

Implementation of FIR Filters

Aim: Write a MATLAB program to implement the following FIR filters using Hanning, Hamming, Rectangular and Triangular windows.

- a) Low Pass Filter
- b) High Pass Filter
- c) Band Pass Filter
- d) Band Stop Filter

Theory

Finite Impulse Response (FIR) filters are a type of digital filter characterized by a finite duration of the impulse response. The window method is a common technique used to design FIR filters. This method involves multiplying an ideal (infinite) impulse response by a window function to create a realizable FIR filter. The FIR filter coefficients $h[n]$ are obtained by multiplying the ideal impulse response by the chosen window function. The frequency response of the FIR filter can be analyzed using the Discrete Fourier Transform (DFT). The window function influences the main lobe width and side lobe levels in the frequency response. A narrower main lobe provides better frequency resolution, while lower side lobes reduce spectral leakage.

Windows:

1. Rectangular window:

$$w_{rec}(n) = 1, \quad -M \leq n \leq M.$$

2. Triangular (Bartlett) window:

$$w_{tri}(n) = 1 - \frac{|n|}{M}, \quad -M \leq n \leq M.$$

3. Hanning window:

$$w_{han}(n) = 0.5 + 0.5 \cos\left(\frac{n\pi}{M}\right), \quad -M \leq n \leq M.$$

4. Hamming window:

$$w_{ham}(n) = 0.54 + 0.46 \cos\left(\frac{n\pi}{M}\right), \quad -M \leq n \leq M.$$

Filters:

$$\begin{aligned}\text{Lowpass:} \quad h(n) &= \begin{cases} \frac{\Omega_c}{\pi} & n = 0 \\ \frac{\sin(\Omega_c n)}{n\pi} & \text{for } n \neq 0 \end{cases} \quad -M \leq n \leq M \\ \text{Highpass:} \quad h(n) &= \begin{cases} \frac{\pi - \Omega_c}{\pi} & n = 0 \\ -\frac{\sin(\Omega_c n)}{n\pi} & \text{for } n \neq 0 \end{cases} \quad -M \leq n \leq M \\ \text{Bandpass:} \quad h(n) &= \begin{cases} \frac{\Omega_H - \Omega_L}{\pi} & n = 0 \\ \frac{\sin(\Omega_H n)}{n\pi} - \frac{\sin(\Omega_L n)}{n\pi} & \text{for } n \neq 0 \end{cases} \quad -M \leq n \leq M \\ \text{Bandstop:} \quad h(n) &= \begin{cases} \frac{\pi - \Omega_H + \Omega_L}{\pi} & n = 0 \\ -\frac{\sin(\Omega_H n)}{n\pi} + \frac{\sin(\Omega_L n)}{n\pi} & \text{for } n \neq 0 \end{cases} \quad -M \leq n \leq M\end{aligned}$$

Program

a) Low Pass Filter

```
wc = 0.5*pi;
N=50;
alpha = (N-1)/2;
n=0:1:N-1;
hd=sin(wc*(n-alpha))./(pi*(n-alpha));
%LPFhamming
w1=hamming(N);
hn=hd.*w1';
w=0:0.01:pi;
h1=freqz(hn,1,w);
subplot(4,2,1);
plot(w/pi,10*log10(abs(h1)));
title('LPF using hamming window');
xlabel('normalized frequency');
ylabel('magnitude in db');
%LPFhanning
w2=hanning(N);
```

```

hn=hd.*w2';
w=0:0.01:pi;
h2=freqz(hn,1,w);
subplot(4,2,3);
plot(w/pi,10*log10(abs(h2)));
title('LPF using hanning window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%LPFrect
w3=boxcar(N);
hn=hd.*w3';
w=0:0.01:pi;
h3=freqz(hn,1,w);
subplot(4,2,5);
plot(w/pi,10*log10(abs(h3)));
title('LPF using rectangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%LPFtri
w4=bartlett(N);
hn=hd.*w4';
w=0:0.01:pi;
h4=freqz(hn,1,w);
subplot(4,2,7);
plot(w/pi,10*log10(abs(h4)));
title('LPF using triangular window');

```

```
xlabel('normalized frequency');
ylabel('magnitude in db');
%hamming
subplot(4,2,2);
stem(w1);
title('hamming window sequence');
xlabel('no of samples');
ylabel('amplitude');
%hanning
subplot(4,2,4);
stem(w2);
title('hanning window sequence');
xlabel('no of samples');
ylabel('amplitude');
%rectangular
subplot(4,2,6);
stem(w3);
title('rectangular window sequence');
xlabel('no of samples');
ylabel('amplitude');
%triangular
subplot(4,2,8);
stem(w4);
title('tirangular window sequence');
xlabel('no of samples');
ylabel('amplitude');
```

b) High Pass Filter

```
clc;
clear all;
close all;
wc = 0.5*pi;
N=50;
alpha = (N-1)/2;
n=0:1:N-1;
hd=(sin(pi*(n-alpha))-sin(wc*(n-alpha)))/(pi*(n-alpha));
%HPFhamming
w1=hamming(N);
hn=hd.*w1';
w=0:0.01:pi;
h1=freqz(hn,1,w);
subplot(4,2,1);
plot(w/pi,10*log10(abs(h1)));
title('HPF using hamming window');
xlabel('normalized frequency');
ylabel('magnitude in db');
%HPFhanning
w2=hanning(N);
hn=hd.*w2';
w=0:0.01:pi;
h2=freqz(hn,1,w);
subplot(4,2,3);
plot(w/pi,10*log10(abs(h2)));
```

```

title('HPF using hanning window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%HPFrect

w3=boxcar(N);
hn=hd.*w3';
w=0:0.01:pi;
h3=freqz(hn,1,w);
subplot(4,2,5);
plot(w/pi,10*log10(abs(h3)));
title('HPF using rectangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%HPFtri

w4=bartlett(N);
hn=hd.*w4';
w=0:0.01:pi;
h4=freqz(hn,1,w);
subplot(4,2,7);
plot(w/pi,10*log10(abs(h4)));
title('HPF using triangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%WINDOWS

%hamming
subplot(4,2,2);

```

```
stem(w1);
title('hamming window sequence');
xlabel('no of samples');
ylabel('amplitude');

%hanning
subplot(4,2,4);
stem(w2);
title('hanning window sequence');
xlabel('no of samples');
ylabel('amplitude');

%rectangular
subplot(4,2,6);
stem(w3);
title('rectangular window sequence');
xlabel('no of samples');
ylabel('amplitude');

%triangular
subplot(4,2,8);
stem(w4);
title('tirangular window sequence');
xlabel('no of samples');
ylabel('amplitude');
```

c) Band Pass Filter

```
clc;
clear all;
close all;
wcl= 0.25*pi;
wc2 = 0.75*pi;
N=50;
%N = input('enter the value of N');
alpha = (N-1)/2;
n=0:1:N-1;
hd=(sin(wcl*(n-alpha+eps))-sin(wc2*(n-alpha+eps)))./(pi*(n-
alpha+eps));
%BPFFhamming
w1=hamming(N);
hn=hd.*w1';
w=0:0.01:pi;
h1=freqz(hn,1,w);
subplot(4,2,1);
plot(w/pi,10*log10(abs(h1)));
title('BPF using hamming window');
xlabel('normalized frequency');
ylabel('magnitude in db');
%BPFFhanning
w2=hanning(N);
hn=hd.*w2';
w=0:0.01:pi;
```



```

h2=freqz(hn,1,w);
subplot(4,2,3);
plot(w/pi,10*log10(abs(h2)));
title('BPF using hanning window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%BPFFrect
w3=boxcar(N);
hn=hd.*w3';
w=0:0.01:pi;
h3=freqz(hn,1,w);
subplot(4,2,5);
plot(w/pi,10*log10(abs(h3)));
title('BPF using rectangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%BPFFtri
w4=bartlett(N);
hn=hd.*w4';
w=0:0.01:pi;
h4=freqz(hn,1,w);
subplot(4,2,7);
plot(w/pi,10*log10(abs(h4)));
title('BPF using triangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

```

```

%WINDOWS

%hamming
subplot(4,2,2);
stem(w1);
title('hamming window sequence');
xlabel('no of samples');
ylabel('amplitude');

%hanning
subplot(4,2,4);
stem(w2);
title('hanning window sequence');
xlabel('no of samples');
ylabel('amplitude');

%rectangular
subplot(4,2,6);
stem(w3);
title('rectangular window sequence');
xlabel('no of samples');
ylabel('amplitude');

%triangular
subplot(4,2,8);
stem(w4);
title('tirangular window sequence');
xlabel('no of samples');
ylabel('amplitude');

```

d) Band Stop Filter

```
wc1= 0.25*pi;
wc2 = 0.75*pi;
N=50;

%N = input('enter the value of N');

alpha = (N-1)/2;

n=0:1:N-1;

hd=(sin(wc1*(n-alpha+eps))-sin(wc2*(n-alpha+eps))+sin(pi*(n-
alpha+eps)))./(pi*(n-alpha+eps));

%BSFhamming

w1=hamming(N);

hn=hd.*w1';

w=0:0.01:pi;

h1=freqz(hn,1,w);

subplot(4,2,1);

plot(w/pi,10*log10(abs(h1)));

title('BSF using hamming window');

xlabel('normalized frequency');

ylabel('magnitude in db');

%BSFhanning

w2=hanning(N);

hn=hd.*w2';

w=0:0.01:pi;

h2=freqz(hn,1,w);

subplot(4,2,3);

plot(w/pi,10*log10(abs(h2)));
```

```

title('BSF using hanning window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%BSFrect

w3=boxcar(N);
hn=hd.*w3';
w=0:0.01:pi;
h3=freqz(hn,1,w);
subplot(4,2,5);
plot(w/pi,10*log10(abs(h3)));
title('BSF using rectangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%BSFtri

w4=bartlett(N);
hn=hd.*w4';
w=0:0.01:pi;
h4=freqz(hn,1,w);
subplot(4,2,7);
plot(w/pi,10*log10(abs(h4)));
title('BSF using triangular window');
xlabel('normalized frequency');
ylabel('magnitude in db');

%WINDOWS

%hamming
subplot(4,2,2);

```

```

stem(w1);
title('hamming window sequence');
xlabel('no of samples');
ylabel('amplitude');

%hanning
subplot(4,2,4);
stem(w2);
title('hanning window sequence');
xlabel('no of samples');
ylabel('amplitude');

%rectangular
subplot(4,2,6);
stem(w3);
title('rectangular window sequence');
xlabel('no of samples');
ylabel('amplitude');

%triangular
subplot(4,2,8);
stem(w4);
title('tirangular window sequence');
xlabel('no of samples');
ylabel('amplitude');

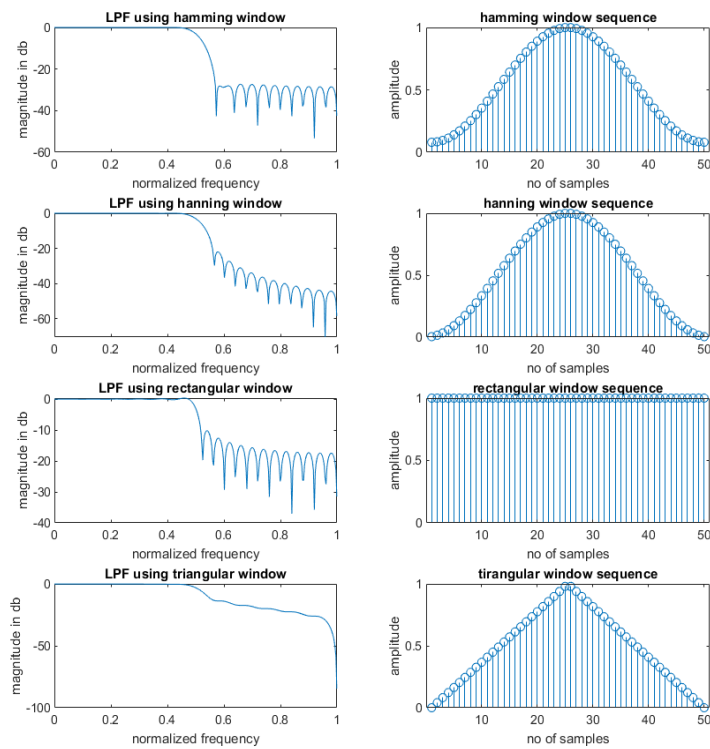
```

Result

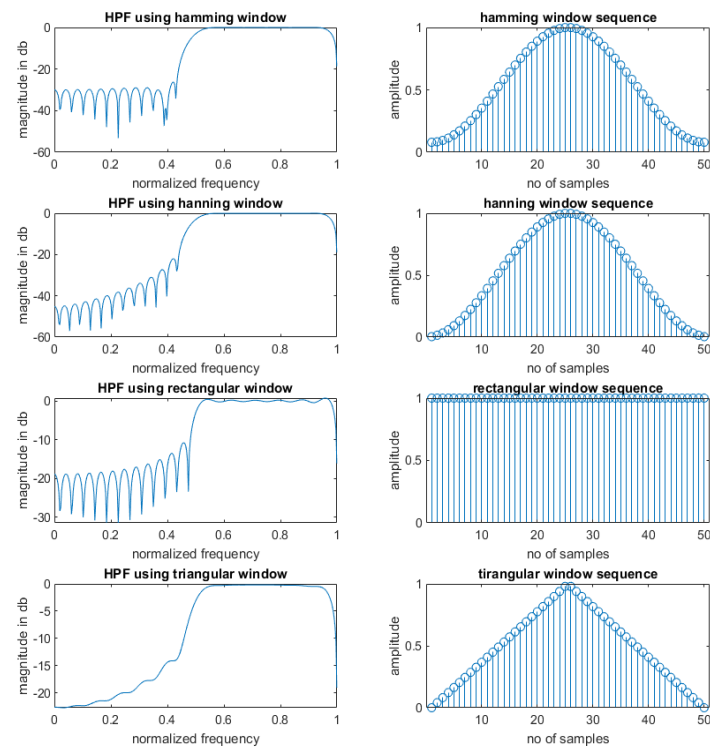
Implemented FIR filters using Window method.

Observation

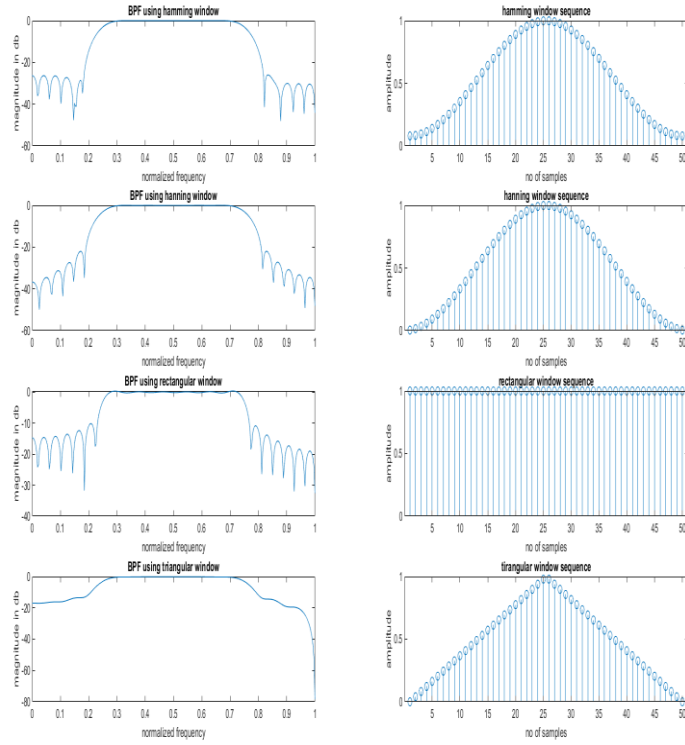
a) Low Pass Filter



b) High Pass Filter



c) Band Pass Filter



d) Band Stop Filter

