

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

Практическое задание №4  
по дисциплине  
*«Имитационное моделирование робототехнических систем»*

Студент:  
*Группа № R4133C*  
*Звонков Г.Е*

Преподаватель:  
*Ракшин. Е.А*

Санкт-Петербург  
2025

## Цели и задачи работы

Дополнить модель из 3 лабораторной приводами, синтезировать ПД регулятор.  $q^{des} = AMP \cdot \sin(FREQ \cdot t) + BIAS$

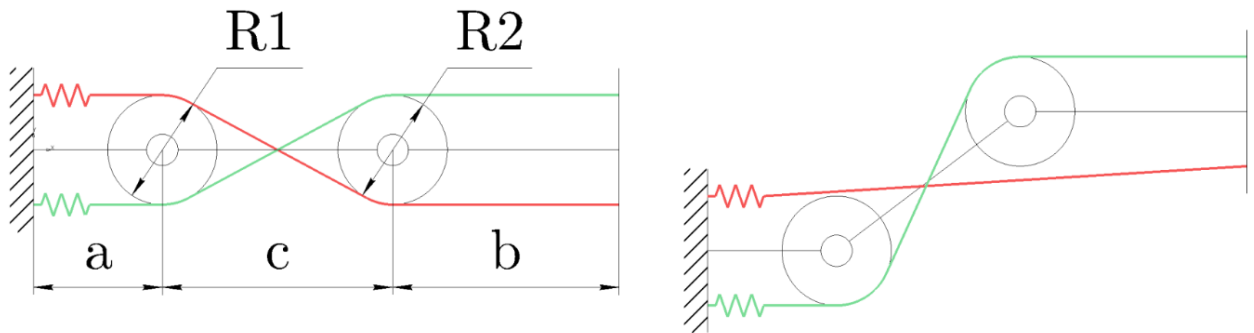


Рисунок 1 – 2R механизм

Таблица №1 – параметры механизма

$q_i$	AMP	FREQ	BIAS
$q_1$	36,33	2,5	43,5
$q_2$	33,54	2,08	-38,6

Дополним код модели добавив актуаторы и сенсоры

```

<sensor>

  <framepos objtype="site" objname="carriage"/>

  <!-- Сенсоры позиции и скоростей-->

  <jointpos joint="A" name="q1"/>
  <jointvel joint="A" name="dq1"/>
  <jointpos joint="B" name="q2"/>
  <jointvel joint="B" name="dq2"/>

</sensor>

<actuator>

  <motor name="motor_t1" joint="A" gear="3" />
  <motor name="motor_t2" joint="B" gear="3" />

</actuator>
  
```

Уравнение ПД регулятора

$$\tau = Kp \cdot e + Kd \cdot \dot{e}$$

Где  $\dot{e} = (\dot{q}_r - \dot{q})$ ,  $Kp = [150 \ 260]$ ,  $Kd = [3 \ 4]$

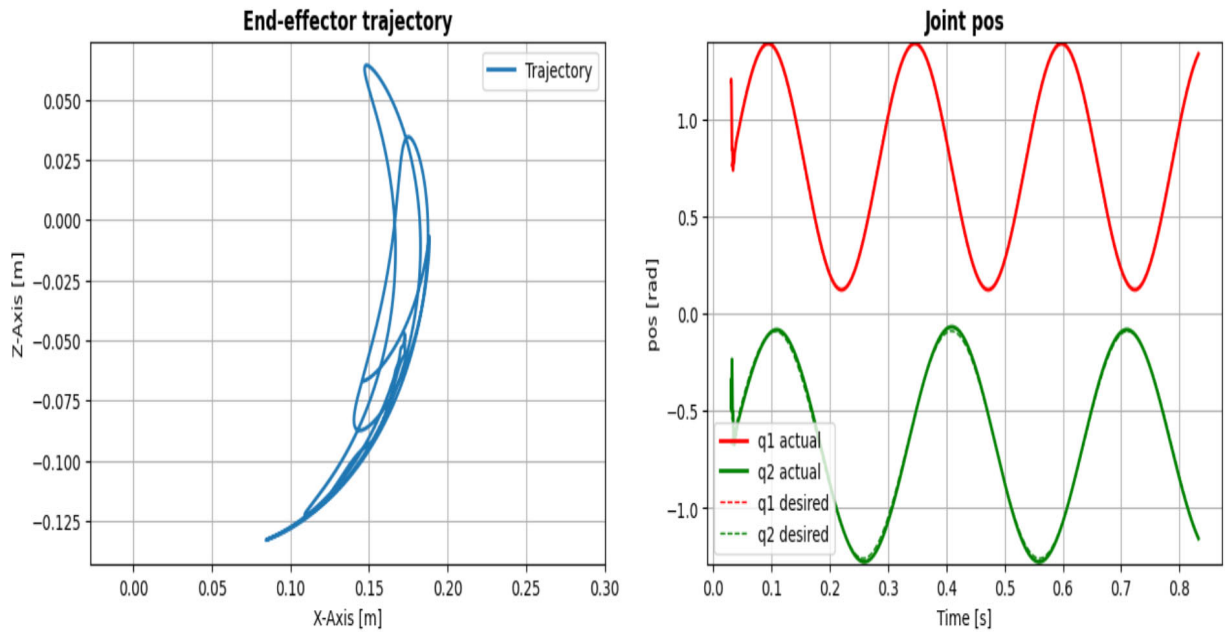


Рисунок 2 –Графики позиции Энд-эффектора и углов звеньев

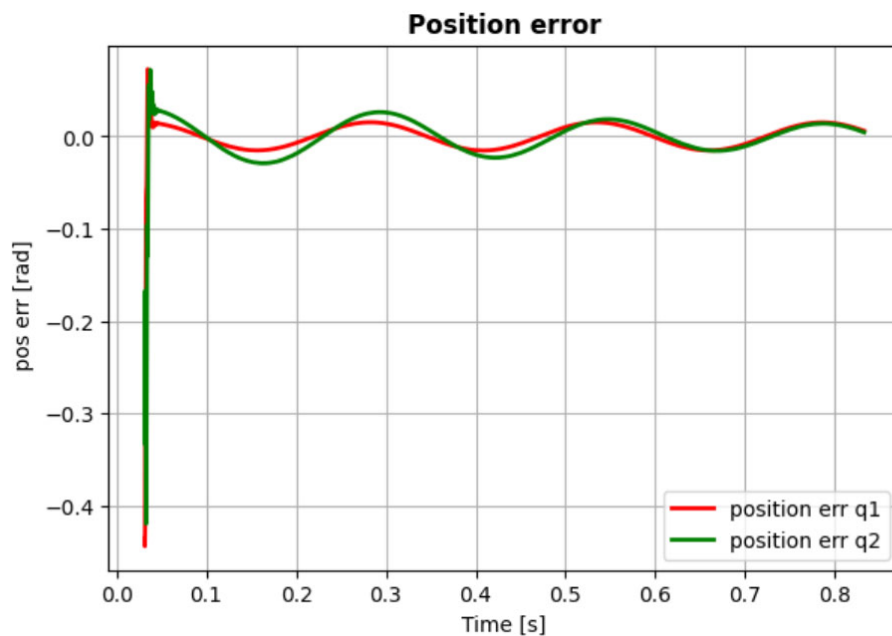


Рисунок 3 – График ошибки позиционирования

## **Выводы**

В результате работы была доработана модель 2R механизма в среде MuJoCo путем добавления актуаторов и синтеза ПД регулятора движения.

Коэффициенты регулятора были подобраны эмпирическим путем, при которых получилось добиться наименьшей ошибки позиционирования.

## Приложение А

### Листинг А.1 – XML

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

<mujoco>

<!-- Параметры 2R механизма -->

R1, m  R2, m  a, m  b, m  c, m
0.045  0.023  0.052  0.084  0.053


  <option timestep="1e-4"/>
  <option integrator="RK4"/>
  <option gravity="0 0 -9.8"/>

<!-- Настройки сцены -->

  <asset>

    <texture type="skybox" builtin="gradient" rgb1="0.85 0.9 1" rgb2="0.65 0.7 0.8" 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"/>

    <material name="init_mat" rgba="0.85 0.85 0.85 1"/>

    <material name="bright_blue" rgba="0.2 0.6 1 1"/>

  </asset>

<!-- Создание Объектов -->

  <worldbody>

    <light pos="0 0 10"/>

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

    <camera name="side view" pos="0.1 -1.5 1.0" euler="0 90 0" fovy="60"/>

    <camera name="upper view" pos="0 0 1.5" euler="0 0 0"/>

    <!-- Начальная плоскость -->

    <body name="Init" pos="0 0 0" euler="0 90 0">

      <geom type="box" size=" 0.05 0.01 0.002" material="bright_blue"/>

      <site name="Tendon1_pos" pos="0.0225 0 0" type="sphere" size="0.001"/>

      <site name="Tendon2_pos" pos="-0.0225 0 0" type="sphere" size="0.001"/>

    </body>

  </worldbody>

</mujoco>
```

```

<!-- Создание объекта для удержания каретки в горизонтальном положении -->

<body name="carriage_vertical" pos="0.189 0 0">

  <site name="carriage_weld" pos="0 0 0" type="sphere" size="0.001"/>

  <joint name="cv_x" type="slide" axis="1 0 0"/>

  <joint name="cv_y" type="slide" axis="0 0 1"/>

  <geom type="box" size="0.0002 0.0002 0.0002 " mass="0.01" contype="0"/>

</body>

<!-- Создание вложенных объектов, описывающих систему 2R механизма-->

<!-- Первое звено -->

<body name = "Link1" pos = '0 0 0' euler = '0 0 0'>

  <geom type = "cylinder" pos = "0.026 0 0" euler = "0 90 0" mass = "0.01" size="0.002 0.026" rgba="0 0 1
0.5"/>

  <!-- Второе звено -->

  <body name = "Link2" pos = '0.052 0 0' euler = '0 0 0'>

    <joint name="A" type="hinge" axis="0 1 0" stiffness="0.1" damping="0.5"/>

    <geom type="cylinder" mass = "0.01" pos="0.0265 0 0" size="0.002 0.0265" euler="0 90 0" rgba="0 0
1 0.5"/>

    <!-- Блок цилиндр 1 -->

    <geom name="Block1" type="cylinder" size="0.0225 0.01" pos="0 0 0" euler="90 0 0" rgba="1 0.5 0
0.5" mass = "0.001" />

    <site name="Tendon2_b1" pos="0 0 -0.022515" type="sphere" size="0.001"/>

    <site name="Tendon1_b1" pos="0 0 0.022515" type="sphere" size="0.001"/>

    <site name="SB1" pos="0 0 0" type="sphere" size="0.001"/>

    <!-- Третье звено -->

    <body name = "Link3" pos = '0.053 0 0' euler = '0 0 0'>

      <!-- #Создаем Joint -->

      <joint name="B" type="hinge" axis="0 1 0" stiffness="0.1" damping="0.5" />

      <geom type="cylinder" pos="0.042 0 0" size="0.002 0.042" euler="0 90 0" rgba="0 0 1 0.5" mass =
"0.01" />

      <!-- Блок цилиндр 2 -->

      <geom name="Block2" type="cylinder" size="0.0115 0.01" pos="0 0 0" euler = "90 0 0" mass =
"0.01" rgba="1 0.5 0 0.5"/>

      <site name="Tendon1_b2" pos="0 0 0.01150001" type="sphere" size="0.001"/>

      <site name="Tendon2_b2" pos="0 0 -0.01150001" type="sphere" size="0.001"/>

      <site name="SB2" pos="0 0 0" type="sphere" size="0.001"/>

```

```

<site name="carriage" pos="0.084 0 0" type="sphere" size="0.001"/>

<!-- #Создаем Каретку -->

<body name = "Carriage" pos = '0.084 0 0' euler = '0 0 0'>

    <geom type="box" size="0.002 0.002 0.0115" pos="0 0 0" rgba="0.9 0.9 0.9 0.5" mass = "0.01"/>

    <!-- #Добавляем sites для тендонов каретки-->

    <site name="Tendon1_end" pos="0 0 0.0115" type="sphere" size="0.001"/>

    <site name="Tendon2_end" pos="0 0 -0.0115" type="sphere" size="0.001"/>

</body>

</body>

</body>

<!-- Фиктивные точки для корректного построения маршрута тендона -->

<body name="Fictionous point1" pos="0.0785 0 0" >

    <geom type="box" size="0.00001 0.0001 0.0001" contype="0" mass = "0.001"/>

    <site name="FP1" pos="0 0 0" type="cylinder" size="0.0001 0.00001"/>

    <joint name="F1x" type="slide" axis="1 0 0"/>

    <joint name="F1y" type="slide" axis="0 0 1"/>

</body>

</worldbody>

<!-- Тендон красный -->

<tendon>

    <spatial name = "Tendon1" width = "0.001" stiffness = "10" damping = "2" springlength = "0.005" rgba="1 0
0 1">

        <site site = "Tendon2_pos"/>

        <geom geom="Block1" sidesite="Tendon1_b1"/>

        <site site = "FP1"/>

        <geom geom="Block2" sidesite="Tendon2_b2"/>

        <site site="Tendon2_end"/>

    </spatial>

</tendon>

<!-- Тендон зелёный -->

<tendon>

```

```

    <spatial name = "Tendon2" width = "0.001" stiffness = "10" damping = "2" springlength = "0.005" rgba="0 1
0 1">

