

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное
учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)
Факультет системы управления и робототехники

ОТЧЕТ

по дисциплине

«Имитационное моделирование робототехнических систем»

по теме:

Лабораторная работа 3 - Аналитическое решение ОДУ

Студент:

Группа № R4133с

Шумкарбек кызы Нурзада

Преподаватель:

инженер, ассистент

Ракишин Е. А.

Санкт-Петербург 2025

СОДЕРЖАНИЕ

Основная часть.....	3
Заключение.....	9

Основная часть

Variant 1 - tendon connected 2R planar mechanism:

Даны коэффициенты:

$R1, m = 0.014$

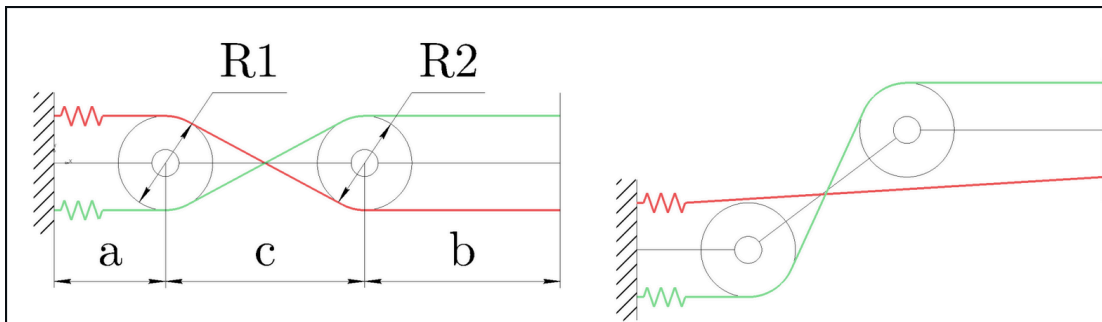
$R2, m = 0.048$

$a, m = 0.062$

$b, m = 0.092$

$c, m = 0.036$

TENDON



Task 3:

Choose one of the passive mechanisms according to your list and model.xml files.

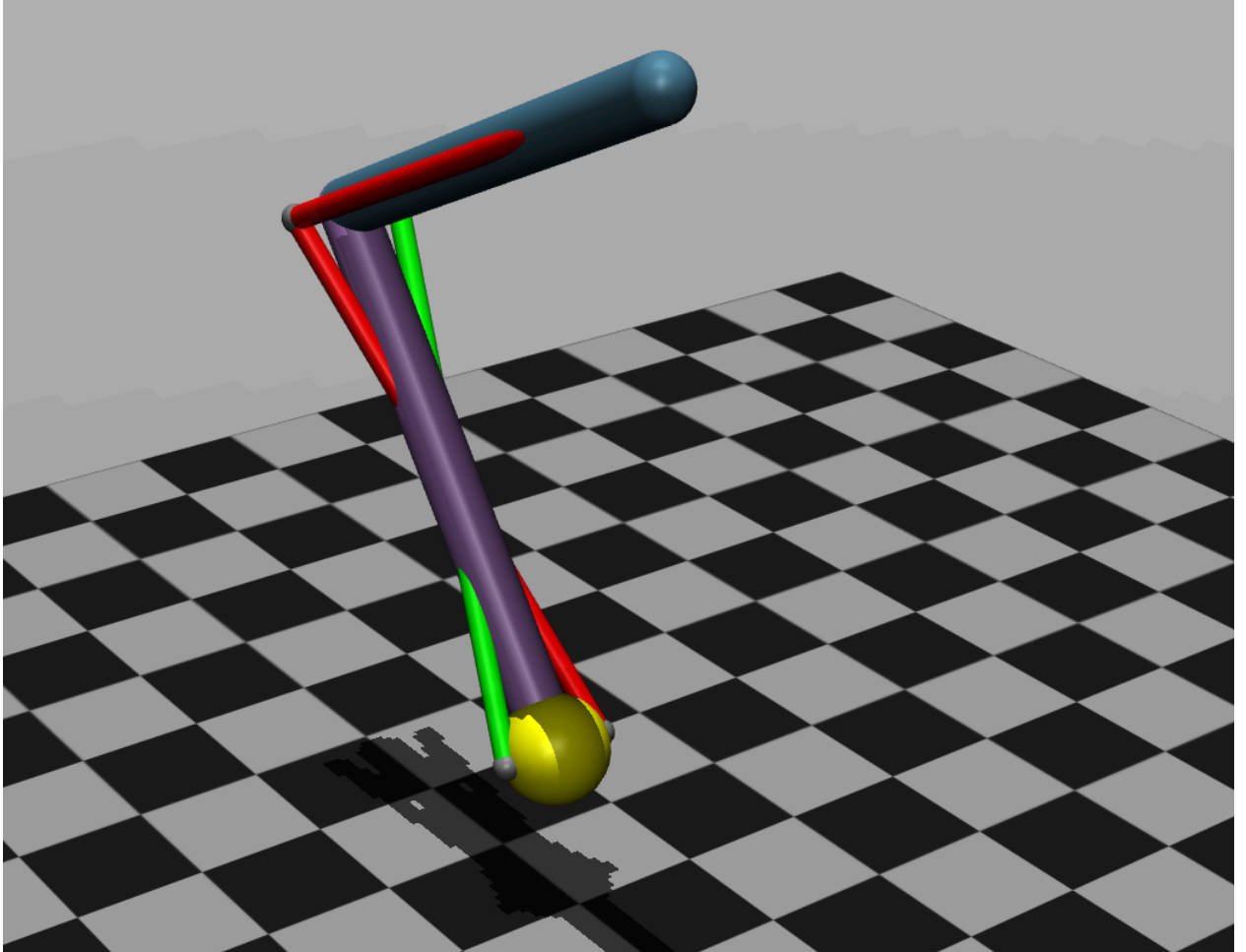
Write python script with model, data and viewer methods. Run the simulation.

