

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

**ОТЧЕТ**  
**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**  
**«АНИМАЦИЯ СИСТЕМЫ»**  
**ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И**  
**ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»**  
**ВАРИАНТ ЗАДАНИЯ № 29**

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

Полиха Александр Владимирович \_\_\_\_\_  
подпись, дата

Проверил и принял

Авдюшкин А.Н. \_\_\_\_\_  
подпись, дата

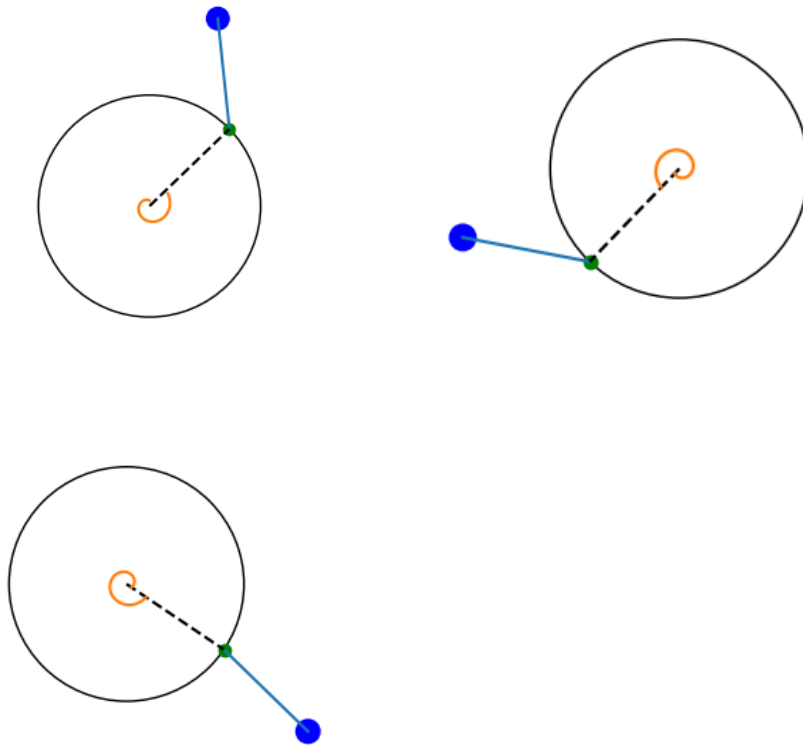
с оценкой \_\_\_\_\_

Москва, 2023

## Задание

Построить анимацию движения системы с помощью Python.

## Результат



## Код

```
import numpy as n
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from matplotlib.patches import Circle
import math

step = 1000
t = n.linspace(0,10,step)
x = n.sin(t)
phi = n.sin(2*t)

fgr = plt.figure()
gr = fgr.add_subplot(1,1,1)
gr.axis('equal')
gr.set_xlim((-3, 3))
gr.set_ylim((-3, 3))
```

```

R = 1
RB = 0.1
L = 1

circle = Circle((0, 0), R, fill=False)
circle_patch = gr.add_patch(circle)

Xa = n.cos(phi)
Ya = n.sin(phi)
Xb = Xa + n.cos(phi)
Yb = Ya + n.sin(phi)

pA = gr.plot(Xa[0], Ya[0], marker='o', color='g')[0]
pB = gr.plot(Xb[0], Yb[0], marker='o', color='b')[0]
AB = gr.plot([Xa[0], Xb[0]], [Ya[0], Yb[0]])[0]
B_circle = Circle((Xb[0], Yb[0]), RB, fill=True, color='b')
gr.add_patch(B_circle)

OA = gr.plot([0, Xa[0]], [0, Ya[0]], 'k--')[0]

# Spiral
Ns = 1
r1 = 0.05
r2 = 0.2
numpts = n.linspace(0, 1, 50*Ns+1)
Betas = numpts * (Ns * 2*n.pi - phi[0])
Xs = (r1 + (r2-r1) * numpts) * n.cos(Betas + n.pi/2)
Ys = (r1 + (r2-r1) * numpts) * n.sin(Betas + n.pi/2)

Spiral = gr.plot(Xs, Ys)[0]

def update(i):
    global Xa, Ya, Xb, Yb, B_circle, OA
    psi = n.cos(t[i] * 3)
    theta = n.radians(i)
    Xa = n.cos(theta)
    Ya = n.sin(theta)
    Xb = Xa + n.cos(theta + psi) * L
    Yb = Ya + n.sin(theta + psi) * L
    pA.set_data(Xa, Ya)
    pB.set_data(Xb, Yb)
    AB.set_data([Xa, Xb], [Ya, Yb])
    B_circle.set_center((Xb, Yb))
    OA.set_data([0, Xa], [0, Ya])

    Betas = numpts * (Ns * 2*n.pi)
    Xs = (r1 + (r2-r1) * numpts) * n.cos(Betas+theta)
    Ys = (r1 + (r2-r1) * numpts) * n.sin(Betas+theta)

    Spiral.set_data(Xs, Ys)

```

```
    return circle_patch, pA, pB, AB, B_circle, OA, Spiral  
  
anim = FuncAnimation(fgr, update, frames=step, interval=1, blit=True)  
  
plt.show()
```

## Вывод

Лабораторная работа выполнена. Научился строить анимацию движения системы с помощью Python.