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 №2.1

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«ДОСЛІДЖЕННЯ ПАРАМЕТРІВ АЛГОРИТМУ ДИСКРЕТНОГО ПЕРЕТВОРЕННЯ ФУР'Є»

Виконала:	Перевірив:
студентка групи ІП-84	викладач
Скрипник Єлена Сергіївна	Регіда Павло Геннадійович
номер залікової книжки: 8422	

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**, **2π** - один період низьких частот. Щоб підвищити точність треба збільшити інтервал **T.**

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

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:

$$W_N^{pk} = e^{-jk\Delta t\Delta\omega p}$$

Ці поворотні коефіцієнти записуються в ПЗУ, тобто є константами.

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

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

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

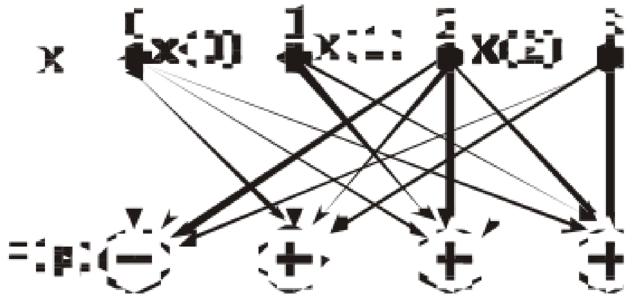
Ці коефіцієнти повторюються (тому і **p** до **N-1**, і **k** до **N-1**, а **(N-1) • (N-1)**) з періодом **N(2π)**.. Т.ч. в ПЗУ треба зберігати N коефіцієнтів дійсних і уявних частин. Якщо винести знак коефіцієнта можна зберігати **N/2** коефіцієнтів.

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:

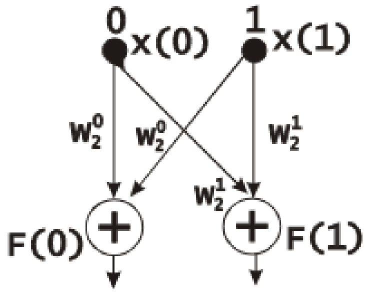
k \ p	0	1	2	3
0	W_4^0	W_4^0	W_4^0	W_4^0
1	W_4^0	W_4^1	W_4^2	W_4^3
2	W_4^0	W_4^2	W_4^0	W_4^2
3	W_4^0	W_4^3	W_4^2	W_4^1

Різних тут всього 4 коефіцієнта:

$$W_4^0 = \cos\left(\frac{2\pi}{4}\cdot 0\right) - j\sin\left(\frac{2\pi}{4}\cdot 0\right) = 1 \qquad (W_4^1 = -j;\ W_4^2 = -1;\ W_4^3 = +j)$$

Можна в пам'яті зберігати тільки 2, а решта брати з "-", якщо $\frac{N}{2}-1<pk$. 4 ДПФ це вироджені перетворення, по модулю ці коефіцієнти = 1 і всі 4 ДПФ можуть реалізуватися на 24-х суматора. Це буде далі використовуватися в реалізації ШПФ з основою 4.

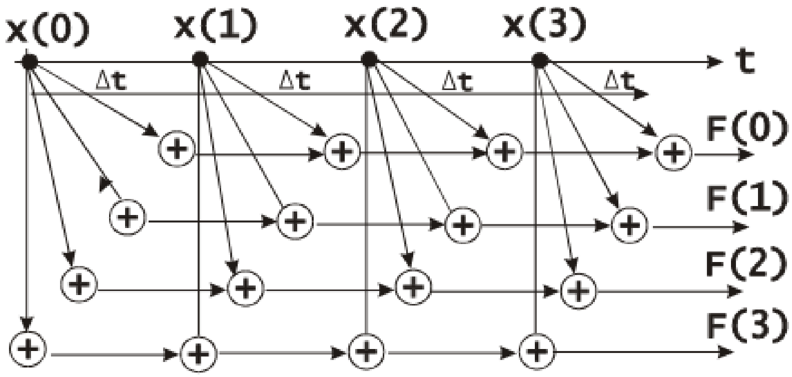
2ДПФ реалізується ще простіше:



($W_2^0=+1$; $W_2^1=-1$)

<u>Спеціальна схема реалізації ДПФ з активним використанням пауз між відліками</u>

При реалізації ДПФ можна організувати обробку в темпі надходження даних. Реалізація схеми в БПФ з активним використанням пауз на 4-х точках виглядає так:



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:

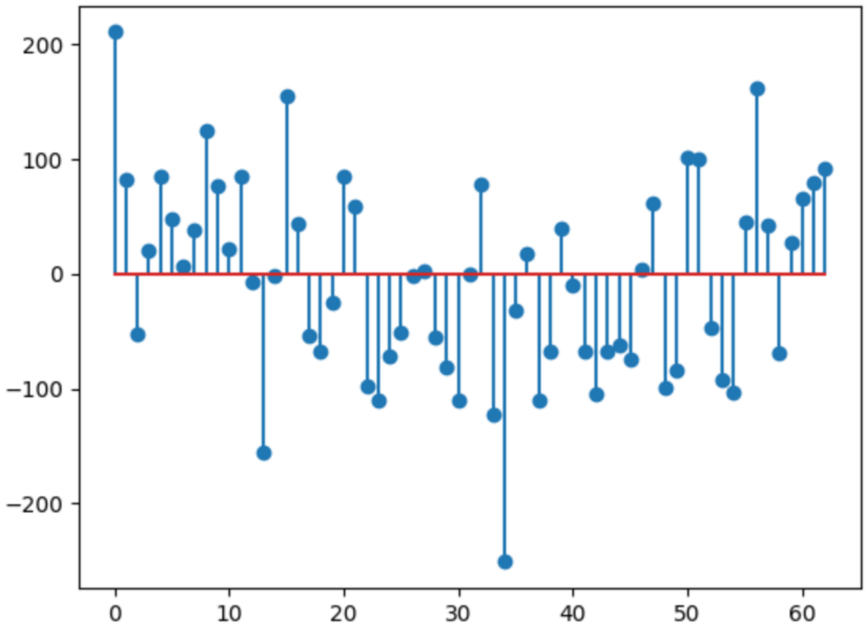
22 – число гармонік в сигналі = 10; гранична частота = 1200; кількість дискретних відліків = 64.

Лістинг програми:

```
import matplotlib.pyplot as plt
import random
```

```
import math

n = 10
omegaMax = 1200
N = 64

k = 128
tau = 64

def Plot(g):
    A = []
    fi = []
    for i in range(n):
        A.append(random.random())
        fii = random.random() * omegaMax
        fi.append(fii)
    for i in range(k):
        res = 0
        for j in range(n):
            res += A[j] * math.sin((omegaMax / (j + 1)) * i + fi[j])
        g.append(res)
        yy = i

def Fourier(g):
    Fp = []
    W = []
    Re = []
    Im = []
    for i in range(N):
        Re.append(math.sin(i * 2 * math.pi / 4))
        Im.append(math.cos(i * 2 * math.pi / 4))
        W.append(math.sqrt((Re[i] * Re[i]) + (Im[i] * Im[i])))
    for k in range(N - 1):
        Wpk = 0
        for p in range(N - 1):
            Wpk = Wpk + W[(p * k) % N]
        Fp.append(g[k] * Wpk)
    return Fp



if __name__ == "__main__":
    x = []
    Plot(x)
    Fp = Fourier(x)
    plt.stem(Fp)
    plt.show()
```

Результат виконання:



Висновки:

На даній лабораторній роботі було здійснено ознайомлення з принципами реалізації спектрального аналізу випадкових сигналів на основі алгоритму перетворення Фур'є, вивчено та досліджено особливості даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.