practical Filters

Ideal Filters are given by a frequency response, consisting of *perfect* passbands and stopbands. These are not actually FIR filters because there is no finite set of filter coefficients that will produce the ideal frequency response. In the `dltidemo` GUI, you can choose ideal lowpass filters (LPF), highpass filters (HPF) and bandpass filters (BPF).
The ideal LPFs and HPFs have one parameter for the cutoff frequency which determines the boundary between the ideal passband and the ideal stopband. The ideal BPF has a parameter for center frequency which determines where the band is located; its bandwidth (in this GUI) is always 0.4π. All the ideal filters have an additional parameter for the slope of the phase of $H(e^{j\hat{\omega}})$.

(a) Use the `dltidemo` to view the *sinusoid-in gives sinusoid-out* behavior of an ideal bandpass filter (BPF). Choose the center frequency to be 0.4π. Then the two bandedge frequencies for the BPF will be $\hat{\omega}_\ell = 0.2\pi$ and $\hat{\omega}_u = 0.6\pi$. These bandedges define the extent of the ideal passband of the BPF.

(b) Set the input to be $x[n] = 1.5 + 0.9\cos(0.55\pi n)$. Determine which frequency components are present in the output signal.

(c) You can also include an ideal linear phase in the frequency response, so choose the phase slope as -2. Then find the output for the same input as in the previous part. Describe how the output has changed; relate the delay in the output to the phase slope of the frequency response.

(d) When the input is $x[n] = 1.5 + 0.9\cos(0.65\pi n)$, determine the output. Explain why it is zero.

Practical Filters which are approximations to the ideal filters by length-L FIR filters. The ones shown in `dltidemo` were designed using MATLAB's `fir1` function for digital filter design. The GUI has length-15 LPFs and HPFs, and length-21 BPFs. The LPF and HPF designs have one parameter for the cutoff frequency which determines the boundary between the non-ideal passband and stopband. The BPF has a parameter for center frequency which determines where the passband is located. The default bandedges are $\pm 0.2\pi$ from the center frequency, but since this filter is not ideal the frequency response at the bandedges has a magnitude of 0.5.
Notation: the cutoff frequency for HPFs and LPFs will be called $\hat{\omega}_c$.

Note: The running average FIR filter is an *approximate* lowpass filter, but it is not a very good one because it is not very close to an ideal LPF. Better filters can be designed with computer-aided optimization algorithms, or with MATLAB's `fir1` function.

(a) Use the `dltidemo` to view the *sinusoid-in gives sinusoid-out* behavior of an ideal bandpass filter (BPF). Choose the center frequency for the BPF to be $\hat{\omega} = 0.4\pi$. Then the desired bandedges of the passband will be $\hat{\omega}_\ell = 0.2\pi$ and $\hat{\omega}_u = 0.6\pi$. Since the filter is not ideal, these two bandedges define an approximate extent of the BPF's passband.

(b) Set the input to be $x[n] = 1.5 + 0.9\cos(0.65\pi n)$. Determine the output signal. Compare to the output obtained in part (d) for Ideal Filters. Explain why is is not zero.

(c) The output signal might contain both frequency components. Relate the amplitudes of the output signal's two frequency components to the non-ideal nature of the frequency response. Right-click to get values from the frequency response plot.

3 In-Lab Exercises

The main objective of this lab is to use a MATLAB GUI to demonstrate the frequency response, as well as the *sinusoid-in gives sinusoid-out* property of LTI systems. If you are working in the ECE lab it is **NOT** necessary to install the GUI; otherwise, you should install the *SP-First* toolbox. The frequency response demo, `dltidemo`, is part of the *SP-First Toolbox*.

3.1 LTI Frequency Response Demo

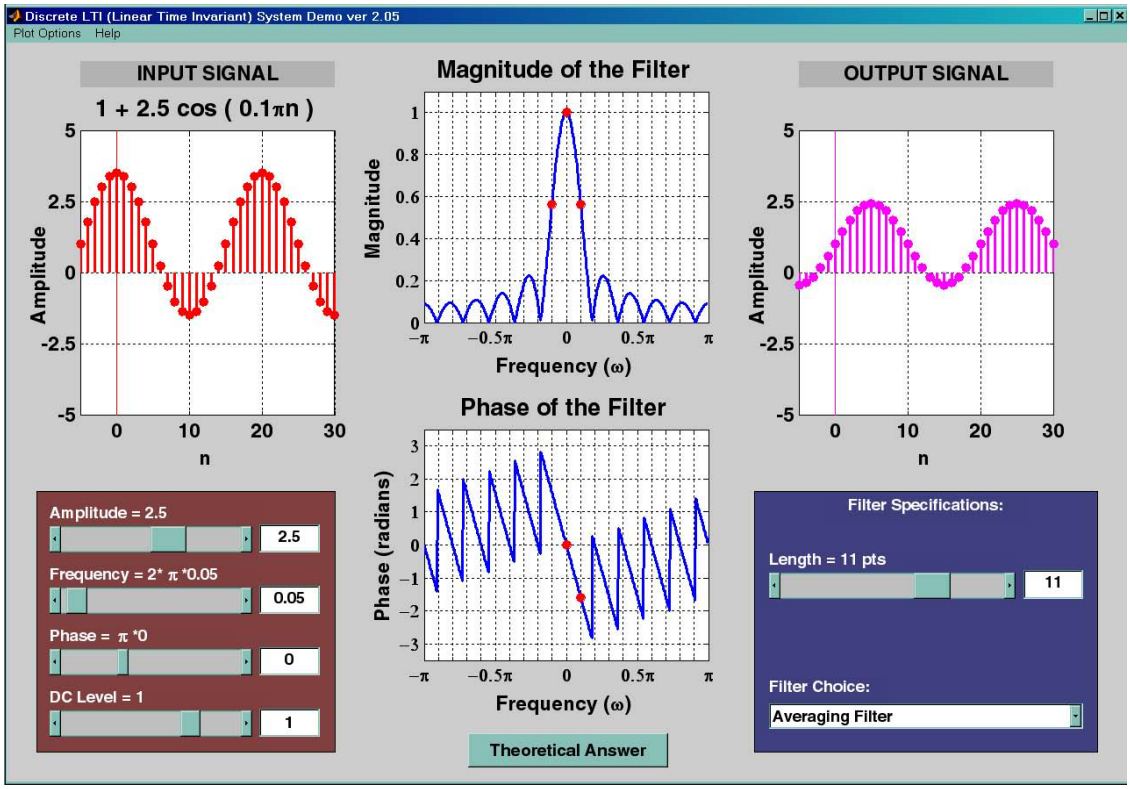

Figure 1: DLTI demo interface. The frequency label should be $\hat{\omega}$, but MATLAB won't display the hat in $\hat{\omega}$.

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; just click on the `Theoretical Answer` button located at the bottom-middle part of the window. The digital filter can be changed by choosing different `Filter Choice` options in the `Filter Specifications` box in the lower right-hand corner.
Perform the following steps with the `dltidemo` GUI:

(a) Set the input to $x[n] = 1.8\cos(0.2\pi(n-3))$; note that this sinusoid has a positive peak at $n = 3$.

(b) Set the digital filter to be a 7-point averager. Using the middle panels that show the frequency response, measure the magnitude and phase of the frequency response at the input frequency. Right-click to get values from the frequency response plot.

(c) Give an equation that explains how the time delay is related to the phase of $H(e^{j\hat{\omega}})$ at $\hat{\omega} = 0.2\pi$. Remember that phases differing by integer multiples of 2π are the same.

(d) Convert the phase to time-index delay. Then you can determine a formula for the output signal that can be written in the form: $y[n] = A\cos(\hat{\omega}_0(n - n_7))$, where n_7 is an integer. Using n_5 from $y[n]$ and the fact that the input signal had a peak at $n = 3$, it should be easy verify how much the peak of the cosine wave has been shifted. This is called the *time delay* through the filter.

Instructor Verification (separate page)

(e) Now, change the frequency of the input signal so that the output will be exactly zero. Sinusoidal components at other frequencies $\hat{\omega}$ will also be nulled—make a list of these nulled frequencies in the

range $0 \leq \hat{\omega} \leq \pi$. There are many choices for this frequency; list them all.
Hint: Recall the Dirichlet form for the frequency response of the averaging filter, and where it has regularly spaced zeros versus $\hat{\omega}$.

(f) When the output of an FIR filter is zero, the FIR filter acts as a *Nulling Filter* for a certain input frequency. Design a second-order nulling filter that will remove the cosine component from the following signal: $x[n] = 1 + \cos(0.4\pi n)$. List the three filter coefficients of the nulling filter. Also, the DC component will not be removed, so determine the DC level of the output signal.

Instructor Verification (separate page)

3.2 Ideal Filters and Practical Filters

In `dltidemo`, it is possible to choose from two classes of filters: ideal filters and practical filters.

3.2.1 Ideal Filters

(a) Define the input signal to be $x[n] = 1.8\cos(0.46\pi n)$.

(b) LPF: Set the filter to be an ideal LPF with a cutoff frequency of $\hat{\omega}_c = 2\pi(0.29)$. Determine a value for the phase slope so that the output will be delayed by 3, i.e., $y[n] = 1.8\cos(0.46\pi(n-3))$. In addition, get the formula for the output signal from the *Theoretical Answer*. Explain why the phase of the *theoretical* output signal is not equal to -1.38π. Consider multiples of 2π in the phase which must be taken into account when relating the phase to the delay.

(c) HPF: Change to an ideal HPF; use the same phase slope as in the previous part. Determine the minimum value of the cutoff frequency so that the output signal will be zero.

