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

Куценко Борис Дмитриевич

Преподаватель: Чекина Евгения Алексеевна

Оценка:

Дата:

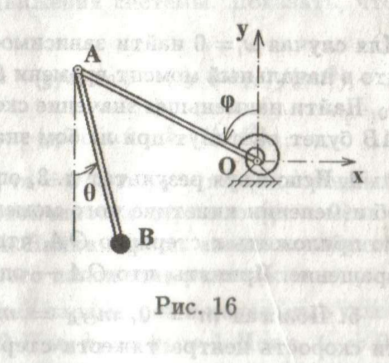
Москва, 2021

**Вариант №«16»**

**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.animation import FuncAnimation**

**import matplotlib.patches as pat**

**import sympy as sp**

**import math**

**Steps = 1000**

**t = sp.Symbol('t')**

**phi = 4 \* sp.sin(t)**

**thetta = math.pi / 4 \* 5 \* t**

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

**OA = 5**

**AB = 4**

**X\_A = OA \* sp.cos(phi)**

**Y\_A = OA \* sp.sin(phi)**

**X\_B = X\_A + AB \* sp.sin(thetta)**

**Y\_B = Y\_A - AB \* sp.cos(thetta)**

**VX\_B = sp.diff(X\_B,t)**

**VY\_B = sp.diff(Y\_B,t)**

**VX\_A = sp.diff(X\_A,t)**

**VY\_A = sp.diff(Y\_A,t)**

**V\_A = sp.sqrt(VX\_A \*\* 2 + VY\_A \*\* 2)**

**V\_B = sp.sqrt(VX\_B \*\* 2 + VY\_B \*\* 2)**

**Nv= 3**

**R1 = 0.001**

**R2 = 0.4**

**Ksi = np.linspace(0, 0, )**

**fig = plt.figure(figsize=(10,10))**

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

**ax.axis('equal')**

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

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

**Phi = np.zeros\_like(T)**

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

**XB = np.zeros\_like(T)**

**YB = np.zeros\_like(T)**

**XA = np.zeros\_like(T)**

**YA = np.zeros\_like(T)**

**Alpha =np.zeros\_like(T)**

**Phi =np.zeros\_like(T)**

**VB =np.zeros\_like(T)**

**VA =np.zeros\_like(T)**

**for i in range(len(T)):**

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

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

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

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

**XB[i] = sp.Subs(X\_B, t, T[i])**

**YB[i] = sp.Subs(Y\_B, t, T[i])**

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

