Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Лабораторна робота №1

з предмету «Моделювання систем»

Виконав студент 3-го курсу

Групи ІПС-31

Навка Гліб Олександрович

2021

Завдання

1. Написати програму, яка б за допомогою дискретного перетворення Фур'є визначала суттєві вклади частот f_i, $i = 1, 2, \ldots, r$ за спостереженнями $\hat{y}(t_i)$, $i = 1, 2, \ldots N$. Спостереження записані у файлі, що додається.

2. Записати функціонал похибки ( ), виходячи з кількості знайдених параметрів f_i, $i = 1, 2, \ldots, k - 3$ в першій лабораторній роботі.

3. Записати систему лінійних алгебраїчних рівнянь ( ).

4. Створити програму знаходження a_j, $j = 1, 2, \ldots, k+1$.

Розв’язок

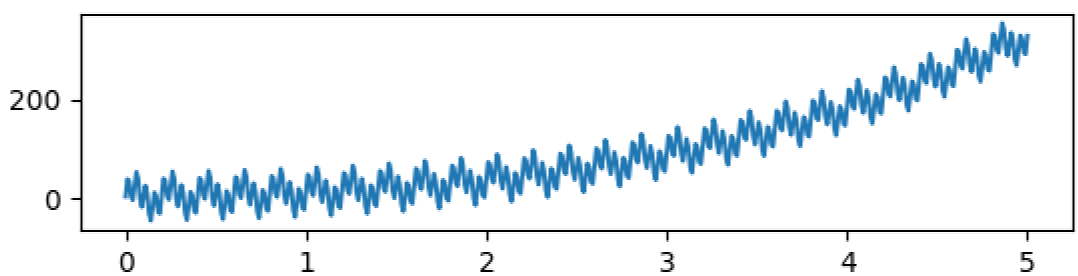
Варіант №5

За умовою $t_{i+1} - t_i = \Delta t = 0.01$

Інтервал спостереження [0, T], T = 5.

```
measurements = np.array(open("f5.txt").read().split(), float)

t = 5
dt = 0.01
time = np.arange(0, t + dt, dt)
```

Будуємо графік спостережень. Від має вигляд:

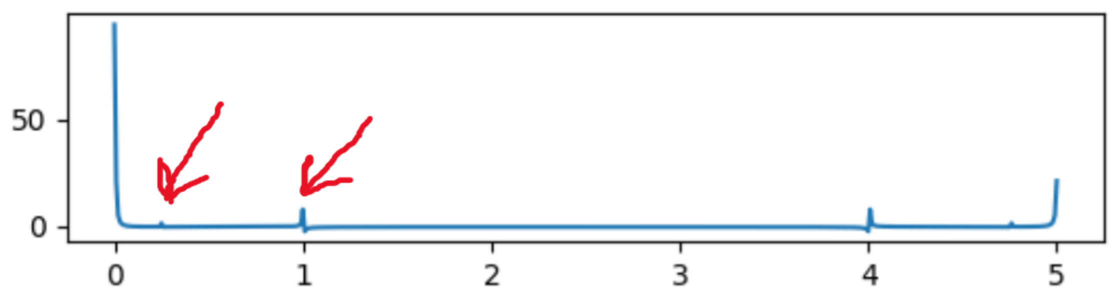


Далі ми будуємо дискретне перетворення Фур’є, яку рахуємо по формулі

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

```
# Discrete Fourier transform
n = time.shape[0]
n_range = np.arange(n)
k = n_range.reshape((n, 1))
m = np.exp(-2j * np.pi * k * n_range / n)

transformed_data = (m @ measurements) / n
transformed_half_data = transformed_data[:transformed_data.shape[0] // 2
- 1]
```

Тепер можемо намалювати графік перетворення Фур’є:



Як бачимо з графіка, локальних максимуми у нас два

```
extremes = np.array(
    argrelextrema(transformed_half_data, np.greater)
)
frequency1 = extremes[0][0] / t
frequency2 = extremes[0][1] / t
```

Значення частот

```
frequency1: 5.0
frequency2: 20.0
```

Записуємо систему лінійних алгебраїчних рівнянь

$$F(a_1, a_2, \ldots, a_{k+1}) =$$
$$= \frac{1}{2} \sum_{j=0}^{N-1} \left(a_1 t_j^3 + a_2 t_j^2 + a_3 t_j + \sum_{i=4}^{k} a_i \sin(2\pi f_{i-3} t_j) + a_{k+1} - \hat{y}(t_j) \right)^2 .$$

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

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

Для цього записуємо систему рівнянь

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

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

functions = [
    time ** 3,
    time ** 2,
    time,
    np.sin(2.0 * np.pi * frequency1 * time),
    np.sin(2.0 * np.pi * frequency2 * time),
    np.ones(n)
]


for i in range(a.shape[0]):
    for j in range(a.shape[1]):
        a[i, j] = np.sum(functions[i] * functions[j])
```

Розв'язуємо цю систему методом Гауса та знаходимо коефіцієнти при частотах a_j

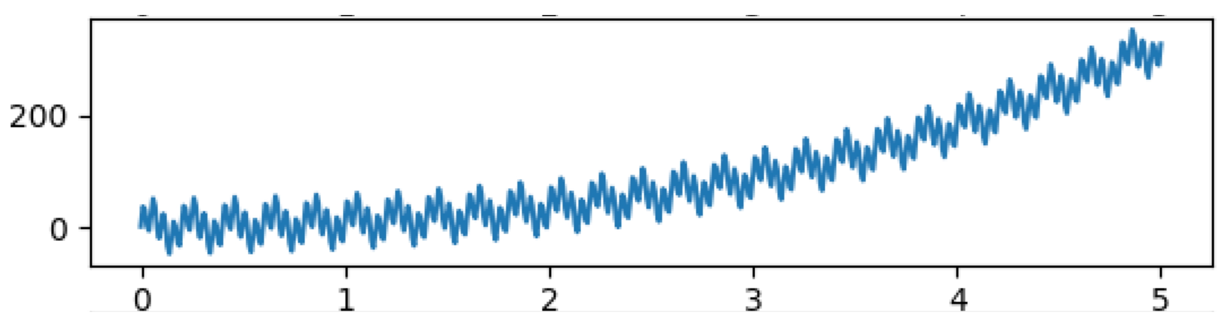
```
c = np.array([
    np.sum(measurements * time ** 3),
    np.sum(measurements * time ** 2),
    np.sum(measurements * time),
    np.sum(measurements * np.sin(2. * np.pi * frequency1 * time)),
    np.sum(measurements * np.sin(2. * np.pi * frequency2 * time)),
    np.sum(measurements)
])

solution = np.linalg.inv(a) @ c
```

Отримав такий розв'язок:

[2.00000095 1.9999931   5.00001328 22.00000314 30.00000021 2.99999345]

Далі можемо записати апроксимуючу функцію та побудувати її графік

```
approximated_functions = solution @ functions
```



Також у завданні сказано знайти середньоквадратичну похибку

```
mse = mean_squared_error(measurements, approximated_functions)
```

Маємо таку відповідь:

9.094032128274329e-10

Висновок

Як бачимо, графік апроксимуючої функції та графік наших спостережень збігаються. Це означає, що отримана відповідь є правильна. Також отримали дуже малу похибку, що також дуже добре.