Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Компьютерные науки и прикладная математика»

Лабораторная работа № 3
по курсу "Теоретическая механика и компьютерное моделирование"

Динамика системы

Выполнил студент группы М8О-206Б-21
Синюков А.С.
Преподаватель: Чекина Евгения Алексеевна

Оценка:
Дата: 26.12.2022

Москва, 2022

Вариант № 24

Задание:

Невесомая трубка, выгнутая в форме кругового кольца радиуса r, закреплена на платформе, которая имеет массу m_1 и может скользить без трения по горизонтальной плоскости. В трубке движется без трения точечный груз M массы m_2.

Составить выражения для кинетической и потенциальной энергии системы, вычислить обобщенные силы.

Используя уравнения Лагранжа второго рода, показать, что дифференциальные уравнения движения системы имеют вид

$(m_1 + m_2)\ddot{s} + m_2 r(\ddot{\theta}cos(\theta) - \dot{\theta}^2 sin(\theta)) = 0$

$\ddot{s}cos(\theta) + r\ddot{(\theta)} + gsin(\theta) = 0$

Записать интеграл энергии системы и циклический интеграл.

Рисунок:

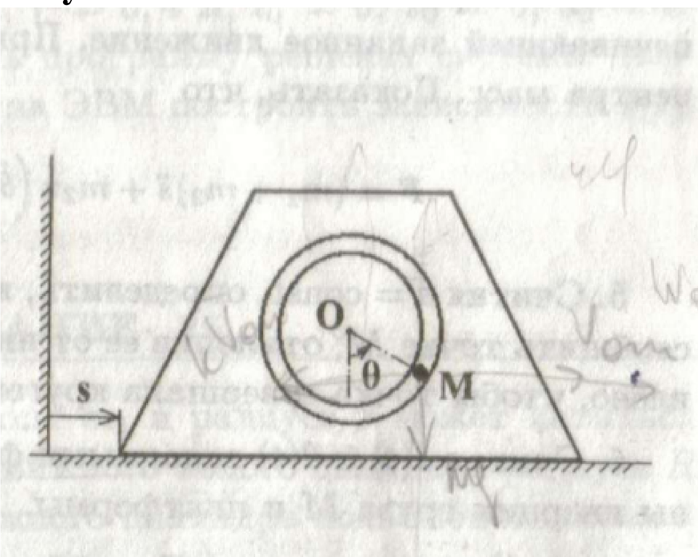

Рис. 24

Текст программы:

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

def Platform(x0, y0): #линия в форме трапеции из 5 точек
    PX = [x0 - 10, x0 - 5, x0 + 5, x0 + 10, x0 - 10]
    PY = [y0 - 7.5, y0 + 10, y0 + 10, y0 - 7.5, y0 - 7.5]
    return PX, PY
```

```
def formY(y, t, f_v, f_om):
    y1, y2, y3, y4 = y
    dy_dt = [y3, y4, f_v(y1, y2, y3, y4), f_om(y1, y2, y3, y4)]
    return dy_dt

#Начальные данные
radius = 0.4
mass_m1 = 20
mass_m2 = 5
g = 9.8

# Составление законов движений, уравнений Лагранжа, диффуров
t = sp.Symbol('t')

s = sp.Function('s')(t)
phi = sp.Function('phi')(t)
v = sp.Function('v')(t)
om = sp.Function('om')(t)


x = s * sp.cos(math.pi) + 0.8
y = -s * sp.sin(math.pi) + 7.5

xa = x + 5 * sp.sin(phi)
ya = y - 5 * sp.cos(phi)


v_2_cm = v**2+(om**2)*(radius**2)/2 + 2*v*om*radius*sp.cos(phi)
moment_inertia = (mass_m2 * radius*radius) / 2

kin_energy = (mass_m1 * v * v)/2 + ((mass_m2 * v_2_cm)/2 + (moment_inertia * om * om)/2)
pot_energy = -(mass_m2 * g * radius * sp.cos(phi))

L = kin_energy - pot_energy

ur1 = sp.diff(sp.diff(L, v), t) - sp.diff(L, s)
ur2 = sp.diff(sp.diff(L, om), t) - sp.diff(L, phi)

a11 = ur1.coeff(sp.diff(v, t), 1)
a12 = ur1.coeff(sp.diff(om, t), 1)
a21 = ur2.coeff(sp.diff(v, t), 1)
a22 = ur2.coeff(sp.diff(om, t), 1)
b1 = -(ur1.coeff(sp.diff(v, t), 0)).coeff(sp.diff(om, t), 0).subs([(sp.diff(s, t), v), (sp.diff(phi, t), om)
b2 = -(ur2.coeff(sp.diff(v, t), 0)).coeff(sp.diff(om, t), 0).subs([(sp.diff(s, t), v), (sp.diff(phi, t), om)

det = a11 * a22 - a12 * a21
det1 = b1 * a22 - b2 * a12
det2 = a11 * b2 - b1 * a21
```

```
dv_dt = det1 / det
dom_dt = det2 / det

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

y0 = [0, sp.rad(0), 1, 2]

f_v = sp.lambdify([s, phi, v, om], dv_dt, "numpy")
f_om = sp.lambdify([s, phi, v, om], dom_dt, "numpy")
sol = odeint(formY, y0, T, args=(f_v, f_om))

