Московский авиационный институт

(национальный исследовательский университет)

Институт № 8 «Информационные технологии и прикладная математика»

**Лабораторная работа №2**

**по курсу «Теоретическая механика и компьютерное моделирование»**

**Анимация системы**

Выполнил студент группы М8О-207Б-20

Чекменев Вячеслав Алексеевич

Преподаватель: Чекина Евгения Алексеевна

Оценка:

Дата:

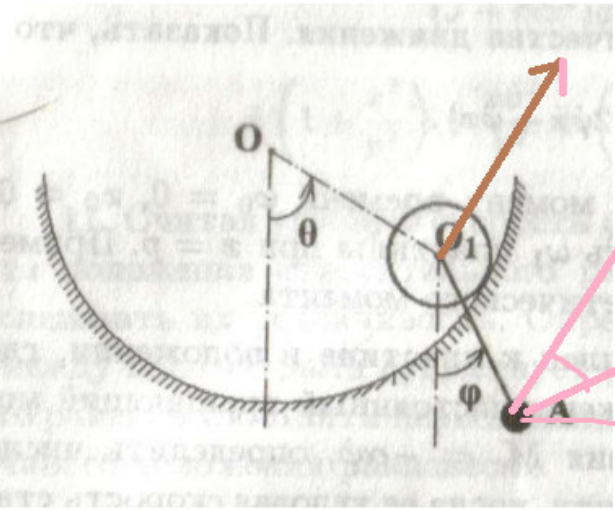
Москва, 2021

**Вариант №«25»**

**Задание:**

Реализовать анимацию движения механической системы используя язык программирования Python.

**Механическая система:**

****

**Текст программы:**

import numpy as np

import matplotlib.pyplot as plt

import sympy as sp

from matplotlib.animation import FuncAnimation

import math

Steps = 1001

R\_Ground = 6

R\_Circle = R\_Ground/6

l = R\_Ground/2 # length of the palka between O1 and A

t = sp.Symbol('t')

thetta = sp.cos(10\*t)

phi = sp.cos(16\*t)

omega\_phi = sp.diff(phi, t)

omega\_thetta = sp.diff(thetta, t)

# Ground

alpha = np.linspace(-math.pi, 0, Steps)

X\_Ground = R\_Ground + R\_Ground \* np.cos(alpha)

Y\_Ground = R\_Ground + R\_Ground \* np.sin(alpha)

# Point O

X\_O = R\_Ground

Y\_O = R\_Ground

# circle

beta = np.linspace(0, 2\*math.pi, Steps)

X\_Circle = R\_Circle \* np.cos(beta)

Y\_Circle = R\_Circle \* np.sin(beta)

# Point O1

X\_O1 = -(R\_Ground - R\_Circle) \* sp.sin(thetta) + R\_Ground

Y\_O1 = -(R\_Ground - R\_Circle) \* sp.cos(thetta) + R\_Ground

# point A

X\_A = X\_O1 + l\*sp.sin(phi)

Y\_A = Y\_O1 - l\*sp.cos(phi)

#constructing corresponding arrays

T = np.linspace(0, 10, Steps)

Thetta = np.zeros\_like(T)

Phi = np.zeros\_like(T)

Omega\_phi = np.zeros\_like(T)

Omega\_thetta = np.zeros\_like(T)

XO1 = np.zeros\_like(T)

YO1 = np.zeros\_like(T)

XA = np.zeros\_like(T)

YA = np.zeros\_like(T)

#filling arrays with corresponding values

for i in range(len(T)):

Thetta[i] = sp.Subs(thetta, t, T[i])

Phi[i] = sp.Subs(phi, t, T[i])

Omega\_phi[i] = sp.Subs(omega\_phi, t, T[i])

Omega\_thetta[i] = sp.Subs(omega\_thetta, t, T[i])

XO1[i] = sp.Subs(X\_O1, t, T[i])

YO1[i] = sp.Subs(Y\_O1, t, T[i])

XA[i] = sp.Subs(X\_A, t, T[i])

YA[i] = sp.Subs(Y\_A, t, T[i])

# graphs

fig\_for\_graphs = plt.figure(figsize=[13, 7])

ax\_for\_graphs = fig\_for\_graphs.add\_subplot(2, 2, 1)

ax\_for\_graphs.plot(T, Phi, color='blue')

ax\_for\_graphs.set\_title("Phi(t)")

ax\_for\_graphs.set(xlim=[0, 10])

ax\_for\_graphs.grid(True)

ax\_for\_graphs = fig\_for\_graphs.add\_subplot(2, 2, 2)

ax\_for\_graphs.plot(T, Thetta, color='red')

ax\_for\_graphs.set\_title('Thetta(t)')

ax\_for\_graphs.set(xlim=[0, 10])

ax\_for\_graphs.grid(True)

ax\_for\_graphs = fig\_for\_graphs.add\_subplot(2,2,3)

ax\_for\_graphs.plot(T, Omega\_phi, color='green')

ax\_for\_graphs.set\_title("phi'(t) = omega\_phi(t)")

ax\_for\_graphs.set(xlim=[0, 10])

ax\_for\_graphs.grid(True)

ax\_for\_graphs = fig\_for\_graphs.add\_subplot(2, 2, 4)

ax\_for\_graphs.plot(T, Omega\_thetta, color='black')

ax\_for\_graphs.set\_title("thetta'(t) = omega\_thetta(t)")

ax\_for\_graphs.set(xlim=[0, 10])

ax\_for\_graphs.grid(True)

# some settings

fig = plt.figure()

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

ax.axis("equal")

ax.set(xlim=(0, 12), ylim=(0, 12))

# plot first zero state

Ground = ax.plot(X\_Ground, Y\_Ground, color='black', linewidth=2)

Point\_O = ax.plot(X\_O, Y\_O, color='red', linewidth=4)

Draw\_palka = ax.plot([X\_O, XO1[0]], [Y\_O, YO1[0]], 'r--')[0]

Draw\_Circle = ax.plot(X\_Circle + XO1[0], Y\_Circle + YO1[0], color='blue', linewidth=1)[0]

Draw\_point\_O1 = ax.plot(XO1[0], YO1[0], color='blue', linewidth=3, marker='o')[0]

Draw\_point\_A = ax.plot(XA[0], YA[0], 'r', marker='o', markersize=15)[0]

Draw\_palka\_O1\_A = ax.plot([XO1[0], XA[0]], [YO1[0], YA[0]], 'b')[0]

# function for updating state of the system

def kinoteatr\_five\_zvezd\_na\_novokuzneckoy(i):

Draw\_point\_O1.set\_data(XO1[i], YO1[i])

Draw\_Circle.set\_data(X\_Circle + XO1[i], Y\_Circle + YO1[i])

Draw\_palka.set\_data([X\_O, XO1[i]], [Y\_O, YO1[i]])

Draw\_point\_A.set\_data(XA[i], YA[i])

Draw\_palka\_O1\_A.set\_data([XO1[i], XA[i]], [YO1[i], YA[i]])

return [Draw\_point\_O1, Draw\_Circle, Draw\_palka, Draw\_point\_A]

anime = FuncAnimation(fig, kinoteatr\_five\_zvezd\_na\_novokuzneckoy,

frames=Steps, interval=1)

# show figure

plt.show()

# anime.save("cringe.gif")

**Результат работы:**

|  |
| --- |
|  |
|  |

