大连理工大学实验报告

学院（系）：电信学部　专业：电子信息工程（英语强化）　班级：电英 1801

姓　　名：童博涵　学号：201883032　组：

实验时间：　实验室：　实验台：

指导教师签字：　成绩：

实验二 含噪语音信号的分析

一、实验题目和结果

1. 函数代码：

```
% --- Executes on selection change in filter_choose.
function filter_choose_Callback(hObject, eventdata, handles)
% hObject    handle to filter_choose (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 filter_choose contents as cell array
%        contents{get(hObject,'Value')} returns selected item from filter_choose
%用于选择不同滤波器之后界面中参数输入框的变化
contents = cellstr(get(hObject,'String'));
if get(handles.IIR_btn,'Value') == 1

    if ((get(hObject,'Value')) == 3 || ...
    (get(hObject,'Value')) == 4 || ...
    (get(hObject,'Value')) == 7 || ...
    (get(hObject,'Value')) == 8 || ...
    (get(hObject,'Value')) == 11 || ...
    (get(hObject,'Value')) == 12)

        set(handles.fL2_edit,'Visible','on');
        set(handles.fH2_edit,'Visible','on');
        set(handles.text11,'Visible','on');
        set(handles.text12,'Visible','on');
    else
        set(handles.fL2_edit,'Visible','off');
```

```
        set(handles.fH2_edit,'Visible','off');
        set(handles.text11,'Visible','off');
        set(handles.text12,'Visible','off');
    end
end

if get(handles.IIR_btn,'Value') == 1

    if ((get(hObject,'Value')) == 1 || ...
    (get(hObject,'Value')) == 2 || ...
    (get(hObject,'Value')) == 3 || ...
    (get(hObject,'Value')) == 4)
        set(handles.n_edit,'Visible','off');
        set(handles.text13,'Visible','off');

    else
        set(handles.n_edit,'Visible','on');
        set(handles.text13,'Visible','on');
    end
end

if get(handles.FIR_btn,'Value') == 1
    set(handles.n_edit,'Visible','on');
    set(handles.text13,'Visible','on');
    set(handles.fL2_edit,'Visible','off');
    set(handles.fH2_edit,'Visible','off');
    set(handles.text11,'Visible','off');
    set(handles.text12,'Visible','off');
    set(handles.text8,'Visible','off');
    set(handles.text9,'Visible','off');
    set(handles.Rp_edit,'Visible','off');
    set(handles.Rs_edit,'Visible','off');
end
```

```
% --- Executes on button press in file_btn.
function file_btn_Callback(hObject, eventdata, handles)
% hObject    handle to file_btn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%用于选择要播放的文件
global y1
global y_copy
[file, path] = uigetfile({'*.mp3;*.wav;*.mp4;*.m4a','audio files(*.mp3;*.wav;*.mp4;*.m4a)'},...
```

```
    '选择音频文件');
if length(file) == 1 && length(path) == 1
    h = warndlg('请选择文件','警告','modal');
    return;
end
handles.filename = strcat(path,file);
[handles.y,handles.fs] = audioread(handles.filename);
y1 = handles.y;
y_copy = handles.y;
% player = audioplayer(handles.y,handles.fs);
total_sec = length(handles.y)/handles.fs;
set(handles.play_slider,'Max',total_sec);
set(handles.play_slider,'Value',0);
if ~isempty(timerfind)
    stop(timerfind);
    delete(timerfind);
end
h = timer;
handles.tmr = h;
% h.ExecutionMode = 'fixedRate';
% h.Period = 1;
% h.timerFcn = {@disptime,handles};
% start(h);
set(handles.tmr,'ExecutionMode','fixedRate');   %定时器，循环执行，循环定时。
set(handles.tmr,'Period',1);    %定时器，定时间隔 1 秒
set(handles.tmr,'TimerFcn',{@timer_sub,handles});    %定时器，定时会触发 TimerFcn 函数，定时函数
(TimerFcn)触发用户自定义的函数(disptime 函数)
if size(handles.y) == [length(handles.y) 2] %多声道

    minutes = fix(total_sec(:,1)/60);
    seconds = fix(total_sec(:,1)-minutes*60);
    set(handles.right_btn,'Visible','On');
    set(handles.stereo_btn,'Visible','On');
    set(handles.left_btn,'String','左声道');
    if get(handles.left_btn,'Value') == 1 %选择左声道
        y1 = y1(:,1);
    elseif get(handles.right_btn,'Value') == 1 %选择右声道
        y1 = y1(:,2);
    else %立体声
        y1 = y1;
    end

else
    minutes = fix(total_sec/60); %单声道
    seconds = fix(total_sec-minutes*60);
```

```
    set(handles.right_btn,'Visible','Off');
    set(handles.stereo_btn,'Visible','Off');
    set(handles.left_btn,'String','单声道');
end

if minutes <10 %显示音频时长
    set(handles.time_text,'String',strcat('00:00/0',num2str(minutes),':',num2str(seconds)));
elseif seconds < 10
    set(handles.time_text,'String',strcat('00:00/',num2str(minutes),':0',num2str(seconds)));
else
    set(handles.time_text,'String',strcat('00:00/',num2str(minutes),':',num2str(seconds)));
end

t = 1:1:length(handles.y);
t = t./handles.fs;


left_channel_en = get(handles.left_btn,'Value');
right_channel_en = get(handles.right_btn,'Value');
stereo_channel_en = get(handles.stereo_btn,'Value');
% 在界面中画出音频波形
right_flag = get(handles.right_btn,'Visible') == 'on';
right_flag = all(right_flag(:)==1);
if left_channel_en == 1
    y_t = handles.y(:,1);
    y_t = y_t';
    plot(handles.sig_show,t,y_t)
    xlabel(handles.sig_show,'s')
    ylabel(handles.sig_show,'amplitude')
    fft_y = fft(handles.y);
    fft_y = fft_y(:,1);
    N = length(t);
    dw = (0:N-1)*handles.fs/N-handles.fs/2;
    plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
elseif right_channel_en == 1 && right_flag == 1
    y_t = handles.y(:,2);
    y_t = y_t';
    plot(handles.sig_show,t,y_t)
    xlabel(handles.sig_show,'r')
    ylabel(handles.sig_show,'amplitude')
    fft_y = fft(handles.y);
    fft_y = fft_y(:,2);
    N = length(t);
    dw = (0:N-1)*handles.fs/N-handles.fs/2;
    plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
```

```
else
    y_t = handles.y(:,1);
    y_t = y_t';
    plot(handles.sig_show,t,y_t,'r')
    hold on
    y_t2 = handles.y(:,2);
    y_t2 = y_t2';
    plot(handles.sig_show,t,y_t2,'b')
    xlabel(handles.sig_show,'s')
    ylabel(handles.sig_show,'amplitude')
    fft_y = fft(handles.y);
    N = length(t);
    dw = (0:N-1)*handles.fs/N-handles.fs/2;
    plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
end
guidata(hObject, handles);
```

