Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Кафедра інтелектуальних програмних систем

Моделювання складних систем

Звіт

з лабораторної роботи №3

студентки 3-го курсу

групи ІПС-31

Совгирі Анни Олегівни

Варіант №5

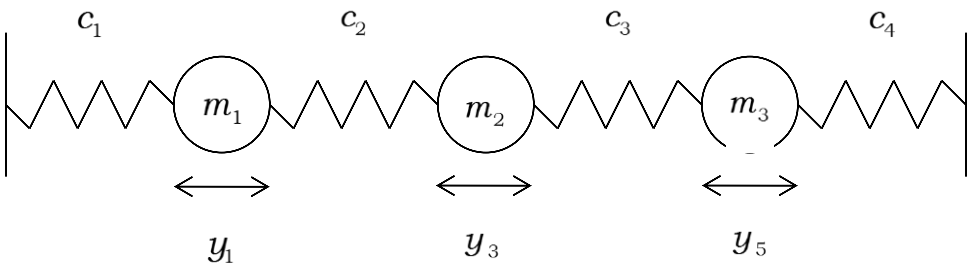
Київ

2025

Параметрична ідентифікація параметрів з використанням функцій чутливості.

Постановка задачі

Для математичної моделі коливання трьох мас m_1, m_2, m_3, які поєднані між собою пружинами з відповідними жорсткостями c_1, c_2, c_3, c_4, і відомої функції спостереження координат моделі $\overline{y}(t),\ t \in [t_0, t_k]$ потрібно оцінити частину невідомих параметрів моделі з використанням функції чутливості.



Математична модель коливання трьох мас описується наступною системою

$$\frac{dy}{dt} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{(c_2 + c_1)}{m_1} & 0 & \frac{c_2}{m_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{c_2}{m_2} & 0 & -\frac{(c_2 + c_3)}{m_2} & 0 & \frac{c_3}{m_2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{c_3}{m_3} & 0 & -\frac{(c_4 + c_3)}{m_3} & 0 \end{pmatrix} y = Ay\,.$$

Показник якості ідентифікації параметрів невідомих параметрів β має вигляд

$$I(\beta) = \int_{t_0}^{t_k} (\overline{y}(t) - y(t))^T (\overline{y}(t) - y(t))dt\,.$$

Якщо представити вектор невідомих параметрів $\beta = \beta_0 + \Delta\beta$, де β_0 –початкове наближення вектора параметрів,

$$\Delta\beta = \left(\int_{t_0}^{t_k} U^T(t) U(t) dt \right)^{-1} \int_{t_0}^{t_k} U^T(t) (\overline{y}(t) - y(t))dt\,.$$

Матриці чутливості $U(t)$ визначається з наступної матричної системи диференціальних рівнянь

$$\frac{dU(t)}{dt} = \frac{\partial(Ay)}{\partial y^T} U(t) + \frac{\partial(Ay)}{\partial \beta^T},$$

$$U(t_0) = 0,\ \beta = \beta_0.$$

В даному випадку $\dfrac{\partial(Ay)}{\partial y^T} = A$.

Спостереження стану моделі проведені на інтервалі часу $t_0 = 0,\ t_k = 50,\ \Delta t = 0.2$
.
Для чисельного інтегрування застосувати метод Рунге-Кутта 4-го порядку:

$$\frac{dy}{dt} = f(y,t), y(t_0) = y_0 ,$$

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4),$$

де

$$k_1 = hf(y_n, t_n) .$$

$$k_2 = hf(y_n + \frac{1}{2}k_1, t_n + \frac{1}{2}h),$$

$$k_3 = hf(y_n + \frac{1}{2}k_2, t_n + \frac{1}{2}h),$$

$$k_4 = hf(y_n + k_3, t_n + h),$$

$$t_{n+1} = t_n + h .$$

Варіант 5:

Вектор оцінюваних параметрів $\beta = (c_1, c_4, m_2)^T$, початкове наближення $\beta_0 = (0.1,\ 0.08,\ 21)^T$, відомі параметри $c_2 = 0.3, c_3 = 0.2$, $m_1 = 12,\ m_3 = 18$, ім’я файлу з спостережуваними даними y5.txt.

Теоретичні відомості

Параметрична ідентифікація — це процес визначення невідомих параметрів моделі за експериментальними даними. У нашій задачі потрібно знайти параметри

$$\beta = (c_i, m_j)^T,$$

які найбільш точно відтворюють спостереження $\bar{y}(t)$.