**VB[i] = sp.Subs(V\_B, t, T[i])**

**VA[i] = sp.Subs(V\_A, t, T[i])**

**alpha = np.linspace(0, Nv\*6.283+Phi[0], Steps)**

**X\_SpiralSpr = -(R1 + alpha \* (R2 - R1) / alpha[-1]) \* np.sin(alpha)**

**Y\_SpiralSpr = (R1 + alpha \* (R2 - R1) / alpha[-1]) \* np.cos(alpha)**

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

**R\_Circle = 0.5**

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

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

**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, VA, color='red')**

**ax\_for\_graphs.set\_title('Va(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, VB, color='black')**

**ax\_for\_graphs.set\_title("Vb(t)")**

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

**ax\_for\_graphs.grid(True)**

**OX = ax.plot([-12, 12], [-0.5,-0.5], 'black', linestyle = '--')**

**Draw\_Spring = ax.plot(X\_SpiralSpr, Y\_SpiralSpr, color='#666666')[0]**

**Draw\_OA=ax.plot([0, XA[0]], [0, YA[0]], color='#808080')[0]**

**Draw\_AB=ax.plot([XA[0], XB[0]], [YA[0], YB[0]], color='#808080' )[0]**

**PointB = ax.plot(XB[0], YB[0])[0]**

**PointA = ax.plot(XA[0], YA[0], color='#a0a0a0', marker='o')[0]**

**Draw\_Circle = ax.plot(X\_Circle + XB[0], Y\_Circle + YB[0], color='black', linewidth=1)[0]**

**triangle = pat.Polygon([(0,0), (-0.5, -0.5), (0.5, -0.5)], color='#d3d3d3')**

**ax.add\_patch(triangle)**

**def update(i):**

**PointB.set\_data(XB[i],YB[i])**

**Draw\_OA.set\_data([0, XA[i]], [0, YA[i]])**

**PointA.set\_data(XA[i], YA[i])**

**Draw\_AB.set\_data([XA[i], XB[i]], [YA[i], YB[i]])**

**Draw\_Circle.set\_data(X\_Circle + XB[i], Y\_Circle + YB[i])**

**alpha = np.linspace(0, Nv\*6.28+Phi[i], 100)**

**X\_SpiralSpr = -(R1 + alpha \* (R2 - R1) / alpha[-1]) \* np.sin(alpha - 1.57)**

**Y\_SpiralSpr = (R1 + alpha \* (R2 - R1) / alpha[-1]) \* np.cos(alpha - 1.57)**

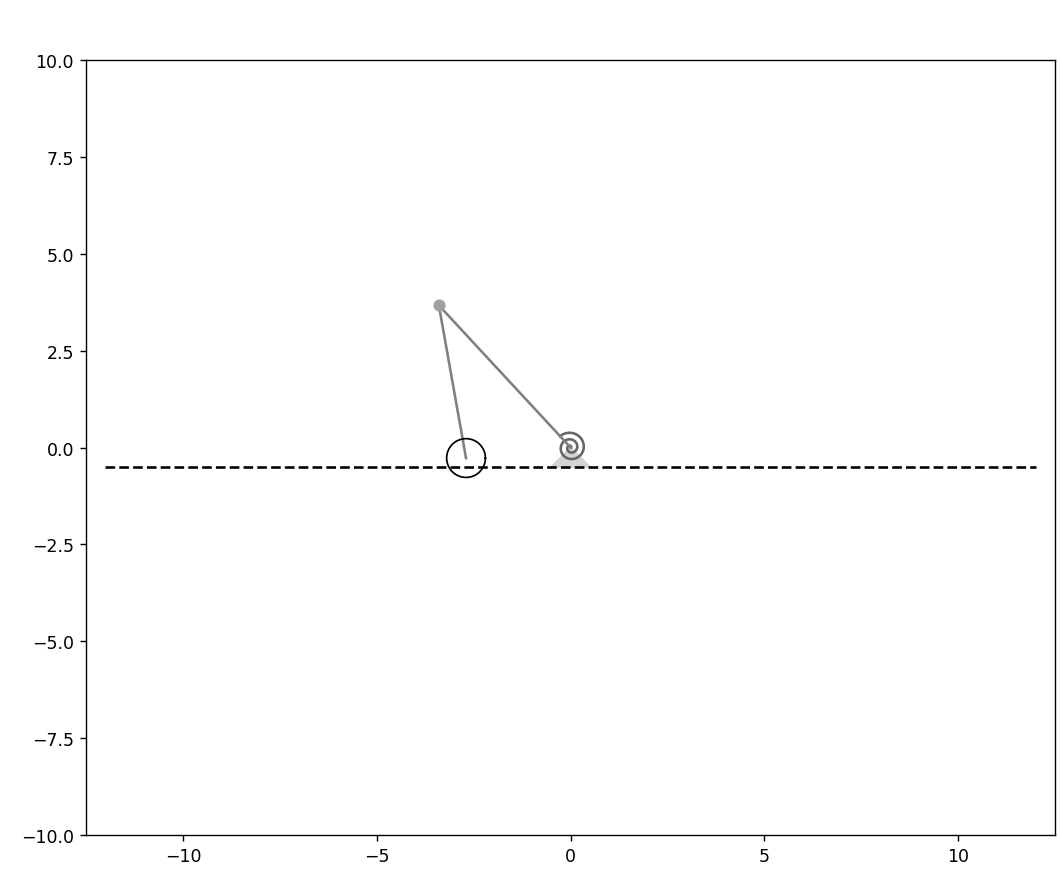
**Draw\_Spring.set\_data(X\_SpiralSpr, Y\_SpiralSpr)**

