

Міністерство освіти і науки України  
Національний технічний університет України "Київський політехнічний інститут  
імені Ігоря Сікорського"  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки

## **ЗВІТ**

### **Лабораторна робота №1.2**

з дисципліни  
«Інтелектуальні вбудовані системи»

на тему  
«Дослідження автокореляційної і взаємнокореляційної функцій  
випадкових сигналів»

Виконав:  
Василиненко Д.Д.  
Студент групи ІП-84  
Перевірив:  
Регіда Павло Геннадійович

Київ 2021

## Завдання

Для згенерованого випадкового сигналу з Лабораторної роботи N 1 відповідно до заданого варіантом (Додаток 1) розрахувати його автокореляційної функцію. Згенерувати копію даного сигналу і розрахувати взаємнокореляційну функцію для 2-х сигналів. Розробити відповідну програму і вивести отримані значення і графіки відповідних параметрів.

## Основні теоретичні відомості

Значення автокореляційної функції фізично представляє зв'язок між значенням однієї і тієї ж величини, тобто для конкретних моментів  $t_k, \tau_s$ , значення  $R_{xx}(t, \tau)$  оцінюється друге змішаним центральним моментом 2-х перетинів випадкових процесів  $x(t_k), x(t_k + \tau_s)$

$$R_{xx}(t, \tau_s) = \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N \overbrace{(x_i(t_k) - M_x(t_k))}^{x(t_k)} \cdot \overbrace{(x_i(t_k + \tau_s) - M_x(t_k + \tau_s))}^{x(t_k + \tau_s)}$$

для кожного конкретного інтервалу потрібно проходити по всім  $t_k$  (перетинах).

Центральні значення можна замінити:

$$\begin{aligned} & \overline{x(t_k)}, \overline{x(t_k + \tau_s)}, \text{ тобто їх } M_x = 0 \\ & \left[ \begin{aligned} R_{xx}(t, \tau) &= \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N \overline{x_i(t)} \cdot \overline{x_i(t + \tau)} \\ R_{xx}(t, \tau) &= \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N \overline{x_i(t)} \cdot \overline{x_i(t + \tau)} \end{aligned} \right] \end{aligned}$$

Обчислення кореляційної функції  $R_{xx}(t, \tau)$  є відносно складним, оскільки необхідно попереднє обчислення математичного очікування  $M_x$  для виконання кількісної оцінки, іноді виповнюється ковариационної функцією:

$$C_{xx}(t, \tau) = \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N x_i(t) \cdot x_i(t + \tau)$$

## Завдання по варіанту (4):

- Число гармонік в сигналі - 12
- Гранична частота - 2400
- Кількість дискретних відліків - 1024

## Вихідний код:

### correlation.py

```
import numpy as np

def selfcorrelation(x):
    N = len(x)
    Mx = np.mean(x)
    correlation = []
    for tau in range(N / 2):
        r = 0
        for time in range(N / 2):
            r += (x[time] - Mx) * (x[time + tau] - Mx)
        correlation.append(r / ((N / 2) - 1))
    return correlation

def correlation(x, y = []):
    N = len(x)
    Mx = np.mean(x)
    My = np.mean(y)
    correlation = []
    for tau in range(N / 2):
        r = 0
        for time in range(N / 2):
            r += (x[time] - Mx) * (y[time + tau] - My)
        correlation.append(r / ((N / 2) - 1))
    return correlation
```

### rsg.py

```
import random
import math
import time

def getRandomSignal(harmonicsAmount, limitFrequency, N):
    signal = [0] * N
    for i in range(harmonicsAmount):
        w = limitFrequency / (i+1)
        A = random.random()
```

```

    Fi = random.random()
    for t in range(N):
        signal[t] += (A * math.sin(w * t + Fi))
    return signal

