

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ Т. ШЕВЧЕНКА  
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

# Моделювання систем

Лабораторна робота 1

**Виконав**

студент групи ІС-31

А.С. ХОМА

Київ-2018

## Умова

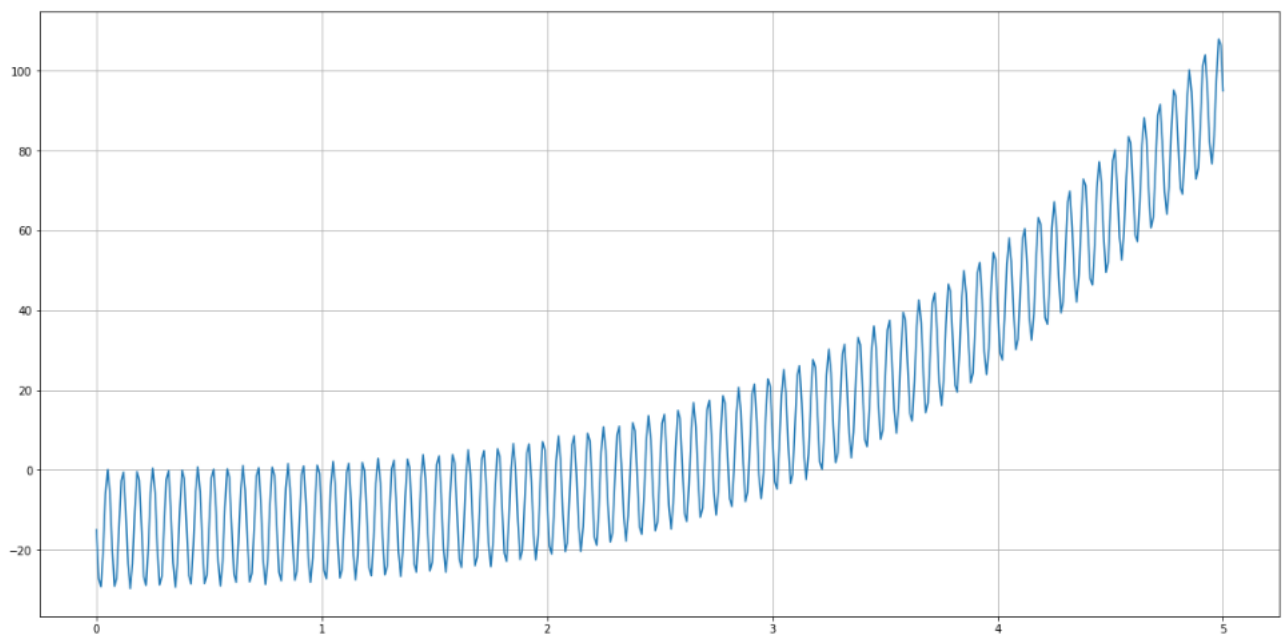
Визначити модель в класі функцій

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

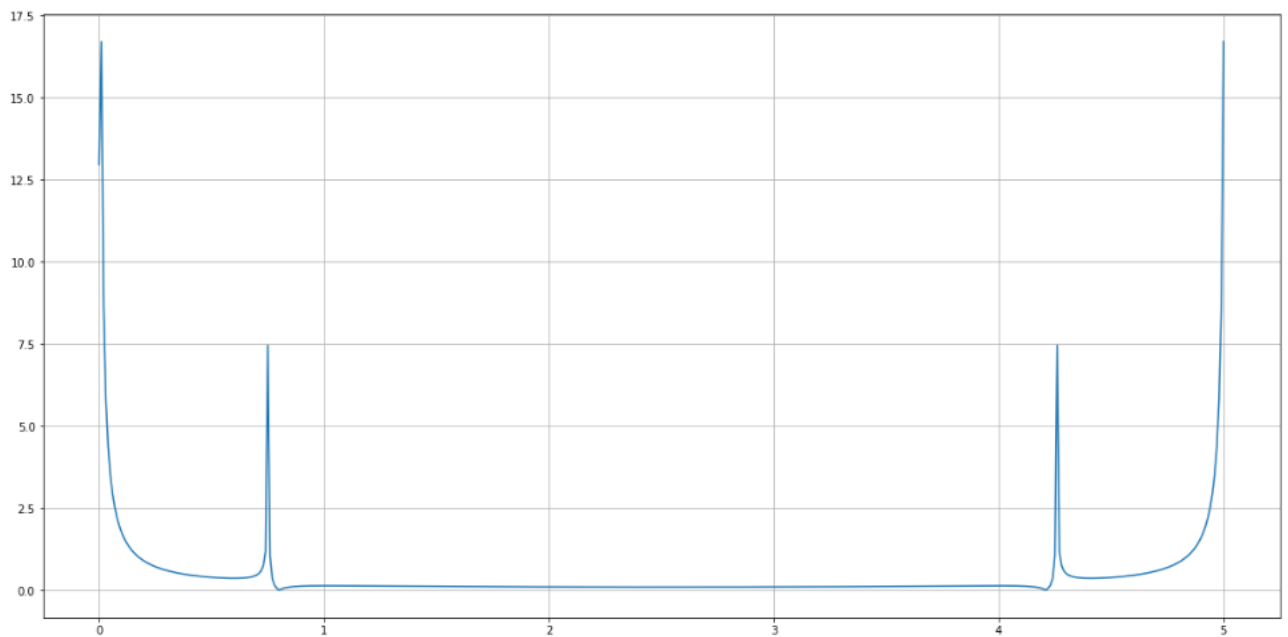
для спостережуваної дискретної функції  $\hat{y}(t_i)$ ,  $i = 1, 2, \dots, N$ ,  $t_{i+1} - t_i = \Delta t = 0.01$ , інтервал спостереження  $[0, T]$ ,  $T = 5$ .

## Хід роботи

### 1. Переглянемо наші дані



### 2. Побудуємо дискретне перетворення Фур'є



3. Визначимо компоненту найбільшого впливу

```
biggest_value = []

for i in range(3, n // 2):
    if np.max(frequency[i - 3:i + 3]) == frequency[i]:
        biggest_value.append(i)
        print(frequency[i])

main_frequency = biggest_value[0] / T
```

4. Розв'яжемо систему рівнянь, де апроксимуємо суму синусів як синус найбільшого впливу перетворення Фур'є

```
b = np.array([np.sum(data * time ** 3), np.sum(data * time ** 2), np.sum(data * time),
              np.sum(data * np.sin(2. * np.pi * main_frequency * time)), np.sum(data)])

a = np.zeros((b.shape[0], b.shape[0]))

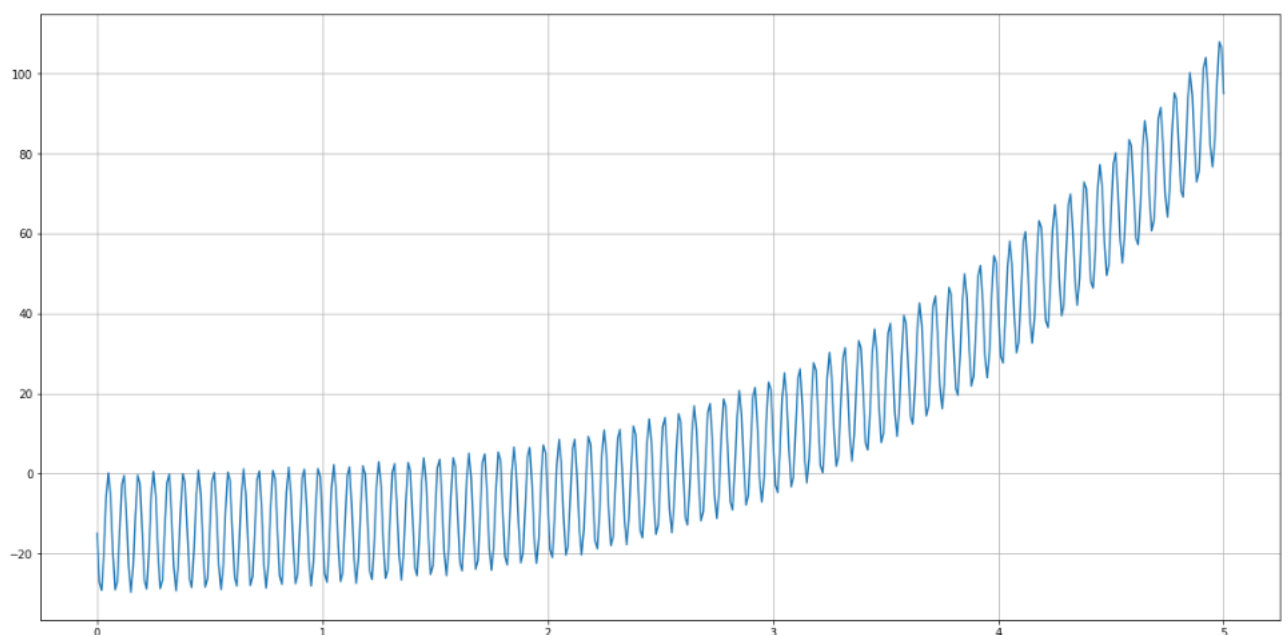
functions = [time ** 3, time ** 2, time, np.sin(2. * np.pi * main_frequency * time), np.ones(n)]

for i in range(b.shape[0]):
    for j in range(b.shape[0]):
        a[i, j] = np.sum(functions[i] * functions[j])

solution = np.linalg.inv(a) @ b.T
```

5. Побудуємо отриманий поліном та порахуємо середньоквадратичну похибку

```
approximated_func = np.dot(solution, functions)
plt.grid(True)
plt.plot(time, approximated_func);
```



```
print("Mean squared error: ", mean_squared_error(data, approximated_func))
```

```
Mean squared error: 6.760852363614448e-10
```