

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 2.1
з дисципліни «Інтелектуальні вбудовані системи»
на тему «Дослідження параметрів алгоритму дискретного
перетворення Фур'є»

Виконала:
студентка групи ІП-83
Мазур С. В.

Перевірив:
асистент 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 $x(k)$

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

$$\omega \rightarrow \omega_p \rightarrow p\Delta\omega \rightarrow p \quad \Delta\omega = \frac{2\pi}{T}$$

На всьому інтервалі подання сигналів T , 2π - один період низьких частот. Щоб підвищити точність треба збільшити інтервал T .

$$t \rightarrow t_k \rightarrow k\Delta t \rightarrow k; \quad \Delta t = \frac{T}{N} = \frac{1}{k_{\text{зм}}} \cdot f'_{\text{zp}}.$$

ДПФ - проста обчислювальна процедура типу звірки (тобто Σ -є парних множень), яка за складністю також має оцінку $N^2 + N$. 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.

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

Завдання на лабораторну роботу

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 №14

Число гармонік в сигналі $n - 6$

Гранична частота, $\omega_{\text{гр}}$ - 2100
Кількість дискретних відліків, N - 1024

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 = 6 # number of harmonics
w = 2100 # max frequency
N = 1024 # number of discrete 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

# function for calculating Discrete Fourier Transform coefficient
def dftCoeff ( pk , N ):
    exp = 2 *math.pi*pk/N
    return complex (math.cos(exp), -math.sin(exp))

# function for calculating Discrete Fourier Transform
def dft ( signals ):
    N = len (signals)
    spectrum = []
```

```

    for p in range (N):
        sum = 0
    for k in range (N):
        sum += signals[k] * dftCoeff(p*k, N)
    spectrum.append( abs ( sum ))
return spectrum

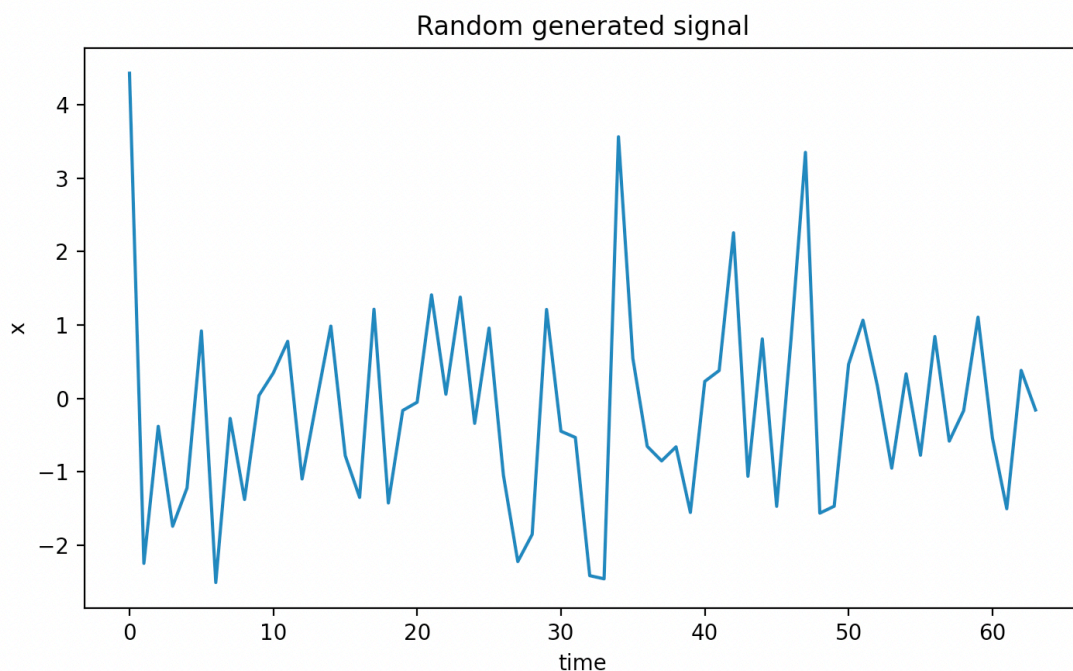
signals = generateSignals(n, w, N)

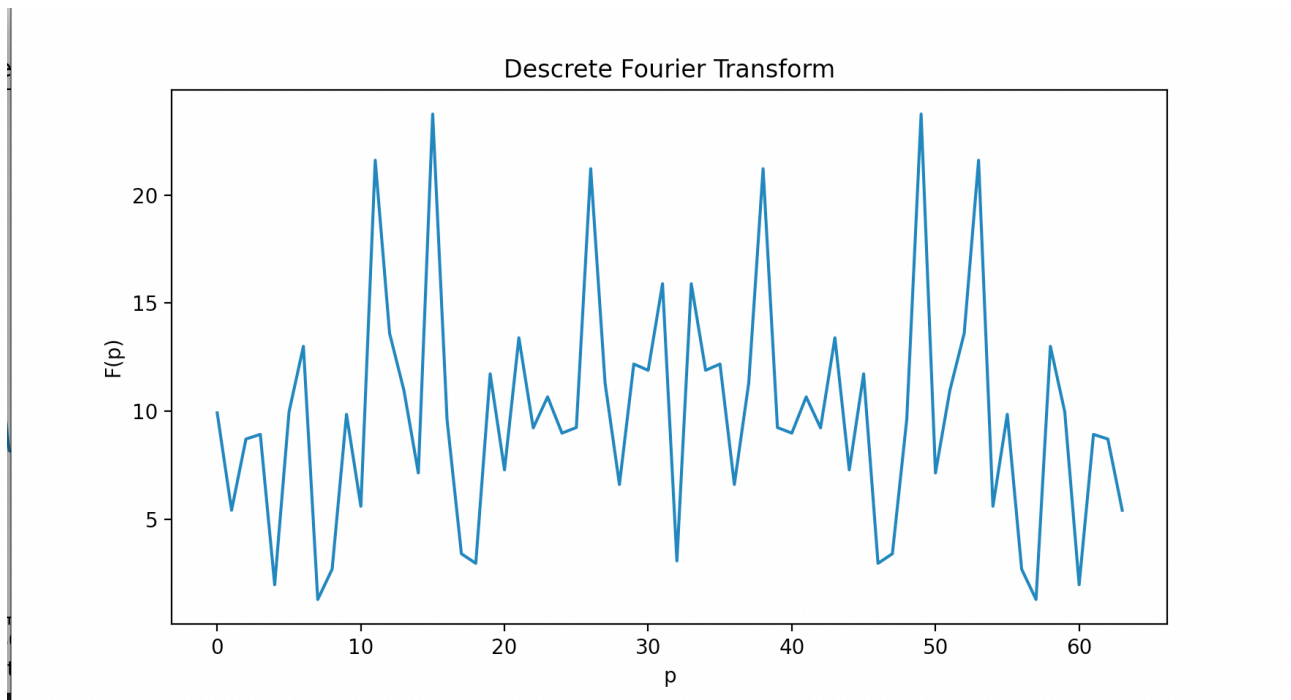
# plotting
# signals
plt.plot(signals)
plt.xlabel( 'time' )
plt.ylabel( 'x' )
plt.title( 'Random generated signal' ) plt.figure()

# dft
plt.plot(dft(signals))
plt.xlabel( 'p' )
plt.ylabel( 'F(p)' )
plt.title( 'Descrete Fourier Transform' )
plt.show()

```

Результат роботи програми





Висновки

Під час виконання лабораторної роботи ознайомилися з принципами реалізації спектрального аналізу випадкових сигналів на основі алгоритму перетворення Фур'є, вивчили та дослідили особливостей даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.