

## 将音频信号转换为频域

---

- import numpy as np
- from scipy.io import wavfile
- import matplotlib.pyplot as plt
- ''' 功能作用：将音频信号转换为频域 '''
- # 读取输入文件
- sampling\_freq, audio = wavfile.read('input\_freq.wav')
- # 标准化数值
- audio = audio / (2.\*\*15)
- # 提取音频数据的长度
- len\_audio = len(audio)
- ''' 傅里叶变换，傅里叶级数 是不同的
- 前者: 主要强调的是 数据无重复
- 后者: 主要强调的是 数据有一定周期和重复性'''
- # 应用傅里叶变换'
- "操作：
- 1.傅里叶变换是关于中心点对称的，
- 2.因此只需要转换信号的前半部分。
- 3.最终目标是提取功率信号，
- 4.因此需要先将信号的值平方'''
- transformed\_signal = np.fft.fft(audio)
- half\_length = np.ceil((len\_audio + 1) / 2.0)
- transformed\_signal = abs(transformed\_signal[0:int(half\_length)])
- transformed\_signal /= float(len\_audio)
- transformed\_signal \*\*= 2
- # 提取转换信号的长度
- len\_ts = len(transformed\_signal)
- # 将部分信号乘以\*2if len\_audio % 2:
- transformed\_signal[1:len\_ts] \*= 2else:
- transformed\_signal[1:len\_ts-1] \*= 2
- # 获取功率信号
- power = 10 \* np.log10(transformed\_signal)
- # 建立时间轴 X 需要根据采样频率对其进行缩放，并将其转换成秒
- x\_values = np.arange(0, half\_length, 1) \* (sampling\_freq / len\_audio) / 1000.0
- # 绘制信号图plt.figure()
- plt.plot(x\_values, power, color='black')
- plt.xlabel('Freq (in kHz)')
- plt.ylabel('Power (in dB)')
- plt.show()

---

幕布 - 思维概要整理工具

---