Project: Two-channel Audio Equalizer Design on the Adruino Due Platform

Zijian Fan
960723-0316

Yichen Yang
970916-9594

2018-11-28

Abstract

In this project, we'll discuss how to design an equalizer that separates the low-pass and high-pass component, and by modifying relative coefficients, we can mix those 2 components with different ratio and observe the result. Moreover, we'll also discuss about the coefficient quantization error and analyse how it affects the filter's performance by calculating its SQNR.

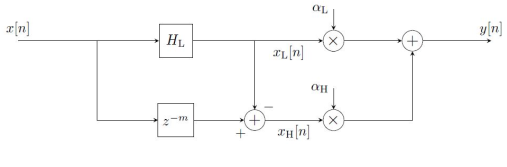
1 Introduction

Separating different frequency components of a signal is very useful and widespread implemented in many application. An realization is called equalizer which consists of low-pass and high-pass filter pair. Through reading materials, we know an effective equalizer model is shown in Figure 1, from which we know that an effective way to design a low-cost and high performance filter is crucial and significant, and the quantization error is also inevitable. Thus, we assume the registers storing filter coefficient have only limited F digits.

2 Theory

We can see that in Figure 1, $x_L[n]$ denotes the Low-Frequency component and $x_H[n]$ denotes the High-Frequency component. We multiply these two components by two different factors α_L and α_H, and the signal with different LF and HF component can be rebuilt.

Figure 1: A typical equalizer's structure



Firstly, it is clear that if both α_L and α_H in 1 are equal to 1, then the output y[n] is just the delayed version of input signal x[n]:
$x_H[n] = x[n-m] - x_L[n]$
$y[n] = x_H[n]\alpha_H + x_L[n]\alpha_L = x_H[n] + x_L[n]$
$y[n] = x[n-m]$

Then we need to design the proper filter $H_L[v]$ so that the LF component can be perfectly preserved while the HF component can be effectively suppressed. Assume the sample rate equals 48kHz, in the project, we try to design a filter with cutoff frequency at 3kHZ and at least 40dB suppression at the stop band(6kHz 24kHz).

A possible way to design this low-pass filter is to plot the ideal model in the frequency domain, then take the inverse Fourier transform and finally multiply it by a Hamming window. Correspondingly, we use fdatool in the matlab to design the filter with the cutoff frequency at 3kHz and sample rate of 48kHz and choosing Hamming window.
$h[n] = Hamming_N[n] * \int_{-0.5}^{0.5} H(v)exp(j2\pi nv)dv$

In order to maintain the high-pass filter $h_H[n]$ is also a linear phase filter, we should select a proper delay m. From Figure 1,we deduce the following:
If the input signal $x[n] = \delta[n]$, then the fourier transform of $x_H[n]$ follows:

$$X_H(v) = e^{-j2\pi vm} - H_L(v)X(v) = e^{-j2\pi mv} - H_L(v) \tag{1}$$

Considering $H_L(v)$ is a Low-pass linear phase Type 1 filter(order N is odd), we can split the transfer function into the following:

$$H_L(v) = A(v)e^{-j2\pi v\frac{N-1}{2}} \tag{2}$$

In the (2), A(v) is an amplitude function with real value and the exponential part is the linear phase. Observing (1) and (2), we can find that when $m = \frac{N-1}{2}$ holds, the spectrum $X_H(v)$ can be written as:

Figure 2: Impulse response of LP filter

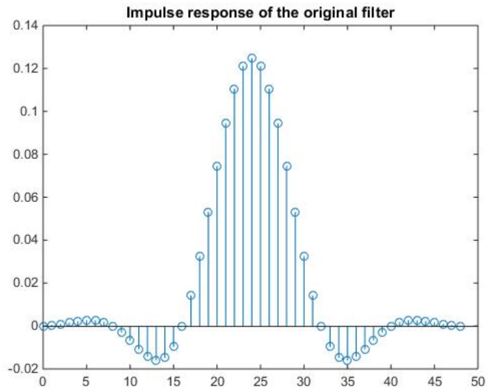
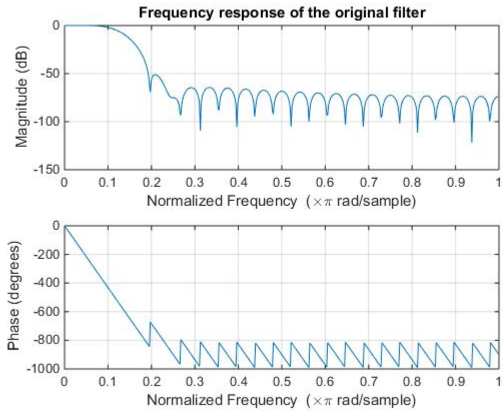


Figure 3: Frequency response of LP filter



$$X_H(v) = exp(-j2\pi v\frac{N-1}{2})(1 - A(v)) \qquad (3)$$

Through (3), it suffices to say that when $m = \frac{N-1}{2}$, $h_H[n]$ becomes a Type I linear phase filter.

In the fixed point implementation, we create a new filter $h_Q[n] = [h_L[n] * 2^{-F}]$, where $[\cdot]$ denotes rounding to the closest integer.Because Arduino uses a 12-bit ADC, we choose F as 12.

Even though the filter H_Q can be implemented exactly, the multiplication with 2^{-F} followed by integer quantization introduces quantization noise. And the noise can be calculated as:

$$\mathrm{SQNR} = \frac{\mathrm{E}\left\{x_{\mathrm{L}}^{2}(n)\right\}}{\mathrm{E}\left\{\left(x_{\mathrm{L}}(n) - x_{\mathrm{L}}^{\mathrm{Q}}(n)\right)^{2}\right\}} \qquad (4)$$

$$\mathrm{SQNR} = \frac{\mathrm{E}\left\{x^{2}(n)\right\}\sum_{i=0}^{48} h_{\mathrm{L}}^{2}(i)}{\mathrm{E}\left\{x^{2}(n)\right\}\sum_{l=0}^{48}\left(h_{\mathrm{L}}(l) - h_{\mathrm{L}}^{\mathrm{Q}}(l)\right)^{2}} \qquad (5)$$

The derivation from (4) to (5) is shown in the appendix.

3 Numerical Results

Impulse and frequency response of LP filter are shown in 3 and 3.

We use 48-order type I Linear phase FIR filter, equipped with Hamming window, and finally we design the filter whose impulse response and frequency response are shown in 3, through which we can see that the suppression at 6kHz and above is approximately 50dB, which meet our demands.

Impulse and frequency response of HP filter are shown in 3 and 3.

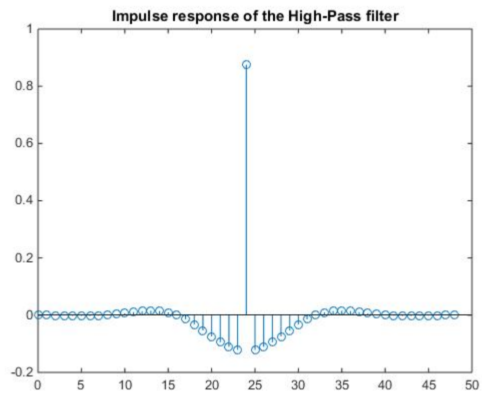
Setting $m = \frac{N-1}{2}$, we can plot the impulse response and frequency response of the High-pass filter, which is shown in 3.

However,in practice we need integer as the filter's coefficient, so quantization error is inevitable. Naturally, the higher quantization bit corresponds to smaller error, but increasing quantization bit also means more cost. Thus, in order not to risk internal overflow in the later implementation on Arduino Due, we keep F below 16. We find that when F equals to 9, the quantization filter will meet the minimum requirement. So we choose F a little bit lager than 9. Frequency response of Fixed point implementation filter(F=12) is shown in 3.

From 3, we see when $w = 2\pi \times 1/8$, the suppression is larger than 50dB. And when$w = 2\pi \times 1/16$, the suppression is 6dB, which means the cutoff frequency $\nu_c = 1/16$.

To estimate the quantization error and compare

Figure 4: Impulse response of HP filter



it with theoretical result, we generate a random sequence with 10000 numbers uniformly distributed in $\left[-2^{11}, 2^{11}\right]$, and filter it with the LP filter and quantization filter separately. Then we calculate the power of the sequence and the mean square error in order to get the SQNR $\approx$ 56.92dB. According to (5), SQNR $\approx$ 56.99dB.

4 Conclusion

In this project, we apply the knowledge from DSP to the equalizer design. First, we design the LP filter and the HP filter. In order to implement the filter with integer arithmetics, we create a quantized LP filter. Thus, the error caused by quantization is unavoidable, according to our trials and theoretical calculation, the SNR is very high.

Appendix

Derivation from (4) to (5):

$$
\begin{aligned}
& \mathrm{E}\left\{x_{\mathrm{L}}^{2}\left(\mathrm{n}\right)\right\} \\
&= \mathrm{E}\left\{\left(x\left(n\right) * h_{\mathrm{L}}\left(n\right)\right)^{2}\right\} \\
&= \mathrm{E}\left\{\left(\sum_{l=0}^{48} h_{\mathrm{L}}\left(l\right) x\left(n-l\right)\right)\left(\sum_{m=0}^{48} h_{\mathrm{L}}\left(m\right) x\left(n-m\right)\right)\right\} \\
&= \sum_{l=0}^{48} \sum_{m=0}^{48} h_{\mathrm{L}}\left(l\right) h_{\mathrm{L}}\left(m\right) r_{x}\left(l-m\right) \\
&= \sum_{l=0}^{48} \sum_{m=0}^{48} h_{\mathrm{L}}\left(l\right) h_{\mathrm{L}}\left(m\right) \sigma^{2} \delta\left(l-m\right) \\
&= \sum_{l=0}^{48} h_{\mathrm{L}}^{2}\left(l\right) \sigma^{2} \\
& \mathrm{E}\left\{\left(x_{\mathrm{L}}\left(n\right)-x_{\mathrm{L}}^{\mathrm{Q}}\left(n\right)\right)^{2}\right\} \\
&= \mathrm{E}\left\{\left(x\left(n\right) * \left(h_{\mathrm{L}}\left(n\right)-h_{\mathrm{L}}^{\mathrm{Q}}\left(n\right)\right)\right)^{2}\right\} \\
&= \sum_{l=0}^{48}\left(h_{\mathrm{L}}\left(l\right)-h_{\mathrm{L}}^{Q}\left(l\right)\right) \sigma^{2} \\
& \mathrm{SQNR} = \frac{\mathrm{E}\left\{x^{2}(n)\right\} \sum_{i=0}^{50} h_{\mathrm{L}}^{2}(i)}{\mathrm{E}\left\{x^{2}(n)\right\} \sum_{l=0}^{50}\left(h_{\mathrm{L}}(l)-h_{\mathrm{L}}^{\mathrm{Q}}(l)\right)^{2}} \\
&= \frac{\sum_{i=0}^{50} h_{\mathrm{L}}^{2}(i)}{\sum_{l=0}^{50}\left(h_{\mathrm{L}}(l)-h_{\mathrm{L}}^{\mathrm{Q}}(l)\right)^{2}}
\end{aligned}
$$

Figure 5: Frequency response of HP filter

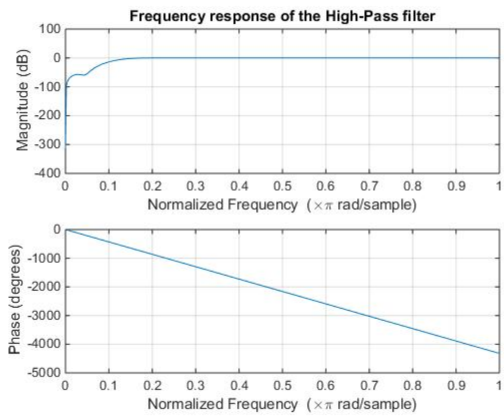
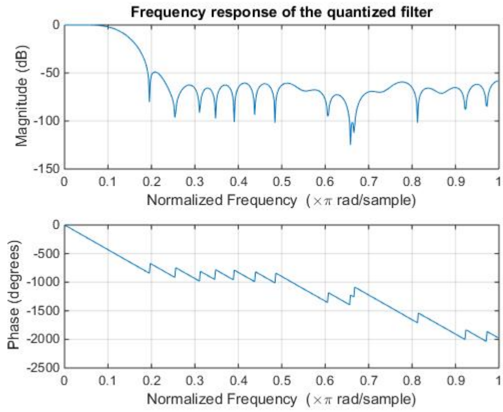


Figure 6: Frequency response of quantized LP filter(F=12)



References

[1] Tobias Oetiker et al. *The Not So Short Introduction to LaTeX 2ε* `http://tug.ctan.org/info/lshort/english/lshort.pdf`.