Digital Signal Processing

Homework 2

Federico Michelotto - 1218322

2019/2020

Contents

1 Exercise 1

Since the problem consists to reject the original signal at specific frequencies, I decided to use an IIR notch filter to accomplish this task. More precisely, for the first N harmonics related to the fundamental frequency of $f_0 = 235$ Hz, I applied an IIR second order notch filter to eliminate one harmonic at a time. That is, a cascade of N second order IIR notch filters with the following transfer function.

$$H(z) = \frac{1 - z_1 z^{-1}}{1 - p_1 z^{-1}} \frac{1 - z_2 z^{-1}}{1 - p_2^{-1}} \frac{(1 - p_1)(1 - p_2)}{(1 - z_1)(1 - z_2)}$$

The last part is a constant factor to normalize H(z) such that H(0) = 1. With poles and zeros set as follow:

$$z_1 = e^{i\theta}$$
$$z_2 = e^{-i\theta}$$
$$p_1 = re^{i\theta}$$
$$p_2 = re^{-i\theta}$$

where theta is the Normalized frequency of the k-th harmonics kf_0 of the k-th filter:

$$\theta = kf_0 2\pi T, k \in [1, N]:$$

The r constant has been set at 0.99 after some trials. The frequency response of one of these filters is shown in Fig.1.

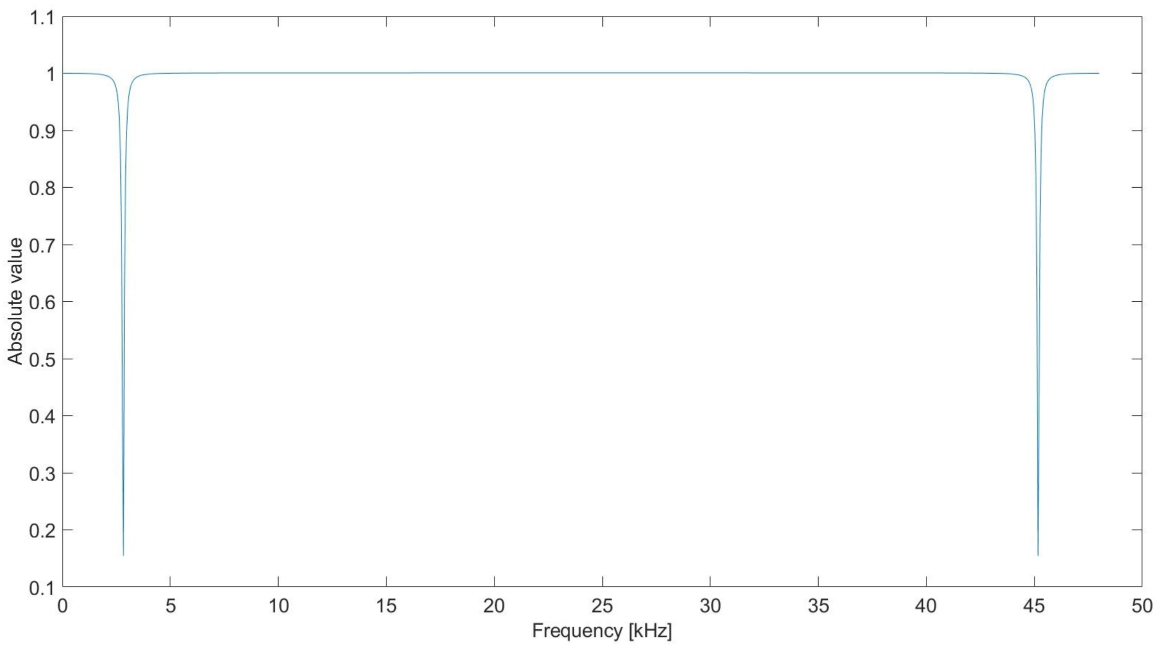

Figure 1: N-th IIR notch filter with cut-off frequency 2820 Hz, N = 12.

Using this technique the noise caused by the vuvuzela was sensibly reduced, but also attenuated a little bit the sound of the commentator.A visual understanding of what the filters have made on the original signal can be seen plotting the Power Spectral Density before and after, using the built-in function pwelch provided by Matlab.

Comparing Fig.2 and Fig.3 can be noted that the filtering had some effectiveness.

