

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**  
**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**  
**«ДИНАМИКА СИСТЕМЫ»**  
**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И**  
**ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**  
**ВАРИАНТ ЗАДАНИЯ № 13**

Выполнил(а) студент группы М8О-208Б-22

Былькова Кристина Алексеевна \_\_\_\_\_  
подпись, дата

Проверил и принял

Зав. каф. 802, Бардин Б.С. \_\_\_\_\_  
подпись, дата

с оценкой \_\_\_\_\_  
подпись, дата

Москва, 2023

Задание: проинтегрировать систему дифференциальных уравнений движения системы с двумя степенями свободы с помощью средств Python. Построить графики законов движения системы и указанных в задании реакций для разных случаев системы.

Текст программы:

```
import matplotlib.pyplot as p
from matplotlib.animation import FuncAnimation
import numpy as n

from scipy.integrate import odeint

def SystDiffEq(y, t, m1, m2, a, b, g):
    # y[0,1,2,3] = phi, psi, phi', psi'
    # dy[0,1,2,3] = phi', psi', phi'', psi''

    dy = n.zeros_like(y)
    dy[0] = y[2] # тривиальные уравнения вида

    dy[1] = y[3] # dphi = phi', dpsi = psi'

    # представим систему уравнений движения в виде
    # линейной относительно вторых производных
    # системы  $A \cdot q'' = B$ , где
    #  $q = (\phi; \psi)$ ,  $A = A(\phi, \psi)$ ,
    #  $B = B(\phi, \psi, \phi', \psi')$ :
    #
    #  $a_{11} \cdot \phi'' + a_{12} \cdot \psi'' = b_1$ 
    #  $a_{21} \cdot \phi'' + a_{22} \cdot \psi'' = b_2$ 
    # коэффициенты первого уравнения

    a11 = (m1+m2)*2*a
    a12 = -m2*b*(1-n.cos(y[1]-y[0]))
    b1 = -((m1+m2)*g*n.sin(y[0])-m2*b*(n.sin(y[1]-
y[0])*y[3]**2))

    # коэффициенты второго уравнения
    a21 = a*(1-n.cos(y[1]-y[0]))
    a22 = -2*b
    b2 = a*n.sin(y[1]-y[0])*y[2]**2+g*n.sin(y[1])

    # решение правилом Крамера
    dy[2] = (b1 * a22 - b2 * a12) / (a11 * a22 - a12 * a21)
    dy[3] = (a11 * b2 - a21 * b1) / (a11 * a22 - a12 * a21)
    return dy

# Определяем величины, заданные для системы
m1 = 4
m2 = 2
r1 = 1
```

```

r2 = 0.125
r3 = 0.05
g=9.81
a=r1-r3
b=r1-r2

# Определяем сетку времени
T = n.linspace(0, 10, 100)
y0 = [n.pi/6, n.pi/3, 0, n.pi/3]

# Функция для численного интегрирования уравнений движения
Y = odeint(SystDiffEq, y0, T, (m1,m2,a,b,g))

Phi = Y[:,0]
Psi = Y[:,1]
Phit = Y[:,2]
Psit = Y[:,3]

fgrp = p.figure()
plPhi = fgrp.add_subplot(4,1,1)
plPhi.plot(T, Phi)

plPsi = fgrp.add_subplot(4,1,2)
plPsi.plot(T, Psi)

Phitt = n.zeros_like(T)
Psitt = n.zeros_like(T)
# Вычисляем вторые производные координат по времени
for i in range(len(T)):
    Phitt[i] = SystDiffEq(Y[i], T[i], m1,m2,a,b,g)[2]
    Psitt[i] = SystDiffEq(Y[i], T[i], m1,m2,a,b,g)[3]

# Строим графики функций
N2 = m2*(g*n.cos(Psi)+b*Psit**2+a*(Phit**2*n.cos(Psi-Phi)-
Phitt*n.sin(Psi-Phi)))
Ftr = m2*(g*n.sin(Psi)+b*Psitt+a*(Phit**2*n.sin(Psi-
Phi)+Phitt*n.cos(Psi-Phi)))
plRA = fgrp.add_subplot(4,1,3)
plRA.plot(T, N2)
plNK = fgrp.add_subplot(4,1,4)
plNK.plot(T, Ftr)
fgrp.show()

fgr = p.figure()
plt = fgr.add_subplot(1,1,1)
plt.axis('equal')

plt.plot([- (2*r1), (2*r1)], [0,0], '--')
plt.plot([0,0], [- (2*r1), (2*r1)], '--')

Alp = n.linspace(0, 2*n.pi, 100)
Xsh = r3 * n.sin(Alp)

```

```

Ysh = r3 * n.cos(Alp)
Shtift = plt.plot(Xsh,Ysh)[0]

Xc1 = a * n.sin(Phi[0])
Yc1 = a * -n.cos(Phi[0])

Xa1 = r1 * n.sin(Alp)
Ya1 = r1 * n.cos(Alp)
A1 = plt.plot(Xa1+Xc1,Ya1+Yc1)[0]

Xc2 = Xc1+b * n.sin(Psi[0])
Yc2 = Yc1+b * -n.cos(Psi[0])

Xa2 = r2 * n.sin(Alp)
Ya2 = r2 * n.cos(Alp)
A2 = plt.plot(Xa2+Xc2,Ya2+Yc2)[0]

def run(i):
    Xc1 = a * n.sin(Phi[i])
    Yc1 = a * -n.cos(Phi[i])
    Xc2 = Xc1+b * n.sin(Psi[i])
    Yc2 = Yc1+b * -n.cos(Psi[i])
    A1.set_data(Xa1+Xc1,Ya1+Yc1)
    A2.set_data(Xa2+Xc2,Ya2+Yc2)
    return

anim = FuncAnimation(fgr, run, frames = len(T), interval = 1)
fgr.show()

```

Результат работы программы:

