

浙江大学



信号与系统 实验报告

实验名称 综合设计性实验—声音的分析、处理

姓名 黄隆铃

学号 3180105524

实验日期 2020 年 5 月 31 日 ~ 6 月 1 日

目录

1 实验原理	1
1.1 声音处理中的数模转换过程	1
1.2 wav 文件	1
1.3 采样率	1
1.4 滤波器	1
2 声音处理一: 基本特征	2
2.1 Matlab 代码	2
2.2 实验结果	3
3 声音处理二: 重采样	3
3.1 Matlab 代码	4
3.2 实验结果	5
4 声音处理三: 滤波	6
4.1 Matlab 代码	6
4.2 实验结果	8

插图

1 原始声音信号的时域和频域特征	3
2 时域特征比较	5
3 频域特征比较	6
4 低通滤波器	9
5 高通滤波器	9
6 滤波结果的时域比较	10
7 滤波结果的频域比较	10

1 实验原理

1.1 声音处理中的数模转换过程

电脑中的声音文件是用数字 0 和 1 来表示的。在电脑上录音的本质就是把模拟声音信号转换成数字信号，而在播放时则是把数字信号还原成模拟声音信号输出。

1.2 wav 文件

Matlab 读取 wav 文件后，可以发现得到一个 $N \times 2$ 的数组，查阅资料得知这个数组里的第一列为左声道，第二列为右声道。经过实验发现确实是这样：带着耳机，经过代码 $x(:,2) = 0$ 之后只有左耳可以听到声音， $x(:,1) = 0$ 之后只有右耳可以听到声音。

1.3 采样率

人耳能够感觉到的最高频率为 20kHz，根据采样定理，要满足人耳的听觉要求，需要以高于最高频率的两倍进行采样才可以完整重建信号。因此要至少每秒进行 40k 次采样，即采样频率（采样率）40kHz。

在当今的主流采集卡上，采样频率一般共分为 22.05KHz、44.1KHz (44100Hz)、48KHz 三个等级，22.05 KHz 只能达到 FM 广播的声音品质，44.1KHz 则是理论上的 CD 音质界限，48KHz 则更加精确一些。对于高于 48KHz 的采样频率人耳已无法辨别出来了，所以在电脑上没有多少使用价值。[\[2\]](#)

1.4 滤波器

在 Matlab 中，可以使用 fir1 函数进行滤波器的设计，然后使用 filter 进行滤波。另外，还可以使用 fir2、firls、firpm 或者 filterDesigner（图形界面）进行多种样式的滤波器设计。[\[3\]](#)

2 声音处理一: 基本特征

用 Matlab 读入了林俊杰的一段音乐, 下面给出读入音乐的采样率、时域波形和频谱特征。

2.1 Matlab 代码

```
1 % 该声音文件的采样率、时域波形和频谱等特征
2 clear;clc;clear sound;
3 [x,fs]=audioread(' ../wavs/林俊杰 - 可惜没如果.wav');
4 x = x(floor(length(x)/16*9):floor(length(x)/16*10),:);
5 % 截取一段高潮部分
6 % sound(x,fs); %播放
7 % x(:,1)=0; 验证左右声道
8 fs % 44100 -- 采样率
9 N=length(x);
10 t=0:N-1;
11 w=2*pi*(t/N-0.5)*fs;
12 X=fftshift(fft(x)*1/fs);
13 subplot(211);
14 plot(t,x,'blue') %时域波形
15 title('原始音乐信号时域波形');
16 xlabel('Time/44100s');
17 ylabel('Amplitude');
18 subplot(212); %频谱特征
19 plot(w/(2*pi),abs(X),'red');
20 title('原始音乐信号频谱特征')
21 xlabel('Frequency/Hz');
22 ylabel('Amplitude');
23 axis([-5000 5000 0 1/2]);
24 saveas(gcf,'origin-audio.png')
```

2.2 实验结果

通过 `audioread` 读取文件后，直接查看 `fs` 的大小，得到采样率为 44100 Hz, 即 CD 音质 44.1KHz。

通过时域和频域分析，可以得到时域波形和频谱特征：

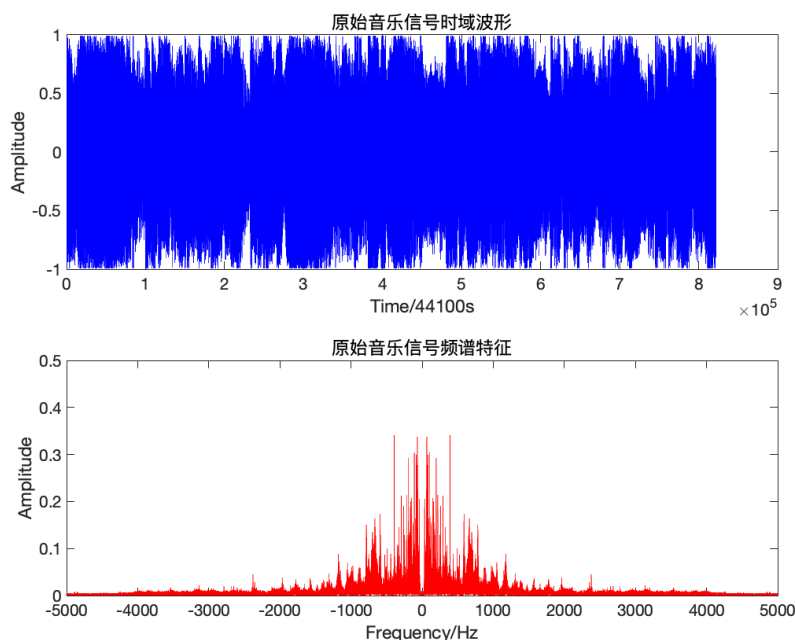


图 1: 原始声音信号的时域和频域特征

3 声音处理二：重采样

下面以较低的采样率 4kHz，对信号进行重采样，并倾听重采样的效果。由于采样率较低，发生了因混叠导致的声音的失真。另外使用 MATLAB 现成的重采样函数 `resample` 进行重采样，发现少了一些奇怪的冒泡似的杂音。根据老师的解释，这是由于 `resample` 函数使用了低通 FIR 滤波器对信号进行反混叠滤波，可避免混叠失真。

3.1 Matlab 代码

```
1 % 重采样
2 clf;clear sound;
3 fs1 = 4000; %重采样率：4KHz
4 x1=x(1:fs/fs1:length(x),:); %手动重采样
5 sound(x1,fs1);
6 x2=resample(x,fs1,fs);% 利用resample函数重采样
7 sound(x2,fs1);
8
9 N1=length(x1);
10 t1=0:N1-1;
11 N2=length(x2);
12 t2=0:N2-1;
13 %时域
14 subplot(211);
15 plot(t1,x1,'blue');
16 xlabel('Time/4000s');
17 ylabel('Amplitude');
18 title('手动重采样');
19 axis([0 8e4 -1.5 1.5]);
20 subplot(212);
21 plot(t2,x2,'red');
22 xlabel('Time/4000s');
23 ylabel('Amplitude');
24 title('resample函数重采样');
25 saveas(gcf,'resample-timezone.png')
26
27 % 频域
28 w1=2*pi*(t1/N1-0.5)*fs1;
29 w2=2*pi*(t2/N2-0.5)*fs1;
30 X1=fftshift(fft(x1)*1/fs1);
31 X2=fftshift(fft(x2)*1/fs1);
32 subplot(211);
33 plot(w1/(2*pi),abs(X1),'blue');
```

```

34 xlabel('Frequency/Hz');
35 ylabel('Amplitude');
36 title('手动重采样');
37 subplot(212);
38 plot(w2/(2*pi),abs(X2),'red');
39 xlabel('Frequency/Hz');
40 ylabel('Amplitude');
41 title('resample函数重采样');
42 saveas(gcf,'resample-frequency.png')

```

3.2 实验结果

首先在直接倾听，听下来有不小的差别，使用 resample 函数进行重采样得到的声音少了许多刺耳杂音。

然后我进行时域和频域分析，分别得到时域和频域特征图像：

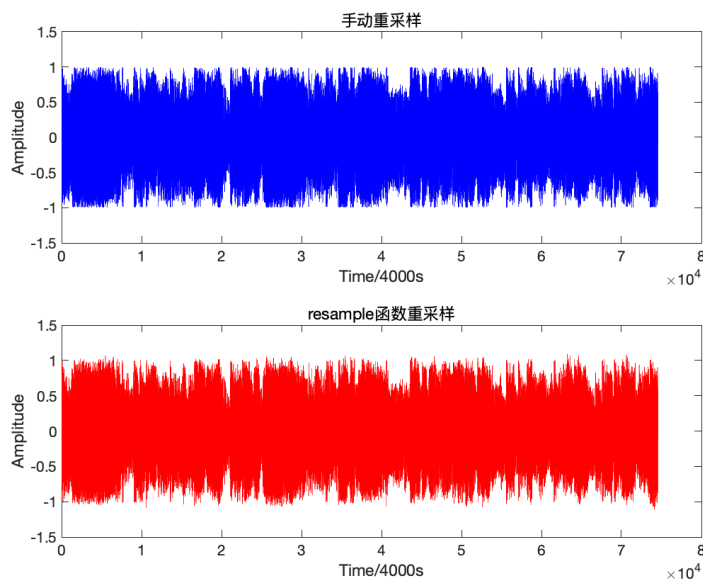


图 2: 时域特征比较

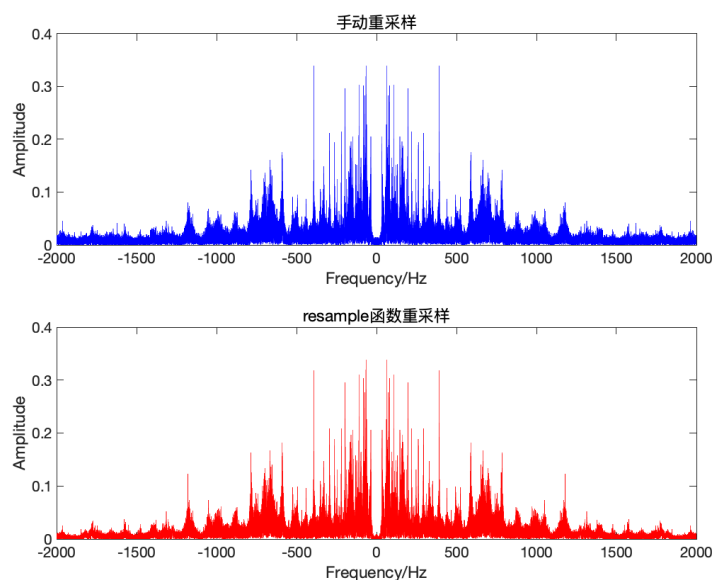


图 3: 频域特征比较

可以看到，手动重采样的结果和 `resample` 函数差不多，时域上看着区别不大，频域上我们可以看到差别：红色的相比蓝色的边缘处振幅小了许多。很容易理解，音乐的频率集中在低频段，那么边缘高频段的就是我听到的杂音，它确实被 `resample` 函数过滤掉修复了。

4 声音处理三：滤波

下面我选取截止频率 $f_c = 3 \text{ KHz}$ ，分别用一个 FIR 低通和高通滤波器对声音信号进行滤波处理。然后倾听滤波后的效果。

4.1 Matlab 代码

```
1 % 滤波
2 clf;clear sound;
3 fc=3000;
4 Wn=fc/(fs/2);% 化为数字频率 f/(fs/2)
5 low_flt = fir1(48,Wn,'low');
```



```

6 high_flt = fir1(48,Wn,'high');
7 figure;
8 freqz(low_flt,1);
9 saveas(gcf,'lowpass-filter.png');
10 figure;
11 freqz(high_flt,1);
12 saveas(gcf,'highpass-filter.png');
13 y1 = filter(low_flt,1,x);
14 sound(y1,fs);
15 y2= filter(high_flt,1,x);
16 sound(y2,fs);
17
18 % 时域
19 figure;
20 subplot(311);
21 plot(t,x,'b');
22 xlabel('Time/44100s');
23 ylabel('Amplitude');
24 title('origin');
25 subplot(312);
26 plot(t,y1,'g');
27 xlabel('Time/44100s');
28 ylabel('Amplitude');
29 title('lowpass');
30 subplot(313);
31 plot(t,y2,'r');
32 xlabel('Time/44100s');
33 ylabel('Amplitude');
34 title('highpass');
35 sgttitle('滤波结果比较 -- 时域');
36 saveas(gcf,'filter-result-timezone.png');
37
38 % 频域
39 Y1=fftshift(fft(y1)*1/fs1);
40 Y2=fftshift(fft(y2)*1/fs1);

```

```
41 figure;
42 subplot(311);
43 plot(w/(2*pi),abs(X),'b');
44 xlabel('Frequency/Hz');
45 ylabel('Amplitude');
46 title('origin');
47 subplot(312);
48 plot(w/(2*pi),abs(Y1),'g');
49 xlabel('Frequency/Hz');
50 ylabel('Amplitude');
51 title('lowpass');
52 subplot(313);
53 plot(w/(2*pi),abs(Y2),'r');
54 xlabel('Frequency/Hz');
55 ylabel('Amplitude');
56 title('highpass');
57 sgtitle('滤波结果比较 -- 频域');
58 saveas(gcf,'filter-result-frequency.png');
```

4.2 实验结果

接下来列出来滤波器的特征和滤波后的音频在时域和频域的比较。

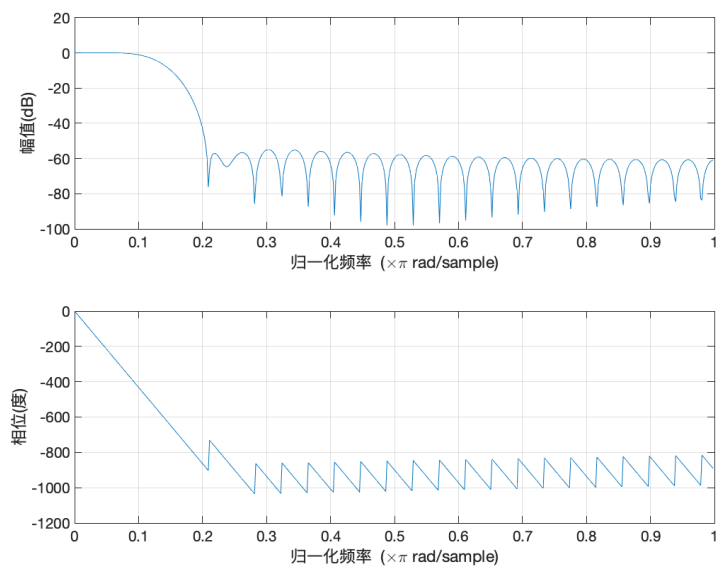


图 4: 低通滤波器

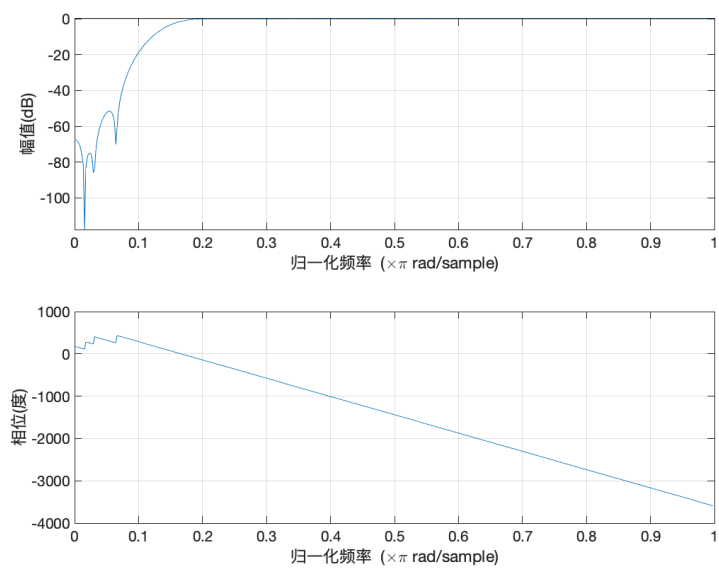


图 5: 高通滤波器

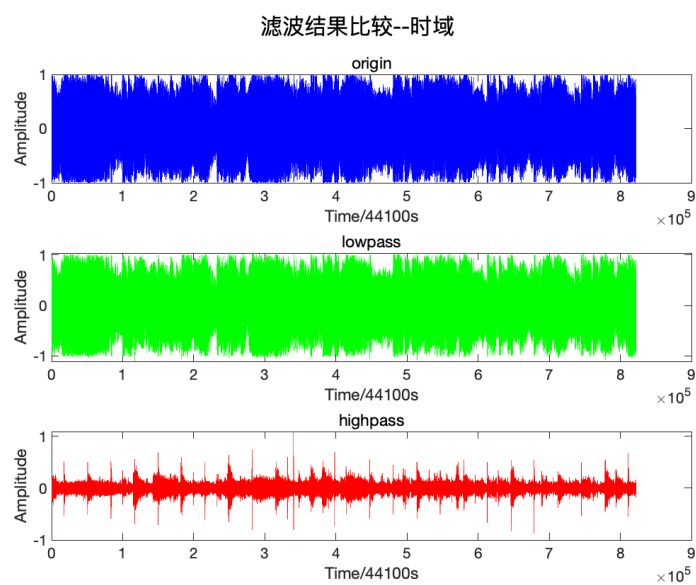


图 6: 滤波结果的时域比较

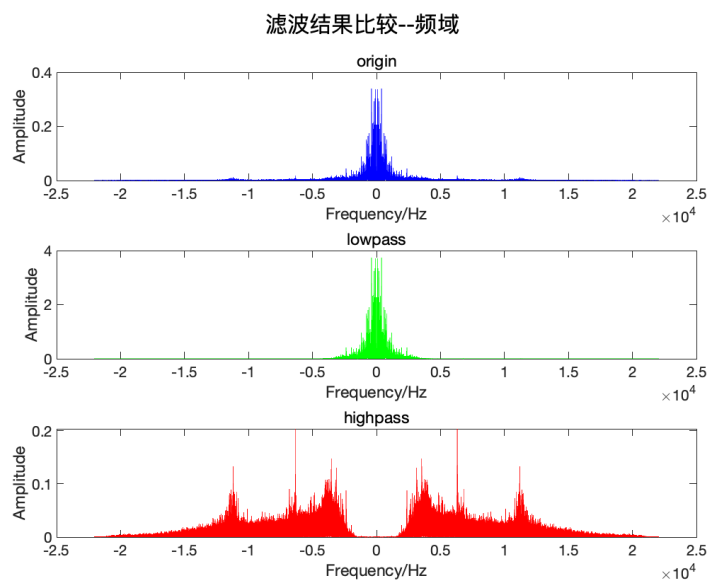


图 7: 滤波结果的频域比较

经过低通滤波器滤波，感觉到脆耳清澈的声音部分少了，留下了相对低沉的声音，但是比较顺畅，仍旧可以听。经过高通滤波器滤波，留下的声音太过刺耳，难以听下去。可见，这美妙的音乐大部分声音集中在低频，但是通过适当的高频声音的叠加，变得更加好听了！

参考网站

- [1] 应用 MatLab 对语音信号进行处理 <https://zhuanlan.zhihu.com/p/57902993>
- [2] 音频基础知识 <https://zhuanlan.zhihu.com/p/30401086>
- [3] 如何快速设计一个 FIR 滤波器 <https://zhuanlan.zhihu.com/p/45520018>