МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №2.1
з дисципліни «Інтелектуальні вбудовані системи»
на тему «ДОСЛІДЖЕННЯ ПАРАМЕТРІВ АЛГОРИТМУ ДИСКРЕТНОГО ПЕРЕТВОРЕННЯ ФУР'Є »

Виконав:
студент гр. ІП-84
Дмитренко Олександр

Перевірив:
Регіда П.Г.

Київ 2021

Основні теоретичні відомості

В основі спектрального аналізу використовується реалізація так званого дискретного перетворювача Фур'є (ДПФ) з неформальним (не формульним) поданням сигналів, тобто досліджувані сигнали представляються послідовністю відліків $x(k)$

$$F_x(p) = \sum_{k=0}^{N-1} x(k) \cdot e^{-jk\Delta tp\Delta\omega}$$

$$\omega \to \omega_p \to p\Delta\omega \to p \qquad \Delta\omega = \frac{2\pi}{T}$$

На всьому інтервалі подання сигналів **T**, $\mathbf{2\pi}$ - один період низьких частот. Щоб підвищити точність треба збільшити інтервал **T.**

$$t \to t_k \to k\Delta t \to k\,; \qquad \Delta t = \frac{T}{N} = \frac{1}{k_{\text{зап}}.} \cdot f'\text{гр}.$$

ДПФ - проста обчислювальна процедура типу звірки (тобто Σ-е парних множень), яка за складністю також має оцінку $\mathbf{N^2 + N}$. Для реалізації ДПФ необхідно реалізувати поворотні коефіцієнти ДПФ:

$$\mathrm{W}_{\mathrm{N}}^{\mathrm{pk}} = \mathrm{e}^{-\mathrm{jk}\Delta\mathrm{t}\Delta\omega\mathrm{p}}$$

Ці поворотні коефіцієнти записуються в ПЗУ, тобто є константами.

$$\mathrm{W}_{\mathrm{N}}^{\mathrm{pk}} = \mathrm{e}^{-\mathrm{jk}\frac{\mathrm{T}}{\mathrm{N}}\mathrm{p}\frac{2\pi}{\mathrm{T}}} = \mathrm{e}^{-\mathrm{j}\frac{2\pi}{\mathrm{N}}\mathrm{pk}}$$

W_N^{pk} не залежать від **T**, а лише від розмірності перетворення **N.** Ці коефіцієнти подаються не в експоненційній формі, а в тригонометричній.

$$W_N^{pk} = \cos\left(\frac{2\pi}{N}pk\right) - j\sin\left(\frac{2\pi}{N}pk\right)$$

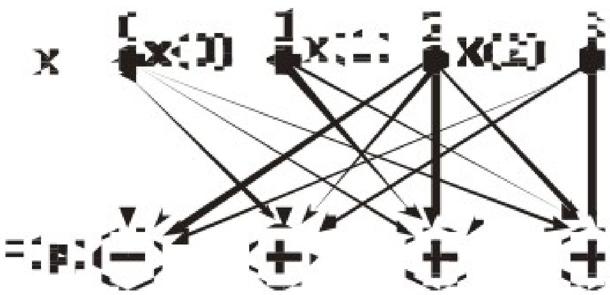
Ці коефіцієнти повторюються (тому і **p** до **N-1**, і **k** до **N-1**, а **(N-1) • (N-1)**) з періодом **N(2π)**.. Т.ч. в ПЗУ треба зберігати N коефіцієнтів дійсних і уявних частин. Якщо винести знак коефіцієнта можна зберігати **N/2** коефіцієнтів.

2π/N- деякий мінімальний кут, на який повертаються ці коефіцієнти. У ПЗУ окремо зберігаються дійсні та уявні частини компілюють коефіцієнтів. Більш загальна форма ДПФ представляється як:

$$F_x(p) = \sum_{k=0}^{N-1} x(k) \cdot W_N^{pk}$$

ДПФ дуже зручно представити у вигляді відповідного графа.

Приклад: граф 4-х точкового ДПФ. ($k = \overline{0,3}$; $p = \overline{0,3}$)



Коефіцієнти зручно представити у вигляді таблиці:

p / k	0	1	2	3
0	W_4^0	W_4^0	W_4^0	W_4^0
1	W_4^0	W_4^1	W_4^2	W_4^3
2	W_4^0	W_4^2	W_4^0	W_4^2
3	W_4^0	W_4^3	W_4^2	W_4^1

Завдання

Для згенерованого випадкового сигналу з Лабораторної роботи N 1 відповідно до заданого варіантом (Додаток 1) побудувати його спектр, використовуючи процедуру дискретного перетворення Фур'є. Розробити відповідну програму і вивести отримані значення і графіки відповідних параметрів

Варіант

Номер залікової книжки - **8507**
Варіант в таблиці — **7**

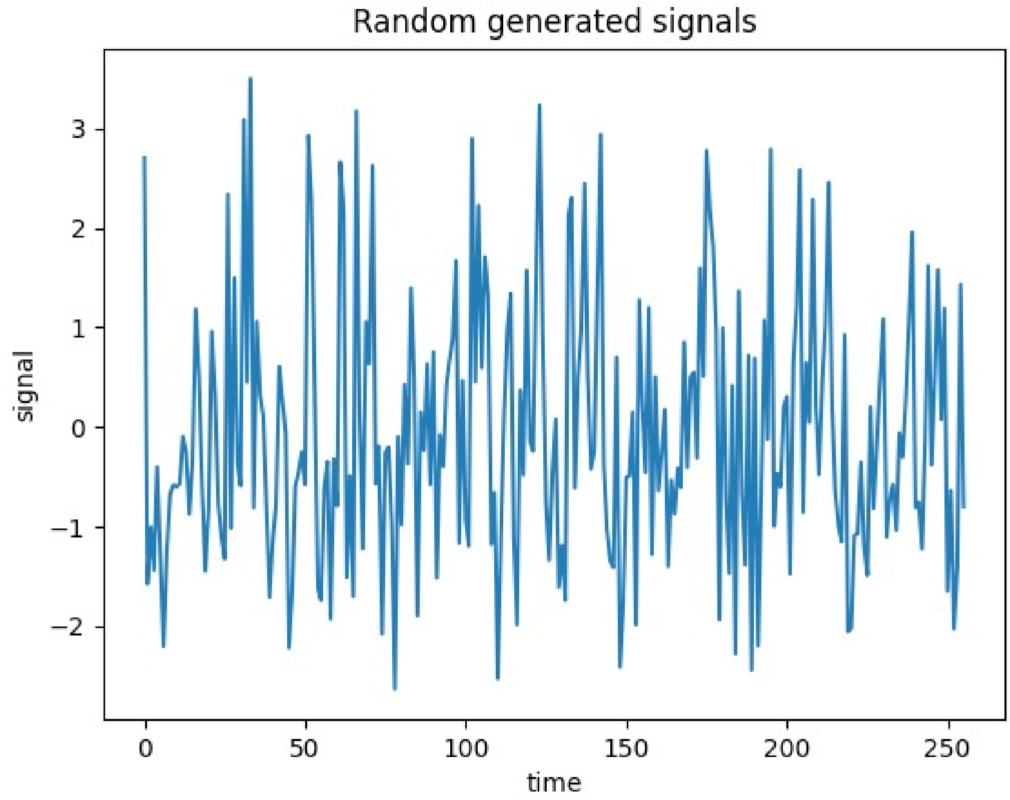
Число гармонік в сигналі n = 10
Гранична частота ω гр = 2700
Кількість дискретних відліків N = 256

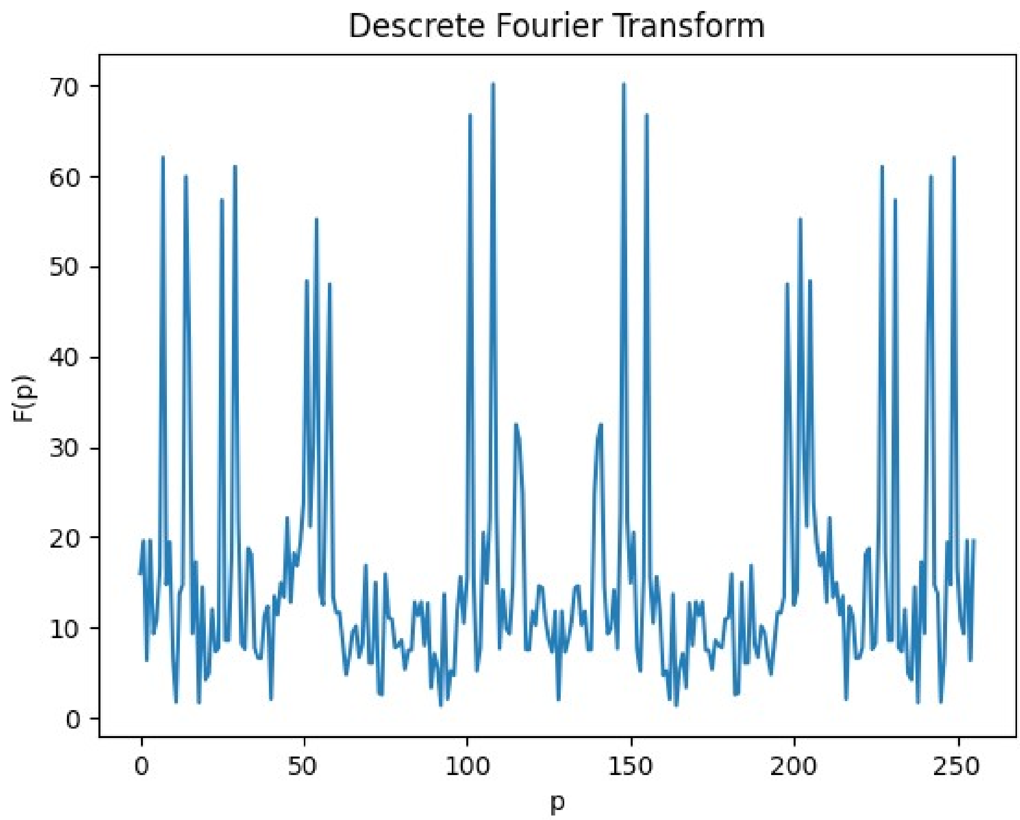
Лістинг програми

```
import numpy as np      # math operations
import matplotlib.pyplot as plt     #graphs

n = 10      # harmonics
w = 2700    #frequency
N = 256     # discrete calls

def signalsGenerator(n,w,N):
    signals = np.zeros(N)
    W = w / n
    for _ in range(n):
        for t in range(N):
            amplitude = np.random.rand()
            phase = np.random.rand()
            signals[t] += (amplitude * np.sin(W * t + phase))
        W += W
    return signals
```

```python
#  Discrete Fourier Transform Coefficient
def fCoeff(pk, N):
    exp = 2*np.pi*pk/N
    return complex(np.cos(exp), -np.sin(exp))


# Discrete Fourier Transform
def fTransform(signals):
    N = len(signals)
    spect=[]
    for p in range (N):
        sum = 0
        for k in range(N):
            sum+= signals[k] * fCoeff(p*k, N)
        spect.append(abs(sum))

    return spect


signal  = signalsGenerator(n,w,N)


plt.plot(signal)
plt.title('Random generated signals')
plt.xlabel('time')
plt.ylabel('signal')
plt.figure()

plt.plot(fTransform(signal))
plt.title('Descrete Fourier Transform')
plt.xlabel('p')
plt.ylabel('F(p)')
plt.show()
```

Результат роботи програми





Висновки

У ході виконання лабораторної роботи було ознайомлено з принципами реалізації спектрального аналізу випадкових сигналів на основі алгоритму перетворення Фур’є. Програма була реалізована на мові Python. В свою чергу вона обчислює спектр за допомогою дискретного перетворення Фур’є та в кінці виводить його графік.