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

Дмитриев Андрей

Проверил:

Полунин А.И.

Белгород, 2024 г.

**Цель работы:**

1. Изучить метод Доступова для оценки вероятностных характеристик фазовых координат систем.
2. Разработать программу для оценки вероятностных характеристик вектора Х на момент времени tk (конкретный вариант).

**Вариант 2:**

1. Вычислим и :

, .

1. Зададим системы случайных величин:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 1 |  | 0 |
| 2 | 0 |  |
| 3 |  |  |
| 4 |  |  |

1. Зададим четыре набора начальных условий:

1) ; 3) ;

2) ; 4) .

1. Создадим программу для интегрирования и расчёта мат. ожидания и дисперсии.

import numpy as np  
from scipy.integrate import odeint  
  
  
def Solve(m, M, D, t):  
 E = [D[0] \* np.sqrt(m),  
 D[1] \* np.sqrt(m)]  
 V = [[M[0] + E[0], M[1]],  
 [M[0], M[1] + E[1]],  
 [M[0] + E[0], M[1] + E[1]],  
 [M[0] - E[0], M[1] - E[1]]]  
 Mr = [integrate(V[0], t),  
 integrate(V[1], t),  
 integrate(V[2], t),  
 integrate(V[3], t)]  
 ret = [[GetMk(col(Mr, 0), m),  
 GetMk(col(Mr, 1), m)],  
 [GetDk(col(Mr, 0), m),  
 GetDk(col(Mr, 1), m)]]  
 return ret  
  
  
def col(arr, i):  
 return [x[i] for x in arr]  
  
  
def GetMk(X, m):  
 return 1 / m \* (X[0] + X[1]) + (X[3] - X[2]) / (2 \* m)  
  
  
def GetDk(X, m):  
 return 1 / m \* (X[0] \*\* 2 + X[1] \*\* 2) + (X[3] \*\* 2 - X[2] \*\* 2) / (2 \* m) - GetMk(X, m) \*\* 2  
  
  
def integrate(X, t):  
 [\_, Xr] = odeint(GetSystem, X, [0, t])  
 return Xr  
  
  
def GetSystem(X, t):  
 return [FirstEquation(X, t), SecondEquation(X, t)]  
  
  
def FirstEquation(X, t):  
 return X[1] + X[0] \*\* 2 - t  
  
  
def SecondEquation(X, t):  
 return cos(X[0] \* X[1])  
  
  
res = Solve(2, [0.5, 0.], [1, 1], 28)  
for i in range(2):  
 str\_m = 'm' + str(i + 1) + 'k'  
 print(str\_m)  
 print(res[0][i])  
  
for i in range(2):  
 str\_d = 'd' + str(i + 1) + 'k'  
 print(str\_d)  
 print(res[1][i])

1. Получим результат:

m1k

1.7599e+00

m2k

5.2601e-01

d1k

4.1710e-01

d2k

9.2190e-04

**Вывод:** В ходе лабораторной работы научились составлять математическую модель, на основе которой написали программу для построения графиков функции, описывающих поведение динамической системы.