**Department of Electrical Engineering and   
Computer Science**

**Faculty Member:** Dr. Ahmad Salman  **Dated:** 06/03/2023

**Semester:** 6th **Section:** BEE 12C

**EE-330 Digital Signal Processing**

Lab 6: Frequency Response and Nulling Filters

Group Members

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | **PLO4 - CLO4** | | **PLO5 -CLO5** | **PLO8 -CLO6** | **PLO9 -CLO7** |
| **Name** | **Reg. No** | **Viva / Quiz / Lab Performance** | **Analysis of data in Lab Report** | **Modern Tool Usage** | **Ethics and Safety** | **Individual and Teamwork** |
|  |  | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** | **5 Marks** |
| Danial Ahmad | 331388 |  |  |  |  |  |
| Muhammad Umer | 345834 |  |  |  |  |  |
| Tariq Umar | 334943 |  |  |  |  |  |
|  |  |  |  |  |  |  |

Table of Contents

[2 Sampling of Audio Signals in MATLAB 3](#_Toc130078565)

[2.1 Objectives 3](#_Toc130078566)

[2.2 Introduction 3](#_Toc130078567)

[2.3 Software 3](#_Toc130078568)

[2.4 Lab Report Instructions 3](#_Toc130078569)

[3 Lab Procedure 5](#_Toc130078570)

[3.1 Introduction 5](#_Toc130078571)

[3.1.1 MATLAB Function for Frequency Response 5](#_Toc130078572)

[3.1.2 Lab Task 1: Frequency Response of the Four-Point Averager 6](#_Toc130078573)

[3.1.3 The MATLAB FIND Function 8](#_Toc130078574)

[3.1.4 Cascading Two Systems 8](#_Toc130078575)

[3.1.5 Lab Task 2 10](#_Toc130078576)

[3.1.6 Nulling Filters for Rejection 10](#_Toc130078577)

[4 Conclusion 13](#_Toc130078578)

# Sampling of Audio Signals in MATLAB

## Objectives

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. As a result, you should learn how to characterize a filter by knowing how it reacts to different frequency components in the input

* Introduction to bandpass filters
* Introduction to Nulling filters
* Cascade systems and their frequency response
* How to extract information from sinusoidal signals

## Introduction

The purpose of this lab is to study the response of Finite Impulse Response (FIR) filters to inputs such as complex exponentials and sinusoids. FIR filters are commonly used in digital signal processing applications to remove noise, smooth signals, or extract specific frequency components. Understanding how filters react to different frequency components in the input is essential in characterizing a filter and using it effectively. In this lab, we will use conv() to implement filters and freqz() to obtain the filter's frequency response.

## Software

MATLAB is a high-level programming language and numerical computing environment. Developed by MathWorks, it provides an interactive environment for numerical computation, visualization, and programming. MATLAB is widely used in various fields, including engineering, science, and finance, due to its capabilities for matrix and vector operations, implementation of algorithms, and creation of graphical representations of data. The objective of this lab is to provide a hands-on experience with the A-to-D sampling and the D-to-A reconstruction processes that are essential for digital image processing. We will also demonstrate a commonly used method of image zooming (reconstruction) that produces “poor” results, which will help illustrate the importance of understanding the underlying principles of digital image processing.

## Lab Report Instructions

All questions should be answered precisely to get maximum credit. Lab report must ensure following items:

* Lab objectives
* MATLAB codes
* Results (graphs/tables) duly commented and discussed
* Conclusion

# Lab Procedure

## Introduction

This lab also introduces two practical filters: bandpass filters and nulling filters. Bandpass filters can be used to detect and extract information from sinusoidal signals, e.g., tones in a touch-tone telephone dialer. Nulling filters can be used to remove sinusoidal interference, e.g., jamming signals in a radar.

