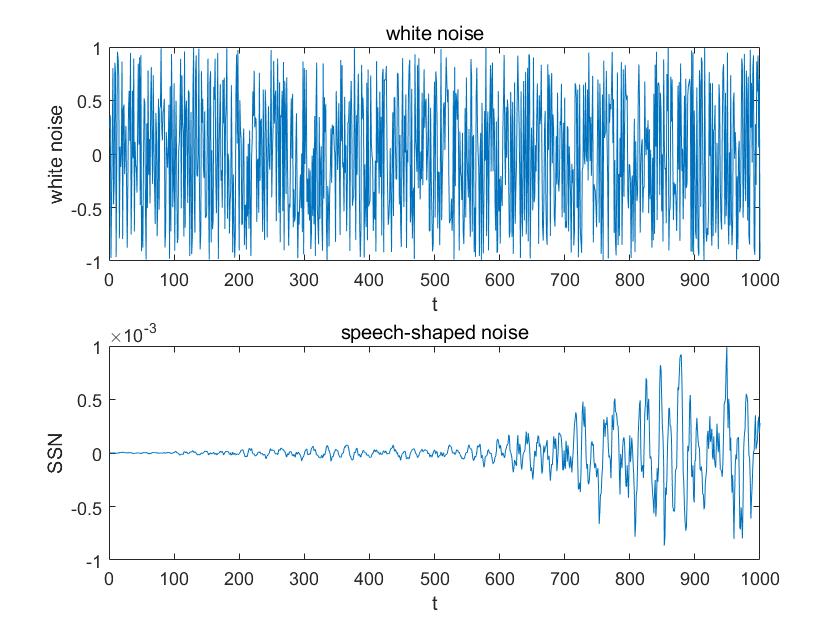
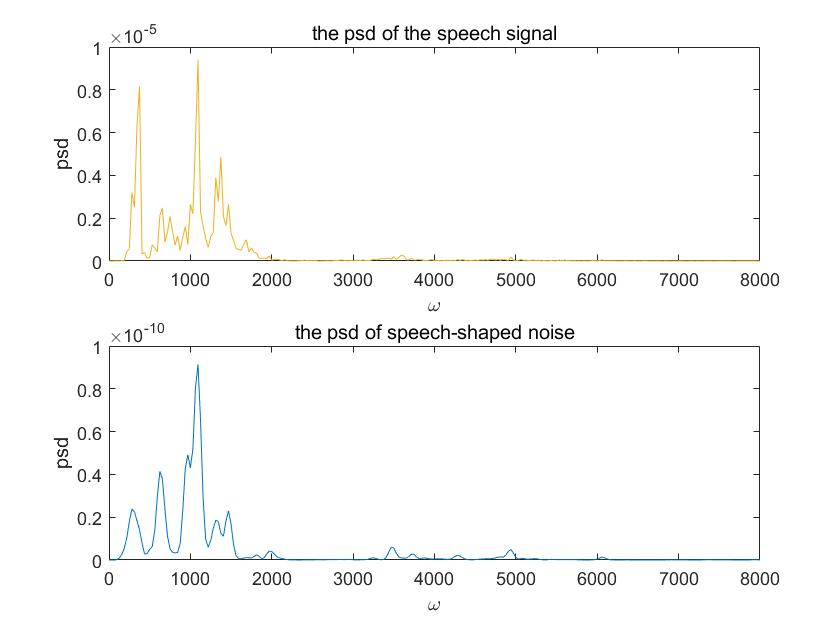
Name 1: 李璇 SID 1: 12010137 Name 2:张林燊SID 2: 12010424

实验报告#5 (Lab#5)

1. Generate a speech-shaped noise (SSN), and plot the spectra of the speech signal and SSN (e.g., use MATLAB function “periodogram” or “pwelch”, or other power spectra density estimation functions).
2. generate long-term spectrum of speech signal
3. generate filter coefficients based on the psd of above speech signal
4. generate white noise (how many points?)
5. perform filtering on white noise signal





It can be noticed that the speech-shaped noise differs from the white noise a lot. And the psd of the speech-shaped noise is similar to that of the speech signal.

**Matlab code:**

[x,fs]= audioread('C\_01\_01.wav');

player = audioplayer(x,fs);

play(player);

noise=1-2\*rand(1,1000);

signal=repmat(x,1,10);

figure(1);

[Pxx1,w1]=pwelch(signal,[],[],512,fs);

subplot(2,1,1);

plot(w1,Pxx1);

title('the psd of the speech signal');

xlabel('\omega');

ylabel('psd');

b=fir2(3000,w1/(fs/2),sqrt(Pxx1/max(Pxx1)));

ssn=filter(b,1,noise);

[Pxx2,w2]= pwelch(ssn,[],[],512,fs);

subplot(2,1,2),plot(w2,Pxx2);

title('the psd of speech-shaped noise');

xlabel('\omega');

ylabel('psd');

figure(2);

subplot(2,1,1),

plot((1:1000),noise),title('white noise');

xlabel('t');

ylabel('white noise');

subplot(2,1,2),

plot((1:length(ssn)),ssn);

title('speech-shaped noise');

