EQ2300 Digital Signal Processing Project

Núria Flores Espinosa
990906-3761

Yu Qin
980627-2879

August 28, 2023

Abstract

In this project, we designed a type I finite impulse response (FIR) low-pass filter (LPF) and a type I FIR high-pass filter (HPF) to implement a two-channel audio equalizer. Our LPF meets the requirement with the nominal normalized cutoff frequency $\nu c = 1/16$ and has a suppression of at least 40 dB for all normalized frequencies above $2\nu c = 1/8$, even taking the noise introduced by the fixed point design into account.

For practical purposes, we also explore the influence of quantization with the usage of fixed-point design for our filter. With our design, the final outcome shows that the output signal to quantization noise ratio (SQNR) is 45dB with a uniformly distributed input.

1 Introduction

The audio equalizer is a common component of our daily-used audio devices. In this project, we need to design a filter for the upcoming experience where we will deploy our equalizer onto the Arduino Due platform. The basic structure of the equalizer is shown in Figure 1.

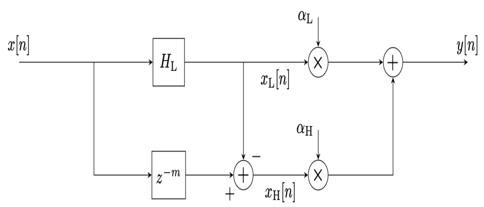

Figure 1: Basic structure

In this structure, we have the input signal $x[n]$, output signal $y[n]$, the LPF H_L, the low frequency part of signal $x_L[n]$ and the high frequency part of signal $x_H[n]$. We will design the filter H_L as a type I FIR filter using window method. In part 2, we will give a detailed explanation of how we could implement such a low-pass filter and use this to generate the corresponding high-pass filter we need.

2 Theory

2.1 Filter design using windows

In order to design the type I FIR LPF using the windows method, we need to clarify what a M-tap type I FIR LPF is. Type I means the taps number M of the filter is an odd number and is symmetric around the midpoint of itself in the time domain. FIR means the filter has a finite number of response in the time domain, which represents the value beyond the M taps should all be 0.

Ideally, we should have a perfect rectangular shape in the frequency domain. However, we can not implement this in the real world since it has an infinite impulse response. We can implement a finite M-tap LPF by creating the $h_1[n], n \in [\frac{-(M-1)}{2}, \frac{(M-1)}{2}]$ as in Equation (1) below.

$$\begin{aligned} h_1[n] &= IDTFT\{H_L\} \\ &= IDTFT\{rect_{2\nu_c}[\nu]\} \\ &= 2\nu_c sinc[2\nu_c n], n \in [\frac{-(M-1)}{2}, \frac{(M-1)}{2}]. \end{aligned} \tag{1}$$

Since sinc function is symmetric with 0, this implementation also make it a type I FIR filter at the same time. But we also need a way to modify the frequency according to what we want. Then we will need the window method [1], given as

$$h_L[n] = h_1[n] \cdot w[n], \tag{2}$$

where $w[n]$ can be any weight value you want to give to each point. As long as $w[n]$ is also real valued and symmetric to the midpoint, the filter will still be a type I FIR filter.

We can generate the corresponding HPF $h_H[n]$ easily in the time domain by

$$h_H[n] = \delta[n-m] - h_L[n], \tag{3}$$

where m is the timestamp of the midpoint of the n point sequence.

In the following sections, we explore the effect of different $w[n]$ and check for which window can satisfy our needs.

3 Numerical Results

3.1 Verification of the output

The aim of this section is to verify that the output $y[n]$ will always be a delayed version of the input $x[n]$ as long as $\alpha_L = \alpha_H = 1$ regardless of the filter H_L used.

From Figure 1, given $\alpha_L = \alpha_H = 1$, we can derive an expression for $y[n]$ as

$$y[n] = x_L[n] + x_H[n], \tag{4}$$

Taking into account that $x_H[n] = x[n-m] - x_L[n]$, we will find

$$y[n] = x_L[n] + x[n-m] - x_L[n] = x[n-m]. \tag{5}$$

Hence, we see that indeed the output of the circuit will be a delayed version of the input, delayed by m.

3.2 Type I FIR LPF Design

This section is focused on the design of the LPF that will filter out low frequencies of the input signal. As we are working with audio signals, the cutoff frequency will be set to 3 kHz, because most of the power in music is located below 3 kHz.

In consequence, the digital filter will have a normalized cutoff frequency of $\nu_c = \frac{3}{48} = \frac{1}{16}$. In addition, it will have a suppression above 40 dB for frequencies above $\nu = \frac{1}{8}$. All these conditions have to be met under the restriction of using a filter with less than $M = 55$ taps due to the platform capabilities.

Taking all this into account, we designed our filter using three different windows: Hanning, Hamming and Chebishev. In order to compare their performance and choose the one that complied best with the restrictions. The results are plotted in Figures 2, 3 and 4.

After analyzing the three results, we first discarded the Chevishev window as it has a higher 3 dB bandwidth, resulting on a slower transition from the pass-band to the stop-band. After comparing the two windows left, we decided on the Hamming window, as its sidelobes decreased faster and its power at $\nu = \frac{1}{8}$ was lower, around -62 dB. While using the Hanning window, the power at $\nu = \frac{1}{8}$ was around -54 dB.

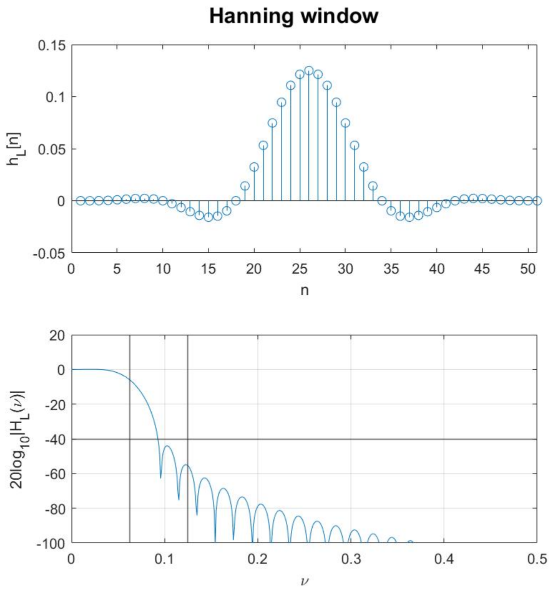

Figure 2: Filter designed using a Hanning window

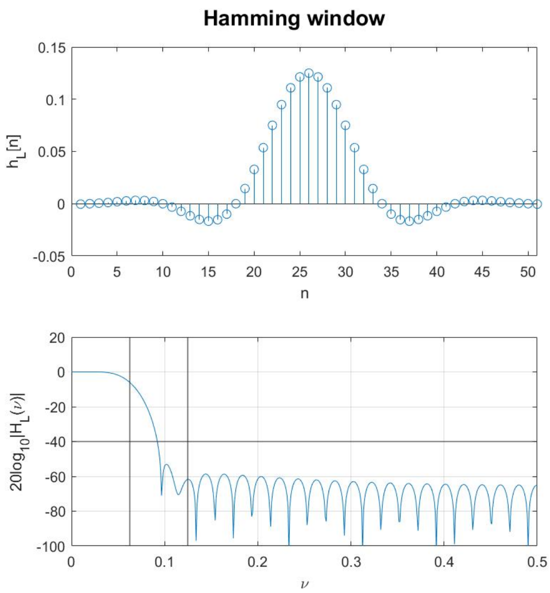

Figure 3: Filter designed using a Hamming window

3.3 Type I FIR HPF Design

From the block scheme on Figure 1, we see that the HPF is constructed by delaying the input signal m samples.

In order to achieve a Type I linear phase filter, we chose $m = 26$ which is equivalent to half the chosen value of taps $M = 51$. By choosing this value the resulting $h_H[n]$ is symmetric around m with even symmetry and has an even filter order,

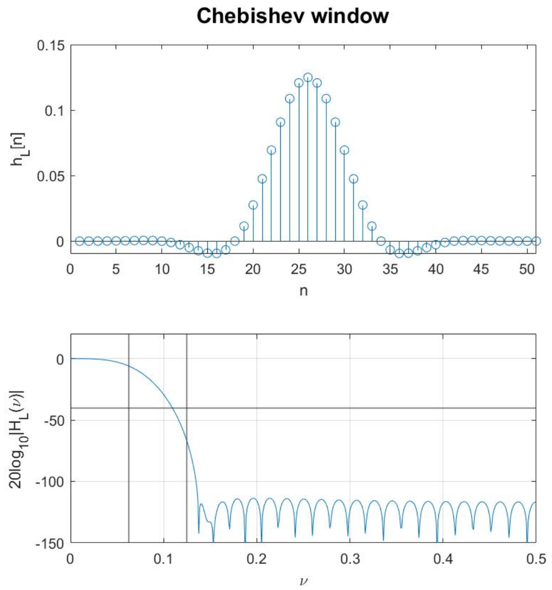

Figure 4: Filter designed using a Chebishev window

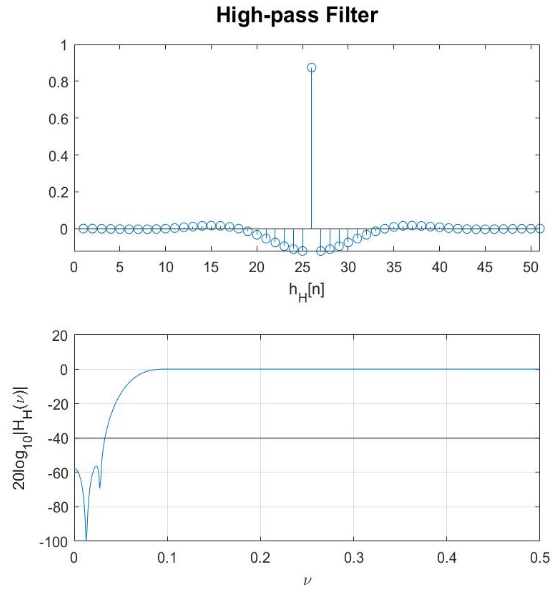

Figure 5: Impulse response and frequency response of the HPF.

which gives us a Type I filter that, by definition, has a linear phase. The impulse response and frequency response of our HPF are plotted in Figure 5.

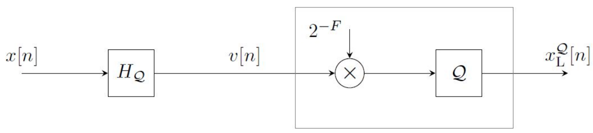

Figure 6: Bloch scheme for the quantized filter.

3.4 Quantization

The LPF designed in the previous sections is not implementable, in reality an equivalent fixed point implementation has to be applied. This fixed point implementation has impulse response $h_Q[n] = 2^{-F}[h_L[n] \cdot 2^F]$ as it is depicted in Figure 6. A suitable F has to be chosen so that the filter still complies with the original specifications, taking into account that the platform capability limits the maximum value of F to 16. With all this under consideration and after running some tests with different values of F, the value chosen is $F = 10$. The result is plotted if Figure 7.

As we can see, for normalized frequencies over $\nu = \frac{1}{8}$, the signal still gets attenuated more than 40 dB, so the quantized filter matches the specifications.

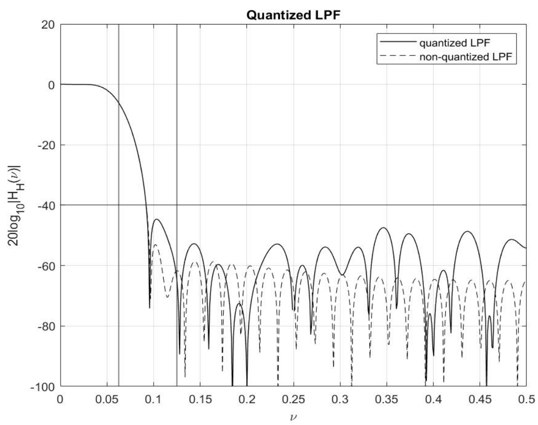

Figure 7: Frequency response of the quantized LPF.

Finally, in order to analyze the performance of our filter in terms of qualtization noise, the signal to quantization noise ratio will be computed following (6).

$$SQNR = \frac{E\{x_L^2[n]\}}{E\{(x_L[n] - x_L^Q[n])^2\}} \tag{6}$$

The input signal $x[n]$ has been created using the function *randi* in Matlab, which creates a random vector of uniformly distibuted integers in the specified range and with the specified vector dimensions. In our case, the vector contained values in the range $[-2^{11}, 2^{11}]$.

Following equation (6) and using $x[n]$ as the input vector, the obtained $SQNR = 45dB$.

4 Conclusions

In this project, we have compared the different effect of Hanning, Hamming and Chebishev windows to our LPF design. We have chosen the Hamming window and design the corresponding HPF, which both show a good performance regarding to the cut-off frequency and side-lobe suppression.

With our design, the final outcome shows that the output signal to quantization noise ratio (SQNR) is 45dB with a uniformly distributed input and have a suppression beyond 40dB with the existence of the quantization noise.

We are looking forward to put our design into real practice in the upcoming course lab.

References

[1] Hayes. M.H, *Statistical digital signal processing and modelling.* John Wiley & Sons, Inc., New York, USA, 1996