****

数字信号处理实验报告

专业：电子信息（专升本）

班别： 电子信息（专升本）201

姓名： 黄子介

学号： 3200442035

**浙大宁波理工学院实验报告**

**姓名 黄子介 学号 3200442035 学院** 信息科学与工程学院

**专业 电子信息 班级 电子信息（专升本）201 课程** 数字信号处理

**实验时间 2021-6-16 实验地点 SF405**   **指导教师 王一刚**

**实验四**

**实验目的：**周期信号及其谐波，采样与欠采样

**实验内容：**

熟悉锯齿波，矩形波，三角波的生成

显示频谱

写一个spectrum函数：将hs改为各元素除以对应的频率fs

**实验结果：**

2-2：

import numpy as np

import matplotlib.pyplot as plt

from thinkdsp import decorate

from thinkdsp import Sinusoid

from thinkdsp import normalize, unbias

from thinkdsp import SawtoothSignal

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

plt.subplot(4,1,1)

sawtooth.make\_spectrum().plot()

decorate(xlabel='Frequency (Hz)')

from thinkdsp import SquareSignal

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

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

plt.subplot(4,1,2)

square.make\_spectrum().plot()

decorate(xlabel='Frequency (Hz)')

from thinkdsp import TriangleSignal

plt.subplot(4,1,3)

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

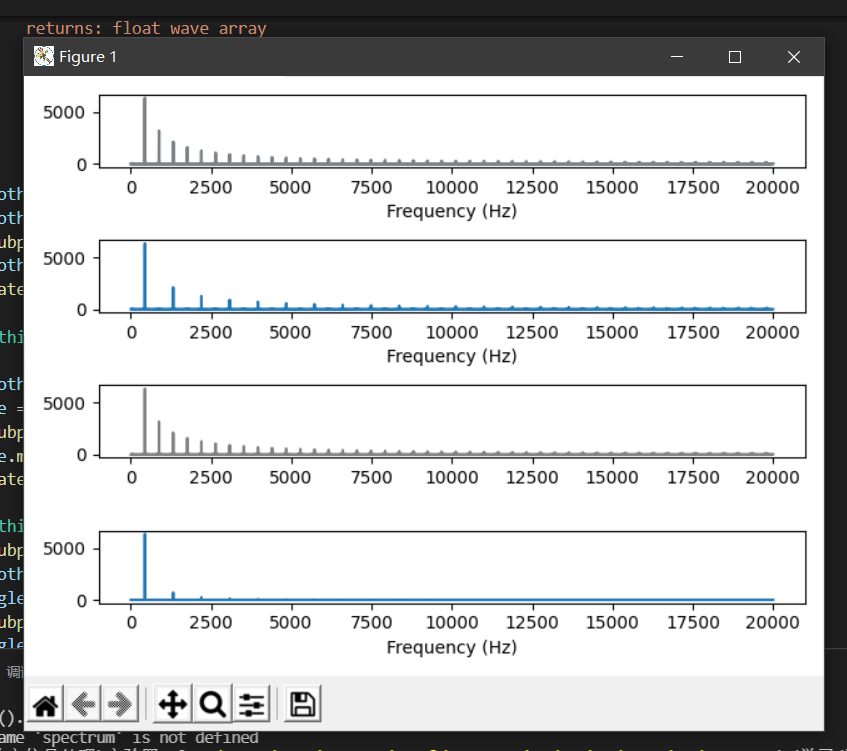
triangle = TriangleSignal(amp=0.79).make\_wave(duration=0.5, framerate=40000)

plt.subplot(4,1,4)

triangle.make\_spectrum().plot()

decorate(xlabel='Frequency (Hz)')

plt.show()



2-3：

import numpy as np

import matplotlib.pyplot as plt

from thinkdsp import SquareSignal

from thinkdsp import decorate

from thinkdsp import SinSignal

square = SquareSignal(1100)

plt.subplot(2,1,1)

square.plot()

decorate(xlabel='Time (s)')

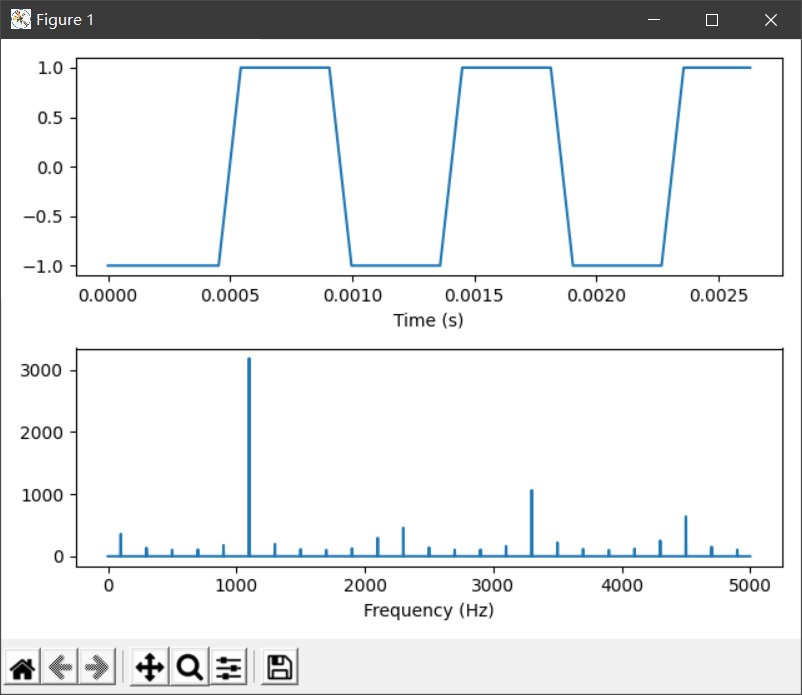
square = SquareSignal(1100).make\_wave(duration=0.5, framerate=10000)

plt.subplot(2,1,2)

square.make\_spectrum().plot()

decorate(xlabel='Frequency (Hz)')

plt.show()



