МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ   
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ**

**ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА»**

**(БГТУ им. В.Г. Шухова)**

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**Лабораторная работа №5**

по дисциплине: Системное моделирование

тема: Оценка вероятностных характеристик фазовых координат систем

Выполнил: ст. группы ПВ-223

Игнатьев Артур Олегович

Проверил:

Полунин Александр Иванович

Белгород 2024 г.

**Вариант 3**

Решение задачи:

1. Вычислим значения и :
2. Зададим системы случайных величин:

|  |  |  |
| --- | --- | --- |
|  | V1 | V2 |
| 1 | 2,83 | 0 |
| 2 | 0 | 2,83 |
| 3 | 2,83 | 2,83 |
| 4 | -2,83 | -2,83 |

1. Зададим четыре набора входных данных:

Далее интегрируем четыре раза систему дифференциальных уравнений, каждый раз используя новый набор начальных условий. В результате получим четыре набора фазовых координат за момент времени tk. Вычисляем значения математического ожидания и дисперсии фазовых координат.

Код программы:

clc; close all; clear;

% Основная функция для решения задачи

% Параметры:

% m - масса

% M - начальные значения вектора [M1, M2]

% D - вектор коэффициентов [D1, D2]

% t - время интегрирования

function ret = Solve(m, M, D, t)

% Вычисление начальных векторов на основе параметров D и m

E = [D(1) \* sqrt(m); D(2) \* sqrt(m)];

% Формирование матрицы начальных условий V

V = [M(1) + E(1), M(2);

M(1), M(2) + E(2);

M(1) + E(1), M(2) + E(2);

M(1) - E(1), M(2) - E(2)];

% Интегрирование систем дифференциальных уравнений для каждой строки матрицы V

Mr = [Integrate(V(1, :), t);

Integrate(V(2, :), t);

Integrate(V(3, :), t);

Integrate(V(4, :), t)];

% Вычисление значений Mk и Dk на основе результатов интегрирования

ret = [[GetMk(Mr(:, 1)', m);

GetMk(Mr(:, 2)', m)];

[GetDk(Mr(:, 1)', m);

GetDk(Mr(:, 2)', m)]];

end

% Функция для вычисления Dk

% Параметры:

% X - вектор значений после интегрирования

% m - масса

function ret = GetDk(X, m)

ret = 1/m \* (X(1)^2 + X(2)^2) + (X(4)^2 - X(3)^2) / (2 \* m) - GetMk(X, m)^2;

end

% Функция для вычисления Mk

% Параметры:

% X - вектор значений после интегрирования

% m - масса

function ret = GetMk(X, m)

ret = 1/m \* (X(1) + X(2)) + (X(4) - X(3)) / (2 \* m);

end

% Функция для численного интегрирования системы дифференциальных уравнений

% Параметры:

% X - начальные условия

% t - время интегрирования

function ret = Integrate(X, t)

[~, Xr] = ode45(@(t, X) GetSystem(X, t), [0 t], X);

ret = Xr;

end

% Определение системы дифференциальных уравнений

% Параметры:

% X - текущие значения

% t - текущее время

function ret = GetSystem(X, t)

ret = [FirstEquation(X, t); SecondEquation(X, t)];

end

% Первая функция дифференциального уравнения

% Параметры:

% X - текущие значения

function ret = FirstEquation(X, ~)

ret = sin(X(1) + X(2));

end

% Вторая функция дифференциального уравнения

% Параметры:

% X - текущие значения

% t - текущее время

function ret = SecondEquation(X, t)

ret = X(1) \* t + 2 \* X(2);

end

% Вызов основной функции с заданными параметрами и вывод результатов

% Параметры:

% 2 - масса m

% [0.1, 0.3] - начальные значения M1 и M2

% [2, 2] - коэффициенты D1 и D2

% 35 - время интегрирования t

res = Solve(2, [0.1 0.3], [2 2], 35);

% Вывод значений Mk

for i = 1 : 2

str\_m = strcat('m', num2str(i), 'k');

disp(str\_m);

disp(res(i));

end

% Вывод значений Dk

for i = 1 : 2

str\_d = strcat('d', num2str(i), 'k');

disp(str\_d);

disp(res(i + 2));

end

Результат работы программы:

