**Project1：Speech synthesis and perception with envelope cue**

|  |  |
| --- | --- |
| **Author** | Name: 谭雍昊 柳嘉伟 郭骏德 彭思杰  Student ID: 11910204 11910123 11910206 11910702 |
| **Introduction**   1. Learn how to synthesize a speech signal based on multi-band envelope cues. 2. Change the numbers of bands and the LPF cutoff frequency to gain the optimum solution. 3. Using different filters provided by MATLAB for comparison. 4. Exploring MATLAB’s GUI and EXE package function. 5. Critical thinking of speech synthesis.   **Project results & Analysis**：  **Requirement task-1**      Fig task1-a    Fig task1-b    Fig task1-c    Fig task1-d    Fig task1-e    Fig task1-f    Fig task1-g    Fig task1-h  sb_1_1sb_1_2  Fig task1-i Fig task1-j  sb_1_3sb_1_64  Fig task1-k Fig task1-l  sb_1_118sb_1_120  Fig task1-m Fig task1-n  sb_1_124  Fig task1-o  **Analysis task-1**  Set the LPF cutoff frequency equal to 50Hz and the number of bands as the only variable,we do trials for N=1,2,4,6,8. During the process of generating the Fig task1-a and Fig task1-b, we can also get the sound for the conditions of different numbers of bands. From the wave files we can directly judge that when N=8, the sound we get is closest to the original compared to other smaller numbers of bands. The reasons are directly shown in the Fig task1-c and Fig task1-d, which are the PSD(power spectrum density)of the signal for different N. Ideally, the larger is N, the more points we can get from the original signal, the better we can restore the signal. Similarly, from Fig task1-e and Fig task1-f, which are the frequency spectrum of the signals , N is also related to the points we can get to for the envelop of the signal, the bigger N is, the better the restored signal can be.  However, in the real case, we find that when N is too large the signal we get will also distort. From Fig task1-g and task1-h, we can see that when N is too large, the loss of low frequency signal is very obvious, while producing high frequency squeal. And from the observation from Fig task1-i to Fig task1-o, we may can attribute this to the transition band of the real filter, which affects the results dominantly when N is large.Meanwhile, what we find it interesting is that when N=118, there aren’t obvious squeals, while the squeals occur when N=119-120. However, the high frequency squeals disappear when N=121-124, while the squeals occur again then N=125. After testing, we find that the filter-shaped frequency band is related to those of the first band-pass filter. Therefore, we suppose the deformation of the butterwidth filter will contribute to the collapse of the low frequency signal firstly.  **Requirement task-2**      Fig task2-a    Fig task2-b    Fig task2-c    Fig task2-d    Fig task2-e    Fig task2-f  sb_2_5  Fig task2-g  sb_2_50  Fig task2-h  sb_2_100  Fig task2-i  sb_2_500  Fig task2-j  sb_3_2  Fig task2-k  sb_3_20  Fig task2-l  sb_3_200  Fig task2-m  sb_3_500  Fig task2-n  **Analysis task-2**  Set the number of the bands equal to 4 and LPF cutoff frequency as the only variable,we do trials for fcutoff=20,50,100,200,400Hz. From Fig task2-a to Fig task2-f, with the increase of the cutoff frequency, the envelop we get becomes more closer to the original signal with the increase of relatively higher frequency. For the frequency spectrum, however, almost remain the same because N doesn’t change. Also, from the observation from Fig task2-g to Fig task2-j, we find that when fcutoff increases, the burrs of the signal increases, showing the increase of high frequency signal,while a very large fcutoff causes the voice to be very emotional due to the loss of high frequency signal. Furthermore, when fcutoff is small but enough to be recognized, there’s no need for a very large fcutoff because a very large fcutoff makes the envelop very close to the original signal, which can be directly illustrated from Fig task2-k to Fig task2-n. However, this also means losing the advantage of envelop---the accessibility to be transported easily. Thus we think wehn fcutoff=50, it’s enough to be recognized.  **Requirement task-3**      Fig task3-a    Fig task3-b    Fig task3-c    Fig task3-d    Fig task3-e    Fig task3-f  **Analysis task-3**  Set SNR equal to -5dB, LPF cutoff frequency equal to 50Hz, N=2,4,6,8,16 respectively. Even though the noise in the background interfere with the signals, we can still get a clearer voice signal with a larger number of N when we draw the Fig task3-a and Fig task3-b simultaneously. The conclusions we can draw from Fig task3-c, Fig task3-d and Fig task3-e, Fig task3-f are similar to the conclusions in task1. The number of peaks we can get in the frequency spectrum and PSD is directly related to the number of N, the larger N is, the more peaks we can get in the PSD and frequency spectrum, the clearer the voice signal we can get, and there are actually not much difference compared to the PSDs and frequency spectrum in the task1.  **Requirement task-4**      Fig task4-a    Fig task4-b    Fig task4-c    Fig task4-d    Fig task4-e    Fig task4-f  **Analysis task-4**  Set SNR equal to -5dB, N equal to 6, LPF =2,4,6,8,16 respectively. Similar to task2, while the cutoff frequency increases, the envelop we get becomes more closer to the original signal with the increase of relatively higher frequency. For the frequency spectrum still almost remain the same because N doesn’t change, whereas the quality of the voice is poorer than the voice without SSN(speech-shaped signal).  **Investigation beyond project tasks (Bonus)**   1. Comparison and analysis of Butterworth, Chebyshev I, Chebyshev II, Bessel and Elliptic filters.   **Analysis**  D:\SUSTECH2020春\信号和系统\lab\proj\巴特沃夫N=8（对比且比）.png巴特沃夫N=8（对比且比）  Fig I-1  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫N=8（对比巴特）.png切比雪夫N=8（对比巴特）  Fig I-2  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫二型N=8（对比巴特）.png切比雪夫二型N=8（对比巴特）  Fig I-3  untitled  Fig I-4  D:\SUSTECH2020春\信号和系统\lab\proj\椭圆N=8（对比巴特）.png椭圆N=8（对比巴特）  Fig I-5  Introduction of filters:  The Butterworth filter: a filter with flat passband and stopband. The passband rate response curve is the flattest without fluctuations. The stopband frequency band gradually drops to zero. The drop is slow, which is easy to cause in the transition band. distortion.  The Chebyshev filter: a filter whose frequency response amplitude and other ripple fluctuations (flat passband, stopband and other ripples or stopband flat, passband and other ripples) in the passband or stopband. The amplitude characteristics in the passband are Wait for ripples. The filter whose frequency response amplitude equals ripple fluctuations in the pass band (or "pass band") is called "I-type Chebyshev filter", and the frequency response amplitude equal ripples in the stop band (or "stop band") The wave filter is called "Type II Chebyshev filter".  