Київський національний університет імені Т. Шевченка
Факультет комп'ютерних наук та кібернетики

Моделювання систем

Лабораторна робота №3

Параметрична ідентифікація параметрів з використанням функцій чутливості

Звіт

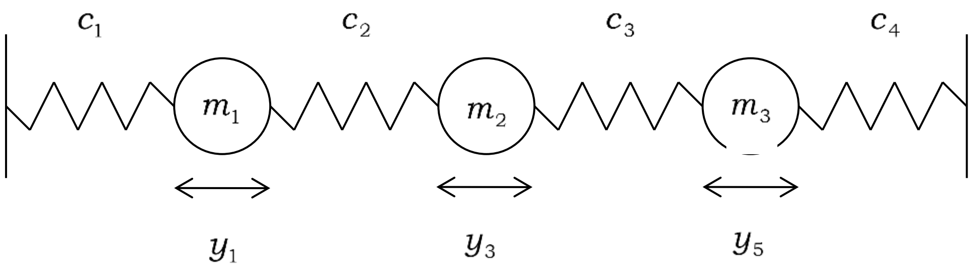
Виконала: студентка групи ІПС-31
Карпишин Ольга

Київ-2019

Умови лабораторної роботи:

Параметрична ідентифікація параметрів з використанням функцій чутливості.

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.



Мій варіант – f4.txt

4) Вектор оцінюваних параметрів $\beta = (m_2, c_3, m_3)^T$, початкове наближення $\beta_0 = (21, 0.15, 11)^T$, відомі параметри $c_1 = 0.14, c_2 = 0.3,\ c_4 = 0.12,\ m_1 = 12$, ім'я файлу з спостережуваними даними y4.txt.

Хід роботи

I. Визначимо матрицю A:

```
function res = defineA(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);
    res = A;
end
```

II. Обчислимо матрицю чутливості U та y за допомогою методу Рунге-Кутта:

```
function res = U_RungeKutt(A, U, h, y, m, c)
    k1 = h * fU(A, U, y, m, c);
    k2 = h * fU(A, U + k1 / 2.0, y, m, c);
    k3 = h * fU(A, U + k2 / 2.0, y, m, c);
    k4 = h * fU(A, U + k3, y, m, c);
    res = U + (k1 + 2 * k2 + 2 * k3 + k4) / 6.0;
end
```

Де $f(U,t) = \frac{d(Ay)}{dy^T} * U(t) + \frac{d(Ay)}{d\beta^T}$ $\qquad$ $f(U,t) = A * U + dAy$

$$dAy = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-c_2 y_1 + (c_2+c_3) y_3 - c_3 y_5}{m_2^2} & \frac{y_5 - y_3}{m_2} & 0 \\ 0 & 0 & 0 \\ 0 & \frac{y_3 - y_5}{m_3} & \frac{-c_3 y_3 + (c_3 + c_4) y_5}{m_3^2} \end{pmatrix}$$

```
function res = fU(A, U, y, m, c)
    dAy = zeros(6,3);
    dAy(4, 1) = (-c(2)*y(1)+(c(2)+c(3))*y(3)-c(3)*y(5))/(m(2)*m(2));
    dAy(4, 2) = (-y(3)+y(5))/m(2);
    dAy(6, 2) = (-y(5)+y(3))/m(3);
    dAy(6, 3) = (-c(3)*y(3)+(c(3)+c(4))*y(5))/(m(3)*m(3));
    res = A * U + dAy;
end

function res = Y_RungeKutt(A, y, h, m, c)
    k1 = h * fy(A, y);
    k2 = h * fy(A, y + k1 / 2.0);
    k3 = h * fy(A, y + k2 / 2.0);
    k4 = h * fy(A, y + k3);
    res = y + (k1 + 2 * k2 + 2 * k3 + k4) / 6.0;
end
```

III. Обчислюємо $\Delta\beta, I(\beta)$, шукаючи інтеграли методом прямокутників.

```
while (I > eps)
    iterationNumber = iterationNumber+1;
    y = explored_y(:,1);
    dy = zeros(M,1);
    U = zeros(6,3);
    B = zeros(3,3);
    b = zeros(3,1);
    I = 0.0;
    i = 2;

    for iter = 1:M
        resulted_y(iter,1) = y(iter,1);
    end

    while(i <= N)
        A = defineA (m,c);
```

```
        Unew = U_RungeKutt(A, U, h, y, m, c);
        yNew = Y_RungeKutt(A, y, h, m, c);
        dyNew = explored_y(:,i) - yNew;
        for iter = 1:M
            resulted_y(iter,i)=y(iter,1);
        end

        B = B + h*(U'*U + Unew'*Unew) / 2.0;
        b = b + h*(U'*dy + Unew'*dyNew) / 2.0;
        I = I + h*(dy'*dy + dyNew'*dyNew) / 2.0;
        U = Unew;
        y = yNew;
        dy = dyNew;
        i = i + 1;
    end
    delta = pinv(B)*b;
    m(2) = m(2) + delta(1);
    c(3) = c(3) + delta(2);
    m(3) = m(3) + delta(3);
    disp(I);
end
```

VI. Отримаємо результати:

m_2=0.12

c_3=12

m_3=18

Кількість ітерація: 5

1) 6.4993
2) 0.7874
3) 0.0059
4) 2.0051e-06
5) 1.1276e-08