Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №1**

**по курсу «Теоретическая механика»**

**Анимация точки**

Выполнил студент группы М8О-207Б-21

Дубровин Дмитрий Константинович

Преподаватель: Беличенко Михаил Валериевич

Оценка: -

Дата: 29.12.2022

Москва, 2022

**Вариант № 8**

**Задание:**

Построить заданную траекторию и анимацию движения точки, а также отобразить стрелки скорости и ускорения. Построить радиус кривизны траектории.

**Закон движения точки:**

Var 8: r(t) = 2+sin(8t), phi(t) = t+0.2\*cos(6t)

**Текст программы**

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

t = sp.Symbol('t')

T = np.linspace(1, 15, 1000)

# Var 8: r(t) = 2+sin(8t), phi(t) = t+0.2\*cos(6t)

r = 2 + sp.sin(8 \* t)

phi = t + 0.2 \* sp.cos(6 \* t)

# переход в декартовы координаты

x = r \* sp.cos(phi)

y = r \* sp.sin(phi)

Vx, Vy = sp.diff(x, t), sp.diff(y, t) # скорость - производная от координат

Wx, Wy = sp.diff(Vx, t), sp.diff(Vy, t) # ускорение - производная от скорости

V, W = sp.sqrt(Vx \*\* 2 + Vy \*\* 2), sp.sqrt(Wx \*\* 2 + Wy \*\* 2) # квадратный корень из составляющих по координатам

F\_func = [sp.lambdify(t, i) for i in [x, y, Vx, Vy, Wx, Wy]] # генератор лямбда функций

[X, Y, Vx, Vy, Wx, Wy] = [func(T) for func in F\_func] # подстановка интервала

fig = plt.figure() # генерация окна

ax = fig.add\_subplot(1, 1, 1) # заголовки

ax.axis('equal'), ax.set\_title("Модель движения точки"), ax.set\_xlabel('x'), ax.set\_ylabel('y'), ax.plot(X, Y), ax.set(

xlim=[-5, 5], ylim=[-5, 5])

P = ax.plot(X[0], Y[0], marker='\*')[0]

k, kf = True, 0.1 # коэффициенты для корректного отображения

def anima(i):

P.set\_data(X[i], Y[i])

VLine = ax.arrow(X[i], Y[i], kf \* Vx[i], kf \* Vy[i], width=0.02, color='darkred',

label='- скорость') # Вектор скорости

WLine = ax.arrow(X[i], Y[i], kf \* Wx[i], kf \* Vy[i], width=0.02, color='indianred',

label='- ускорение') # Вектор ускорения

CVector = ax.arrow(X[i], Y[i], - kf \* ((Vy[i] \* (Vx[i] \*\* 2 + Vy[i] \*\* 2)) / (Vx[i] \* Wy[i] - Wx[i] \* Vy[i])),

kf \* ((Vx[i] \* (Vx[i] \*\* 2 + Vy[i] \*\* 2)) / (Vx[i] \* Wy[i] - Wx[i] \* Vy[i])),

width=0.03, color="r", label='- кривизна') # Вектор кривизны

global k

if k: # легенда для наглядности

ax.legend(ncol=2, # количество столбцов

facecolor='oldlace', # цвет области

edgecolor='r', # цвет крайней линии

title='обозначение векторов', # заголовок

title\_fontsize='10') # размер заголовка

k = False

return P, VLine, WLine, CVector

anim = FuncAnimation(fig, anima, frames=len(T), interval=20, blit=True)

plt.show()

**Результат работы программы:**

|  |  |
| --- | --- |
|  |  |
|  |  |