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

% Hint: get(hObject,'Value') returns toggle state of sw_btn
% 控制音频的播放与暂停
global player
global y_copy
% sound(y_copy,handles.fs)
if isempty(player)
    if handles.flag1 == 0
        player = audioplayer(y_copy,handles.fs);
        handles.flag1 = 1;
    else
        player = audioplayer(handles.y,handles.fs); %生成 audioplayer 对象
    end
end
% disp(handles.fs)
if ~isempty(handles.y)
    f = get(hObject,'Value');
    if f == 1 && handles.flag == 0 %第一次按下
        start(handles.tmr);
        play(player);
        set(hObject,'String','暂停');
```

```
    elseif f == 1 && handles.flag == 1 %继续播放
        resume(player);
        set(hObject,'String','暂停');
    else %暂停
        pause(player);
        handles.flag = 1;
        set(hObject,'String','播放');
        set(handles.filt_sig,'Enable','off');
        set(handles.origin_sig,'Enable','off');
    end
end

s = get(hObject,'String');
if get(hObject,'Value') == 1 || s(1) == '暂'
    set(handles.filt_sig,'Enable','off');
    set(handles.origin_sig,'Enable','off');
else
    set(handles.filt_sig,'Enable','off');
    set(handles.origin_sig,'Enable','off');
end
guidata(hObject, handles);
```

```
% --- Executes on button press in stop_btn.
function stop_btn_Callback(hObject, eventdata, handles)
% hObject    handle to stop_btn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global player
% if ~isempty(handles.y)
stop(player);
if ~isempty(timerfind)
    stop(timerfind);
end
handles.flag = 0;
set(handles.sw_btn,'String','播放');
set(handles.sw_btn,'Value',0);
set(handles.play_slider,'Value',0);
set(handles.time_text,'String','00:00/00:00');
guidata(hObject, handles);
set(handles.filt_sig,'Enable','on');
set(handles.origin_sig,'Enable','on');
```

```
% --- Executes on button press in gen_filter_btn.
function gen_filter_btn_Callback(hObject, eventdata, handles)
% hObject    handle to gen_filter_btn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 生成滤波器进行滤波
global A
global B
global player
global y1
global y_copy
contents = cellstr(get(handles.filter_choose,'String'));
filter_name = contents{get(handles.filter_choose,'Value')};
disp(strcat('name: ',filter_name));
fL = str2double(get(handles.fL_edit,'String'));
fH = str2double(get(handles.fH_edit,'String'));
Rp = str2double(get(handles.Rp_edit,'String'));
Rs = str2double(get(handles.Rs_edit,'String'));
fL1 = str2double(get(handles.fL2_edit,'String'));
fH1 = str2double(get(handles.fH2_edit,'String'));
step = str2double(get(handles.n_edit,'String'));
if step <1
    return
end
if handles.fs == 0 %未输入信号
    fs = 2000;
    Wp = fL/(fs/2);
    Ws = fH/(fs/2);
    Wp1 = fL1/(fs/2);
    Ws1 = fH1/(fs/2);
    disp('using default sample freq 1000Hz')
    if fL < 1000 || fH < 1000
        h = warndlg('freqency should less than 1000Hz','警告','modal');
        return;
    end
else
    Wp = fL/(handles.fs/2);
    Ws = fH/(handles.fs/2);
    Wp1 = fL1/(handles.fs/2);
    Ws1 = fH1/(handles.fs/2);
    disp(strcat('using input sample freq:',num2str(handles.fs),'Hz'));
end

if get(handles.IIR_btn,'Value') == 1
```

```
    disp('IIR')

    Wp2 = [Wp Wp1]; %选择带通和带阻输入两个参数
    Ws2 = [Ws Ws1];
    vi = get(handles.fL2_edit,'Visible');
    vi = (vi(2) == 'f');
    vi = ~all(vi(:)==1);
    if ~(Ws < Wp && Wp < Wp1 && Wp1 < Ws1)&&vi %输入参数错误
        h = warndlg('fH < fL < fL2 < fH2','警告','modal');
        return
    elseif ~(Ws > Wp)&&(~vi)
        h = warndlg('fH > fL ','警告','modal');
        return

    else
        switch filter_name
            case 'Butterworth-lowpass'
                [N,Wn] = buttord(Wp,Ws,Rp,Rs);
                [B,A] = butter(N,Wn,'low');
            case 'Butterworth-highpass'
                [N,Wn] = buttord(Wp,Ws,Rp,Rs);
                [B,A] = butter(N,Wn,'high');
            case 'Butterworth-bandpass'
                [N,Wn] = buttord(Wp2,Ws2,Rp,Rs);
                [B,A] = butter(N,Wn,'bandpass');
            case 'Butterworth-bandstop'
                [N,Wn] = buttord(Wp2,Ws2,Rp,Rs);
                [B,A] = butter(N,Wn,'stop');
            case 'Chebyshev-I-lowpass'
                [N,Wn] = cheb1ord(Wp,Ws,Rp,Rs);
                [B,A] = cheby1(step,N,Wn,'low');
            case 'Chebyshev-I-highpass'
                [N,Wn] = cheb1ord(Wp,Ws,Rp,Rs);
                [B,A] = cheby1(step,N,Wn,'high');
            case 'Chebyshev-I-bandpass'
                Wp2 = [Wp Wp1];
                Ws2 = [Ws Ws1];
                [N,Wn] = cheb1ord(Wp2,Ws2,Rp,Rs);
                [B,A] = cheby1(step,N,Wn,'bandpass');
            case 'Chebyshev-I-bandstop'
                Wp2 = [Wp Wp1];
                Ws2 = [Ws Ws1];
                [N,Wn] = cheb1ord(Wp2,Ws2,Rp,Rs);
                [B,A] = cheby1(step,N,Wn,'stop');
            case 'Chebyshev-II-lowpass'
```

