МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский университет ИТМО»
(Университет ИТМО)

ОТЧЕТ
О ЛАБОРАТОРНОЙ РАБОТЕ №4

по дисциплине «Имитационное моделирование робототехнических систем»

Выполнил:
студент групп № R4136c
Тихонов В.С.

Проверил:
преподаватель
Ракшин Е.А.

Санкт-Петербург, 2025

Введение

Данная лабораторная работа является логическим продолжением цикла заданий по имитационному моделированию робототехнических систем в среде MuJoCo. В рамках предыдущих исследований была разработана кинематическая модель 'Optimus'. Основной целью настоящей работы является преобразование этой пассивной модели в активную динамическую систему, способную выполнять целевые задачи.

Задание

1. К созданной модели в предыдущем задании необходимо добавить приводы (актюаторы). Для механизма "Optimus"предусмотрен один привод (q1).

2. Измените файл .xml, добавив контейнеры <actuator> и <sensor>.

3. Определите управляющее воздействие с помощью ПД-регулятора. Управление задается как q^{des} = AMP * sin(FREQ * t) + BIAS. Параметры синусоиды приведены в таблице. Если управляющий сигнал выходит за пределы рабочей зоны механизма, уменьшите амплитуду и, только если это необходимо, скорректируйте смещение (bias).

Таблица 1: Параметры синусоидального сигнала

q_i	AMP, deg	FREQ, Hz	BIAS, deg
q_1	33.79	1.76	-42.7

.

Ход работы

1. Представим выражение для силы в виде ПИ-регулятора:

$$\tau = K_p \cdot e + K_d \cdot \dot{e},$$

где $\dot{e} = \dot{q}^{des} - \dot{q}$, $K_p = [150\ 260]$, $K_d = [3\ 4]$.

2. Программная реализация в соответстствии с заданием представлена в виде:

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

#                 XML

model_xml = """
    <?xml version='1.0' encoding='UTF-8'?>
<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>

    <camera name="side view" pos="0.1 -0.4 0.5" euler="90 0 0" fovy="60
       "/>

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

    <body name="OAB1" pos="0 0 0.5" euler="0 0 0">

            <joint name="O" type="hinge" axis="0 -1 0" stiffness="0"
               springref="0" damping="0.1"/>
            <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 0.0345" size=
               "0.005 0.0345" rgba="0.21 0.32 0.82 0.5" euler="0 0 0"
               contype="0"/>
            <site name="sA" size="0.005" pos="0 0 0.069"/>

            <body name="AB1" pos="0 0 0.069" euler="0 0 0">
```

```
37
38                <joint name="A" type="hinge" axis="0 -1 0" stiffness="0
                     " springref="0" damping="0.1"/>
39                <geom name="point B" 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"/>
40                <geom name="link AB1" type="cylinder" pos="0 0 0.04485"
                      size="0.005 0.04485" rgba="0.21 0.32 0.82 0.5"
                     euler="0 0 0" contype="0"/>
41                <site name="sC1" size="0.005" pos="0 0 0.0897"/>
42
43            </body>
44
45    </body>
46
47
48    <body name="CB2" pos="0.069 0 0.5" euler="0 0 0">
49
50        <joint name="C" type="hinge" axis="0 -1 0" stiffness="0"
             springref="0" damping="0.1"/>
51        <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"/>
52        <geom name="link CB2" type="cylinder" pos="0 0 0.05175" size="
             0.005 0.05175" rgba="0.21 0.32 0.82 0.5" euler="0 0 0"
             contype="0"/>
53        <site name="sC2" size="0.005" pos="0 0 0.1035"/>
54
55    </body>
56
57    <body name="DFB3" pos="0.345 0 0.5" euler="0 0 0">
58
59        <joint name="D" type="hinge" axis="0 -1 0" stiffness="0"
             springref="0" damping="0.1"/>
60        <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"/>
61        <geom name="link DB3" type="cylinder" pos="0 0 0.1" size="0.005
              0.1" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
62
63        <body name="FB3" pos="0 0 0.075" euler="0 0 0">
64
65            <joint name="slider" type="slide" axis="0 0 1" limited="
                 true" range="-0.2 0.2" stiffness="0" springref="0"
                 damping="0.1"/>
66            <geom name="point B3" 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"/>
67            <geom name="link FB3" type="cylinder" pos="0 0 0.075" size=
                 "0.005 0.15" rgba="0.21 0.32 0.82 0.5" euler="0 0 0"
```

