Project 1 report

张骥霄　杨牧之　冯柏钧　周安然
12012924　12010422　12011124　12011214

May 20, 2022

1 Introduction

To Help deaf people to hear again, today, a prosthetic device, called the cochlear implant, can be implanted in the inner ear and can restore partial hearing to profoundly deaf people.

To verify the principle of the cochlear implant, we do this project - Speech synthesis and perception with envelope cue.

The main steps of our implementation is to filter the original signal with different bandpass filters, do full-wave rectification and pass a low-pass filer, multiple with a sine wave and finally sum up each components and get the result. Specified in the following figure.

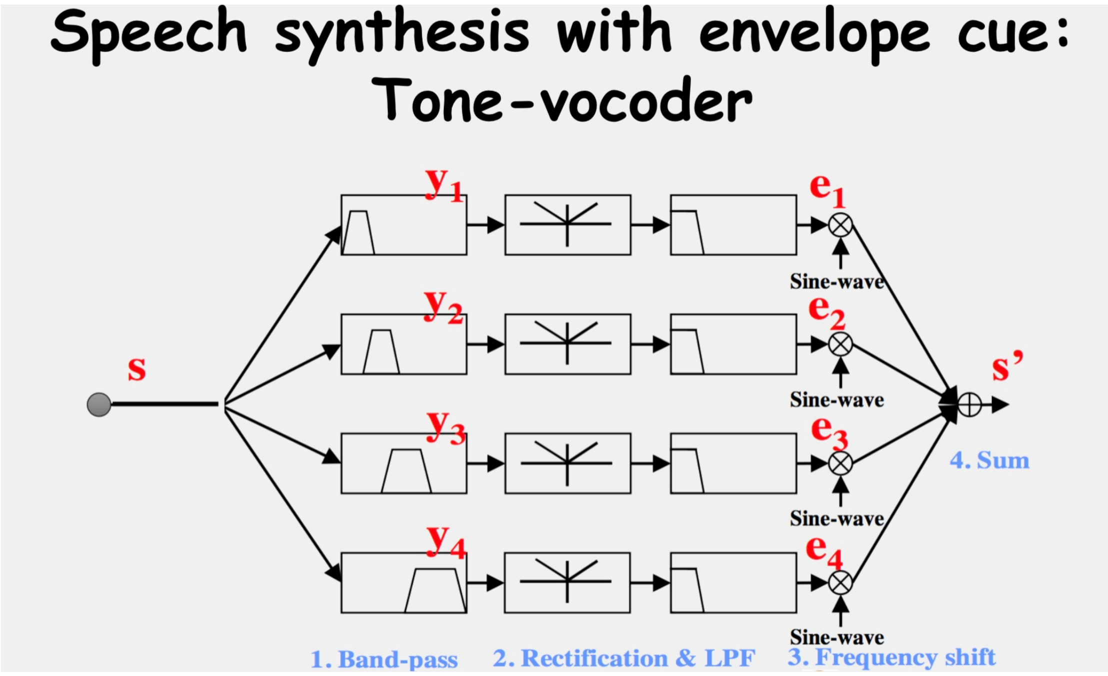
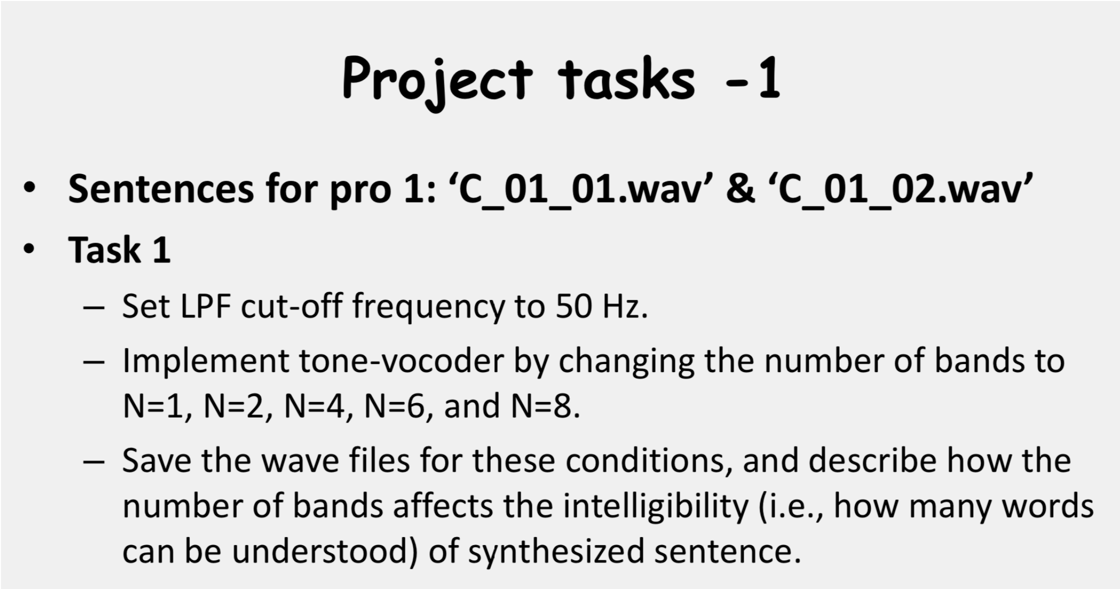


Figure 1: The process of our experiment

2 Task 1(张骥霄)

2.1 Task



2.2 Result

As required, I first set the LPF cut-off frequency to 50Hz as shown below, which suit our needs.

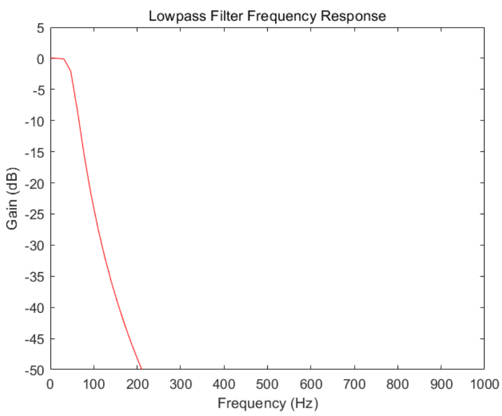


Figure 2: The certification of low-pass filer

Besides, I also check the bandpass filter to see if it is worked as expected, here is one example, which substantiate the correctness of the used filter.

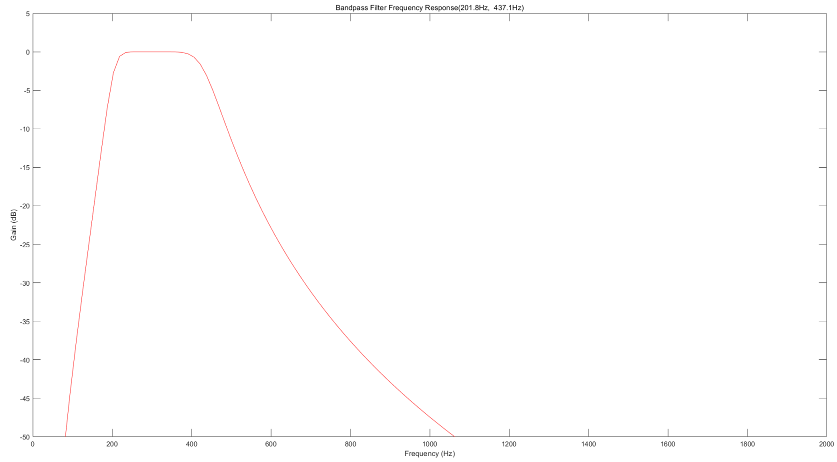


Figure 3: The certification of low-pass filer

I separately process the provided two signals, here is the original signal of the first signal.

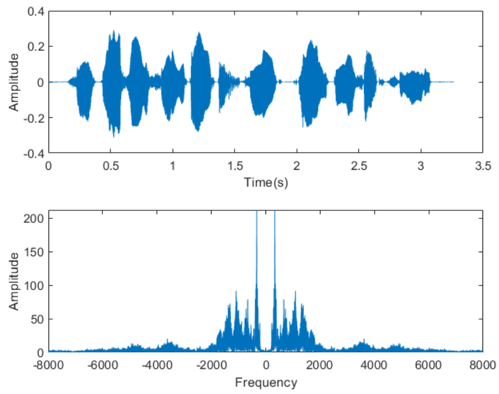


Figure 4: The original signal of the first signal

By set the number of bands $N = 1, 2, 4, 8, 16$, we get the following result, where the first 8 subplots are the waves of processed signals and next 8 subplots are their corresponding Fourier Transform.

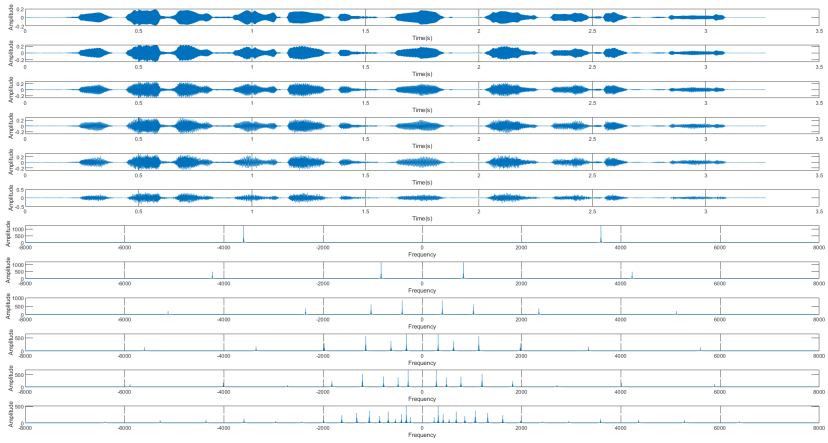


Figure 5: The processed signal of the first signal

Similarly, I processed the second signal using the same wave, here is the original signal and the processed signal of the second signal respectively.

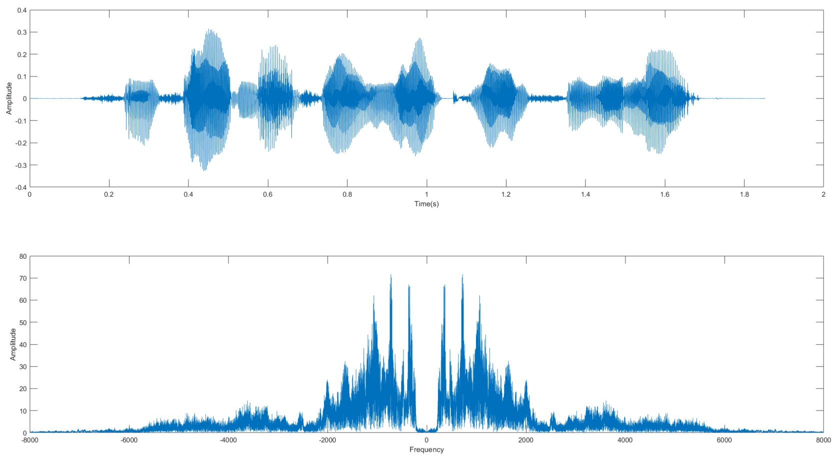


Figure 6: The original signal of the second signal

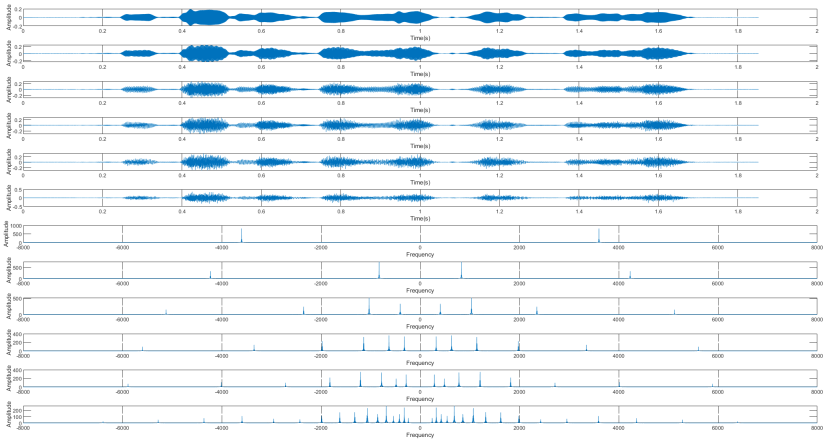


Figure 7: The processed signal of the second signal

Finally, I also draw the power spectrum of the two signal. First I will show the power spectrum of the original signal of first signal.

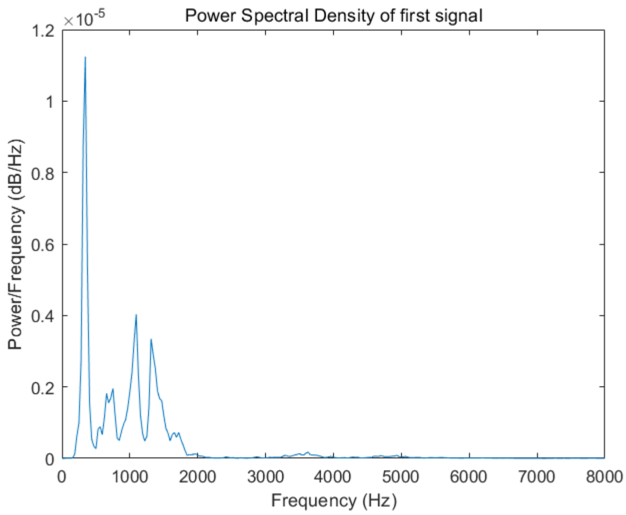


Figure 8: The power spectrum of the original signal of the first signal

Then I set the number of bands $N = 1, 2, 4, 8, 16$, the result are shown in the following picture respectively.

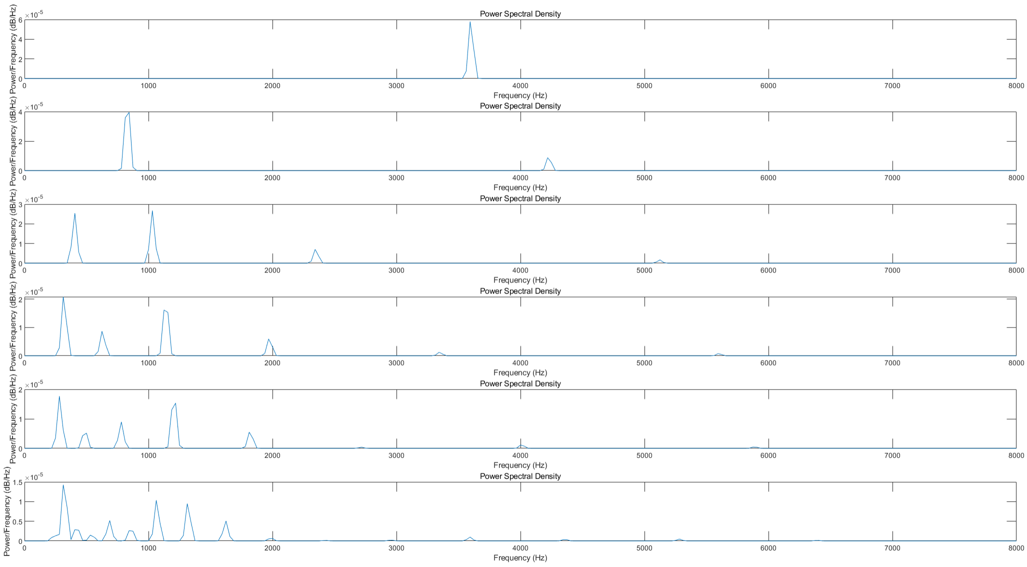


Figure 9: The power spectrum of the processed signal of the first signal

Similarly, the following two pictures are the result of the original and processed signal respectively.

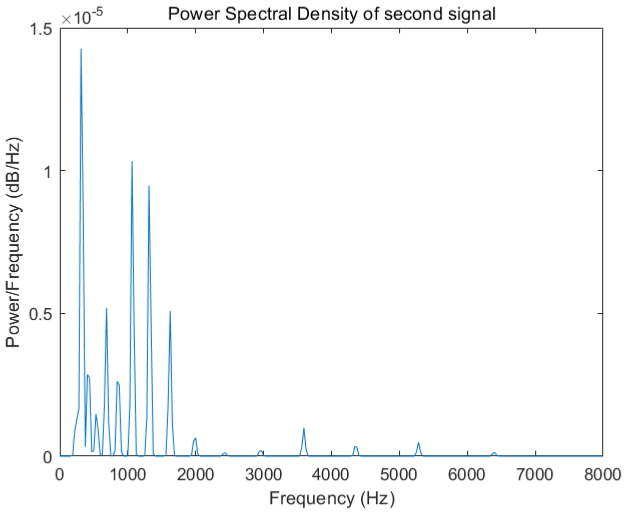


Figure 10: The power spectrum of the original signal of the second signal

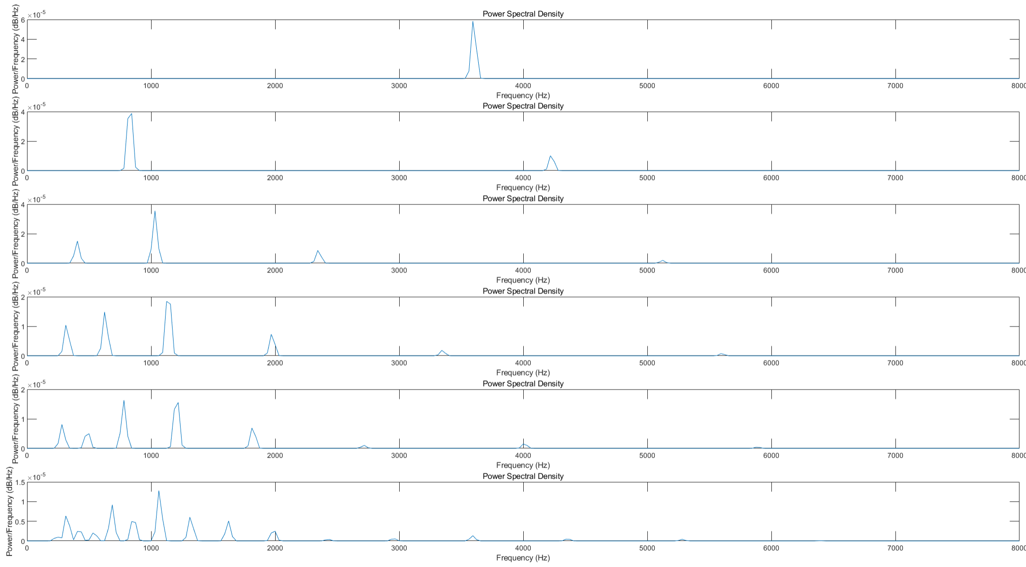


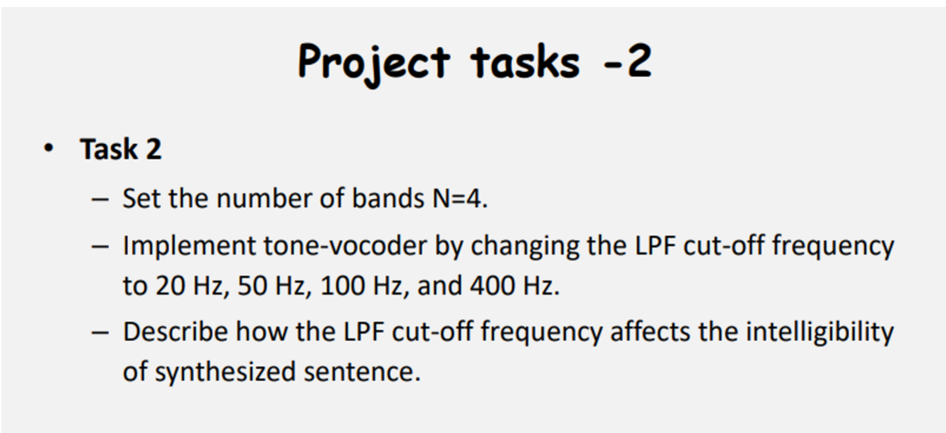
Figure 11: The power spectrum of the processed signal of the second signal

2.3 Analysis

As we can see, when the LPF cut-off frequency is fixed, the more bands the processed signal have, the more information the signal is held, which we can see from above figures, where the bands is more, the Fourier transform and the power spectrum is more close to the original signal's. The audio produced can also prove this point, when the bands is smaller than 4, we cannot understand any words in the audio, as the bands number is bigger than eight, we can roughly hear what words the audio contains, while more bands means clearer audio(When the band number is 64 or 128, it perform little difference compared to the original signal).

3 Task 2 (冯柏钧)

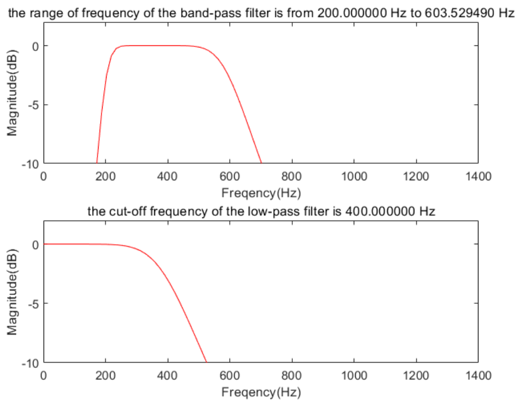
3.1 Task



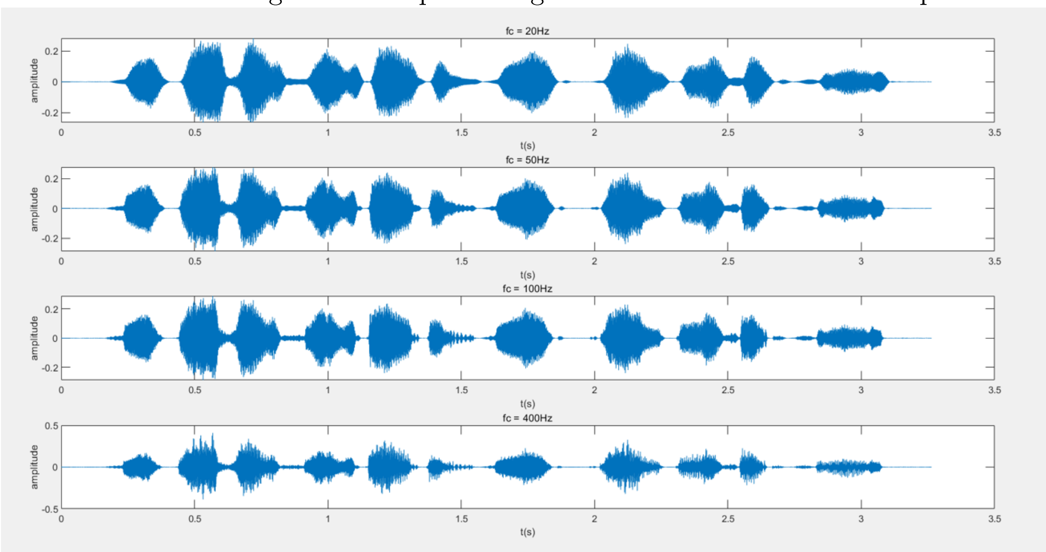
3.2 Result

Before we use the band-pass filer and the low-pass filter to get the envelopes, we must confirm the properties of the filer first. Since my task pays no attention to the exact order of the low-pass filter, I select the 4 as the number of order of all low-pass filters used in my task. In my task, the band number is always 4, so I plot the certification of one band-pass filter and one low-pass filer as an example. Some other configuration of the filters are listed in the figure.

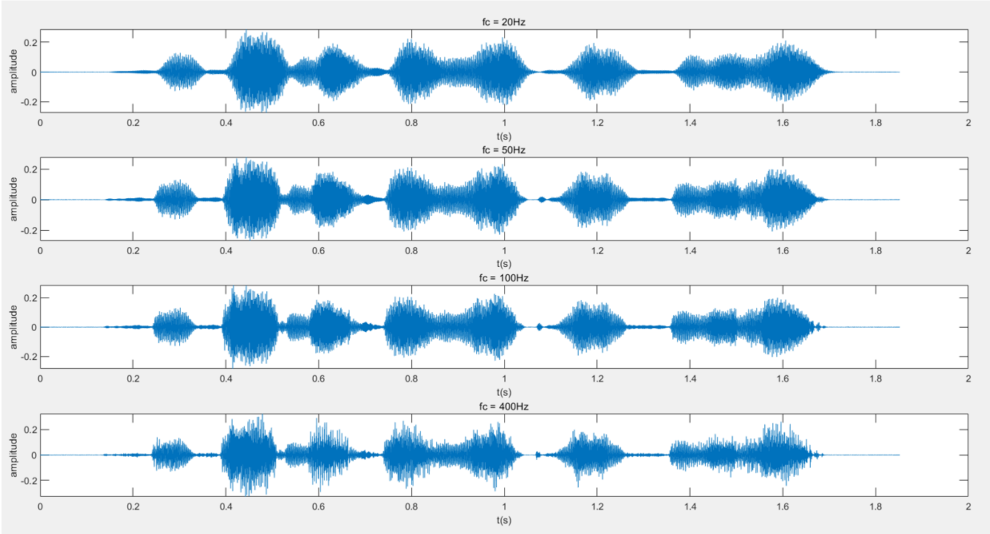
The certification of one band-pass filter and one low-pass filer



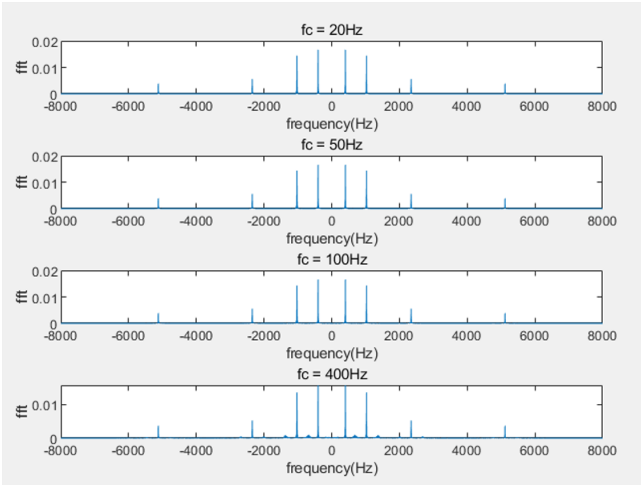
The waveform of signal 1 after processing with different LPF cut-off frequencies



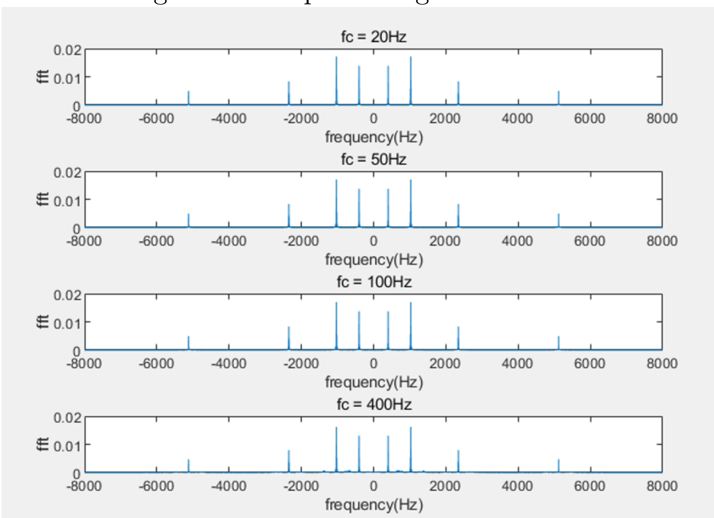
The waveform of signal 2 after processing with different LPF cut-off frequencies



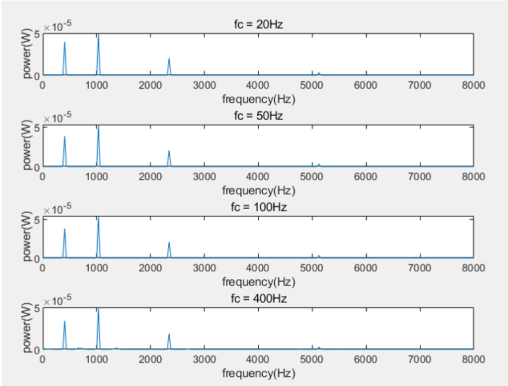
The Fourier transform of signal 1 after processing with different LPF cut-off frequencies



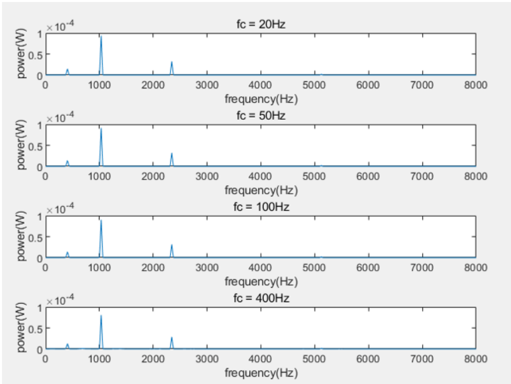
The Fourier transform of signal 2 after processing with different LPF cut-off frequencies



The power of signal 1 after processing with different LPF cut-off frequencies



The power of signal 1 after processing with different LPF cut-off frequencies



3.3 Analysis

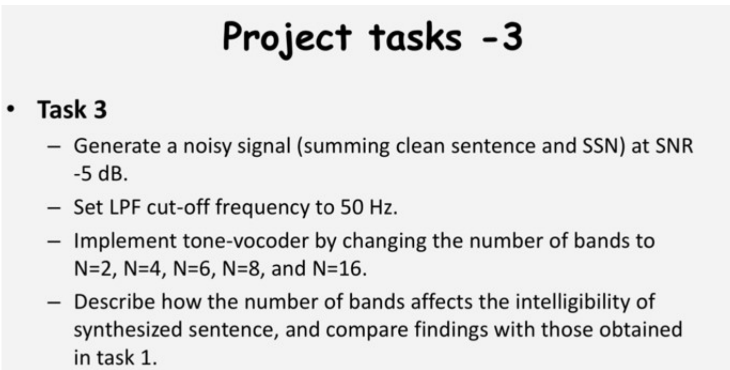
Within the given range, as the LPF cut-off frequency increases, the voice of people becomes clearer.

I test the LPF cut-off frequency to be 4000Hz, it is clearer than the previous work, however, there still exists significant difference with Original file.

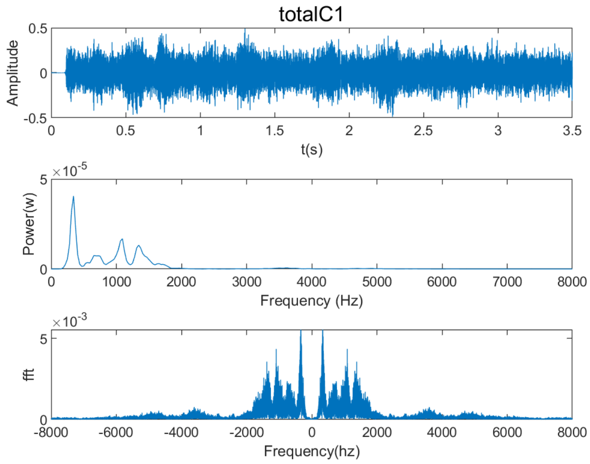
I design my program with the thinking of functional programming. Encapsulate my codes to make my work more clear and readable.

