

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

ПРАКТИЧЕСКАЯ РАБОТА №4
по дисциплине
«Имитационное моделирование робототехнических систем»

Студент:
Группа № R4135c

E.A. Щерблюк

Предподаватель: *E.A. Ракшин*

Санкт-Петербург
2025

Цель работы: реализация системы управления на основе ПД-регулятора для плоскостного механизма 2R с сухожильным соединением в среде имитационного моделирования MuJoCo; к модели, созданной в предыдущем задании, необходимо добавить приводы.

Ход работы

Таблица 1 – Задание варианта №73

Вариант	q1			q2		
	AMP, deg	FREQ, Hz	BIAS, deg	AMP, deg	FREQ, Hz	BIAS, deg
TENDON	41.32	3.11	-30.7	24.69	1.41	24.7

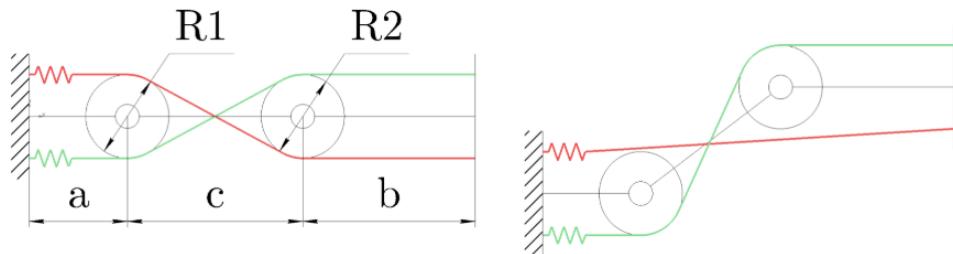


Рисунок 1 – Плоскостной механизм 2R с сухожильным соединением (1 вариант)

По заданному варианту было сформировано XML-описание пассивного механизма, включающее звенья, сайты и две пространственные тросовые передачи (spatial tendons), проходящие через заданные точки крепления и шкивы. Для корректной работы тросовой передачи предусмотрены вспомогательные тела и ограничения (equality), фиксирующие положение промежуточных точек касания на шкивах, а ориентация эффектора удерживается жёсткой связью (weld).

В соответствии с заданием в XML-модель были добавлены два мотор-привода (tendon1_motor, tendon2_motor), управляющие длинами тросов, а также набор сенсоров для измерения текущих длин и скоростей изменения длины тяг, а также положения и скорости подвижного шкива. На основе этой модели создан Python-скрипт, выполняющий загрузку модели, инициализацию данных, вычисление управляющего воздействия с помощью ПД-регулятора $q^{\text{des}} = \text{AMP} \cdot \sin(\text{FREQ} \cdot t) + \text{BIAS}$.

Во время моделирования управляющий сигнал подаётся на оба троса, а положение эффектора считывается с сенсора и используется для построения его траектории. Скриншоты кода приведены в приложении, а результаты моделирования — на рисунках ниже.

```

<actuator>
  <motor name="act_q1" joint="A" gear="1" ctrlrang=-5 5/>
  <motor name="act_q2" joint="B" gear="1" ctrlrang=-5 5/>
</actuator>

<sensor>
  <jointpos name="sens_q1_pos" joint="A"/>
  <jointvel name="sens_q1_vel" joint="A"/>
  <jointpos name="sens_q2_pos" joint="B"/>
  <jointvel name="sens_q2_vel" joint="B"/>
  <framepos objtype="site" objname="effector"/>
</sensor>

