Моделювання систем

Лабораторна робота №3

«Параметрична ідентифікація параметрів з використанням функцій чутливості»

**Варіант 13**

Виконав:

студент групи ІПС-32

Тоцький Олександр

Київ-2020

***Варіант експериментальних даних:*** Вектор оцінюваних параметрів , початкове наближення , відомі параметри , , ім’я файлу з спостережуваними даними y3.txt.

1. Початкові значення

**yStart = dlmread('y3.txt',' ');**

**[M,N] = size(yStart);**

**c(1) = 0.1; % estimate**

**c(2) = 0.3;**

**c(3) = 0.1; % estimate**

**c(4) = 0.12;**

**m(1) = 9; % estimate**

**m(2) = 28;**

**m(3) = 18;**

**h = 0.2;**

**epsilon = 1e-6;**

**I = inf;**

**numberOfIterations = 0;**

1. Математична модель коливання трьох мас:



**function result = getA(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(4) + c(3)) / m(3);**

**result = A;**

**end**

1. Знаходження матриці чутливості U методом Рунге-Кутта 4-го порядку:

В даному випадку .



Тепер можемо застосовувати метод Рунге-Кутта:

**function result = 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);**

**result = U + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6.0;**

**end**

**function result = fU(A, U, y, m, c)**

**dAy = zeros(6,3);**

**dAy(2, 1) = -(y(1) / m(1));**

**dAy(2, 3) = ((c(1)+c(2))\*y(1) - c(2)\*y(3))/ (m(1) \* m(1));**

**dAy(4, 2) = (-y(3)+y(5)) / m(2);**

**dAy(6, 2) = (y(3)-y(5))/ m(3);**

**result = A \* U + dAy;**

**end**

Використовуючи метод Рунге-Кутта знаходимо y:

**function result = Y\_RungeKutt(A, y, h)**

**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);**

**result = y + (k1 + 2 \* k2 + 2 \* k3 + k4) / 6.0;**

**end**

**function result = fy(A, y)**

**result = A\*y;**

**end**

1. Знаходимо I – квадратичне відхилення від спостережуваних даних і розв’язку системи Ay . Якщо воно мінімальне – ми знайшли вектор параметрів В, який найкраще наближує розв’язок системи диференціальних рівнянь відносно наших параметрів.



Рахуємо , де інтеграли знаходимо методом прямокутників (в кожний момент часу рахуємо підінтегральну функцію і додаємо її до попередньої), та перераховуємо : , де  – початкове наближення вектора параметрів.

.

**while (I > epsilon)**

**numberOfIterations = numberOfIterations+1;**

**y = yStart(:,1);**

**dy = zeros(M,1);**

**U = zeros(6,3);**

**firstPart = zeros(3,3);**

**secondPart = zeros(3,1);**

**I = 0.0;**

**i = 2;**

**while(i <= N)**

**A = getA(m,c);**

**Unew = U\_RungeKutt(A, U, h, y, m, c);**

**yNew = Y\_RungeKutt(A, y, h);**

**dyNew = yStart(:,i) - yNew;**

**firstPart = firstPart + h\*(U'\*U + Unew'\*Unew) / 2.0;**

**secondPart = secondPart + 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**

**deltaBeta = inv(firstPart)\*secondPart;**

**c(1) = c(1) + deltaBeta(1);**

**c(3) = c(3) + deltaBeta(2);**

**m(1) = m(1) + deltaBeta(3);**

**disp(I);**

**end**

**Результати:**

* **c1: 0.1400**
* **c3: 0.2000**
* **m1: 12.0000**
* **Number of iterations: 5**