Міністерство освіти і науки України Національний технічний університет України “Київський політехнічний інститут ім. Ігоря Сікорського” Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

Лабораторна робота 2.2
з дисципліни «Інтелектуальні вбудовані системи»

Виконав:

студент групи ІП-83

Подаш А.М.

Перевірив:

асистент Регіда П.Г.

Київ 2021

Основні теоретичні відомості

Швидкі алгоритми ПФ отримали назву схеми Кулі-Тьюкі. Всі ці алгоритми використовують регулярність самої процедури ДПФ і те, що будь-який складний коефіцієнт W_N^{pk} можна розкласти на прості комплексні коефіцієнти.

$$W_N^{pk} = W_N^1 W_N^2 W_N^3$$

Для стану таких груп коефіцієнтів процедура ДПФ повинна стати багаторівневою, не порушуючи загальних функціональних зв'язків графа процедури ДПФ.

Існують формальні підходи для отримання регулярних графів ДПФ. Всі отримані алгоритми поділяються на 2 класи:

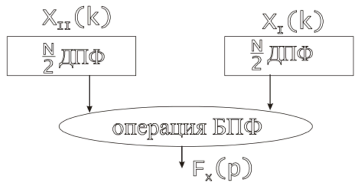
1) На основі реалізації принципу зріджені за часом $\mathbf{X_к}$

2)на основі реалізації принципу зріджені відліків шуканого спектру **F(p).**

Найпростіший принцип зріджені - поділу на парні/непарні пів-послідовності, які потім обробляють паралельно. А потім знаходять алгоритм, як отримати шуканий спектр.

Якщо нам вдасться ефективно розділити, а потім алгоритм отримання спектра, то ми можемо перейти від **N** ДПФ до **N/2** ДПФ.

$$X(k) \to X_{II}(k), \quad X(k) \to X_I(k)$$



Розглянемо формальний висновок алгоритму ШПФ, який реалізує в одноразовому застосуванні принцип проріджування по часу:

$$F_x(p) = \sum_{k=0}^{N-1} X(k) W_N^{pk} = \sum_{k=0}^{N-2} X_{II}(k) W_N^{pk} + \sum_{k=1}^{N-2} X_I(k) W_N^{pk}$$

$$X_{II}(k) \to X(2k^*)\,;\; X_I(k) \to X(2k^*+1)\,;\; k^* = 0; \frac{N}{2} - 1$$

$$F_x(p) = \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*) W_N^{pk} + \sum_{k^*=0}^{\frac{N}{2}-1} X(2k^*+1) W_N^{p(2k^*+1)}$$

$$W_N^{p2k^*} = e^{-j\frac{2\pi}{N}p2k^*} = e^{-j\frac{2\pi}{N/2}pk^*} = W_{\frac{N}{2}}^{pk^*}$$

У цій першій сумі з'явилися коефіцієнти в 2 рази менше.

У другій сумі з'явився множник, який не залежить від k * тобто він може бути винесений за знак суми.

Варіант №18

Число гармонік в сигналі n - 10

Гранична частота, ωгр - 1500

Кількість дискретних відліків, N - 256

Завдання на лабораторну роботу

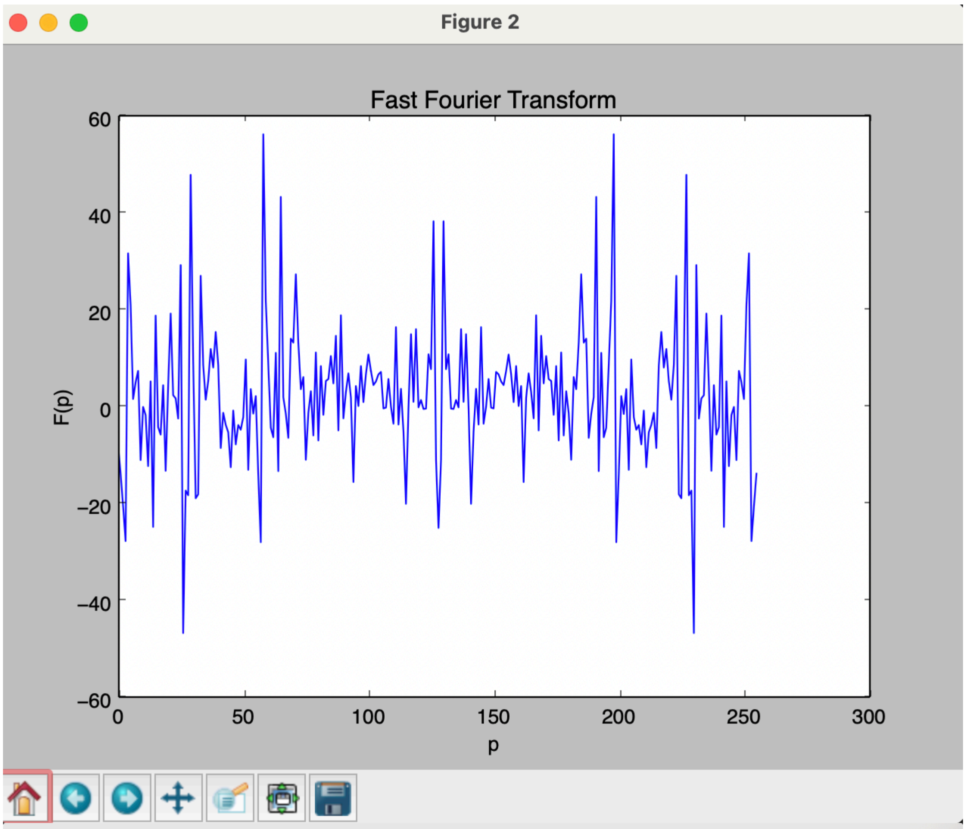
Для згенерованого випадкового сигналу з Лабораторної роботи N 1 відповідно до заданого варіантом (Додаток 1) побудувати його спектр, використовуючи процедуру швидкого перетворення Фур'є з проріджуванням відліків сигналу за часом. Розробити відповідну програму і вивести отримані значення і графіки відповідних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

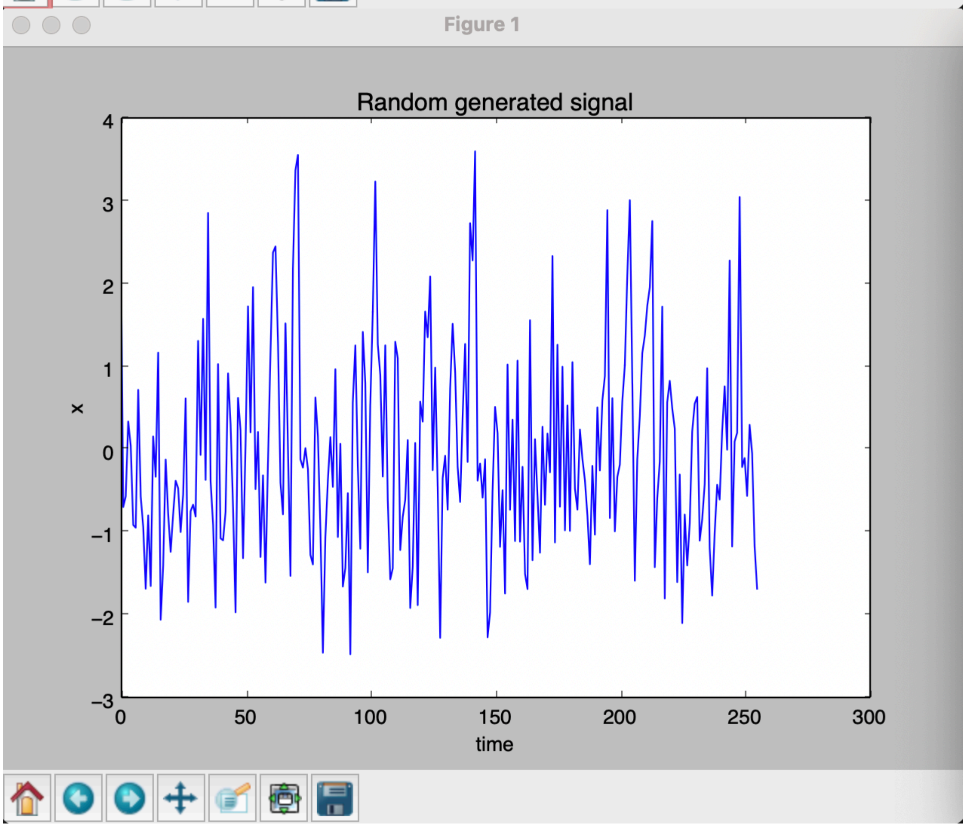
Лістинг програми

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

# constants
n = 10  # number of harmonics
w = 1500  # max frequency
N = 256  # number of descrete calls


# function for calculating random signal
def formula(a, w, t, phi):
    return a*np.sin(w*t+phi)

# function for generation array of signals
def generateSignals(n, w, N):
    signals = [0]*N  # array of signals
    w0 = w/n  # frequency
    for _ in range(n):

        for t in range(N):
            a = np.random.rand()  # amplitude
            phi = np.random.rand()  # phase
            signals[t] += formula(a, w0, t, phi)
        w0 += w0
    return signals

def coeff(pk, N):
    exp = 2*math.pi*pk/N
    return complex(math.cos(exp), -math.sin(exp))
```

```python
# function for calculating Discrete Fourier Transform
def fft(signals):
    N = len(signals)
    l = int(N/2)
    spectrum = [0] * N

    if N == 1:
        return signals
    evens = fft(signals[::2])
    odds = fft(signals[1::2])

    for k in range(l):
        spectrum[k] = evens[k] + odds[k] * coeff(k, N)
        spectrum[k + l] = evens[k] - odds[k] * coeff(k, N)

    return spectrum

signals = generateSignals(n, w, N)

# plotting

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

#fft
plt.plot(fft(signals))
plt.xlabel('p')
plt.ylabel('F(p)')
plt.title('Fast Fourier Transform')
plt.show()
```

Результат роботи програми





Висновки

Під час виконання лабораторної роботи я дослідив принципи реалізації спектрального аналізу випадкових сигналів на основі алгоритму швидкого перетворення Фур'є. Було реалізовано програму обчислення спектру згенерованого сигналу на мові Phyton. Програма обчислює спектр за допомогою швидкого перетворення Фур'є і після цього виводить його графік.