

实验报告：语音信号的频分复用

2018011365 张鹤潇

实验原理

将四段 30 s 音频频谱拼接到一起，保证拼接后频谱的共轭对称性，这就将四段音频调制到了一段音频中；将调制后的信号分解到单独的频段，进行 IFFT，就解调得到原来的四段音频。

实验过程

我收集原始音频的内容是新概念英语自带语音，长度均为 30 s.

实验环境为 Python 3.8.5，直接运行

```
python run.py
```

即可生成结果。

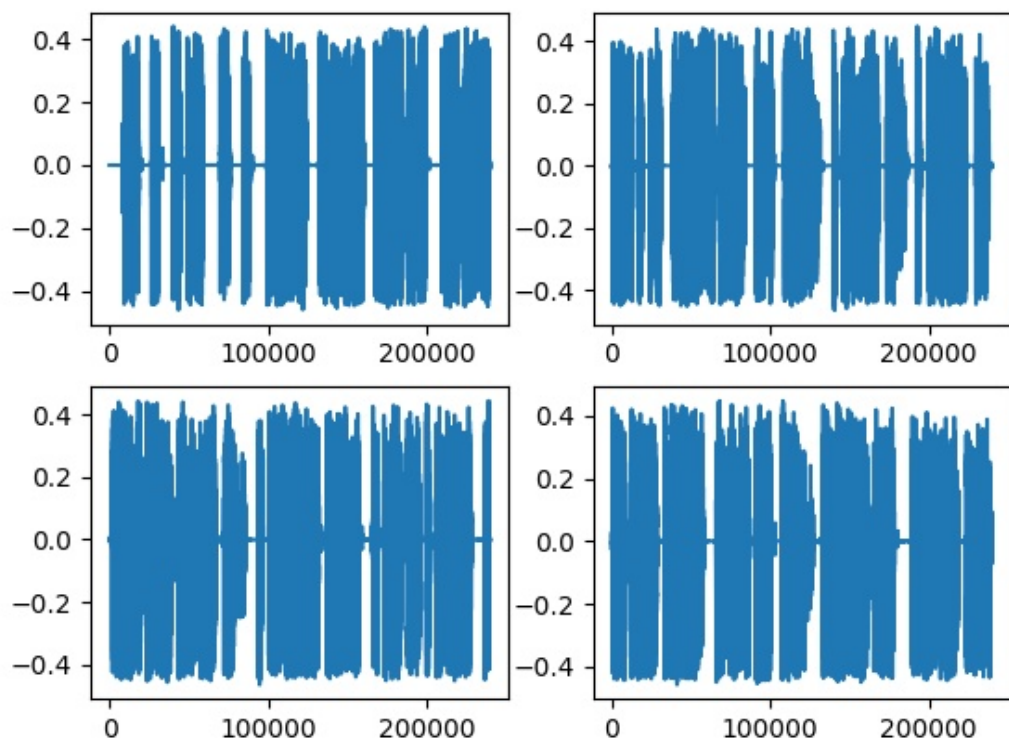
预处理

对读入的音频进行预处理：重采样至 8000 Hz, 滤去 3400 Hz 以上的音频信号.

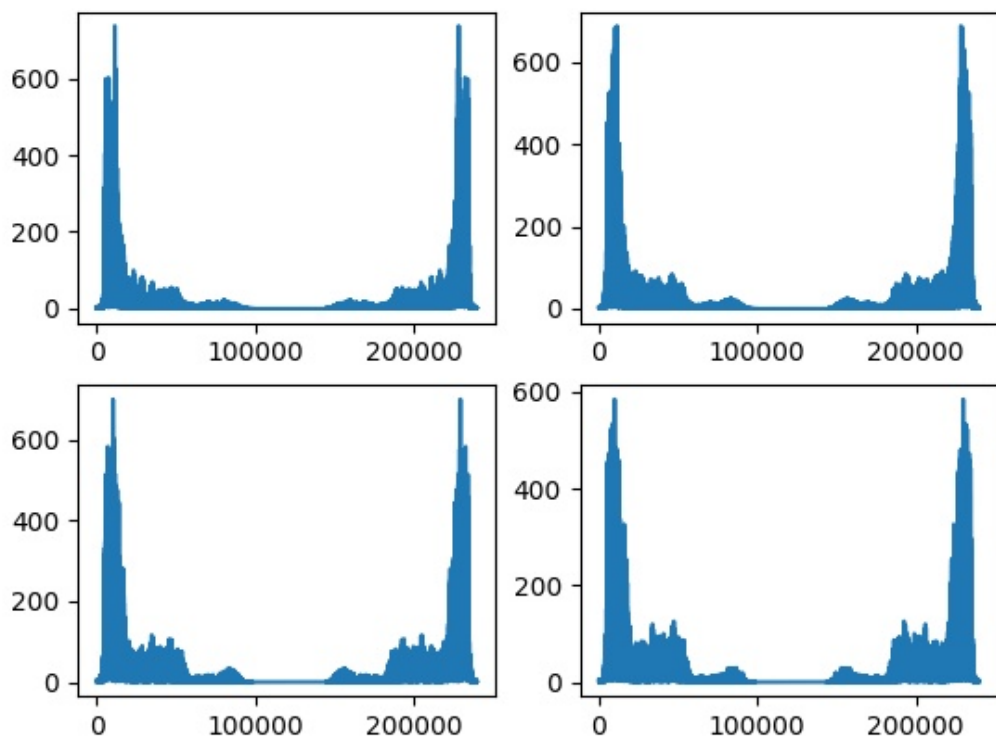
```
# 重采样
y, ori_sr = librosa.load(f'./data/{i}.wav', duration=TIME, sr=SAMPLE_RATE)
# 滤高频
b, a = signal.butter(8, STOP_RATE, fs=ori_sr, btype='lowpass')
y_filted = signal.filtfilt(b, a, y)
```

