Національний Технічний Університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра обчислювальної техніки

**Лабораторна робота №4**

з курсу «Системи реального часу»

Варіант 13

Виконав:

студент 3 курсу ФІОТ

групи ІО-72

Клецун М.С.

Зал. Книжка: №7213

Перевірив:

Регіда П.Г.

Київ 2020

## ***Тема роботи:*** *ДОСЛІДЖЕННЯ АЛГОРИТМУ ШВИДКОГО ПЕРЕТВОРЕННЯ ФУР'Є З*

## *ПРОРІДЖУВАННЯМ ВІДЛІКІВ СИГНАЛІВ У ЧАСІ*

## 

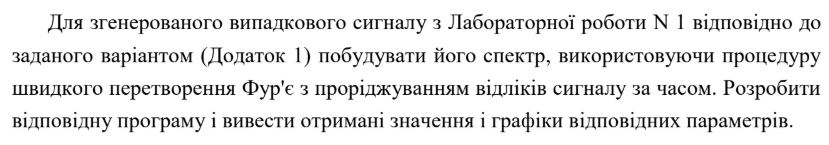
**Варіант №13:**

*n = 10*

*Wgr = 1500*

*N = 256*

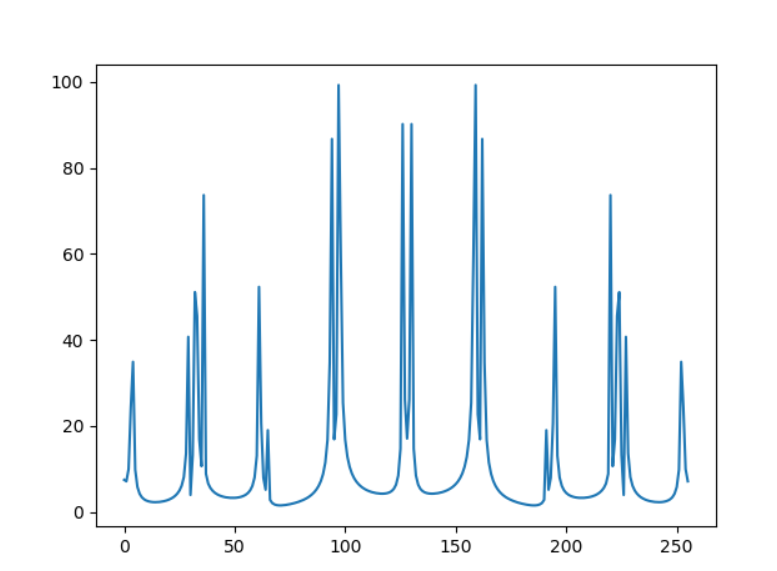
**Завдання**



**Лістинг програми**

import numpy as np  
import random as rnd  
import matplotlib.pyplot as plt  
  
  
def create\_harmon():  
 harmon = [0 for \_ in range(vidlik\_number)]  
 for i in range(number\_of\_harmonics):  
 A = rnd.uniform(min\_number, max\_number)  
 Fi = rnd.uniform(min\_number, max\_number)  
 for t in range(vidlik\_number):  
 harmon[t] += A\*np.sin(frequency/number\_of\_harmonics\*t\*i + Fi)  
 return harmon  
  
  
def fft(x: list):  
 N = len(x)  
 fftt = [[0] \* 2 for i in range(N)]  
 for i in range(N // 2):  
 array1 = [0] \* 2  
 array2 = [0] \* 2  
 for j in range(N // 2):  
 cos = np.cos(4 \* np.pi \* i \* j / N)  
 sin = np.sin(4 \* np.pi \* i \* j / N)  
 array1[0] += x[2 \* j + 1] \* cos # real  
 array1[1] += x[2 \* j + 1] \* sin # imagine  
 array2[0] += x[2 \* j] \* cos # real  
 array2[1] += x[2 \* j] \* sin # imagine  
 cos = np.cos(2 \* np.pi \* i / N)  
 sin = np.sin(2 \* np.pi \* i / N)  
 fftt[i][0] = array2[0] + array1[0] \* cos - array1[1] \* sin # real  
 fftt[i][1] = array2[1] + array1[0] \* sin + array1[1] \* cos # imagine  
 fftt[i + N // 2][0] = array2[0] - (array1[0] \* cos - array1[1] \* sin) # real  
 fftt[i + N // 2][1] = array2[1] - (array1[0] \* sin + array1[1] \* cos) # imagine  
 return fftt  
  
  
  
number\_of\_harmonics = 10  
vidlik\_number = 256  
frequency = 1500  
min\_number = 0  
max\_number = 1  
  
x = create\_harmon()  
fft = fft(x)  
data\_fft = [np.sqrt(fft[i][0] \*\* 2 + fft[i][1] \*\* 2) for i in range(vidlik\_number)]  
  
plt.plot([i for i in range(vidlik\_number)], x)  
plt.show()  
plt.plot([i for i in range(vidlik\_number)], data\_fft)  
plt.show()

**Приклад роботи**



**Висновок**

Під час виконання лабораторної роботи проблем не виникло. Завдання виконане, програма працює та виводить правильний результат.