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-205Б-21

Жилов Андрей Алексеевич

Преподаватель: Беличенко Михаил Валериевич

Оценка:

Дата: 15.12.2022

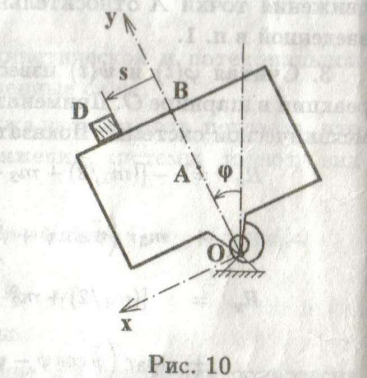
Москва, 2022

**Вариант №10**

**Задание:**

Реализовать анимацию движения механической системы в среде Octave (или Matlab)

**Механическая система:**

****

**Текст программы**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAnimation

import sympy as sp

def Rot(X,Y,Alpha):

RotX = X\*np.cos(Alpha) - Y\*np.sin(Alpha)

RotY = X\*np.sin(Alpha) + Y\*np.cos(Alpha)

return RotX, RotY

t = np.linspace(0, 10, 1001)

phi = 0.4\*np.cos(2.14 \* t)#угол в радианах

k= np.sin(phi)/3

s=np.sin(np.pi\*t\*4)/3

WeightA=4#ширина нижнего блока

HeightA=2#высота нижнего блока

xTA = np.array([-WeightA/2, -WeightA/2, WeightA/2, WeightA/2,0.2\*WeightA, 0, -0.2\*WeightA,-WeightA/2])

yTA= np.array([-HeightA/2, HeightA/2, HeightA/2, -HeightA/2, -HeightA/2, -HeightA/2-0.825,-HeightA/2,-HeightA/2])

xA\_0=0#положение равновесия пружины

yA\_0=HeightA/2+1

xA=xA\_0#+yA\_0\*np.sin(phi)

yA=yA\_0#\*np.cos(phi)

WeightB=0.4

HeightB=0.3

xTB = np.array([-WeightB/2, -WeightB/2, WeightB/2, WeightB/2,-WeightB/2])

yTB= np.array([-HeightB/2, HeightB/2, HeightB/2, -HeightB/2,-HeightB/2])

yB\_0=HeightA+1+HeightB/2

xB\_0=0

xB=xB\_0#-WeightA\*phi

yB=yB\_0

MaxXB=WeightA\*0.38

MinXB=WeightA\*-0.38

lastX=MinXB

def FxB(phi,s):

global lastX

if(phi<=0):

if(lastX>=MaxXB):

lastX=MaxXB

return lastX

lastX=-WeightA\*phi\*2+MinXB

else:

if(lastX<=MinXB):

lastX=MinXB

return lastX

lastX=-WeightA\*phi\*2+MaxXB

return lastX

fig = plt.figure(figsize=[25,25])

ax = fig.add\_subplot(1,1,1)

ax.axis('equal')

ax.set(xlim=[-5,6],ylim=[0,10])

ax.plot([-7,-7,10],[10,0.125,0.125],color=[0,0.5,0],linewidth=2)#ск

N = 4

r1=0.2

r2=1

Beta0 = np.linspace(0,1,50\*N+1)

Betas = Beta0\*(N\*2\*np.pi-phi[0])

xS = -(r1+(r2-r1)\*Betas/(N\*2\*np.pi-phi[0]))\*np.sin(Betas)

yS = (r1+(r2-r1)\*Betas/(N\*2\*np.pi-phi[0]))\*np.cos(Betas)

SpiralnayaPruzzhina = ax.plot(xS, yS, color=[1, 0.5, 0.5])[0]

A=ax.plot(Rot(0,0,phi[1]),color=[1,0,0])[0]

B=ax.plot(Rot(0,0,phi[1]),color=[1,0,0])[0]

def draw(i):

A.set\_data(Rot(xA+xTA,yA+yTA,phi[i]))

B.set\_data(Rot(FxB(phi[i],phi[i])+xTB,yB+yTB,phi[i]))

Betas = Beta0 \* (N \* 2 \* np.pi\*(1-1/37) + phi[i])

xS = -(r1 + (r2 - r1) \* Betas / (N \* 2 \* np.pi + phi[i])) \* np.sin(Betas)

yS = (r1 + (r2 - r1) \* Betas / (N \* 2 \* np.pi + phi[i])) \* np.cos(Betas)

SpiralnayaPruzzhina.set\_data(xS, yS)

return [A,B,SpiralnayaPruzzhina]

animation=FuncAnimation(fig,draw,interval = t[1]-t[0],frames=len(t))

plt.show()

**Результат работы:**

|  |  |
| --- | --- |
|  |  |
|  |  |