

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

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

Студент:

Группа R4135c

Амансахедов М.М.

Преподаватель:

Ассистент

E.A. Ракин

Санкт-Петербург 2025 г.

ЦЕЛЬ РАБОТЫ.

Цель данной работы — разработать и промоделировать закрытую кинематическую цепь коленного механизма Optimus в среде MuJoCo, используя параметры, заданные в таблице варианта:

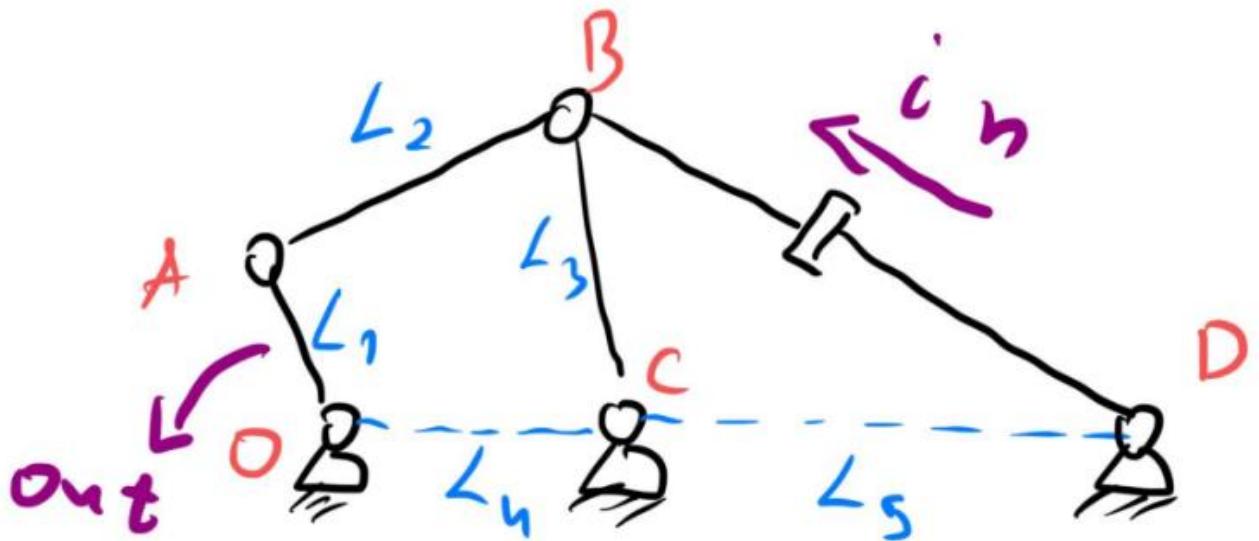


Рисунок 1 – Вариант 2

В рамках работы требуется:

- изучить структуру механизма;
- построить XML-модель в формате MuJoCo;
- реализовать симуляцию механизма средствами Python;
- визуализировать движение и провести краткий анализ результатов

Исходные данные:

L1 (AB)	L2 (BC)	L3 (CD)	L4 (AD)	L5 (CP)
0.036	0.0468	0.054	0.036	0.18

РЕШЕНИЕ

Механизм представляет собой плоскую четырёхзвенную систему с дополнительным звеном СР.

Кинематическая структура:

1. Основная ветка А–В–С–Р, включающая:

- шарнир в точке А;
- звено АВ;
- шарнир В;
- звено ВС;
- точку С;
- звено СР (нижняя часть ноги);
- конечную точку Р.

2. Замыкающая ветка А–Д–С, обеспечивающая закрытую цепь.

3. Замыкание цепи реализовано через связь:

```
<equality>
|   <connect site1="sC1" site2="sC2"/>
</equality>
```

Для моделирования была использована XML-модель, на примере 4bar.xml:

```

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

<option timestep="1e-4"/>
<option gravity="0 0 -9.8"/>

<asset>
    <texture type="skybox" builtin="gradient" rgb1="1 1 1" rgb2=".05 .05 .05" width="265" height="256"/>
    <texture name="grid" type="2d" builtin="checker" rgb1=".1 .1 .1" rgb2=".6 .6 .6" width="300" height="300"/>
    <material name="grid" texture="grid" texrepeat="10 10" reflectance=".2"/>
</asset>

<worldbody>
    <light pos="0 0 10"/>
    <geom type="plane" size="0.5 0.5 0.1" material="grid"/>
    <camera name="side view" pos="0.1 -1.5 1.0" euler="90 0 0" fovy="60"/>
    <camera name="upper view" pos="0 0 1.5" euler="0 0 0"/>
    <body name="ABC1P" pos="0 0 1.5" euler="90 0 0">
        <joint name="A" type="hinge" axis="0 0 1" stiffness="0" springref="0" damping="0"/>
        <geom name="point A" type="cylinder" pos="0 0 0" size="0.02 0.02"
            | rgba=".89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
        <geom name="link AB" type="cylinder" pos="0 -0.018 0" size="0.015 0.018"
            | rgba=".21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
        <body name="BC1P" pos="0 -0.036 0" euler="0 0 0">
            <joint name="B" type="hinge" axis="0 0 1" stiffness="0" springref="0" damping="0.1"/>
            <geom name="point B" type="cylinder" pos="0 0 0" size="0.02 0.02"
                | rgba=".89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
            <geom name="link BC" type="cylinder" pos="0 -0.0234 0" size="0.015 0.0234"
                | rgba=".21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
            <site name="sC1" size="0.01" pos="0 -0.0468 0"/>
            <body name="CP" pos="0 -0.0468 0" euler="0 0 0">
                <geom name="point C1" type="cylinder" pos="0 0 0" size="0.02 0.02"
                    | rgba=".89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
                <geom name="link CP" type="cylinder" pos="0 -0.09 0" size="0.015 0.09"
                    | rgba=".21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
                <site name="sP" size="0.01" pos="0 -0.18 0"/>
            </body>
        </body>
    </body>
</body>

```

```
<body name="DC2" pos="0.036 0 1.5" euler="90 0 0">

    <joint name="D" type="hinge" axis="0 0 1" stiffness="0" springref="0" damping="0.1"/>
    <geom name="point D" type="cylinder" pos="0 0 0" size="0.02 0.02"
        | rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>

    <geom name="link DC" type="cylinder" pos="0 -0.027 0" size="0.015 0.027"
        | rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>

    <site name="sC2" size="0.01" pos="0 -0.054 0"/>

</body>

</worldbody>

<equality>
    <connect site1="sC1" site2="sC2"/>
</equality>

<actuator>
    <position name="A_act" joint="A"/>
</actuator>

<sensor>
    <framepos objtype="site" objname="sP"/>
</sensor>

</mujoco>
```

Для построения модели написан Python код:

```
import os
import time
import numpy as np
import mujoco
import mujoco.viewer
from lxml import etree

L_AB = 0.036
L_BC = 0.0468
L_CD = 0.054
L_AD = 0.036
L_CP = 0.18


def swap_attr(tree, tag, name, attr, value):
    elem = tree.find(f".//{tag}[@name='{name}']")
    if elem is None:
        raise ValueError(f"Element <{tag} name='{name}'> not found")
    elem.set(attr, value)


base_xml = "4bar.xml"
out_xml = "optimus_knee.xml"

tree = etree.parse(base_xml)

swap_attr(tree, "geom", "link AB", "pos", f"0 -{L_AB/2:.5f} 0")
swap_attr(tree, "geom", "link AB", "size", f"0.015 {L_AB/2:.5f}")

swap_attr(tree, "body", "BC1P", "pos", f"0 -{L_AB:.5f} 0")

swap_attr(tree, "geom", "link BC", "pos", f"0 -{L_BC/2:.5f} 0")
swap_attr(tree, "geom", "link BC", "size", f"0.015 {L_BC/2:.5f}")

swap_attr(tree, "site", "sC1", "pos", f"0 -{((L_AB + L_BC)/2):.5f} 0")

swap_attr(tree, "body", "CP", "pos", f"0 -{((L_AB + L_BC)/2):.5f} 0")

swap_attr(tree, "geom", "link CP", "pos", f"0 -{L_CP/2:.5f} 0")
swap_attr(tree, "geom", "link CP", "size", f"0.015 {L_CP/2:.5f}")

swap_attr(tree, "site", "sP", "pos", f"0 -{L_CP:.5f} 0")

swap_attr(tree, "body", "DC2", "pos", f"{L_AD:.5f} 0 1.5")

swap_attr(tree, "geom", "link DC", "pos", f"0 -{L_CD/2:.5f} 0")
swap_attr(tree, "geom", "link DC", "size", f"0.015 {L_CD/2:.5f}")
```

```

swap_attr(tree, "site", "sC2", "pos", f"0 -{L_CD:.5f} 0")

act_elem = tree.find("../actuator/position")
if act_elem is not None:
    act_elem.set("joint", "D")

tree.write(out_xml, pretty_print=True, xml_declaration=True, encoding="UTF-8")
print(f"XML with Optimus parameters saved to: {out_xml}")

model = mujoco.MjModel.from_xml_path(out_xml)
data = mujoco.MjData(model)

P_traj_x = []
P_traj_z = []

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

    site_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SITE, "sP")
    viewer.cam.lookat[:] = data.site_xpos[site_id]

    viewer.cam.distance = 0.3
    viewer.cam.elevation = -20
    viewer.cam.azimuth = 90

    freq = 1.0
    amp = 0.7

    while viewer.is_running():
        data.ctrl[0] = amp * np.sin(2 * np.pi * freq * data.time)
        mujoco.mj_step(model, data)
        viewer.sync()
    freq = 1.0 # Гц
    amp = 0.7 # рад

    while viewer.is_running():
        data.ctrl[0] = amp * np.sin(2 * np.pi * freq * data.time)
        mujoco.mj_step(model, data)
        viewer.sync()

print("Simulation finished.")

```

Результат моделирования:



ВЫВОД.

В ходе выполнения практической работы: Создана XML-модель механизма Optimus с учётом индивидуальных параметров варианта; Разработан Python-скрипт для генерации модели и запуска симуляции; Выполнена визуализация движения четырёхзвенного механизма в MuJoCo.