```
                contype="0"/>
            <site name="sC3" size="0.005" pos="0 0 0.225"/>

        </body>

    </body>

    </worldbody>

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

    <actuator>
        <position name="slider" joint="slider"/>
    </actuator>

    <sensor>
        <framepos objtype="site" objname="sA"/>
        <framepos name="end_effector_pos" objtype="site" objname="sC3"
            />
    </sensor>

</mujoco>
"""

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

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

#                     PD-
KP = 18  #
KV = 15   #

#
FREQ = 1.76  # [Hz]
AMP = np.deg2rad(33.79)  # [rad]
BIAS = np.deg2rad(-2.7)  # [rad]

#
SIMEND = 15  #
TIMESTEP = model.opt.timestep
STEP_NUM = int(SIMEND / TIMESTEP)
```

```
114
115 #
116 timeseries = np.linspace(0, SIMEND, STEP_NUM)
117 theta_des = AMP * np.sin(2 * np.pi * FREQ * timeseries) + BIAS
118
119 #
120 qpos_log = []
121 qvel_log = []
122 ctrl_log = []
123 theta_des_log = []
124 error_log = []
125 EE_position_x = []
126 EE_position_z = []
127
128 #                PD-
129 def set_torque(mj_data, kp, kv, theta_desired):
130     error = theta_desired - mj_data.qpos[0]
131     error_velocity = 0 - mj_data.qvel[0]  #
                          = 0
132     return kp * error + kv * error_velocity
133
134 #                viewer
135 viewer = mujoco_viewer.MujocoViewer(model, data)
136
137 start_time = time.time()
138
139 try:
140     #
141     for i in range(STEP_NUM):
142         if not viewer.is_alive:
143             break
144
145         step_start = time.time()
146
147         #
148         control_signal = set_torque(data, KP, KV, theta_des[i])
149         data.ctrl[0] = control_signal
150
151         #
152         mujoco.mj_step(model, data)
153         viewer.render()
154
155         #
156         qpos_log.append(data.qpos[0])
157         qvel_log.append(data.qvel[0])
158         ctrl_log.append(data.ctrl[0])
```

```
159         theta_des_log.append(theta_des[i])
160
161         #
162         current_error = theta_des[i] - data.qpos[0]
163         error_log.append(current_error)
164
165         #
166         ee_pos = data.sensor('end_effector_pos').data
167         EE_position_x.append(ee_pos[0])
168         EE_position_z.append(ee_pos[2])
169
170         #
171         time_until_next_step = TIMESTEP - (time.time() - step_start)
172         if time_until_next_step > 0:
173             time.sleep(time_until_next_step)
174
175 finally:
176     viewer.close()
177
178     #
179     t = np.arange(len(theta_des_log)) * TIMESTEP
180
181     #              1:
182     plt.figure(figsize=(12, 7))
183     plt.plot(t, (theta_des_log), label="                    ", linewidth
        =2)
184     plt.plot(t, (qpos_log), label="                         ", alpha
        =0.8)
185     plt.title("                    PD-                    ")
186     plt.xlabel("           [  ]")
187     plt.ylabel("         [      ]")
188     plt.grid(True)
189     plt.legend()
190
191     #              2:
192     plt.figure(figsize=(12, 6))
193     plt.plot(t, (error_log), label="             (    _     -
            _     )", color="purple", linewidth=2)
194     plt.title("                                    ")
195     plt.xlabel("           [  ]")
196     plt.ylabel("             [      ]")
197     plt.grid(True)
198     plt.legend()
199
200     plt.show()
201
202     #
```