```

Рисунок 2 – Часть кода для actuator и sensor

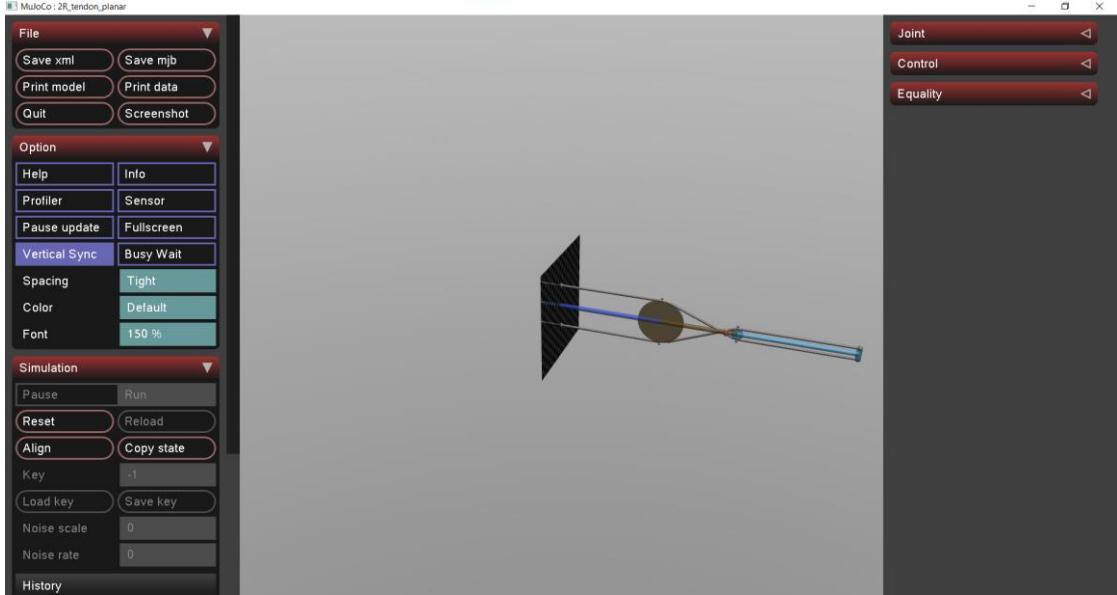


Рисунок 3 – Модель плоскостного механизма 2R с сухожильным соединением

```

pd_q1 = PDController(kp=50, kd=0.3, dt=dt, output_limit=12.0)
pd_q2 = PDController(kp=2, kd=10, dt=dt, output_limit=10.0)