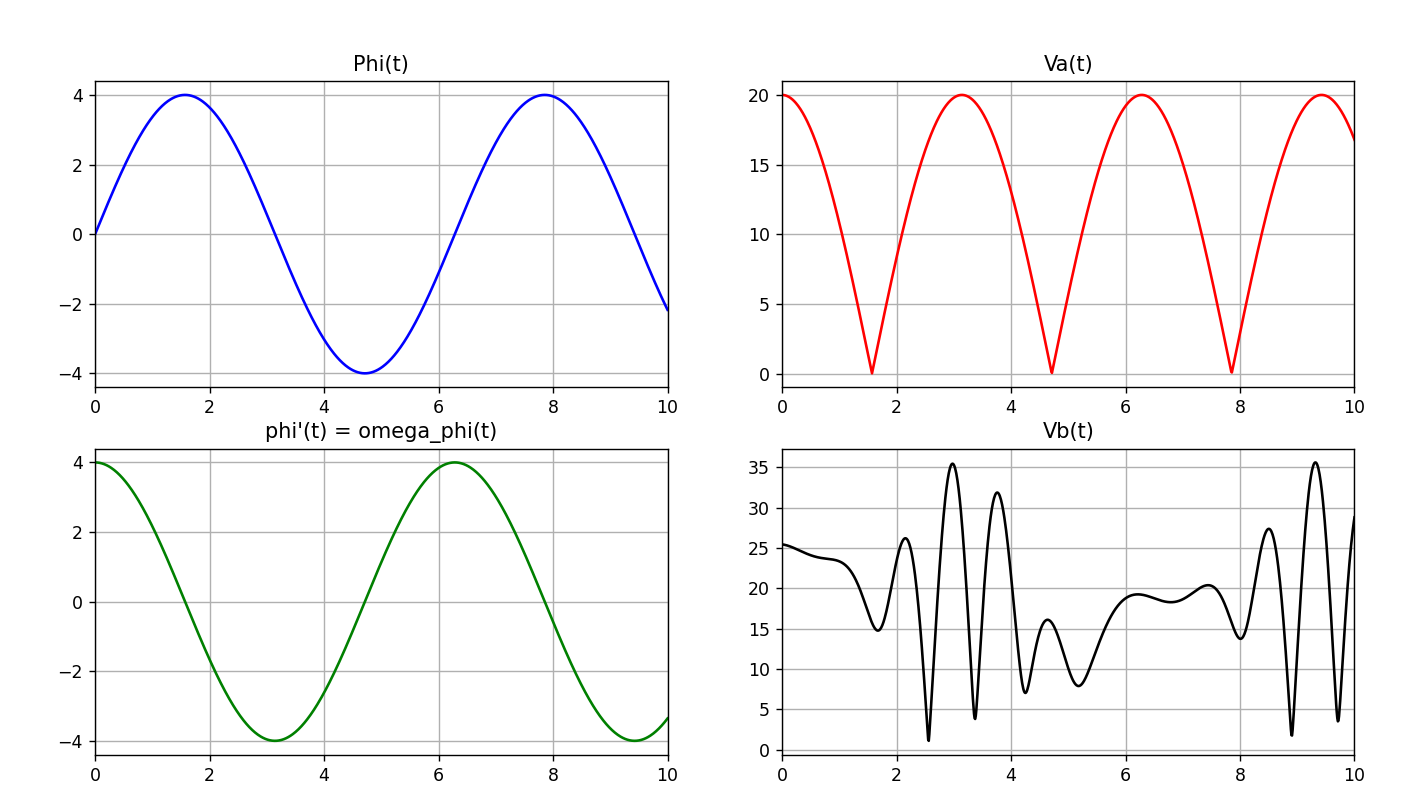
**return [PointB, Draw\_OA, Draw\_Spring, Draw\_AB, PointA, Draw\_Circle]**

**anima = FuncAnimation(fig, update, frames=Steps, interval=1)**

**plt.show()**

**Результат работы:**



****