[ARTEM DUDKO, NIKOLAS TAPANAINEN, BRANDON KEEFE] - [MATLAB Project 2] - [2021]

Table of Contents

[PROBLEM #1] 1
3.a 1
3.b on paper 2
3.c 2
3.d 3
3.e 6
3.f 7
3.g 8

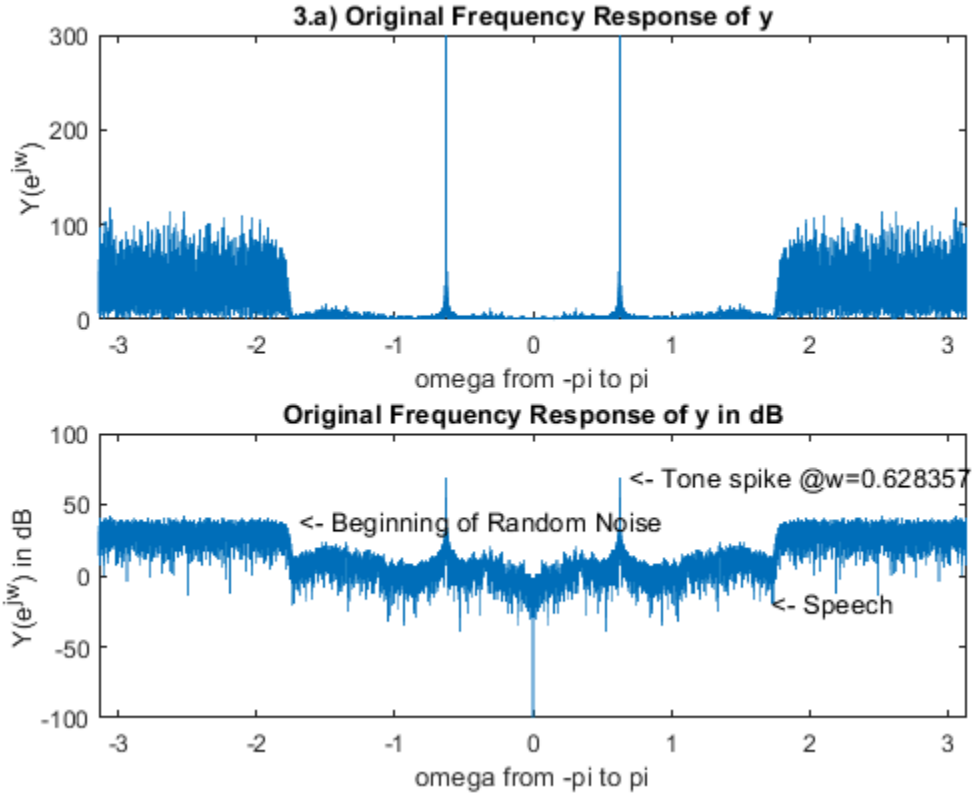
[PROBLEM #1]

```
clc;

%Tutorial problems:
a1tut = [1 -0.8];
b1tut = [2 0 -1];
Ntut = 40;
[H1, omega1] = freqz(b1tut,a1tut,Ntut);


a2tut = [1 -0.8];
b2tut = [2 0 -1];
Ntut = 40;
[H2, omega2] = freqz(b2tut,a2tut,Ntut, 'whole');
```

3.a

```
%%student ID sum = 1
load group1.mat;


Nfft = 2^ceil(log2(length(y)));
Y = fftshift(fft(y,Nfft));    %compute Y(e^jw)
omega5 = (0:(Nfft-1)).*(2*pi/Nfft)-pi;
Y_DB = 20 .* log10(abs(Y));  %compute dB

figure(1)

subplot(2,1,1)
plot(omega5,abs(Y));
xlim([-pi pi])
```

```
ylim([0 300])
title("3.a) Original Frequency Response of y")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w)")

subplot(2,1,2)
plot(omega5,Y_DB);
xlim([-pi pi])
title("Original Frequency Response of y in dB")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w) in dB")
ylim([-100 100])
text(0.7,70,"<- Tone spike @w=0.628357")
text(-1.7,40,"<- Beginning of Random Noise")
text(1.73,-18,"<- Speech")
```

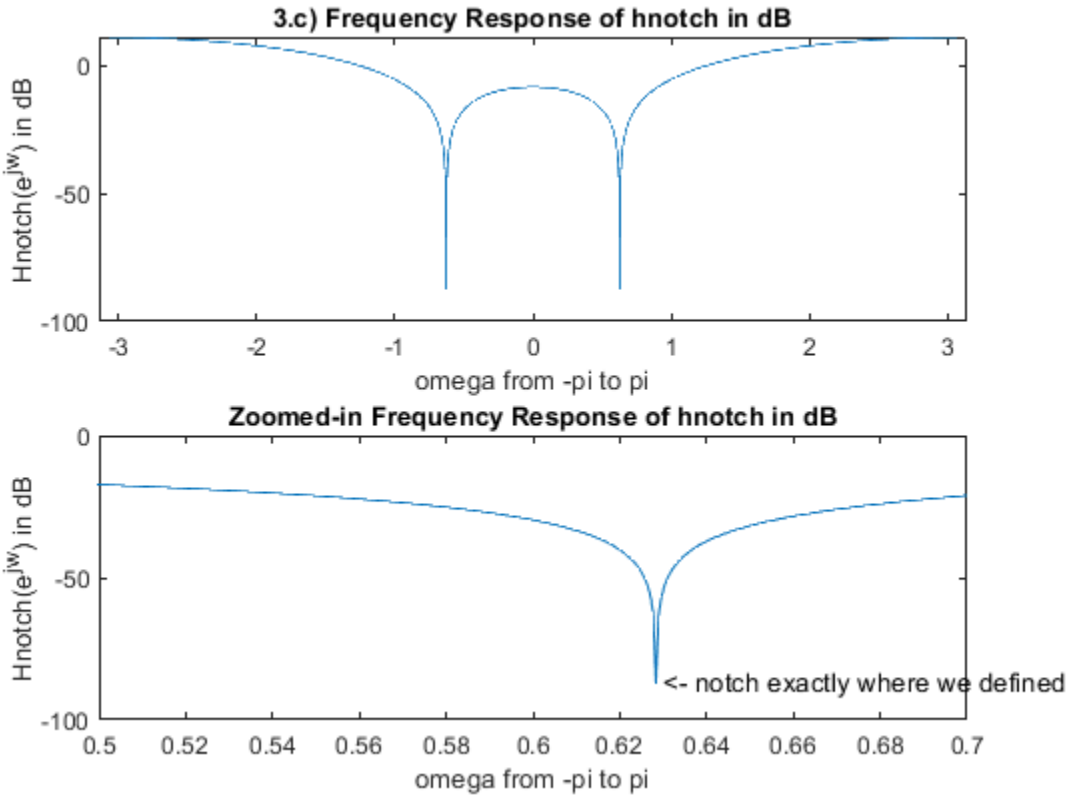


3.b on paper

3.c

```
%impulse response, and its dB form.
hnotch = [1,-2.*cos(0.628357),1];  %w plugged in from part a
[H_3a, omega3] = freqz(hnotch,1,10000,'whole');
H_3a_DB = 20 .* log10(abs(H_3a));
```

```
H_3a_DB_pos = H_3a_DB(1:5000); %the graph of H(e^jw) is from 0 to 2pi,
H_3a_DB_neg = H_3a_DB(5001:10000);  %this makes it -pi to pi
H_3a_DB_shifted = [H_3a_DB_neg; H_3a_DB_pos];
shift = ones(10000,1).*-pi;
omega3shift = omega3 + shift;

%resulting w is very close to part a, w = 0.628319
figure(2)
subplot(2,1,1)
plot(omega3shift,H_3a_DB_shifted);
xlim([-pi pi])
title("3.c) Frequency Response of hnotch in dB")
xlabel("omega from -pi to pi")
ylabel("Hnotch(e^j^w) in dB")

subplot(2,1,2)
plot(omega3shift,H_3a_DB_shifted);
xlim([0.5 0.7])
title("Zoomed-in Frequency Response of hnotch in dB")
xlabel("omega from -pi to pi")
ylabel("Hnotch(e^j^w) in dB")
text(0.63,-86,"<- notch exactly where we defined")
```

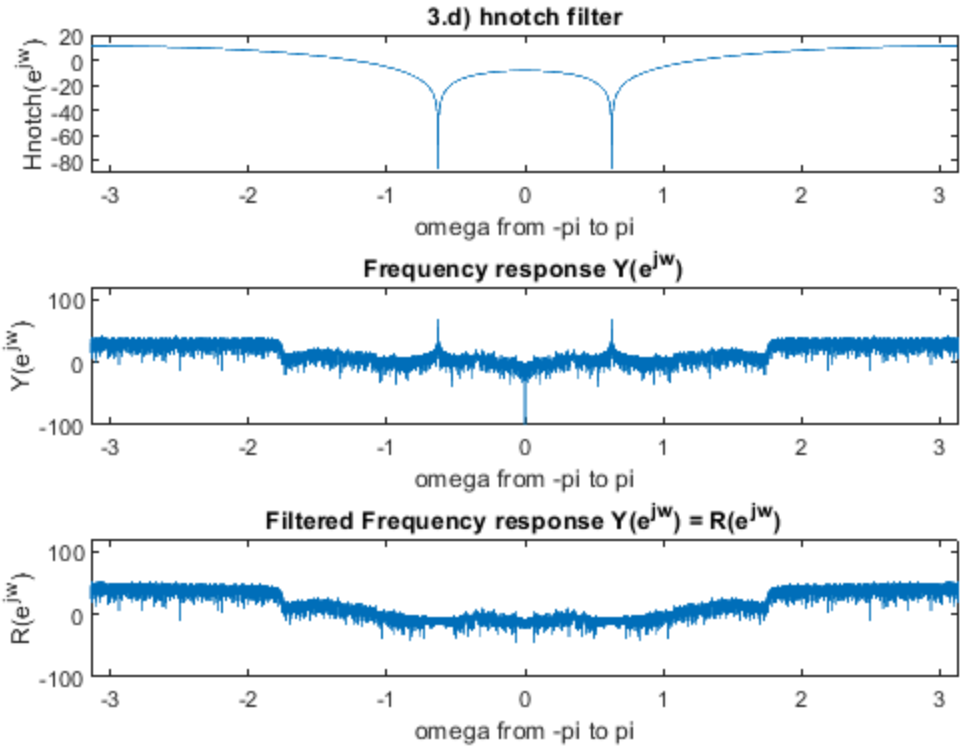


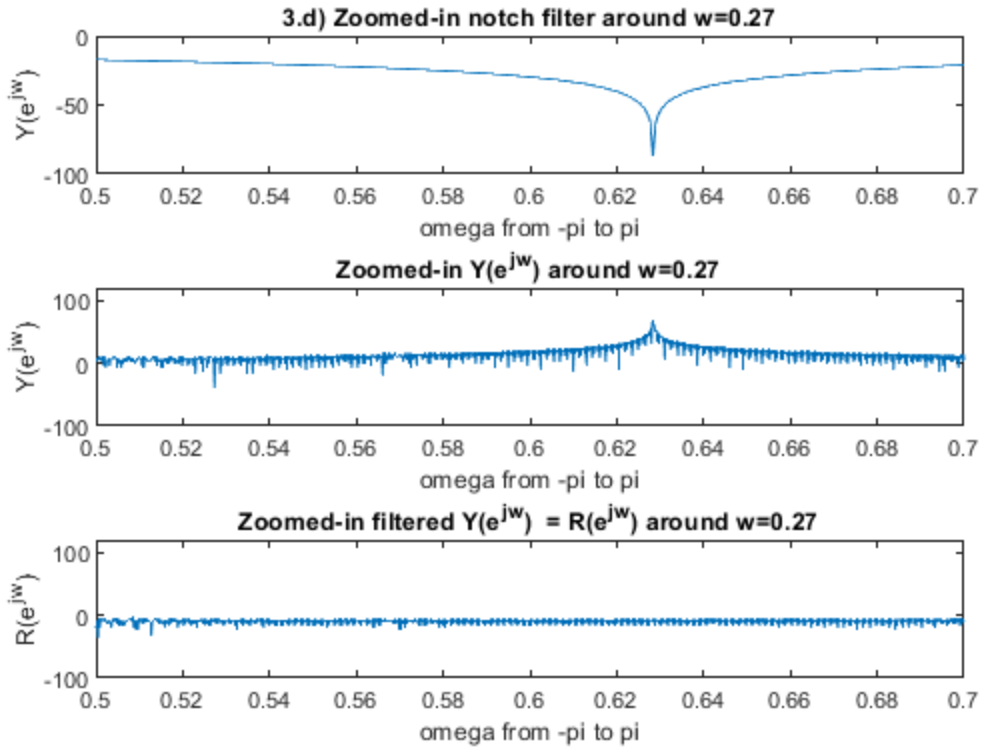
3.d

```
fft_hnotch = fft(hnotch,length(y));  %frequency transform of hnotch
```

```
fft_hnotch_DB = 20 .* log10(abs(fft_hnotch));

r = filter(hnotch,1,y);  %r is filtered y with no notches
R = fftshift(fft(r,Nfft));
R_DB = 20 .* log10(abs(R));


figure(3)   %filter response in dB
subplot(3,1,1)
plot(omega3shift,H_3a_DB_shifted);
xlim([-pi pi])
title("3.d) hnotch filter")
xlabel("omega from -pi to pi")
ylabel("Hnotch(e^j^w)")
ylim([-90 20])

subplot(3,1,2)   %Original frequency response in dB
plot(omega5,Y_DB);
xlim([-pi pi])
ylim([-100 120])
title("Frequency response Y(e^j^w)")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w)")

subplot(3,1,3)   %filtered y frequency in dB, the tone is gone
plot(omega5,R_DB);
xlim([-pi pi])
ylim([-100 120])
title("Filtered Frequency response Y(e^j^w) = R(e^j^w)")
xlabel("omega from -pi to pi")
ylabel("R(e^j^w)")

figure(4)   %figure 4 is a zoomed in version of figure 3

subplot(3,1,1)
plot(omega3shift,H_3a_DB_shifted);
xlim([0.5 0.7])
title("3.d) Zoomed-in notch filter around w=0.27")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w)")

subplot(3,1,2)
plot(omega5,Y_DB);
xlim([-pi pi])
xlim([0.5 0.7])
ylim([-100 120])
title("Zoomed-in Y(e^j^w) around w=0.27")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w)")

subplot(3,1,3)
plot(omega5,R_DB);
xlim([0.5 0.7])
ylim([-100 120])
```

```
title("Zoomed-in filtered Y(e^j^w)  = R(e^j^w) around w=0.27")
xlabel("omega from -pi to pi")
ylabel("R(e^j^w)")
```

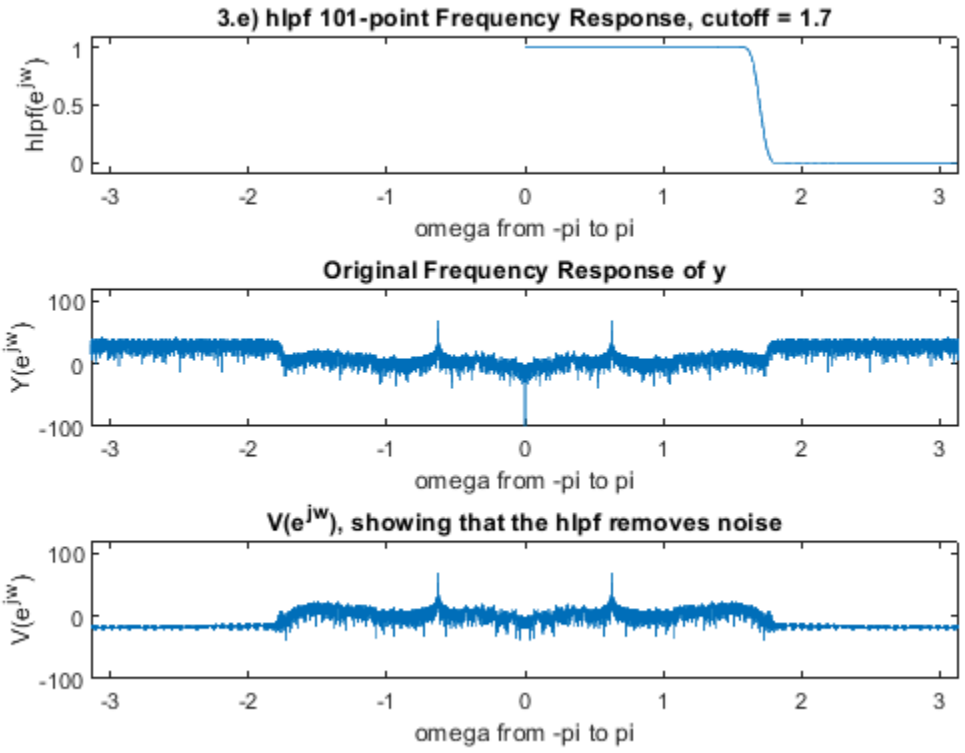




3.e

```
alpha = 1.7./pi;  %cutoff of 1.7
hlpf = fir1(101,alpha);

[lowpass,omega6] = freqz(hlpf,1,512);  %frequency response of the
 lowpass filter

v = filter(hlpf,1,y);    %filtering the original signal y with hlpf to
 get v
V = fftshift(fft(v,Nfft));  %Freq response of filtered signal v
V_DB = 20 .* log10(abs(V));


figure(5)

subplot(311)
plot(omega6,abs(lowpass));
xlim([-pi pi])
title("3.e) hlpf 101-point Frequency Response, cutoff = 1.7")
xlabel("omega from -pi to pi")
ylabel("hlpf(e^j^w)")
ylim([-0.1 1.1])

subplot(312)
```

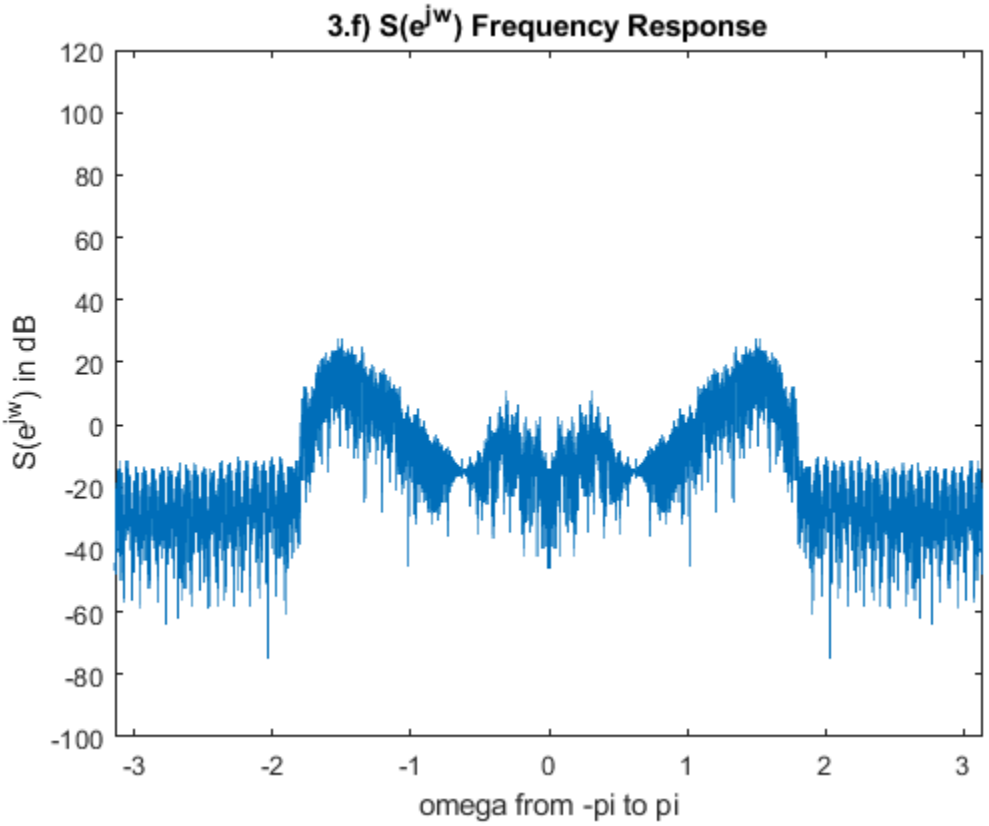
```
plot(omega5,Y_DB);
xlim([-pi pi])
ylim([-100 120])
title("Original Frequency Response of y")
xlabel("omega from -pi to pi")
ylabel("Y(e^j^w)")

subplot(313)
plot(omega5,V_DB);
xlim([-pi pi])
ylim([-100 120])
title("V(e^j^w), showing that the hlpf removes noise")
xlabel("omega from -pi to pi")
ylabel("V(e^j^w)")
```



3.f

```
s = filter(hlpf,1,r);   %s is defined as the filter of r using the
 lowpass
S = fftshift(fft(s,Nfft));
S_DB = 20 .* log10(abs(S));

%speech can be heard from end of script, but both the tone and noise
 are
%gone from s
```

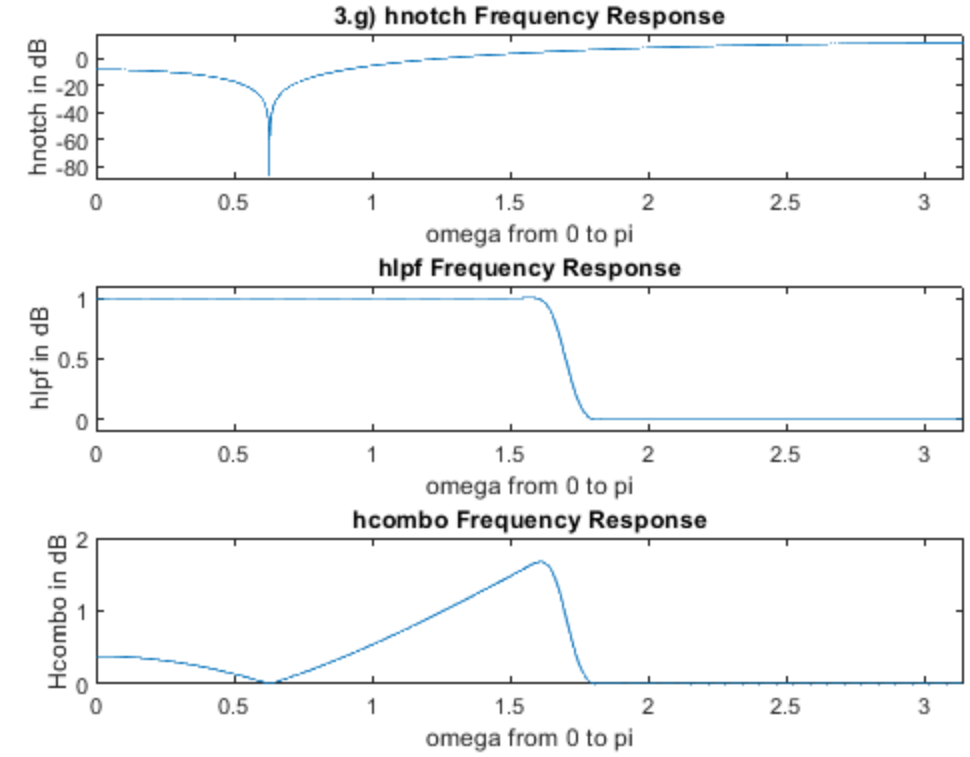
```
figure(6)

plot(omega5,S_DB);    %S(ejw) shows that both the tone and noise are
 removed from s[n]
xlim([-pi pi])
ylim([-100 120])
title("3.f) S(e^j^w) Frequency Response")
xlabel("omega from -pi to pi")
ylabel("S(e^j^w) in dB")
```

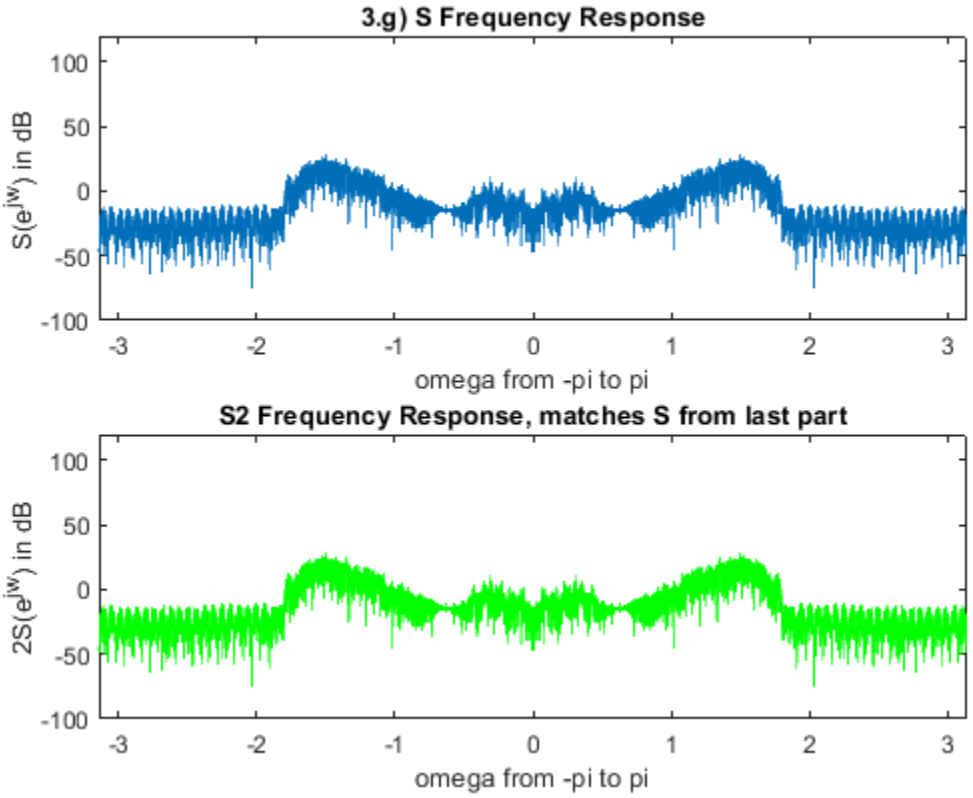


3.g

```
h_combo = conv(hlpf,hnotch);  %h_combo defined as hlpf convolved with
 hnotch
[H_combo,omega7] = freqz(h_combo,1,512);


figure(7)    %this frequncy response looks like a multiplication of
 the original two signals in freq

subplot(311)
plot(omega3shift,H_3a_DB_shifted);
xlim([0 pi])
title("3.g) hnotch Frequency Response")
xlabel("omega from 0 to pi")
ylabel("hnotch in dB")
```

```
ylim([-90 18])

subplot(312)
plot(omega6,abs(lowpass));
xlim([0 pi])
ylim([-0.1 1.1])
title("hlpf Frequency Response")
xlabel("omega from 0 to pi")
ylabel("hlpf in dB")

subplot(313)
plot(omega7,abs(H_combo));
xlim([0 pi])
title("hcombo Frequency Response")
xlabel("omega from 0 to pi")
ylabel("Hcombo in dB")


s2 = filter(h_combo,1,y);  %S2 looks the same as S, so their time
 version are the same too: s = s2
S2 = fftshift(fft(s2,Nfft));
S2_DB = 20 .* log10(abs(S2));


figure(8)  %S vs S2 methods

subplot(211)
plot(omega5,S_DB);
xlim([-pi pi])
ylim([-100 120])
title("3.g) S Frequency Response")
xlabel("omega from -pi to pi")
ylabel("S(e^j^w) in dB")

subplot(212)
plot(omega5,S2_DB,'g');
title("S2 Frequency Response, matches S from last part")
xlabel("omega from -pi to pi")
ylabel("2S(e^j^w) in dB")
xlim([-pi pi])
ylim([-100 120])


%%uncomment to listen
%soundsc(y,10000)  %original signal
%soundsc(v,10000)  %v is random noise free
%soundsc(r,10000)  %r is tone free
%soundsc(s,10000)  %s is tone and random oise free
%soundsc(s2,10000) %s2 is the same as s, but attained using hcombo
```


3.g) hnotch Frequency Response
hnotch in dB
omega from 0 to pi
hlpf Frequency Response
hlpf in dB
omega from 0 to pi
hcombo Frequency Response
Hcombo in dB
omega from 0 to pi


3.g) S Frequency Response
S(e^jw) in dB
omega from -pi to pi
S2 Frequency Response, matches S from last part
2S(e^jw) in dB
omega from -pi to pi

Published with MATLAB® R2021a