Fall 2024
DSP Project Report

Eyad Hany 202101225
Khaled Ashraf 202100751
Amr Ahmed 202100302
Adham Saad 202100163

Communications and Information Engineering
Zewail City of Science and Technology

December 21, 2024

Contents

1 Windowed Sinc FIR Filter Design Method

The objective of this task was to design a low-pass FIR filter using the windowed sinc method. Four different window functions were applied to the sinc function:

- Rectangular
- Blackman
- Chebyshev
- Kaiser

The designed filters were analyzed in terms of their magnitude response, plotted on both linear and logarithmic (dB) scales.

The key goals of this report are:

1. To demonstrate the implementation of FIR filters using window functions.
2. To compare the magnitude response of filters designed with each window.
3. To analyze and explain the differences in performance of these windows.

1.1 Implementation of the Windowed Sinc FIR Filter

The windowed sinc method uses a sinc function (impulse response of an ideal low-pass filter) multiplied by a window function to design a realizable filter. The implementation followed these steps:

User Inputs

- Filter length (N) – must be odd.
- Sampling frequency (f_s).
- Cutoff frequency (f_c).
- Scale choice – linear or logarithmic.

Sinc Function

The sinc function was computed as:

$$h[n] = \frac{\sin(\omega_c \cdot \pi \cdot n)}{\pi \cdot n}$$

where

$$\omega_c = \frac{f_c}{f_s}$$

Window Functions

- Rectangular Window
- Blackman Window
- Chebyshev Window
- Kaiser Window

Normalization of Filter Coefficients

To ensure a unity DC gain, the filter coefficients were normalized:

$$h = \frac{h}{\text{sum}(h)}$$

Frequency Response Analysis

Using MATLAB's `freqz` function, the magnitude response was computed and plotted.

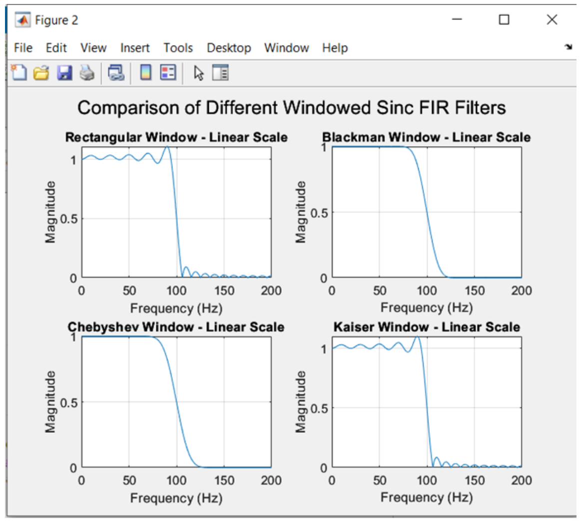
1.2 Results and Analysis

For this output, the following user inputs were used:

- Filter length (N): 41
- Sampling frequency (f_s): 400 Hz
- Cutoff frequency (f_c): 100 Hz
- Scale choice: Linear Scale, Logarithmic scale

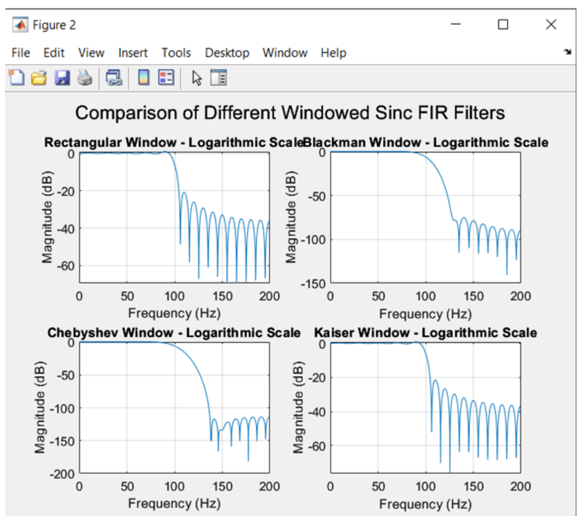
Linear Scale Results

The results include the window functions and their corresponding magnitude responses (linear scale shown).



Logarithmic Scale Results

The results include the window functions and their corresponding magnitude responses (logarithmic scale shown).



1.3 Comments on Window Functions

The window functions determine how the sinc function is "tapered" to reduce the side lobes in the filter response. The plots above show each window type applied to the sinc function:

- **Blackman Window**: Smooth, tapering window with good side lobe attenuation.
- **Chebyshev Window**: Allows controlled ripple in both the passband and stopband.

- **Kaiser Window**: Provides a flexible trade-off between main lobe width and side lobe height.

1.4 Comments on Magnitude Responses

Rectangular Window

- **Linear Scale:** The rectangular window results in a sinc-like frequency response with sharp transitions. However, there are significant ripples in the stopband (high side lobes).
- **Logarithmic Scale:** The stopband attenuation is poor, with side lobes not sufficiently suppressed.
- **Analysis:** The rectangular window has the sharpest transition but the worst stopband performance due to high side lobes.

Blackman Window

- **Linear Scale:** The transition is smoother compared to the rectangular window. Side lobes are much smaller, indicating better stopband attenuation.
- **Logarithmic Scale:** Stopband attenuation is excellent (around -60 dB), but the main lobe is wider.
- **Analysis:** The Blackman window offers good stopband suppression at the cost of a wider transition band.

Chebyshev Window

- **Linear Scale:** The Chebyshev window produces a sharp cutoff with ripples in both the passband and stopband.
- **Logarithmic Scale:** The ripples in the stopband are visible but controlled, providing a balance between main lobe width and stopband attenuation.
- **Analysis:** The Chebyshev window allows for user-defined ripple, balancing transition width and attenuation.

Kaiser Window

- **Linear Scale:** The Kaiser window produces a smooth and controlled transition band.
- **Logarithmic Scale:** The stopband attenuation is adjustable (via the beta parameter), and the transition band is slightly wider than the Chebyshev window.
- **Analysis:** The Kaiser window is flexible, allowing the designer to trade-off between main lobe width and stopband suppression.

2 Least Squares Design of FIR Filter

In this section, we discuss the framework for the design of a Finite Impulse Response (FIR) filter using the Least Squares (LS) and Weighted Least Squares (WLS) approaches.

2.1 Basis Matrix Definition

The design process begins with constructing the basis matrix (A), which represents the exponential relationship between the filter coefficients and the frequency response:

$$A_{k,n} = \exp(-j2\pi f_k(n-1))$$

- (A) is the basis matrix with dimensions $N \times L$.
- (N) is the number of frequency points.
- (L) is the filter length.
- $\{f_k\}$ represents the normalized frequency points.
- (n) is the index of the filter coefficients.

2.2 Least Squares Solution

The Least Squares solution aims to minimize the error between the desired frequency response H_d and the actual response of the filter. The solution for the filter coefficients h_LS is obtained using the MoorePenrose pseudoinverse of A:

$$h_{LS} = \text{real}\left(A^{\dagger} H_d\right)$$

- Where $A\hat{\dagger}$ is the pseudoinverse of A, computed as:

$$A^{\dagger} = \left(A^H A\right)^{-1} A^H$$

- (H) denotes the Hermitian transpose (conjugate transpose) of A.
- (H_d) is the desired frequency response vector.

The real part of the solution is extracted to ensure the filter coefficients hLS remain real-valued, which is a fundamental property of FIR filters.

2.3 Frequency Response of the Designed Filter

The frequency response $H_{LS}\left(e^{j\omega}\right)$ of the LS-designed FIR filter is given by:

$$H_{LS}(e^{j\omega}) = \sum_{n=0}^{L-1} h_{LS}[n]e^{-j\omega n}$$

- $h_{LS}[n]$ are the computed LS filter coefficients.
- ω is the angular frequency.

This equation demonstrates how the frequency response is derived from the time-domain coefficients.

