Київський національний університет імені Тараса Шевченка

Факультет комп’ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Лабораторна робота №1

Виконав студент 3-го курсу

Групи ІПС-32

Михайлов Іван Анатолійович

Варіант №2

2021

Завдання

1. Вивчити означення дискретного перетворення Фур'є і його властивості.

2. Написати програму, яка б за допомогою дискретного перетворення Фур'є визначала суттєві вклади частот за спостереженнями . Спостереження записані у файлі, що додається.

3. Зробити аналіз функції модуля перетворення Фур'є дискретної послідовності і вивести його графік. Вивести знайдені значення .

4. Створити програму знаходження. Вивести відповідні значення функціоналу похибки (2.2), а також графіки спостережень , та математичної моделі (2.1) при знайдених параметрах. Порівняти ці графіки.

5. Оформити в друкованій формі звіт про виконання роботи, в якому викласти результати проведених обчислень.

Теоретичні відомості

1.1 Дискретне паратворення Фур'є

Дискретне перетворення Фур'є для дискретної послідовності $x(j)$, $j = 0, 1, 2, \ldots, N-1$ визначається таким способом

$$c_x(k) = \frac{1}{N} \sum_{m=0}^{N-1} x(m) e^{-i2\pi km/N}. \tag{1.1}$$

Тут $i^2 = -1$ – комплексна одиниця, $e^{i\phi} = \cos\phi + i \sin\phi$.

2.1 Постановка задачі ідентифікації параметрів математичної моделі

Задана математична модель в класі функцій

$$y(t) = a_1 t^3 + a_2 t^2 + a_3 t + \sum_{i=4}^{k} a_i \sin(2\pi f_{i-3} t) + a_{k+1}. \tag{2.1}$$

2.2 Метод найменших квадратів

$$F(a_1, a_2, \ldots, a_{k+1}) =$$
$$= \frac{1}{2} \sum_{j=0}^{N-1} \left(a_1 t_j^3 + a_2 t_j^2 + a_3 t_j + \sum_{i=4}^{k} a_i \sin(2\pi f_{i-3} t_j) + a_{k+1} - \hat{y}(t_j) \right)^2. \tag{2.2}$$

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

$$\frac{\partial F(a_1, a_2, \ldots, a_{k+1})}{\partial a_j} = 0, \tag{2.3}$$

Хід роботи

1. Використовуємо Швидке перетворення Фур'є з бібліотеки та Дискретне перетворення Фур'є.
2. Отримуємо массив комплексних чисел, взяли їх модулі.
3. Отримали графік, що відображає спектр частот від 0 до 100 Гц. Згідно з теоремою Шеннона-Котельникова, для точного відновлення сигналу після дискретизації частота дискретизації повинна бути щонайменше вдвічі більшою за максимальну частоту сигналу. У нашому випадку це частота Найквіста — 50 Гц. Тому для отримання коректних даних обмежуємо спектр частот до діапазону від 0 до 50 Гц, оскільки частоти вище будуть лише дзеркальними копіями нижніх частот і не містять нової інформації.
4. За допомогою вбудованої функції пошуку локальниї максимумів знаходимо індекси значущих частот та їх значення. Створюємо функцію, що відображає модель згідно наданому шаблону (2.1). Додаємо синусоїду з параметром, який знайдений у минулій лабі.
6. Описуємо функціонал похибки.
7. Мінімізуємо за допомогою вбудованного методу найменших квадратів.
8. Формуємо та вирішуємо систему рівннянь беручи частинну похідну та записуючи віжповідні коєфіцієнти у матрицю.
9. Формуємо модель згідно з отриманими параметрами, будуємо графіки порівнюючи отриману систему та вхідні дані.

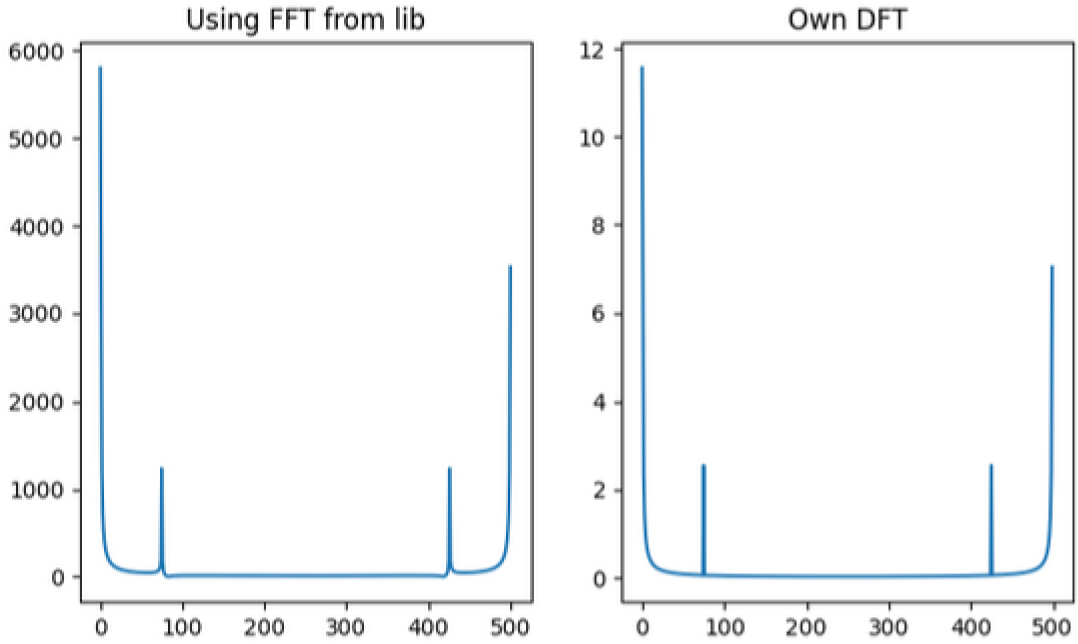
Виконання

Перша частина

Описуємо та використовуємо Дискретне перетворення Фур'є.

Також, для порівняння, використовуємо вбудовану функцію швидкого перетворення Фур'є.

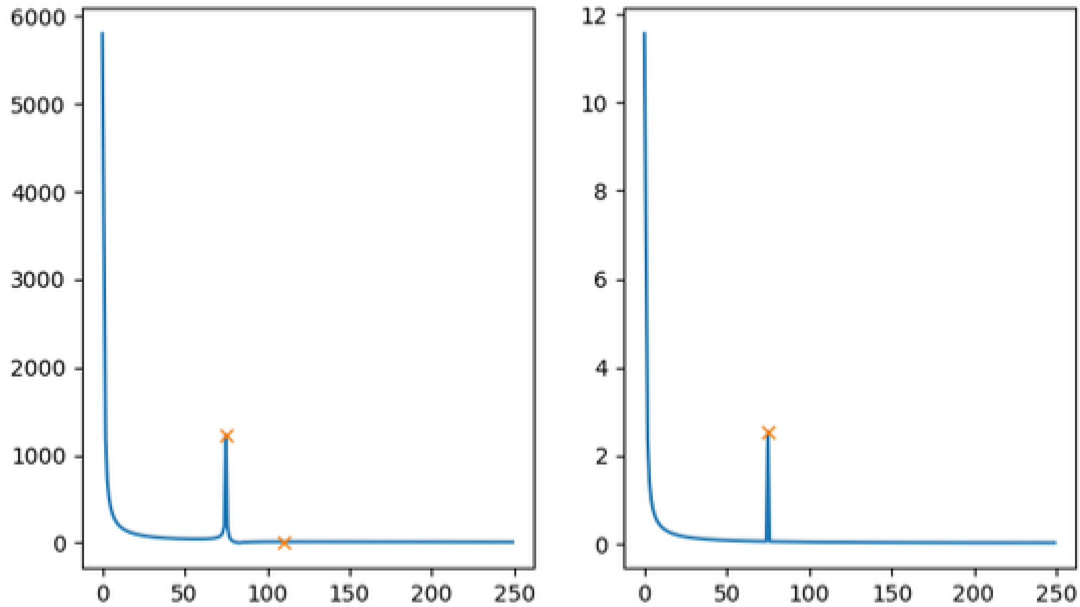
```
dft_result=np.zeros(N,dtype=complex)
for k in range(N):
  for m in range(N):
    dft_result[k] += data_array[m] * np.exp(-1j * np.pi * 2 * k * m / N) / N
```



Обмежуємо спектр згідно з теоремою Шеннона-Котельникова.

Знаходимо індекси локальних максимумів та віповідні частоти.

```
peaks_dft, properties= scipy.signal.find_peaks(np.abs(dft_result)[0:250])
frequencies_dft = [0.2 * num for num in peaks_dft]
print(frequencies_dft)
```



Отримуємо індекси значущих частот та їх величину.

Варто зауважити, що при використанні швидко перетворення Фур’є, ми, отримали 2 локальних максимуми аналітично, але, очевидно, що значуща частота лише 1.

Отриуємо значущу частоту 15Гц.

Друга частина

Формуємо модель згідно з заданим шаблоном, формуючи синусоїду згідно зі знайденою нами частотою 15Гц.

```
# Build the sinusoidal component
f_sin = np.sin(2 * np.pi * frequencies_dft[0] * t)
```

Записуємо повну модель та функціонал похибки, використовуємо вбудований метод найменших квадратів для мінімізації функцоіналу похибки.

```
def model(a, t):
    a1, a2, a3, a4, a5 = a
    return (a1 * t**3 + a2 * t**2 + a3 * t +
            a4 * np.sin(2 * np.pi * 15 * t) + a5)

def residuals(a, t, actual_vals):
    y_pred = model(a, t)
    return y_pred - actual_vals

t = t_vals
actual_vals = data_array
initial_guess = np.zeros(5)

result = least_squares(residuals, initial_guess, args=(t, actual_vals))
optimal_params = result.x
```

Фіксуємо параметри отримані вбудованою функцією.
least_squares: [1.00000054 -7.00000381 5.00000766 5.00000828 2.99999574]

Власноруч візьмемо похідні, та побудуємо СЛАУ для вирішення задачі.

```
# Build the sinusoidal component
f_sin = np.sin(2 * np.pi * frequencies_dft[0] * t)

# Construct the system of equations
A = np.array([
    [np.sum(t**6), np.sum(t**5), np.sum(t**4), np.sum(f_sin * t**3), np.sum(t**3)],
    [np.sum(t**5), np.sum(t**4), np.sum(t**3), np.sum(f_sin * t**2), np.sum(t**2)],
    [np.sum(t**4), np.sum(t**3), np.sum(t**2), np.sum(f_sin * t), np.sum(t)],
    [np.sum(f_sin * t**3), np.sum(f_sin * t**2), np.sum(f_sin * t), np.sum(f_sin * f_sin), np.sum(N * f_sin)],
    [np.sum(t**3), np.sum(t**2), np.sum(t), np.sum(N * f_sin), N]
])

c = np.array([
    np.sum(y * t**3),
    np.sum(y * t**2),
    np.sum(y * t),
    np.sum(y * f_sin),
    np.sum(y)
])
```

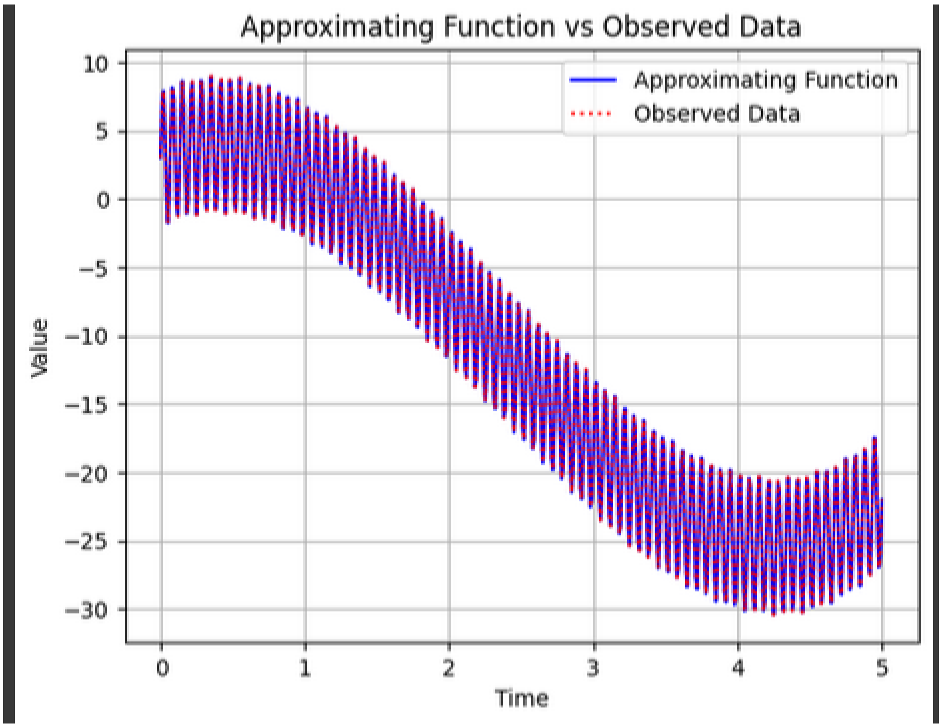
Використаємо numpy.linalg.solve та отримуємо такі параметри.

Manually: [1.00000054 -7.00000381 5.00000766 5.00000828 2.99999574]

Отже результати отримані, запишемо отриману модель.

$$y(t) = 1t^3 - 7t^2 + 5t + 5\sin(30\pi t) + 3$$

Продемонструємо вихідну модель та вхідні дані на графіку.



Бачимо, що модель точно передала нам вхідні значення.

Висновок

Практичним шляхом дослідили як описувати вхідні сигнали за допомогою математичних моделей.

Використали метод найменших квадратів для знаходження відповідних коефіцієнтів моделі та мінімізації похибки.