Для цього вводиться **функція якості (похибки)**:

$$I(\beta) = \int_0^T (\bar{y}(t) - y(t,\beta))^\top (\bar{y}(t) - y(t,\beta))\, dt.$$

Ідея методу — знайти таке β, при якому $I(\beta)$ мінімальне.

Функції чутливості

Функція чутливості показує, як змінюється розв’язок моделі при малих змінах параметрів:

$$U(t) = \frac{\partial x(t, \beta)}{\partial \beta}.$$

Це матриця розмірності (кількість станів) × (кількість параметрів).

Функція чутливості задовольняє власному диференціальному рівнянню, яке інтегрується **паралельно** з основною системою.

$$\dot{U}(t) = A(\beta)\, U(t) + \frac{\partial (Ax)}{\partial \beta}, \quad U(0) = 0.$$

Метод Рунге–Кутти

Для інтегрування основної системи та рівняння чутливостей використовується метод Рунге–Кутти 4-го порядку, який забезпечує баланс між точністю та швидкістю.
На одному кроці він використовує чотири проміжні оцінки:

$$x_{n+1} = x_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).$$

Цей же метод застосовується до матриці чутливостей *U(t).*

Алгоритм виконання

Програмна реалізація параметричної ідентифікації виконується за таким алгоритмом:

1. Зчитування даних

1. Завантажуються експериментальні дані з файлу *y5.txt*.
2. Формується матриця спостережень $\bar{y}(t)$, де кожний рядок відповідає моменту часу.

2. Ініціалізація параметрів

1. Задаються відомі коефіцієнти моделі (жорсткості та маси згідно з варіантом).
2. Вибирається початкове наближення для параметрів, які необхідно оцінити, тобто початковий вектор

$$\beta_0 = (c_1^{(0)}, c_4^{(0)}, m_2^{(0)})^T.$$

3. Встановлюється крок інтегрування та допустима похибка ε.

3. Ітераційний процес (головний цикл)

Цикл уточнення параметрів виконується до досягнення заданої точності.

1) Формування матриці моделі

На поточній ітерації формуються:

- матриця системи A(β), яка описує модель у вигляді системи 1-го порядку;
- допоміжні вирази, необхідні для обчислення часткових похідних, що входять до рівняння чутливості.

2) Чисельне інтегрування

На кожній ітерації паралельно інтегруються:

- **основна система рівнянь руху** для отримання траєкторії $\bar{y}(t)$;
- **система рівнянь чутливості**, що дозволяє обчислити матрицю

$$U(t) = \frac{\partial x(t)}{\partial \beta}.$$

Обидві системи інтегруються методом Рунге–Кутта 4-го порядку на всьому часовому інтервалі.

3) Обчислення інтегральних величин

На основі отриманих значень $y(t)$ та $U(t)$:

- Обчислюються інтеграли, що входять до нормальних рівнянь методу найменших квадратів:

$$\int_0^T U^\top U\, dt, \qquad \int_0^T U^\top (\bar{y} - y)\, dt.$$

- Інтегрування виконується чисельно — шляхом сумування значень на часовій сітці.

4) Корекція параметрів

За розрахованою системою рівнянь визначається вектор поправок:

$$\Delta\beta = \left[\int_0^T U^\top U\, dt\right]^{-1} \left[\int_0^T U^\top (\bar{y} - y)\, dt\right].$$

Далі параметри оновлюються:

$$\beta_{\text{new}} = \beta_{\text{old}} + \Delta\beta.$$

Це наближає модель до експериментальних даних.

4. Перевірка критерію зупинки

Ітераційний процес завершується, якщо виконується хоча б одна з умов:

- інтегральна похибка стала меншою за ε;
- норма поправки параметрів $\|\Delta\beta\|$ є нижчою за поріг точності.

У разі невиконання умов параметри переносяться в наступну ітерацію, і процес повторюється.

5. Формування результатів

Після завершення роботи алгоритму:

1. Виводяться оцінені значення параметрів.
2. Обчислюється фінальний показник якості.
3. Будуються графіки порівняння моделі та експериментальних даних.
4. Результати зберігаються у вигляді логів і зображень.

Програмна реалізація

Програмна частина роботи реалізована мовою *Python* із використанням бібліотек ***NumPy*** (векторизовані обчислення та робота з матрицями), ***Matplotlib*** (побудова графіків для порівняння моделі з експериментальними даними), ***time*** (вимірювання тривалості виконання алгоритму), ***os*** (створення службових каталогів і робота з файлами), а також ***csv*** (збереження даних ітерацій у вигляді таблиць для подальшого аналізу). Нижче наведено основні модулі програми:

`make_logs_folder()`

- Перевіряє наявність директорії `logs/`.
- У разі її відсутності створює папку для збереження проміжних результатів, таблиць *.csv* та графіків.

`load_observations(filename)`

- Зчитує експериментальні дані з текстового файлу (`y5.txt`).
- Построчно перетворює дані у числа типу `float`.
- Формує масив, де рядки відповідають моментам часу, а стовпці — компонентам вектора стану.
- Виконує транспонування, щоб отримати матрицю спостережень розмірності (N,6).

`build_system_matrix(params)`

- Формує матрицю системи A(β) розмірності 6×6 для моделі трьох мас та чотирьох пружин.
- Підставляє у формули значення мас ***m1,m2,m3*** та жорсткостей пружин ***c1,…,c4***.

`rk4_step_state(a_matrix, y_current, dt)`

- Реалізує один крок **методу Рунге–Кутти 4-го порядку** для основної системи диференціальних рівнянь.
- За відомим значенням стану $y(t)$ та матрицею A обчислює наближення $y(t + dt)$.

`rk4_step_sensitivity(a_matrix, source_term, u_current, dt)`

- Аналогічно до попередньої функції, виконує крок методу Рунге–Кутти для **матриці чутливостей** *U(t)*.
- Дає змогу чисельно отримати функції чутливості на всьому часовому інтервалі.

`numerical_jacobian(y_vec_func, beta_names, beta_dict, delta)`

- Обчислює чисельну матрицю похідних $\frac{\partial(Ay)}{\partial\beta}$ методом центральних скінченних різниць.
- Для кожного параметра з вектора β робиться “крок” +δ та −δ, після чого обчислюється різниця значень.
- Повертає матрицю Якобі, яка використовується в рівнянні для *U(t)*.

`simulate_model(params, y0, t_grid, dt)`

- Використовується для побудови фінальних графіків після ідентифікації параметрів.

`identify_parameters(meas_data, fixed_params, beta_names, beta_initial, eps, dt, max_iterations)`

Це **головна обчислювальна функція**, яка реалізує метод параметричної ідентифікації. На основі експериментальних даних та початкових значень параметрів:

- формує матрицю системи A(β) для поточного набору параметрів;
- обчислює чисельні часткові похідні ∂(A*y*)/∂β;
- паралельно інтегрує основну систему та рівняння чутливостей методом Рунге–Кутта;
- накопичує інтеграли, що входять до нормальних рівнянь;
- визначає поправку параметрів Δβ та оновлює вектор β;
- перевіряє критерій зупинки (похибка моделі або мала норма поправки).

Функція виконує цикл уточнення доти, доки не досягнено потрібної точності, після чого повертає знайдені параметри та статистику роботи алгоритму.

`plot_comparison(measured, simulated, t_grid, filename_prefix)`

- Будує порівняльні графіки експериментальних даних (`measured`) та моделі (`simulated`) для всіх шести компонент стану.

`main()`

- Створює службову директорію `logs/`.
- Зчитує експериментальні дані з файлу `y5.txt`.
- Задає відомі параметри варіанту (c2,c3,m1,m3 та початковий вектор невідомих параметрів $\beta_0 = (c_1, c_4, m_2)^T$.
- Викликає функцію **`identify_parameters`** для оцінки параметрів.
- За знайденими параметрами обчислює траєкторію моделі функцією **`simulate_model`**.
- Виводить результати в консоль (початкові й кінцеві параметри, значення функції якості, час виконання).
- Викликає **`plot_comparison`** для візуалізації збігу моделі з експериментальними даними.

Результати

```
Parameter Identification Results
--------------------------------
Initial beta:
c1: 0.100000
c4: 0.080000
m2: 21.000000

Estimated beta:
c1: 0.140000
c4: 0.120000
m2: 27.999993

Metrics:
Cost function I(β): 4.273167e-07
Iterations: 4
Time: 0.08 s
```

Початкове наближення параметрів

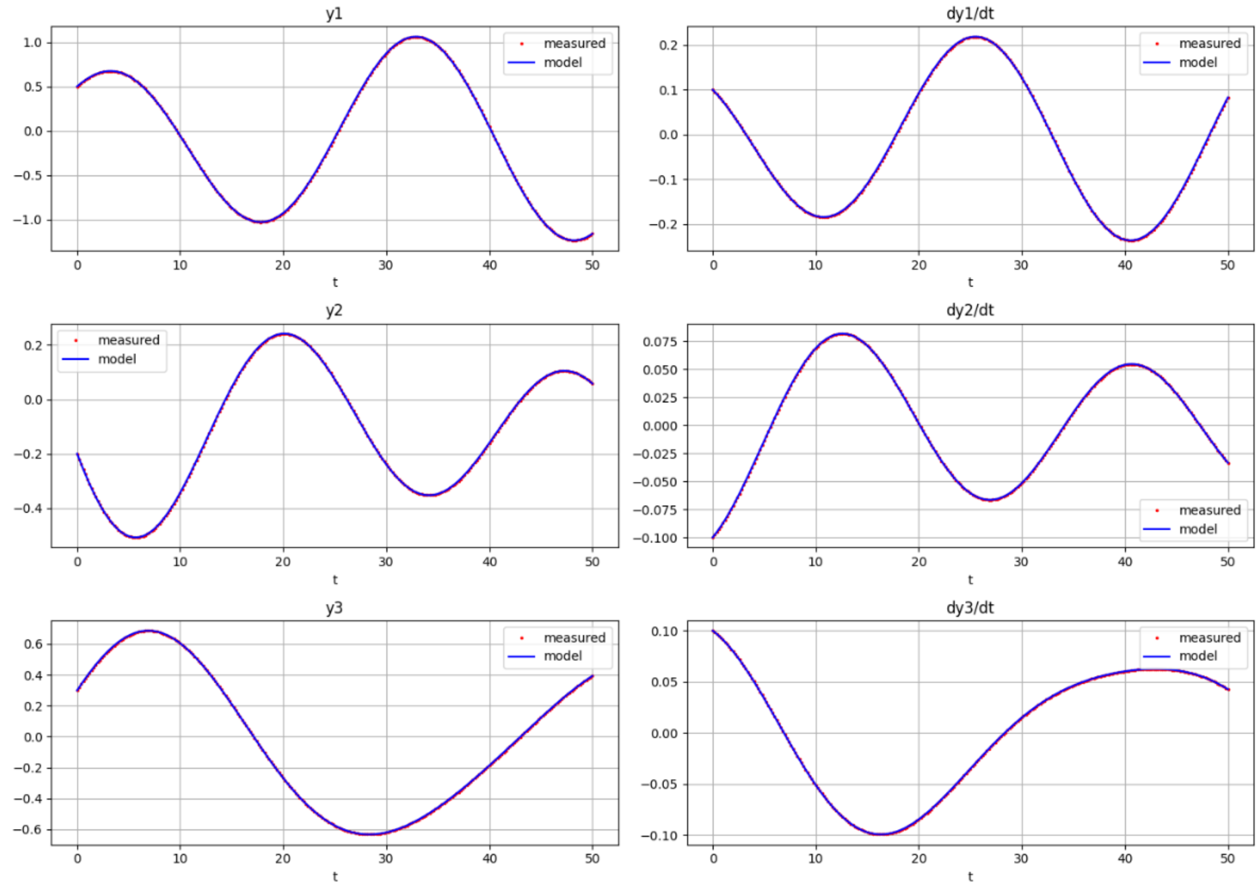
$$\beta_0 = (c_1, c_4, m_2) = (0.1,\ 0.08,\ 21)$$

було уточнене протягом лише **4 ітерацій**, після чого алгоритм досяг критеріїв зупинки. Остаточні оцінки параметрів становлять:

$$c_1 = 0.14, \qquad c_4 = 0.12, \qquad m_2 \approx 28.0.$$

Остаточне значення функції якості I(β) є *надзвичайно малим*, тобто модель практично ідеально відтворює дані.

Час роботи алгоритму — **0.08 с**, що підтверджує високу ефективність чисельної реалізації.



На графіках видно, що синя крива (модель) практично повністю збігається з червоними точками (виміряні значення) для всіх шести компонент:

- y_1, y_2, y_3 — зміщення трьох мас;
- $\dot{y}_1, \dot{y}_2, \dot{y}_3$ — їх швидкості.

Між виміряними та змодельованими значеннями **візуально не спостерігається помітних відхилень**, що свідчить про коректність знайдених параметрів та високу відповідність моделі експериментальним даним.