ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

**ОТЧЕТ**

**О ВЫПОЛОНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ**

**«ДИНАМИКА СИСТЕМЫ»**

**ПО ДИСЦИПЛИНЕ «Основы Компьютерного Моделирования Математических Систем»**

**ВАРИАНТ ЗАДАНИЯ №29**

Выполнил(а) студент группы М8О-215Б-23

Самарский Я.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

Проверил и принял

Ст. преп. каф. 802 Волков Е.В.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

подпись, дата

с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

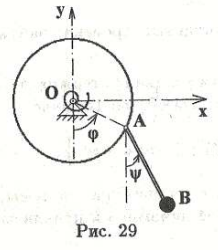
Москва, 2024

**Вариант №29**

**Задание:**

Численно решить дифференциальные уравнения движения механической системы в среде Octave (или Matlab), сделать задание №12 курсовой и построить анимацию движения системы.

**Механическая система:**

****

**Текст программы**

Основная:

from matplotlib.patches import Circle

import matplotlib.pyplot as plt

import numpy as np

import sympy

from matplotlib.animation import FuncAnimation

from sympy.utilities.lambdify import implemented\_function

from scipy.integrate import odeint

ROT\_OFFSET = -np.pi / 2

M1 = 0.1

M2 = 0.05

R = 1

L = 1

G = 9.8

C = 100

K = 0

PHI\_M = 0

def func(y, t, m1, m2, r, l, g, c, k, phi\_m):

    # y = [phi, psi, phi', psi']

    dy = np.zeros\_like(y)

    dy[0] = y[2]

    dy[1] = y[3]

    # a11 \* phi'' + a12 \* psi'' = b1

    # a21 \* phi'' + a22 \* psi'' = b2

    a11 = ((m1 / 2) + m2) \* r

    a12 = m2 \* l \* np.cos(y[0] - y[1])

    b1 = -m2 \* l \* y[3] \*\* 2 \* np.sin(y[0] - y[1]) - m2 \* g \* np.sin(y[0]) - (c / r) \* (y[0] - phi\_m) + (k/r) \* (dy[1] - dy[0])

    a21 = np.cos(y[0] - y[1])

    a22 = l / r

    b2 = y[2] \*\* 2 \* np.sin(y[0] - y[1]) - (g / r) \* np.sin(y[1]) - k \* (y[3] - y[2]) / (m2 \* r \* l)

    dy[2] = (b1 \* a22 - b2 \* a12) / (a11 \* a22 - a12 \* a21)

    dy[3] = (a11 \* b2 - a21 \* b1) / (a11 \* a22 - a12 \* a21)

    return dy

PHI\_0 = 0

PSI\_0 = 10

D\_PHI\_0 = 0

D\_PSI\_0 = 0

steps = 1001

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

y0 = np.array([PHI\_0, PSI\_0, D\_PHI\_0, D\_PSI\_0])

y = odeint(func, y0, t, args=(M1, M2, R, L, G, C, K, PHI\_M))

phi = y[:,0]

psi = y[:,1]

d\_phi = y[:,2]

d\_psi = y[:,3]

dd\_phi = np.zeros\_like(t)

dd\_psi = np.zeros\_like(t)

r\_ox = np.zeros\_like(t)

r\_oy = np.zeros\_like(t)

for i in range(len(t)):

    local\_dy = func(y[i], t[i], M1, M2, R, L, C, K, G, PHI\_M)

    dd\_phi[i] = local\_dy[2]

    dd\_psi[i] = local\_dy[3]

    r\_ox[i] = M2 \* (R \* (dd\_phi[i] \* np.cos(phi[i]) - (dd\_phi[i] \*\* 2) \* np.sin(phi[i])) + L \* (dd\_psi[i] \* np.cos(phi[i]) - (d\_psi[i] \*\* 2) \* np.sin(psi[i])))

    r\_oy[i] = M2 \* (R \* (dd\_phi[i] \* np.sin(phi[i]) - (dd\_phi[i] \*\* 2) \* np.cos(phi[i])) + L \* (dd\_psi[i] \* np.sin(phi[i]) + (d\_psi[i] \*\* 2) \* np.cos(psi[i]))) + (M1 + M2) \* G

fig = plt.figure(figsize=[17,9])

ax1 = fig.add\_subplot(4, 2, 2)

ax1.plot(t, phi)

ax1.set\_title("phi(t)")

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

ax2.plot(t, psi)

ax2.set\_title("psi(t)")

ax2 = fig.add\_subplot(4, 2, 6)

ax2.plot(t, r\_ox)

ax2.set\_title("r\_ox(t)")

ax2 = fig.add\_subplot(4, 2, 8)

ax2.plot(t, r\_oy)

ax2.set\_title("r\_oy(t)")

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

subplot.axis('equal')

subplot.set\_xlim((-3, 3))

subplot.set\_ylim((-3, 3))

R\_MAIN = 1

R\_B = 0.1

L = 1.5

x\_a = np.cos(ROT\_OFFSET + phi)

y\_a = np.sin(ROT\_OFFSET + phi)

x\_b = x\_a + np.cos(ROT\_OFFSET + psi) \* L

y\_b = y\_a + np.sin(ROT\_OFFSET + psi) \* L

# Центральная окружность

main\_circle = Circle((0, 0), R\_MAIN, fill=False)

subplot.add\_patch(main\_circle)

# Прямые AB и AO

line\_ab = subplot.plot([x\_a[0], x\_b[0]], [y\_a[0], y\_b[0]], marker='o')[0]

line\_oa = subplot.plot([0, x\_a[0]], [0, y\_a[0]], 'k--')[0]

# Точка B

b\_circle = Circle((x\_b[0], y\_b[0]), R\_B, fill=True, color='b', zorder=2)

b\_circle\_patch = subplot.add\_patch(b\_circle)

# Спираль

R\_SPIRAL\_1 = 0.05

R\_SPIRAL\_2 = 0.2

radians = np.linspace(0, 2 \* np.pi, 50)

radiuses = np.linspace(R\_SPIRAL\_1, R\_SPIRAL\_2, 50)

X\_n = np.cos(ROT\_OFFSET + radians) \* radiuses

Y\_n = np.sin(ROT\_OFFSET + radians) \* radiuses

spiral = subplot.plot(X\_n, Y\_n)[0]

def Rot2D(X, Y, phi):

    # Поворот двумерной ДСК с помощью матрицы поворота

    X\_r = X \* np.cos(phi) - Y \* np.sin(phi)

    Y\_r = X \* np.sin(phi) + Y \* np.cos(phi)

    return X\_r, Y\_r

def update(i):

    print(i)

    line\_ab.set\_data([x\_a[i], x\_b[i]], [y\_a[i], y\_b[i]])

    line\_oa.set\_data([0, x\_a[i]], [0, y\_a[i]])

    b\_circle.set\_center((x\_b[i], y\_b[i]))

    new\_X\_n, new\_Y\_n = Rot2D(X\_n, Y\_n, phi[i])

    spiral.set\_data(new\_X\_n, new\_Y\_n)

    return line\_ab, line\_oa, b\_circle\_patch, spiral

animation = FuncAnimation(fig, update, frames=steps, interval=10, blit=True)

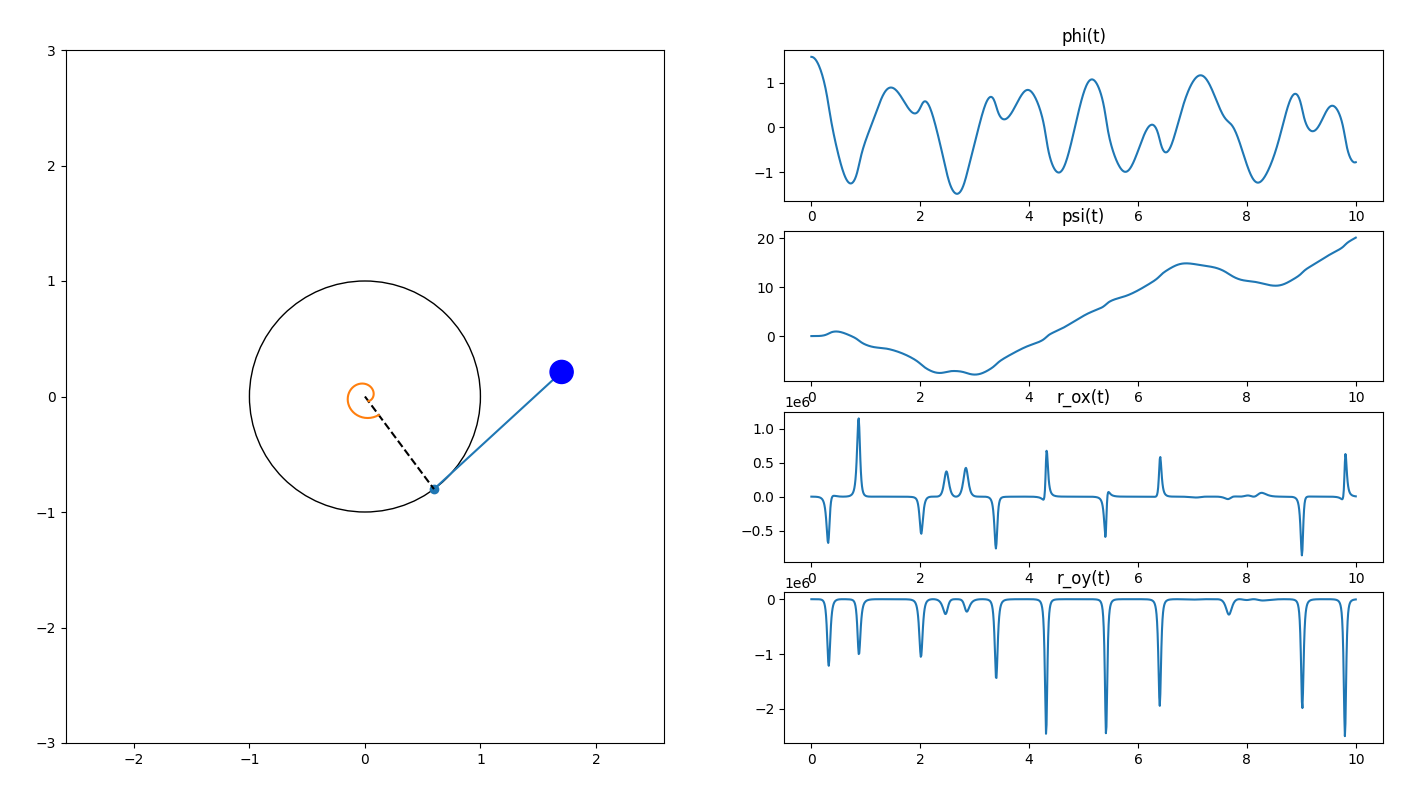
plt.show()

**Результат работы:**

**Тест 1**

M1 = 0.1, M2 = 0.05, R = 1, L = 1, G = 9.8, C = 1, K = 0, PHI\_M = 0

y0 = [pi/2 0 0 0];

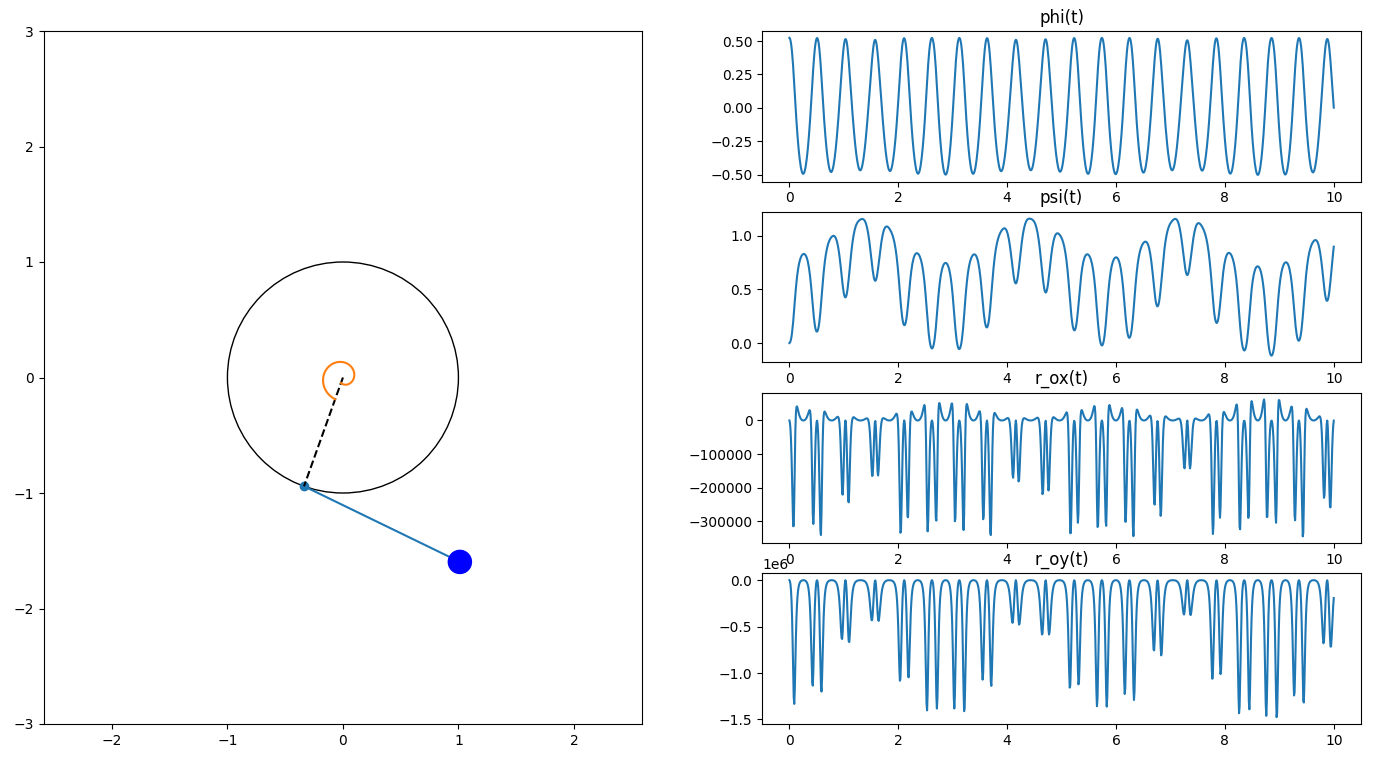
****

Результат: диск колеблется, точка B вращается вокруг точки A

**Тест 2**

M1 = 0.1, M2 = 0.05, R = 1, L = 1, G = 9.8, C = 10, K = 0, PHI\_M = 0

y0 = [pi/6 0 0 0];

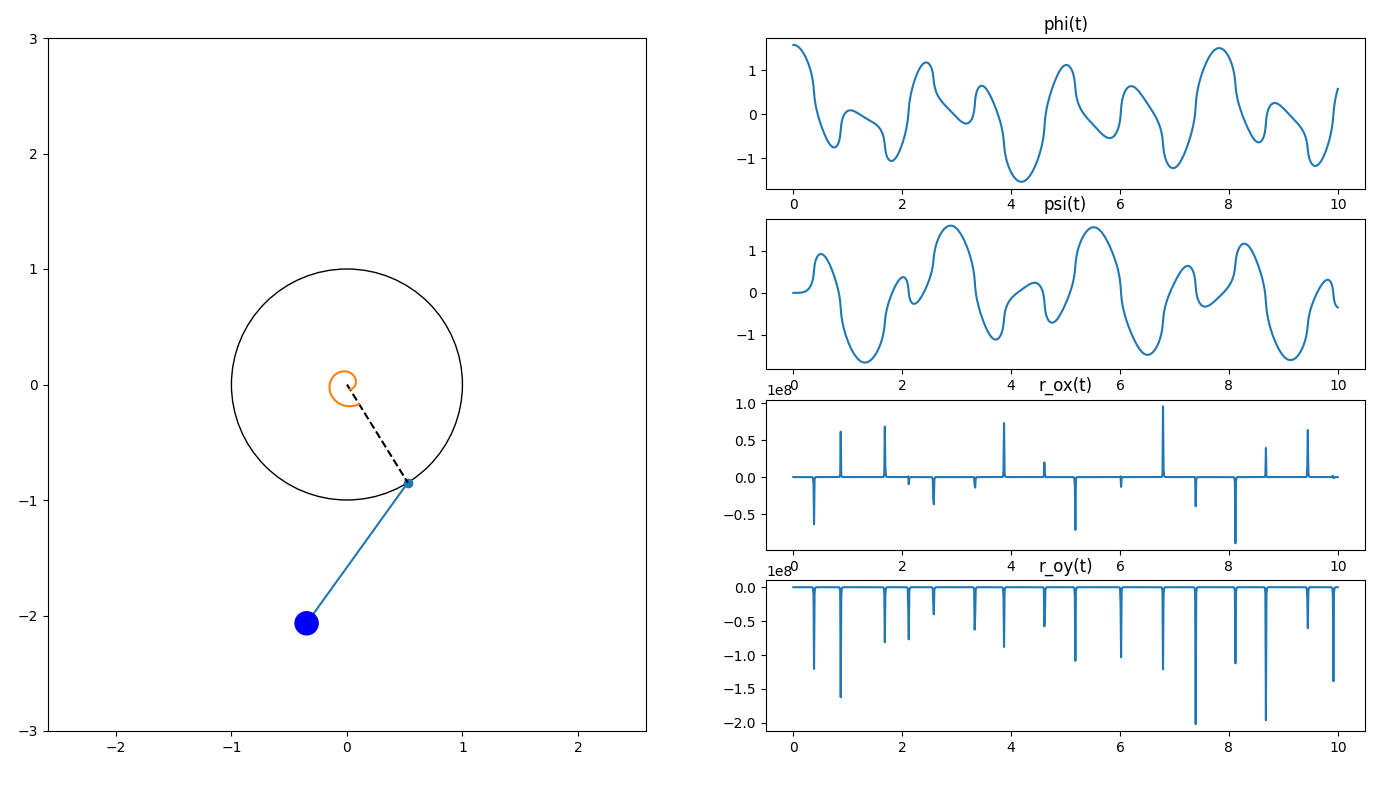
****

Результат: диск быстро колеблется, точка B тоже быстро колеблется

**Тест 3**

M1 = 0.1, M2 = 1, R = 1, L = 1, G = 9.8, C = 1, K = 0, PHI\_M = 0

y0 = [pi/2 0 0 0];

