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
15.04.06 Робототехника и искусственный интеллект

ОТЧЕТ
по дисциплине
«Имитационное моделирование робототехнических систем»

Практическая работа №4
СОЗДАНИЕ СИСТЕМЫ УПРАВЛЕНИЯ В СРЕДЕ MUJOCO

Студент:
Группа № R4136c *507170 О.Е. Попова*

Предподаватель:
Ассистент ФСУиР *373529 Е.А. Ракшин*

Санкт-Петербург 2025

1 Задание на практическую работу

1. To the model you have created in the previous task, you need to add actuators. For Optimus mechanism there is one actuator (q1), for tendon mechanism there are two actuators (q1 and q2).
2. Modify the .xml file by adding `<actuaor>` and `<sensor>` containers (look at the examples in the previous task).
3. Define control effort via PD regulator. The $q^{des} = AMP \cdot sin(FREQ \cdot t) + BIAS$. Look in the table for sine wave parameters. If the conrol sequence goes beyond workspace of the mechanism, decrease the amplitude and tune bias only if needed.

Таблица 1 – Данные для управления системой Optimus (q1)

Amp	*Freq*	*Bias*
13.61°=0.238 rad	2.14 Hz	32.7°=0.571 rad

2 Реализация управления с ПД-регулятором

Для моделирования системы в прошлой работе мы разделили ее на три участка. Первый участок – двухзвенный OAB, второй – однозвенный маятник CB, и последний – DB. Управлять будем длиной DB (in), а считывать – угол поворота O (out).

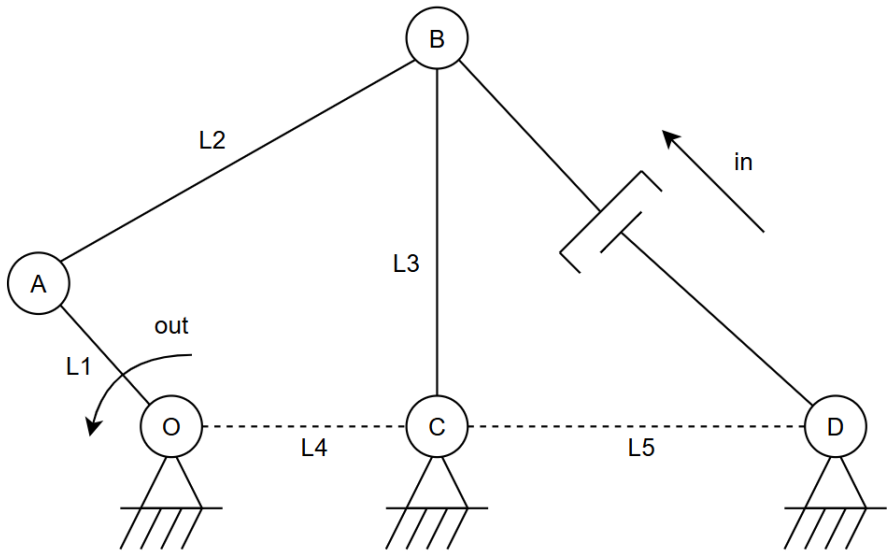

Рисунок 1 – Схема системы Optimus

Возьмем модель из прошлой работы (см. рисунок 2), напишем код в .py, используя MuJoCo (см. приложения). В данной работе использовались **actuator:position** (управление длиной звена DB) и **sensor:jointpos** (считывание угла поворота джоинта O).

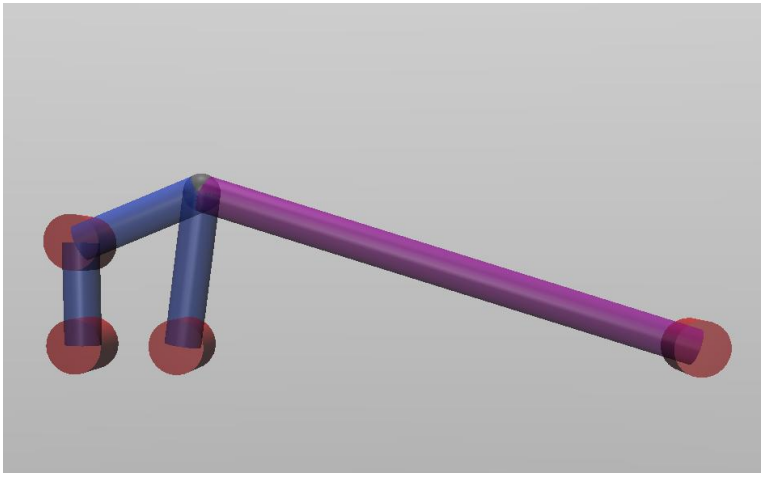

Рисунок 2 – Модель системы Optimus в MuJoCo Viewer

Управление должно осуществляться по закону (2):

$$q^{des}(t) = 0.238 \cdot \sin(2.14t) + 0.571, \quad (1)$$

$$u(t) = k_p\left(q^{des}(t) - q(t)\right) + k_d\left(\dot{q}^{des}(t) - \dot{q}(t)\right). \quad (2)$$

Исходя из формулы (1), крайние положения, которые будет принимать джоинт O, показаны на рисунке 3.

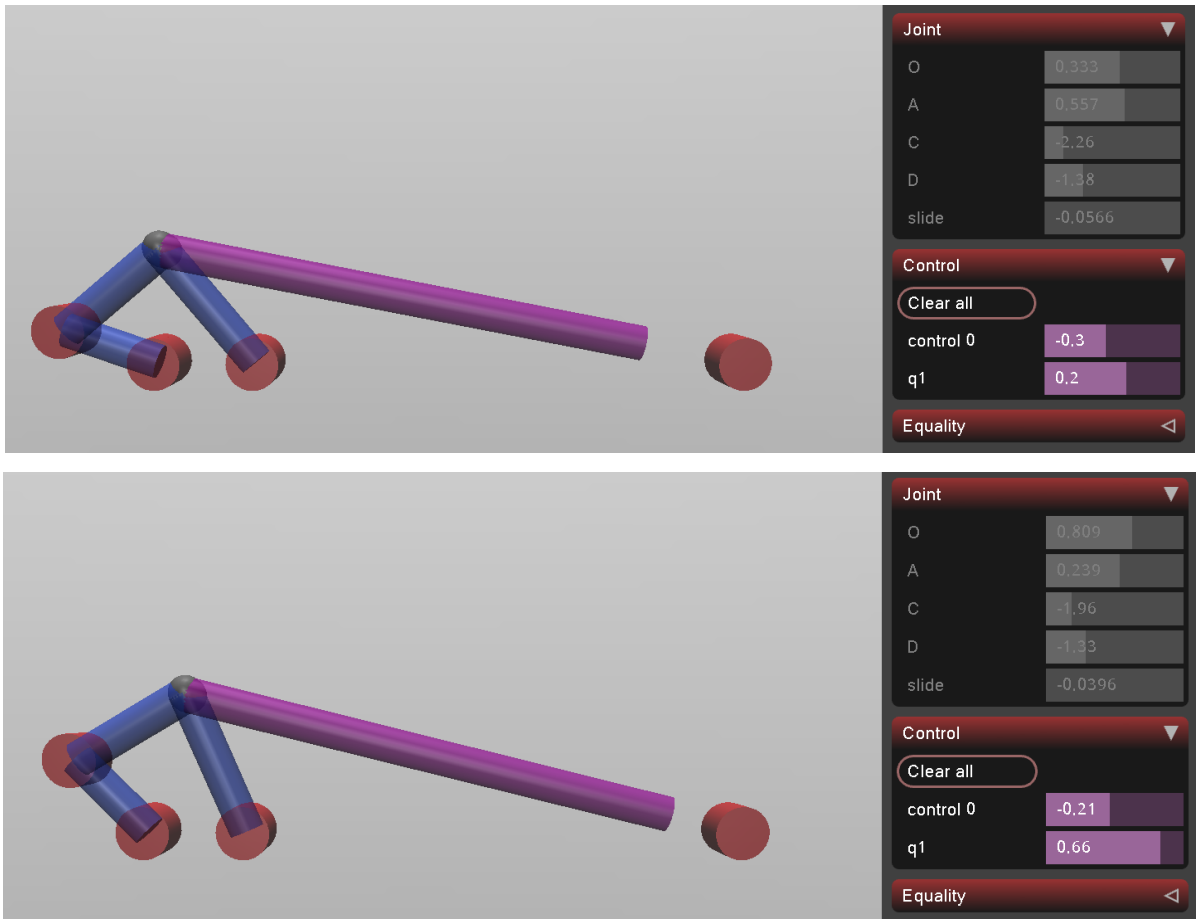

Рисунок 3 – Крайние положения модели (0.333 радиан и 0.809 радиан соотв.)

