ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«АНИМАЦИЯ СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ ДИНАМИЧЕСКИХ СИСТЕМ»**

**ВАРИАНТ ЗАДАНИЯ №20**

Выполнил(а) студент группы М8О-201Б-23

Терентьев М.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

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

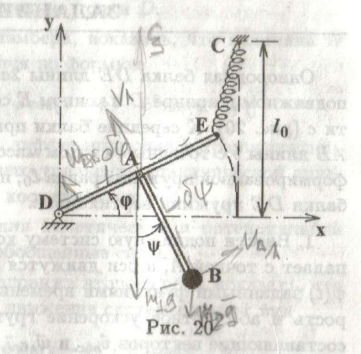
Москва, 2024

**Вариант №20**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**

****

**Код программы:**

import matplotlib.pyplot as plt

import numpy as np

from matplotlib.animation import FuncAnimation

fig = plt.figure()

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

gr.axis('equal')

# Дано:

a = b = l0 = 1

DE = 2 \* a

# Задаю функции phi(t) и psi(t)

step = 3000

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

phi = 2 \* np.sin(6 \* t)

psi = 5 \* t + 0.2 \* np.cos(6 \* t

# Балка DE

Xd = 0

Yd = 0

Xe = Xd + DE \* np.cos(phi)

Ye = Yd + DE \* np.sin(phi)

balkaDE = gr.plot([Xd, Xe[0]], [Yd, Ye[0]], color='black', linewidth=5)[0]

pD = gr.plot(Xd, Yd, marker='o', color='r')[0]

pE = gr.plot(Xe, Ye, marker='o', color='r')[0]

# Пружина

Xc = DE

Yc = l0

pC = gr.plot(Xc, Yc, marker='o', color='r')[0]

def get\_spring(coils, width, start, end):

start, end = np.array(start).reshape((2,)), np.array(end).reshape((2,))

len = np.linalg.norm(np.subtract(end, start))

u\_t = np.subtract(end, start) / len

u\_n = np.array([[0, -1], [1, 0]]).dot(u\_t)

spring\_coords = np.zeros((2, coils + 2))

spring\_coords[:, 0], spring\_coords[:, -1] = start, end

normal\_dist = np.sqrt(max(0, width \*\* 2 - (len \*\* 2 / coils \*\* 2))) / 2

for i in np.arange(1, coils + 1):

spring\_coords[:, -i] = (start

+ ((len \* (2 \* i - 1) \* u\_t) / (2 \* coils))

+ (normal\_dist \* (-1) \*\* i \* u\_n))

return spring\_coords[0, 2:], spring\_coords[1, 2:]

pS = gr.plot(\*get\_spring(70, 0.1, [Xe[0], Ye[0]], [Xc, Yc]), color='black')[0]

# Стержень AB

Xa = Xd + DE / 2 \* np.cos(phi)

Ya = Yd + DE / 2 \* np.sin(phi)

Xb = Xa + b \* np.cos(psi - np.pi / 2)

Yb = Ya + b \* np.sin(psi - np.pi / 2)

sterjenAB = gr.plot([Xa[0], Xb[0]], [Ya[0], Yb[0]],

color='black', linewidth=1)[0]

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

pB = gr.plot(Xb, Yb, marker='o', color='black', markersize=20)[0

def run(i):

balkaDE.set\_data([Xd, Xe[i]], [Yd, Ye[i]])

pE.set\_data(Xe[i], Ye[i])

pS.set\_data(\*get\_spring(70, 0.1, [Xe[i], Ye[i]], [Xc, Yc]))

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

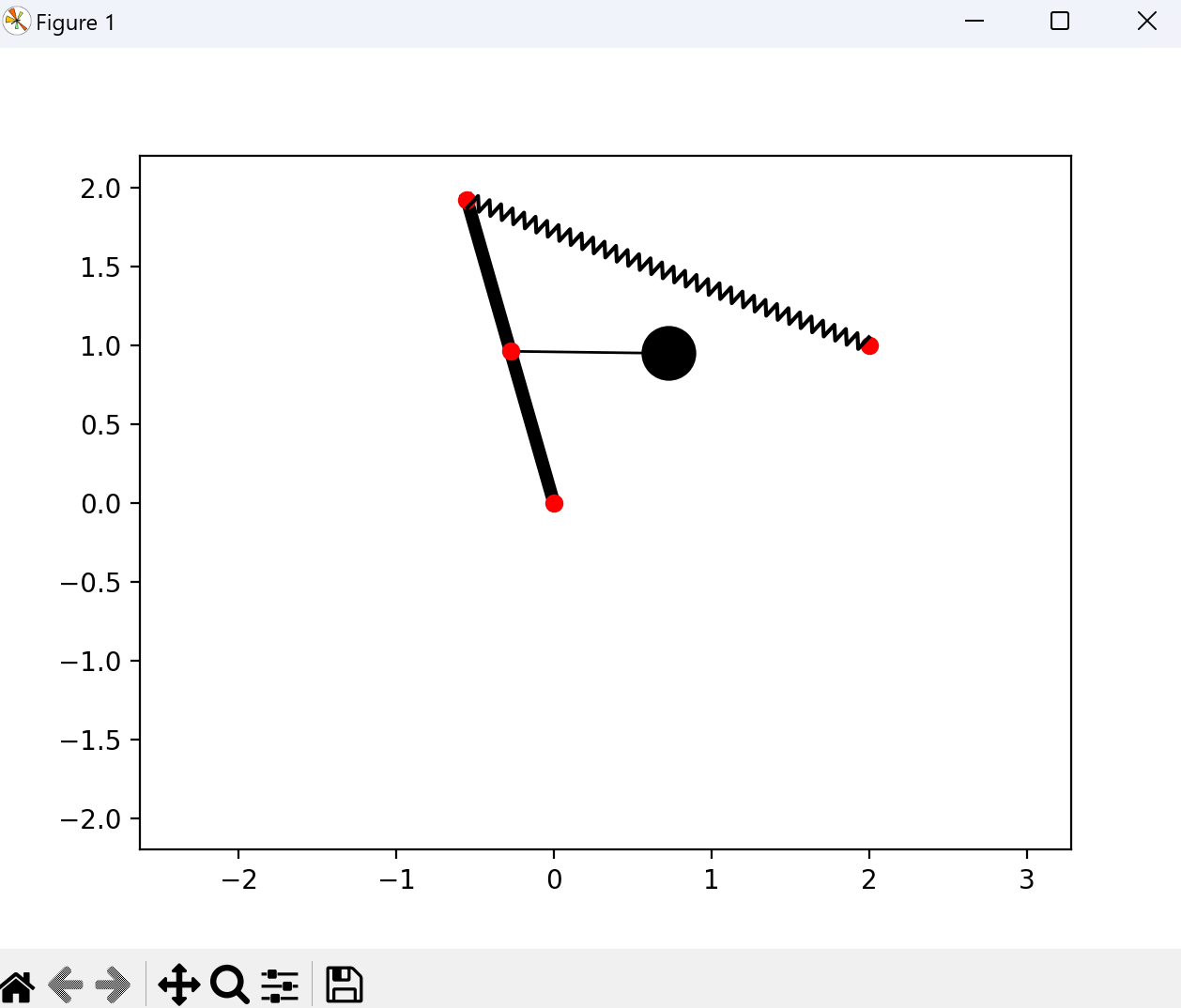
pB.set\_data(Xb[i], Yb[i])

