

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:

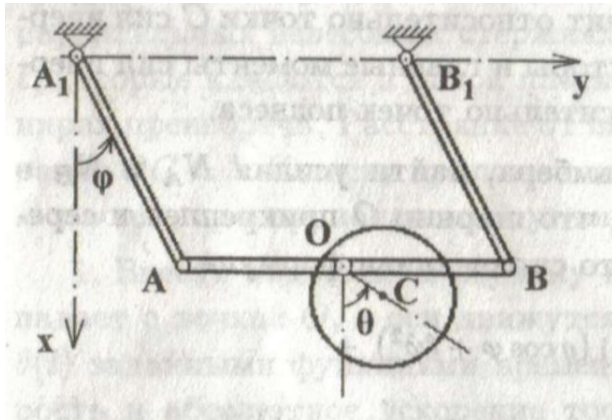
Москва, 2021

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,

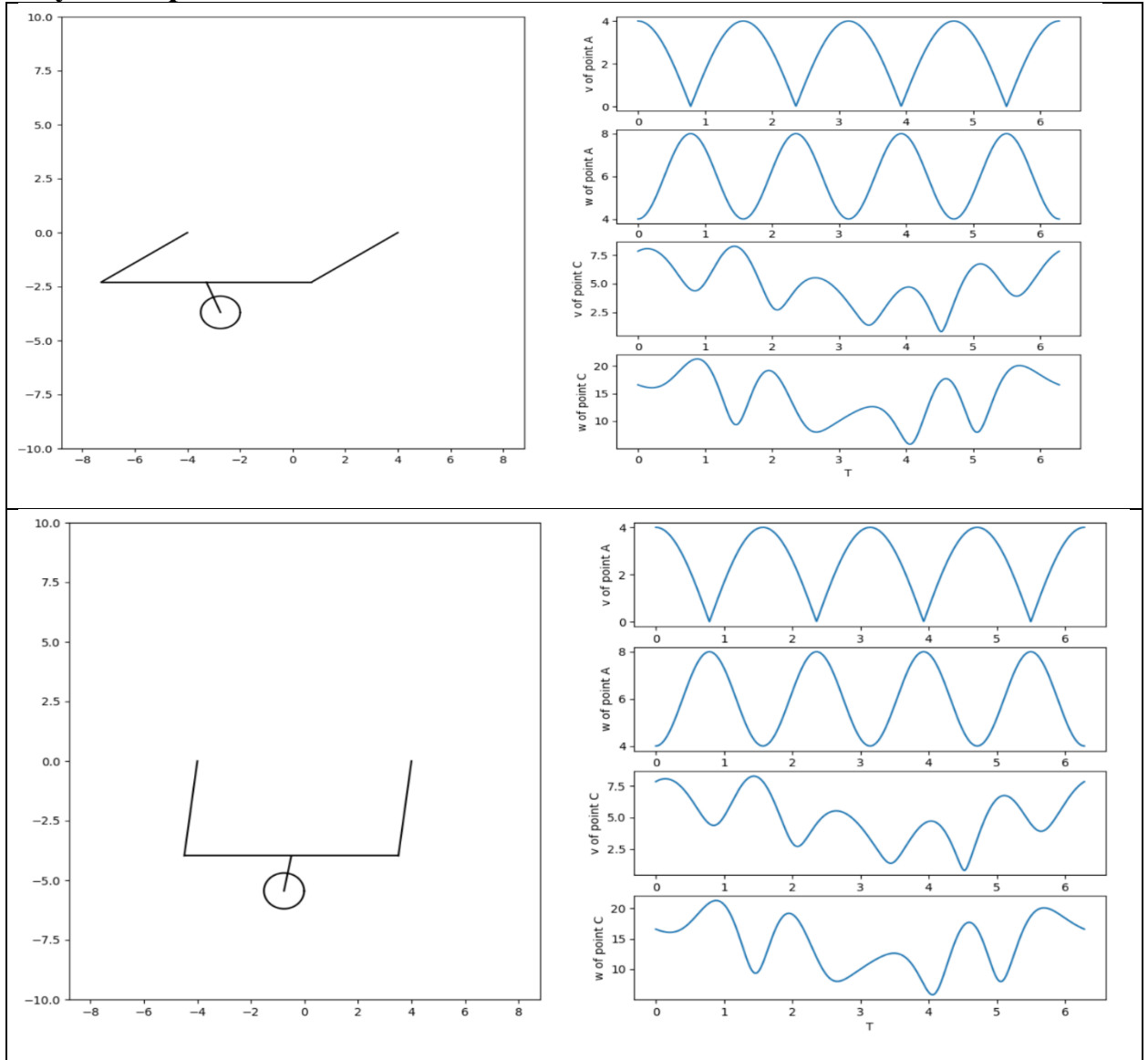
beam_length = 4
beam_length_2 = 1.5
t = sp.Symbol('t')
phi = sp.sin(t) * sp.cos(t) + math.pi/4
tetta = 3 * t
# speed and acceleration of point A
vxa = sp.diff(sp.sin(phi) * beam_length, t)
vya = sp.diff(sp.cos(phi) * beam_length, t)
va = (vxa**2 + vya**2)**0.5
wa = (sp.diff(vxa, t)**2 + sp.diff(vya, 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.