Для полученной модели построим графики ее поведения при подаче (1) с ПД-регулятором. Коэффициенты ПД-регулятора представлены в (3), графики желаемого и реального положений джоинта O во времени, а также динамической ошибки представлены на рисунке 4.

$$k_p = 650, k_d = 12. \quad (3)$$

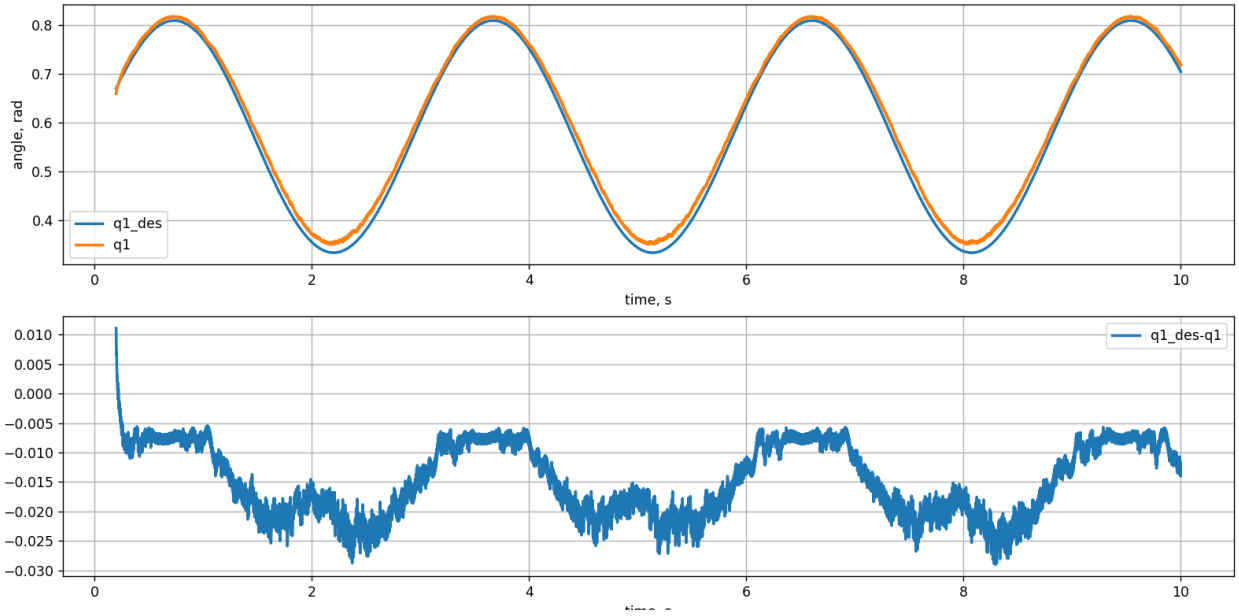

Рисунок 4 – График положения джоинта O во времени

3 Выводы

В ходе выполнения практической работы был проведен синтез ПД-регулятора. Проведен подбор коэффициентов регулятора так, чтобы ошибка в положении была приемлемой (<5%). Практика настройки коэффициентов П и Д компонент полностью совпала с теорией: коэффициент П усиливал воздействие для устранения текущей ошибки, а коэффициент Д сглаживал воздействие, предсказывая будущую ошибку. Этап с добавлением актуатора и сенсора был взят в большей мере с прошлой работы, в этой работе изменений в xml файле было немного.

