МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский Авиационный Институт» (Национальный Исследовательский Университет)

Институт: №8 «Компьютерные науки и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Лабораторная работа №2

по курсу «Теоретическая механика»

Ⅲ семестр

«Кинематика системы»

|  |  |
| --- | --- |
| Группа | М8О-206Б-21 |
| Студент | Чистяков К.С. |
| Преподаватель | Чекина Е.А. |
| Оценка |  |
| Дата |  |

Москва, 2023

Оглавление

[*Задание:* 3](#_Toc126762743)

[*Механическая система*: 3](#_Toc126762744)

[*Текст программы:* 3](#_Toc126762745)

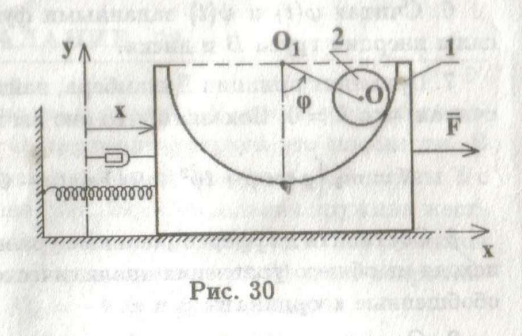
[*Результат работы программы:* 6](#_Toc126762746)

Вариант 30

# *Задание:*

Реализовать анимацию движения механической системы, считая и заданными функциями.

# *Механическая система*:

******

# *Текст программы:*

# Подключаем библиотеки

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

# Функция для задания шарика

def circle(x, y, r):

cx = [x + r \* np.sin(i / 100) for i in range(0, 628)]

cy = [y + r \* np.cos(i / 100) for i in range(0, 628)]

return (cx, cy)

# Функция для задания выемки

def recess(R):

rx = [2 + R \* np.sin(i / 100) for i in range(160, 470)]

ry = [3 + R \* np.cos(i / 100) for i in range(160, 470)]

return (rx, ry)

# Определение t как символа (это будет независимая переменная)

t = sp.Symbol('t')

# Задание закона движения s и phi

s = 2 \* sp.sin(1.5 \* t) + 2

phi = sp.cos(1.5 \* t) + sp.pi

# Радиус внутренней выемки

R = 1.5

# Задание параметров шарика

r = 0.5 # Радиус шарика

Cx = (R - r) \* sp.sin(phi) + 2

Cy = (R - r) \* sp.cos(phi) + 3

# Нахождение модуля скорости и ускорения

Vr = sp.diff(phi, t) \* (R-r) # Относительная скорость

Ve = sp.diff(s, t) # Абсолютная скорость

VmodC = sp.sqrt(Ve\*\*2 + Vr\*\*2 + 2 \* Vr \* Ve \* sp.sin(phi)) # Модуль скорости

We = sp.diff(Ve,t) # Абсолютное ускорение

Wrn = phi\*\*2 \* (R-r) # Wrn - одно из слагаемых относительного ускорения

Wrtau = sp.diff(sp.diff(phi, t), t) # Wrtau - одно из слагаемых относительного ускорения

Wx = - Wrn \* sp.sin(phi) - We - Wrtau \* sp.cos(phi) # Ускорение разложенное по x

Wy = Wrn \* sp.cos(phi) - Wrtau \* sp.sin(phi) # Ускорение разложенное по y

WmodC = sp.sqrt(Wx\*\*2 + Wy\*\*2) # Модуль ускорения

# Задание пружины

n = 20 # Количество витков

k = 1 / (n - 2)

width = 0.3 # Ширина пружины

# Создание пустых массивов, заполненных нулями

Sx = np.zeros(n)

Sy = np.zeros(n)

Sx[0] = 0

Sx[n-1] = 1

Sy[0] = -0.3

Sy[n-1] = 0

for i in range(n-2):

Sx[i+1] = k \* (i + 1)

Sy[i+1] = width \* (-1)\*\*i

# Модуль растяжения пружины

tension = abs(s)

# Основание вала

O1 = np.array([0, 0, 4, 4])

O2 = np.array([3, 0, 0, 3])

# Построение функций

T = np.linspace(0, 10, 500)

tension\_def = sp.lambdify(t, tension)

Cx\_def = sp.lambdify(t, Cx)

Cy\_def = sp.lambdify(t, Cy)

VmodB\_def = sp.lambdify(t, VmodC)

WmodB\_def = sp.lambdify(t, WmodC)

Tension = tension\_def(T)

C\_x = Cx\_def(T)

C\_y = Cy\_def(T)

V\_C = VmodB\_def(T)

W\_C = WmodB\_def(T)

# Рисование

fig = plt.figure(figsize=[16, 7])

# Первая часть

ax1 = fig.add\_subplot(1, 2, 1)

ax1.set(xlim=[-1, 9], ylim=[-1, 9])

ax1.set\_xlabel('Ось x')

ax1.set\_ylabel('Ось y')

OsY = ax1.plot([0, 0], [0, 5], 'black')[0]

OsX = ax1.plot([0, 8.5], [0, 0], 'black')[0]

Box = ax1.plot(O1 + Tension[0], O2, 'brown')[0]

Recess = ax1.plot(recess(R)[0] + Tension[0], recess(R)[1], 'brown')[0]

line1 = ax1.plot([Tension[0] + 0.5, Tension[0]], [3, 3], 'brown')[0]

line2 = ax1.plot([Tension[0] + 3.5, Tension[0] + 4], [3, 3], 'brown')[0]

Spring = ax1.plot(Sx \* Tension[0], Sy + 1.5, 'red')[0]

Cirle, = ax1.plot(circle(C\_x[0], C\_y[0], r)[0] + Tension[0], circle(C\_x[0], C\_y[0], r)[1], 'green')

# Вторая часть

ax2 = fig.add\_subplot(2, 2, 2)

ax2.set(xlim=[0, 10], ylim=[V\_C.min(), V\_C.max()])

TVx = [T[0]]

TVy = [V\_C[0]]

TV, = ax2.plot(TVx, TVy, '-')

plt.title('Скорость')

ax2.set\_xlabel('T')

ax2.set\_ylabel('V')

ax3 = fig.add\_subplot(2, 2, 4)

ax3.set(xlim=[0,10], ylim=[W\_C.min(), W\_C.max()])

TWx = [T[0]]

TWy = [W\_C[0]]

TW, = ax3.plot(TWx, TWy, '-')

plt.title('Ускорение')

ax3.set\_xlabel('T')

ax3.set\_ylabel('W')

plt.subplots\_adjust(wspace=0.3, hspace=0.4)

# Анимация

def Animation(i):

Box.set\_data(O1 + Tension[i], O2)

Recess.set\_data(recess(R)[0] + Tension[i], recess(R)[1])

line1.set\_data([Tension[i] + 0.5, Tension[i]], [3, 3])

line2.set\_data([Tension[i] + 3.5, Tension[i] + 4], [3, 3])

Spring.set\_data(Sx \* Tension[i], Sy + 1.5)

Cirle.set\_data(circle(C\_x[i], C\_y[i], r)[0] + Tension[i], circle(C\_x[i], C\_y[i], r)[1])

TVx.append(T[i])

TVy.append(V\_C[i])

TWx.append(T[i])

TWy.append(W\_C[i])

TV.set\_data(TVx, TVy)

TW.set\_data(TWx, TWy)

if i == 500-1:

TVx.clear()

TVy.clear()

TWx.clear()

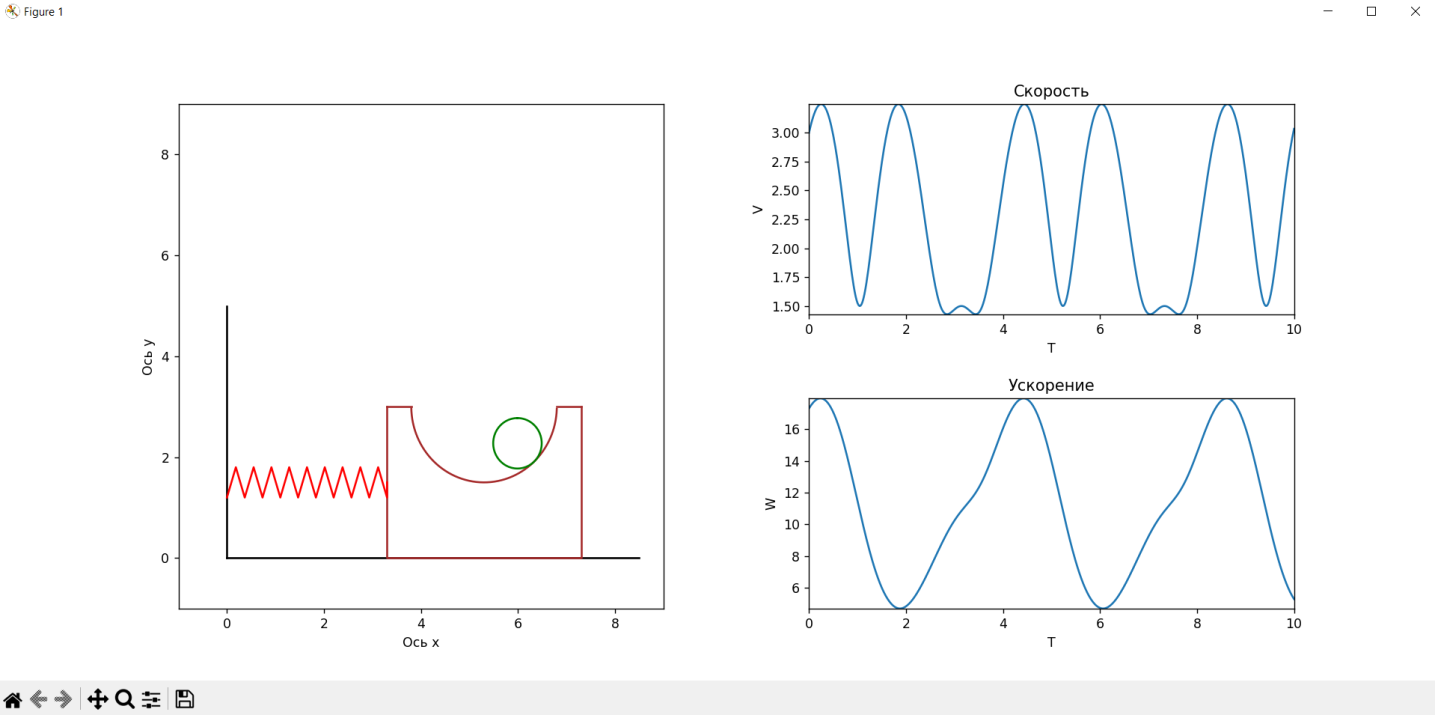
TWy.clear()

return Box, Recess, line1, line2, Spring, Cirle, TV, TW

plot = FuncAnimation(fig, Animation, frames=1000, interval=0.001)

plt.show()

# *Результат работы программы:*

******