

**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 №2.2

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«ДОСЛІДЖЕННЯ АЛГОРИТМУ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є З
ПРОРІДЖУВАННЯМ ВІДЛІКІВ СИГНАЛІВ У ЧАСІ»

Виконав:

студент групи ПІ-84
Тимофеєнко Павло Вікторович
номер залікової книжки: 8523

Перевірів:

викладач
Регіда Павло Геннадійович

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.

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

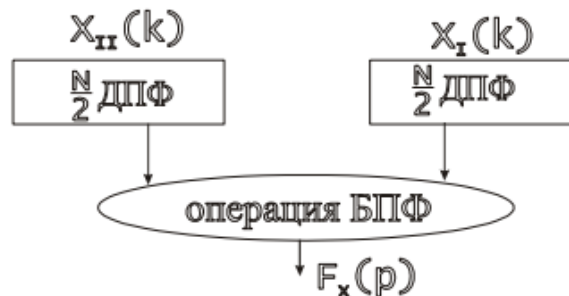
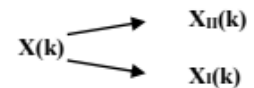
Для стану таких груп коефіцієнтів процедура ДПФ повинна стати багаторівневою, не порушуючи загальних функціональних зв'язків графа процедури ДПФ.

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

- 1) На основі реалізації принципу зрізжених за часом X_k
- 2) на основі реалізації принципу зрізжених відліків шуканого спектру $F(p)$.

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

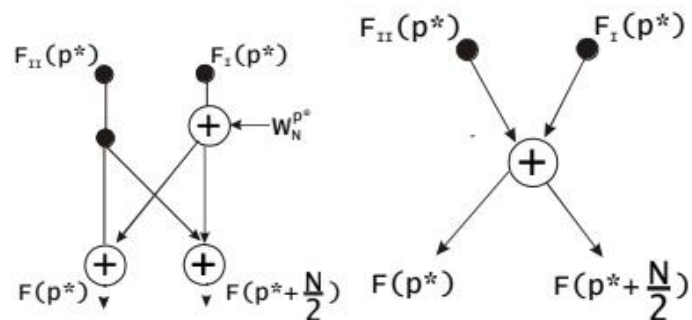
Якщо нам вдасться ефективно розділити, а потім алгоритм отримання спектра, то ми можемо перейти від N ДПФ до $N/2$ ДПФ.



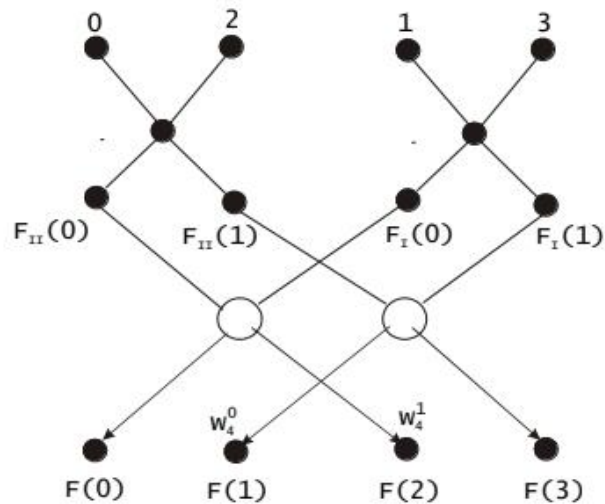
Особливості апаратної реалізації базової операції БПФ з проріджуванням за часом.

В процесорах цифрової обробки сигналів спеціальне АЛУ орієнтоване на виконання реалізації ШПФ або за часом, або по частоті. Функції, реалізовані базовими операціями БПФ, визначаються базовими алгоритмами. Операнди представлені як комплексні величини. Особливість операції - 2 операнда на вході, на виході так само 2 результату.

Графічно базову операцію БПФ описують так:



Цю базову операцію називають метеликом через форми графа. Для спрощеного аналізу процедури БПФ використовують символічне зображення.



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.

Варіант-23

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

Лістинг програми

Index_fft.py

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

plt.style.use('bmh')

harmonics_count = 8
frequency = 1500
N = 1024

signals = rsg.generateSignal(harmonics_count,frequency,N)
AFFT = np.abs(fft.FFT(signals))

fig, (ax1, ax2) = plt.subplots(nrows=2, ncols=1)
fig.subplots_adjust(hspace=0.5)
fig.set_size_inches(12,6)

ax1.plot(signals)
ax1.set_xlim(0, int(N/4))
ax1.set(xlabel='Time', ylabel='Signal(t)',
        title='Random generated signals')

ax2.plot(AFFT)
ax2.set_xlim(0, int(N/4))
ax2.set(xlabel='Time', ylabel='Freq (Hz)',
```

```

        title='Frequency spectrum(FFT)')

fig.savefig("plot(fft).png")
plt.show()

```

rsg.py

```

import random
import math

def generateSignal(harmonics_count,frequency,N):
    signal = [0] * N

    for i in range (harmonics_count):

        W = (frequency / harmonics_count) * (i+1)
        Amplitude = random.random()
        Phase = random.random()

        for t in range(N):
            signal[t] += (Amplitude * math.sin(W * t + Phase))

    return signal

```

fft.py

```

import math

def calcCoef(p, N):
    return (math.cos(2.0 * math.pi * p / N) + math.sin(2.0 * math.pi * p / N) * 1j)

def FFT(signals):

    N = len(signals)
    if N == 1:
        return signals

    FFT_spec = [0] * N

    evens = FFT(signals[::2])
    odds = FFT(signals[1::2])

    length = N/2

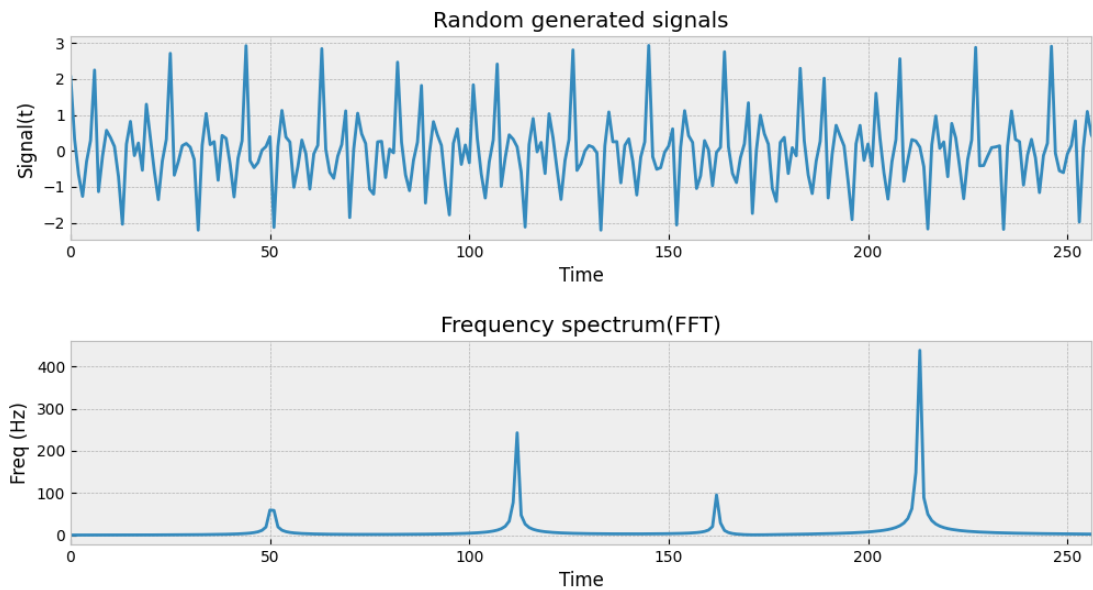
    for p in range(int(length)):
        w = calcCoef(p, N)
        incalc = odds[p] * w

```

```
FFT_spec[p] = evens[p] + incalc
FFT_spec[p + int(length)] = evens[p] - incalc

return FFT_spec
```

Результати роботи програми



Висновок

Був ознайомлений з принципами реалізації прискореного спектрального аналізу випадкових сигналів на основі алгоритму швидкого перетворення Фур'є. Вивчив та дослідив особливості даного алгоритму з використанням засобів моделювання і сучасної програмної оболонки Python.