实 验 报 告

课程名称：	《数字信号处理实验》
姓名学号：	黎子建 **201630257912**
	蒲 尧 **201630258438**
	敖梓茗 **201630257165**
	邢 靖 **201630258803**
学生专业：	信息工程 **3** 班
开课学期：	**2018-2019** 年度第一学期

电子与信息学院

2018 年 9 月 制

学生实验守则

（2018 年 5 月修订）

一、学生进入实验室必须严格遵守实验室的各项规章制度，服从教师的安排。

二、进入实验室，要注意环境安全和实验过程的安全，慎防意外发生。若发生事故，应及时向实验指导人员报告，听从教师指挥，并采取相应的措施，减少事故造成的伤害和损失。

三、实验前必须认真预习，明确实验目的、内容和步骤，了解仪器设备的操作规程和实验物品的特性。

四、实验课不得迟到、早退，不得擅自缺课；因病或其它原因不能上课者应履行请假手续；中途因故离场，须经指导老师批准；衣冠不整不得进入实验室，不准带与实验课无关的物品进实验室。

五、实验过程中，应规范操作，认真观察并如实记录，实验结果须经实验指导人员认可。

六、实验报告应独立、按时完成，不得抄袭、臆造实验报告。

七、爱护仪器设备，节约用水、用电和实验材料。未经许可，不许动用与本实验无关的仪器设备及其它物品，不准私自将公物拿出实验室。

八、实验完毕，应做好仪器设备的复位工作以及关闭相关的水源、电源和气源，清洁实验台面和仪器设备，打扫室内卫生并得到实验指导人员允许后方可离开实验室。

九、对违反实验室规章制度和实验操作规程造成事故和损失的，视其情节对责任者按章处理。

设计性实验:语音及音乐信号的采样、滤波及处理

地　　点：	31　楼　405　房；	**实验台号：**	
实验日期与时间：	2018.11.26	**评　　分：**	
预习检查纪录：		**实验教师：**	

一、实验目的

1、　理解采样率和量化级数对语音信号的影响；
2、　设计滤波器解决实际问题。
3、　了解回声的产生和梳妆滤波器；
4、　混音效果的原理和均衡器的设计；

二、实验原理

实验提示

（1）推荐录音及播放软件：CoolEdit；

（2）分析语音及音乐信号的频谱，根据信号的频率特性理解采样定律对信号数字化的工程指导意义；

（3）可用带阻滤波器对 50Hz 交流电噪声进行去噪处理；

（4）也可研究设计自适应滤波器对 50Hz 噪声及其它随机环境噪声进行滤波处理。

（5）回声产生可以使用梳妆滤波器，y(n)=x(n)+ax(n-R), a<1(回声衰减系数)；或者传输函数为 $H(z)=\frac{\alpha+z^{-R}}{1+\alpha z^{-R}},|\alpha|<1$ 的全通滤波器实现；比较这两种实现方式的区别，分析为什么会有这样的区别；

（6）可以用许多一阶和二阶参数可调的滤波器级联来实现均衡器的功能，滤波器的结构选择结构要求是调整方便，最好调一个参数只影响

一个应用指标，且可调参数少；

三、实验要求

I、利用电脑的声卡录一段语音信号及音乐信号，

（1）观察使用不同采样率及量化级数所得到的信号的听觉效果，从而确定对不同信号的最佳的采样率；

（2）分析音乐信号的采样率为什么要比语音的采样率高才能得到较好的听觉效果；

（3）注意观察信号中的噪声（特别是 50hz 交流电信号对录音的干扰，设计一个滤波器去除该噪声。

II、对一段语音信号及音乐信号

（1）设计函数实现一段语音或音乐的回声产生；

（2）设计均衡器，使得得不同频率的混合音频信号，通过一个均衡器后，增强或削减某些频率区域，以便修正低频和高频信号之间的关系；

有图形交互界面更佳。

四、实验源码及结果分析

I、 声音及音乐信号的量化采样及滤波

（1） 源码

① 采样 量化 50hz 滤波器

```
function varargout = Visualization(varargin)
% VISUALIZATION MATLAB code for Visualization.fig
```

```
%     VISUALIZATION, by itself, creates a new VISUALIZATION or
raises the existing
%     singleton*.
%
%     H = VISUALIZATION returns the handle to a new VISUALIZATION
or the handle to
%     the existing singleton*.
%
%     VISUALIZATION('CALLBACK',hObject,eventData,handles,...)
calls the local
%     function named CALLBACK in VISUALIZATION.M with the given
input arguments.
%
%     VISUALIZATION('Property','Value',...) creates a new
VISUALIZATION or raises the
%     existing singleton*.  Starting from the left, property value
pairs are
%     applied to the GUI before Visualization_OpeningFcn gets
called.  An
%     unrecognized property name or invalid value makes property
application
%     stop.  All inputs are passed to Visualization_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Visualization

% Last Modified by GUIDE v2.5 26-Nov-2018 23:08:32

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Visualization_OpeningFcn, ...
                   'gui_OutputFcn',  @Visualization_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
% End initialization code - DO NOT EDIT


% --- Executes just before Visualization is made visible.
function Visualization_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Visualization (see VARARGIN)

% Choose default command line output for Visualization
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Visualization wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Visualization_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%以下为程序主体|以下为程序主体|以下为程序主体|以下为程序主体|以下为程序主体
% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
rs = audioread('11.26.aaaa.wav');   %读取文件
FS = 8000;      %采样率8K
n = length(rs);    %取文件的长度
A = get(handles.slider1,'Value');   %取滑动条的读数
rs = Mix50Hz(rs,n,A); %因为50Hz分量太小，所以人工混入50Hz的噪声
RS_e = fft(rs);
mag_RS_e = abs(RS_e);
f = (0:n-1)*(FS/n);        %频率范围
axes(handles.axes1);       %定位到图框1绘制图形（左上）
stem(handles.axes1,f,mag_RS_e,'.');grid on;
axis([0 4000 0 inf]);
title("原频谱/Hz");
ylabel('Magnitude');
%-------------------------------------------------------
axes(handles.axes2);       %定位到图框2绘制图形（右上）
stem(handles.axes2,f,mag_RS_e,'.');grid on;
axis([20 80 0 8]);
title("原频谱20-80Hz部分/Hz");
ylabel('Magnitude');
%-------------------------------------------------------
[rs] = RM50Hz(rs);     %调用50Hz陷波器
n = length(rs);
RS_e = fft(rs);
mag_RS_e = abs(RS_e);
f = (0:n-1)*(FS/n);    %频率范围
axes(handles.axes3);   %定位到图框3绘制图形（左下）
stem(handles.axes3,f,mag_RS_e,'.');grid on;
axis([20 80 0 8]);
title("滤波后20-80Hz部分/Hz");
ylabel('Magnitude');
%-------------------------------------------------------
axes(handles.axes4);   %定位到图框4绘制图形（右下）
stem(handles.axes4,f,mag_RS_e,'.');grid on;
axis([0 4000 0 inf]);
title("滤波后频谱/Hz");
ylabel('Magnitude');
%以上为程序主体|以上为程序主体|以上为程序主体|以上为程序主体|以上为程序主体

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double


% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine
range of slider


% --- Executes during object creation, after setting all
properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
```

```
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end
```

② 用于加噪的 50hz 信号

```
function [ y ] = Mix50Hz( x,N,A )
Fs = 8000;
Ts = 1/Fs;
n = 1:N;
fiftyHz = A*sin(2*pi*50*n*Ts)+1;
y = fiftyHz.*x';
end
```

II、回声及均衡器源码

（1）源码：

①录音程序：

```
function [y]=record0(Fs,T)

R = audiorecorder(Fs, 16 ,1) ;

%创建一个保存音频信息的对象，它包含采样率，时间和录制的音频信息等

等。

%44100表示采样为44100Hz（可改为8000, 11025, 22050等，此数值越大，录入的

声音质量越好，相应需要的存储空间越大）

%16为用16bits存储，2为两通道即立体声（也可以改为1即单声道）。

record(R);

%开始录制，此时对着麦克风说话即可。

pause(T);

%暂停录制。

stop(R);

%停止录制

myspeech = getaudiodata(R);
```

```
%得到以n*2列数字矩阵存储的刚录制的音频信号。
sound(myspeech,Fs);
%对这个矩阵你就可以用各种滤波器进行处理，或者把它和别的音频混音等等。你也可以画出它的波形（如果时间较长画波形可能会花点时间）。
audiowrite('test.wav',myspeech,Fs) %存储.wav音频文件，在这里文件名为test.wav
y=audioread('test.wav');
```

②回声和均衡器

```
function varargout = dspshiyan(varargin)
% DSPSHIYAN MATLAB code for dspshiyan.fig
%      DSPSHIYAN, by itself, creates a new DSPSHIYAN or raises the existing
%      singleton*.
%
%      H = DSPSHIYAN returns the handle to a new DSPSHIYAN or the handle to
%      the existing singleton*.
%
%      DSPSHIYAN('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in DSPSHIYAN.M with the given input arguments.
%
%      DSPSHIYAN('Property','Value',...) creates a new DSPSHIYAN or raises the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before dspshiyan_OpeningFcn gets called.  An
```

```
%           unrecognized property name or invalid value makes property application
%           stop.    All inputs are passed to dspshiyan_OpeningFcn via varargin.
%
%           *See GUI Options on GUIDE's Tools menu.    Choose "GUI allows only one
%           instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES


% Edit the above text to modify the response to help dspshiyan


% Last Modified by GUIDE v2.5 26-Nov-2018 13:47:18


% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',              mfilename, ...
                                'gui_Singleton',    gui_Singleton, ...
                                'gui_OpeningFcn', @dspshiyan_OpeningFcn, ...
                                'gui_OutputFcn',    @dspshiyan_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
% End initialization code - DO NOT EDIT



% --- Executes just before dspshiyan is made visible.
function dspshiyan_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to dspshiyan (see VARARGIN)

% Choose default command line output for dspshiyan
handles.output = hObject;

% Update handles structure
```

```
guidata(hObject, handles);


% UIWAIT makes dspshiyan wait for user response (see UIRESUME)
% uiwait(handles.figure1);



% --- Outputs from this function are returned to the command line.
function varargout = dspshiyan_OutputFcn(hObject, eventdata, handles)
% varargout   cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on button press in pushbutton4.
%回声框里的“录音并播放”按钮
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    structure with handles and user data (see GUIDATA)
% 根据语音还是音乐选择采样率，语音为“1”，音乐为“2”
if get (handles.popupmenu5,'Value' )==1;
    Fs=8000;
elseif get (handles.popupmenu5, 'Value' )==2;
    Fs=44100;
else
    errordlg('error! ');
end
% 根据“时长”框里的数选择时长（s）
if get (handles.popupmenu6,'Value' )==1;
    T=1;
elseif get (handles.popupmenu6, 'Value' )==2;
    T=2;
elseif get (handles.popupmenu6, 'Value' )==3;
    T=3;
elseif get (handles.popupmenu6, 'Value' )==4;
    T=4;
elseif get (handles.popupmenu6, 'Value' )==5;
    T=5;
elseif get (handles.popupmenu6, 'Value' )==6;
    T=6;
```

```
elseif get (handles.popupmenu6, 'Value' )==7;
    T=7;
elseif get (handles.popupmenu6, 'Value' )==8;
    T=8;
else
    errordlg('error! ');
end
y=record0(Fs,T);%调用录音函数，返回信号
axes(handles.axes1);%定位到图框1绘制图形（左上）
plot(handles.axes1,y);
title('录制音频时域');
Y=abs(fft(y));
axes(handles.axes2);%定位到图框2绘制图形（右上）
plot(handles.axes2,Y(1:round(length(y)/2)-1));
title('录制音频频域');


% --- Executes on button press in pushbutton1.
%回声框里的“梳妆滤波器”按钮
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% 根据语音还是音乐选择采样率，语音为“1”，音乐为“2”
if get (handles.popupmenu5,'Value' )==1;
    Fs=8000;
elseif get (handles.popupmenu5, 'Value' )==2;
    Fs=44100;
else
    errordlg('error! ');
end
y=audioread('test.wav');%读取wav格式音频文件
%y=getappdata(handles.pushbutton4,'y');
a=0.5;%设置衰减系数
R=2000;%延时的时间长短
B=[1,zeros(1,R-1),a,zeros(1,R-1),a/2];%系统分子
A=[1];%系统分母
[h,w]=freqz(B,A);%求系统频响特性的函数
y2=filter(B,A,y);%滤波输出
Y2=fft(y2,length(y));%滤波输出的fft
Y3=abs(Y2);
sound(y2,Fs);%播放处理后的声音信号
axes(handles.axes3);%定位到图框3绘制图形（左下）
plot(handles.axes3,y2);
title('梳妆滤波结果的时域图');
```

```
axes(handles.axes4);%定位到图框4绘制图形（右下）
plot(handles.axes4,Y3(1:round(length(y)/2)-1));
title('梳妆滤波结果的频域图');



% --- Executes on button press in pushbutton2.
%回声框里的“全通滤波器”按钮
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 根据语音还是音乐选择采样率，语音为“1”，音乐为“2”
if get (handles.popupmenu5,'Value' )==1;
    Fs=8000;
elseif get (handles.popupmenu5, 'Value' )==2;
    Fs=44100;
else
    errordlg('error! ');
end
y=audioread('test.wav');%读取wav格式音频文件
a=0.5;%设置衰减系数
R=2000;%延时的时间长短
```

```
B=[a,zeros(1,R-1),1];%系统分子

A=[1,zeros(1,R-1),a];%系统分母

[h,w]=freqz(B,A);%求系统频响特性的函数

y2=filter(B,A,y);%滤波输出

Y2=fft(y2,length(y));%滤波输出的fft

Y3=abs(Y2);

sound(y2,Fs);%播放处理后的声音信号

axes(handles.axes3);%定位到图框3绘制图形（左下）

plot(handles.axes3,y2);

title('全通滤波结果的时域图');

axes(handles.axes4);%定位到图框4绘制图形（右下）

plot(handles.axes4,Y3(1:round(length(y)/2)-1));

title('全通滤波结果的频域图');



% --- Executes on button press in pushbutton8.

%均衡器框里的“录音并播放”按钮

function pushbutton8_Callback(hObject, eventdata, handles)

% hObject    handle to pushbutton8 (see GCBO)

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

% handles    structure with handles and user data (see GUIDATA)

% 根据语音还是音乐选择采样率，语音为“1”，音乐为“2”
```

```
if get (handles.popupmenu4,'Value' )==1;
    Fs=8000;
elseif get (handles.popupmenu4, 'Value' )==2;
    Fs=44100;
else
    errordlg('error! ');
end
% 根据“时长”框里的数选择时长（s）
if get (handles.popupmenu7,'Value' )==1;
    T=1;
elseif get (handles.popupmenu7, 'Value' )==2;
    T=2;
elseif get (handles.popupmenu7, 'Value' )==3;
    T=3;
elseif get (handles.popupmenu7, 'Value' )==4;
    T=4;
elseif get (handles.popupmenu7, 'Value' )==5;
    T=5;
elseif get (handles.popupmenu7, 'Value' )==6;
    T=6;
elseif get (handles.popupmenu7, 'Value' )==7;
    T=7;
```

```
elseif get (handles.popupmenu7, 'Value' )==8;
    T=8;
else
    errordlg('error! ');
end
y=record0(Fs,T);
axes(handles.axes1);%定位到图框1绘制图形（左上）
plot(handles.axes1,y);
title('录制音频时域');
Y=abs(fft(y));
axes(handles.axes2);%定位到图框2绘制图形（右上）
plot(handles.axes2,Y(1:round(length(y)/2)-1));
title('录制音频频域');



% --- Executes on button press in pushbutton5.
%均衡器框里的“均衡并播放”按钮
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[yy,Fs]=audioread('test.wav');
```

```
Y1=fft (yy);
fband =[100 200 500 1000 2000 3999 8000 16000 20000];% 频率分段
%根据语音还是音乐选择频段的多少，语音为“1”，音乐为“2”
%若为语音，则最高频为3999Hz，即num=5，否则最高频为20kHz,即num=8
if get (handles.popupmenu4,'Value' )==1;
    num=5;
elseif get (handles.popupmenu4, 'Value' )==2;
    num=8;
else
    errordlg('error! ');
end
y=zeros (size (yy));
%N为“阶数（1-4）”框里所选的阶数
N=get (handles.popupmenu3,'Value' );
%handles.slideri (i=1,...,8)表示从左到右的8个滑动条
slider=[handles.slider1,handles.slider2,handles.slider3,handles.slider4,handles.slider5,hand
les.slider6,handles.slider7,handles.slider8];
%handles. popupmenu1为“滤波器型号”框
%其中巴特沃兹为“1”，切比雪夫I型为“2”，切比雪夫II型为“3”，椭圆为
“4”
for i=1:num;
    k(i)=get (slider(i),'Value' )
```

```
        if get (handles. popupmenu1,'Value' )==1;
            [b a]=butter (N, 2*fband(i:i+1)/Fs);
        elseif get (handles. popupmenu1, 'Value' )==2;
            [b a]=cheby1 (N, 0.5, 2*fband(i:i+1)/Fs);
        elseif get (handles. popupmenu1,'Value' )==3;
            [b a]=cheby2 (N, 20, 2*fband(i:i+1)/Fs);
        elseif get (handles. popupmenu1, 'Value' )==4;
            [b a]=ellip(N, 0.5, 20, 2*fband(i:i+1)/Fs);
        else
            errordlg(' No filter type chosen or filter type error! ');
        end
        y=y+k(i)*filter (b, a, yy);
end
Y2=fft(y);
Y3=abs(Y2);
sound(y,Fs);
axes(handles.axes3);
plot(handles.axes3,y);
title('均衡后音频时域');
axes(handles.axes4);
plot(handles.axes4,Y3(1:round(length(y)/2)-1));
title('均衡后音频频域');
```

```
% --- If Enable == 'on', executes on mouse press in 5 pixel border.
% --- Otherwise, executes on mouse press in 5 pixel border or over pushbutton2.
function pushbutton2_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% 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 axes1


% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
% hObject    handle to slider1 (see GCBO)
```

```
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%              get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to slider1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider2_Callback(hObject, eventdata, handles)
% hObject      handle to slider1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
```

```
% handles        structure with handles and user data (see GUIDATA)


% Hints: get(hObject,'Value') returns position of slider
%                get(hObject,'Min') and get(hObject,'Max') to determine range of slider



% --- Executes during object creation, after setting all properties.
function slider2_CreateFcn(hObject, eventdata, handles)
% hObject        handle to slider1 (see GCBO)
% eventdata     reserved - to be defined in a future version of MATLAB
% handles        empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
% hObject        handle to slider3 (see GCBO)
% eventdata     reserved - to be defined in a future version of MATLAB
% handles        structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider4_Callback(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%            get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider6_Callback(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%            get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%            get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider7_Callback(hObject, eventdata, handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%            get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function slider8_Callback(hObject, eventdata, handles)
% hObject    handle to slider8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function slider8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on selection change in popupmenu1.
function popupmenu1_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu1



% --- Executes during object creation, after setting all properties.
function popupmenu1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes during object creation, after setting all properties.
function uipanel1_CreateFcn(hObject, eventdata, handles)
```

```
% hObject    handle to uipanel1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a double



% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on mouse press over axes background.
function axes1_ButtonDownFcn(hObject, eventdata, handles)
% hObject    handle to axes1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on selection change in popupmenu3.
```

```
function popupmenu3_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu3 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu3


% --- Executes during object creation, after setting all properties.
function popupmenu3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on selection change in popupmenu4.
function popupmenu4_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu4 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu4


% --- Executes during object creation, after setting all properties.
function popupmenu4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
```

```
get(0,'defaultUicontrolBackgroundColor'))
        set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in popupmenu5.
function popupmenu5_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu5 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu5


% --- Executes during object creation, after setting all properties.
function popupmenu5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
```

```
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on selection change in popupmenu6.
function popupmenu6_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu6 contents as cell
array
%           contents{get(hObject,'Value')} returns selected item from popupmenu6



% --- Executes during object creation, after setting all properties.
function popupmenu6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

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


% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on selection change in popupmenu7.
function popupmenu7_Callback(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu7 contents as cell
array
%        contents{get(hObject,'Value')} returns selected item from popupmenu7



% --- Executes during object creation, after setting all properties.
```

```
function popupmenu7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to popupmenu7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

（2）实验结果

（一）采样率和量化级数对于信号的影响

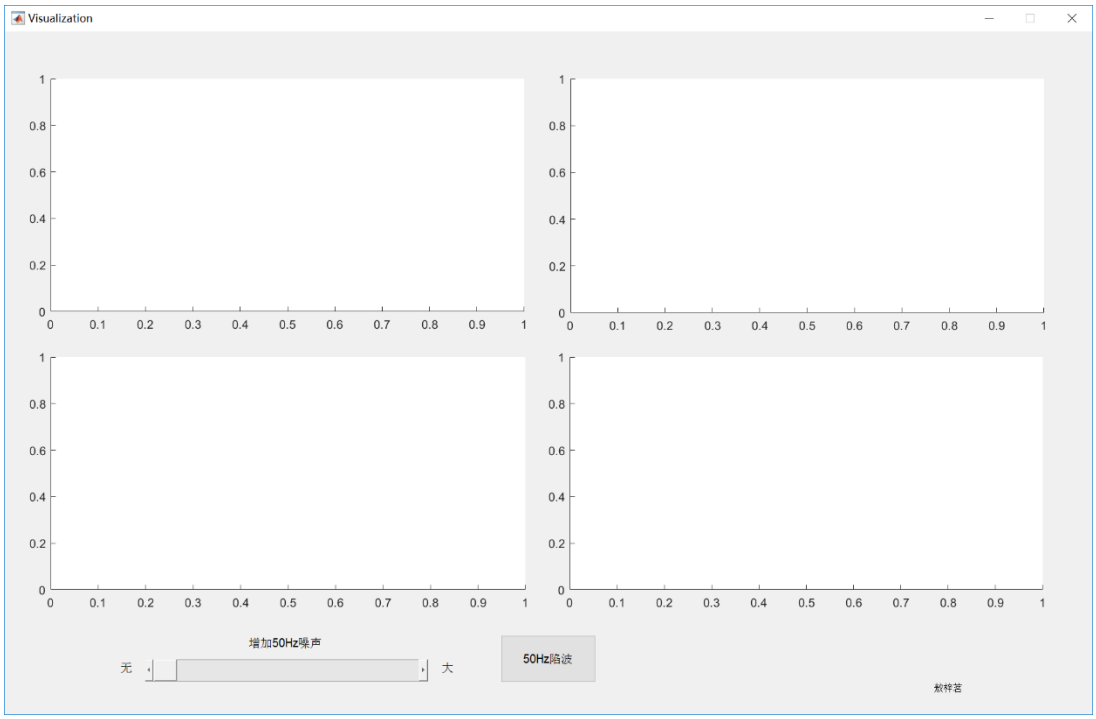
1.采样及量化的结果

分别使用了 8k,16k,32k 三种采样率录制了三段语音信号,量化级数均采用 16 位,通过反复的回放对比,发现虽然高于 8k 的采样率的声音更加饱满,音质更佳,但是对于语音信号来说,8 采样率用于通话交流系统已经足够了,采样率的提高对于音质的提升不大,反而需要更大的信道带宽,得不偿失,同样的,之后又分别录制了 8k,16k 以及 32k 三种采样率的音乐信号,通过反复对比可以明显感觉到 8k 采样率的音乐明显少了许多频率成分,失真严重,高音部分基本听不到了,总之对于音乐信号,8k 采样率太低不适合音乐的欣赏,16k 采样率较 8k 来说频率成分丰富了很多,音乐更加优美动听,32k 采样率在此基础上音质有所提升,但个人感觉提升不是很大。

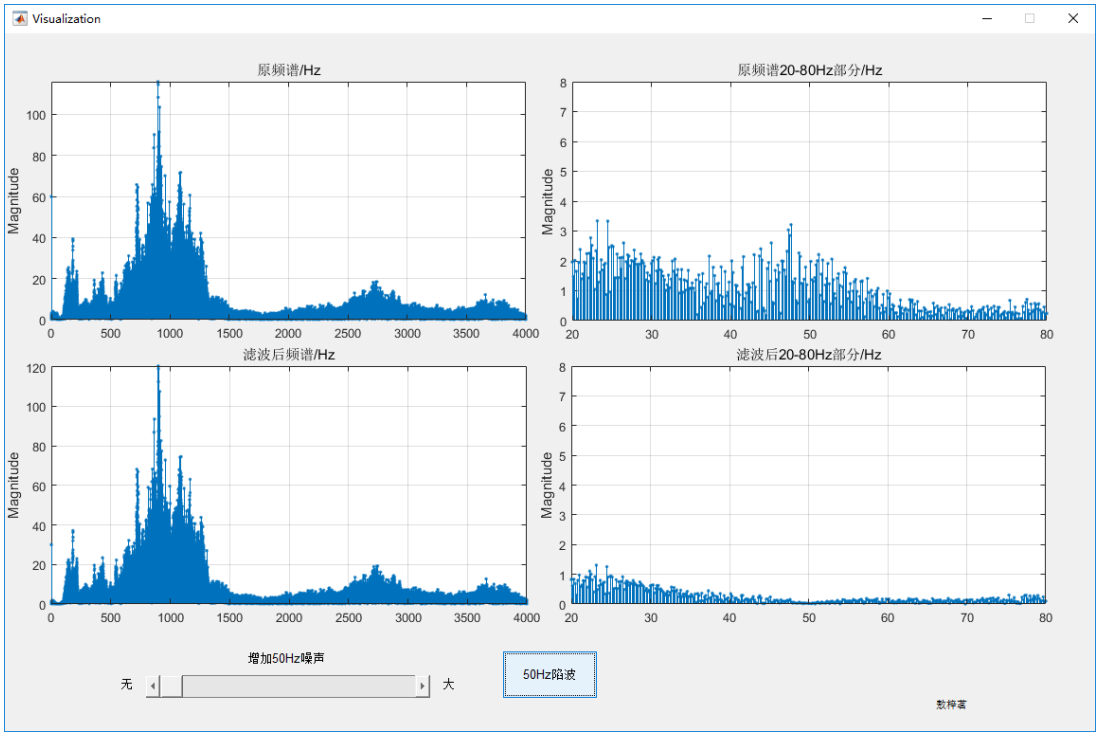
16k 音乐信号的三种不同量化电平(8bit,16bit,32bt)的效果,显然地 8bit 对应只有 256 个量化电平,音乐听起来相当粗糙,明不合适,16bit 已经达到了 65536 个量化电平,是当前音乐主流的量化数,32b1t 占用了大量了存储空间,音质提升也不

大。

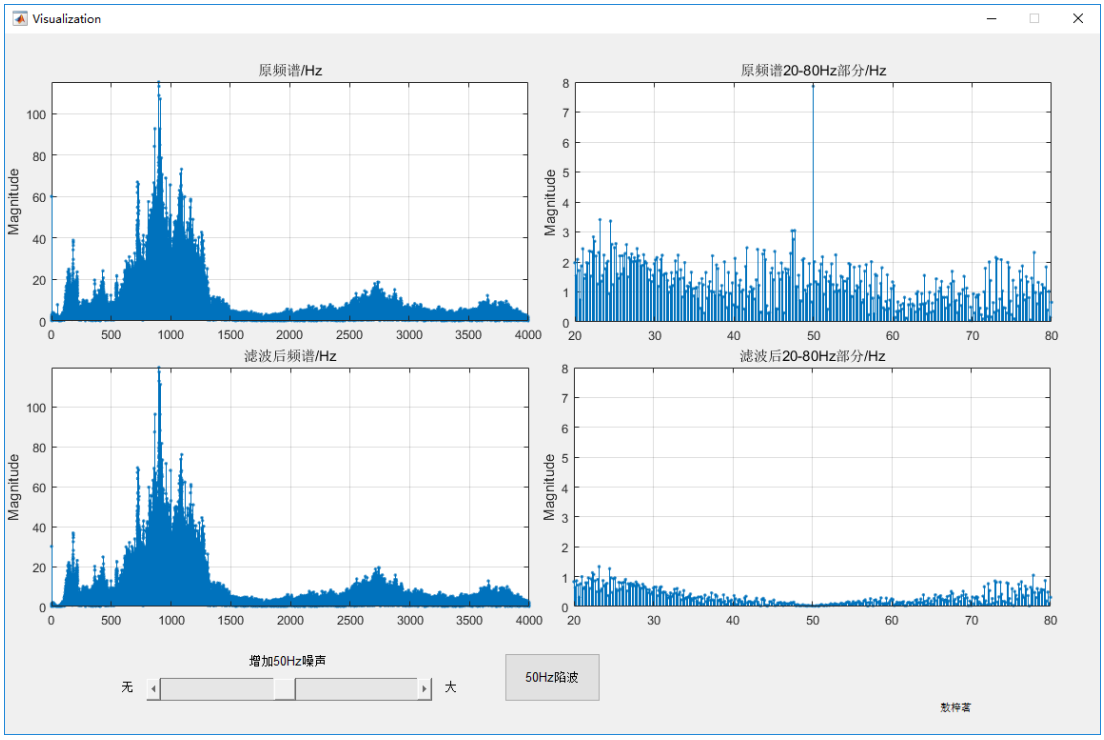
2.滤波器运行结果如图:



(a) 初始界面



(b) 不加噪声时



(c)加了噪声时

（二）回声及均衡器实验

1.回声及均衡器实验过程分析

音频信号分为语音和音乐，采样率分别 8kHz 和 44100Hz。下面的实验结果图中，均衡器横坐标对应频率：

横坐标（0-2.5）在语音时	横坐标（0-14）在语音时
100Hz~0.0625	100Hz~0.0636
200Hz~0.125	200Hz~0.127
500Hz~0.3125	500Hz~0.31
1kHz~0.625	1kHz~0.64
2kHz~1.25	2kHz~1.27
3999Hz~2.49	3999Hz~2.54
	8kHz~5.09
	16KHz~10.18
	20KHz~12.72

回声：

我们设计的滤波器如下：

梳妆滤波器：y(n)=x(n)+ax(n-R)+0.5*ax(n-R), a<1(回声衰减系数)；

全通滤波器：$H(z)=\frac{\alpha+z^{-R}}{1+\alpha z^{-R}},|\alpha|<1$

均衡器：

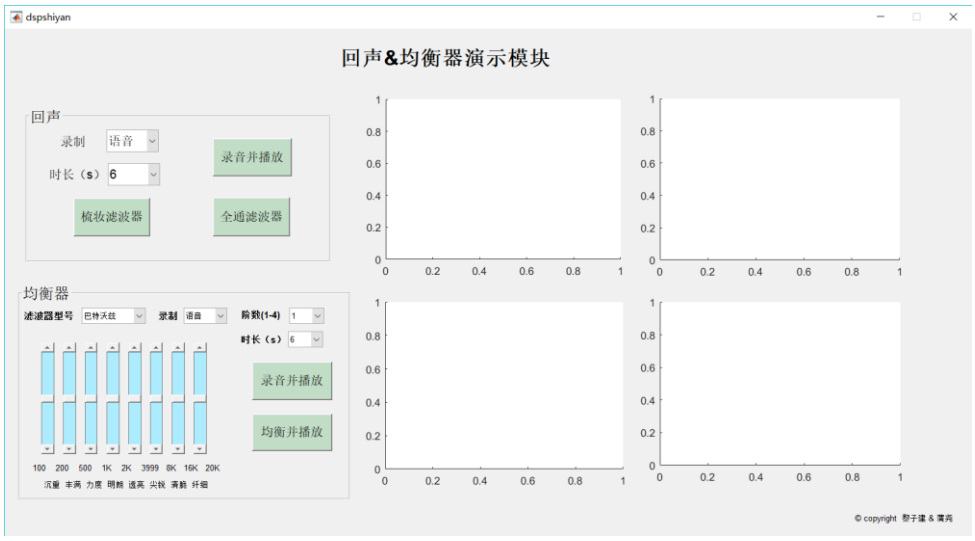
根据所选的阶数 N、均衡的频率范围 Wn 以及滤波器种类实现均衡效果，函数如下：

```
[b a]=butter (N, 2*fband(i:i+1)/Fs);
[b a]=cheby1 (N, 0.5, 2*fband(i:i+1)/Fs);
[b a]=cheby2 (N, 20, 2*fband(i:i+1)/Fs);
[b a]=ellip(N, 0.5, 20, 2*fband(i:i+1)/Fs);
```

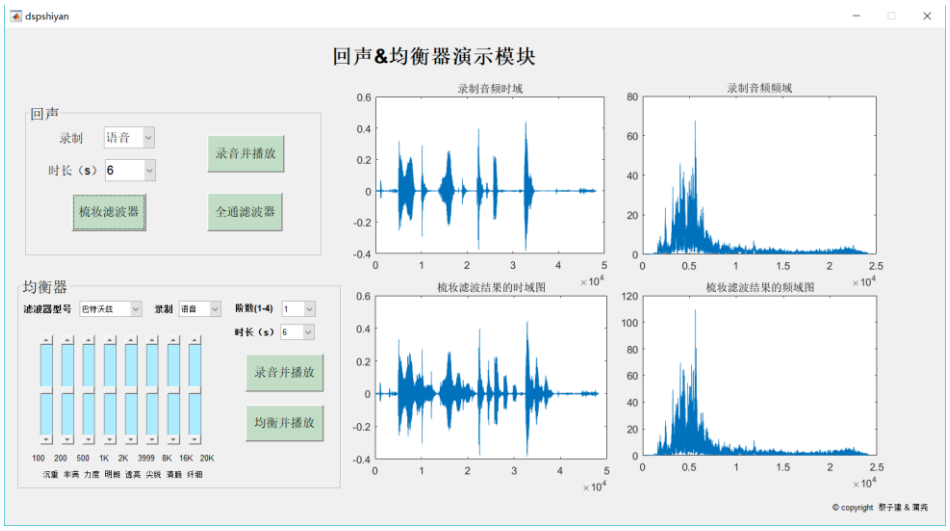
本设计中的均衡器有八段，分别用八个滚动条来表示各个频率段经过带通滤波后信号幅值的加权值大小（范围是 0-10）。用户可选取四种分别基于巴特沃斯法、切比雪夫 I、切比雪夫 II 、椭圆法的均衡器。

用户的选择功能可用一个下拉菜单来实现。对于滤波器的阶数本应该是有一定的函数进行计算的，本设计中由于各个滤波器的频率大小有很大的跨越，经过计算和实验，得出了满足要求的阶数的范围

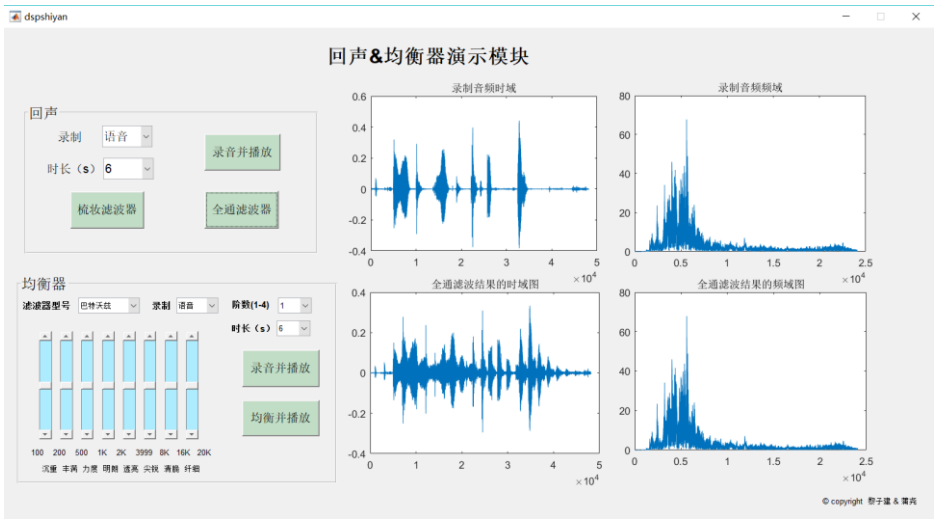
2.回声及均衡器运行截图



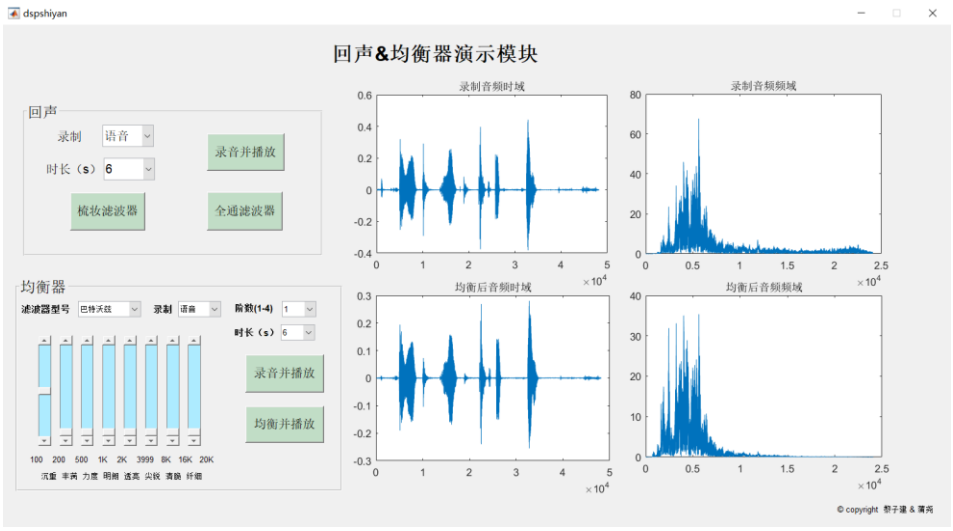
(A)回声及均衡器演示模块



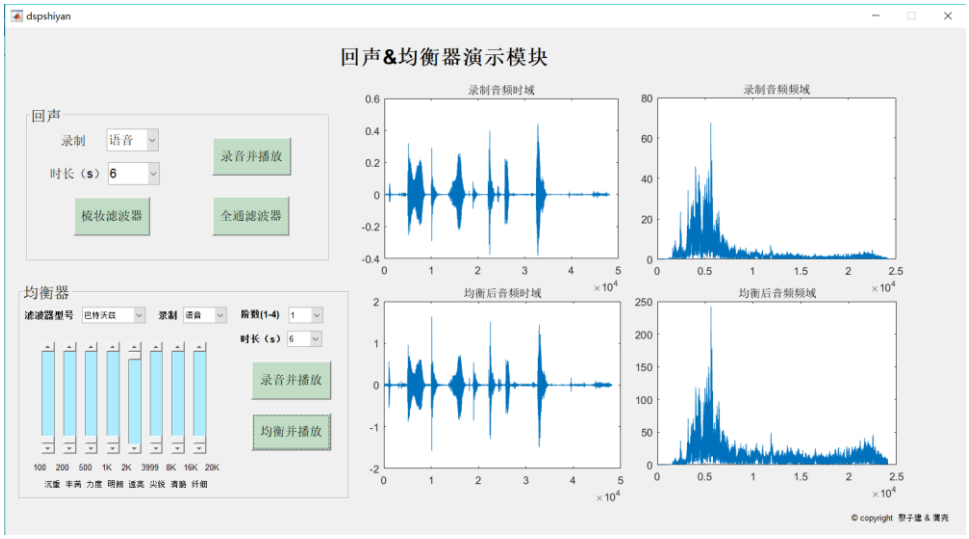
(B)梳妆滤波器（语音）



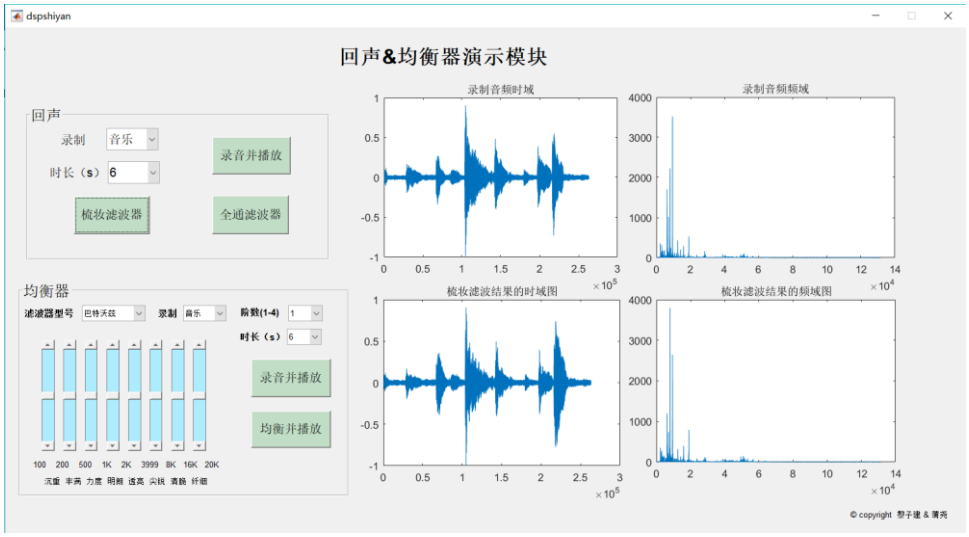
（C）全通滤波器（语音）



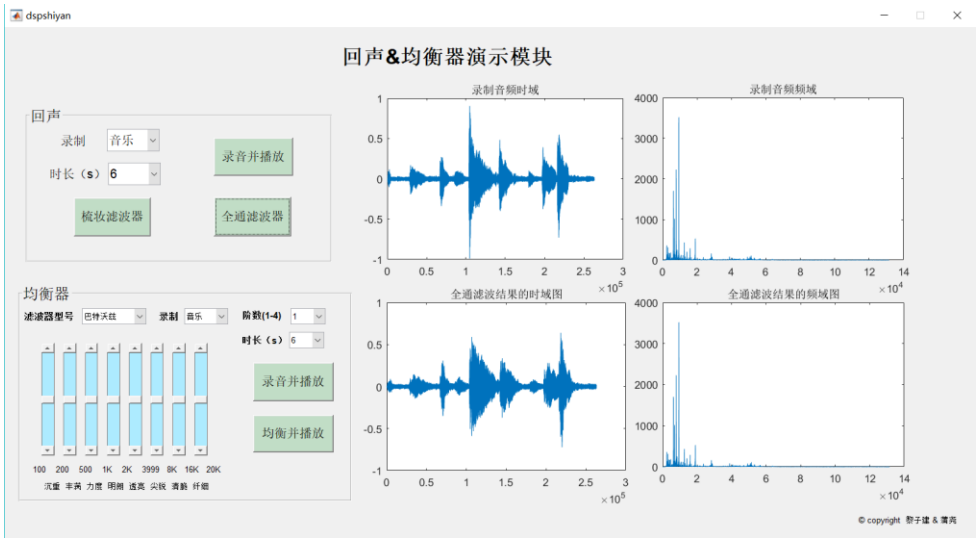
（D）均衡器（语音沉重）



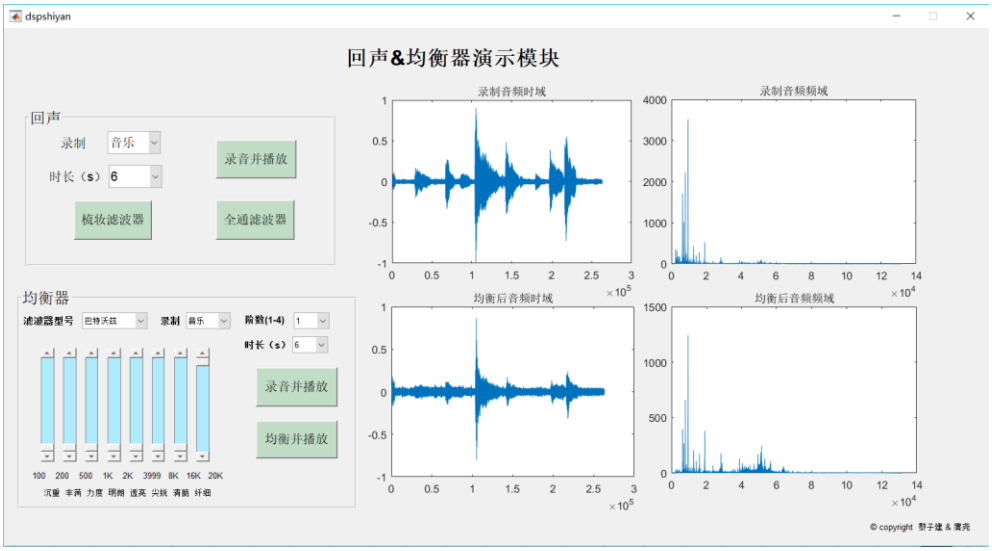
（E）均衡器（语音透亮）



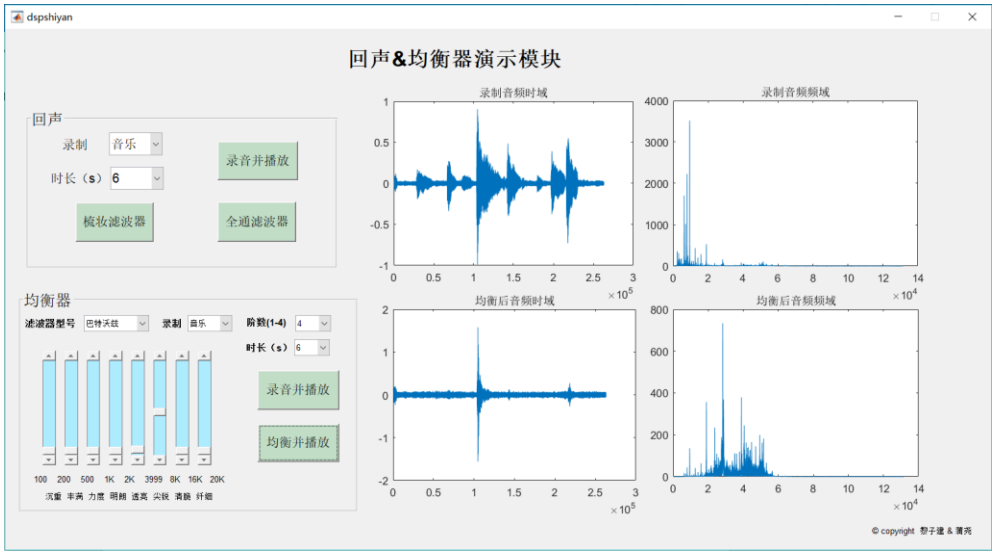
(F)梳妆滤波器（音乐）



（G）全通滤波器（音乐）



（H）均衡器（音乐）



（H）均衡器（音乐 4 阶）

五、实验结论

1. 对于语音信号，采用 8kHz 的采样率，8 位量化电平数已经足以进行语音交流，而对于音乐信号，由于频率范围较大，最好能达到 16khz 以上的采样率，量化电平 16bit 足够了。

