Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук і кібернетики

Звіт

з лабораторної роботи №1

з моделювання систем

Виконав:

Студент групи ІПС-32

Гончаренко Ілля Сергійович

Київ

2023

Варіант 15

1. Суть задачи:

   Визначити модель в класі функцій

$$y(t) = a_1t^3 + a_2t^2 + a_3t + \sum_{i=4}^{k} a_i sin(2\pi f_{i-3}t) + a_{k+1}$$

Для спостережуваної дискретної функції $y(t_i), i = 1,2,...,N$, (відповідний файл f15.txt), $t_{i+1} - t_i = \Delta t = 0.01$, інтервал спостереження $[0,T], T = 5$

Для цього нам потрібно виконати дискретне перетворення Фур'є для дискретної послідовності $y(t_m), m = 0,1,2,...,N-1$

$$c_y(k) = \frac{1}{N}\sum_{m=0}^{N-1} y(t_m)e^{-i2\pi km/N}$$

Тепер нам потрібно визначити частоти з найбільшим вкладом в дискретному перетворенні Фур'є. Для цього візьмемо момент де модуль набуває найбільшого значення(тобто локальний екстремум). І тоді виконаємо множення $k * \Delta f = k * /T$ (де k* - локальні екстремуми, а f* - частоти з найбільшим вкладом).

Знайшовши все необхідне можна перейти до визначення невідомих параметрів $a_i, i = k+1$, де будемо застосовувати метод найменших квадратів. Для цього записуємо функціонал похибки

$$F(a_1, a_2, \ldots, a_{k+1}) = \frac{1}{2}\sum_{j=0}^{N-1}(a_1t_j^3 + a_2t_j^2 + a_3t_j + \sum_{i=4}^{k} a_i sin(2\pi f_{i-3}t_j) + a_{k+1} - y(t_j))^2$$

$a_i, i = 1,2,...,k+1$ - беремо з умови

$$F(a_1, a_2, \ldots, a_{k+1}) -> \min_{a_1,a_2,\ldots,a_{k+1}}$$

Записуємо систему рівнянь:

$$\frac{\partial F(a_1, a_2, \ldots, a_{k+1})}{\partial a_j} = 0$$

Ця система є системою лінійних алгебраїчних рівнянь. Розв'язавши цю систему одним із відомих методів, знаходимо $a_i, i = 1,2,...,k+1$

2. Хід роботи:

Реалізовано мовою Python

Почнемо з ініціалізації всіх необхідних параметрів

```
# Завантаження данних
observations = np.loadtxt('f15.txt')

T = 5
N = len(observations)
dt = 0.01
t = np.arange(0, N*dt, dt)
```

Виконуємо дискретне перетворення Фур'є та визначаємо його за модулем:

```
# 1. Дискретне перетворення Фур'є
f_transform = np.fft.fft(observations) / N
frequencies = np.fft.fftfreq(N, dt)

# Модуль перетворення Фур'є
f_magnitude = np.abs(f_transform)
```

Знаходимо і виводимо локальні максимуми(суттєві частоти):
Хочу зазначити що :N //2 - береться для того щоб визначити локальні максимуми лише для лівої частини тому що у дискретному

