THE MINISTRY OF SCIENCE AND HIGHER EDUCATION
OF THE RUSSIAN FEDERATION

ITMO University
(ITMO)

Faculty of Control Systems and Robotics

SYNOPSIS
for the subject
“Simulation of Robotic Systems”

on the topic:
practice 4.

Student:
R4131c, 506185 *Movsesyan Arthur*

Tutor:
Assistant, MSc *Rakshin Egor*

Saint Petersburg 2025

Task

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.

My parameters:

	Q1			Q2	
AMP, deg	FREQ, Hz	Bias, deg	AMP, deg	FREQ, Hz	Bias, deg
40	2.1	9.5	28.45	2.46	-11.5

Python Code Implementation:

```python
import mujoco
import mujoco_viewer
import numpy as np
import matplotlib.pyplot as plt
import os

model_xml = """
<mujoco model="tendon">
  <option gravity="0 -9.81 0" integrator="Euler"/>
  <statistic center="0 0 0" extent="0.2"/>

  <visual>
    <rgba haze="0.9 0.9 0.95 1"/>
  </visual>

  <default>
    <joint axis="0 1 0" damping="0.00005"/>
    <geom type="capsule"/>
  </default>

  <asset>
    <texture name="texplane" type="2d" builtin="checker"
             rgb1="0.2 0.2 0.2" rgb2="0.2 0.2 0.2"
             width="512" height="512" mark="none"/>
    <material name="matplane" reflectance="0"
              texture="texplane" texrepeat="1 1" texuniform="true"/>
  </asset>

  <worldbody>
    <light pos="0 0 0.25"/>
    <light pos="0 0 3" dir="0 0 -1" directional="false"/>
    <geom name="floor" pos="0 0 -0.14" size="2 2 0.1"
          type="plane" material="matplane" conaffinity="15"
condim="3"/>

    <body pos="0 0 0">
      <geom name="left_bound" type="box"
            size="0.002 0.03 0.03"
            rgba="0.3 0.5 1 1" pos="0 0 0"/>
      <geom fromto="0 0 0 0.039 0 0"
            rgba="0.8 0.8 0.3 0.6" size="0.002"
            contype="0" conaffinity="0"/>
      <site name="s1" pos="0.002 0 0.014" size="0.002" rgba="1 1 0
1"/>
      <site name="s2" pos="0.002 0 -0.014" size="0.002" rgba="1 1 0
1"/>

      <body pos="0.039 0 0">
        <joint name="elbow"/>
        <geom fromto="0 0 0 0.073 0 0"
```

```
              rgba="0.8 0.8 0.3 0.6" size="0.002"/>
        <body name="bullet_body" pos="0 0 0">
          <joint name="bullet_hinge" type="hinge"
                 axis="0 1 0" stiffness="5" damping="0.005"/>
          <geom name="Pulley" type="cylinder"
                fromto="0 0.005 0 0 -0.005 0"
                size="0.014"
                rgba="0.3 0.3 0.9 0.9"/>
          <site name="s3" pos="0 0 0.014" size="0.002" rgba="1 1 0
1"/>
          <site name="s4" pos="0 0 -0.014" size="0.002" rgba="1 1 0
1"/>
        </body>

        <body pos="0.073 0 0">
          <joint name="wrist"/>
          <geom fromto="0 0 0 0.043 0 0"
                rgba="0.8 0.3 0.6 1" size="0.002"/>
          <body name="pulley2_body" pos="0 0 0">
            <joint name="pulley2_hinge" type="hinge"
                   axis="0 1 0" stiffness="5" damping="0.005"/>
            <geom name="Pulley2" type="cylinder"
                  fromto="0 0.005 0 0 -0.005 0"
                  size="0.042"
                  rgba="0.3 0.3 0.9 0.9"/>
            <site name="s5" pos="0 0 0.042" size="0.002" rgba="1 1 0
1"/>
            <site name="s6" pos="0 0 -0.042" size="0.002" rgba="1 1 0
1"/>
            <site name="s7" pos="0.043 0 0.042" size="0.002" rgba="1 1
0 1"/>
            <site name="s8" pos="0.043 0 -0.042" size="0.002" rgba="1
1 0 1"/>
          </body>

          <body>
            <geom name="right_bound" type="box"
                  size="0.002 0.03 0.03"
                  rgba="0.3 0.5 1 1"
                  pos="0.043 0 0"/>
          </body>
        </body>
      </body>
    </body>
  </worldbody>

  <tendon>
    <spatial name="tendon1" stiffness="5" rgba="0.4 0.2 1 1"
width="0.0015">
      <site site="s1"/>
```

```
      <geom geom="Pulley" sidesite="s3"/>
      <site site="s3"/>
      <geom geom="Pulley2" sidesite="s6"/>
      <site site="s8"/>
    </spatial>

    <spatial name="tendon2" stiffness="5" rgba="0 0.85 0.7 1"
width="0.0015">
      <site site="s2"/>
      <geom geom="Pulley" sidesite="s4"/>
      <site site="s4"/>
      <geom geom="Pulley2" sidesite="s5"/>
      <site site="s7"/>
    </spatial>
  </tendon>

  <actuator>
    <motor name="q1" joint="elbow" ctrlrange="-10 10" />
    <motor name="q2" joint="wrist" ctrlrange="-10 10" />
  </actuator>

  <sensor>
    <jointpos name="elbow_pos" joint="elbow"/>
    <jointvel name="elbow_vel" joint="elbow"/>
    <jointpos name="wrist_pos" joint="wrist"/>
    <jointvel name="wrist_vel" joint="wrist"/>
  </sensor>
</mujoco>
"""

with open('tendon_mechanism.xml', 'w') as f:
    f.write(model_xml)

model = mujoco.MjModel.from_xml_path('tendon_mechanism.xml')
data = mujoco.MjData(model)

q1_motor_id = model.actuator("q1").id
q2_motor_id = model.actuator("q2").id

Kp = 0.1
Kd = 0.002

a1 = np.deg2rad(40.0)
f1 = 2.1
b1 = np.deg2rad(9.5)

a2 = np.deg2rad(28.45)
f2 = 2.46
b2 = np.deg2rad(-11.5)
```

```python
dt = 0.001
steps = 5000

time_log = np.arange(steps) * dt
q1_log = []
q2_log = []
q1_des_log = []
q2_des_log = []

viewer = mujoco_viewer.MujocoViewer(model, data)

try:
    for i in range(steps):
        t = i * dt

        q1_des = b1 + a1 * np.sin(2 * np.pi * f1 * t)
        q2_des = b2 + a2 * np.sin(2 * np.pi * f2 * t)

        q1 = data.sensordata[0]
        q2 = data.sensordata[2]
        dq1 = data.sensordata[1]
        dq2 = data.sensordata[3]

        u1 = Kp * (q1_des - q1) - Kd * dq1
        u2 = Kp * (q2_des - q2) - Kd * dq2

        data.ctrl[q1_motor_id] = u1
        data.ctrl[q2_motor_id] = u2

        mujoco.mj_step(model, data)
        viewer.render()

        q1_log.append(q1)
        q2_log.append(q2)
        q1_des_log.append(q1_des)
        q2_des_log.append(q2_des)

finally:
    viewer.close()
    if os.path.exists('tendon_mechanism.xml'):
        os.remove('tendon_mechanism.xml')

plt.figure(figsize=(10, 12))

plt.subplot(2, 1, 1)
plt.plot(time_log, q1_log, label="q1_real")
plt.plot(time_log, q1_des_log, label="q1_des", linestyle='--')
plt.title("q1")
plt.legend()
plt.grid(True)
```

```python
plt.xlabel("time, s")
plt.ylabel("a, rad")

plt.subplot(2, 1, 2)
plt.plot(time_log, q2_log, label="q2_real")
plt.plot(time_log, q2_des_log, label="q2_des", linestyle='--')
plt.title("q2")
plt.legend()
plt.grid(True)
plt.xlabel("time, s")
plt.ylabel("a, rad")

plt.tight_layout()
plt.show()
```

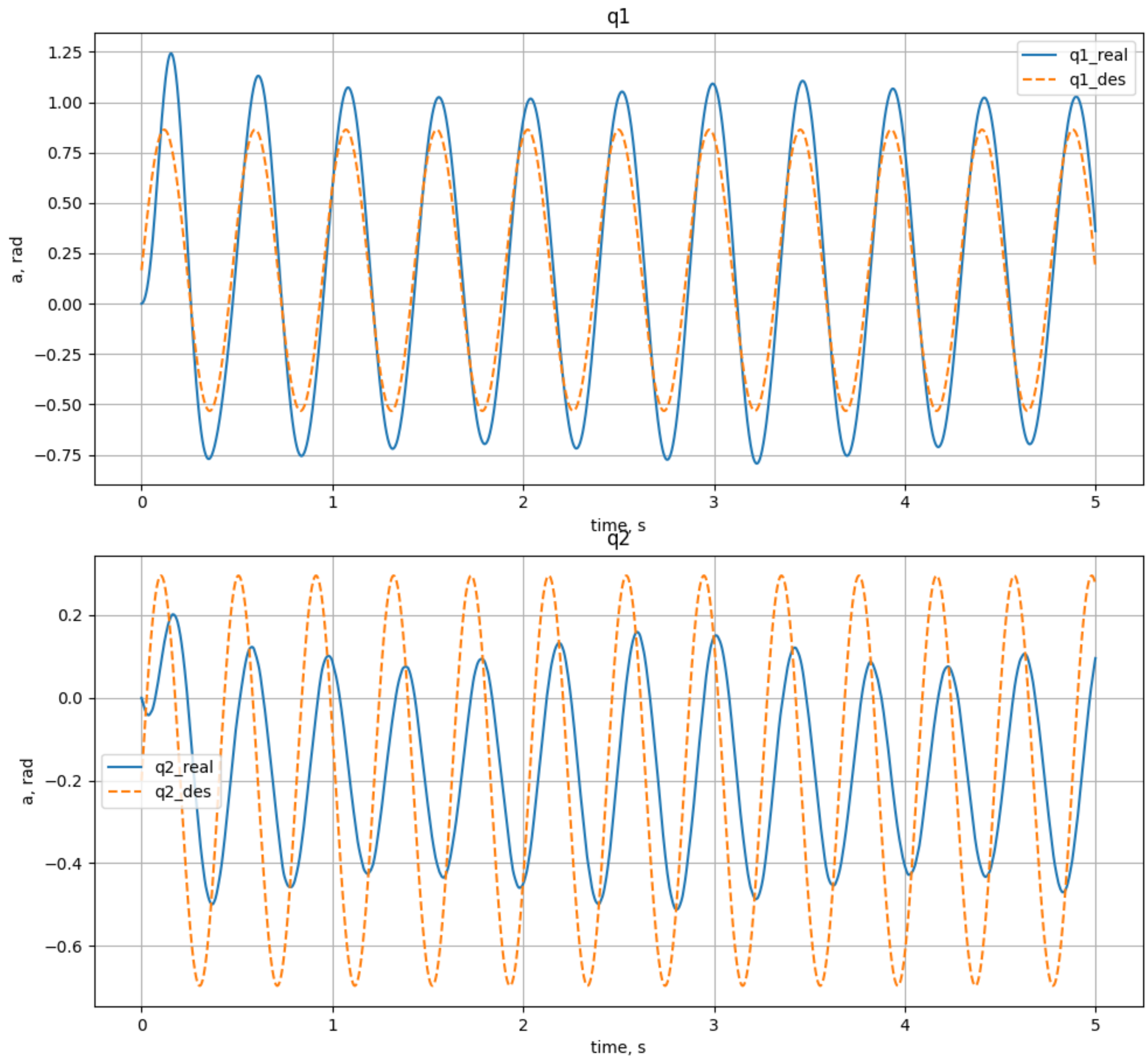

Fig.1. Positions of q1 and q2 actuators.

Conclusion:

As part of this practical assignment, a dynamic mechanical model was designed and validated. The system comprises two cylinders linked by cross-mounted joints, along with a movable right wall that responds to external control inputs.

The model was implemented in Mujoco, enabling realistic simulation of its motion and behavior. This allowed for detailed observation of how individual structural components react to applied control signals.

To facilitate active control and position monitoring, actuators and sensors were integrated into the system. A PD controller was developed and tested. The resulting graphs plotting q1 and q2 over time clearly illustrate the controller's effectiveness in stabilizing and guiding the system's motion.