Московский авиационный институт
(национальный исследовательский университет)
Институт № 8 «Информационные технологии и прикладная математика»

Лабораторная работа №4
по курсу «Теоретическая механика»
Малые колебания

Выполнил студент группы М8О-206Б-20
Семин Александр Витальевич
Преподаватель: Сухов Е. А.
Дата: 23.12.21
Оценка:

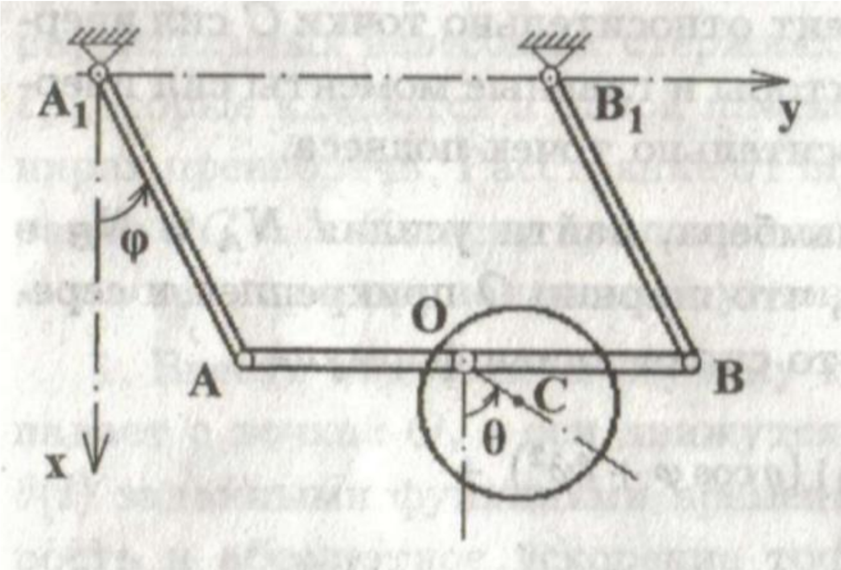
Москва, 2021

Вариант № «21»

Задание:

Реализовать анимацию движения механической системы в среде Python на основе уравнений Лагранжа 2-го рода для малых колебаний, точка D закреплена в точке B.

Механическая система:



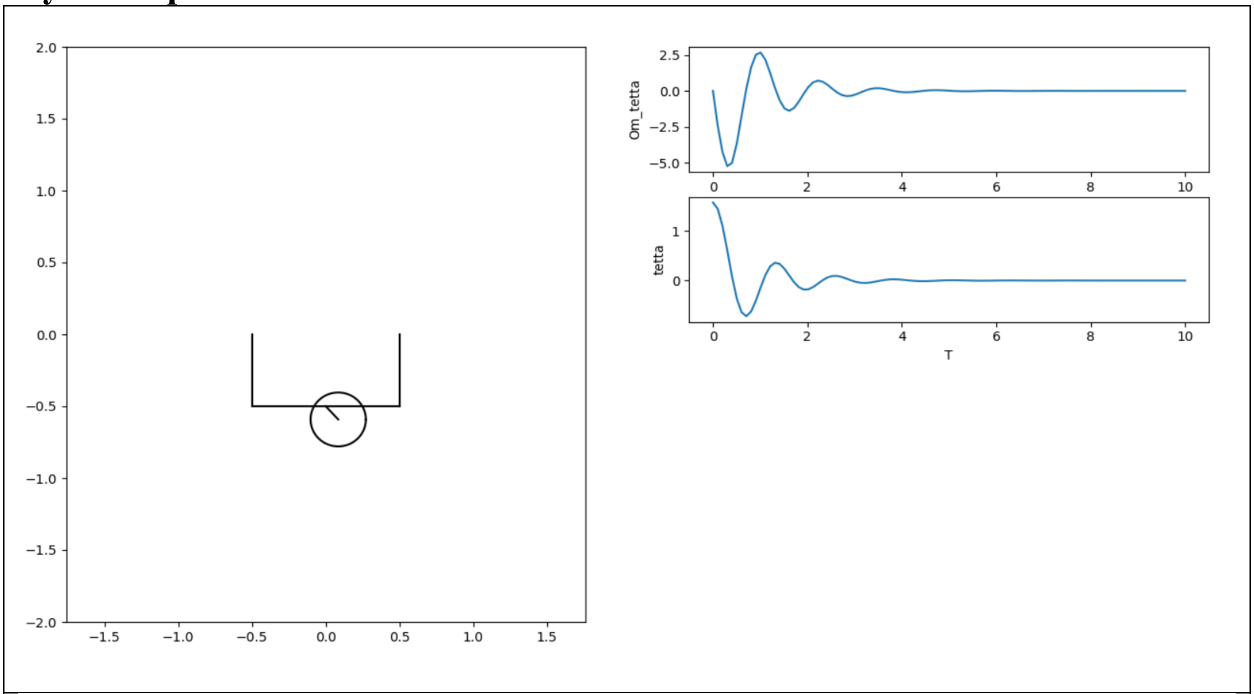
Текст программы:

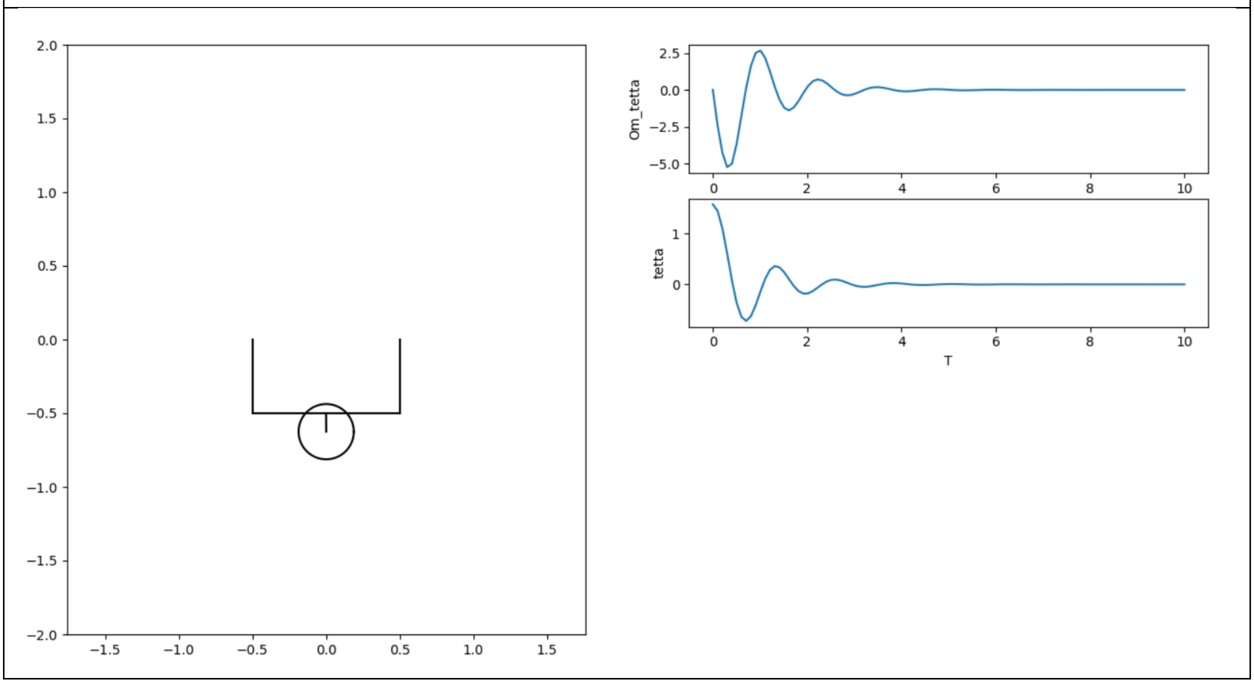
```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
from scipy.integrate import odeint
import sympy as sp
import math


def Circle(X, Y):
    CX = [X + 0.75 * math.cos(i/100) * R for i in range(0, 628)]
    CY = [Y + 0.75 * math.sin(i/100) * R for i in range(0, 628)]
    return CX, CY


def anima(i):
    Beam_1.set_data([-0.5, -0.5], [-0.5, 0])
    Beam_2.set_data([0.5, 0.5], [-0.5, 0])
    Beam_3.set_data([-0.5, 0.5], [-0.5, -0.5])
    Beam_4.set_data([0, 0 + X_o[i]], [-0.5, -0.5 + Y_o[i]])
    circle.set_data(*Circle(0 + X_o[i], -0.5 + Y_o[i]))
    return Beam_1, Beam_2, Beam_3, Beam_4, circle,

def formY(y, t, fV, fOm):
    y1,y2,y3,y4 = y
    dydt = [y3,y4,fV(y1,y2,y3,y4),fOm(y1,y2,y3,y4)]
    return dydt

def formY2(y, t, fOm):
    y1,y2 = y
    dydt = [y2,fOm(y1,y2)]
    return dydt
```

```
m_2 = 5
m_1 = 5
g = 10
l = 1
b = 0.125
k = 0.5
R = 0.25

t = sp.Symbol('t')
phi = 0
tetta = sp.Function('tetta')(t)
om = 0
om_2 = sp.Function('om_2')(t)
E_ab = (m_2 * om**2 * l**2) / 2
Vx_c = om * l * sp.sin(phi)
Vy_c = om * l * sp.cos(phi)
Vx_o = om_2 * sp.sin(tetta) * b
Vy_o = om_2 * sp.cos(tetta) * b
J_cir = (m_1 * R**2) / 2
E_cir = m_1 * ((Vx_o + Vx_c)**2 + (Vy_o + Vy_c)**2) / 2 + (J_cir *
om_2**2)/2

Ekin = E_ab + E_cir

Pi1 = m_2 * g * (l * (1 - sp.cos(phi)) + b)
Pi2 = m_1 * g * (l * (1 - sp.cos(phi)) + b * (1 - sp.cos(tetta)))
Epot = Pi1 + Pi2
M = - k * om_2
L = Ekin - Epot


ur1 = sp.diff(sp.diff(L, om_2), t) - sp.diff(L, tetta) - M
print(ur1)
# ur2 = sp.diff(sp.diff(L, om), t) - sp.diff(L, phi)

#вычисление вторых производных (DV/dt и dom/dt) с использованием
метода Крамера
a11 = ur1.coeff(sp.diff(om_2, t), 1)
# a12 = ur1.coeff(sp.diff(om, t), 1)
# a21 = ur2.coeff(sp.diff(om_2, t), 1)
# a22 = ur2.coeff(sp.diff(om, t), 1)
# b1 = -(ur1.coeff(sp.diff(om_2, t), 0)).coeff(sp.diff(om, t),
0).subs([(sp.diff(tetta, t), om_2), (sp.diff(phi, t), om)])
b1 = -ur1.coeff(sp.diff(om_2, t),0).subs(sp.diff(tetta, t), om_2);
# b2 = -(ur2.coeff(sp.diff(om_2, t), 0)).coeff(sp.diff(om, t),
0).subs([(sp.diff(tetta, t), om_2), (sp.diff(phi, t), om)])

# detA = a11*a22-a12*a21
# detA1 = b1*a22-b2*a21
# detA2 = a11*b2-b1*a21

# dom_tettadt = detA2/detA
# dom_phidt = detA1/detA
print(b1, a11)
domdt = b1/a11
print(domdt)
```

```
countOfFrames = 100

# Построение системы д/у
T = np.linspace(0, 10, countOfFrames)

# f_om_tetta = sp.lambdify([phi, tetta, om_2, om], dom_tettadt,
"numpy")
f_om_phi = sp.lambdify([tetta, om_2], domdt, "numpy")
y0 = [math.pi/2, 0]
sol = odeint(formY2, y0, T, args = (f_om_phi, ))

# X_ab_func = sp.lambdify(phi, sp.sin(phi) * l)
# Y_ab_func = sp.lambdify(phi, - sp.cos(phi) * l)

X_o_func = sp.lambdify(tetta, - sp.sin(tetta) * b)
Y_o_func = sp.lambdify(tetta, - sp.cos(tetta) * b)

X_o = X_o_func(sol[:, 1])
Y_o = Y_o_func(sol[:, 1])

fig = plt.figure(figsize = (17, 8))

ax1 = fig.add_subplot(1, 2, 1)
ax1.axis('equal')
ax1.set(xlim = [-2, 2], ylim = [-2, 2])

ax2 = fig.add_subplot(4, 2, 2)
ax2.plot(T, sol[:, 1])

ax2.set_xlabel('T')
ax2.set_ylabel('Om_tetta')

ax3 = fig.add_subplot(4, 2, 4)
ax3.plot(T, sol[:, 0])

ax3.set_xlabel('T')
ax3.set_ylabel('tetta')

# ax3 = fig.add_subplot(4, 2, 4)
# ax3.plot(T, sol[:, 2])

# ax3.set_xlabel('T')
# ax3.set_ylabel('Om_tetta')

Beam_1, = ax1.plot([-0.5, -0.5], [-0.5, 0], 'black')
Beam_2, = ax1.plot([0.5, 0], [0.5, -0.5], 'black')
Beam_3, = ax1.plot([-0.5, 0.5], [-0.5, -0.5], 'black')
Beam_4, = ax1.plot([0, 0 + X_o[0]], [-0.5, -0.5 + Y_o[0]], 'black')
circle, = ax1.plot(*Circle(0 + X_o[0], -0.5 + Y_o[0]), 'black')
anim = FuncAnimation(fig, anima, frames=countOfFrames, interval=100,
blit=True)
plt.show()
```

Результат работы:





$T = \frac{J_0 \cdot \dot{\theta}^2}{2}$, $J_0 = J_c + m_2 b^2$, $J_c = \frac{m_2 r^2}{2}$.

$$T = \frac{J_0 \dot{\theta}^2}{2}$$

$$J_0 = m_2 b^2 + J_c$$

$$J_c = \frac{m_2 r^2}{2}$$

Сл-но коэффициента:

$$a = m_2 b^2 + \frac{1}{2} m_2 r^2$$

по ф-ле Тейлора

$$\Pi = -m_2 g b \cos\theta = -m_2 g b\left(1 - \frac{\theta^2}{2}\right) = -m_2 g b + \frac{m_2 g b \theta^2}{2}$$

Сл-но коэф. c

$$c = m_2 g b$$

Итого ф-ла

$$\left(m_2 b^2 + \frac{1}{2} m_2 r^2\right) \ddot{\theta} + m_2 g b \theta = -k \dot{\theta}$$

$$\text{Сл-но } T^{(\text{период})} = 2\pi \sqrt{\frac{a}{c}} = 2\pi = \sqrt{\frac{m_2 b^2 + \frac{1}{2} m_2 r^2}{m_2 g b}}$$

1) $k = 0,1$.

$$\left(m_2 b^2 + \frac{1}{2} m_2 r^2\right) \frac{\partial^2 \theta}{\partial t^2} + m_2 g b \theta = -0,1 \frac{\partial \theta}{\partial t}$$

2) $k = 10$

$$\left(m_2 b^2 + \frac{1}{2} m_2 r^2\right) \frac{\partial^2 \theta}{\partial t^2} + m_2 g b \theta = -10 \frac{\partial \theta}{\partial t}$$

Вывод:

В процессе выполнения работы я познакомился с явлением малых колебаний. Они представляют собой распространенный тип движения механических систем, имеющих положение устойчивого равновесия, при малом отклонении от которого возникают силы, стремящиеся вернуть систему в исходное положение. Так что все движение происходит в малой окрестности этого устойчивого положения равновесия.