****

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： 王雯倩

学号： 3200442003

**浙大宁波理工学院实验报告**

**姓名 王雯倩 学号 3200442003 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 2021-6-19 实验地点 指导教师**

**实验四.信号的谐波**

**实验目的：**

**了解不同信号的谐波，并分析**

**实验内容：**

1.计算锯齿信号的频谱，并将其谐波结构跟三角波方波进行比较

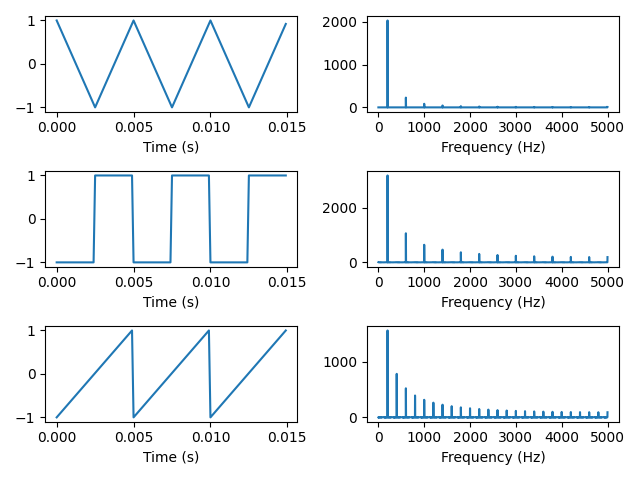
2.创建一个1100Hz的方波信号，并用一个wave来采样它

3.创建一个440Hz的三角波，创建一个0.01s的Wave，绘制波形

4.创建并计算spectrum，并绘制他

**实验结果：**

1. 计算锯齿信号的频谱，并将其谐波结构跟三角波方波进行比较



代码：

from thinkdsp import TriangleSignal

from thinkdsp import decorate

import matplotlib.pyplot as plt

import numpy as np

#三角波

signal = TriangleSignal(200)

duration = signal.period\*3

segment = signal.make\_wave(duration, framerate=10000)

plt.subplot(3,2,1)

segment.plot()

decorate(xlabel='Time (s)')

wave = signal.make\_wave(duration=0.5, framerate=10000)

spectrum = wave.make\_spectrum()

plt.subplot(3,2,2)

spectrum.plot()

decorate(xlabel='Frequency (Hz)')

#方波

from thinkdsp import SquareSignal

signal = SquareSignal(200)

duration = signal.period\*3

segment = signal.make\_wave(duration, framerate=10000)

plt.subplot(3,2,3)

segment.plot()

decorate(xlabel='Time (s)')

wave = signal.make\_wave(duration=0.5, framerate=10000)

spectrum = wave.make\_spectrum()

plt.subplot(3,2,4)

spectrum.plot()

decorate(xlabel='Frequency (Hz)')

#sawtoothSignal类

from thinkdsp import Sinusoid

from thinkdsp import normalize, unbias

class SawtoothSignal(Sinusoid):

    def evaluate(self, ts):

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

        frac, \_ = np.modf(cycles)

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

        return ys

signal = SawtoothSignal(200)

duration = signal.period\*3

segment = signal.make\_wave(duration, framerate=10000)

plt.subplot(3,2,5)

segment.plot()

decorate(xlabel='Time (s)')

wave = signal.make\_wave(duration=0.5, framerate=10000)

spectrum = wave.make\_spectrum()

plt.subplot(3,2,6)

spectrum.plot()

decorate(xlabel='Frequency (Hz)')

plt.show()

1. 创建一个1100Hz的方波信号，并用一个wave来采样它

代码：

from thinkdsp import TriangleSignal,SquareSignal

from thinkdsp import decorate

import matplotlib.pyplot as plt

import numpy as np

square = SquareSignal(1100)

cysquare = square.make\_wave(duration=2, framerate=10000)

cysquare.play('temp1.wav')

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

square.make\_spectrum().plot()

decorate(xlabel='Frequency (Hz)')

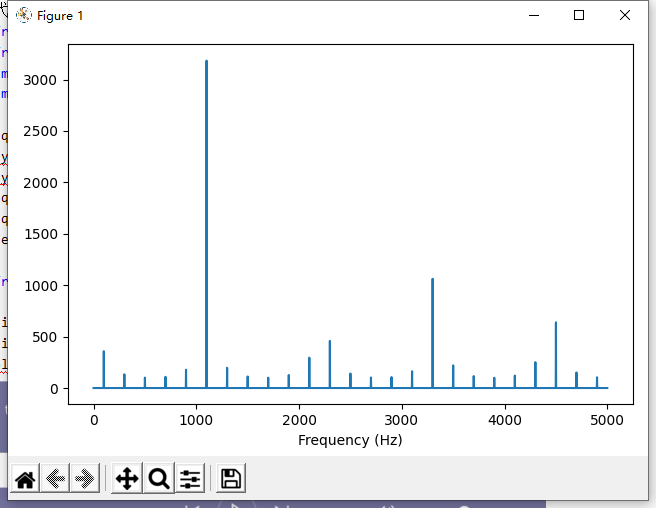
from thinkdsp import SinSignal

sin=SinSignal(500).make\_wave(duration=2, framerate=10000)

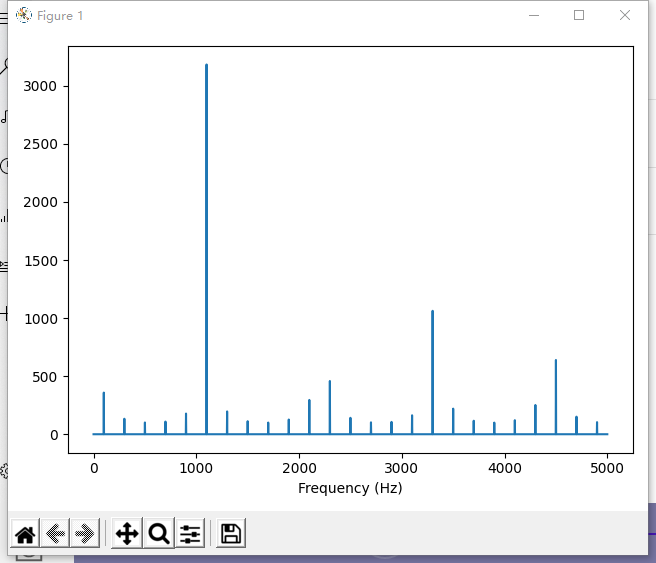
sin.play('temp2.wav')

plt.show()









Temp1比Temp2声音更加尖锐。

1. ①创建一个440Hz的三角波，创建一个0.01s的Wave，绘制波形

②创建一个Spectrum对象并打印spectrum.hs[0]

③设spectrum.hs[0]=100，这对波形的影响是什么？

代码：

from thinkdsp import Spectrum, TriangleSignal,SquareSignal

from thinkdsp import decorate

import matplotlib.pyplot as plt

import numpy as np

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

plt.subplot(2,1,1)

triangle.plot()

decorate(xlabel="time(s)")

spectrum=triangle.make\_spectrum()

print(spectrum.hs[0])

spectrum.hs[0] = 100

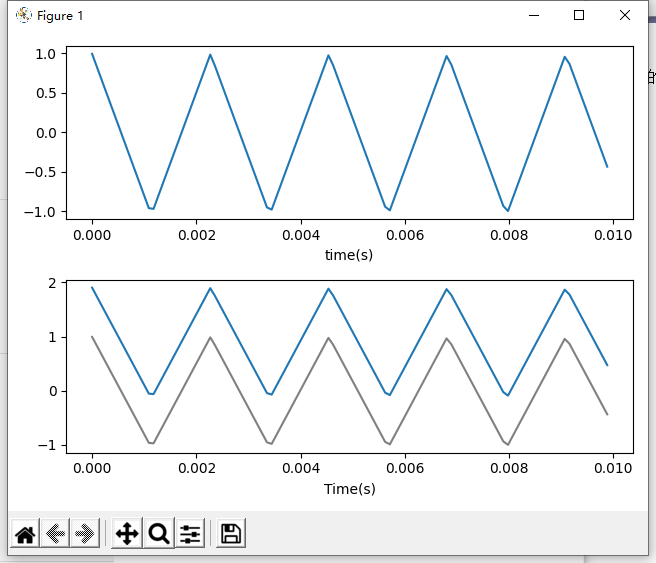
plt.subplot(2,1,2)

triangle.plot(color="gray")

spectrum.make\_wave().plot()

decorate(xlabel='Time(s)')

plt.show()



1. 写一个以spectrum为参数的函数修改它，将hs的各个元素除以对应的频率fs

代码：

from thinkdsp import Spectrum, TriangleSignal,SquareSignal, play\_wave

from thinkdsp import decorate

import matplotlib.pyplot as plt

import numpy as np

def filter\_spectrum(spectrum):

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

    spectrum.hs[0] = 0

wave = TriangleSignal(freq=440).make\_wave(duration=5)

#生成原音频

wave.play("temp3.wav")

spectrum = wave.make\_spectrum()

#绘制spectrum图像

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

filter\_spectrum(spectrum)

spectrum.scale(440)

#生成使用函数修改后的spectrum所绘制的图像

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

decorate(xlabel='Frequency (Hz)')

#从修改过的spectrum绘制一个波形并生成音频试听

spectrum.make\_wave().play("temp4.wav")

plt.show()

