

Київський національний університет імені Тараса Шевченка  
Факультет комп'ютерних наук та кібернетики  
Кафедра інтелектуальних програмних систем  
Моделювання систем

Лабораторна робота №1  
Виконала студентка 3-го курсу  
Групи ІПС-31  
Величко Діана Сергіївна  
Варіант 3

2024

### Завдання :

Визначити модель в класі функції

$$y(t) = a_1 t^3 + a_2 t^2 + a_3 t + \sum_{i=4}^k a_i \sin(2\pi f_{i-3} t) + a_{k+1} \text{ для спостережуваної}$$

дискретної функції  $\hat{y}(t_i)$ ,  $i = 1, 2, \dots, N$ ,  $t_{i+1} - t_i = \Delta t = 0.01$ , інтервал спостереження  $[0, T]$ ,  $T = 5$ .

Використати дискретне перетворення Фур'є для дискретної послідовності  $x(i)$ ,  $i = 0, 1, 2, \dots, N - 1$

$$c_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} x(m) e^{-i2\pi km/N}.$$

### Спостереження :

3 5.9886 7.8541 7.9026 6.1342 3.2426 0.35049 -1.4196 -1.374 0.4875 3.471 6.454 8.3138  
8.3568 6.5829 3.6859 0.78837 -0.98707 -0.94665 0.90963 3.888 6.8659 8.7207 8.7587 6.98  
4.0781 1.1758 -0.6043 -0.56853 1.2832 4.257 7.2304 9.0809 9.1145 7.3314 4.4254 1.5189 -  
0.26533 -0.23361 1.6141 4.584 7.5535 9.4002 9.4301 7.6433 4.7336 1.8236 0.03584 0.064109  
1.9084 4.875 7.8413 9.6847 9.7115 7.9216 5.0089 2.0959 0.30521 0.33063 2.1722 5.136 8.0996  
9.9404 9.9646 8.1723 5.2571 2.3418 0.54878 0.57195 2.4113 5.373 8.3345 10.1733 10.1956  
8.4014 5.4844 2.5672 0.77255 0.79407 2.6318 5.592 8.5521 10.3895 10.4104 8.6148 5.6966  
2.7783 0.98252 1.003 2.8397 5.799 8.7582 10.5948 10.6149 8.8187 5.8999 2.981 1.1847 1.2047  
3.0411 6 8.9589 10.7953 10.8153 9.019 6.1001 3.1813 1.3851 1.4052 3.2418 6.201 9.1603  
10.997 11.0175 9.2217 6.3034 3.3852 1.5896 1.6105 3.4479 6.408 9.3682 11.2059 11.2274  
9.4328 6.5156 3.5986 1.8044 1.8267 3.6655 6.627 9.5887 11.4281 11.4512 9.6582 6.7429  
3.8277 2.0354 2.0596 3.9004 6.864 9.8278 11.6694 11.6948 9.9041 6.9911 4.0784 2.2885  
2.3153 4.1587 7.125 10.0916 11.9359 11.9642 10.1764 7.2664 4.3567 2.5699 2.5998 4.4465  
7.416 10.3859 12.2336 12.2653 10.4811 7.5746 4.6686 2.8855 2.9191 4.7696 7.743 10.7168  
12.5685 12.6043 10.8242 7.9219 5.02 3.2413 3.2793 5.1341 8.112 11.0904 12.9467 12.9871  
11.2116 8.3141 5.4171 3.6432 3.6862 5.546 8.529 11.5125 13.374 13.4196 11.6495 8.7574  
5.8658 4.0974 4.1459 6.0114 9 11.9892 13.8565 13.908 12.1438 9.2576 6.3721 4.6098 4.6644  
6.5361 9.531 12.5266 14.4002 14.4582 12.7005 9.8209 6.942 5.1863 5.2477 7.1262 10.128  
13.1305 15.0111 15.0761 13.3256 10.4531 7.5814 5.8331 5.9019 7.7878 10.797 13.807 15.6953  
15.7679 14.025 11.1604 8.2965 6.5561 6.6328 8.5267 11.544 14.5621 16.4586 16.5395 14.8049  
11.9486 9.0932 7.3612 7.4465 9.349 12.375 15.4019 17.3071 17.3969 15.6712 12.8239 9.9775  
8.2546 8.349 10.2608 13.296 16.3322 18.2468 18.346 16.6299 13.7921 10.9554 9.2422 9.3463  
11.2679 14.313 17.3591 19.2837 19.393 17.687 14.8594 12.0328 10.33 10.4445 12.3764 15.432  
18.4887 20.4239 20.5438 18.8484 16.0316 13.2159 11.5239 11.6494 13.5923 16.659 19.7268  
21.6732 21.8043 20.1203 17.3149 14.5106 12.8301 12.9671 14.9217 18 21.0795 23.0377  
23.1807 21.5086 18.7151 15.9229 14.2545 14.4036 16.3704 19.461 22.5529 24.5234 24.6789  
23.0193 20.2384 17.4588 15.803 15.9649 17.9445 21.048 24.1528 26.1363 26.3048 24.6584

21.8906 19.1242 17.4818 17.6571 19.6501 22.767 25.8853 27.8825 28.0646 26.4318 23.6779  
 20.9253 19.2968 19.486 21.493 24.624 27.7564 29.7678 29.9642 28.3457 25.6061 22.868  
 21.2539 21.4577 23.4793 26.625 29.7722 31.7983 32.0096 30.406 27.6814 24.9583 23.3593  
 23.5782 25.6151 28.776 31.9385 33.98 34.2067 32.6187 29.9096 27.2022 25.6189 25.8535  
 27.9062 31.083 34.2614 36.3189 36.5617 34.9898 32.2969 29.6056 28.0387 28.2897 30.3587  
 33.552 36.747 38.8211 39.0805 37.5252 34.8491 32.1747 30.6246 30.8926 32.9786 36.189  
 39.4011 41.4924 41.769 40.2311 37.5724 34.9154 33.3828 33.6683 35.772 39 42.2298 44.3389  
 44.6334 43.1134 40.4726 37.8337 36.3192 36.6228 38.7447 41.991 45.2392 47.3666 47.6796  
 46.1781 43.5559 40.9356 39.4397 39.7621 41.9028 45.168 48.4351 50.5815 50.9135 49.4312  
 46.8281 44.227 42.7505 43.0923 45.2524 48.537 51.8236 53.9897 54.3413 52.8786 50.2954  
 47.7141 46.2575 46.6192 48.7993 52.104 55.4107 57.597 57.9689 56.5265 53.9636 51.4028  
 49.9666 50.3489 52.5496 55.875 59.2025 61.4095 61.8023 60.3808 57.8389 55.2991 53.884  
 54.2874 56.5094 59.856 63.2048 65.4332 65.8474 64.4475 61.9271 59.409 58.0156 58.4407  
 60.6845 64.053 67.4237 69.6741 70.1104 68.7326 66.2344 63.7384 62.3674 62.8149 65.081  
 68.472 71.8653 74.1383 74.5972 73.242 70.7666 68.2935 66.9453 67.4158 69.7049 73.119  
 76.5354 78.8316 79.3137 77.9819 75.5299 73.0802 71.7555 72.2495 74.5623 78

### **Послідовність виконання :**

1. Знаходимо  $\Delta f = T1$ .
2. Для всіх  $k = 0, 1, \dots, N - 1$  визначаємо модуль перетворення Фур'є  $|\hat{y}(k)|$ , за спостереженнями  $\hat{y}(t_j)$ ,  $j = 0, 1, \dots, N - 1$ .
3. Визначаємо локальні максимуми  $k^*$  модуля перетворення Фур'є  $|\hat{y}(k)|$ ,  $k = 0, 1, \dots, [N/2] - 1$ .
4. Знаходимо частоти  $f^* = k^* \Delta f$ .

