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 $\phi(t), \psi(t)$

Подвижная система координат — Точка О с осями параллельными x, y

Переносное движение — движение точки О в неподвижной системе координат

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

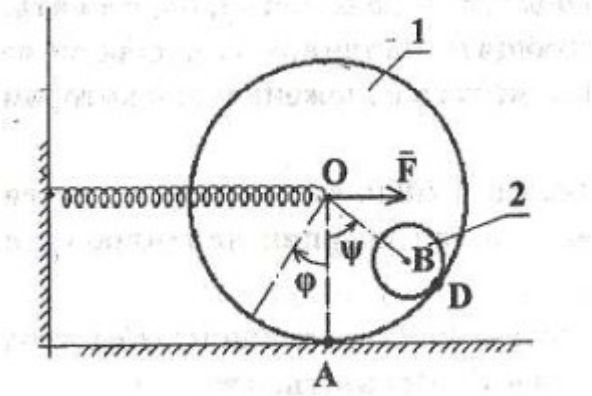

Рис. 31

Закон движения точки:

$\phi(t)=\frac{1}{2}\sin(2t), \qquad \psi(t)=t$

(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: