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 №21**

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

Клименко В.М.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Авдюшкин А.Н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

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

Москва, 2023

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

*Код:*

import math

import numpy

import sympy

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

circle\_radius = 0.75

def draw\_circle(X, Y):

CX = [X + circle\_radius \* math.cos(i / 10) for i in range(0, 64)]

CY = [Y + circle\_radius \* math.sin(i / 10) for i in range(0, 64)]

return CX, CY

def animation(i):

beam1.set\_data([-4, -4 + XA[i]], [0, YA[i]])

beam2.set\_data([4, 4 + XA[i]], [0, YA[i]])

beam3.set\_data([-4 + XA[i], 4 + XA[i]], [YA[i], YA[i]])

beam4.set\_data([XA[i], XA[i] + XO[i]], [YA[i], YA[i] + YO[i]])

circle.set\_data(\*draw\_circle(XA[i] + XO[i], YA[i] + YO[i]))

return beam1, beam2, beam3, beam4, circle

t = sympy.Symbol('t')

phi = sympy.sin(t) \* sympy.cos(t) + math.pi/4

theta = 3 \* t

beam1\_length = 4

beam2\_length = 1.5

# speed and acceleration of point A

A\_x\_velocity = sympy.diff(sympy.sin(phi) \* beam1\_length, t)

A\_y\_velocity = sympy.diff(sympy.cos(phi) \* beam1\_length, t)

A\_velocity = (A\_x\_velocity\*\*2 + A\_y\_velocity\*\*2)\*\*0.5

A\_acceleration = (sympy.diff(A\_x\_velocity, t)\*\*2 + sympy.diff(A\_y\_velocity, t)\*\*2)\*\*0.5

# speed and acceleration of point C

C\_x\_velocity = sympy.diff(sympy.sin(theta) \* beam2\_length, t) + A\_x\_velocity

C\_y\_velocity = sympy.diff(sympy.cos(theta) \* beam2\_length, t) + A\_y\_velocity

C\_velocity = (C\_x\_velocity\*\*2 + C\_y\_velocity\*\*2)\*\*0.5

C\_acceleration = (sympy.diff(C\_x\_velocity, t)\*\*2 + sympy.diff(C\_y\_velocity, t)\*\*2)\*\*0.5

iterations = 500

T = numpy.linspace(0, 2\*math.pi, num=iterations)

XO = numpy.zeros\_like(T)

YO = numpy.zeros\_like(T)

XA = numpy.zeros\_like(T)

YA = numpy.zeros\_like(T)

VA = numpy.zeros\_like(T)

WA = numpy.zeros\_like(T)

VC = numpy.zeros\_like(T)

WC = numpy.zeros\_like(T)

for i in numpy.arange(iterations):

XO[i] = sympy.Subs(beam2\_length \* sympy.cos(theta), t, T[i])

YO[i] = sympy.Subs(beam2\_length \* sympy.sin(theta), t, T[i])

XA[i] = sympy.Subs(beam1\_length \* sympy.cos(2 \* phi), t, T[i])

YA[i] = sympy.Subs(-math.sqrt(beam1\_length\*\*2 - XA[i]\*\*2), t, T[i])

VA[i] = sympy.Subs(A\_velocity, t, T[i])

WA[i] = sympy.Subs(A\_acceleration, t, T[i])

VC[i] = sympy.Subs(C\_velocity, t, T[i])

WC[i] = sympy.Subs(C\_acceleration, t, T[i])

fig = plt.figure()

ax1 = fig.add\_subplot(1, 1, 1)

ax1.axis('equal')

ax1.set(xlim=[-10, 10], ylim=[-10, 10])

beam1, = ax1.plot([-4, -4 + XA[0]], [0, YA[0]], 'black')

beam2, = ax1.plot([4, 4 + XA[0]], [0, YA[0]], 'black')

beam3, = ax1.plot([-4 + XA[0], 4 + XA[0]], [YA[0], YA[0]], 'black')

beam4, = ax1.plot([XA[0], XA[0] + XO[0]], [YA[0], YA[0] + YO[0]], 'black')

circle, = ax1.plot(\*draw\_circle(XA[0] + XO[0], YA[0] + YO[0]), 'black')

anim = FuncAnimation(fig, animation, frames=iterations - 1, interval=0, cache\_frame\_data=False, blit=True)

plt.show()

*Скриншот:*

