信号处理原理-实验2

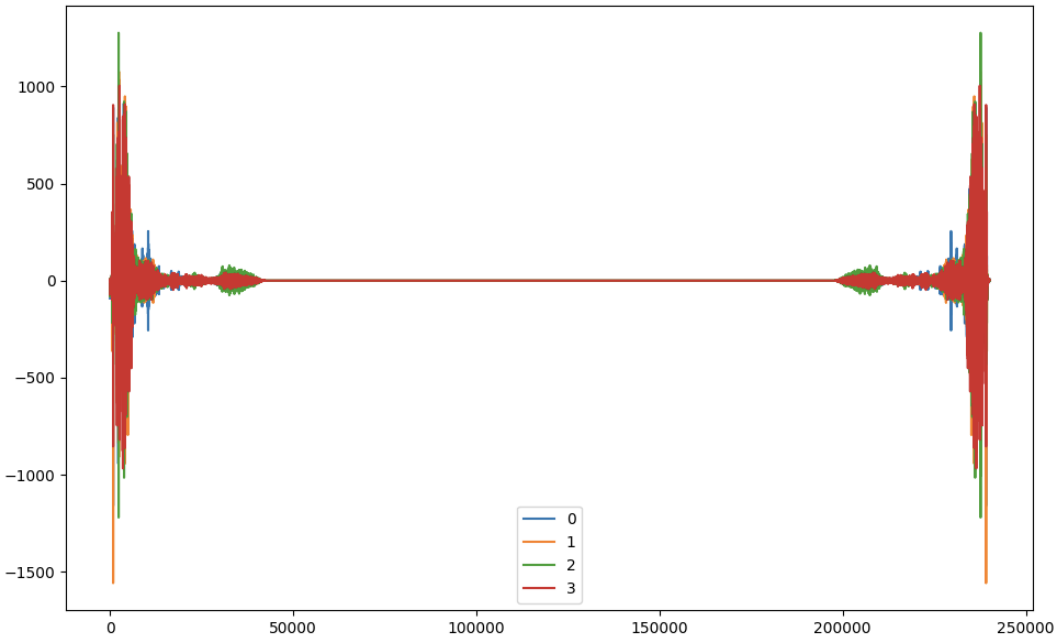
Tags	
Status	
Priority	
Due Date	
Created By	
Date Created	@January 6, 2022 2:35 AM
Finished	☐
Schedule	
Type	
Book	
Course	信号处理原理
Hours Remain	
Semester	2021fall
URL	

信号处理原理-实验2

陈张萌2017013678

先录制一段音频，再对原音频进行预处理，得到采样率为8000Hz的30s音频4段。

对其做FFT得到结果如下：



编码处理

现在希望将其叠加起来传输，而我们需要使得叠加后的信号也是实信号，需要满足频率域共轭对称条件，因此在对频率域拼接时也需要让其仍然共轭对称。因此对于每一段FFT后的结果，将其切成两半后对称拼接起来，如下：

```
for i in range(4):
    pinqilai.extend(fft_result[i][0:int(N*SR/2)])
for i in range(4-1,-1,-1):
    pinqilai.extend(fft_result[i][int(N*SR/2):N*SR])
    pinqilai = np.fft.ifft(pinqilai, 4*N*SR)
```

结果为一段采样率48000Hz，20s的音频

解码处理

读取上面的音频，进行解码，先对其进行fft，再将fft结果分离成4段fft，

```
        result = []
        for i in range(4):
            result.append([])
        for i in range(4):
            result[i].extend(pinqilais[j][i*int(N*SR/2):(i+1)*int(N*SR/2)])
        for i in range(4-1,-1,-1):
            result[i].extend(pinqilais[j][2*N*SR+(4-1-i)*int(N*SR/2):2*N*SR+(4-i)*int(N*SR)])
        # fft
        for i in range(4):
            result[i]=np.fft.ifft(result[i],N*SR)
```

再分别对每段fft结果求其ifft，就得到解码后的音频。

分帧传输

和上面叙述的过程同理，只是在每个步骤的最开始先将一段30s的音频切分成多段即可。

注意事项

使用librosa库读写音频时需要指定采样率