



ИТМО
МФКТУ
ФСУИР

Практическая работа №3

Дисциплина: Имитационное моделирование
робототехнических систем

Студент: Воробьев Роман, R4133с

Преподаватель: Ракшин Егор Александрович

Санкт-Петербург

2025

Условия задачи

Параметры системы: $L_1 = 0,052$ м, $L_2 = 0,0676$ м, $L_3 = 0,078$ м, $L_4 = 0,052$ м, $L_5 = 0,26$ м.

Необходимо промоделировать механизм с замкнутой кинематикой и получить траекторию end-effector-a.

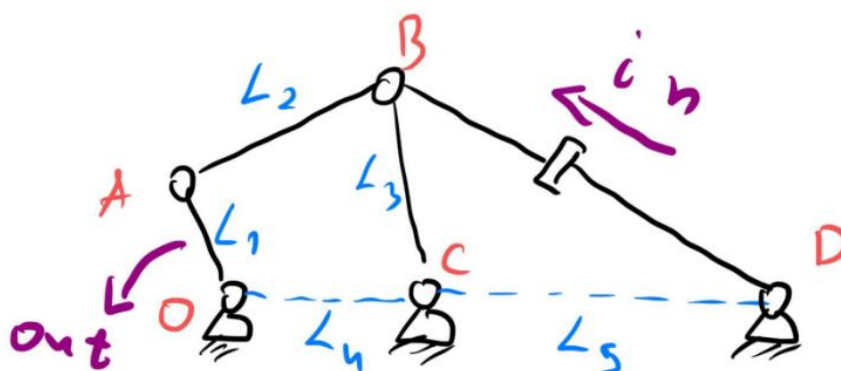


Рисунок 1 – моделируемый механизм

Детали работы

По данным в задании длинам звеньев создан xml-файл, описывающий конструкцию механизма. Код файла представлен в Приложении 1.

Механизм замкнутый, поэтому для построения его дерева в xml-файле он разбивается на 3 механизма с открытой кинематикой: АОВ, СВ и DB, где

- АОВ – двухзвенный плоский механизм с 2 вращательными кинематическими парами,
- СВ – коромысло,
- BD – фактически кулиса.

Отдельные части механизма после описания в `<worldbody>` соединяются специально созданными site-ми с помощью функции `<connect>`.

Для моделирования в MuJoCo и снятия данных используется скрипт на Python, приведенный в Приложении 2.

Результаты

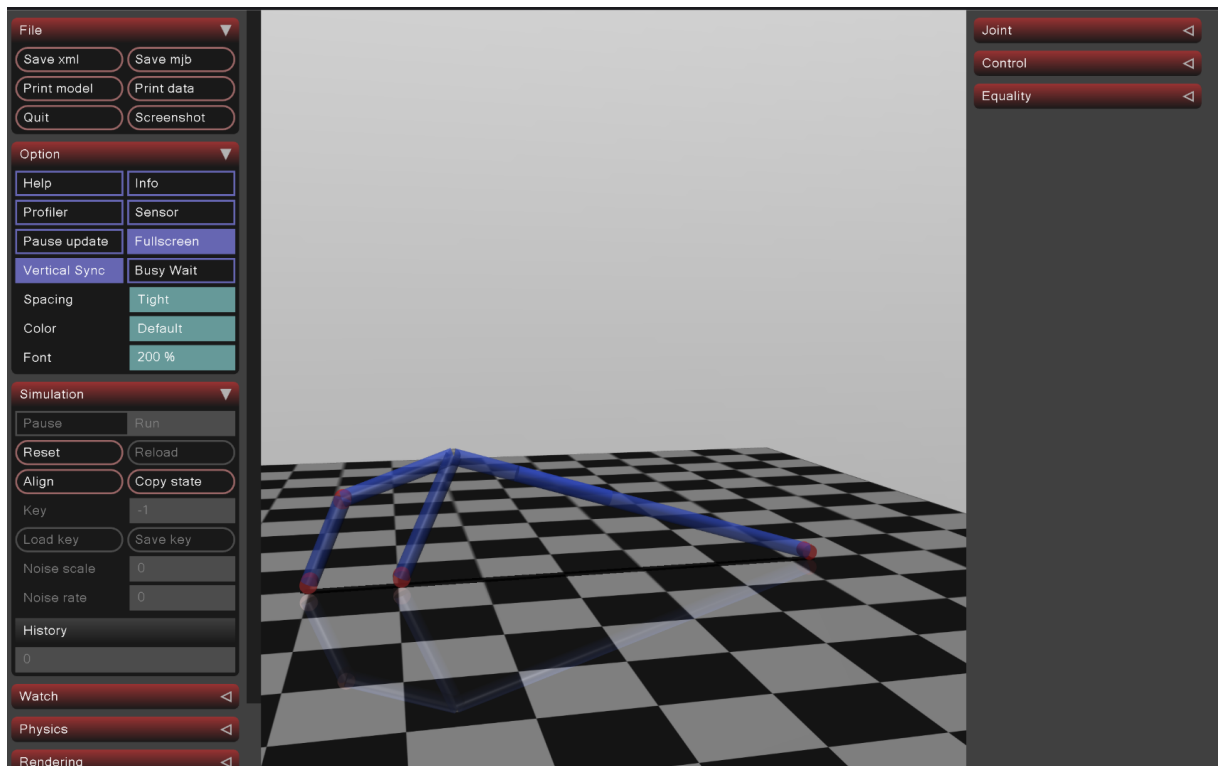


Рисунок 2 – Графическое изображение моделируемого механизма

В звене DB создан slider, с диапазоном перемещений $[-0,03 \text{ м}; 0,03 \text{ м}]$.

