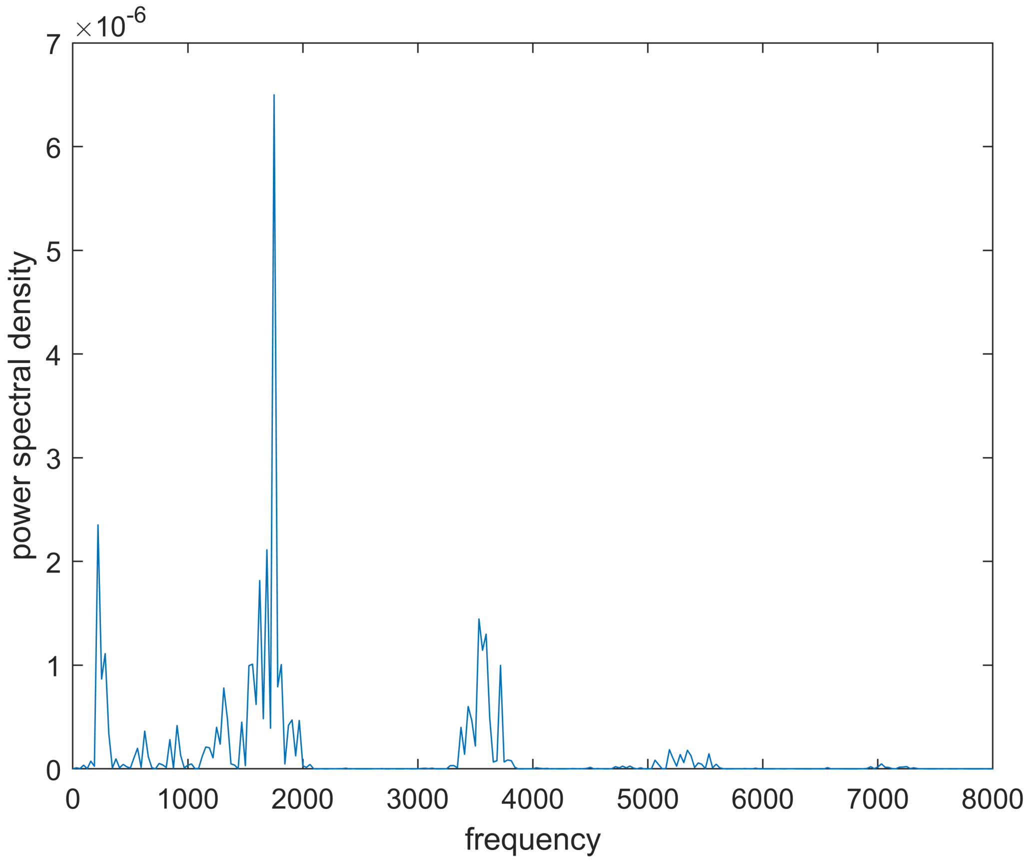
Report of Signals and Systems Lab Assignment 5

Written by HUANG Guanchao, SID 11912309 and GONG Xinrui, SID 11911233.

> In the following tests, we used the first speech signal, namely `C_01_01.wav` as material.

Part 1

The power spectrum of the speech signal is shown in the figure below.



The length of the noise should correspond to that of the speech signal.

The MATLAB script is shown in the code block below.

```
close all
[y, fs] = audioread('C_01_01.wav');
sig = repmat(y, 5, 1);
[Pxx, w] = periodogram(sig, [], 512, fs);

plot(w, Pxx)
title('power spectrum density of the speech signal')
xlabel('frequency')
ylabel('power spectral density')

b = fir2(3000, w / (fs / 2), sqrt(Pxx / max(Pxx)));
[N, ~] = size(sig);
```

```
noise = 1 - 2 * rand(N, 1);
ssn = filter(b, 1, noise);
```

Part 2

To set the SNR to be -5, assuming the coefficient of `sig` is 1, and that of `ssn` is a, according to the property of `norm()`, we have

$$\mathrm{SNR} = 20\lg\frac{\mathrm{norm}(\mathrm{sig})}{\mathrm{norm}(a\,\mathrm{ssn})} = -5 \Rightarrow a = \frac{\mathrm{norm}(\mathrm{sig})}{\mathrm{norm}(\mathrm{ssn})}10^{1/4}.$$

To verify this, try running

```
snr = 20 * log10(norm(sig) / norm(ssn * norm(sig) / norm(ssn) * 10^(1/4)))
```

The output is exactly

```
snr =
    -5
```

To adjust the intensity, run

```
y = y / norm(y) * norm(sig);
norm(sig)
norm(y)
```

The output is again exactly

```
ans =
   27.6355

ans =
   27.6355
```

Therefore, the norm, as well as the energy in signal `sig` and `y` is identical.

