

Министерство науки и высшего образования Российской Федерации
федеральное
государственное автономное образовательное учреждение высшего
образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»



Имитационное моделирование робототехнических систем

Практическая работа №2

Вариант 2

Студент, группа:

Сергеева Е. С., R4150

Tg:@serkaterina

Преподаватель:

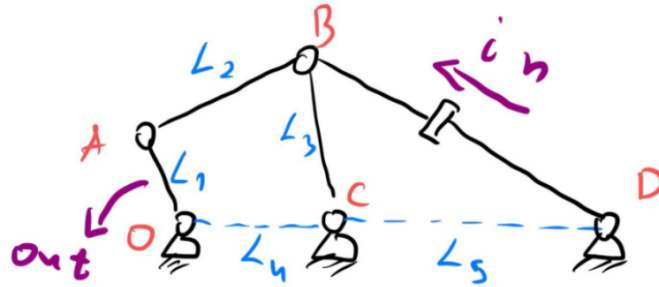
Ракшин Егор Александрович

Санкт-Петербург,

2025

Цель работы: выполнить моделирование системы, реализующей движение коленного сустава посредством механизма колена типа Optimus.

Начальные условия:



Optimus' knee closed-chain mechanism

Таблица начальных условий

L1, m	L2, m	L3, m	L4, m	L5, m
0.046	0.0598	0.069	0.046	0.23

Описание модели в среде MuJoCo:

```
<?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="0.8 0.9 1"
    rgb2="0.2 0.3 0.6" width="265" height="256"/>
    <texture name="grid" type="2d" builtin="checker" rgb1="0.1 0.1
    0.1" rgb2="0.3 0.3 0.3" width="300" height="300"/>
    <material name="grid" texture="grid" texrepeat="10 10"
    reflectance="0.1"/>
  </asset>

  <worldbody>
    <light pos="0 0 10" diffuse="0.8 0.8 0.8" specular="0.5 0.5 0.5"/>
    <light pos="2 2 5" diffuse="0.5 0.5 0.5" specular="0.3 0.3 0.3"/>
    <geom type="plane" size="0.5 0.5 0.1" material="grid"/>

    <!-- -§-€-İ-Б-С-А-Ї-≤-∞-Љ-Љ-Л-μ -В-Љ-З-İ-€ -->
    <site name="fixed_O" pos="0 0 0.02" size="0.008"/>
    <site name="fixed_C" pos="0.046 0 0.02" size="0.008"/>
    <site name="fixed_D" pos="0.069 0 0.02" size="0.008"/>

    <!-- -Ю-°-Э-Ю-Т-Э-Р-í -¶-Х-Я-Ю-І-Ъ-Р О-А-В - -Ъ-†-Ш-Т-Ю-®-Ш-Я -->
    <body name="OAB" pos="0 0 0.02" euler="0 0 0">
      <joint name="O" type="hinge" axis="0 -1 0" damping="0.1"/>
      <geom name="point O" type="sphere" pos="0 0 0" size="0.008"/>
      <geom name="link OA" type="cylinder" pos="0.023 0 0"
      size="0.005 0.023" euler="0 90 0"/>
      <site name="sA" size="0.004" pos="0.046 0 0"/>

      <body name="AB" pos="0.046 0 0" euler="0 0 0">
        <joint name="A" type="ball" damping="0.1"/>

```

```

        <geom name="point A" type="sphere" pos="0 0 0"
size="0.008"/>
        <geom name="link AB" type="cylinder" pos="0.0299 0 0"
size="0.005 0.0299" euler="0 90 0"/>
        <site name="sB" size="0.004" pos="0.0598 0 0"/>
    </body>
</body>

<!-- -Ч-Т-Х-Э-Ю C-B - -Б-Ю-†-Ю-Б-Б-°-Ы-Ю -->
<body name="CB" pos="0.046 0 0.02" euler="0 0 0">
    <joint name="C" type="hinge" axis="0 -1 0" damping="0.1"/>
    <geom name="point C" type="sphere" pos="0 0 0" size="0.008"/>
    <geom name="link CB" type="cylinder" pos="0.0345 0 0"
size="0.005 0.0345" euler="0 90 0"/>
    <site name="sB_CB" size="0.004" pos="0.069 0 0"/>
</body>

<!-- -Ч-Т-Х-Э-Ю D-B - -Ф-Ю-Я-Ю-Ы-Э-Ш-Г-Х-Ы-ђ-Э-Ю-Х -Ч-Т-Х-Э-Ю -->
<body name="DB" pos="0.069 0 0.02" euler="0 0 0">
    <joint name="D" type="hinge" axis="0 -1 0" damping="0.1"/>
    <geom name="point D" type="sphere" pos="0 0 0" size="0.008"/>
    <geom name="link DB" type="cylinder" pos="-0.023 0 0"
size="0.005 0.023" euler="0 90 0"/>
    <site name="sB_DB" size="0.004" pos="-0.046 0 0"/>
</body>

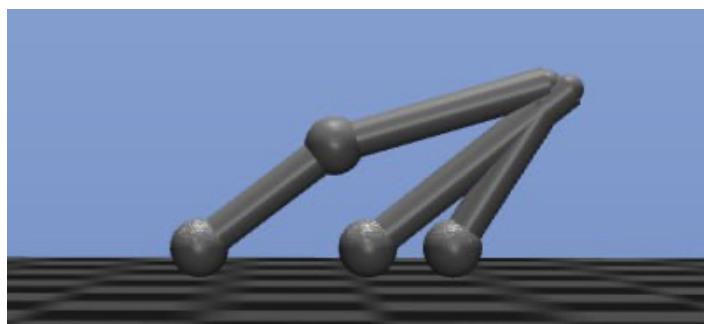
</worldbody>

<equality>
    <!-- -°-Ю-Х-Ф-Ш-Э-Х-Э-Ш-Х -Т-°-Х-• -Г-†-Х-• -Ч-Т-Х-Э-ђ-Х-Т -Т -Г-
Ю-І-Б-Х Б -->
    <connect site1="sB" site2="sB_CB"/>
    <connect site1="sB" site2="sB_DB"/>
</equality>

</mujoco>

```

Результат визуальной модуляции:



Визуальная модель в среде MoJoCo

Для работы с моделью с помощью Python используем код:

```

# Загружаем обновленную модель
m = mujoco.MjModel.from_xml_path('optimus.xml')
d = mujoco.MjData(m)

with mujoco.viewer.launch_passive(m, d, key_callback=key_callback) as
viewer:

```

```

start = time.time()

# Устанавливаем начальное положение для анимации
d.qpos[0] = 0.5 # Начальный угол для кривошипа

while viewer.is_running():
    step_start = time.time()

    if not paused:

        mujoco.mj_step(m, d)

        # Синхронизация визуализации
        viewer.sync()

        # Регулировка скорости для реального времени
        time_until_next_step = m.opt.timestep - (time.time() -
step_start)

        if time_until_next_step > 0:
            time.sleep(time_until_next_step)

if __name__ == "__main__":
    main()

```

Выводы: выполнить моделирование коленного сустава робота, подготовленная модель может быть использована для дальнейших исследований, в том числе имитационного моделирования.