Project Assignment

Hareekeshav Sethumadhavan Srinivasan

October 22, 2024

Abstract

This report outlines the design and implementation of a fixed-point filter for equalization, focusing on the transformation of a low-pass filter to an equivalent fixed-point representation. The primary goal was to address the challenge of adapting a signal processing filter to operate efficiently in integer arithmetic while maintaining comparable performance.

1 Introduction

One fundamental technique in Digital Signal Processing is equalization, which involves modifying the frequency response of a system to achieve desired characteristics in signal processing.

The primary focus is to address the challenge of implementing an effective equalizer using fixed-point arithmetic. The problem revolves around transforming a low-pass filter designed in floating-point representation into an equivalent fixed-point filter to reduce computational complexity.

In the end we compute the SQNR of the filter that was designed assuming that the input lies in some range due to Arduino Due analog to digital converter.

2 Theory

2.1 Verification of y[n] = x[n-m]

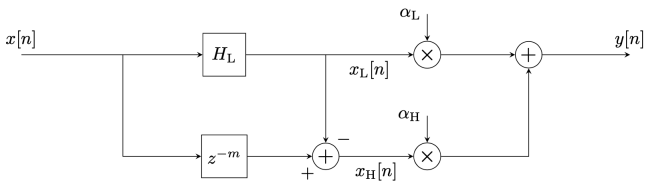

Figure 1: Schematic Diagram of the Equalizer

From the diagram, the signal at time index n, y[n], is expressed as:

$$y[n] = (\alpha_L \times x_L[n]) + (\alpha_H \times x_H[n])$$

When α_L and α_H both equal 1, the equation simplifies to:

$$y[n] = x_L[n] + x_H[n]$$

Substituting $x_L[n] = x[n] * h_L$ and $x_H[n] = x[n-m] - x_L[n]$ into the above equations yields:

$$y[n] = (x[n] * h_L) + (x[n-m] - x_L[n])$$
$$y[n] = (x[n] * h_L) + (x[n-m] - (x[n] * h_L))$$
$$y[n] = x[n-m]$$

2.2 Low-Pass Filter Design

The design of a low-pass filter involves specifying certain parameters to attain the desired frequency response. Commonly used method is the window-based design. Window Method involves designing the filter by applying a window function to ideal filter coefficients, optimizing the window to meet desired specifications. We define some cut-off frequency for our design, typically where we want the signal to start attenuating. For our case, we want nominal normalized cutoff frequency of vc = 1/16 = 3/48, corresponding to an equivalent cutoff frequency of 3 kHz for the analog signal before sampling. The filter need to have a suppression of at least 40 dB for all normalized frequencies above vc = 1/8 = 6/48.

We first compute the Ideal impulse response equation, which is given by

$$\mathrm{h}_I[n] = \begin{cases} \frac{\sin(2\pi\nu_c(n-M/2))}{\pi(n-M/2)} & \text{if } n \neq M/2 \\ 2\nu_c & \text{if } n = M/2 \end{cases}$$

For the window, we use Hamming Window and generate a Hamming window of length M+1 . The equation h[n] represents the windowed filter coefficients

$\mathrm{h}[\mathrm{n}] = \mathrm{h}_I[n] \times w[n]$

Delaying the ideal impulse response by M/2 samples ensures that the center of the impulse (or the peak of the response) aligns with the middle sample of the filter kernel, considering that the filter length is M+1 samples. This ensures that the filter is linear as per the requirements. We choose the filter order M to be 50. It provides better stopband attenuation and sharper cutoff compared to lower orders, also it offers improved characteristics without excessively increasing computational load. Hamming window offers control over the transition band width and sidelobe levels in the frequency response.

In MATLAB,we first generate an index array from 0 to M. We then use the freqz function to compute the frequency response of the designed filter.

The code is generated and the impulse response and magnitude frequency response of the designed filter is visualized as in figure 2 and figure 3

2.3 High pass filter design

The aim of designing this high-pass filter is to complement the low-pass filter by extracting frequency components above a certain frequency while suppressing lower frequencies. The high-pass filter is created by subtracting the low-pass filter's response from an impulse at position 'm'. This approach uses the principle of filtering through delay and subtraction to create the desired high-pass characteristics.

The MATLAB code uses the pre-designed low pass filter (h_l). Using the selected delay value (m), the code positions an impulse to create a high-pass filter from the low-pass one. Subtracting h_l from this impulse at position 'm', it generates the high-pass filter's impulse response (h_H). The 'freqz' function computes the high-pass filter's frequency response (H_H). This approach transforms the low-pass filter by delay and subtraction, creating a high-pass filter attuned to specific frequency components.

The delay value 'm' determines the point at which the high-pass filter's impulse response will start. By choosing 'm', the code effectively positions the point where the high-pass response initiates. For a Type I linear phase FIR filter, the symmetry of the filter coefficients plays a crucial role.

The delay 'm' is chosen to maintain symmetry in the high-pass filter. By ensuring the correct delay and appropriate centering, the high-pass filter will exhibit Type I linear phase characteristics similar to the low-pass filter.

Choosing 'm' as half of the filter length ensures that the center of the high-pass filter aligns correctly with the center of the low-pass filter. This alignment

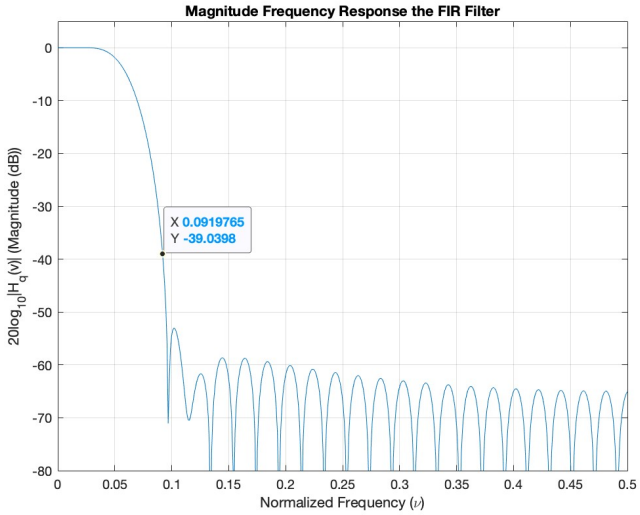

Figure 2: Magnitude of Frequency response

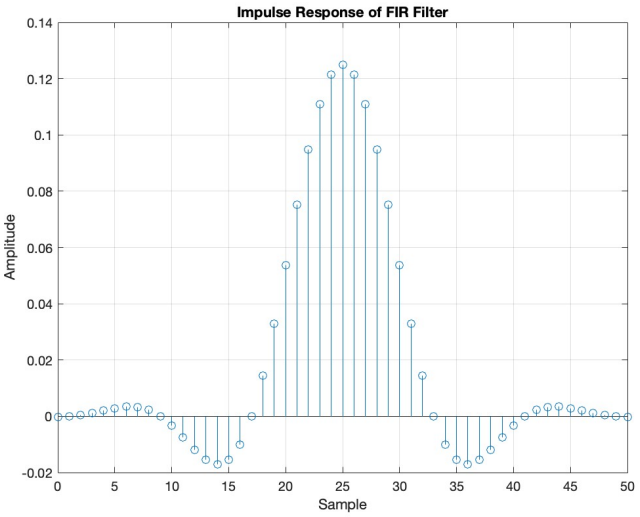

Figure 3: Impulse response

helps in maintaining the symmetry and linear phase properties, essential characteristics of a Type I linear phase filter. So therefore we chose m=25 and plot the magnitude of frequency (figure 5) and the impulse response (figure 4).

2.4 Fixed point implementation

The purpose of the fixed-point implementation is to represent the filter coefficients in a fixed-point format suitable for integer arithmetic. This allows performing filter operations with reduced computational complexity, beneficial for platforms with limited resources or to reduce hardware complexity in signal processing applications. F represents the number of fractional bits used in the fixed-point representation. A higher F value generally results in a smoother magnitude response curve due to increased precision in the fixed-point representation. The choice of F=10 for these plots provides a balance between achieving a reasonable level of precision and computational efficiency.

