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

ЗВІТ

Лабораторна робота №2.1

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«Дослідження параметрів алгоритму дискретного перетворення Фур'є»

Виконав:
Василиненко Д.Д.

Студент групи ІП-84

Перевірив:

Регіда Павло Геннадійович

Київ 2021

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.

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 $x(k)$

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

$$\omega \to \omega_p \to p\Delta\omega \to p \qquad \Delta\omega = \frac{2\pi}{T}$$

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

$$t \to t_k \to k\Delta t \to k; \qquad \Delta t = \frac{T}{N} = \frac{1}{k_{зап.}} \cdot f'гр.$$

ДПФ - проста обчислювальна процедура типу звірки (тобто Σ-е 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 по варіанту (4):

- Число гармонік в сигналі - 12
- Гранична частота - 2400

- Кількість дискретних відліків - 1024

Вихідний код:

f.py

```
import math

def fCoef(pk, N):
  arg = 2 * math.pi * pk / N
  return complex(math.cos(arg), -math.sin(arg))

def f(signal):
  N = len(signal)
  spectre = list()
  for p in range(N):
    sum = 0
    for k in range(N):
      x = signal[k]
      w = fCoef(p*k, N)
      sum += w * x
    res = abs(sum)
    spectre.append(res)
  return spectre
```

rsg.py

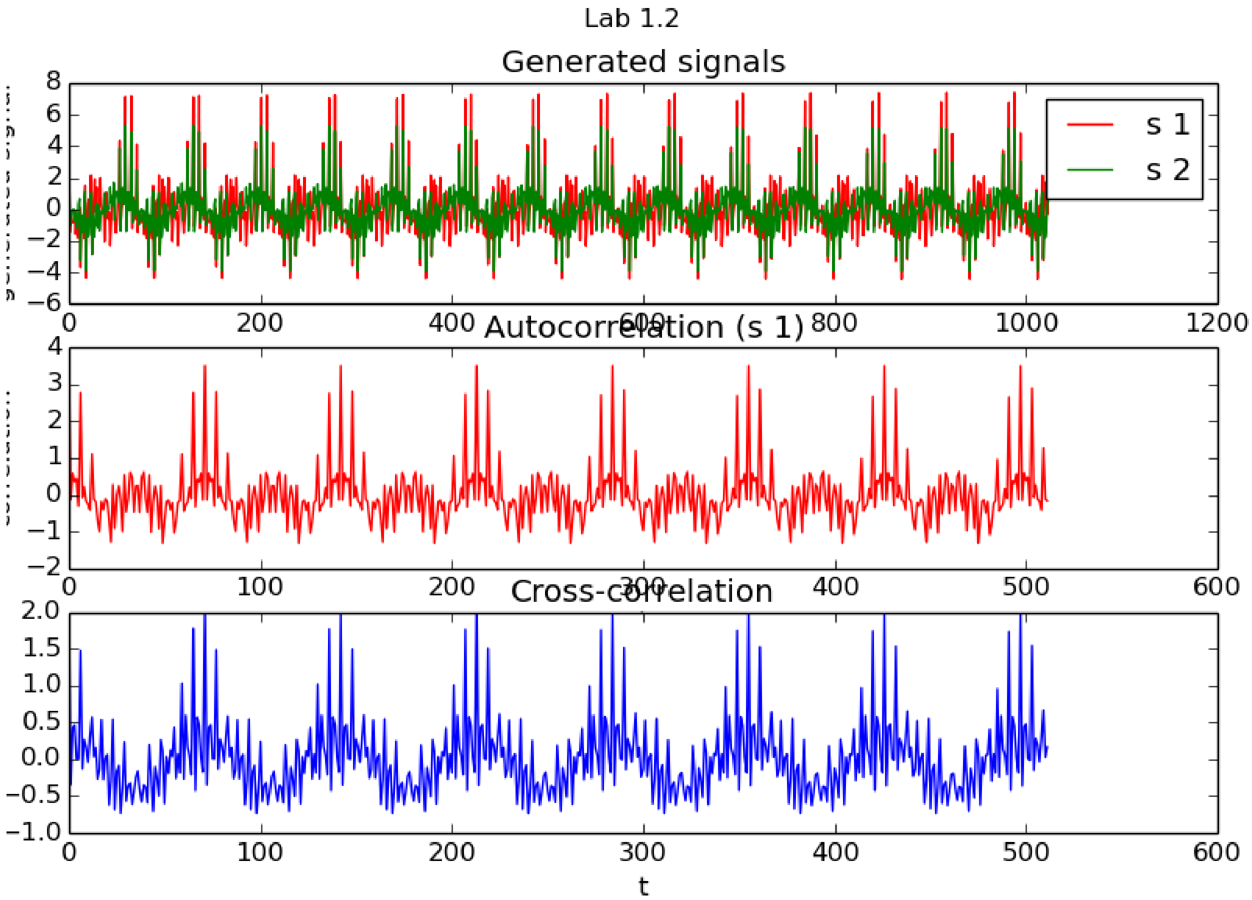
```
import random
import math
import time

def getRandomSignal(harmonicsAmount, limitFrequency, N):
    signal = [0] * N
    for i in range (harmonicsAmount):
        w = (limitFrequency / harmonicsAmount) * (i+1)
        A = random.random()
        Fi = random.random()
        for t in range(N):
            signal[t] += (A * math.sin(w * t + Fi))
```

```
    return signal
```

index.py

```
import matplotlib
import matplotlib.pyplot as plt
import numpy as np
import rsg
import f

n = 12
W = 2400
N = 1024
time = range(N)
signal = rsg.getRandomSignal(n, W, N)

s = f.f(signal)

fig, (ax1, ax2) = plt.subplots(2, 1)
fig.suptitle('Lab 2.1')

ax1.plot(signal, linewidth=0.8)
ax1.set_title('Generated signal')
ax1.set(xlabel='time', ylabel='generated signal')

ax2.plot(s, color='r', linewidth=0.8)
ax2.set_title('Discrete Fourier transform')
ax2.set(xlabel='p', ylabel='F(p)')
plt.show()
```

Результати роботи програми



Висновки

У ході роботи було згенеровано два сигнали. Під час виконання лабораторної роботи на прикладі яких обраховувалися автокореляція та взаємна кореляція. Було отримано 3 графіки: згенеровані сигнали, накладені одне на одного, автокореляції для сигналу X, взаємної кореляції двох сигналів.