МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

(Университет ИТМО)

Факультет

Систем управления и робототехники

ОТЧЕТ

по лабораторной работе № 3

по дисциплине

«Имитационное моделирование робототехнических систем»

Студент:

Группа № R4137c А. Альмахмуд

Преподаватели:

профессор, И.И. Борисов

инженер, ассистент, Е.А. Ракшин

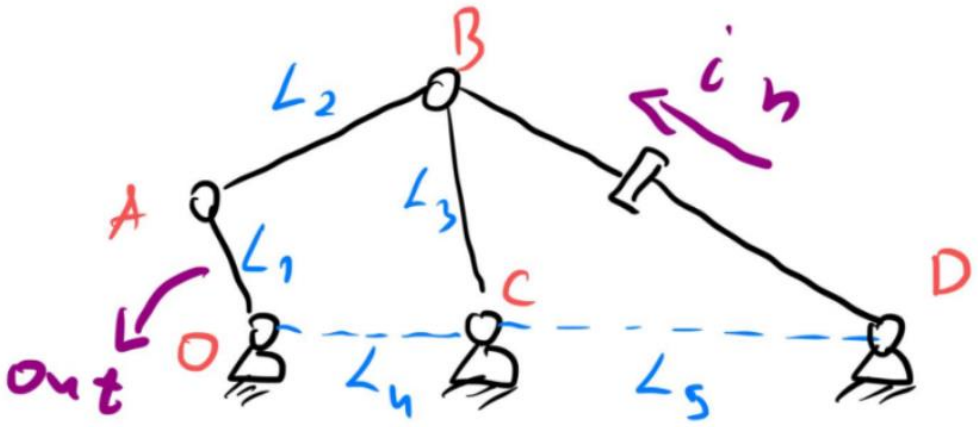
Санкт-Петербург 2025

1- ОПИСАНИЕ ЗАДАНИЯ

Цель работы: выполнить имитационное моделирование механизма замкнутой кинематики, обеспечивающей движение колена робота Optimus.

1.1 Исходные данные

Была получена система, представленная



Рисунке 1 Схема системы

Как показано на Рисунке 1, движение осуществляется за счет мотора,

установленного между сочленениями B и D.

Исходные данные для выполнения задания представлены в Таблице 1.

Таблица 1 — Исходные данные

Параметр	L1 (m)	L2 (m)	L3 (m)	L4 (m)	L5 (m)
Значение	0.071	0.0923	0.1065	0.071	0.355

2- Колено робота Optimus

2.1 Построение модели

xml код модели

```
<mujoco model="optimus-knee">
    <compiler angle="radian"/>

    <default>
        <joint limited="false"/>
        <geom size="0.0025 0 0" type="capsule" contype="0" conaffinity="0"
rgba="0.9 0.7 0.1 1"/>
        <site size="0.0025 0.005 0.005"/>
        <default class="base">
            <joint limited="true" range="-10 20"/>
            <geom size="0.003 0 0" type="sphere" rgba="0.9 0.1 0.1 1"/>
        </default>
    </default>

    <asset>
        <texture type="skybox" builtin="gradient" rgb1="0.3 0.5 0.7" rgb2="0 0 0"
width="64" height="384"/>
        <texture type="2d" name="texplane" builtin="checker" mark="cross"
rgb1="0.2 0.3 0.4" rgb2="0.1 0.15 0.2" markrgb="0.8 0.8 0.8" width="512"
height="512"/>
        <material name="matplane" texture="texplane" texuniform="true"
texrepeat="3 3" reflectance="0.5"/>
    </asset>

    <worldbody>
        <body>
            <geom name="O" class="base"/>

            <body>
                <joint pos="0 0 0" axis="0 1 0"/>
                <geom name="L1" size="0.0025 0.0350" pos="0 0 0.0350" quat="0 1 0
0"/>

                <body pos="0 0 0.071">
                    <joint pos="0 0 0" axis="0 1 0"/>
                    <geom name="L2" size="0.0025 0.04600" pos="0 0 0.04600"
quat="0 1 0 0"/>
```

```
                    <geom name="A" class="base"/>
                    <site name="A" pos="0 0 0.0923"/>
                </body>
            </body>
        </body>

        <body pos="0.071 0 0">
            <geom name="C" class="base"/>

            <body>
                <joint class="base" pos="0 0 0" axis="0 1 0" range="-0.174533
0.349066"/>
                <geom name="L3" size="0.0025 0.05100" pos="0 0 0.05100" quat="0 1
0 0"/>
                <geom name="B" class="base" pos="0 0 0.1065"/>
                <geom class="base" pos="0 0 0.1065"/>
                <site name="B" pos="0 0 0.1065"/>
            </body>
        </body>

        <body pos="0.426 0 0" quat="0.793353 0 -0.608761 0">
            <geom name="D" class="base"/>
            <site name="D" pos="0 0 0"/>
        </body>
    </worldbody>

    <equality>
        <connect site1="B" site2="A"/>
    </equality>

    <actuator>
        <general cranksite="B" slidersite="D" ctrlrange="0 3"
cranklength="0.309"/>
    </actuator>
</mujoco>
```

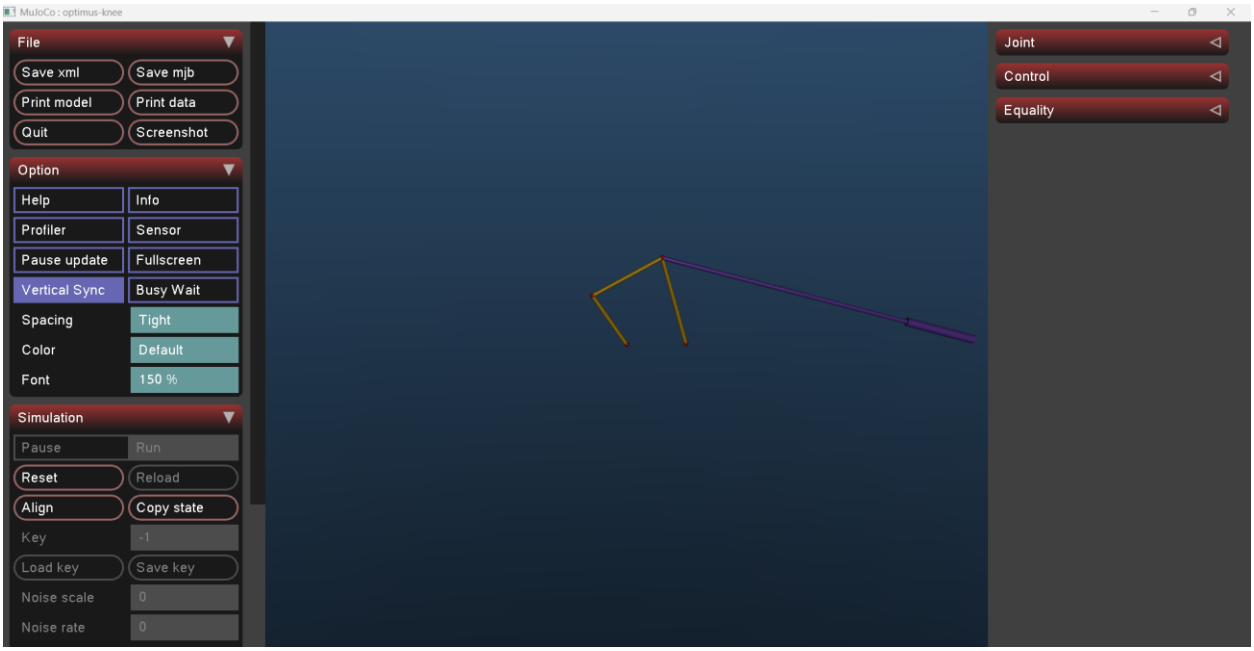
2.2 Симуляция

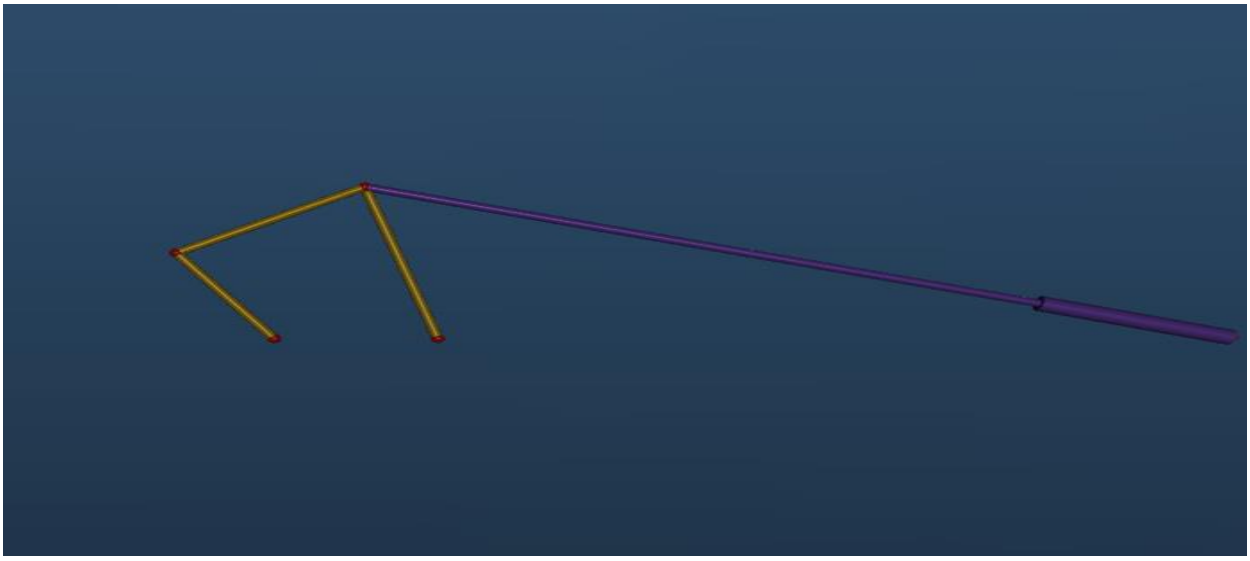
Код python для симуляции системы

```python
import mujoco
import mujoco.viewer
import numpy as np
import matplotlib.pyplot as plt

# ------------------------------
# Load the model
# ------------------------------
xml_path = r"C:\Users\User1\Desktop\New folder\New folder\knee.xml"
model = mujoco.MjModel.from_xml_path(xml_path)
data = mujoco.MjData(model)

# ------------------------------
# IDs for actuator and joint
# ------------------------------
joint_id_A = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_JOINT, "A")
actuator_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_ACTUATOR, "linear_motor")
ctrl_range = model.actuator_ctrlrange[actuator_id]

# Arrays to store simulation data
angles = []
displacements = []

# ------------------------------
# Launch viewer
# ------------------------------
with mujoco.viewer.launch_passive(model, data) as viewer:

    t_end = 5.0
    dt = model.opt.timestep
    num_steps = int(t_end / dt)

    for step in range(num_steps):
        # Calculate linear displacement
        displacement = ctrl_range[0] + (ctrl_range[1] - ctrl_range[0]) * (step / num_steps)
        data.ctrl[actuator_id] = displacement

        # Step the simulation
        mujoco.mj_step(model, data)

        # Record joint angle
        angle = data.qpos[joint_id_A]
```

```
        angles.append(np.degrees(angle))
        displacements.append(displacement)

        # Update viewer
        if viewer.is_running():
            viewer.sync()
        else:
            break

# ------------------------------
# Plot the results
# ------------------------------
angles = np.array(angles)
displacements = np.array(displacements)

plt.figure(figsize=(8,6))
plt.plot(displacements, angles, color='blue', linewidth=2, label="OA Joint Angle")
plt.xlabel("Linear Displacement (m)")
plt.ylabel("Joint Rotation Angle (degrees)")
plt.title("Optimus Knee Simulation: OA Angle vs Displacement")
plt.grid(True)
plt.legend()
plt.show()
```

2.3 Модель колена робота Optimus в MuJoCo viewer





2.4 Графики

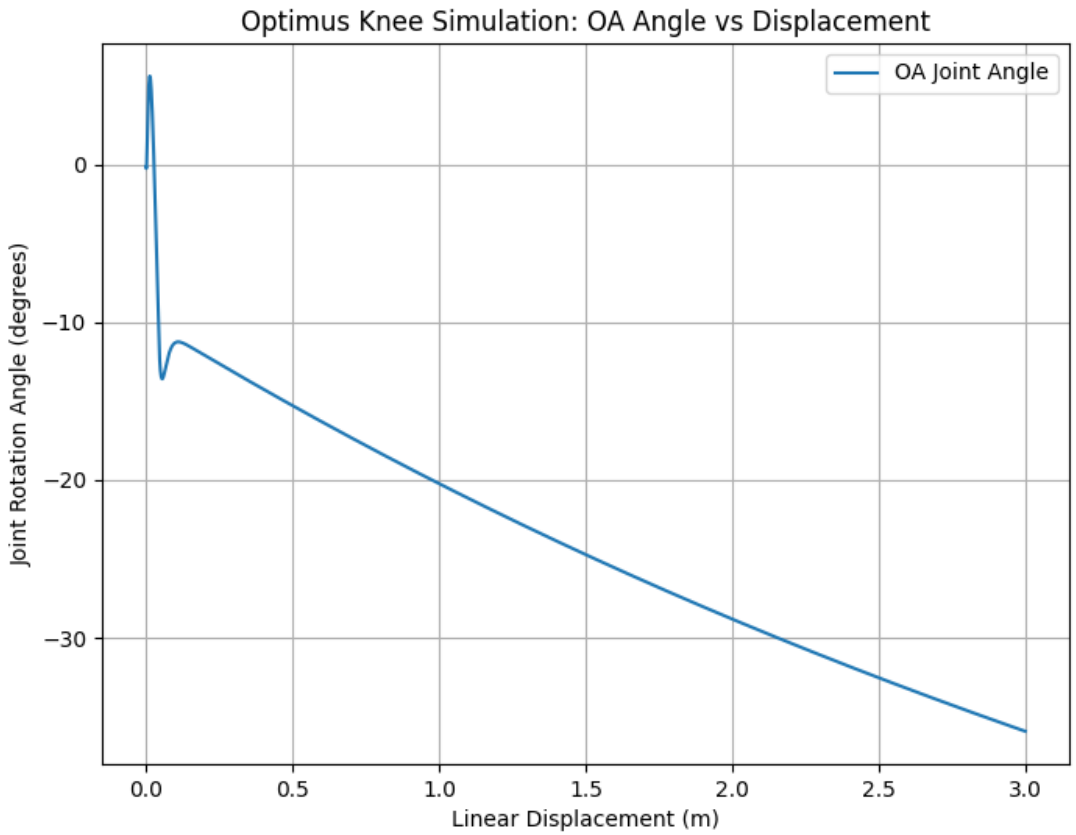


График вращение кривошипа OA как функция линейного перемещения линейного

3- Заключение

В данной лабораторной работе нами был успешно смоделирован и протестирован коленным механизм Optimus, для этого мы создали XML-файл для моделирования в среде MuJoCo и написали Python-скрипт для запуска и анализа моделирования. В ходе моделирования было продемонстрировано корректное движение механизма в рамках заданных ограничений, что подтверждает успех нашего моделирования.