Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №2**

**по курсу «Теоретическая механика»**

**Анимация системы**

Выполнил студент группы М8О-201Б-20

Сазонов Вадим Кириллович

Оценка:

Дата:

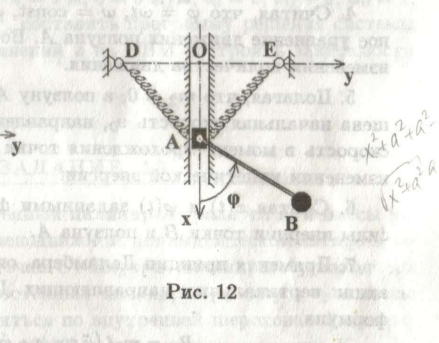
Москва, 2021

**Вариант №12«Фантастическая система»**

**Задание:**

Реализовать анимацию движения механической системы в среде Python

**Механическая система:**



**Текст программы**

Основная :

import sympy as sp

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

from matplotlib.patches import Rectangle

from matplotlib.patches import Circle

import math

Frames = 50

Interval\_Frame = 0

Repeat\_Delay\_Anim = 0

# initializing

a = 0.5 # y-size of rectangle

b = 1 # x-size of rectangle

t = sp.Symbol("t") # t is a symbol variable

# point A (center of rectangle):

ax = 0 # можно задать x(t) для блока

ay = sp.sin(t) # можно задать y(t) для блока

vxrec = sp.diff(ax, t)

vyrec = sp.diff(ay, t)

# point B (center of circle):

br = 1.5 # можно задать r(t) для шара

bfi = 3 \* sp.sin(t) # можно задать fi(t) шара

bx = br \* sp.cos(bfi)

by = br \* sp.sin(bfi) + ay

vxcirc = sp.diff(bx, t)

vycirc = sp.diff(by, t)

# constant arrays

L1X = [-a/2 - 0.01, -a/2 - 0.01]

L2X = [a/2 + 0.01, a/2 + 0.01]

LY = [-1 - b/2, 1 + b/2]

# initializing arrays

T = np.linspace(0, 2 \* np.pi, Frames)

AX = np.zeros\_like(T)

AY = np.zeros\_like(T)

BR = np.zeros\_like(T)

BFI = np.zeros\_like(T)

BX = np.zeros\_like(T)

BY = np.zeros\_like(T)

VXREC = np.zeros\_like(T)

VYREC = np.zeros\_like(T)

VXCIRC = np.zeros\_like(T)

VYCIRC = np.zeros\_like(T)

# progress bar (loading bar)

# filling arrays

for i in range(len(T)):

AY[i] = sp.Subs(ay, t, T[i])

BR[i] = sp.Subs(br, t, T[i])

BFI[i] = sp.Subs(bfi, t, T[i])

BX[i] = sp.Subs(bx, t, T[i])

BY[i] = sp.Subs(by, t, T[i])

VXREC[i] = sp.Subs(vxrec, t, T[i])

VYREC[i] = sp.Subs(vyrec, t, T[i])

VXCIRC[i] = sp.Subs(vxcirc, t, T[i])

VYCIRC[i] = sp.Subs(vycirc, t, T[i])

print()

# start plotting

fig = plt.figure()

ax0 = fig.add\_subplot(1, 2, 1)

ax0.axis("equal")

# ax0.set(xlim=[-2.5, 2.5], ylim=[-3, 3])

# plotting environment

ax0.plot(L1X, LY, color="grey") # left wall

ax0.plot(L2X, LY, color="grey") # right wall

sl, = ax0.plot([-2, -a/2], [0, AY[0]], color="brown") # left spring (rope)

sr, = ax0.plot([2, a/2], [0, AY[0]], color="brown") # right spring (rope)

ax0.plot(-2, 0, marker=".", color="black") # left joint

ax0.plot(2, 0, marker=".", color="black") # right joint

rect = plt.Rectangle((-a/2, -b/2 + AY[0]),

a, b + AY[0], color="black") # rectangle

circ = plt.Circle((BX[0], BY[0]), 0.1, color="grey") # circle

# plotting radius vector of B

R\_vector, = ax0.plot([0, BX[0]], [0, BY[0]], color="grey")

# adding statistics

ax2 = fig.add\_subplot(4, 2, 2)

ax2.plot(T, VXREC)

ax2.set\_xlabel("T")

ax2.set\_ylabel("VxRectangle")

# ax2.axis("equal")

ax2.set(xlim=[0, 2 \* np.pi], ylim=[-1, 1])

ax3 = fig.add\_subplot(4, 2, 4)

ax3.plot(T, VYREC)

ax3.set\_xlabel("T")

ax3.set\_ylabel("VyRectangle")

# ax3.axis("equal")

ax3.set(xlim=[0, 2 \* np.pi])

ax4 = fig.add\_subplot(4, 2, 6)

ax4.plot(T, VXCIRC)

ax4.set\_xlabel("T")

ax4.set\_ylabel("VxCircle")

# ax4.axis("equal")

ax4.set(xlim=[0, 2 \* np.pi])

ax5 = fig.add\_subplot(4, 2, 8)

ax5.plot(T, VYCIRC)

ax5.set\_xlabel("T")

ax5.set\_ylabel("VyCircle")

# ax5.axis("equal")

ax5.set(xlim=[0, 2 \* np.pi])

plt.subplots\_adjust(wspace=0.6, hspace=1)

# function for initializing the positions

def init():

rect.set\_y(-b/2)

ax0.add\_patch(rect)

circ.center = (0, 0)

ax0.add\_patch(circ)

return rect, circ

# function for recounting the positions

def animation(i):

rect.set\_y(AY[i] - b/2)

sl.set\_data([-2, -a/2], [0, AY[i]])

sr.set\_data([2, a/2], [0, AY[i]])

R\_vector.set\_data([0, BX[i]], [AY[i], BY[i]])

circ.center = (BX[i], BY[i])

return sl, sr, rect, R\_vector, circ,

# animating function

anim = FuncAnimation(fig, animation,init\_func=init, frames=50, interval=100,blit=False, repeat=True)

plt.show()

**Результат работы:**

