Experiment Name:

Experiment Name:

Design and Analysis of a Lowpass FIR Filter using Hanning Window

Objective:

1. To design an FIR filter that allows frequencies within the specified passband.
2. To attenuate frequencies in the stopband, achieving a clear cutoff.
3. To analyze the frequency response of the designed filter to ensure it meets the required specifications.

Theory:

An FIR (Finite Impulse Response) filter is a digital filter commonly used for its stability, linear phase response, and simplicity in design. FIR filters achieve frequency selectivity by summing past input values with a fixed set of coefficients, resulting in a finite response to an impulse input.

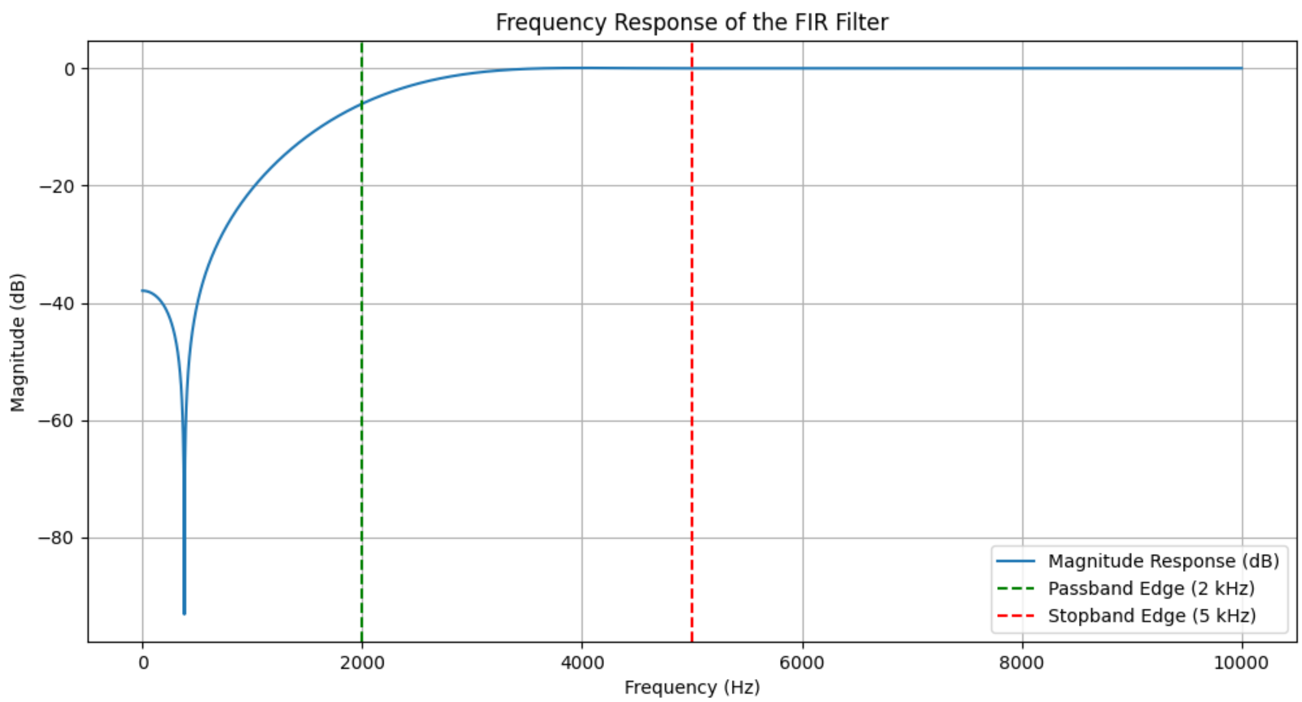
In this experiment, we are creating a bandpass FIR filter to pass frequencies within a range and attenuate frequencies outside this range.

- Passband and Stopband Edges: The passband edge (2 kHz) defines the frequency range allowed to pass with minimal attenuation, while the stopband edge (5 kHz) marks the point beyond which frequencies are attenuated.
- Sampling Frequency (Fs): This is set to 20 kHz, defining the highest frequency accurately representable (10 kHz), which ensures the filter's performance is within the desired frequency range.
- Filter Length (M): The filter length, or the number of taps (21 in this case), defines the filter's complexity and sharpness. Longer filters offer sharper cutoffs but are computationally heavier.
- Hanning Window: The Hanning window is applied to the filter coefficients to shape the filter's frequency response, minimizing ripples and sidelobes. This window reduces artifacts in the passband and smoothens the transition band, producing a clean, reliable output.

Python Source Code:

```
import numpy as np

import matplotlib.pyplot as plt
```

```
from scipy.signal import firwin, freqz

# Define filter specifications
fs = 20000  # Sampling frequency in Hz
fp = 2000   # Passband edge frequency in Hz
fsb = 5000  # Stopband edge frequency in Hz
M = 21      # Filter length

# Normalize frequencies
normalized_fp = fp / (fs / 2)  # Normalize passband edge frequency
normalized_fs = fsb / (fs / 2)  # Normalize stopband edge frequency

# Design the FIR filter using Hanning window
filter_coefficients = firwin(M, normalized_fp, window='hann', pass_zero=False)

# Frequency response
w, h = freqz(filter_coefficients, worN=8000)

# Plot frequency response
plt.figure(figsize=(12, 6))
plt.plot(w * fs / (2 * np.pi), 20 * np.log10(np.abs(h)), label='Magnitude Response (dB)')
plt.axvline(fp, color='green', linestyle='--', label="Passband Edge (2 kHz)")
plt.axvline(fsb, color='red', linestyle='--', label="Stopband Edge (5 kHz)")
plt.xlabel('Frequency (Hz)')
plt.ylabel('Magnitude (dB)')
plt.title('Frequency Response of the FIR Filter')
plt.grid()
plt.legend()
plt.show()
```


Frequency Response of the FIR Filter
0
−20
−40
−60
−80
Magnitude (dB)
0
2000
4000
6000
8000
10000
Frequency (Hz)
Magnitude Response (dB)
Passband Edge (2 kHz)
Stopband Edge (5 kHz)

Output: