Санкт-Петербургский Политехнический Университет Петра Великого
Институт компьютерных наук и технологий
Кафедра компьютерных систем и программных технологий

Отчёт по лабораторной работе

Телекоммуникационные технологии
Сигналы телекоммуникационных систем

Работу выполнил:
Крылов И.С.
Группа: 33501/4
Преподаватель:
Богач Н.В.

Санкт-Петербург
2018

Содержание

1 Цель работы

Познакомиться со средствами генерации и визуализации простых сигналов.

2 Программа работы

Промоделировать синусоидальный и прямоугольный сигналы с различными параметрами. Получить их спектры и вывести на график.

3 Теоретический материал

Сигнал - это изменение одной величины в зависимости от другой. В математическом понимании сигнал это функция. В основе обработки сигналов лежит их анализ, целью которого является сравнение сигналов между собой для выявления их сходства и различия. Одним из наиболее востребованных способов изучения и сравнения сигналов является спектральный анализ - преобразование Фурье временного сигнала (изменяющегося во времени) в частотную область для получения спектра этого сигнала.

В основе спектрального анализа лежит идея разложения сигнала на более простые сигналы в базисе ортогональных функций. Так, если в качетсве базиса выбрать функцию синуса, то спектр анализируемого сигнала будет представлять собой набор синусоид, составляющих этот сигнал. Такое разложение возможно с помощью дискретного преобразования Фурье(ДПФ). Для более чёткого понимаю приницпа работы ДПФ следует рассмотреть разложение в ряд Фурье.

Разложению в ряд Фурье подвергаются периодические сигналы, представляемые в виде суммы гармонических функций(в общем виде $s(t) = A\cos(t\omega + \phi)$), либо комплексных экспонент с частотами, образующими арифметическую прогрессию. Для того чтобы такое разложение существовало, фрагмент сигнала длительностью в один период должен удовлетворять условиям Дирихле:

1. нет разрывов второго рода
2. число разрывов первого рода конечно
3. число экстремумов конечно

В синусно-косинусной форме интеграл Фурье имеет следующий вид:

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k cos(k\omega_1 t) + b_k sin(k\omega_1 t)) \tag{1}$$

, где ω_1 - круговая частота, $k\omega_1$ - гармоники, а коэффициенты ряда a_k и b_k рассчитываются по формулам:

$$a_k = \frac{2}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} s(t)\cos(k\omega_1 t)dt, \quad b_k = \frac{2}{T}\int_{\frac{-T}{2}}^{\frac{T}{2}} s(t)\sin(k\omega_1 t)dt \tag{2}$$

Константа a_0 расчитывается по общей формуле для a_k.

Несложно получить комлексную форму ряда Фурье. Для этого, воспользовавшись формулами тригонометрических преобразований, сократим подинтегральное слагаемое синус, получив косинус той же частоты с иной амлитудой и некоторой начальной фазой. Такая форма представления ряда Фурье называется вещественной. Представим косинус в виде полусуммы комплексных экспонент $\cos x = \frac{1}{2}(e^{jx} + e^{-jx})$. Такое представление следует из формулы Эйлера $e^{jx} = \cos x + j \sin x$. Подставив новое значение косинуса в вещественную форму ряда Фурье, будем трактовать экспоненты с отрицательными степенями как члены ряда с отрицательными номерами. Результатом этих преобразований станет искомая комлексная форма ряда Фурье:

$$s(t) = \sum_{k=-\infty}^{+\infty} C_k e^{-jk\omega_1 t} \tag{3}$$

Выразим коэффициенты C_k ряда Фурье в комлексной форме через коэффициенты a_k и b_k и получим формулу непосредственного расчёта коэффициентов C_k:

$$C_k = \frac{1}{T} \int_{\frac{-T}{2}}^{\frac{T}{2}} s(t) e^{-jk\omega_1 t} dt \tag{4}$$

Если $s(t)$ - четная функция, то коэффициенты ряда C_k будут вещественные, иначе - чисто мнимые.

Для перехода от ряда Фурье к преобразованию удобно будет воспользоваться подходом увеличения количества повторений одиночных импульсов, при это заполним промежутки нулевыми значениями. Из формулы (4) следует, что при перерасчёте коэффициентов ряда нам придётся вычислить тот же интеграл, но для более тесно расположенных частот $w_k = lw_1$. Изменение пределов интеграла не будет играть роли засчёт нулевых значений. Единственным изменением будет уменьшение общего уровня гармоник, вызванного увеличением периода T.

С ростом периода следования, гармоники располагаются ближе друг к другу, а общий уровень спектральных составляющих уменьшается. Если устремить период к бесконечности, то формула для расчёта коэффициентов комплексного ряда Фурье претерпит следующие изменения:

1. частота становится непрерывным параметром преобразования
2. исчезает множитель $\frac{1}{T}$
3. результатом вычислений становится функция частоты $S(w)$ - спектральная функция (плотность) сигнала $s(t)$

В результате формула ряда Фурье преобретает следующий вид:

$$S(w) = \int_{-\infty}^{+\infty} s(t) e^{-j\omega t} dt \tag{5}$$

В формуле самого ряда Фурье суммирование заменяется на интегрирование:

$$s(t) = \frac{1}{2pi} \int_{-\infty}^{+\infty} S(w) e^{j\omega t} dw \tag{6}$$

Полученное выражение называется обратным преобразованием Фурье.

4 Ход работы

Для обработки цифровых сигналов воспользуемся готовым пакетом ThinkX и встроенными в него модулями ThinkDSP и ThinkPlot. Для представления сигнала в модуле thinkdsp.py есть класс signal, задающий Python-представление математической функции, и от которого наследуется класс для представления синусоидального сигнала - Sinusoid. Для создания синусоидального сигнала определена функция SinSignal, в тело которой передаются три параметра: freq - частота в Герцах, amp - амлитуда в относительных единицах, offset - фазовый сдвиг в радианах.

```
In [2]: from __future__ import print_function, division
        %matplotlib inline
        import thinkdsp
        import thinkplot
        import matplotlib.pyplot as plt
        import numpy as np
        from ipywidgets import interact, interactive, fixed
        import ipywidgets as widgets
        from IPython.display import display
```

Для обработки сигнала в thinkdsp.py определён класс wave. Wave - это сигнал, обрабатываемый в последовательности моментов времени. Каждый момент врмени называется кадром (frame). Для создания экземпляра wave из существующего объекта сигнала, в классе signal определён метод make_wave с соответсвующими входными параметрами: duration - длина wave в секундах, start - время старта в секундах, framerate - число кадров (выборок) в секунду(по умолчанию устанавливается значение 11025 кадров в секунду - стандртная часта выборки, используемая в звуковых файлах разного формата). Wave поддерживает метод plot, использующий pyplot.

```
In [3]: sin_sig1 = thinkdsp.SinSignal(freq=50, amp=0.5, offset=0)
        sin_sig2 = thinkdsp.SinSignal(freq=100, amp=0.7, offset=0)
        sin_sig3 = thinkdsp.SinSignal(freq=250, amp=0.8, offset=0)
        sin_sig4 = thinkdsp.SinSignal(freq=500, amp=1, offset=0)
```

```
In [4]: thinkplot.preplot(num=4, rows=2, cols=2)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sin_sig1')
        wave1 = sin_sig1.make_wave(framerate=11025)
        wave1.plot(color='blue')
        thinkplot.subplot(2)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sin_sig2')
        wave2 = sin_sig2.make_wave(framerate=11025)
        wave2.plot(color='red')
        thinkplot.subplot(3)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sin_sig3')
        wave3 = sin_sig3.make_wave(framerate=11025)
        wave3.plot(color='green')
        thinkplot.subplot(4)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sin_sig4')
        wave4 = sin_sig4.make_wave(framerate=11025)
        wave4.plot(color='black')
```

