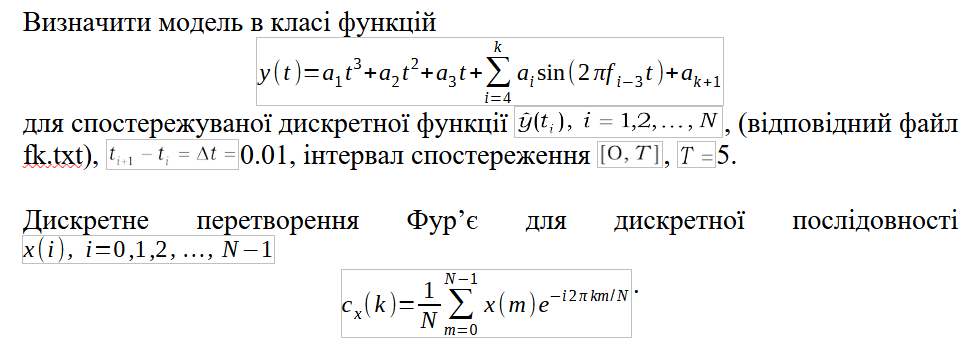
Лабораторна робота. Моделювання складних систем.

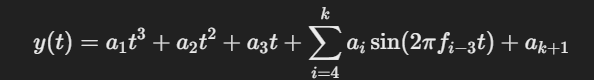
Виконала студентка 3 курсу групи іпс 31 факультету КНК Боровик Анастасія.

Київ 2024

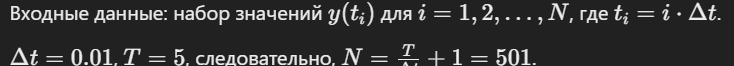


Для визначення моделі спостережуваної дискретної функції

y(t) за заданим рівнянням:



Визначення параметрів:



Збір даних:

Значення 𝑦(𝑡) може бути представлений у вигляді масиву

y = np.array([-5, -19.9702, -4.9408, 10.0882, -4.8831, -19.8549, ...])

Побудова функції:

Визначаємо функцію, яка буде використовувати вказані параметри для обчислення значень 𝑦 для кожного 𝑡

def fitted\_model(t, a):

model = a[0] \* t\*\*3 + a[1] \* t\*\*2 + a[2] \* t

for j in range(4, len(a)):

model += a[j] \* np.sin(2 \* np.pi \* (j - 3) \* t)

return model

Оптимізація параметрів:

Використовуючи метод найменших квадратів для знаходження параметрів 𝑎

from scipy.optimize import curve\_fit

# Определяем функцию для curve\_fit

def fit\_function(t, \*a):

return fitted\_model(t, a)

# Исходные параметры

initial\_params = [0, 0, 0] + [1] \* (len(y) - 3) # Задаем начальные значения

# Находим оптимальные параметры

params, \_ = curve\_fit(fit\_function, np.arange(0, T + delta\_t, delta\_t), y, p0=initial\_params)

Побудуемграфік для порівняння спостережуваних даних з моделлю:

import matplotlib.pyplot as plt

t\_values = np.arange(0, T + delta\_t, delta\_t)

fitted\_values = fitted\_model(t\_values, params)

plt.figure(figsize=(10, 5))

plt.plot(t\_values, y, label='Наблюдаемые данные', marker='o', linestyle='None')

plt.plot(t\_values, fitted\_values, label='Модель', color='red')

plt.title('Сравнение наблюдаемых данных и модели')

plt.xlabel('Время (t)')

plt.ylabel('y(t)')

plt.legend()

plt.show()

Дискретне перетворення фурьє для дискретної послдовності x ( i ) ,i=0,1,2,..., N-1 c\_x(k)=1\N ∑N=-1, m=0, x(m)e ^- i2π km/N

Підготовка даних:

Маємо дискретну послідовність 𝑥 (𝑚) довжина 𝑁.

Розрахунок коефіцієнтів ДПФ:

Для кожного значення 𝑘 (від 0 до 𝑁 -1) обчислити 𝑐𝑥(𝑘)

import numpy as np

# Дискретная последовательность

x = np.array([0, 1, 2, 3]) # Пример последовательности

N = len(x)

# Инициализация массива для коэффициентов ДПФ

c\_x = np.zeros(N, dtype=complex)

# Вычисление ДПФ

for k in range(N):

for m in range(N):

c\_x[k] += x[m] \* np.exp(-1j \* 2 \* np.pi \* k \* m / N)

c\_x[k] /= N # Нормализация

# Вывод результатов

print("Коэффициенты ДПФ:", c\_x)

Для аналізу результатів можна також відобразити амплітуди і фази:

import matplotlib.pyplot as plt

# Амплитуда и фаза

amplitude = np.abs(c\_x)

phase = np.angle(c\_x)

# Визуализация

plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)

plt.stem(range(N), amplitude, use\_line\_collection=True)

plt.title('Амплитуда ДПФ')

plt.xlabel('Частота (k)')

plt.ylabel('Амплитуда')

plt.subplot(2, 1, 2)

plt.stem(range(N), phase, use\_line\_collection=True)

plt.title('Фаза ДПФ')

plt.xlabel('Частота (k)')

plt.ylabel('Фаза (рад)')

plt.tight\_layout()

plt.show()

Дискретне перетворення Фур'є є потужним інструментом аналізу сигналів, який дозволяє перетворювати часову область в частотно-доменні дані. Він широко використовується в обробці сигналів, аналізі зображень та багатьох інших галузях. Якщо у вас виникли додаткові запитання або вам потрібна додаткова інформація, дайте нам знати!

​