DSP - Laboratory 4

Daniele Orsuti
daniele.orsuti@phd.unipd.it

December 10, 2021

Abstract

Fourth laboratory experience [1] for the course Digital Signal Processing (DSP) a.y. 21/22. Topic: Audio signal processing using IIR filters.

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.

1 Vuvuzela noise reduction with notch filters

As in the previous laboratories, this document is intended to guide you step by step toward the solution. Please, read carefully the following tasks and feel free to look at Matlab documentation to gain familiarity with the functions used. During the laboratory we will sequentially show the solution of each task.

In the 2rd laboratory we had already worked with notch filters, in that case, however, the designed filter was a very simple FIR filter, which was not able to effectively remove the annoying tone from the corrupted input signal. In this laboratory, we could have used the same audio file to show the better performance that could be achieved using an IIR notch filter. You can verify that as homework, but to have fun with something different, we changed the audio file in this laboratory.
For more insight on the difference between FIR and IIR filters, refer to the optional task in Section 3.

After this short introduction we are ready to start.

1.1 Objective

A football match recording from 2010 World Cup has an annoying vuvuzela noise that prevents from clearly hearing the commentator's voice. Help the broadcaster to remove the annoying sound from the audio file *vuvuzela.wav*.

- Consider that the vuvuzela is a plastic horn, about 65 centimetres long, which produces a loud monotone note at about 235 Hz (first harmonic, the following harmonics are at about 465 Hz, 705 Hz, etc.).

2 Tasks

- Load the audio signal in Matlab (use the command *audioread* https://www.mathworks.com/help/matlab/ref/audioread.html). Please, note that *audioread* returns both the sampled audio data ($y[nT_s]$) and the sampling rate of the audio file (named $F_s = 1/T_s$ in what follows). F_s will be an important parameter in the following elaborations [2].

[1] All information and material presented here is intended to be used for educational or informational purposes only.
[2] For a review on sampling rate (i.e, F_s) and sampling period (i.e., T_s) you can give a quick look at https://en.wikipedia.org/wiki/Sampling_(signal_processing)

- Plot the loaded signal $y[nT_s]$ as a function of time. Plot the magnitude of the Fourier transform (in dB) of $y[nT_s]$ and verify the presence of the vuvuzela's harmonics (verify that tones are present at the frequencies given in Section 1.1). Proceed as follows:
  - For the time domain plot, define the appropriate temporal axis, i.e., the MATLAB vector

    ```
    t = T_s * [0:length(y)-1]
    ```

  - For the frequency domain plot use *freqz* command. Remember to specify F_s:

    ```
    [H, f_hat] = freqz(y,1,Fs)
    ```

    Important note: In the previous laboratories, when using freqz, we got familiar with the normalized frequencies (set by default in MATLAB). We would like now to convert from normalized frequencies to frequencies in Hz. This allows to simplify the next elaborations since the vuvuzela harmonics are specified in Hz. The conversion process is depicted in Figure 1. Following the figure, to plot the frequency response, the following normalization needs to be performed

    ```
    f_hat = f_hat / (pi) * (Fs /2)
    ```

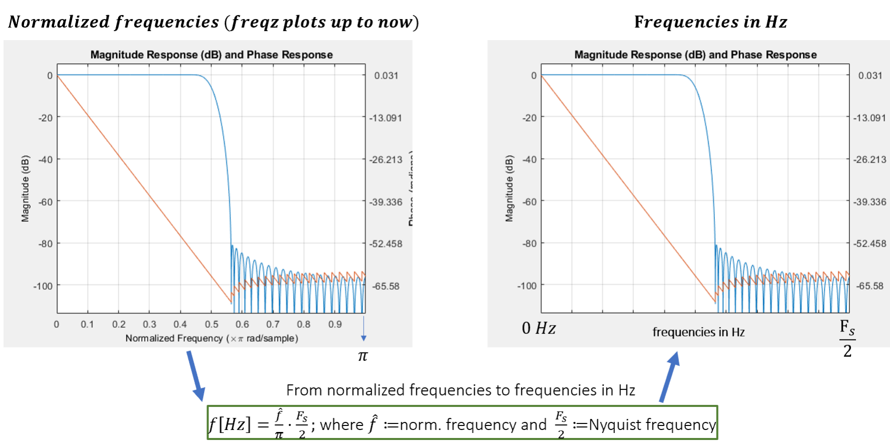

Figure 1: Illustrative example: from normalized frequencies to frequencies in Hz. In the figure, a low pass filter (LPF) is depicted for the only purpose of providing an example. Indeed **in what follows we will not be dealing with LPFs, but with notch filters.**

