**浙大宁波理工学院实验报告**

**姓名 王瑞泓 学号 3200442031 学院 信息科学与工程学院**

**专业 电子信息 班级 电信专升本201班 课程 数字信号处理**

**实验时间 2021-06-09 实验地点 SL 405**   **指导教师 王一刚**

**实验四**

**实验目的：**

1. 了解锯齿波、方波、三角波的频谱区别
2. 了解周期信号及其谐波，采样与欠采样

**实验内容：**

**利用python及thinkdsp库**

**实验结果：**

from thinkdsp import Sinusoid

from thinkdsp import normalize, unbias

import numpy as np

import matplotlib.pyplot as plt

from thinkdsp import SquareSignal

from thinkdsp import TriangleSignal

class SawtoothSignal(Sinusoid):

    '''锯齿波的类'''

    def evaluate(self, ts):

        #cycle是自起始时间的周期数

        cycles = self.freq \* ts + self.offset / np.pi / 2

        frac, \_ = np.modf(cycles)

        #np.modf()能将一个数分为小数部分和整数部分

        ys = normalize(unbias(frac), self.amp)

        #frac的序列介于0到1

        #unbias将波形下移使其中心位于0

        #normalize将波形归一化到给定的振幅（amp）

        return ys

plt.rcParams['font.sans-serif']=['SimHei'] #加上这一句就能在图表中显示中文

plt.rcParams['axes.unicode\_minus']=False #用来正常显示负号

plt.subplot(511)#产生5行1列的第一个图

plt.title(u"锯齿波")

sawtooth = SawtoothSignal().make\_wave(duration=0.5, framerate=40000)

#制作波形长度为0.5s，每秒40000帧

sawtooth.make\_spectrum().plot()

plt.subplot(513)

plt.title(u"锯齿波（灰）&方波")

sawtooth.make\_spectrum().plot(color='gray')#将原锯齿波显示为灰色便于观察

square = SquareSignal(amp=0.5).make\_wave(duration=0.5, framerate=40000)

#将方波的振幅调整为0.5便于比较

square.make\_spectrum().plot()

plt.subplot(515)

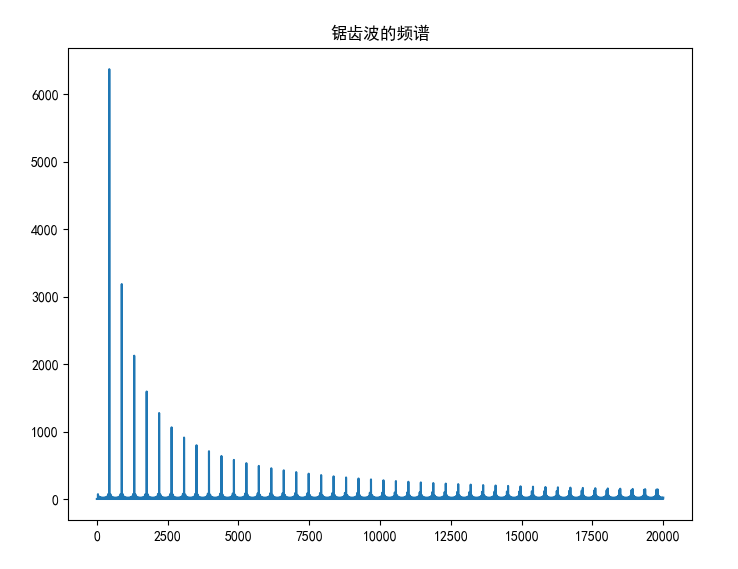
plt.title(u"锯齿波（灰）&三角波")

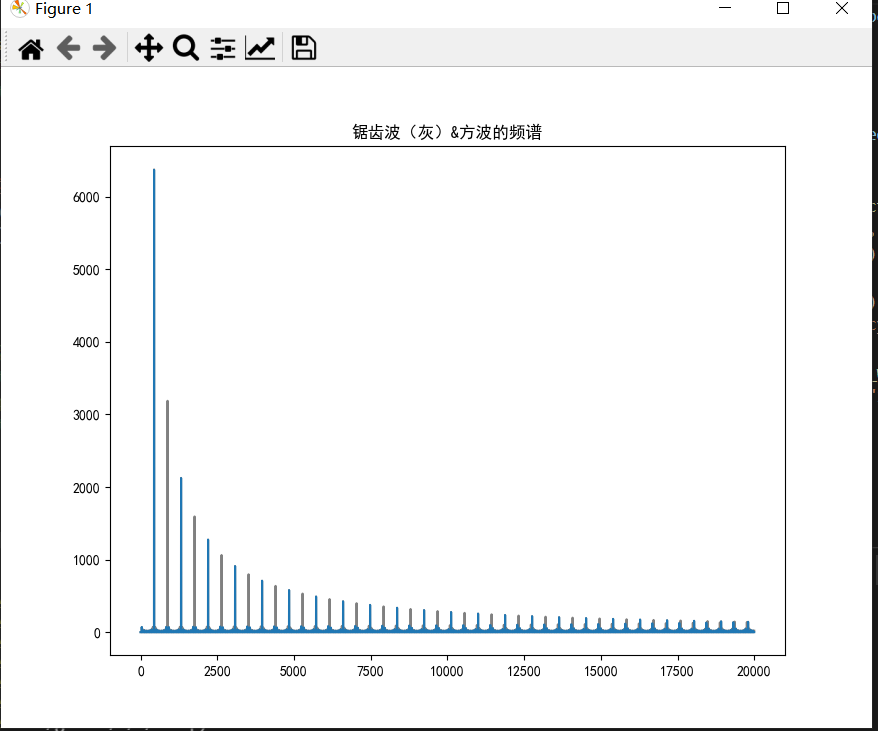
sawtooth.make\_spectrum().plot(color='gray')

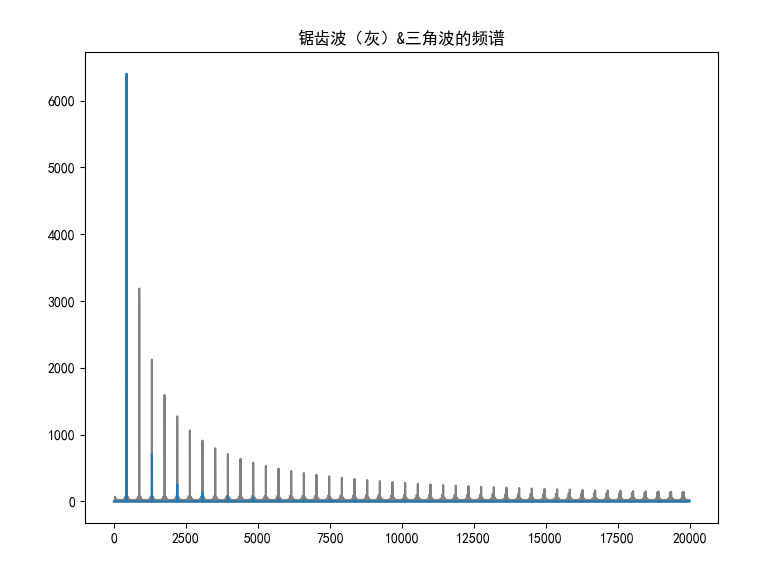
triangle = TriangleSignal(amp=0.79).make\_wave(duration=0.5, framerate=40000)

triangle.make\_spectrum().plot()

plt.show()







方波和三角波没有偶数次谐波和奇数次谐波。与锯齿波相比，三角波衰减的更快（指数衰减1/f²）。

2-3

import matplotlib.pyplot as plt

from thinkdsp import SquareSignal

from thinkdsp import decorate

square = SquareSignal(1100).make\_wave(duration=0.5, framerate=10000)

#创建一个频率为1100HZ方波,然后对其采样长度为0.5s，采样帧率为10000帧每秒

wave2=square.make\_spectrum()

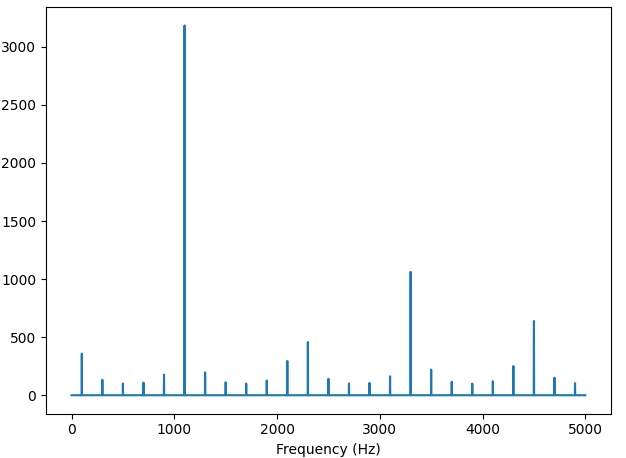
#filtered = wave.make\_wave()

#filtered.write(filename="2.5.wav")

wave2.plot()

decorate(xlabel='Frequency (Hz)')

plt.show()



基波在1100赫兹，3次谐波在3300赫兹，但5次谐波，应该在5500赫兹，却在4500赫兹。第7次谐波，应该在7700赫兹，却在2300赫兹。

2-4

import matplotlib.pyplot as plt

from thinkdsp import TriangleSignal

from thinkdsp import decorate

triangle = TriangleSignal().make\_wave(duration=0.01)

#创建一个时长为0.01s的三角波

#triangle.plot()

#decorate(xlabel='Time (s)')

spectrum = triangle.make\_spectrum()

print('hs[0]=',spectrum.hs[0])

spectrum.hs[0] = 100

triangle.plot(color='gray')

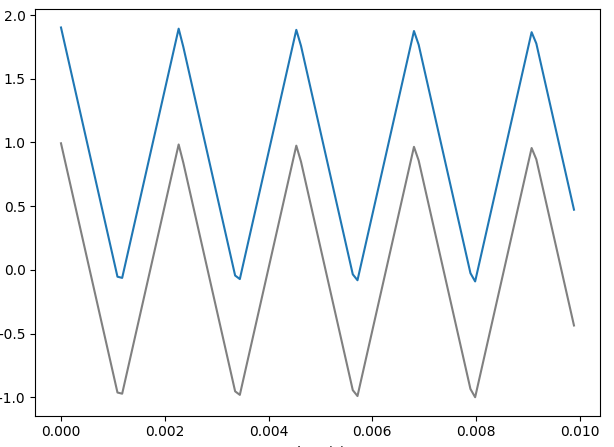
#在图上吧原三角波变为灰色

spectrum.make\_wave().plot()

decorate(xlabel='Time (s)')

plt.show()





相当于在交流量上叠加一个直流偏移

2-5

import matplotlib.pyplot as plt

from thinkdsp import TriangleSignal

from thinkdsp import decorate

def filter\_spectrum(spectrum):

    # avoid division by 0

    spectrum.hs[1:] /= spectrum.fs[1:]

    #将hs各个元素除以fs

    spectrum.hs[0] = 0

    #fs[0]=0不能除以0，所以自动让他等于0

wave1 = TriangleSignal(freq=440).make\_wave(duration=0.5)

#建立一个频率为440hz的采样长度为0.5s的三角波wave

wave1.make\_audio()

plt.subplot(311)

spectrum = wave1.make\_spectrum()

#创建wave的频谱

spectrum.plot( color='gray')

filter\_spectrum(spectrum)

spectrum.scale(440)#self.hs = factor\*self.hs

spectrum.plot()

decorate(xlabel='Frequency (Hz)')

wave2 = spectrum.make\_wave()

plt.subplot(312)

wave1.plot()

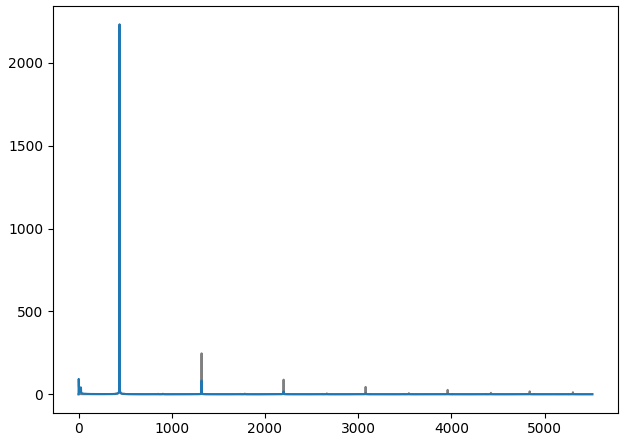
plt.subplot(313)

wave2.plot()

#filtered = spectrum.make\_wave()

#filtered.write(filename="2.5.wav")

plt.show()



它的作用就像一个低通滤波器。

使三角波听起来像正弦波