Examples of .xml models are in the "Examples" folder.

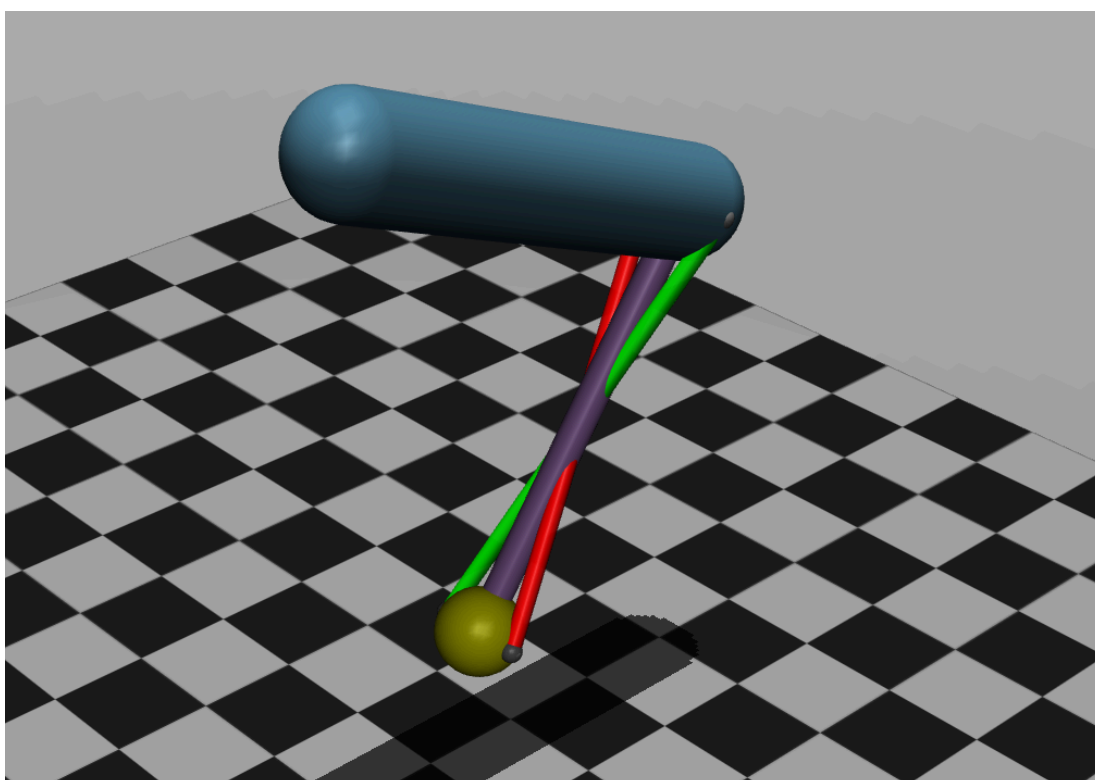
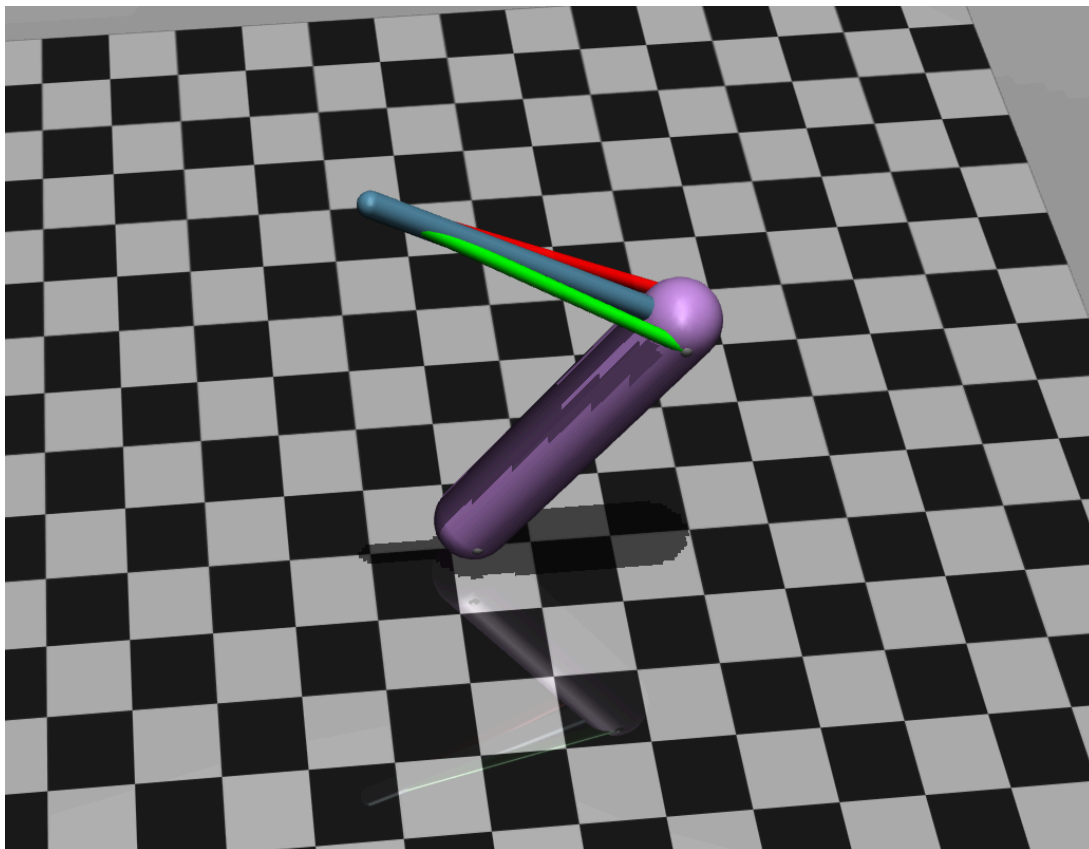
Выполнение работы:

Model

Ниже представлен 2R манипулятор с тендонами в симуляционной среде MuJoCo с одинаковыми радиусами:



Задаем коэффициенты: $R_1, m = 0.014$, $R_2, m = 0.048$, $a, m = 0.062$, $b, m = 0.092$, $c, m = 0.036$.



Описываем настройки графических ресурсов (ассеты) для сцены в MuJoCo:

```
mujoco>
option timestep="1e-4"/>
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>
```

Собираем иерархию 2R манипулятора:

```
<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 pos="0 0 0.3">
    <joint name="joint1" type="hinge" axis="0 1 0"/>
    <geom type="capsule" fromto="0 0 0.2 0 0" size="0.007" mass="0.0001" rgba="1 0 0 1"/>

    <site name="tendon1_site1" pos="0.2 0.02 0"/>
    <site name="tendon2_site1" pos="0.2 -0.02 0"/>

    <body pos="0.2 0 0">
      <joint name="joint2" type="hinge" axis="0 1 0"/>
      <geom type="capsule" fromto="0 0 0.2 0 0" size="0.024" mass="0.0001" rgba="0 1 0 1"/>

      <site name="tendon1_site2" pos="0.2 -0.02 0"/>
      <site name="tendon2_site2" pos="0.2 0.02 0"/>

      <body pos="0.2 0 0">
        <geom type="sphere" size="0.02" rgba="1 1 0 1"/>
      </body>
    </body>
  </body>

  <site name="base_tendon1" pos="0 0.03 0.3"/>
  <site name="base_tendon2" pos="0 -0.03 0.3"/>
</worldbody>
```

```
<tendon>
  <spatial name="tendon1" width="0.004" rgba="1 0 0 0.8">
    <site site="base_tendon1"/>
    <site site="tendon1_site1"/>
    <site site="tendon1_site2"/>
  </spatial>

  <spatial name="tendon2" width="0.004" rgba="0 1 0 0.8">
    <site site="base_tendon2"/>
    <site site="tendon2_site1"/>
    <site site="tendon2_site2"/>
  </spatial>
</tendon>
</mujoco>
```

