

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

Лабораторна робота №1.2
з дисципліни
«Інтелектуальні вбудовані системи»
на тему
«Дослідження автокореляційної і взаємною-
кореляційної функцій випадкових сигналів»

Виконала:
студентка
групи ІП-83
Гомілко Д. В.

Перевірів:
Регіда П. Г.

Київ 2021

Основні теоретичні відомості, необхідні для виконання лабораторної роботи

Значення автокореляційної функції фізично представляє зв'язок між значенням однієї і тієї ж величини, тобто для конкретних моментів t_k, τ_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:

$$\overline{x}(t_k), \overline{x}(t_k, \tau_s), \text{ тобто їх } M_x = 0$$

$$\left[\begin{array}{l} R_{xx}(t, \tau) = \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N \overline{x}_i(t) \cdot \overline{x}_i(t + \tau) \\ R_{xx}(t, \tau) = \lim_{N \rightarrow \infty} \frac{1}{N-1} \sum_{i=1}^N \overline{x}_i(t) \cdot \overline{x}_i(t + \tau) \end{array} \right]$$

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

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

Умови завдання для варіанту

Для згенерованого випадкового сигналу з Лабораторної роботи 1 відповідно до заданого варіантом 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.

Варіант: 04 (номер заліковки — 8304):

Варіант	Число гармонік в сигналі, n	Гранична частота, ω	Кількість дискретних відліків, N
4	12	2400	1024

Лістинг програми із заданими умовами завдання

signalGenerator.py

```
import random

import math

def createSignal(harmonics, maxFrequency, calls):
    sumsArr = [0] * calls
    frequencyStep = maxFrequency / harmonics
    for i in range(harmonics):
        frequency = frequencyStep * (i + 1)
        amplitude = random.random()
        phase = random.random()
        for t in range(calls):
            sumsArr[t] += amplitude * math.sin(frequency * t + phase)
    return sumsArr
```

calc.py

```
mean = lambda vals: sum(vals) / len(vals)

def variance(vals):
    M = mean(vals)
    N = len(vals)
    return sum((x - M) ** 2 for x in vals) / (N - 1)

def correlation(signalX, signalY = []):
    N = len(signalX)
    calcRange = N / 2
    Mx = mean(signalX)
    My = mean(signalY) if signalY else Mx
    comparedSignal = signalY or signalX
    correlation = []
    for tau in range(int(calcRange)):
        res = 0
        for t in range(int(calcRange)):
            res += (signalX[t] - Mx) * (comparedSignal[t + tau] - My)
        correlation.append(res / (calcRange - 1))
    return correlation
```

complexity.py

```
import time

import signalGenerator
import calc

def getGenerationComplexity(stepsCount, maxFrequency, calls):
    elapsed = []
    harmonics = []
    for i in range(stepsCount):
        count = int(10 * (i + 1))
        harmonics.append(count)
        start = time.perf_counter()
        signalGenerator.createSignal(count, maxFrequency, calls)
        stop = time.perf_counter()
        elapsed.append(stop - start)
    return harmonics, elapsed

def getCorrealtionComplexity(stepsCount, harmonics, maxFrequency):
    elapsed = []
    harmonics = []
    for i in range(stepsCount):
```

```

count = int(10 * (i + 1))
harmonics.append(count)
signal = signalGenerator.createSignal(count, maxFrequency, count)
start = time.perf_counter()
calc.correlation(signal)
stop = time.perf_counter()
elapsed.append(stop - start)
return harmonics, elapsed

```

lab1-2.py

```

import matplotlib.pyplot as plt

import signalGenerator
import calc
import complexity

HARMONICS = 12
MAX_FREQUENCY = 2400
DISCRETE_CALLS = 1024
COMPLEXITY_COUNT_LOOPS = 200

signalX = signalGenerator.createSignal(
    HARMONICS,
    MAX_FREQUENCY,
    DISCRETE_CALLS
)
signalY = signalGenerator.createSignal(
    HARMONICS,
    MAX_FREQUENCY,
    DISCRETE_CALLS
)
autocorrelation = calc.correlation(signalX)
correlation = calc.correlation(signalX, signalY)

harmonics, time = complexity.getCorrealtionComplexity(
    COMPLEXITY_COUNT_LOOPS,
    MAX_FREQUENCY,
    DISCRETE_CALLS
)

len = int(DISCRETE_CALLS / 2)
correlationRange = list(range(len))

fig, axs = plt.subplots(2, 2)
plt.subplots_adjust(left=0.05, bottom=0.1, right=0.97, wspace=0.1)
fig.suptitle('Lab 1.2')

axs[0, 0].plot(signalX, color='r', linewidth=0.75, label='signal X')
axs[0, 0].plot(signalY, color='g', linewidth=0.75, label='signal Y')
axs[0, 0].set_title('Generated signals')
axs[0, 0].set_xlabel='time', ylabel='generated signal')
axs[0, 0].legend()

axs[0, 1].plot(correlationRange, autocorrelation, color='r', linewidth=0.8)
axs[0, 1].set_title('Autocorrelation (signal X)')
axs[0, 1].set_xlabel='t', ylabel='correlation')

axs[1, 0].plot(correlationRange, correlation, linewidth=0.8)
axs[1, 0].set_title('Cross-correlation')
axs[1, 0].set_xlabel='t', ylabel='correlation')

axs[1, 1].plot(harmonics, time)

```

```

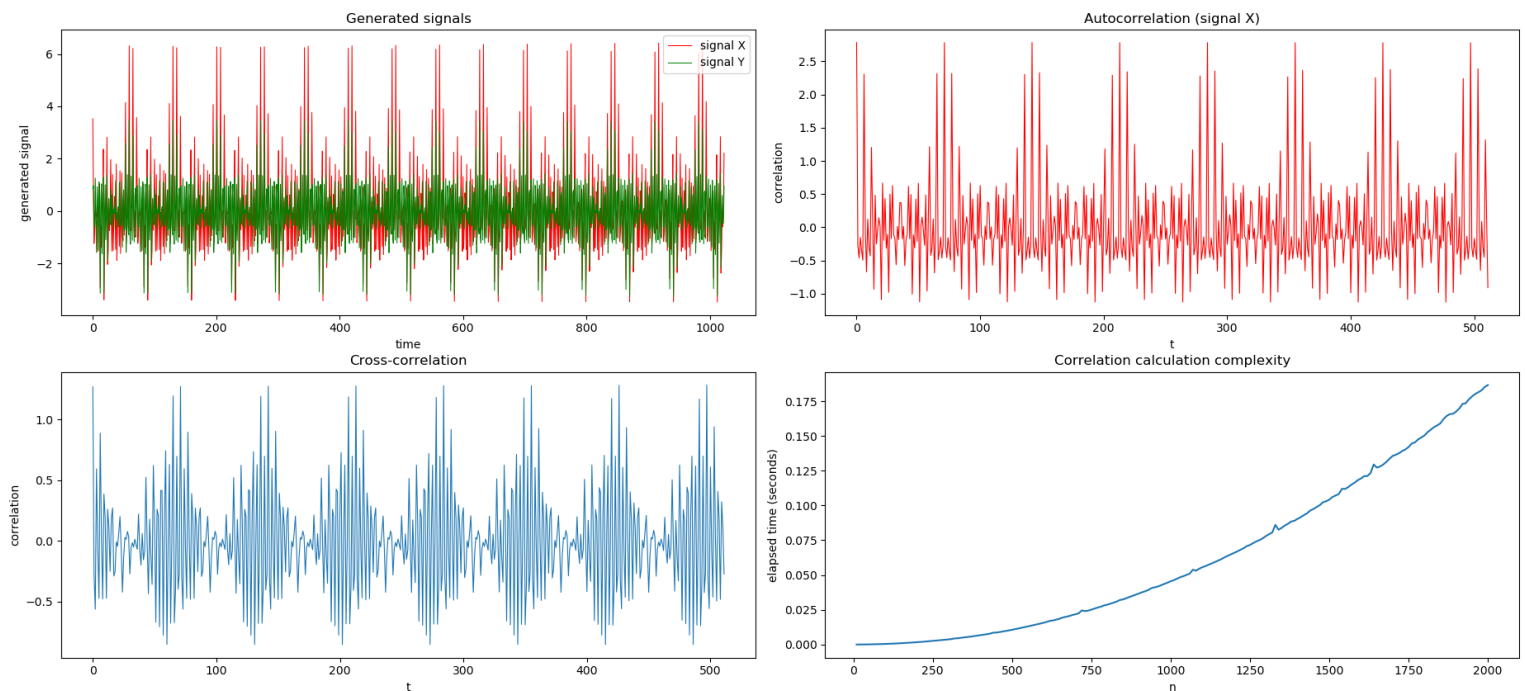
axs[1, 1].set_title('Correlation calculation complexity')
axs[1, 1].set(xlabel='n', ylabel='elapsed time (seconds)')

plt.show()
fig.savefig('graphs/lab1-2.png')

```

Результати виконання кожної програми

Графіки згенерованого сигналу та його копії, залежності автокореляції сигналу X або взаємної кореляції сигналу X та Y від інтервалу τ та складності обчислення кореляції:



Висновки щодо виконання лабораторної роботи

Під час виконання лабораторної роботи ми ознайомилися з принципами побудови автокореляційної і взаємної кореляційної функцій. У ході роботи було згенеровано два сигнали, на прикладі яких обраховувалися автокореляція та взаємна кореляція. У результаті було отримано 4 графіки: згенеровані сигнали, накладені одне на одного, автокореляції для сигналу X, взаємної кореляції двох сигналів, а також графік складності роботи алгоритму з обчислення кореляції. Під час підрахунку складності поступово збільшувалося значення кількості дискретних відліків для згенерованих сигналів. Таким чином, мету даної лабораторної роботи можна вважати досягнутою.