



---

THE MINISTRY OF SCIENCE AND HIGHER EDUCATION OF  
THE RUSSIAN FEDERATION

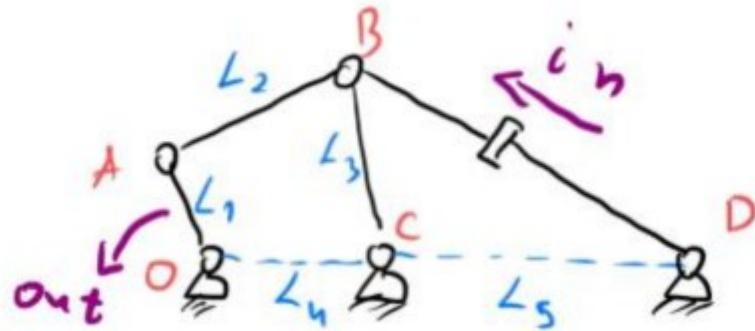
ITMO University  
(ITMO)

Faculty of Control Systems and Robotics  
Simulation of Robotic Systems

task 3

Ali Ahmad  
ISU 475789

### Optimus' knee closed-chain mechanism:

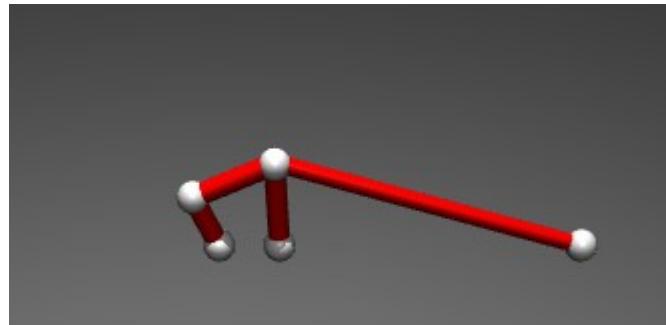


The parameters of the figure are shown in the following table:

Parameter	$L_1$	$L_2$	$L_3$	$L_4$	$L_5$
Value (m)	0.057	0.0741	0.0855	0.057	0.285

First of all, we wrote the XML file to build the closed-loop mechanism in MuJuCo. We created the links and joints. The next step is to create a Python script to load the mechanism model from an XML file that defines all the bodies and joints and then to display the mechanism in the MuJoCo environment.

The model in the MuJoCo environment is shown in the following figure:



In this task, we made an XML file to model the passive Optimus knee closed chain mechanism using MuJoCo environment and we write a Python script using mujoco.viewer library to run the simulation

## xml file:

```
<mujoco model="Optimus">

    <option gravity="0 -9.81 0" timestep="0.0001" solver="Newton" iterations="200" tolerance="1e-8"/>

    <default>

        <site rgba="0 0 1 1"/>

    </default>

    <worldbody>

        <light pos="0 0 3"/>

        <geom type="plane" size="5 5 0.1" rgba=".5 .5 .5 .7"/>

        <body pos="0 0 0.5">

            <joint name="O" axis="0 1 0" range="-90 90" limited="true" />

            <geom name="O" type="sphere" size="0.015" rgba="1 1 1 1"/>

            <geom name="L1" type="cylinder" pos="0 0 0.0285" size="0.01 0.0285" rgba="1 0 0 1"/>

            <body pos="0 0 0.057" euler="0 0 0">Arm) -->

                <joint name="A" axis="0 1 0" range="-90 90" limited="true"/>

                <geom name="A" type="sphere" pos="0 0 0" size="0.015" rgba="1 1 1 1"/>

                <geom name="L2" type="cylinder" pos="0 0 0.03705" size="0.01 0.03705" rgba="1 0 0 1"/>

                <site name="s1" pos="0 0 0.0741" size="0.005"/>

            </body>

        </body>

        <body name="H" pos="0.057 0 0.5">

            <joint name="C" axis="0 1 0" range="-90 90" limited="true"/>

            <geom name="C" type="sphere" size="0.015" rgba="1 1 1 1"/>

            <geom name="L3" type="cylinder" size="0.01" fromto="0 0 0 0 0.0855" rgba="1 0 0 1"/>

            <geom type="sphere" size="0.005" mass="0.002" rgba="1 0 0 1"/>

            <site name="s2" pos="0 0 0.0855" size="0.005"/>

            <body pos="0 0 0.0855">

                <geom name="B" type="sphere" pos="0 0 0" size="0.015" rgba="1 1 1 1"/>

            </body>

        </body>

    </worldbody>

</mujoco>
```

```

</body>
<body pos="0.342 0 0.5" >
  <geom name="D" type="sphere" size="0.015" rgba="1 1 1 1"/>
  <site name="D" pos="0 0 0"/>
  <geom name="D_to_B_link" type="cylinder" fromto="0 0 0 -0.285 0 0.0855" size="0.008"
    rgba="1 0 0 1"/>
  <site name="s3" pos="-0.285 0 0.0855" size="0.005"/>
</body >
</worldbody>
<equality>
  <connect site1="s1" site2="s2"/>
  <connect site1="s2" site2="s3"/>
</equality>
</mujoco>

```

### **Python Code:**

```

import mujoco.viewer
XML_PATH = 'optimus_knee1.xml'
if __name__ == '__main__':
    model = mujoco.MjModel.from_xml_path(XML_PATH)
    data = mujoco.MjData(model)
    with mujoco.viewer.launch_passive(model, data) as viewer:
        data.time = 0
        while viewer.is_running() and data.time < 80:
            mujoco.mj_step(model, data)
            viewer.sync()

```