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 ВЗАЄМНОКОРЕЛЯЦІЙНОЇ ФУНКЦІЙ ВИПАДКОВИХ СИГНАЛІВ»

|  |  |  |
| --- | --- | --- |
|  |  |  |

Виконав: Перевірив:

студент групи ІП-83 викладач  
Фібрук Руслан Сергійович Регіда Павло Геннадійович  
номер залікової книжки: 8320

Київ 2021

**Основні теоретичні відомості**

Значення автокореляційної функції фізично представляє зв'язок між значенням однієї і тієї ж величини, тобто для конкретних моментів k t , s τ , значення R (t,τ ) xx оцінюється друге змішаним центральним моментом 2-х перетинів випадкових процесів x(tk), x(tk + τ). Дуже важливим виявляється не тільки обчислення автокореляційної функції (τ ) Rxx , але і обчислення взаємної кореляційної функції (τ ) Rxy для двох випадкових процесів x(y), y(t), для якої не можна на основі зовнішнього спостереження сказати, чи є залежність між ними. τ - випробувальний інтервал, на конкретному значенні якого досліджується взаємний вплив.

**Завдання на лабораторну роботу**

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 взаімнокорреляціонную функцію для 2-х сигналів. Розробити відповідну програму і вивести отримані значення і графіки відповідних параметрів.

Варiант**:** 20

Число гармонік в сигналі: 6

Гранична частота: 1700

Кількість дискретних відліків: 1024

**Лістинг програми**

import numpy as np

import matplotlib.pyplot as plt

HARMONICS\_COUNT = 6

MAX\_FREQUENCY = 1700

DISCRETE\_TIMES\_COUNT = 1024

MAX\_INTERVAL = 256

def rand\_sig(harmonics\_count, max\_freq, discr\_times\_count):

sig = np.zeros(discr\_times\_count)

freq\_start = max\_freq / harmonics\_count

for harmonic\_index in range(harmonics\_count):

amplitude = np.random.uniform(0.0, 1000.0)

phase = np.random.uniform(-np.pi / 2, np.pi / 2)

freq = freq\_start \* (harmonic\_index + 1)

for time in range(discr\_times\_count):

sig[time] += amplitude \* np.sin(freq \* time + phase)

return sig

def math\_expectation(sig):

sum = 0

for i in range(len(sig)):

sum += sig[i]

return sum / len(sig)

def dispersion(sig):

math\_exp = math\_expectation(sig)

sum = 0

for i in range(len(sig)):

sum += (sig[i] - math\_exp) \*\* 2

return sum / (len(sig) - 1)

def correlation(sig1, sig2, interval):

M1 = math\_expectation(sig1)

M2 = math\_expectation(sig2)

sum = 0

for time in range(len(sig1) - interval):

sum += (sig1[time] - M1) \* (sig2[time + interval] - M2)

return sum / (len(sig1) - 1)

def autocorrelation(sig, interval):

return correlation(sig, sig, interval)

sig1 = rand\_sig(HARMONICS\_COUNT, MAX\_FREQUENCY, DISCRETE\_TIMES\_COUNT)

sig2 = rand\_sig(HARMONICS\_COUNT, MAX\_FREQUENCY, DISCRETE\_TIMES\_COUNT)

autocorrelation\_values = np.zeros(MAX\_INTERVAL)

crosscorrelation\_values = np.zeros(MAX\_INTERVAL)

for interval in range(MAX\_INTERVAL):

autocorrelation\_values[interval] = autocorrelation(sig1, interval)

crosscorrelation\_values[interval] = correlation(sig1, sig2, interval)

plt.plot(range(MAX\_INTERVAL), autocorrelation\_values)

plt.xlabel("Interval")

plt.ylabel("Autocorrelation value")

plt.show()

plt.plot(range(MAX\_INTERVAL), crosscorrelation\_values)

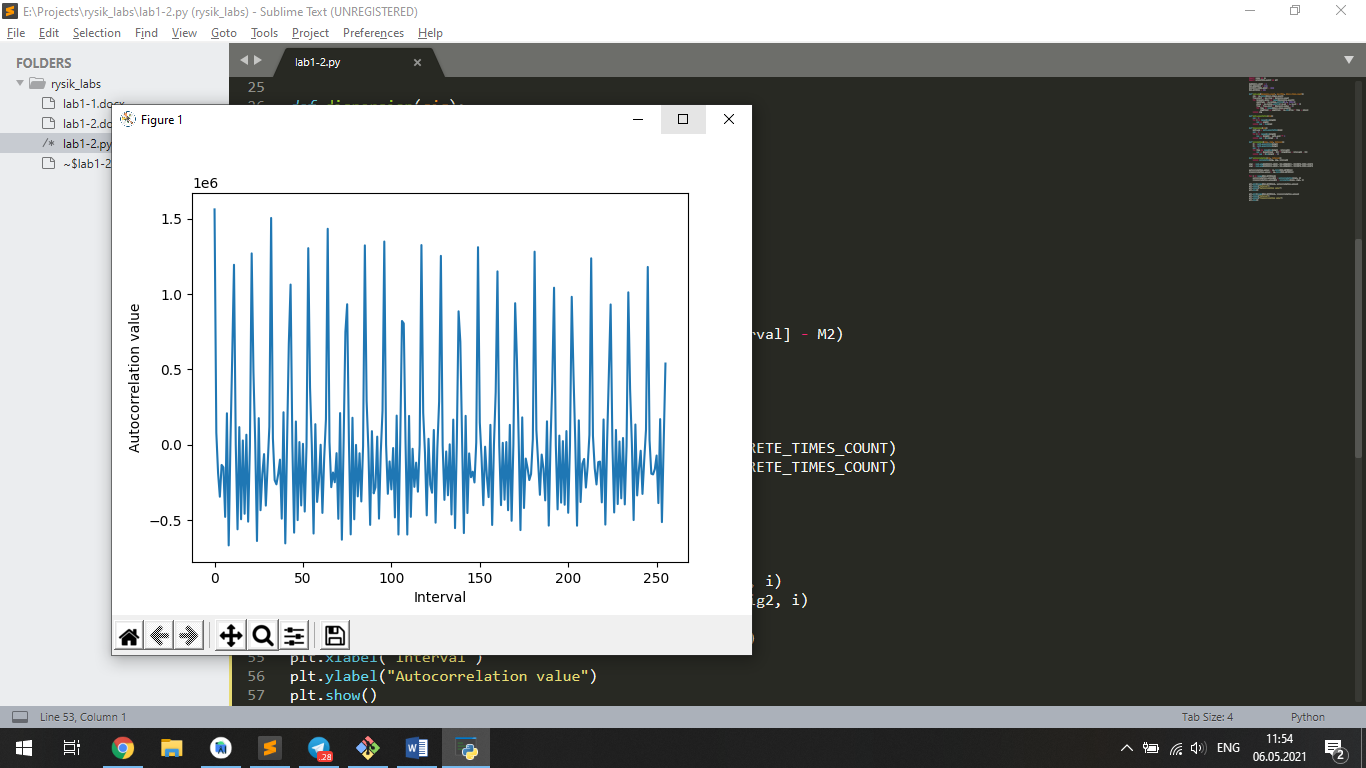
plt.xlabel("Interval")

plt.ylabel("Crosscorrelation value")

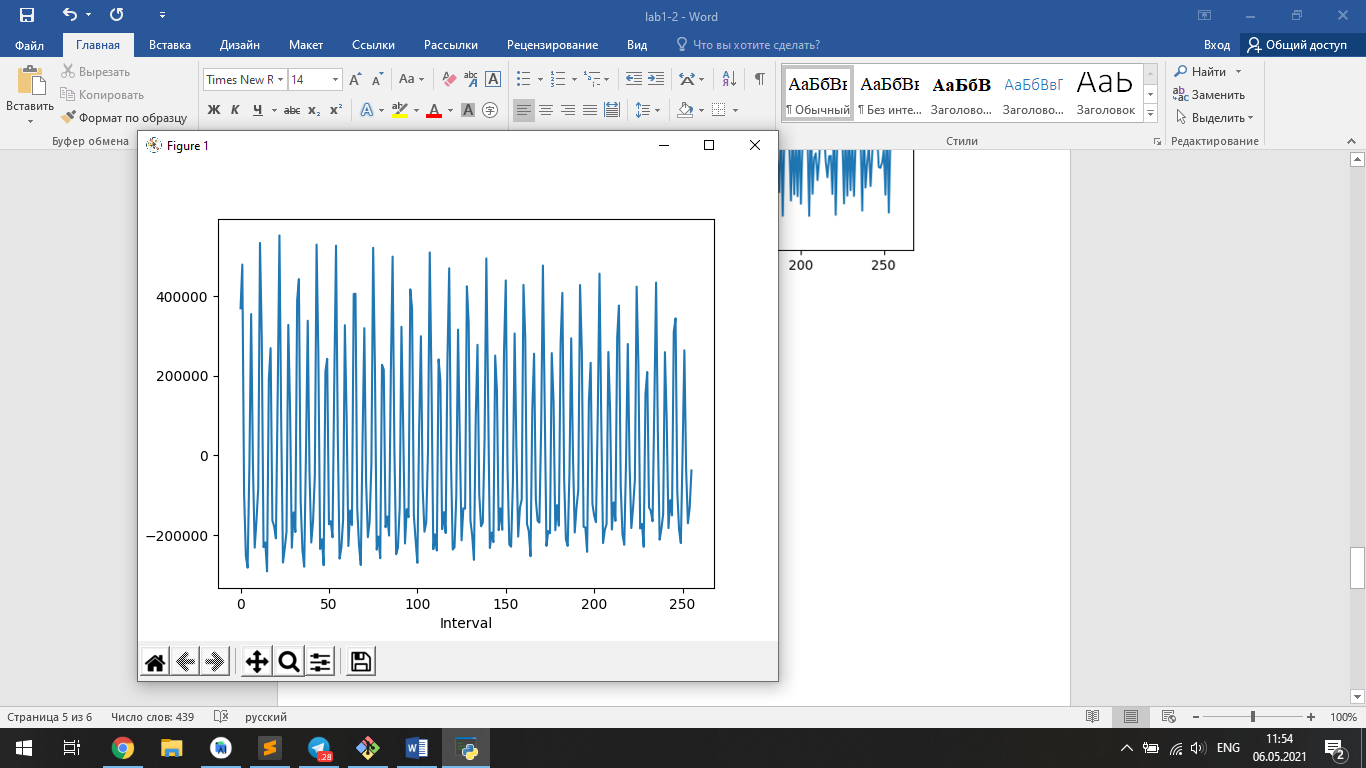
plt.show()

**Результати роботи програми**

Автокореляція:



Взаємна кореляція:



**Висновки**

При виконанні цієї лабораторної роботи ми навчилися обчислювати значення автокореляційної функції для згенерованого випадкового сигналу та взаємнокореляційної – для двох сигналів.