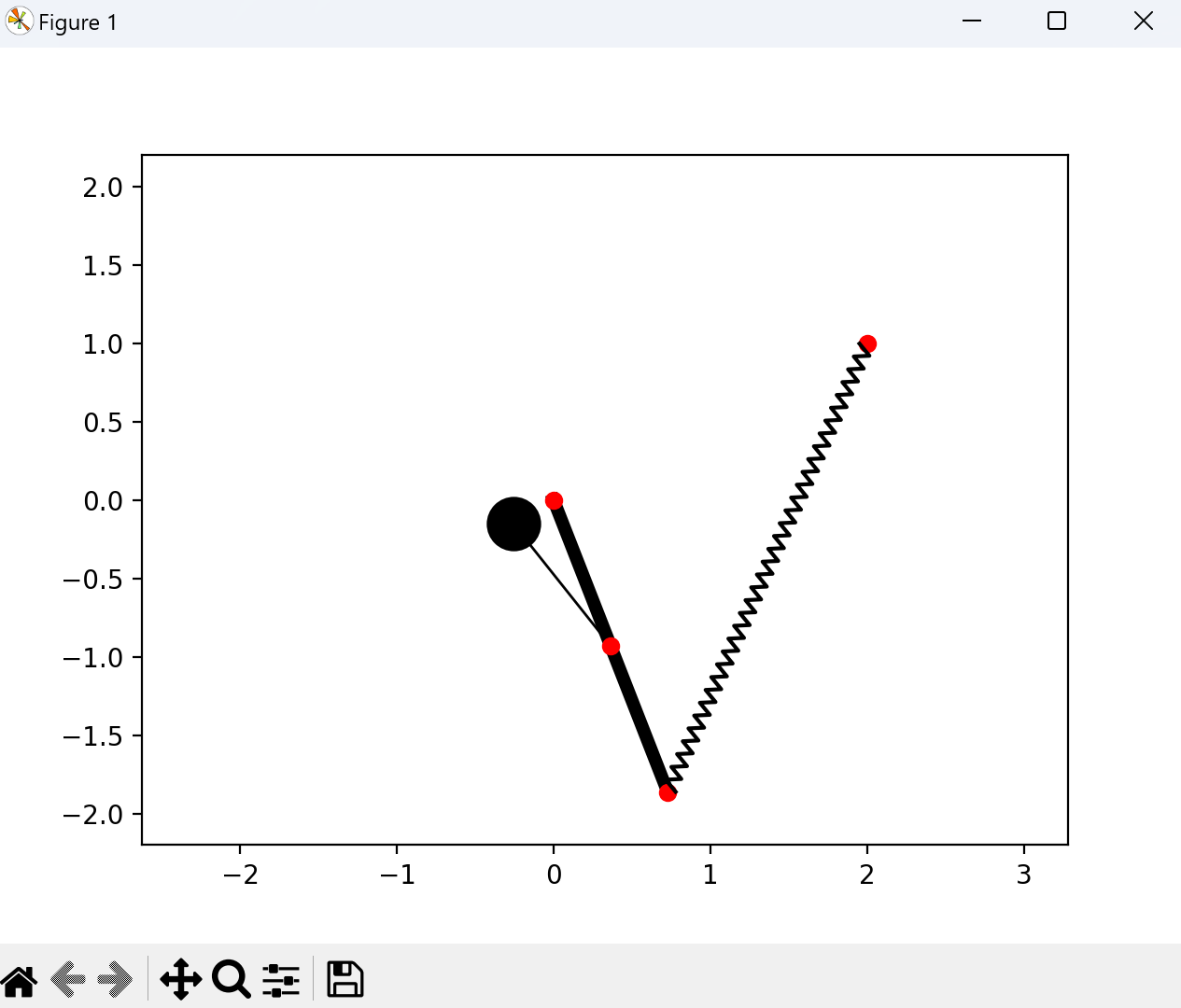
sterjenAB.set\_data([Xa[i], Xb[i]], [Ya[i], Yb[i]])

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

plt.show()

**Работа программы:**





**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.

Начало формы