4 Task 3

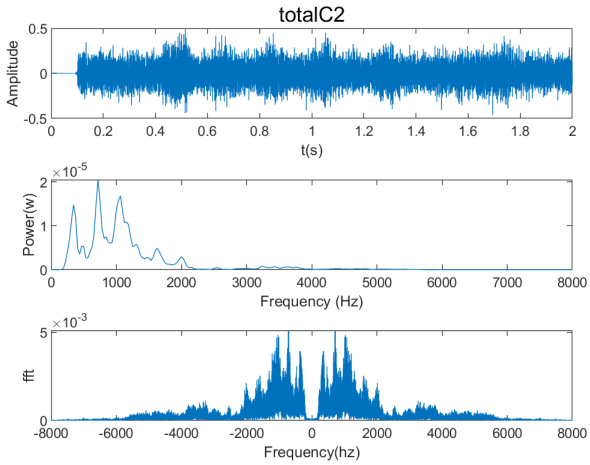
4.1 Task



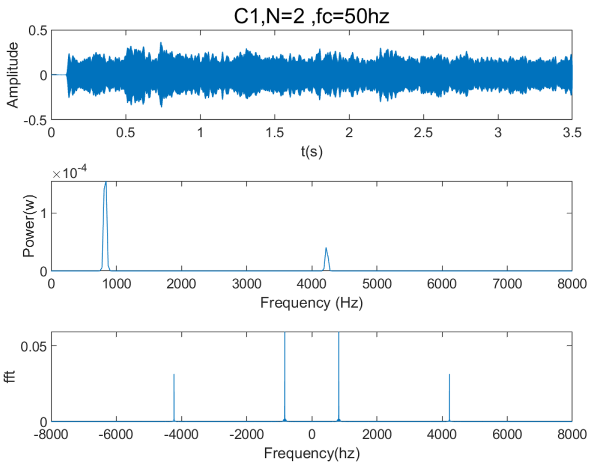
4.2 Result



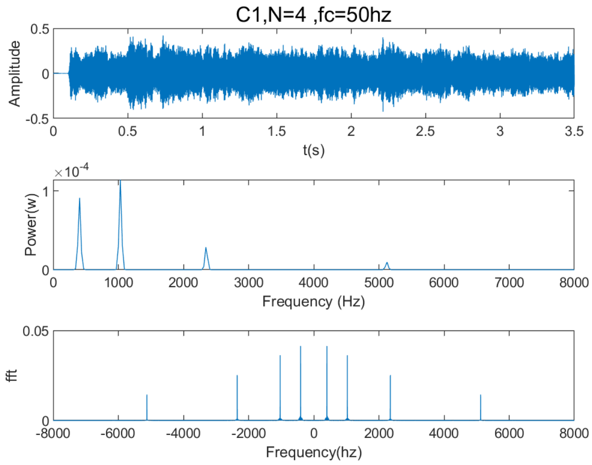
signal1 total



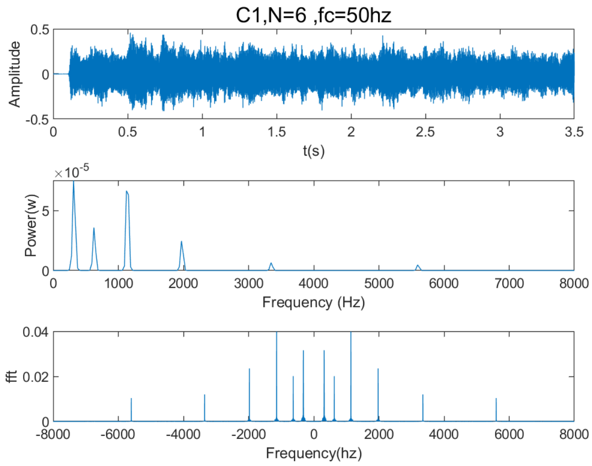
signal2 total



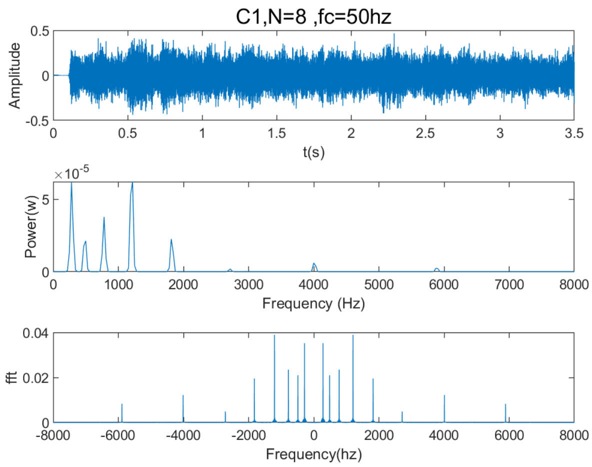
signal1 N=2



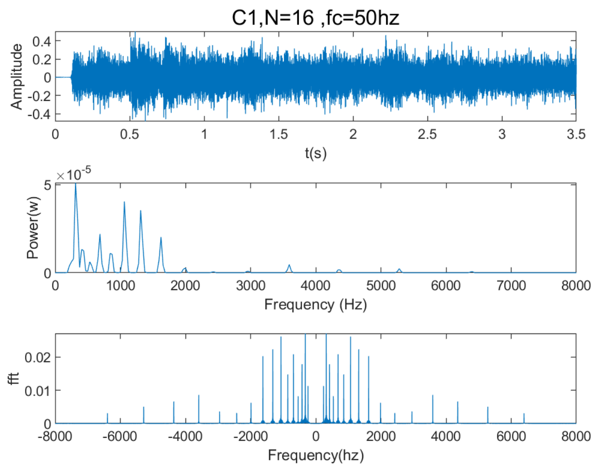
signal1 N=4



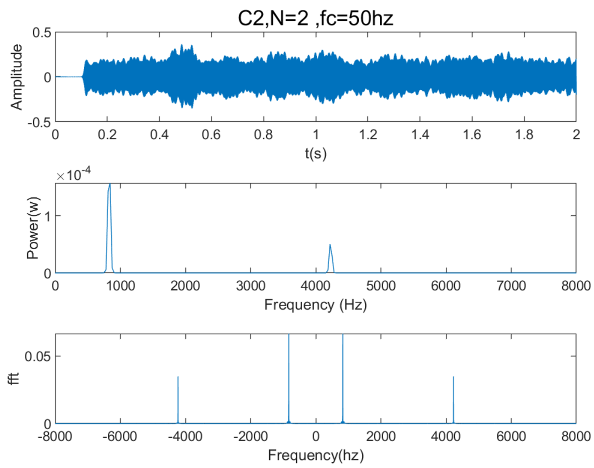
signal1 N=6



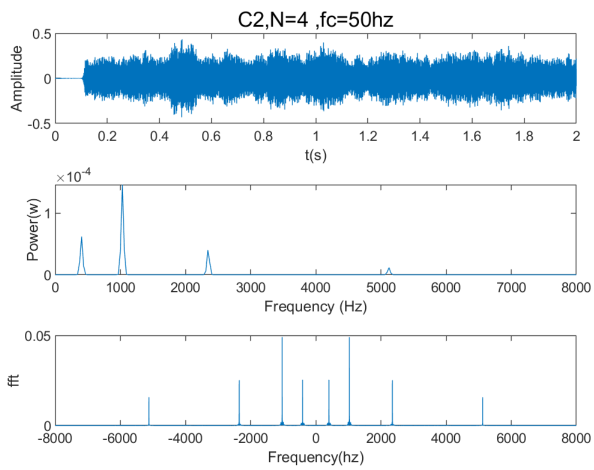
signal1 N=8



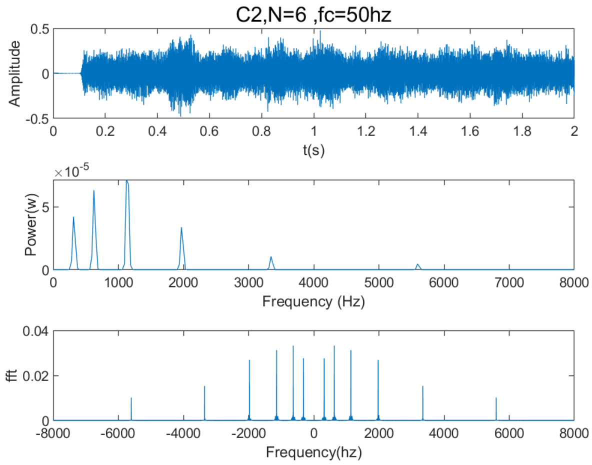
signal1 N=16



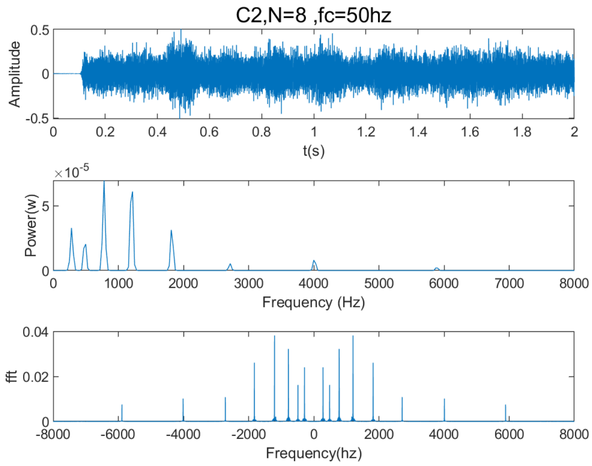
signal2 N=2



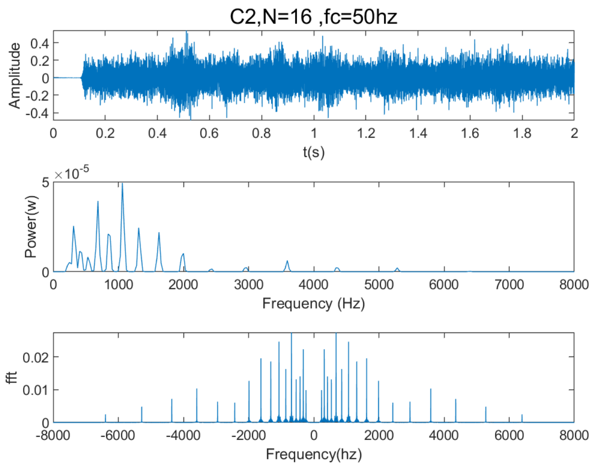
signal2 N=4



signal2 N=6

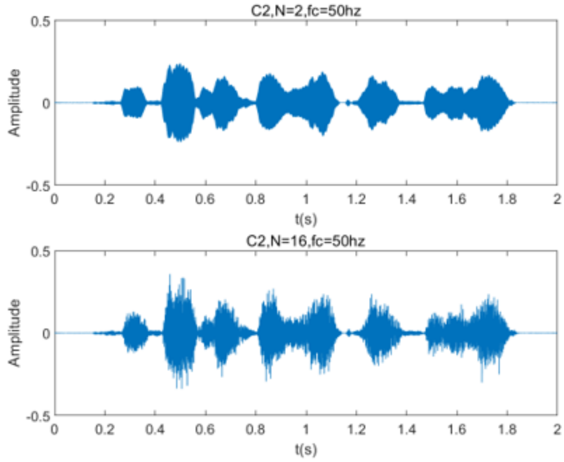


signal2 N=8

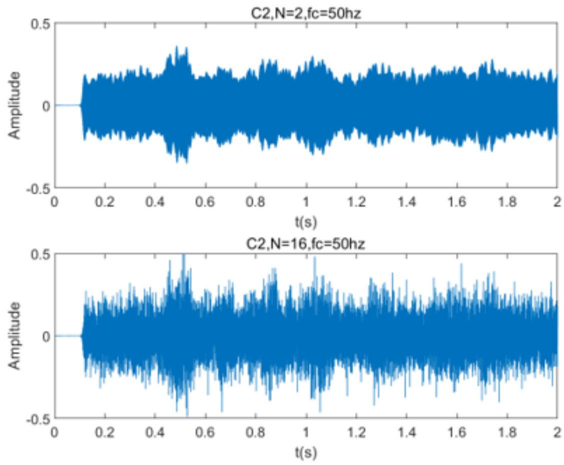


signal2 N=16

4.3 Analysis



Original signal

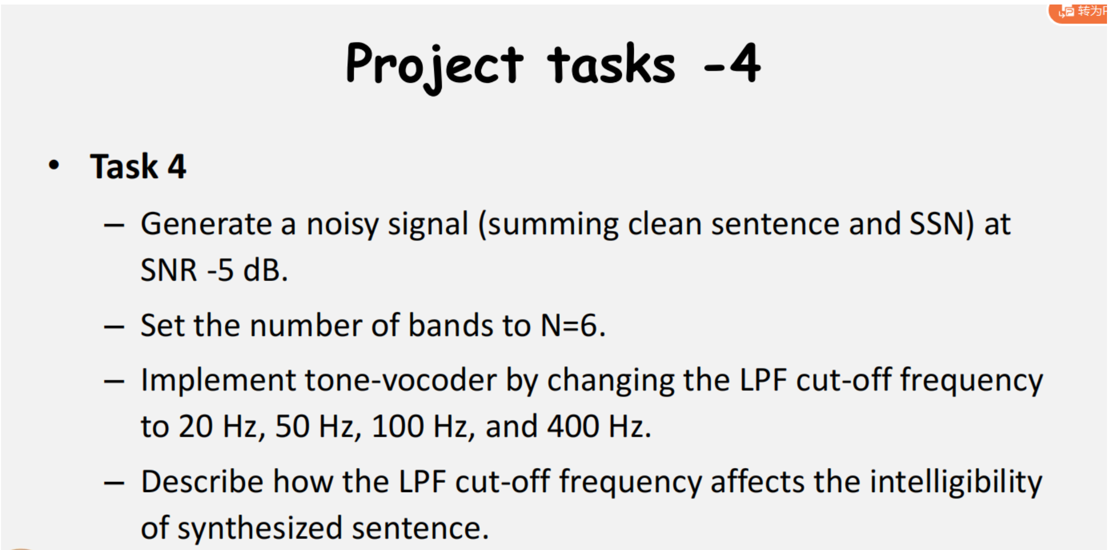


Synthetic signal

I compared N2 and N16 of the original signal with the signal with noise respectively. We can see from the figure that when N=2, the byte of the speech signal is not clearly distinguished from noise, and what we heard just now is almost noise.When N=16, there is a distinct distinction between speech signal and noise.Note: In a certain range, the larger N is, the smaller the masking effect of noise is.

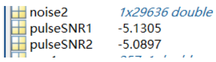
5 Task 4(周安然)

5.1 Task



5.2 Result

1. Firstly, we make sure we have set SNR equals to -5.



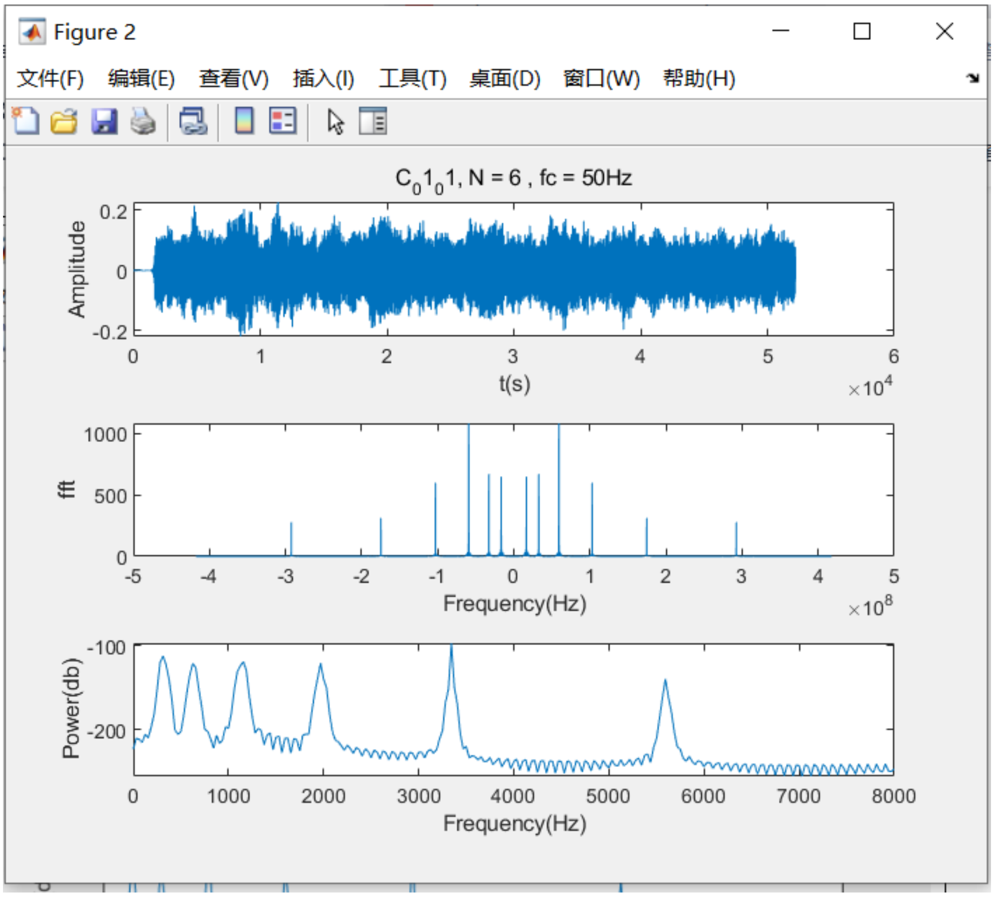
This is C$_0$1$_0$1, $SSN1andy1 = C_01_01 + SSN1$.

This is C$_0$1$_0$2, $SSN2andy2 = C_01_02 + SSN2$.

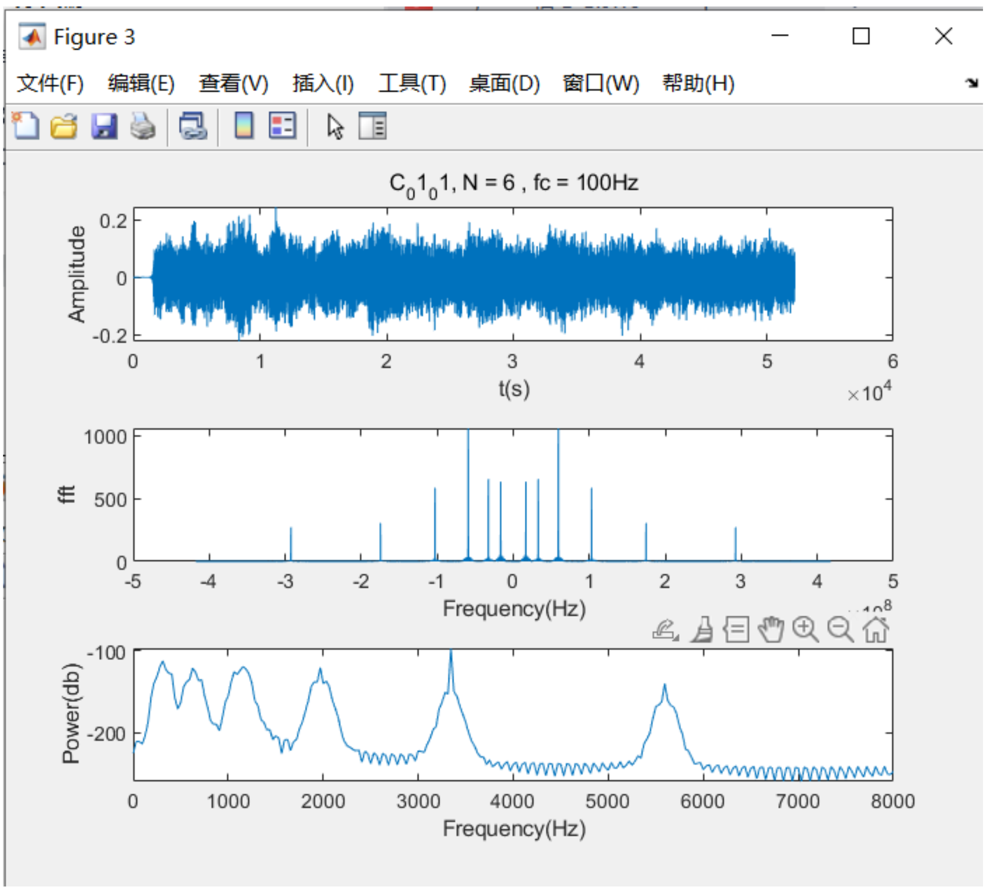
From the above pictures, we can see that We could find that the noise covers the most area of clean sentence at SNR = -5.

2. According to the request of task4, I set N = 6 and four different LPF cut-off frequency. Here are pictures which contains psd-Time, fft-Frequency and Power-Frequency of y1 and y2 with different LPF cutoff frequency.

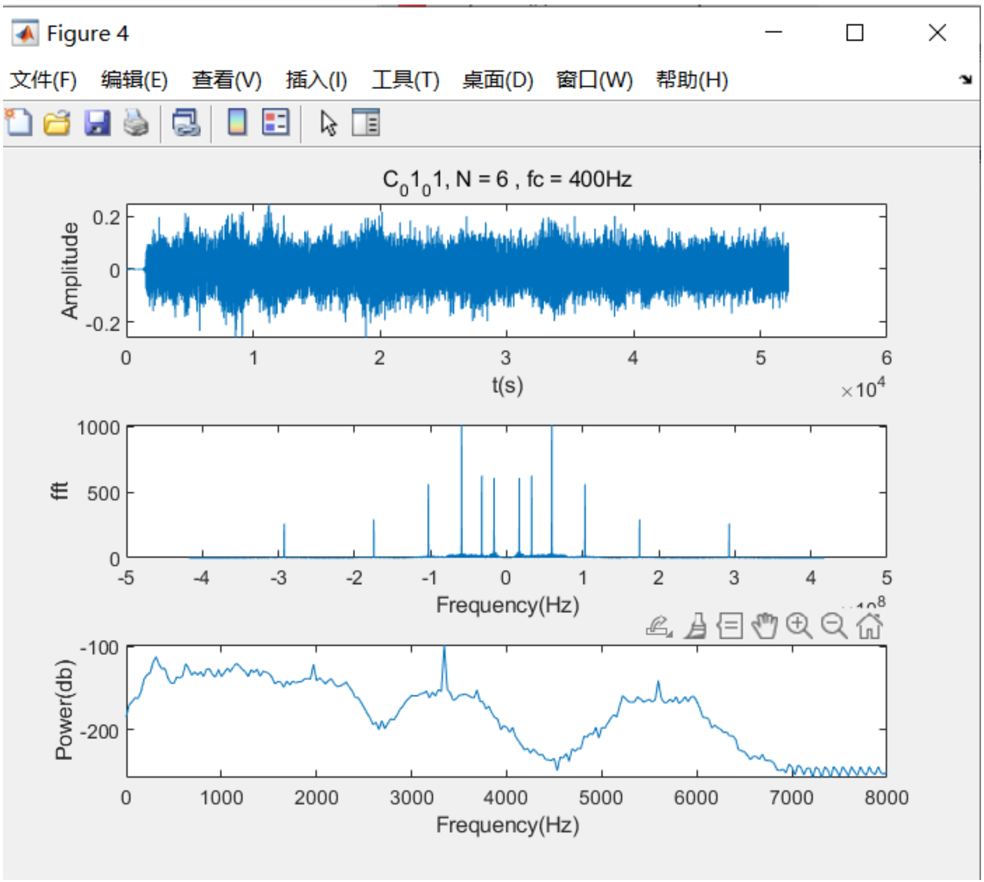
For y1 = C$_0$1$_0$1 $+ SSN1$



Synthetic signal

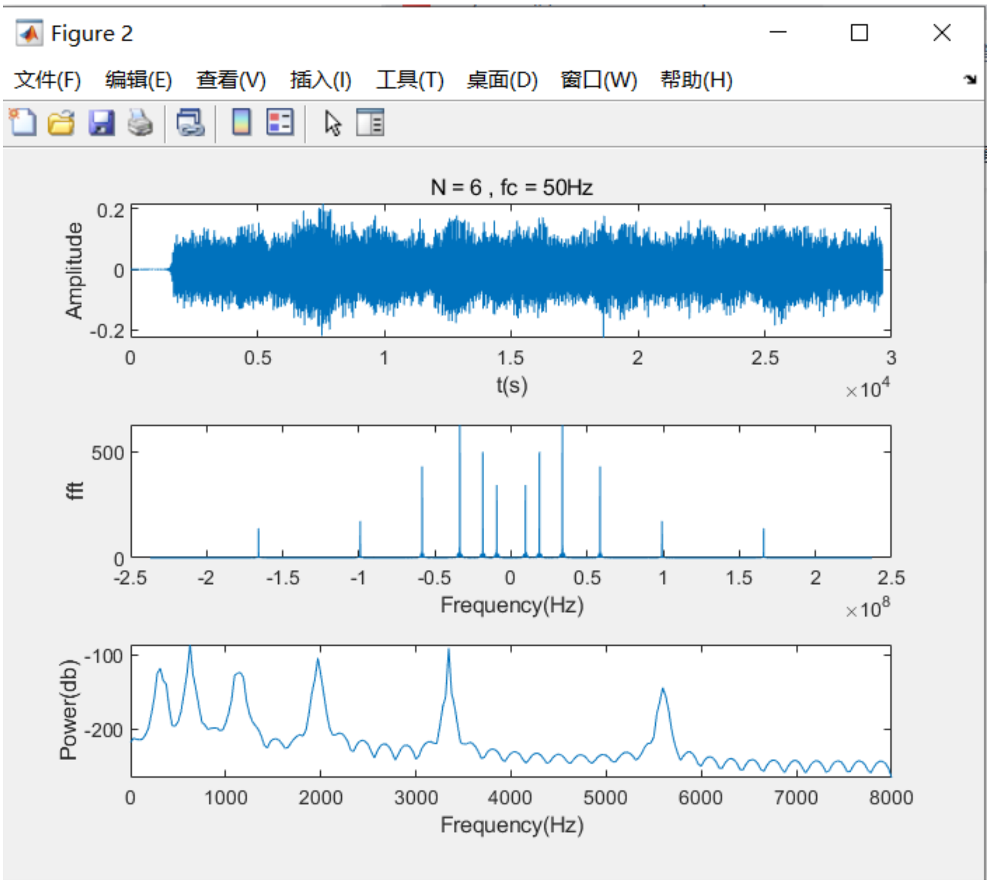


Synthetic signal

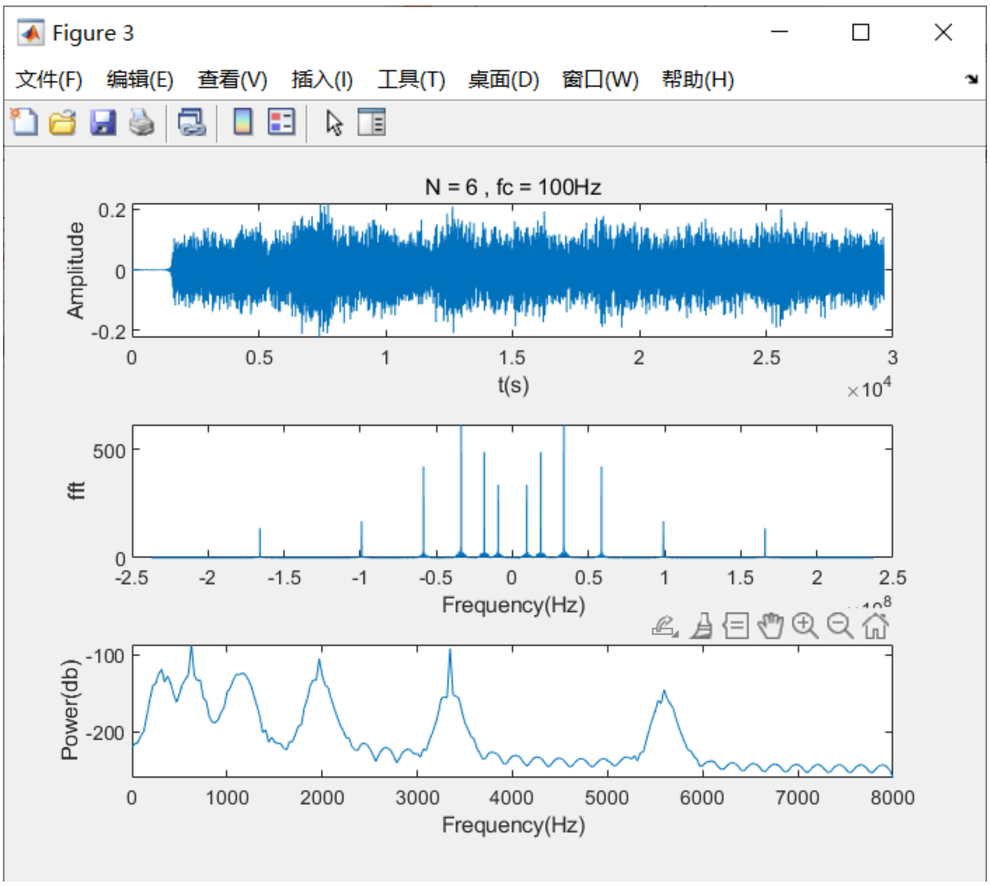


Synthetic signal

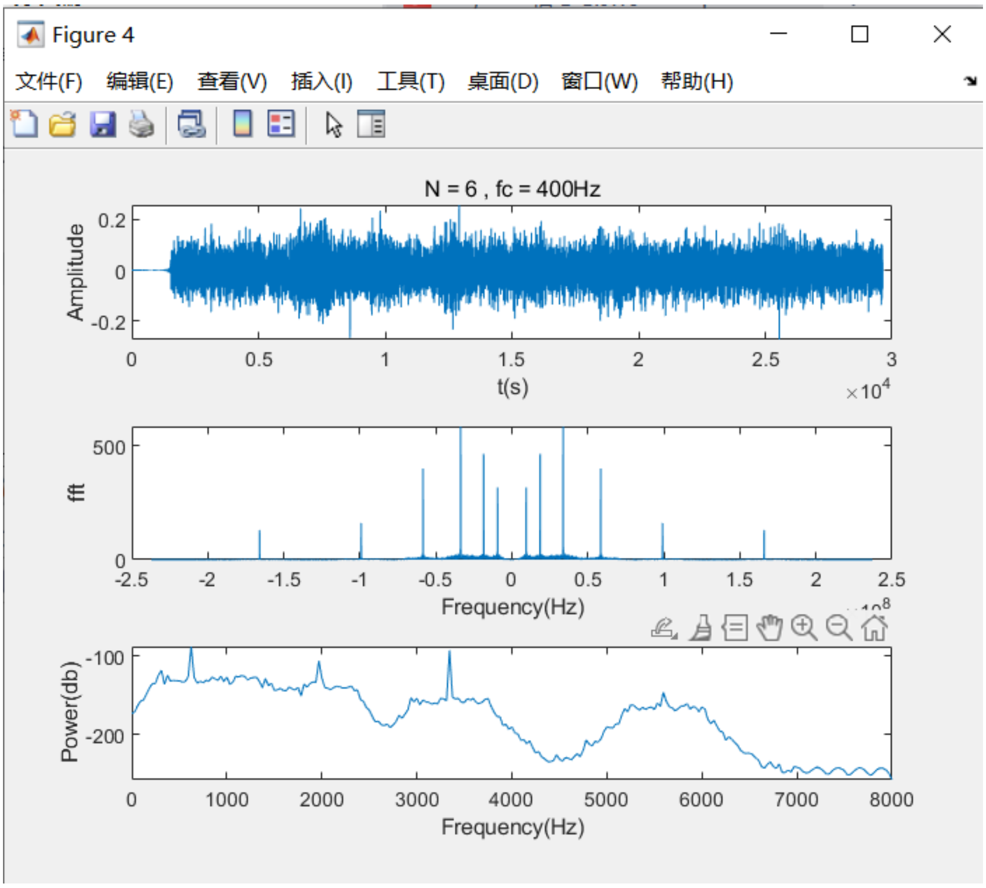
For y2 = $C_0 1_0 2 + SSN2$



Synthetic signal

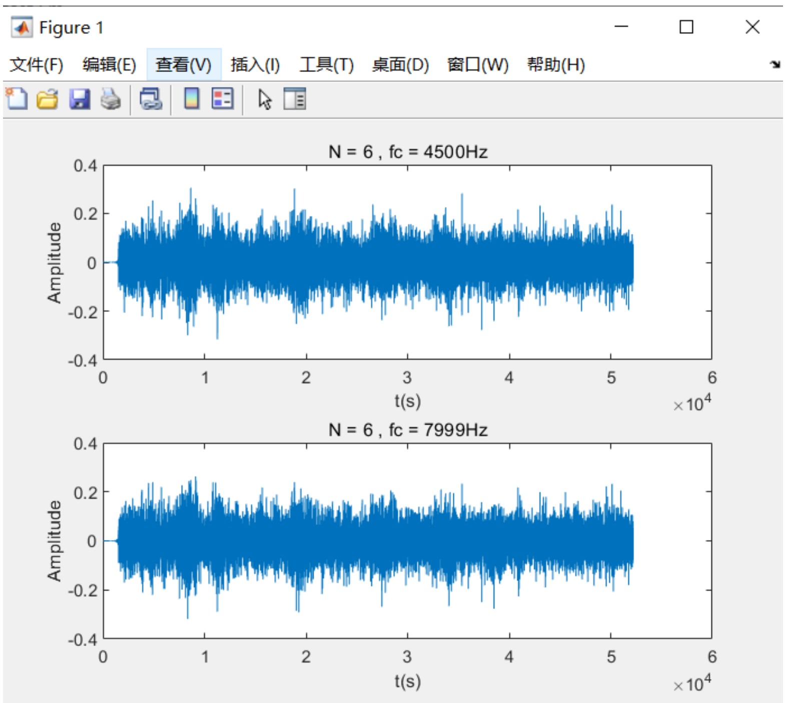


Synthetic signal

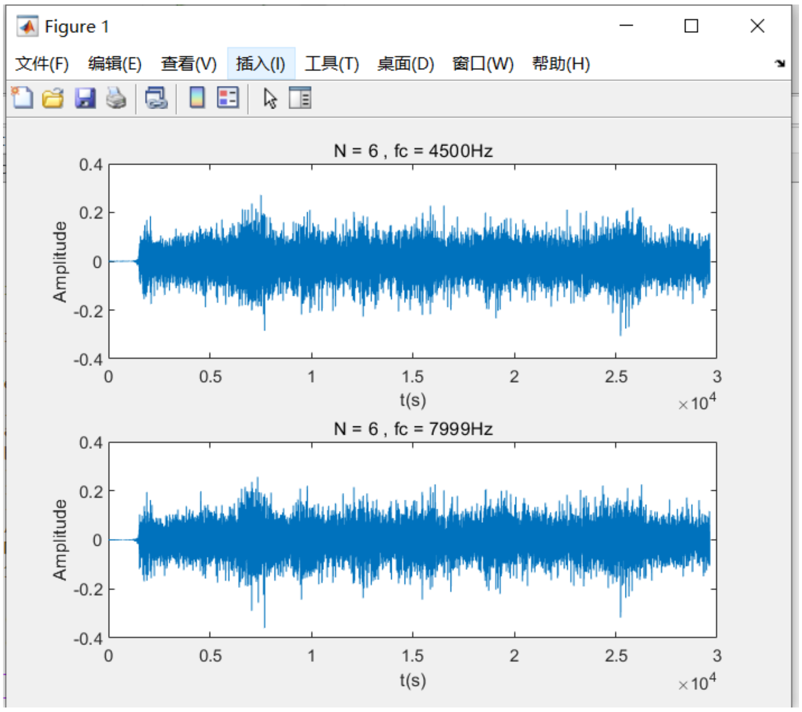


Synthetic signal

3. I change the LPF cut-pff frequency to a higher level for each file in order to find how the LPF cut-off frequency affects the intelligibility of synthesized sentence.



Synthetic signal



Synthetic signal

5.3 Analysis

Compare the result pictures in 1, it is obvious that Speech-shaped-noise greatly interferes the intelligibility of synthesized sentence. In the case where N(N = 6) is same,we can roughly distinguish speech when the LPF cut-off frequency covers from 100Hz to 400Hz. Therefore, we can analyze that as the LPF cut-off frequency increases in a certain range, the signal's identification will increases . I test the LPF cut-off frequency to be 4500Hz and 7999Hz, it is clearer than the previous work. However, because of interference of SSN, it is still very hard to hear the voice clear.

6 Experience

In this project, we applied what we have learned in this course comprehensively, which help us learn more about how to use the knowledge of signal and system to solve some real-world problems. In task1, I discover the effect of the bands not only by hear the generated audio, but also through the wave curve, Fourier Transform in frequency domain and the energy spectrum. My task4 is similar to task2. Both tasks are analyse how the LPF cut-off frequency affects the intelligibility of synthesized sentence. But task4 add SSN in the voice file. I use a function program of task2 to make my work more clear and readable.

7 Code

7.1 Task 1

```

[y1, fs1] = audioread('C_01_01.wav');
[b_low,a_low] = butter(4, 50/(fs1/2), 'low');
```

```
[h, f]= freqz(b_low,a_low,512,fs1);

nfft = 512;
Winodws = hann(nfft);
noverlap = nfft/2;

figure(5);
plot(f, 20 * log10(abs(h)), 'r' );
axis([0 1000 -50 5]);
title('Lowpass Filter Frequency Response');
xlabel('Frequency (Hz)');
ylabel('Gain (dB)');

[Pxx1, w1] = pwelch(y1, Winodws, noverlap, nfft, fs1);
figure(6)
plot(w1, Pxx1);
title('Power Spectral Density of first signal');
xlabel('Frequency (Hz)');
ylabel('Power/Frequency (dB/Hz)');


figure(1)
N = [1, 2, 4, 6, 8, 16];
for i = 1:length(N)
    result_wave = zeros(length(y1),1);
    n = N(i);
    for j = 1:n
        d_low = (27.2784-5.77322) / n * (j-1) + 5.77322;
        d_high = (27.2784-5.77322) / n * (j) + 5.77322;
        low_frequency = 165.4 * (10^(0.06*d_low) - 1);
        high_frequency = 165.4 * (10^(0.06*d_high) - 1);
        wave = get_band(y1, fs1, low_frequency, high_frequency,b_low,
            a_low);
        result_wave = result_wave + wave;
    end
    result_wave = result_wave * norm(y1)/ norm(result_wave);
    fft_result_wave = fftshift(fft(result_wave));
    t = linspace(0, length(result_wave)/fs1, length(result_wave));
    line = linspace(0, fs1, length(fft_result_wave)) - fs1/2;
    subplot(6,1,i);
    [Pxx1, w1] = pwelch(result_wave, Winodws, noverlap, nfft, fs1);
    plot(w1, Pxx1);
```

```
    title('Power Spectral Density');
    xlabel('Frequency (Hz)');
    ylabel('Power/Frequency (dB/Hz)');

end

fft_y1  = fftshift(fft(y1));
t = linspace(0, length(y1)/fs1, length(y1));
line = linspace(0, fs1, length(fft_y1)) - fs1/2;
figure(2)
subplot(2,1,1);
plot(t, y1);
xlabel('Time(s)');
ylabel('Amplitude');
subplot(2,1,2);
plot(line, abs(fft_y1));
xlabel('Frequency');
ylabel('Amplitude');


[y2, fs2] = audioread('C_01_02.wav');

[b_low,a_low] = butter(4, 50/(fs2/2), 'low');

[Pxx2, w2] = pwelch(y2, Winodws, noverlap, nfft, fs2);
figure(7)
plot(w1, Pxx1);
title('Power Spectral Density of second signal');
xlabel('Frequency (Hz)');
ylabel('Power/Frequency (dB/Hz)');



figure(3)
N = [1, 2, 4, 6, 8, 16];
%N = [1];
for i = 1:length(N)
    result_wave = zeros(length(y2),1);
    n = N(i);
    for j = 1:n
        d_low = (27.2784-5.77322) / n * (j-1) + 5.77322;
        d_high = (27.2784-5.77322) / n * (j) + 5.77322;
        low_frequency = 165.4 * (10^(0.06*d_low) - 1)
```

```
        high_frequency = 165.4 * (10^(0.06*d_high) - 1)
        wave = get_band(y2, fs2, low_frequency, high_frequency,b_low,
            a_low);
        result_wave = result_wave + wave;
    end
    result_wave = result_wave * norm(y2)/ norm(result_wave);
    fft_result_wave = fftshift(fft(result_wave));
    t = linspace(0, length(result_wave)/fs2, length(result_wave));
    line = linspace(0, fs2, length(fft_result_wave)) - fs2/2;
    subplot(6,1,i);
    [Pxx2, w2] = pwelch(result_wave, Winodws, noverlap, nfft, fs2);
    plot(w2, Pxx2);
    title('Power Spectral Density');
    xlabel('Frequency (Hz)');
    ylabel('Power/Frequency (dB/Hz)');
    audiowrite(['C_01_02_', num2str(n), '.wav'], result_wave, fs1);
end

% figure(4);
fft_y2  = fftshift(fft(y2));
t = linspace(0, length(y2)/fs2, length(y2));
line = linspace(0, fs2, length(fft_y2)) - fs2/2;



function wave = get_band(s, fs, low_frequency, high_frequency,b_low,
    a_low)
    len = length(s)/fs;
    t = (0:1/fs:len - 1/fs);
    [b, a] = butter(4, [low_frequency, high_frequency] / (fs / 2));
    [h,f] = freqz(b,a,512,fs);
    % figure(6)
    % plot(f, 20 * log10(abs(h)), 'r' );
    % axis([0 2000 -50 5]);
    % title('Bandpass Filter Frequency Response(201.8Hz, 437.1Hz)');
    % xlabel('Frequency (Hz)');
    % ylabel('Gain (dB)');

    y = filter(b, a, s);
    y = abs(y);
    mid_frequency = (low_frequency + high_frequency) / 2;
    sin_wave = sin(2 * pi * mid_frequency * t);
    sin_wave = sin_wave.';
```

```
    envelope = filter(b_low, a_low, y);
    wave = envelope .* sin_wave;
    % wave = envelope;
end
```

7.2 Task 2

```
clc
clear
%the number of bands
N = 4;
%get the origin signal
[x1, fs1] = audioread('C_01_01.wav');
[x2, fs2] = audioread('C_01_02.wav');
%the lower bound of the frequency
fl = 200;
%the higher bound of the frequency
fh = 7000;
%enum cut-off frequencies of the LPF
fc1 = 20;
fc2 = 50;
fc3 = 100;
fc4 = 400;
x1 = x1';
x2 = x2';
%get the new signals
y11 = get_wave(x1, fs1, N, fl, fh, fc1);
y12 = get_wave(x1, fs1, N, fl, fh, fc2);
y13 = get_wave(x1, fs1, N, fl, fh, fc3);
y14 = get_wave(x1, fs1, N, fl, fh, fc4);

y21 = get_wave(x2, fs2, N, fl, fh, fc1);
y22 = get_wave(x2, fs2, N, fl, fh, fc2);
y23 = get_wave(x2, fs2, N, fl, fh, fc3);
y24 = get_wave(x2, fs2, N, fl, fh, fc4);

%time scale for the new signals
t1 = (0:1/fs1:length(x1)/fs1 - 1/fs1);
t2 = (0:1/fs2:length(x2)/fs2 - 1/fs2);

% plot new signals in time filed
figure(2)
```

```
subplot(411)
plot(t1,y11),title('fc = 20Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(412)
plot(t1,y12),title('fc = 50Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(413)
plot(t1,y13),title('fc = 100Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(414)
plot(t1,y14),title('fc = 400Hz'),xlabel('t(s)'),ylabel('amplitude')

figure(3)
subplot(411)
plot(t2,y21),title('fc = 20Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(412)
plot(t2,y22),title('fc = 50Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(413)
plot(t2,y23),title('fc = 100Hz'),xlabel('t(s)'),ylabel('amplitude')
subplot(414)
plot(t2,y24),title('fc = 400Hz'),xlabel('t(s)'),ylabel('amplitude')

%define the x-axis fot the fourier transform of the new signals
freq1 = linspace(-fs1/2,fs1/2-1,length(x1));
freq2 = linspace(-fs2/2,fs2/2-1,length(x2));

%plot the fourier transform
figure(4)
subplot(411)
plot(freq1,fftshift(abs(fft(y11./(length(y11)))))),title('fc = 20Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(412)
plot(freq1,fftshift(abs(fft(y12./(length(y12)))))),title('fc = 50Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(413)
plot(freq1,fftshift(abs(fft(y13./(length(y13)))))),title('fc = 100Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(414)
plot(freq1,fftshift(abs(fft(y14./(length(y14)))))),title('fc = 400Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')

figure(5)
subplot(411)
plot(freq2,fftshift(abs(fft(y21./(length(y21)))))),title('fc = 20Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(412)
```

```
plot(freq2,fftshift(abs(fft(y22./(length(y22))))),title('fc = 50Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(413)
plot(freq2,fftshift(abs(fft(y23./(length(y23))))),title('fc = 100Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')
subplot(414)
plot(freq2,fftshift(abs(fft(y24./(length(y24))))),title('fc = 400Hz'),
    xlabel('frequency(Hz)'),ylabel('fft')

%calculate the power of the new signals
[p11,w11] = pwelch(y11,[],[],512,fs1);
[p12,w12] = pwelch(y12,[],[],512,fs1);
[p13,w13] = pwelch(y13,[],[],512,fs1);
[p14,w14] = pwelch(y14,[],[],512,fs1);
[p21,w21] = pwelch(y21,[],[],512,fs2);
[p22,w22] = pwelch(y22,[],[],512,fs2);
[p23,w23] = pwelch(y23,[],[],512,fs2);
[p24,w24] = pwelch(y24,[],[],512,fs2);

%plot new signals in power field
figure(6)
subplot(411)
plot(w11,p11),title('fc = 20Hz'),xlabel('frequency(Hz)'),ylabel('power(
    W)')
subplot(412)
plot(w12,p12),title('fc = 50Hz'),xlabel('frequency(Hz)'),ylabel('power(
    W)')
subplot(413)
plot(w13,p13),title('fc = 100Hz'),xlabel('frequency(Hz)'),ylabel('power
    (W)')
subplot(414)
plot(w14,p14),title('fc = 400Hz'),xlabel('frequency(Hz)'),ylabel('power
    (W)')

figure(7)
subplot(411)
plot(w21,p21),title('fc = 20Hz'),xlabel('frequency(Hz)'),ylabel('power(
    W)')
subplot(412)
plot(w22,p22),title('fc = 50Hz'),xlabel('frequency(Hz)'),ylabel('power(
    W)')
subplot(413)
plot(w23,p23),title('fc = 100Hz'),xlabel('frequency(Hz)'),ylabel('power
```

```
    (W)')
subplot(414)
plot(w24,p24),title('fc = 400Hz'),xlabel('frequency(Hz)'),ylabel('power
    (W)')

%record the new signals
audiowrite('P1_1_fc_=_20.wav',y11,fs1);
audiowrite('P1_1_fc_=_50.wav',y12,fs1);
audiowrite('P1_1_fc_=_100.wav',y13,fs1);
audiowrite('P1_1_fc_=_400.wav',y14,fs1);

audiowrite('P1_2_fc_=_20.wav',y21,fs2);
audiowrite('P1_2_fc_=_50.wav',y22,fs2);
audiowrite('P1_2_fc_=_100.wav',y23,fs2);
audiowrite('P1_2_fc_=_400.wav',y24,fs2);

%encapsulate the process with a function
function res = get_wave(in, fs, N, fl, fh, fc)
  %predefine the size of the result
  res = zeros(1,length(in));
  %calculte the physical distance of human ear corresponding to given frequencies
  dl = log10(fl/165.4 + 1)/0.06;
  dh = log10(fh/165.4 + 1)/0.06;
  d = (dh - dl)/N;
  len = length(in)/fs;
  %define the time scale for the sine signals
  t = (0:1/fs:len - 1/fs);
  for n = 1:N
      %calculate the lower bound and the higher bound of the band-pass
      %filer
      l = 165.4*(10^(0.06*(dl + d*(n-1)))-1);
      h = 165.4*(10^(0.06*(dl + d*n))-1);
      [b, a] = butter(4, [l, h]/(fs/2));
      wave = filter(b, a, in);
      %full-wave recitification
      wave = abs(wave);
      [x, y] = butter(4, fc/(fs/2));
      %plot the certification for the filter
      if (n == 1)
      [h1 ,f1] = freqz(b,a,512,fs);
      [h2 ,f2] = freqz(x,y,512,fs);
      figure(1)
      subplot(211)
```

```
        plot(f1,20*log10(abs(h1)),'r'),axis([0 1400 -10 2]),title(sprintf("the range of frequency of the band-pass filter is from %f Hz to xlabel('Freqency(Hz)'),ylabel('Magnitude(dB)')
        subplot(212)
        plot(f2,20*log10(abs(h2)),'r'),axis([0 1400 -10 2]),title(sprintf("the cut-off frequency of the low-pass filter is %f Hz",fc));
        xlabel('Freqency(Hz)'),ylabel('Magnitude(dB)')
        end
        env = filter(x, y, wave);
        sin_sig = sin(2 * pi * (l + (h - l)/2)* t);
        %sum up
        res = sin_sig.* env + res;
    end
    %normalization
    res = res*norm(in)/norm(res);
end



%Q2
```

7.3 Task 3

```
clear;
clc;
[y1, fs1] = audioread('C_01_01.wav');
[y2, fs2] = audioread('C_01_02.wav');
sig1 = repmat(y1, 1, 10);
sig2 = repmat(y2, 1, 10);
nfft = 512;
Winodws = hann(nfft);
noverlap = nfft/2;
[Pxx1, w1] = pwelch(sig1, Winodws, noverlap, nfft, fs1);
[Pxx2, w2] = pwelch(sig2, Winodws, noverlap, nfft, fs2);
b1 = fir2(3000, w1/(fs1/2), sqrt(Pxx1/max(Pxx1)));
b2 = fir2(3000, w2/(fs2/2), sqrt(Pxx2/max(Pxx2)));
[h1, wh1] = freqz(b1, 1, 128);
[h2, wh2] = freqz(b2, 1, 128);
noise_1 = 1-2*rand(1,52215);
noise_2 = 1-2*rand(1,29636);
ssn1 = filter(b1, 1, noise_1)./1.065;
ssn2 = filter(b2, 1, noise_2)./1.53;
```

```
SNR1=20*log10(norm(y1)/norm(ssn1'))
SNR2=20*log10(norm(y2)/norm(ssn2'))
k1=y1+ssn1';
k2=y2+ssn2';
[Pyy1,w1] =pwelch(ssn1',Winodws, noverlap, nfft, fs2);
[Pyy2,w2] =pwelch(k2,Winodws, noverlap, nfft, fs2);
[b_low,a_low] = butter(4, 50/(fs1/2), 'low');
 N = [1, 2, 4, 6, 8, 16, 32, 64, 128, 256];
for i = 1:length(N)
    result_wave1 = zeros(length(y1),1);
    result_wave2 = zeros(length(y1),1);
    n = N(i);
    for j = 1:n
        d_low = (27.2784-5.77322) / n * (j-1) + 5.77322;
        d_high = (27.2784-5.77322) / n * (j) + 5.77322;
        low_frequency = 165.4 * (10^(0.06*d_low) - 1);
        high_frequency = 165.4 * (10^(0.06*d_high) - 1);
        wave1 = get_band(ssn1', fs1, low_frequency, high_frequency,
            b_low, a_low);
         wave2 = get_band(k1, fs1, low_frequency, high_frequency,b_low,
            a_low);
        result_wave1 = result_wave1 + wave1;
        result_wave2 = result_wave2 + wave2;
    end
    result_wave1 = result_wave1 * norm(ssn1')/ norm(result_wave1);
    result_wave2 = result_wave2 * norm(k1)/ norm(result_wave2);
    audiowrite(['(n)C_01_01_', num2str(n), '.wav'], result_wave1, fs1);
    audiowrite(['(k)C_01_01_', num2str(n), '.wav'], result_wave2, fs1);
end
[b_low,a_low] = butter(4, 50/(fs2/2), 'low');
N = [1, 2, 4, 6, 8, 16, 32, 64, 128, 256];
for i = 1:length(N)
    result_wave1 = zeros(length(y2),1);
    result_wave2 = zeros(length(y2),1);
    n = N(i);
    for j = 1:n
        d_low = (27.2784-5.77322) / n * (j-1) + 5.77322;
        d_high = (27.2784-5.77322) / n * (j) + 5.77322;
        low_frequency = 165.4 * (10^(0.06*d_low) - 1);
        high_frequency = 165.4 * (10^(0.06*d_high) - 1);
        wave1 = get_band(ssn2', fs2, low_frequency, high_frequency,
            b_low, a_low);
         wave2 = get_band(k2, fs2, low_frequency, high_frequency,b_low,
```

```
                a_low);
            result_wave1 = result_wave1 + wave1;
            result_wave2 = result_wave2 + wave2;
        end
        result_wave1 = result_wave1 * norm(ssn2')/ norm(result_wave1);
        result_wave2 = result_wave2 * norm(k2)/ norm(result_wave2);
        audiowrite(['(n)C_01_02_', num2str(n), '.wav'], result_wave1, fs2);
        audiowrite(['(k)C_01_02_', num2str(n), '.wav'], result_wave2, fs2);
end
 audiowrite(['C_01_01_ssn1.wav'],ssn1, fs1);
   audiowrite(['C_01_02_ssn2.wav'], ssn2, fs2);
 audiowrite(['C_01_01_total1.wav'],k1, fs1);
   audiowrite(['C_01_02_total2.wav'], k2, fs2);
[yx2, fsx2] = audioread('(k)C_01_02_2.wav');
player1 = audioplayer(yx2, fsx2);
play(player1)
[Pyy2,w2] =pwelch(yx2,Winodws, noverlap, nfft, fsx2);
subplot(3,1,1)
t2=linspace(0,2,29636);
plot(t2,yx2);
xlabel('t(s)')
ylabel('Amplitude');
title("C2,N=2 ,fc=50hz",'fontsize', 13)
subplot(3,1,2)
plot(w2, Pyy2);
xlabel('Frequency (Hz)')
ylabel('Power(w)');
subplot(3,1,3)
xx2=fft(yx2)./length(yx2);
t2=linspace(-8000,8000,29636);
plot(t2,abs(fftshift(xx2)));
xlabel('Frequency(hz)')
ylabel('fft');
[yx1, fsx1] = audioread('(k)C_01_01_2.wav');
[Pyy1,w1] =pwelch(yx1,Winodws, noverlap, nfft, fsx1);
subplot(3,1,1)
t1=linspace(0,3.5,52215);
plot(t1,yx1);
xlabel('t(s)')
ylabel('Amplitude');
title("C1,N=2 ,fc=50hz",'fontsize', 13)
subplot(3,1,2)
plot(w1, Pyy1);
```

```
xlabel('Frequency (Hz)')
ylabel('Power(w)');
subplot(3,1,3)
xx1=fft(yx1)./length(yx1);
t1=linspace(-8000,8000,52215);
plot(t1,abs(fftshift(xx1)));
xlabel('Frequency(hz)')
ylabel('fft');
[Pyy1,w1] =pwelch(k1,Winodws, noverlap, nfft, fsx1);
[Pyy2,w2] =pwelch(ssn2',Winodws, noverlap, nfft, fsx2);
subplot(3,1,1)
t1=linspace(0,3.5,52215);
plot(t1,k1);
xlabel('t(s)')
ylabel('Amplitude');
title("totalC1",'fontsize', 13)
subplot(3,1,2)
plot(w1, Pyy1);
xlabel('Frequency (Hz)')
ylabel('Power(w)');
subplot(3,1,3)
xx1=fft(k1)./length(k1);
t1=linspace(-8000,8000,52215);
plot(t1,abs(fftshift(xx1)));
xlabel('Frequency(hz)')
ylabel('fft');
[yx1, fsx1] = audioread('C_01_02_2.wav');
[yx2, fsx2] = audioread('C_01_02_16.wav');
subplot(2,1,1)
t2=linspace(0,2,29636);
plot(t2,yx1);
axis([0,2,-0.5,0.5]);
xlabel('t(s)')
ylabel('Amplitude');
title("C2,N=2,fc=50hz")
subplot(2,1,2)
t1=linspace(0,2,29636);
plot(t2,yx2);
axis([0,2,-0.5,0.5]);
xlabel('t(s)')
ylabel('Amplitude');
title("C2,N=16,fc=50hz")
```

7.4 Task 4

```

clc;
clear;

%Set the number of bands to N = 6
N = 6;
%Changing LPF Cutoff
fl = 200;
fh = 7000;
fc1 = 20;
fc2 = 50;
fc3 = 100;
fc4 = 400;

%Generate a noisy signal at SNR -5dB

[s1,fs1] = audioread('C_01_01.wav');
sig1 = s1';
[pxx1,w1] = pwelch(sig1,[],[],512,fs1);
b1 = fir2(3000,w1/(fs1/2),sqrt(pxx1/max(pxx1)));
noise1 = 1-2*rand(1,length(s1));
SSN1 = filter(b1,1,noise1)./0.65;
E1 = norm(sig1);
E2 = norm(SSN1);
SSN1 = sqrt(E1/E2)*SSN1;
y1 = sig1 + SSN1;
y1 = y1/norm(y1)*norm(sig1);
%sound(y1,fs1);
y11 = get_wave(y1, fs1, 6, fl, fh, fc1);
y12 = get_wave(y1, fs1, 6, fl, fh, fc2);
y13 = get_wave(y1, fs1, 6, fl, fh, fc3);
y14 = get_wave(y1, fs1, 6, fl, fh, fc4);

audiowrite('P1_1_fc_=_20.wav',y11,fs1);
audiowrite('P1_1_fc_=_50.wav',y12,fs1);
audiowrite('P1_1_fc_=_100.wav',y13,fs1);
audiowrite('P1_1_fc_=_400.wav',y14,fs1);

figure(1)
subplot(3,1,1)
plot(1:length(y1),y11)
```

```
title('C_01_01, N = 6 , fc = 20Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs1*(-length(y1)/2:length(y1)/2-1);
plot(f,fftshift(abs(fft(y11))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx1,w1] = pwelch(y11,[],[],512,fs1);
plot(w1,20*log10(pxx1))
xlabel('Frequency(Hz)');ylabel('Power(db)')

figure(2)
subplot(3,1,1)
plot(1:length(y1),y12)
title('C_01_01, N = 6 , fc = 50Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs1*(-length(y1)/2:length(y1)/2-1);
plot(f,fftshift(abs(fft(y12))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx1,w1] = pwelch(y12,[],[],512,fs1);
plot(w1,20*log10(pxx1))
xlabel('Frequency(Hz)');ylabel('Power(db)')


figure(3)
subplot(3,1,1)
plot(1:length(y1),y13)
title('C_01_01, N = 6 , fc = 100Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs1*(-length(y1)/2:length(y1)/2-1);
plot(f,fftshift(abs(fft(y13))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx1,w1] = pwelch(y13,[],[],512,fs1);
plot(w1,20*log10(pxx1))
xlabel('Frequency(Hz)');ylabel('Power(db)')

figure(4)
subplot(3,1,1)
plot(1:length(y1),y14)
```

```
title('C_01_01, N = 6 , fc = 400Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs1*(-length(y1)/2:length(y1)/2-1);
plot(f,fftshift(abs(fft(y14))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx1,w1] = pwelch(y14,[],[],512,fs1);
plot(w1,20*log10(pxx1))
xlabel('Frequency(Hz)');ylabel('Power(db)')


clc;
clear;

%Set the number of bands to N = 6
N = 6;
%Changing LPF Cutoff
fl = 200;
fh = 7000;
fc1 = 20;
fc2 = 50;
fc3 = 100;
fc4 = 400;
%Generate a noisy signal at SNR -5dB
[s2,fs2] = audioread('C_01_02.wav');
sig2 = s2';
[pxx2,w2] = pwelch(sig2,[],[],512,fs2);
b2 = fir2(3000,w2/(fs2/2),sqrt(pxx2/max(pxx2)));
noise2 = 1-2*rand(1,length(s2));
SSN2 = filter(b2,1,noise2)./0.7;
E3 = norm(sig2);
E4 = norm(SSN2);
SSN2 = sqrt(E3/E4)*SSN2;
y2 = sig2 + SSN2;
y2 = y2/norm(y2)*norm(sig2);
%sound(y2,fs2)
y21 = get_wave(y2, fs2, 6, fl, fh, fc1);
y22 = get_wave(y2, fs2, 6, fl, fh, fc2);
y23 = get_wave(y2, fs2, 6, fl, fh, fc3);
y24 = get_wave(y2, fs2, 6, fl, fh, fc4);

audiowrite('P1_2_fc_=_20.wav',y21,fs2);
```

```
audiowrite('P1_2_fc_=_50.wav',y22,fs2);
audiowrite('P1_2_fc_=_100.wav',y23,fs2);
audiowrite('P1_2_fc_=_400.wav',y24,fs2);

figure(1)
subplot(3,1,1)
plot(1:length(y2),y21)
title('N = 6 , fc = 20Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs2*(-length(y2)/2:length(y2)/2-1);
plot(f,fftshift(abs(fft(y21))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx2,w1] = pwelch(y21,[],[],512,fs2);
plot(w1,20*log10(pxx2))
xlabel('Frequency(Hz)');ylabel('Power(db)')

figure(2)
subplot(3,1,1)
plot(1:length(y2),y22)
title('N = 6 , fc = 50Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs2*(-length(y2)/2:length(y2)/2-1);
plot(f,fftshift(abs(fft(y22))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx2,w1] = pwelch(y22,[],[],512,fs2);
plot(w1,20*log10(pxx2))
xlabel('Frequency(Hz)');ylabel('Power(db)')

figure(3)
subplot(3,1,1)
plot(1:length(y2),y23)
title('N = 6 , fc = 100Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs2*(-length(y2)/2:length(y2)/2-1);
plot(f,fftshift(abs(fft(y23))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx2,w1] = pwelch(y23,[],[],512,fs2);
```

```
plot(w1,20*log10(pxx2))
xlabel('Frequency(Hz)');ylabel('Power(db)')

figure(4)
subplot(3,1,1)
plot(1:length(y2),y24)
title('N = 6 , fc = 400Hz')
xlabel('t(s)');ylabel('Amplitude')
subplot(3,1,2)
f = fs2*(-length(y2)/2:length(y2)/2-1);
plot(f,fftshift(abs(fft(y24))))
xlabel('Frequency(Hz)');ylabel('fft')
subplot(3,1,3)
[pxx2,w1] = pwelch(y24,[],[],512,fs2);
plot(w1,20*log10(pxx2))
xlabel('Frequency(Hz)');ylabel('Power(db)')

%Function need to be used
function res = get_wave(in, fs, N, fl, fh, fc)
  res = zeros(1,length(in));
  dl = log10(fl/165.4 + 1)/0.06;
  dh = log10(fh/165.4 + 1)/0.06;
  d = (dh - dl)/N;
  len = length(in)/fs;
  t = (0:1/fs:len - 1/fs);
  for n = 1:N
      l = 165.4*(10^(0.06*(dl + d*(n-1)))-1);
      h = 165.4*(10^(0.06*(dl + d*n))-1);
      [b, a] = butter(4, [l, h]/(fs/2));
      wave = filter(b, a, in);
      wave = abs(wave);
      [x, y] = butter(4, fc/(fs/2));
      env = filter(x, y, wave);
      sin_sig = sin(2 * pi * (l + (h - l)/2)* t);
      res = sin_sig.* env + res;
  end
  res = res*norm(in)/norm(res);
end
```