X_def = sp.lambdify(s, x)
Y_def = sp.lambdify(s, y)
XA_def = sp.lambdify([s, phi], xa)
YA_def = sp.lambdify([s, phi], ya)
Cord_def = sp.lambdify(t, t)

X = X_def(sol[:, 0])
Y = Y_def(sol[:, 0])
XA = XA_def(sol[:, 0], sol[:, 1])
YA = YA_def(sol[:, 0], sol[:, 1])
Cord = Cord_def(T)


fig = plt.figure(figsize = (20, 10))

ax1 = fig.add_subplot(1, 2, 1)
ax1.axis('equal')
ax1.set(xlim=[-20, 20], ylim=[-20, 30])
ax1.plot([X.min() - 10, X.max() + 10], [0, 0], 'black')
ax1.plot([X.min() - 10, X.min() - 10], [0, Y.max() + 15], 'black')

PrX, PrY = Platform(X[0], Y[0])
Prism = ax1.plot(PrX, PrY, 'blue')[0]

radius, = ax1.plot([X[0], XA[0]], [Y[0], YA[0]], 'black')

Phi = np.linspace(0, 6.28, 20)
r = 0.2
Point = ax1.plot(XA[0] + r * np.cos(Phi), YA[0] + r * np.sin(Phi), 'blue')[0]

ax2 = fig.add_subplot(4, 2, 2)
ax2.set(xlim=[0, 15], ylim=[-1.5, 1.5])
Vgx = [Cord[0]]
Vgy = [sol[:, 2][0]]
V_graph, = ax2.plot(Vgx, Vgy, 'blue')
ax2.set_ylabel('V')

ax3 = fig.add_subplot(4, 2, 4)
```

```
ax3.set(xlim=[0, 15], ylim=[-4.0, 4.0])
Omgx = [Cord[0]]
Omgy = [sol[:, 3][0]]
Om_graph, = ax3.plot(Omgx, Omgy, 'orange')
ax3.set_ylabel('OMEGA')

plt.subplots_adjust(wspace = 0.2, hspace = 0.2)

# Анимируем
def anima(i):
    PrX, PrY = Platform(X[i], Y[i])
    Prism.set_data(PrX, PrY)
    radius.set_data([X[i], XA[i]], [Y[i], YA[i]])
    Point.set_data(XA[i] + r * np.cos(Phi), YA[i] + r * np.sin(Phi))
    Vgx.append(Cord[i])
    Vgy.append(sol[:, 2][i])
    Omgx.append(Cord[i])
    Omgy.append(sol[:, 3][i])
    V_graph.set_data(Vgx, Vgy)
    Om_graph.set_data(Omgx, Omgy)
    return Prism, radius, Point, V_graph, Om_graph

anim = FuncAnimation(fig, anima, frames = 1000, interval = 1, blit = True)

plt.show()
```

Выводы формул:

$$v_{цт}^2 = v^2 + \frac{\omega^2 r^2}{2} + 2 v \omega r \cos\theta$$

$$J = \frac{m r^2}{2}$$

$$T = \frac{M v^2}{2} + \left[\frac{m v_{цт}^2}{2} + \frac{J \omega^2}{2} \right]$$

$$\Pi = -m g r \cos\theta$$

$$L = T - \Pi = \frac{M v^2}{2} + \frac{m v^2}{2} + \frac{m \omega^2 r^2}{4} + m v \omega r \cos\theta +$$

$$+ \frac{m r^2 \omega^2}{4} + m g r \cos\theta$$

1) $$\frac{\partial L}{\partial v} = M v + m v + m \omega r \cos\theta$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) = M a + m a + m \varepsilon r \cos\theta - m \omega^2 r \sin\theta$$

$$\frac{\partial L}{\partial s} = 0$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial v}\right) - \frac{\partial L}{\partial s} = 0 \Rightarrow a(M+m) + m r(\varepsilon \cos\theta -$$

$$- \omega^2 \sin\theta) = \ddot{s}(M+m) + m r(\ddot{\theta} \cos\theta - \dot{\theta}^2 \sin\theta) = 0$$

2) $$\frac{\partial L}{\partial \omega} = \frac{m \omega r^2}{2} + m v r \cos\theta + \frac{m \omega r^2}{2} =$$

$$= m \omega r^2 + m v r \cos\theta$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega}\right) = m \varepsilon r^2 + m a r \cos\theta - m v \omega r \sin\theta$$

$$\frac{\partial L}{\partial \theta} = - m v \omega r \sin\theta - m g r \sin\theta$$

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \omega}\right) - \frac{\partial L}{\partial \theta} = m\varepsilon r^2 + \cancel{m}ar\cos\theta -$$

$$- \cancel{m v \omega r \sin\theta} + \cancel{m v \omega r \sin\theta} + \cancel{m} g r \sin\theta =$$

$$= \varepsilon r^2 + a\cos\theta + g\sin\theta =$$

$$= \ddot{\theta} r^2 + \ddot{s}\cos\theta + g\sin\theta = 0$$

Результат работы программы: