Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №2**

**по курсу «Теоретическая механика»**

**Анимация системы**

Выполнила студентка группы М8О-201Б-20

Гусева Софья Романовна

Преподаватель: Беличенко Михаил Валериевич

Оценка:

Дата: 27.12.2021

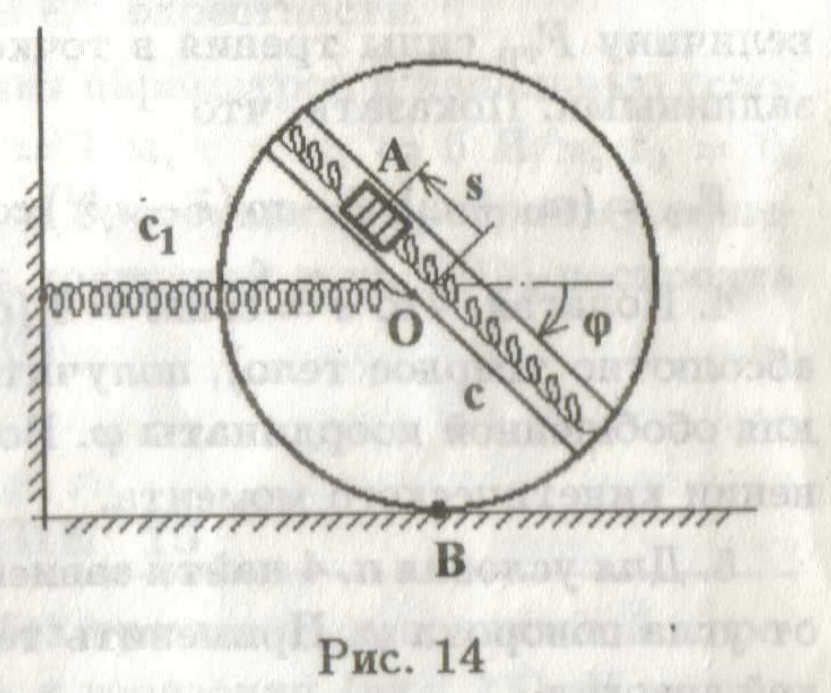
Москва, 2021

**Вариант №2 «Фантастическая система»**

**Задание:**

Реализовать анимацию движения механической системы на языке Python.

**Механическая система:**

****

**Текст программы**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

fig = plt.figure(figsize=(10, 5))

ax = fig.add\_subplot(1, 2, 1)

ax.axis('equal')

ax.set\_xlim([-3, 3])

ax.set\_ylim([-3, 3])

t = sp.Symbol('t')

OX = 0.0

OY = -2

U2 = t \* 2

U3 = sp.sin(U2)

x1 = sp.sin(U2)

y1 = OY

vx1 = sp.diff(x1, t)

vy1 = sp.diff(y1, t)

x2 = x1

y2 = y1 - 1

x3 = sp.sin(U3) + x1

y3 = sp.cos(U3) + y1

# vx3 = sp.diff(x3, t)

# vy3 = sp.diff(y3, t)

x4 = -sp.sin(U3) + x1

y4 = -sp.cos(U3) + y1

vx4 = sp.diff(x4, t)

vy4 = sp.diff(y4, t)

x5 = sp.sin(U3) \* U3 + x1

y5 = sp.cos(U3) \* U3 + y1

vx5 = sp.diff(x5, t)

vy5 = sp.diff(y5, t)

def updateScrew(poly, angl, sz, ox, oy):

XY = [(sp.sin(angl + ct1) \* sz + ox, sp.cos(angl + ct1) \* sz + oy) for ct1 in np.linspace(0, math.pi \* 2, 7)]

poly.set\_xy(XY)

xmax = 6.28 \* 4

T = np.linspace(0, xmax, 500)

X1 = np.zeros\_like(T)

Y1 = np.zeros\_like(T)

VX1 = np.zeros\_like(T)

VY1 = np.zeros\_like(T)

X2 = np.zeros\_like(T)

Y2 = np.zeros\_like(T)

VX2 = np.zeros\_like(T)

VY2 = np.zeros\_like(T)

X3 = np.zeros\_like(T)

Y3 = np.zeros\_like(T)

VX3 = np.zeros\_like(T)

VY3 = np.zeros\_like(T)

X4 = np.zeros\_like(T)

Y4 = np.zeros\_like(T)

VX4 = np.zeros\_like(T)

VY4 = np.zeros\_like(T)

X5 = np.zeros\_like(T)

Y5 = np.zeros\_like(T)

VX5 = np.zeros\_like(T)

VY5 = np.zeros\_like(T)

for i in np.arange(len(T)):

X1[i] = sp.Subs(x1, t, T[i])

Y1[i] = sp.Subs(y1, t, T[i])

VX1[i] = sp.Subs(vx1, t, T[i])

VY1[i] = sp.Subs(vy1, t, T[i])

X2[i] = sp.Subs(x2, t, T[i])

Y2[i] = sp.Subs(y2, t, T[i])

X3[i] = sp.Subs(x3, t, T[i])

Y3[i] = sp.Subs(y3, t, T[i])

X4[i] = sp.Subs(x4, t, T[i])

VX4[i] = sp.Subs(vx4, t, T[i])

VY4[i] = sp.Subs(vy4, t, T[i])

Y4[i] = sp.Subs(y4, t, T[i])

X5[i] = sp.Subs(x5, t, T[i])

Y5[i] = sp.Subs(y5, t, T[i])

VX5[i] = sp.Subs(vx5, t, T[i])

VY5[i] = sp.Subs(vy5, t, T[i])

gr1 = fig.add\_subplot(6, 2, 2)

gr1.plot(T, VX1)

gr1.set\_xlabel('T')

gr1.set\_ylabel('VXO')

grl1 = gr1.axvline(0, color='r')

gr2 = fig.add\_subplot(6, 2, 4)

gr2.plot(T, VY1)

gr2.set\_xlabel('T')

gr2.set\_ylabel('VYO')

grl2 = gr2.axvline(0, color='r')

gr3 = fig.add\_subplot(6, 2, 6)

gr3.plot(T, VX4)

gr3.set\_xlabel('T')

gr3.set\_ylabel('VXA')

grl3 = gr3.axvline(0, color='r')

gr4 = fig.add\_subplot(6, 2, 8)

gr4.plot(T, VY4)

gr4.set\_xlabel('T')

gr4.set\_ylabel('VYA')

grl4 = gr4.axvline(0, color='r')

gr5 = fig.add\_subplot(6, 2, 10)

gr5.plot(T, VX5)

gr5.set\_xlabel('T')

gr5.set\_ylabel('VXB')

grl5 = gr5.axvline(0, color='r')

gr6 = fig.add\_subplot(6, 2, 12)

gr6.plot(T, VY5)

gr6.set\_xlabel('T')

gr6.set\_ylabel('VYB')

grl6 = gr6.axvline(0, color='r')

plt.subplots\_adjust(wspace=0.3, hspace=0.7)

circle1 = plt.Circle((0, 0), 1.0, color='red')

circle2 = plt.Circle((0, 0), 0.95, color='white')

hinge1 = plt.Circle((0, 0), 0.03, color='red')

hinge2 = plt.Circle((0, 0), 0.03, color='blue')

circle3 = plt.Circle((0, 0), 0.1, color='orange')

line1 = plt.Polygon(((0, 0), (1, 1)), 0.1, color='green')

line3 = plt.Polygon(((0, 0), (1, 1)), 0.1, color='green')

ax.add\_patch(circle1)

ax.add\_patch(circle2)

ax.add\_patch(line1)

ax.add\_patch(line3)

ax.add\_patch(circle3)

ax.add\_patch(hinge1)

ax.add\_patch(hinge2)

ax.legend([hinge1, hinge2, circle3], ['O', 'A', 'B'])

def anima(t1):

circle1.center = X1[t1], Y1[t1]

circle2.center = X1[t1], Y1[t1]

circle3.center = X5[t1], Y5[t1]

hinge1.center = X1[t1], Y1[t1]

hinge2.center = X4[t1], Y4[t1]

line1.set\_xy(((-3, -2), (X1[t1], Y1[t1])))

# line2.set\_xy(((X1[t1],Y1[t1]),(X2[t1],Y2[t1])))

line3.set\_xy(((X4[t1], Y4[t1]), (X3[t1], Y3[t1])))

grl1.set\_xdata(t1 / 500 \* xmax)

grl2.set\_xdata(t1 / 500 \* xmax)

grl3.set\_xdata(t1 / 500 \* xmax)

grl4.set\_xdata(t1 / 500 \* xmax)

grl5.set\_xdata(t1 / 500 \* xmax)

grl6.set\_xdata(t1 / 500 \* xmax)

return []

# animation function

anim = FuncAnimation(fig, anima, frames=500, interval=10, blit=False)

plt.show()

**Результат работы:**