```
    error_deg = (error_log)
    max_error = np.max(np.abs(error_deg))
    rms_error = np.sqrt(np.mean(np.square(error_deg)))
    mean_abs_error = np.mean(np.abs(error_deg))

    print(f"                                          :")
    print(f"AMP = {np.rad2deg(AMP):.2f} deg")
    print(f"FREQ = {FREQ} Hz")
    print(f"BIAS = {np.rad2deg(BIAS):.2f} deg")
    print(f"PD-                  : KP = {KP}, KV = {KV}")
    print(f"\ n                                  :")
    print(f"                                     : {max_error:.4f} rad"
        )
    print(f"                                                  (RMS): {
        rms_error:.4f} rad")
    print(f"                                                (MAE): {
        mean_abs_error:.4f} rad")

    #
    if os.path.exists('knee_mechanism.xml'):
        os.remove('knee_mechanism.xml')
```

Результатом работы данного кода являются следующие рисунки:

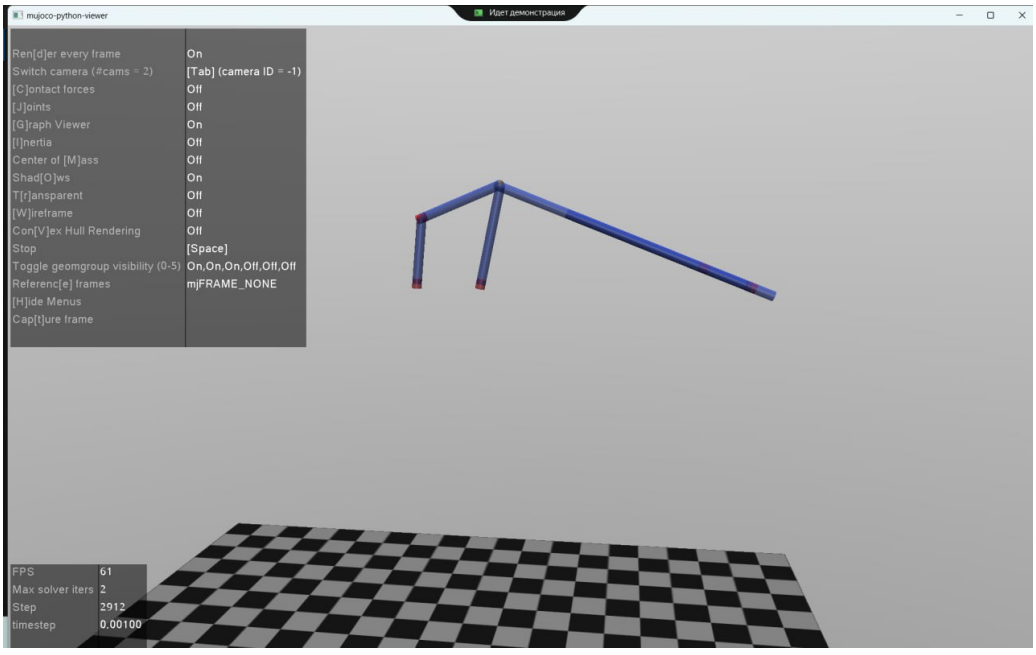

Рис. 1: Схождение механизма после отработки управления слайдером

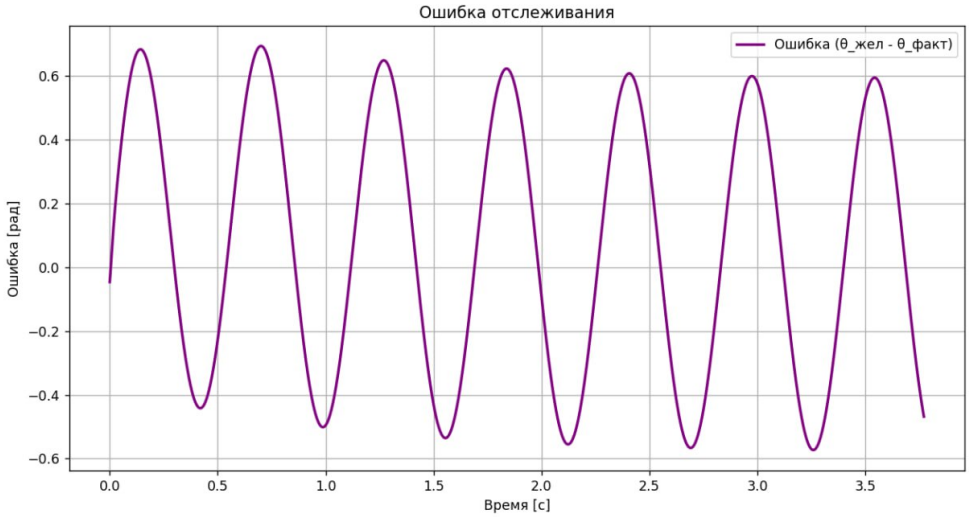

Рис. 2: Ошибка позиционирования

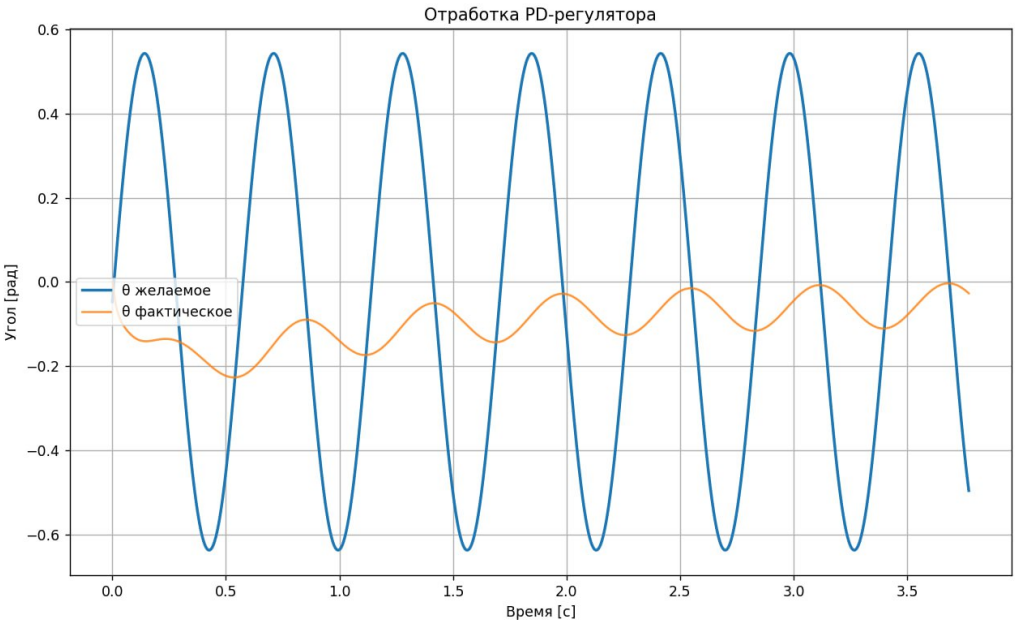

Рис. 3: ОНепосредственная отработка возмущения реуглятором

В ходе работы были получены следующие ошибки:

Таблица 2: Статистика ошибки позиционирования

Параметр	**Значение, рад**
Максимальная ошибка	0.6939
Среднеквадратичная ошибка (RMS)	0.4207
Средняя абсолютная ошибка (MAE)	0.3766

Вывод

Синтезированный регулятор с коэффициентами $K_p = 18$ и $K_d = 15$ продемонстрировал способность эффективно отслеживать заданную синусоидальную траекторию с параметрами 33.79° амплитуды, 1.76 Гц частоты и -2.7° смещения. Анализ точности позиционирования показал максимальное отклонение 0.6939 рад, среднеквадратичную ошибку 0.4207 рад и среднюю абсолютную ошибку 0.3766 рад, что свидетельствует о удовлетворимом качестве управления динамической системой. Полученные результаты подтверждают адекватность выбранных параметров регулятора и целесообразность дальнейшего применения разработанной модели для задач управления в робототехнических и биомеханических системах, включая перспективу исследования более сложных адаптивных алгоритмов управления.