Experiment No:09 Date:24-10-2024

IMPLEMENTATION OF FIR FILTER

Aim

To design and implement FIR filters by various window methods.

Theory

In digital signal processing, **Finite Impulse Response (FIR) filters** are widely used because they provide a linear phase response, making them ideal for applications where phase distortion must be minimized. FIR filters can be implemented using various design methods, among which the **windowing method** is simple and effective. An FIR filter has a finite impulse response, achieved by truncating the ideal, infinite impulse response. This truncation is performed with a **window function** to make the filter realizable and control the ripple and decay rates in the frequency response.

Window functions modify the ideal impulse response to reduce unwanted frequency components and control the sharpness of the transition between passband and stopband. The following are the equations for the commonly used windows:

1. **Rectangular Window**:
    - The rectangular window is defined as:

      $w(n)=1, 0\leq n\leq N-1$

      It has a narrow main lobe but high sidelobes, leading to poor stopband attenuation.
2. **Hamming Window**:
    - The Hamming window is defined as:

      $w(n)=0.54-0.46\cos(2\pi n/N-1), 0\leq n\leq N-1$

      This window reduces the sidelobe levels significantly compared to the rectangular window, providing better stopband attenuation.
3. **Hanning Window**:
    - The Hanning window is defined as:

      $w(n)=0.5(1-\cos(2\pi n/N-1)), 0\leq n\leq N-1$

      The Hanning window has lower sidelobe levels than the rectangular window, though it sacrifices some sharpness in the transition.
4. **Bartlett Window**:

- The Bartlett window, also known as the triangular window, is defined as: $w(n)=2n/N-1, 0 \le n \le N-1/2$

  $=2-2n/N-1, N-1/2<n \le N-1$

  The Bartlett window effectively reduces sidelobes and offers a gradual transition, though the main lobe is wider than the rectangular window.

The FIR filters are designed by specifying different forms for the desired impulse response hd[n] and then using the window functions to obtain realizable filter coefficients h[n].

1. **Low-Pass Filter (LPF)**:
   - The ideal impulse response for a low-pass filter is: $hd(n)=\sin(\omega c(n-\alpha))/\pi(n-\alpha), \alpha=N-1/2$
2. **High-Pass Filter (HPF)**:
   - The high-pass filter is derived by subtracting the low-pass response from a delta function: $hd(n)=\sin(\pi(n-\alpha))-\sin(\omega c(n-\alpha))/\pi(n-\alpha)$
3. **Band-Pass Filter (BPF)**:
   - For a band-pass filter with cutoff frequencies ωc1and ωc2, the ideal impulse response is: $hd(n)=\sin(\omega c2(n-\alpha))-\sin(\omega c1(n-\alpha))/\pi(n-\alpha)$
4. **Band-Stop Filter (BSF)**:
   - The band-stop filter blocks a frequency band by combining low-pass and high-pass responses: $hd(n)=\sin(\omega c(n-\alpha))-\sin(\omega c1(n-\alpha))+\sin(\pi(n-\alpha))/\pi(n-\alpha)$

Program

```
%lowpass filter

clc;

clear all;

close all;

wc=0.5*pi;

N=input("Enter the value of N:");

alpha=(N-1)/2;

eps=0.001;

n=0:1:N-1;

hd=sin(wc*(n-alpha+eps))./(pi*(n-alpha+eps));%eqn forLPF

w=0:0.01:pi;

wr=boxcar(N);%rectangular window
```

```
hn=hd.*wr';
h=freqz(hn,1,w);
figure();
subplot(3,2,1);
plot(w/pi,10*log10(abs(h)));
title("Low pass filter using rectangular window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whm=hamming(N);%hamming window
hn=hd.*whm';
h=freqz(hn,1,w);
subplot(3,2,2);
plot(w/pi,10*log10(abs(h)));
title("Low pass filter using hamming window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whn=hanning(N);%hanning window
hn=hd.*whn';
h=freqz(hn,1,w);
subplot(3,2,3);
plot(w/pi,10*log10(abs(h)));
title("Low pass filter using hanning window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
wbr=bartlett(N);%bartlett window
hn=hd.*wbr';
h=freqz(hn,1,w);
```

```
subplot(3,2,4);
plot(w/pi,10*log10(abs(h)));
title("Low pass filter using bartlett window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
subplot(3,2,5);%plotting rectangular window
stem(wr);
title("Rectangular window sequence");
xlabel("Number of samples");
ylabel("Amplitude");
grid("on");

%highpass filter
wc=0.5*pi;
N=input("Enter the value of N:");
alpha=(N-1)/2;
eps=0.001;
n=0:1:N-1;
hd=((sin(pi*(n-alpha+eps))-sin(wc*(n-alpha+eps)))./(pi*(n-
alpha+eps)));%eqn for HPF
w=0:0.01:pi;
wr=boxcar(N);%rectangular window
hn=hd.*wr';
h=freqz(hn,1,w);
subplot(3,2,1);
plot(w/pi,10*log10(abs(h)));
title("High pass filter using rectangular window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
```

```
grid("on");
whm=hamming(N);%hamming window
hn=hd.*whm';
h=freqz(hn,1,w);
subplot(3,2,2);
plot(w/pi,10*log10(abs(h)));
title("High pass filter using hamming window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whn=hanning(N);%hanning window
hn=hd.*whn';
h=freqz(hn,1,w);
subplot(3,2,3);
plot(w/pi,10*log10(abs(h)));
title("High pass filter using hanning window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
wbr=bartlett(N);%bartlett window
hn=hd.*wbr';
h=freqz(hn,1,w);
subplot(3,2,4);
plot(w/pi,10*log10(abs(h)));
title("High pass filter using bartlett window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
subplot(3,2,5);%plotting hamming window
stem(whm);
```

```
title("Hamming window sequence");
xlabel("Number of samples");
ylabel("Amplitude");
grid("on");

%band pass filter

wc=0.5*pi;
wc1=0.9*pi;
N=input("Enter the value of N:");
alpha=(N-1)/2;
eps=0.001;
n=0:1:N-1;
hd=(sin(wc1*(n-alpha+eps))-sin(wc*(n-alpha+eps)))./(pi*(n-
alpha+eps));%eqn for bandpass
w=0:0.01:pi;
wr=boxcar(N);%rectangular window
hn=hd.*wr';
h=freqz(hn,1,w);
subplot(3,2,1);
plot(w/pi,10*log10(abs(h)));
title("Band pass filter using rectangular window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whm=hamming(N);%hamming window
hn=hd.*whm';
h=freqz(hn,1,w);
subplot(3,2,2);
plot(w/pi,10*log10(abs(h)));
```

```
title("Band pass filter using hamming window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whn=hanning(N);%hanning window
hn=hd.*whn';
h=freqz(hn,1,w);
subplot(3,2,3);
plot(w/pi,10*log10(abs(h)));
title("Band pass filter using hanning window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
wbr=bartlett(N);%bartlett window
hn=hd.*wbr';
h=freqz(hn,1,w);
subplot(3,2,4);
plot(w/pi,10*log10(abs(h)));
title("Band pass filter using bartlett window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
subplot(3,2,5);%plotting hanning window
stem(whn);
title("Hanning window sequence");
xlabel("Number of samples");
ylabel("Amplitude");
grid("on");
```

```
%band stop filter

wc=0.5*pi;
wc1=0.9*pi;
N=input("Enter the value of N:");
alpha=(N-1)/2;
eps=0.001;
n=0:1:N-1;
hd=(sin(wc*(n-alpha+eps))-sin(wc1*(n-alpha+eps))+sin(pi*(n-
alpha+eps)))./(pi*(n-alpha+eps));%eqn for BSF
w=0:0.01:pi;
wr=boxcar(N);%rectangular window
hn=hd.*wr';
h=freqz(hn,1,w);
subplot(3,2,1);
plot(w/pi,10*log10(abs(h)));
title("Band stop filter using rectangular window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whm=hamming(N);%hamming window
hn=hd.*whm';
h=freqz(hn,1,w);
subplot(3,2,2);
plot(w/pi,10*log10(abs(h)));
title("Band stop filter using hamming window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
whn=hanning(N);%hanning window
```

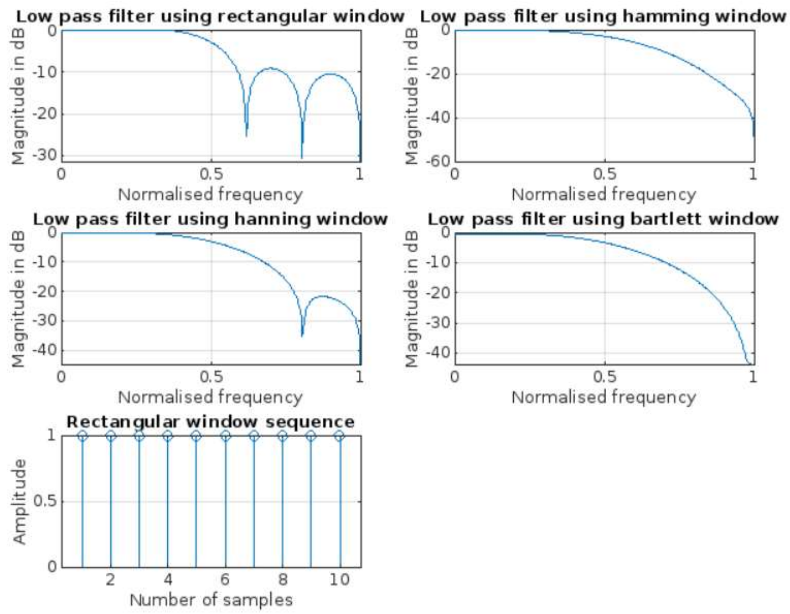
```
hn=hd.*whn';
h=freqz(hn,1,w);
subplot(3,2,3);
plot(w/pi,10*log10(abs(h)));
title("Band stop filter using hanning window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
wbr=bartlett(N);%bartlett window
hn=hd.*wbr';
h=freqz(hn,1,w);
subplot(3,2,4);
plot(w/pi,10*log10(abs(h)));
title("Band stop filter using bartlett window");
xlabel("Normalised frequency");
ylabel("Magnitude in dB");
grid("on");
subplot(3,2,5);%plotting bartlett window
stem(wbr);
title("Bartlett window sequence");
xlabel("Humber of samples");
ylabel("Amplitude");
grid("on");
```

Result

Designed and implemented FIR filter by various window method.

Observation

Low pass filter



High pass filter