Joel Marte, Sarnoh Lepolu
12/8/2020
ECE 320

MATLAB Project 2

Main Body

1) Freqz Tutorial

   Before getting the frequency response, we specified the difference equation with the variables a and b(coefficients). The freqz function ([H omega]=freqz(b,a,N)) takes in these coefficients and returns to us H (value of frequency response) and omega (frequency in Wk from 0 to pi). With the additional argument 'whole' in the freqz function([H omega]=freqz(b,a,N, 'whole')), frequency sample size goes from 0 to 2pi. In our case, we wanted 4 evenly spaced frequencies so N = 4 and as promised we get four values of H and omega.

2) Filtering Speech Signals

   (a) After loading the y[n] DT signal, we ran soundsc(y, fs) -- fs being the base frequency of the speech which is 10,000Hz. There was a lot of noise and a high pitch tone in the signal, making it hard to hear the speech. The following code was used to compute the DTFT of signal y[n] and generate the plots for the magnitude |Y(e^j*omega)|:

```
% set fft size to be the power of 2 greater than the signal length
Nfft = 2^ceil(log2(length(y)));
% fftshift reorders the values of the DTFT to go from -pi to pi
Y =fftshift(fft(y, Nfft));
% set appropriate omega samples for X from -pi to pi
omegaX = (0:(Nfft-1))*(2*pi/Nfft)-pi;
```

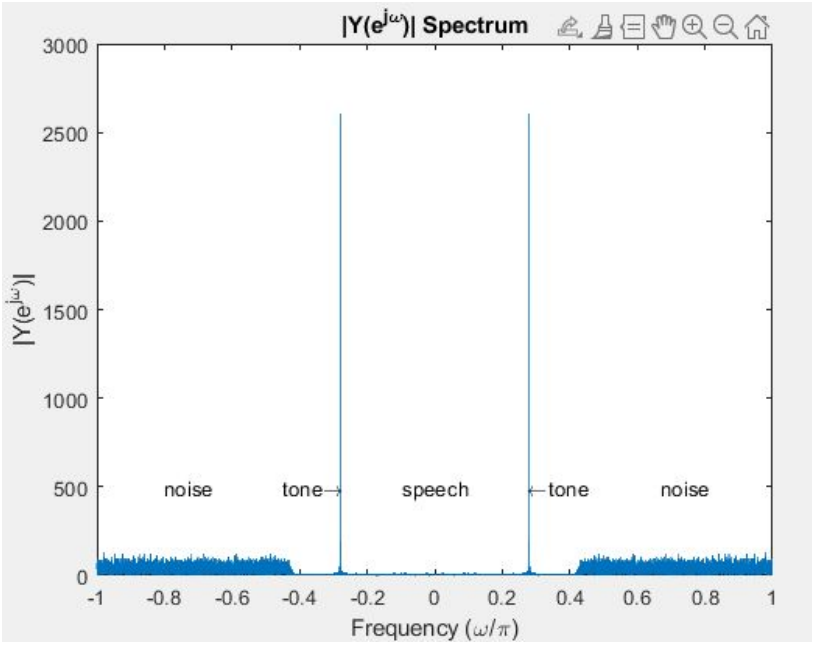

Figure. 1: Magnitude of Y(e^j*omega) Spectrum

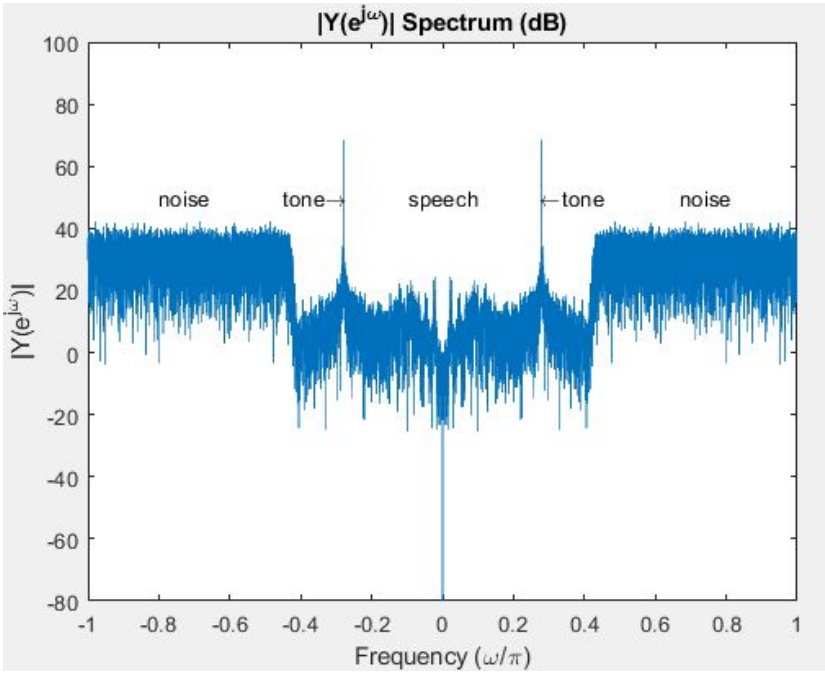

Figure. 2: Magnitude of Y(e^j*omega) Spectrum in dB

(b) A Notch Filter is also known as a band stop filter. It is designed to remove specific interfering signals called the stop band frequency. Figure 3 shows the solution of an impulse response, which gain of zero at, If w = +-wn then H(ejw)=0

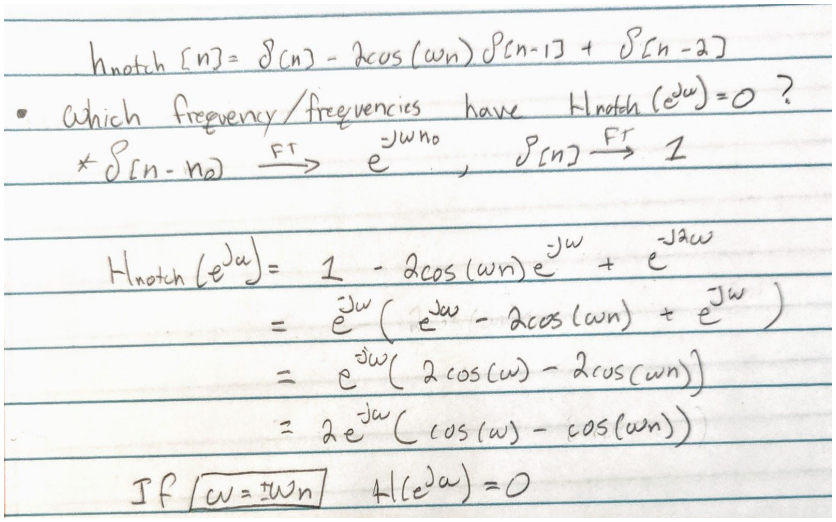

Figure. 3

(c) To get the specific frequency removed, we manually zoomed in to where the tone was on the Magnitude of Y(e^j*omega). Similarly to the freqz tutorial, we used the freqz function to compute the frequency response Hnotch and generate its plot. Note that Hnotch has a gain of zero at approximately the same frequency shown in figure 4.

