ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**

**ВАРИАНТ ЗАДАНИЯ №27**

Выполнил(а) студент группы М8О-201Б-22

Чибугаев И.А..\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

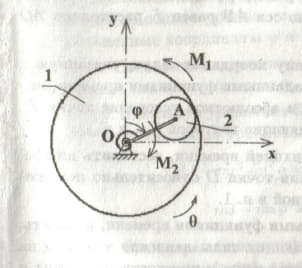
подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва, 2024

*Задание:* Реализовать анимацию движения механической системы используя язык программирования Python.

*Вариант:*



*Код:*

import numpy as n

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

*#задается интервал*

step = 500

t = n.linspace(0, 10, step)

*#задается функция*

phi = n.sin(2\*t)

theta = n.linspace(0, -2\*n.pi, step)

O = 0

r2 = 0.5*#внешний радиус пружины*

r1 = 0.2*#внутренний радиус пружины*

R1 = 3 *#радиус кольца*

*#координаты точки А*

Xa = (R1 - 1) \* n.cos(n.pi/2+phi)

Ya = (R1 - 1) \* n.sin(n.pi/2+phi)

*#координаты точки Б*

Xb = - R1 \* n.cos(n.pi/2+theta)

Yb = R1 \* n.sin(n.pi/2+theta)

*#окно и грaфик*

fgr = plt.figure()

grf = fgr.add\_subplot(1, 1, 1)

grf.axis('equal')

grf.set(xlim = [-5, 5], ylim = [-5, 5])

grf.set\_aspect( 1 )

*#треугольник*

grf.plot([0,0.5,-0.5,0], [0,-1,-1,0], color = 'black')

*#точки на графике*

p1 = grf.plot(O, O, marker = 'o', color = 'black')[0]

pA = grf.plot(Xa[0], Ya[0], marker = 'o', color = 'black')[0]

pB = grf.plot(Xb[0], Yb[0], marker = 'o', color = 'black')[0]

*#прямая ОА*

O1 = grf.plot([Xa[0], O],[Ya[0], O],color = 'black')[0]

*#большая окружность*

circle = plt.Circle(( O, O ), R1 , fill = False)

grf.add\_artist(circle)

*#маленькая окружность*

circleA = plt.Circle(( Xa[0], Ya[0]), 1 , fill = False)

grf.add\_artist(circleA)

*#спиральная пружина*

Ns = 2

numpnts = n.linspace(0, 1, 50\*Ns+1)

Betas = numpnts\*(Ns \* 2\*n.pi - phi[0])

Xs = ((r2-r1)\*numpnts)\*n.cos(Betas + n.pi/2)

Ys = ((r2-r1)\*numpnts)\*n.sin(Betas + n.pi/2)

SpPruzh = grf.plot(Xs, Ys, color = 'black')[0]

*#анимация*

def run(i):

    pA.set\_data([Xa[i]], [Ya[i]])

    pB.set\_data([Xb[i]], [Yb[i]])

    O1.set\_data([Xa[i], O], [Ya[i], O])

    circleA.center = (Xa[i],Ya[i])

    Betas = numpnts\*(Ns \* 2\*n.pi - phi[i])

    Xs = -((r2-r1)\*numpnts)\*n.cos(Betas+n.pi/2)

    Ys = ((r2-r1)\*numpnts)\*n.sin(Betas+n.pi/2)

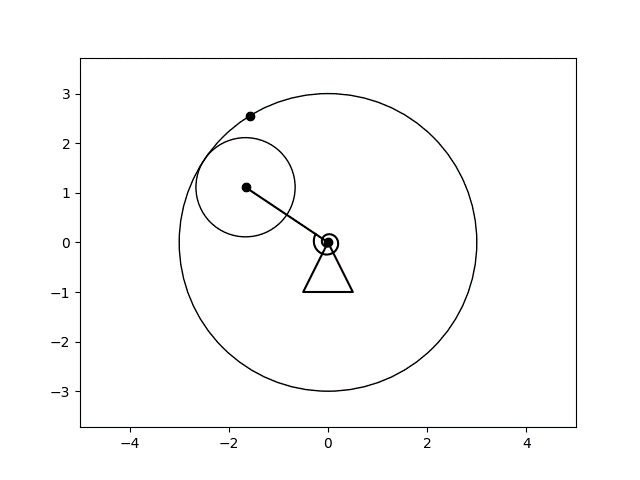
    SpPruzh.set\_data(Xs, Ys)

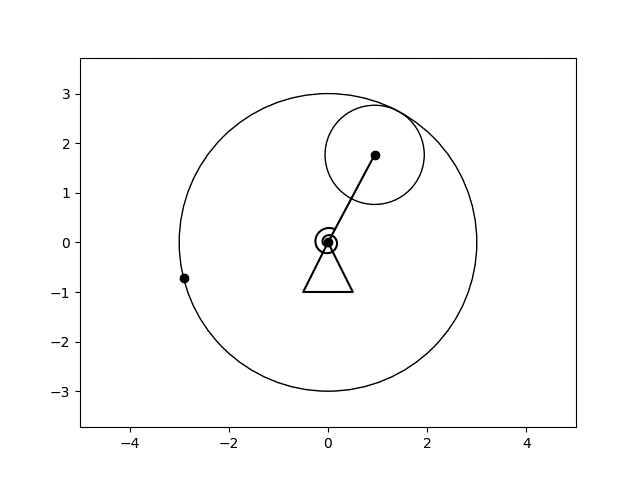
    return

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

fgr.show()

*Скриншот:*





*Вывод:* В ходе лабораторной работы была создана анимация механической системы с использованием библиотек numpy и matplotlib. Шаги включали в себя определение параметров системы, расчет координат точек, и создание анимации. Комментарии были добавлены для понимания кода, и использовались различные параметры для улучшения визуального восприятия. Также рассматривались вопросы управления скоростью анимации и решения проблем с её остановкой.