```
                [N,Wn] = cheb2ord(Wp,Ws,Rp,Rs);
                [B,A] = cheby2(step,N,Wn,'low');
            case 'Chebyshev-II-highpass'
                [N,Wn] = cheb2ord(Wp,Ws,Rp,Rs);
                [B,A] = cheby2(step,N,Wn,'high');
            case 'Chebyshev-II-bandpass'
                Wp2 = [Wp Wp1];
                Ws2 = [Ws Ws1];
                [N,Wn] = cheb2ord(Wp2,Ws2,Rp,Rs);
                [B,A] = cheby2(step,N,Wn,'bandpass');
            case 'Chebyshev-II-bandstop'
                Wp2 = [Wp Wp1];
                Ws2 = [Ws Ws1];
                [N,Wn] = cheb2ord(Wp2,Ws2,Rp,Rs);
                [B,A] = cheby2(step,N,Wn,'stop');
        end
    end

    [H,w] = freqz(B,A);
%     disp(abs(H/max(H)))

    db = 20*log10(abs(H/max(H)));
%     if (any(isnan(db(:))) || step > 50) && get(handles.n_edit,'Visible') == 'on'
%         h = warndlg('滤波器阶次过大','警告','modal');
%     end

    plot(handles.filter_show,w/pi,db)
    xlabel(handles.filter_show,'归一化频率')
    title(handles.filter_show,'滤波器频响')
    handles.y_filter = filtfilt(B,A,y_copy);
    disp('filter in filt')
    disp(size(handles.y_filter))
%     handles.y = handles.y_filter;
%     y1 = handles.y;
    if handles.fs ~= 0
        player = audioplayer(handles.y_filter,handles.fs);
    end
    if get(handles.filt_sig,'Value') == 1
        t = 1:1:length(handles.y);
        t = t./handles.fs;
        fft_y = fft(handles.y_filter);
        fft_y = fft_y(:,1);
        N = length(t);
        dw = (0:N-1)*handles.fs/N-handles.fs/2;
        plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
```

```
        plot(handles.sig_show,t,handles.y_filter(:,1))
    end

else
    disp('FIR')
    if Ws < Wp
        h = warndlg('fL < fH ','警告','modal');
        return
    else
        switch filter_name
            case 'hamming-lowpass'
                b = fir1(step,Wp,'low');
                [H,w] = freqz(b,1);
            case 'hamming-highpass'
                b = fir1(step,Ws,'high');
                [H,w] = freqz(b,1);
            case 'hamming-bandpass'
                b = fir1(step,[Wp Ws],'bandpass');
                [H,w] = freqz(b,1);
            case 'hamming-bandstop'
                b = fir1(step,[Wp Ws],'stop');
                [H,w] = freqz(b,1);
            case 'kaiser-lowpass'
                b = fir1(step,Wp,'low',kaiser(step+1,0.5));
                [H,w] = freqz(b,1);
            case 'kaiser-highpass'
                b = fir1(step,Ws,'high',kaiser(step+1,0.5));
                [H,w] = freqz(b,1);
            case 'kaiser-bandpass'
                b = fir1(step,[Wp Ws],'bandpass',kaiser(step+1,0.5));
                [H,w] = freqz(b,1);
            case 'kaiser-bandstop'
                b = fir1(step,[Wp Ws],'stop',kaiser(step+1,0.5));
                [H,w] = freqz(b,1);
        end
    end
            db = 20*log10(abs(H/max(H)));
            plot(handles.filter_show,w/pi,db)

            if get(handles.filt_sig,'Value') == 1
                handles.y_filter = filtfilt(b,1,y_copy(:,1));
                t = 1:1:length(handles.y);
                t = t./handles.fs;
                fft_y = fft(handles.y_filter);
                fft_y = fft_y(:,1);
```

```
                N = length(t);
                dw = (0:N-1)*handles.fs/N-handles.fs/2;
                plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
                plot(handles.sig_show,t,handles.y_filter(:,1))
                if handles.fs ~= 0
                    player = audioplayer(handles.y_filter,handles.fs);
                end
            end

end
```

```
% --- Executes when selected object is changed in uibuttongroup_channel.
function uibuttongroup_channel_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup_channel
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 选择声道
current_selection = get(handles.uibuttongroup_channel,'SelectedObject');
current_tag = get(current_selection,'Tag');
t = 1:1:length(handles.y);
t = t./handles.fs;
if ~isempty(handles.y)
    switch current_tag
        case 'left_btn'
            y_t = handles.y(:,1);
            y_t = y_t';
            plot(handles.sig_show,t,y_t)
            xlabel(handles.sig_show,'s')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            fft_y = fft_y(:,1);
            N = length(t);
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
        case 'right_btn'
            y_t = handles.y(:,2);
            y_t = y_t';
            plot(handles.sig_show,t,y_t)
            xlabel(handles.sig_show,'s')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            fft_y = fft_y(:,2);
            N = length(t);
```

```
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
        case 'stereo_btn'
            y_t = handles.y;
            y_t = y_t';
            plot(handles.sig_show,t,y_t,'r')
            hold on
            y_t2 = handles.y(:,2);
            y_t2 = y_t2';
            plot(handles.sig_show,t,y_t2,'b')
            xlabel(handles.sig_show,'s')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            N = length(t);
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
    end
end

guidata(hObject, handles);
```

```
function timer_sub(hObject,eventdata,handles)
%用于实时显示波形
global player
global y1
y2 = y1;
% disp(size(y2));
Fs = player.SampleRate;
cur_second = fix(player.CurrentSample/Fs);
set(handles.play_slider,'Value',cur_second);
cur_minute = fix(cur_second/60);
cur_second_left = cur_second-cur_minute*60;
if(cur_minute < 10)
    tmp1 = strcat('0',num2str(cur_minute));
else
    tmp1 = num2str(cur_minute);
end
if(cur_second_left < 10)
    tmp2 = strcat('0',num2str(cur_second_left));
else
    tmp2 = num2str(cur_second_left);
end
```

