## **概念**

### **傅立叶变换**

**Fourier Transform**，一种线性积分变换，用于信号在时域（或空域）和频域之间的变换。

时域：即

### **离散傅立叶变换**

**DFT，Discrete Fourier Transform**，是傅立叶变换在时域和频域上都呈离散的形式，将信号的时域采样变换为其DTFT（离散时间傅立叶变换）的频域采样。

### **快速傅立叶变换**

**Fast Fourier Transform，FFT**，是快速计算离散傅立叶变换（DFT）的方法

### **基波和谐波**

在复杂的周期性震荡中，包含**基波**和**谐波**，和该振荡最长周期相等的正弦波分量称为基波。相应于这个周期的频率称为**[基波频率](https://baike.baidu.com/item/%E5%9F%BA%E6%B3%A2%E9%A2%91%E7%8E%87/10790188" \t "/home/jimzeus/x/_blank)**。频率等于基波频率的整倍数的正弦波分量称为谐波。

### **滤波器**

**filter**，过滤特定频率信号的器件/装置/程序/函数，根据其作用大致分为

* **低通滤波器**：允许低频（低于某个频率）信号通过
* **高通滤波器**：允许高频（高于某个频率）信号通过
* **带通滤波器**：允许某个频带（bandwidth，某两个频率之间）信号通过，通常是高通滤波器和低通滤波器的级联
* **带阻滤波器**：减弱某个频带（bandwidth，某两个频率之间）信号通过，通常是高通滤波器和低通滤波器的级联

### **FIR滤波器**

**Finite Impulse Response Filter，有限脉冲响应滤波器**，数字滤波器的一种。这类滤波器对于脉冲输入信号的响应最终趋向于0，因此是有限的，从而得名。

FIR滤波器是一线性系统，输入信号x(0), x(1), ... , x(n)，经过该系统后的输出信号y(n)可表示为：

IMG_256

其中h0, h1, ..., hN是滤波器的脉冲相应，通常称为**滤波器的系数**，N是**滤波器的阶数**。

### **IIR滤波器**

**Infinite Impulse Response Filter，无限脉冲响应滤波器**，数字滤波器的一种。

## **接口和程序**

ffmpeg：

* wave
  + open()：打开wave文件，返回Wave\_read或者Wave\_write对象
  + Wave\_read：读方式打开的wave文件对象
    - getparams()：返回nchannels（通道数，单声道、多声道），sampwidth（数据宽度，8bit或者16bit），framerate（44.1K、48K），nframes（总数据帧数），comptype（压缩方式），compname。
  + Wave\_write：写方式打开的wave文件对象
* scipy
  + signal：
    - lfilter()：用FIR或者IIR过滤器滤波
      * b：FIR或者IIR滤波器的系数（即firwin函数的返回值）
      * a：IIR滤波器的系数
      * x：输入的信号序列
    - firwin()：根据参数生成FIR滤波器的系数（coefficients）
      * numtaps：FIR滤波器系数的长度
      * cutoff：截断频率
      * fs：信号的采样频率
    - remez()：用Remez算法生成滤波器
    - freqz()：
  + fft：参见numpy.fft
  + fftpack
    - helper.fftfreq()：返回离散的频率分布，等于numpy.fft.helper.fftfreq()
      * n：返回的频率数组的长度
      * d：采样间隔（单位为秒）
  + io
    - wavfile
      * read()：读取wave文件，参数：路径，返回值：采样频率和数据
      * write()：写入wave文件
* numpy：
  + core.function\_base.linspace()：
  + core.fromnumeric.cumsum()：累加函数
  + lib.function\_base.insert()：在ndarray中插入值
  + fft：快速傅立叶变换相关包
    - fftpack.fft()：快速傅立叶变换，返回频域的序列，长度与输入相同（即每个元素代表频率/总长度的
      * a：输入的时域数据序列，其长度等于频率×时间
    - fftpack.ifft()：
    - helper.fftfreq()：
* pyaudio：PortAudio的python bindings，声卡操作，实时的声音输入输出
  + PyAudio
    - open()：返回一个pyaudio.Stream

参考：

声音的输入输出：<http://bigsec.net/b52/scipydoc/wave_pyaudio.html>

数字信号系统：<http://bigsec.net/b52/scipydoc/filters.html>

FFT学习：<http://bigsec.net/b52/scipydoc/fft_study.html>