2. 现在的电脑电源滤波效果较好，录音频谱中 50Hz 的交流噪声分量很小，所以演示时为了体现陷波器的效果，需要额外混进 50Hz 的正弦波噪声进行演示；陷波器本质是滤波器，其过渡带大小会影响到周围频谱分量的大小，能够观察到进行 50Hz 陷波后其周围频率分量均有降低。

人工混进噪声频率后，其他频率分量大小均有变化，所以特定频率的噪声也会对整体的语音效果造成影响。

3. 梳妆滤波器和全通滤波器，梳妆滤波器是 FIR 滤波器，发现梳妆滤波器返回

的回声的次数与自己设计的延时函数的个数有关，我们设计的是有 1 个和 2 个延时：

y(n)=x(n)+ax(n-R), a<1(回声衰减系数)；听到有 1 个回声。

y(n)=x(n)+ax(n-R)+0.5*ax(n-R), a<1(回声衰减系数)；听到有 2 个回声。

而全通滤波器是 IIR 滤波器，所以回声会有很多段，更加贴合实际。

4. 均衡器通过改变不同频域的信号功率通过加权和达到实现不同音效的效果。语音信号和音乐信号经过均衡器后各频率分量幅度响应较之前更加均匀。阶数越高，滤波效果更好。

六、实验感想

邢靖：

通过本次试验，我理解了采样率和量化级数对语音信号和音乐信号的影响，明白的语音信号和音乐信号的区别，对回声合成和均衡器原理有了一定的了解。加深了对数字信号处理的理解。

蒲尧：

通过本次与同学合作完成 DSP 课程设计，学到很多东西：

了解了梳妆滤波器是通过设计多个延时器实现的，而全通滤波器是不断反馈实现回声效果。比较两者，前者的回声效果较为呆板，而后者的效果更加真实。

还了解了均衡器的设计，通过设置阶数 N，Wp、Ws、Rp、Rs、滤波器种类等参数，实现了对语音及音乐信号的均衡。滤波器的作用是将这些干扰成分滤除，也就是让特定频段的信号通过达到对信号筛选的效果。应用 MATLAB 语言进行 FIR 数字滤波器的设计仿真时，可根据设计要求随时改变参数，以使滤波器达到最优化。在经典的滤波器中，通过对 IIR 数字滤波器的设计研究，应用 Butterworth 滤波器、Chebyshevel 型滤波器、ChebyshevII 型滤波器以及椭圆滤波器四种形式，分别用这四种方法设计的均衡器进行比较仿真，得到其不同的仿真特性。虽然效果并不是十分理想，但对课本知识有了更加深入的了解。

通过使用 guide 指令，实现了人机交互，总体设计的界面比较友好。

黎子建：

在此次的课程设计实验过程中，利用 matlab 对一段语音及音乐信号的采样、滤波及处理，我对采样率和量化级数对语音信号的影响有了更深的理解。利用 matlab 产生回声，我对梳妆滤波器和全通滤波器有了全面的了解；利用 matlab 产生均衡器对信号进行均衡，我对均衡器的原理也有了更深的理解。

在此次实验，我主要负责回声和均衡器部分。

在回声部分，我们做了梳妆滤波器和全通滤波器，发现梳妆滤波器返回的回声的次数与自己设计的延时函数的个数有关，而全通滤波器是 IIR 滤波器，所以回声会有很多段，更加贴合实际。

在均衡器部分，我们直接调用了 matlab 自带的函数，如巴特沃兹滤波器、切比雪夫 I 型滤波器、切比雪夫 II 型滤波器和椭圆滤波器，来设计我们的滤波器。语音信号和音乐信号经过均衡器后各频率分量幅度响应较之前更加均匀。

总而言之，在此次课程设计实验中，我对 DSP 的一些概念有了更深的理解，对 matlab 的操作也愈加熟练。

敖梓茗：

本次实验我负责做 50Hz 陷波滤波器，还体会了不同采样率和量化比特对音质的影响。50Hz 陷波滤波器的实验中我学会了陷波滤波器的设计，以及陷波器过渡带对周围频率信号

的影响。还有我学会了写图形界面，以后改参数就不用老是改源程序，可以在图形界面上直观地调整参数大小。另外如果该实验用的是笔记本电脑录音，要想采集到 50Hz 噪声信号必须接上电源，最好还能把电池拆下来，用老旧的台式机来做实验可能交流电噪声对录音的影响会更明显一点。

在做采样率对话音、音乐声音质量的实验时，我能直观感受到不同采样率（从 44.1K 降到 8K）和不同量化比特（从 16bit 降到 8bit）对音质的影响。不过需要注意的是，因为我们电脑的录音设备太烂，我在用 CD 音质(44.1KHz, 16bit)录音再把它降采样到 8K 时，音质并无太大变化，从 16bit 降到 8bit 也同理。所以要体会采样率和量化比特不同对音质的影响，我们需要选择好的音源和一副好的耳机，方能体会到所谓无损音质和有损音质的差别。