

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

# Телекоммуникационные технологии

Отчет по лабораторной работе №2

Корреляция

**Работу**

**выполнил:**

Чугунов А.А.

Группа: 33501/4

**Преподаватель:**

Богач Н.В.

Санкт-Петербург  
2017

## 1. Цель работы

Познакомиться с понятием корреляции и функцией корреляции.

## 2. Теоретическая информация

В данной лабораторной работе будем рассматривать корреляционный анализ. Его смысл состоит в количественном измерении степени сходства различных сигналов. Для этого будем использовать специальные корреляционные функции. Так, для получения взаимной корреляции двух последовательностей, имеем следующую формулу:

$$r_{12}(j) = \frac{1}{N} \sum_{n=0}^{N-1} x_1(n)x_2(n+j) = r_{12}(-j) = \frac{1}{N} \sum_{n=0}^{N-1} x_2(n)x_1(n-j) \quad (1)$$

Здесь  $j$  - это смещение одного сигнала относительно другого.

Так же можно ввести аналогичную формулу для непрерывной временной области:

$$r_{12}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x_1(t)x_2(t+\tau)dt \quad (2)$$

Расчет корреляции можно ускорить, воспользовавшись следующей формулой:

$$r_{12}(j) = \frac{1}{N} F_D^{-1}[X_1^*(k)X_2(k)] \quad (3)$$

Здесь,  $F_D^{-1}$  - обратное преобразование Фурье. При различной длине сигналов выполнить расчет корреляции можно путем добавления к двум последовательностям дополняющих нулей. Если последовательность  $x_1(n)$  имеет длину  $N_1$ , а последовательность  $x_2(n)$  —  $N_2$ , то  $x_1(n)$  дополняется  $(N_2 - 1)$  нулями, а  $x_2(n)$  —  $(N_1 - 1)$  нулями. Далее на основе двух расширенных последовательностей рассчитывается взаимная корреляция.

## 3. Ход выполнения работы

Имеем сигнал, состоящий из нулей и единиц -  $[0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0]$  и синхросылку -  $[1, 0, 1]$ . Задача - найти положение синхросылки в сигнале. Изначально преобразуем все нули в -1. Это необходимо для корректной работы алгоритма быстрого расчета корреляции сигналов, так как сигнал будет дополняться нулями. Корреляцию рассчитаем с помощью встроенной функции `correlate` библиотеки `numpy`. Алгоритм для быстрого расчета корреляции напишем самостоятельно используя преобразования Фурье из той же библиотеки. Результаты представлены на Рисунке. 3.1

Листинг 1: CorrelLab.py

```
1 from scipy.fftpack import fft
2 import numpy as np
3 import time as time
4 import matplotlib.pyplot as plt
5 from scipy import signal
6
7 sig = np.asarray([0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0])
8 syncSig = np.asarray([1, 0, 1])
9
10 sig = sig + sig - 1
```

```

11 syncSig = syncSig + syncSig - 1
12
13 print(sig)
14 print(syncSig)
15
16 correlation = np.asarray([])
17 t = np.arange(1e6)
18 sum = 0
19 for ti in t:
20     beforeTime = time.time()
21     correlation = np.correlate(sig, syncSig)
22     afterTime = time.time()
23     sum = sum + afterTime - beforeTime
24 print('time_needed_for_correlation:', sum / len(t))
25
26 sig = np.asarray([-1, -1, -1, 1, -1, 1, -1, 1, 1, 1, -1, -1, -1, -1, 1, -1, 0,
    ↪ 0])
27 syncSig = np.asarray([1, -1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0])
28 sum = 0
29 finalCorrel = np.asarray([])
30 for di in t:
31     beforeTime = time.time()
32     conjY = np.conjugate(np.fft.fft(syncSig))
33     y1Fft = np.fft.fft(sig)
34     multiplication = conjY * y1Fft
35     finalCorrel = np.fft.ifft(multiplication)
36     afterTime = time.time()
37     sum = sum + afterTime - beforeTime
38 print('time_needed_for_fast_correlation:', sum / len(t))
39 print(correlation)
40 print(finalCorrel[:14].real)

```

```

[-1 -1 -1  1 -1  1 -1  1  1 -1 -1 -1 -1  1 -1]
[ 1 -1  1]
time needed for correlation:  1.4900157451629638e-06
time needed for fast correlation:  1.94175283908844e-05
[-1  1 -3  3 -3  3 -1  1 -1  1 -1  1 -3]
[-1.  1. -3.  3. -3.  3. -1.  1. -1.  1. -1. -1.  1. -3.]

```

Рисунок 3.1. Результаты работы прямого и быстрого расчета корреляции

Наблюдаем более быструю работу прямого алгоритма, что является очень странным. Можем сделать предположение, что быстрый алгоритм работает эффективней с большим количеством данных. Стоит отметить, что оба алгоритма выполнили свою задачу и определили положение синхропосылки. В данном случае начало синхропосылки отмечено цифрой 3.

## 4. Выводы

Проделав лабораторную работу, рассмотрели понятие корреляции и научились пользоваться двумя алгоритмами её расчета. Установили, что прямой алгоритм поиска корреляции более быстро рассчитывает корреляцию, чем быстрый алгоритм.