

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

ОТЧЕТ
О ВЫПОЛНЕНИИ ЛАБОРАТОРНОЙ РАБОТЫ
«АНИМАЦИЯ СИСТЕМЫ»
ПО ДИСЦИПЛИНЕ «ТЕОРЕТИЧЕСКАЯ МЕХАНИКА И
ОСНОВЫ КОМПЬЮТЕРНОГО МОДЕЛИРОВАНИЯ»
ВАРИАНТ ЗАДАНИЯ № 2

Выполнил(а) студент группы М8О-208Б-21

Белоносов К. _____
подпись, дата

Проверил и принял

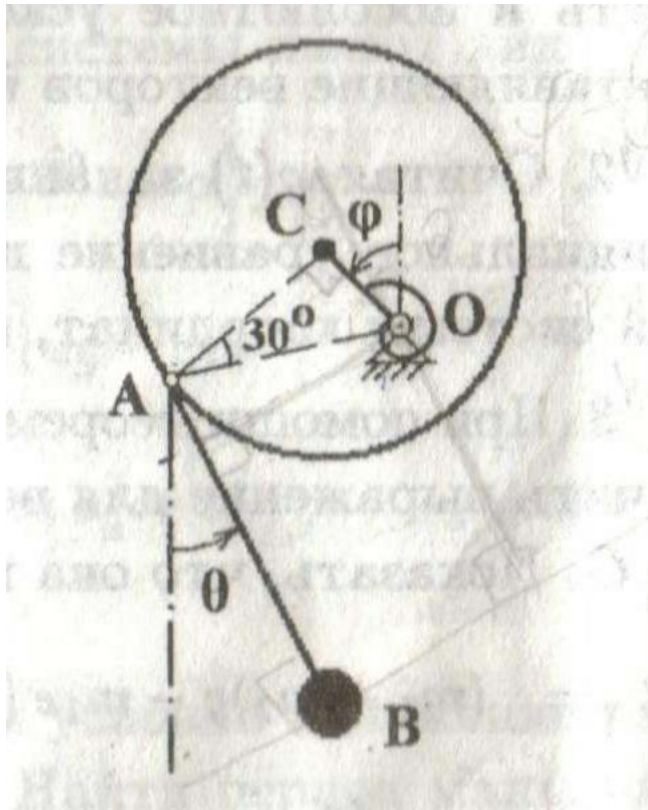
Зав. каф. 802, Бардин Б.С. _____
подпись, дата

с оценкой _____

Москва, 2022

Задание: построить анимацию движения системы с помощью Python.

Механическая система:



Текст программы:

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

```
Steps = 1001
```

```
t_fin = 20
```

```
t = np.linspace(0, t_fin, Steps)
```

```
R = 1
```

```
phi = 2*np.sin(1.7*t) + 5*np.cos(1.2*t)
```

```
ksi = 4*np.sin(2*t) + 5*np.cos(3*t)
```

```
psi = np.linspace(0, 6.28, 20)
```

```
C = 1
```

```
R1 = 0.2
```

```
R2 = R/(3**0.5)/2
```

```
l = 1
```

```
X_Wheel = R*np.sin(psi)
```

```
Y_Wheel = R*np.cos(psi)
```

```
X_C = (R/(3**0.5))*np.sin(phi)
```

```
Y_C = (R/(3**0.5))*np.cos(phi)
```

```
X_A = X_C + R*np.sin(phi - 1.57)
```

```
Y_A = Y_C + R*np.cos(phi - 1.57)
```

```
X_B = X_A + l*np.cos(ksi)
```

```
Y_B = Y_A + l*np.sin(ksi)
```

```
thetta = np.linspace(0, C*6.28-phi[0], 100)
```

```
X_SpiralSpr = -(R1 + thetta*(R2-R1)/thetta[-1])*np.sin(thetta)
```

```
Y_SpiralSpr = (R1 + thetta*(R2-R1)/thetta[-1])*np.cos(thetta)
```

```
fig = plt.figure(figsize=[15, 7])
```

```
ax = fig.add_subplot(1, 1, 1)
```

```
ax.axis('equal')
```

```
ax.set(xlim=[-6, 6], ylim=[-6, 6])
```

```
Circ = ax.plot(X_C[0]+R * np.cos(phi), Y_C[0]+R * np.sin(phi), 'yellow')[0]
```

```
Point_O = ax.plot(0, 0, marker='o')[0]
```

```
Point_C = ax.plot(X_C[0], Y_C[0], marker='o')[0]
```

```
Point_A = ax.plot(X_A[0], Y_A[0], marker='o')[0]
```

```
Point_B = ax.plot(X_B[0], Y_B[0], marker='o', markersize=20)[0]
```

```
Line_CO = ax.plot([0, X_C[0]], [0, Y_C[0]], 'blue')[0]
```

```
Line_AB = ax.plot([X_A[0], X_B[0]], [Y_A[0], Y_B[0]])[0]
```

```
Drawed_SpiralSpring = ax.plot(X_SpiralSpr, Y_SpiralSpr)[0]
```

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

```
    Point_C.set_data(X_C[i], Y_C[i])
```

```
    Point_A.set_data(X_A[i], Y_A[i])
```

```
    Point_B.set_data(X_B[i], Y_B[i])
```

```
    Line_AB.set_data([X_A[i], X_B[i]], [Y_A[i], Y_B[i]])
```

```
    Circ.set_data(X_C[i] + R * np.cos(phi), Y_C[i] + R * np.sin(phi))
```

```
    Line_CO.set_data([0, X_C[i]], [0, Y_C[i]])
```

```
    theta = np.linspace(0, C * 6.28 - phi[i], 100)
```

```
    X_SpiralSpr = -(R1 + theta * (R2 - R1) / theta[-1]) * np.sin(theta)
```

```
    Y_SpiralSpr = (R1 + theta * (R2 - R1) / theta[-1]) * np.cos(theta)
```

```
    Drawed_SpiralSpring.set_data(X_SpiralSpr, Y_SpiralSpr)
```

```
    return [Point_C, Point_A, Point_B, Circ, Line_CO, Line_AB, Drawed_SpiralSpring]
```

```
anim = FuncAnimation(fig, anima, frames=len(t), interval=3)
```

```
plt.show()
```

Результаты работы:

