

**Московский авиационный институт
(Национальный исследовательский университет)**

Институт: «Информационные технологии и прикладная математика»

Кафедра: 806 «Вычислительная математика и программирование»

Дисциплина: «Теоретическая механика»

**Лабораторная работа № 2
по курсу «Теоретическая механика»
Анимация системы.**

Студент: Попов И. П.

Группа: М80-206Б-20

Преподаватель: Сухов Е.А.

Дата: 15.12

Оценка:

Москва, 2021

Вариант №19

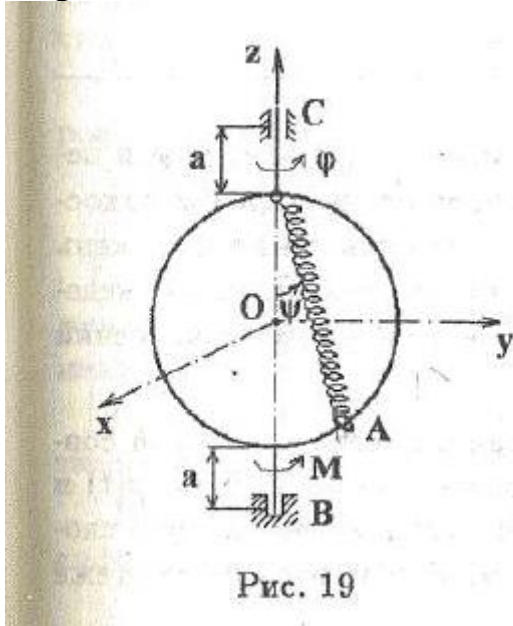


Рис. 19

Задание:

Реализовать анимацию движения механической системы используя язык программирования Python.

Листинг программы:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import sympy as sp
import math

# Ввод переменной t и радиусов необходимых окружностей + ввод угла поворота шариков
t = sp.Symbol('t')
R = 2

# Построение графика и подграфика с выравниванием осей
fig = plt.figure(figsize=(17, 8))
ax1 = fig.add_subplot(1, 2, 1)
ax1.axis('equal')

phi = np.linspace(0, 2 * math.pi, 500)
psi = np.linspace(-math.pi/2, 0, 500)

conline, = ax1.plot([sp.sin(2*psi[0]) * R * np.abs(sp.cos(phi[0]))], 0), [-1, R], 'black')
```

```
P, = ax1.plot(sp.sin(2*psi[0]) * R * np.abs(sp.cos(phi[0])), sp.cos(2*psi[0]) * R, marker='o', color='black')
Circ, = ax1.plot(R * np.abs(sp.cos(phi[0])) * np.cos(phi), R * np.sin(phi), 'black')
```

#Доп графики

```
ax2 = fig.add_subplot(4, 2, 2)
```

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

```
x = sp.sin(t)+2
```

```
y = sp.cos(t+math.pi)+2
```

```
Vx = sp.diff(x, t)
```

```
Vy = sp.diff(y,t)
```

```
T = np.linspace(0, 10, 1000)
```

```
VX = np.zeros_like(T)
```

```
VY = np.zeros_like(T)
```

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

```
    VX[i] = sp.Subs(Vx, t, T[i])
```

```
    VY[i] = sp.Subs(Vy, t, T[i])
```

```
ax2.plot(T, VX)
```

```
ax2.set_xlabel('T')
```

```
ax2.set_ylabel('VX')
```

```
ax3 = fig.add_subplot(4, 2, 4)
```

```
ax3.plot(T, VY)
```

```
ax3.set_xlabel('T')
```

```
ax3.set_ylabel('VY')
```

```
def anima(i):
```

```
    P.set_data(sp.sin(2*psi[i]) * R * np.abs(sp.cos(phi[i])), sp.cos(2*psi[i]) * R)
```

```
    online.set_data([sp.sin(2*psi[i]) * R * np.abs(sp.cos(phi[i])), 0], [sp.cos(2*psi[i]) * R, R])
```

```
    Circ.set_data(R * np.abs(sp.cos(phi[i])) * np.cos(phi), R * np.sin(phi))
```

```
    return Circ, P, online
```

```
anim = FuncAnimation(fig, anima, frames=500, interval=1, blit=True)
```

```
plt.show()
```

Результат работы

Figure 1

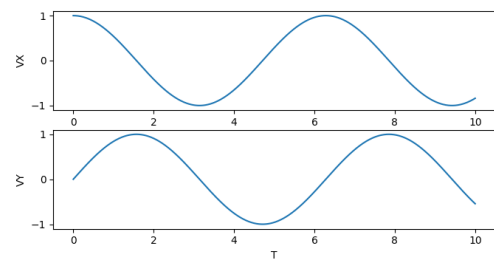
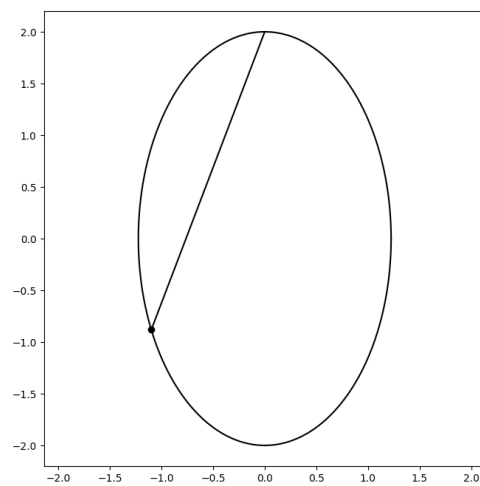
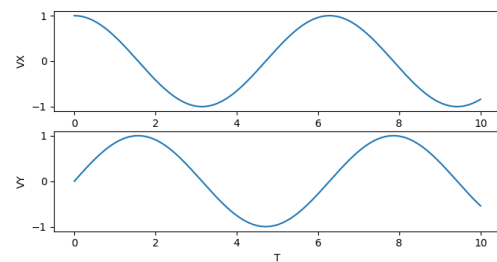
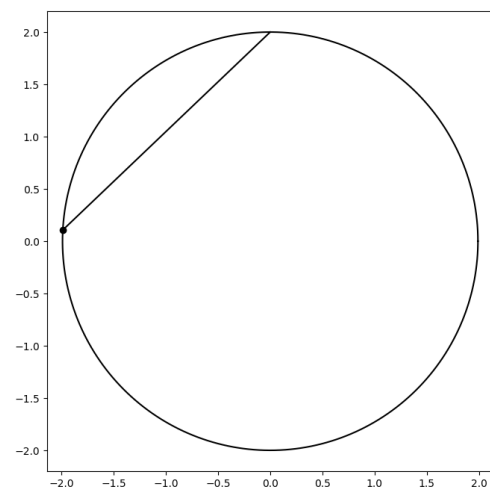


Figure 1



Вывод

В процессе выполнения данной лабораторной работы я использовал полученные на курсе «Теоретическая механика» знания для написания программы анимации системы из своего варианта задания.

Хотелось бы отметить удобство построения дополнительных графиков, даже в моей работе они могут наглядно представить процессы изменения величин, что уж говорить об их полезности в более серьезных проектах.