

**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 №1.1**  
з дисципліни  
«Інтелектуальні вбудовані системи»  
на тему  
«Дослідження і розробка моделей випадкових сигналів.  
Аналіз їх характеристик»

Виконав:  
студент групи ІП-84  
Гудь В.В.  
№ залікової книжки: ІП-8405

Перевірив:  
викладач  
Регіда П.Г.

Київ 2021

## Теоретичні відомості

СРЧ обов'язково пов'язані з деякою зовнішнім середовищем. СРЧ забезпечує контроль за зміною параметрів зовнішнього середовища і в ряді випадків забезпечує управління параметрами середовища через деякі впливу на неї. Параметри середовища представляються деякою зміною фізичного середовища. При вимірах фізичного параметра ми отримуємо певний електричний сигнал на вході вимірювального датчика. Для подання такого електричного сигналу можна використовувати різні моделі. Найкращою моделлю досліджуваного сигналу є відповідна математична інтерпретація випадкового процесу. Випадковий сигнал або процес завжди представляється деякою функцією часу  $x(t)$ , значення якої не можна передбачити з точністю засобів вимірювання або обчислень, які б кошти моделі ми не використовували.

Для випадкового процесу його значення можна передбачити лише основні його характеристики: математичне сподівання  $M x(t)$ , дисперсію  $D x(t)$ , автокореляційну функцію  $R_{xx}(t, \tau)$ ,  $R_{xy}(t, \tau)$ .

Ці характеристики для випадкового нестационарного процесу теж є функціями часу, але вони детерміновані. Для оцінки цих характеристик використовуються СРВ, які повинні обробити значну кількість інформації; для отримання їх при нестационарному процесі необхідно мати безліч реалізацій цього процесу.

### Умови завдання

Варіант 5:

$n = 14$ ,  $\omega_{gr} = 2000$ ,  $N = 264$

### Вихідний код

```
import numpy as np
import matplotlib.pyplot as plotter
n = 14
w = 2000
N = 264

def generate_signal(amplitude, phase, frequency, time):
    return amplitude * np.sin(frequency * time + phase) # Press Ctrl+F8 to toggle the breakpoint.

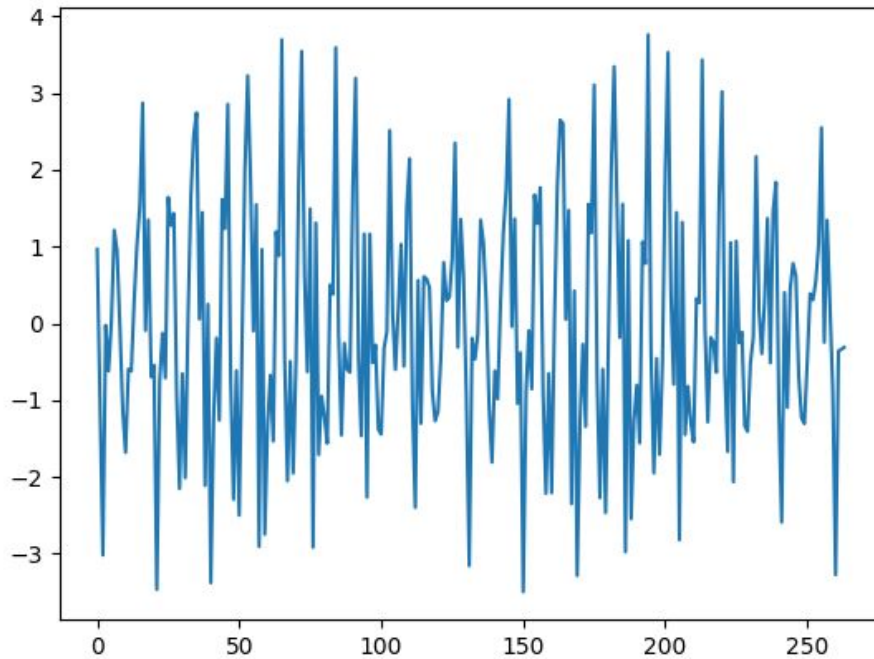
def signal_generator(harmonics, frequency_max, samples_amount):
    signal_collection = np.zeros(samples_amount)
    for f in range(1, harmonics + 1):
        phase = np.random.uniform((-np.pi / 2), (np.pi / 2))
        amplitude = np.random.uniform(0, 1)
        frequency = (f * frequency_max) / harmonics
        for time in range(samples_amount):
            signal_collection[time] += generate_signal(amplitude, phase, frequency, time)
    return signal_collection

signal = signal_generator(n, w, N)
```

```
plotter.plot(signal)
plotter.show()
```

```
print('Expected Value is', np.average(signal))
print('Variance is', np.var(signal))
```

### Результати роботи програми



```
Expected Value is -0.0162814184747356
Variance is 2.3971154211809984
```

### Висновки

У ході виконання лабораторної роботи ми ознайомилися з принципами генерації випадкових сигналів, вивчили та дослідили їх основні параметри з використанням засобів моделювання та сучасних програмних оболонок.