Модели дискретных сигналов. Задача 1 (отборочный этап)

Постановка задачи

Известно, что дискретный сигнал имеет вид

$$x_k = A\sin(\omega k + \varphi), \tag{1}$$

где $\omega \in [2\pi, 3\pi]$, $\varphi \in \left[0, \frac{\pi}{2}\right]$, а также выполнена периодичность

$$x_{k+N} = x_k \quad \forall k \in \mathbb{Z}. \tag{2}$$

В файле (приложение A.txt) дана последовательность чисел $\{x_k\}$; требуется определить параметры ω, A, φ. Численные ответы представить с точностью до 0.0001; приложить рассуждения, аналитические выкладки, листинг программы и график.

1 Исходные данные и выбор периода

Из файла считывается L чисел (длина файла). Файл содержит несколько повторов одного и того же периода сигнала, то есть значения идут блоками длины N и повторяются. Фундаментальный период N находится программно как минимальный делитель L, при котором блоки длины N совпадают с допуском порядка 10^{-8}. Для данного набора данных получено $N = 20$. Дальнейшие вычисления параметров выполняются по одному периоду $\{x_k\}_{k=0}^{N-1}$, то есть используется срез $x[:N]$.

2 Аналитическая часть

2.1 Связь периода и угловой частоты

Подставим (1) в условие периодичности (2):

$$A\sin(\omega(k+N) + \varphi) = A\sin(\omega k + \varphi) \quad \forall k.$$

Так как $\sin(\cdot)$ имеет период 2π, равенство выполняется при

$$\omega N = 2\pi m, \quad m \in \mathbb{Z}. \tag{3}$$

Отсюда:

$$\omega = \frac{2\pi m}{N}. \tag{4}$$

С учётом ограничения $\omega \in [2\pi, 3\pi]$ и найденного $N = 20$:

$$2\pi \le \frac{2\pi m}{20} \le 3\pi \quad \Rightarrow \quad 20 \le m \le 30.$$

Таким образом, перебираются целые $m = 20, 21, \ldots, 30$.

2.2 Линеаризация и оценивание параметров

Модель (1) приводится к линейной форме по параметрам:

$$A \sin(\omega k + \varphi) = A\left(\sin(\omega k)\cos\varphi + \cos(\omega k)\sin\varphi\right) = B\sin(\omega k) + C\cos(\omega k),$$

где $B = A\cos\varphi$, $C = A\sin\varphi$. Для фиксированного ω оцениваются B и C методом наименьших квадратов:

$$\min_{B,C} \sum_{k=0}^{N-1} \left(x_k - (B\sin(\omega k) + C\cos(\omega k))\right)^2.$$

После нахождения B и C восстанавливаем

$$A = \sqrt{B^2 + C^2}, \qquad \varphi = \operatorname{atan2}(C, B). \tag{5}$$

Значение φ после atan2 приводится к диапазону $[0, 2\pi)$ (при необходимости добавлением 2π). По условию задачи выбирается представитель в $[0, \pi/2]$; в полученном решении $\varphi \approx 0.81$ уже лежит в требуемом интервале.

2.3 Критерий выбора m

Для каждого m строится приближение $\hat{x}_k$ и вычисляется среднеквадратичная ошибка:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(x_k - \hat{x}_k)^2}.$$

Выбирается m, дающий минимальное значение RMSE.

3 Численные результаты

В результате перебора $m \in \{20, \dots, 30\}$ минимум ошибки достигается при $m = 23$. Тогда

$$\omega = \frac{2\pi \cdot 23}{20} = 7.225663103256524\ldots$$

и

$$A = 17.00000000000001\ldots, \qquad \varphi = 0.8099999999999954\ldots$$

Ошибка аппроксимации:

$$\mathrm{RMSE} \approx 7.845656339423094 \cdot 10^{-14}.$$

Ответ (точность 0.0001)

$$\boxed{\omega = 7.2257}, \qquad \boxed{A = 17.0000}, \qquad \boxed{\varphi = 0.8100}.$$

4 График

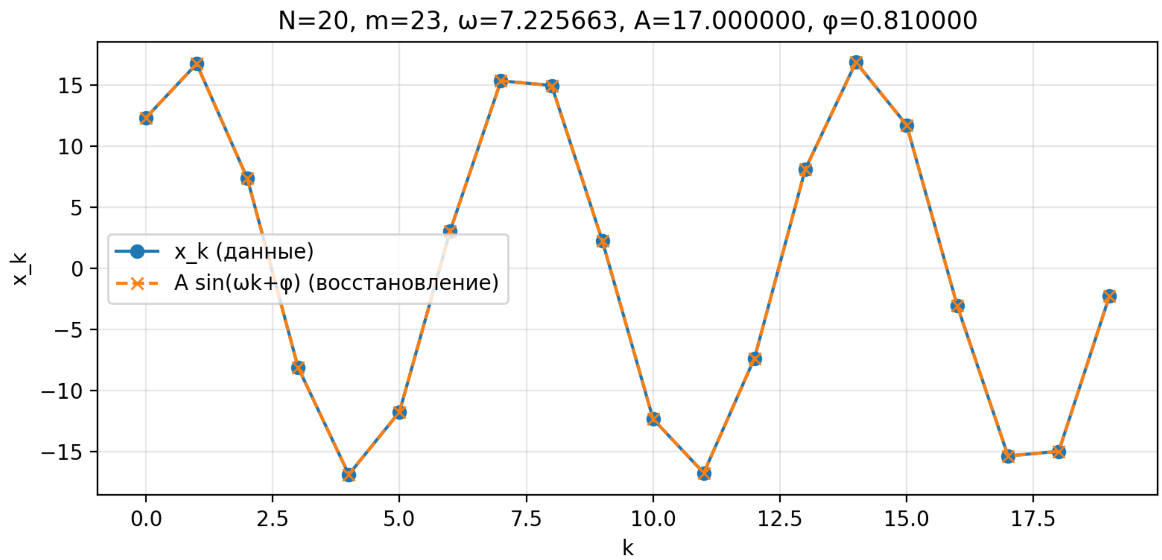

Рис. 1: Сравнение x_k (данные) и $\hat{x}_k = A\sin(\omega k + \varphi)$ на одном периоде ($N = 20$).

5 Листинг программы (Python)

```
import numpy as np, math, pathlib
import numpy.linalg as la
import matplotlib.pyplot as plt

path = "A.txt"

x = np.array([float(v) for v in pathlib.Path(path).read_text().split()])
L = len(x)

def detect_period(x, tol=1e-8):
    L = len(x)
    for p in range(1, L + 1):
        if L % p != 0:
            continue
        ok = True
        for i in range(0, L - p, p):
            if np.max(np.abs(x[i:i+p] - x[i+p:i+2*p])) > tol:
                ok = False
                break
        if ok:
            return p
    return L

N = detect_period(x)
xN = x[:N]
k = np.arange(N)

```

```
m_min = int(math.ceil(1.0 * N))
m_max = int(math.floor(1.5 * N))

best = None
for m in range(m_min, m_max + 1):
    omega = 2 * math.pi * m / N
    M = np.column_stack([np.sin(omega * k), np.cos(omega * k)])

    B, C = la.lstsq(M, xN, rcond=None)[0]
    xhat = M @ np.array([B, C])

    rmse = la.norm(xhat - xN) / math.sqrt(N)
    A = math.hypot(B, C)
    phi = math.atan2(C, B)

    cand = (rmse, m, omega, A, phi, B, C, xhat)
    if best is None or cand[0] < best[0]:
        best = cand

rmse, m, omega, A, phi, B, C, xhat = best

print("N =", N)
print("m =", m)
print("omega =", omega)
print("A =", A)
print("phi =", phi)
print("RMSE =", rmse)

print("omega_4dp =", "{:.4f}".format(omega))
print("A_4dp =", "{:.4f}".format(A))
print("phi_4dp =", "{:.4f}".format(phi))

plt.figure(figsize=(8, 4))
plt.plot(k, xN, marker='o', linestyle='-', label='data x_k')
plt.plot(k, xhat, marker='x', linestyle='--', label='fit A*sin(omega*k+phi)')
plt.xlabel("k")
plt.ylabel("x_k")
plt.title("N={}, m={}, omega={:.6f}, A={:.6f}, phi={:.6f}".format(N, m, omega, A, phi))
plt.grid(True, alpha=0.3)
plt.legend()
plt.tight_layout()
plt.savefig("fit_plot.png", dpi=200)
plt.close()
```