**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

автоматизированныхсистем

Лабораторная работа №5

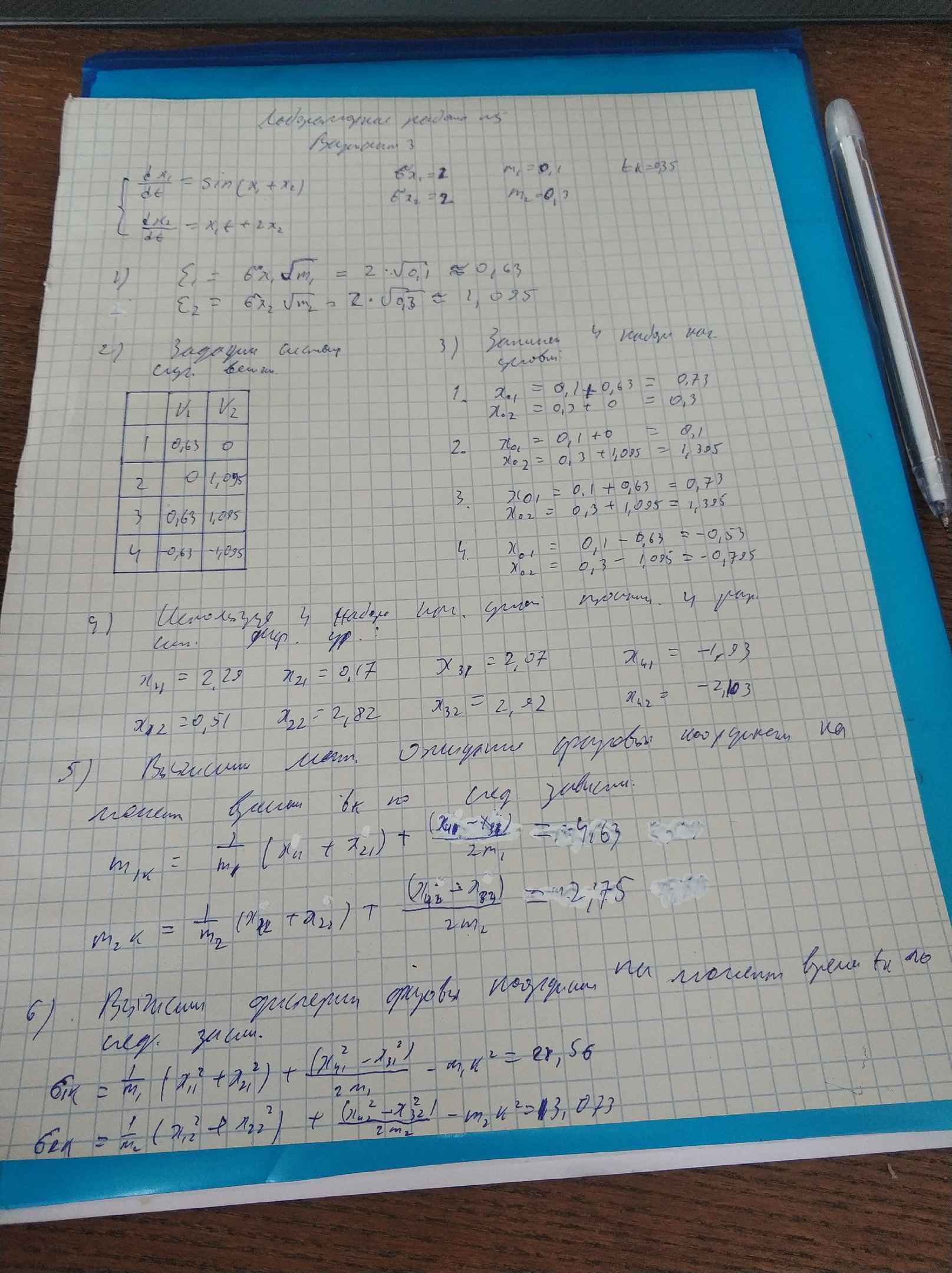
дисциплина: Системное моделирование

тема: Оценка вероятностных характеристик фазовых координат систем

Выполнил ст. группы ВТ-22

Воскобойников Илья

Белгород 2020



Текст программы

import matplotlib.pyplot as plt  
from math import \*  
  
import numpy as np  
from scipy.integrate import ode  
  
ts = []  
ys = []  
gamm = []  
  
def get\_steps(t, y):  
 ts.append(t)  
 ys.append(list(y.copy()))  
  
def f(t, y, \_):  
 x1, x2 = y  
 return [sin(x1 + x2) , x1\*t+2\*x2]  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 tmax = 0.35  
  
 om1 = 2  
 om2 = 2  
 m1 = 0.1  
 m2 = 0.3  
  
 E1 = sqrt(4.5)  
 E2 = sqrt(3.2)  
  
 x011 = m1 + E1  
 x012 = m2 + 0  
 x01 = [x011, x012]  
  
 x021 = m1 + 0  
 x022 = m2 + E2  
 x02 = [x021, x022]  
  
 x031 = m1 + E1  
 x032 = m2 + E2  
 x03 = [x031, x032]  
  
 x041 = m1 - E1  
 x042 = m2 - E2  
 x04 = [x041, x042]  
  
 x\_ans = []  
  
 for index, complect in enumerate([x01, x02, x03, x04]):  
 y0, t0 = complect, 0.2  
 ODE = ode(f)  
 ODE.set\_integrator(**'dopri5'**, max\_step=0.1, nsteps=100)  
 ODE.set\_solout(get\_steps)  
 fig, ax = plt.subplots()  
 fig.set\_facecolor(**'white'**)  
 ODE.set\_initial\_value(y0, t0)  
 ODE.set\_f\_params(0)  
 ODE.integrate(tmax)  
 T = np.array(ts)  
 Y = np.array(ys)  
 G = np.array(gamm)  
 plt.plot(T[:], Y[:, 1], linewidth=3, label=**'f%i'** % 2)  
 plt.plot(T[:], Y[:, 0], linewidth=3, label=**'f%i'** % 1)  
 plt.grid(True)  
 plt.legend(loc=**'best'**)  
 plt.show()  
 x1 = ys[-1][0]  
 x2 = ys[-1][1]  
 x\_ans.append([x1, x2])  
 ys = []  
 ts = []  
  
 print(**f"x**{index}**1 =** {x1}**"**)  
 print(**f"x**{index}**2 =** {x2}\n\n**"**)  
  
 m1k = (1/m1)\*(x\_ans[0][0]+x\_ans[1][0]) + (x\_ans[3][0]-x\_ans[2][0])/(2\*m1)  
 m2k = (1/m2)\*(x\_ans[0][1]+x\_ans[1][1]) + (x\_ans[3][1]-x\_ans[2][1])/(2\*m2)  
  
 print(**"m1k = "** + str(m1k))  
 print(**"m2k = "** + str(m2k))  
  
 print()  
 om1k = (1/m1)\*(x\_ans[0][0]\*\*2 + x\_ans[1][0]\*\*2) + (x\_ans[3][0]\*\*2 - x\_ans[2][0]\*\*2)/(2\*m1) - m1k\*m1k  
 om2k = (1/m2)\*(x\_ans[0][1]\*\*2 + x\_ans[1][1]\*\*2) + (x\_ans[3][1]\*\*2 - x\_ans[2][1]\*\*2)/(2\*m2) - m2k\*m2k  
  
 print(**"om1k = "** + str(om1k))  
 print(**"om2k = "** + str(om2k))