The Bessel filter: a linear filter with the flattest group delay (linear phase response), that is, the flattest amplitude and phase response, and is commonly used in audio crossover systems. Has the flattest amplitude and phase response. The phase response of the bandpass (usually the area of interest of the user) is almost linear.  Elliptic filter: a filter with ripple characteristics in passband and stopband, so the passband and stopband have good approximation characteristics.  Comparison of filters:  At the same order: the amplitude-frequency curve of the elliptic filter has the steepest drop, followed by the Chebyshev filter, and then the Butterworth filter, and the Bessel filter has the gentlest drop.  At the same order: Butterworth filter passband is the flattest, stopband drops slowly; Chebyshev filter passband and other ripples, stopband drops faster; Bessel filter passband and other ripples, stopband Slow down. That is to say, the frequency selection characteristic of the amplitude-frequency characteristic is the worst. However, the Bessel filter has the best linear phase characteristics; the elliptic filter has a ripple in the pass band (flat or equal ripple in the stop band), and the stop band drops the fastest.  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_4.png切比雪夫对比巴特沃夫task1_4D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_4.png切比雪夫对比巴特沃夫task1_4  Fig I-6 Fig I-7  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫2型对比巴特沃夫task1_7.png切比雪夫2型对比巴特沃夫task1_7D:\SUSTECH2020春\信号和系统\lab\proj\贝塞尔对比巴特沃夫task1_1.png贝塞尔对比巴特沃夫task1_1Fig I-8 Fig I-9  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫2型对比巴特沃夫task1_7.png切比雪夫2型对比巴特沃夫task1_7  Fig I-10  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_3.png切比雪夫对比巴特沃夫task1_3D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_6.png切比雪夫对比巴特沃夫task1_6  Fig I-11 Fig I-12  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫2型对比巴特沃夫task1_9.png切比雪夫2型对比巴特沃夫task1_9D:\SUSTECH2020春\信号和系统\lab\proj\贝塞尔对比巴特沃夫task1_3.png.png贝塞尔对比巴特沃夫task1_3.png  Fig I-13 Fig I-14  D:\SUSTECH2020春\信号和系统\lab\proj\椭圆对比巴特沃夫task1_6.png椭圆对比巴特沃夫task1_6  Fig I-15  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_2.png切比雪夫对比巴特沃夫task1_2D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫对比巴特沃夫task1_5.png切比雪夫对比巴特沃夫task1_5  Fig I-16 Fig I-17  D:\SUSTECH2020春\信号和系统\lab\proj\切比雪夫2型对比巴特沃夫task1_8.png切比雪夫2型对比巴特沃夫task1_8D:\SUSTECH2020春\信号和系统\lab\proj\贝塞尔对比巴特沃夫task1_2.png贝塞尔对比巴特沃夫task1_2  Fig I-18 Fig I-19  D:\SUSTECH2020春\信号和系统\lab\proj\椭圆对比巴特沃夫task1_5.png椭圆对比巴特沃夫task1_5  Fig I-20  We plot the frequency response of these 5 filters, using n=4, fcut=50Hz. As shown from Fig I-1 to Fig I-5, the responses of filters are as described above.  Fig I-6 to Fig I-10 are the original and output signals of task-1 when we use Butterworth, Chebyshev I, Chebyshev II, Bessel and elliptic filters as the filters in vocoder().  Similarly, we use these parameters do plot the power spectrum (Fig I-11 to Fig I-15) and frequency spectrum (Fig I-16 to Fig I-20) of them. We take N as 8,16,64 and cutoff frequency fcut=50Hz. Actually, we can’t see much difference from these figures, except figures of Bessel filter. Therefore, we think figure can’t judge the intelligibility of different output signals effectively.  We listen to the outputs and found that Butterworth is the best among them. Secondly, Chebyshev I. Elliptic gave us a satisfying result when N is very large. Chebyshev II results in a bad output signal. Bessel is the worst, and we suppose that is because MATLAB only provides us with “Bessel analog filter design”, instead of classic digital filter, while the other four have digital filter functions. In this case, we can explain why Bessel filter has bad figures of power spectrum and frequency spectrum (only contains high frequency level) and bad sound.  In conclusion, we suggest using Butterworth filter do finish these tasks.   1. Design a simple human-computer interface, and publish the voice processing program into executable files, such as exe, DLL, .NET, etc.   **Analysis**    Fig E-1    Fig E-2  As shown in Fig E-1 and Fig E-2, we construct a simple GUI surface and package it to an EXE file.   1. Further thinking: RMS-SNR evaluation   **Analysis**      Table 1  We want to find out more ways beyond judging the output signals by figures and audios. SNR is a useful way to compare output signal and original signal. The equation is to firstly derive the mean square power of the original signal Ps=sum((s-(mean(s))).^2). Secondly, derive the mean square power of the noise (separated from the output signal). Pn=sum((s-s1).^2). Finally, snr=10log10(Ps/Pn) to get the SNR. We use the outputs of task-1 as an example, and as table 1 shown, most of the results is satisfying. When N=64, we may get the best extent.  **Note**: Please indicate meaning of the symbols in all expressions. Please indicate the coordinate and unit in all figures. | |
| **Experience**  We encountered many difficulties during this project but we managed to solve all these through close communication and sincere teamwork. During the project, we searched many essays about cochlear implant to figure out how it really works, we learnt to use MATLAB to express sin function, design band-pass filter, how to evaluate the effect of different filter and also learn to design gui interface. What’s more, we have a better understanding of how the number of bands affects the intelligibility and how the cut-off frequency of the LPF filter affect the intelligibility of the synthetic sound. Moreover, wo figured out how noise disturb the synthesis and all we also have a glance look at four different filters which are commonly used in the real life. We can say that this project gave us a opportunity to have a glance at what real-time research is like. We even plan to do more further study about this project to see whether we can get some interesting outcome.  Through the project, not only do we learn a lot about using MATLAB and cochlear implants from this unforgettable process, we also harvest precious friendship and get valuable experience. This project seemed to be insignificant at the very beginning but it eventually left a profound influence on all of us: It taught us how to cooperate with each other, how to express ourselves more precisely and most importantly: how real teamwork is like. The project helped us to apply what we have learnt on the lecture class to some real study, which gave us lot of experience and gave us a better understanding of the knowledge that we have learned previously. We truly believe that through this project we indeed learnt a lot and we have fully confidence that we can apply what we learn in future work.  Thanks Dr. Wu for teaching us MATLAB skills and understanding of Fourier series, also thanks TA for marking our work and giving feedback. | |
| **Score** |  |

字体：英文Times new Roman；中文宋体，正文五号

Code

%vocoder

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fcutoff is to design LPF cutoff frequency (need research)

% N is the parts to divide the bandpass filters (need research)

% fs comes from the original sound s

% s\_out is the sound after finally normalized

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function [s\_out,a\_out] = vocoder(s,N,fcutoff,fs)

d200 = log10(200/165.4+1)/0.06;

d7000 = log10(7000/165.4+1)/0.06;

% dmid=(d200+d7000)/1;

% fpassbands = 165.4\*(10.^(0.06\*dmid)-1); % 计算出对应的passbands sets

dsets = linspace(d200,d7000,N+1); % 计算出耳蜗距离vector sets

fpassbands = 165.4\*(10.^(0.06\*dsets)-1); % 计算出对应的passbands sets

s\_out = zeros(1,length(s));

for i=1:N

flow=fpassbands(i);

fhigh=fpassbands(i+1);

fmid=(flow+fhigh)/2;

[b,a]=butter(4,[flow fhigh]/(fs/2)); % 获取 ith bandpass filter

y=filter(b,a,s);

yabs=abs(y); % 整流

[b\_lowpass,a\_lowpass]=butter(4,fcutoff/(fs/2)); %获取低通滤波器

yenv=filter(b\_lowpass,a\_lowpass,yabs); % 提取包络

dt=1/fs:1/fs:length(s)/fs;

sin\_env = sin(2\*pi\*fmid\*dt); %对应频率的 ith sine函数

s\_out=s\_out+yenv.\*sin\_env;

end

a\_out=fft(s\_out,length(s\_out));

a\_out=fftshift(a\_out);

s\_out=s\_out\*(norm(s)/norm(s\_out)); % normalize

end

Task-1

clc;

clear;

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

fcutoff=50;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1]=vocoder(s,1,fcutoff,fs); % 使用声码器进行处理

[s2,a2]=vocoder(s,2,fcutoff,fs);

[s3,a3]=vocoder(s,4,fcutoff,fs);

[s4,a4]=vocoder(s,6,fcutoff,fs);

[s5,a5]=vocoder(s,8,fcutoff,fs);

[s6,a6]=vocoder(s,16,fcutoff,fs);

[s7,a7]=vocoder(s,64,fcutoff,fs);

[s8,a8]=vocoder(s,90,fcutoff,fs); %与切比雪夫相比更好 ，但是N=120几乎没了

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=1'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by N=1'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

%用于对比

figure(1)

subplot(411),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(412),plot(t,s1),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

subplot(413),plot(t,s2),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

subplot(414),plot(t,s2),grid on,title('signal processed by N=64'),xlabel('time/s'),ylabel('sig Amp');

figure(4)

subplot(411),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(412),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

subplot(413),plot(w,abs(a6)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a7)),grid on,title('spectrum of signal processed by N=64'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx6,w]=pwelch(s6,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx7,w]=pwelch(s7,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(411),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(412),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(413),plot(w,10\*log10(Pxx6)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

subplot(414),plot(w,10\*log10(Pxx7)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=64');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(311),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(312),plot(w,10\*log10(Pxx1)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=1');

subplot(313),plot(w,10\*log10(Pxx2)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=2');

figure(6)

subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=4');

subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=6');

subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

audiowrite('task1\_s1.wav',s1,fs);

audiowrite('task1\_s2.wav',s2,fs);

audiowrite('task1\_s3.wav',s3,fs);

audiowrite('task1\_s4.wav',s4,fs);

audiowrite('task1\_s5.wav',s5,fs);

audiowrite('task1\_s6.wav',s6,fs);

audiowrite('task1\_s7.wav',s7,fs);

audiowrite('task1\_s8.wav',s8,fs);

Ps=sum((s-(mean(s))).^2);%signal power

Pn1=sum((s-s1).^2); %noise power

snr1=10\*log10(Ps/Pn1); % 其中I-original signal，In-noise signal，snr

Pn2=sum((s-s2).^2); %noise power

snr2=10\*log10(Ps/Pn2); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn3=sum((s-s3).^2); %noise power

snr3=10\*log10(Ps/Pn3); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn4=sum((s-s4).^2); %noise power

snr4=10\*log10(Ps/Pn4); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn5=sum((s-s5).^2); %noise power

snr5=10\*log10(Ps/Pn5); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn6=sum((s-s6).^2); %noise power

snr6=10\*log10(Ps/Pn6); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn7=sum((s-s7).^2); %noise power

snr7=10\*log10(Ps/Pn7); % 其中I是纯信号，In是带噪信号，snr是信噪比

Pn8=sum((s-s8).^2); %noise power

snr8=10\*log10(Ps/Pn8); % 其中I是纯信号，In是带噪信号，snr是信噪比

sound(s8,fs); %发现巴特沃夫高阶表现好

Task-2

clc;

clear;

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

N=4;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1] = vocoder(s,N,20,fs);

[s2,a2] = vocoder(s,N,50,fs);

[s3,a3] = vocoder(s,N,100,fs);

[s4,a4] = vocoder(s,N,200,fs);

[s5,a5] = vocoder(s,N,400,fs); %?效果很差？

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by f\_{cut}=20Hz'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by f\_{cut}=50Hz'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by f\_{cut}=100Hz'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by f\_{cut}=200Hz'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by f\_{cut}=400Hz'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by f\_{cut}=20Hz'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by f\_{cut}=50Hz'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by f\_{cut}=100Hz'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by f\_{cut}=200Hz'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by f\_{cut}=400Hz'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(311),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(312),plot(w,10\*log10(Pxx1)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=20Hz');

subplot(313),plot(w,10\*log10(Pxx2)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=50Hz');

figure(6)

subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=100Hz');

subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=200Hz');

subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=400Hz');

audiowrite('task2\_s1.wav',s1,fs);

audiowrite('task2\_s2.wav',s2,fs);

audiowrite('task2\_s3.wav',s3,fs);

audiowrite('task2\_s4.wav',s4,fs);

audiowrite('task2\_s5.wav',s5,fs);

sound(s5,fs); % 听得出来

Task-3

clc;

clear;

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

fcutoff = 200;

s=s';

Nlength=length(s);

l=Nlength;

ws=(-pi:2\*pi/l:pi-pi/l)\*fs;

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

ws=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

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

sig=repmat(s,1,10);

[Pxx,w]=pwelch(sig,[],[],512,fs); % 这里选取的采样点参数是512

b=fir2(3000,w/(fs/2),sqrt(Pxx/max(Pxx))); % 选择3000阶数的加窗FIR滤波器，计算b参数

[h,wh]=freqz(b,1,512); % 采样参数采用默认值512

ssn=filter(b,1,noise);

Assn=norm(s)/10^(-5/20); % 求出满足-5的SNR对应的噪音幅值

ssn=(Assn./norm(ssn))\*ssn; % 调整原SSN的幅值

%ssn=norm(s)\*ssn/10^(-5/20)/norm(ssn)

SNR=20\*log10(norm(s)/norm(ssn)); % 验证调整幅值后的SNR

fprintf('>> 验证的SNR为: %fdB\n',SNR);

y=s+ssn;

y2=y\*norm(s)/norm(y); % 最后进行了normalize

ay2=fft(y2,length(y2));

ay2=fftshift(ay2); %求y2的fft

fprintf('>>norm\_ssn: %f\n>>norm\_y2: %f',norm(ssn),norm(y2));

[s1,a1]=vocoder(y2,2,fcutoff,fs); % 使用声码器处理

[s2,a2]=vocoder(y2,4,fcutoff,fs);

[s3,a3]=vocoder(y2,6,fcutoff,fs);

[s4,a4]=vocoder(y2,8,fcutoff,fs);

[s5,a5]=vocoder(y2,16,fcutoff,fs);

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(3)

% subplot(311),plot(ws,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(ws,abs(a1)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(ws,abs(a2)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(ws,abs(a3)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(ws,abs(a4)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(ws,abs(a5)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

audiowrite('task3\_s1.wav',s1,fs);

audiowrite('task3\_s2.wav',s2,fs);

audiowrite('task3\_s3.wav',s3,fs);

audiowrite('task3\_s4.wav',s4,fs);

audiowrite('task3\_s5.wav',s5,fs);

% sound(s,fs);

% pause(3);

% sound(y2,fs);

% pause(3)

sound(s5,fs);

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(311),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(312),plot(w,10\*log10(Pxx1)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=2');

subplot(313),plot(w,10\*log10(Pxx2)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=4');

figure(6)

subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=6');

subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

Task-4

clc;clear;

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

N=6;

s=s';

Nlength=length(s);

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

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

sig=repmat(s,1,10);

[Pxx,ws]=pwelch(sig,[],[],512,fs); % 这里选取的采样点参数是512

b=fir2(3000,ws/(fs/2),sqrt(Pxx/max(Pxx))); % 选择3000阶数的加窗FIR滤波器，计算b参数

[h,wh]=freqz(b,1,512); % 采样参数采用默认值512

ssn=filter(b,1,noise);

Assn=norm(s)/10^(-5/20); % 求出满足-5的SNR对应的噪音幅值

ssn=(Assn./norm(ssn))\*ssn; % 调整原SSN的幅值

%ssn=norm(s)\*ssn/10^(-5/20)/norm(ssn)

SNR=20\*log10(norm(s)/norm(ssn)); % 验证调整幅值后的SNR

fprintf('>> 验证的SNR为: %fdB\n',SNR);

y=s+ssn;

y2=y\*norm(s)/norm(y); % 最后进行了normalize

ay2=fft(y2,length(y2));

ay2=fftshift(ay2); %求y2的fft

[s1,a1] = vocoder(y2,N,20,fs);

[s2,a2] = vocoder(y2,N,50,fs);

[s3,a3] = vocoder(y2,N,100,fs);

[s4,a4] = vocoder(y2,N,200,fs);

[s5,a5] = vocoder(y2,N,400,fs);

figure(1)

subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(312),plot(t,s1),grid on,title('signal processed by f\_{cut}=20Hz'),xlabel('time/s'),ylabel('sig Amp');

subplot(313),plot(t,s2),grid on,title('signal processed by f\_{cut}=50Hz'),xlabel('time/s'),ylabel('sig Amp');

figure(2)

subplot(311),plot(t,s3),grid on,title('signal processed by f\_{cut}=100Hz'),xlabel('time/s'),ylabel('sig Amp');

subplot(312),plot(t,s4),grid on,title('signal processed by f\_{cut}=200Hz'),xlabel('time/s'),ylabel('sig Amp');

subplot(313),plot(t,s5),grid on,title('signal processed by f\_{cut}=400Hz'),xlabel('time/s'),ylabel('sig Amp');

figure(3)

subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by f\_{cut}=20Hz'),xlabel('freq.'),ylabel('fft');

subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by f\_{cut}=50Hz'),xlabel('freq.'),ylabel('fft');

figure(4)

subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by f\_{cut}=100Hz'),xlabel('freq.'),ylabel('fft');

subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by f\_{cut}=200Hz'),xlabel('freq.'),ylabel('fft');

subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by f\_{cut}=400Hz'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(311),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(312),plot(w,10\*log10(Pxx1)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=20Hz');

subplot(313),plot(w,10\*log10(Pxx2)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=50Hz');

figure(6)

subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=100Hz');

subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=200Hz');

subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('fcut=400Hz');

audiowrite('task4\_s1.wav',s1,fs);

audiowrite('task4\_s2.wav',s2,fs);

audiowrite('task4\_s3.wav',s3,fs);

audiowrite('task4\_s4.wav',s4,fs);

audiowrite('task4\_s5.wav',s5,fs);

sound(s1,fs);

bonus

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% fcutoff is to design LPF cutoff frequency (need research)

% N is the parts to divide the bandpass filters (need research)

% fs comes from the original sound s

% s\_out is the sound after finally normalized

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%test for bessel filter

clc;

clear;

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

s=s';

N=8;

fcutoff=50;

d200 = log10(200/165.4+1)/0.06;

d7000 = log10(7000/165.4+1)/0.06;

dsets = linspace(d200,d7000,N+1);

fpassbands = 165.4\*(10.^(0.06\*dsets)-1);

s\_out = zeros(1,length(s));

for i=1:N

flow=fpassbands(i);

fhigh=fpassbands(i+1);

fmid=(flow+fhigh)/2;

%

% [b,a]=butter(4,[flow fhigh]/(fs/2)); % 获取 ith bandpass filter

[b,a] = besself(4,[flow fhigh]/(fs/2),'bandpass');

% [b, a] = besself(4, [flow fhigh] / (fs / 2));

[h,f]=freqz(b,a,128,fs);

hold on;

plot(f,20\*log10(abs(h)));

y=filter(b,a,s);

yabs=abs(y); % 整流

[b\_lowpass,a\_lowpass]=besself(4,fcutoff/(fs/2)); %获取低通滤波器

yenv=filter(b\_lowpass,a\_lowpass,yabs); % 提取包络

dt=1/fs:1/fs:length(s)/fs;

sin\_env = sin(2\*pi\*fmid\*dt); %对应频率的 ith sine函数

s\_out=s\_out+yenv.\*sin\_env;

end

xlabel('freqz'),ylabel('frequency response/dB');

title('bessel');

hold off;

a\_out=fft(s\_out,length(s\_out));

a\_out=fftshift(a\_out);

s\_out=s\_out\*(norm(s)/norm(s\_out)); % normalize

sound(s\_out,fs);

%test for ellip

clc;

clear;

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

s=s';

N=100;

fcutoff=50;

d200 = log10(200/165.4+1)/0.06;

d7000 = log10(7000/165.4+1)/0.06;

dsets = linspace(d200,d7000,N+1);

fpassbands = 165.4\*(10.^(0.06\*dsets)-1);

s\_out = zeros(1,length(s));

for i=1:N

flow=fpassbands(i);

fhigh=fpassbands(i+1);

fmid=(flow+fhigh)/2;

% Rp=3;

% Rs=40;

% [n,Wn]=cheb1ord(flow/(fs/2),fhigh/(fs/2),Rp,Rs);

% fprintf('>> i=%d %f\n',i,n);

% [b,a]=cheby1(n,Rp,Wn);

% n = 4; Rp = 0.5;

% Wn = [flow fhigh]/(fs/2);

% [b,a] = cheby1(n,Rp,Wn);

Rp =3; Rs =30 ;

Wp =[flow fhigh] /(fs/2);

Ws = [flow-20 fhigh+20] /(fs/2);

[ n , Wn] = ellipord (Wp , Ws , Rp , Rs);

[ b , a] = ellip(n , Rp, Rs ,Wn);

% [b,a]=butter(4,[flow fhigh]/(fs/2));

y=filter(b,a,s);

yabs=abs(y);

[h,f]=freqz(b,a,128,fs);

hold on;

plot(f,20\*log10(abs(h)));

fprintf('>> checkpint one for i = %d\n',i);

[b\_lowpass,a\_lowpass]=butter(4,fcutoff/(fs/2));

yenv=filter(b\_lowpass,a\_lowpass,yabs);

fprintf('>> checkpint two for i = %d\n',i);

dt=1/fs:1/fs:length(s)/fs;

sin\_env = sin(2\*pi\*fmid\*dt); % sine to get envelop

fprintf('>> checkpint three for i = %d\n',i);

s\_out=s\_out+yenv.\*sin\_env;

end

xlabel('freqz'),ylabel('frequency response/dB');

title('ellip');

hold off;

fprintf('>> checkpint end');

s\_out=s\_out\*(norm(s)/norm(s\_out)); % normalize

fprintf('>> checkpint final end');

sound(s\_out,fs);

%cheby

function [s\_out,a\_out] = cheby(s,N,fcutoff,fs)

d200 = log10(200/165.4+1)/0.06;

d7000 = log10(7000/165.4+1)/0.06;

% dmid=(d200+d7000)/1;

% fpassbands = 165.4\*(10.^(0.06\*dmid)-1); % 计算出对应的passbands sets

dsets = linspace(d200,d7000,N+1); % 计算出耳蜗距离vector sets

fpassbands = 165.4\*(10.^(0.06\*dsets)-1); % 计算出对应的passbands sets

s\_out = zeros(1,length(s));

for i=1:N

flow=fpassbands(i);

fhigh=fpassbands(i+1);

fmid=(flow+fhigh)/2;

n = 4; Rp = 0.5;

Wn = [flow fhigh]/(fs/2);

[b,a] = cheby1(n,Rp,Wn);

y=filter(b,a,s);

yabs=abs(y);

[h,f]=freqz(b,a,128,fs);

% hold on;

% plot(f,20\*log10(abs(h)));

[b\_lowpass,a\_lowpass]=butter(4,fcutoff/(fs/2));

yenv=filter(b\_lowpass,a\_lowpass,yabs);

dt=1/fs:1/fs:length(s)/fs;

sin\_env = sin(2\*pi\*fmid\*dt);

s\_out=s\_out+yenv.\*sin\_env;

end

a\_out=fft(s\_out,length(s\_out));

a\_out=fftshift(a\_out);

s\_out=s\_out\*(norm(s)/norm(s\_out)); % normalize

fprintf('>> checkpint final end');

%cheby2

function [s\_out,a\_out] = cheby02(s,N,fcutoff,fs)

d200 = log10(200/165.4+1)/0.06;

d7000 = log10(7000/165.4+1)/0.06;

% dmid=(d200+d7000)/1;

% fpassbands = 165.4\*(10.^(0.06\*dmid)-1); % 计算出对应的passbands sets

dsets = linspace(d200,d7000,N+1); % 计算出耳蜗距离vector sets

fpassbands = 165.4\*(10.^(0.06\*dsets)-1); % 计算出对应的passbands sets

s\_out = zeros(1,length(s));

for i=1:N

flow=fpassbands(i);

fhigh=fpassbands(i+1);

fmid=(flow+fhigh)/2;

% Rp=3;

% Rs=40;

%

% [n,Wn]=cheb1ord(flow/(fs/2),fhigh/(fs/2),Rp,Rs);

% fprintf('>> i=%d %f\n',i,n);

% [b,a]=cheby1(n,Rp,Wn);

n = 4; Rp = 0.5;

Wn = [flow fhigh]/(fs/2);

[b,a] = cheby2(n,Rp,Wn);

% [b,a]=butter(4,[flow fhigh]/(fs/2));

y=filter(b,a,s);

yabs=abs(y); % 鏁存祦

[h,f]=freqz(b,a,128,fs);

% hold on;

% plot(f,20\*log10(abs(h)));

[b\_lowpass,a\_lowpass]=butter(4,fcutoff/(fs/2));

yenv=filter(b\_lowpass,a\_lowpass,yabs);

dt=1/fs:1/fs:length(s)/fs;

sin\_env = sin(2\*pi\*fmid\*dt);

s\_out=s\_out+yenv.\*sin\_env;

end

a\_out=fft(s\_out,length(s\_out));

a\_out=fftshift(a\_out);

s\_out=s\_out\*(norm(s)/norm(s\_out)); % normalize

fprintf('>> checkpint final end');

%Tests for these filters:

%test for cheby1

clc;

clear;

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

fcutoff=50;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1]=cheby(s,1,fcutoff,fs); % 使用声码器进行处理

[s2,a2]=cheby(s,2,fcutoff,fs);

[s3,a3]=cheby(s,4,fcutoff,fs);

[s4,a4]=cheby(s,6,fcutoff,fs);

[s5,a5]=cheby(s,8,fcutoff,fs);

[s6,a6]=cheby(s,16,fcutoff,fs);

[s7,a7]=cheby(s,64,fcutoff,fs);

[s8,a8]=cheby(s,90,fcutoff,fs); %刚好90没掉

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=1'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by N=1'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

figure(1)

subplot(411),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(412),plot(t,s1),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

subplot(413),plot(t,s2),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

subplot(414),plot(t,s2),grid on,title('signal processed by N=64'),xlabel('time/s'),ylabel('sig Amp');

figure(4)

subplot(411),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(412),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

subplot(413),plot(w,abs(a6)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a7)),grid on,title('spectrum of signal processed by N=64'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx6,w]=pwelch(s6,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx7,w]=pwelch(s7,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(411),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(412),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(413),plot(w,10\*log10(Pxx6)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

subplot(414),plot(w,10\*log10(Pxx7)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=64');

% figure(6)

% subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

audiowrite('task1\_s1\_c.wav',s1,fs);

audiowrite('task1\_s2\_c.wav',s2,fs);

audiowrite('task1\_s3\_c.wav',s3,fs);

audiowrite('task1\_s4\_c.wav',s4,fs);

audiowrite('task1\_s5\_c.wav',s5,fs); %放这个

audiowrite('task1\_s6\_c.wav',s6,fs); %放这个

audiowrite('task1\_s7\_c.wav',s7,fs); %放这个

audiowrite('task1\_s8\_c.wav',s8,fs);

% Ps=sum((s-(mean(s))).^2);%signal power

% Pn=sum((s-s5).^2); %noise power

%

% snr1=10\*log10(Ps/Pn); % 其中I是纯信号，In是带噪信号，snr是信噪比

% sound(s5,fs);

% pause(3);

% sound(s6,fs);

% pause(3);

% sound(s7,fs);

sound(s8,fs); %高阶不行

%test for cheby2

clc;

clear;

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

fcutoff=50;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1]=cheby02(s,1,fcutoff,fs); % 使用声码器进行处理

[s2,a2]=cheby02(s,2,fcutoff,fs);

[s3,a3]=cheby02(s,4,fcutoff,fs);

[s4,a4]=cheby02(s,6,fcutoff,fs);

[s5,a5]=cheby02(s,8,fcutoff,fs);

[s6,a6]=cheby02(s,16,fcutoff,fs);

[s7,a7]=cheby02(s,64,fcutoff,fs);

[s8,a8]=cheby02(s,90,fcutoff,fs); %刚好90没掉

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=1'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by N=1'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

figure(1)

subplot(411),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(412),plot(t,s1),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

subplot(413),plot(t,s2),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

subplot(414),plot(t,s2),grid on,title('signal processed by N=64'),xlabel('time/s'),ylabel('sig Amp');

figure(4)

subplot(411),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(412),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

subplot(413),plot(w,abs(a6)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a7)),grid on,title('spectrum of signal processed by N=64'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx6,w]=pwelch(s6,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx7,w]=pwelch(s7,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(411),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(412),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(413),plot(w,10\*log10(Pxx6)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

subplot(414),plot(w,10\*log10(Pxx7)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=64');

% figure(6)

% subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% audiowrite('task1\_s1\_c2.wav',s1,fs);

% audiowrite('task1\_s2\_c2.wav',s2,fs);

% audiowrite('task1\_s3\_c2.wav',s3,fs);

% audiowrite('task1\_s4\_c2.wav',s4,fs);

audiowrite('task1\_s5\_c2.wav',s5,fs); %放这个

audiowrite('task1\_s6\_c2.wav',s6,fs); %放这个

audiowrite('task1\_s7\_c2.wav',s7,fs); %放这个

audiowrite('task1\_s8\_c2.wav',s8,fs);

% Ps=sum((s-(mean(s))).^2);%signal power

% Pn=sum((s-s5).^2); %noise power

%

% snr1=10\*log10(Ps/Pn); % 其中I是纯信号，In是带噪信号，snr是信噪比

% sound(s5,fs);

% pause(3);

% sound(s6,fs);

% pause(3);

% sound(s7,fs);

sound(s8,fs); %高阶不行

%test for bessel

clc;

clear;

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

fcutoff=50;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1]=bes(s,1,fcutoff,fs); % 使用声码器进行处理

[s2,a2]=bes(s,2,fcutoff,fs);

[s3,a3]=bes(s,4,fcutoff,fs);

[s4,a4]=bes(s,6,fcutoff,fs);

[s5,a5]=bes(s,8,fcutoff,fs);

[s6,a6]=bes(s,16,fcutoff,fs);

[s7,a7]=bes(s,64,fcutoff,fs);

[s8,a8]=bes(s,90,fcutoff,fs); %刚好90没掉

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=1'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by N=1'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

figure(1)

subplot(411),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(412),plot(t,s1),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

subplot(413),plot(t,s2),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

subplot(414),plot(t,s2),grid on,title('signal processed by N=64'),xlabel('time/s'),ylabel('sig Amp');

figure(4)

subplot(411),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(412),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

subplot(413),plot(w,abs(a6)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a7)),grid on,title('spectrum of signal processed by N=64'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx6,w]=pwelch(s6,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx7,w]=pwelch(s7,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(411),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(412),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(413),plot(w,10\*log10(Pxx6)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

subplot(414),plot(w,10\*log10(Pxx7)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=64');

% figure(6)

% subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

audiowrite('task1\_s1\_b.wav',s1,fs);

audiowrite('task1\_s2\_b.wav',s2,fs);

audiowrite('task1\_s3\_b.wav',s3,fs);

audiowrite('task1\_s4\_b.wav',s4,fs);

audiowrite('task1\_s5\_b.wav',s5,fs); %放这个

audiowrite('task1\_s6\_b.wav',s6,fs); %放这个

audiowrite('task1\_s7\_b.wav',s7,fs); %放这个

audiowrite('task1\_s8\_b.wav',s8,fs);

% Ps=sum((s-(mean(s))).^2);%signal power

% Pn=sum((s-s5).^2); %noise power

%

% snr1=10\*log10(Ps/Pn); % 其中I是纯信号，In是带噪信号，snr是信噪比

% sound(s5,fs);

% pause(3);

% sound(s6,fs);

% pause(3);

% sound(s7,fs);

sound(s8,fs); %高阶不行

%test for elliptic

clc;

clear;

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

fcutoff=50;

s=s';

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs/(2\*pi);

a0=fft(s,length(s));

a0=fftshift(a0);

t=1/fs:1/fs:length(s)/fs;

[s1,a1]=ellipp(s,1,fcutoff,fs); % 使用声码器进行处理

[s2,a2]=ellipp(s,2,fcutoff,fs);

[s3,a3]=ellipp(s,4,fcutoff,fs);

[s4,a4]=ellipp(s,6,fcutoff,fs);

[s5,a5]=ellipp(s,8,fcutoff,fs);

[s6,a6]=ellipp(s,16,fcutoff,fs);

[s7,a7]=ellipp(s,64,fcutoff,fs);

[s8,a8]=ellipp(s,200,fcutoff,fs); %对比，N=120仍有而巴特沃夫没了 N=300还有？？？

% figure(1)

% subplot(311),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s1),grid on,title('signal processed by N=1'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s2),grid on,title('signal processed by N=2'),xlabel('time/s'),ylabel('sig Amp');

%

% figure(2)

% subplot(311),plot(t,s3),grid on,title('signal processed by N=4'),xlabel('time/s'),ylabel('sig Amp');

% subplot(312),plot(t,s4),grid on,title('signal processed by N=6'),xlabel('time/s'),ylabel('sig Amp');

% subplot(313),plot(t,s5),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

% figure(3)

% subplot(311),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a1)),grid on,title('spectrum of signal processed by N=1'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a2)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

%

% figure(4)

% subplot(311),plot(w,abs(a3)),grid on,title('spectrum of signal processed by N=4'),xlabel('freq.'),ylabel('fft');

% subplot(312),plot(w,abs(a4)),grid on,title('spectrum of signal processed by N=6'),xlabel('freq.'),ylabel('fft');

% subplot(313),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

figure(1)

subplot(411),plot(t,s),grid on,title('original signal'),xlabel('time/s'),ylabel('sig Amp');

subplot(412),plot(t,s1),grid on,title('signal processed by N=8'),xlabel('time/s'),ylabel('sig Amp');

subplot(413),plot(t,s2),grid on,title('signal processed by N=16'),xlabel('time/s'),ylabel('sig Amp');

subplot(414),plot(t,s2),grid on,title('signal processed by N=64'),xlabel('time/s'),ylabel('sig Amp');

figure(4)

subplot(411),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(412),plot(w,abs(a5)),grid on,title('spectrum of signal processed by N=8'),xlabel('freq.'),ylabel('fft');

subplot(413),plot(w,abs(a6)),grid on,title('spectrum of signal processed by N=16'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a7)),grid on,title('spectrum of signal processed by N=64'),xlabel('freq.'),ylabel('fft');

[Pxx0,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx1,w]=pwelch(s1,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx2,w]=pwelch(s2,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx3,w]=pwelch(s3,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx4,w]=pwelch(s4,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx5,w]=pwelch(s5,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx6,w]=pwelch(s6,[],[],512,fs); % 这里选取的采样点参数是512

[Pxx7,w]=pwelch(s7,[],[],512,fs); % 这里选取的采样点参数是512

figure(5)

subplot(411),plot(w,10\*log10(Pxx0)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('original');

subplot(412),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=8');

subplot(413),plot(w,10\*log10(Pxx6)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=16');

subplot(414),plot(w,10\*log10(Pxx7)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');title('N=64');

% figure(6)

% subplot(311),plot(w,10\*log10(Pxx3)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(312),plot(w,10\*log10(Pxx4)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

% subplot(313),plot(w,10\*log10(Pxx5)),grid on,xlabel('frequency/Hz'),ylabel('pwelch/dB');

audiowrite('task1\_s1\_e.wav',s1,fs);

audiowrite('task1\_s2\_e.wav',s2,fs);

audiowrite('task1\_s3\_e.wav',s3,fs);

audiowrite('task1\_s4\_e.wav',s4,fs);

audiowrite('task1\_s5\_e.wav',s5,fs); %放这个

audiowrite('task1\_s6\_e.wav',s6,fs); %放这个

audiowrite('task1\_s7\_e.wav',s7,fs); %放这个

audiowrite('task1\_s8\_e.wav',s8,fs);

% Ps=sum((s-(mean(s))).^2);%signal power

% Pn=sum((s-s5).^2); %noise power

%

% snr1=10\*log10(Ps/Pn); % 其中I是纯信号，In是带噪信号，snr是信噪比

% sound(s5,fs);

% pause(3);

% sound(s6,fs);

% pause(3);

% sound(s7,fs);

sound(s7,fs);

%GUI

function varargout = whyme(varargin)

gui\_Singleton = 1;

gui\_State = struct('gui\_Name', mfilename, ...

'gui\_Singleton', gui\_Singleton, ...

'gui\_OpeningFcn', @whyme\_OpeningFcn, ...

'gui\_OutputFcn', @whyme\_OutputFcn, ...

'gui\_LayoutFcn', [] , ...

'gui\_Callback', []);

if nargin && ischar(varargin{1})

gui\_State.gui\_Callback = str2func(varargin{1});

end

if nargout

[varargout{1:nargout}] = gui\_mainfcn(gui\_State, varargin{:});

else

gui\_mainfcn(gui\_State, varargin{:});

end

% --- Executes just before whyme is made visible.

function whyme\_OpeningFcn(hObject, eventdata, handles, varargin)

handles.output = hObject;

% Update handles structure

guidata(hObject, handles);

% --- Outputs from this function are returned to the command line.

function varargout = whyme\_OutputFcn(hObject, eventdata, handles)

varargout{1} = handles.output;

function pushbutton1\_Callback(hObject, eventdata, handles)

[filename,filepath]=uigetfile('\*.wav','打开文件');

filep=strcat(filepath,filename);

[s,fs]=audioread(filep);

s=s';

prompt={'输入N的值','输入截止频率'};%设置提示字符串

name='Enter trapeaia Data';%设置标题

numlines=1;%指定输入数据的行数

defAns={'2','60'};%设定默认值

Resize='on';%设定对话框尺寸可调节

answer=inputdlg(prompt,name,numlines,defAns,'on')

num1=answer(1);

num2=answer(2);

N1=str2num(num1{1});

fcutoff1=str2num(num2{1});

[s1,a1]=vocoder(s,N1,fcutoff1,fs);

l=length(s);

w=(-pi:2\*pi/l:pi-pi/l)\*fs;

a0=fft(s,length(s));

a0=fftshift(a0);

subplot(411),plot(s),grid on,title('original signal'),xlabel('t'),ylabel('y');

subplot(412),plot(s1),grid on,title('signal processed by input N'),xlabel('t'),ylabel('y');

subplot(413),plot(w,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(414),plot(w,abs(a1)),grid on,title('spectrum of signal processed by input N'),xlabel('freq.'),ylabel('fft');

audiowrite(filep,s1,fs);

sound(s);

% --- Executes on button press in pushbutton2.

function pushbutton2\_Callback(hObject, eventdata, handles)

[filename,filepath]=uigetfile('\*.wav','打开文件');

filep=strcat(filepath,filename);

[s,fs]=audioread(filep);

prompt={'输入N的值','输入截止频率'};%设置提示字符串

name='Enter trapeaia Data';%设置标题

numlines=1;%指定输入数据的行数

defAns={'2','60'};%设定默认值

Resize='on';%设定对话框尺寸可调节

answer=inputdlg(prompt,name,numlines,defAns,'on')

num1=answer(1);

num2=answer(2);

N1=str2num(num1{1});

fcutoff1=str2num(num2{1});

fcutoff=50;

s=s';

Nlength=length(s);

l=Nlength;

ws=(-pi:2\*pi/l:pi-pi/l)\*fs;

a0=fft(s,length(s));

a0=fftshift(a0);

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

sig=repmat(s,1,10);

[Pxx,w]=pwelch(s,[],[],512,fs); % 这里选取的采样点参数是512

b=fir2(3000,w/(fs/2),sqrt(Pxx/max(Pxx))); % 选择3000阶数的加窗FIR滤波器，计算b参数

[h,wh]=freqz(b,1,512); % 采样参数采用默认值512

% ssn=filter(b,1,noise);

Anoise=norm(s)/10^(-5/20); % 求出满足-5的SNR对应的噪音幅值

noise=(Anoise./norm(noise))\*noise; % 调整原SSN的幅值

ssn=filter(b,1,noise);

SNR=20\*log10(norm(s)/norm(noise)); % 验证调整幅值后的SNR

fprintf('>> 验证的SNR为: %fdB\n',SNR);

y=s+ssn;

y2=y\*norm(s)/norm(y); % 最后进行了normalize

ay2=fft(y2,length(y2));

ay2=fftshift(ay2); %求y2的fft

fprintf('>>norm\_ssn: %f\n>>norm\_y2: %f',norm(ssn),norm(y2));

[s1,a1]=vocoder(y2,2,fcutoff1,fs); % 使用声码器处理

[s5,a5]=vocoder(y2,N1,fcutoff1,fs);

figure(1)

subplot(311),plot(s),grid on,title('original signal'),xlabel('t'),ylabel('y');

subplot(312),plot(y2),grid on,title('new signal with noise'),xlabel('t'),ylabel('y');

subplot(313),plot(s1),grid on,title('signal processed by N=2'),xlabel('t'),ylabel('y');

figure(2)

subplot(311),plot(ws,abs(a0)),grid on,title('spectrum of original signal'),xlabel('freq.'),ylabel('fft');

subplot(312),plot(ws,abs(ay2)),grid on,title('spectrum of noise signal'),xlabel('freq.'),ylabel('fft');

subplot(313),plot(ws,abs(a1)),grid on,title('spectrum of signal processed by N=2'),xlabel('freq.'),ylabel('fft');

audiowrite('task3\_s1.wav',s1,fs);

audiowrite('task3\_s5.wav',s5,fs);

sound(s5);

% --- Executes on button press in pushbutton3.

function pushbutton3\_Callback(hObject, eventdata, handles)

close;

% hObject handle to pushbutton3 (see GCBO)

function axes8\_CreateFcn(hObject, eventdata, handles)

% hObject handle to axes8 (see GCBO)

pic=imread('信号图像.jpg');

imshow(pic);

% eventdata reserved - to be defined in a future version of MATLAB

% handles empty - handles not created until after all CreateFcns called

% Hint: place code in OpeningFcn to populate axes8