《数字信号处理》实验报告

数字滤波器 的设计实现与使用

学生姓名 赵展

学　　号 U202117282

专业班级 种子2101班

实验平台 Matlab R2023b on Windows

联系方式 15225929727

2023年11月5日

目录

1 实验目的

- 掌握使用FFT进行信号谱分析的方法；
- 设计数字滤波器对指定的语音信号进行滤波处理。

2 实验内容

- 使用Matlab的fft 函数对语音信号进行频谱分析，找出干扰信号的频谱；
- 设计数字滤波器滤除语音信号中的干扰分量，并进行播放对比。

3 实验过程与结果分析

3.1 理想带阻滤波器

对信号源函数进行傅里叶变换和频谱分析，然后设计理想带阻滤波器进行滤波。其中带阻滤波器的脚本函数如下：

```
% band reject filter design
function [afterfilter] = band_reject(audio_data, f, fs)
w = 2 * pi * (f / fs);
A =-2 * cos(w);
h_n = [1, A, 1];
afterfilter = conv(audio_data, h_n);
```

该滤波器令$|He^{j\omega}| = 0$可以求出$A = -2\cos(\omega)$，在时域上卷积计算就可以将特殊的噪音部分的波滤掉。首先将原信号波进行频谱分析，在matlab中输入以下代码，到如图1所示的结果

```
% read original audio wave
clear all;
close all;
[audio_data, fs] = audioread('..\dsp_lab\lab2\SunshineSquare.wav');
time = (length(audio_data)- 1) / fs; % get audio time
n = 0 : length(audio_data)- 1;
tt = 0 : 1 / fs : time; % generate time vector
% freq domain frequency resolution
freq_res = n * fs / length(audio_data);
fft_audio_data = fft(audio_data); % do fft to audio_data
subplot(2,1,1);
plot(tt, audio_data);
title('原始声音信号时域波形');
subplot(2,1,2);
```

```
plot(freq_res, abs(fft_audio_data)); % plot freqency domain data
title('原始声音信号频域波形');
```

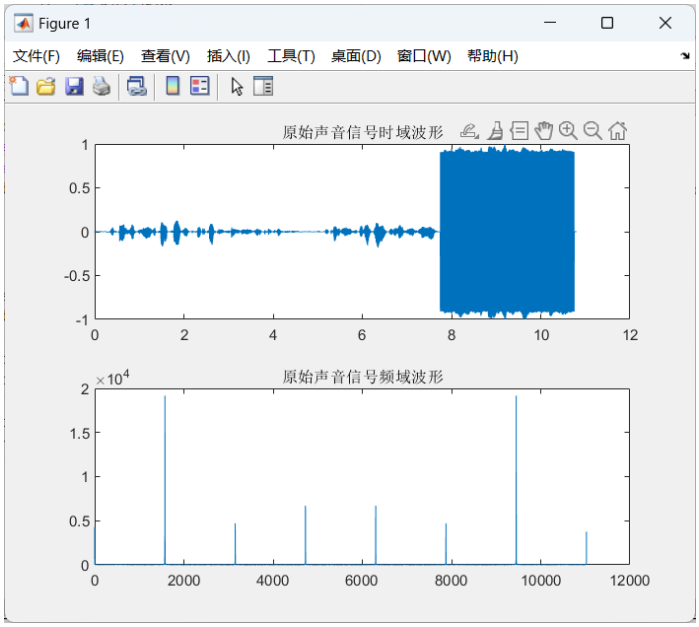

图 1: 原信号时域波形和频谱

然后我先后进行了四次滤波处理，不断滤除噪声。

第一次滤波

第一次滤除0Hz左右的噪声，使用如下代码，得到如图2所示的波形结果。

```
% read original audio wave
clear all;
close all;
[audio_data, fs] = audioread('..\dsp_lab\lab2\SunshineSquare.wav');
time = (length(audio_data) -1)/ fs; % get audio time
n = 0 : length(audio_data)- 1;
tt = 0 : 1 / fs : time; % generate time vector
% freq domain frequency resolution
freq_res = n * fs / length(audio_data);
audio_1 = band_reject(audio_data, 0, fs);
audio_1 = audio_1(1 : end-2); % remove the last two elements.
fft_audio_1 = fft(audio_1);
subplot(2,1,1);
plot(tt, audio_1);
title('滤波一次后声音信号时域波形');
subplot(2,1,2);
```

```
plot(freq_res, abs(fft_audio_1));
title('滤波一次后声音信号频域波形');
```

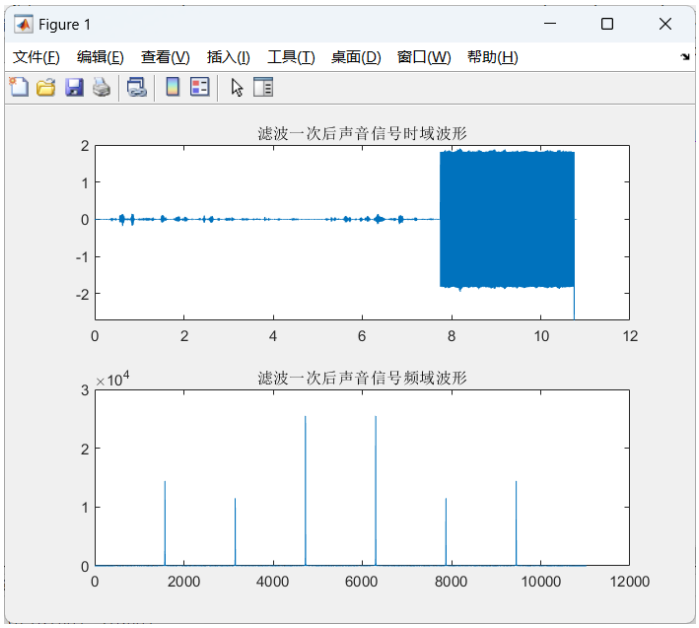

图 2: 第一次滤波后的时域频域信号

第二、三次滤波

第二、三次滤除1574.99Hz、3149.97Hz左右的噪声，使用相似的代码，得到如图3，4所示的波形结果。

第四次滤波

第四次滤除4724.96Hz左右的噪声，使用如下代码，得到如图5所示的波形结果。第四次滤波后我同时将经过四次滤波之后的信号播放出来，可以听到这时已经几乎没有噪声了。

```
% read original audio wave
clear all;
close all;
[audio_data, fs] = audioread('..\dsp_lab\lab2\SunshineSquare.wav');
time = (length(audio_data)- 1) / fs; % get audio time
n = 0 : length(audio_data)- 1;
tt = 0 : 1 / fs : time; % generate time vector
% freq domain frequency resolution
freq_res = n * fs / length(audio_data);
audio_1 = band_reject(audio_data, 0, fs);
audio_1 = audio_1(1 : end-2); % remove the last two elements.
fft_audio_1 = fft(audio_1);
```

