Mini-Project Report
Department of Electronic Engineering
The Chinese University of Hong Kong

Course: ELEG4501

Name: Su, Shang-Yen

SID: 1155032058

Part A: Create and observe noisy signal

A.1. In Fig. 1, the first plot result is x[n]; and the second plot result is v[n]. To generate y[n] (the 3rd plot result), I added x[n] and v[n] directly.

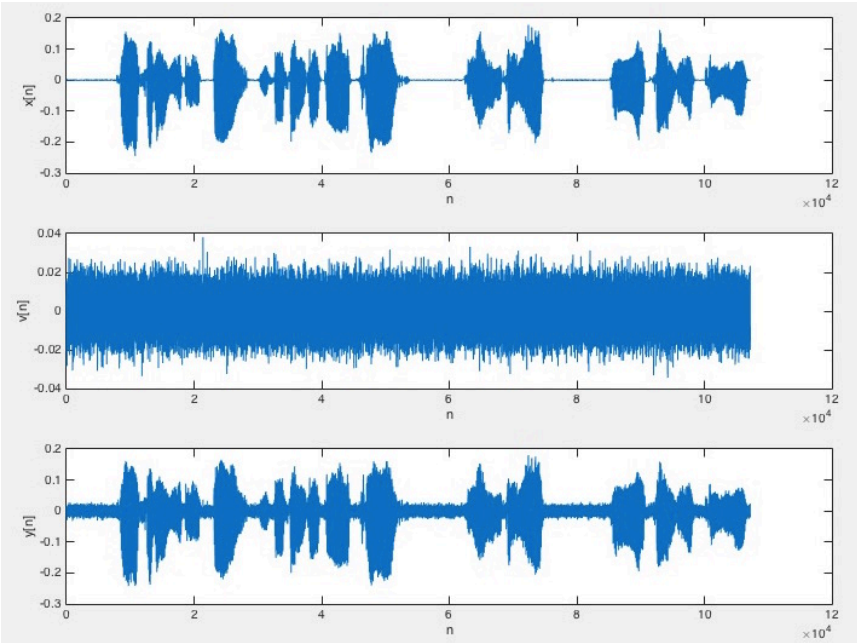

Fig. 1. x[n] (upper), v[n] (middle), y[n] (lower)

A.2. There are 107200 units of data for each of x[n] and y[n], and the length for each is 7 seconds. To extract a short-time frame from 0.8 sec to 0.83 sec, we cut the data from 16000*0.8 to 16000*0.83. Then, we calculate the log-magnitude spectra by formula 20*log(abs(fft(x[n]))) (The result is shown as Fig. 2).

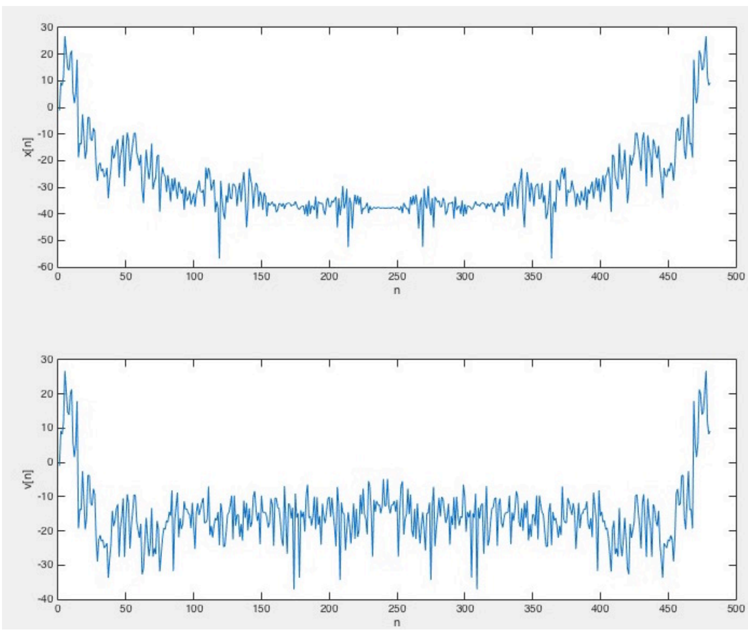

Fig. 2. Log-magnitude spectra results for x[n] (upper), y[n] (lower)

To compare these two spectra results in Fig. 2, we can find that the noisy signal change more strongly.

Part B: Calculate spectrogram

B.1. There are two factor in for using function “spectrogram”: x, window, noverlap, and nfft. x is voice data; window is the length of window; noverlap is the length of window multiple the frame overlap ratel; nfft is FFT length. The spectrogram results are shown from Fig. 3 to Fig. 6.

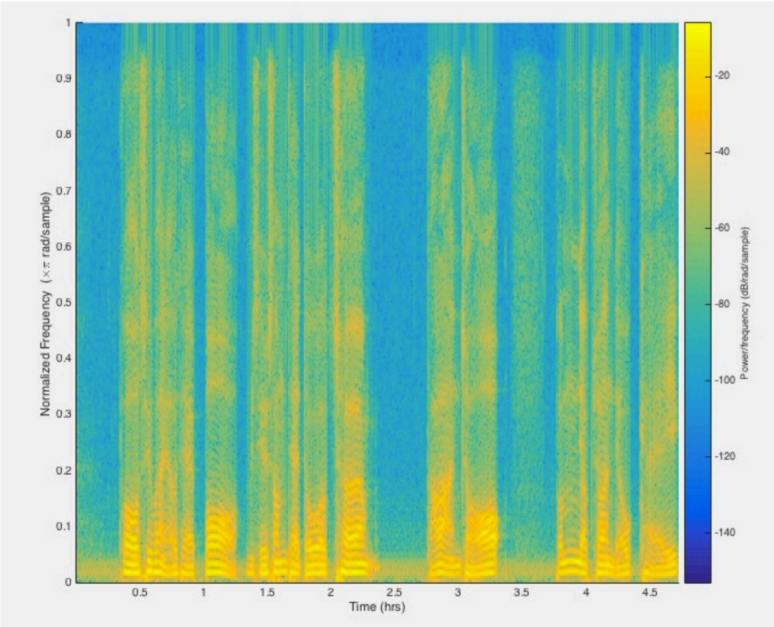

Fig. 3. Spectrogram result for x[n] (window = 500)

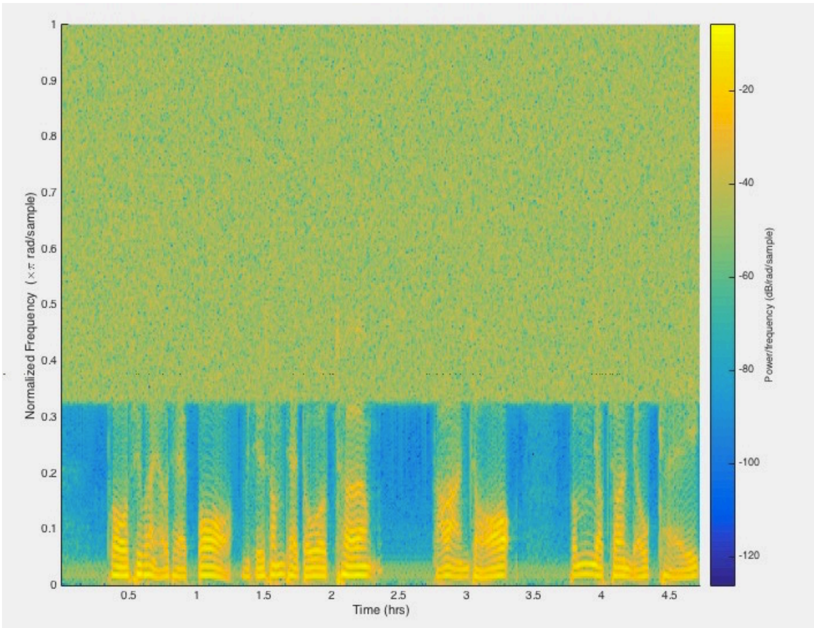

Fig. 4. Spectrogram result for y[n] (window = 500)