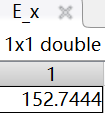
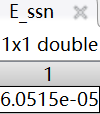
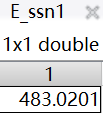
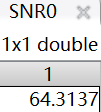
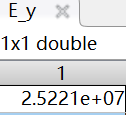
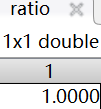
xlabel('t');

ylabel('SSN');

1. Read a speech signal , adjust the SNR ( to the above SSN) to -5dB, let , and normalize the energy of in relative to , i.e., modify the energy of so that it equals to that of .
2. adjust the intensity of SSN so that

1. adjust the intensity of so that

, that is energy of equals to energy of

First we calculate the energy of the signal x by E\_x=norm(x)\*norm(x) the energy of the ssn by E\_ssn=norm(ssn)\*norm(ssn). Then we get the SNR0=64.3137dB. Then let SNR1=-5dB, we get ssn1 which satisfies SNR=-5dB by ssn1=ssn\*((E\_ssn1/E\_ssn)^(0.5)) and E\_ssn1. Then we get y=x+ssn1 and E\_y. Then we get the signal y1 which has the same energy as x. Then, through normalization, we got the ratio of norm(x) to norm(y1), which equals to 1. That verifies x and y1 has the same energy.

**Matlab code:**

[x,fs]= audioread('C\_01\_01.wav');

%assignment1

player = audioplayer(x,fs);

play(player);

noise=1-2\*rand(1,1000);

signal=repmat(x,1,10);

figure(1);

[Pxx1,w1]=pwelch(signal,[],[],512,fs);

subplot(2,1,1);

plot(w1,Pxx1);

title('the psd of the speech signal');

xlabel('\omega');

ylabel('psd');

b=fir2(3000,w1/(fs/2),sqrt(Pxx1/max(Pxx1)));

ssn=filter(b,1,noise);

[Pxx2,w2]= pwelch(ssn,[],[],512,fs);

subplot(2,1,2),plot(w2,Pxx2);

title('the psd of speech-shaped noise');

xlabel('\omega');

ylabel('psd');

figure(2);

subplot(2,1,1),

plot((1:1000),noise),title('white noise');

xlabel('t');

ylabel('white noise');

subplot(2,1,2),

plot((1:length(ssn)),ssn);

title('speech-shaped noise');

xlabel('t');

ylabel('SSN');

%assignment2

E\_x=norm(x)\*norm(x);

E\_ssn=norm(ssn)\*norm(ssn);

SNR0=20\*log10(norm(x)/norm(ssn));

E\_ssn1=E\_x/(10^(-5/10));

ssn1=ssn\*((E\_ssn1/E\_ssn)^(0.5));

SNR1=20\*log10(norm(x)/norm(ssn1));

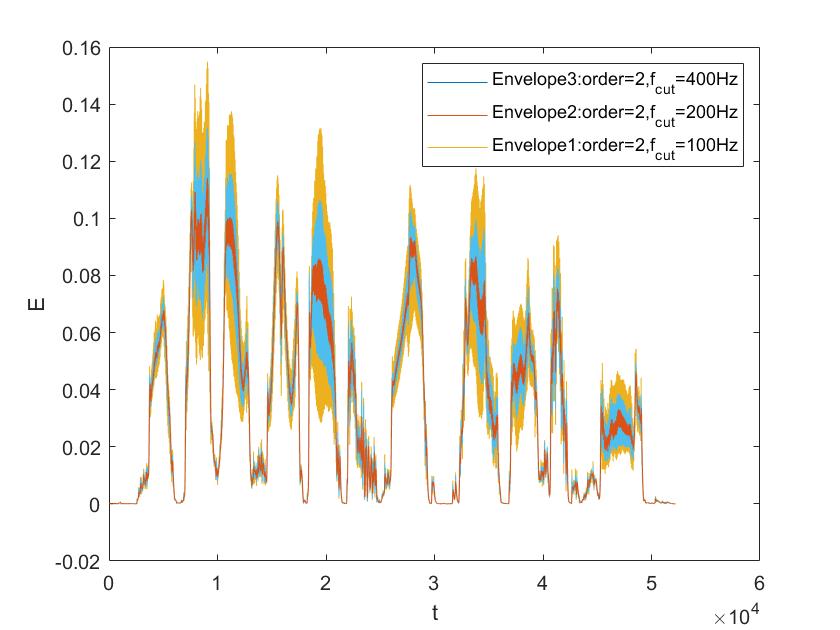
y=x+ssn1;