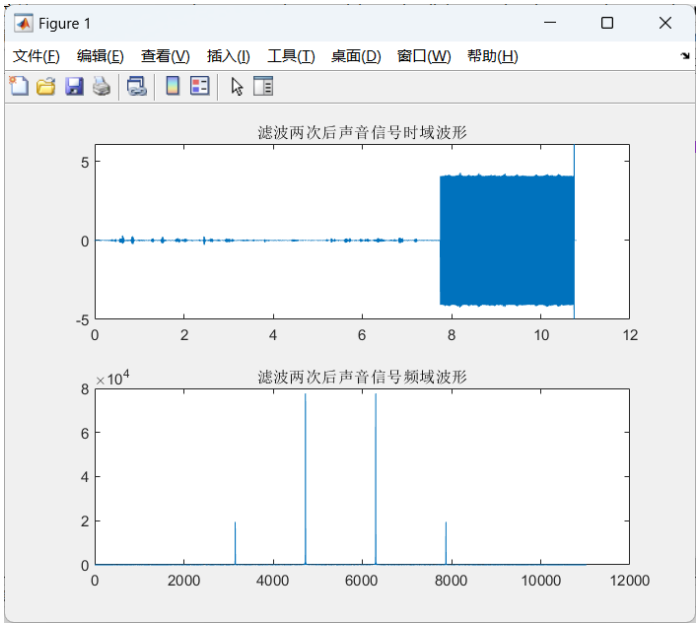

图 3: 第二次滤波后的时域频域信号

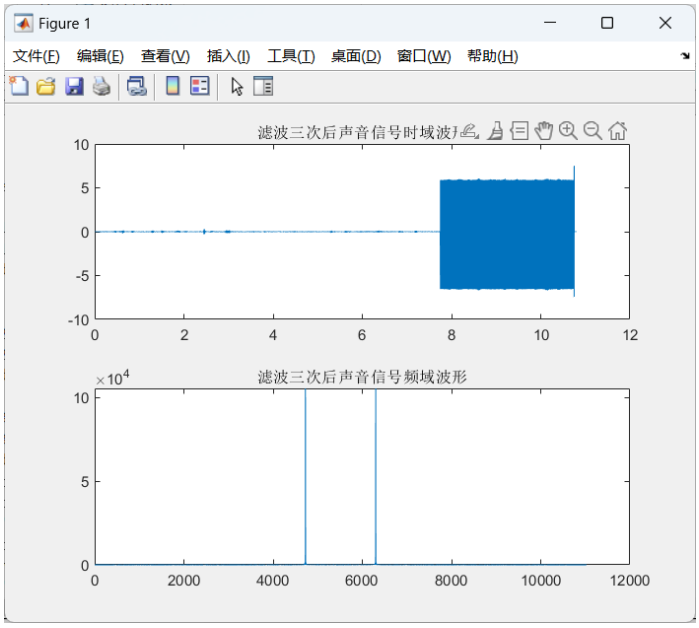

图 4: 第三次滤波后的时域频域信号

```
audio_2 = band_reject(audio_1, 1574.99, fs);
audio_2 = audio_2(1 : end-2); % remove the last two elements.
fft_audio_2 = fft(audio_2);
audio_3 = band_reject(audio_2, 3149.97, fs);
audio_3 = audio_3(1 : end-2); % remove the last two elements.
fft_audio_3 = fft(audio_3);
subplot(2,1,1);
audio_4 = band_reject(audio_3, 4724.96, fs);
audio_4 = audio_4(1 : end-2); % remove the last two elements.
fft_audio_4 = fft(audio_4);
audio_4(abs(audio_4) > 0.9) = 0; % remove the pluse
audio_4 = audio_4 * 2.5; % enhance the volume.
subplot(2,1,1);
plot(tt, audio_4);
title('滤波四次后声音信号时域波形');
subplot(2,1,2);
plot(freq_res, abs(fft_audio_4));
title('滤波四次后声音信号频域波形');
soundsc(audio_4,fs);
```

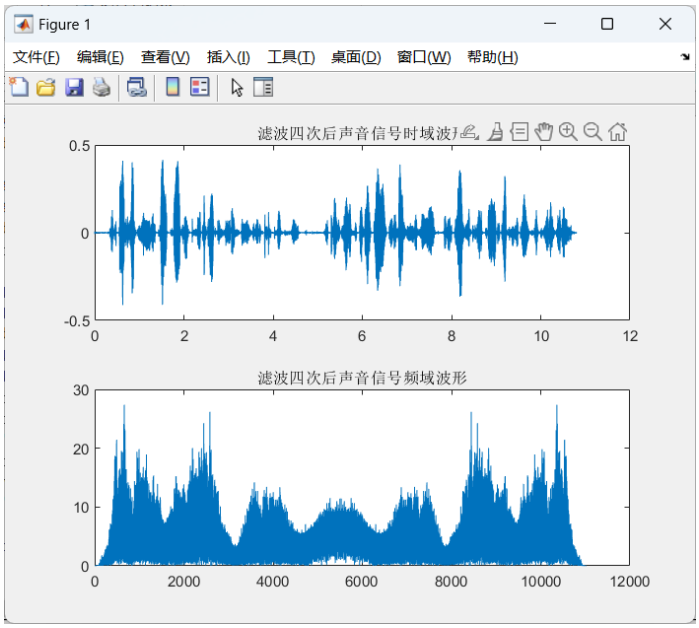

图 5: 第四次滤波后的时域频域信号

最后就可以使用 audiowrite函数生成滤过噪音之后的音频信号保存起来。

3.2 数字巴特沃斯带阻、高通滤波器

同样根据原始信号的频谱制作相应的巴特沃斯带阻滤波器和高通滤波器，使用高通滤波器滤除0Hz附近的噪音，使用带阻滤波器滤除其他频率的噪音。以下是巴特沃斯带阻滤波器代码：

```
% butter band reject filter design
function [afterfilter] = butter_band_reject(audio_data, f, fs)
wp_d=f-50;
wp_s=f+50;
ws_d=f-150;
ws_s=f+150;
wp=[wp_d wp_s]*2/fs;
ws=[ws_d ws_s]*2/fs;
Ap=3;As=18;
[N,wn]=buttord(wp,ws,Ap,As);
butter_fliter=butter(N,wn,'stop');
afterfilter = conv(audio_data,butter_fliter);
```

以下是巴特沃斯高通滤波器代码：

```
function [afterfilter] = butter_high_pass(audio_data, f, fs)
Ap=2;As=15;
F_sh=f+50;
F=2000;
wp=F*2/fs; ws=F_sh*2/fs;
[N,wn]=buttord(wp,ws,Ap,As);
butter_fliter=butter(N,wn,'high');
afterfilter = conv(audio_data,butter_fliter);
```

然后主程序代码如下：

```
% read original audio wave
clear all;
close all;
[audio_data, fs] = audioread('..\dsp_lab\lab2\SunshineSquare.wav');
time = (length(audio_data) -1)/ fs; % get audio time
n = 0 : length(audio_data)- 1;
tt = 0 : 1 / fs : time; % generate time vector
% freq domain frequency resolution
freq_res = n * fs / length(audio_data);
audio_1 = butter_band_reject(audio_data, 1574.99, fs);
audio_1 = audio_1(1 : end-4); % remove the last two elements.
```

```
fft_audio_1 = fft(audio_1);
audio_2 = butter_band_reject(audio_1, 3149.97, fs);
audio_2 = audio_2(1 : end-4); % remove the last two elements.
fft_audio_2 = fft(audio_2);
audio_3 = butter_band_reject(audio_2, 4724.96, fs);
audio_3 = audio_3(1 : end-6); % remove the last two elements.
fft_audio_3 = fft(audio_3);
audio_4 = butter_high_pass(audio_3, 0, fs);
audio_4 = audio_4(1 : end-1); % remove the last two elements.
% audio_4(abs(audio_4) > 3) = 0; % remove the pluse
fft_audio_4 = fft(audio_4);
subplot(2,1,1);
plot(tt, audio_4);
title('滤波四次后声音信号时域波形');
subplot(2,1,2);
plot(freq_res, abs(fft_audio_4));
title('滤波四次后声音信号频域波形');
soundsc(audio_4,fs);
```

最终得到如图6的结果 也可以清晰地听到音频信号的内容。但是很明显使用该滤波器滤掉噪声

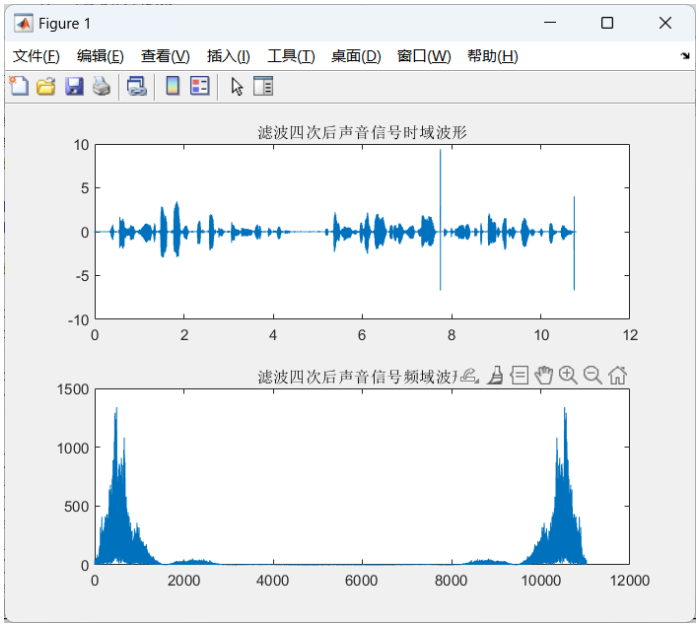

图 6: 第四次滤波后的时域频域信号

的同时也滤掉了部分原信号。

4 实验小结

本次实验，在前置实验的基础上，我用Matlab分析了音频信号的频谱，使用两种方法滤掉了音频信号中的干扰分量，第一次直观的感受到了数字滤波器的作用效果。