Increasing to much the number of the filters causes also an attenuation of the rest of the signal, not just the noise, for this reason I found as acceptable a value of N equal to 12 to remove the first 12 harmonics.

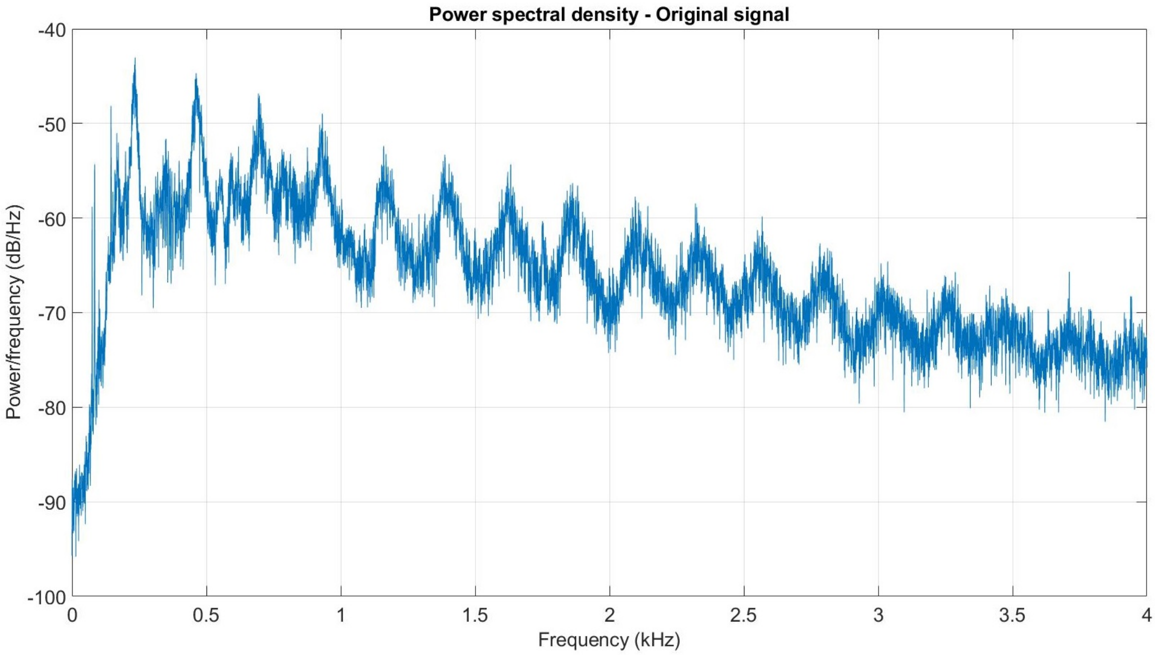

Figure 2: Power Spectral Density original signal, x-axis limited up to 4 kHz.

