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-206Б-20,

Семин Александр Витальевич

Преподаватель: Сухов Е. А

Дата: 12.12

Оценка:

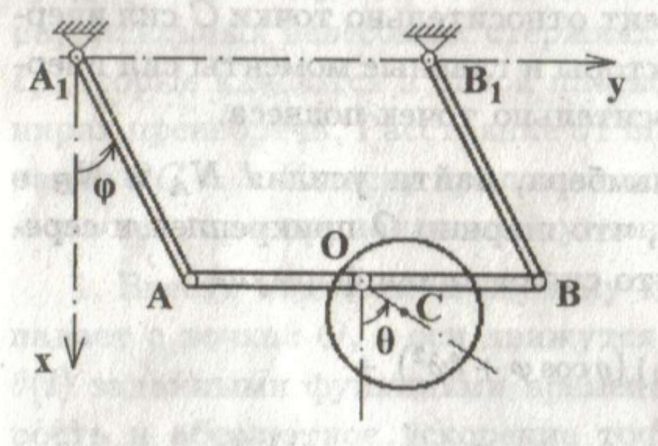
Москва, 2021

**Вариант №«21»**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**

****

**Текст программы:**

import numpy as np

import matplotlib.pyplot as plt

from matplotlib.patches import Circle

from matplotlib.animation import FuncAnimation

import sympy as sp

import math

def Circle(X, Y):

CX = [X + 0.75 \* math.cos(i/100) for i in range(0, 628)]

CY = [Y + 0.75 \* math.sin(i/100) for i in range(0, 628)]

return CX, CY

def anima(i):

Beam\_1.set\_data([- 4, - 4 + XA[i]], [0, YA[i]])

Beam\_2.set\_data([4, 4 + XA[i]], [0, YA[i]])

Beam\_3.set\_data([- 4 + XA[i], 4 + XA[i]], [YA[i], YA[i]])

Beam\_4.set\_data([XA[i], XA[i] + XO[i]], [YA[i], YA[i] + YO[i]])

circle.set\_data(\*Circle(XA[i] + XO[i], YA[i] + YO[i]))

return Beam\_1, Beam\_2, Beam\_3, Beam\_4, circle,

beam\_length = 4

beam\_length\_2 = 1.5

t = sp.Symbol('t')

phi = sp.sin(t) \* sp.cos(t) + math.pi/4

tetta = 3 \* t

*# speed and acceleration of point A*

vxa = sp.diff(sp.sin(phi) \* beam\_length, t)

vya = sp.diff(sp.cos(phi) \* beam\_length, t)

va = (vxa\*\*2 + vya\*\*2)\*\*0.5

wa = (sp.diff(vxa, t)\*\*2 + sp.diff(vya, t)\*\*2)\*\*0.5

*# speed and acceleration of point C*

vxc = sp.diff(sp.sin(tetta) \* beam\_length\_2, t) + vxa

vyc = sp.diff(sp.cos(tetta) \* beam\_length\_2, t) + vya

vc = (vxc\*\*2 + vyc\*\*2)\*\*0.5

wc = (sp.diff(vxc, t)\*\*2 + sp.diff(vyc, t)\*\*2)\*\*0.5

T = np.linspace(0, 2\*math.pi, 1000)

XA = np.zeros\_like(T)

YA = np.zeros\_like(T)

XO = np.zeros\_like(T)

YO = np.zeros\_like(T)

VA = np.zeros\_like(T)

WA = np.zeros\_like(T)

VC = np.zeros\_like(T)

WC = np.zeros\_like(T)

for i in np.arange(len(T)):

XA[i] = sp.Subs(beam\_length \* sp.cos(2 \* phi), t, T[i])

YA[i] = sp.Subs(-math.sqrt(beam\_length\*\*2 - XA[i]\*\*2), t, T[i])

XO[i] = sp.Subs(beam\_length\_2 \* sp.cos(tetta), t, T[i])

YO[i] = sp.Subs(beam\_length\_2 \* sp.sin(tetta), t, T[i])

VA[i] = sp.Subs(va, t, T[i])

WA[i] = sp.Subs(wa, t, T[i])

VC[i] = sp.Subs(vc, t, T[i])

WC[i] = sp.Subs(wc, t, T[i])

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

ax1 = fig.add\_subplot(1, 2, 1)

ax1.axis('equal')

ax1.set(xlim=[-10, 10], ylim=[-10, 10])

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

ax2.plot(T, VA)

ax2.set\_xlabel('T')

ax2.set\_ylabel('v of point A')

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

ax3.plot(T, WA)

ax3.set\_xlabel('T')

ax3.set\_ylabel('w of point A')

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

ax4.plot(T, VC)

ax4.set\_xlabel('T')

ax4.set\_ylabel('v of point C')

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

ax5.plot(T, WC)

ax5.set\_xlabel('T')

ax5.set\_ylabel('w of point C')

Beam\_1, = ax1.plot([-2, -2 + XA[0]], [0, YA[0]], 'black')

Beam\_2, = ax1.plot([2, 2 + XA[0]], [0, YA[0]], 'black')

Beam\_3, = ax1.plot([- 4 + XA[0], 4 + XA[0]], [YA[0], YA[0]], 'black')

Beam\_4, = ax1.plot([XA[0], XA[0] + XO[0]], [YA[0], YA[0] + YO[0]], 'black')

circle, = ax1.plot(\*Circle(XA[0] + XO[0], YA[0] + YO[0]), 'black')

anim = FuncAnimation(fig, anima, frames=1000, interval=0.01, blit=True)

plt.show()

**Результат работы:**

|  |
| --- |
|  |
|  |

**Вывод:**

С помощью полученных на курсе знаний я написал программу анимации системы для своего варианта.

Можно отметить удобство построения графиков, которые даже в моей работе помогли наглядно представить процессы изменения величин.