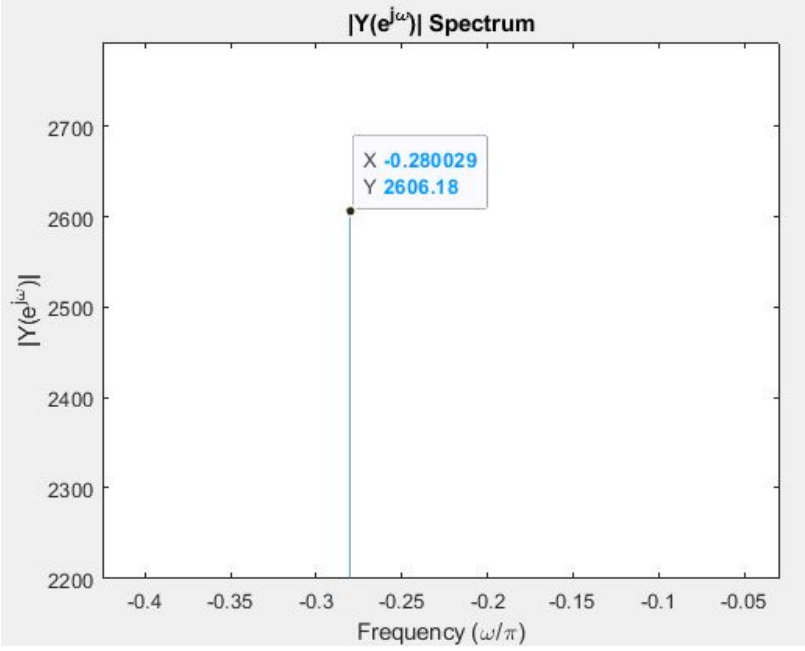

Figure.4: Manual Zoom-in of Tone

```
Wn=0.280029*pi;
hnotch = [1 -2*cos(Wn) 1];
[Hnotch, omegaHnotch] = freqz(hnotch, 1);
```

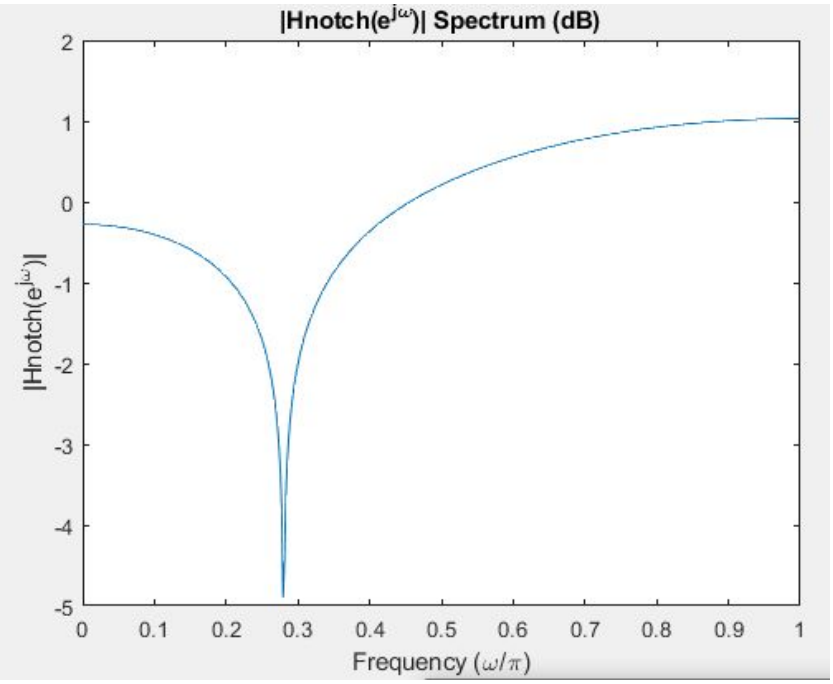

Figure. 5: Magnitude of Hnotch Spectrum in dB

(d) Although our Hnotch spectrum (dB) is not perfectly flat, This command runs y[n] signal through notch filter to remove tone, and it does so successfully:

```
r = filter(hnotch, 1, y);
soundsc(r, fs)
```

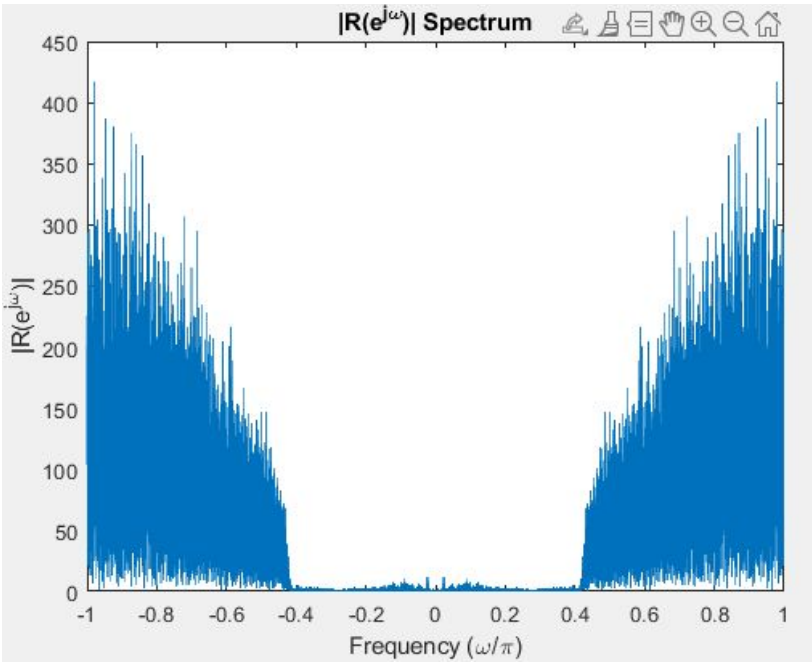

Figure. 6: Magnitude of R(e^j*omega) Spectrum with no Tone

(e) Looking at the figure 6, we concluded that the cutoff frequency would have to be approximately above frequency value 0.4. This cutoff frequency should be able to remove all of that noise in the signal y[n]. The following code computes what we have described:

```
alpha = 0.416809; % cut-off frequency to remove noise based of Fig.1
hlpf = fir1(100, alpha);
```

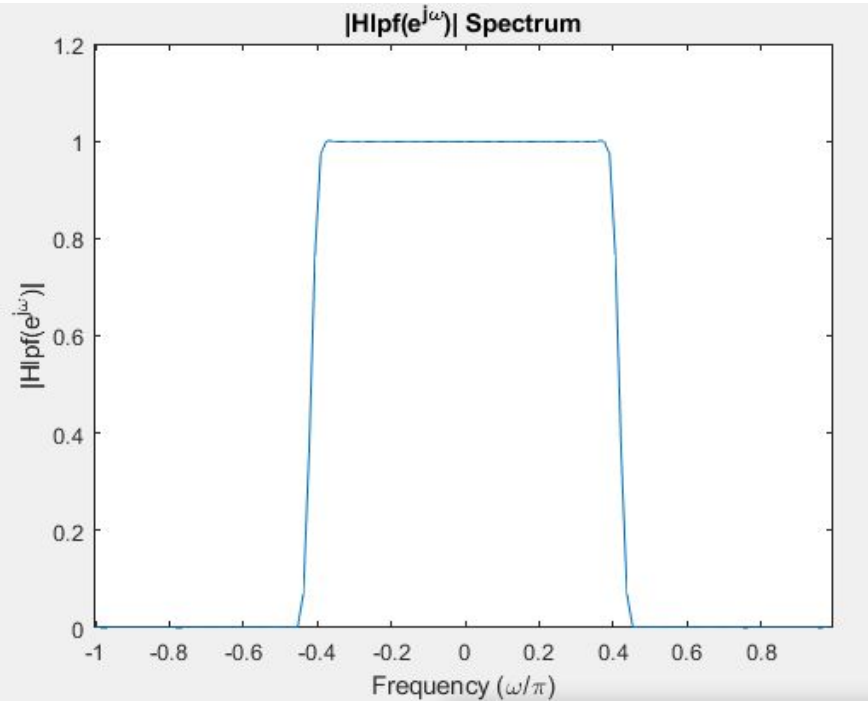

Figure 7: Lowpass Filter to Remove Noise

(f) The following command filters signal r with impulse response hlpf with output s. The speech is easier to understand, it says, “It was the best of times, it was the worst of times.” I do still hear a faint noise in the background. Can we use a lower frequency to remove the faint noise that we still hear in the background without interfering with our output.

s = filter(hlpf, 1, r);

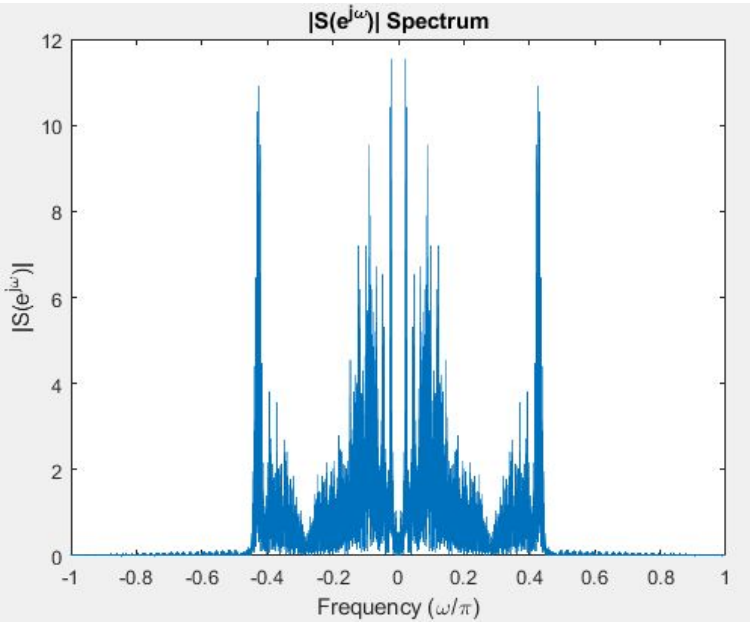

Figure. 8: Filtered Speech Signal Without Noise or Tone

(g) Due to the linearity property,we can convolve both of our filters and reap the benefits of both. Variable hcombo will have the features of both filters now. The following code will remove the noise and the tone in a single filtering operation and then filters signal y with hcombo.

```
hcombo = conv(hlpf, hnotch);
s2 = filter(hcombo, 1, y);
```

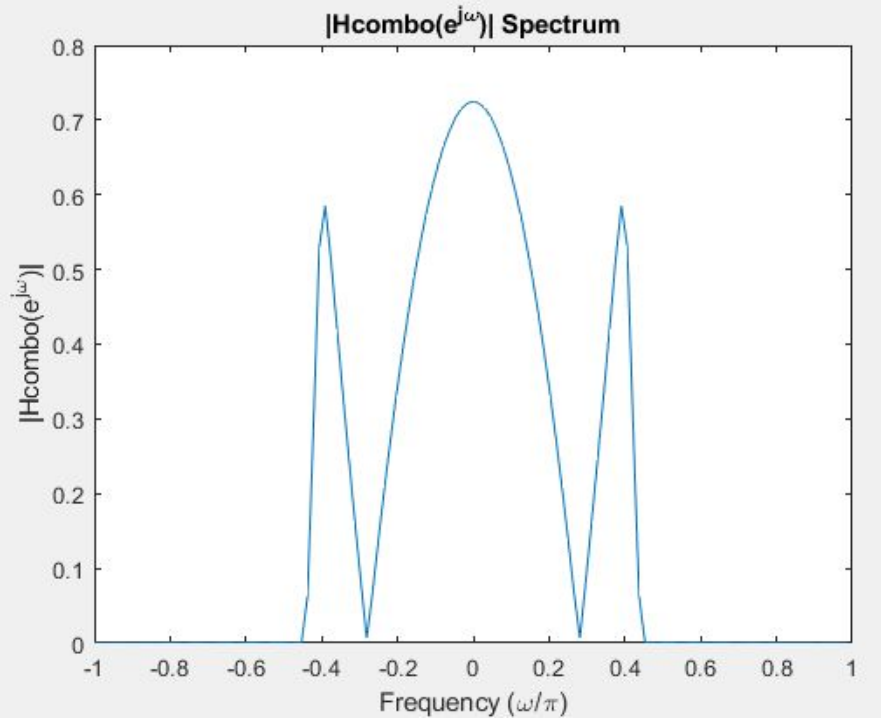

Figure. 9: Frequency Response of hlpf and hnotch Filters Convolved

Breakdown of Individual Contributions

Joel - Sarnoh and I contributed to writing the code for this project. I wrote most of the report.
Sarnoh - Project code was contributed by both Joel & I as well as the report. Joel contributed to most of the report.

References

[1] “Add text descriptions to data points - MATLAB text.” [Online]. Available: https://www.mathworks.com/help/matlab/ref/text.html. [Accessed: 05-Dec-2020].

[2] “Load variables from file into workspace - MATLAB load.” [Online]. Available: https://www.mathworks.com/help/matlab/ref/load.html. [Accessed: 05-Dec-2020].

[3] “DFT Example: Rectangular Pulse - YouTube”[Online]. Available: https://www.youtube.com/watch?v=pUQIHdz0YiE [Accessed: 05-Dec-2020].

[4] “DFT Example: Rectangular Pulse - YouTube” [Online]. Available: https://www.youtube.com/watch?v=WqF2j-HdGl0&feature=emb_logo [Accessed: 05-Dec-2020].

[5] “Butterworth Filter Design in Matlab - YouTube” [Online]. Available: https://www.youtube.com/watch?v=7H0KdeawwBc&feature=emb_logo [Accessed: 05-Dec-2020].

[6] “Designing Chebyshev Type 2 Filters in Matlab - YouTube” [Online]. Available: https://www.youtube.com/watch?v=dFoZcQ1T0I4&feature=emb_logo [Accessed: 05-Dec-2020].

[7] “Recording for 30 November Office Hours” [Online]. Available: https://umassd.zoom.us/rec/play/getAvFu1K6CPIrZetSlLTRbMvyGu-OAmh6wEYgLs1LDlnRTtRx2QbcO9lu19Aw5_jaC14vrZhmfyvTmM.kIcIND4yWQps_oLQ?continueMode=true&_x_zm_rtaid=dRPTrFuZQiiG1QzEWvBRYw.1607359975561.91f91ed18a5f38d715fcd28810ecc242&_x_zm_rhtaid=367 [Accessed: 06-Dec-2020].

Appendix

Matlab Script

```
% Joel Marte
% Sarnoh W Lepolu
% ECE320
% MATLAB Projet 2

% 3.2 tutorial: freqz
% (2a)
clc, clear;
a = [1, -.8];
b = [2 0 -1];

% (2b)
n1 = 4;
n2 = 4;

[H1, omega1] = freqz(b, a, n1);
[H2, omega2] = freqz(b, a, n2, 'whole');
H1; omega1;
% (2c)
H2;
omega2;

%%% (a)
load('group9.mat')
% soundsc(y, 10000)
fs = 10000;
% set fft size to be the power of 2 greater than the signal length
Nfft = 2^ceil(log2(length(y)));
% fftshift reorders the values of the DTFT to go from -pi to pi
Y =fftshift(fft(y, Nfft));
% set appropriate omega samples for X from -pi to pi
omegaX = (0:(Nfft-1))*(2*pi/Nfft)-pi;

figure(1)
clf
% plot magnitude samples vs frequency normalized by pi
plot(omegaX/pi, abs(Y));
```

```
noise = text(-.8, 500, 'noise'); noise2 = text(.67, 500, 'noise');
speech = text(-.095, 500, 'speech');
tone = text(-.45, 500, 'tone\rightarrow'); tone2 = text(.27, 500, '\leftarrowtone');
xlabel('Frequency (\omega/\pi)')
ylabel('|Y(e^{j\omega})|')
title('|Y(e^{j\omega})| Spectrum')

figure(2)
plot(omegaX/pi, 20*log10(abs(Y)));
dBnoise = text(-.8, 50, 'noise'); dBnoise2 = text(.67, 50, 'noise');
dBspeech = text(-.095, 50, 'speech');
dBtone = text(-.45, 50, 'tone\rightarrow'); dBtone2 = text(.27, 50, '\leftarrowtone');
xlabel('Frequency (\omega/\pi)')
ylabel('|Y(e^{j\omega})|')
title('|Y(e^{j\omega})| Spectrum (dB)')
ylim([-80 100])




%%% (c)

Wn=0.280029*pi;
hnotch = [1 -2*cos(Wn) 1];
[Hnotch, omegaHnotch] = freqz(hnotch, 1);

% frequency response of notch filter

figure(3)
plot(omegaHnotch/pi, 2*log10(abs(Hnotch)));
xlabel('Frequency (\omega/\pi)')
ylabel('|Hnotch(e^{j\omega})|')
title('|Hnotch(e^{j\omega})| Spectrum (dB)')

%%% (d)

% run y signal through notch filter to remove tone
r = filter(hnotch, 1, y);
% soundsc(r, fs)
% The tone is comepletely gone
```

```
NfftR = 2^ceil(log2(length(r)));

R =fftshift(fft(r, NfftR));
omegaR = (0:(NfftR-1))*(2*pi/NfftR)-pi;



figure(4)
plot(omegaR/pi, abs(R));
xlabel('Frequency (\omega/\pi)')
ylabel('|R(e^{j\omega})|')
title('|R(e^{j\omega})| Spectrum')

%%% (e)
alpha = 0.416809; % cut-off frequency to remove noise based of Fig.1
hlpf = fir1(100, alpha);

NfftHlpf = 2^ceil(log2(length(hlpf)));

Hlpf =fftshift(fft(hlpf, NfftHlpf));

omegaHlpf = (0:(NfftHlpf-1))*(2*pi/NfftHlpf)-pi;


figure(5)
plot(omegaHlpf/pi, abs(Hlpf));
xlabel('Frequency (\omega/\pi)')
ylabel('|Hlpf(e^{j\omega})|')
title('|Hlpf(e^{j\omega})| Spectrum')

%%% (f)
s = filter(hlpf, 1, r);
soundsc(s, fs)
% The voice says, "It was the best of times, it was the worst of times"
% I do still hear a faint noise in the backgroud

NfftS = 2^ceil(log2(length(s)));

S =fftshift(fft(s, NfftS));
```

```
omegaS = (0:(NfftS-1))*(2*pi/NfftS)-pi;



figure(6)
plot(omegaS/pi, abs(S));
xlabel('Frequency (\omega/\pi)')
ylabel('|S(e^{j\omega})|')
title('|S(e^{j\omega})| Spectrum')

% (g)
hcombo = conv(hlpf, hnotch);

NfftHcombo = 2^ceil(log2(length(hcombo)));

Hcombo = fftshift(fft(hcombo, NfftHcombo));
omegaHcombo = (0:(NfftHcombo-1))*(2*pi/NfftHcombo)-pi;

figure(7)
plot(omegaHcombo/pi, abs(Hcombo));
xlabel('Frequency (\omega/\pi)')
ylabel('|Hcombo(e^{j\omega})|')
title('|Hcombo(e^{j\omega})| Spectrum')

s2 = filter(hcombo, 1, y);
% soundsc(s2, fs)

% Yes, s2 removed the tone and the noise. It sounds just like the s.
```