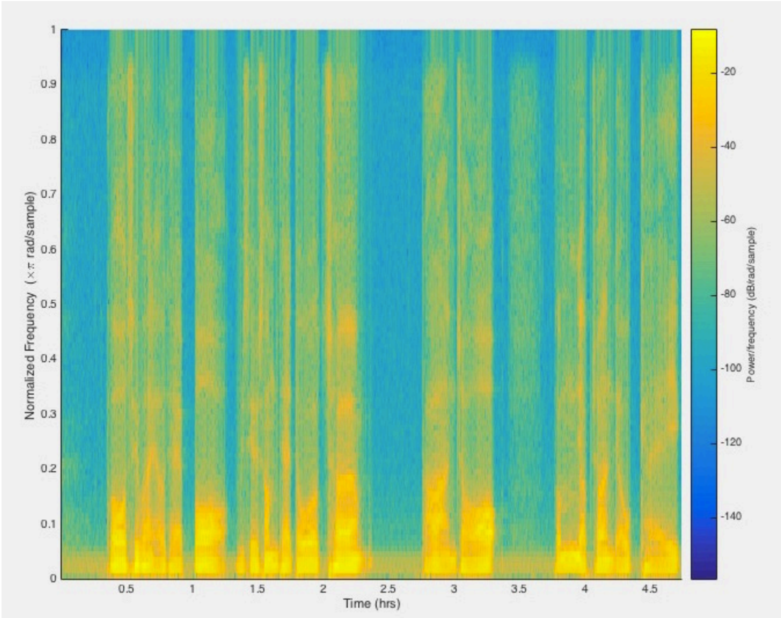

Fig. 5. Spectrogram result for x[n] (window = 200)

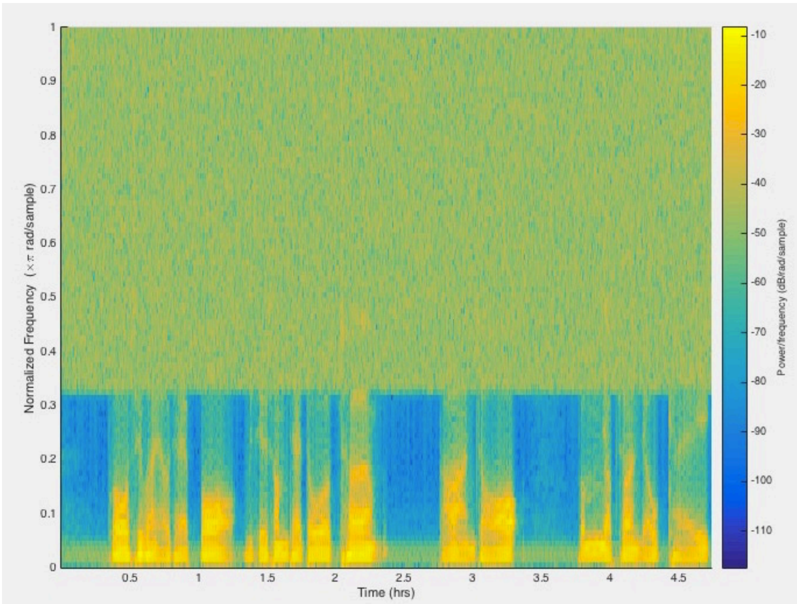

Fig. 6. Spectrogram result for y[n] (window = 200)

We can find that for spectrogram results of noisy signal, there are the monochromatic areas.

B.2. For better frequency resolution, the spectrum on spectrogram result consists with many horizontal "lines". We can find that the spectrogram result with larger window length has more clear lines. In other hand, for better time resolution, the spectrum on spectrogram result consists with many wide vertical "lines". The spectrogram result with smaller window length satisfies this rule. Thus, we can conclude that for larger window length, the frequency resolution is better; for smaller window length, the time resolution is better.

Part C: Design FIR filters to remove noise

C.1. For our FIR filters, we set N = 50, and the pass-band is from 330/FS to 3300/FS. It almost truncates the information for frequency larger than 0.48.

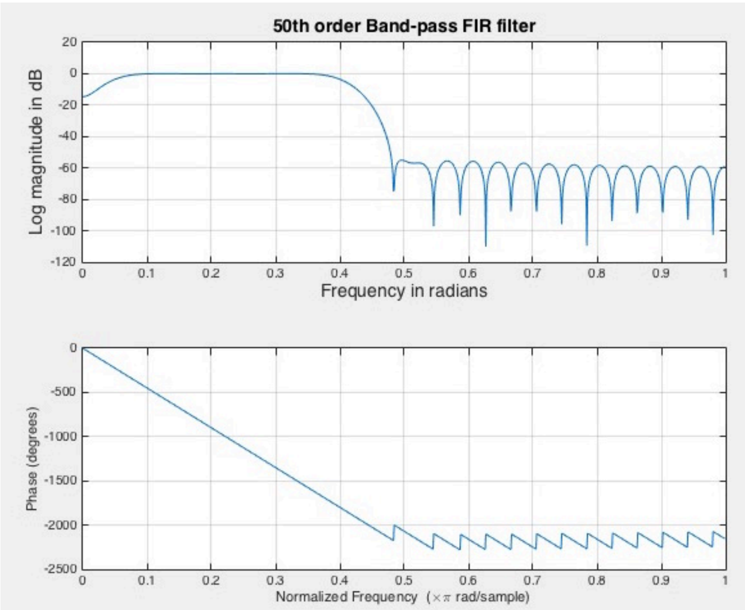

Fig. 7. Log magnitude response result of FIR filter

C.2. The window length is 500, and the frame overlap rate is 50%. The spectrogram result is shown in Fig. 8.

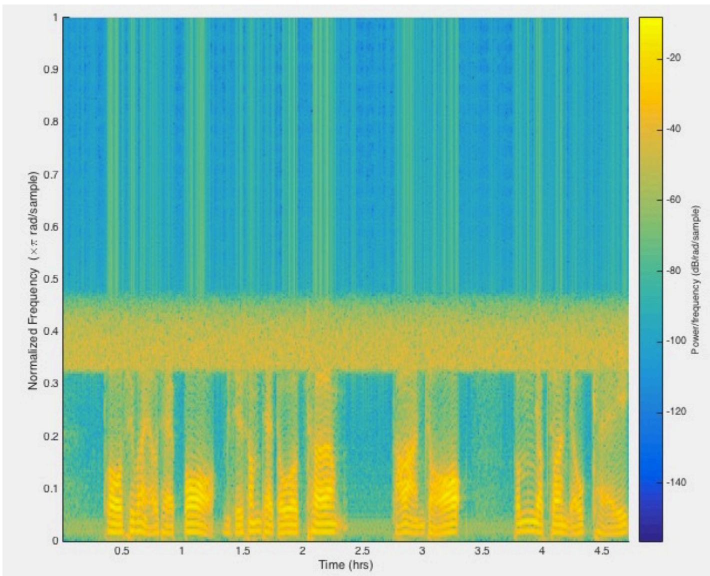

Fig. 8. Spectrogram result for y[n] with FIR filter

The filter removes some noise. However, we still can hear some noise clearly.

C.3. We can find that there is a noise area in the middle of spectrogram result (Normalized frequency from 0.33 to 0.48). It means that we should truncate more information. Thus, we have better re-design the filter.

C.4. The band-pass range is change to 0.0375 to 0.26. In Fig. 9, we can see that the information for frequency larger than 0.33 is almost truncated.

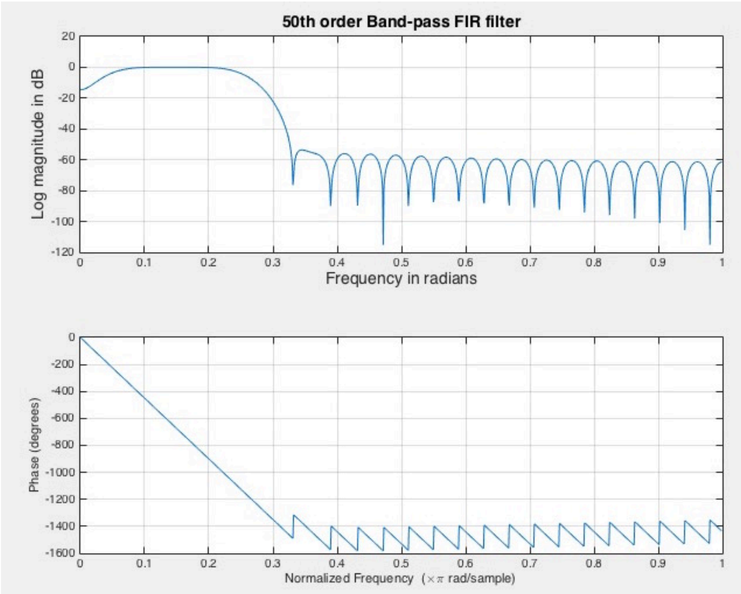

Fig. 9. Log magnitude response result of modified FIR filter

C.5. The spectrogram result is shown in Fig. 10.

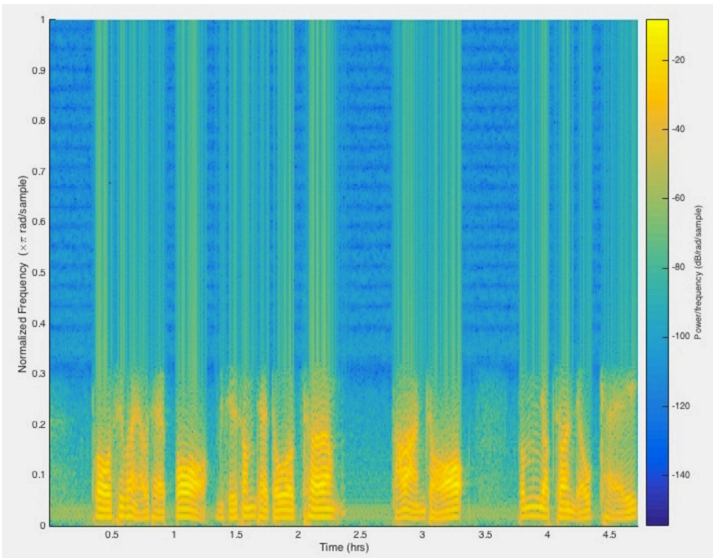

Fig. 10. Spectrogram result for y[n] with modified FIR filter

The noise area is removed. Meanwhile, we hardly hear any noise when we play the auditory.

C.6. The noise sound auditory in C.5 is removed, while we can hear very clearly noise sound in C.2. Besides, the human voice is a little bit weaker, but it’s not very obvious.

Appendix

```
%%%Part A.
[x,FS] = audioread('speech.wav');
[v,FS] = audioread('noise.wav');

%%%A.1
y = x + v;

subplot(3,1,1);
plot(x);
xlabel('n');
ylabel('x[n]');
subplot(3,1,2);
plot(v);
xlabel('n');
ylabel('v[n]');
subplot(3,1,3);
plot(y);
xlabel('n');
ylabel('y[n]');
%%%

%%%A.2
s1 = FS*0.80;
s2 = FS*0.83;
x_cut = x(s1:s2);
y_cut = y(s1:s2);
X_CUT_FFT = fft(x_cut);
Y_CUT_FFT = fft(y_cut);
log_x = 20*log10(abs(X_CUT_FFT));
log_y = 20*log10(abs(Y_CUT_FFT));

subplot(2,1,1);
plot(log_x);
xlabel('n');
```

```
ylabel('x[n]');
subplot(2,1,2);
plot(log_y);
xlabel('n');
ylabel('v[n]');
```

Part B.

```
[x,FS] = audioread('speech.wav');
[v,FS] = audioread('noise.wav');
y = x + v;

%%%B.1
W1 = 500; W2 = 200;
ov = 0.5;
noverlap1 = W1*ov;
noverlap2 = W2*ov;

%%%w=500
spectrogram(x, W1, noverlap1, W1,'yaxis');
spectrogram(y, W1, noverlap1, W1,'yaxis');
%%%w=200
spectrogram(x, W2, noverlap2, W2,'yaxis');
spectrogram(y, W2, noverlap2, W2,'yaxis');
```

Part C.

```
[x,FS] = audioread('speech.wav');
[v,FS] = audioread('noise.wav');
y = x + v;

%%%C.1
N = 50;
w_cut = [300*2/FS, 3300*2/FS];
hbpn_b = fir1(N, w_cut, 'band');
freqz(hbpn_b,1,1024);
y_bpf = filter(hbpn_b,1,y);
```

```
%%soundsc(y_bpf,FS);
[h_b, w_b]=freqz(hbpn_b,1,1024);
xlabel('Frequency in radians','FontSize',14);
ylabel('Log magnitude in dB','FontSize',14);
title([num2str(N),'th order Band-pass FIR filter'],'FontSize',14);

%%%C.2
W1 = 500;
ov = 0.5;
noverlap1 = W1*ov;

w=500;
spectrogram(y_bpf, W1, noverlap1, W1,'yaxis');

%%%C.4
N = 50;
w_cut = [0.0375, 0.26];
hbpn_b = fir1(N, w_cut, 'band');
freqz(hbpn_b,1,1024);
y_bpf = filter(hbpn_b,1,y);
soundsc(y_bpf,FS);
[h_b, w_b]=freqz(hbpn_b,1,1024);
xlabel('Frequency in radians','FontSize',14);
ylabel('Log magnitude in dB','FontSize',14);
title([num2str(N),'th order Band-pass FIR filter'],'FontSize',14);

%%%C.5
W1 = 500;
ov = 0.5;
noverlap1 = W1*ov;

w=500;
spectrogram(y_bpf, W1, noverlap1, W1,'yaxis');
```