**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

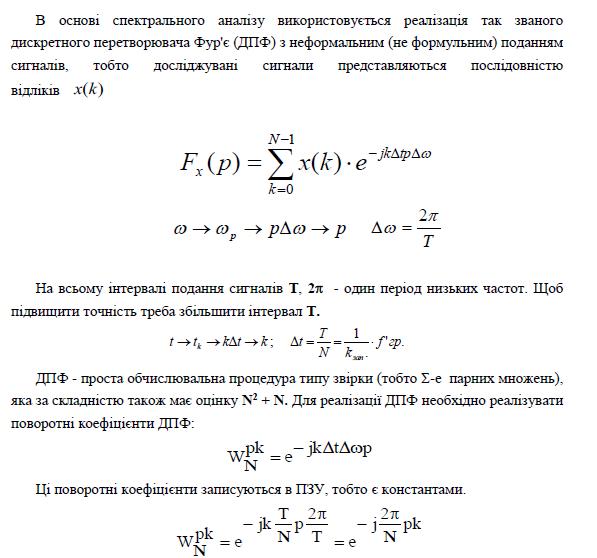
перетворення Фур’є»

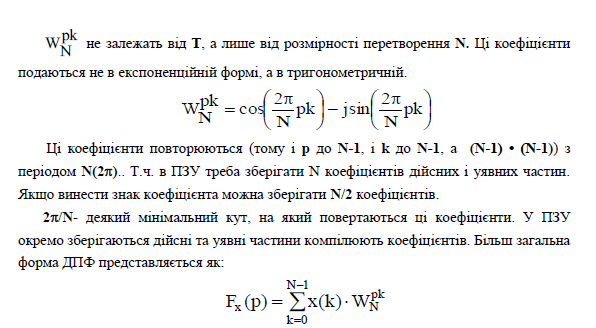
Виконав: Перевірив:

студент групи ІП-84 викладач  
Кучін Владислав Дмитрович Регіда Павло Геннадійович  
номер залікової книжки: 8415

Київ 2021

**Основні теоретичні відомості**



****

**Завдання на лабораторну роботу**

Для згенерованого випадкового сигналу з лабораторної роботи №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-15**

Число гармонік в сигналі: 14.

Гранична частота: 2500.

Кількість дискретних відліків: 64

**Лістинг програми**

import numpy as np

import matplotlib.pyplot as plp

W\_MAX = 2500

HARMONICS\_QUANTITY = 14

DISCRETE\_SAMPLES\_NUMBER = 64

def calculate\_value(amplitude, phase, frequency, time):

return amplitude \* np.sin(frequency \* time + phase)

def create\_signal(w\_max, harmonics\_quantity, discrete\_samples\_number):

sum\_values = np.zeros(discrete\_samples\_number)

sampling\_frequency = 2 \* w\_max

for w in range(1, harmonics\_quantity + 1):

phase = np.random.uniform(-np.pi / 2, np.pi / 2)

amplitude = np.random.uniform(0.0, 1000.0)

frequency = 2 \* np.pi \* w\_max / harmonics\_quantity \* w

for t in range(discrete\_samples\_number):

sum\_values[t] += calculate\_value(amplitude, phase, frequency, t / sampling\_frequency)

return sum\_values

def calc\_w\_pkn(p, k, n):

return complex(np.cos(2 \* np.pi / n \* p \* k), np.sin(2 \* np.pi / n \* p \* k))

def discrete\_fourier\_transform(signal):

result = np.zeros(len(signal), dtype=np.complex)

for p in range(len(signal)):

for k in range(len(signal)):

w = calc\_w\_pkn(p, k, len(signal))

result[p] += w \* signal[k]

return abs(result)

SIGNAL = create\_signal(W\_MAX, HARMONICS\_QUANTITY, DISCRETE\_SAMPLES\_NUMBER)

dft = discrete\_fourier\_transform(SIGNAL)

fft = abs(np.fft.fft(SIGNAL))

time\_period = DISCRETE\_SAMPLES\_NUMBER / (2 \* W\_MAX)

plp.subplot(221)

plp.title('Вихідний сигнал')

plp.plot(np.linspace(0, time\_period, num=DISCRETE\_SAMPLES\_NUMBER), SIGNAL)

plp.subplot(223)

plp.title('Дискретне перетворення Фур\'є (частотний спектр)')

plp.plot(np.linspace(0, 2 \* W\_MAX, num=DISCRETE\_SAMPLES\_NUMBER), dft)

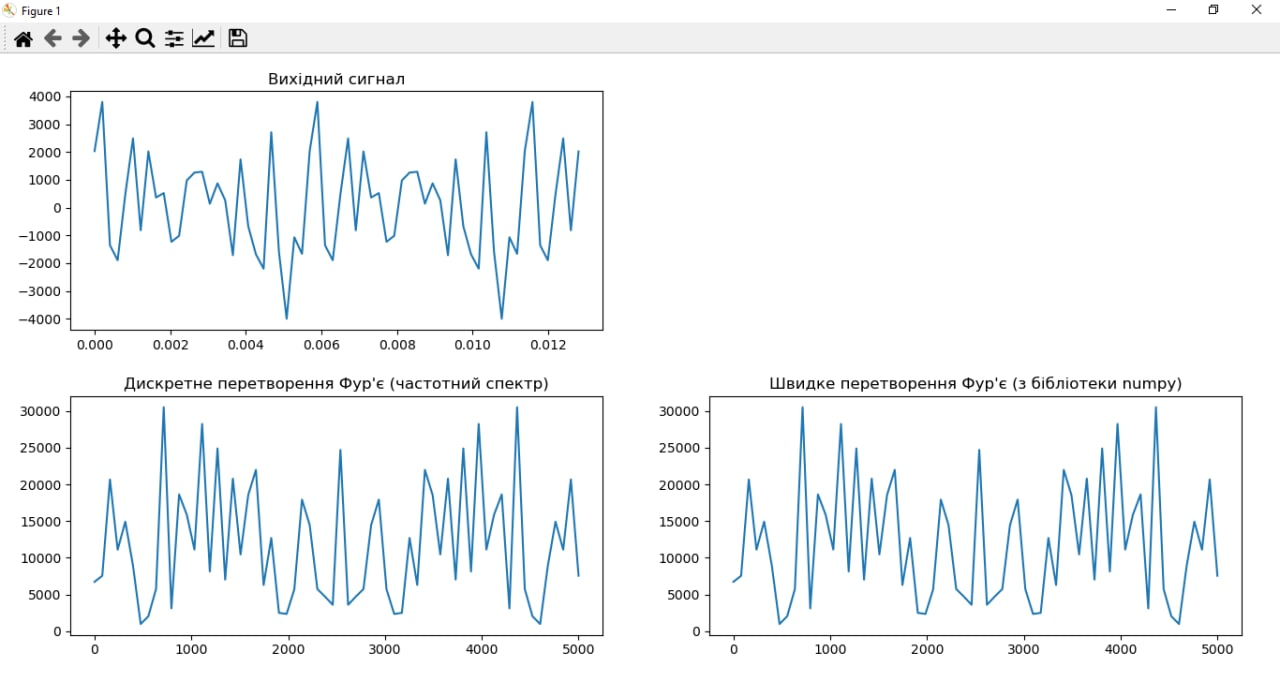
plp.subplot(224)

plp.title('Швидке перетворення Фур\'є (з бібліотеки numpy)')

plp.plot(np.linspace(0, 2 \* W\_MAX, num=DISCRETE\_SAMPLES\_NUMBER), fft)

plp.show()

**Результати роботи програми**



**Висновки**

Під час даної лабораторної роботи ми вивчили, як виділяти частотний спектр з сигналу за допомогою дискретного перетворення Фур’є.