UNIVERSITY OF SCIENCE AND TECHNOLOGY
Digital Signal Processing (CIE 442)

Digital Filtering of an Audio Signal

Tasneem Muhammad 202101031
Nada Nabil 202101220
Aya Sherif 202100642
Rghda Salah 202101510

Dr. Mahmoud Abdelaziz
Academic Year
Fall 2024

Part 1: Digital FIR Filter Design Tool

1. Windowed Sinc FIR Filter Design Method

The Windowed Sinc FIR filter design method is a widely used approach to create linear-phase digital filters. This method generates an ideal low-pass filter using a sinc function and then applies a window to control side lobes and improve stopband attenuation. This report documents the implementation of this filter design method using MATLAB, including four different window functions: Rectangular, Blackman, Chebyshev, and Kaiser. The frequency response is analyzed in both linear and logarithmic scales.

Implementation Details:

1. Filter Length Selection

- **User Input:** The user specifies the filter length (‘N’). Odd lengths are recommended for symmetric filters.
- **Reason:** Symmetry ensures a linear phase, which is desirable in FIR filters.

2. Cutoff Frequency Input

- **User Input:** The normalized cutoff frequency (‘fc’) is specified in the range (0, 0.5).
- **Mapping:** This corresponds to the desired passband edge in normalized frequency.

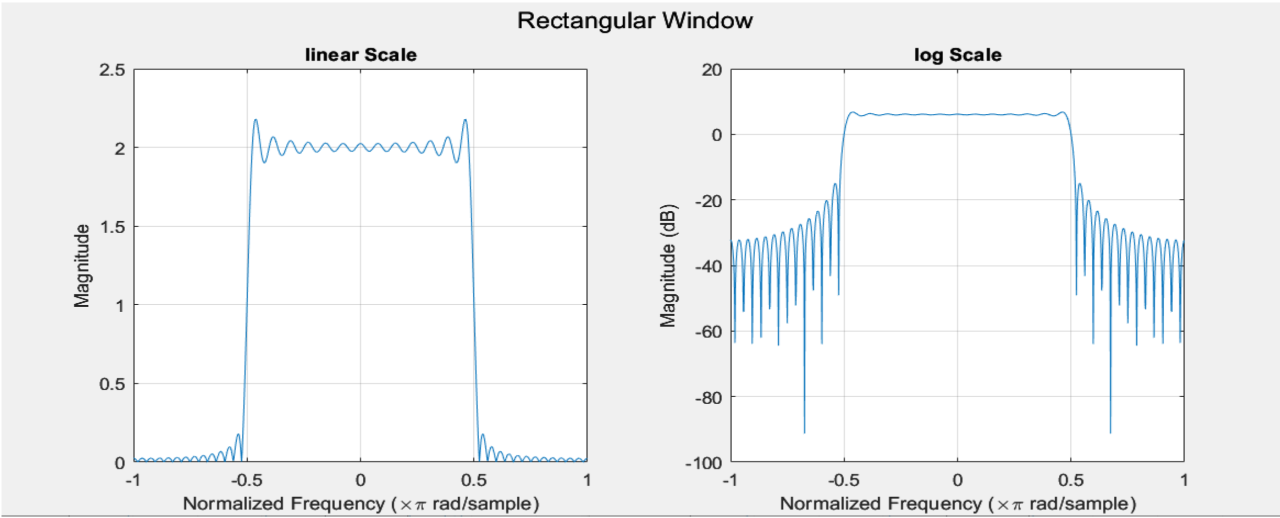
3. Window Functions

- **Rectangular Window:** Simple, but exhibits significant side lobes in the frequency response.
- **Blackman Window:** Provides better side lobe attenuation at the cost of a wider transition band.
- **Chebyshev Window:** Allows control over the side lobe levels via the ripple parameter.
- **Kaiser Window:** Offers flexibility with the beta parameter, balancing attenuation and transition width.

Figures and Comments:

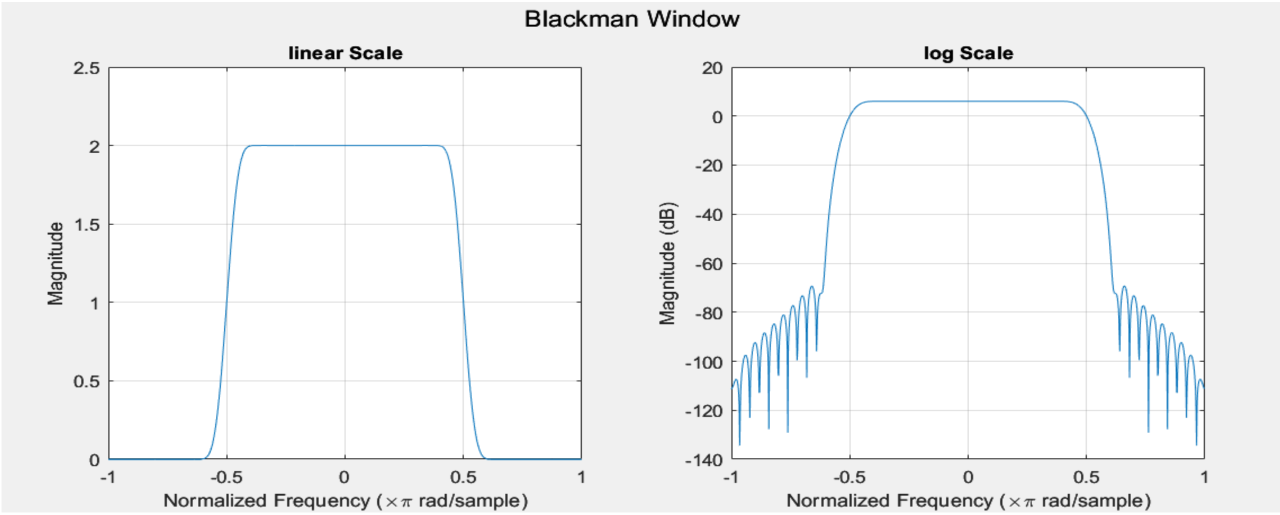
When the length of the filter impulse = 101

1. **Rectangular Window**



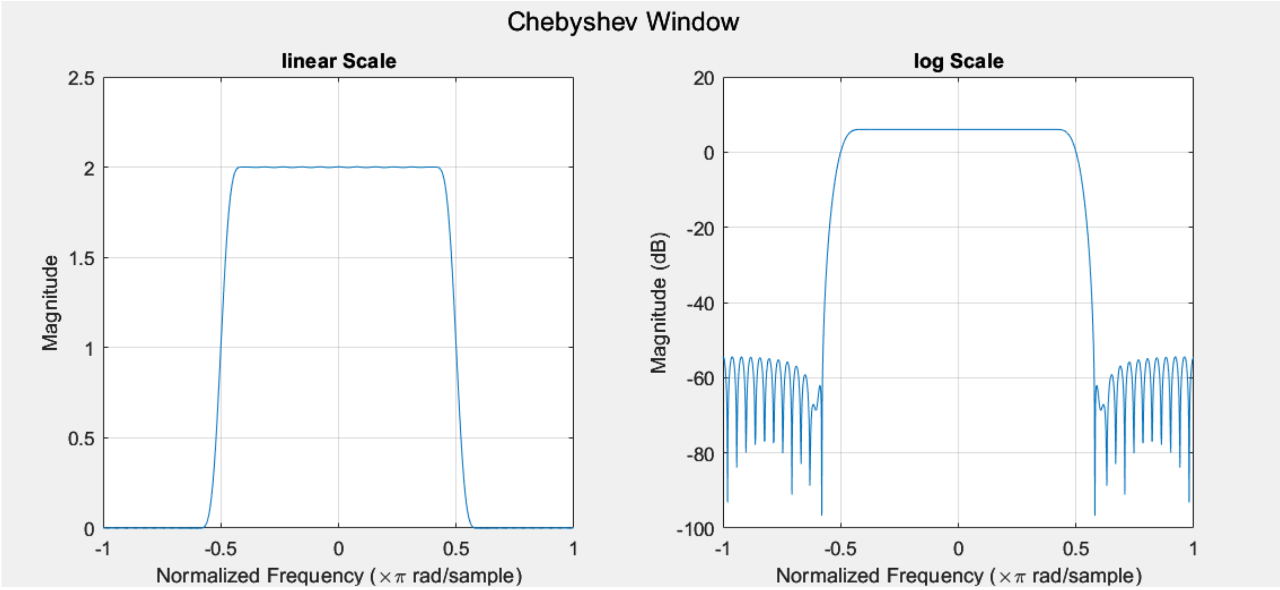
- **Observation:**
  - The rectangular window produces significant side lobes, leading to poor stopband attenuation.
  - **Pass Region**: Sharpest passband among windows but suffers from ripples (Gibbs phenomenon).
  - The figure shows the magnitude response with noticeable oscillations in the stopband.

2. Blackman Window



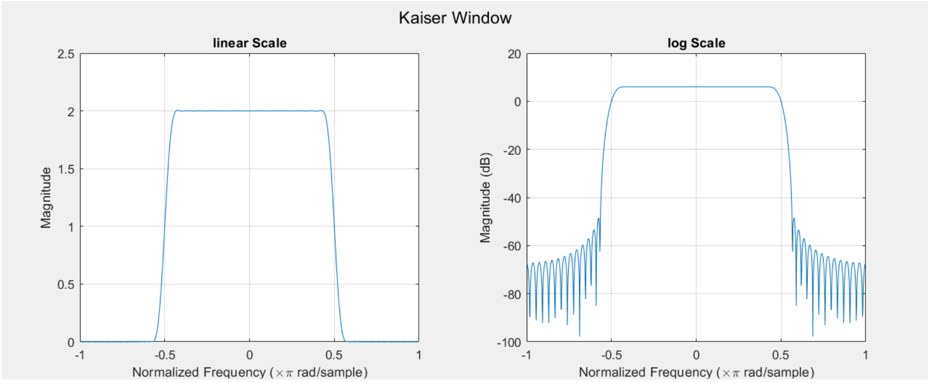
- **Observation:**
  - A tapering window that reduces sidelobe levels using additional terms (cosine terms) in the window function.
  - The Blackman window improves stopband attenuation but results in a wider transition band.
  - **Pass Region**: Slightly broader compared to the rectangular window, with reduced ripples.
  - **Transition Region**: Wider than the rectangular window, but with much smaller sidelobes.
  - **Stop Region**: Excellent stopband attenuation,
  - The magnitude response displays reduced side lobes compared to the rectangular window.

3. Chebyshev Window



- **Observation:**
  - The Chebyshev window allows control of side lobe levels. In this case, a ripple of 50 dB was used.
  - **Pass Region**: Passband ripples can be controlled (not completely ripple-free).
  - **Transition Region**: Narrower than Blackman, dependent on the design parameter.
  - The magnitude response shows narrower transition bands with controlled side lobes.
  - **Stop Region**: Good stopband attenuation,

4. Kaiser Window



- **Observation:**
  - The Kaiser window provides a balance between stopband attenuation and transition width. A beta value of 5 was used.
  - **Pass Region**: Highly customizable; ripples in the passband can be minimized by adjusting β\betaβ
  - **Transition Region**: Variable; can be adjusted to achieve a balance between sharpness and sidelobe attenuation.
  - The magnitude response demonstrates good overall performance.
  - Computationally more complex than the simpler windows.

Window	Pass Region	Transition Region	Stop Region	Trade-offs
Rectangular	Sharp but with ripples	Wide	Poor attenuation (-21 dB)	Simple, sharp passband; high spectral leakage.
Blackman	Smooth with minimal ripples	Wider	Excellent attenuation (-74 dB)	Low leakage, wider transition; blurs close frequencies.
Chebyshev	Controlled ripples	Moderate	Good attenuation (equiripple)	Narrow transition, but uniform stopband ripples may not suit all applications.
Kaiser	Adjustable	Adjustable (based on β)	Good attenuation (varies with β)	Highly flexible; balance between sharpness and attenuation.

2. Least Squares (LS) and Weighted LS (WLS) FIR Filter Design Methods

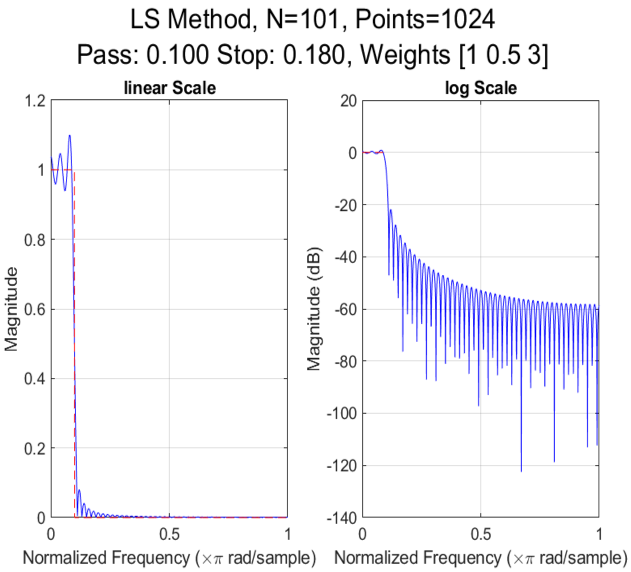
This section explores the design of FIR filters using the Least Squares (LS) and Weighted Least Squares (WLS) methods. These techniques optimize the filter coefficients to approximate a desired frequency response while minimizing the error across specified frequency bands. The LS method assigns equal importance to all frequency points, while the WLS method allows for prioritizing specific bands using user-defined weights.

