Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №2**

**по курсу «Теоретическая механика»**

**Анимация системы**

Выполнил студент группы М8О-203Б-20

Сорокин Никита Эдуардович

Преподаватель: Беличенко Михаил Валериевич

Оценка:

Дата: 10.12.2021

Москва, 2021

**Вариант № «31»**

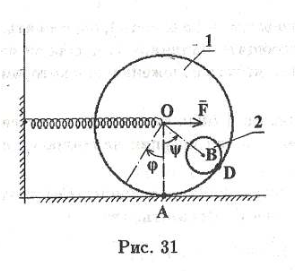
**Задание:**

Анимировать сложное движения твердого тела. Заданы законы

Подвижная система координат — Точка О с осями параллельными х, у

Переносное движение — движение точки О в неподвижной системе координат

Относительное движение — движение точки В (и ее окружности) в подвижной системе координат



**Закон движения точки:**

(законы движения можно устанавливать в программе любыми)

**Текст программы**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

Steps = 1001

t = np.linspace(0, 20, Steps)

phi = 0.5 + np.sin(2 \* t) # здесь можно задавать свои законы движения

psi = t

pi = 3.14

R1 = 5

R1\_array = np.full(Steps, R1, dtype=int)

r\_0 = 5

Point\_O = R1 \* phi + r\_0

theta = np.linspace(0, 2 \* pi, 30)

Circle1\_X = R1 \* np.cos(theta)

Circle1\_Y = R1 \* np.sin(theta)

Ground\_X = [0, 0, 20]

Ground\_Y = [16, 0, 0]

Line\_OH\_X = [0, 0]

Line\_OH\_Y = [0, R1]

Point\_A\_X = Point\_O + R1 \* np.cos(- phi - pi / 2)

Point\_A\_Y = R1\_array + R1 \* np.sin(- phi - pi / 2)

R2 = 1

Point\_B\_X = Point\_O + (R1 - R2) \* np.cos(psi)

Point\_B\_Y = R1\_array + (R1 - R2) \* np.sin(psi)

Circle2\_X = R2 \* np.cos(theta)

Circle2\_Y = R2 \* np.sin(theta)

AnglesCount = 30

MaxWidth = 0.2

Spring\_X = np.zeros(AnglesCount)

Spring\_Y = np.zeros(AnglesCount)

Spring\_X[AnglesCount - 1] = 1

k = AnglesCount - 2

for i in range(AnglesCount - 2):

Spring\_X[i + 1] = (i + 1) / k - 1 / (2 \* k)

Spring\_Y[i + 1] = ((-1) \*\* i) \* MaxWidth

Spring\_Length = Point\_O

Figure = plt.figure(figsize=[15,8])

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

ax.axis("equal")

Drawed\_Ground = ax.plot(Ground\_X, Ground\_Y, color="black", linewidth=3)

Drawed\_Point\_O = ax.plot(Point\_O[0], R1, marker="o")[0]

Drawed\_Point\_H = ax.plot(Point\_O[0], 0, marker="o")[0]

Drawed\_Point\_A = ax.plot(Point\_A\_X[0], Point\_A\_Y[0], marker="o")[0]

Drawed\_Point\_B = ax.plot(Point\_B\_X[0], Point\_B\_Y[0], marker="o")[0]

Drawed\_Circle1 = ax.plot(Point\_O[0] + Circle1\_X, R1 + Circle1\_Y)[0]

Drawed\_Circle2 = ax.plot(Point\_B\_X[0] + Circle2\_X, Point\_B\_Y[0] + Circle2\_Y)[0]

Drawed\_Line\_OH = ax.plot(Line\_OH\_X, Line\_OH\_Y)[0]

Drawed\_Line\_OA = ax.plot([Point\_O[0], Point\_A\_X[0]], [R1\_array[0], Point\_A\_Y[0]])[0]

Drawed\_Line\_OB = ax.plot([Point\_O[0], Point\_B\_X[0]], [R1\_array[0], Point\_B\_Y[0]])[0]

Drawed\_Spring = ax.plot(Spring\_X \* Spring\_Length[0], Spring\_Y + R1)[0]

def Movement(i) :

Drawed\_Point\_O.set\_data(Point\_O[i], R1)

Drawed\_Point\_H.set\_data(Point\_O[i], 0)

Drawed\_Point\_A.set\_data(Point\_A\_X[i], Point\_A\_Y[i])

Drawed\_Point\_B.set\_data(Point\_B\_X[i], Point\_B\_Y[i])

Drawed\_Circle1.set\_data(Point\_O[i] + Circle1\_X, R1 + Circle1\_Y)

Drawed\_Circle2.set\_data(Point\_B\_X[i] + Circle2\_X, Point\_B\_Y[i] + Circle2\_Y)

Drawed\_Line\_OH.set\_data(Line\_OH\_X + Point\_O[i], Line\_OH\_Y)

Drawed\_Line\_OA.set\_data([Point\_O[i], Point\_A\_X[i]], [R1\_array[i], Point\_A\_Y[i]])

Drawed\_Line\_OB.set\_data([Point\_O[i], Point\_B\_X[i]], [R1\_array[i], Point\_B\_Y[i]])

Drawed\_Spring.set\_data(Spring\_X \* Spring\_Length[i], Spring\_Y + R1)

Animation = FuncAnimation(Figure, Movement, frames=Steps, interval=10)

plt.show()

**Результат работы программы:**

