

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Национальный исследовательский университет ИТМО»  
(Университет ИТМО)

Международный научно-образовательный центр  
Физикиnanoструктур

Практическая работа № 3  
по дисциплине  
*«Имитационное моделирование робототехнических систем»*

по теме:  
Инициализация пассивного механизма в среде MuJoCo

Студент:  
*Группа № R4134c*

Д.С. Черных  
И.О. Фамилия

Предподаватель:  
*Инженер, младший научный сотрудник*

Е.А. Ракшин  
И.О. Фамилия

Санкт-Петербург 2025

**Цель работы:** исследовать пассивный механизм, создать симуляцию и сделать выводы.

**Оборудование и программные среды:** ПК, Visual studio insider.

**Задание:**

Составить скрипт на Python для моделирования пассивного механизма, согласно варианту.

**Ход работы**

Согласно варианту в ходе работы будет рассматриваться сухожильный плоский механизм со следующими параметрами:

$$R_1 = 0,018 \text{ м}, \quad R_2 = 0,026 \text{ м}, \quad a = 0,034 \text{ м}, \quad b = 0,054 \text{ м}, \\ c = 0,084 \text{ м}$$

Схема механизма представлена на рисунке 1.

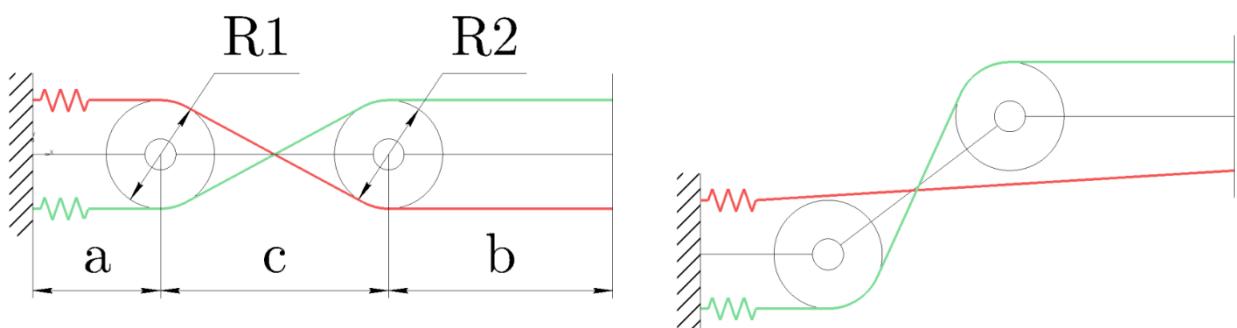


Рисунок 1 – Заданная схема механизма

Программный код решение представлен ниже:

```
# tendon_simple.py
import mujoco
import mujoco_viewer
import numpy as np

# Параметры
LEN_A, LEN_B, LEN_C = 0.034, 0.084, 0.054
R_R1, R_R2 = 0.018, 0.026

def make_xml():
    return f"""
<mujoco model="TendonCross">
    <option timestep="0.002" gravity="0 0 0"/>
    <worldbody>
        <geom name="floor" type="plane" size="2 2 0.01" rgba="0.6 0.8 0.6 1"/>
        <site name="floor_red" pos="-0.03 0 0" rgba="1 0 0 1"/>
        <site name="floor_green" pos=" 0.03 0 0" rgba="0 1 0 1"/>
        <camera name="cam" pos="0.3 0.8 0.3" xyaxes="-1 0 0 0 0 1"/>

        <body name="link_a" pos="0 0 0">
            <geom type="capsule" size="0.005 {LEN_A/2:.3f}" fromto="0 0 0 0 {LEN_A}"/>
            <body name="R1" pos="0 0 {LEN_A}">
```

```

<joint name="R1_joint" type="hinge" axis="0 1 0"/>
<geom type="cylinder" size="{R_R1} 0.01" rgba="0.8 0.8 0.8 1"
euler="90 0 0"/>
    <site name="R1_top" pos="{R_R1} 0 0" rgba="0 1 0 1"/>
    <site name="R1_bot" pos="-{R_R1} 0 0" rgba="1 0 0 1"/>

    <body name="link_b" pos="0 0 0">
        <geom type="capsule" size="0.005 {LEN_B/2:.3f}" fromto="0 0 0 0
{LEN_B}"/>
        <body name="R2" pos="0 0 {LEN_B}">
            <joint name="R2_joint" type="hinge" axis="0 1 0"/>
            <geom type="cylinder" size="{R_R2} 0.01" rgba="0.8 0.8 0.8 1"
euler="90 0 0"/>
                <site name="R2_top" pos="{R_R2} 0 0" rgba="1 0 0 1"/>
                <site name="R2_bot" pos="-{R_R2} 0 0" rgba="0 1 0 1"/>

                <body name="link_c" pos="0 0 0">
                    <geom type="capsule" size="0.005 {LEN_C/2:.3f}" fromto="0 0 0 0
0 {LEN_C}"/>
                    <body name="right_wall" pos="0 0 {LEN_C}">
                        <geom type="box" size="0.1 0.05 0.005" rgba="0.5 0.5 0.8 1"/>
                        <site name="wall_red" pos="-0.03 0 0.0025" rgba="0 1 0 1"/>
                        <site name="wall_green" pos=" 0.03 0 0.0025" rgba="1 0 0 1"/>
                    </body>
                </body>
            </body>
        </body>
    </body>
</worldbody>

<tendon>
    <spatial name="tendon_red" width="0.003" rgba="1 0 0 0.9">
        <site site="floor_red"/><site site="R1_bot"/><site site="R2_top"/><site
site="wall_green"/>
    </spatial>
    <spatial name="tendon_green" width="0.003" rgba="0 1 0 0.9">
        <site site="floor_green"/><site site="R1_top"/><site
site="R2_bot"/><site site="wall_red"/>
    </spatial>
</tendon>
</mujoco>
".encode()

# Запуск
model = mujoco.MjModel.from_xml_string(make_xml())

```

```

data = mujoco.MjData(model)
viewer = mujoco_viewer.MujocoViewer(model, data, width=800,
height=600)

# Индексы
addr_R1 = model.jnt_qposadr[mujoco.mj_name2id(model,
mujoco.mjtObj.mjOBJ_JOINT, "R1_joint")]
addr_R2 = model.jnt_qposadr[mujoco.mj_name2id(model,
mujoco.mjtObj.mjOBJ_JOINT, "R2_joint")]

t = 0.0
while viewer.is_alive:
    data.qpos[addr_R1] = 0.6 * np.sin(t)
    data.qpos[addr_R2] = -0.6 * np.sin(t)
    t += 0.02
    mujoco.mj_forward(model, data)
    viewer.render()

viewer.close()
print("Готово.")

```

На рисунке 1 представлен результат работы данного кода.

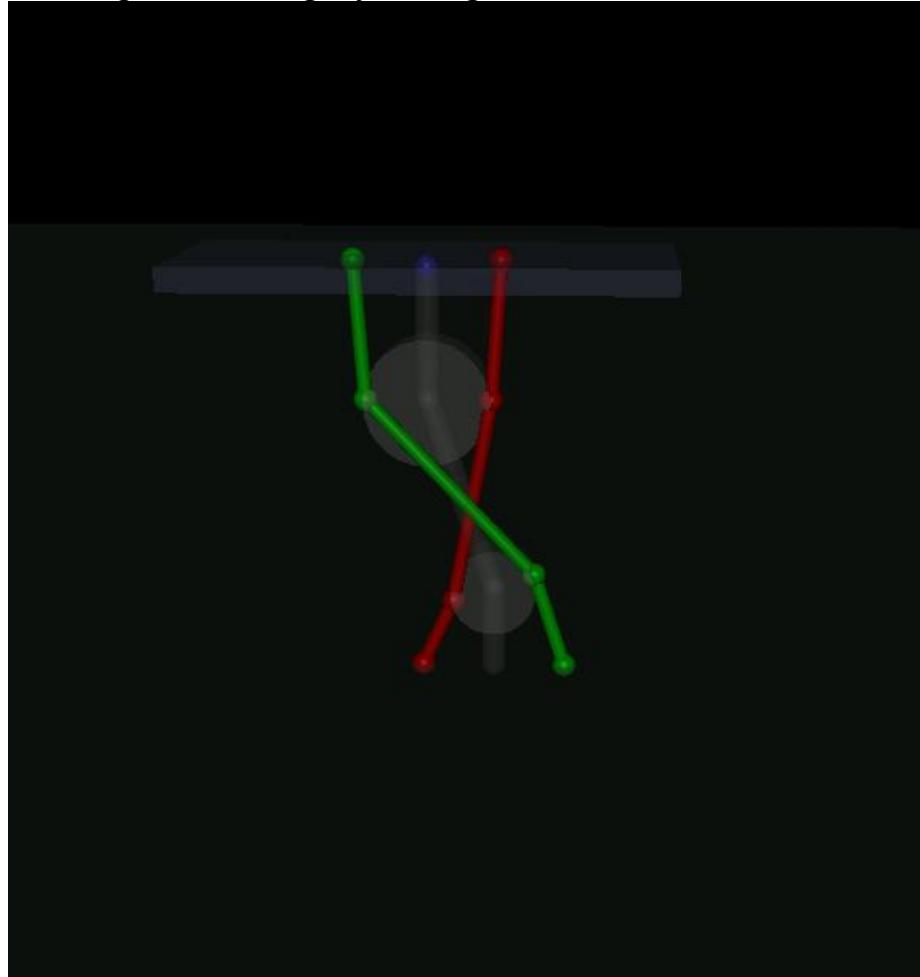


Рисунок 1 – Результат работы кода

## **Вывод**

В ходе работы был разработан код симуляции плоского сухожильного механизма в программной среде MiJoso.