编解码音频

预处理后的时域信号如图，



频域信号如图,



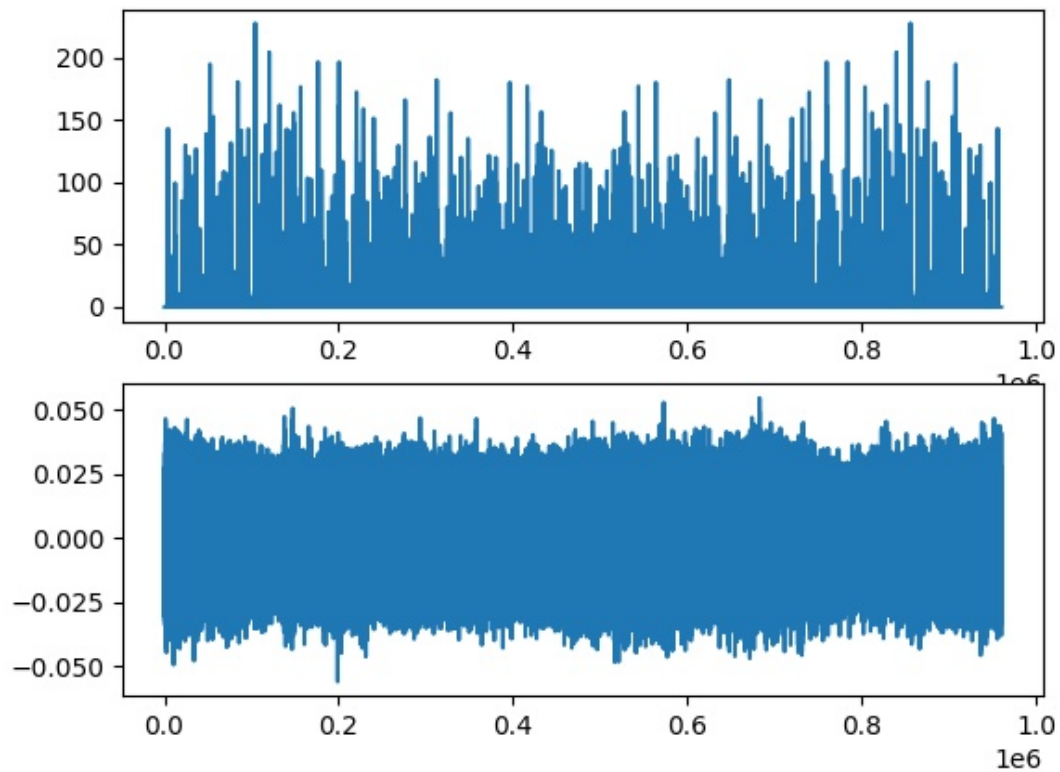
将频谱按左右两段拼接到一起, 完成音频的调制

```

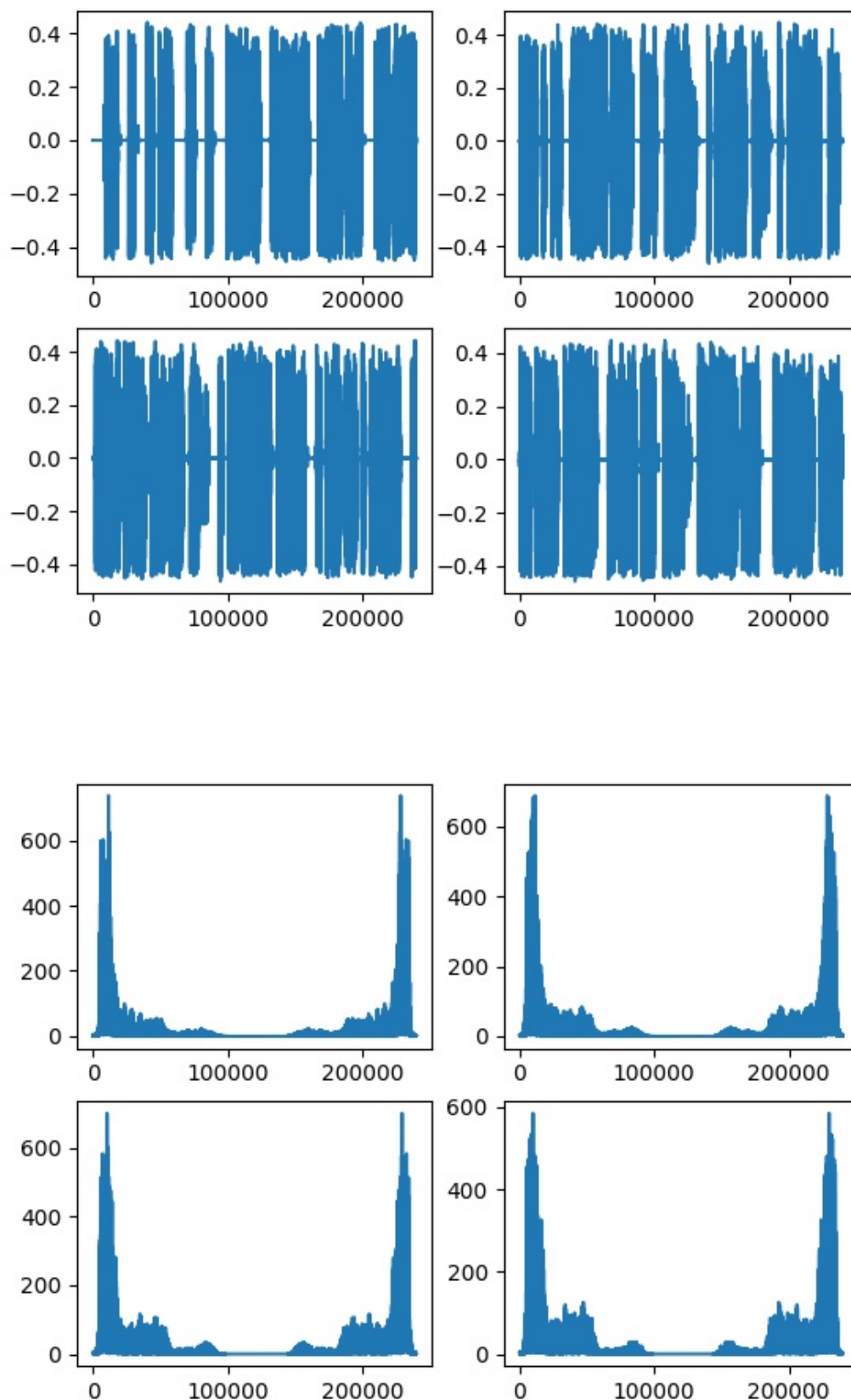
spectrums = [np.fft.fft(audio) for audio in split_audio(audios)]
sample_len = SAMPLE_RATE * SPLIT_TIME // 2
# 0~pi/2
range1 = [sample[:sample_len] for sample in spectrums]
# pi/2~pi
range2 = [sample[-sample_len:] for sample in reversed(spectrums)]
f = np.concatenate(range1 + range2)

```

调制后的时域和频域信号如图，可以看到共轭对称性得到了保持。



将调制后的信号解调到各段音频中，得到的音频时域和频域如图，几乎与原始音频完全一致，声音效果也很好。



MSE 的计算

计算恢复信号和原始信号的 MSE,

```
mse = np.mean(np.sum((np.array(audios) - np.array(processed_audios))**2, 1))
print(f'time segment = {SPLIT_TIME}s, mse = {mse}')
```

分帧频分复用

将 30 s 音频频切为多个等长片段，每个片段长度为 N，分别对每一片段进行频分复用，结果如下，

```
time segment = 1s, mse = 0.0008947254632884388  
time segment = 2s, mse = 0.00020562659101838534  
time segment = 5s, mse = 6.062963884718852e-05  
time segment = 10s, mse = 2.8169981426642077e-05  
time segment = 30s, mse = 2.194722459389537e-05
```