Experiment No:

Experiment Name:

Design and Analysis of a Lowpass FIR Filter using Hamming Window

Objective:

1. To design a lowpass FIR filter that meets specific passband and stopband requirements.
2. To analyze the filter's frequency response, ensuring it meets attenuation goals beyond the stopband.

Theory:

A **lowpass FIR (Finite Impulse Response) filter** allows signals below a specified frequency (the passband edge) to pass, while attenuating frequencies above a stopband edge. FIR filters are popular for their stability and consistent linear phase.

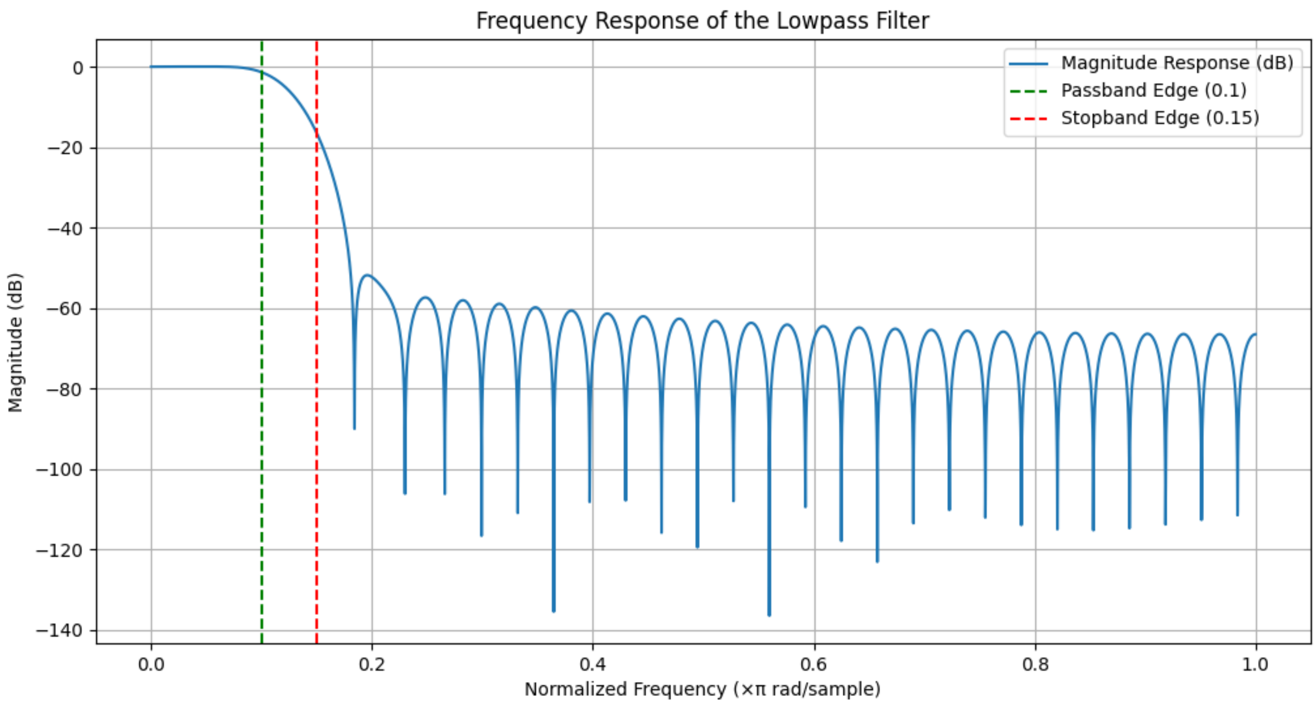
1. **Filter Length (M)**: The filter length, 61 in this case, affects how sharp the transition between passband and stopband can be. Longer filters have sharper transitions but require more computation.
2. **Passband and Stopband Edges**: The passband edge (fp) is 0.1, and the stopband edge (fs) is 0.15 in normalized frequency, indicating that frequencies below fp pass through, while those above fs are attenuated.
3. **Cutoff Frequency (fc)**: Set midway between the passband and stopband edges, this cutoff frequency defines the boundary for allowed frequencies.
4. **Hamming Window**: The Hamming window minimizes ripples in the passband and offers good transition smoothness, making it ideal for lowpass FIR filters. The window is applied to shape the frequency response of the filter coefficients, ensuring clear attenuation beyond the stopband edge.

Python Source Code:

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import firwin, freqz


# Define filter specifications
```

```
M = 61          # Filter length
fp = 0.1        # Passband edge frequency (normalized)
fs = 0.15       # Stopband edge frequency (normalized)

# Set the cutoff frequency to the average of the passband and stopband edges
fc = (fp + fs) / 2

# Design the FIR lowpass filter using the Hamming window
filter_coefficients = firwin(M, fc, window='hamming', pass_zero=True)

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

# Plot frequency response
plt.figure(figsize=(12, 6))
plt.plot(w / np.pi, 20 * np.log10(np.abs(h)), label='Magnitude Response (dB)')
plt.axvline(fp, color='green', linestyle='--', label="Passband Edge (0.1)")
plt.axvline(fs, color='red', linestyle='--', label="Stopband Edge (0.15)")
plt.xlabel('Normalized Frequency (×π rad/sample)')
plt.ylabel('Magnitude (dB)')
plt.title('Frequency Response of the Lowpass Filter')
plt.grid()
plt.legend()
plt.show()
```

Output:



Frequency Response of the Lowpass Filter
Magnitude Response (dB)
Passband Edge (0.1)
Stopband Edge (0.15)
Magnitude (dB)
0
−20
−40
−60
−80
−100
−120
−140
0.0
0.2
0.4
0.6
0.8
1.0
Normalized Frequency (×π rad/sample)