FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION OF HIGHER EDUCATION
NATIONAL RESEARCH UNIVERSITY ITMO
ITMO UNIVERSITY

FACULTY OF CONTROL SYSTEMS AND ROBOTICS

PROGRAM:
ROBOTICS AND ARTIFICIAL INTELLIGENCE

LABORATORY REPORT
SIMULATION OF ROBOTIC SYSTEMS
PRACTICAL WORK №4

STUDENT
Zongo Sabane

PROFESSOR
Rakshin Egor Aleksandrovich

Academic Year: 2025

STATEMENT

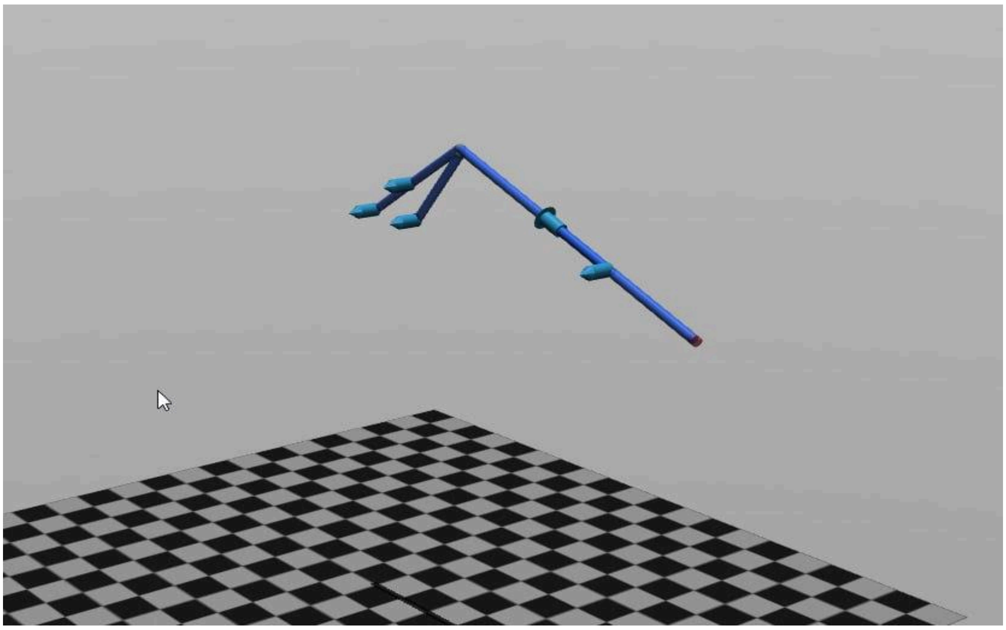
For this task, we have to add an actuator and sensors to the previous model (Optimus' knee closed chain mechanism) we have created, and define control effort via PD regulator. The qdes=AMP · sin(FREQ · t)+BIAS.

Previous model:

```
import mujoco
import mujoco.viewer
import time


xml = """

<mujoco>
  <option timestep="0.001"/>
  <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"/>


    <!-- Branch 1: L1 + L2 -->
    <body name="OAB1" pos="0 0 0.5">
      <joint name="O" type="hinge" axis="0 -1 0" stiffness="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 0.036" size="0.005 
0.036" rgba="0.2 0.4 0.9 0.7"/>
```

```xml
      <body name="AB1" pos="0 0 0.072">
        <joint name="A" type="hinge" axis="0 -1 0" damping="0.1"/>
        <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_AB1" type="cylinder" pos="0 0 0.0468" size="0.005 0.0468" rgba="0.2 0.4 0.9 0.7"/>
        <site name="sC1" pos="0 0 0.0936" size="0.005"/>
      </body>
    </body>


    <!-- Branch 2: L3 -->
    <body name="CB2" pos="0.072 0 0.5">
      <joint name="C" type="hinge" axis="0 -1 0" damping="0.1"/>
       <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_CB2" type="cylinder" pos="0 0 0.054" size="0.005 0.054" rgba="0.2 0.4 0.9 0.7"/>
      <site name="sC2" pos="0 0 0.108" size="0.005"/>
    </body>


    <!-- Branch 3: L6 + L7 -->
    <body name="DFB3" pos="0.36 0 0.5">
      <joint name="D" type="hinge" axis="0 -1 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_DF" type="cylinder" pos="0 0 0.036" size="0.005 0.036" rgba="0.2 0.4 0.9 0.7"/>
      <body name="FB3" pos="0 0 0.072">
        <joint name="slider" type="slide" axis="0 0 1" limited="true" range="-0.2 0.2" damping="0"/>
        <geom name="point F" 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_FB3" type="cylinder" pos="0 0 0.18" size="0.005 0.18" rgba="0.2 0.4 0.9 0.7"/>
        <site name="sC3" pos="0 0 0.36" size="0.005"/>
      </body>
    </body>
  </worldbody>


  <!-- all branches meet at common point -->
```

```
  <equality>
    <connect  site1="sC1" site2="sC2"/>
    <connect  site1="sC1" site2="sC3"/>
  </equality>
</mujoco>
"""

model = mujoco.MjModel.from_xml_string(xml)
data = mujoco.MjData(model)

with mujoco.viewer.launch_passive(model, data) as viewer:
    start = time.time()
    while viewer.is_running() and time.time() - start < 35:
        mujoco.mj_step(model, data)
        viewer.sync()
```



Optimus’ knee closed chain mechanism

SOLVING

To realize the requested model, we will modify the previous script by adding actuator, sensors and defining a PD regulator.

Parameters: q des=AMP · sin(FREQ · t)+BIAS

AMP, deg	FREQ, Hz	BIAS, deg
44.34	2.64	22.8

I. SCRIPT

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

# Parameters
AMP_deg = 44.34
FREQ = 2.64
BIAS_deg = 22.8

#degree to radian
AMP = np.deg2rad(AMP_deg)
BIAS = np.deg2rad(BIAS_deg)

Kp = 3.0
Kd = 0.12


xml = """
<mujoco>
  <option timestep="0.001"/>
  <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"/>
```

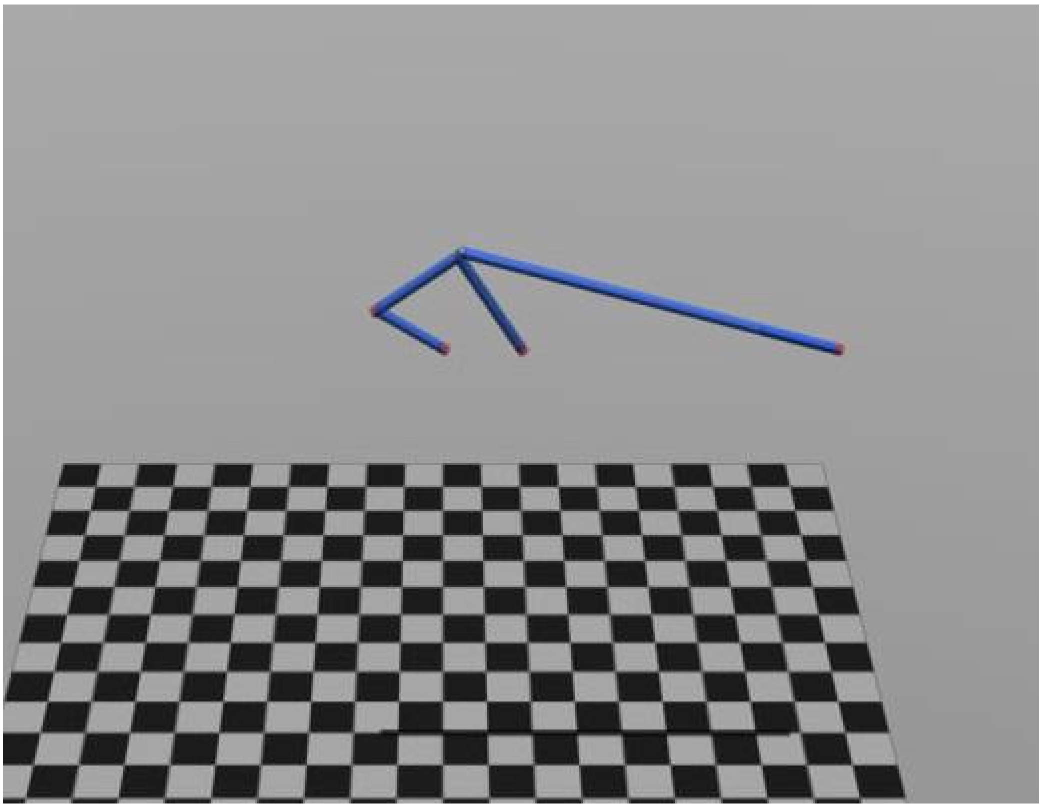
```
    <texture name="grid" type="2d" builtin="checker" rgb1="0.1 0.1 0.1"
rgb2="0.6 0.6 0.6" width="300" height="300"/>
    <material name="grid" texture="grid" texrepeat="10 10"
reflectance="0.2"/>
  </asset>

  <worldbody>
    <light pos="0 0 10"/>
    <geom type="plane" size="0.5 0.5 0.1" material="grid"/>

    <!-- Branch 1: L1 + L2 -->
    <body name="OAB1" pos="0 0 0.5">
      <joint name="O" type="hinge" axis="0 -1 0" stiffness="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 0.036" size="0.005
0.036" rgba="0.2 0.4 0.9 0.7"/>
      <body name="AB1" pos="0 0 0.072">
        <joint name="A" type="hinge" axis="0 -1 0" damping="0.1"/>
        <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_AB1" type="cylinder" pos="0 0 0.0468"
size="0.005 0.0468" rgba="0.2 0.4 0.9 0.7"/>
        <site name="sC1" pos="0 0 0.0936" size="0.005"/>
      </body>
    </body>

    <!-- Branch 2: L3 -->
    <body name="CB2" pos="0.072 0 0.5">
      <joint name="C" type="hinge" axis="0 -1 0" damping="0.1"/>
      <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_CB2" type="cylinder" pos="0 0 0.054" size="0.005
0.054" rgba="0.2 0.4 0.9 0.7"/>
      <site name="sC2" pos="0 0 0.108" size="0.005"/>
    </body>

    <!-- Branch 3: L6 + L7 -->
    <body name="DFB3" pos="0.36 0 0.5">
      <joint name="D" type="hinge" axis="0 -1 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_DF" type="cylinder" pos="0 0 0.036" size="0.005
0.036" rgba="0.2 0.4 0.9 0.7"/>
      <body name="FB3" pos="0 0 0.072">
        <joint name="slider" type="slide" axis="0 0 1" limited="true"
range="-0.25 0.25" damping="0"/>
        <geom name="point F" 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_FB3" type="cylinder" pos="0 0 0.18"
size="0.005 0.18" rgba="0.2 0.4 0.9 0.7"/>
        <site name="sC3" pos="0 0 0.36" size="0.005"/>
      </body>
    </body>
  </worldbody>

  <!-- Fermeture cinématique -->
  <equality>
    <connect site1="sC1" site2="sC2"/>
    <connect site1="sC1" site2="sC3"/>
  </equality>

  <!-- Actionneurs -->
  <actuator>
    <motor name="motor_O" joint="O"/>
    <!-- Optionnel : ajouter d'autres moteurs si besoin -->
  </actuator>

  <!-- Capteurs -->
  <sensor>
    <jointpos name="sensor_O_pos" joint="O"/>
    <jointvel name="sensor_O_vel" joint="O"/>
  </sensor>
</mujoco>
"""

model = mujoco.MjModel.from_xml_string(xml)
data = mujoco.MjData(model)

motor_id = 0
times = []
q_des_log = []
q_log = []
dq_log = []
```

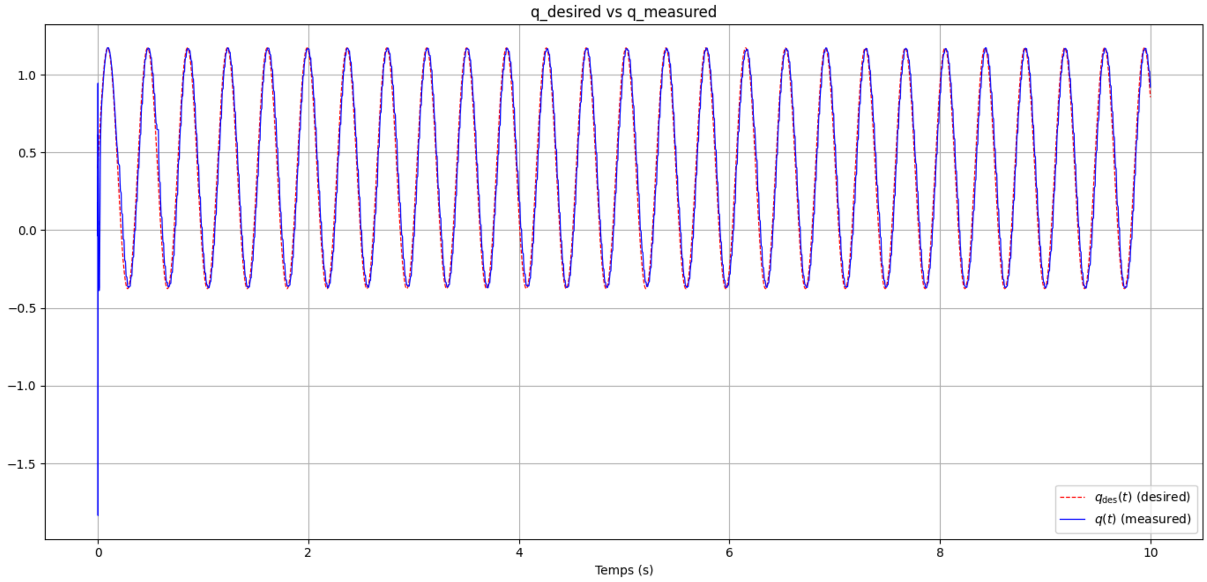
```python
with mujoco.viewer.launch_passive(model, data) as viewer:
    start_time = time.time()
    while viewer.is_running() and (time.time() - start_time) < 10:
        t = time.time() - start_time
        q_des = AMP * np.sin(2 * np.pi * FREQ * t) + BIAS

        # read joint 0 position and velocity
        q = data.sensor("sensor_0_pos").data[0]
        dq = data.sensor("sensor_0_vel").data[0]

        # PD Command
        ctrl = Kp * (q_des - q) - Kd * dq

        # Apply command
        data.ctrl[0] = ctrl

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

        times.append(t)
        q_log.append(q)
        dq_log.append(dq)
        q_des_log.append(q_des)

        # Plot results
plt.figure(figsize=(24, 8))

plt.subplot(1, 3, 1)
plt.plot(times, q_log, 'b',linewidth=0.8, label='q (joint 0 position)')
plt.xlabel('Temps (s)')
plt.ylabel('Angle (rad)')
plt.title('Joint 0 angle vs time')
plt.grid(True)
plt.legend()

plt.subplot(1, 3, 2)
plt.plot(times, dq_log, 'k-',linewidth=0.8, label='dq (joint 0
velocity)')
plt.xlabel('Temps (s)')
plt.ylabel('Angular velocity (rad/s)')
plt.title('joint 0 velocity vs time')
plt.grid(False)
plt.legend()
```

```python
plt.subplot(1, 3, 3)
plt.plot(times, q_des_log, 'r--',linewidth=0.8,
label=r'$q_{\mathrm{des}}(t)$ (desired)')
plt.plot(times, q_log, 'b',linewidth=0.8, label=r'$q(t)$ (measured)')
plt.xlabel('Temps (s)')
plt.ylabel('Angle (rad)')
plt.title('q_desired vs q_measured')
plt.legend()
plt.grid(True)

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

II. SIMULATION RESULT



Optimus' knee closed chain mechanism

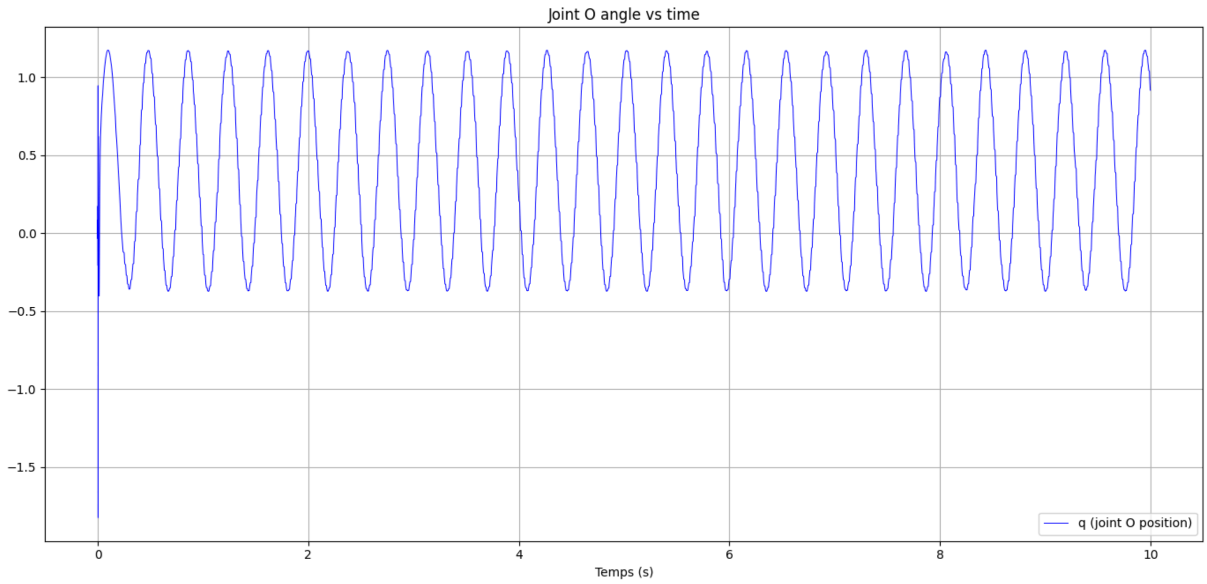


q_des vs q over the first 10 seconds

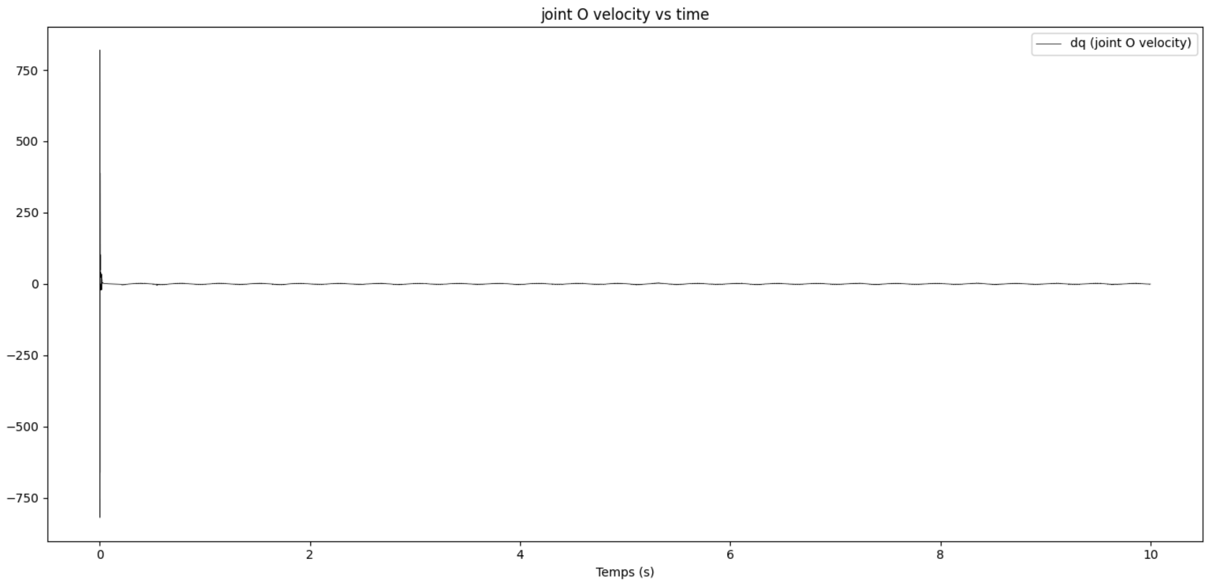
III. CONCLUSION

The simulation demonstrates that the closed-chain robotic mechanism successfully tracks the desired sinusoidal trajectory in a general sense, adhering to the prescribed frequency of 2.64 Hz. However, noticeable phase lag, waveform distortion, and an initial transient highlight the dynamic limitations inherent to the constrained kinematic structure. These deviations arise from the coupling between passive branches and the high-frequency excitation, which together restrict the system's ability to perfectly reproduce the reference signal. Overall, the results validate the correct implementation of the PD controller and confirm that the mechanism operates within a bounded region of its feasible workspace.

ANNEX



Joint O angle vs time
1.0
0.5
0.0
−0.5
−1.0
−1.5
q (joint O position)
0
2
4
6
8
10
Temps (s)



joint O velocity vs time
dq (joint O velocity)
750
500
250
0
−250
−500
−750
0
2
4
6
8
10
Temps (s)