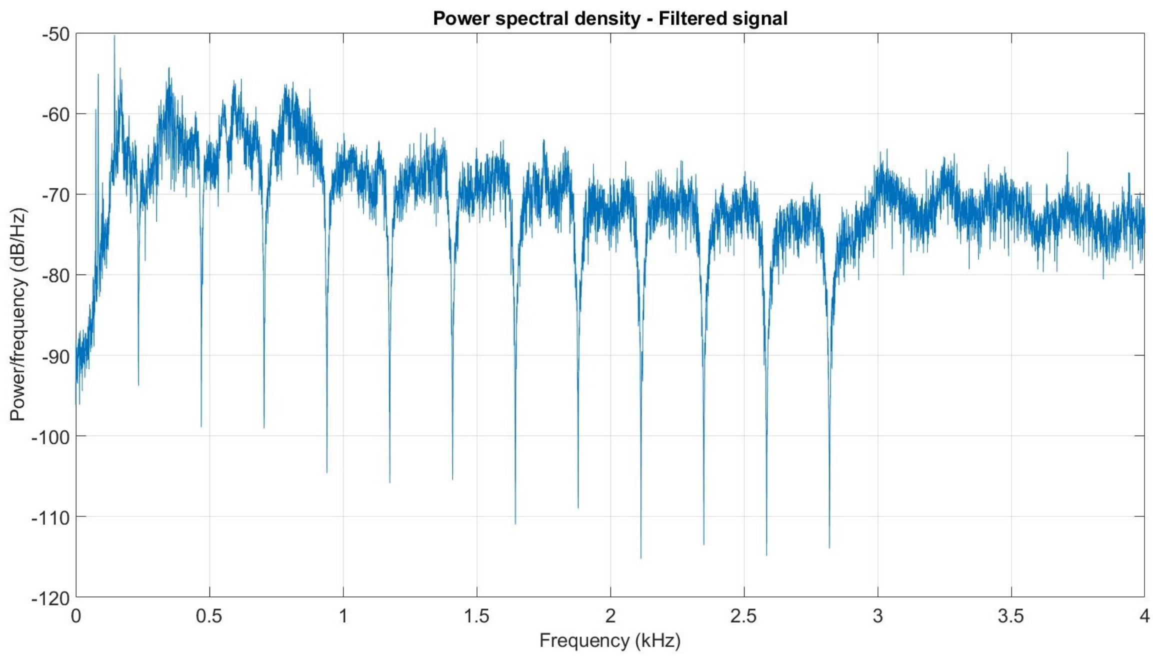

Figure 3: Power Spectral Density filtered signal, x-axis limited up to 4 kHz.

2 Exercise 2

Starting from the analog transfer function $H_a(s)$ we can use the transfer method to map it to the z-domain.

Doing this transformation we have to be careful, we have to apply a scaling of $\frac{2}{T}$ to the relationship that link the s-domain to the z-domain to successfully apply the transform method.

$$s = \frac{2}{T}\frac{1 - z^{-1}}{1 + z^{-1}}$$

The analog transfer function $H_a(s) = \frac{1+sT_2}{(1+sT_1)(1+sT3)}$ is mapped in the frequency domain through the following relationship:

$$H_{F,ref}(f) = H_a(i \cdot tg(\pi f T)(\frac{2}{T}))$$

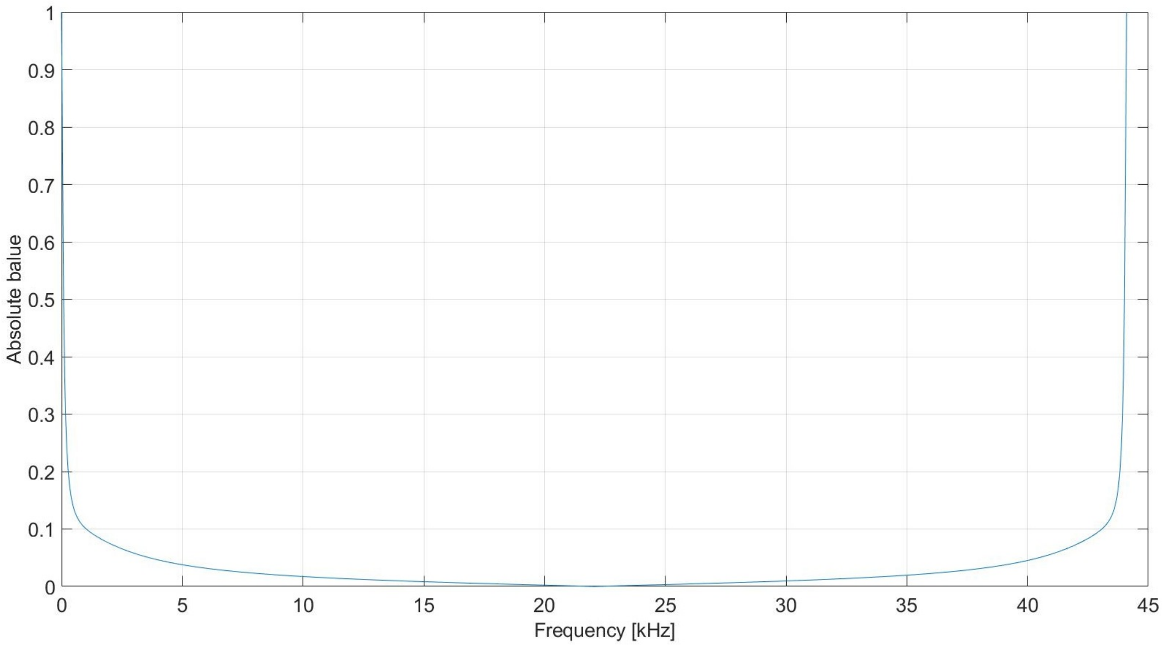

Figure 4: Reference frequency response $H_{F,ref}(f)$.

