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 взаємноюкореляційної функцій випадкових сигналів»

Виконав:
Тимофеєнко П.В.

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

Теоретичні відомості

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 $x(t_k), x(t_k+\tau_s)$

$$R_{xx}(t,\tau_s) = \lim_{N\to\infty} \frac{1}{N-1}\sum_{i=1}^{N} \overbrace{(x_i(t_k) - M_x(t_k))}^{\overset{0}{x}(t_k)} \cdot \overbrace{(x_i(t_k+\tau_s) - M_x(t_k+\tau_s))}^{\overset{0}{x}(t_k+\tau_s)}$$

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

Центральні значення можна замінити:

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

$$\left[\begin{aligned} R_{xx}(t,\tau) &= \lim_{N\to\infty} \frac{1}{N-1}\sum_{i=1}^{N} \overset{0}{x}_i(t)\cdot \overset{0}{x}_i(t+\tau) \\ R_{xx}(t,\tau) &= \lim_{N\to\infty} \frac{1}{N-1}\sum_{i=1}^{N} \overset{0}{x}_i(t)\cdot \overset{0}{x}_i(t+\tau) \end{aligned}\right]$$

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:

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

У завданнях управління частіше використовується нормована кореляційна функція:

$$S_{xx}(t,\tau) = \frac{R_{xx}(t,\tau)}{D_x(t)} < 1$$

Дослідження нестандартних випадкових сигналів вимагає значних обсягів пам'яті, тому в більшості наукових досліджень приймається гіпотеза про стаціонарності випадкового сигналу на інтервалі ($t_0 \ldots t_1$).

Кореляційна функція для стаціонарного сигналу:

$$R_x(\tau_s) = \lim_{N\to 0} \cdot \frac{1}{N-1} \cdot \sum_{i=1}^{N} \underbrace{(x_i(t_k) - M_x)}_{X(t_k)} \cdot \underbrace{(x_i(t_k + \tau_s) - M_x)}_{x(t_s)} =$$

$$= \lim_{n\to 0} \cdot \frac{1}{n-1} \cdot (x_i(t_k) - M_x) \cdot (x_i(t_k + \tau_s) - M_x)$$

$x(t)$ в межах однієї реалізації показує наскільки швидко змінюється сигнал.

Коваріаційна функція для стаціонарного сигналу:

$$C_{xx}(\tau) = \lim_{N\to 0} \cdot \frac{1}{n-1} \cdot \sum_{k=1}^{n} Lx(t_k) \cdot x(t_k + \tau)$$

показує ступінь зв'язності між значеннями одного і того ж сигналу.

Таким чином для стаціонарних і ергодичні процесів обчислення параметрів сигналів реалізуються шляхом усереднення за часом у межах однієї реалізації.

Статистичне вимірювання зв'язків між двома стаціонарними випадковими процесами

Дуже важливим виявляється не тільки обчислення автокореляційної функції $R_{xx}(\tau)$, але і обчислення взаємної кореляційної функції $R_{xy}(\tau)$ для двох випадкових процесів $x(y)$, $y(t)$, для якої не можна на основі зовнішнього спостереження сказати, чи є залежність між ними. Для розрахунку взаємної кореляційної функції:

$$R_{xy}(\tau) = \lim_{n\to 0} \cdot \frac{1}{n-1} \cdot \sum_{i=1}^{n} \underbrace{(x_i(t_k) - M_x)}_{X(t_k)} \cdot \underbrace{(y(t_k + \tau) - M_y)}_{y(t_k - \tau)} =$$

τ - випробувальний інтервал, на конкретному значенні якого досліджується взаємний вплив.

Завдання за варіантом(23)

Число гармонік в сигналі n	Гранична частота	Кількість дискретних відліків, N
8	1500	1024

Лістинг

Index_correlation.py

```
import matplotlib.pyplot as plt
import numpy as np
```

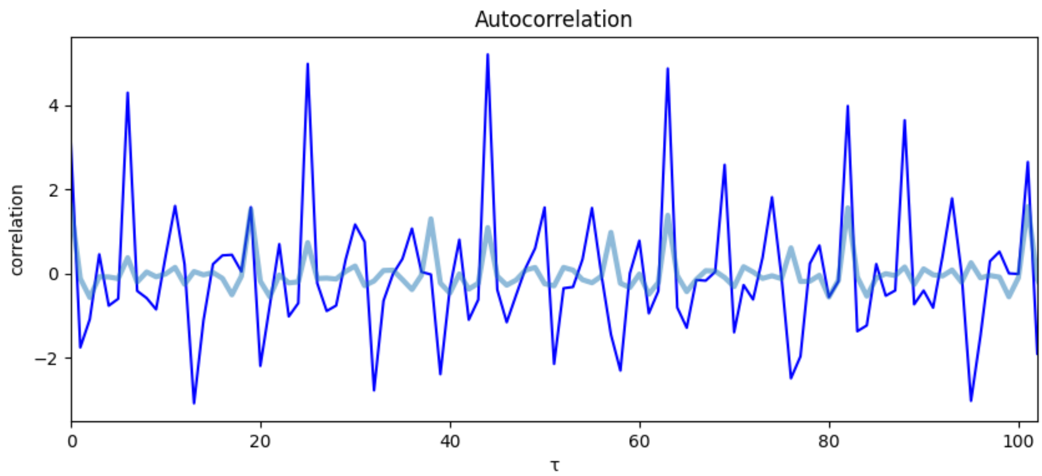
```python
import rsg
import correlation as cor

harmonics_count = 8
frequency = 1500
N = 1024

signal = rsg.generateSignal(harmonics_count,frequency,N)
signal_copy = rsg.generateSignal(harmonics_count,frequency,N)

fig1, ax1 = plt.subplots()
fig1.set_size_inches(10,4)

ax1.plot(list(range(N)), signal, c = 'b', label="signal")
ax1.plot(list(range(int(N/2))), cor.autocorrelation(signal),
         linewidth=3, label="correlation", alpha=0.5)

ax1.set_xlim(0, int(N/10))
ax1.set(ylabel='correlation', xlabel='τ',
        title='Autocorrelation')

##########
fig2, ax2 = plt.subplots()
fig2.set_size_inches(10,4)


ax2.plot(list(range(N)), signal, c = 'b', label="signal")
ax2.plot(list(range(N)), signal_copy, c = 'g', label="signal")
ax2.plot(list(range(int(N/2))), cor.correlation(signal,signal_copy),
         linewidth=3, label="correlation", alpha=0.5)

ax2.set_xlim(0, int(N/10))
ax2.set(ylabel='correlation', xlabel='τ',
        title='Cross-correlation')

fig1.savefig("autocorrelation.png")
fig2.savefig("crosscorrelation.png")
plt.show()
```

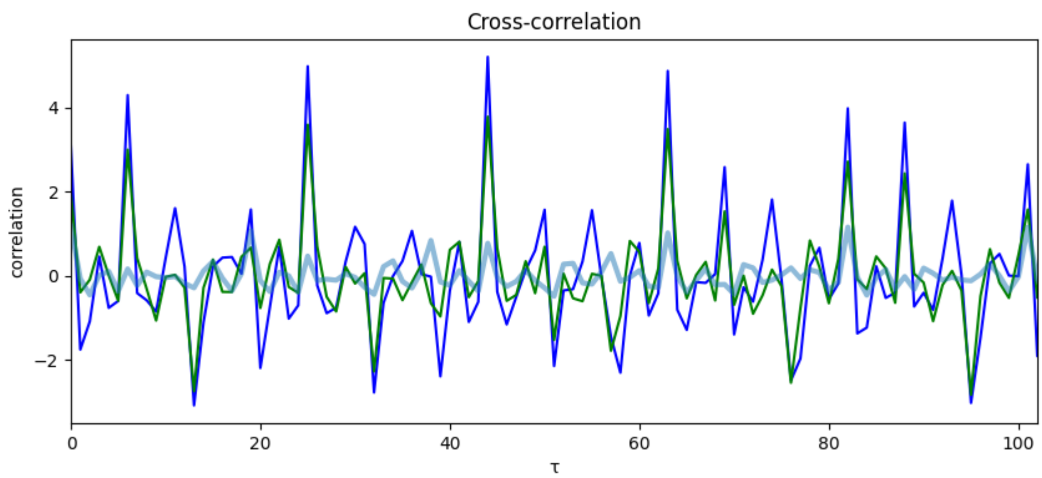
correlation.py

```python
import numpy as np
import math

def correlation(signal1,signal2):
    result = []

    m1 = np.mean(signal1)
    m2 = np.mean(signal2)
```

```python
    d1 = np.std(signal1)
    d2 = np.std(signal2)

    l = len(signal1) // 2

    for tau in range(l):
        cov = 0

        for i in range(l):
            cov += (signal1[i]-m1)*(signal2[i + tau]-m2) / (l-1)

        result.append(cov / (math.sqrt(d1) * math.sqrt(d2)))

    return result


def autocorrelation(signal1):
    return correlation(signal1,signal1)
```

Результат роботи програми





Висновок

У ході даної лаб. роботи ознайомився з принципами побудови автокорелляціонной і взаємної кореляційної функцій, вивчив та дослідив їх основні параметри з використанням засобів Python.