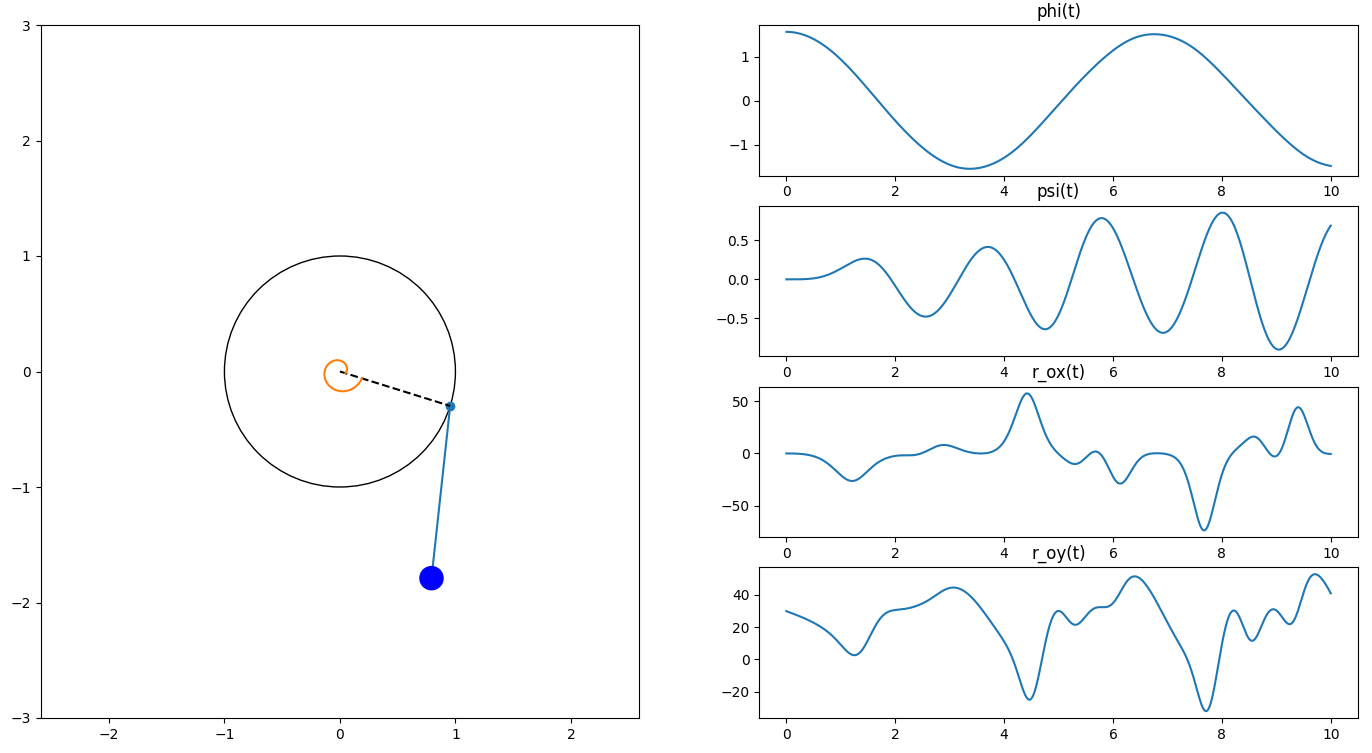
****

Результат: диск колеблется с некоторыми подёргиваниями, точка B достаточно медленно качается

**Тест 4**

M1 = 3, M2 = 0.05, R = 1, L = 1, G = 9.8, C = 1, K = 0, PHI\_M = 0

y0 = [pi/2 0 0 0];

****

Результат: диск очень медленно вращается, точка B колеблется, быстро следуя за точкой A.

**Вывод:**

В третьей лабораторной работе используя численные методы, опираясь на физические параметры, мной были рассчитаны углы вращения элементов и координаты для графиков. Используя алгоритм из второй лабораторной работы, была отрисована механическая система, а также добавлены графики углов phi и psi и проекций реакции оси диска.