The target of this exercise is to use the transform method to build a second order IIR filter that has a frequency response that approximates $H_{F,ref}(f)$.

Applying the transform method we can find the poles and zeros of the transfer function of order two:

$$z_1 = \frac{2 - \frac{T}{T2}}{2 + \frac{T}{T2}}$$
$$z_2 = -1$$
$$p_1 = \frac{2 - \frac{T}{T1}}{2 + \frac{T}{T1}}$$
$$p_2 = \frac{2 - \frac{T}{T3}}{2 + \frac{T}{T3}}$$

where the transfer function is:

$$H(z) = K\frac{(1 - z_1 z^{-1})(1 - z_2 z^{-1})}{(1 - p_1 z^{-1})(1 - p_2 z^{-1})}$$

with K that is a constant factor. Since we want that $H_F(f) = 1$, we set

$$K = \frac{(1 - p1)(1 - p2)}{2(1 - z1)}$$

Know, knowing that $H_F(f) = H(z = e^{i2\pi fT})$ we can plot the frequency response of our filter, Fig.5.

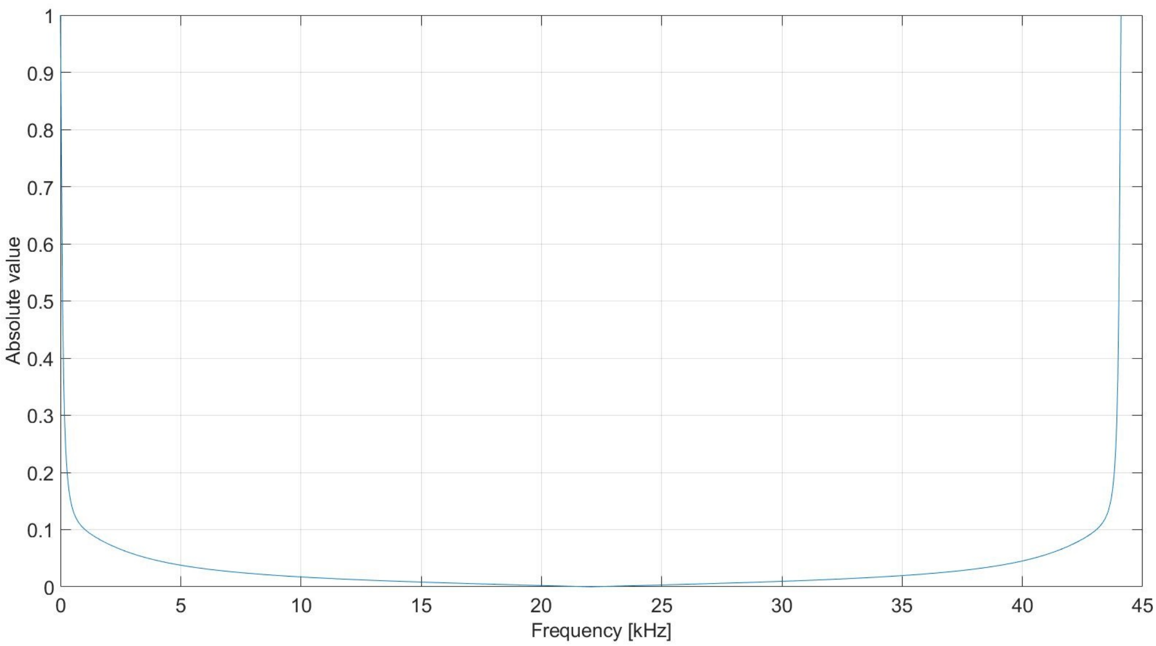

Figure 5: $H_F(f)$ frequency response.

The outcome approximates very well the reference frequency response, and the filtered audio signal has improved in sound quality with respect to the original one, that was what we expected.

2.1 Advanced

Using the direct method to minimize the squared error to the reference frequency response $H_{F,ref}(f)$, the outcome is shown below.