На рисунке 3 представлен график траектории точки В при движении слайдера между своими конечными положениями. Ожидаемо это дуга окружности.

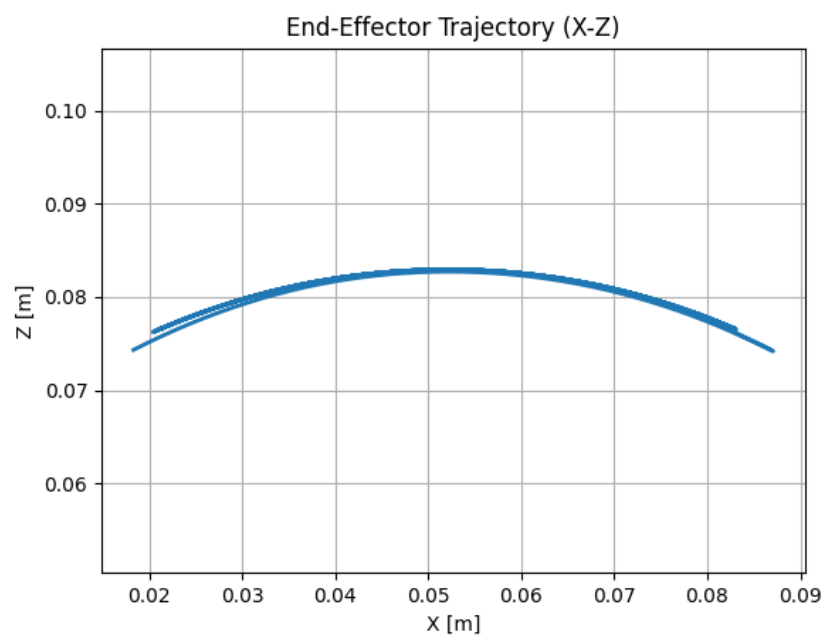


Рисунок 3 – Траектория точки В

Приложение 1

```
<?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="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="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="OAB" pos="0 0 0.005" euler="90 0 0">
```

```
      <joint name="O" type="hinge" axis="0 0 1" stiffness="0"
springref="0" damping="0"/>
```

```
      <geom name="point O" type="cylinder" pos="0 0 0" size="0.005 0.005"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
```

```
      <geom name="link OA" type="cylinder" pos="0 0.026 0" size="0.005
0.026" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
      <body name="AB" pos="0 0.052 0" euler="0 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="0.005
0.005" rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
```

```
        <geom name="link AB" type="cylinder" pos="0 0.0338 0"
size="0.005 0.0338" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
        <site name="sB1" size="0.004" pos="0 0.0676 0"/>
```

```
      </body>
```

```
    </body>
```

```
    <body name="CB" pos="0.052 0 0.005" euler="90 0 0">
```

```
      <joint name="C" type="hinge" axis="0 0 1" stiffness="0"
springref="0" damping="0"/>
```

```
      <geom name="point C" type="cylinder" pos="0 0 0" size="0.005 0.005"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
```

```
      <geom name="link CB" type="cylinder" pos="0 0.039 0" size="0.005
0.039" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
      <site name="sB2" size="0.004" pos="0 0.078 0"/>
```

```
    </body>
```

```
    <body name="DSB" pos="0.312 0 0.005" euler="90 0 0">
```

```
      <joint name="D" type="hinge" axis="0 0 1" stiffness="0"
springref="0" damping="0"/>
```

```
      <geom name="point D" type="cylinder" pos="0 0 0" size="0.005 0.005"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
```

```
      <geom name="link DS" type="cylinder" pos="0 0.1 0" size="0.005 0.1"
rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
      <body name="SB" pos="0 0.17 0" euler="0 0 0">
```

```
        <joint name="S" type="slide" axis="0 1 0" range="-0.03
0.03"/>
```

```
        <geom name="link SB" type="cylinder" pos="0 0.05 0"
size="0.005 0.05" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
        <site name="sB3" size="0.004" pos="0 0.1 0"/>
```

```
      </body>
```

```

        </body>
    </worldbody>

    <equality>
        <connect site1="sB1" site2="sB2"/>
        <connect site1="sB1" site2="sB3"/>
    </equality>

    <actuator>
        <position name="S" joint="S"/>
    </actuator>
</mujoco>

```

Приложение 2

```

import time
import mujoco
import mujoco.viewer
import matplotlib.pyplot as plt

paused = False

def main():
    # Загружаем xml модель
    f = 'lab3.xml'
    m = mujoco.MjModel.from_xml_path(f)
    d = mujoco.MjData(m)

    # site конечного звена
    EE_SITE = "sB1"
    ee_x, ee_y, ee_z = [], [], []

    # запуск viewer
    with mujoco.viewer.launch_passive(m, d) as viewer:
        while viewer.is_running():
            step_start = time.time()
            if not paused:
                mujoco.mj_step(m, d)

                viewer.sync()
                # координаты B
                sid = m.site(EE_SITE).id
                pos = d.site_xpos[sid]
                ee_x.append(pos[0])
                ee_y.append(pos[1])
                ee_z.append(pos[2])

            # синхронизация под реальное время
            dt = m.opt.timestep - (time.time() - step_start)
            if dt > 0:
                time.sleep(dt)

        fig = plt.figure()
        ax = fig.add_subplot(111)
        ax.plot(ee_x[1500:], ee_z[1500:], linewidth=2)
        ax.set_title("End-Effector Trajectory (X-Z)")
        ax.set_xlabel("X [m]")
        ax.set_ylabel("Z [m]")
        ax.grid(True)
        ax.axis("equal")
        plt.show()
        input("\n ")

if __name__ == "__main__":
    main()

```