Національний Технічний Університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та Обчислювальної Техніки

Кафедра обчислювальної техніки

**Лабораторна робота №2.1**

з курсу «Системи реального часу»

Варіант 1

Виконав:

студент 3 курсу ФІОТ

групи ІО-72

Бутенко В.О.

Зал. Книжка: №7201

Перевірив:

Регіда П.Г.

Київ 2020

## ***Тема роботи:*** *ДОСЛІДЖЕННЯ ПАРАМЕТРІВ АЛГОРИТМУ ДИСКРЕТНОГО ПЕРЕТВОРЕННЯ ФУР'Є*

## ***Мета роботи:*** *ознайомлення з принципами реалізації спектральногот аналізу випадкових сигналів на основі алгоритму перетворення Фур'є, вивчення та дослідження особливостей даного алгоритму з використанням засобів моделювання і сучасних програмних оболонок.*

**Варіант №1:**

*n = 6*

*Wgr = 1200*

*N = 64*

**Завдання**

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.

**ДОП ЗАВДАННЯ:**

Використовуючи бібліотеку Numpy, порахувати значення, побудувати графіки та порівняти їх з графіками, які були створені самостійно в Лаб 2.1.

**Лістинг програми**

**import** math  
**import** matplotlib.pyplot **as** plt  
**from** random **import** \*  
**import** time  
**from** numpy **import** fft  
**import** numpy **as** np  
  
n = 6 *# число гармонік в сигналі*Wmax = 1200 *# гранична частота*N = 64 *# число дискретних відліків*x = [0 **for** i **in** range(N)] *# масив сигналів  
  
# Генерація сигналів*start = time.time()  
**for** i **in** range(n):  
 A = uniform(0, 1)  
 F = uniform(0, 1)  
 **for** j **in** range(0, N):  
 x[j] += A \* math.sin(Wmax / n \* j \* i + F)  
  
end = time.time()  
print(**"Час генерації сигналів"**, end - start)  
Fr = []  
Fi = []  
  
  
**def** dpf(signals, N):  
 **for** i **in** range(N):  
 Fr.append(0)  
 Fi.append(0)  
 **for** j **in** range(N):  
 Fr[i] += x[j] \* math.cos(-2 \* math.pi \* i \* j / N)  
 Fi[i] += x[j] \* math.sin(-2 \* math.pi \* i \* j / N)  
  
  
start = time.time()  
dpf(x, N)  
end = time.time()  
print(**"Час ДПФ"**, end - start)  
F = np.array([Fr[i] + 1j \* Fi[i] **for** i **in** range(N)])  
Fl = fft.fft(x)  
  
fig = plt.figure()  
  
g1 = fig.add\_subplot(2, 2, 1)  
g2 = fig.add\_subplot(2, 2, 2)  
g3 = fig.add\_subplot(2, 2, 3)  
g4 = fig.add\_subplot(2, 2, 4)  
  
g1.plot(range(N), Fr)  
g2.plot(range(N), Fi)  
g3.plot(range(N), Fl.real)  
g4.plot(range(N), Fl.imag)  
  
g1.set(title=**"Fr"**)  
g2.set(title=**"Fi"**)  
g3.set(title=**"Fr generated with Numpy"**)  
g4.set(title=**"Fi generated with Numpy"**)  
  
plt.show()

## Результат роботи



