



**Міністерство освіти і науки України  
Національний технічний університет України "Київський політехнічний  
інститут імені Ігоря Сікорського"  
Факультет інформатики та обчислювальної техніки  
Кафедра обчислювальної техніки**

## **ЗВІТ**

про виконання лабораторної роботи №2.2

з дисципліни

**«Інтелектуальні вбудовані системи»**

на тему:

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

Перевірив:  
асистент кафедри ОТ  
Регіда П. Г

ВИКОНАВ:  
студент 3 курсу  
групи ІП-83, ФІОТ  
Мінченко В.Ю.  
Залікова книжка №8315  
Варіант – 18

Київ 2021

### 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 18:

Варіант	Число гармонік в сигналі n	Гранична частота, $\omega_{\text{гp}}$	Кількість дискретних відліків, N
18	10	1500	256

Програмний код:

fft.py

```
import math

def calc_w(p, k, N):
    argument = 2.0 * math.pi * p * k / N
    return complex(math.cos(argument), -math.sin(argument))

def fast_fourier_transform(signal):
    N = len(signal)
    spectre = [0] * N
    if N == 1: return signal
    even_signal, odd_signal = signal[::2], signal[1::2]
    even_transformed = fast_fourier_transform(even_signal)
    odd_transformed = fast_fourier_transform(odd_signal)
    for k in range(0, int(N / 2)):
        w = calc_w(1, k, N)
        spectre[k] = even_transformed[k] + w * odd_transformed[k]
        spectre[k + int(N / 2)] = even_transformed[k] - w * odd_transformed[k]
    return [*map(lambda el: abs(el), spectre)]
```

signal\_generator.py

```
import random
import numpy as np

def generate_signal(signal_harmonics, frequency, discrete_calls):
    signals = np.zeros(discrete_calls)
    for i in range(signal_harmonics):
        frequency_step = frequency / signal_harmonics * (i + 1)
        amplitude = random.random()
        phase = random.random()
        for t in range(discrete_calls):
            signals[t] += amplitude * np.sin(frequency_step * t + phase)
    return signals
```

## Lab2\_2.py

```
import matplotlib.pyplot as plt
from lab2 import signal_generator
import fft

HARMONICS = 10
FREQUENCY = 1500
DISCRETE_CALLS = 256

signal = signal_generator.generate_signal(
    HARMONICS,
    FREQUENCY,
    DISCRETE_CALLS
)
spectrum_fft = fft.fast_fourier_transform(signal)
print(spectrum_fft)
fig, (ax1, ax2) = plt.subplots(2)
fig.suptitle('Laboratorka 2.2')
fig.set_size_inches(18.5, 10.5)

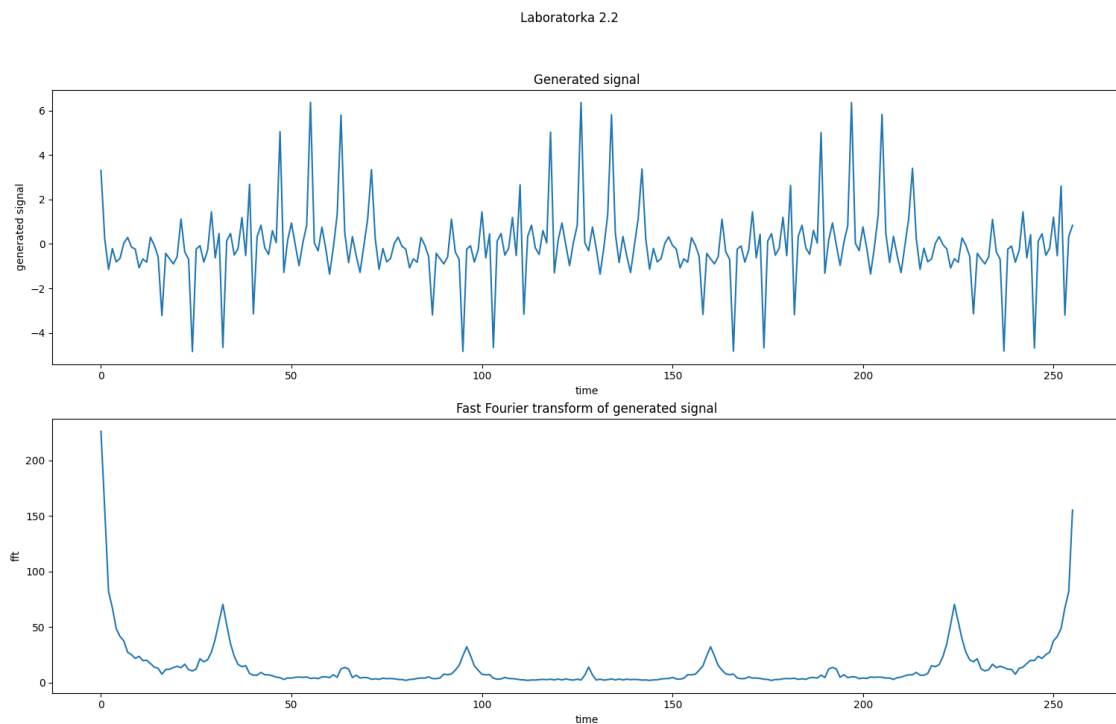
ax1.plot(signal)
ax1.set_title('Generated signal')
ax1.set_xlabel='time', ylabel='generated signal'

ax2.plot(spectrum_fft)
ax2.set_title('Fast Fourier transform of generated signal')
ax2.set_xlabel='time', ylabel='fft'
fig.savefig('lab2_2.png')

plt.show()
```

Результати роботи програми:

Графіки згенерованого сигналу та реалізація швидкого перетворення Фур'є з проріджуванням відліків сигналу за часом:



## Висновки:

Отже, в ході лабораторної роботи, ми вдосконалили практичні вміння генерувати випадковий сигнал згідно формули та вхідних значень за допомогою власноруч написаної програми, а також обраховували спектр отриманого випадкового сигналу за допомогою алгоритму швидкого перетворення Фур'є. Результати наведено в звіті та в репозиторії. Кінцеву мету було досягнуто.