```

Рисунок – 4 Параметры ПД-регулятора

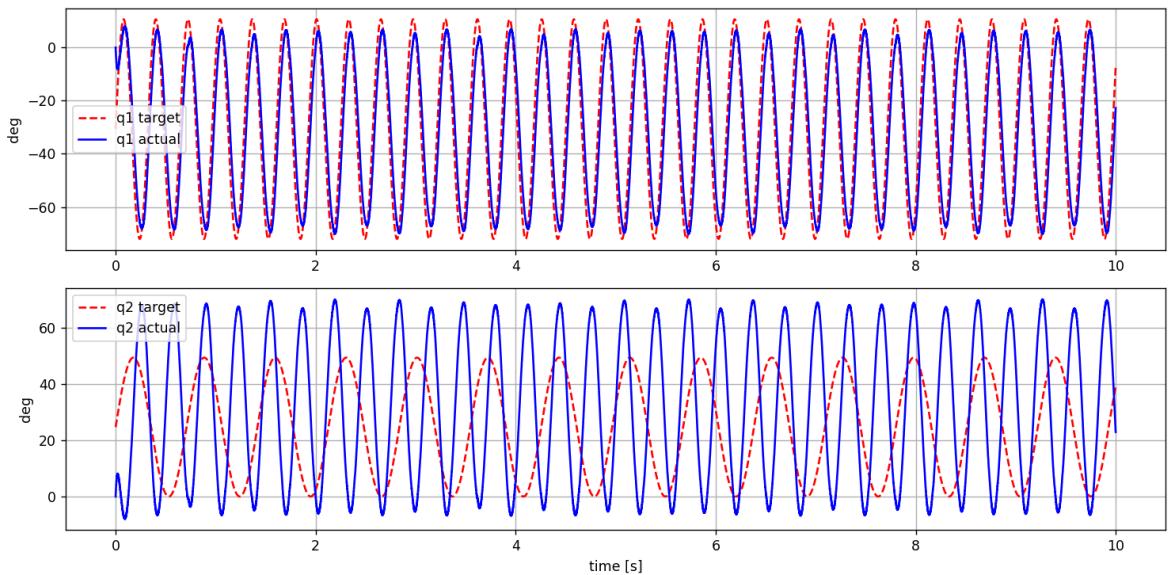


Рисунок 5 – Траектории движения

Вывод: в ходе выполнения практической работы в среде MuJoCo был смоделирован плоский механизм 2R с сухожильным соединением. Проведённое моделирование демонстрирует корректную передачу усилия через тросы и стабильное поведение всех элементов механизма. Управление длинами тяг осуществлялось с помощью ПД-регулятора, формирующего управляющее усилие на основе отклонения текущей длины тяги от заданной синусоидальной траектории. Полученные результаты подтверждают корректность xml файла и его соответствие заданной структуре системы.

ПРИЛОЖЕНИЕ

```
1 import mujoco
2 import mujoco.viewer
3 import numpy as np
4 import time
5 import matplotlib.pyplot as plt
6 |
7 from tendon import generate_tendon_xml
8
9 class PDController:
10     def __init__(self, kp, kd, dt, output_limit=None):
11         self.kp = kp
12         self.kd = kd
13         self.dt = dt
14         self.prev_error = 0
15         self.limit = output_limit
16
17     def calculate(self, target, current):
18         error = target - current
19         derivative = (error - self.prev_error) / self.dt
20         output = self.kp * error + self.kd * derivative
21         self.prev_error = error
22
23         if self.limit:
24             output = np.clip(output, -self.limit, self.limit)
25
26     return output
27
28 params_q1 = {'amp': np.deg2rad(41.32), 'freq': 3.11, 'bias': np.deg2rad(-30.7)}
29 params_q2 = {'amp': np.deg2rad(24.69), 'freq': 1.41, 'bias': np.deg2rad(24.7)}
30
31 def get_target(t, params):
32     return params['amp'] * np.sin(2 * np.pi * params['freq'] * t) + params['bias']
33
34 R1, R2, a, b, c = 0.04, 0.01, 0.091, 0.095, 0.065
35 xml = generate_tendon_xml(R1, R2, a, b, c)
36 model = mujoco.MjModel.from_xml_string(xml)
37 data = mujoco.MjData(model)
38
39 dt = 1e-4
40 sim_time = 10
41 steps = int(sim_time / dt)
42
43 pd_q1 = PDController(kp=4.0, kd=0.1, dt=dt, output_limit=8.0)
44 pd_q2 = PDController(kp=4.0, kd=0.1, dt=dt, output_limit=8.0)
45
46 time_log = []
47 q1_target_log, q1_actual_log = [], []
48 q2_target_log, q2_actual_log = [], []
49
50 with mujoco.viewer.launch_passive(model, data) as viewer:
51     for i in range(steps):
52         t = data.time
53
54         q1_des = get_target(t, params_q1)
55         q2_des = get_target(t, params_q2)
56
57         q1_curr = data.sensordata[0]
58         q2_curr = data.sensordata[2]
59
60         data.ctrl[0] = pd_q1.calculate(q1_des, q1_curr)
61         data.ctrl[1] = pd_q2.calculate(q2_des, q2_curr)
62
63         mujoco.mj_step(model, data)
64
65         if i % 10 == 0:
66             time_log.append(t)
67             q1_target_log.append(q1_des)
68             q1_actual_log.append(q1_curr)
69             q2_target_log.append(q2_des)
70             q2_actual_log.append(q2_curr)
71
72         if i % int(1 / dt / 30) == 0:
73             viewer.sync()
74         if not viewer.is_running():
75             break
76
77 plt.figure(figsize=(12, 6))
78
79 plt.subplot(2, 1, 1)
80 plt.plot(time_log, np.rad2deg(q1_target_log), 'r--', label='q1 target')
81 plt.plot(time_log, np.rad2deg(q1_actual_log), 'b-', label='q1 actual')
82 plt.ylabel('deg'); plt.grid(); plt.legend()
83
84 plt.subplot(2, 1, 2)
85 plt.plot(time_log, np.rad2deg(q2_target_log), 'r--', label='q2 target')
86 plt.plot(time_log, np.rad2deg(q2_actual_log), 'b-', label='q2 actual')
87 plt.ylabel('deg'); plt.xlabel('time [s]'); plt.grid(); plt.legend()
88
89 plt.tight_layout()
90 plt.show()
```

Рисунок 1 – Python код

```

C:\> Users > Eliz > Desktop > tendon.py > generate_tendon_xml
1 def generate_tendon_xml(R1: float, R2: float, a: float, b: float, c: float):
2     stiffness = 100
3
4     # actuators для шарниров A и B (q1 и q2) и сенсоры
5     return f"""
6         <mujoco model="2R_tendon_planar">
7
8             <option timestep="1e-4"/>
9             <option integrator="RK4"/>
10            <option gravity="0 0 0"/>
11
12            <asset>
13                <texture type="skybox" builtin="gradient" rgb1="1 1 1" rgb2="0.5 0.5 0.5" width="265" height="256"/>
14                <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"/>
15                <material name="grid" texture="grid" texrepeat="10 10" reflectance="0.2"/>
16            </asset>
17
18            <worldbody>
19                <light pos="0 0 10"/>
20
21                <camera name="side view" pos="0.1 -1.5 1.0" euler="90 0 0" fovy="60"/>
22                <camera name="upper view" pos="0 0 1.5" euler="0 0 0"/>
23
24                <body name="wall" pos="0 0 0" euler="0 90 0">
25                    <geom type="plane" size="0.05 0.05 0.01" material="grid"/>
26                    <site name="t1_wall" pos="{R1 / 2} 0 0" type="sphere" size="0.002"/>
27                    <site name="t2_wall" pos="{-R1 / 2} 0 0" type="sphere" size="0.002"/>
28                </body>
29
30                <body name="mid_body_t1" pos="{a + c / 2} 0 0">
31                    <site name="t1_mid" pos="0 0 0" type="sphere" size="0.001"/>
32                    <joint name="mid_joint_x_t1" type="slide" axis="1 0 0"/>
33                    <joint name="mid_joint_y_t1" type="slide" axis="0 0 1"/>
34                    <geom type="sphere" size="0.002" mass="0.0001" rgba="0.86 0.43 0.54 0.5" contype="0"/>
35                </body>
36
37                <body name="mid_body_t2" pos="{a + c / 2} 0 0">
38                    <site name="t2_mid" pos="0 0 0" type="sphere" size="0.001"/>
39                    <joint name="mid_joint_x_t2" type="slide" axis="1 0 0"/>
40                    <joint name="mid_joint_y_t2" type="slide" axis="0 0 1"/>
41                    <geom type="sphere" size="0.002" mass="0.0001" rgba="0.86 0.43 0.54 0.5" contype="0"/>
42                </body>
43
44                <body name="effector_link" pos="{a + b + c} 0 0">
45                    <site name="effector_world" pos="0 0 0" type="sphere" size="0.001"/>
46                    <joint name="effector_x" type="slide" axis="1 0 0"/>
47                    <joint name="effector_y" type="slide" axis="0 0 1"/>
48                    <geom type="sphere" size="0.002" mass="0.0001" rgba="0.86 0.43 0.54 0.5" contype="0"/>
49                </body>
50
51                <body name="link1" pos="0 0 0" euler="0 0 0">
52                    <geom type="cylinder" pos="{a / 2} 0 0" size="0.002 {a / 2}" euler="0 90 0" rgba="0.21 0.32 0.82 0.5" contype="0"/>
53
54                <body name="link2" pos="{a} 0 0" euler="0 0 0">
55                    <!-- Joint A (q1) -->
56                    <joint name="A" type="hinge" axis="0 1 0" stiffness="0" springref="0" damping="0.05"/>
57                    <geom type="cylinder" pos="{c / 2} 0 0" size="0.002 {c / 2}" euler="0 90 0" rgba="0.42 0.32 0.12 0.5" contype="0"/>
...

