NOTCH FILTER DESIGN

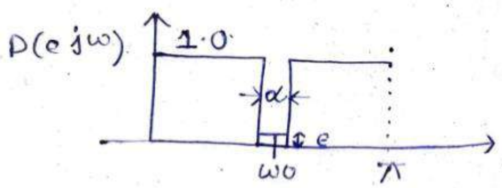
Name: Ponduru Shalini **Date:**30/10/2023

Roll No:21EE10052

Q)

Notch Filter Design:

Given the specification of Type I Linear phase Notch Filter to suppress power frequency components in an ECG signal. Design a filter of suitable order.



$T_s = 1 \times 10^{-4}$

$\alpha = 2\pi \times 1 \times T_s$

$W_{stop} = 50$

$e = 0.001$

$\omega_0 = 2\pi \times 50 \times T_s$

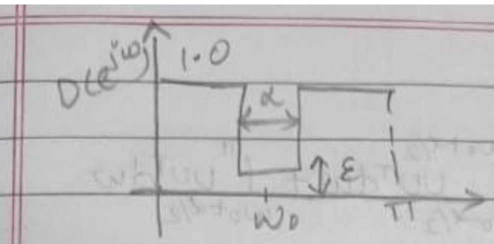
Plot the following:

(a) $h_0, h_1, \dots h_M$

(b) The ideal frequency response

(c) The actual frequency response (both angle and magnitude) i.e. $\angle H(e^{j\omega})$ and $|H(e^{j\omega})|$

Description:



$$|H(e^{jw})| = \underbrace{[\alpha_0 \ \alpha_1 \ \alpha_2 \ \cdots \ \alpha_{M/2}]}_{\alpha^T} \underbrace{\begin{bmatrix} 1 \\ \cos w \\ \cos 2w \\ \vdots \\ \cos \frac{M}{2} w \end{bmatrix}}_{V}$$

$$|H(e^{jw})| = a^T v = v^T a.$$

$$E = \int_0^{\pi} [D(e^{jw}) - |H(e^{jw})|]^2 dw$$

$$= \int_0^{w_0-\alpha/2} [1 - V^T a]^2 dw + \int_{w_0-\alpha/2}^{w_0+\alpha/2} W[\varepsilon - V^T a]^2 dw + \int_{w_0+\alpha/2}^{\pi} [1 - V^T a]^2 dw$$

$$\frac{dE}{d\alpha} = -\int_0^{w_0-\alpha/2} 2(1 - V^T a) V dw - \int_{w_0-\alpha/2}^{w_0+\alpha/2} 2W(\varepsilon - V^T a) V dw - 2\int_{w_0+\alpha/2}^{\pi} (1 - U^T a) V dw = 0$$

$$\Rightarrow \alpha\left[\int_0^{w_0-\alpha/2} UU^T dw + W\int_{w_0-\alpha/2}^{w_0+\alpha/2} UU^T dw + \int_{w_0+\alpha/2}^{\pi} UU^T dw\right]$$

$$= \int_0^{w_0-\alpha/2} U dw + \varepsilon W \int_{w_0-\alpha/2}^{w_0+\alpha/2} U dw + \int_{w_0+\alpha/2}^{\pi} U dw$$

$$Qa = P$$

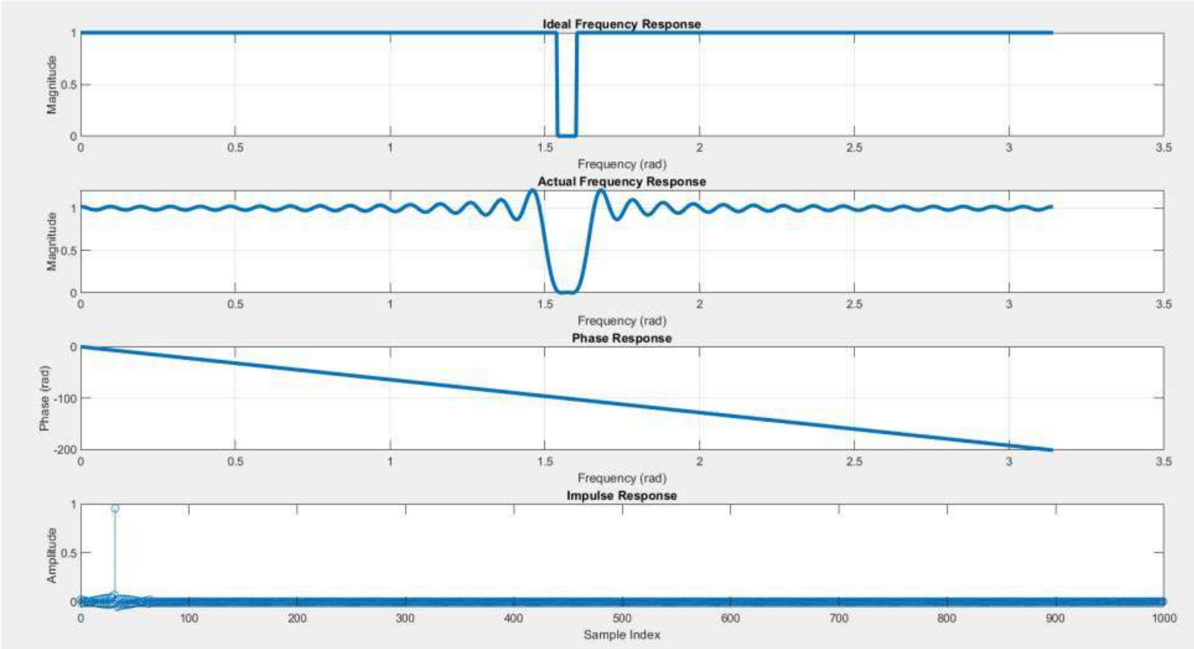
$$Q = \int_0^{w_0-\alpha/2} UU^T dw + W\int_{w_0-\alpha/2}^{w_0+\alpha/2} UU^T dw + \int_{w_0+\alpha/2}^{\pi} UU^T dw$$

