DSP - Laboratory 2

Annalisa Gallina
annalisa.gallina.1@studenti.unipd.it

October 30, 2023

Abstract

Second laboratory experience [1] for the course Digital Signal Processing (DSP) a.y. 23/24. Topics: FIR filters design.

Before starting with the exercises make sure you have installed **Signal Processing Toolbox** (`https://www.mathworks.com/products/signal.html`.), and make sure you have headphones handy for the second exercise.

1 FIR Notch filters for tones removal in audio signals

Caveat: Please use headphones when listening to the audio signal. It is highly suggested to start with a low volume to avoid the sound bothering you.

In this exercise we are going to design a simple FIR filter (actually, a cascade of FIR filters) to remove annoying sounds (sinusoidal tones) from an audio signal. Although the impulse response of the filter is very simple, we can still achieve the desired denoising operation. An alternative approach would be to design a band stop filter (willing students can enjoy trying out), hereafter the straightforward approach (with simple impulse response) is adopted. Figure 1 shows the typical shape of the magnitude of the frequency response of a notch filter used to attenuate a particular frequency.

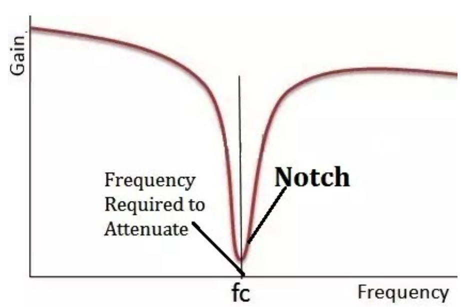

Figure 1: Example: magnitude of the frequency response of a notch filter.

Objective: Write a MATLAB program that removes unwanted tones from an audio file. The file *'SunshineSquare.wav'* has had some unwanted tones added. Our job is to remove the tones so you can hear the message in a clearer way.

Approach: There at two steps needed to remove the tones. First, determine the frequencies of the interfering tones, and second, filter out those frequencies.

To read the audio file and plot the spectrogram of it (`https://www.mathworks.com/help/signal/ref/spectrogram.html`). Use the following code.

[1] All information and material presented here is intended to be used for educational or informational purposes only. Credits to Daniele Orsuti daniele.orsuti@phd.unipd.it

```
[xx, fs] = audioread('SunshineSquare.wav');
figure;
spectrogram(xx,fs);
```

Note, fs is the sampling rate of the wav file and is important to specify it.

A weighted three-point averager is enough to remove one frequency at a time. Given the impulse response:

$$h[n] = [1, A, 1]$$

find the frequency response $H(z) = H(e^{j\hat{\omega}})$ in terms of A (see the first task). Find the values of A needed to remove each of the unwanted frequencies. Once you have the correct values, this code can be used to remove one frequency at a time:

```
hh = [1, AA, 1];
yy = filter(hh, 1, xx);
```

You will have to fill in your values for **AA**. You can check the frequency response of your filter by using freqz:

```
ww = -pi:pi/100:pi; % 1/100 is the frequency step
HH = freqz(hh, 1 ,ww); % 1 = a
plot(ww ,20*log10(abs(HH)));
```

Hint: You will have to use multiple filters. Once you have it working, combine those filters into one filter.

1.1 First task

From the spectrogram plot determine the frequencies associated with the sinusoidal disturbing tones.

1.2 Second task

Let's find the impulse response expression. If we want to eliminate a sinusoidal input signal, then we actually have to remove two signals, since

$$x[n] = \cos(\hat{\omega}_0 n) = \frac{1}{2}\left(e^{j\hat{\omega}_0}\right)^n + \frac{1}{2}\left(e^{-j\hat{\omega}_0}\right)^n .$$

Each complex exponential can be removed with a first-order FIR filter, and then the two filters would be cascaded to form the second-order nulling filter that removes the cosine. The first complex exponential signal is nulled by a filter with system function

$$H_1(z) = H_1(e^{j\hat{\omega}}) = 1 - z_1 z^{-1}$$

where $z_1 = e^{j\hat{\omega}_0}$ because $H_1(z_1) = 0$ at $z_1 = z_1$. Similarly $H_s(z) = 1 - z_2 z^{-1}$ removes the second complex exponential. By cascading the two filters, we obtain:

$$H(z) = H_1(z) \cdot H_2(z) = \cdots = 1 - 2\cos(\hat{\omega}_o) z^{-1} + z^{-2} . \quad (1)$$

The final results in Equation 1 follows from straightforward algebraic operations (it is suggested to check the final result).

Normalize Equation 1 to obtain a unit DC gain frequency response (DC corresponds to $\hat{\omega} = 0$ or equivalently to $z = e^{j\hat{\omega}} = 1$).

1.3 Third task

Design the filters with adequate coefficients. Plot the overall frequency response obtained by cascading the designed filters.

1.4 Fourth task

Filter the audio signal, plot the spectrogram of it and reproduce the original and filtered signal. Does the filtering operation produce satisfactory results ?