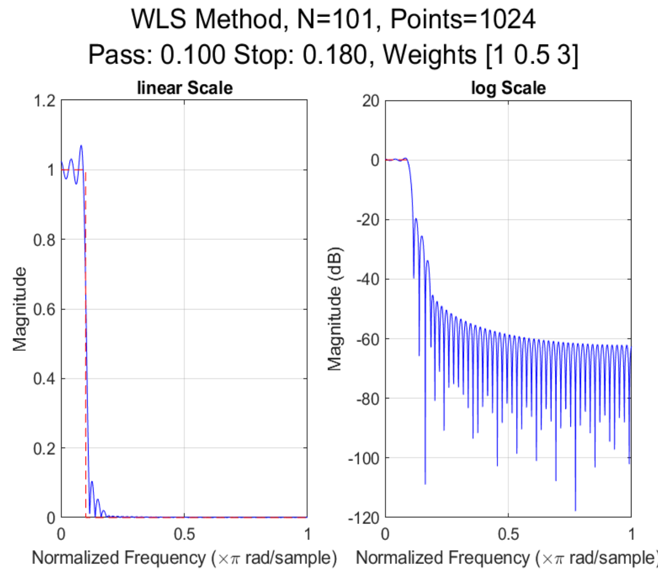
Implementation Details:

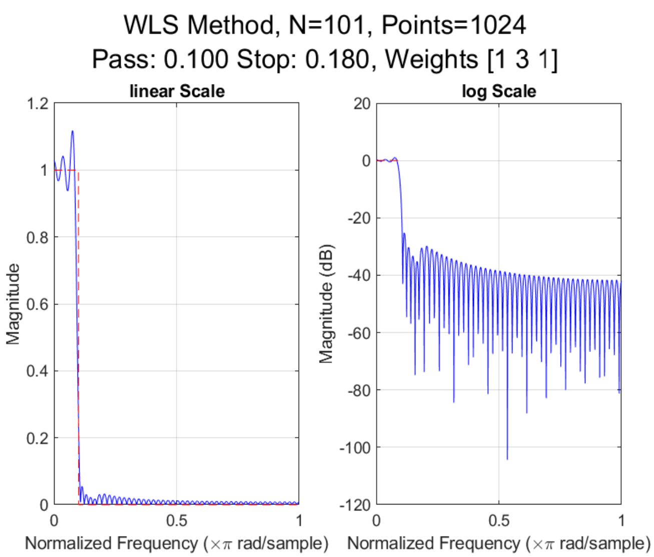
1. **User Inputs**:
    a. Filter length (N): Determines the number of coefficients in the FIR filter.
    b. Number of frequency response points (M): Specifies the granularity of the frequency grid.
    c. Desired response (H_d): Defines the target frequency response, e.g., a low-pass filter.
    d. Weights (WLS only): Specifies the importance of passband, transition band, and stopband.
    e. Plot Scale: Allows visualization of the magnitude response in linear or logarithmic scale.
2. **Least Squares Method**:
    a. The LS method minimizes the squared error between the desired and actual frequency responses over the entire frequency range.
3. **Weighted Least Squares Method**:
    a. The WLS method extends LS by incorporating a weighting matrix to emphasize certain frequency bands.

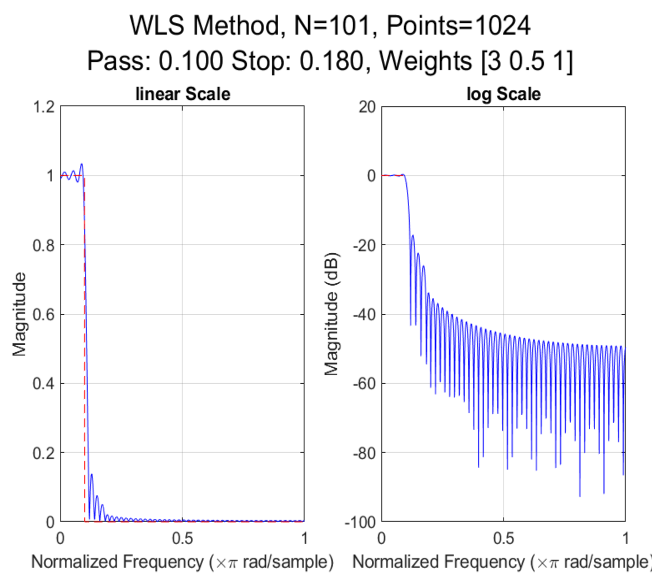
Figures and Comments:

LS VS WLS





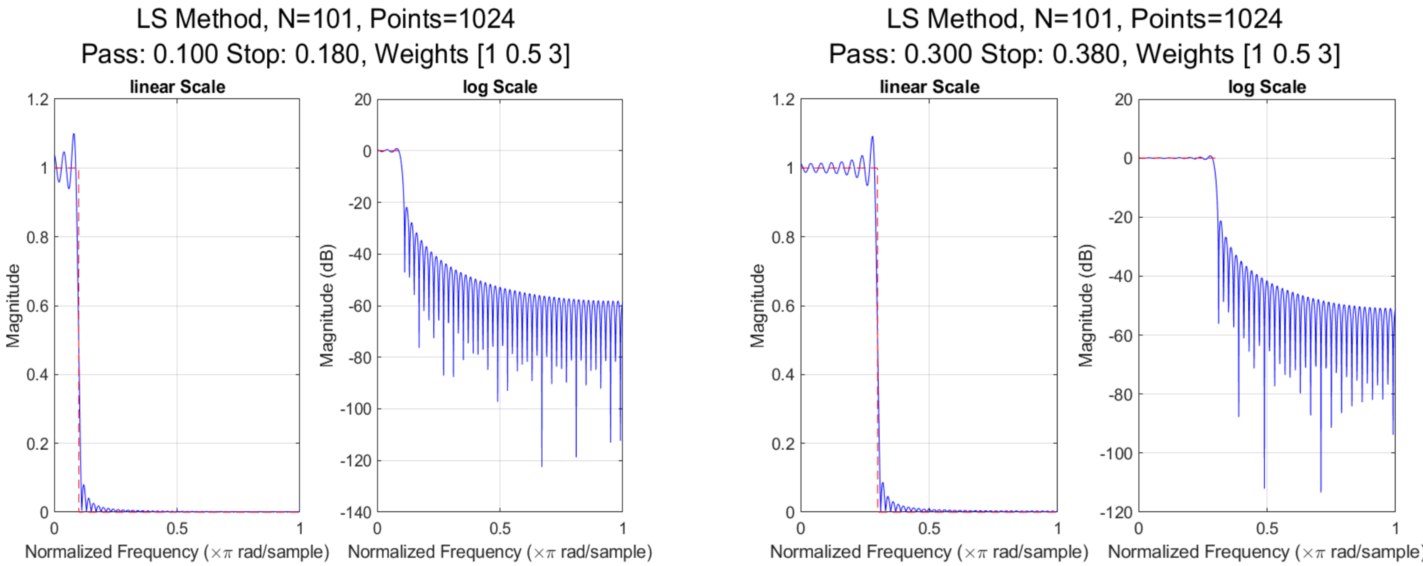




Comment:

The figures illustrate the performance differences between Least Squares (LS) and Weighted Least Squares (WLS) methods under three weighting scenarios: prioritizing the pass region, transition region, and stop region. In the pass region, WLS demonstrates superior flatness and reduced ripples compared to LS, though at the expense of stopband attenuation and transition sharpness. For the transition region, WLS achieves a sharper roll-off, significantly narrowing the transition width, but introduces higher ripple levels in both the pass and stop regions. When the stop region is prioritized, WLS achieves excellent attenuation and sidelobe suppression, outperforming LS, but sacrifices passband flatness and transition sharpness. These comparisons highlight the trade-offs inherent in WLS design

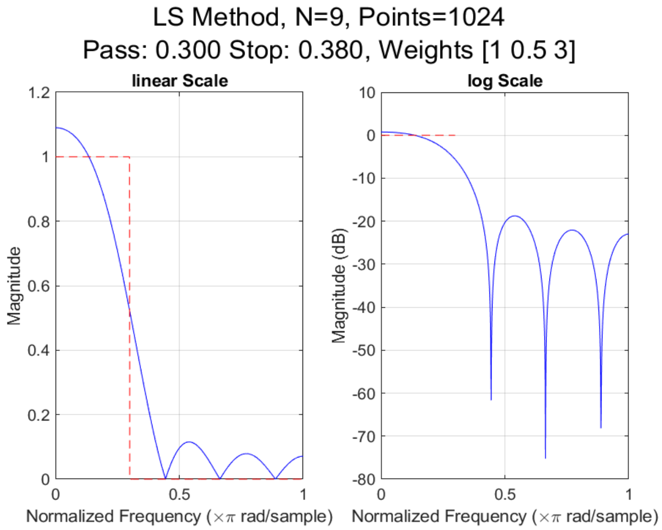
Passpoint VS Stoppoint

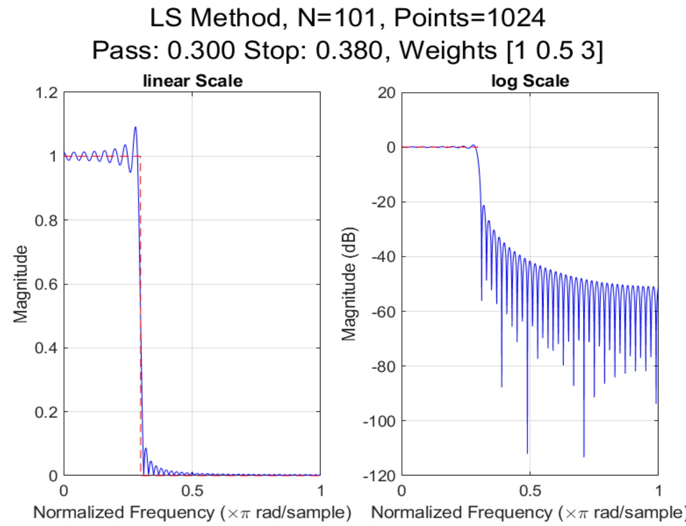


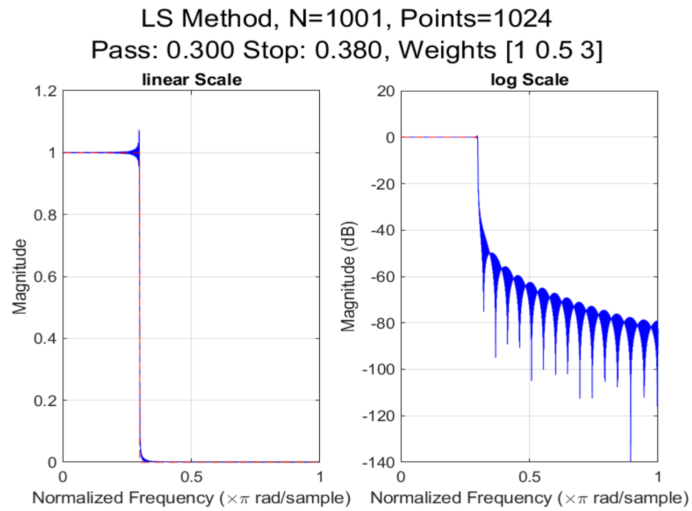
Comment

Here, we adjust the cutoff frequency to control the precise point at which we filter the signal. This parameter is crucial for determining the boundary between the desired signal and the noise, ensuring that we effectively attenuate noise while preserving the relevant components of the signal. By selecting the appropriate cutoff frequency, we can fine-tune the filter to target specific frequency ranges where noise is present, optimizing the overall performance of the system.

Filter lengths



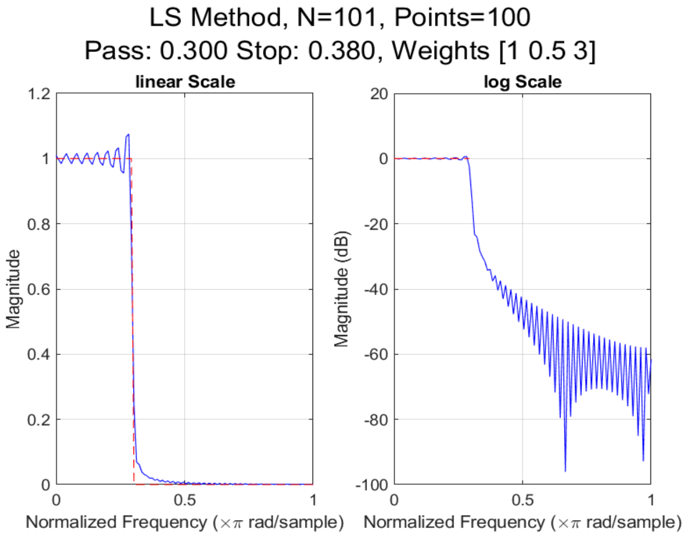


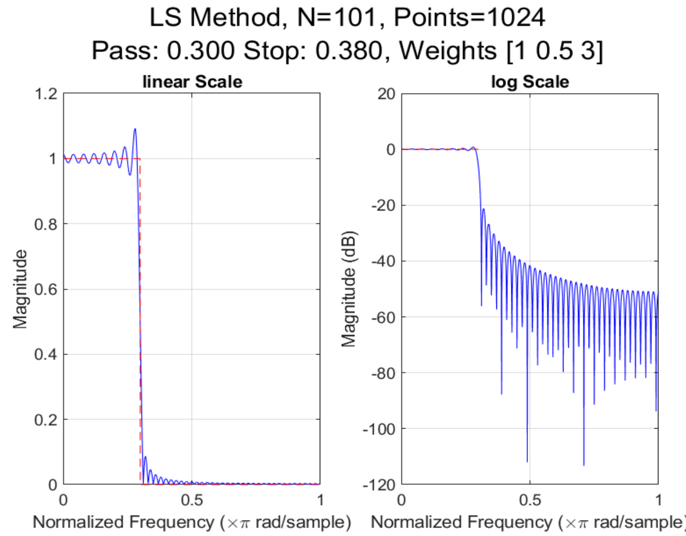


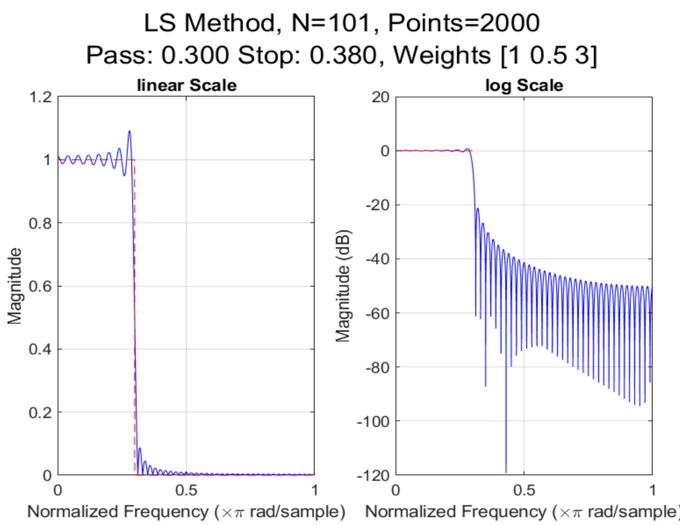
Comment:

The filter length plays a crucial role in determining the performance of a designed filter. A small filter length results in a filter with a coarse approximation of the desired frequency response, leading to a less accurate filter, a wider transition region, and potentially higher ripple in the passband and stopband. A medium filter length strikes a balance between accuracy and computational efficiency, providing a better approximation of the desired response with reduced ripples and a more controlled transition region. However, it may still have some limitations in stopband attenuation or transition sharpness. A large filter length improves the filter's performance significantly, resulting in a finer approximation of the ideal filter response, with a narrower transition region and better stopband attenuation. However, it increases computational complexity and processing time, which may not be suitable for real-time or resource-constrained applications. Therefore, the filter length must be chosen based on the trade-off between performance and computational constraints.

Number of frequency response points







Comment:

The number of frequency response points in filter design significantly affects the accuracy and resolution of the filter's frequency characteristics. A small value results in a coarse representation of the frequency response, which can lead to an inaccurate filter design with noticeable deviations from the desired response, especially in the transition and stop regions. A medium value provides a better balance, allowing for a more accurate approximation of the filter's frequency response with finer details in the passband, transition, and stopband regions. This improves the filter's overall performance without excessively increasing computational complexity. However, a large value leads to a highly detailed frequency response, offering very precise control over the filter's characteristics, especially in the transition region. While this enhances the filter's performance, it increases computational burden and may introduce longer design times and higher memory usage, which can be impractical in real-time or resource-constrained applications. Thus, the number of frequency response points should be chosen carefully based on the desired filter precision and available computational resources.

Discusion on Part 1:

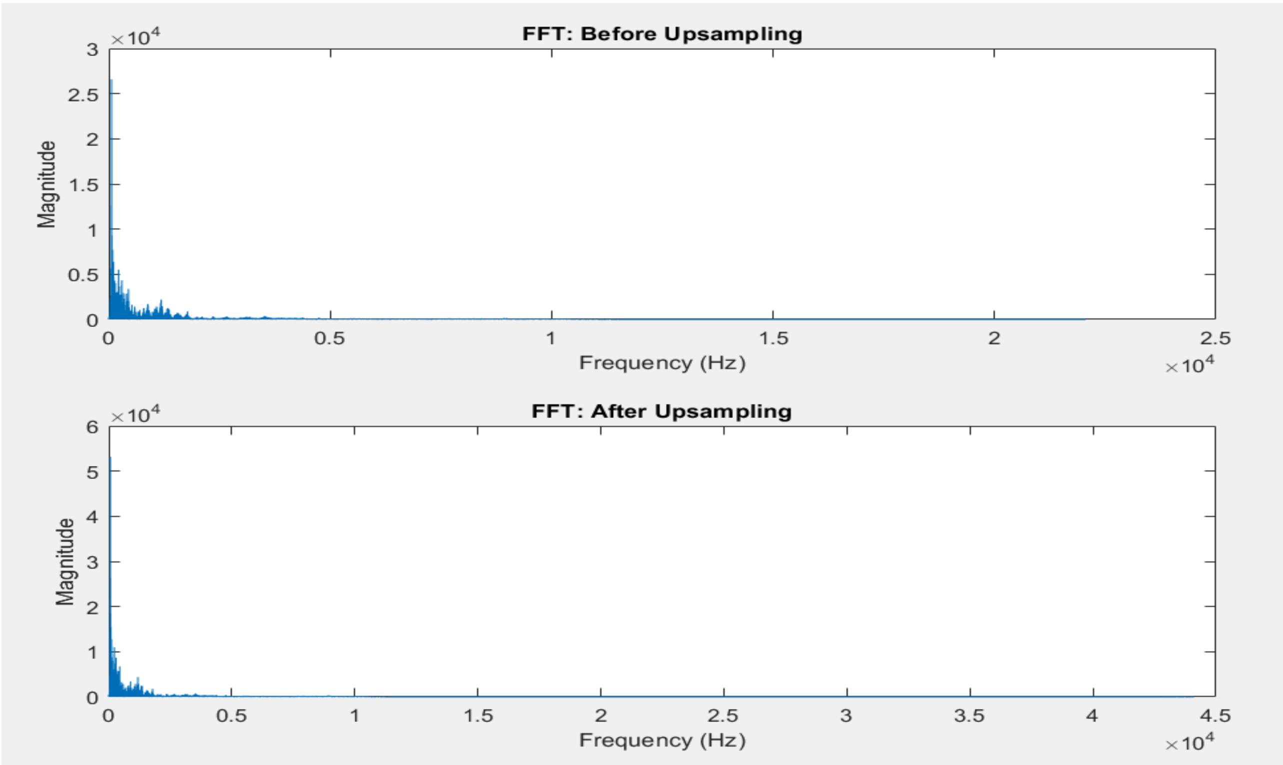
- Increasing the filter length (N) improves performance, resulting in sharper transitions and reduced ripples.
- Adjusting the weights in the WLS method allows customization of the filter's characteristics:
  - Higher stopband weights enhance attenuation at the expense of passband accuracy.
  - Higher passband weights improve fidelity in the passband but reduce stopband attenuation.
- The choice of linear or logarithmic scale significantly impacts the visualization:
  - Linear scale highlights the passband accuracy.
  - Logarithmic scale emphasizes stopband attenuation and dynamic range.

Part 2: Practical Filtering Problem

1. Audio File Processing

- Upsample the signal by 2 using `resample`.
- Plot the frequency spectrum before and after upsampling.

Figures and Comments:



- **Comments:**

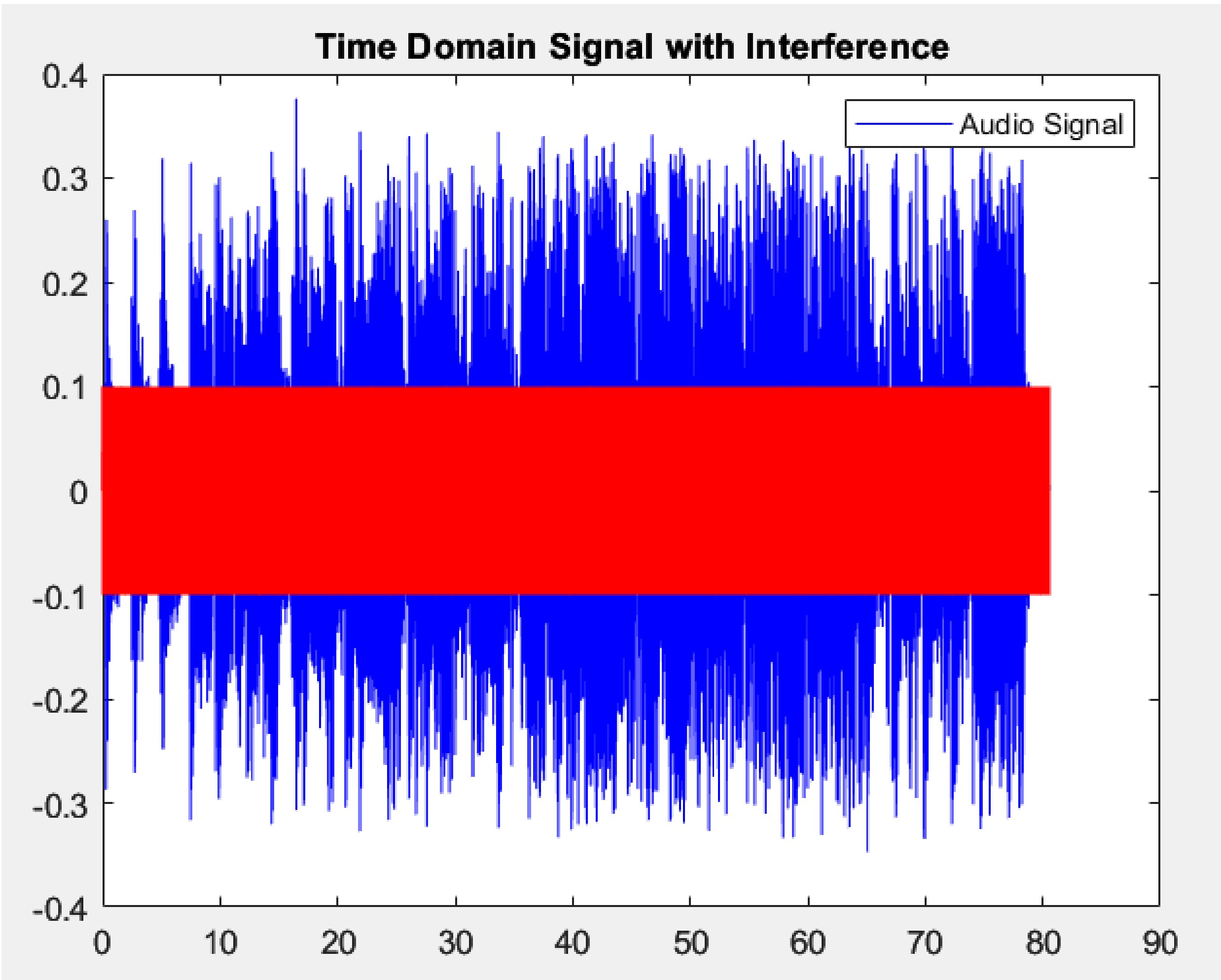
The plot of the frequency spectrum before and after upsampling reveals the impact of increasing the sample rate on the signal's frequency content. Before upsampling, the spectrum typically displays a limited frequency range corresponding to the original sampling rate, with the highest frequency component (Nyquist frequency) determined by half the sampling rate. After upsampling, the spectrum is expanded, showing more frequency points in the range up to the new Nyquist frequency. However, the original signal's frequency content remains unchanged—what upsampling does is insert additional zeroes in the frequency domain, effectively increasing the resolution of the frequency representation. This often results in the appearance of spectral replicas or "images" (aliasing) unless appropriate filtering is applied to remove unwanted components. To prevent these unwanted effects, a low-pass filter is typically applied after upsampling to remove the high-frequency images and preserve the integrity of the original signal's content.

