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 №23**

Выполнил(а) студент группы М8О-207Б-22

Лебедько Платон Владимирович\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

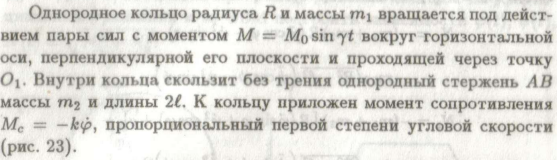
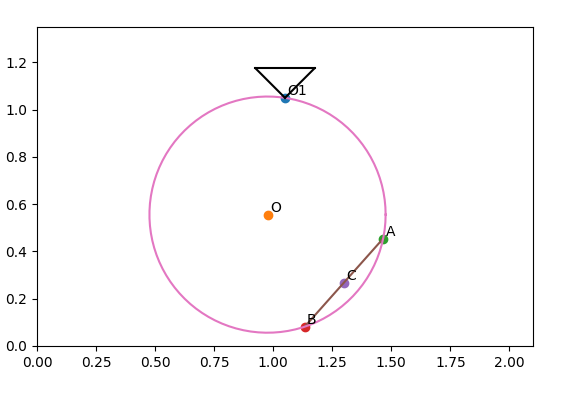
Зав. каф. 802, Бардин Б.С.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

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

Москва, 2023

*Задание:* построить анимацию движения системы с помощью Python.

1. *Условия задачи 23 варианта:*  
   
2. *Рисунок получившейся физической модели:  
   *
3. *Код программы:*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import math

# Вспомогательные

forLetters = 0.2

# Определяем отрисовку

steps = 1001

t\_fin = 20

t = np.linspace(0, t\_fin, steps)

# Функции

phi = 3\*np.sin(1.8\*t)

psi = 0.5\*np.sin(2\*t)

# Константы

R = 10

l = 5

# Вычисленные константы

alpha = -math.acos(l / R)

OC = -R \* math.sin(alpha)

# Задаем положение

O1X = 2 \* R + 1

O1Y = 2 \* R + 1

# относительно т. О1

OX = O1X + R \* np.sin(phi)

OY = O1Y - R \* np.cos(phi)

# относительно т. О

CX = OX + OC \* np.sin(psi)

CY = OY - OC \* np.cos(psi)

AX = OX + R \* np.cos(alpha + psi)

AY = OY + R \* np.sin(alpha + psi)

BX = OX + R \* np.cos(math.pi - alpha + psi)

BY = OY + R \* np.sin(math.pi - alpha + psi)

fig = plt.figure()

ax1 = fig.add\_subplot()

ax1.axis('equal')

plt.gca().set\_adjustable("box")

ax1.set(xlim=[0, 4 \* R + 2], ylim=[0, 4 \* R + 2])

ax1.plot(O1X, O1Y, marker = 'o')

plt.text(O1X + forLetters, O1Y + forLetters, 'O1')

OPoint = ax1.plot(OX[0], OY[0], marker = 'o')[0]

APoint = ax1.plot(AX[0], AY[0], marker = 'o')[0]

BPoint = ax1.plot(BX[0], BY[0], marker = 'o')[0]

CPoint = ax1.plot(CX[0], CY[0], marker = 'o')[0]

OText = plt.text(OX[0] + forLetters, OY[0] + forLetters, 'O')

AText = plt.text(AX[0] + forLetters, AY[0] + forLetters, 'A')

BText = plt.text(BX[0] + forLetters, BY[0] + forLetters, 'B')

CText = plt.text(CX[0] + forLetters, CY[0] + forLetters, 'C')

ABLine = ax1.plot([ AX[0], BX[0] ], [ AY[0], BY[0] ])[0]

phiForCirc = np.linspace(0, 2\*math.pi, 100)

Circ = ax1.plot(OX[0] + R \* np.cos(phiForCirc), OY[0] + R \* np.sin(phiForCirc))[0]

def anima(i):

OPoint.set\_data([OX[i]], [OY[i]])

APoint.set\_data([AX[i]], [AY[i]])

BPoint.set\_data([BX[i]], [BY[i]])

CPoint.set\_data([CX[i]], [CY[i]])

OText.set\_position([OX[i] + forLetters, OY[i] + forLetters])

AText.set\_position([AX[i] + forLetters, AY[i] + forLetters])

BText.set\_position([BX[i] + forLetters, BY[i] + forLetters])

CText.set\_position([CX[i] + forLetters, CY[i] + forLetters])

ABLine.set\_data([ AX[i], BX[i] ], [ AY[i], BY[i] ])

Circ.set\_data(OX[i] + R \* np.cos(phiForCirc), OY[i] + R \* np.sin(phiForCirc))

anim = FuncAnimation(fig, anima, frames = steps, interval = 100)

anim\_running = True

def onClick(event):

global anim\_running

if anim\_running:

anim.event\_source.stop()

anim\_running = False

else:

anim.event\_source.start()

anim\_running = True

fig.canvas.mpl\_connect('button\_press\_event', onClick)

plt.show()

1. *Вывод*

Я построил анимацию движения системы с помощью Python. Я научился строить и анимировать движение окружности и отрезка, лежащего внутри этой окружности.