Data

```
data = mujoco.MjData(model)

data.qpos[0] = np.deg2rad(45)
data.qpos[1] = np.deg2rad(-30)
```

```
model = mujoco.MjModel.from_xml_path(f2)
```

```
data = mujoco.MjData(model)
```

```
data.qpos[0] = np.deg2rad(45)
data.qpos[1] = np.deg2rad(-30)
```

```
SIMEND = 10
TIMESTEP = 0.001
STEP_NUM = int(SIMEND / TIMESTEP)
```

```
EE_position_x = []
EE_position_z = []
joint1_angles = []
joint2_angles = []
```

Viewer methods

```
with mujoco.viewer.launch_passive(model, data) as viewer:
    viewer.cam.distance = 2.0
    viewer.cam.azimuth = 45
    viewer.cam.elevation = -20

    for i in range(STEP_NUM):
        if viewer.is_running:

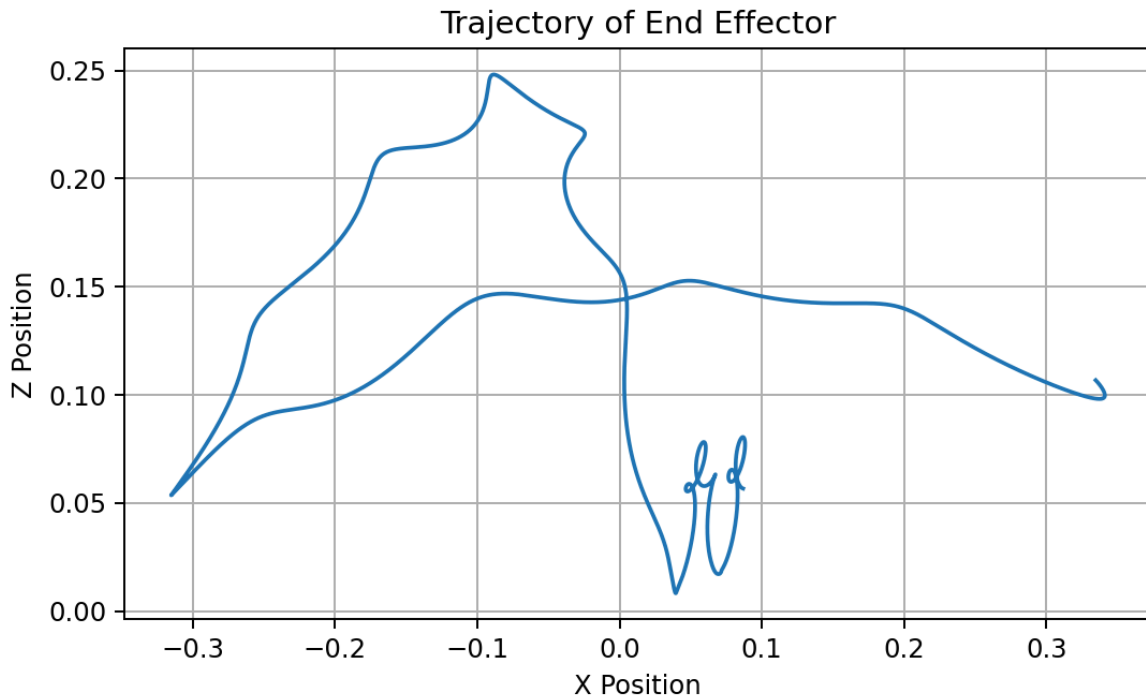
            position_EE = data.body(3).xpos
            EE_position_x.append(position_EE[0])
            EE_position_z.append(position_EE[2])

            if model.nq >= 1:
                joint1_angles.append(data.qpos[0])
            if model.nq >= 2:
                joint2_angles.append(data.qpos[1])

            if model.nu >= 1:
                data.ctrl[0] = np.sin(i * 0.01) * 0.5
            if model.nu >= 2:
                data.ctrl[1] = np.cos(i * 0.01) * 0.3

            mujoco.mj_step(model, data)
            viewer.sync()
            time.sleep(0.001)
```


Определяем траекторию End Effector:



Это траектория движения концевой End Effector 2R-манипулятора.

В ходе лабораторной работы разработана модель 2R планарного манипулятора с тендонами, демонстрирует принципы работы кинематических цепей и эффективность сухожильных систем для передачи усилия в робототехнических устройствах.