4 Листинг кода на Python

```
import mujoco
```

```
import mujoco.viewer
import matplotlib.pyplot as plt
import numpy as np

f = "C:\\Users\\olivi\\Downloads\\cylinder.xml"

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

sim_time = 10
dt = model.opt.timestep
num_of_steps = int(sim_time / dt)
time_series = np.linspace(0, sim_time, num_of_steps)
O_desired = []
O_position = []
O_delta = []
t = 0.0

viewer = mujoco.viewer.launch_passive(model, data)

AMP = 0.238
FREQ = 2.14
BIAS = 0.571
q1_des_0 = BIAS
kp = 650
kd = 12

for i in range(num_of_steps):
    if not viewer.is_running():
        break

    q1_des = AMP * np.sin(FREQ * t) + BIAS
    q1 = data.sensordata[0]
    dq1_des = (q1_des - q1_des_0)/dt
    dq1 = data.sensordata[1]

    data.ctrl[0] = kp*(q1_des-q1) + kd*(dq1_des-dq1)

    O_desired.append(q1_des)
    O_position.append(q1)
    O_delta.append(q1_des-q1)
    t += dt
    q1_des_0 = q1_des
    mujoco.mj_step(model, data)
    viewer.sync()

viewer.close()

plt.subplot(2, 1, 1)
plt.plot(time_series[200:], O_desired[200:], '-', linewidth=2, label="q1_des")
```

```
plt.plot(time_series[200:], O_position[200:], '-', linewidth=2, label="q1")
plt.xlabel("time, s")
plt.ylabel("angle, rad")
plt.legend()
plt.grid()
plt.subplot(2, 1, 2)
plt.plot(time_series[200:], O_delta[200:], '-', linewidth=2, label="q1_des-q1")
plt.xlabel("time, s")
plt.ylabel("angle, rad")
plt.legend()
plt.grid()
plt.show()
```

5 Листинг модели в xml

```
<mujoco>

    <option timestep="1e-3"/>
    <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="OAB1" pos="0 0 0.5" euler="-90 0 -90">

            <joint name="O" type="hinge" axis="0 0 1" stiffness="0" springref="0"
damping="0.1"/>
            <geom name="point O" type="cylinder" pos="0 0 0" size="0.015 0.015"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
            <geom name="link OA" type="cylinder" pos="0 -0.028 0" size="0.01
0.028" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
```

```
            <body name="AB1" pos="0 -0.056 0" euler="0 0 90">

                <joint name="A" type="hinge" axis="0 0 1" stiffness="0"
springref="0" damping="0.1"/>
                <geom name="point A" type="cylinder" pos="0 0 0" size="0.015
0.015" rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
                <geom name="link AB" type="cylinder" pos="0 -0.0364 0" size="0.01
0.0364" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
                <site name="sB1" size="0.01" pos="0 -0.0728 0"/>

            </body>

        </body>

        <body name="CB2" pos="0.056 0 0.5" euler="-90 0 -90">

            <joint name="C" type="hinge" axis="0 0 1" stiffness="0" springref="0"
damping="0.1"/>
            <geom name="point C" type="cylinder" pos="0 0 0" size="0.015 0.015"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
            <geom name="link CB" type="cylinder" pos="0 -0.042 0" size="0.01
0.042" rgba="0.21 0.32 0.82 0.5" euler="90 0 0" contype="0"/>
            <site name="sB2" size="0.01" pos="0 -0.084 0"/>

        </body>

        <body name="DB3" pos="0.336 0 0.5" euler="-90 0 -90">

            <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="0.015 0.015"
rgba="0.89 0.14 0.16 0.5" euler="0 0 0" contype="0"/>

            <body name="slider" pos="-0.28 0 0" euler="0 0 0">

                <joint name="slide" type="slide" axis="1 0 0" range="0 1"
damping="0.1"/>
                <geom name="link DB" type="cylinder" pos="0.14 0 0" size="0.01
0.14" rgba="0.62 0.1 0.62 0.5" euler="0 90 0" contype="0"/>
                <site name="sB3" size="0.01" pos="0 0 0"/>

            </body>

        </body>

    </worldbody>

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

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

    <actuator>
        <position joint="slide" ctrlrange="-3 0" ctrllimited="true" kp="100"/>
    </actuator>

    <sensor>
        <jointpos name="O" joint="O"/>
        <jointvel name="dO" joint="O"/>
    </sensor>

</mujoco>
```