```
total_minutes = fix(fix(player.TotalSamples/Fs)/60);
total_seconds_left = fix(player.TotalSamples/Fs)-60*total_minutes;
if(total_minutes < 10)
    tmp3 = strcat('0',num2str(total_minutes));
else
    tmp3 = num2str(total_minutes);
end
if(total_seconds_left < 10)
    tmp4 = strcat('0',num2str(total_seconds_left));
else
    tmp4 = num2str(total_seconds_left);
end
time_show = strcat(tmp1,':',tmp2,'/',tmp3,':',tmp4);
set(handles.time_text,'String',time_show);

if(cur_second >= fix(player.TotalSamples/Fs))
    zero_mat = zeros(1,(cur_second+1)*Fs-length(y2));
%     disp(size(y2))
%     disp(size(zero_mat))
    y2 = horzcat(y2(1,:),zero_mat);
    stop(handles.tmr);
%     delete(handles.tmr);
    handles.flag = 0;
    set(handles.sw_btn,'Value',0);
    set(handles.sw_btn,'string','播放');
    set(handles.filt_sig,'Enable','on');
    set(handles.origin_sig,'Enable','on');
    disp('end')
    return
end
t1 = cur_second+1/Fs:1/Fs:(cur_second+1);
% y2 = y2(:,1);
f1 = y2((cur_second)*Fs+1:(cur_second+1)*Fs);
max_y = abs(max(y1));
% f1 = y1(cur_second*Fs:(cur_second+1)*Fs,1);
plot(handles.sig_live,t1,f1)
% title(handles.sig_live,'实时显示');
axis(handles.sig_live,[-inf inf -max_y(1) max_y(1)])

f1_fft = fft(f1);
N = length(t1);
fm = (0:N-1)*Fs/N-Fs/2;
plot(handles.freq_live,fm*2*pi,fftshift(abs(f1_fft)))
```

```
% --- Executes when selected object is changed in uibuttongroup_filter.
function uibuttongroup_filter_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup_filter
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%
current_selection = get(handles.uibuttongroup_filter,'SelectedObject');
current_tag = get(current_selection,'Tag');
switch current_tag
    case 'IIR_btn'
        set(handles.n_edit,'Visible','off');
        set(handles.text13,'Visible','off');
        set(handles.filter_choose,'Value',1);
        set(handles.text8,'Visible','on');
        set(handles.text9,'Visible','on');
        set(handles.Rp_edit,'Visible','on');
        set(handles.Rs_edit,'Visible','on');
        set(handles.filter_choose,'String',{
            'Butterworth-lowpass'...
            'Butterworth-highpass',...
            'Butterworth-bandpass',...
            'Butterworth-bandstop',...
            'Chebyshev-I-lowpass',...
            'Chebyshev-I-highpass',...
            'Chebyshev-I-bandpass',...
            'Chebyshev-I-bandstop',...
            'Chebyshev-II-lowpass',...
            'Chebyshev-II-highpass',...
            'Chebyshev-II-bandpass',...
            'Chebyshev-II-bandstop'});
    case 'FIR_btn'
        set(handles.fL2_edit,'Visible','off');
        set(handles.fH2_edit,'Visible','off');
        set(handles.text11,'Visible','off');
        set(handles.text12,'Visible','off');
        set(handles.text8,'Visible','off');
        set(handles.text9,'Visible','off');
        set(handles.Rp_edit,'Visible','off');
        set(handles.Rs_edit,'Visible','off');
        set(handles.n_edit,'Visible','on');
        set(handles.text13,'Visible','on');
        set(handles.filter_choose,'Value',1);
        set(handles.filter_choose,'String',{
            'hamming-lowpass',...
```

```
            'hamming-highpass',...
            'hamming-bandpass',...
            'hamming-bandstop',...
            'kaiser-lowpass',...
            'kaiser-highpass',...
            'kaiser-bandpass',...
            'kaiser-bandstop'});

end
```

```
% --- Executes when selected object is changed in uibuttongroup_ctrl.
function uibuttongroup_ctrl_SelectionChangedFcn(hObject, eventdata, handles)
% hObject    handle to the selected object in uibuttongroup_ctrl
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 控制输出原始或滤波波形
global y_copy
global y1
global A
global B
global player
% disp(A)

current_selection = get(handles.uibuttongroup_ctrl,'SelectedObject');
current_tag = get(current_selection,'Tag');
disp(current_tag)
t = 1:1:length(handles.y);
t = t./handles.fs;
if ~isempty(handles.y)
    switch current_tag
        case 'filt_sig'
            disp('filter')
            if isempty(A)
                h = warndlg('请选择滤波器','警告','modal');
                set(handles.origin_sig,'Value',1);
                set(handles.filt_sig,'Value',0);
                return
            end
            y1  = filtfilt(B,A,y1);
            handles.y = filtfilt(B,A,handles.y);
            player = audioplayer(y1,handles.fs);
            left_channel_en = get(handles.left_btn,'Value');
            right_channel_en = get(handles.right_btn,'Value');
```

```
        stereo_channel_en = get(handles.stereo_btn,'Value');

        right_flag = get(handles.right_btn,'Visible') == 'on';
        right_flag = all(right_flag(:)==1);
        if left_channel_en == 1
            y_t = handles.y(:,1);
            y_t = y_t';
            plot(handles.sig_show,t,y_t)
            xlabel(handles.sig_show,'s')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            fft_y = fft_y(:,1);
            N = length(t);
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
        elseif right_channel_en == 1 && right_flag == 1
            y_t = handles.y(:,2);
            y_t = y_t';
            plot(handles.sig_show,t,y_t)
            xlabel(handles.sig_show,'r')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            fft_y = fft_y(:,2);
            N = length(t);
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
        else
            y_t = handles.y(:,1);
            y_t = y_t';
            plot(handles.sig_show,t,y_t,'r')
            hold on
            y_t2 = handles.y(:,2);
            y_t2 = y_t2';
            plot(handles.sig_show,t,y_t2,'b')
            xlabel(handles.sig_show,'s')
            ylabel(handles.sig_show,'amplitude')
            fft_y = fft(handles.y);
            N = length(t);
            dw = (0:N-1)*handles.fs/N-handles.fs/2;
            plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
        end
    case 'origin_sig'
        disp('origin')
        y1 = y_copy;
        handles.y = y_copy;
```

