****

数字信号处理实验报告

专业： 电子信息工程

班别： 电信专升本201 班

姓名： 陈黎军

学号： 3200442029

**浙大宁波理工学院实验报告**

**姓名 陈黎军 学号 3200442029 学院** 信息科学与工程学院

**专业 电子信息工程 班级 电信专升本201 课程** 数字信号处理

**实验时间 2021-5-26 实验地点 指导教师**

**实验二**

**实验目的：**

熟悉一维声音信号

**实验内容：**

1、下载好声音样本

2、截取长约半秒的高音稳定片段，绘制频谱。

3、过滤部分谐音后，再播放于原声音对比

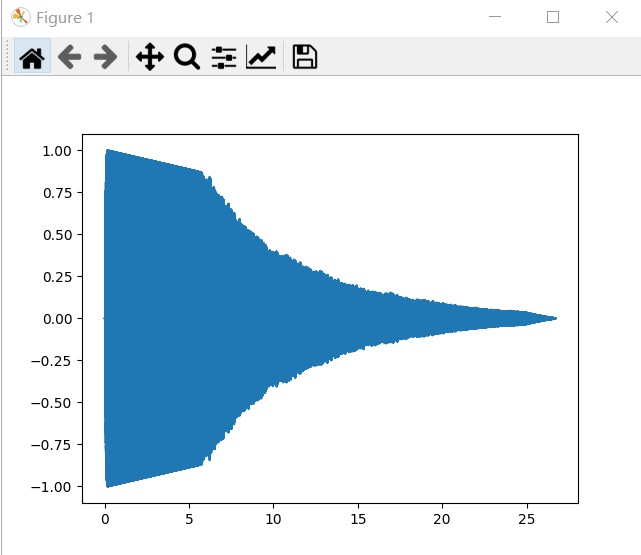
4、用sinsignal创建的波，试听

5、加入与基频不是倍数关系的频率元素后，试听对比

6、创建一个名为stretch的函数来对波形进行修改，使变快或变慢

**实验结果：**

1. 样本频谱



2、截取片段后频谱

import os

import matplotlib.pyplot as plt

import thinkdsp as dsp

from thinkdsp import read\_wave

from thinkdsp import decorate

from IPython.display import display

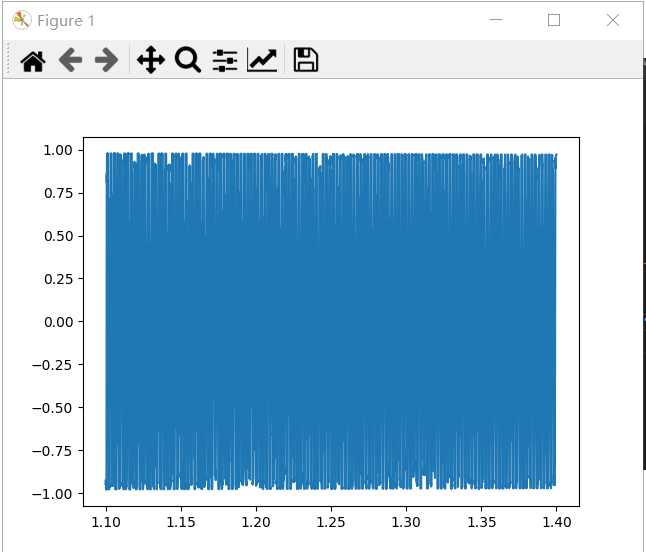
from ipywidgets import interact, fixed

wave = read\_wave('C:/Users/38407/Desktop/2/数字信号处理/python/3python练习第一章/18871\_\_zippi1\_\_sound-bell-440hz.wav')

wave.normalize()

wave.make\_audio()

plt.show()



3、过滤掉部分谐音后

import os

import matplotlib.pyplot as plt

import thinkdsp as dsp

from thinkdsp import read\_wave

from thinkdsp import decorate

from IPython.display import display

from ipywidgets import interact, fixed

wave = read\_wave('C:/Users/38407/Desktop/2/数字信号处理/python/3python练习第一章/18871\_\_zippi1\_\_sound-bell-440hz.wav')

wave.normalize()

wave.make\_audio()

#wave.plot()

segment = wave.segment(start=1.1, duration=0.3)

segment.make\_audio()

#segment.plot()

#segment.segment(start=1.1, duration=0.005).plot()

spectrum = segment.make\_spectrum()

#spectrum.plot(high=1000)

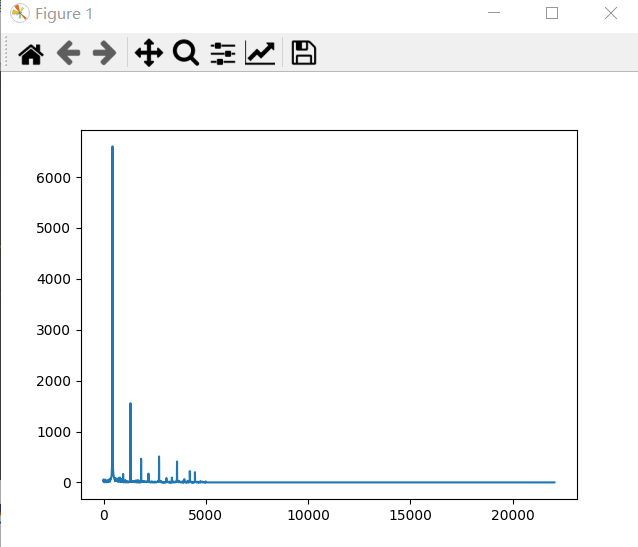
spectrum.low\_pass(5000,0.1)

spectrum.make\_wave().make\_audio()

spectrum.plot()

wave.write(filename='output1-2.wav')

plt.show()



（过滤掉后，声音明显变低沉，谐音频率越高，音色越亮）

4、用sinsignal创建的波

import os

import matplotlib.pyplot as plt

from thinkdsp import read\_wave

from thinkdsp import decorate

from IPython.display import display

from ipywidgets import interact, fixed

from thinkdsp import SinSignal

wave2 = read\_wave('C:/Users/38407/Desktop/2/数字信号处理/python/3python练习第一章/18871\_\_zippi1\_\_sound-bell-440hz.wav')

signal = (SinSignal(freq=400, amp=1.0) +

          SinSignal(freq=600, amp=0.5) +

          SinSignal(freq=800, amp=0.25))

signal.plot()

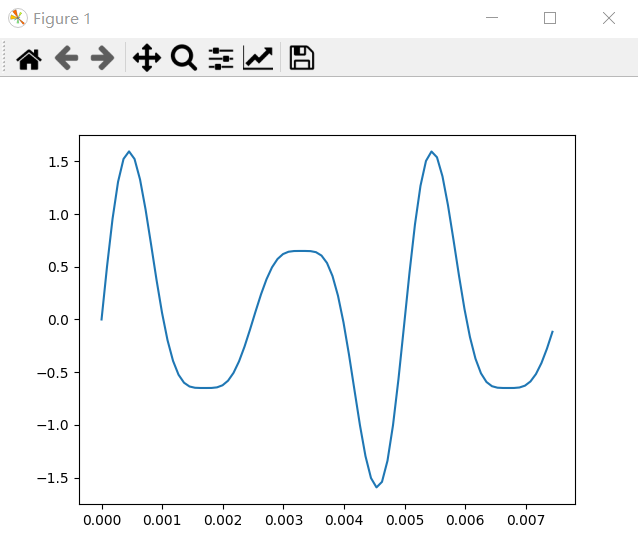
wave2 = signal.make\_wave(duration=1)

wave2.apodize()

wave2.make\_audio()

wave2.write(filename='output1-3.wav')

plt.show()

声音低沉并且一致

5、加入与基频不是倍数关系的频率元素后

import os

import matplotlib.pyplot as plt

from thinkdsp import read\_wave

from thinkdsp import decorate

from IPython.display import display

from ipywidgets import interact, fixed

from thinkdsp import SinSignal

wave2 = read\_wave('C:/Users/38407/Desktop/2/数字信号处理/python/3python练习第一章/18871\_\_zippi1\_\_sound-bell-440hz.wav')

signal = (SinSignal(freq=400, amp=1.0) +

          SinSignal(freq=600, amp=0.5) +

          SinSignal(freq=800, amp=0.25))

#signal.plot()

wave2 = signal.make\_wave(duration=1)

wave2.apodize()

wave2.make\_audio()

#spectrum = wave2.make\_spectrum()

#spectrum.plot(high=2000)

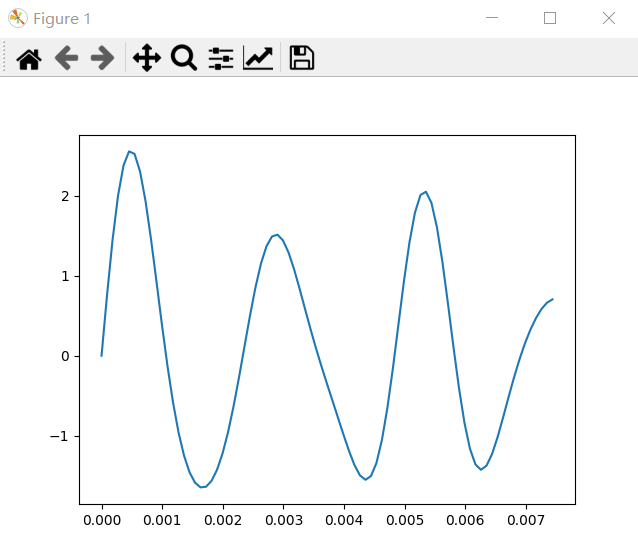
signal += SinSignal(freq=450)

signal.make\_wave().make\_audio()

signal.plot()

wave2.write(filename='output1-3.wav')

plt.show()



音调有一点点区别

6、创建一个名为stretch的函数来对波形进行修改，使变快或变慢

import os

import matplotlib.pyplot as plt

from thinkdsp import read\_wave

from thinkdsp import decorate

from IPython.display import display

from ipywidgets import interact, fixed

from thinkdsp import SinSignal

wave3 = read\_wave('C:/Users/38407/Desktop/2/数字信号处理/python/3python练习第一章/18871\_\_zippi1\_\_sound-bell-440hz.wav')

wave3.normalize()

wave3.make\_audio()

def stretch(wave, factor):

    wave.ts \*= factor

    wave.framerate /= factor

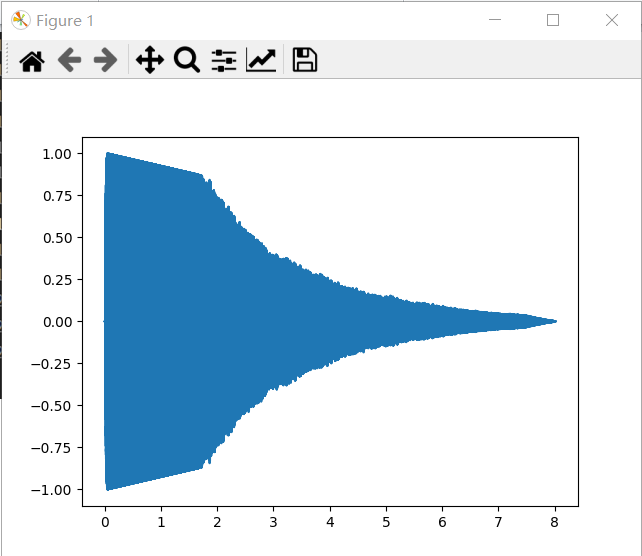
stretch(wave3, 0.3)

wave3.make\_audio()

wave3.plot()

wave3.write(filename='output1-4.wav')

plt.show()



速度变快