ASSIGNMENT 3

PROJECT REPORT

DIGITAL SIGNAL PROCESSING AND STORAGE

By shinu shaji (c0761203)

stebin yohannan (C0770947)

Topic :- design a notch filter to eliminate 1 KHz ,2 KHz and 3 KHz frequency noises .

Step 1:

**Filter design**

a = [1 -2\*r\*cos(2\*pi\*fn/fs) r^2];

b = [1 -2\*cos(2\*pi\*fn/fs)];

b = b/sum(b)\*sum(a); %normalization of the coefficients .

* This equation is used to generate the “a” and “b” coefficient of the notch filter . where fn is the noise frequency and fs is the sampling frequency .
* Use this to find the filter coefficients for 1 KHz ,2 KHz and 3 Khz notch filters.

Step 2:

**Cascading the filter coefficents.**

A = conv(a1,a2);

B = conv(b1,b2);

* **H(n) = H(n1)\*H(n2)**
* Multiple filters can be cascaded by convolution of the filter coefficients

Step 3:

**Plot the filter response.**

[h, m] = freqz(B, A);

plot((m)/2\*pi\*fs,abs(h));

* Filter response can found using the freqz() function.
* Where A and B are the coefficients of the cascaded filter and h is the magnitude and m is in radians .
* Frequency = radians /2\*pi\*f
* Plot() function is used to plot the graph.

Step 4:

**Generate the signals 1KHz ,2KHz and 3 Khz and add it to the signal.**

fs = 8000;

t = (0:(4551-1))/fs;

noise1 = cos(2\*pi\*1000\*t);

noise2 = cos(2\*pi\*2000\*t);

noise3 = cos(2\*pi\*3000\*t);

summ = noise1+signal+noise2+noise3;

* Cos() function can be used to generate cos signal.
* fs is the sampling frequency .
* t would generate 4551 samples of the cos function .
* summ is the summation of all the signals .

step 5:

**Filter the signal.**

y2 = filter(B, A, summ);

* filter() function is used to filter a given signal .
* B and A are the filter coefficients and summ is the signal .

**Step 6:**

**Play the signals.**

**player1 = audioplayer(norm\_signal,fs,8)**

**play(player1)**

**pause(1)**

**player1 = audioplayer(summ,fs,8)**

**play(player1)**

**pause(2);**

**player2 = audioplayer(y2,fs,8)**

**play(player2)**

* Audioplayer(signal , sampling freq , bit rate) function is used to create an object to play the signal .
* Play() function is used to play the object .
* Pause(time) wait for the signal to play.

Step 7 :

**Plot the signal**

subplot(3,1,1);

plot(t\*1000\*14.05,abs(fft(signal)))

title("pure signal");

subplot(3,1,2);

plot(t\*1000\*14.05,abs(fft(summ)));

title("noise induced");

subplot(3,1,3)

plot(t\*1000\*14.05,abs(fft(y2)));

title("filtered signal")

* Fft() function is used to find the frequency response

**Full program:**

clc;clear;

%[aud,ff] = audioread("asd.ogg");

fs = 8000;

t = (0:(4551-1))/fs;

signal = cos(2\*pi\*4000\*t);

norm\_signal = (signal - min(signal)) / ( max(signal) - min(signal) )

fn1 = 1000;

fn2 = 2000;

fn3 = 3000;

noise1 = cos(2\*pi\*1000\*t);

noise2 = cos(2\*pi\*2000\*t);

noise3 = cos(2\*pi\*3000\*t);

summ = noise1+signal+noise2+noise3;

%summ = aud+noise1+noise2+noise3;

norm\_summ = (summ - min(summ)) / ( max(summ) - min(summ) )

r = 0.99;

b = [1 -2\*cos(2\*pi\*fn1/fs) 1];

a1 = [1 -2\*r\*cos(2\*pi\*fn1/fs) r^2]; % filter coefficients

b1 = b/sum(b)\*sum(a1);

%[H1, om] = freqz(b, a);

%y2 = filter(b, a, summ);

b = [1 -2\*cos(2\*pi\*fn2/fs) 1];

a2 = [1 -2\*r\*cos(2\*pi\*fn2/fs) r^2]; % filter coefficients

b2 = b/sum(b)\*sum(a2);

%[H2, om] = freqz(b, a);

%y2 = filter(b, a, y2);

b = [1 -2\*cos(2\*pi\*fn3/fs), 1];

a3 = [1 -2\*r\*cos(2\*pi\*fn3/fs) r^2]; % filter coefficients

b3 = b/sum(b)\*sum(a3);

%[H3, om] = freqz(b, a);

%y2 = filter(b, a, y2);

A = conv(a1,a2);

A = conv(A,a3);

B = conv(b1,b2);

B = conv(B,b3);

y2 = filter(B, A, summ);

[h,m] = freqz(B, A);

plot((m)/2\*pi\*fs,abs(h))

norm\_filt = (y2 - min(y2)) / ( max(y2) - min(y2) )

pause(1)

player1 = audioplayer(norm\_signal,fs,8)

play(player1)

pause(1)

player1 = audioplayer(summ,fs,8)

play(player1)

pause(2);

player2 = audioplayer(y2,fs,8)

play(player2)

##audiowrite("filered.ogg",norm\_filt,fs);

##audiowrite("summ.ogg",norm\_summ,fs);

##audiowrite("audd.ogg",signal,fs);

pause(1);

##

subplot(3,1,1);

plot(t\*1000\*14.05,abs(fft(signal)))

title("pure signal");

subplot(3,1,2);

plot(t\*1000\*14.05,abs(fft(summ)));

title("noise induced");

subplot(3,1,3)

plot(t\*1000\*14.05,abs(fft(y2)));

title("filtered signal")