Experiment No:

Experiment Name:

Design and Analysis of an FIR Bandpass Filter.

Objective:

1. To design an FIR bandpass filter using specified edge frequencies.
2. To allow specific frequencies within the passband to pass while attenuating frequencies outside this range.
3. To eliminate unwanted signals outside the passband, achieving a clear and targeted frequency response.
4. To analyze the frequency response of the designed filter to ensure it meets desired specifications.

Theory:

A bandpass filter is a type of digital filter that allows frequencies within a specified range, known as the passband, to pass while attenuating frequencies outside this range. In this experiment, we are designing an FIR (Finite Impulse Response) bandpass filter with specified passband and stopband edge frequencies.

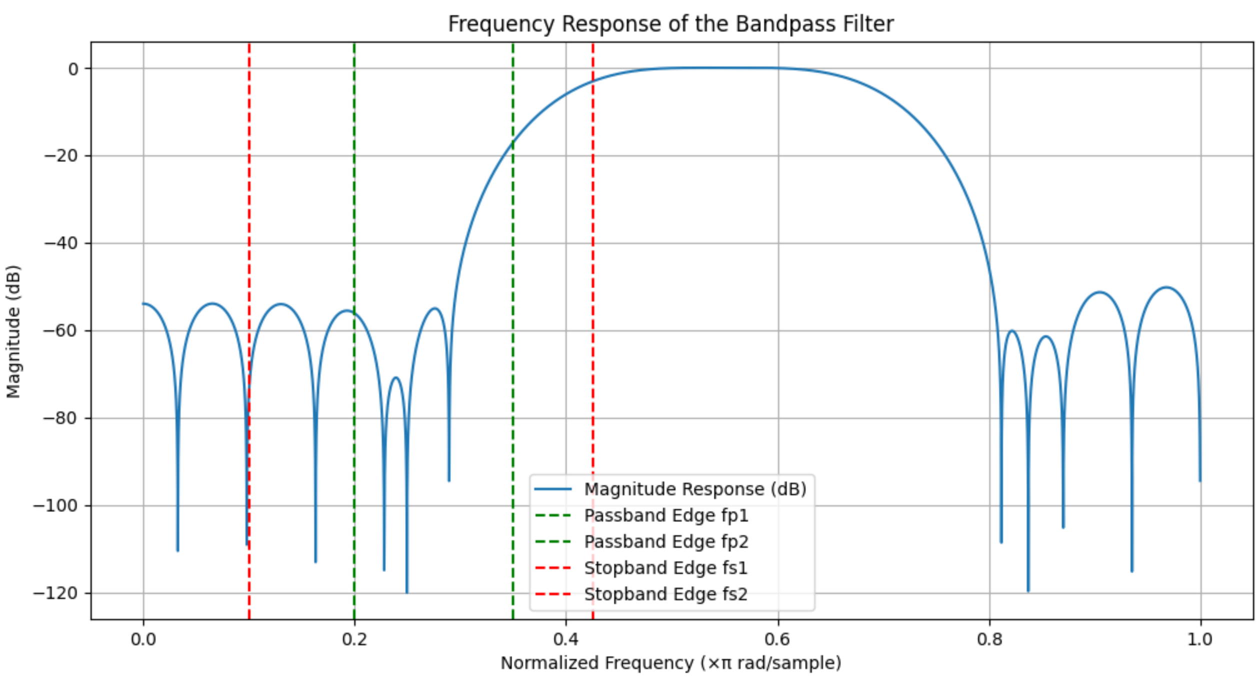
1. Filter Length (M): The number of taps in the FIR filter. A longer filter length often results in a sharper filter response.
2. Passband Edge Frequencies: These frequencies define the range of frequencies that the filter allows to pass. For this filter, the passband edge frequencies are $f_p1 = 0.2$ and $f_p2 = 0.35$.
3. Stopband Edge Frequencies: Frequencies outside the passband where the filter significantly attenuates the signal. Here, the stopband edge frequencies are $f_s1 = 0.1$ and $f_s2 = 0.425$
4. FIR Filter Design: Using the windowing method, specifically the Hamming window, we design the FIR filter based on the desired frequency specifications. The firwin function from the SciPy library is used to generate the filter coefficients for this design.

Python Source Code:

```
import numpy as np

import matplotlib.pyplot as plt

from scipy.signal import firwin, freqz


# Define filter specifications
```

```
M = 32  # Filter length
fs = 1  # Sampling frequency (normalized for design, so fs=1)
fp1, fp2 = 0.2, 0.35  # Passband edge frequencies
fs1, fs2 = 0.1, 0.425  # Stopband edge frequencies

# Calculate the cutoff frequencies for firwin
cutoff = [fp1, fp2]

# Design the FIR bandpass filter
filter_coefficients = firwin(M, cutoff, pass_zero=False, fs=fs)

# 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(fp1, color='green', linestyle='--', label="Passband Edge fp1")
plt.axvline(fp2, color='green', linestyle='--', label="Passband Edge fp2")
plt.axvline(fs1, color='red', linestyle='--', label="Stopband Edge fs1")
plt.axvline(fs2, color='red', linestyle='--', label="Stopband Edge fs2")
plt.xlabel('Normalized Frequency (×π rad/sample)')
plt.ylabel('Magnitude (dB)')
plt.title('Frequency Response of the Bandpass Filter')
plt.grid()
plt.legend()
plt.show()
```

Output:



Frequency Response of the Bandpass Filter
0
−20
−40
−60
−80
−100
−120
Magnitude (dB)
0.0
0.2
0.4
0.6
0.8
1.0
Normalized Frequency (×π rad/sample)
Magnitude Response (dB)
Passband Edge fp1
Passband Edge fp2
Stopband Edge fs1
Stopband Edge fs2