**Київський національний університет імені Тараса Шевченка**

**Факультет комп’ютерних наук та кібернетики**

**Моделювання систем**

**Лабораторна робота №3**

**Варіант 14**

**Виконав:**

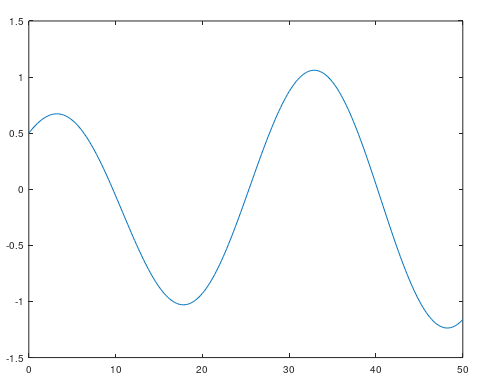
студент групи ІПС-31

Точаненко Владислав Володимирович

**Київ-2019**

Вектор оцінюваних параметрів , початкове наближення , відомі параметри , , ім’я файлу з спостережуваними даними y4.txt.

**1. Побудуємо графік функції спостереження координат моделі:**



**2. Визначимо матрицю А:**

function res = 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(3) + c(4)) / m(3);

    res = A;

end

**3. Обчислимо матрицю U 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;

endfunction

Де

function res = fU(A, U, y, m, c)

    dAy = zeros(6,3);

    dAy(6, 1) = -(y(5) / m(3));

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

    dAy(6, 3) = -(c(3) \* y(3) - (c(3) + c(4)) \* y(5)) / (m(3) \* m(3));

    res = A \* U + dAy;

end

**4. Обчислюємо y за допомогою методу Рунге-Кутта:**

function res = Y\_RungeKutt(A, y, h, m, c)

    k1 = h \* fy(A, y, m, c);

    k2 = h \* fy(A, y + k1 / 2.0, m, c);

    k3 = h \* fy(A, y + k2 / 2.0, m, c);

    k4 = h \* fy(A, y + k3, m, c);

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

endfunction

**5. Обчислюємо , шукаючи інтеграли методом прямокутників.**

while (I > eps)

    numIter = numIter+1;

    y = yExperimental(:,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

        yGained(iter,1) = y(iter,1);

    end

    while(i <= N)

        A = getA(m,c);

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

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

        dyNew = yExperimental(:,i) - yNew;

        for iter = 1:M

            yGained(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;

    c(4) = c(4) + delta(1);

    m(1) = m(1) + delta(2);

    m(3) = m(3) + delta(3);

end

**6. Отримаємо результати:**

c4 = 0.12  
m1 = 12.0  
m3 = 18.0  
Iterations = 8

Iteration #1 = 8.9860  
Iteration #2 = 105.03  
Iteration #3 = 11.904  
Iteration #4 = 1.8651  
Iteration #5 = 0.21910  
Iteration #6 = 0.0096417  
Iteration #7 = 0.000038239  
Iteration #8 = 0.000000011974

