

**федеральное государственное автономное образовательное  
учреждение высшего образования**

# **«Национальный исследовательский университет ИТМО»**

**Факультет систем управления и информатики**

**Практическая работа 3**

Выполнил:

Нагорный Л.А.

Группа:

R4135с

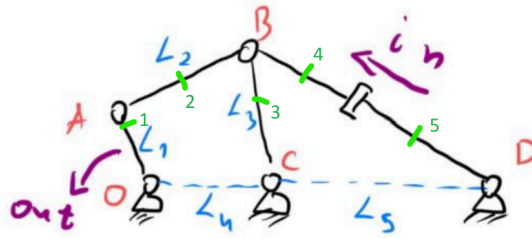
Преподаватель:

Ракшин Е.А.

Санкт-Петербург

2025

## 1. Задание варианта



Параметры системы:

$l_1$	$l_2$	$l_3$	$l_4$	$l_5$
0.069	0.0897	0.1035	0.069	0.345

## 2. Решение

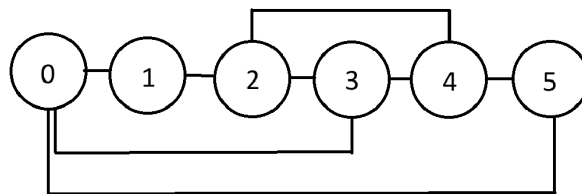


Рисунок 1 - граф механизма

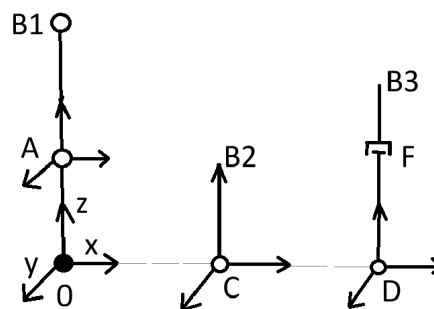


Рисунок 2 - Конфигурация разомкнутой схемы механизма

В результате анализа кинематической схемы механизма был построен соответствующий структурный граф. Моделирование звеньев механизма с заданными геометрическими параметрами выполнено в среде MuJoCo. Для элемента DFB3 реализовано поступательное соединение типа "slide" (ползун), обеспечивающее линейное перемещение вдоль заданной оси.

XML файл разомкнутой системы имеет вид:

```
<?xml version='1.0' encoding='UTF-8'?>
<mujoco>
```

```

<option timestep="1e-3"/>
<option gravity="0 0 -9.8"/>

<asset>
  <texture type="skybox" builtin="gradient" rgb1="1 1 1" rgb2="0.5
0.5 0.5" width="265" height="256"/>
  <texture name="grid" type="2d" builtin="checker" rgb1="0.1 0.1 0.1"
rgb2="0.6 0.6 0.6" width="300" height="300"/>
  <material name="grid" texture="grid" texrepeat="10 10"
reflectance="0.2"/>
</asset>

<worldbody>
  <light pos="0 0 10"/>
  <geom type="plane" size="1.0 1.0 0.1" material="grid"/>

  <!-- Первое звено OAB1 -->
  <body name="OAB1" pos="0 0 1.5" euler="0 0 0">
    <joint name="O" type="hinge" axis="0 -1 0" limited="true"
range="-180 180" stiffness="0" damping="0"/>
    <geom name="point_O" type="sphere" pos="0 0 0" size="0.008"
rgba="0.89 0.14 0.16 1.0"/>
    <geom name="link_OA" type="cylinder" pos="0 0 0.0345"
size="0.005 0.0345" rgba="0.21 0.32 0.82 0.8" euler="0 0 0"/>

    <body name="AB1" pos="0 0 0.069" euler="0 0 0">
      <joint name="A" type="hinge" axis="0 -1 0" limited="true"
range="-180 180" stiffness="0" damping="0.1"/>
      <geom name="point_A" type="sphere" pos="0 0 0" size="0.008"
rgba="0.89 0.14 0.16 1.0"/>
      <geom name="link_AB1" type="cylinder" pos="0 0 0.04485"
size="0.005 0.04485" rgba="0.21 0.32 0.82 0.8" euler="0 0 0"/>
      <site name="end_B1" type="sphere" size="0.01" pos="0 0
0.0897" rgba="1.0 1.0 0.0 1.0"/>
    </body>
  </body>

  <!-- Второе звено CB2 -->
  <body name="CB2" pos="0.2 0 1.5" euler="0 0 0">
    <joint name="C" type="hinge" axis="0 -1 0" limited="true"
range="-180 180" stiffness="0" damping="0.1"/>
    <geom name="point_C" type="sphere" pos="0 0 0" size="0.008"
rgba="0.14 0.89 0.16 1.0"/>

```

```

    <geom name="link_CB2" type="cylinder" pos="0 0 0.05175"
size="0.005 0.05175" rgba="0.14 0.89 0.16 0.8" euler="0 0 0"/>
    <site name="end_B2" type="sphere" size="0.01" pos="0 0 0.1035"
rgba="0.0 1.0 1.0 1.0"/>
  </body>

  <!-- Третье звено DFB3 с поступательной парой -->
  <body name="DFB3" pos="-0.2 0 1.5" euler="0 0 0">
    <joint name="D" type="hinge" axis="0 -1 0" limited="true"
range="-180 180" stiffness="0" damping="0"/>
    <geom name="point_D" type="sphere" pos="0 0 0" size="0.008"
rgba="0.89 0.14 0.82 1.0"/>
    <geom name="link_DB3" type="cylinder" pos="0 0 0.0345"
size="0.005 0.0345" rgba="0.89 0.14 0.82 0.8" euler="0 0 0"/>

    <body name="FB3" pos="0 0 0.069" euler="0 0 0">
      <joint name="slider" type="slide" axis="0 0 1"
limited="true" range="-0.3 0.3" stiffness="0" damping="0"/>
      <geom name="point_F" type="sphere" pos="0 0 0" size="0.008"
rgba="0.89 0.14 0.82 1.0"/>
      <geom name="link_FB3" type="cylinder" pos="0 0 0.1725"
size="0.005 0.1725" rgba="0.89 0.14 0.82 0.8" euler="0 0 0"/>
      <site name="end_B3" type="sphere" size="0.01" pos="0 0
0.345" rgba="1.0 0.5 0.0 1.0"/>
    </body>
  </body>
</worldbody>
</mujoco>

```

Разомкнутое представление имеет вид:

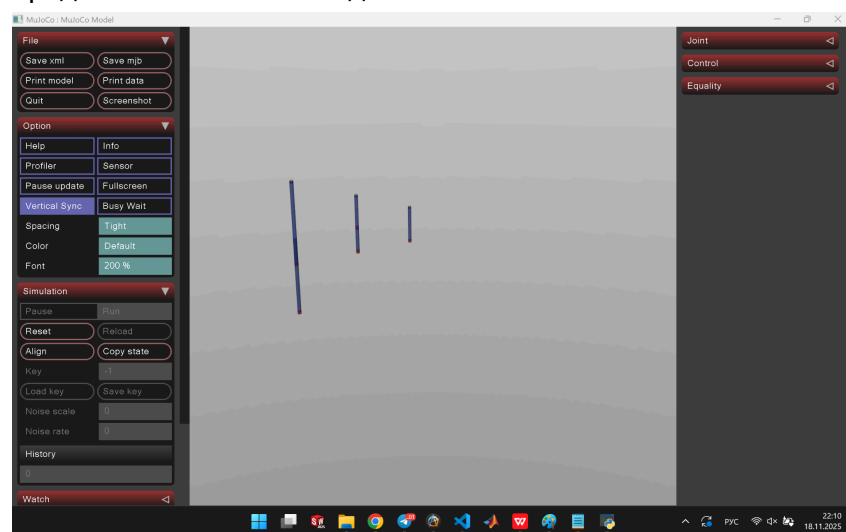
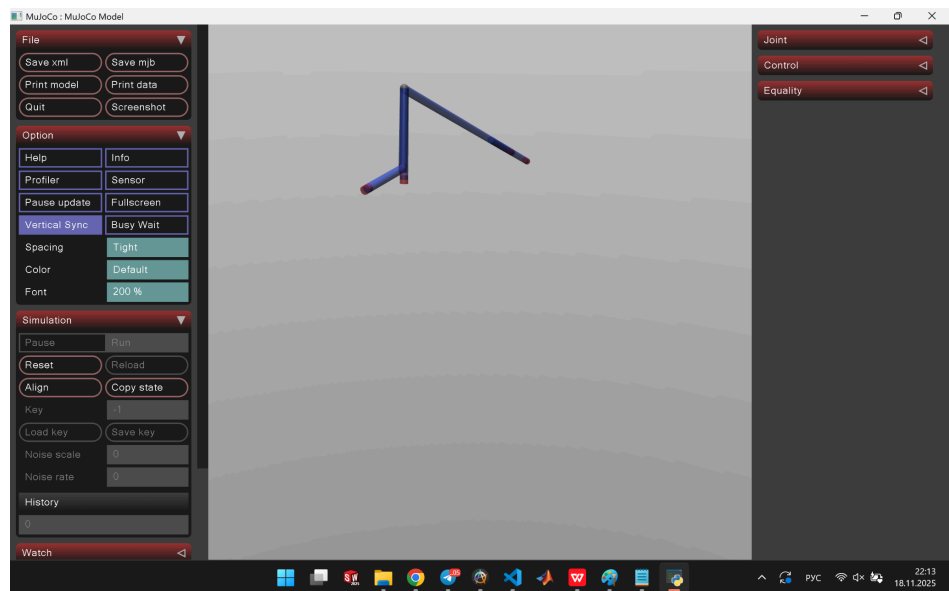
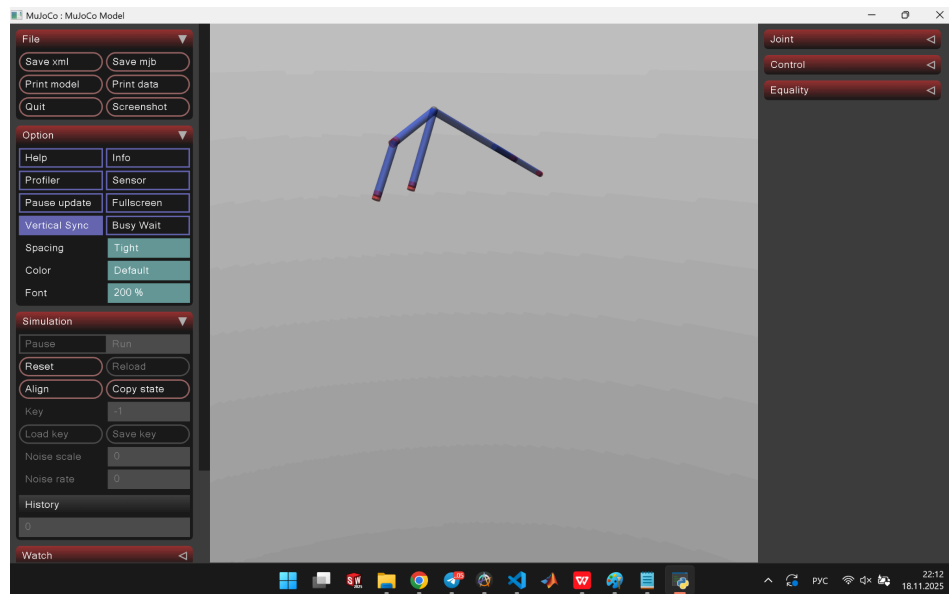


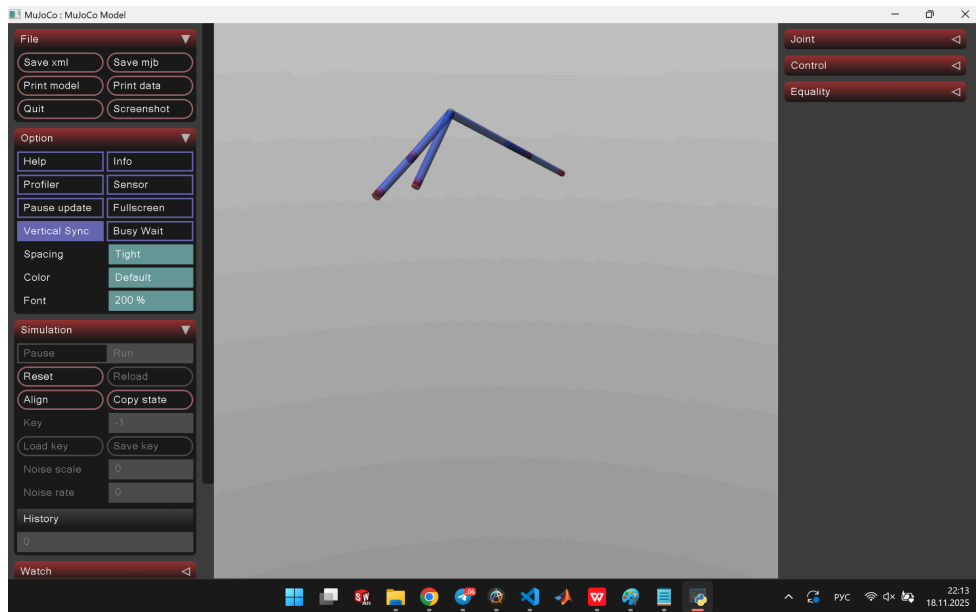
Рисунок 3 - Представление конфигурации разомкнутой схемы механизма

Соединение сайтов B1, B2 и B3 в XML было сделано с помощью контейнера <equality> и элемента <connect>:

```
<connect site1="sC1" site2="sC2"/> <!-- Соединяет B1 с B2 -->
<connect site1="sC1" site2="sC3"/> <!-- Соединяет B1 с B3 -->
```

### 3. Результаты





Получившийся код:

```
import time
import mujoco
import mujoco.viewer
import numpy as np

XML_PATH = "optimus.xml"

def print_model_info(model):
    print("\nModel summary:")
    print(f"  nq = {model.nq}")
    print(f"  nv = {model.nv}")
    print(f"  nu = {model.nu}")

    print("\nJoints:")
    for jid in range(model.njnt):
        name = mujoco.mj_id2name(model, mujoco.mjtObj.mjOBJ_JOINT, jid)
        print(f"  {jid}: {name}")

def set_joint_angle(model, data, joint_name, angle_rad):
    """Установка угла одного DOF-шарнира в новом API."""
    jid = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_JOINT,
joint_name)
    if jid == -1:
        print(f"[Warning] Joint '{joint_name}' not found!")
        return
```

```

qpos_addr = model.jnt_qposadr[jid]
data.qpos[qpos_addr] = angle_rad

def apply_configuration(model, data, config_name, joint_angles):
    """Применяет заданную конфигурацию суставов"""
    print(f"\n== Applying configuration: {config_name} ==")

    data.qpos[:] = 0.0

    for joint_name, angle_deg in joint_angles.items():
        set_joint_angle(model, data, joint_name, np.deg2rad(angle_deg))
        print(f"    {joint_name}: {angle_deg}°")

    mujoco.mj_forward(model, data)

CONFIGURATIONS = {
    "1. Вытянутая позиция": {
        "O": 45,
        "A": 30,
        "C": 15,
        "D": -20,
        "slider": 0.1
    },
    "2. Сложенная позиция": {
        "O": -30,
        "A": -60,
        "C": -45,
        "D": 25,
        "slider": -0.15
    },
    "3. Асимметричная позиция": {
        "O": 60,
        "A": -15,
        "C": 40,
        "D": -35,
        "slider": 0.05
    }
}

# -----
# Основная функция

```

```

# -----

def main():
    # Загружаем модель
    print("Loading model from:", XML_PATH)
    model = mujoco.MjModel.from_xml_path(XML_PATH)
    data = mujoco.MjData(model)

    print_model_info(model)

    # -----
    # Запускаем viewer
    print("\nLaunching viewer...")

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

        # Настройка камеры
        viewer.cam.lookat[:] = np.array([0.25, 0.0, 1.3])
        viewer.cam.distance = 0.8
        viewer.cam.elevation = -20
        viewer.cam.azimuth = 30

        # Демонстрация каждой конфигурации
        for config_name, joint_angles in CONFIGURATIONS.items():

            # Применяем конфигурацию
            apply_configuration(model, data, config_name, joint_angles)

            print(f"Displaying {config_name} for 5 seconds...")

            # Показываем конфигурацию в течение 5 секунд
            for step in range(5000):
                mujoco.mj_step(model, data)
                viewer.sync()
                time.sleep(0.001)

            # Пауза между конфигурациями
            time.sleep(1.0)

        # Финальная конфигурация - возврат в нейтральное положение
        print("\n=== Returning to neutral position ===")
        apply_configuration(model, data, "Neutral", {
            "O": 0, "A": 0, "C": 0, "D": 0, "slider": 0

```



```
    })

    for step in range(3000):
        mujoco.mj_step(model, data)
        viewer.sync()
        time.sleep(0.001)

    print("\nSimulation finished.")

if __name__ == "__main__":
    main()
```

#### 4. Вывод

В ходе выполнения работы была создана модель механизма, содержащего как вращательные, так и поступательные кинематические пары. Исходно механизм был реализован в виде разомкнутой кинематической цепи, где каждое звено было описано с использованием библиотеки MuJoCo в соответствии с заданными геометрическими параметрами в формате XML. На следующем этапе произведено замыкание цепи с образованием замкнутого контура, после чего выполнено тестирование работоспособности разработанной модели.