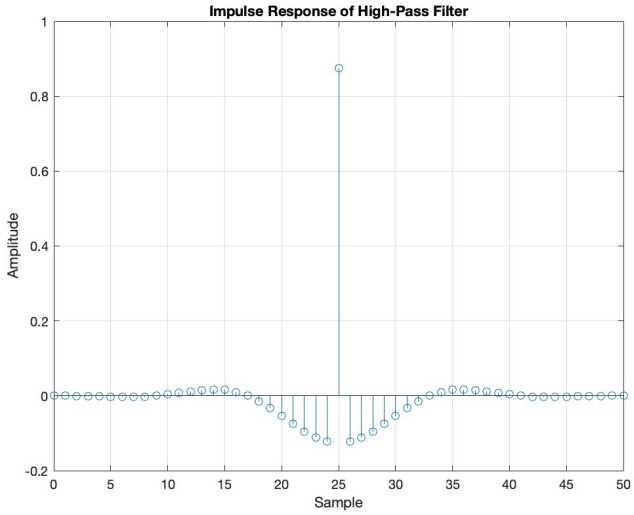

Figure 4: Impulse Response

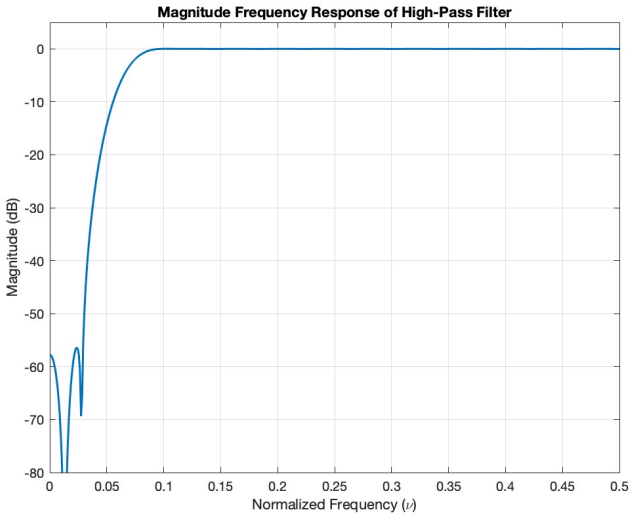

Figure 5: Magnitude Frequency Response

2.5 Computation of SQNR

SQNR measures the ratio between the signal power and the quantization noise power.

SQNR = E{ $x_L^2[n]$}/$E\{(x_L[n] - x_{q_L}[n])^2\}$

The average signal power is calculated as

$$E[x^2] = \text{mean}\left(\text{conv}(x, h_l)\right)^2$$

The average quantization noise power is calculated as

$$E[x_q^2] = \text{mean}\left(\text{conv}(x, h_Q)\right)^2$$

Finally, the Signal-to-Quantization-Noise Ratio (SQNR) is given by

$$\text{SQNR} = 10 \cdot \log_{10}\left(\frac{E[x^2]}{E[x^2] - E[x_q^2]}\right)$$

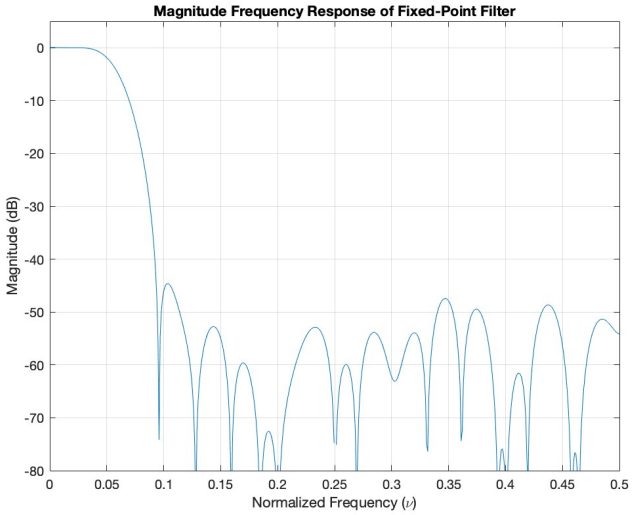

Figure 6: Magnitude Frequency response of fixed point implemented filter

Where h_l represents the low pass filter coefficients and h_Q represents the quantized coefficients. x is generated by a uniform rand function and the pdf of that is taken to compute the signal power and quantization noise power. This ratio represents the strength of the signal compared to the quantization noise.For F=10, we get a SQNR of 46.9 dB.

3 Conclusions

In conclusion, this report has presented the design and implementation of low-pass and high-pass filters for an equalizer, meeting the specified cutoff frequencies and suppression requirements.Moreover, the investigation into fixed-point implementation provided insights into real-world applicability by quantizing filter coefficients for integer arithmetic. The Signal-to-Quantization-Noise Ratio (SQNR) calculation offered an empirical measure of the filter's performance under quantization

Moving forward, optimizing filter design approaches can offer a balance between transition width and passband characteristics.

Appendix

3.1 Choice of filter order

Higher filter orders allow for sharper transitions between passband and stopband. M=50 balances performance in meeting specifications and Offers improved characteristics

3.2 Variations in value of F

adjusting the value of F in fixed-point implementations involves a trade-off between precision and frequency response accuracy. It also changes the value of SQNR as shown in figure 8.

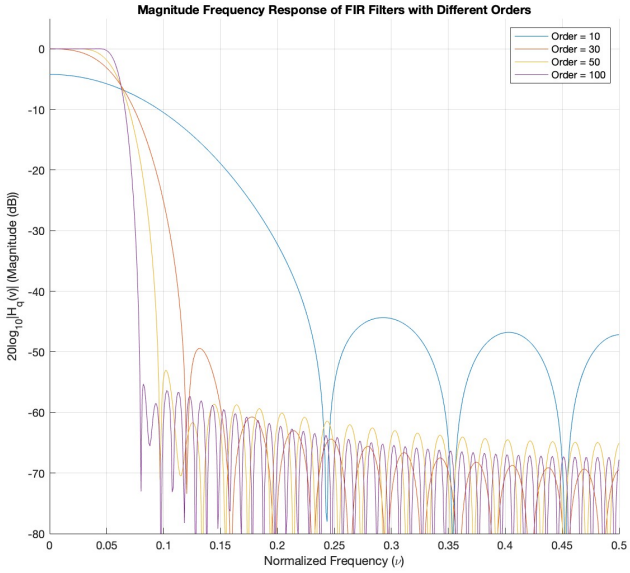

Figure 7: Magnitude frequency plots for different values of filter order

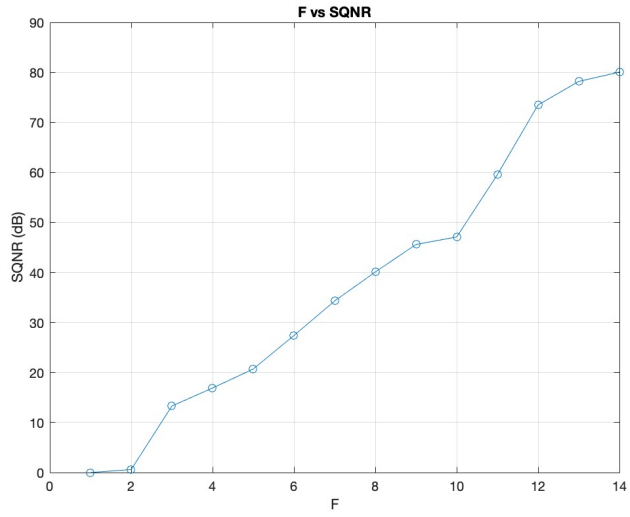

Figure 8: SQNR vs F

4 References

1. EQ2300 Digital signal processing lecture 4 : Filter design using windows
2. EQ2300 Digital signal processing lecture 5 : Quantization