Київський національний університет імені Тараса Шевченка

Факультет комп`ютерних наук та кібернетики

Лабораторна робота №3(в. 9)

із курсу «Моделювання систем»

Виконала

студентка 3 курсу

групи ІПС-33

Постернак Таїсія

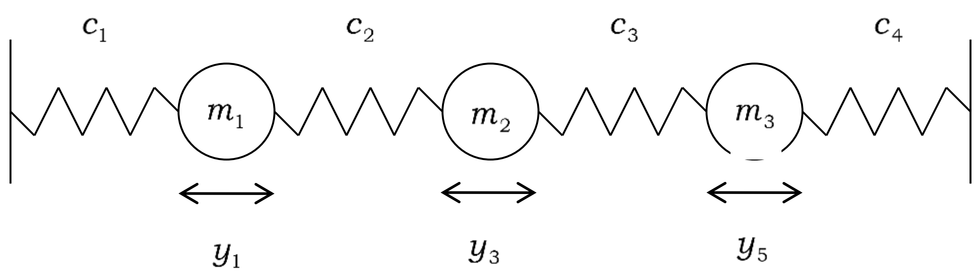
Київ, 2023

Тема.

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.



За умовою варіанта №9, вектор оцінюваних параметрів $\beta = (c_2, c_4, m_1)^T$, початкове наближення $\beta_0 = (0.2,\ 0.1,\ 9)^T$, відомі параметри $c_1 = 0.14, c_3 = 0.2$, $m_2 = 28, m_3 = 18$, ім’я файлу з спостережуваними даними y9.txt.

Спостереження стану моделі проведені на інтервалі часу $t_0 = 0,\ t_k = 50,\ \Delta t = 0.2$.

Теорія.

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

$$\frac{dU(t)}{dt} = \frac{\partial(Ay)}{\partial y^T} U(t) + \frac{\partial(Ay)}{\partial \beta^T},$$
$$U(t_0) = 0,\ \beta = \beta_0. \tag{1}$$

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

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

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

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

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

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

Хід роботи.

Щоб програмно обчислити формулу (1), скористаємося тим, що в даному випадку $\frac{\partial(Ay)}{\partial y^T} = A$. Також маємо, що

$$Ay = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -\frac{c_2 + c_1}{m_1} & 0 & \frac{c_2}{m_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \frac{c_2}{m_2} & 0 & -\frac{c_2 + c_3}{m_2} & 0 & \frac{c_3}{m_3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{c_3}{m_3} & 0 & -\frac{c_4 + c_3}{m_3} & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} y_2 \\ \frac{-c_1 y_1 - c_2 y_1 + c_2 y_3}{m_1} \\ y_4 \\ \frac{c_2 y_1 - c_2 y_3 - c_3 y_3 + c_3 y_5}{m_2} \\ y_6 \\ \frac{c_3 y_3 - c_3 y_5 - c_4 y_5}{m_3} \end{pmatrix}$$

Отже, для $\beta = (c_2, c_4, m_1)^T$

$$\frac{\partial(Ay)}{\partial \beta^T} = \begin{pmatrix} 0 & 0 & 0 \\ \frac{-y_1 + y_3}{m_1} & 0 & \frac{c_1 y_1 + c_2 y_1 - c_2 y_3}{{m_1}^2} \\ 0 & 0 & 0 \\ \frac{y_1 - y_3}{m_2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -\frac{y_5}{m_3} & 0 \end{pmatrix}$$

Зчитуємо дані спостереження та задаємо відомі змінні, а також початкові наближення решти. Виконуємо алгоритм доки не досягнемо бажаної точності. Для обчислення інтегралів будемо використовувати метод прямокутників.

Отримали наступні результати:

```
[1]: I = 2.7438
[2]: I = 0.020496
[3]: I = 0.00054962
[4]: I = 8.0834e-06
[5]: I = 9.6579e-08

Result:
  c2 = 0.3
  c4 = 0.12
  m1 = 12
```

Код програми (MATLAB).

```
clear
clc

h = 0.2;
t0 = 0;
tk = 50;
eps = 1e-6;

data = dlmread('y9.txt');
[size_m, size_n] = size(data);
c = [0.14, 0.2, 0.2, 0.1]';
m = [9, 28, 18]';


I = inf;
step = 0;
while I > eps
  step = step + 1;

  y = data(:, 1);
  deltaY = zeros(6, 1);
  U = zeros(6, 3);
  integral_UU = zeros(3, 3);
  integral_Uy = zeros(3, 1);
  I = 0.0;
  A = countA(m, c);

  for i = 2:size_n
    U_next = RungeKutta_U(A, U, y, m, c, h);
    y_next = RungeKutta_y(A, y, h);
    deltaY_next = data(:, i) - y_next;

    integral_UU = integral_UU + 0.5*h*(U'*U + U_next'*U_next);
    integral_Uy = integral_Uy + 0.5*h*(U'*deltaY + U_next'*deltaY_next);

    I = I + 0.5*h*(deltaY'*deltaY + deltaY_next'*deltaY_next);

    U = U_next;
    y = y_next;
    deltaY = deltaY_next;
  end

  delta = pinv(integral_UU)*integral_Uy;
  c(2) = c(2) + delta(1);
  c(4) = c(4) + delta(2);
  m(1) = m(1) + delta(3);

  disp(['[', num2str(step), ']: I = ', num2str(I)])
end

disp(' ')
disp('Result: ')
disp(['  c2 = ', num2str(c(2))])
```

```
disp(['  c4 = ', num2str(c(4))])
disp(['  m1 = ', num2str(m(1))])


function result = countA(m, c)
  A = zeros(6, 6);

  A(1, 2) = 1;

  A(2, 1) = -(c(1)+c(2))/m(1);
  A(2, 3) = c(2)/m(1);

  A(3, 4) = 1;

  A(4, 1) = c(2)/m(2);
  A(4, 3) = -(c(2)+c(3))/m(2);
  A(4, 5) = c(3)/m(2);

  A(5, 6) = 1;

  A(6, 3) = c(3)/m(3);
  A(6, 5) = -(c(3)+c(4))/m(3);

  result = A;
end

function result = RungeKutta_U (A, U, y, m, c, h)
  k1 = h*dU_dt(A, U, y, m, c);
  k2 = h*dU_dt(A, U + 0.5*k1, y, m, c);
  k3 = h*dU_dt(A, U + 0.5*k2, y, m, c);
  k4 = h*dU_dt(A, U + k3, y, m, c);

  result = U + (k1 + 2*k2 + 2*k3 + k4)/6;
end
function result = dU_dt (A, U, y, m, c)
  dAy_db = zeros(6, 3);
  dAy_db(2, 1) = (y(3)-y(1))/m(1);
  dAy_db(4, 1) = (y(1)-y(3))/m(2);
  dAy_db(6, 2) = -y(5)/m(3);
  dAy_db(2, 3) = (c(1)*y(1)+c(2)*y(1)-c(3)*y(3))/m(1)^2;

  result = A*U + dAy_db;
end

function result = RungeKutta_y (A, y, h)
  k1 = h*dy_dt(A, y);
  k2 = h*dy_dt(A, y + 0.5*k1);
  k3 = h*dy_dt(A, y + 0.5*k2);
  k4 = h*dy_dt(A, y + k3);

  result = y + (k1 + 2*k2 + 2*k3 + k4)/6;
end
function result = dy_dt (A, y)
  result = A*y;
end
```