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 № 2**

Выполнил(а) студент группы М8О-208Б-21

Белоносов К.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

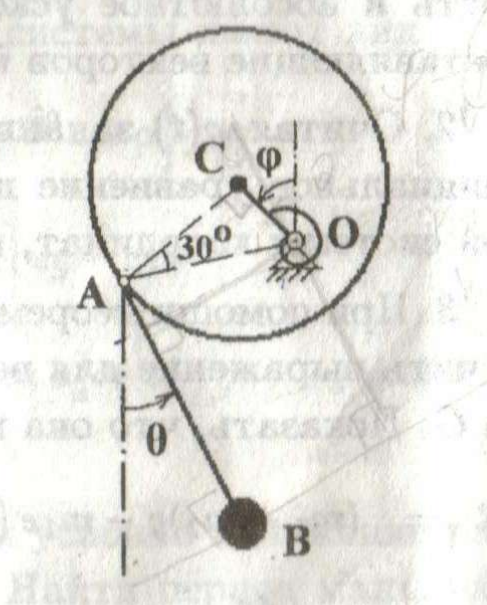
подпись, дата

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

Москва, 2022

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

*Механическая система:*



*Текст программы:*

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

Steps = 1001

t\_fin = 20

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

R = 1

phi = 2\*np.sin(1.7\*t) + 5\*np.cos(1.2\*t)

ksi = 4\*np.sin(2\*t) + 5\*np.cos(3\*t)

psi = np.linspace(0, 6.28, 20)

C = 1

R1 = 0.2

R2 = R/(3\*\*0.5)/2

l = 1

X\_Wheel = R\*np.sin(psi)

Y\_Wheel = R\*np.cos(psi)

X\_C = (R/(3\*\*0.5))\*np.sin(phi)

Y\_C = (R/(3\*\*0.5))\*np.cos(phi)

X\_A = X\_C + R\*np.sin(phi - 1.57)

Y\_A = Y\_C + R\*np.cos(phi - 1.57)

X\_B = X\_A + l\*np.cos(ksi)

Y\_B = Y\_A + l\*np.sin(ksi)

thetta = np.linspace(0, C\*6.28-phi[0], 100)

X\_SpiralSpr = -(R1 + thetta\*(R2-R1)/thetta[-1])\*np.sin(thetta)

Y\_SpiralSpr = (R1 + thetta\*(R2-R1)/thetta[-1])\*np.cos(thetta)

fig = plt.figure(figsize=[15, 7])

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

ax.axis('equal')

ax.set(xlim=[-6, 6], ylim=[-6, 6])

Circ = ax.plot(X\_C[0]+R \* np.cos(phi), Y\_C[0]+R \* np.sin(phi), 'yellow')[0]

Point\_O = ax.plot(0, 0, marker='o')[0]

Point\_C = ax.plot(X\_C[0], Y\_C[0], marker='o')[0]

Point\_A = ax.plot(X\_A[0], Y\_A[0], marker='o')[0]

Point\_B = ax.plot(X\_B[0], Y\_B[0], marker='o', markersize=20)[0]

Line\_CO = ax.plot([0, X\_C[0]], [0, Y\_C[0]], 'blue')[0]

Line\_AB = ax.plot([X\_A[0], X\_B[0]], [Y\_A[0], Y\_B[0]])[0]

Drawed\_SpiralSpring = ax.plot(X\_SpiralSpr, Y\_SpiralSpr)[0]

def anima(i):

Point\_C.set\_data(X\_C[i], Y\_C[i])

Point\_A.set\_data(X\_A[i], Y\_A[i])

Point\_B.set\_data(X\_B[i], Y\_B[i])

Line\_AB.set\_data([X\_A[i], X\_B[i]], [Y\_A[i], Y\_B[i]])

Circ.set\_data(X\_C[i] + R \* np.cos(phi), Y\_C[i] + R \* np.sin(phi))

Line\_CO.set\_data([0, X\_C[i]], [0, Y\_C[i]])

thetta = np.linspace(0, C \* 6.28 - phi[i], 100)

X\_SpiralSpr = -(R1 + thetta \* (R2 - R1) / thetta[-1]) \* np.sin(thetta)

Y\_SpiralSpr = (R1 + thetta \* (R2 - R1) / thetta[-1]) \* np.cos(thetta)

Drawed\_SpiralSpring.set\_data(X\_SpiralSpr, Y\_SpiralSpr)

return [Point\_C, Point\_A, Point\_B, Circ, Line\_CO, Line\_AB, Drawed\_SpiralSpring]

anim = FuncAnimation(fig, anima, frames=len(t), interval=3)

plt.show()

*Результаты работы:*

|  |
| --- |
|  |
|  |
|  |
|  |