2.4 Weighted Least Squares (WLS) Approach

To achieve greater control over the performance in different frequency bands, a weight function W(f) is introduced in the LS formulation. The weight function prioritizes the error minimization in specific bands such as the passband and stopband. The weight function W(f) is defined as:

$$W(f) = \begin{cases} W_{\text{pass}}, & f \leq f_p \\ W_{\text{stop}}, & f \geq f_s \end{cases}$$

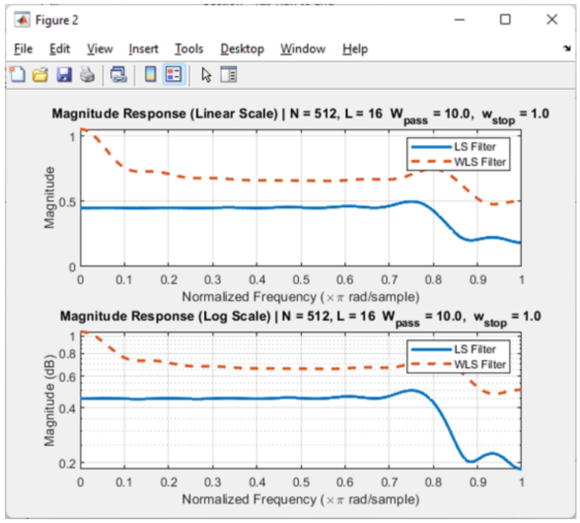
- W_{pass} is the weight assigned to the passband to control ripples.
- W_{stop} is the weight assigned to the stopband to ensure strong attenuation.
- f_p is the normalized frequency edge of the passband.
- f_s is the normalized frequency edge of the stopband.

The desired frequency response $H_d\left(f\right)$ corresponding to an ideal low-pass filter is expressed as:

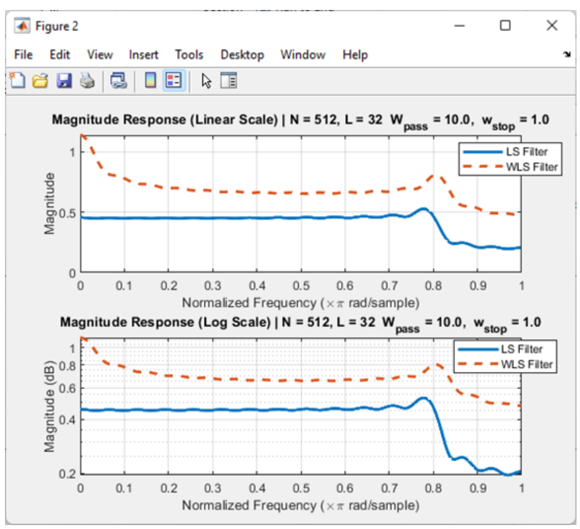
$$H_d(f) = \begin{cases} 1, & f \leq f_p \\ 0, & f \geq f_s \end{cases}$$

2.5 Varying the Filter Length (L)

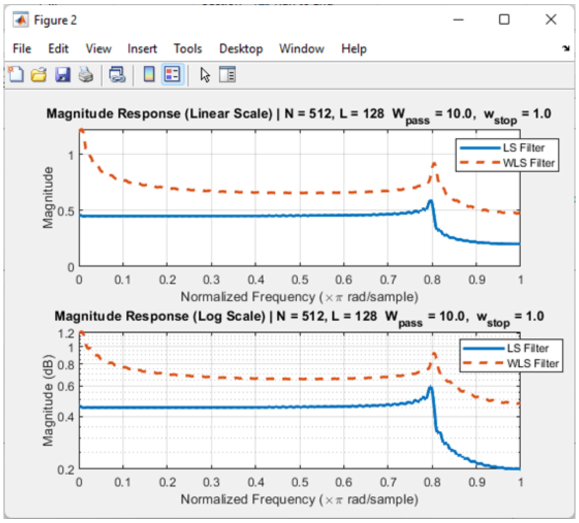
The filter length determines the number of coefficients in the impulse response. A larger L generally improves the approximation of the desired frequency response but increases computational cost and sharpens the transition bands.



For L=16, the filter exhibits wider transition bands, and the frequency response deviates from the ideal shape. The passband ripple is slightly noticeable, and the stopband attenuation is suboptimal. This demonstrates the limitation of small filter lengths in achieving precise performance.

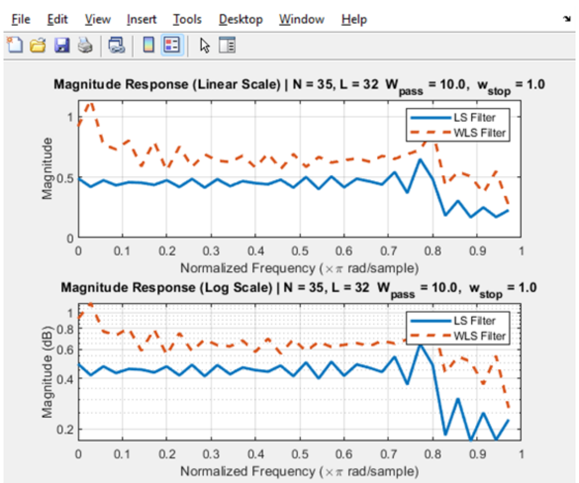


Increasing L = 32 improves the filter's response. The transition bands become sharper, and the approximation of the desired frequency response is significantly better. Passband ripple is reduced, and stopband attenuation improves.

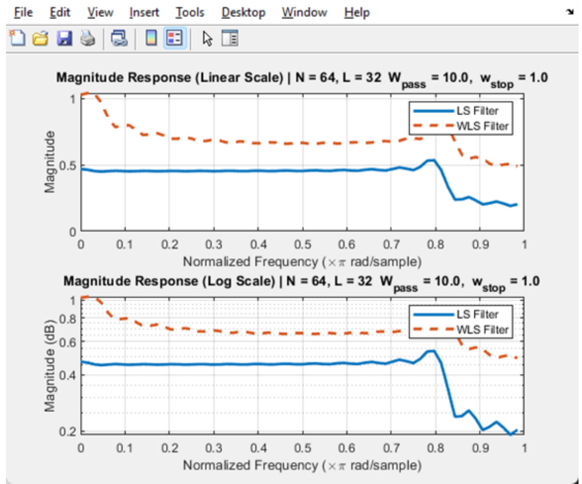


For L=128, the filter achieves near-ideal performance. The transition bands are very sharp, and both passband ripple and stopband attenuation are well-controlled. However, the computational cost increases with the filter length.

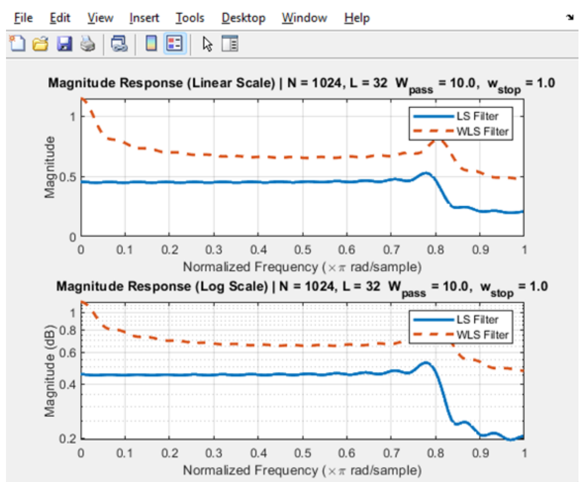
2.6 Varying N (Number of Frequency Points)



With N=35, the frequency response appears less smooth due to the coarse resolution of the frequency grid. The overall shape is visible, but finer details and variations are not captured.

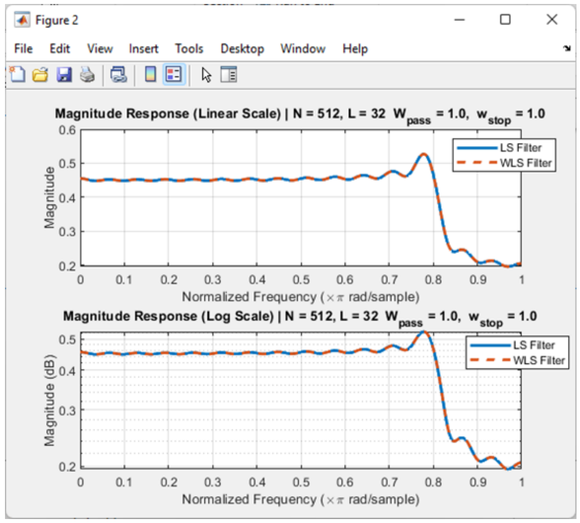


Increasing N to 64 results in a smoother and more continuous frequency response. The resolution is sufficient to capture the overall behavior of the filter with greater detail.

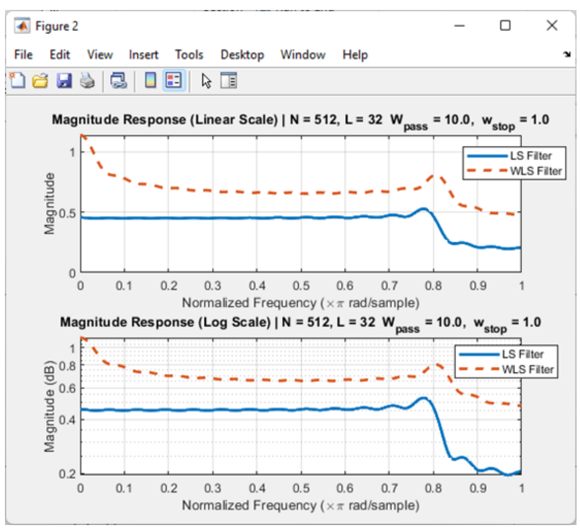


For N=1024, the frequency response achieves very high resolution. The plot is extremely smooth, and small variations in both the passband and stopband are clearly visible, showcasing the fine details of the filter performance.

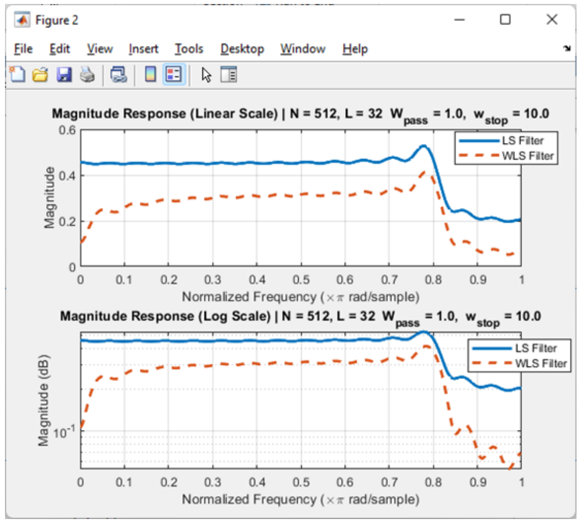
2.7 Varying Wpass and Wstop (Weights)



With equal weights Wpass=1, Wstop=1, the filter balances the errors between the passband and stopband. The passband ripple is moderate, and the stopband attenuation is decent, but there are visible trade-offs in both regions.



With higher emphasis on the passband Wpass=10, Wstop=1, the filter minimizes the ripple in the passband, achieving excellent performance there. However, the stopband attenuation slightly worsens, showing some residual magnitude.



When the stopband weight is increased (Wstop=10, Wpass=1), the filter prioritizes minimizing the stopband magnitude. As a result, the stopband attenuation improves significantly, but the passband exhibits more noticeable ripple.

2.8 Comments on LS & WLS Parameters

- **Filter Length (L)**: Increasing L sharpens the transition bands and improves accuracy but at higher computational costs.
- **Frequency Points (N)**: Increasing N smoothens the frequency response, enhancing resolution and capturing finer details.
- **Weights (W_{pass} and W_{stop})**: Adjusting weights prioritizes performance in either the passband or stopband.

3 Part 2

3.1 Reading the audio signal and Upsampling

The audio signal was read using Fs=22k where Fs is the sampling frequency based on the Nyquist theorem as it satisfies the following equation

$$f_s \geq 2f_{\max}$$

Where:
- f_s: Sampling frequency
- f_{max}: Maximum frequency component of the signal

since we are using upsampling factor (L) of 2, we should use double of the original frequency we used in sampling the audio as the new sampling frequency:-

$$f_{snew} = 2f_s$$

3.2 Frequency spectrum before and after upsampling

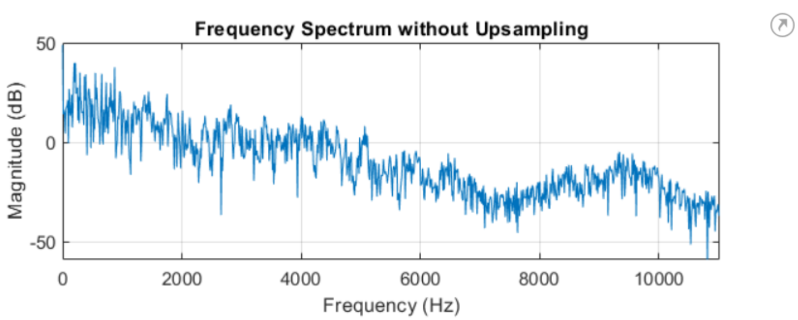


Figure 1: frequency spectrum(magnitude) before upsampling

The figure shows that the audio signal frequency components extends from 0k up to 11k which is half the sampling frequency used

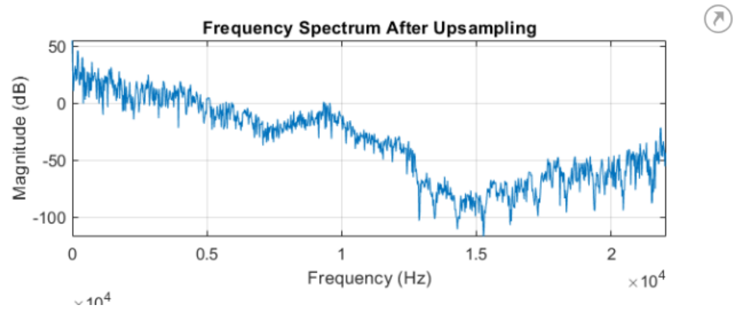


Figure 2: frequency spectrum(magnitude) after upsampling

analyzing the frequency after upsampling at factor L=2,doubled the frequency spectrum occupied by the signal that's why the nyquist freq. is increased to double hence we used double the sampling frequency later to hear the audio However, the signal orginal spectrum that was maxed at 11k remains unchanged, but the freq. spectrum now have increased room in the frequency domain due to the higher sampling rate.(fsnew)
observation after hearing the audio upsampled:the Upsampling process increased the sampling frequency that should be used with no improvement to the audio quality as no new information was added to the signal.

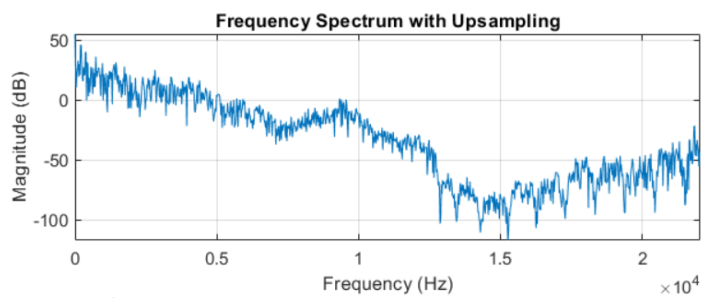


Figure 3: frequency spectrum(phase) after upsampling

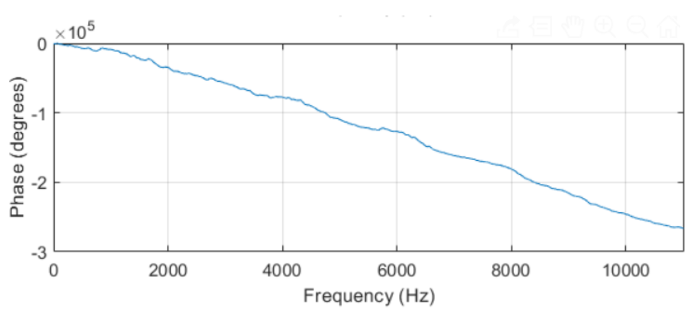


Figure 4: frequency spectrum(phase) before upsampling

3.3 adding interference

as seen from the spectrum frequency the audio signal significant freq components lies under 10k according to the magnitude, after that there is minimal energy and zero energy at higher frequencues that are beyond the nyquist frequence,since the interference freq should be less than the nyquist and still doesn't overlap with useful signal frequencies, we choose intereference freq.to be 15k

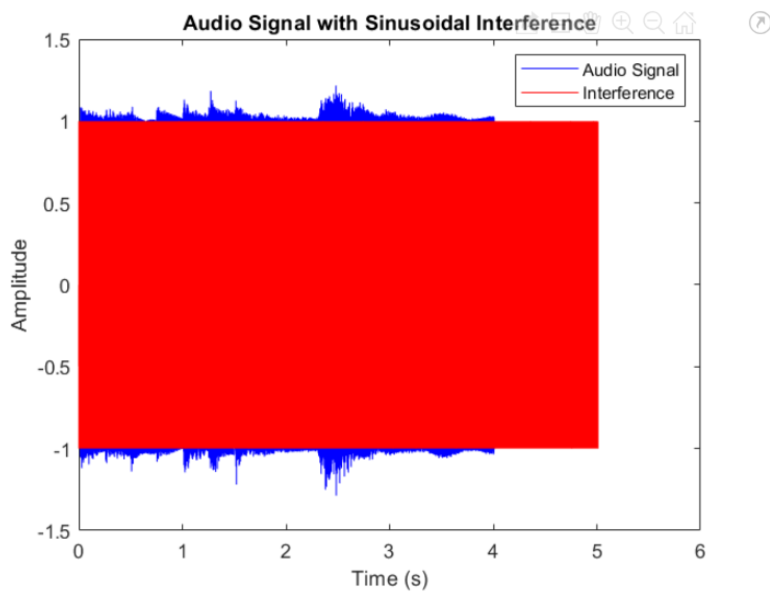


Figure 5: Audio signal with interference(Amplitude Vs time)

3.4 frequency spectrum after adding the interference

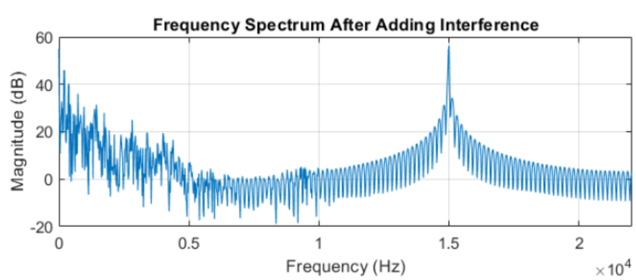


Figure 6: frequency spectrum(Magnitude) after adding interference with f=15k

As shown in Fig.6, the interference appears as a spike in the freq. domain at 15k(interference frequency chosen),since the interference frequency was far from the major freq.components of the original audio,the original spectrum ranging from(0HZ to 11k HZ) remains almost intact after the interference creates it's distinct peak at it's frequency

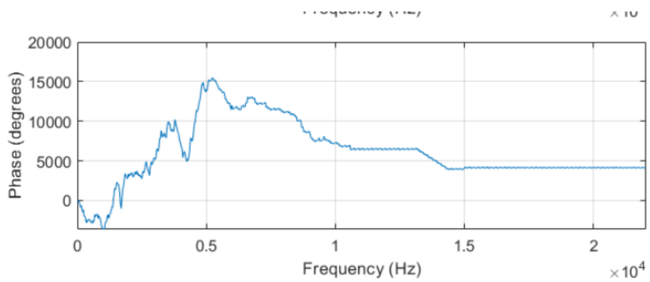


Figure 7: frequency spectrum(phase) after adding sinusoidal interference with f=15k

3.5 playing audio after adding the interference

The audio was again played with the new sampling frequency(Fsnew)

Observations:-
The original audio can still be listened and distinguished but the added interference is perceived as a noise playing in the background.
the higher the scaled amplitude of the interference signal, the more the original audio is hard to be heard as the interference noise is more louder (in terms of the magnitude)

3.6 Filter Design

In this step, we designed a digital FIR filter using the filter design tool to attenuate the interference signal while preserving the useful audio signal as much as possible. The filter design was carried out iteratively, adjusting the parameters to find a suitable compromise between performance and complexity. We used the Kaiser window method to design the FIR filter.

The process included the following iterations: - First, a low-pass FIR filter was designed with a cutoff frequency of 15 kHz, targeting the removal of interference while retaining the audio signal. The filter length was chosen to be 51 taps, which provided a reasonable balance between computational complexity and filter performance. - Next, various window functions (Kaiser, Chebyshev, Rectangular, Blackman) were tested, with the Kaiser window yielding the best performance in terms of reducing the interference while maintaining the original audio content.

The final filter was designed with the following parameters: - Filter length $N = 51$ - Cutoff frequency $f_c = 15\,\text{kHz}$ - Window type: Kaiser window with $\beta = 8$

The filter coefficients were computed and normalized to ensure the filter's gain was unity.

3.7 Filter the Audio Signal with Interference

After designing the FIR filter, we applied it to the audio signal with interference. The filter coefficients obtained from the previous step were used to filter the noisy audio signal.

```
% Filtering the audio signal with interference
audio_filtered = filter(h_filter, 1, audio_with_interference);
```

The filtered audio signal is now ready for analysis.

3.8 Plot the Frequency Spectrum After Filtering

After filtering the audio signal, we analyzed the frequency spectrum of the filtered audio. The frequency spectrum was plotted to verify that the interference at 15 kHz was attenuated while retaining the useful audio frequencies.

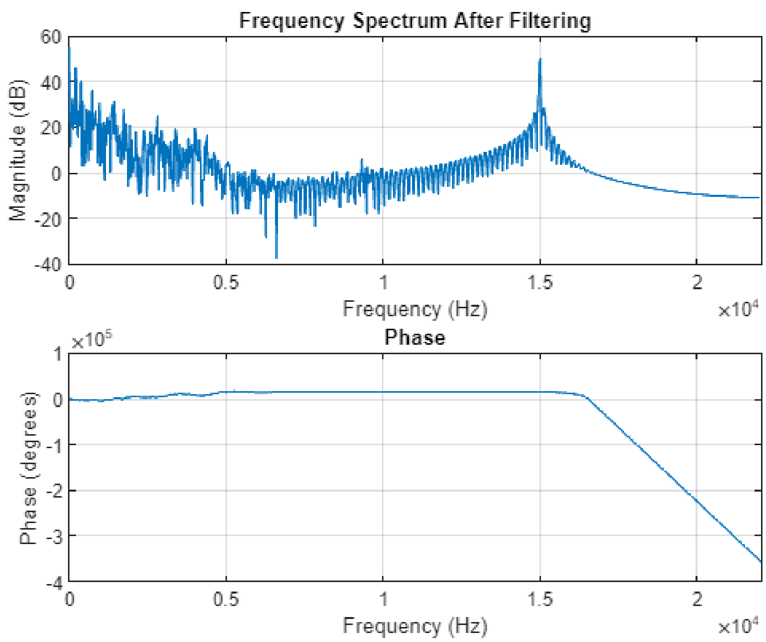


Figure 8: Frequency Spectrum (Magnitude) After Filtering

As seen from the frequency spectrum, the interference at 15 kHz has been significantly attenuated, and the useful frequencies of the audio signal remain largely unchanged.

3.9 Listen to the Audio Signal After Filtering

Finally, we listened to the filtered audio signal to assess the effectiveness of the filter in removing the interference. The filtered audio signal was played using the following command:

```
sound(audio_filtered, fs_new);
```

Upon listening to the audio, we observed that the interference noise at 15 kHz was effectively removed, and the original audio signal was clear and distinguishable. The quality of the audio was preserved, with no significant distortion introduced by the filtering process.