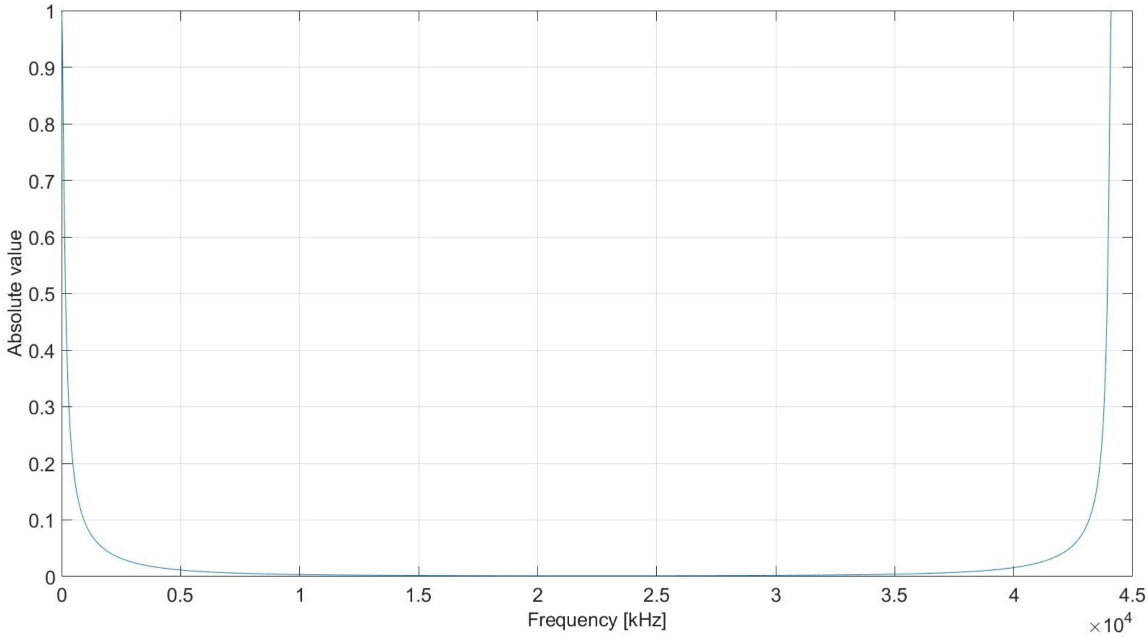

Figure 6: $H_{opt}(f)$ frequency response.

Despite the frequency response is pretty close to the reference one, the audio quality returned using this filter is not as good as the one got with the transform method.

3 Exercise 3

We have an audio signal with a sampling frequency $F_s = 44.1$ kHz and we want to correctly resample it to 48 kHz given the relationship below.