- Design a very simple IIR Notch filter to reduce the vuvuzela sound (the filter will remove the first harmonic). The generic second order IIR notch filter transfer function ($M = N = 2$) is given by:

$$H(z) = b_0 \frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})} = b_0 \frac{1 - 2\cos\theta_0 z^{-1} + z^{-2}}{1 - 2r\cos\theta_0 z^{-1} + r^2 z^{-2}}$$

  where $z_1 = e^{i\theta_0} = z_2^*$, $p_1 = re^{i\theta_0} = p_2^*$, $r \approx 1$ and finally $b_0 = \frac{1-2r\cos\theta_0+r^2}{2-2\cos\theta_0}$ to obtain as maximum gain one. In order to design a filter which notches the generic frequency $\bar{f}$ we need $\theta_0 = 2\pi\bar{f}\frac{1}{F_s}$ where $F_s = 1/T_s$ and T_s is the sampling period [3]

  Let $r \in \mathcal{R} := \{0.95\, 0.99\}$,

  - Plot the magnitude and phase of frequency response of the designed notch filter $\forall\, r \in \mathcal{R}$.

[3] Note that if we name $\theta_0 := \hat{f}$, then we find that to obtain θ_0 we are just converting the frequency $\bar{f}$ [Hz] to the normalized frequency.

- Plot the poles and zeros location of the designed filter $\forall\, r \in \mathcal{R}$.

- Filter the audio signal (use MATLAB's filter function) $\forall r \in \mathcal{R}$. Make sure that the filtering operation is performed using a notch filter with DC gain of 0 dB. Plot in the same figure the original and filtered spectrum of the audio signal.

  Play the filtered signal. Is the filter able to effectively reduce the vuvuzela sound?

- **[Optional:] Cascade of notch filters** If you are not satisfied with the result, you can keep on cascading notch filters to remove not only the fundamental harmonic but also the following ones. Below is given the list of the frequencies of the harmonics which can be visualized in the spectra of the audio signal.

```
f_1h=235; %[Hz] 1st harmonic freq.
f_2h=465;%[Hz] 2nd harmonic freq.
f_3h=694;%705; %[Hz] 3rd harmonic freq.
f_4h=932;%[Hz] 4th harmonic  freq.
f_5h=1160;%[Hz] 5th harmonic freq.
f_6h=1389;%[Hz] 6th harmonic freq.
f_7h=1632.5;%[Hz] 7th harmonic freq.
f_8h=1864;%[Hz] 8th harmonic freq.
f_9h=2115;%[Hz] 9th harmonic freq.
f_10h=2325;%[Hz] 10th harmonic freq.
f_11h=2568;%[Hz] 11th harmonic freq.
f_12h=2784;%[Hz] 12th harmonic freq.
```

  Plot the overall frequency response of the cascaded notch filters and the spectrum of the filtered signal.

3 [Optional] Group delay

The group delay [4] of a filter is a measure of the average time delay of the filter as a function of frequency. The group delay is defined as the negative first derivative of the filter's phase response with respect to the angular frequency.

$$\tau_g(\omega) := -\frac{d\phi(\omega)}{d\omega}$$

use the matlab function *grpdelay* (https://www.mathworks.com/help/signal/ref/grpdelay.html) to plot the group delay of the IIR filter designed in this laboratory, and the group delay of the FIR notch filter designed in the 2rd laboratory. Compare the results obtained.

[4] Further details at https://en.wikipedia.org/wiki/Group_delay_and_phase_delay