```

Рисунок 2 – Xml файл часть 1

```

59         <geom name="pulley1" type="cylinder" size="{R1 / 2} 0.001" pos="0 0 0" euler="90 0 0" rgba="0.42 0.32 0.12 0.5" contype="0"/>
60         <site name="side_r1_t1" pos="0 0 {-R1 / 2 - 0.002}" type="sphere" size="0.001"/>
61         <site name="side_r1_t2" pos="0 0 {R1 / 2 + 0.002}" type="sphere" size="0.001"/>
62         <site name="pulley1_side" pos="0 0 0" type="sphere" size="0.001"/>
63             <body name="link3" pos="[{c} 0 0" euler="0 0 0">
64                 <!-- Joint B (q2) -->
65                 <joint name="B" type="hinge" axis="0 1 0" stiffness="0" springref="0" damping="0.05"/>
66                 <geom type="cylinder" pos="{b / 2} 0 0" size="0.002 {b / 2}" euler="0 90 0" rgba="0.34 0.65 0.84 0.5" contype="0"/>
67                 <geom type="box" pos="{b} 0 0" size="0.002 0.002 {R2 / 2}" rgba="0.34 0.65 0.84 0.5" mass="0" contype="0"/>
68
69                 <site name="t1_end" pos="{b} 0 {R2 / 2}" type="sphere" size="0.002"/>
70                 <site name="t2_end" pos="{b} 0 {-R2 / 2}" type="sphere" size="0.002"/>
71
72                 <geom name="pulley2" type="cylinder" size="{R2 / 2} 0.001" pos="0 0 0" euler="90 0 0" rgba="0.34 0.65 0.84 0.5" contype="0"/>
73                 <site name="side_r2_t1" pos="0 0 {R2 / 2 + 0.002}" type="sphere" size="0.001"/>
74                 <site name="side_r2_t2" pos="0 0 {-R2 / 2 - 0.002}" type="sphere" size="0.001"/>
75                 <site name="pulley2_side" pos="0 0 0" type="sphere" size="0.001"/>
76
77                 <site name="effector" pos="{b} 0 0" type="sphere" size="0.001"/>
78             </body>
79         </body>
80     </worldbody>
81
82     <tendon>
83         <spatial name="tendon1_1" width="0.001" stiffness="{stiffness}" damping="10" springlength="0.005">
84             <site site="t1_wall"/>
85             <geom geom="pulley1" sidesite="side_r1_t1"/>
86             <site site="t1_mid"/>
87             <geom geom="pulley2" sidesite="side_r2_t1"/>
88             <site site="t1_end"/>
89         </spatial>
90     </tendon>
91
92     <tendon>
93         <spatial name="tendon2_1" width="0.001" stiffness="{stiffness}" damping="10" springlength="0.005">
94             <site site="t2_wall"/>
95             <geom geom="pulley1" sidesite="side_r1_t2"/>
96             <site site="t2_mid"/>
97             <geom geom="pulley2" sidesite="side_r2_t2"/>
98             <site site="t2_end"/>
99         </spatial>
100    </tendon>
101
102    <equality>
103        <weld site1="effector" site2="effector_world" torquescale="100"/>
104
105        <connect site1="t1_mid" site2="pulley1_side"/>
106        <connect site1="t1_mid" site2="pulley2_side"/>
107        <connect site1="t2_mid" site2="pulley1_side"/>
108        <connect site1="t2_mid" site2="pulley2_side"/>
109    </equality>
110
111    <!-- ЗАДАНИЕ: Добавляем Actuator и Sensor для q1 и q2 -->
112    <actuator>
113        <!-- Управляем шарнирами А и В напрямую для выполнения траектории q1, q2 -->
114        <motor name="act_q1" joint="A" gear="1" ctrlrange="-5 5"/>
115        <motor name="act_q2" joint="B" gear="1" ctrlrange="-5 5"/>
116    </actuator>
117
118    <sensor>
119        <jointpos name="sens_q1_pos" joint="A"/>
120        <jointvel name="sens_q1_vel" joint="A"/>
121        <jointpos name="sens_q2_pos" joint="B"/>
122        <jointvel name="sens_q2_vel" joint="B"/>
123        <framepos objtype="site" objname="effector"/>
124    </sensor>

```

Рисунок 3 – Xml файл часть 2