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 № 31

Выполнил(а) студент группы М8О-203Б-22

Теребаев К.Д._____
подпись, дата

Проверил и принял

Авдюшкин А.Н._____
подпись, дата

с оценкой _____

Москва, 2023

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

Законы движения системы:

$$2(m_1+m_2)R\ddot{\varphi}+m_2(R-r)\left[(1+\cos\psi)\ddot{\psi}-\dot{\psi}^2\sin\psi\right]+cR\varphi=$$

$$=F_0\sin\gamma t,\qquad R(1+\cos\psi)\ddot{\varphi}+2(R-r)\ddot{\psi}+g\sin\psi=0.$$

Проекции реакции оси блока:

$$F_A=(m_1+m_2)R\ddot{\varphi}+m_2(R-r)\left(\ddot{\psi}\cos\psi-\dot{\psi}^2\sin\psi\right)+cR\varphi-F,$$

$$N_A=(m_1+m_2)\,g\;+\;m_2(R-r)\left(\ddot{\psi}\sin\psi+\dot{\psi}^2\cos\psi\right).$$

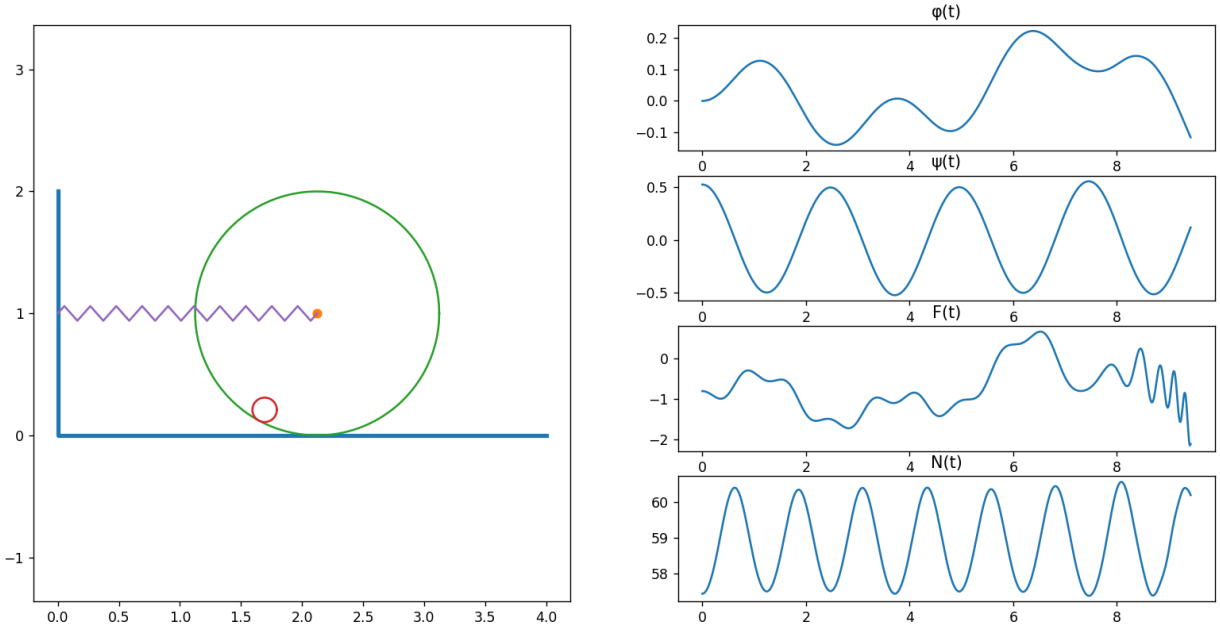
Текст программы:

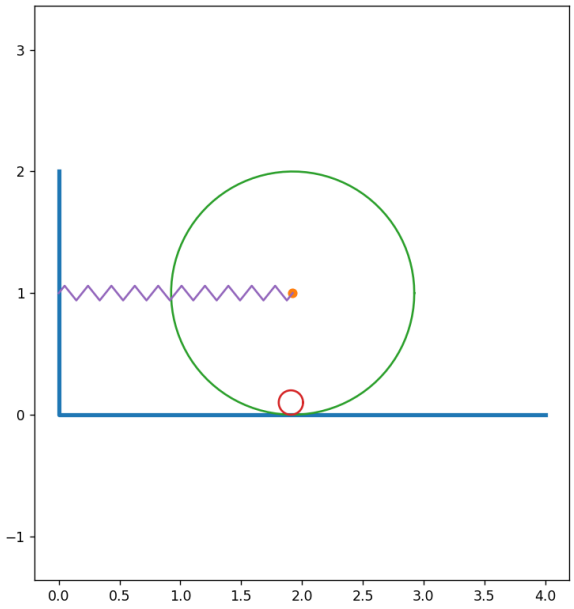
```
import numpy as n
import matplotlib.pyplot as plt
from scipy.integrate import odeint
from matplotlib.animation import FuncAnimation


def sys_diff_eq(y, t, m1, m2, R, r, c, F, gamma, g):
    # y = [phi, psi, phi', psi', t] -> dy = [phi', psi', phi'', psi'', t]
    dy = n.zeros_like(y)
    dy[0] = y[2]
    dy[1] = y[3]
    dy[4] = y[4] + 3 * n.pi / 1000

    # a11 * phi'' + a12 * psi'' = b1
    # a21 * phi'' + a22 * psi'' = b2

    a11 = 2 * (m1 + m2) * R
    a12 = m2 * (R - r) * (1 + n.cos(y[1]))
    b1 = F * n.sin(gamma * y[4]) + m2 * (R - r) * y[3] ** 2 * n.sin(y[1]) - c * R *
y[0]

    a21 = R * (1 + n.cos(y[1]))
    a22 = 2 * (R - r)
    b2 = -g * n.sin(y[1])

    det_a = a11 * a22 - a12 * a21
    det_a1 = b1 * a22 - a12 * b2
    det_a2 = a11 * b2 - a21 * b1

    dy[2] = det_a1 / det_a
    dy[3] = det_a2 / det_a
```

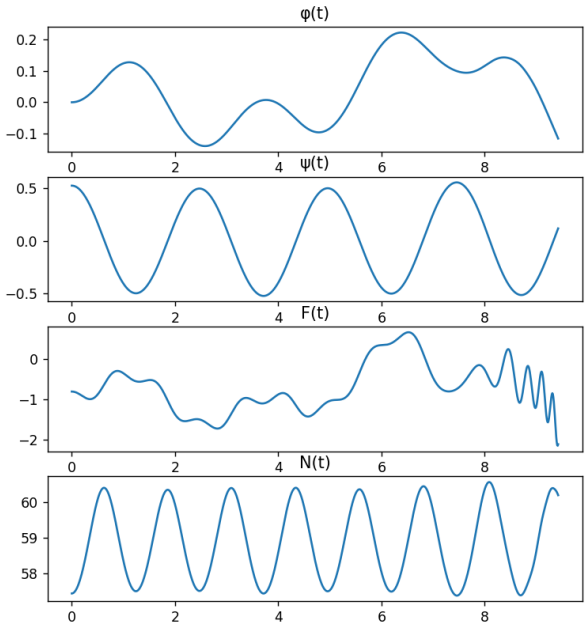
```
    return dy


m1 = 5
m2 = 1
R = 1
r = 0.1
c = 10
F = 1
gamma = n.pi / c
g = 9.81

y0 = [0, n.pi / 6, 0, 0, 0]

x0 = 2
L = 0.9

step = 1000
t = n.linspace(0, 3 * n.pi, step)

Y = odeint(sys_diff_eq, y0, t, (m1, m2, R, r, c, F, gamma, g))

phi = Y[:, 0]
psi = Y[:, 1]
phi_t = Y[:, 2]
psi_t = Y[:, 3]

x = n.zeros_like(t)
phi_tt = n.zeros_like(t)
psi_tt = n.zeros_like(t)
F_a = n.zeros_like(t)
N_a = n.zeros_like(t)

for i in range(len(t)):
    x[i] = phi[i] * R
    phi_tt[i] = sys_diff_eq(Y[i], t[i], m1, m2, R, r, c, F, gamma, g)[2]
    psi_tt[i] = sys_diff_eq(Y[i], t[i], m1, m2, R, r, c, F, gamma, g)[3]
    F_a[i] = (m1 + m2) * R * phi_tt[i] + m2 * (R - r) * (psi_tt[i] * n.cos(psi[i])) \
        - psi_t[i]**2 * n.sin(psi[i]) + c * R * phi[i] - F
    N_a[i] = (m1 + m2) * g + m2 * (R - r) * (psi_tt[i] * n.sin(psi[i]) + psi_t[i]**2
* n.cos(psi[i]))

fgr = plt.figure()

gr = fgr.add_subplot(4, 2, (1, 7))
gr.axis('equal')

phi_plt = fgr.add_subplot(4, 2, 2)
phi_plt.plot(t, phi)
phi_plt.set_title("ϕ(t)")
psi_plt = fgr.add_subplot(4, 2, 4)
psi_plt.plot(t, psi)
psi_plt.set_title("ψ(t)")
F_a_plt = fgr.add_subplot(4, 2, 6)
F_a_plt.plot(t, F_a)
F_a_plt.set_title("F(t)")
N_a_plt = fgr.add_subplot(4, 2, 8)
N_a_plt.plot(t, N_a)
N_a_plt.set_title("N(t)")

gr.plot([0, 0, 4], [2, 0, 0], linewidth=3)
```

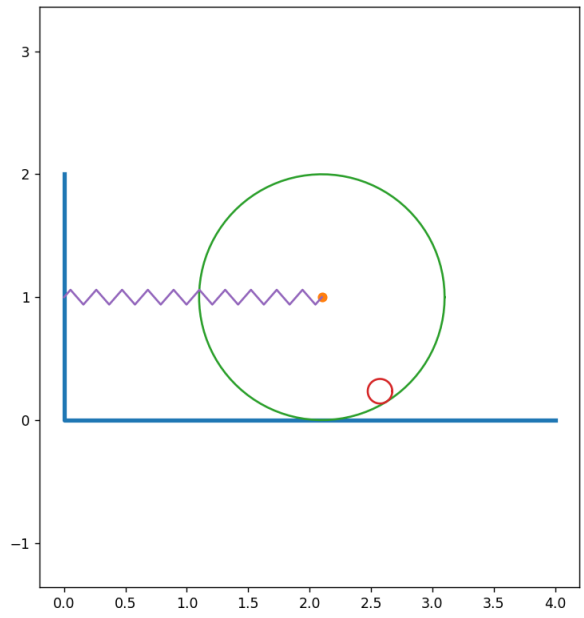
```
Xa = x0 + x
Ya = R

Xb = Xa + L * n.sin(psi)
Yb = Ya - L * n.cos(psi)

pA = gr.plot(Xa[0], Ya, marker='o')[0]

Alp = n.linspace(0, 2 * n.pi, 100)
Xc = n.cos(Alp)
Yc = n.sin(Alp)

Main_cylinder = gr.plot(Xc * R + Xa[0], Yc * R + Ya)[0]
Sub_cylinder = gr.plot(Xc * r + Xb[0], Yc * r + Yb[0])[0]

Np = 20
Xp = n.linspace(0, 1, 2 * Np + 1)
Yp = 0.06 * n.sin(n.pi / 2 * n.arange(2 * Np + 1))

Spring = gr.plot((x0 + x[0]) * Xp, Yp + R)[0]


def run(i):
    pA.set_data([Xa[i]], [Ya])
    Main_cylinder.set_data([Xc * R + Xa[i]], [Yc * R + Ya])
    Sub_cylinder.set_data([Xc * r + Xb[i]], [Yc * r + Yb[i]])
    Spring.set_data([(x0 + x[i]) * Xp], [Yp + R])
    return [pA, Main_cylinder, Sub_cylinder, Spring]


anim = FuncAnimation(fgr, run, frames=step, interval=1)

plt.show()
```

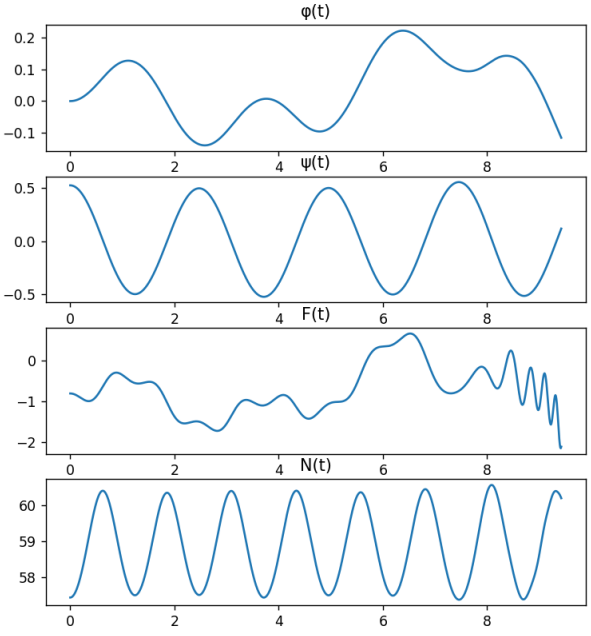
Результат работы:




3
2
1
0
−1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


φ(t)
0.2
0.1
0.0
−0.1
ψ(t)
0.5
0.0
−0.5
F(t)
0
−1
−2
N(t)
60
59
58
0
2
4
6
8


3
2
1
0
−1
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


φ(t)
0.2
0.1
0.0
−0.1
ψ(t)
0.5
0.0
−0.5
F(t)
0
−1
−2
N(t)
60
59
58
0
2
4
6
8