$$F_{s1} = \frac{L}{M} F_s = \frac{160}{147} F_s$$

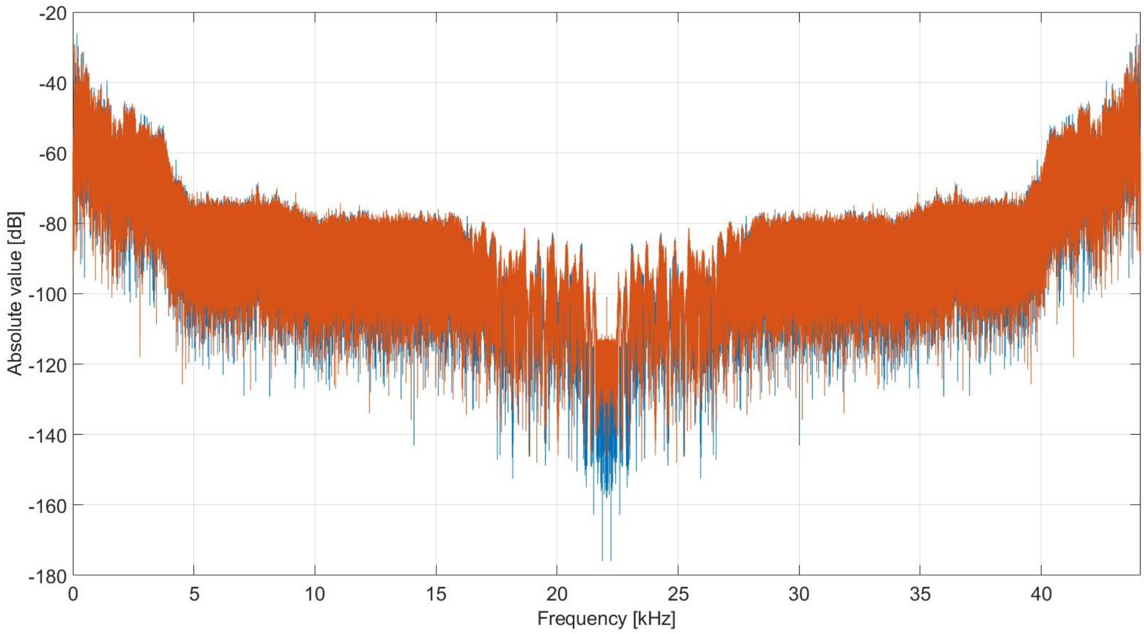

Figure 7: $X(f)$ frequency response of the original signal at 44.1 kHz.

To do so we first apply an interpolation system of factor L, and then we apply a decimation system of factor M.
The interpolation system consists in two phases:

1. Insert L-1 zeros between samples of the original signal in time.

2. Apply a low pass filter with cut-off frequency of $F_s/2$.

The first step will cause the sampling frequency to become equal to $F_{s2} = LF_s$ because of the fact that now the sampling period is $\frac{T}{L} = T_2$.
The decimation system, instead, consists in these two stages:

1. Apply a low pass filter with cut-off frequency of $F_{s1}/2 = \frac{F_{s2}}{2M}$.

2. Modify the sampling period from T_2 to T_1 such that $T_1 = MT_2$, this means to one keep value every M.

When the interpolation system is followed by the decimation system, there are two filters in cascade, and since they are both low pass filters, we can simply apply the one with the lowest cut-off frequency.

This mean that when $L > M$ (we are upsampling our signal) the low pass filter of the interpolation system has the lowest cut-off, and when $M > L$ (downsampling) the low pass filter with the lowest cut-off is the decimation one.

Knowing this, I used the *firpmord* method to design a filter with a cut-off of $F_s/2$ and with a sampling frequency of $F_{s2} = LF_s$.

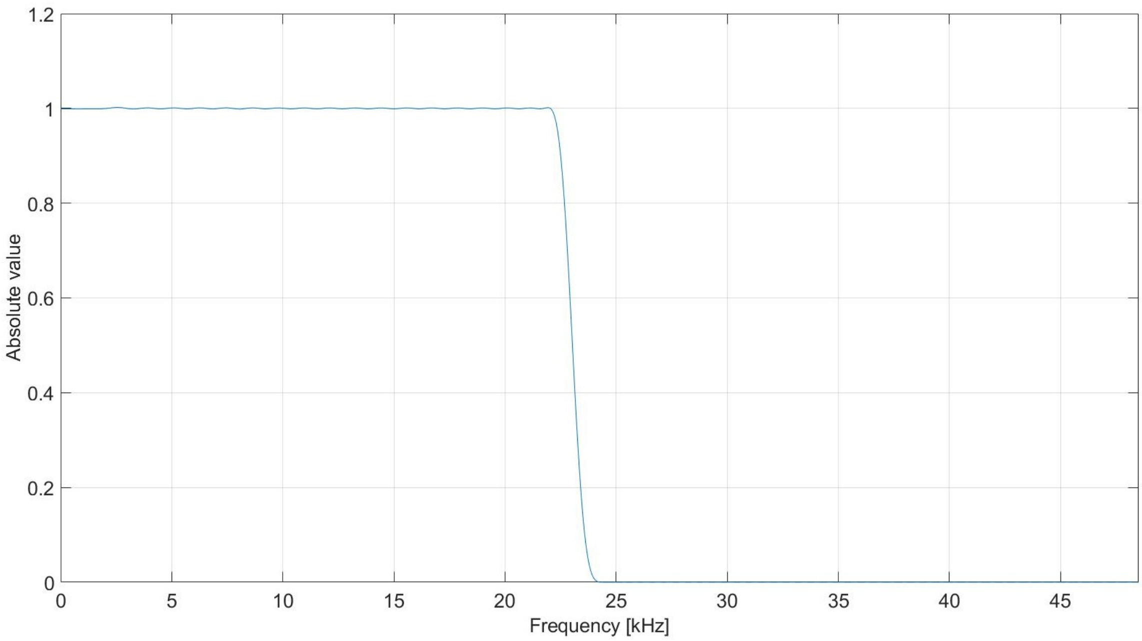

Figure 8: $H(f)$ frequency response, x-axis limited up to F_s.

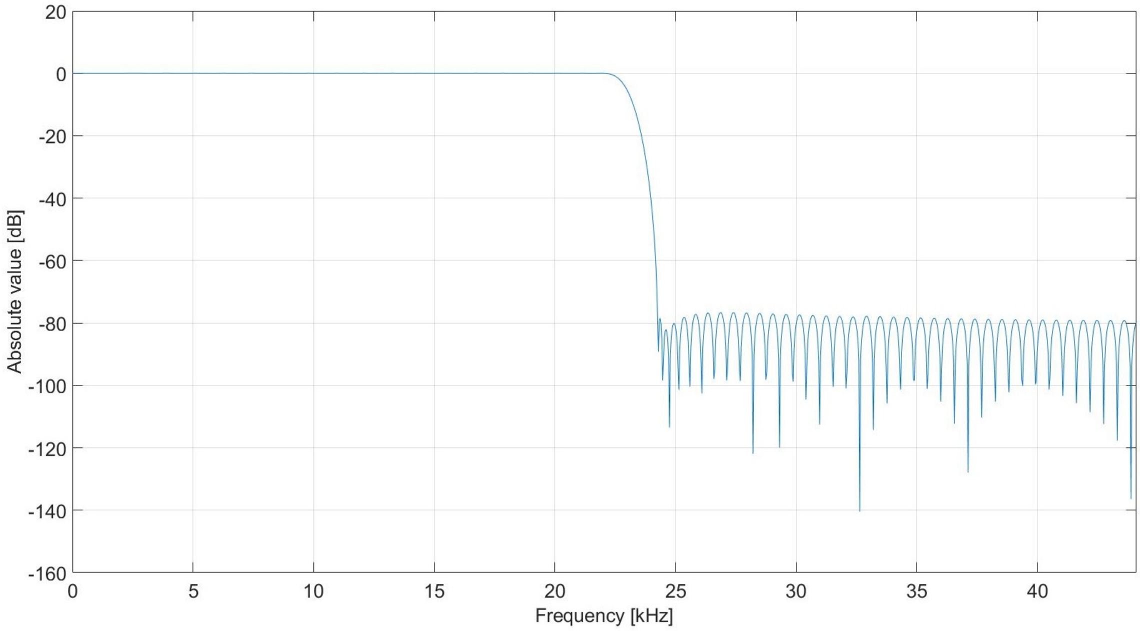

Figure 9: $H(f)$ dB frequency response, x-axis limited up to F_s.

To note, that since L isn't a small number F_{s2} is pretty big, $F_{s2} = 7056$ kHz. This (and the specifications of the filter) led to a filter with order 12473.

In Fig.10 is shown the frequency response of the output signal, and compared with the frequency response of the original signal of Fig.7 it seems that there aren't major discrepancies. The audio signal resulted (to my ears) as good as the original one.