2-4：

import numpy as np

import matplotlib.pyplot as plt

from thinkdsp import TriangleSignal

from thinkdsp import decorate

from thinkdsp import SinSignal

triangle = TriangleSignal(440).make\_wave(duration=0.01)

plt.subplot(2,1,1)

triangle.plot()

decorate(xlabel='Time (s)')

spectrum = triangle.make\_spectrum()

spectrum.hs[0]

print(spectrum.hs[0])

spectrum.hs[0] = 100

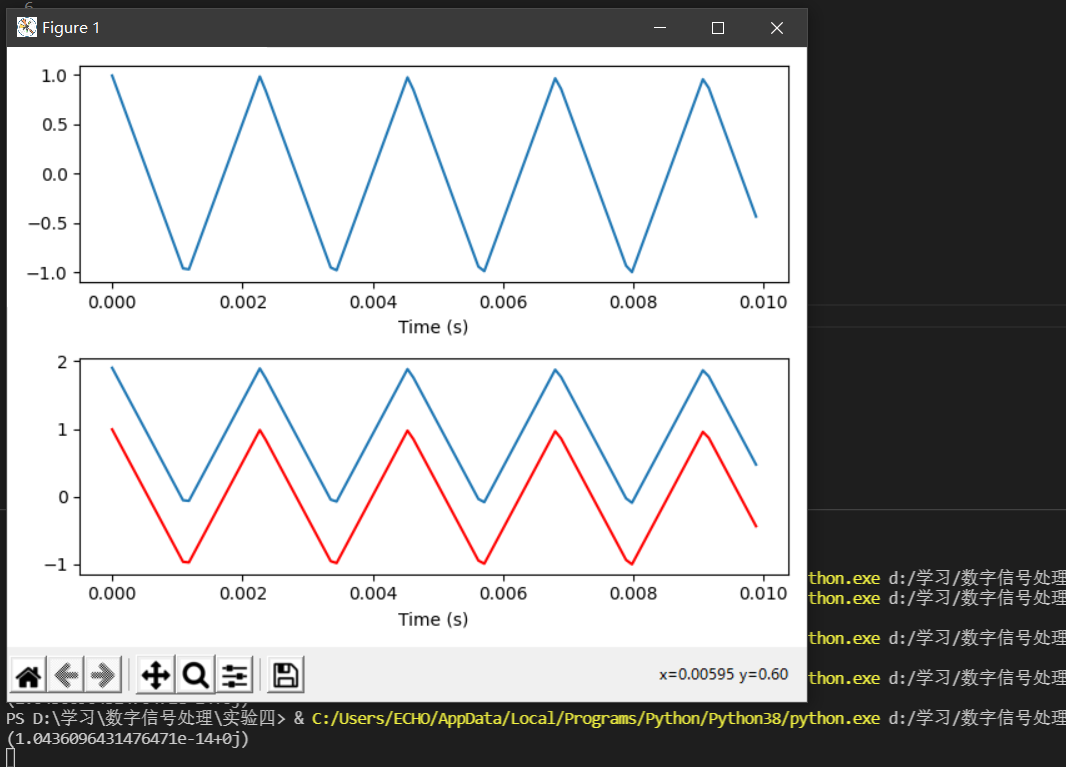
plt.subplot(2,1,2)

triangle.plot(color='gray')

spectrum.make\_wave().plot()

decorate(xlabel='Time (s)')

plt.show()



2-5：

import numpy as np

import matplotlib.pyplot as plt

from thinkdsp import TriangleSignal

from thinkdsp import SquareSignal

from thinkdsp import SawtoothSignal

from thinkdsp import decorate

from thinkdsp import SinSignal

def filter\_spectrum(spectrum):

    """Divides the spectrum through by the fs.

    spectrum: Spectrum object

    """

    # avoid division by 0

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

    spectrum.hs[0] = 0

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

spectrum = wave.make\_spectrum()

plt.subplot(3,1,1)

spectrum.plot(high=10000, color='red')

filter\_spectrum(spectrum)

spectrum.scale(440)

spectrum.plot(high=10000)

decorate(xlabel='Frequency (Hz)')

wave2=square = SquareSignal(1100).make\_wave(duration=0.5)

spectrum2 = wave2.make\_spectrum()

plt.subplot(3,1,2)

spectrum2.plot(high=10000, color='red')

filter\_spectrum(spectrum2)

spectrum2.scale(440)

spectrum2.plot(high=10000)

decorate(xlabel='Frequency (Hz)')

wave3=sawtooth = SawtoothSignal().make\_wave(duration=0.5)

spectrum3 = wave3.make\_spectrum()

plt.subplot(3,1,3)

spectrum3.plot(high=10000, color='red')

filter\_spectrum(spectrum3)

spectrum3.scale(440)

spectrum3.plot(high=10000)

decorate(xlabel='Frequency (Hz)')

plt.show()