Part 3

Firstly, do full wave rectification with running

```
y = abs(y);
```

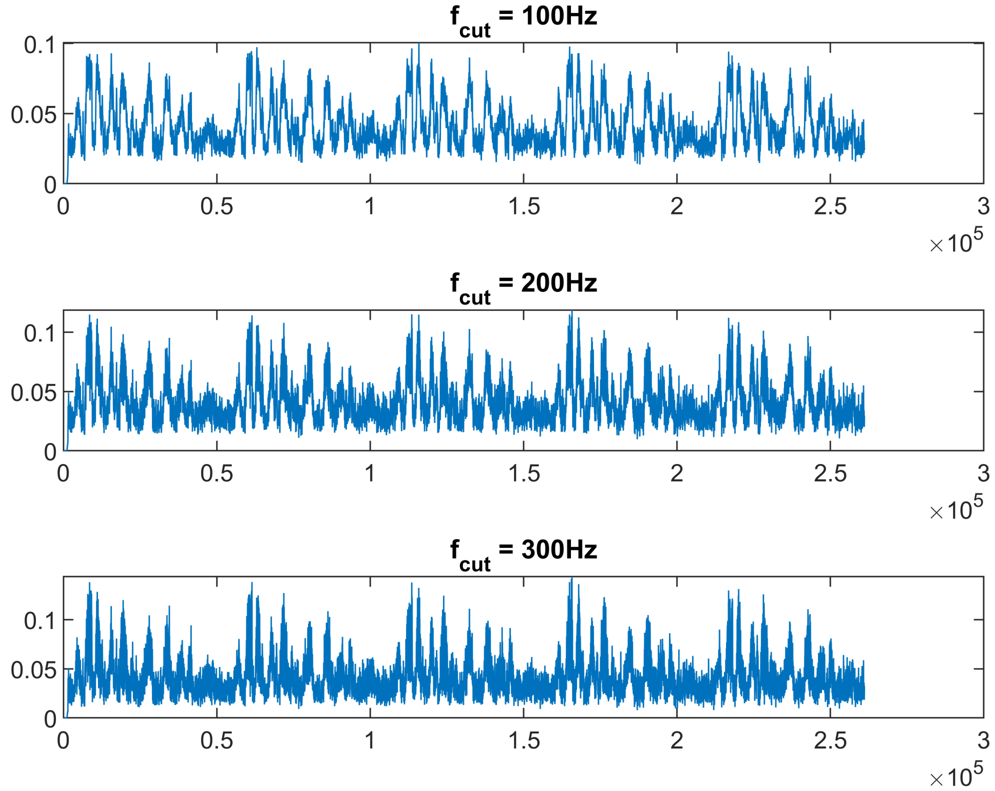
Then, generate and apply 2-nd order low-pass Butterworth filter of different cutoff frequency.

```
[b, a] = butter(2, 100 / (fs / 2));
env = filter(b, a, y);
subplot(3, 1, 1)
plot(env)
title('f_{cut} = 100Hz')
hold on
```

```
[b, a] = butter(2, 200 / (fs / 2));
env = filter(b, a, y);
subplot(3, 1, 2)
plot(env)
title('f_{cut} = 200Hz')
hold on

[b, a] = butter(2, 300 / (fs / 2));
subplot(3, 1, 3)
env = filter(b, a, y);
plot(env)
title('f_{cut} = 300Hz')
```

The figure is shown below.



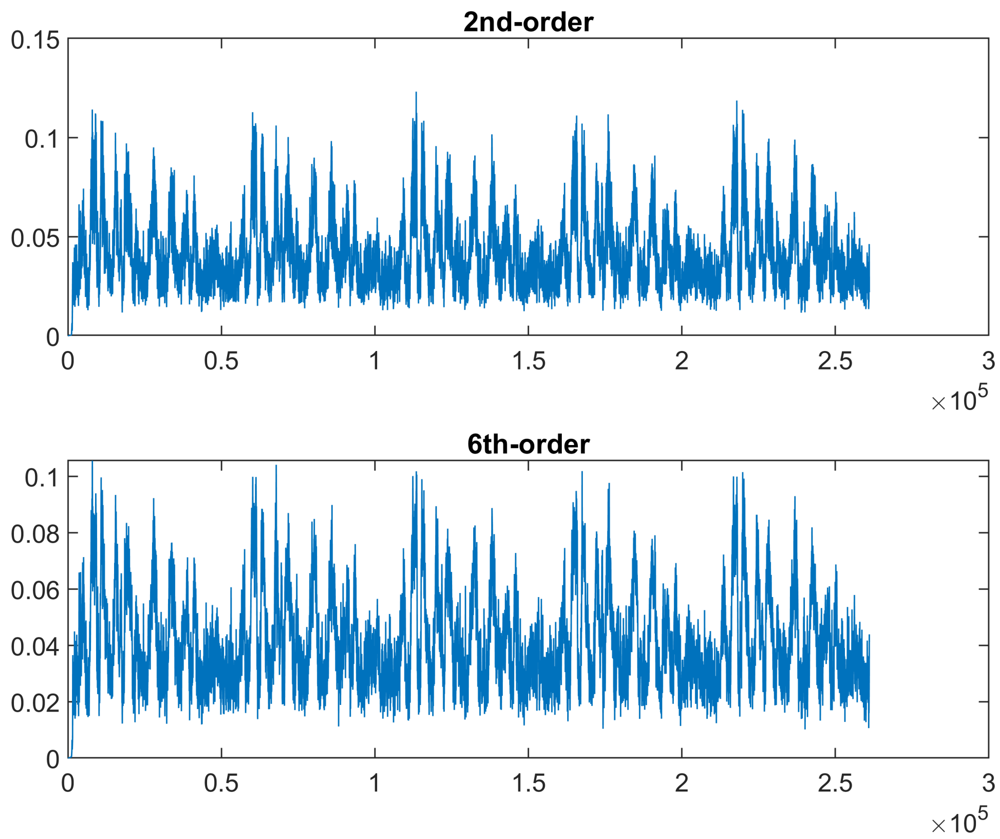
According to the waveform, it is apparently that the higher the cutoff frequency is, the "denser" the waveform of envelope extracted obtained.

Again, try low-pass Butterworth filter with 6th-order.

```
[b, a] = butter(2, 200 / (fs / 2));
env = filter(b, a, y);
subplot(2, 1, 1)
plot(env)
title('2nd-order')
hold on

[b, a] = butter(6, 200 / (fs / 2));
env = filter(b, a, y);
subplot(2, 1, 2)
plot(env)
title('6th-order')
```

The figure is shown below.



Compared with that of the 2nd-order filter, the wave form of envelope extracted using 6th-order filter is even sharper, thus has better performance.