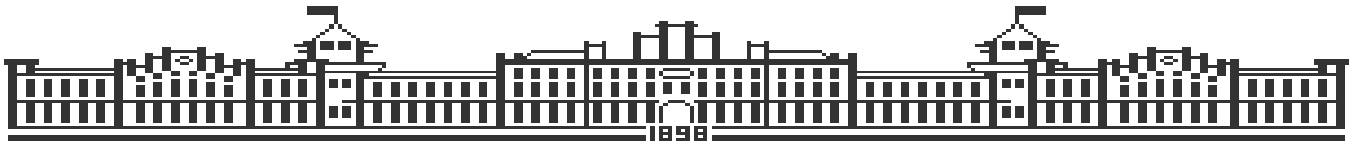


Міністерство освіти і науки України
Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

ЗВІТ

про виконання лабораторної роботи №1.2

з дисципліни

«Інтелектуальні вбудовані системи»

на тему:

«ДОСЛІДЖЕННЯ АВТОКОРЕЛЯЦІЙНОЇ І ВЗАЄМНОЮКОРЕЛЯЦІЙНОЇ ФУНКЦІЙ ВИПАДКОВИХ СИГНАЛІВ»

Перевірив:
асистент кафедри ОТ
Регіда П. Г

ВИКОНАВ:
студент 3 курсу
групи ІП-83, ФІОТ
Мінченко В.Ю.
Залікова книжка №8315
Варіант – 18

Київ 2021

Завдання на лабораторну роботу

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

2.3. Зміст звіт

Варіант 18:

Варіант	Число гармонік в сигналі n	Гранична частота, $\omega_{гр}$	Кількість дискретних відліків, N
18	10	1500	256

Програмний код:

Complexity_generator.py

```
import time as t
import signal_generator


def calculate_time(signal_harmonics, start_frequency, number_discrete_calls):
    time = []
    for n in range(number_discrete_calls):
        start_time = t.time()
        signal_generator.generate_signal(signal_harmonics, start_frequency, n)
        end_time = t.time()
        time.append(end_time - start_time)
    return time
```

signal_generator.py

```
import random
import numpy as np


def generate_signal(signal_harmonics, frequency, discrete_calls):
    signals = np.zeros(discrete_calls)
    for i in range(signal_harmonics):
        frequency_step = frequency / signal_harmonics * (i + 1)
        amplitude = random.random()
        phase = random.random()
        for t in range(discrete_calls):
            signals[t] += amplitude * np.sin(frequency_step * t + phase)
    return signals
```

statistics_utils.py

```python
import numpy as np


def math_expectation(data):
    return np.mean(data)


def math_variance(data):
    return np.var(data)

def cross_correlation(signal1_data, signal2_data):
    result = np.correlate(signal1_data, signal2_data, mode='same')
    return result


def auto_correlation(signal_data):
    result = np.correlate(signal_data, signal_data, mode='full')
    return result[result.size // 2:]
```

Lab1_2.py

```python
import matplotlib.pyplot as plt
import signal_generator
import complexity_generator
import statistics_utils

HARMONICS = 10
FREQUENCY = 1500
DISCRETE_CALLS = 256

signal_first = signal_generator.generate_signal(
  HARMONICS,
  FREQUENCY,
  DISCRETE_CALLS
)
signal_second = signal_generator.generate_signal(
  HARMONICS,
  FREQUENCY,
  DISCRETE_CALLS
)

auto_correlation_signal = statistics_utils.auto_correlation(signal_first)
cross_correlation_signal = statistics_utils.cross_correlation(signal_first,
signal_second)

fig, (((ax00), (ax01)), ((ax10), (ax11))) = plt.subplots(2, 2)
fig.suptitle('Laboratorka 1.2')
fig.set_size_inches(18.5, 10.5)

ax00.plot(signal_first)
ax00.set_title('Generate first signal')
ax00.set(xlabel='time', ylabel='generated signal')

ax01.plot(signal_second)
ax01.set_title('Generate second signal')
ax01.set(xlabel='time', ylabel='generated signal')

ax10.plot(auto_correlation_signal)
ax10.set_title('Autocorrelation of first signal')
ax10.set(xlabel='tau', ylabel='correlation')
```

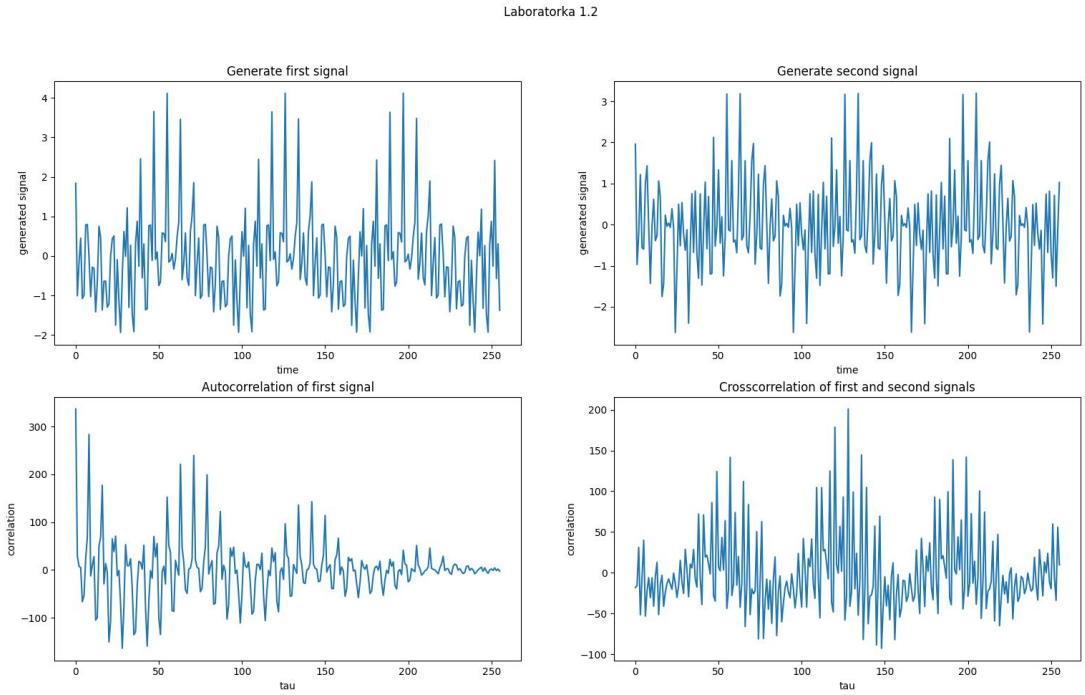
```
ax11.plot(cross_correlation_signal)
ax11.set_title('Crosscorrelation of first and second signals')
ax11.set(xlabel='tau', ylabel='correlation')

fig.savefig('lab1_2.png')

plt.show()
```

Результати роботи програми:

Графіки згенерованого сигналу та складності алгоритму обчислень:



Висновки:

Отже, в ході лабораторної роботи, ми вдосконалили практичні вміння генерувати випадковий сигнал згідно формули та вхідних значень за допомогою власноруч написаної програми, а також обраховували автокореляції та взаємну кореляцію відповідного одного і двох сигналів. Результати наведено в звіті та в репозиторії. Кінцеву мету було досягнуто.