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-207Б-20

Михеева Кристина Олеговна

Преподаватель: Чекина Евгения Алексеевна

Оценка:

Дата: 30.11.21

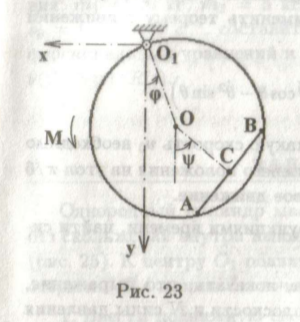
Москва, 2021

**Вариант №«23»**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**

****

**Текст программы:**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

PI = math.pi

t = sp.Symbol('t')

rCircle = 0.5

l = 0.25

rodLength = OA = 2 \* l

t0 = phi0 = wO0 = wC0 = 0 # initial data

psi0 = PI/6

OC = math.sqrt(3) \* l # distance between point O and point C

# parameters of motion

phi = sp.sin(t)

psi = sp.sin(-t + 0.55)

xpO = -rCircle \* sp.sin(phi) # X component of point O

ypO = rCircle \* sp.cos(phi) # Y component of point O

vpO = sp.diff(phi, t) \* rCircle # speed of point O

xpC = xpO + OC \* sp.sin(psi)

ypC = ypO + OC \* sp.cos(psi)

xpA = xpC + l \* sp.cos(psi)

ypA = ypC - l \* sp.sin(psi)

xpB = xpC - l \* sp.cos(psi)

ypB = ypC + l \* sp.sin(psi)

#Point O

vxpO = -vpO \* sp.cos(phi)

vypO = -vpO \* sp.sin(phi)

vo = (vxpO\*\*2 + vypO\*\*2)\*\*0.5

wo = ((sp.diff(vxpO, t)\*\*2 + sp.diff(vypO, t)\*\*2)\*\*0.5)

#Point C

vxRelativepC = sp.diff(psi, t) \* OC \* sp.sin(psi)

vyRelativepC = sp.diff(psi, t) \* OC \* sp.cos(psi)

vxAbspC = vxRelativepC + vxpO

vyAbspC = vyRelativepC + vypO

vc = (vxAbspC\*\*2 + vyAbspC\*\*2)\*\*0.5

wc = (sp.diff(vxAbspC, t)\*\*2 + sp.diff(vyAbspC, t)\*\*2)\*\*0.5

# constructing corresponding arrays

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

Phi = np.zeros\_like(T)

Psi = np.zeros\_like(T)

XpO = np.zeros\_like(T)

YpO = np.zeros\_like(T)

VxpO = np.zeros\_like(T)

VypO = np.zeros\_like(T)

XpC = np.zeros\_like(T)

YpC = np.zeros\_like(T)

XpA = np.zeros\_like(T)

YpA = np.zeros\_like(T)

XpB = np.zeros\_like(T)

YpB = np.zeros\_like(T)

VxpC = np.zeros\_like(T)

VypC = np.zeros\_like(T)

VO = np.zeros\_like(T)

VC = np.zeros\_like(T)

WO = np.zeros\_like(T)

WC = np.zeros\_like(T)

# filling arrays with corresponding values

for i in np.arange(len(T)):

Phi[i] = sp.Subs(phi, t, T[i])

Psi[i] = sp.Subs(psi, t, T[i])

XpO[i] = sp.Subs(xpO, t, T[i])

YpO[i] = sp.Subs(ypO, t, T[i])

VxpO[i] = sp.Subs(vxpO, t, T[i])

VypO[i] = sp.Subs(vypO, t, T[i])

XpC[i] = sp.Subs(xpC, t, T[i])

YpC[i] = sp.Subs(ypC, t, T[i])

XpA[i] = sp.Subs(xpA, t, T[i])

YpA[i] = sp.Subs(ypA, t, T[i])

XpB[i] = sp.Subs(xpB, t, T[i])

YpB[i] = sp.Subs(ypB, t, T[i])

VxpC[i] = sp.Subs(vxAbspC, t, T[i])

VypC[i] = sp.Subs(vyAbspC, t, T[i])

VO[i] = sp.Subs(vo, t, T[i])

VC[i] = sp.Subs(vc, t, T[i])

WO[i] = sp.Subs(wo, t, T[i])

WC[i] = sp.Subs(wc, t, T[i])

# here we start to plot

fig = plt.figure(figsize=(17, 10))

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

ax1.axis('equal')

ax1.set(xlim=[-2, 2], ylim=[-2, 3])

ax1.invert\_xaxis()

ax1.invert\_yaxis()

ax1.plot(0, 0, marker='o', color='blue') # point O1

PCIRCLE, = ax1.plot(XpO[0], YpO[0], 'b', marker='o', markersize=3)

PpA, = ax1.plot(XpA[0], YpA[0], 'g', marker='o', markersize=2)

PpC, = ax1.plot(XpC[0], YpC[0], 'r', marker='o', markersize=3)

PpB, = ax1.plot(XpB[0], YpB[0], 'black', marker='o', markersize=2)

Rod, = ax1.plot([XpA[0], XpB[0]], [YpA[0], YpB[0]], 'r')

# plotting initial positions

ax2 = fig.add\_subplot(4, 2, 2)

ax2.plot(T, VO)

plt.title('V of the Circle O')

plt.xlabel('t values')

plt.ylabel('Vo values')

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

ax3.plot(T, WO)

plt.title('W of the Circle O')

plt.xlabel('t values')

plt.ylabel('Wo values')

ax4 = fig.add\_subplot(4, 2, 6)

ax4.plot(T, VC)

plt.title('V of the Point C')

plt.xlabel('t values')

plt.ylabel('Vc values')

ax5 = fig.add\_subplot(4, 2, 8)

ax5.plot(T, WC)

plt.title('W of the Point C')

plt.xlabel('t values')

plt.ylabel('Wc values')

plt.subplots\_adjust(wspace=0.3, hspace=0.7)

# function for recounting the positions

def anima(i):

CIRCLE = plt.Circle((XpO[i], YpO[i]), rCircle, color='b', fill=False)

ax1.add\_artist(CIRCLE)

PCIRCLE.set\_data(XpO[i], YpO[i])

PpC.set\_data(XpC[i], YpC[i])

PpA.set\_data(XpA[i], YpA[i])

Rod.set\_data([XpA[i], XpB[i]], [YpA[i], YpB[i]])

PpB.set\_data(XpB[i], YpB[i])

return PCIRCLE, CIRCLE, PpC, Rod, PpA, PpB,

# animation function

anim = FuncAnimation(fig, anima, frames=1000, interval=10, blit=True)

plt.show()

**Результат работы:**

|  |
| --- |
|  |