```

## index.py

```

import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import rsg
import correlation as corr

n = 12
W = 2400
N = 1024
time = range(N)
x = rsg.getRandomSignal(n, W, N)
y = rsg.getRandomSignal(n, W, N)

autocorrelation = corr.selfcorrelation(x)
correlation = corr.correlation(x, y)

print(autocorrelation)
print(correlation)

corrR = list(range(N / 2))

fig, (ax1, ax2, ax3) = plt.subplots(3, 1)
plt.subplots_adjust(left=0.05, bottom=0.1, right=0.97, wspace=0.1)
fig.suptitle('Lab 1.2')

ax1.plot(x, color='r', label='s 1')
ax1.plot(y, color='g', label='s 2')
ax1.set_title('Generated signals')
ax1.set_xlabel='time', ylabel='generated signal')
ax1.legend()

ax2.plot(corrR, autocorrelation, color='r')
ax2.set_title('Autocorrelation (s 1)')
ax2.set_xlabel='t', ylabel='correlation')

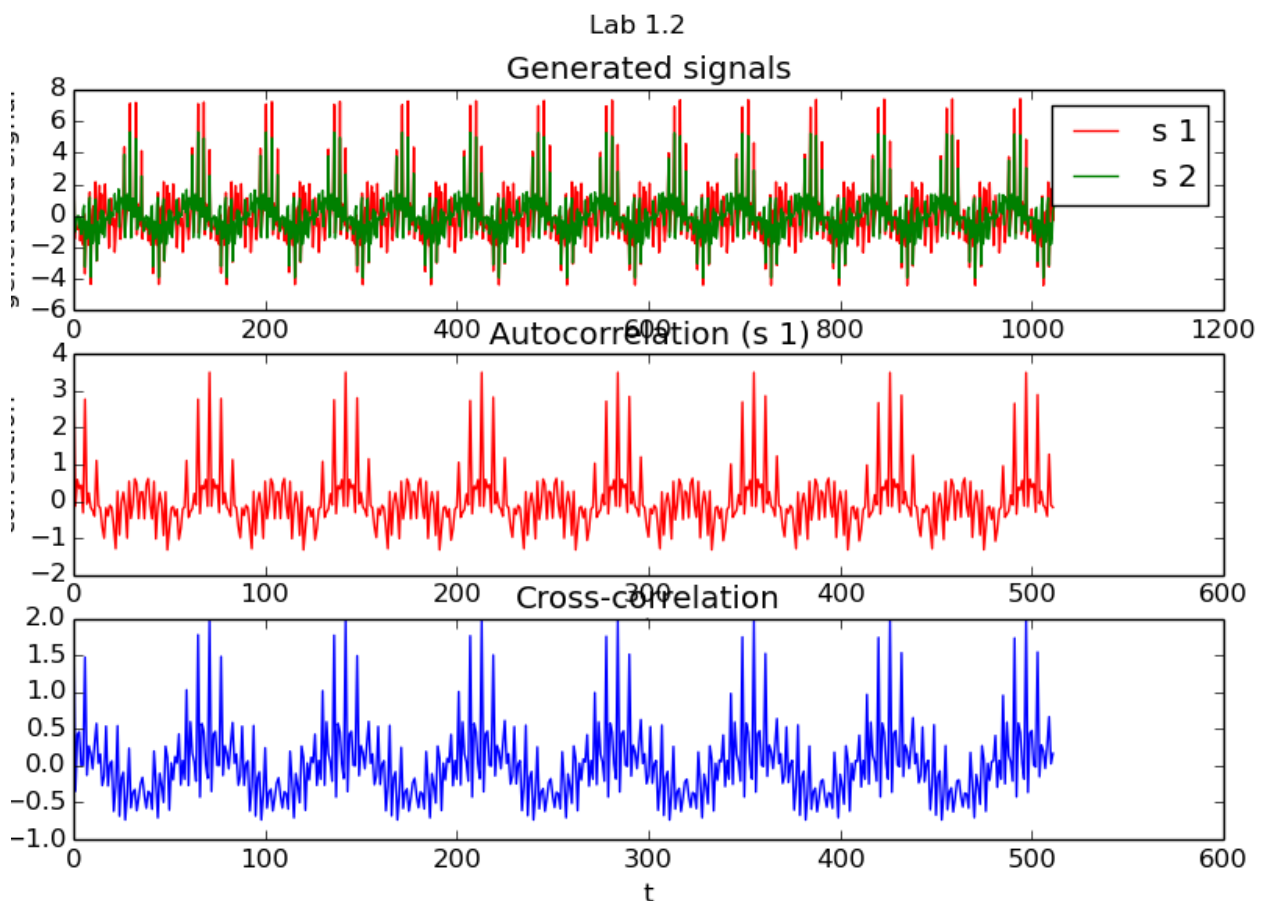
ax3.plot(corrR, correlation)
ax3.set_title('Cross-correlation')
ax3.set_xlabel='t', ylabel='correlation')

```

```
fig.savefig("lab1.png")

plt.show()
```

## Результати роботи програми



## Висновки

У ході роботи було згенеровано два сигнали. Під час виконання лабораторної роботи на прикладі яких обраховувалися автокореляція та взаємна кореляція. Було отримано 3 графіки: згенеровані сигнали, накладені одне на одного, автокореляції для сигналу X, взаємної кореляції двох сигналів.