Московский авиационный институт
(национальный исследовательский университет)
Институт № 8 «Информационные технологии и прикладная математика»

Лабораторная работа №3
по курсу «Теоретическая механика»
Динамика системы

Выполнил студент группы М8О-203Б-20
Сорокин Никита Эдуардович
Преподаватель: Беличенко Михаил Валериевич
Холостова Ольга Владимировна

Оценка:

Дата: 10.05.2022

Москва, 2022

Вариант №«31»

Задание:

Численно решить дифференциальные уравнения движения механической системы с помощью Python, сделать задание №12 курсовой и построить анимацию движения системы.

12. Задавая численные значения параметров и начальные условия: $m_1 = 5$ кг, $m_2 = 1$ кг, $R = 1$ м, $r = 0,1$ м, $c = 10$ Н/м, $F_0 = 1$ Н, $\gamma = \pi$ c^{-1}, $t_0 = 0$, $\varphi_0 = 0$, $\psi_0 = \pi/6$, $\dot{\varphi} = \dot{\psi} = 0$, составить программу решения системы дифференциальных уравнений и на ЭВМ построить зависимости $\varphi(t)$, $\psi(t)$, $F_A(t)$, $N_A(t)$.

Механическая система:

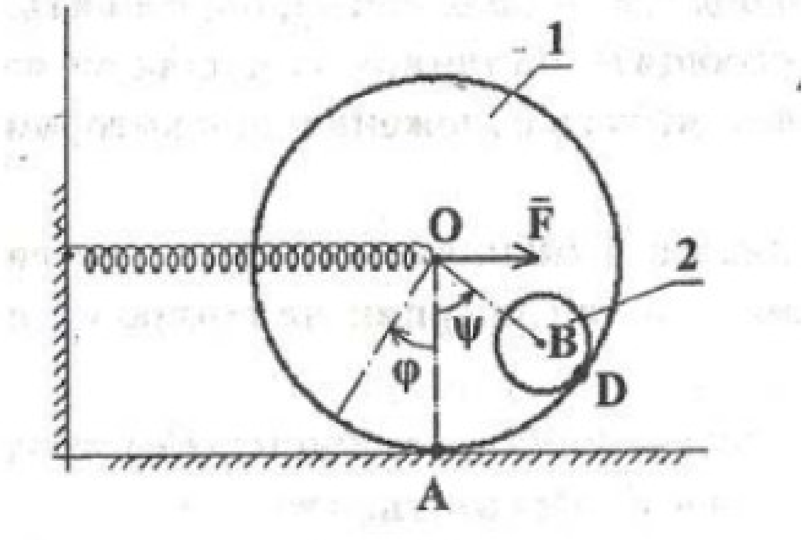


Рис. 31

Текст программы

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import sympy as sp
from scipy.integrate import odeint

pi = 4 * np.arctan(1)
g = 9.81
full_output = 1

# y = (phi, psi, phi', psi')

def odesys(y, t, m1, m2, R, r, c, F0, gamma, g):
    der_y = np.zeros(4)
    der_y[0] = y[2]     # phi' = y[2]
    der_y[1] = y[3]     # psi' = y[3]

    a11 = 2 * (m1 + m2) * R
    a12 = m2 * (R - r) * (1 + np.cos(y[1]))
    a21 = R * (1 + np.cos(y[1]))
    a22 = 2 * (R - r) * r

    b1 = (F0 * np.sin(gamma * t)) - (c * R * y[0]) + (m2 * (R - r) * y[3] * y[3] * np.sin(y[1]))
    b2 = - g * np.sin(y[1])

    der_y[2] = (b1*a22 - b2*a12)/(a11*a22 - a12*a21)    # phi'' находится по методу Крамера
    der_y[3] = (b2*a11 - b1*a21)/(a11*a22 - a12*a21)    # psi'' находится по методу Крамера

    return der_y

# Заданные числовые значения и начальные условия

####################

m1 = 5
m2 = 0.5
R = 1
r = 0.1
c = 10
F0 = 1
gamma = pi

phi0 = 0
psi0 = pi / 6
der_phi0 = 0
der_psi0 = 0

####################

Steps = 1000
t_fin = 20
t = np.linspace(0, t_fin, Steps)

y0 = [phi0, psi0, der_phi0, der_psi0]
```

```
Y = odeint(odesys, y0, t, (m1, m2, R, r, c, F0, gamma, g))
print(Y)

# der - derivative

phi = Y[:, 0]
psi = -pi/2 + Y[:, 1]

der_phi = Y[:, 2]
der_psi = Y[:, 3]

der2_phi = [odesys(y, t, m1, m2, R, r, c, F0, gamma, g)[2] for y,t in zip(Y,t)]
der2_psi = [odesys(y, t, m1, m2, R, r, c, F0, gamma, g)[2] for y,t in zip(Y,t)]

print(der2_phi[1])

F_A = (m1 + m2)*R*der2_phi[1] + m2*(R - r)*(der2_psi*np.cos(psi) - der_psi*der_psi*np.sin(psi)) + c*R*phi -
F0*np.sin(gamma*t)
N_A = (m1 + m2)*g + m2*(R - r)*(der2_psi*np.sin(psi) - der_psi*der_psi*np.cos(psi))

fig_for_graphs = plt.figure(figsize=[13,7])

ax_for_graphs = fig_for_graphs.add_subplot(2, 2, 1)
ax_for_graphs.plot(t, phi, color='blue')
ax_for_graphs.set_title("phi(t)")
ax_for_graphs.set(xlim=[0, t_fin])
ax_for_graphs.grid(True)

ax_for_graphs = fig_for_graphs.add_subplot(2,2,2)
ax_for_graphs.plot(t,psi + pi/2,color='red')
ax_for_graphs.set_title('psi(t)')
ax_for_graphs.set(xlim=[0,t_fin])
ax_for_graphs.grid(True)

ax_for_graphs = fig_for_graphs.add_subplot(2,2,3)
ax_for_graphs.plot(t,F_A,color='green')
ax_for_graphs.set_title("F_A(t)")
ax_for_graphs.set(xlim=[0,t_fin])
ax_for_graphs.grid(True)

ax_for_graphs = fig_for_graphs.add_subplot(2,2,4)
ax_for_graphs.plot(t,N_A,color='black')
ax_for_graphs.set_title('N_A(t)')
ax_for_graphs.set(xlim=[0,t_fin])
ax_for_graphs.grid(True)

# Отсюда начинается лр2

R1_array = np.full(Steps, R, dtype=int)

r_0 = 2

Point_O = R * phi + r_0
```

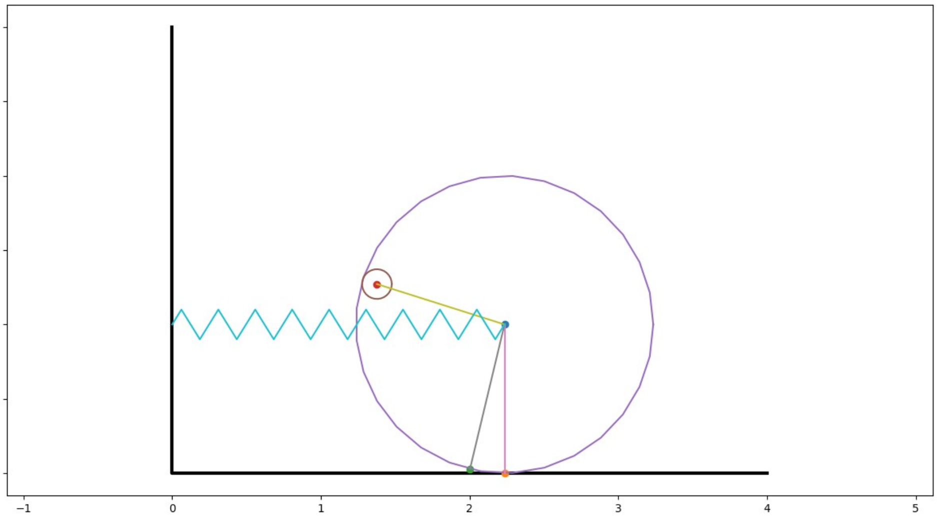
```python
theta = np.linspace(0, 2 * pi, 30)
Circle1_X = R * np.cos(theta)
Circle1_Y = R * np.sin(theta)

Ground_X = [0, 0, 4]
Ground_Y = [3, 0, 0]

Line_OH_X = [0, 0]
Line_OH_Y = [0, R]

Point_A_X = Point_O + R * np.cos(- phi - pi / 2)
Point_A_Y = R1_array + R * np.sin(- phi - pi / 2)



Point_B_X = Point_O + (R - r) * np.cos(psi)
Point_B_Y = R1_array + (R - r) * np.sin(psi)

Circle2_X = r * np.cos(theta)
Circle2_Y = r * np.sin(theta)

AnglesCount = 20
MaxWidth = 0.1

Spring_X = np.zeros(AnglesCount)
Spring_Y = np.zeros(AnglesCount)

Spring_X[AnglesCount - 1] = 1

k = AnglesCount - 2

for i in range(AnglesCount - 2):
    Spring_X[i + 1] = (i + 1) / k - 1 / (2 * k)
    Spring_Y[i + 1] = ((-1) ** i) * MaxWidth

Spring_Length = Point_O



Figure = plt.figure(figsize=[15,8])
ax = Figure.add_subplot(1, 1, 1)
ax.axis("equal")

Drawed_Ground = ax.plot(Ground_X, Ground_Y, color="black", linewidth=3)

Drawed_Point_O = ax.plot(Point_O[0], R, marker="o")[0]
Drawed_Point_H = ax.plot(Point_O[0], 0, marker="o")[0]
Drawed_Point_A = ax.plot(Point_A_X[0], Point_A_Y[0], marker="o")[0]
Drawed_Point_B = ax.plot(Point_B_X[0], Point_B_Y[0], marker="o")[0]

Drawed_Circle1 = ax.plot(Point_O[0] + Circle1_X, R + Circle1_Y)[0]
Drawed_Circle2 = ax.plot(Point_B_X[0] + Circle2_X, Point_B_Y[0] + Circle2_Y)[0]

Drawed_Line_OH = ax.plot(Line_OH_X, Line_OH_Y)[0]
Drawed_Line_OA = ax.plot([Point_O[0], Point_A_X[0]], [R1_array[0], Point_A_Y[0]])[0]
Drawed_Line_OB = ax.plot([Point_O[0], Point_B_X[0]], [R1_array[0], Point_B_Y[0]])[0]
```

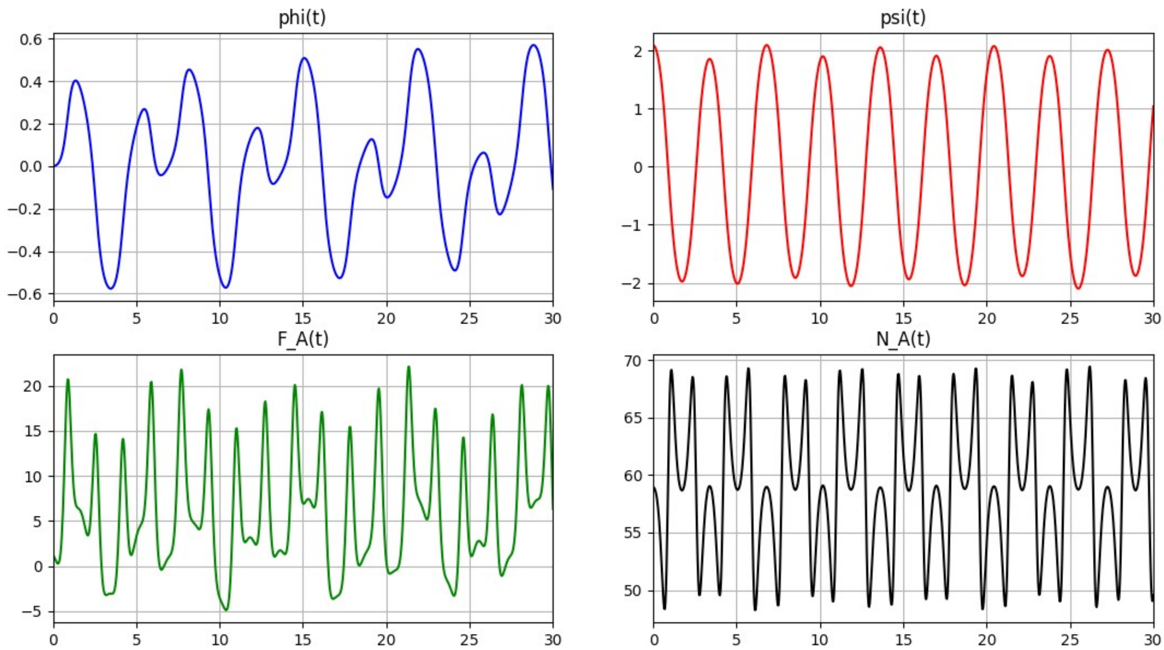
```python
Drawed_Spring = ax.plot(Spring_X * Spring_Length[0], Spring_Y + R)[0]


def Movement(i) :
    Drawed_Point_O.set_data(Point_O[i], R)
    Drawed_Point_H.set_data(Point_O[i], 0)
    Drawed_Point_A.set_data(Point_A_X[i], Point_A_Y[i])
    Drawed_Point_B.set_data(Point_B_X[i], Point_B_Y[i])

    Drawed_Circle1.set_data(Point_O[i] + Circle1_X, R + Circle1_Y)
    Drawed_Circle2.set_data(Point_B_X[i] + Circle2_X, Point_B_Y[i] + Circle2_Y)

    Drawed_Line_OH.set_data(Line_OH_X + Point_O[i], Line_OH_Y)
    Drawed_Line_OA.set_data([Point_O[i], Point_A_X[i]], [R1_array[i], Point_A_Y[i]])
    Drawed_Line_OB.set_data([Point_O[i], Point_B_X[i]], [R1_array[i], Point_B_Y[i]])

    Drawed_Spring.set_data(Spring_X * Spring_Length[i], Spring_Y + R)


Animation = FuncAnimation(Figure, Movement, frames=Steps, interval=10)

plt.show()
```

Результат работы:

Тест 1. Набор данных похожий на набор из №12

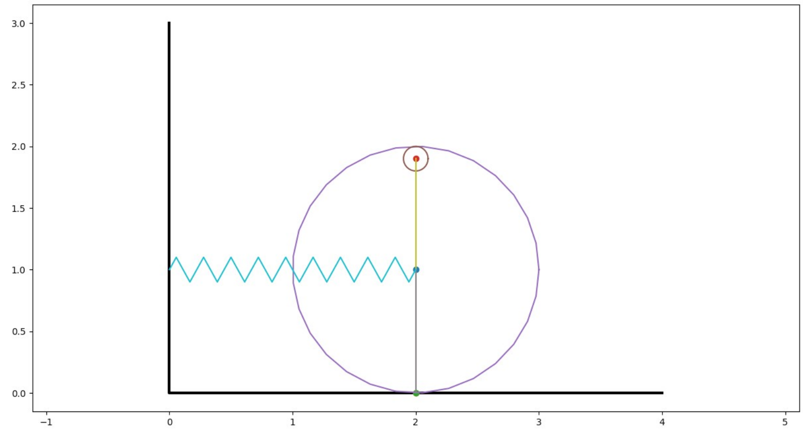
```
m1 = 5
m2 = 1
R = 1
r = 0.1
c = 10
F0 = 1
gamma = pi

phi0 = 0
psi0 = 2 * pi / 3
der_phi0 = 0
der_psi0 = 0
```

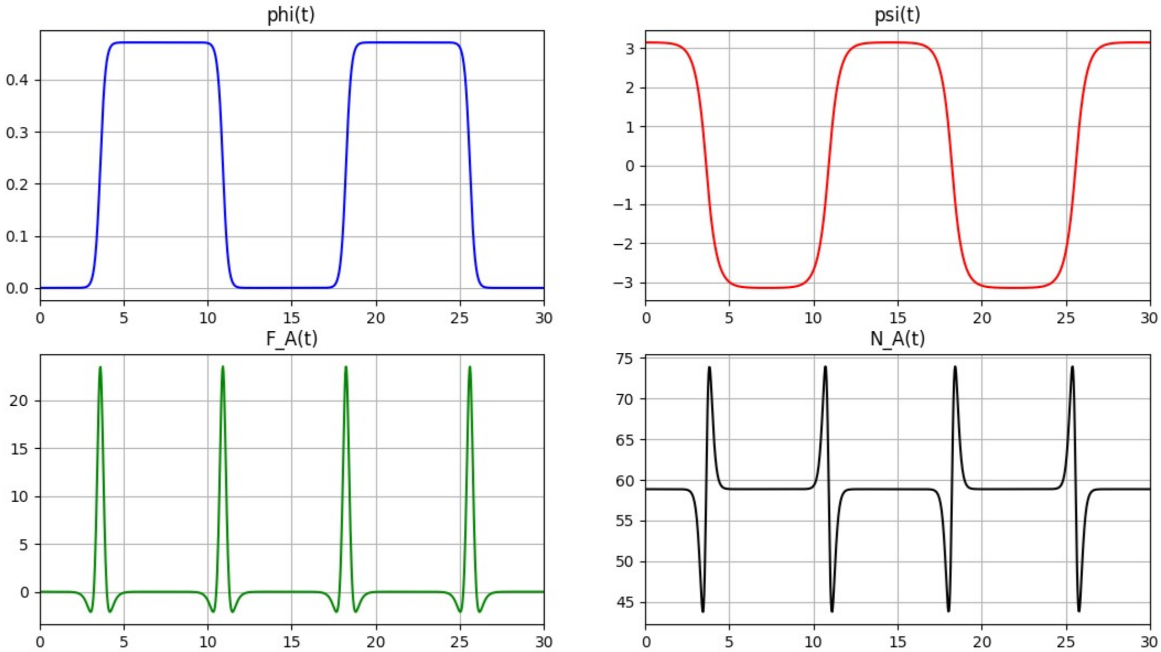




Тест 2. Положение неустойчивого равновесия

```
m1 = 5
m2 = 0.5
R = 1
r = 0.1
c = 0
F0 = 0
gamma = pi

phi0 = 0
psi0 = pi
der_phi0 = 0
der_psi0 = 0
```





Тест 3. Сильная пружина

```
m1 = 5
m2 = 1
R = 1
r = 0.1
c = 20
F0 = 0
gamma = pi

phi0 = 1
psi0 = 0
der_phi0 = 0
der_psi0 = 0
```