Code:

```
close all
clc
global M2
% Notch Filter Design
%Parameters
M=128;
Ts=1e-04;
alpha=100*2*pi*Ts;
Wstop=50;
eps=1e-03;
wo=2*pi*2500*Ts;
M2=M/2;
L=M/2+1;
% Construct Q and p matrix
w1=linspace(0,wo-alpha/2,20000);
w2=linspace(wo-alpha/2,wo+alpha/2,20000);
w3=linspace(wo+alpha/2,pi,20000);
Q=zeros(L);p=zeros(L,1);
for n=1:length(w1)
 Q=Q+fn1(w1(n));
 p=p+fn2(w1(n));
end
for n=1:length(w2)
 Q=Q+Wstop*fn1(w2(n));
 p=p+eps*Wstop*fn2(w2(n));
end
for n=1:length(w3)
 Q=Q+fn1(w3(n));
 p=p+fn2(w3(n));
end
a=Q\p;
%Frequency response
ww=linspace(0,pi,1000)';
for k=1:length(ww)
 if ww(k)<=wo-alpha/2
 ideal(k)=1;
 elseif ww(k)<wo+alpha/2
 ideal(k)=eps;
 else
 ideal(k)=1;
 end
end
subplot(411)
plot(ww,ideal,'LineWidth',3),grid on, shg;
for k=1:length(ww)
 Hw(k)=a'*fn2(ww(k));
end
subplot(412)
plot(ww,Hw,'LineWidth',3), grid on,shg
phase=-ww*(M/2);
subplot(413)
plot(ww,phase,'LineWidth',3),grid on, shg
H=Hw.*exp(1j*phase);
hn=real(ifft(H));
N=length(hn);
n=0:(N-1);
```

```matlab
subplot(414)
stem(n,hn);close all
clc
global M2
% Notch Filter Design
%Parameters
M=128;
Ts=1e-04;
alpha=100*2*pi*Ts;
Wstop=50;
eps=1e-03;
wo=2*pi*2500*Ts;
M2=M/2;
L=M/2+1;
% Construct Q and p matrix
w1=linspace(0,wo-alpha/2,20000);
w2=linspace(wo-alpha/2,wo+alpha/2,20000);
w3=linspace(wo+alpha/2,pi,20000);
Q=zeros(L);p=zeros(L,1);
for n=1:length(w1)
 Q=Q+fn1(w1(n));
 p=p+fn2(w1(n));
end
for n=1:length(w2)
 Q=Q+Wstop*fn1(w2(n));
 p=p+eps*Wstop*fn2(w2(n));
end
for n=1:length(w3)
 Q=Q+fn1(w3(n));
 p=p+fn2(w3(n));
end
a=Q\p;
%Frequency response
ww=linspace(0,pi,1000)';
for k=1:length(ww)
 if ww(k)<=wo-alpha/2
 ideal(k)=1;
 elseif ww(k)<wo+alpha/2
 ideal(k)=eps;
 else
 ideal(k)=1;
 end
end
subplot(411)
plot(ww,ideal,'LineWidth',3),grid on, shg;
for k=1:length(ww)
 Hw(k)=a'*fn2(ww(k));
end
subplot(412)
plot(ww,Hw,'LineWidth',3), grid on,shg
phase=-ww*(M/2);
subplot(413)
plot(ww,phase,'LineWidth',3),grid on, shg
H=Hw.*exp(1j*phase);
hn=real(ifft(H));
N=length(hn);
n=0:(N-1);
subplot(414)
stem(n,hn);
```

Plot:



Discussion:

The MATLAB code I'm analyzing is all about designing and evaluating a notch filter. It begins by cleaning up the workspace and initializing a global variable, `M2`. The parameters of the notch filter are then set. These parameters include the filter's length (`M`), the sampling period (`Ts`), notch filter characteristics like `alpha` and `Wstop`, and the center frequency of the notch (`wo`). After parameter initialization, the code creates matrices `Q` and `p` to be used in the filter design process. The frequency range from 0 to π is divided into three segments (`w1`, `w2`, and `w3`), and these matrices are populated by calling two functions: `fn1(w)` and `fn2(w)`. The heart of the code is solving a linear system (`Q * a = p`) to determine the filter coefficients. These coefficients represent the frequency response of the notch filter. To understand how well the filter design meets the desired specifications, the code calculates the ideal frequency response based on the specified notch characteristics (`wo`, `alpha`, and `eps`). It then plots the ideal and actual frequency responses, allowing us to visually assess the filter's performance. The code also computes and plots the phase response, which shows how the phase of the signal changes with frequency. Furthermore, it calculates the impulse response of the filter by performing the inverse Fourier transform of the product of the frequency response and the phase response. The impulse response is then plotted, giving insights into how the filter behaves in the time domain. In summary, this code serves as a practical example of designing and evaluating a notch filter in MATLAB. It demonstrates various aspects of filter design and analysis, making it a valuable resource for those working on digital signal processing applications