

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Институт №8 "Информационные технологий и прикладная математика"
Кафедра 806 "Вычислительная математика и программирование"

Лабораторная работа №1
по курсу "Теоретическая механика и основы компьютерного моделирования"
3 семестр

Студент: Леухин М. В.
Группа: М8О-206Б-20
Преподаватель: Сухов Е. А.
Подпись: _____

Москва, 2021

Содержание

1	Теоретическая часть	3
2	Листинг программы	3
3	Результат работы программы	6
3.1	$v_0 = 5, R = 10$	6
3.2	$v_0 = 25, R = 10$	7

1 Теоретическая часть

В данной лабораторной работе необходимо вывести уравнения движения материальной точки, которая находится на движущейся со скоростью v_0 окружности радиуса R . Движение такой точки можно представить как сумму движений — движение центра окружности со скоростью v_0 и движение точки по окружности, которое вызвано тем, что колесо при движении поворачивается. В таком случае имеем:

$$\begin{aligned}c_x(t) &= v_0 t \\m_x(t) &= -R \cos\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right) \\x(t) &= c_x(t) + m_x(t) = v_0 t + R \cos\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right)\end{aligned}$$

$$\begin{aligned}c_y(t) &= R = \text{const} \\m_y(t) &= R \sin\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right) \\y(t) &= c_y(t) + m_y(t) = R + R \sin\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right)\end{aligned}$$

Выведем далее формулы скорости и ускорения:

$$\begin{aligned}v_x(t) = \dot{x}(t) &= v_0 + v_0 \sin\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right) \\v_y(t) = \dot{y}(t) &= v_0 \cos\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right) \\w_x(t) = \dot{v}_x(t) &= \frac{v_0^2}{R} \cos\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right) \\w_y(t) = \dot{v}_y(t) &= -\frac{v_0^2}{R} \sin\left(\frac{v_0 t}{R} - \frac{\pi}{2}\right)\end{aligned}$$

2 Листинг программы

```
1 import numpy as np
2 import sympy as sp
3 import math
4 import matplotlib.pyplot as plt
5 from matplotlib.animation import FuncAnimation
6
7 v0 = 5
8 R = 10
9
10 t = sp.Symbol('t')
11 x = v0 * t - R * sp.cos((v0 * t) / R - np.pi / 2)
12 y = R + R * sp.sin((v0 * t) / R - np.pi / 2)
13 vx = sp.diff(x, t)
14 vy = sp.diff(y, t)
15 wx = sp.diff(vx, t)
16 wy = sp.diff(vy, t)
17 cx = v0 * t
18
19 tn = np.linspace(0, 20, 1000)
20 xn = np.zeros_like(tn)
21 yn = np.zeros_like(tn)
22 vxn = np.zeros_like(tn)
23 vyn = np.zeros_like(tn)
24 wxn = np.zeros_like(tn)
```

```

25 wyn = np.zeros_like(tn)
26 cxn = np.zeros_like(tn)
27
28 for i in range(len(tn)):
29     xn[i] = sp.Subs(x, t, tn[i])
30     yn[i] = sp.Subs(y, t, tn[i])
31     vxn[i] = sp.Subs(vx, t, tn[i])
32     vyn[i] = sp.Subs(vy, t, tn[i])
33     wxn[i] = sp.Subs(wx, t, tn[i])
34     wyn[i] = sp.Subs(wy, t, tn[i])
35     cxn[i] = sp.Subs(cx, t, tn[i])
36
37 fig = plt.figure()
38 ax = fig.add_subplot(1, 1, 1)
39 ax.axis('equal')
40 ax.plot(xn, yn, linestyle="—", color="gray")
41 ax.axhline(y=0, color='gray')
42 ax.axvline(x=0, color='gray')
43
44
45 def update(i):
46     # точка
47     point.set_data(xn[i], yn[i])
48     # окружность
49     global circle
50     circle.remove()
51     circle = plt.Circle((cxn[i], R), R, edgecolor="#ff924a",
52                          facecolor="#fad1b6")
53     ax.add_patch(circle)
54     # линиявектораскорости
55     velocity.set_data([xn[i], xn[i] + vxn[i]], [yn[i], yn[i] + vyn[i]])
56     # координатыконцавектораскорости ">"
57     varrow_x, varrow_y = Rot2D(varrow_x0, varrow_y0,
58                                math.atan2(vyn[i], vxn[i]))
59     varrow.set_data(varrow_x + xn[i] + vxn[i], varrow_y + yn[i] +
60                    vyn[i])
61     # линиявектораускорения
62     acceleration.set_data([xn[i], xn[i] + wxn[i]], [yn[i], yn[i] +
63                    wyn[i]])
64     # координатыконцавектораускорения ">"
65     warrow_x, warrow_y = Rot2D(warrow_x0, warrow_y0,
66                                math.atan2(wyn[i], wxn[i]))
67     warrow.set_data(warrow_x + xn[i] + wxn[i], warrow_y + yn[i] +
68                    wyn[i])
69     # центростремительности
70     circle_center.set_data(cxn[i], R)
71     # всёпрямизну
72     curvature = (vxn[i] ** 2 + vyn[i] ** 2) / math.sqrt(wxn[i] ** 2 +
73                    wyn[i] ** 2)
74     k = curvature / math.sqrt(vxn[i] ** 2 + vyn[i] ** 2)
75     curvature_line.set_data([xn[i], xn[i] + vyn[i] * k], [yn[i], yn[i]
76                    - vxn[i] * k])
77     curvature_center.set_data(xn[i] + vyn[i] * k, yn[i] - vxn[i] * k)

```

```

70     global curvature_circle
71     curvature_circle.remove()
72     curvature_circle = plt.Circle((xn[i] + vyn[i] * k, yn[i] - vxn[i]
73         * k), curvature, alpha=0.1)
74     ax.add_patch(curvature_circle)
75
76     return point, velocity, varrow, circle, acceleration, warrow,
77         circle_center, curvature_line, curvature_center,
78         curvature_circle
79
80 def Rot2D(X, Y, Alpha):
81     RX = X * np.cos(Alpha) - Y * np.sin(Alpha)
82     RY = X * np.sin(Alpha) + Y * np.cos(Alpha)
83     return RX, RY
84
85 # точка
86 point = ax.plot(xn[0], yn[0], marker=".", color="black")[0]
87 # окружность
88 circle = plt.Circle((0, R), R, edgecolor="#ff924a",
89     facecolor="#fad1b6")
90 ax.add_patch(circle)
91 # линия вектора скорости
92 velocity = ax.plot([xn[0], xn[0] + vxn[0]], [yn[0], yn[0] + vyn[0]],
93     color="#2ec4f2", label="$v$")[0]
94 # координаты конца вектора скорости
95 varrow_x0 = np.array([-0.2 * R, 0, -0.2 * R])
96 varrow_y0 = np.array([0.1 * R, 0, -0.1 * R])
97 varrow_x, varrow_y = Rot2D(varrow_x0, varrow_y0, math.atan2(vyn[0],
98     vxn[0]))
99 varrow = ax.plot(varrow_x + xn[0] + vxn[0], varrow_y + yn[0] + vyn[0],
100     color="#2ec4f2")[0]
101 # линия ускорения
102 acceleration = ax.plot([xn[0], xn[0] + wxn[0]], [yn[0], yn[0] +
103     wyn[0]], color="#bd0404", label="$w$")[0]
104 # координаты конца вектора ускорения
105 warrow_x0 = np.array([-0.2 * R, 0, -0.2 * R])
106 warrow_y0 = np.array([0.1 * R, 0, -0.1 * R])
107 warrow_x, warrow_y = Rot2D(warrow_x0, warrow_y0, math.atan2(wyn[0],
108     wxn[0]))
109 warrow = ax.plot(warrow_x + xn[0] + wxn[0], warrow_y + yn[0] + wyn[0],
110     color="#bd0404")[0]
111 # центр окружности
112 circle_center = ax.plot(xn[0], R, marker=".", color="#ff924a")[0]
113 # всё про кривизну
114 curvature = (vxn[0] ** 2 + vyn[0] ** 2) / math.sqrt(wxn[0] ** 2 +
115     wyn[0] ** 2)
116 k = curvature / math.sqrt(vxn[0] ** 2 + vyn[0] ** 2)
117 curvature_line = ax.plot([xn[0], xn[0] + vyn[0] * k], [yn[0], yn[0] -
118     vyn[0] * k], linestyle="—", color="#157c9e", label="$\\rho$")[0]
119 curvature_center = ax.plot(xn[0] + vyn[0] * k, yn[0] - vyn[0] * k,
120     marker='.', color="#157c9e")[0]

```

```

110 | curvature_circle = plt.Circle((xn[0] + wxn[0] * k, yn[0] + wyn[0] *
      | k), curvature, alpha=0.1)
111 | ax.add_patch(curvature_circle)
112 |
113 | a = FuncAnimation(fig, update, frames=len(tn), interval=10)
114 |
115 | ax.legend()
116 | plt.show()

```

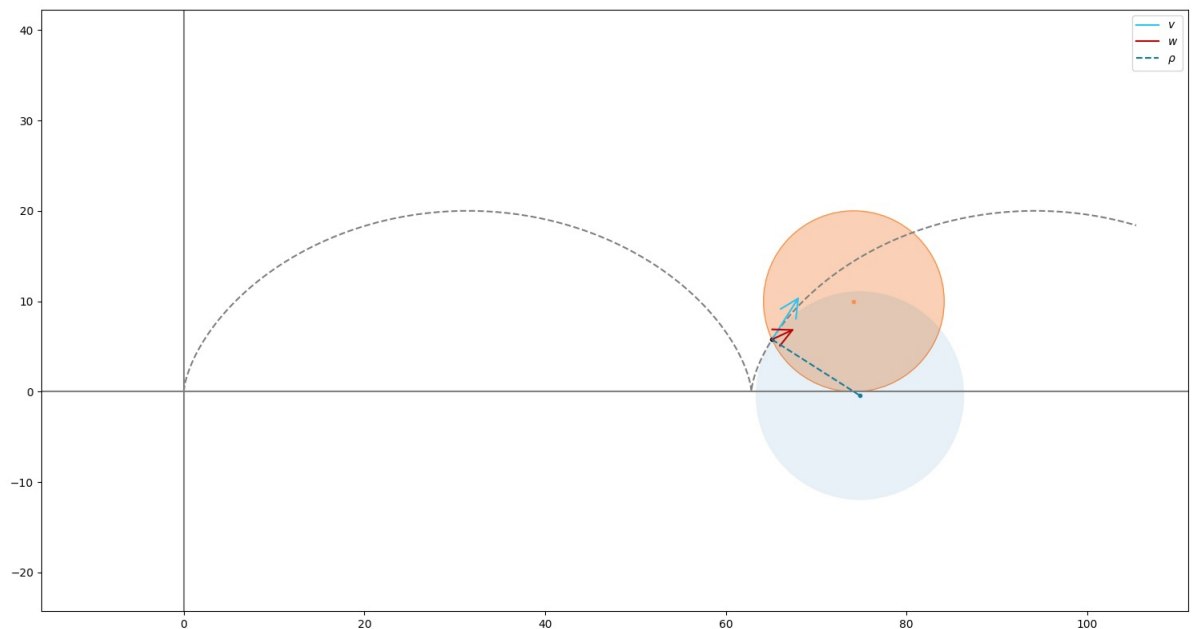
3 Результат работы программы

3.1 $v_0 = 5, R = 10$

```

x(t) = 5*t - 10*cos(t/2 - 1.5707963267949)
y(t) = 10*sin(t/2 - 1.5707963267949) + 10
vx(t) = 5*sin(t/2 - 1.5707963267949) + 5
vy(t) = 5*cos(t/2 - 1.5707963267949)
wx(t) = 5*cos(t/2 - 1.5707963267949)/2
wy(t) = -5*sin(t/2 - 1.5707963267949)/2

```



3.2 $v_0 = 25, R = 10$

```
x(t) = 25*t - 10*cos(5*t/2 - 1.5707963267949)
y(t) = 10*sin(5*t/2 - 1.5707963267949) + 10
vx(t) = 25*sin(5*t/2 - 1.5707963267949) + 25
vy(t) = 25*cos(5*t/2 - 1.5707963267949)
wx(t) = 125*cos(5*t/2 - 1.5707963267949)/2
wy(t) = -125*sin(5*t/2 - 1.5707963267949)/2
```