The Upsampling signal has more samples, and the frequency spectrum shows an increase in the resolution of the frequency bins

2. Adding Sinusoidal Interference

Figures and Comments:

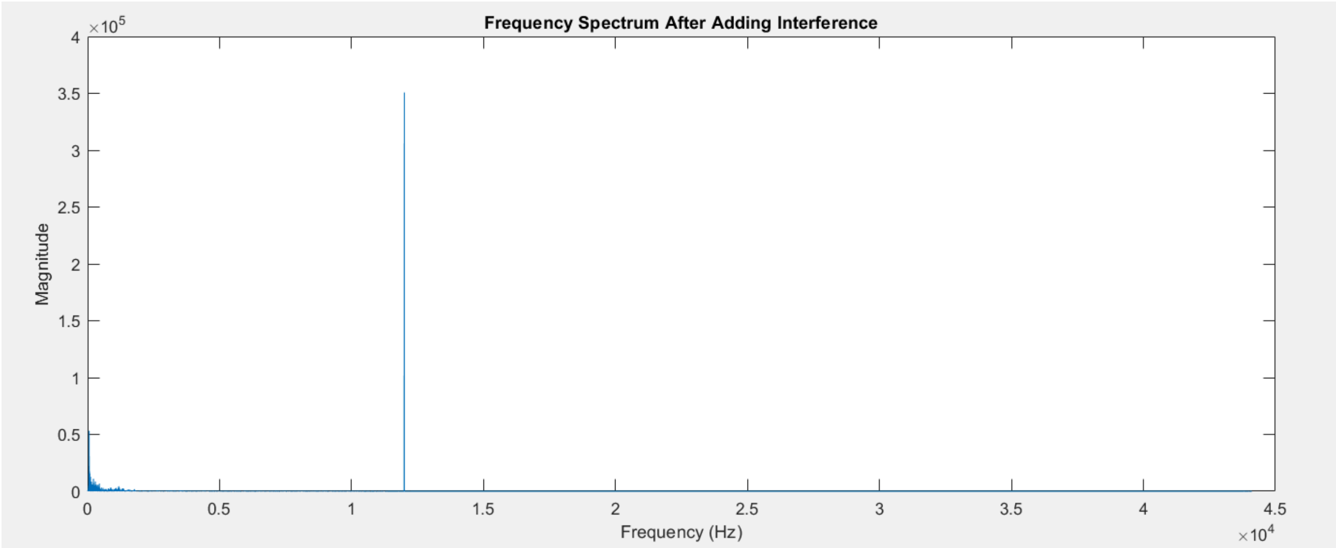
Plot the time-domain signals of audio and interference:



Comment:

This figure illustrates a time-domain signal, represented by a blue waveform labeled "Audio Signal," with a significant interference or masking effect indicated by a bold red rectangular region.

Plot the frequency spectrum of the combined signal: `interferenceFreq = 12000`



- **Comments:**

This figure represents the frequency spectrum of a combined signal after the introduction of interference at a specific frequency, identified as interferenceFreq = 12000 Hz. The spectrum shows a prominent spike at 12000 Hz, indicating the strong presence of the interference signal in the frequency domain. This peak stands out significantly against the rest of the spectrum, where the magnitudes are comparatively minimal. The plot effectively highlights how the interference dominates the signal, emphasizing its high energy at this specific frequency. The visualization is useful for identifying and potentially mitigating such interference in signal processing applications.

3. Listening Tests

- Play the audio signal with interference (using the new sampling rate).

Comments:

The interference introduces a prominent tonal artifact into the audio, which can be perceived as an unwanted high-pitched tone or noise, depending on its amplitude relative to the original signal. Such interference can significantly degrade the audio quality by masking or distorting the original content, making it less intelligible or unpleasant to listen to. Finally we want to thereby restore the audio to its intended quality.

---

4. FIR Filter Design to Remove Interference

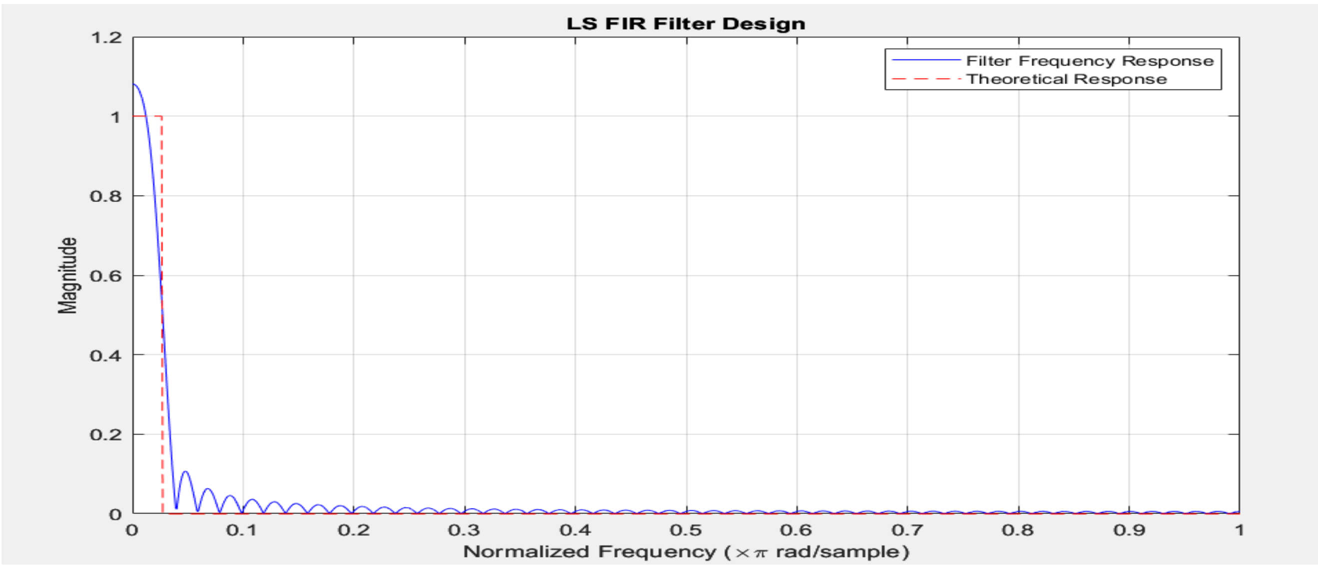
Figures and Comments:

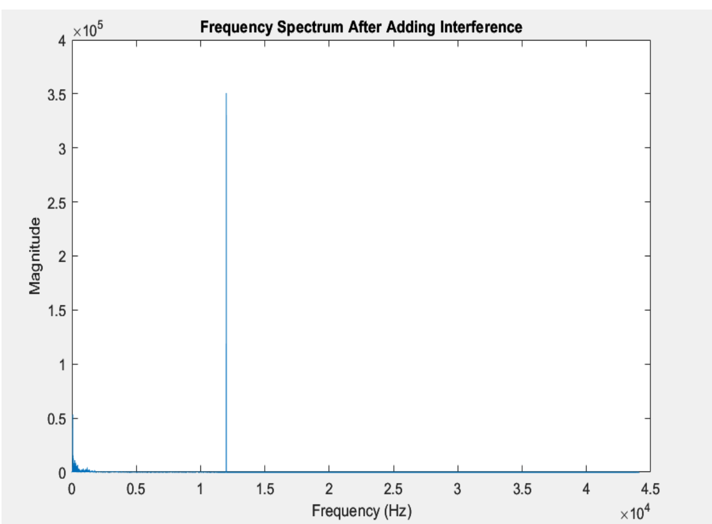
Design a filter using the tool from Part 1 to attenuate interference.

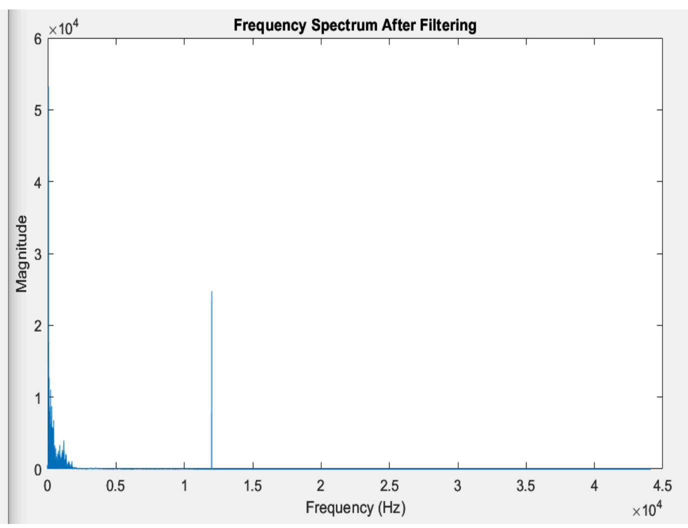
We tried different combinations of the parameter to conduct the best coefficients to design our optimal filter,

The filter is used to attenuate the high-frequency noise while preserving the desired audio content.

```
Designing FIR filter...
Enter the filter length (odd number preferred): 101
Enter the number of frequency response points: 1024
Choose the design method (LS/WLS): LS
Choose the scale for magnitude response (linear/log): linear
Enter the normalized passband cutoff frequency (0-1): 0.027
Enter the normalized stopband cutoff frequency (0-1): 0.035
Playing filtered audio signal...
```



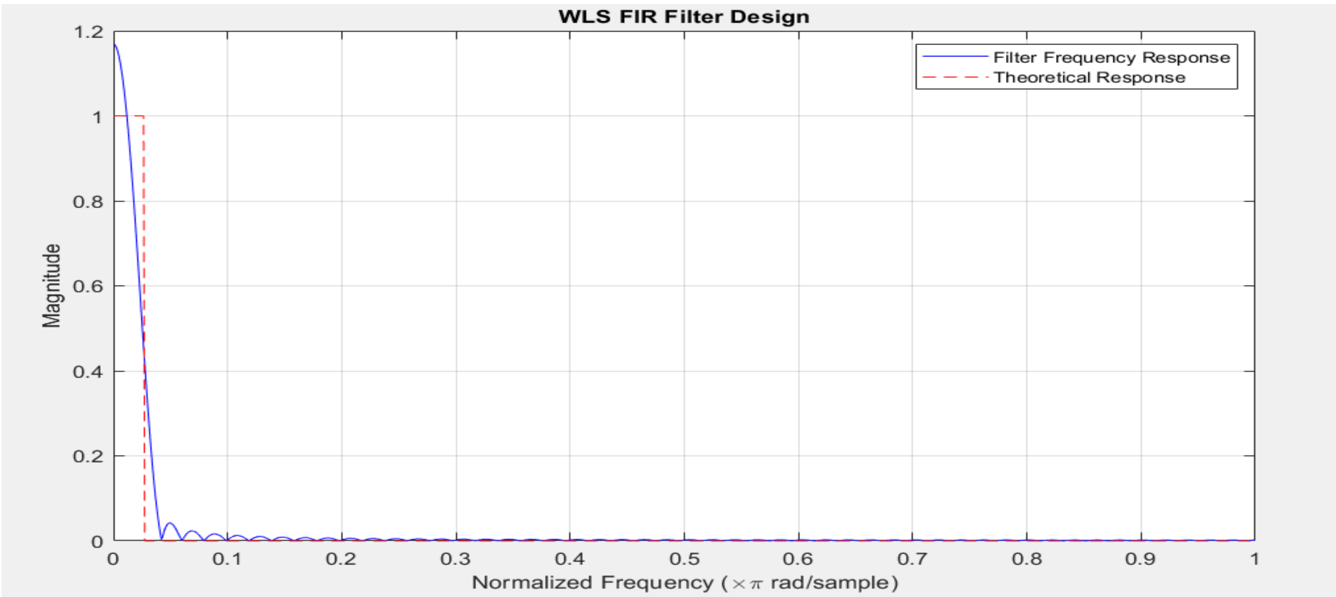


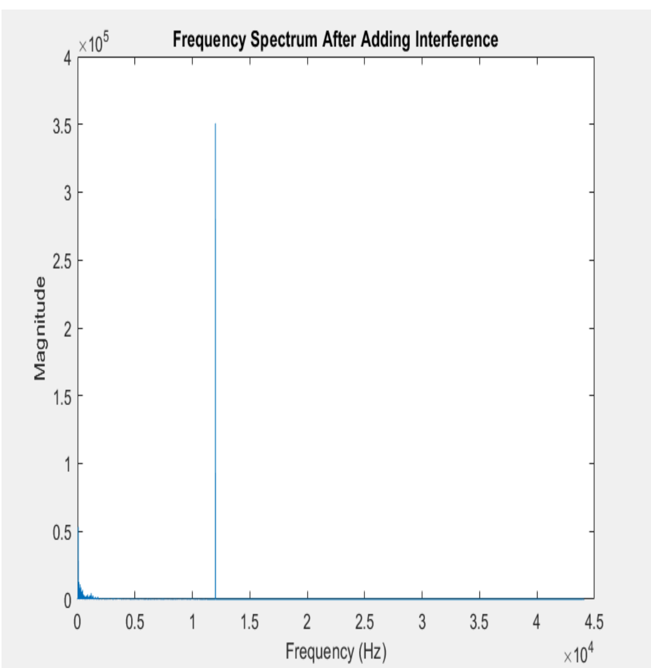


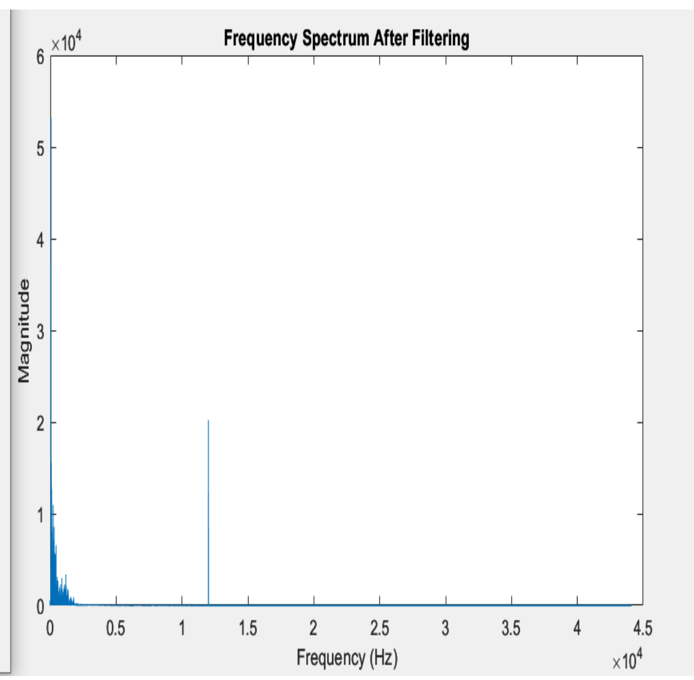
Comment:

While the filter effectively reduces the magnitude of the interference at 12 kHz (as observed in the right spectrum after filtering), it does not entirely eliminate it, leaving residual noise at the interference frequency. This incomplete suppression suggests either insufficient filter order or design limitations in targeting the interference frequency. The residual noise could still impact the audio quality, especially if the interference is prominent relative to the original signal. To improve results, further refinement of the filter, such as increasing its order

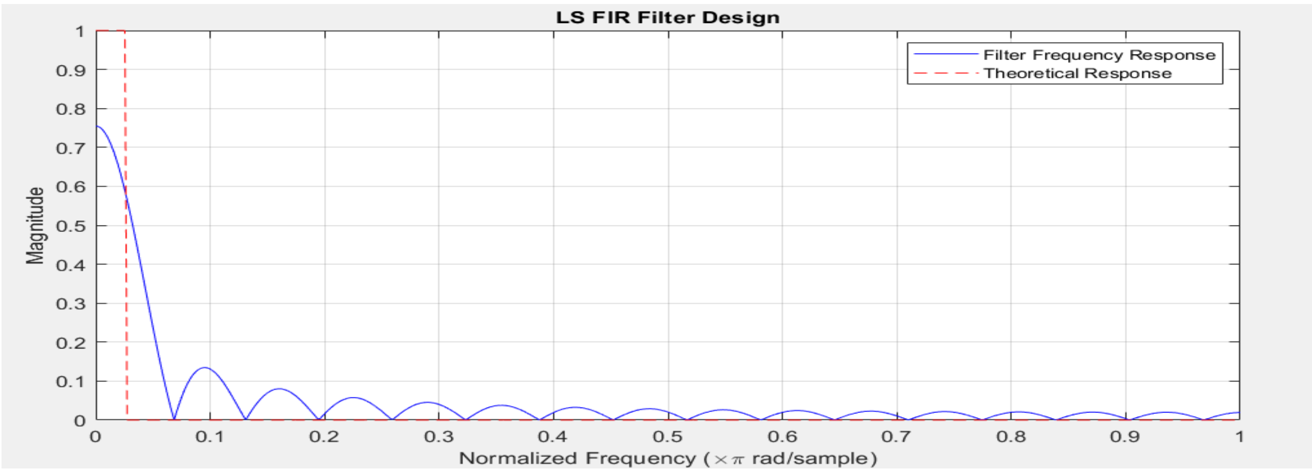
```
Enter the filter length (odd number preferred): 101
Enter the number of frequency response points: 1024
Choose the design method (LS/WLS): WLS
Choose the scale for magnitude response (linear/log): linear
Enter the normalized passband cutoff frequency (0-1): 0.027
Enter the normalized stopband cutoff frequency (0-1): 0.035
Enter the weight for the passband: 1
Enter the weight for the transition band: 0.5
Enter the weight for the stopband: 3
Playing filtered audio signal...
```

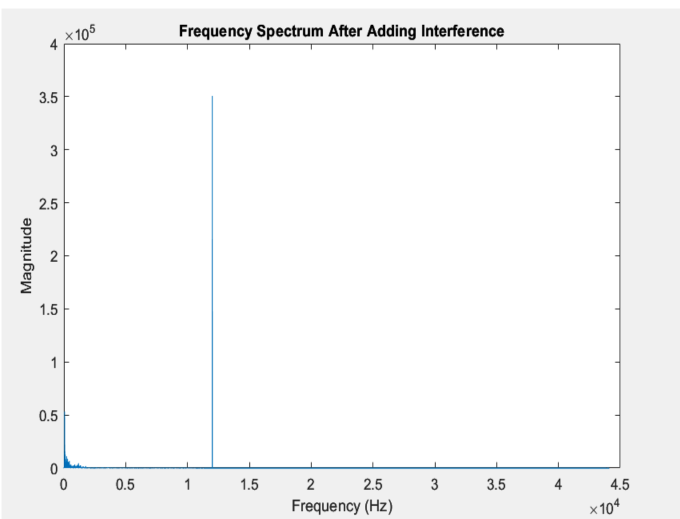


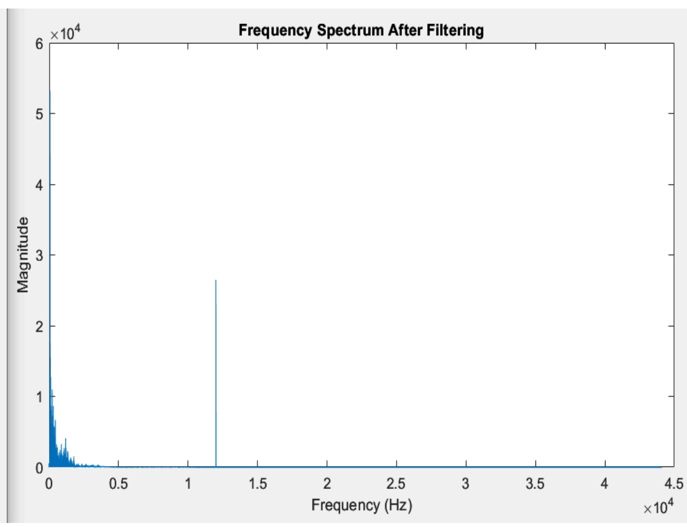


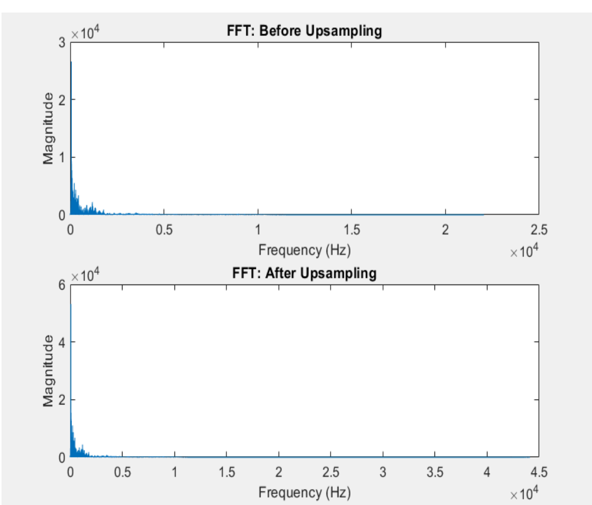


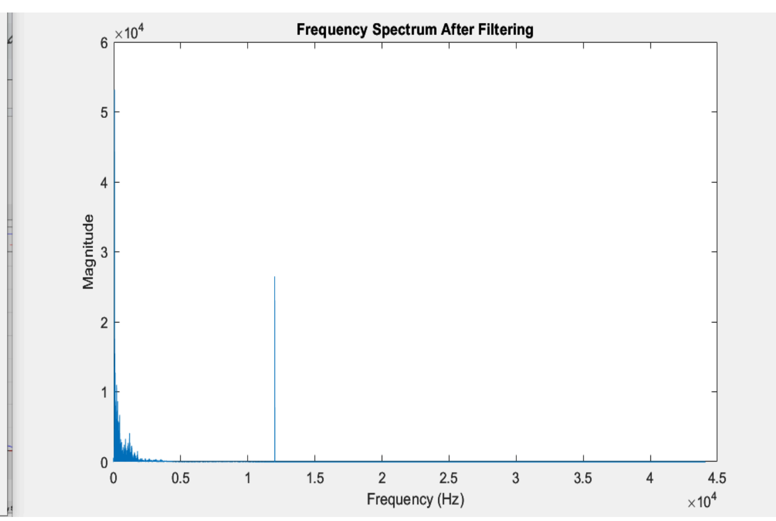
```
Playing audio signal with interference...
Designing FIR filter...
Enter the filter length (odd number preferred): 31
Enter the number of frequency response points: 512
Choose the design method (LS/WLS): LS
Choose the scale for magnitude response (linear/log): linear
Enter the normalized passband cutoff frequency (0-1): 0.027
Enter the normalized stopband cutoff frequency (0-1): 0.035
Playing filtered audio signal...
```











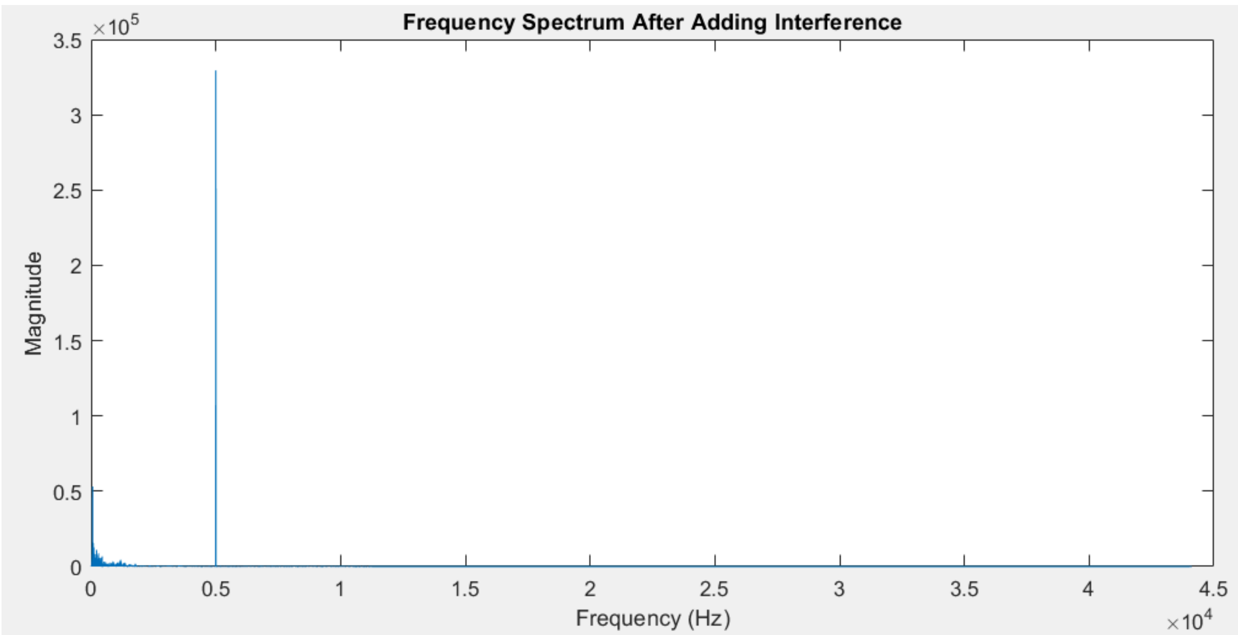
In last Figure, We tried to show that we get the original signal “upsamplinf signal” after the Filtering process

---

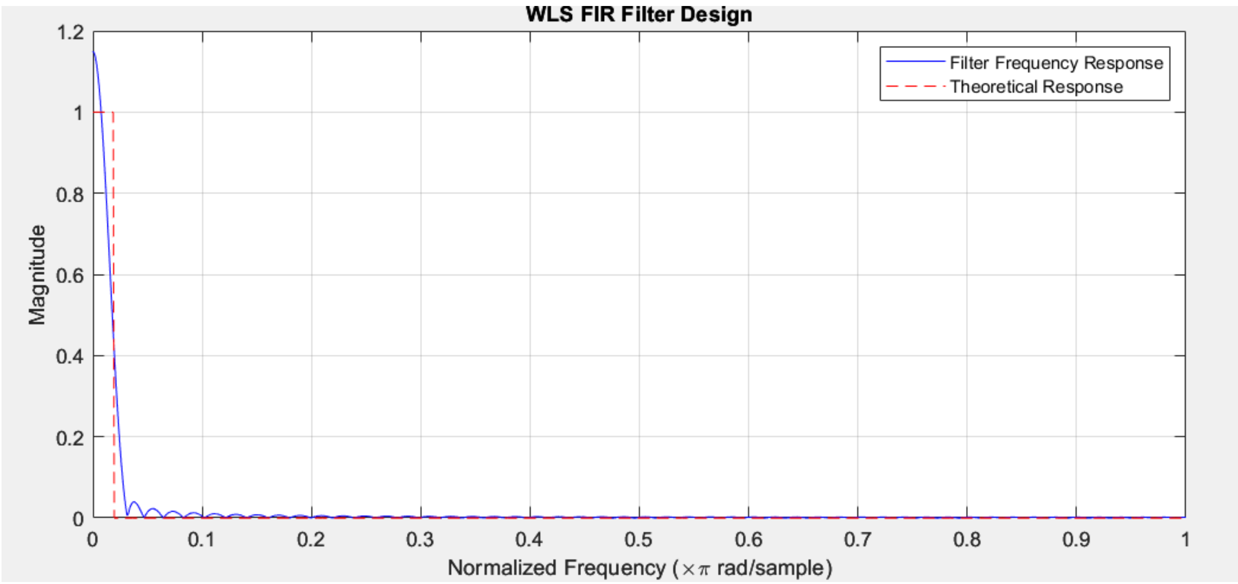
Here we will try another value for interferenceFreq = 5000 HZ

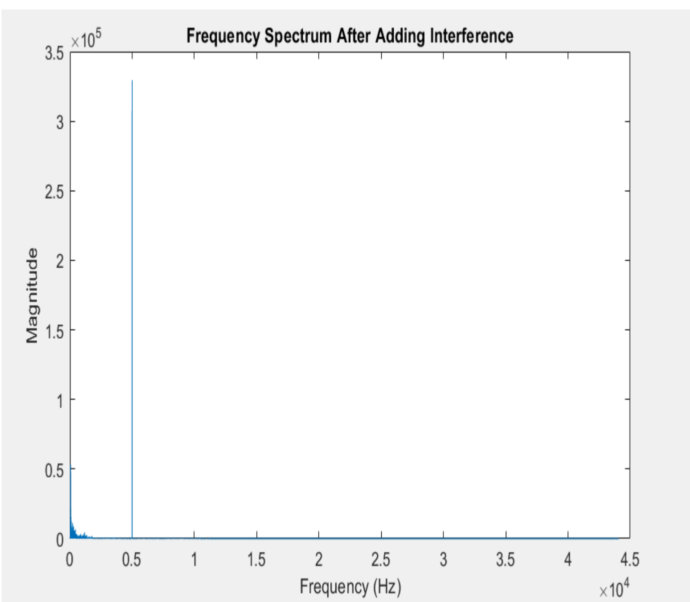
Note: that if we reduce the interferenceFreq sothat it became more close to the signal than before when we used 12000

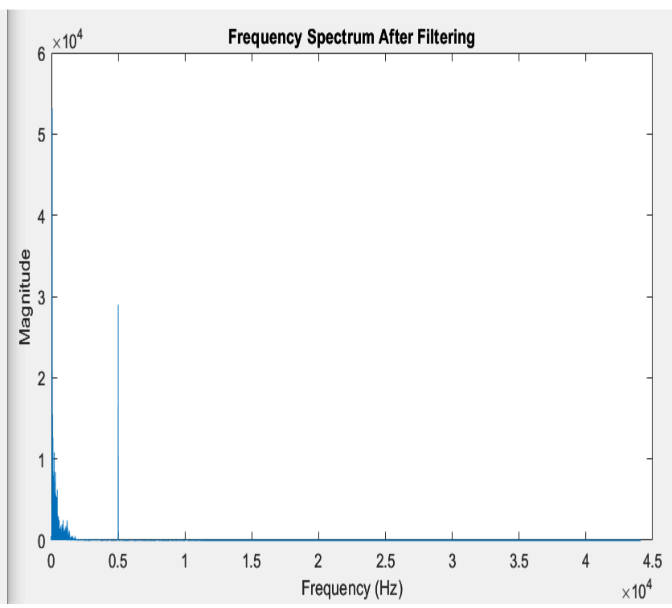
And that will may cause some distortion on the signal or we will need to reduce the cutoff frequency



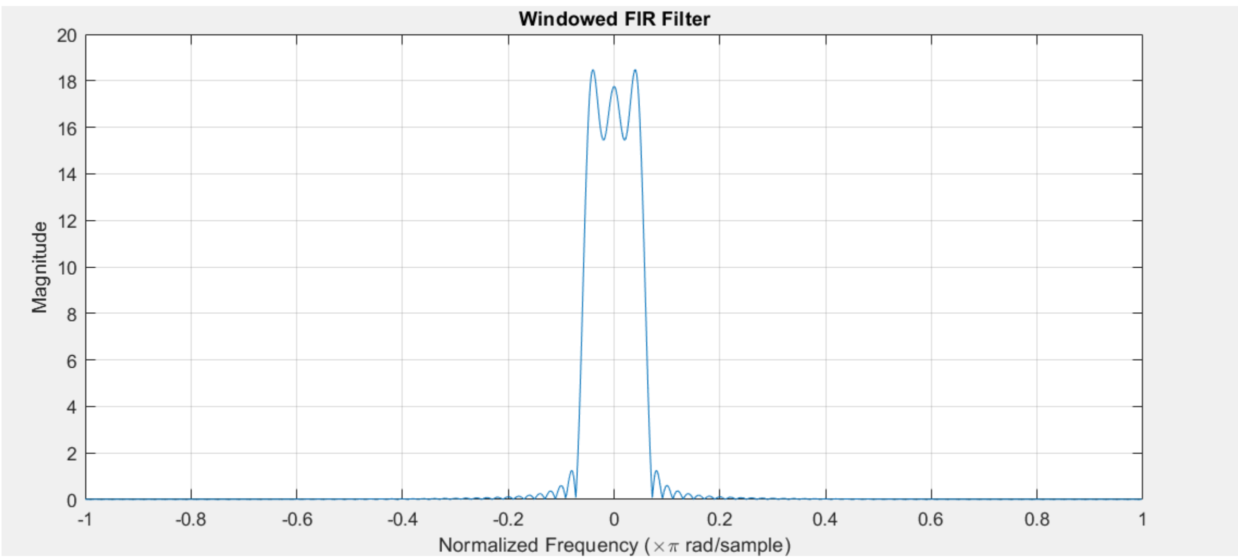
```
Enter the filter length (odd number preferred): 101
Enter the number of frequency response points: 1024
Choose the design method (LS/WLS): WLS
Choose the scale for magnitude response (linear/log): linear
Enter the normalized passband cutoff frequency (0-1): 0.019
Enter the normalized stopband cutoff frequency (0-1): 0.026
Enter the weight for the passband: 1
Enter the weight for the transition band: 0.5
Enter the weight for the stopband: 4
Playing filtered audio signal...
```

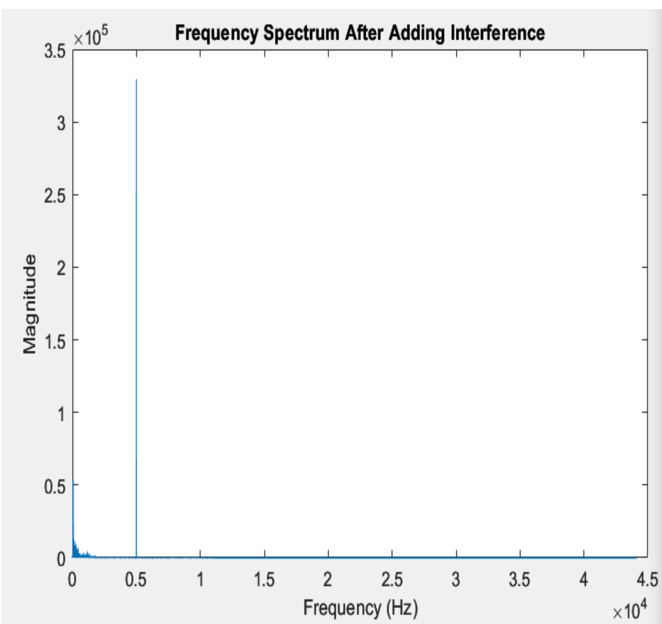


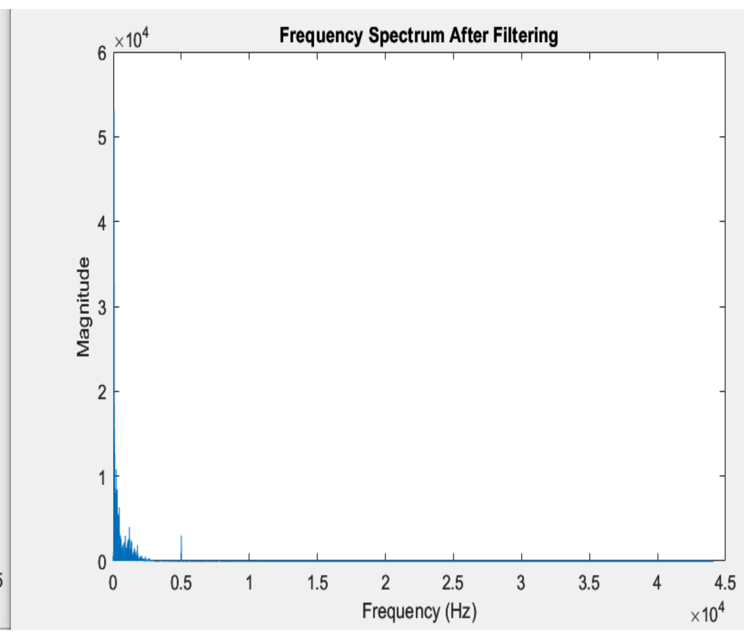




```
Enter the filter length (odd number preferred): 101
Enter the normalized cutoff frequency (0 < fc < 0.5): 0.03
Choose the scale for magnitude response (linear/log): linear
Choose a window type:
1. Rectangular
2. Blackman
3. Chebyshev
4. Kaiser
Enter the number corresponding to the window type: 1
Playing filtered audio signal...
```



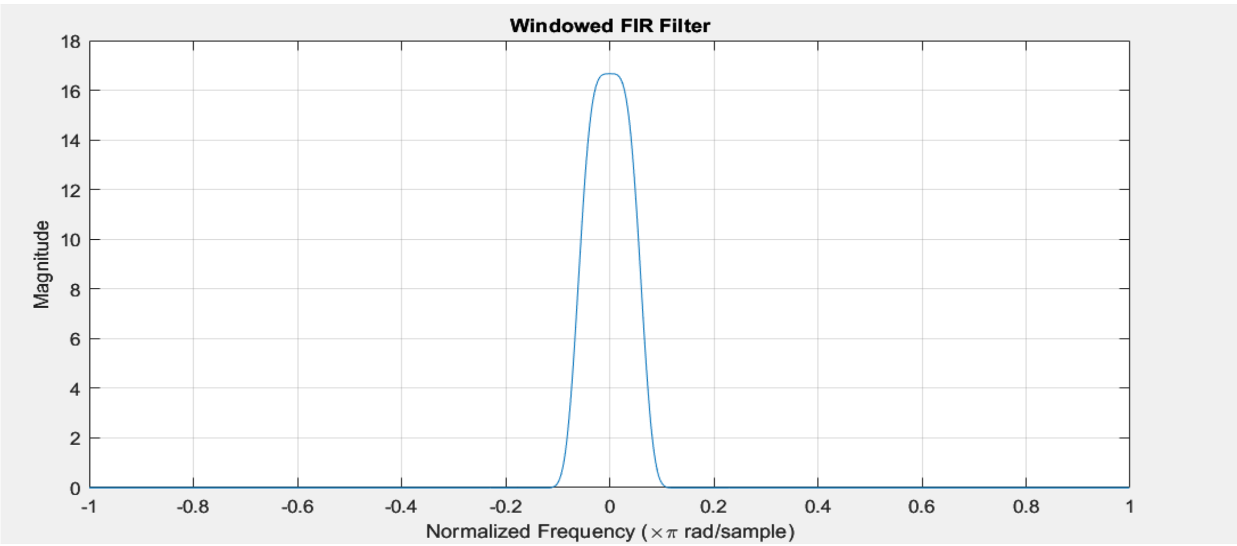


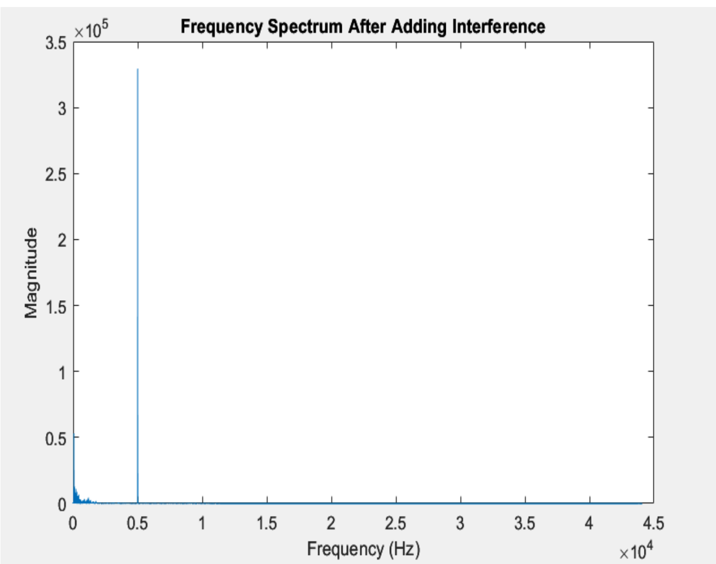


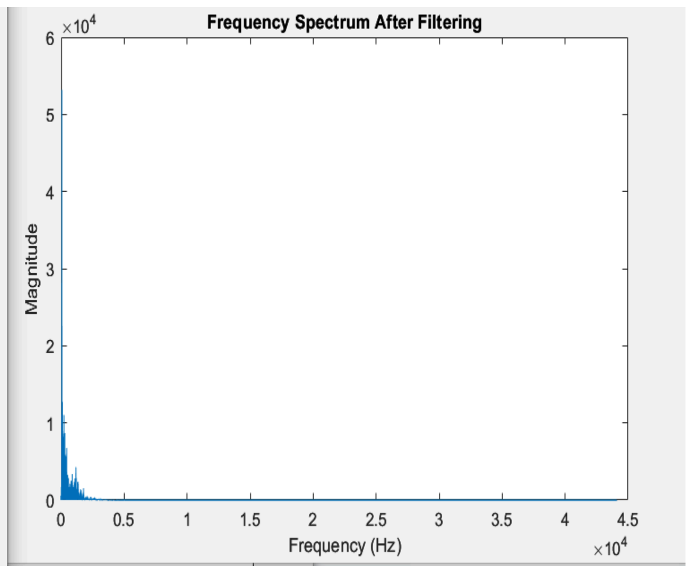
Comment:

Filtering using the Window method demonstrates significantly better results compared to the outputs obtained from LS and WLS methods. As observed, the interference noise is nearly eliminated, showing substantial improvement. However, a small amount of residual noise is still present.

```
Enter the filter length (odd number preferred): 101
Enter the normalized cutoff frequency (0 < fc < 0.5): 0.03
Choose the scale for magnitude response (linear/log): linear
Choose a window type:
1. Rectangular
2. Blackman
3. Chebyshev
4. Kaiser
Enter the number corresponding to the window type: 2
Playing filtered audio signal...
```



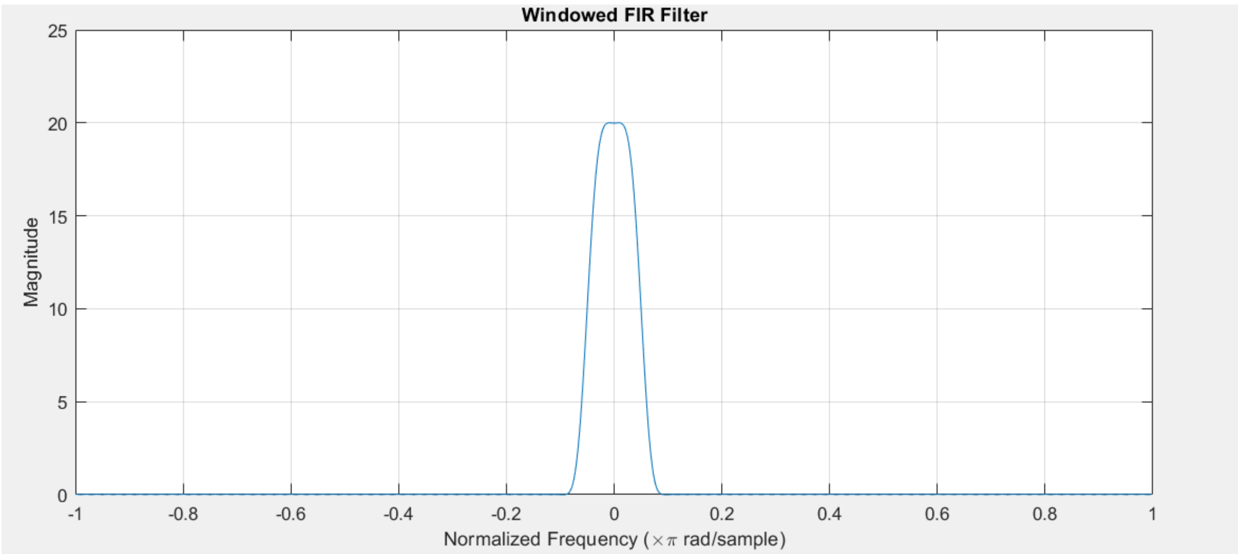


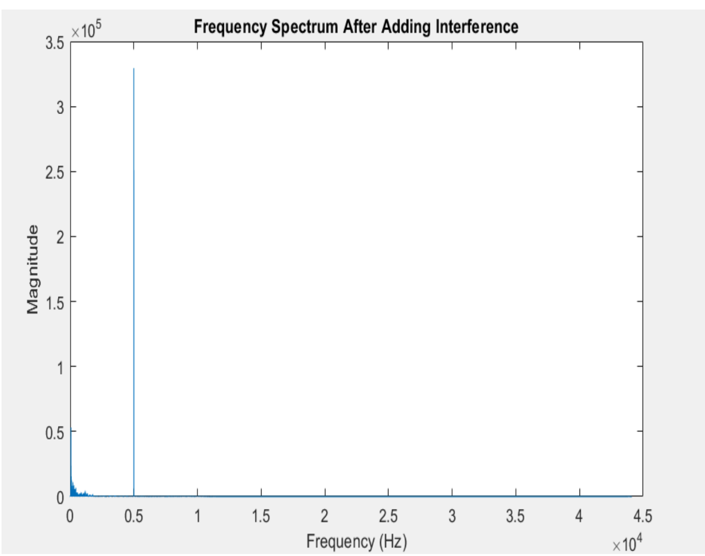


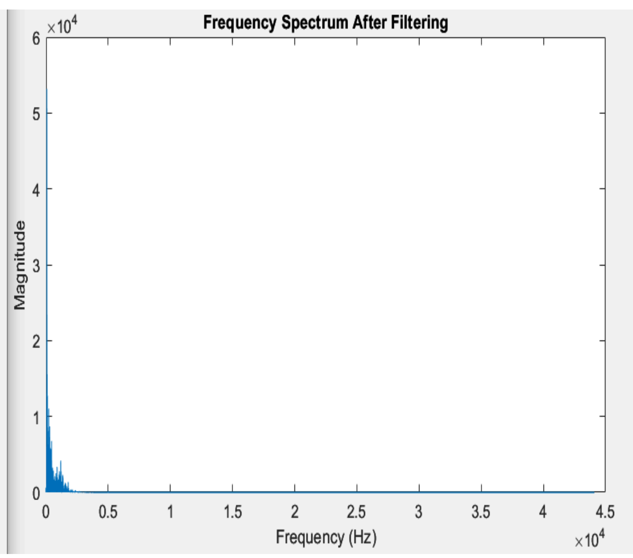
Comment:

Using the Blackman window for filtering produced excellent results, with the interference noise completely eliminated after filtering. This demonstrates the superior performance of the Blackman window in achieving smoother transitions and better frequency attenuation compared to the rectangular window.

```
Enter the filter length (odd number preferred): 101
Enter the normalized cutoff frequency (0 < fc < 0.5): 0.025
Choose the scale for magnitude response (linear/log): linear
Choose a window type:
1. Rectangular
2. Blackman
3. Chebyshev
4. Kaiser
Enter the number corresponding to the window type: 3
Playing filtered audio signal...
```







Comment:

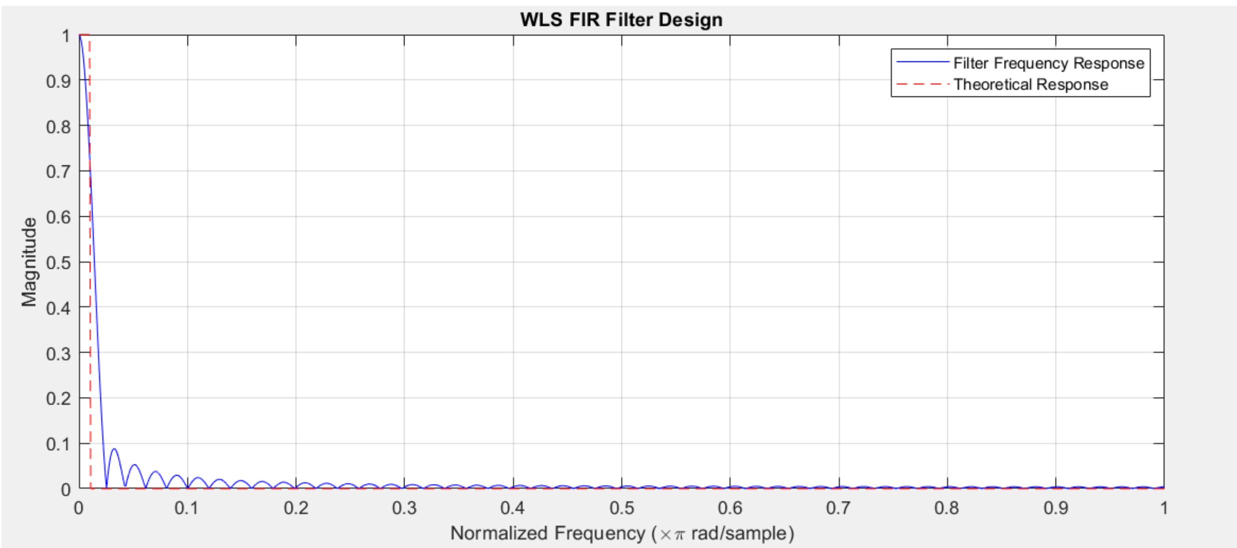
Chebyshev window showed good results also like blackman windo

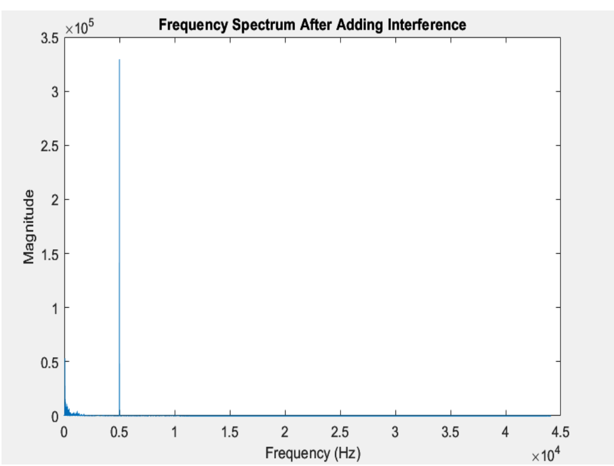
```
%%
% Step 8:  Apply the filter to remove interference from the signal
filteredSignal = filter(h, 1, signalWithInterference);
filteredSignal = filter(h, 1, filteredSignal);
filteredSignal = filter(h, 1, filteredSignal);
```

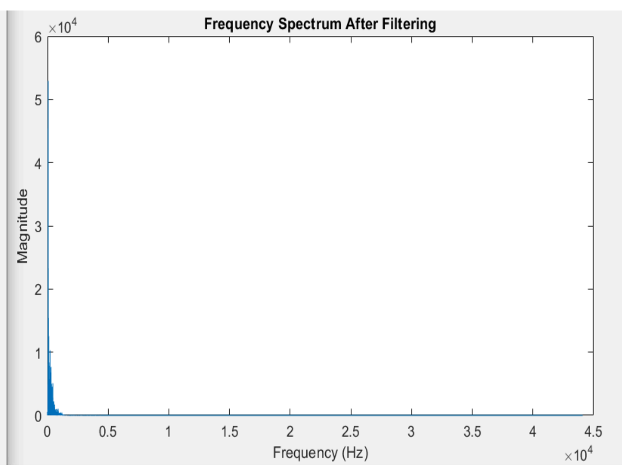
Command Window

```
Enter the filter length (odd number preferred): 101
Enter the number of frequency response points: 1024
Choose the design method (LS/WLS): WLS
Choose the scale for magnitude response (linear/log): linear
Enter the normalized passband cutoff frequency (0-1): 0.01
Enter the normalized stopband cutoff frequency (0-1): 0.018
Enter the weight for the passband: 2
Enter the weight for the transition band: 0.5
Enter the weight for the stopband: 3
Playing filtered audio signal...
```

5000



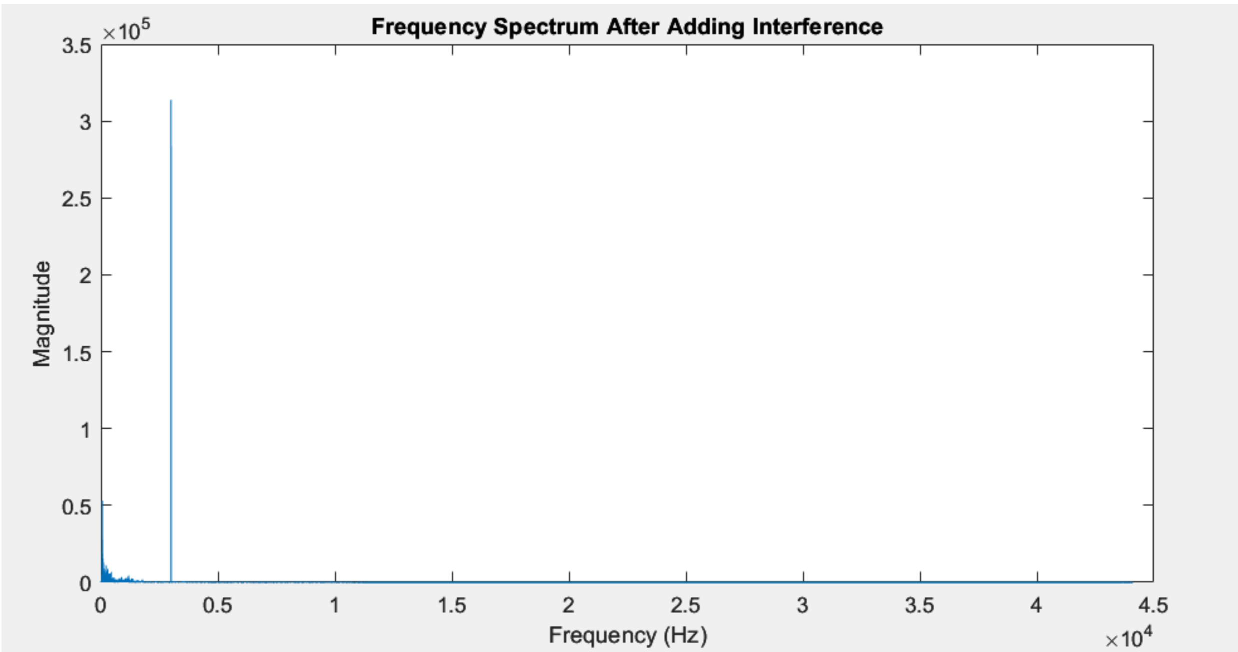




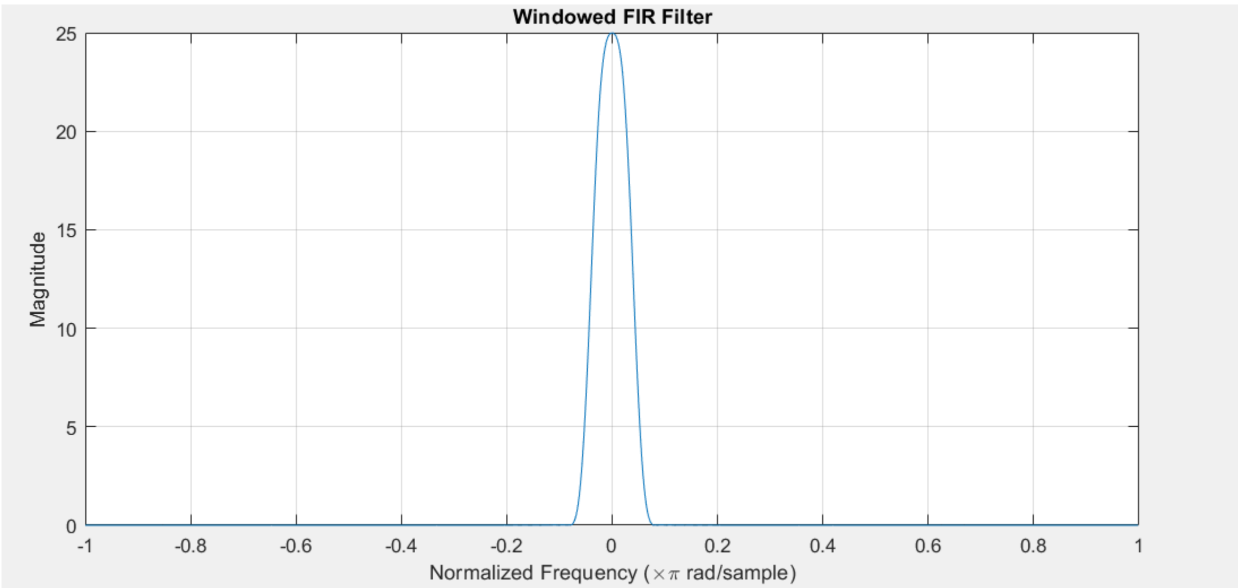
Comment:
We applied three iterations of filtering using the WLS method, which yielded significantly better results compared to a single iteration. This highlights the positive impact of multiple iterations in enhancing the effectiveness of the filtering process.

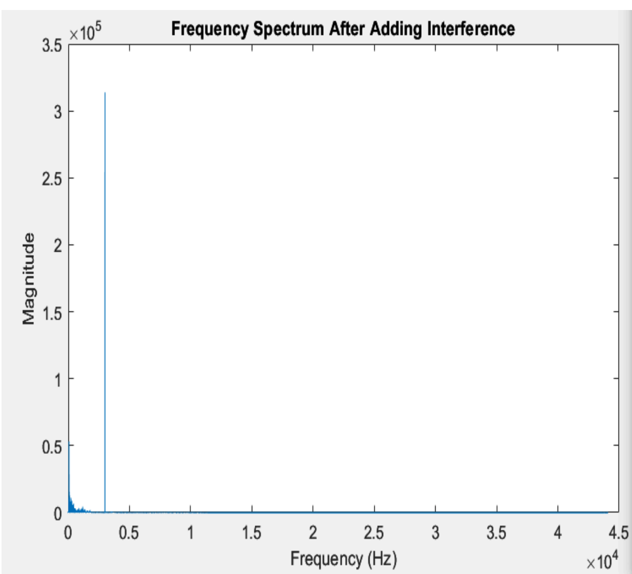
Here we will try another value for interferenceFreq = 3000 HZ

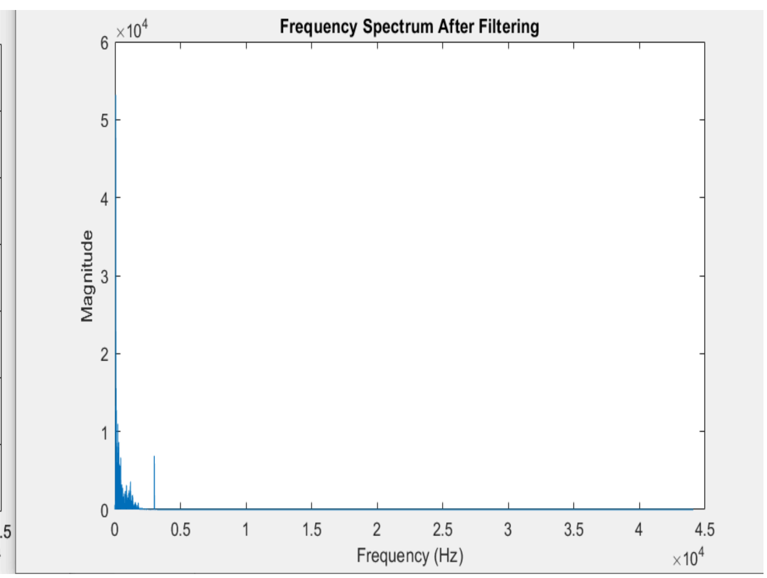
Note: that if we reduce the interferenceFreq sothat it became more close to the signal



```
Enter the filter length (odd number preferred): 101
Enter the normalized cutoff frequency (0 < fc < 0.5): 0.02
Choose the scale for magnitude response (linear/log): linear
Choose a window type:
1. Rectangular
2. Blackman
3. Chebyshev
4. Kaiser
Enter the number corresponding to the window type: 3
Playing filtered audio signal...
```







Comment:

As we reduced the interference frequency, making it closer to the signal, it became more challenging for the same filter parameters to effectively eliminate all the interference noise. While the results were still promising, further adjustments are needed to fully remove the remaining interference.

5. The Optimal Solution:

Based on our previous trials, we have identified several key points that will guide us in choosing the optimal solution:

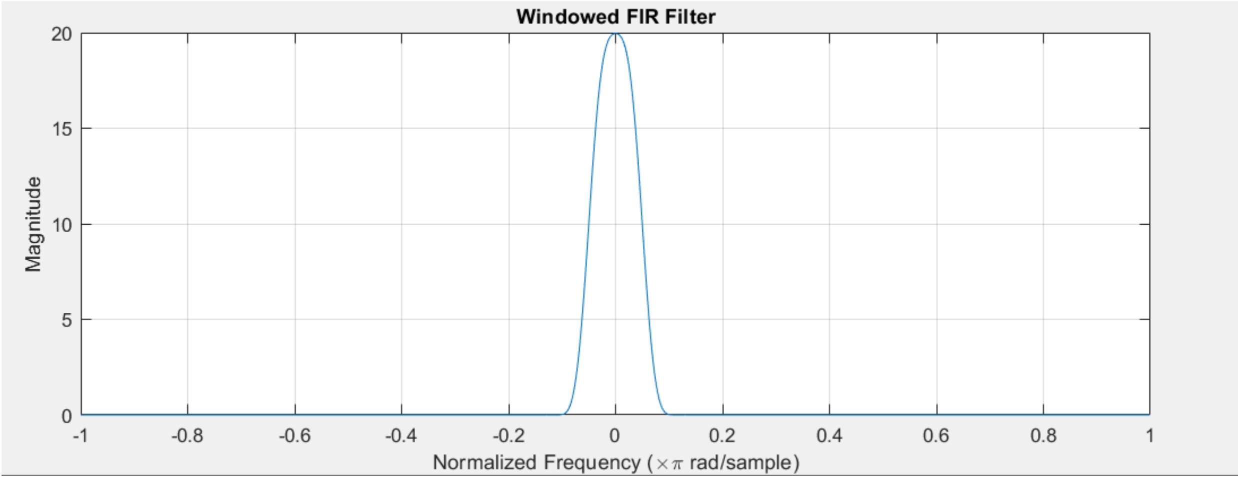
- Window filtering produces significant results.
- Repeating the filtering iterations improves the output.
- Reducing the interference frequency requires readjusting the cutoff frequency.
- Lowering the interference frequency makes the filtering process more challenging.
- Increasing the interference frequency simplifies the filtering process by distancing the noise from the original signal.
- Repeating the filtering iterations, particularly with LS and WLS, yields enhanced results.
- It is important to consider both the filter's performance and its complexity. In our case, filter length will serve as an indicator of complexity."

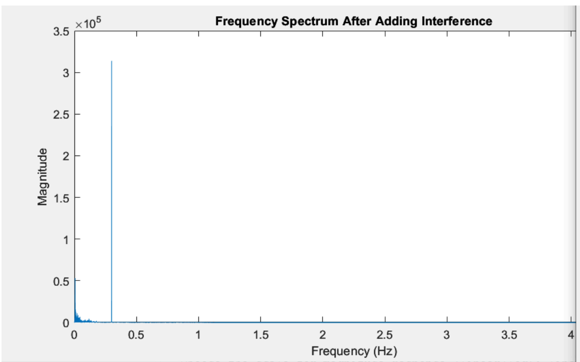
Conditions Used:

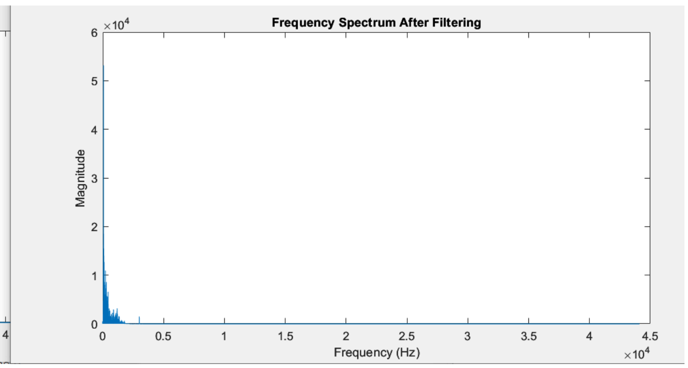
```
Enter the filter length (odd number preferred): 101
Enter the normalized cutoff frequency (0 < fc < 0.5): 0.025
Choose the scale for magnitude response (linear/log): linear
Choose a window type:
1. Rectangular
2. Blackman
3. Chebyshev
4. Kaiser
Enter the number corresponding to the window type: 2
Playing filtered audio signal...
```

interferenceFreq = 3000;

Figures:







Comment:

This filter design successfully removed all interference and restored the original signal without any noise. It achieved this using a filter length of 101, which indicates the filter's complexity, and importantly, no additional iterations were needed, which is advantageous for maintaining low complexity. We set the interference frequency to 3000, which was relatively close to the original signal, presenting a challenge for the filter. However, our design effectively handled this and produced satisfactory results.

This would result in an audio output that is free of the high-pitched tone or noise caused by the interference, preserving the clarity and naturalness of the original content.

---

6. Audio Filtering and Final Results

- Listen to the filtered signal.
- You can find the restored audio saved in the project folder.

Comments:

This would result in an audio output that is free of the high-pitched tone or noise caused by the interference, preserving the clarity and naturalness of the original content. However, if the filter is too aggressive, it might unintentionally attenuate frequencies near the interference, causing a loss of audio fidelity.