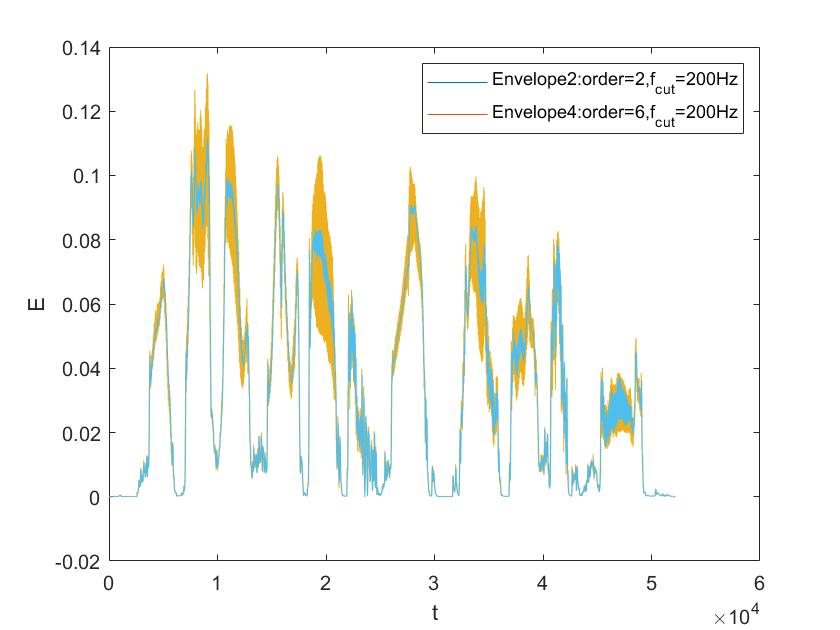
E\_y=norm(y)\*norm(y);

y1=y\*((E\_x/E\_y)^0.5);

ratio=norm(y1)/norm(x);

1. Extract speech envelope
2. full wave rectification:
3. low-pass filtering
   1. With 2nd order low-pass filter and cut-off frequency . Plot these three envelope waveforms in one plot, and describe the difference among them.
   2. With 2nd and 6th order low-pass filter and cut-off frequency . Plot these two envelope waveforms in one plot, and describe the difference between them.

 It could be noticed that when fcut is becoming larger, the envelop becomes smoother and has a smaller range. That is to say, smaller fcut can get a smoother plot.



It is noticed that when the order is increasing, the envelop is smoother and has a smaller range. That is to say, larger order can get a smoother plot. Thus, in order to generate a smooth envelope, we need to use a high-order low-pass filter.

**Matlab code:**

[x,fs]= audioread('C\_01\_01.wav');

%assignment1

player = audioplayer(x,fs);

play(player);

noise=1-2\*rand(1,1000);

signal=repmat(x,1,10);

figure(1);

[Pxx1,w1]=pwelch(signal,[],[],512,fs);

subplot(2,1,1);

plot(w1,Pxx1);

title('the psd of the speech signal');

xlabel('\omega');

ylabel('psd');

b=fir2(3000,w1/(fs/2),sqrt(Pxx1/max(Pxx1)));

ssn=filter(b,1,noise);

[Pxx2,w2]= pwelch(ssn,[],[],512,fs);

subplot(2,1,2),plot(w2,Pxx2);

title('the psd of speech-shaped noise');

xlabel('\omega');

ylabel('psd');

figure(2);

subplot(2,1,1),

plot((1:1000),noise),title('white noise');

xlabel('t');

ylabel('white noise');

subplot(2,1,2),

plot((1:length(ssn)),ssn);

title('speech-shaped noise');

xlabel('t');

ylabel('SSN');

%assignment2

E\_x=norm(x)\*norm(x);

E\_ssn=norm(ssn)\*norm(ssn);

SNR0=20\*log10(norm(x)/norm(ssn));

E\_ssn1=E\_x/(10^(-5/10));

ssn1=ssn\*((E\_ssn1/E\_ssn)^(0.5));

SNR1=20\*log10(norm(x)/norm(ssn1));

y=x+ssn1;

E\_y=norm(y)\*norm(y);

y1=y\*((E\_x/E\_y)^0.5);

ratio=norm(y1)/norm(x);

%assignment 3

figure(3);

[b1,a1]= butter(2,100/(fs/2));

[b2,a2]= butter(2,200/(fs/2));

[b3,a3]= butter(2,300/(fs/2));

yy1= filter(b1,a1,abs(signal));

yy2=filter(b2,a2,abs(signal));

yy3= filter(b3,a3,abs(signal));

plot((1:length(yy3)),yy3);

xlabel('t');

ylabel('E');

hold on;

plot((1:length(yy2)),yy2);

hold on;

plot((1:length(yy1)),yy1);

legend('Envelope3:order=2,f\_c\_u\_t=400Hz','Envelope2:order=2,f\_c\_u\_t=200Hz','Envelope1:order=2,f\_c\_u\_t=100Hz');

figure(4);

[b4,a4]=butter(6,200/(fs/2));

yy4=filter(b4,a4,abs(signal));

plot((1:length(yy2)),yy2);

xlabel('t');

ylabel('E');

hold on;

plot((1:length(yy4)),yy4);

legend('Envelope2:order=2,f\_c\_u\_t=200Hz','Envelope4:order=6,f\_c\_u\_t=200Hz');