Instructor Verification (separate page)

3.2.2 Practical Filters

Since ideal filters cannot be implemented with numerical computation, it is necessary to study how much degradation there will be when actual *implementable* filters are used. In this section, the filters are order-14 FIR filters with 15 filter coefficients. For both tests, use the same input signal as before: $x[n] = 1.8\cos(0.46\pi n)$. When you observe the output signal, think about the following question: Do you expect the signal to be in the stopband or passband of the filter, i.e., do you expect the output to be zero or to be equal to the input?

(a) LPF: Set the filter type to a length-15 LPF, with its cutoff frequency at $\hat{\omega}_c = 2\pi(0.29)$. The cutoff frequency determines the boundary between the stopband and the passband. Use the GUI to determine the output signal *passed* by the LPF. Comment on how close this filter is to the ideal.

(b) In the lowpass case, the output can be written as $y[n] = A_{\text{out}}\cos(0.46\pi(n-n_d))$, where n_d represents a delay. Use the time-domain plots, or the phase slope and phase value to determine n_d which must be an integer. Comment on the degradation of the output amplitude from the ideal.

(c) HPF: Set the filter type to a length-15 HPF, with its cutoff frequency at $\hat{\omega}_c = 2\pi(0.29)$. The cutoff frequency determines the boundary between the stopband and the passband. Use the GUI to determine how well the output signal *rejected* by the HPF versus an ideal filter, i.e., determine the amplitude of the output signal which should be close to zero.

Instructor Verification (separate page)

4 Lab Report: Removing Interference 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 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, $A\cos(120\pi t + \varphi)$. In this section, you will design an FIR nulling filter as shown in Section 2.4 to remove interfering sinusoids, 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. Make a spectrogram (in dB).

(b) Design a cascade of two FIR nulling filters to remove the sinusoids completely. This can be accomplished by finding the numerical values of the filter coefficients for each second-order nulling filter. Combine the cascaded filters into one equivalent FIR filter, and give the filter coefficients of the equivalent filter.

(c) Plot the magnitude frequency response of the cascaded nulling filter designed in the previous part. Indicate the frequencies where the nulls are found.

(d) Process the corrupted signal, `xxbad`, through the nulling filters. Make two spectrograms (in dB): one for the input signal and the other for the output signal. Point out features that verify that the nulling filters operated correctly.

Lab #8
ECE-2026 Spring-2018
WORKSHEET & 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: _____ gtLoginUserName: _____ Date: _____

Part	Observations
3.1(b)	Magnitude and phase of the frequency response of the length-7 averager, at the input frequency.
3.1(c)	Give an equation that explains how the filter's *delay* is related to the phase of $H(e^{j\hat{\omega}})$ at $\hat{\omega} = 0.2\pi$.
3.1(d)	Formula for output from a length-7 averager written in the form $y[n] = A\cos(\hat{\omega}_0(n - n_5))$

Verified:_____ Date/Time:_____

3.1(e)	List of all frequencies nulled by the running-average FIR filter in part (b), or give a formula.
3.1(e)	Length-3 nulling filter, list three filter coefficients. Give the DC level of the output signal.

Verified:_____ Date/Time:_____

3.2.1(b)	Phase slope for Ideal HPF. Explain why phase of $y[n]$ formula is not equal to -1.38π?
3.2.1(c)	Cutoff frequency for Ideal HPF to get $y[n] = 0$.

Verified:_____ Date/Time:_____

3.2.2(a)	Output of length-15 LPF with $\hat{\omega}_c = 2\pi(0.29)$ which can be expressed as $A_{\text{out}}\cos(\hat{\omega}_0 n + \varphi)$
3.2.2(b)	LPF output: Determine the delay n_d so that the output can be expressed with the formula, $A_{\text{out}}\cos(\hat{\omega}_0(n - n_d))$
3.2.2(c)	Output of length-15 HPF with $\hat{\omega}_c = 2\pi(0.29)$. Give formula and explain why $y[n]$ is not exactly zero.

Verified:_____ Date/Time:_____

Lab #8
ECE-2026 Spring-2018
LAB REPORT QUESTION

Turn this page in to your lab grading TA at the very beginning of your next scheduled Lab time.

Name: _____ gtLoginUserName: _____ Date: _____

Update on lab report policy: (1) Explain your approach or results to earn partial credits; and (2) Turn in your code if required in obtaining your results.

Part 4: Design an FIR nulling filter to remove two interfering sinusoids, 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. Make a spectrogram (in dB).

(b) Design a cascade of two FIR nulling filters to remove the sinusoids completely. This can be accomplished by finding the numerical values of the filter coefficients for each second-order nulling filter. Combine the cascaded filters into one equivalent FIR filter, and give the filter coefficients of the equivalent filter.

(c) Plot the frequency response of the cascaded nulling filter designed in the previous part. Indicate the frequencies where the nulls are found.

(d) Process the corrupted signal, `xxbad`, through the nulling filters. Make two spectrograms (in dB): one for the input signal and the other for the output signal. Point out features that verify that the nulling filters operated correctly.