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 №2
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

Поля трубка кольцевой формы радиуса R может вращаться вокруг вертикальной оси Oz. Момент инерции трубки относительно оси Oz равен J_z. Внутри трубки без трения движется материальная точка массы m. На трубку действует момент внешних сил относительно оси Oz, равный $M_{Oz} = -c\varphi$ (φ — угол поворота трубки вокруг оси Oz, c — постоянная).

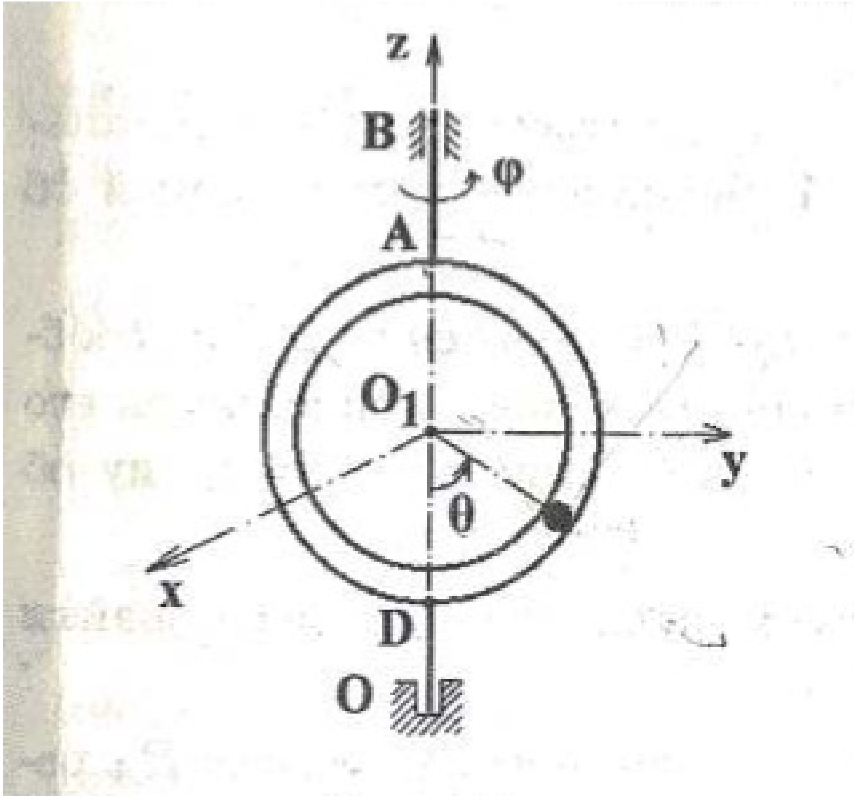


Рис. 11

Нужно вывести подвижную систему координат, связанную с вращающейся трубкой. Считая $\varphi(t)$ и $\theta(t)$ заданными функциями времени, вычислить абсолютную скорость и абсолютное ускорение материальной точки. Изобразить составляющие векторов v и w.

Рассмотрим сначала движение точки в кольцевой трубке. В ней точка просто совершает движение по окружности. Считая, что двумерная плоскость, в которой движется точка, образована осью Oz и неким вектором $O\rho$, который вращается вместе с кольцевой трубкой, можем рассчитать координаты материальной точки в этой двумерной плоскости. Учитывая, что в начальный момент времени точка имеет координаты $z = -R$ и $\rho = 0$, получаем $z = -R\cos\theta$ и $\rho = R\sin\theta$. Теперь рассмотрим совокупность движений точки по окружности и самой кольцевой трубки. Вращение кольцевой трубки не влияет на изменение координаты z точки, однако влияет на кодринаты x и y. Как говорилось ранее, вектор $O\rho$ вращается вместе с кольцевой трубкой, поэтому если в какой-то момент времени точка имеет координату ρ в двумерной плоскости, то её координаты x и y в трёхмерном пространстве соответственно равны $\rho\cos\varphi$ и $\rho\sin\varphi$. В итоге получаем:

$$x(t) = R\sin\theta\cos\varphi$$
$$y(t) = R\sin\theta\sin\varphi$$
$$z(t) = -R\cos\theta$$

Пусть функции $\varphi(t)$ и $\theta(t)$ заданы следующим образом: $\varphi(t) = k_\varphi \pi t, \theta(t) = k_\theta \pi t$. Найдём тогда составляющие скоростие и ускорения точки:

$$v_x(t) = \dot{x}(t) = Rk_\theta \cos\theta \cos\varphi - Rk_\varphi \sin\theta \sin\varphi$$
$$v_y(t) = \dot{y}(t) = Rk_\theta \cos\theta \sin\varphi + Rk_\varphi \sin\theta \cos\varphi$$
$$v_z(t) = \dot{z}(t) = Rk_\theta \sin\theta$$

$$w_x(t) = \dot{v}_x(t) = -Rk_\theta^2 \sin\theta \cos\varphi - Rk_\theta k_\varphi \cos\theta \sin\varphi - Rk_\varphi k_\theta \cos\theta \sin\varphi - Rk_\varphi^2 \sin\theta \cos\varphi$$
$$w_y(t) = \dot{v}_y(t) = -Rk_\theta^2 \sin\theta \sin\varphi + Rk_\theta k_\varphi \cos\theta \cos\varphi + Rk_\varphi k_\theta \cos\theta \cos\varphi - Rk_\varphi^2 \sin\theta \sin\varphi$$
$$w_z(t) = \dot{v}_z(t) = Rk_\theta^2 \cos\theta$$

2 Листинг программы

```
import numpy as np
import sympy as sp
import math
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation

v0 = 5
R = 10

t = sp.Symbol('t')
x = v0 * t - R * sp.cos((v0 * t) / R - np.pi / 2)
y = R + R * sp.sin((v0 * t) / R - np.pi / 2)
vx = sp.diff(x, t)
vy = sp.diff(y, t)
wx = sp.diff(vx, t)
wy = sp.diff(vy, t)
cx = v0 * t

tn = np.linspace(0, 20, 1000)
xn = np.zeros_like(tn)
yn = np.zeros_like(tn)
vxn = np.zeros_like(tn)
vyn = np.zeros_like(tn)
wxn = np.zeros_like(tn)
wyn = np.zeros_like(tn)
cxn = np.zeros_like(tn)

for i in range(len(tn)):
    xn[i] = sp.Subs(x, t, tn[i])
    yn[i] = sp.Subs(y, t, tn[i])
    vxn[i] = sp.Subs(vx, t, tn[i])
    vyn[i] = sp.Subs(vy, t, tn[i])
    wxn[i] = sp.Subs(wx, t, tn[i])
    wyn[i] = sp.Subs(wy, t, tn[i])
    cxn[i] = sp.Subs(cx, t, tn[i])

fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
```

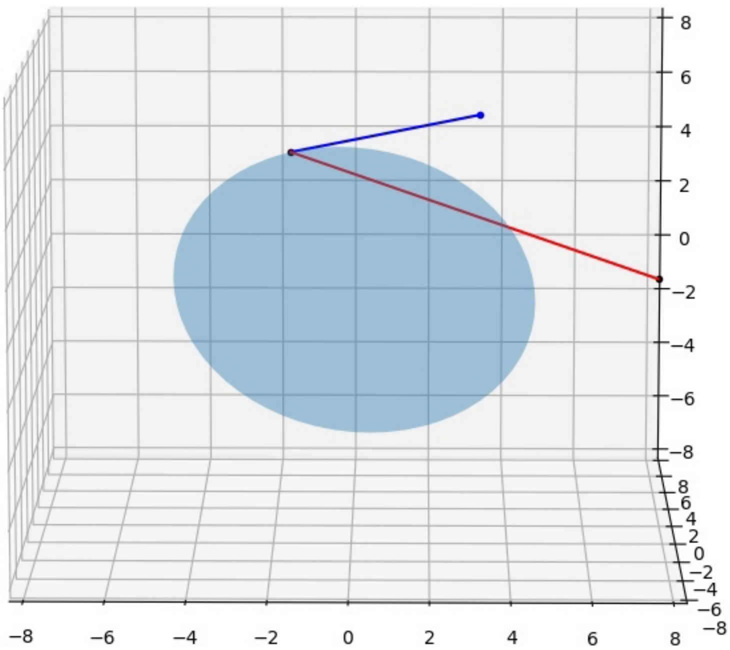
```
ax.axis('equal')
ax.plot(xn, yn, linestyle="--", color="gray")
ax.axhline(y=0, color='gray')
ax.axvline(x=0, color='gray')


def update(i):
    # точка
    point.set_data(xn[i], yn[i])
    # окружность
    global circle
    circle.remove()
    circle = plt.Circle((cxn[i], R), R, edgecolor="#ff924a", facecolor="#fad1b6")
    ax.add_patch(circle)
    # линиявектораскорости
    velocity.set_data([xn[i], xn[i] + vxn[i]], [yn[i], yn[i] + vyn[i]])
    # координатыконцавектораскорости ">"
    varrow_x, varrow_y = Rot2D(varrow_x0, varrow_y0, math.atan2(vyn[i], vxn[i]))
    varrow.set_data(varrow_x + xn[i] + vxn[i], varrow_y + yn[i] + vyn[i])
    # линиявектораускорения
    acceleration.set_data([xn[i], xn[i] + wxn[i]], [yn[i], yn[i] + wyn[i]])
    # координатыконцавектораускорения ">"
    warrow_x, warrow_y = Rot2D(warrow_x0, warrow_y0, math.atan2(wyn[i], wxn[i]))
    warrow.set_data(warrow_x + xn[i] + wxn[i], warrow_y + yn[i] + wyn[i])
    # центрокружности
    circle_center.set_data(cxn[i], R)
    # всёпрокривизну
    curvature = (vxn[i] ** 2 + vyn[i] ** 2) / math.sqrt(wxn[i] ** 2 + wyn[i] ** 2)
    k = curvature / math.sqrt(vxn[i] ** 2 + vyn[i] ** 2)
    curvature_line.set_data([xn[i], xn[i] + vyn[i] * k], [yn[i], yn[i] - vxn[i] * k])
    curvature_center.set_data(xn[i] + vyn[i] * k, yn[i] - vxn[i] * k)
    global curvature_circle
    curvature_circle.remove()
    curvature_circle = plt.Circle((xn[i] + vyn[i] * k, yn[i] - vxn[i] * k), curvature, alpha=0.1)
    ax.add_patch(curvature_circle)

    return point, velocity, varrow, circle, acceleration, warrow, circle_center, curvature_line, curvature_center, curvature_circle


def Rot2D(X, Y, Alpha):
    RX = X * np.cos(Alpha) - Y * np.sin(Alpha)
    RY = X * np.sin(Alpha) + Y * np.cos(Alpha)
```

```
    return RX, RY


# точка
point = ax.plot(xn[0], yn[0], marker=".", color="black")[0]
# окружность
circle = plt.Circle((0, R), R, edgecolor="#ff924a",
    facecolor="#fad1b6")
ax.add_patch(circle)
# линиявектораскорости
velocity = ax.plot([xn[0], xn[0] + vxn[0]], [yn[0], yn[0] + vyn[0]],
    color="#2ec4f2", label="$v$")[0]
# координатыконцавектораскорости     ">"
varrow_x0 = np.array([-0.2 * R, 0, -0.2 * R])
varrow_y0 = np.array([0.1 * R, 0, -0.1 * R])
varrow_x, varrow_y = Rot2D(varrow_x0, varrow_y0, math.atan2(vyn[0],
    vxn[0]))
varrow = ax.plot(varrow_x + xn[0] + vxn[0], varrow_y + yn[0] + vyn[0],
    color="#2ec4f2")[0]
# линияусоркения
acceleration = ax.plot([xn[0], xn[0] + wxn[0]], [yn[0], yn[0] +
    wyn[0]], color="#bd0404", label="$w$")[0]
# координатыконцавектораускорения     ">"
warrow_x0 = np.array([-0.2 * R, 0, -0.2 * R])
warrow_y0 = np.array([0.1 * R, 0, -0.1 * R])
warrow_x, warrow_y = Rot2D(warrow_x0, warrow_y0, math.atan2(wyn[0],
    wxn[0]))
warrow = ax.plot(warrow_x + xn[0] + wxn[0], warrow_y + yn[0] + wyn[0],
    color="#bd0404")[0]
# центрокружности
circle_center = ax.plot(xn[0], R, marker=".", color="#ff924a")[0]
# всёпрокривизну
curvature = (vxn[0] ** 2 + vyn[0] ** 2) / math.sqrt(wxn[0] ** 2 +
    wyn[0] ** 2)
k = curvature / math.sqrt(vxn[0] ** 2 + vyn[0] ** 2)
curvature_line = ax.plot([xn[0], xn[0] + vyn[0] * k], [yn[0], yn[0] -
    vyn[0] * k], linestyle="--", color="#157c9e", label="$\\rho$")[0]
curvature_center = ax.plot(xn[0] + vyn[0] * k, yn[0] - vyn[0] * k,
    marker='.', color="#157c9e")[0]
curvature_circle = plt.Circle((xn[0] + wxn[0] * k, yn[0] + wyn[0] *
    k), curvature, alpha=0.1)
ax.add_patch(curvature_circle)

a = FuncAnimation(fig, update, frames=len(tn), interval=10)

ax.legend()
plt.show()
```

3 Результат работы программы

3.1 $R = 5, k_\varphi = 0.4, k_\theta = 0.4$

```
x(t) =  5*sin(1.25663706143592*t)*cos(1.25663706143592*t)
y(t) =  5*sin(1.25663706143592*t)**2
z(t) =  -5*cos(1.25663706143592*t)
vx(t) =  -6.28318530717959*sin(1.25663706143592*t)**2 + 6.28318530717959*cos(1.25663706143592*t)**2
vy(t) =  12.5663706143592*sin(1.25663706143592*t)*cos(1.25663706143592*t)
vz(t) =  6.28318530717959*sin(1.25663706143592*t)
wx(t) =  -31.5827340834859*sin(1.25663706143592*t)*cos(1.25663706143592*t)
wy(t) =  -15.791367041743*sin(1.25663706143592*t)**2 + 15.791367041743*cos(1.25663706143592*t)**2
wz(t) =  7.89568352087149*cos(1.25663706143592*t)
```



3.2 $R = 5, k_\varphi = 0, k_\theta = 0.4$

```
x(t) =  5*sin(1.25663706143592*t)
y(t) =  0
z(t) =  -5*cos(1.25663706143592*t)
vx(t) =  6.28318530717959*cos(1.25663706143592*t)
vy(t) =  0
vz(t) =  6.28318530717959*sin(1.25663706143592*t)
wx(t) =  -7.89568352087149*sin(1.25663706143592*t)
wy(t) =  0
wz(t) =  7.89568352087149*cos(1.25663706143592*t)
```