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:

студент групи ІП-83
Фібрук Руслан Сергійович
номер залікової книжки: 8320

Перевірив:

викладач
Регіда Павло Геннадійович

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.

Варіант**:** 20

Число гармонік в сигналі: 6

Гранична частота: 1700

Кількість дискретних відліків: 1024

Лістинг програми

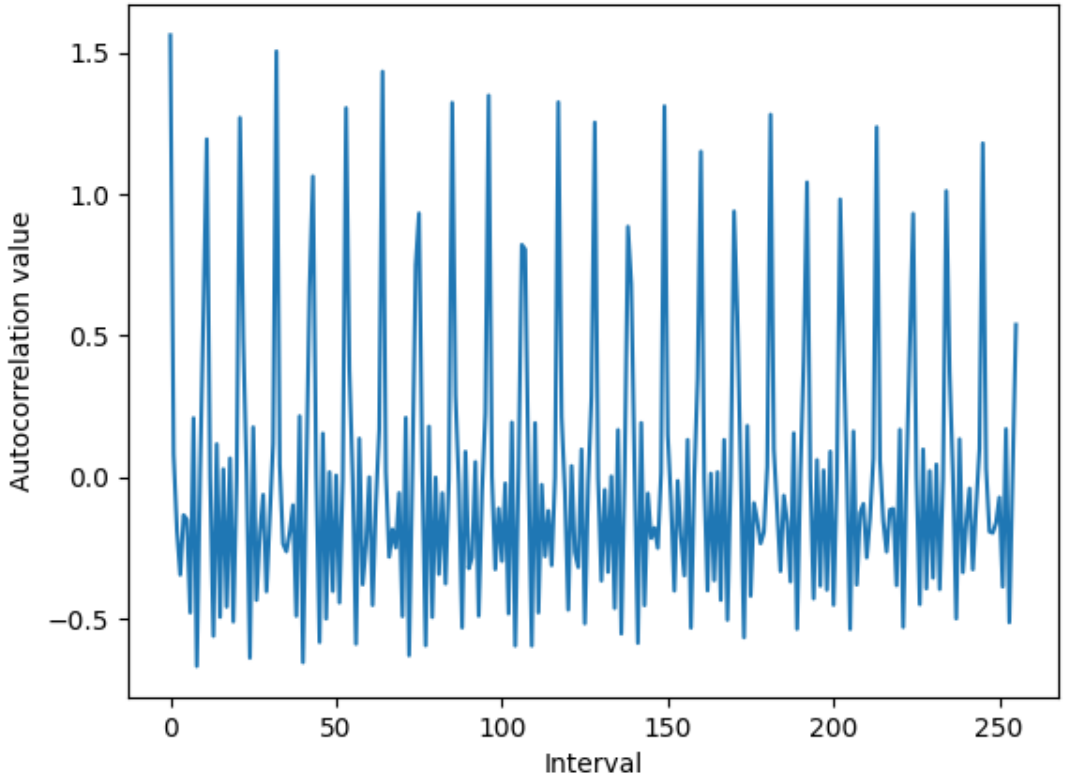
```
import numpy as np

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
```

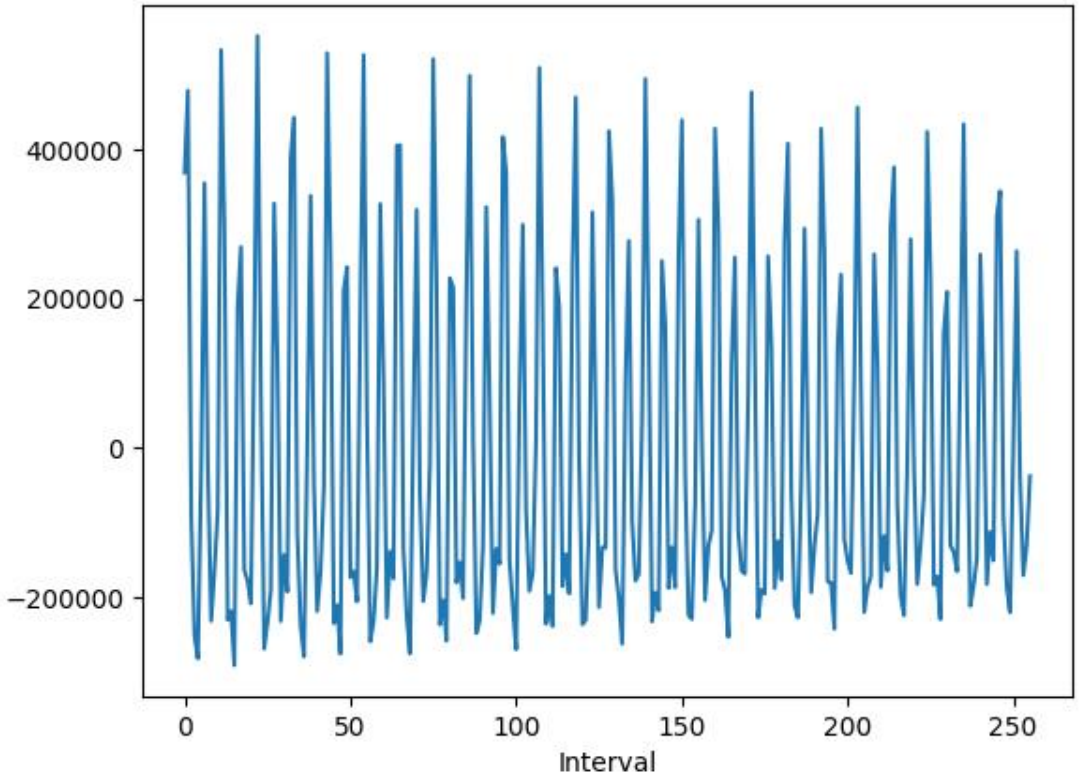
```
    for time in range(len(sig1) - interval):
        sum += (sig1[time] - M1) * (sig2[time + interval] - M2)
    return sum / (len(sig1) - 1)

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.