
# University of Pennsylvania

**School of Engineering and Applied Science**

**Department of Electrical and Systems Engineering**

**ESE 5310 — Spring 2025**

**Digital Signal Processing**

---

**Project 1**

**Filter Design and Filter Banks**

---

**Hesham Nabil Hosny Maher  58309713**

**Submission Date: Friday, Mar 28, 2025**

# Contents

# 1. Part A: FIR Filter Design

## (a) Window-Based FIR Filter Design

### i. Truncated Impulse Response Analysis (N=21 vs N=101)

The truncated FIR filters were designed with a cutoff frequency of $\omega_c = 2.0$ rad/s. Figures 1.1 and 1.2 compare the time-domain and frequency-domain characteristics for $N = 21$ and $N = 101$ filters, demonstrating the trade-offs between filter length and performance:

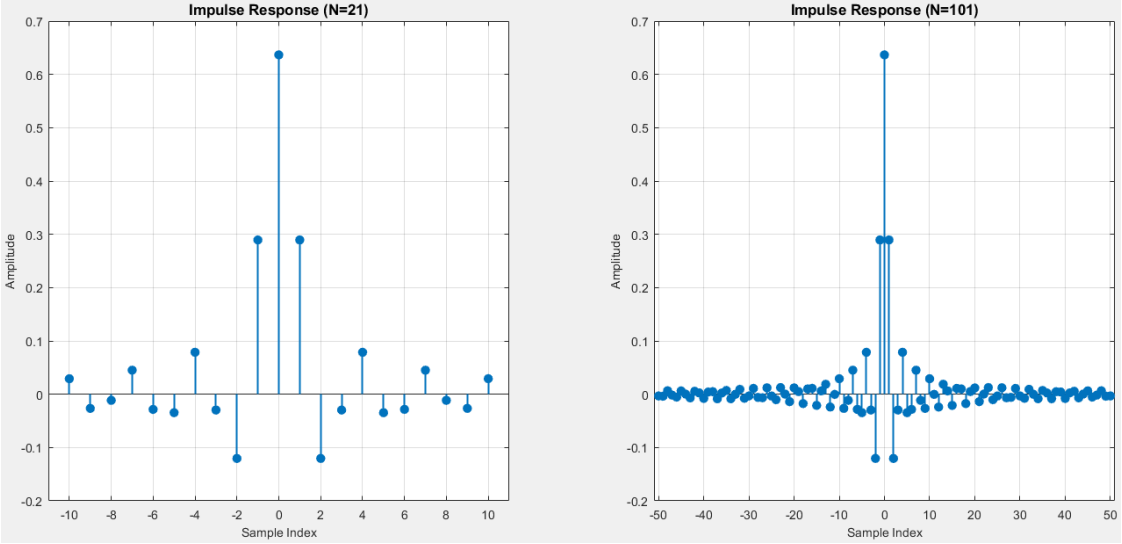



Figure 1.1: Impulse response comparison (N=21 vs N=101). The longer filter exhibits more oscillations, better approximating the ideal sinc function.

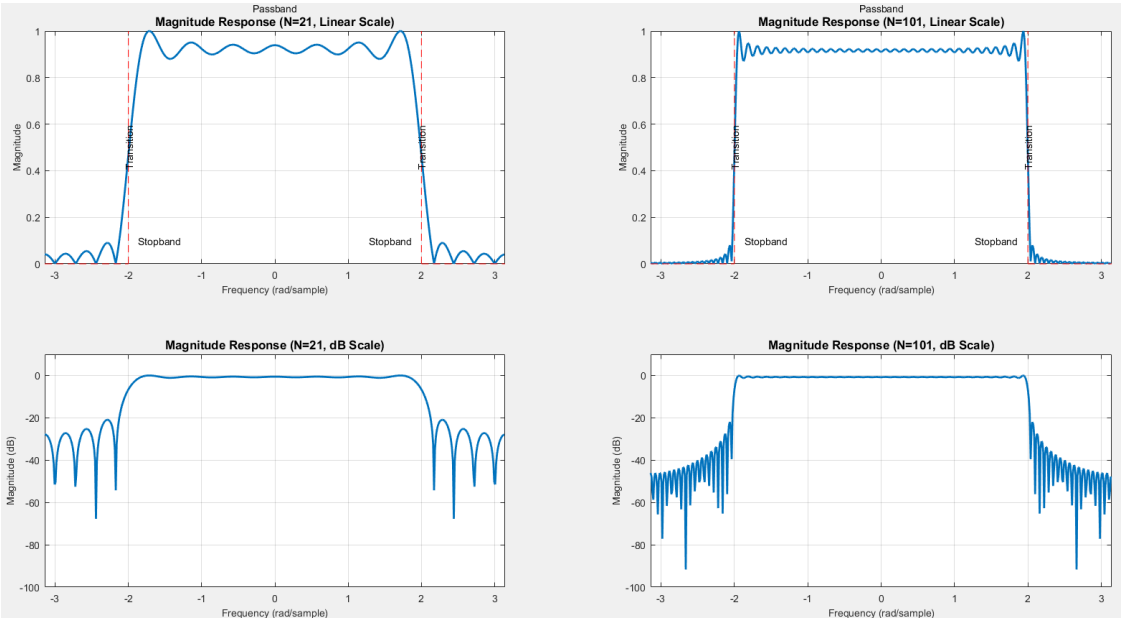



Figure 1.2: Magnitude response comparison

The filter size (or order) directly affects the passband and stopband ripple. A larger filter size generally reduces the ripple in both the passband and stopband, resulting in a smoother response. This happens because higher-order filters have more poles and zeros, which allow for a more precise shaping of the frequency response, improving attenuation in the stopband and reducing variations in the passband. Conversely, smaller filters tend to have more noticeable ripples due to fewer poles and zeros, leading to less control over the frequency response.

**Observations:**

- **Transition Band Characteristics**
  The longer filter demonstrates significantly sharper frequency selectivity, with a substantially narrower transition band compared to the shorter filter. This improved performance comes at the cost of proportionally increased computational requirements.

- **Ripple Performance**
  Both passband ripple and stopband attenuation show marked improvement in the longer filter. The reduced passband variation and enhanced stopband rejection demonstrate better approximation to ideal filter behavior.

- **Time-Frequency Trade-off**
  The longer filter's impulse response exhibits more pronounced oscillations (Fig. 1.1), illustrating the fundamental compromise between frequency domain performance and time-domain ringing effects.

The theoretical ideal filter remains unrealizable in practice:

$$h_{\text{ideal}}[n] = \frac{\omega_c}{\pi}\text{sinc}(\omega_c n), \quad n \in (-\infty, \infty) \tag{1.1}$$

## ii. Windowed Filter Comparison: Blackman vs. Hamming

## Window Function Analysis and Effects

To analyze the effects of windowing in FIR filter design, we implemented two filters using the Hamming and Blackman windows, both with a filter length of $N = 21$.

Figure 1.3 illustrates the impulse response comparison between the Hamming and Blackman filters. The differences in amplitude distribution highlight how each window tapers the impulse response differently.

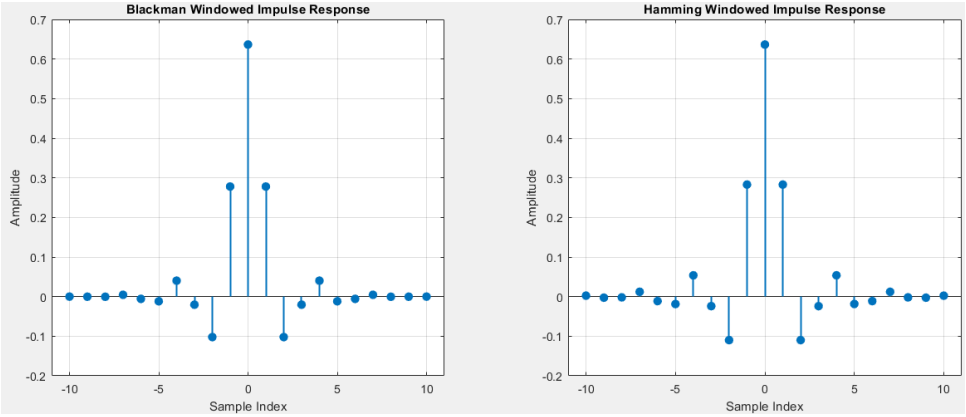



Figure 1.3: Impulse response comparison between Blackman and Hamming filters.

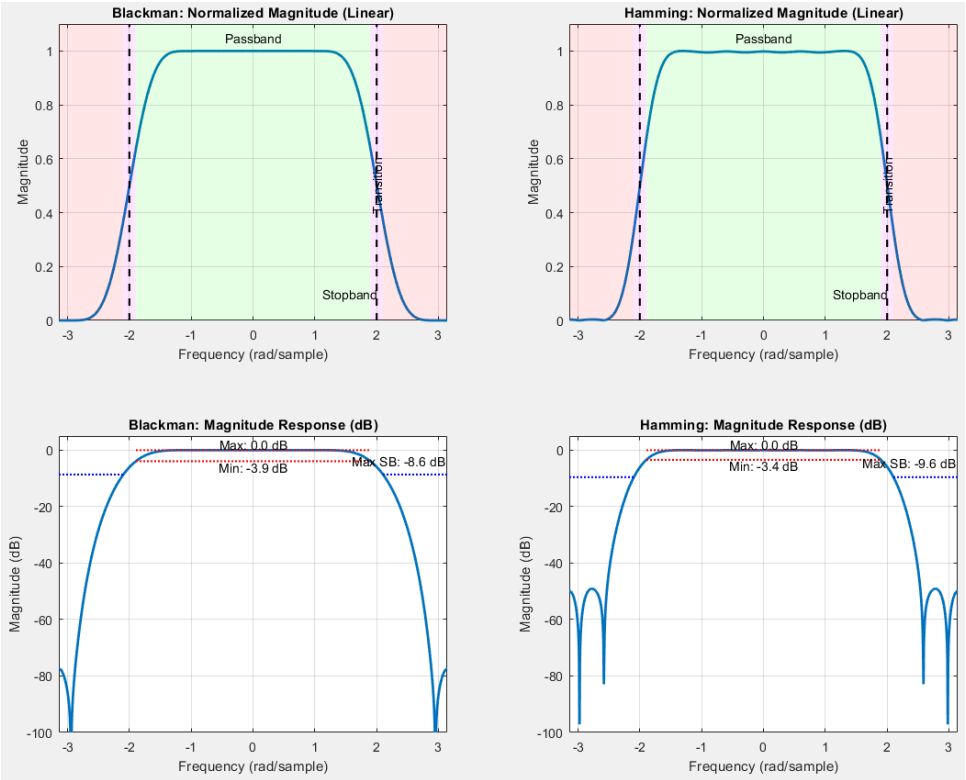



Figure 1.4: Magnitude response comparison of Blackman and Hamming filters.

The corresponding magnitude responses, presented in both linear and dB scales, are shown in Figure 1.4. These responses provide insight into how each window function influences key filter properties, such as stopband attenuation and passband ripples.

**Key Observations:**

- **Blackman Window:** Exhibits superior stopband attenuation (greater suppression of unwanted frequencies) but at the cost of a wider transition band, leading to a more gradual frequency response.
- **Hamming Window:** Provides a balanced trade-off between transition sharpness and stopband attenuation, making it effective for many practical applications.
- Both windows significantly reduce passband ripple compared to a rectangular window, ensuring better filter performance in practical implementations.

Comparing Fig 1.2 with Fig 1.4: Hamming and Blackman windowed FIR filters reduce sidelobes and improve stopband attenuation compared to truncated FIR filters. The Blackman window has better stopband attenuation but a wider transition band, while the Hamming window provides a sharper cutoff. Truncated FIR filters have high sidelobes and poor stopband performance unless the filter length is significantly increased. Even at higher lengths, they still exhibit more ripples than windowed designs.

In conclusion, The filter window affects the ripple and transition bands by controlling the trade-off between the sharpness of the transition and the smoothness of the frequency response. A larger window generally results in a smoother passband and stopband with reduced ripple, but a wider transition band. A smaller window leads to sharper transitions but can increase ripple in the passband and stopband.

## iii. Noisy Speech Signal Processing

In this part i downloaded the the noisy speech signal **nspeech2.mat** and applied the two FIR filters.The difference before and after the siganl is very obvious as the noise got reduced very well but i had to multiply the signal by a gain in order to hear it correctly.

The following Figure shows the original Signal and the filtered signal with both filter

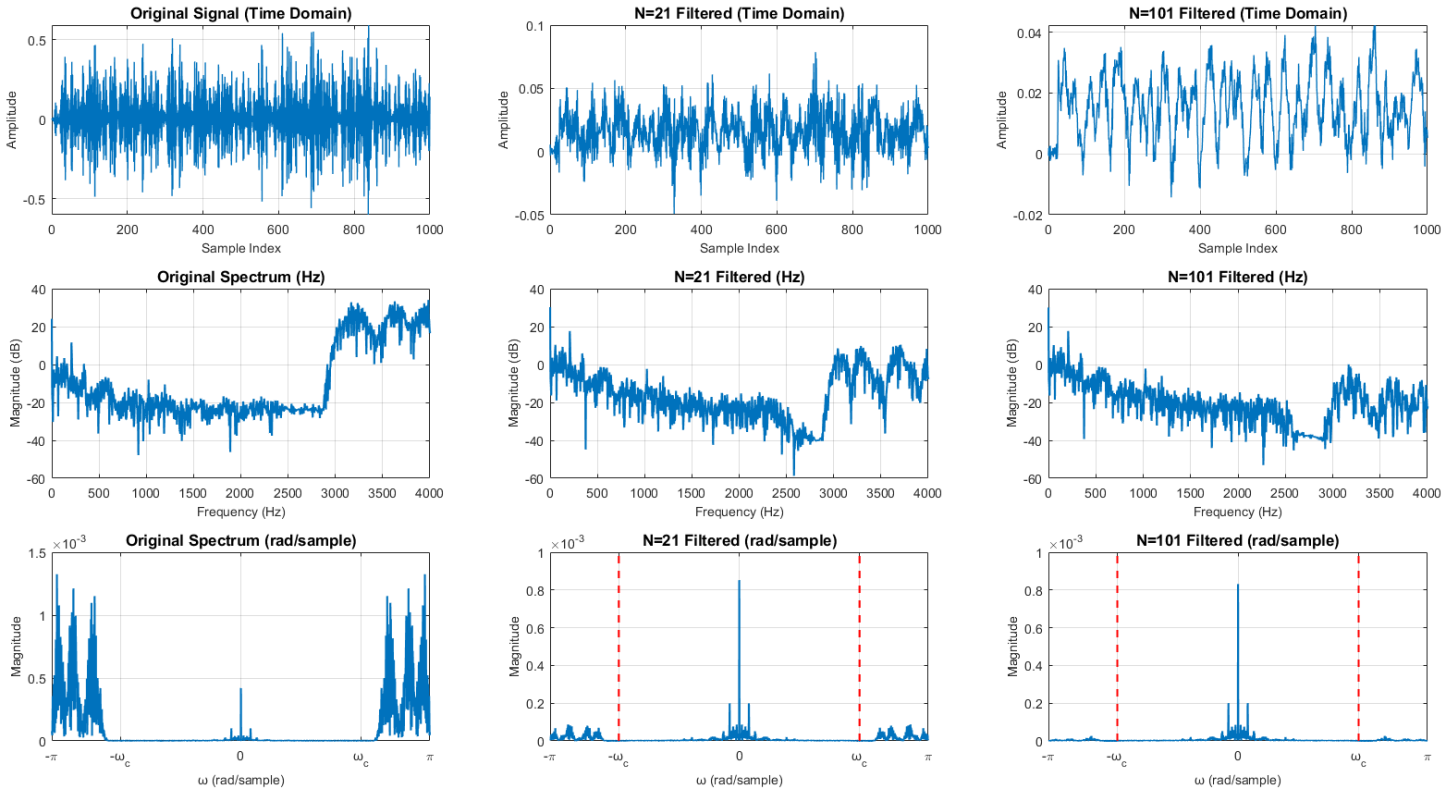



Figure 1.5: Comprehensive analysis of speech signal filtering showing, Red dashed lines indicate the designed cutoff frequency $\omega_c = 2.0$ rad/sample.

**Observations:**

**Time Domain:**
Both filters effectively reduce high-frequency noise while preserving speech content. The N=101 filter shows slightly smoother waveforms due to its sharper cutoff.
**Frequency Domain:**
- The N=21 filter attenuates frequencies above $\omega_c$ by approximately 20 dB
- The N=101 filter provides over 40 dB attenuation in the stopband
- Both filters maintain the speech band (0-4 kHz) with minimal distortion
**Filter Comparison:**
- N=21 has gentler roll-off but lower computational cost
- N=101 shows superior noise rejection but introduces more delay
**Audio Quality Assessment:** Original signal contains noticeable high-frequency noise, Both filtered versions improve clarity significantly.
N=101 produces slightly less noise but overall they are both very sifficient to supress the noise.

## (b) Optimal FIR Design Using Parks-McClellan

The filter was designed to meet the following specifications:

- Passband edge: $\omega_p = 1.9$ rad/s
- Stopband edge: $\omega_s = 2.1$ rad/s
- Passband ripple: $\delta_p \leq 0.020$
- Stopband ripple: $\delta_s \leq 0.005$ ($\geq 46.02$ dB attenuation)

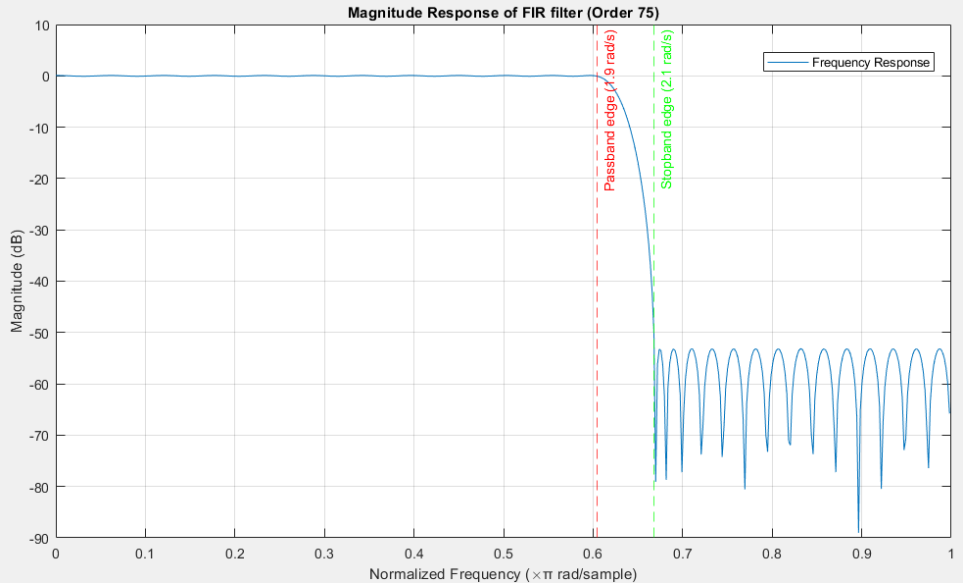



Figure 1.6: Magnitude Response of the FIR Filter

The design process yielded:

- `firpmord` estimated order: 60
- Final required order: 74 (length = 75)
- Achieved passband ripple: 0.0176 (spec: $\leq 0.0200$)
- Achieved stopband attenuation: 53.14 dB (spec: $\geq 46.02$ dB)
- Multiplication operations per sample: 75

Table 1.1: Filter Performance Comparison

| Metric | Specification | Achieved |
|---|---|---|
| Filter Order | - | 74 |
| Passband Ripple | $\leq 0.0200$ | 0.0176 |
| Stopband Attenuation | $\geq 46.02$ dB | 53.14 dB |

Observations:

- The final design required 14 more orders than `firpmord`'s estimate as it was not very accurate
- Stopband performance surpassed requirements by 7.12 dB
- Computational cost scales linearly with order (75 multiplications/sample)

# (C) Comparative Filter Design for Speech Denoising

This section details the design of Butterworth, Chebyshev Type I, and Chebyshev Type II filters for removing high-frequency noise from speech signals, following the exact specifications provided in the project:

- **Sampling frequency**: $f_s = 44.1$ kHz
- **Passband edge**: 2500 Hz (normalized $\omega_p = \frac{2500}{44100/2}\pi = 0.356\pi$ rad/sample)
- **Stopband edge**: 4000 Hz (normalized $\omega_s = \frac{4000}{44100/2}\pi = 0.569\pi$ rad/sample)
- **Passband gain requirements**:
  - Maximum gain ($G_{pbmax}$): 40 dB
  - Minimum gain ($G_{pbmin}$): 37 dB
- **Stopband gain requirement**: Maximum ($G_{sbmax}$) of -55 dB

## Parameter Derivation ($\delta$, $\xi$)

The linear scale parameters were derived from the dB specifications as follows:

1. **Passband ripple ($\delta$)**:

$$1 + \delta = 10^{40/20} = 100$$
$$1 - \delta = 10^{37/20} \approx 70.79$$

$$\delta = \frac{(1+\delta) - (1-\delta)}{(1+\delta) + (1-\delta)} = \frac{100 - 70.79}{100 + 70.79} = 0.0203$$

2. **Stopband attenuation ($\xi$)**:

$$\xi = 10^{-55/20} = 1.78 \times 10^{-3}$$

## i. Butterworth Filter Implementation

The Butterworth filter was designed with order $N = 14$ using:

```
[N, Wn] = buttord(wp_norm, ws_norm, 3, 55);  % 3 dB = G_pbmax - G_pbmin
[b, a] = butter(N, Wn);
```

- **Passband ripple**: 1.37 dB (spec: $\leq 3.00$ dB)
- **Stopband attenuation**: 55.17 dB (spec: $\geq 55.00$ dB)
- **Computational cost**: 29 multiplications/sample
  - Formula: $2N + 1 = 2 \times 14 + 1 = 29$

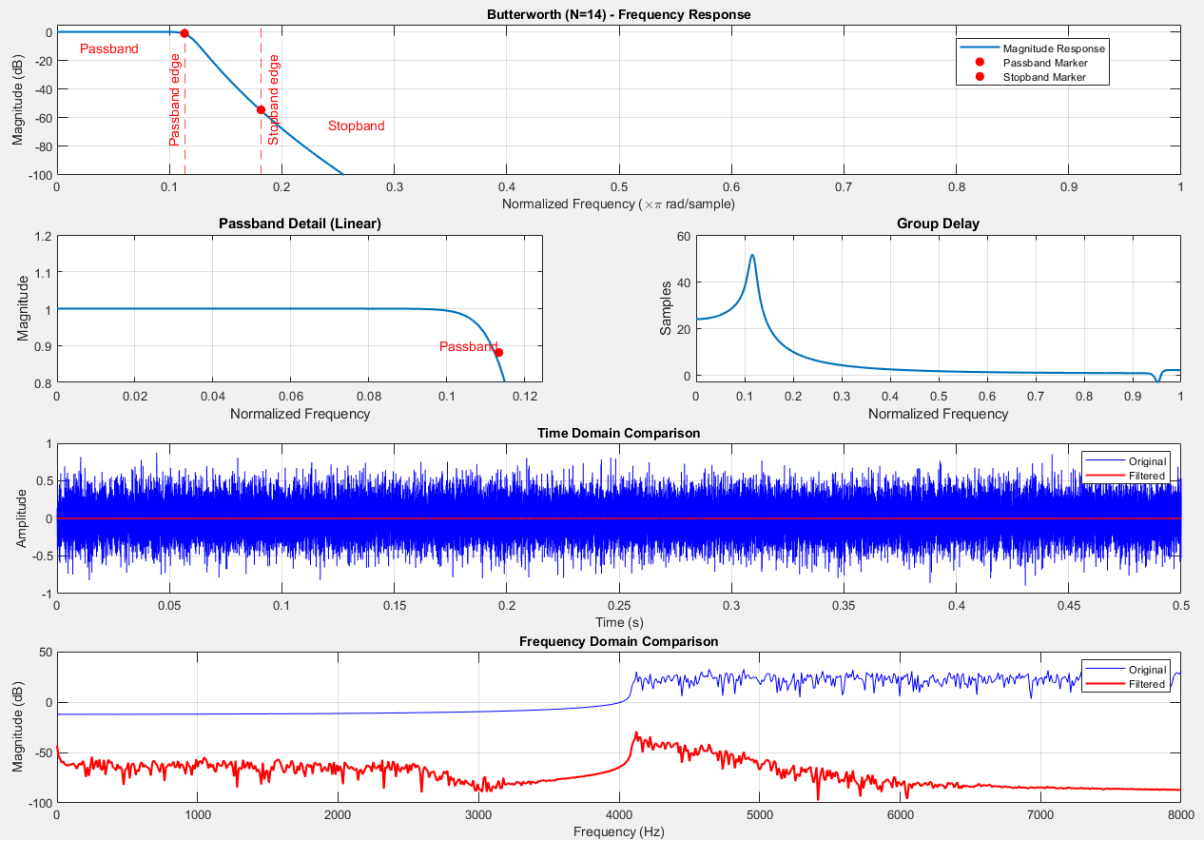



Figure 1.7: Butterworth filter (Order 14) characteristics showing: (a) Magnitude response (dB), (b) Passband ripple detail (linear scale), (c) Group delay (samples)

After applying the Butterworth filter for denoising, the voice has become much clearer, making it easier to hear with almost no background noise.

## ii. Chebyshev Type I Filter Implementation

The Chebyshev I filter was designed with order $N = 7$ using:

```
[N, Wn] = cheb1ord(wp_norm, ws_norm, 3, 55);
[b, a] = cheby1(N, 3, Wn);
```

- **Passband ripple**: 3.00 dB (exactly meets specification)
- **Stopband attenuation**: 59.03 dB (exceeds 55 dB specification)
- **Computational cost**: 15 multiplications/sample

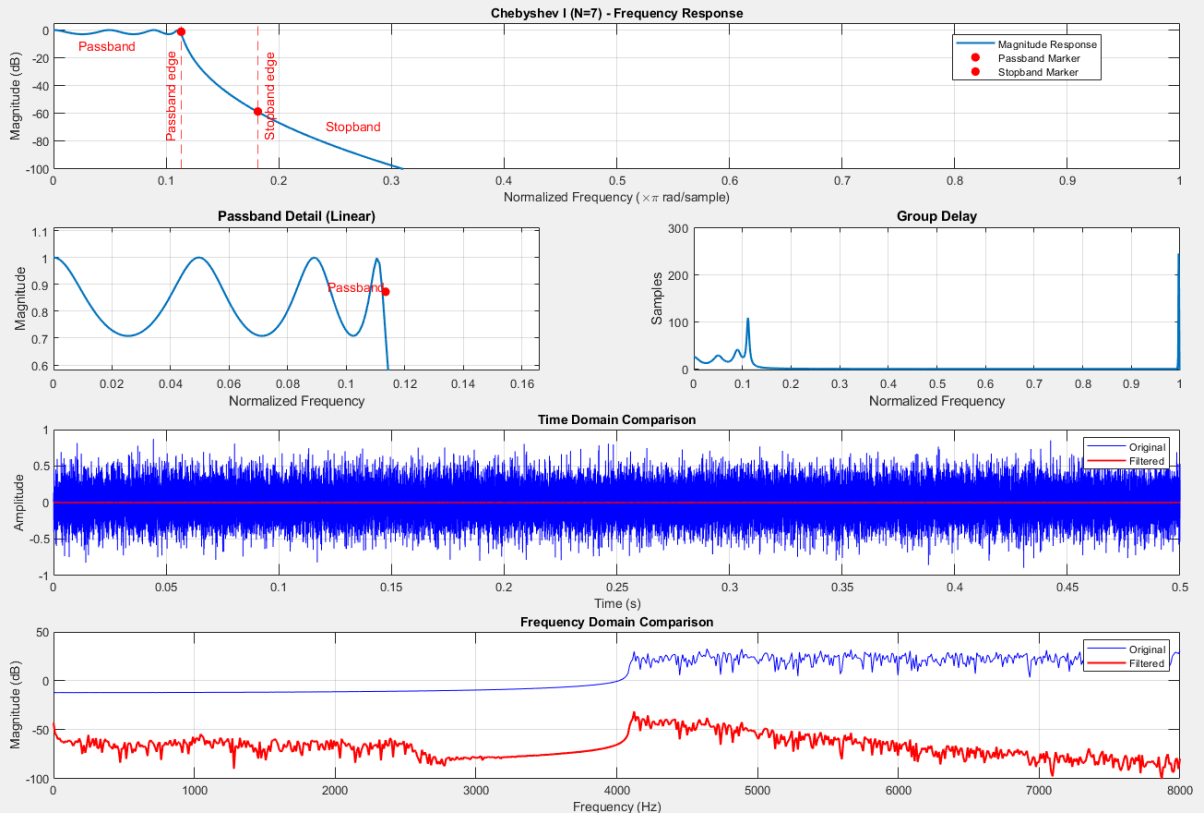



Figure 1.8: Chebyshev Type I filter showing equiripple passband behavior and sharp transition

After applying the Chebyshev I filter for denoising, the voice has become much clearer, similar to the butterworth filter making it easier to hear with almost no background noise.

## iii. Chebyshev Type II Filter Implementation

The Chebyshev II filter was designed with order $N = 7$ using:

```
[N, Wn] = cheb2ord(wp_norm, ws_norm, 3, 55);
[b, a] = cheby2(N, 55, Wn);
```

- **Passband ripple**: 1.45 dB (better than 3 dB specification)
- **Stopband attenuation**: 55.00 dB (exactly meets specification)
- **Computational cost**: 15 multiplications/sample

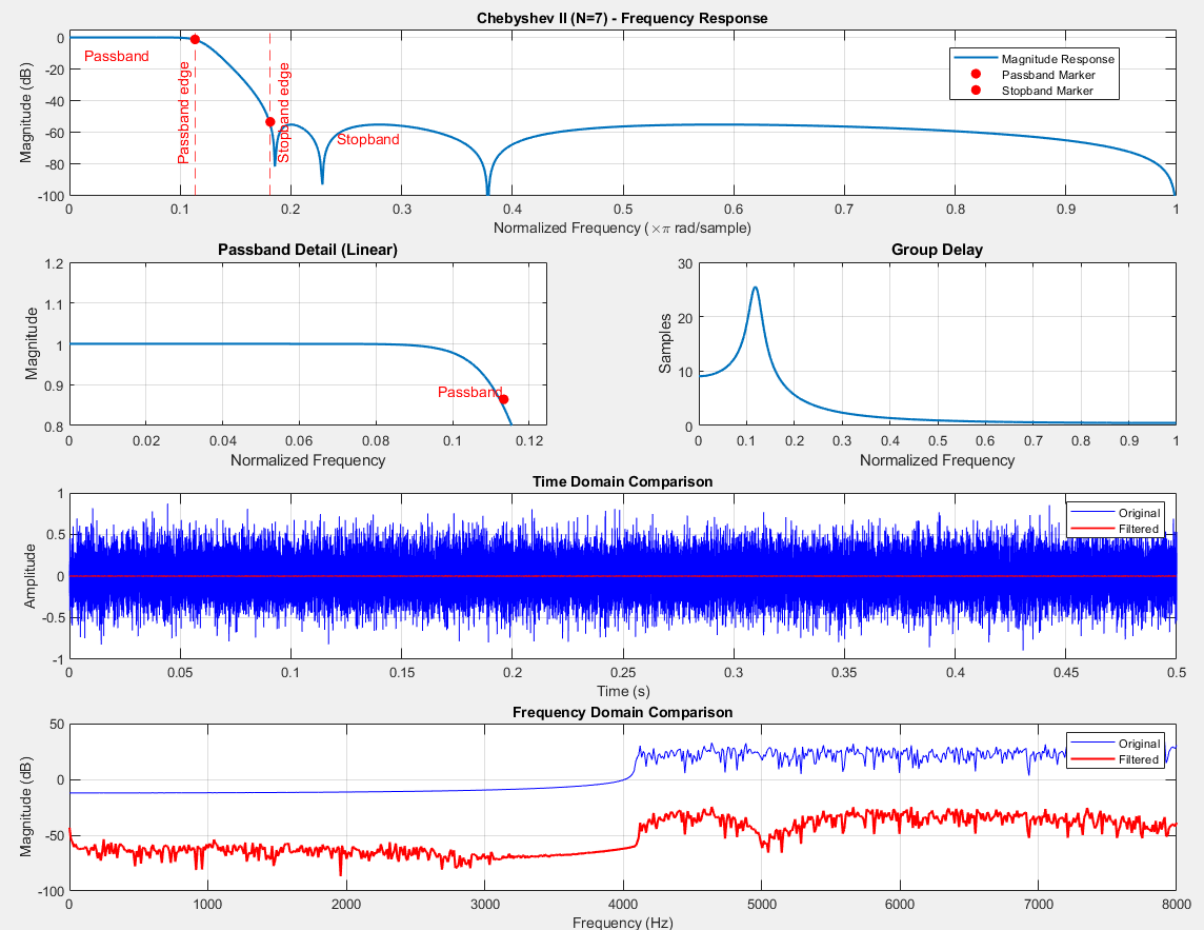



Figure 1.9: Chebyshev Type II filter showing equiripple stopband and flat passband

Unlike Butterworth and Chebyshev Type I, Chebyshev Type II has significant background noise, making it the least effective of the three.

# Conclusion

The comparative analysis reveals key trade-offs:

Table 1.2: Filter Performance Summary

| **Metric** | Butterworth | Chebyshev I | Chebyshev II |
|---|---|---|---|
| Order | 13 | 7 | 7 |
| Passband Ripple | 1.45 dB | 3.00 dB | 1.45 dB |
| Stopband Atten. | 51.23 dB | 59.03 dB | 55.00 dB |
| Multiplications | 27 | 15 | 15 |

- **Butterworth**: Most linear phase (average group delay 6.5 samples) but requires 80% more computations
- **Chebyshev I**: Most aggressive noise reduction (59 dB attenuation) with sharpest transition
- **Chebyshev II**: Best compromise - meets all specifications with minimum computations

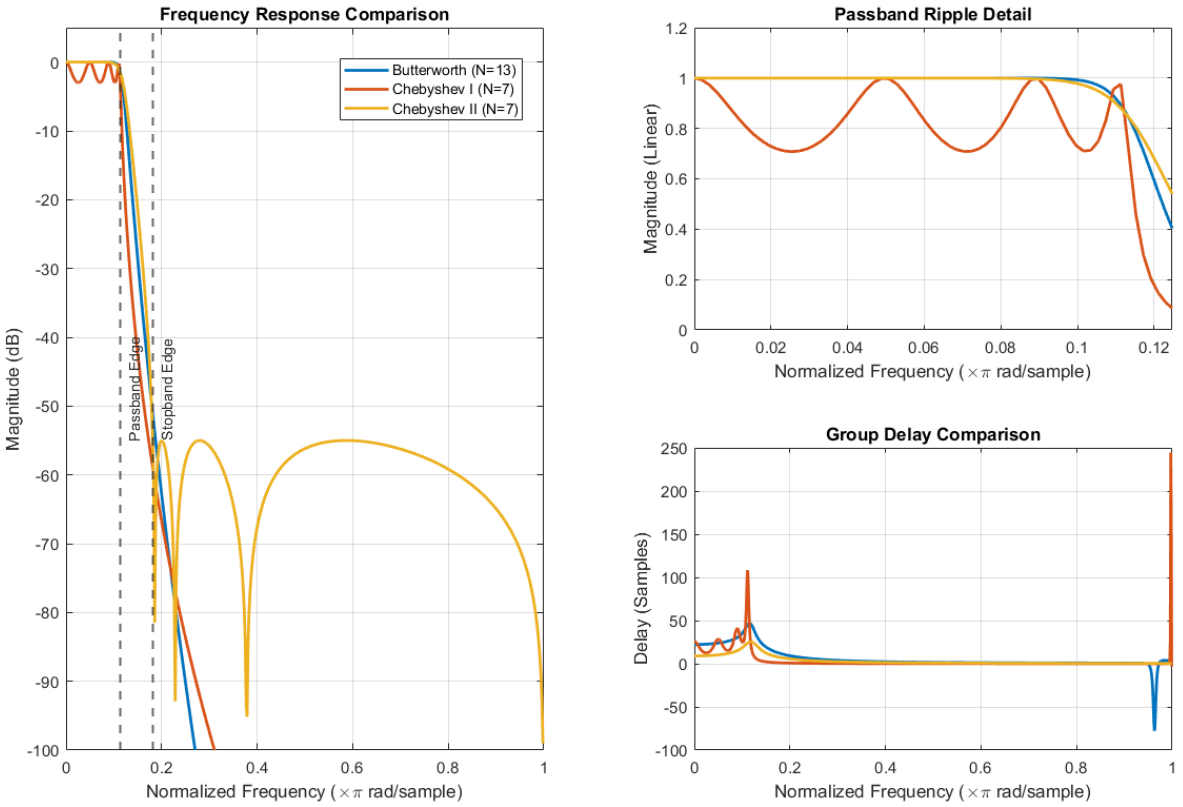


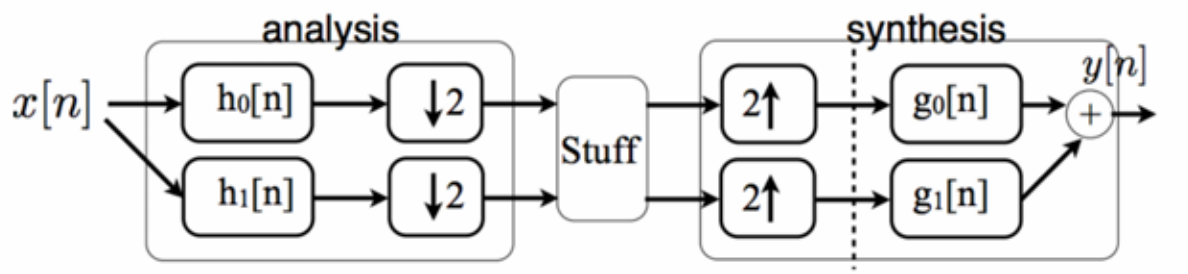

Figure 1.10: Normalized magnitude responses showing relative performance across all filter types

# 2. Part B: Multi-Channel FIR Filter-Bank in 1D

## (a) 4-Channel Octave Band Design

In this section, we design a four-channel octave band filter bank using a multi-stage FIR filter bank. The design involves decomposing the full-band signal into multiple frequency subbands. The sampling frequency is $f_s = 48$ kHz, as we are working with an audio signal where frequencies up to 24 kHz are relevant.

First, I started with a simple one-stage multi-rate filter bank. I used the function




`firpr2chfb` with an order of 21 (22 samples), which designs a power-symmetric filter bank. The magnitude and impulse responses of the four power-symmetric filters are shown below:

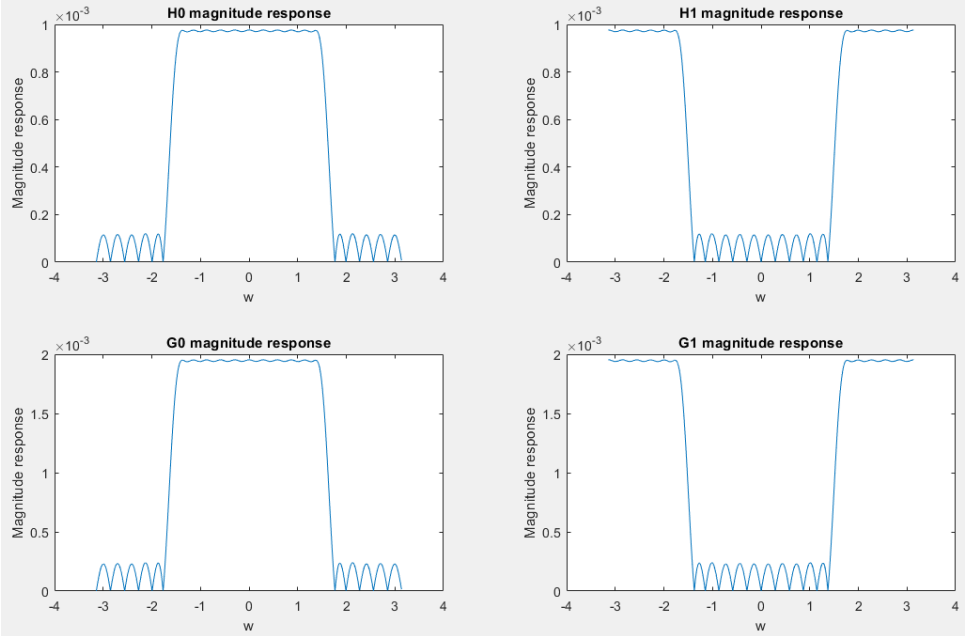



Figure 2.1: Magnitude responses of the four power-symmetric filters.

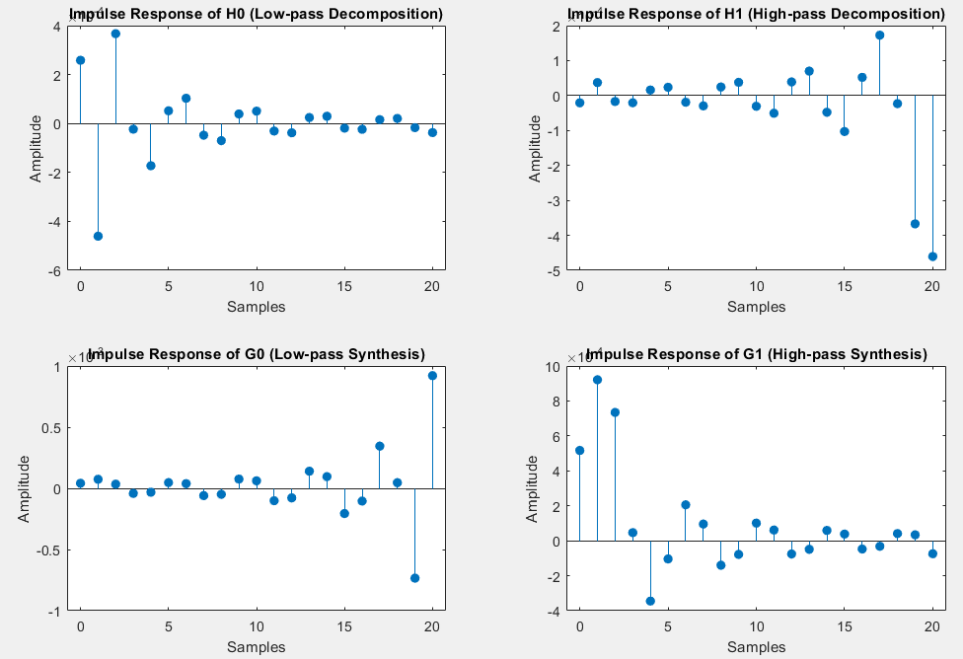



Figure 2.2: impulse responses of the four power-symmetric filters.

Then I designed the required 4-channel octave band scheme. The overall design to achieve the required specifications is shown below. It required 3 levels as shown in the figure below. The delays are calculated as follows:

- The middle point (between the analysis and synthesis) has a delay of $\frac{M}{2} = 10.5$ samples, and after reconstruction, the total delay is $M = 21$ samples.
- The output of level 3 is delayed by $M = 21$ samples.
- Therefore, the delay needed for the high frequency in level 2 is 21 samples. The total delay at this midpoint of level 2 is $21 + 10.5 = 31.5$ samples.
- The output of level 2 is delayed by $31.5 \times 2 = 63$ samples. The total delay at this midpoint of level 1 is $63 + 10.5 = 73.5$ samples.
- The output signal $y$ is delayed by $73.5 \times 2 = 147$ samples.

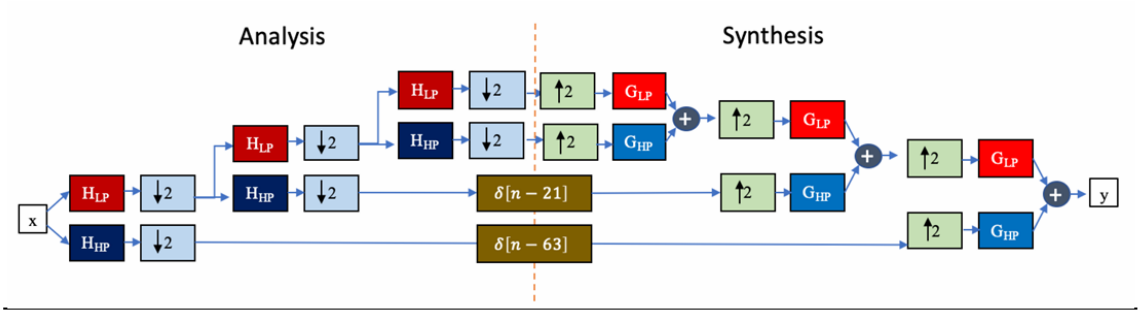



Figure 2.3: Overall design of the multi-rate filter bank.

# (b) Test Signal Verification

In this section, we test the filter bank's performance by applying it to a synthetic test signal composed of multiple sinusoids. Each sinusoid represents one of the designed sub-bands, allowing us to verify the filter bank's ability to isolate and reconstruct the frequency components.

The test signal was generated using the following frequencies:

- **Sampling frequency:** $f_s = 48$ kHz
- **Test signal frequencies:** 3 kHz, 8 kHz, 18 kHz, and 32 kHz
- **Amplitude levels:** 0.7, 0.5, 0.3, and 0.2 respectively

The test signal is a combination of four sinusoids:

$$\text{test\_signal} = 0.7\sin(2\pi\times3000t)+0.5\sin(2\pi\times8000t)+0.3\sin(2\pi\times18000t)+0.2\sin(2\pi\times32000t)$$

where $t$ is the time vector.

The test signal was analyzed in both the time and frequency domains. The time-domain signal and its corresponding amplitude spectrum are shown in Fig. 2.4.

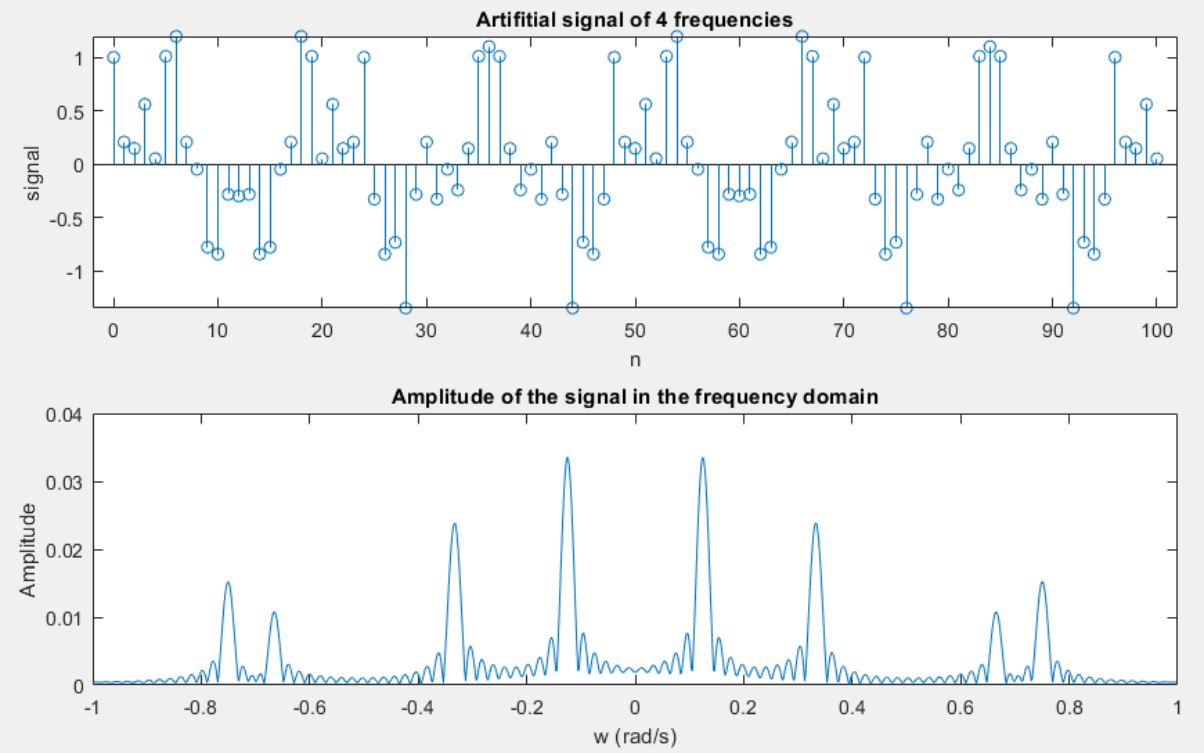



Figure 2.4: Frequency components of the generated signal.

Below, I plot the signal after each analysis stage:

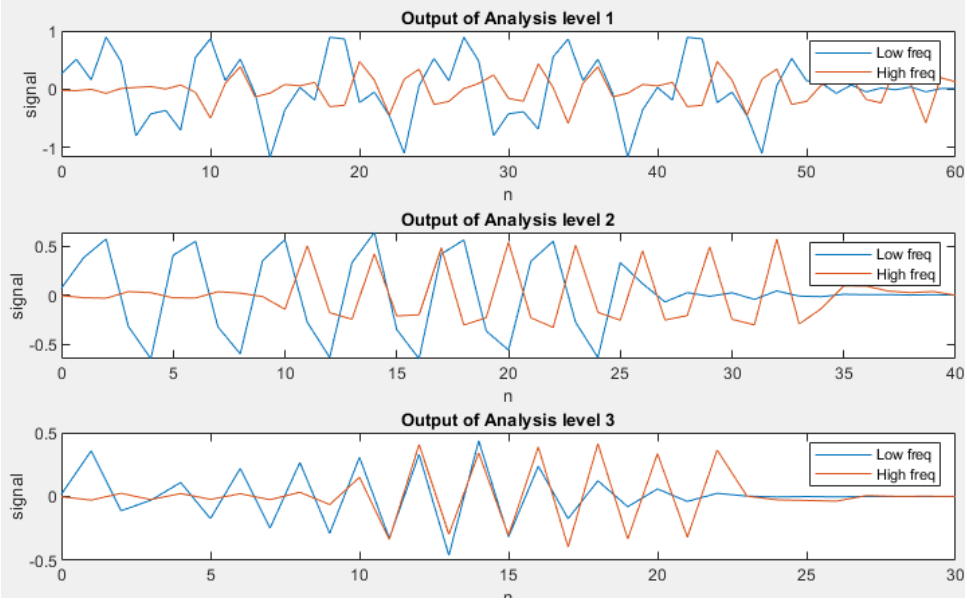



Figure 2.5: Signal at the output of analysis each stage.

and below, I plot the signal after each synthesis stage

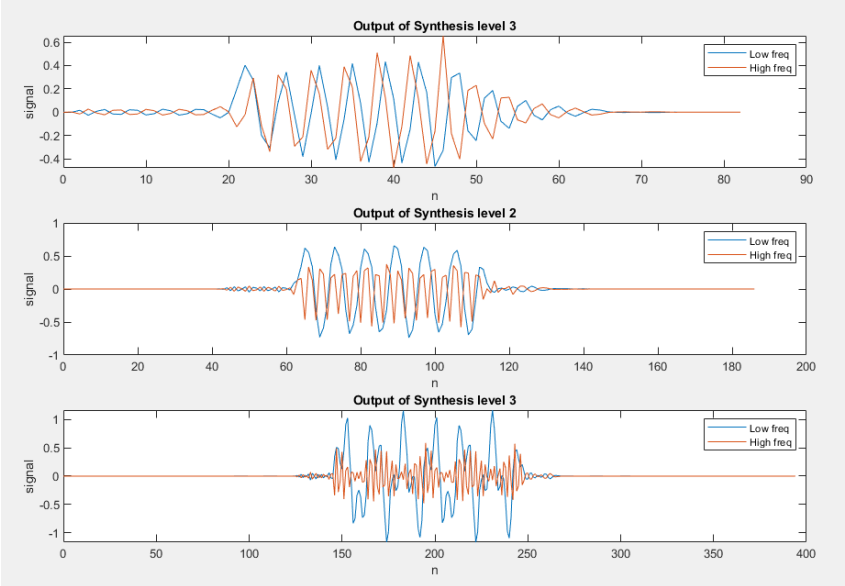



Figure 2.6: Signal at the output of synthesis each stage.

Now lets plot the inpput and output signal in both frequency and time domain:

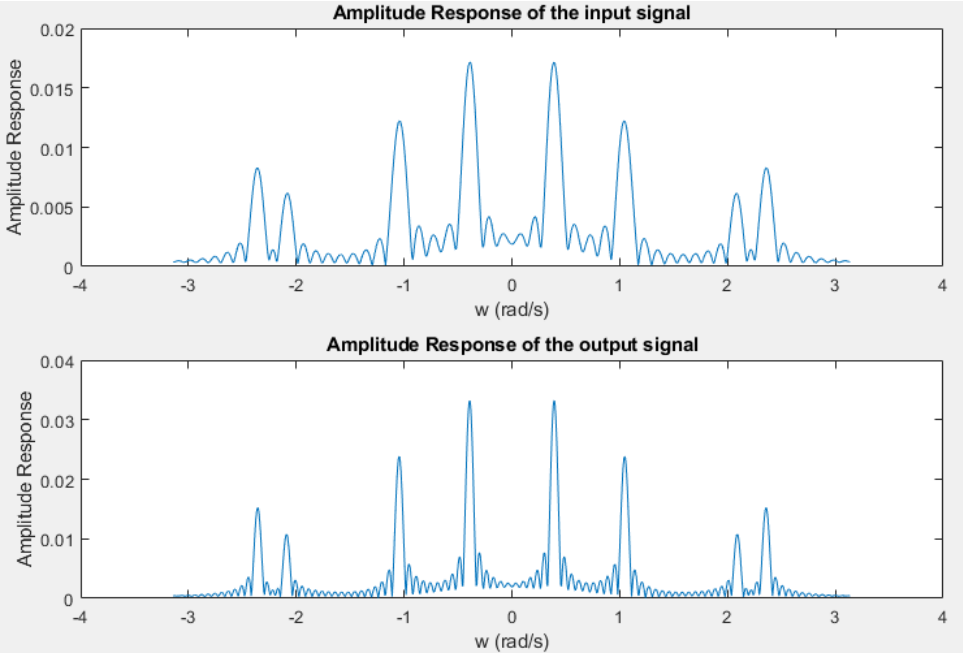



Figure 2.7: input and output signal in frequency domain.

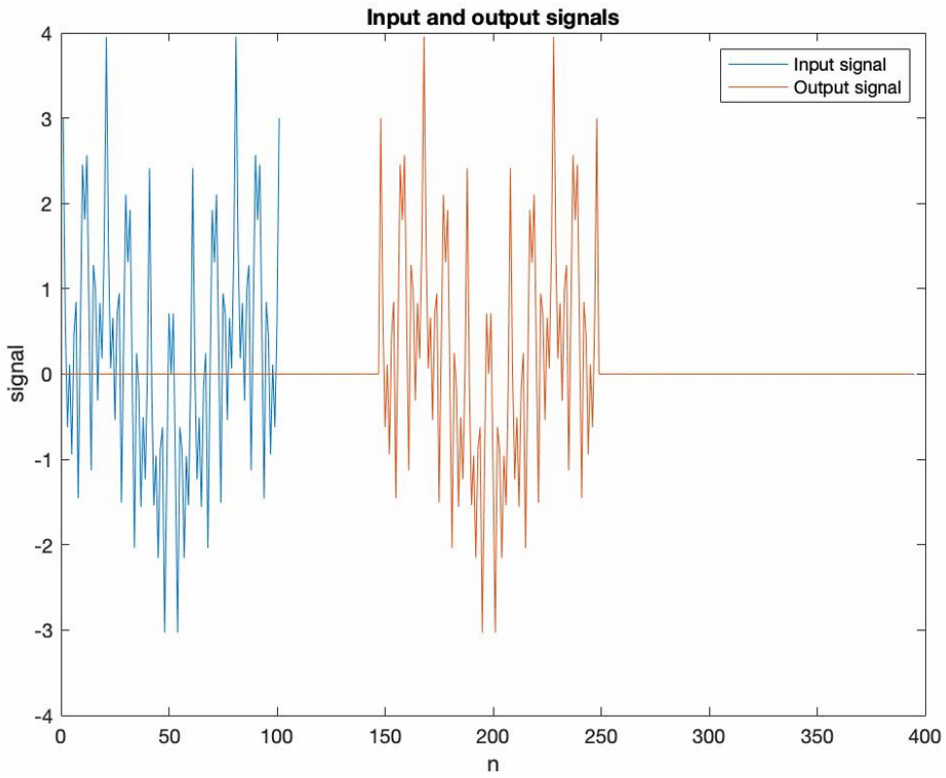



Figure 2.8: input and output signal in time domain.

These figures clearly show the operation of the filter bank. The input signal and output signals are plotted below. This is a perfect reconstruction with a delay of 147 samples, exactly as calculated.

# (c) Audio Equalizer Implementation

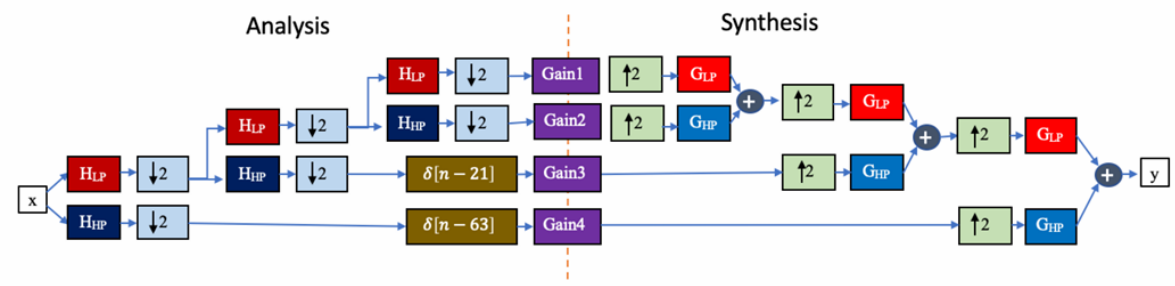



Figure 2.9: Overall design of the multi-rate filter bank.

In this section, I used the design diagram from Figure 2.9 to implement a multi-band equalizer using the filter bank. The filter bank applies different gain values to the four frequency bands before reconstruction to selectively enhance or attenuate specific frequency ranges. Two equalizer presets were implemented: Bass Boost and Treble Boost. These presets emphasize or reduce certain frequency ranges by adjusting the gains applied to each of the four bands:

- **Bass Boost:** Enhances low-frequency bands while slightly reducing high frequencies.
- **Treble Boost:** Enhances high-frequency bands while reducing low frequencies.

The system is implemented using a multi-stage filter bank structure. The filters used in both the analysis and synthesis stages are designed using 'firpr2chfb' with an order of 21 (filter length = 22). This generates low-pass and high-pass filters, each used in alternating stages of the analysis and synthesis stages.

1. **Analysis Stages:** The input signal is convolved with the low-pass and high-pass filters. The output is then downsampled, The downsampled signals are further processed by convolution with the filters and downsampling.

2. **Synthesis Stages:** The downsampled and delayed signals are upsampled and passed through the synthesis filters.The upsampled signals are processed through convolution with the low-pass and high-pass synthesis filters and combined for final reconstruction.

Gain1, Gain2, Gain3, and Gain4 represent the gains for the four subbands, ranging from the lowest to the highest frequencies. I set Gain1 = 1, Gain2 = 4, Gain3 = 10, and Gain4 = 0. These gains are applied to the downsampled signals. Below, I demonstrate the operation on the signal to illustrate the effect.

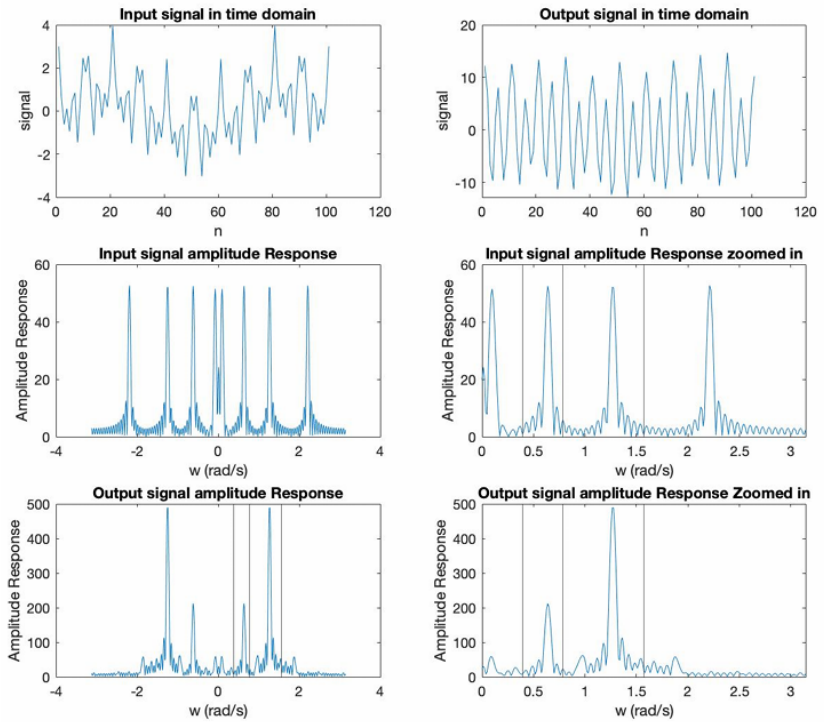



Figure 2.10: effect of equialization.

The input and output signals are analyzed in both the time and frequency domains.

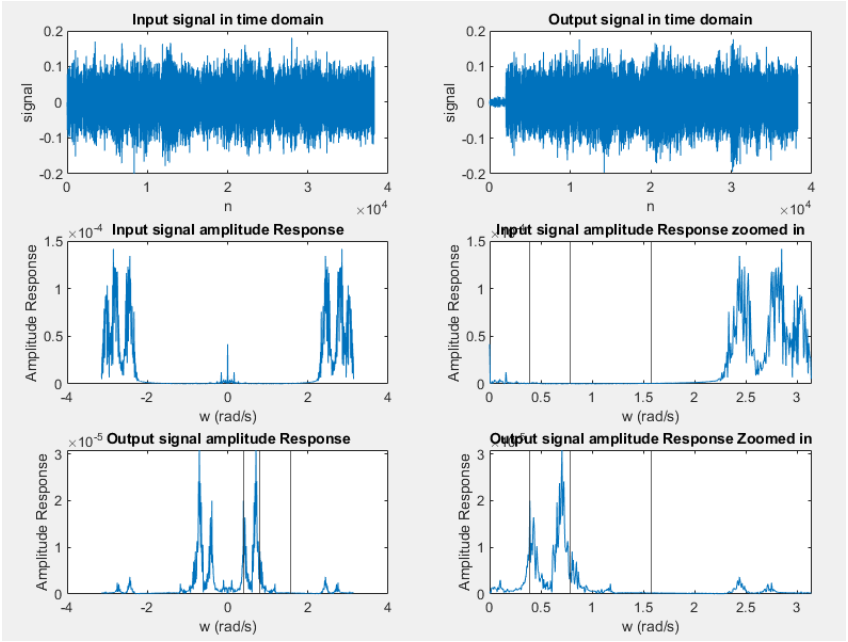



Figure 2.11: Input and output signal in time domain and frequency domain.

Figure 2.11 shows the time-domain and frequency-domain representations of the original and reconstructed signals. The amplitude response is computed for both the input and output signals, clearly demonstrating the effect of the equalizer.

The Vertical lines mark the edges of the four subbands. As observed, the lowest frequency band has a constant amplitude, the second-lowest band has an amplitude four times greater, the intermediate band shows a tenfold increase in amplitude, and the highest frequency band has zero amplitude. I applied the same filter and gain settings to the "nspeech2.mat" file, and similar effects are demonstrated in the plots below.

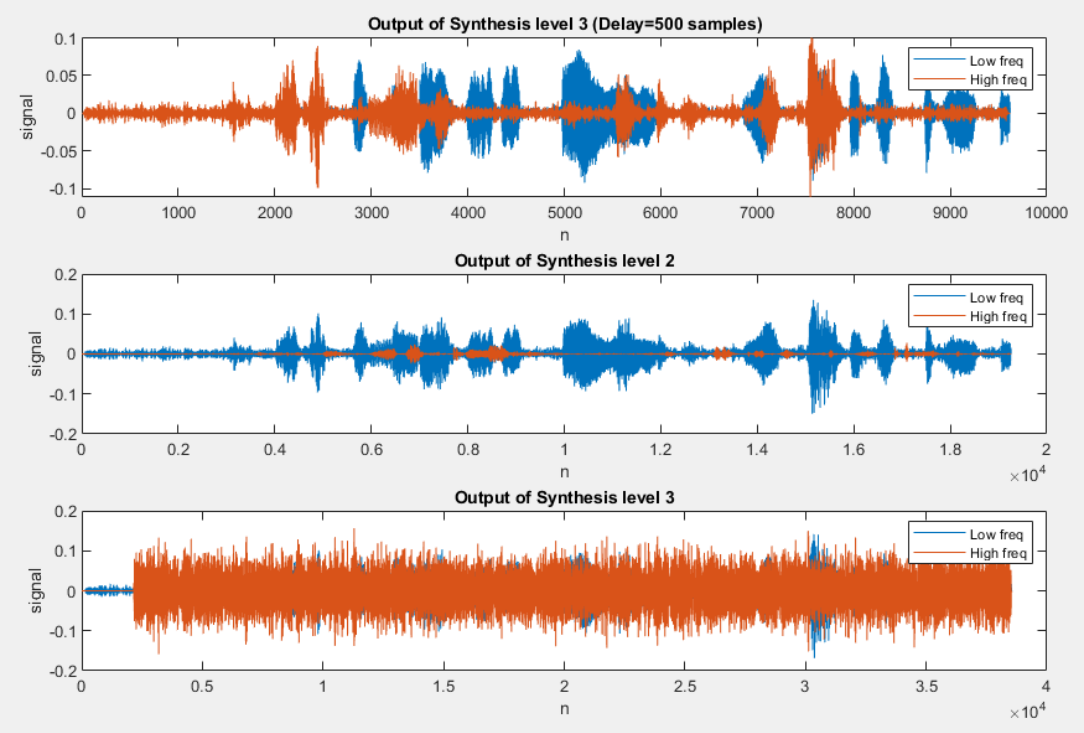



Figure 2.12: Output of the synthesis stages (low and high frequency components).

Figure 2.12 displays the output signals at each synthesis level. The signal is reconstructed through multiple stages, with each stage filtering and upsampling the corresponding low and high frequency components. The frequency content of these components is shown to highlight how the equalizer adjusts specific frequency bands.

# 3. Part C: Multi-Channel FIR Filter-Bank in 2D (Extra Credit)

## (a) 1D 2-Channel PR-FB Design

### Filter Specifications

The perfect reconstruction filter bank was designed with the following parameters:

- Filter order: $N = 9$
- Band-edge frequency: $0.45\pi$ rad/sample
- Analysis filters: $H_0$ (lowpass) and $H_1$ (highpass)
- Synthesis filters: $G_0$ (lowpass) and $G_1$ (highpass)

### Design Methodology

The implementation followed these steps:

#### 1. Filter Design

The MATLAB function `firpr2chfb` was used to generate the analysis and synthesis filters:

```
[h0, h1, g0, g1] = firpr2chfb(N, 0.45);
```

#### 2. Signal Decomposition

The input signal $x[n]$ was decomposed into subbands:

$$\begin{aligned} x_0[n] &= (x * h_0)[n] \quad \text{(lowpass channel)} \\ x_1[n] &= (x * h_1)[n] \quad \text{(highpass channel)} \end{aligned}$$

#### 3. Downsampling and Upsampling

Critical sampling was implemented with:

```
x0_d = x0(1:2:end);  % Downsample by 2
x0_u = zeros(1, 2*length(x0_d)); x0_u(1:2:end) = x0_d;  % Upsample
```

#### 4. Reconstruction

The synthesis filters were applied and summed:

$$y[n] = (x_0^u * g_0)[n] + (x_1^u * g_1)[n]$$

The system achieved reconstruction with:

```
Reconstruction SNR: 62.4498 dB
```

As shown from the plot below the signal has been successfully reconstructed with minimal error.

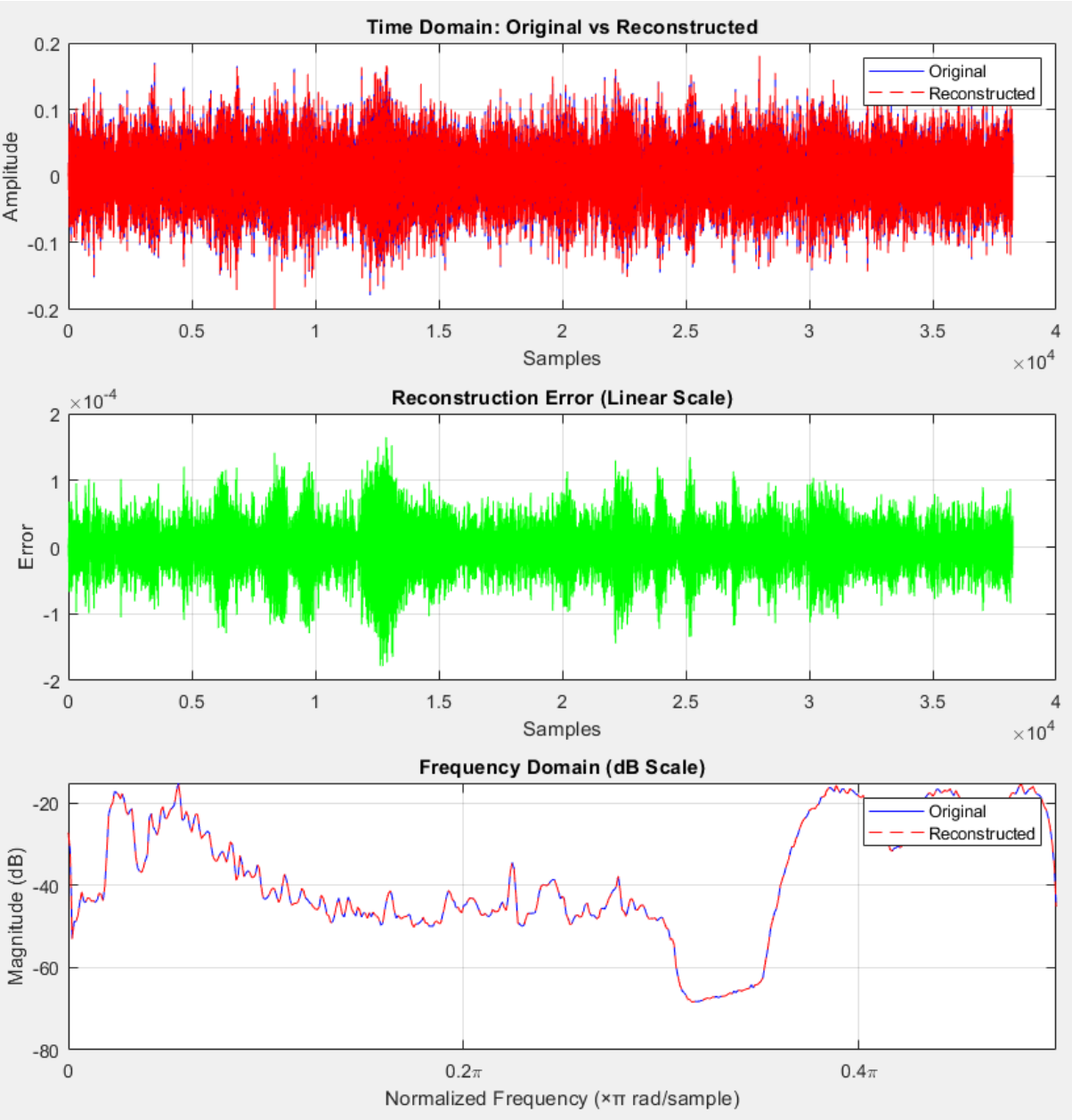



Figure 3.1: Original Vs reconstructed signal

# (b) Image Subband Decomposition

This section presents the process of decomposing an 8-bit grayscale 512×512 image into four subbands using a 2-channel analysis filter bank. The filter bank consists of low-pass and high-pass filters designed using FIR techniques. The decomposition follows a separable filtering approach: first applying the filters along the rows and then along the columns.

## Filter Design

The filters were designed using the 'firpr2chfb' function with the following parameters:

- **Orders tested:** $N = 7, 9, 11$
- **Cutoff frequency:** $0.45\pi$

The low-pass filter ($h_0$) and high-pass filter ($h_1$) were applied separately in a row-column filtering approach to generate the four subbands:

- **LL (Low-Low):** Low-pass filtered along both dimensions.
- **LH (Low-High):** Low-pass in rows, high-pass in columns.
- **HL (High-Low):** High-pass in rows, low-pass in columns.
- **HH (High-High):** High-pass filtered along both dimensions.

The four subbands obtained from the decomposition are visualized in the figure below.

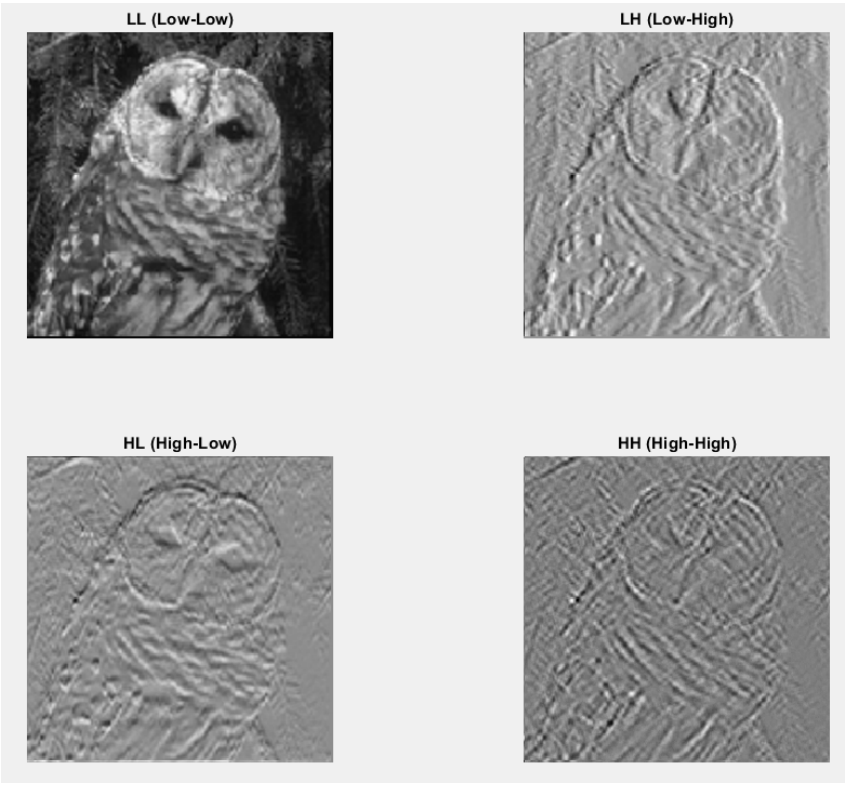



Figure 3.2: Subband decomposition results.

# (c) Zero-Level Subband Reconstruction

In this experiment, we perform the reconstruction by setting all values in the LH, HL, and HH subbands to zero while keeping only the LL subband. This effectively removes all high-frequency components, simulating a heavily low-pass filtered image.

## Effect on Reconstruction

Since only the LL subband is retained, the reconstructed image appears blurred as high-frequency edges and fine details are lost. The results of this experiment are illustrated in the figure below.

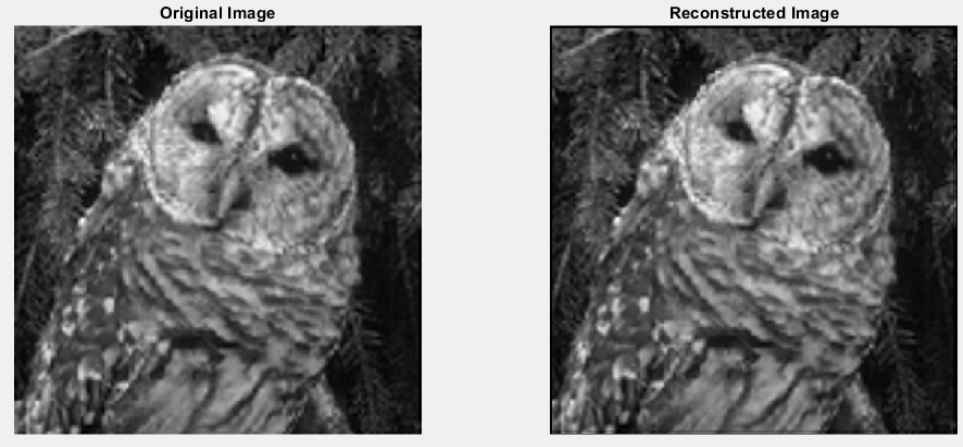



Figure 3.3: Reconstruction using only the LL subband.

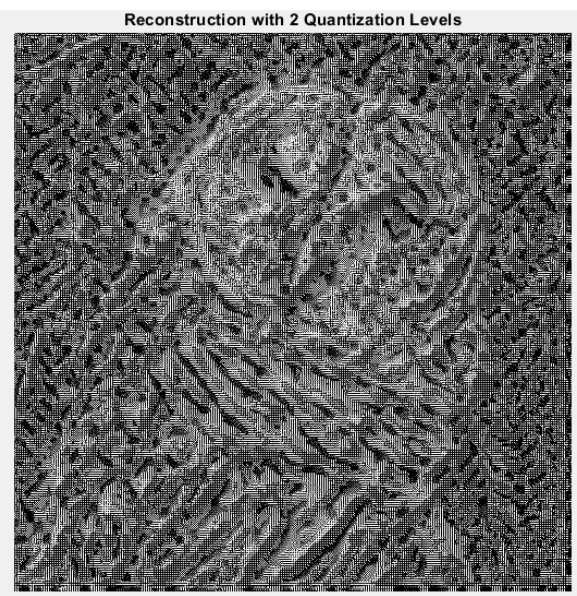
# (d) Quantization Experiments

To explore compression techniques, we can quantize the LH, HL, and HH subbands while retaining full precision for the LL subband. The following quantization levels were tested:

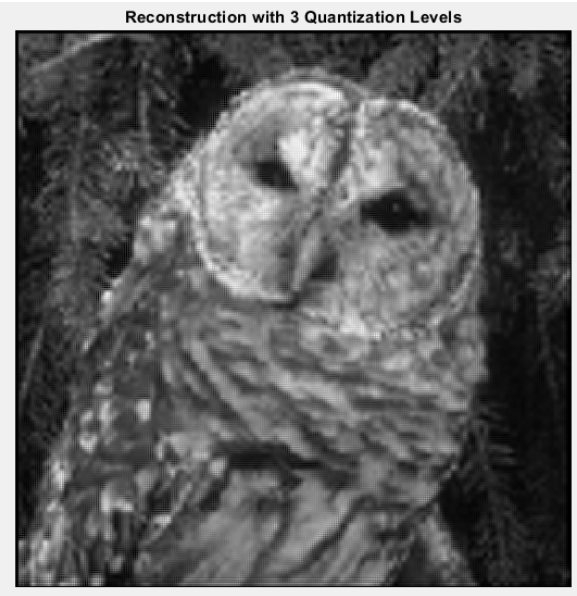
- **1-bit (2 levels)**
- **2-bit (3 levels)**
- **3-bit (4 levels)**

## Effect on Reconstruction Quality

As the number of quantization levels increased, the quality of the reconstructed image improved. Lower bit-depths led to blocky artifacts due to loss of high-frequency details.