Після знаходження частот із найбільшим вкладом, переходимо до визначення невідомих параметрів  $a_j$ ,  $j = 1, 2, \dots, k+1$ . Для їх визначення застосовуємо метод найменших квадратів. Отже, записуємо функціонал похибки

$$F(a_1, a_2, \dots, a_{k+1}) = \sum_{j=0}^{N-1} (a_1 t_j^3 + a_2 t_j^2 + a_3 t_j + \sum_{i=1}^k a_i \sin(2\pi f_i - 3t_i) + a_{k+1} - \hat{y}(t_j))^2$$

Параметри  $a_j$ ,  $j = 1, 2, \dots, k+1$  шукаємо з умови

$$F(a_1, a_2, \dots, a_{k+1}) \rightarrow \min_{a_1, a_2, \dots, a_{k+1}}$$

Для цього записуємо систему рівнянь  $\delta F(a_1, a_2, \dots, a_{k+1}) = 0$ ,

$$\delta a_j$$

$j = 1, 2, \dots, k+1$ . Ця система є системою лінійних алгебраїчних рівнянь. Розв'язавши одним з відомих методів, знаходимо  $a_j$ ,  $j = 1, 2, \dots, k+1$ .

**$[0, T]$ ,  $T = 5$**  - інтервал спостереження

**$\hat{y}(t_i)$**  - спостереження в дискретні моменти часу

**$t_i \in [0, T]$ ,  $i = 0, 1, \dots, N - 1$ ,  $t_{i+1} - t_i = \Delta t = 0.01$ .**

## Реалізація та код розв'язку :

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

# візьмемо із файлу (варіант 3)
data = np.loadtxt('/Users/macbookpro/Desktop/MCC/Lab1/f3.txt')

# ввід параметрів
T = 5
N = len(data)
delta_t = T / N
time_values = np.linspace(0, T, N)

# обчислення перетворення Фур'є
def calculate_dft(signal):
    N = len(signal)
    exp_factor = -2j * np.pi / N
    return np.array([np.sum(signal * np.exp(exp_factor * k * np.arange(N)))
    for k in range(N)]) / N

fourier_transform = calculate_dft(data)

# визначимо частоти
frequencies = np.fft.fftfreq(N, delta_t)
magnitude_fourier = np.abs(fourier_transform)

# визначимо піки
half_range = N // 2
peak_indices, _ = find_peaks(magnitude_fourier[:half_range])

# відбір значущих частот
min_threshold = 1
significant_frequencies = frequencies[peak_indices]
filtered_frequencies =
significant_frequencies[np.abs(significant_frequencies) > min_threshold]

# метод найменших квадратів
def sine_wave(time, frequency):
    return np.sin(2 * np.pi * frequency * time)

def construct_matrix_and_vector(time, signal, freq):
    sine_t = sine_wave(time, freq[0])
    time_powers = [np.sum(time ** p) for p in range(7)]

    # матриця
    matrix = np.array([
        [time_powers[6], time_powers[5], time_powers[4], np.sum(sine_t * time
** 3), time_powers[3]],
        [time_powers[5], time_powers[4], time_powers[3], np.sum(sine_t * time
** 2), time_powers[2]],
        [time_powers[4], time_powers[3], time_powers[2], np.sum(sine_t *
time), time_powers[1]],
        [np.sum(sine_t * time ** 3), np.sum(sine_t * time ** 2),
np.sum(sine_t * time), np.sum(sine_t ** 2), np.sum(sine_t)],
```

```

        [time_powers[3], time_powers[2], time_powers[1], np.sum(sine_t), N]
    ])

    # вектор
    vector = np.array([
        np.sum(signal * time ** 3),
        np.sum(signal * time ** 2),
        np.sum(signal * time),
        np.sum(signal * sine_t),
        np.sum(signal)
    ])

    return matrix, vector

# вирішення системи рівнянь
def solve_least_squares(time, signal, freq):
    matrix, vector = construct_matrix_and_vector(time, signal, freq)
    return np.linalg.solve(matrix, vector)

parameters = solve_least_squares(time_values, data, filtered_frequencies)
rounded_parameters = np.round(parameters).astype(int)

# виведення результатів
print("важливі частоти:", filtered_frequencies)
print("підібрані параметри a:", rounded_parameters)

# рівняння моделі
def display_model_equation(params, freqs):
    equation = f"y(t) = {params[0]} * t^3 + {params[1]} * t^2 + {params[2]} * "
    t + {params[3]} * sin(2π * {freqs[0]} * t) + {params[4]}"
    print("рівняння моделі:", equation)

display_model_equation(rounded_parameters, filtered_frequencies)

# побудова графіків
def plot_observations(time, signal):
    plt.figure(figsize=(10, 5))
    plt.plot(time, signal)
    plt.title('спостереження y(t) в залежності від часу')
    plt.xlabel(' час (в секундах)')
    plt.ylabel('y(t)')
    plt.grid(True)
    plt.show()

def plot_fourier_transform(freqs, magnitudes):
    plt.figure()
    plt.plot(time_values, np.abs(fourier_transform))
    plt.title('модуль перетворення Фур\'є')
    plt.xlabel('частота')
    plt.ylabel('|c_y(k)|')
    plt.grid(True)
    plt.show()

def plot_peaks(freqs, magnitudes, peaks):
    plt.figure(figsize=(10, 5))
    plt.plot(freqs[:half_range], magnitudes[:half_range])
    plt.plot(freqs[peaks], magnitudes[peaks], 'x')

```

```

plt.title('модуль перетворення Фур\'є з піками')
plt.xlabel('частота')
plt.ylabel('|c_y(k)|')
plt.grid(True)
plt.show()

def generate_model(params, time, freq):
    return (params[0] * time ** 3 + params[1] * time ** 2 + params[2] * time
            + params[3] * np.sin(2 * np.pi * freq[0] * time) + params[4])

def plot_model(time, model):
    plt.figure(figsize=(10, 5))
    plt.plot(time, model)
    plt.title('графік моделі з підібраними параметрами')
    plt.xlabel('час (в секундах)')
    plt.ylabel('y(t)')
    plt.grid(True)
    plt.show()

# відображення графіків
plot_observations(time_values, data)
plot_fourier_transform(frequencies, magnitude_fourier)
plot_peaks(frequencies, magnitude_fourier, peak_indices)

fitted_model = generate_model(rounded_parameters, time_values,
                              filtered_frequencies)
plot_model(time_values, fitted_model)

# середньоквадратична похибка
mse_value = np.mean((data - fitted_model) ** 2)
print(f"середньоквадратична похибка: {mse_value}")

```

## Завантажимо дані з файлу:

```
data = np.loadtxt('/Users/macbookpro/Desktop/MCC/Lab1/f3.txt')
```

## Ініціалізуємо надані нам змінні:

```

# ввід параметрів
T = 5
N = len(data)
delta_t = T / N
time_values = np.linspace(0, T, N)

```

## Обчислимо дискретне перетворення Фур'є :

```

# обчислення перетворення Фур'є
def calculate_dft(signal):
    N = len(signal)
    exp_factor = -2j * np.pi / N
    return np.array([np.sum(signal * np.exp(exp_factor * k * np.arange(N)))
                     for k in range(N)]) / N

fourier_transform = calculate_dft(data)

```

Проведемо аналіз , визначимо частоти коливань , та їх піки . Виділимо пікові точки на графіку :

```
# визначимо частоти
frequencies = np.fft.fftfreq(N, delta_t)
magnitude_fourier = np.abs(fourier_transform)

# визначимо піки
half_range = N // 2
peak_indices, _ = find_peaks(magnitude_fourier[:half_range])

# відбір значущих частот
min_threshold = 1
significant_frequencies = frequencies[peak_indices]
filtered_frequencies =
significant_frequencies[np.abs(significant_frequencies) > min_threshold]
```

Використаємо метод найменших квадратів, щоб знайти невідомі параметри  $a_i, i=1, 2, \dots, k+1$ :

```
# метод найменших квадратів
def sine_wave(time, frequency):
    return np.sin(2 * np.pi * frequency * time)

def construct_matrix_and_vector(time, signal, freq):
    sine_t = sine_wave(time, freq[0])
    time_powers = [np.sum(time ** p) for p in range(7)]
```

Вирішимо рівняння системи графіків :

```
# вирішення системи рівнянь
def solve_least_squares(time, signal, freq):
    matrix, vector = construct_matrix_and_vector(time, signal, freq)
    return np.linalg.solve(matrix, vector)

parameters = solve_least_squares(time_values, data, filtered_frequencies)
rounded_parameters = np.round(parameters).astype(int)
```

Побудуємо графіки :

```
# побудова графіків
def plot_observations(time, signal):
    plt.figure(figsize=(10, 5))
    plt.plot(time, signal)
    plt.title('спостереження y(t) в залежності від часу')
    plt.xlabel(' час (в секундах)')
    plt.ylabel('y(t)')
    plt.grid(True)
    plt.show()

def plot_fourier_transform(fregs, magnitudes):
    plt.figure()
    plt.plot(time_values, np.abs(fourier_transform))
    plt.title('модуль перетворення Фур\''є')
```

```

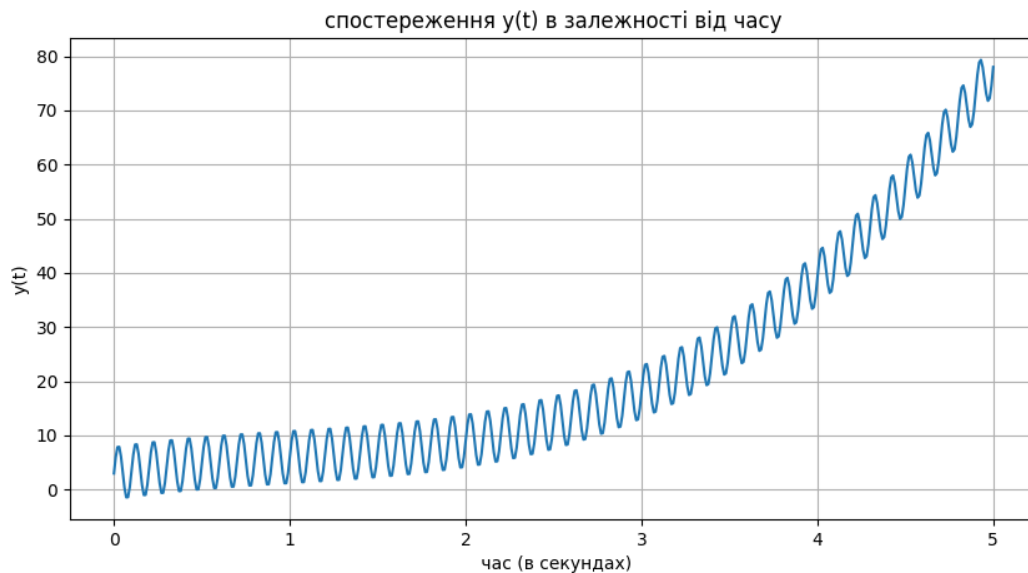
plt.xlabel('частота')
plt.ylabel('|c_y(k)|')
plt.grid(True)
plt.show()

def plot_peaks(freqs, magnitudes, peaks):
    plt.figure(figsize=(10, 5))
    plt.plot(freqs[:half_range], magnitudes[:half_range])
    plt.plot(freqs[peaks], magnitudes[peaks], 'x')
    plt.title('модуль перетворення Фур\'є з піками')
    plt.xlabel('частота')
    plt.ylabel('|c_y(k)|')
    plt.grid(True)
    plt.show()

def generate_model(params, time, freq):
    return (params[0] * time ** 3 + params[1] * time ** 2 + params[2] * time
            + params[3] * np.sin(2 * np.pi * freq[0] * time) + params[4])

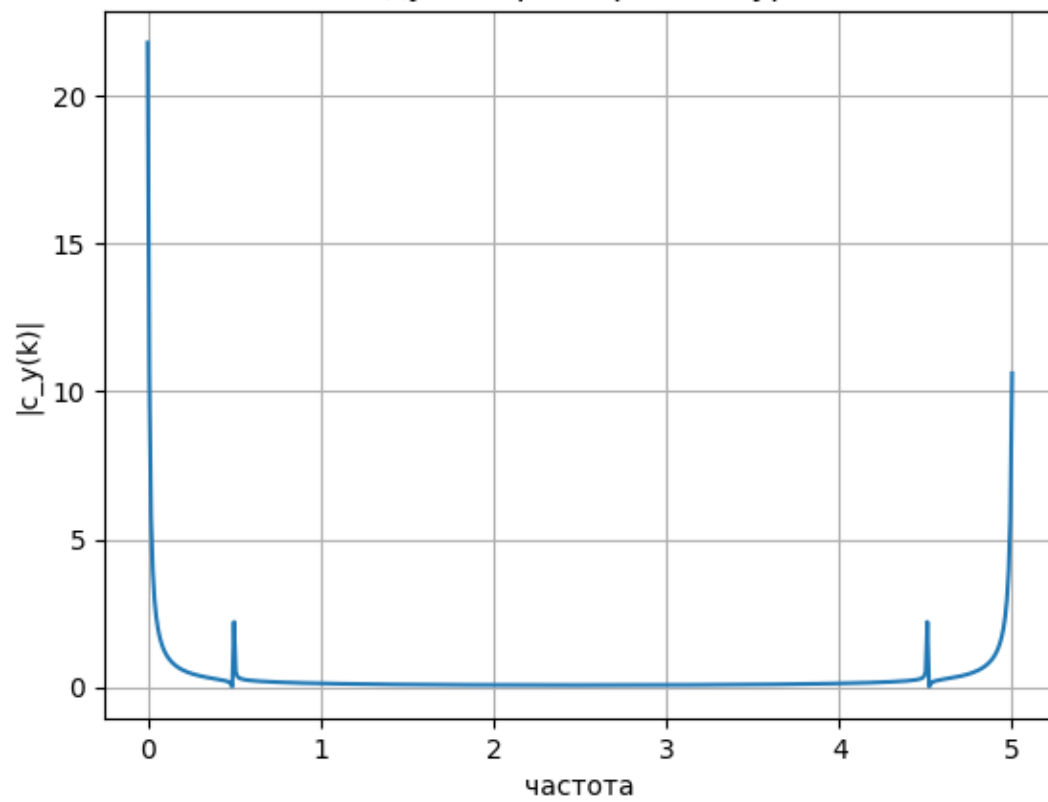
def plot_model(time, model):
    plt.figure(figsize=(10, 5))
    plt.plot(time, model)
    plt.title('графік моделі з підібраними параметрами')
    plt.xlabel('час (в секундах)')
    plt.ylabel('y(t)')
    plt.grid(True)
    plt.show()

```

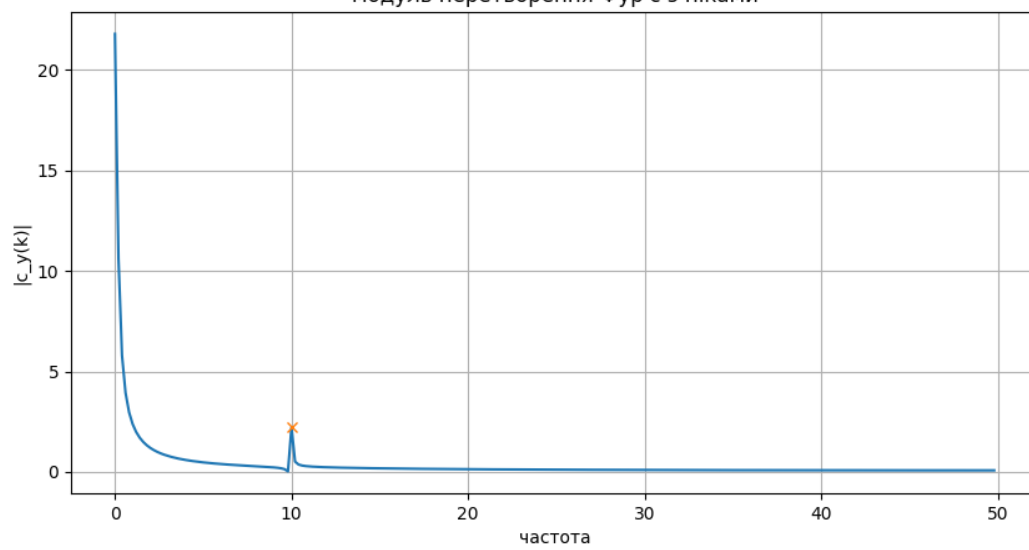




модуль перетворення Фур'є



модуль перетворення Фур'є з піками



Як можемо побачити, найбільший вклад мають частоти 0 та 10. Вклад частоти 0 ми відкидаємо, адже він пов'язаний з вкладом поліноміальної частини нашої моделі.

Вклад частот 10Гц.

Періодична модель нашої моделі матиме вигляд:  $\sin(2 \cdot 10\pi \cdot t)$ .

Вигляд моделі:  $y(t) = a_1 t^3 + a_2 t^2 + a_3 t + a_4 \sin(2 \cdot 10\pi \cdot t) + a_5$ .

**Обчислюємо похибку та будуємо графік:**

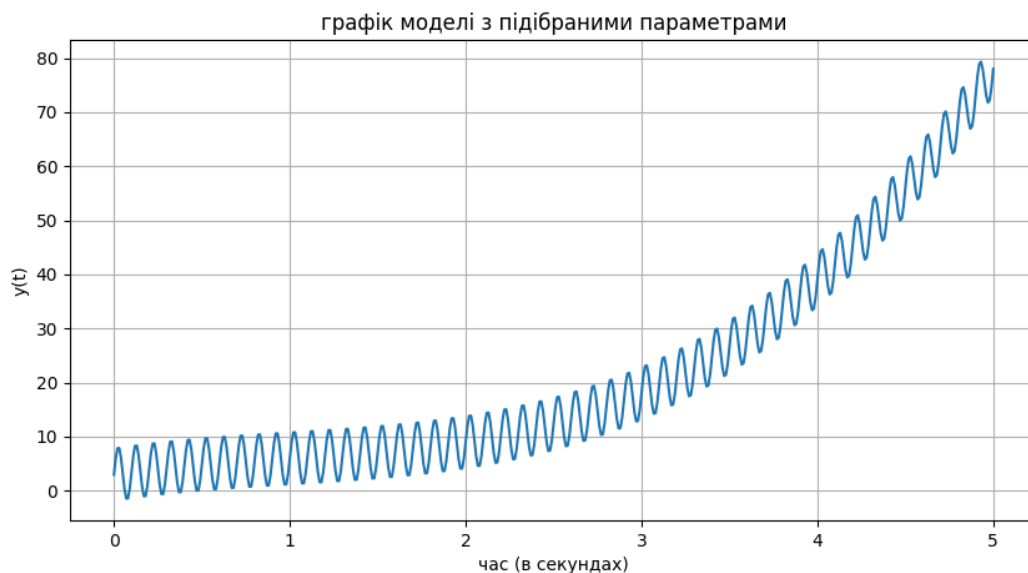
```
# середньоквадратична похибка
mse_value = np.mean((data - fitted_model) ** 2)
print(f"середньоквадратична похибка: {mse_value}")
```

середньоквадратична похибка: 7.195832317432848e-10

**Результати:**

підібрані параметри a: [ 1 -3 5 5 3]

Апроксимуюча функція :  $y(t) = 1 \cdot t^3 + -3 \cdot t^2 + 5 \cdot t + 5 \cdot \sin(2\pi \cdot 10.0 \cdot t) + 3$



**Висновок :**

Під час виконання лабораторної роботи ми :

- вивчили означення дискретного перетворення Фур'є і його властивості.
- створили програму, яка б за допомогою дискретного перетворення Фур'є визначала суттєві вклади частот  $f_i$ ,  $i = 1, 2, \dots, r$  за спостереженнями  $y^*(t_i)$ ,  $i = 1, 2, \dots, N$ . Спостереження записані у файлі, що додається.
- зробити аналіз функції модуля перетворення Фур'є дискретної послідовності  $y^*(t_i)$ ,  $i = 1, 2, \dots, N$  і вивели його графік. Вивели знайдені значення  $f_i$ ,  $i = 1, 2, \dots, r$ .
- створили цей звіт із результатами своєї роботи .