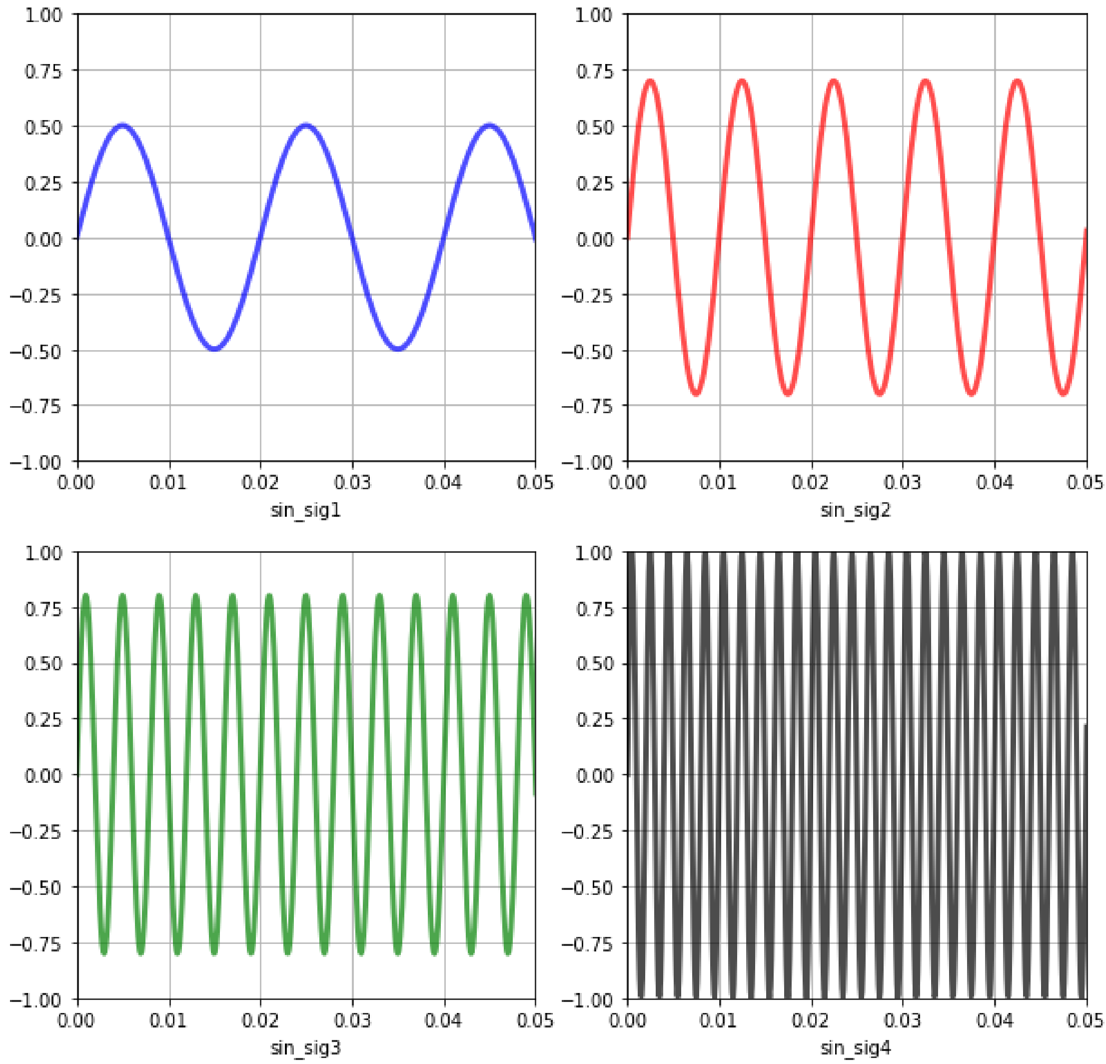

Image 1: Sinusoid signals

Для представления спектра сигнала в модуле thinkdsp.py определён класс Spectrum, поддерживающий функцию plot. Класс wave поддерживает make_spectrum, возвращающий спектр.

```
In [5]: thinkplot.preplot(num=4, rows=2, cols=2)
        plt.axis([-2000, 2000, 0, 6000])
        plt.xlabel('sin_sig1 spectrum')
        wave1.make_spectrum(full=True).plot(color='blue')
        thinkplot.subplot(2)
        plt.axis([-2000, 2000, 0, 6000])
        plt.xlabel('sin_sig2 spectrum')
        wave2.make_spectrum(full=True).plot(color='red')
        thinkplot.subplot(3)
        plt.axis([-2000, 2000, 0, 6000])
        plt.xlabel('sin_sig3 spectrum')
        wave3.make_spectrum(full=True).plot(color='green')
        thinkplot.subplot(4)
        plt.axis([-2000, 2000, 0, 6000])
        plt.xlabel('sin_sig4 spectrum')
        wave4.make_spectrum(full=True).plot(color='black')
```

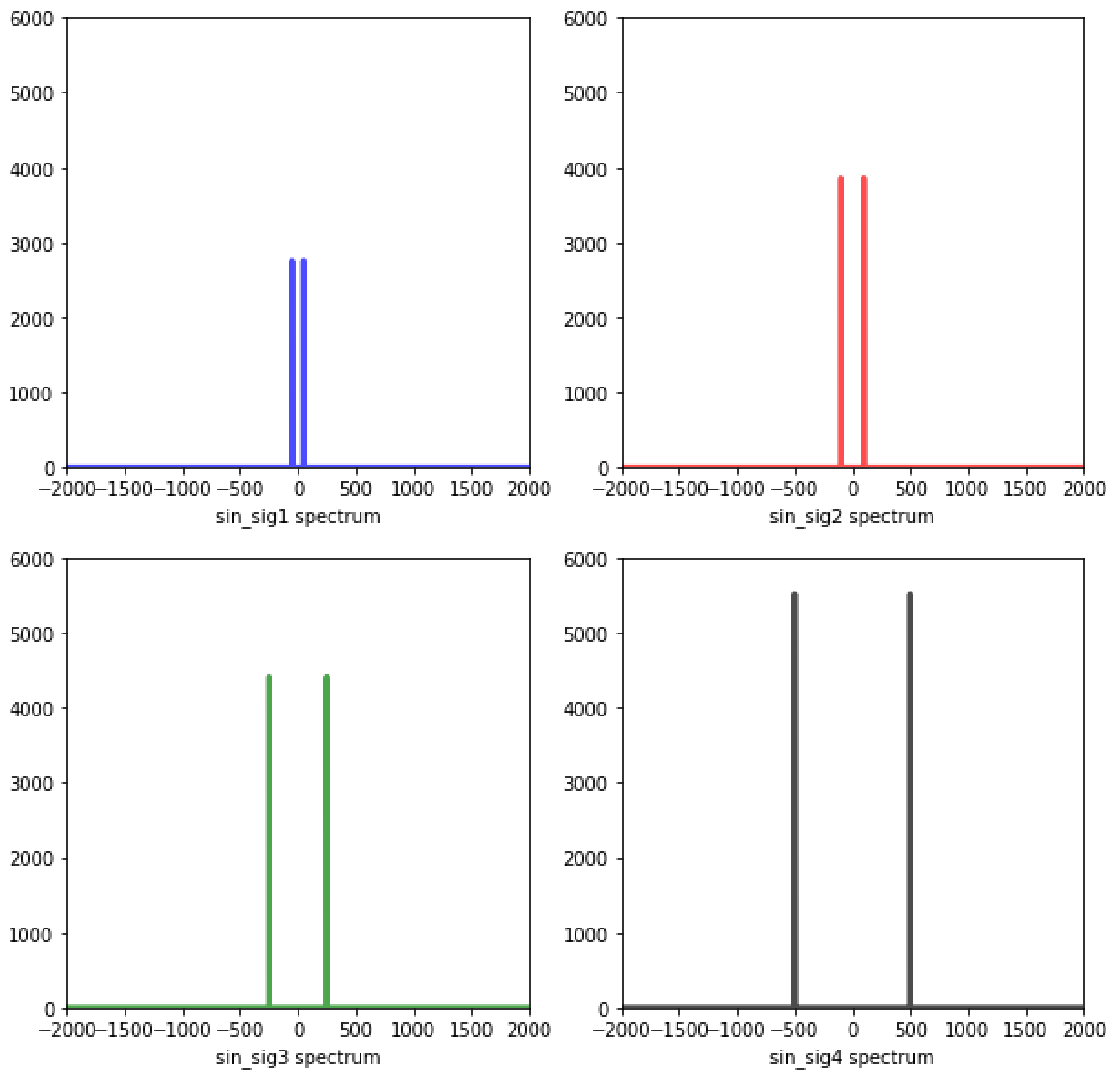

Image 2: Sinusoid signals' spectrums

Выше представлены спектры соответсвующих заданных синосуидальных сигналов. У синусоиды только одна частотная компонента, чем объясняется единственный пик в её спектре.

Для создания прямоугольного сигнала thinkdsp.py предоставляет функцию SquareSignal, принимающую на вход те же параметры что и SinSignal.

```
In [ ]: sq_sig1 = thinkdsp.SquareSignal(freq=50, amp=0.5, offset=0)
        sq_sig2 = thinkdsp.SquareSignal(freq=100, amp=0.7, offset=0)
        sq_sig3 = thinkdsp.SquareSignal(freq=250, amp=0.8, offset=0)
        sq_sig4 = thinkdsp.SquareSignal(freq=500, amp=1, offset=0)
```

```
In [8]: thinkplot.preplot(num=4, rows=2, cols= 2)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sq_sig1')
        sq_wave1 = sq_sig1.make_wave(framerate=11025)
        sq_wave1.plot(color='blue')
        thinkplot.subplot(2)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sq_sig2')
        sq_wave2 = sq_sig2.make_wave(framerate=11025)
        sq_wave2.plot(color='red')
        thinkplot.subplot(3)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sq_sig3')
        sq_wave3 = sq_sig3.make_wave(framerate=11025)
        sq_wave3.plot(color='green')
        thinkplot.subplot(4)
        plt.grid(True)
        plt.axis([0, 0.05, -1, 1])
        plt.xlabel('sq_sig4')
        sq_wave4 = sq_sig4.make_wave(framerate=11025)
        sq_wave4.plot(color='black')
```

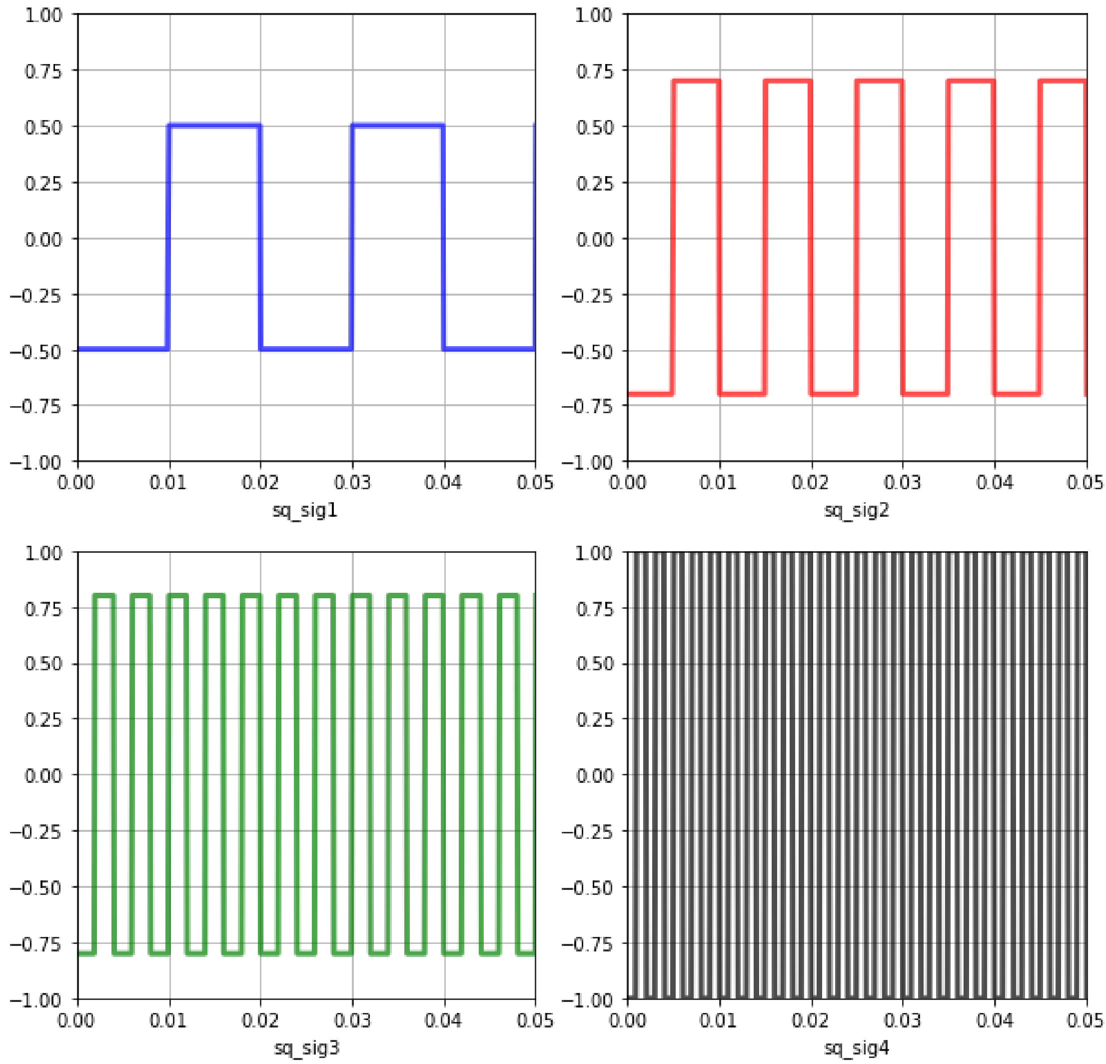

Image 3: Square signals

```
In [9]: thinkplot.preplot(num=4, rows=2, cols=2)
        plt.axis([-2000, 2000, 0, 7500])
        plt.xlabel('sq_sig1 spectrum')
        sq_spectre1 = sq_wave1.make_spectrum(True)
        sq_spectre1.plot(color='blue')
        thinkplot.subplot(2)
        plt.axis([-4000, 4000, 0, 7500])
        plt.xlabel('sq_sig2 spectrum')
        sq_spectre2 = sq_wave2.make_spectrum(True)
        sq_spectre2.plot(color='red')
        thinkplot.subplot(3)
        plt.axis([-6000, 6000, 0, 7500])
        plt.xlabel('sq_sig3 spectrum')
        sq_spectre3 = sq_wave3.make_spectrum(True)
        sq_spectre3.plot(color='green')
        thinkplot.subplot(4)
        plt.axis([-6000, 6000, 0, 7500])
        plt.xlabel('sq_sig4 spectrum')
        sq_spectre4 = sq_wave4.make_spectrum(True)
        sq_spectre4.plot(color='black')
```

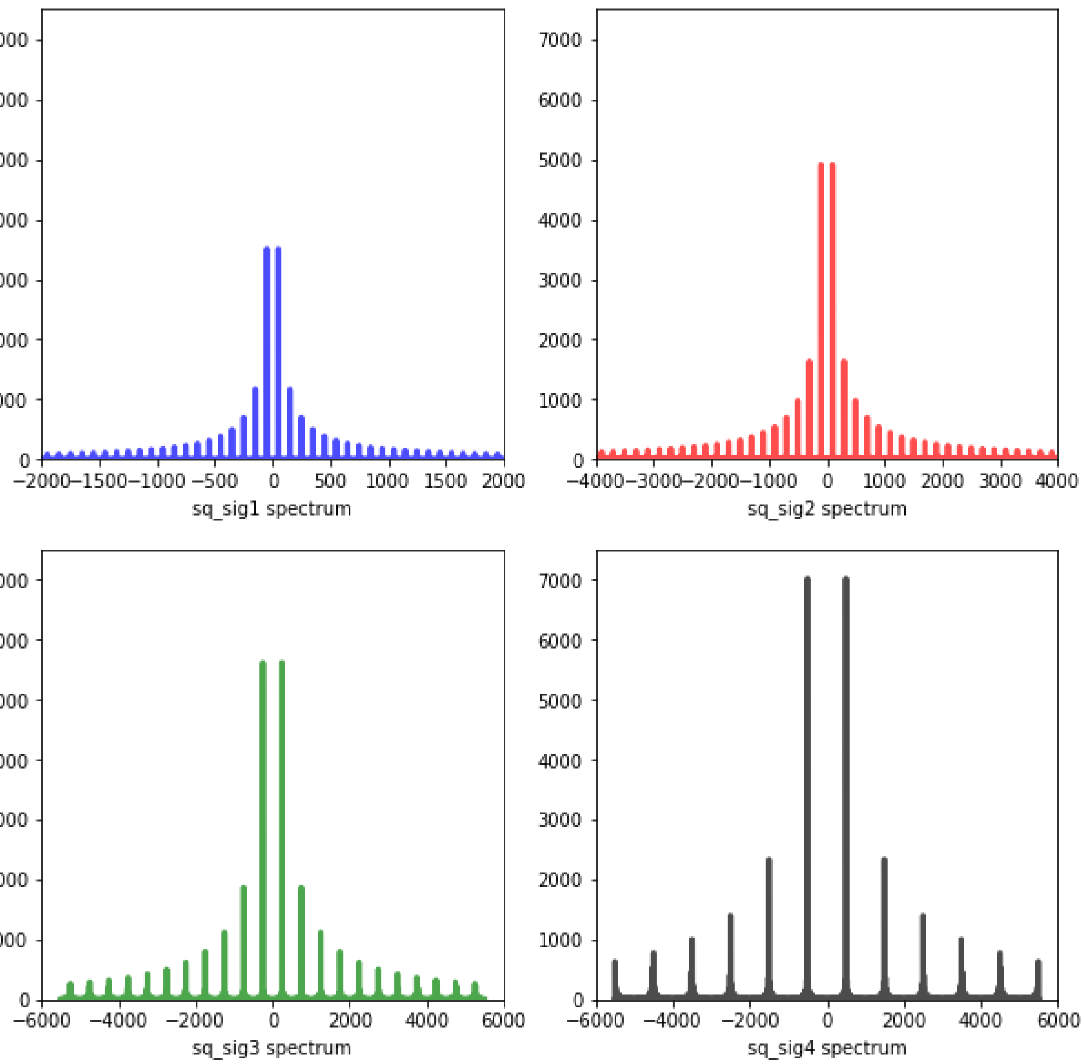

Image 4: Square signals' spectrums

На рисунках выше представлены графики спектров соответсвующих прямоугольных сигналов. Прямоугольный сигнал содержит только нечётные гармоники. Амлитуда гармоник спадает пропорционально частоте. С ростом частоты сигнала, растёт расстояние между пиками в его спектре.

5 Выводы

В проделанной лабораторной работе были рассмотрены основные параметры синусоидального и прямоугольного сигналов. Был изучен один из основных способов исследования сигналов - спектральный анализ. Было наглядно продемонстрировано что с ростом периода следования сигнала, общий уровень спектральных составляющих уменьшается и наоборот, чем дальше располагаются гармоники друг от друга, тем амплитуда выше.