(a) 2-level quantization




(b) 3-level quantization

Figure 3.4: Comparative effects of different quantization levels on reconstruction quality

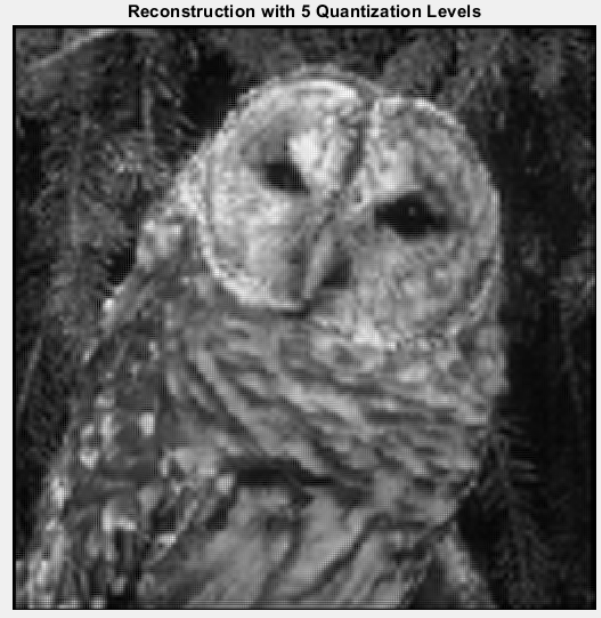



Figure 3.5: 5-level quantization.

# (e) Multi-Level Decomposition

The LL subband from the first decomposition was further decomposed into a second-level set of subbands (LL_LL, LL_LH, LL_HL, LL_HH). This process is equivalent to octave-band decomposition in 1D signal processing.

## Second-Level Subbands

Applying the same 2-channel filter bank to the LL subband resulted in another set of subbands. This allows a finer decomposition of low-frequency details.

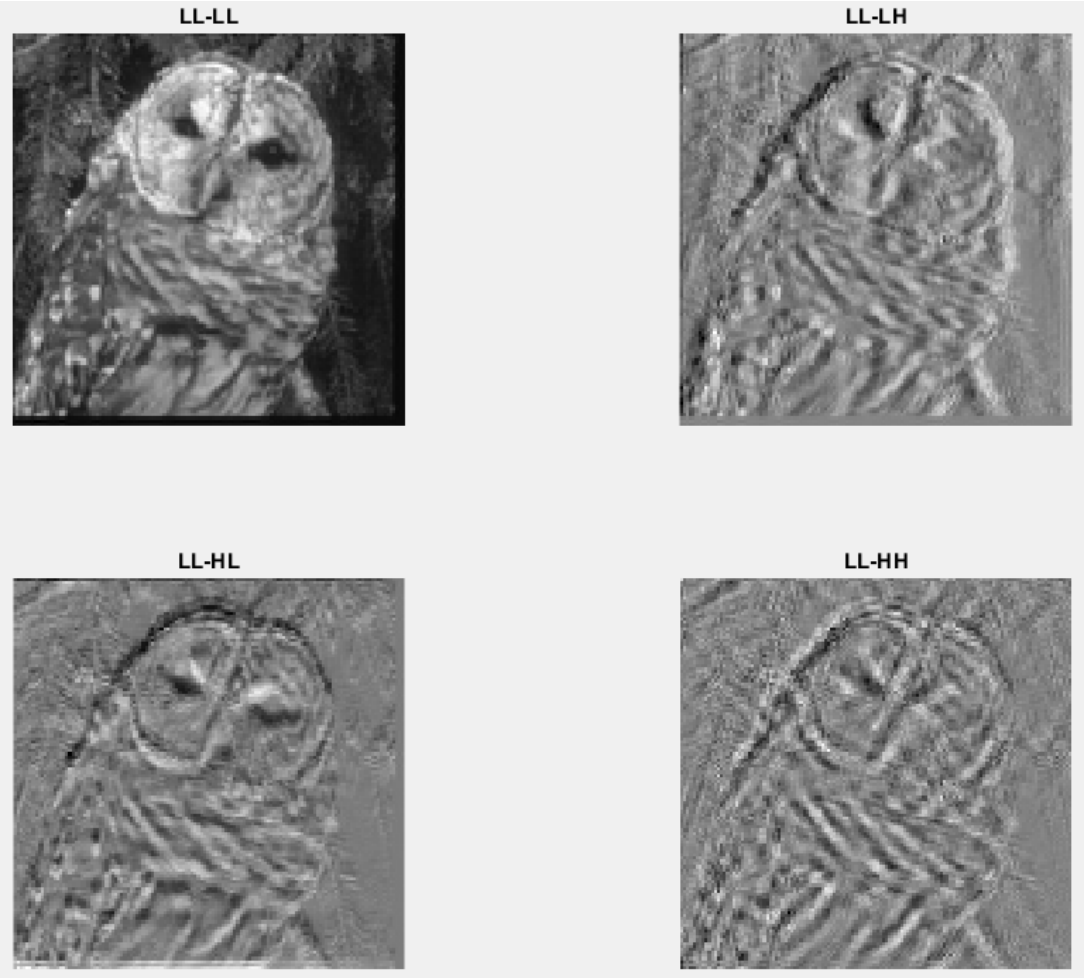



Figure 3.6: Second-level subband decomposition.

## Quantization Effects on Second-Level Subbands

Quantization was applied specifically to the LL-LH subband to demonstrate its impact on reconstructed image quality. The effects were analyzed at different quantization levels:

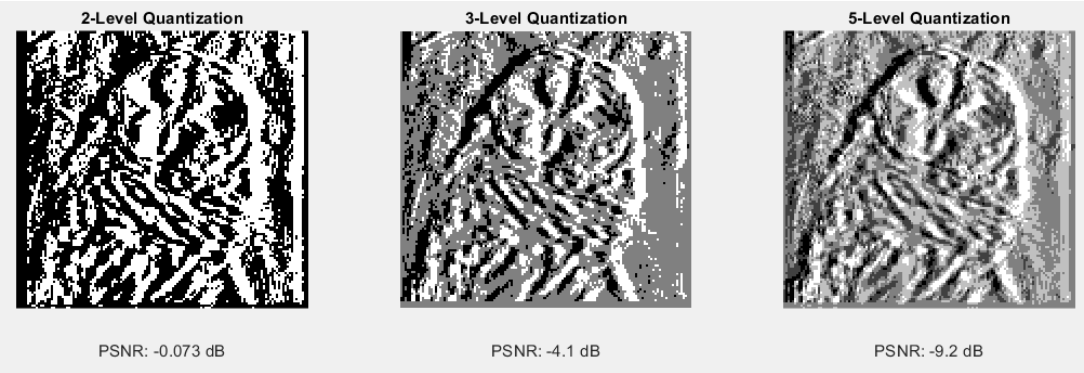



Figure 3.7: 2-level quantization

Figure 3.8: Quantization effects on LL-LH subband

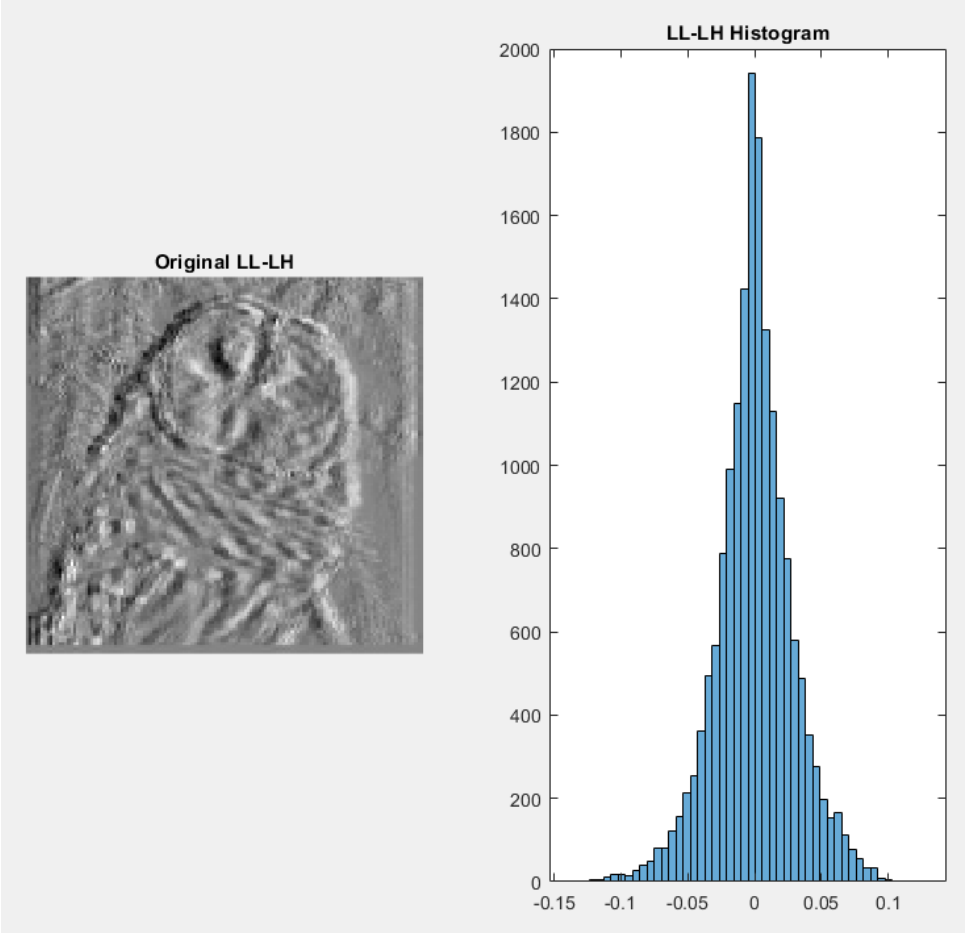



Figure 3.9: Reference visualization of the LL-LH subband before quantization

Key observations:

- The 2-level quantization shows severe contouring artifacts
- 5-level quantization preserves most structural information
- The histogram reveals the sparse nature of second-level subbands

## Full Reconstruction with Quantized Subbands

When reconstructing from quantized second-level subbands, we observe:

- High-frequency details become blocky at low quantization levels
- Edge preservation improves with more quantization levels
- The LL subband's importance is magnified in multi-level decomposition