### MATLAB Function for Frequency Response

MATLAB has a built-in function called freqz()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 in the range - to

For FIR filters, the second argument of freqz(,1,) must always be equal to 1.

b = [0.5, 0.5]; % filter coeffs

w = -pi:(pi / 100):pi;

H = freqz(b, 1, w); % <--freekz.m is an alternative

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid



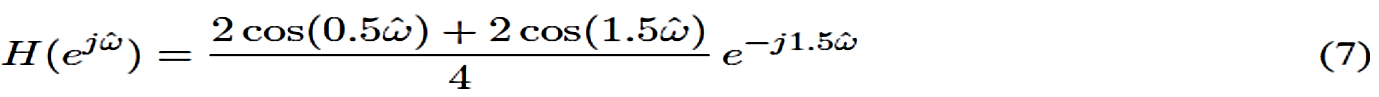


### Lab Task 1: Frequency Response of the Four-Point Averager

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



The frequency response for the 4-point running average operator is given by:



1. Implement (7) directly in MATLAB. Use a vector that includes 400 samples between –π and π for 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 is not very informative.

n = -5:5;

x\_n = (n == 0);

L = 4;

y\_n = *zeros*(1, length(n));

w = -pi:(pi / 400):pi;

H = ((2\*cos(0.5\*w) + 2\*cos(1.5\*w))/4).\*exp(-1i\*1.5\*w);

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid





1. In this part, use freqz.m in MATLAB to compute numerically (from the filter coefficients and plot its magnitude and phase versus . Write the appropriate MATLAB code to plot both the magnitude and phase of Follow the example in Section 8.1.2. 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)?

b = 1/4\**ones*(1, 4);

H = freqz(b, 1, w);

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid





**Comments:** We obtain the same magnitude and phase response using either method.

### 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:

xx = 1.4:0.33:5,

jkl = find(round(xx)==3),

xx(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:

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, 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 probably 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 average in Section 8.1.4.

mag\_zeros =

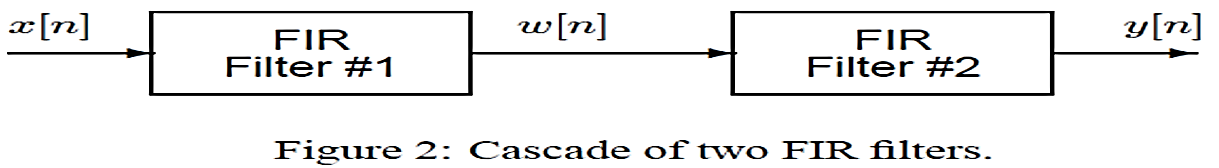
     1   201   601   801

The above vector ‘mag\_zeros’ aligns with the four point average magnitude plot, and we obtain zeros at and .

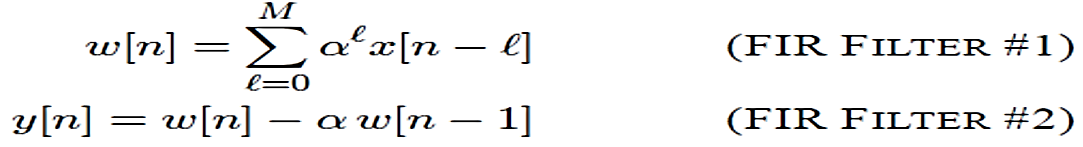
### Cascading Two Systems

More complicated systems are often made up from simple building blocks. In Fig. 2, two FIR filters are

shown connected “in cascade.”



Assume that the system in Fig. 2 is described by the two equations



1. Use freqz() in MATLAB to get the frequency responses for the case where α = 0.8 and M = 9. Plot the magnitude and phase of the frequency response for Filter #1, and for Filter #2. Which one of these filters is a *lowpass filter*?

n = 0:9;

alpha = 0.8;

**Filter 1**

b1 = alpha.^n;

w = -pi:(pi / 400):pi;

H1 = freqz(b1, 1, w);

figure

subplot(2, 1, 1);

plot(w, abs(H1))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H1))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid

**Filter 2**

b2 = [1, -alpha, *zeros*(1, 8)];

H2 = freqz(b2, 1, w);

figure

subplot(2, 1, 1);

plot(w, abs(H2))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H2))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid

|  |  |
| --- | --- |
| Filter 1 | Filter 2 |

1. Plot the magnitude and phase of the frequency response of the overall cascaded system.

H = H1.\*H2;

figure

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid





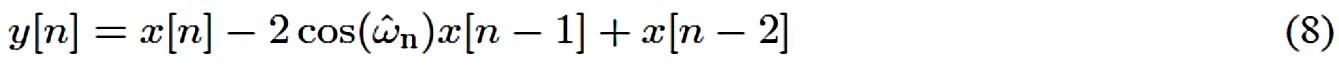
1. Explain how the individual frequency responses in part(a) are combined to get the overall frequency response in part (b). Comment on the magnitude combinations as well as the phase combinations.

Filter 1 and Filter 2 are in series (cascade) arrangement, hence the individual frequencies are multiplied to obtain the overall cascaded system. The overall response is obtained by multiplying the impulse response of Filter 1 and Filter 2 with each other as convolution in time domain is multiplication in frequency domain.

### Lab Task 2

### Nulling Filters for Rejection

Nulling filters are filters that completely eliminate some frequency component. If the frequency is = 0 or = π, then a two-point FIR filter will do the nulling. The simplest possible general nulling filter can have as few as three coefficients. If is the desired nulling frequency, then the following length-3 FIR filter



will have a zero in its frequency response at = . For example, a filter designed to completely eliminate signals of the form would have the following coefficients because we would pick the desired nulling frequency to be = 0.5π. b0 = 1, b1 = −2 cos(0.5π) = 0, b2 = 1.

1. Design a filtering system that consists of the *cascade of two FIR nulling filters* that will eliminate the following input frequencies: = 0.44π, and = 0.7π. For this part, derive the filter coefficients of both nulling filters.

b\_p44pi = [1 -2 \* cos(0.44 \* pi) 1];

b\_7pi = [1 -2 \* cos(0.7 \* pi) 1];

b = conv(b\_044, b\_07);

w = -pi:(pi / 400):pi;

H = freqz(b, 1, w);

figure

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

grid

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')

grid





1. Generate an input signal x[n] that is the sum of three sinusoids:



Make the input signal 150 samples long over the range 0 < n < 149.

b\_p44pi = [1 -2 \* cos(0.44 \* pi) 1];

b\_7pi = [1 -2 \* cos(0.7 \* pi) 1];

b = conv(b\_044, b\_07);

w = -pi:(pi / 400):pi;

H = freqz(b, 1, w);

figure

subplot(2, 1, 1);

plot(w, abs(H))

title('Magnitude Response')

subplot(2, 1, 2);

plot(w, angle(H))

title('Phase Response')

xlabel('Normalized Radian Frequency')



1. Use filter to filter the sum of three sinusoids signal x[n] through the filters in part (a).

y\_n = filter(b, 1, x\_n);

1. 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 ≥ 5.



stem(n, y\_n)

title('Task 2.c')

xlabel('Sample Points [n]')

grid

Formula:

1. Plot the mathematical formula determined in (d) with MATLAB to show that it matches the filter output from filter over the range 5 ≤n ≤ 40.

n\_obv = 5:40;

figure

filtered\_x\_n = 10\*cos(0.3\*pi\*(n\_obv-2));

stem(n\_obv, filtered\_x\_n)

title('Task 2.d')

xlabel('Sample Points [n]')

grid



1. 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.

The output signal is different for the first few points because of the start-up transient; when two FIR filters are cascaded, the resulting filter has a length equal to the sum of the individual filter lengths minus one. In our case, the length of each filter is 3, so the length of the cascade filter is 5. Since the length of the filter is 5, the first 4 points will be affected by the filter’s initial conditions.

# Conclusion

In this lab, we have studied the response of FIR filters to various inputs, including complex exponentials and sinusoids. We have learned how to characterize a filter by its frequency response and how to design and implement bandpass and Nulling filters. We have also explored the concept of cascade systems and learned how to analyze their frequency response.