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 №24**

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

Фокин Л.А.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

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

подпись, дата

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

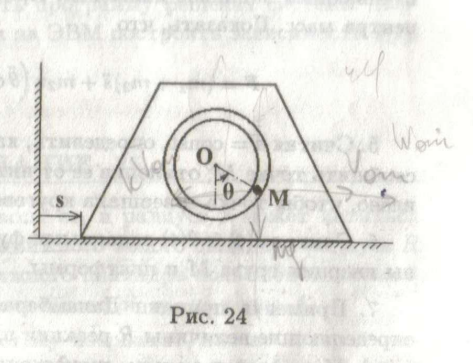
Москва, 2024

**Вариант №24**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**



**Код программы:**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

# Функция для построения трапеции

def trapezoid(*x0*, *y0*, *width*=10, *height*=17.5):

    px = [*x0* - *width*, *x0* - *width* / 3, *x0* + *width* / 3, *x0* + *width*, *x0* - *width*]

    py = [*y0* - *height* / 2, *y0* + *height* / 2, *y0* + *height* / 2, *y0* - *height* / 2, *y0* - *height* / 2]

    return px, py

# Временные параметры

T = np.linspace(0, 20, 1000)

# Движение центра и угла

s = 4 \* np.cos(3 \* T)

phi = 4 \* np.sin(T - 10)

# Координаты центра

x\_center = -s

y\_center = np.ones\_like(T) \* 7.5

# Радиус точки

radius\_length = 5

x\_A = x\_center - radius\_length \* np.sin(phi)

y\_A = y\_center + radius\_length \* np.cos(phi)

# Настройка фигуры и осей

fig, ax = plt.subplots(*figsize*=(10, 6))

ax.axis('equal')

ax.set\_xlim(x\_center.min() - 10, x\_center.max() + 10)

ax.set\_ylim(y\_center.min() - 15, y\_center.max() + 15)

# Отображаемые элементы

trap\_x, trap\_y = trapezoid(x\_center[0], y\_center[0])

trap, = ax.plot(trap\_x, trap\_y, 'r')  # Трапеция

radius\_line, = ax.plot([x\_center[0], x\_A[0]], [y\_center[0], y\_A[0]], 'k')  # Радиус

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

point\_circle, = ax.plot([], [], 'bo')  # Точка на радиусе

orbit\_circle, = ax.plot([], [], *color*="gray", *linestyle*="--")  # Окружность

# Анимация

def update(*frame*):

    # Обновляем трапецию

    trap\_x, trap\_y = trapezoid(x\_center[*frame*], y\_center[*frame*])

    trap.set\_data(trap\_x, trap\_y)

    # Обновляем радиус

    radius\_line.set\_data([x\_center[*frame*], x\_A[*frame*]], [y\_center[*frame*], y\_A[*frame*]])

    # Обновляем точку на радиусе

    point\_circle.set\_data([x\_A[*frame*]], [y\_A[*frame*]])

    # Обновляем окружность

    orbit\_x = x\_center[*frame*] + radius\_length \* np.cos(theta)

    orbit\_y = y\_center[*frame*] + radius\_length \* np.sin(theta)

    orbit\_circle.set\_data(orbit\_x, orbit\_y)

    return trap, radius\_line, point\_circle, orbit\_circle

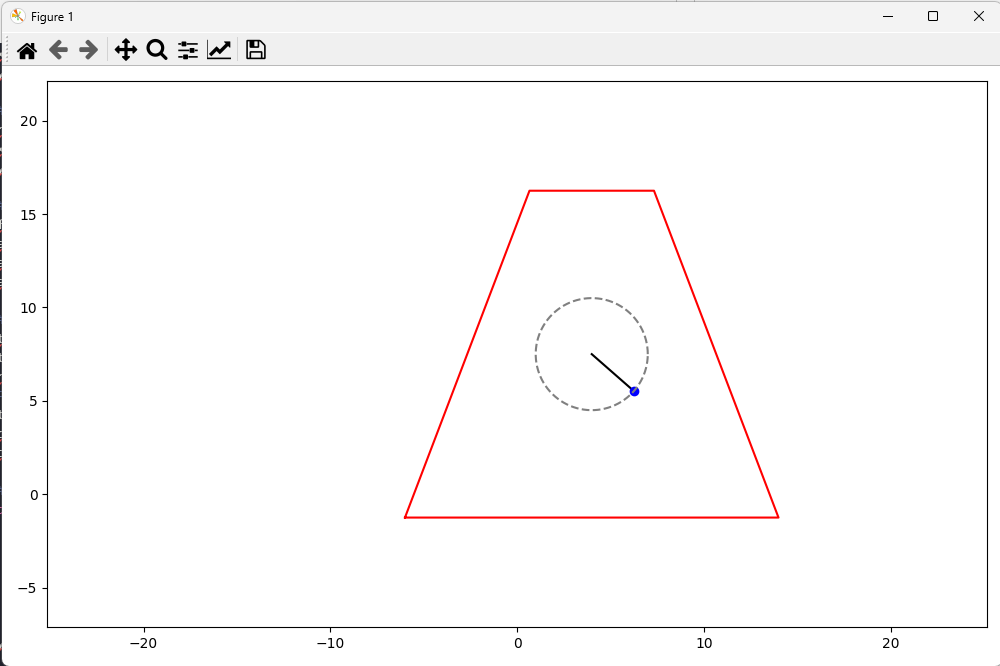
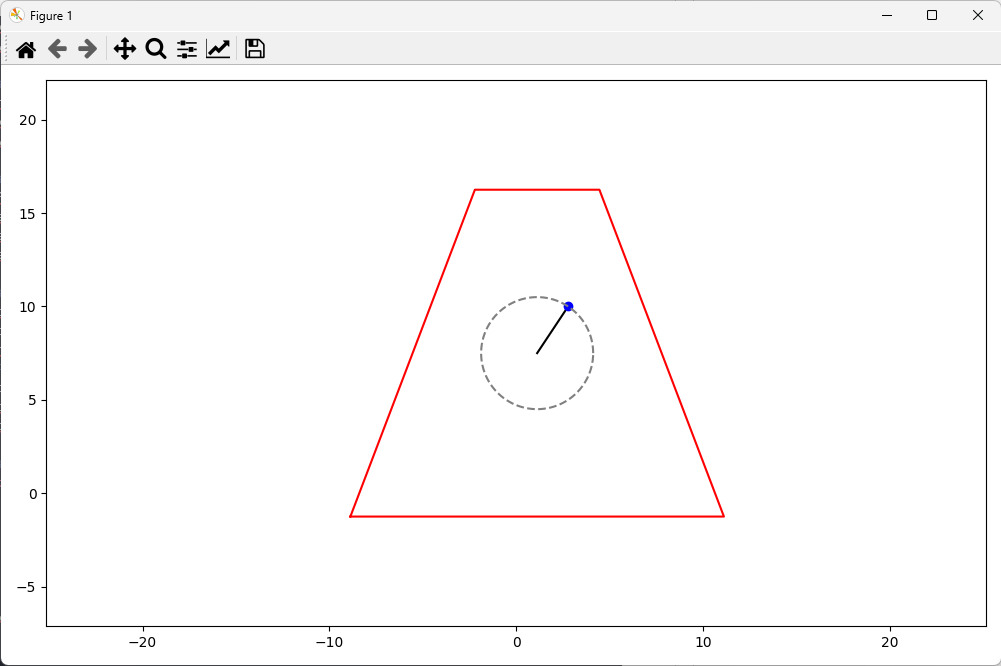
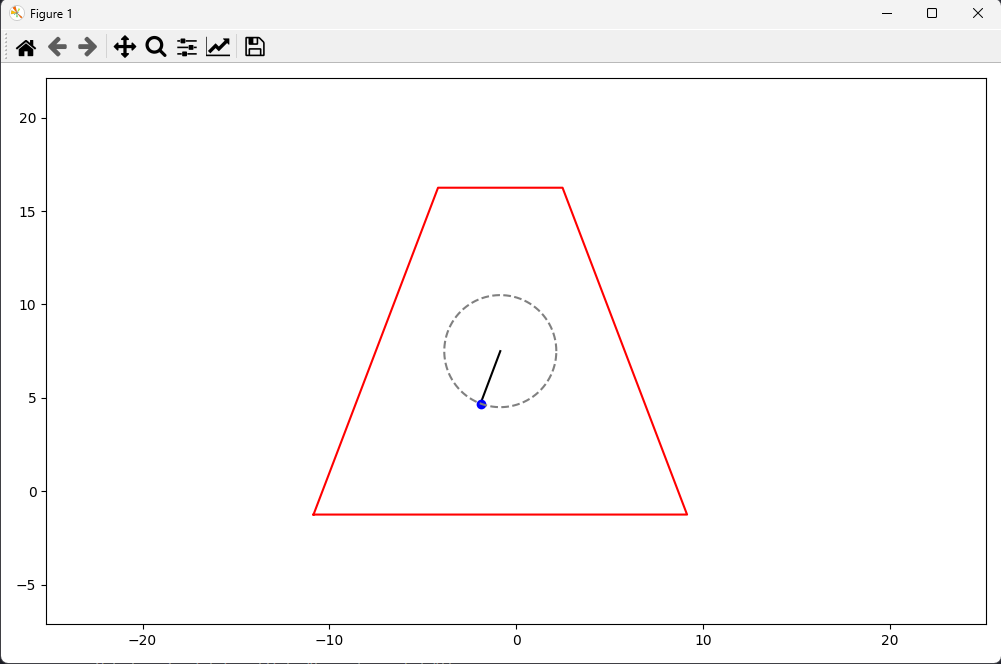
# Создаем анимацию

ani = FuncAnimation(fig, update, *frames*=len(T), *interval*=20, *blit*=True)

plt.tight\_layout()

plt.show()

**Работа программы:**



**Вывод:** в ходе выполнения лабораторной работы была реализована анимация движения механической системы с использованием языка программирования Python. Для создания анимации были использованы библиотеки numpy для работы с массивами и вычислений, а также matplotlib для визуализации данных и создания графиков. В частности, модуль matplotlib.animation был применен для создания анимации, где функция FuncAnimation позволила обновлять координаты элементов системы (трапеции, радиуса и точки на радиусе) в зависимости от времени. Для моделирования движения системы были рассчитаны координаты центра трапеции, угол поворота и положение точки на радиусе. Анимация демонстрирует динамическое изменение положения элементов системы, что позволяет визуально оценить их движение.Начало формы