**Experiment 8: Linear Phase F. I. R. Filter Design using Windowing Method**

|  |  |
| --- | --- |
| **Name** | Prathamesh Mane |
| **UID no. & Branch** | **2022200078 (B1)** |
| **Experiment No.** | 8 |

Code:

% Linear Phase FIR Filter Design using Windowing Method

clc;

clear;

close all;

disp("Step 1: Accept user input specifications");

disp('Enter filter type (LPF, HPF, BPF, BSF): ');

filter\_type = input('Enter filter type (LPF, HPF, BPF, BSF): ', 's');

Fs = input('Enter the sampling frequency (Hz): ');

if strcmpi(filter\_type, 'LPF') || strcmpi(filter\_type, 'HPF')

Fp = input('Enter passband frequency (Hz): ');

Fs\_stop = input('Enter stopband frequency (Hz): ');

else

Fp1 = input('Enter lower passband frequency (Hz): ');

Fp2 = input('Enter upper passband frequency (Hz): ');

Fs\_stop1 = input('Enter lower stopband frequency (Hz): ');

Fs\_stop2 = input('Enter upper stopband frequency (Hz): ');

end

Ap = input('Enter passband attenuation (dB): ');

As = input('Enter stopband attenuation (dB): ');

N = input('Enter the filter order (N): ');

if mod(N, 2) == 0

warning('Filter order should be odd for linear phase. Incrementing N by 1.');

N = N + 1;

end

disp("step 2: Select appropriate window function");

if As <= 21

window\_type = 'rectwin';

elseif As <= 44

window\_type = 'hann';

elseif As <= 53

window\_type = 'hamming';

else

window\_type = 'blackman';

end

fprintf('Selected window: %s\n', window\_type);

disp("Step 3: Normalize frequencies");

if strcmpi(filter\_type, 'LPF') || strcmpi(filter\_type, 'HPF')

Wp = Fp / (Fs / 2)

Ws = Fs\_stop / (Fs / 2)

Wn = [Wp];

else

Wp1 = Fp1 / (Fs / 2);

Wp2 = Fp2 / (Fs / 2);

Ws1 = Fs\_stop1 / (Fs / 2);

Ws2 = Fs\_stop2 / (Fs / 2);

Wn = [Wp1 Wp2];

end

disp("Step 4: Design the filter using the selected window")

switch lower(filter\_type)

case 'lpf'

b = fir1(N-1, Wn, 'low', window(window\_type, N))

case 'hpf'

b = fir1(N-1, Wn, 'high', window(window\_type, N))

case 'bpf'

b = fir1(N-1, Wn, 'bandpass', window(window\_type, N))

case 'bsf'

b = fir1(N-1, Wn, 'stop', window(window\_type, N))

otherwise

error('Invalid filter type.');

end

disp("Step 5: Frequency response");

[H, f] = freqz(b, 1, 1024, Fs);

figure;

subplot(2, 1, 1);

plot(f, 20\*log10(abs(H)));

grid on;

xlabel('Frequency (Hz)');

ylabel('Magnitude (dB)');

title('Magnitude Spectrum');

xlim([0 500]);

ylim([-70, 5]);

hold on;

yline(-Ap, 'r--', 'Passband Attenuation (Ap)');

yline(-As, 'g--', 'Stopband Attenuation (As)');

hold off;

subplot(2, 1, 2);

plot(f, angle(H));

grid on;

xlabel('Frequency (Hz)');

ylabel('Phase (radians)');

title('Phase Spectrum');

disp("Step 7: Impulse response");

figure;

stem(b, 'filled');

grid on;

xlabel('Samples');

ylabel('Amplitude');

title('Impulse Response of FIR Filter');

disp('Observe the phase spectrum for linearity. Linear phase indicates symmetry in the impulse response.');