

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ
«АНИМАЦИЯ ТОЧКИ»
ПО ДИСЦИПЛИНЕ «ОСНОВЫ КОМПЬЮТЕРНОГО
МОДЕЛИРОВАНИЯ МАТЕМАТИЧЕСКИХ СИСТЕМ»
ВАРИАНТ ЗАДАНИЯ № 2

Выполнил(а) студент группы М8О-209Б-23

Борисов Д.С. _____
подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В. _____
подпись, дата

с оценкой _____

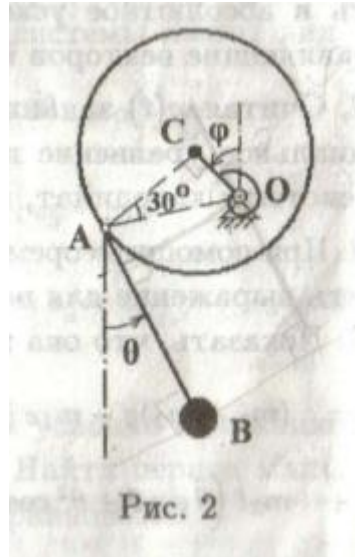
Москва, 2024

Вариант № 2

Задание:

Реализовать анимацию движения механической системы в среде Python.

Механическая система:



Текст программы

```
# Импорт необходимых библиотек
import math
import numpy as np
from matplotlib.animation import FuncAnimation
import matplotlib.pyplot as plt

# Указание параметров моделирования
G = 9.81 # Ускорение свободного падения
L = 8 # Длина стержня
ALPHA = 30 # Угол между AC и AO
R = 7 # радиус диска
C = 10 # Жесткость пружины
E = R / math.sqrt(3) # Расстояние от центра диска до
# горизонтальной оси
TIME = np.linspace(0, 10, 1001) # Полупериод вращения диска
PHI = np.sin(2.1 * TIME) # Угол наклона диска во время вращения
TAU = np.sin(PHI - math.pi / 6) # Начальный угол  $\theta$ , задаётся равным нулю.

# Задание координат точек, где системой отсчета является точка O
X_O = 0
Y_O = 0
X_C = X_O - E * np.sin(PHI)
Y_C = Y_O + E * np.cos(PHI)
X_A = X_C - R * np.sin(math.pi / 2 + PHI)
Y_A = Y_C + R * np.cos(math.pi / 2 + PHI)
X_B = X_A + L * np.sin(TAU)
Y_B = Y_A - L * np.cos(TAU)

def draw_arc(X, Y, radius):
    C_X = [X + radius * math.cos(i / 100) for i in range(0, 628)]
    C_Y = [Y + radius * math.sin(i / 100) for i in range(0, 628)]
```

```

    return C_X, C_Y

# Создаем график и устанавливаем для него параметры
fig = plt.figure(figsize=[13, 9])
ax = fig.add_axes([0.1, 0.1, 0.8, 0.8])
ax.axis('equal')
ax.set(xlim=[-25, 25], ylim=[-25, 25])

# Количество витков или число, определяющее, сколько раз спираль делает
# полный оборот
spiral_branches = 1.1
# Начальный радиус спирали
R1 = 0.2
# Конечный радиус спирали
R2 = 6
# Массив углов для создания спирали (приблизительно равный 2pi)
spiral_angle = np.linspace(0, spiral_branches * 6.28 - PHI[0], 100)

# Вычисление координаты по x для отрисовки спирали Архимеда
spiral_spring_x = -(R1 * spiral_angle * (R2 - R1) / spiral_angle[-1]) *
np.sin(spiral_angle)
# Вычисление координаты по y для отрисовки спирали Архимеда
spiral_spring_y = (R1 * spiral_angle * (R2 - R1) / spiral_angle[-1]) *
np.cos(spiral_angle)

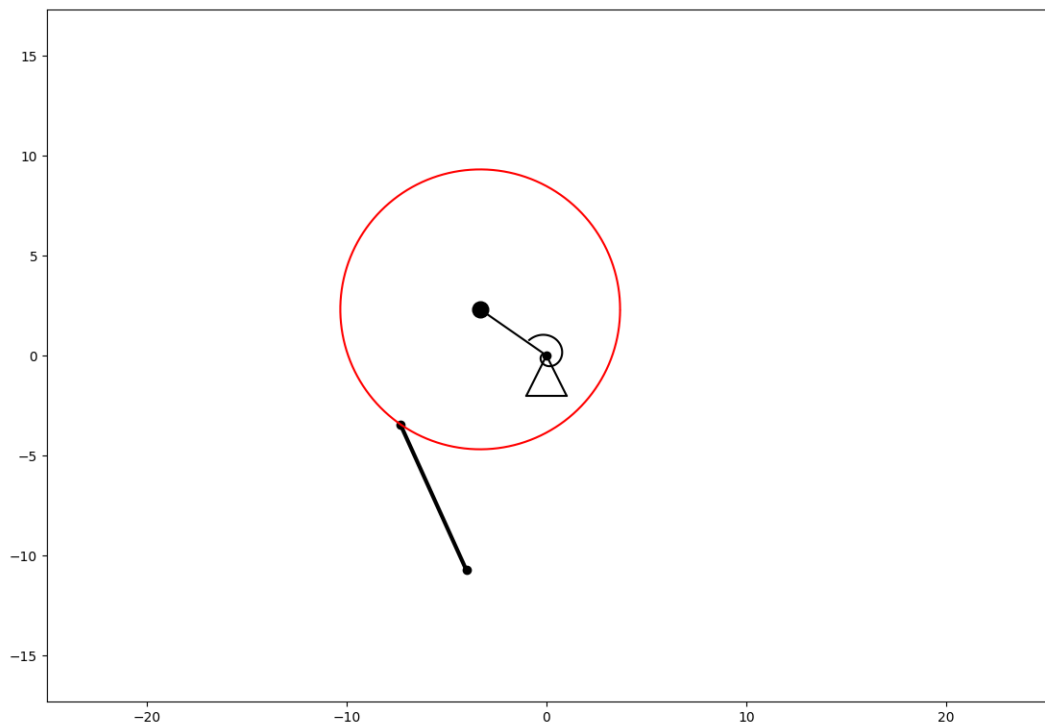
spiral_spring = ax.plot(spiral_spring_x + X_O, spiral_spring_y + Y_O,
color='black')[0]
point_C = ax.plot(X_C[0], Y_C[0], marker='o', markersize=12,
color='black')[0]
point_O = ax.plot(X_O, Y_O, marker='o', color='black')[0]
point_A = ax.plot(X_A, Y_A, marker='o', color='black')[0]
point_B = ax.plot(X_B, Y_B, marker='o', color='black')[0]
line_AB = ax.plot([X_A[0], X_B[0]], [Y_A[0], Y_B[0]], color='black',
linewidth=3)[0]
line_OC = ax.plot([X_O, X_C[0]], [Y_O, Y_C[0]], color='black')[0]
disk_arc, = ax.plot(*draw_arc(X_C[0], Y_C[0], R), 'red')
triangle, = ax.plot([-1, 0, 1], [-2, 0, -2], color='black')
line_tr = ax.plot([-1, 1], [-2, -2], color='black')[0]

# функция для отрисовки текущего состояния системы
def draw(i):
    disk_arc.set_data(*draw_arc(X_C[i], Y_C[i], R))
    point_O.set_data(X_O, Y_O)
    point_C.set_data(X_C[i], Y_C[i])
    point_A.set_data(X_A[i], Y_A[i])
    line_OC.set_data([X_O, X_C[i]], [Y_O, Y_C[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]])
    spiral_angle = np.linspace(0, spiral_branches * 5.6 + PHI[i], 100)
    spiral_spring_x = -(R1 * spiral_angle * (R2 - R1) / spiral_angle[-1]) *
np.sin(spiral_angle)
    spiral_spring_y = (R1 * spiral_angle * (R2 - R1) / spiral_angle[-1]) *
np.cos(spiral_angle)
    spiral_spring.set_data(spiral_spring_x + X_O, spiral_spring_y + Y_O)
    return [disk_arc, point_O, point_C, line_OC, spiral_spring, point_A,
point_B, line_AB]

anim = FuncAnimation(fig, draw, frames=1000, interval=10)
plt.show()

```

Результат работы программы:



Вывод:

В результате лабораторной работы я реализовал анимацию движения механической системы в среде Python, научился строить графики для характеристик движения механической системы.