```
            handles.flag1 = 1;
            player = audioplayer(y_copy,handles.fs);
            left_channel_en = get(handles.left_btn,'Value');
            right_channel_en = get(handles.right_btn,'Value');
            stereo_channel_en = get(handles.stereo_btn,'Value');

            right_flag = get(handles.right_btn,'Visible') == 'on';
            right_flag = all(right_flag(:)==1);
            if left_channel_en == 1
                y_t = handles.y(:,1);
                y_t = y_t';
                plot(handles.sig_show,t,y_t)
                xlabel(handles.sig_show,'s')
                ylabel(handles.sig_show,'amplitude')
                fft_y = fft(handles.y);
                fft_y = fft_y(:,1);
                N = length(t);
                dw = (0:N-1)*handles.fs/N-handles.fs/2;
                plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
            elseif right_channel_en == 1 && right_flag == 1
                y_t = handles.y(:,2);
                y_t = y_t';
                plot(handles.sig_show,t,y_t)
                xlabel(handles.sig_show,'r')
                ylabel(handles.sig_show,'amplitude')
                fft_y = fft(handles.y);
                fft_y = fft_y(:,2);
                N = length(t);
                dw = (0:N-1)*handles.fs/N-handles.fs/2;
                plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
            else
                y_t = handles.y(:,1);
                y_t = y_t';
                plot(handles.sig_show,t,y_t,'r')
                hold on
                y_t2 = handles.y(:,2);
                y_t2 = y_t2';
                plot(handles.sig_show,t,y_t2,'b')
                xlabel(handles.sig_show,'s')
                ylabel(handles.sig_show,'amplitude')
                fft_y = fft(handles.y);
                N = length(t);
                dw = (0:N-1)*handles.fs/N-handles.fs/2;
                plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
            end
```

```
%     disp(current_tag)
    end
end

guidata(hObject, handles);
```

```
% --- Executes on button press in mix_btn.
function mix_btn_Callback(hObject, eventdata, handles)
% hObject    handle to mix_btn (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% 混合噪音
global y_copy
SNR = str2double(get(handles.SNR_edit,'String'));
if SNR >0
    t = 1:1:length(handles.y);
    t = t'./handles.fs;
    handles.y = awgn(y_copy,SNR,'measured');
    y_copy = handles.y;
%     sound(handles.y,handles.fs)
    left_channel_en = get(handles.left_btn,'Value');
    right_channel_en = get(handles.right_btn,'Value');
    stereo_channel_en = get(handles.stereo_btn,'Value');

    right_flag = get(handles.right_btn,'Visible') == 'on';
    right_flag = all(right_flag(:)==1);
    if left_channel_en == 1
        y_t = handles.y(:,1);
        y_t = y_t';
        plot(handles.sig_show,t,y_t)
        xlabel(handles.sig_show,'s')
        ylabel(handles.sig_show,'amplitude')
        fft_y = fft(handles.y);
        fft_y = fft_y(:,1);
        N = length(t);
        dw = (0:N-1)*handles.fs/N-handles.fs/2;
        plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
    elseif right_channel_en == 1 && right_flag == 1
        y_t = handles.y(:,2);
        y_t = y_t';
        plot(handles.sig_show,t,y_t)
        xlabel(handles.sig_show,'r')
        ylabel(handles.sig_show,'amplitude')
```

```
        fft_y = fft(handles.y);
        fft_y = fft_y(:,2);
        N = length(t);
        dw = (0:N-1)*handles.fs/N-handles.fs/2;
        plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
    else
        y_t = handles.y(:,1);
        y_t = y_t';
        plot(handles.sig_show,t,y_t,'r')
        hold on
        y_t2 = handles.y(:,2);
        y_t2 = y_t2';
        plot(handles.sig_show,t,y_t2,'b')
        xlabel(handles.sig_show,'s')
        ylabel(handles.sig_show,'amplitude')
        fft_y = fft(handles.y);
        N = length(t);
        dw = (0:N-1)*handles.fs/N-handles.fs/2;
        plot(handles.freq_show,dw*2*pi,fftshift(abs(fft_y)))
    end
end
```

2.滤波前后结果

如下图所示，图 1 为加入噪声前原始信号，图 2 为加入 10dB 噪声后信号，图 3、图 4、图 5、图 6 为分别使用不同通带的巴特沃斯滤波器滤波结果。

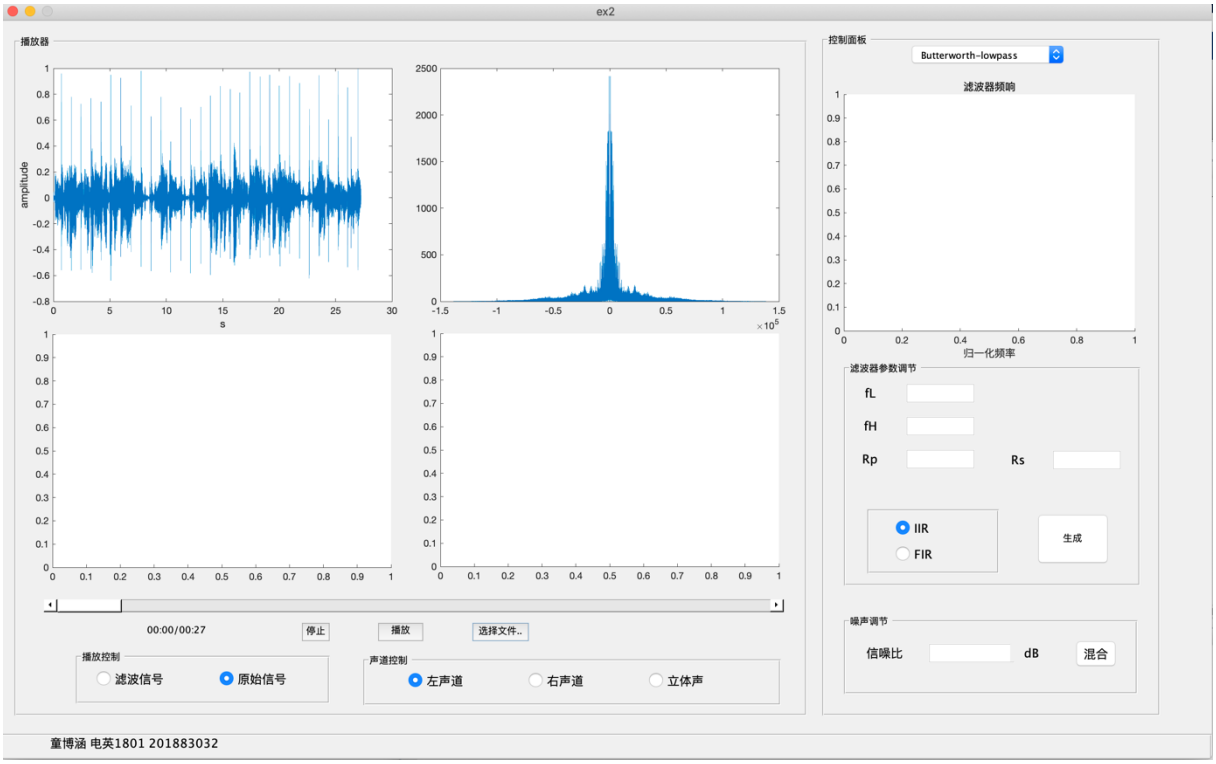

图 1. 原始信号

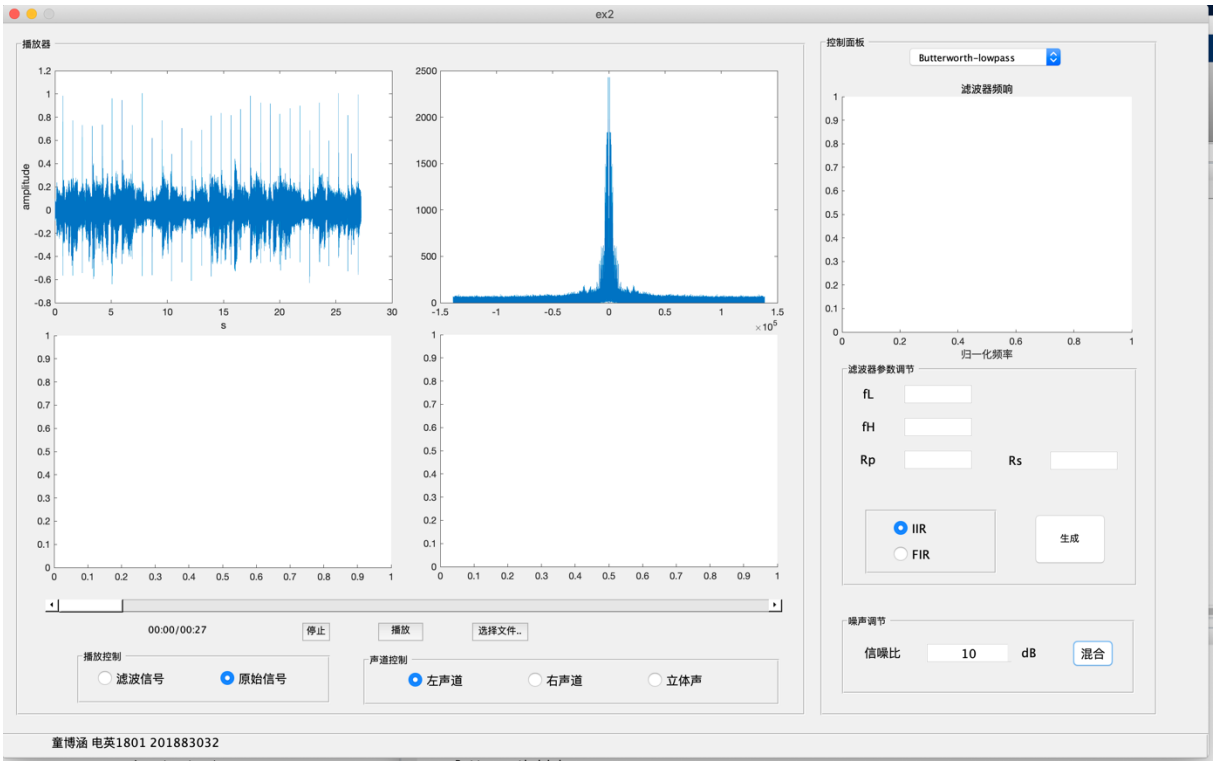

图 2. 添加 10dB 噪声后信号

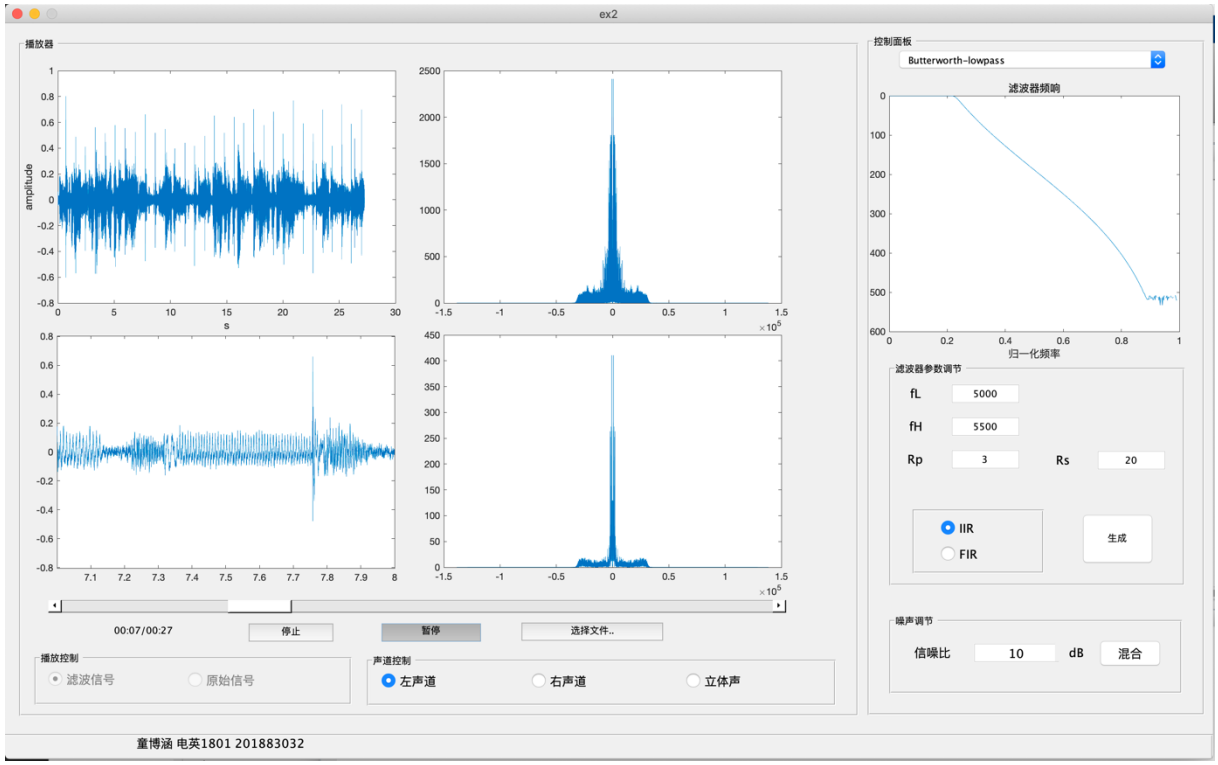

图 3. 低通滤波后结果

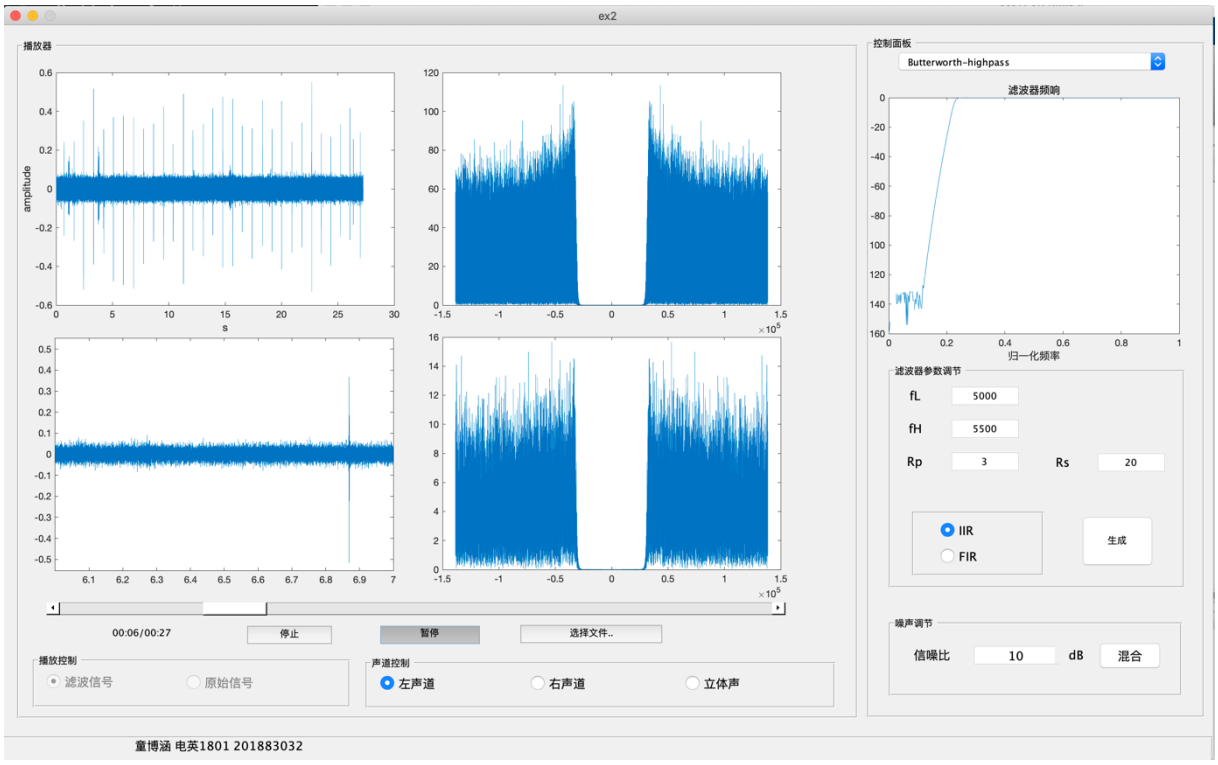

图 4. 高通滤波后结果