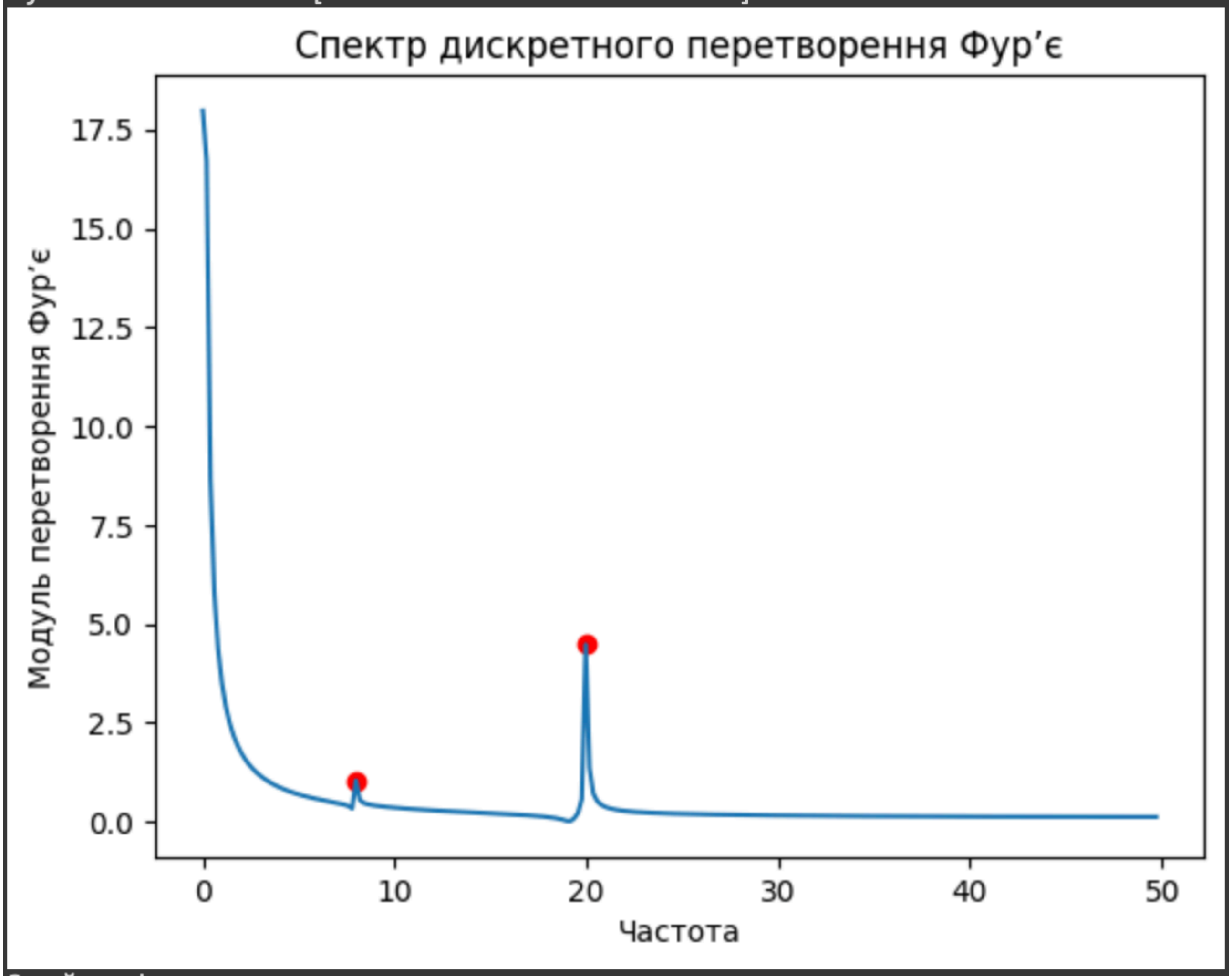
перетворенні Фур’є спектр є симетричним відносно нуля(дзеркальним) тому беремо лише одну частину

```
# Локальні максимуми
peaks, _ = find_peaks(f_magnitude[:N // 2])
peak_frequencies = frequencies[peaks]

print("Суттєві частоти:", peak_frequencies)
```

Виконуємо побудову графіку:
(Не забуваємо про :N // 2)

```
# 2. Побудова графіку модуля перетворення Фур’є
plt.figure()
plt.plot(frequencies[:N // 2], f_magnitude[:N // 2])
plt.scatter(peak_frequencies, f_magnitude[peaks], color='red')
plt.xlabel('Частота')
plt.ylabel('Модуль перетворення Фур’є')
plt.title('Спектр дискретного перетворення Фур’є')
plt.show()
```

```
Суттєві частоти: [ 7.98403194 19.96007984]
```



Отримали 2 значущих вклади частоти 7.98 і 19.96 Гц(близьке до 0 ігнорується, це поліноміальний вклад)

Тепер приступаємо до визначення параметрів найменших квадратів
I

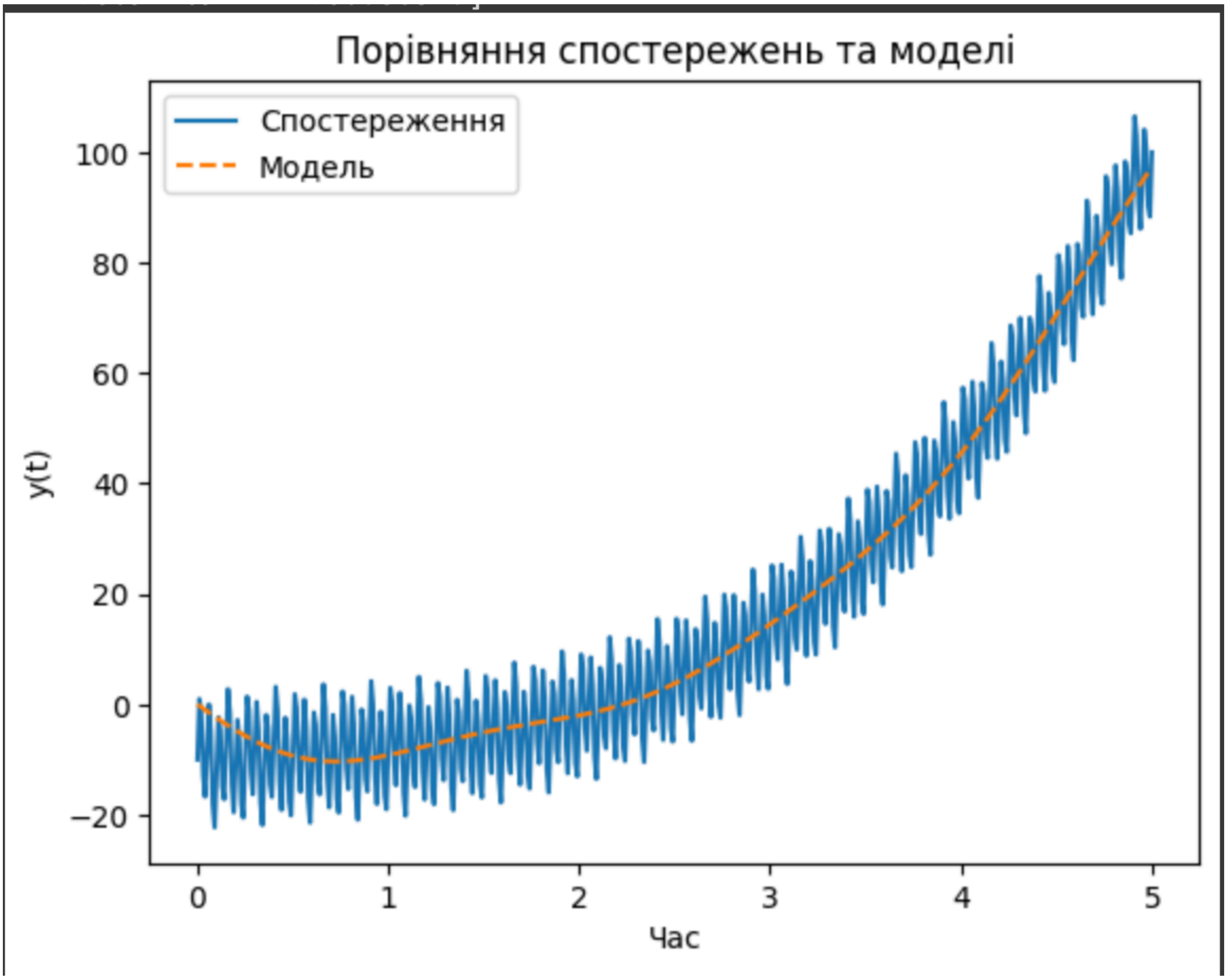
```
# 3. Метод найменших квадратів
def model(t, a1, a2, a3, *params):
    k = len(params) // 2
    y = a1 * t**3 + a2 * t**2 + a3 * t
    for i in range(k):
        fi = params[i]
        ai = params[k + i]
        y += ai * np.sin(2 * np.pi * fi * t - 3 * t)
    return y

# Ініціалізація параметрів
initial_guess = [1, 1, 1] + [1] * (len(peak_frequencies) * 2)

# 4. Підбір параметрів за методом найменших квадратів
params, covariance = curve_fit(model, t, observations, p0=initial_guess)

print("Знайдені параметри:", params)
```

```
Знайдені параметри: [  0.11100943   6.05587157 -14.2325163    0.85191096   1.02857289
  -2.30921694  -1.60780517]
```

Будуємо графік апроксимованої функції, і порівняємо його з моделлю

```
# 5. Обчислення значень за моделлю
fitted_values = model(t, *params)

plt.figure()
plt.plot(t, observations, label='Спостереження')
plt.plot(t, fitted_values, label='Модель', linestyle='--')
plt.xlabel('Час')
plt.ylabel('y(t)')
plt.legend()
plt.title('Порівняння спостережень та моделі')
plt.show()
```

Отримали графік:



Маємо апроксимуючу функцію:

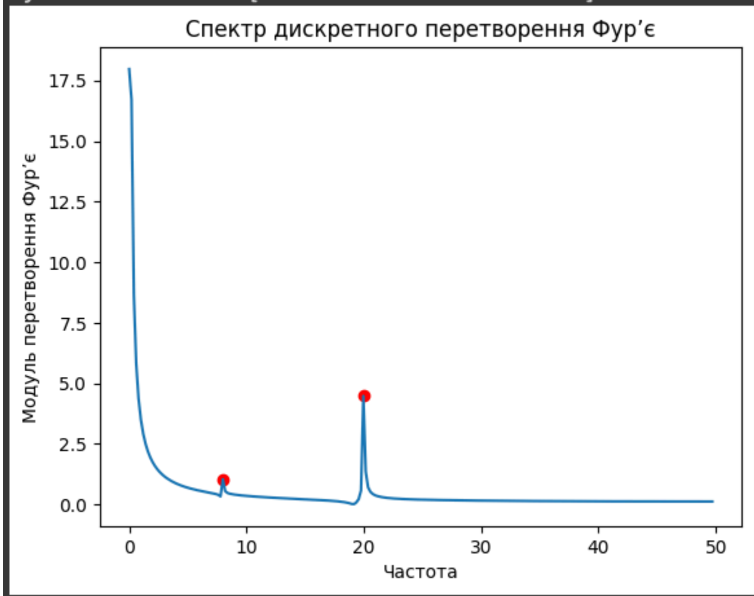
$$y(t) = 0.111t^3 + 6.056t^2 - 14.233t + 0.852sin(2\pi \cdot 1.028t - 3t) + (-2.309)sin(2\pi \cdot (-1.608)t - 3t)$$

Повний код:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks
from scipy.optimize import curve_fit

# Завантаження даних
observations = np.loadtxt('f15.txt')

T = 5
N = len(observations)
dt = 0.01
t = np.arange(0, N*dt, dt)

# 1. Дискретне перетворення Фур'є
f_transform = np.fft.fft(observations) / N
frequencies = np.fft.fftfreq(N, dt)

# Модуль перетворення Фур'є
f_magnitude = np.abs(f_transform)

# Локальні максимуми
peaks, _ = find_peaks(f_magnitude[:N // 2])
peak_frequencies = frequencies[peaks]

print("Суттєві частоти:", peak_frequencies)

# 2. Побудова графіку модуля перетворення Фур'є
plt.figure()
plt.plot(frequencies[:N // 2], f_magnitude[:N // 2])
plt.scatter(peak_frequencies, f_magnitude[peaks], color='red')
plt.xlabel('Частота')
plt.ylabel('Модуль перетворення Фур'є')
plt.title('Спектр дискретного перетворення Фур'є')
plt.show()

# 3. Метод найменших квадратів
def model(t, a1, a2, a3, *params):
    k = len(params) // 2
    y = a1 * t**3 + a2 * t**2 + a3 * t
    for i in range(k):
        fi = params[i]
        ai = params[k + i]
        y += ai * np.sin(2 * np.pi * fi * t - 3 * t)
    return y

# Ініціалізація параметрів
initial_guess = [1, 1, 1] + [1] * (len(peak_frequencies) * 2)

# 4. Підбір параметрів за методом найменших квадратів
params, covariance = curve_fit(model, t, observations, p0=initial_guess)

print("Знайдені параметри:", params)

# 5. Обчислення значень за моделлю
fitted_values = model(t, *params)

plt.figure()
plt.plot(t, observations, label='Спостереження')
plt.plot(t, fitted_values, label='Модель', linestyle='--')
plt.xlabel('Час')
plt.ylabel('y(t)')
plt.legend()
plt.title('Порівняння спостережень та моделі')
plt.show()
```

Повний вивід програми:

```
Суттєві частоти: [ 7.98403194 19.96007984]
```




```
Знайдені параметри: [   0.11100943    6.05587157 -14.2325163     0.85191096    1.02857289
  -2.30921694  -1.60780517]
```