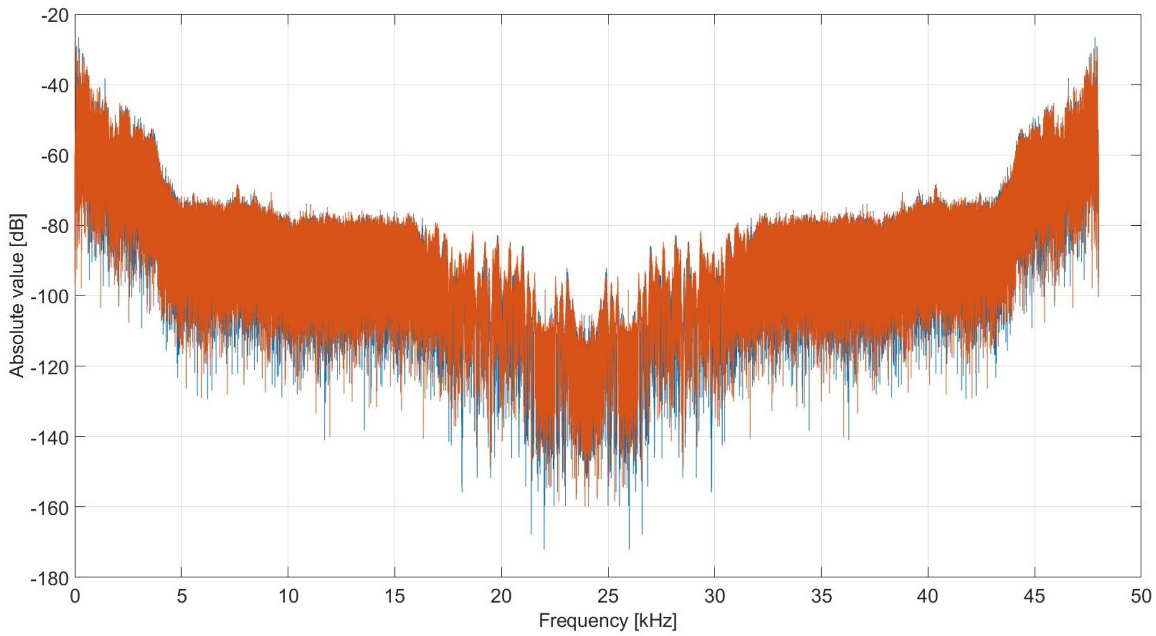

Figure 10: $Y(f)$ frequency response of the output signal at 48 kHZ.

3.1 Advanced

To implement the multistage approach it is sufficient to put in practice the considerations made above about the cut-off frequency of the low-pass filter.

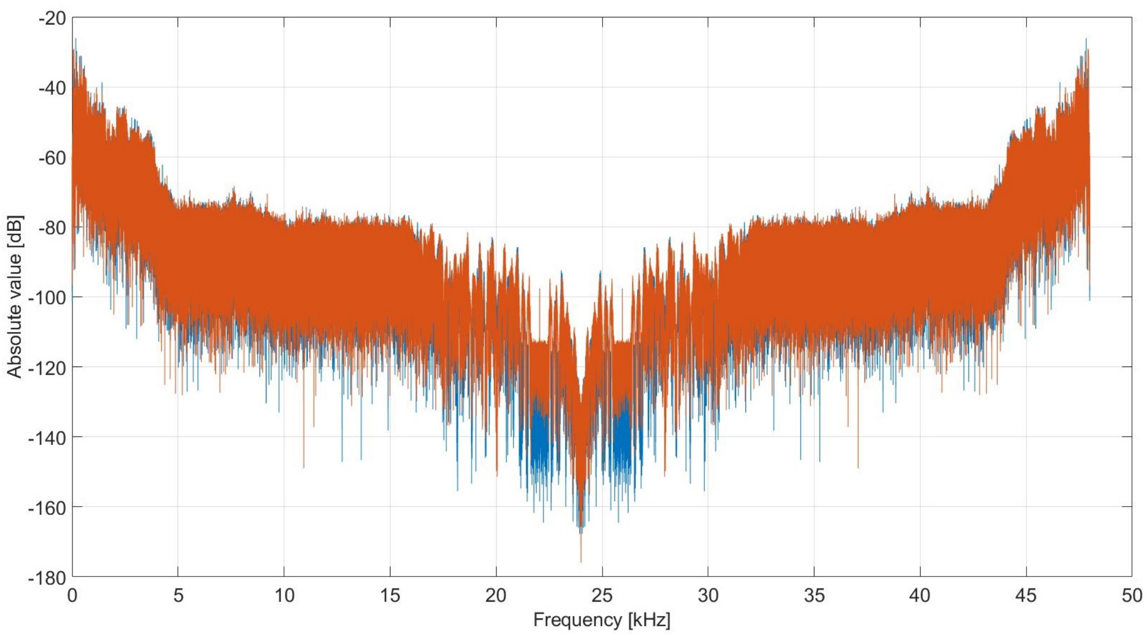

Figure 11: $Y_{adv}(f)$ frequency response of the output signal at 48 kHZ using a multistage approach.

The modular approach is much faster than than the classical one, since it can be avoided the use of filters of very high order. The audio quality at the output seems to be of the same quality of the one without the modular approach.