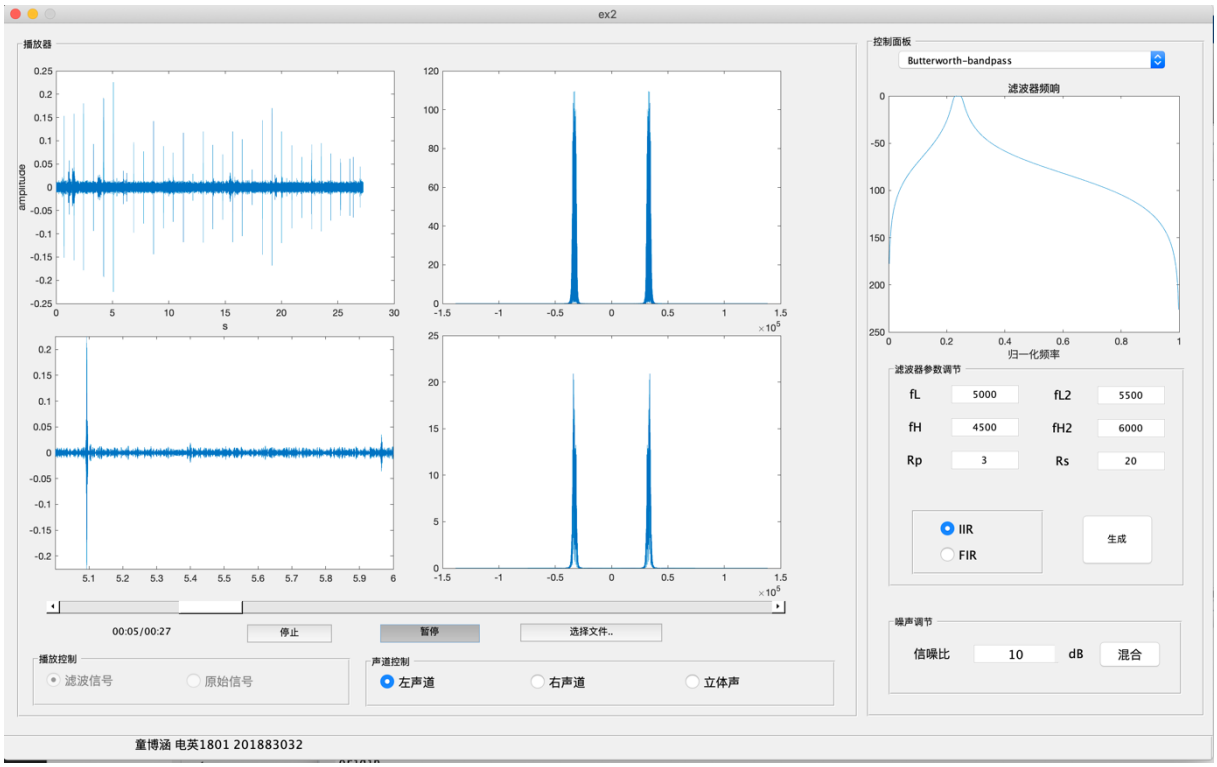

图 5. 带通滤波后结果

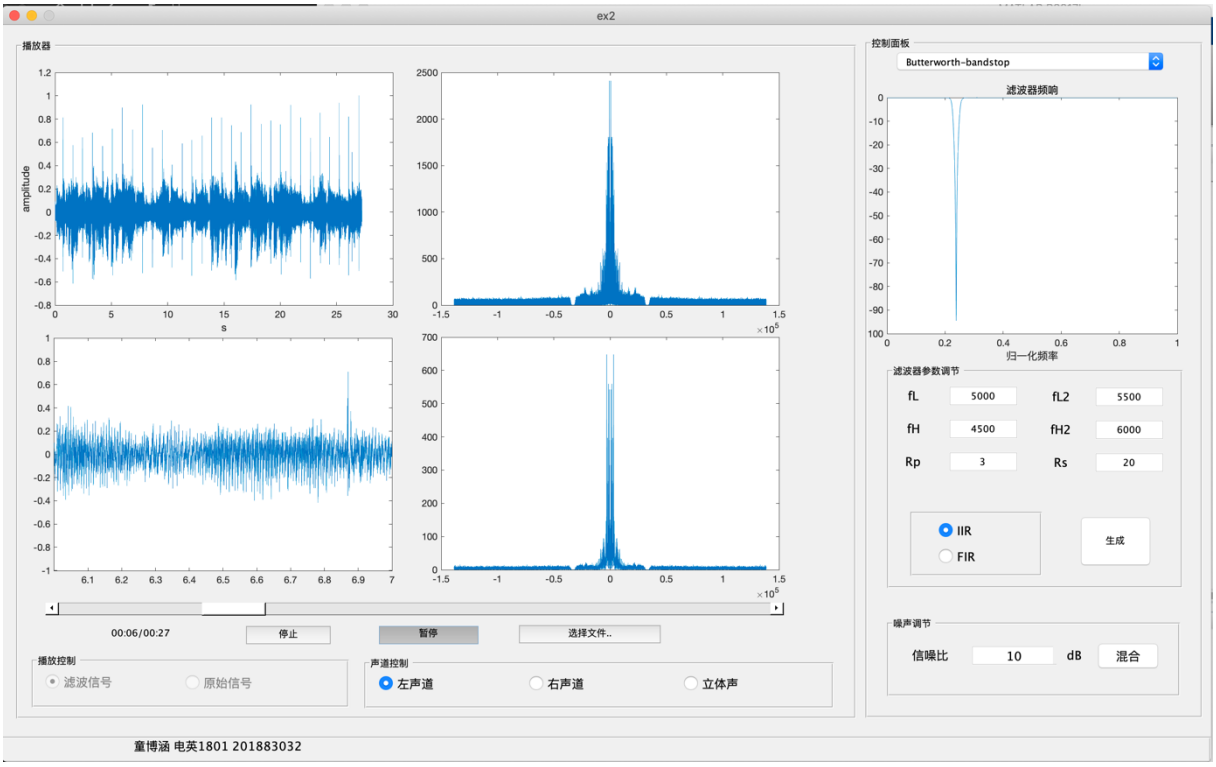

图 6. 带阻滤波后结果

二、实验总结

1. 为了模仿常见的播放器功能，使用 matlab 中的计时器功能，设置计时间隔为 1s，函数 timer_sub 由计时器触发，在界面中实时画出时域和频域波形，并使用滑动条和文本显示播放进度。
2. 除了以巴特沃斯低通滤波器为原型设计 IIR 数字滤波器，进一步使用切比雪夫 I 型和切比雪夫 II 型滤波器设计 IIR 数字滤波器，使用汉明窗和凯泽窗设计了 FIR 数字滤波器。由结果看，IIR 型数字滤波器在处理音频信号中的滤波效果较好。
3. 多个界面控件的设计需要经过多次调试，如更改声道、信号种类、滤波器类型等，需要联合考虑界面中的其他控件是否正常工作。
4. 实验代码略为臃肿，需要进一步优化，多个部分应以模块化方式调用。关于程序中多次调用全局变量的问题，由于在计时器触发的函数 timer_sub 中无法更新句柄变量，若更新会使得调用出错，则在其余函数中使用全局变量保存相关操作后信号结果。