        <site site = "Tendon1_pos"/>

        <geom geom="Block1" sidesite="Tendon2_b1"/>

        <site site = "FP1"/>

        <geom geom="Block2" sidesite="Tendon1_b2"/>

        <site site="Tendon1_end"/>

    </spatial>

</tendon>

<!-- Ограничение удерживающее каретку в горизонтальном положении -->
<equality>

    <connect site1="FP1" site2="SB1"/>

    <connect site1="FP1" site2="SB2"/>

</equality>

<!-- Привода тендонов -->

<actuator>

    <motor name="motor_t1" joint="A" gear="3" />

    <motor name="motor_t2" joint="B" gear="3" />

</actuator>

<sensor>

    <framepos objtype="site" objname="carriage"/>

    <!-- Сенсоры позиции тендонов -->

    <jointpos joint="A" name="q1"/>

    <jointvel joint="A" name="dq1"/>

    <jointpos joint="B" name="q2"/>

    <jointvel joint="B" name="dq2"/>

</sensor>

</mujoco>

```



## Листинг А.2 – Код решения

```
SIMEND = 20

TIMESTEP = 0.001

STEP_NUM = int(SIMEND / TIMESTEP)

timeseries = np.linspace(0, SIMEND, STEP_NUM)


A1 = np.deg2rad(36.33)
A2 = np.deg2rad(33.54)

F1 = 2.5
F2 = 2.08

B1 = np.deg2rad(43.5)
B2 = np.deg2rad(-38.6)

q_d1 = A1 * np.sin(F1 * timeseries) + B1
q_d2 = A2 * np.sin(F2 * timeseries) + B2
dq_d1 = A1 * F1 * np.cos(F1 * timeseries)
dq_d2 = A2 * F2 * np.cos(F2 * timeseries)


site_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SITE, "carriage")
q1_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SENSOR, "q1")
q2_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SENSOR, "q2")
q1_v_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SENSOR, "dq1")
q2_v_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SENSOR, "dq2")


KP = [150, 260]
KV = [0.3, 0.4]

joint_A_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_JOINT, "A")
joint_B_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_JOINT, "B")


def set_torque(mj_data, KP, KV, q_desired, dq_des):

    tendon1_pos = mj_data.qpos[joint_A_id]
    tendon2_pos = mj_data.qpos[joint_B_id]

    tendon1_vel = mj_data.sensordata[q1_v_id]
    tendon2_vel = mj_data.sensordata[q2_v_id]
```

```

data.ctrl[0] = KP[0] * (q_desired[0] - tendon1_pos) + KV[0] * (dq_des[0] - tendon1_vel)
data.ctrl[1] = KP[1] * (q_desired[1] - tendon2_pos) + KV[1] * (dq_des[1] - tendon2_vel)

EE_position_x = []
EE_position_z = []
q1_pos = []
q2_pos = []
simulation_times = []
completed_steps = 0
EE_position_x = []
EE_position_z = []
site_id = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SITE, "carriage")
viewer = mujoco_viewer.MujocoViewer(model,
                                     data,
                                     title="2R",
                                     width=1920,
                                     height=1080)

for i in range(STEP_NUM):
    if viewer.is_alive:

        set_torque(data, KP, KV, [q_d1[i], q_d2[i]], [dq_d1[i], dq_d2[i]])

        position_EE = data.site_xpos[site_id]
        EE_position_x.append(position_EE[0])
        EE_position_z.append(position_EE[2])
        # Сравним сенсоры и qpos
        current_q1 = data.qpos[joint_A_id]
        current_q2 = data.qpos[joint_B_id]

        q1_pos.append(current_q1)
        q2_pos.append(current_q2)

```

```

simulation_times.append(data.time)

mujoco.mj_step(model, data)

viewer.render()

completed_steps += 1

else:
    break

viewer.close()

q_d1_completed = q_d1[:completed_steps]
q_d2_completed = q_d2[:completed_steps]
midlength = int(STEP_NUM/2)

plt.figure(figsize=(15, 10))

# График 1: Траектория конечного эффектора
plt.subplot(2, 2, 1)
plt.plot(EE_position_x, EE_position_z, '-', linewidth=2, label='Trajectory')
plt.title('End-effector trajectory', fontsize=12, fontweight='bold')
plt.xlabel('X-Axis [m]')
plt.ylabel('Z-Axis [m]')
plt.axis('equal')
plt.grid()
plt.legend()

# График 2:
plt.subplot(2, 2, 2)
plt.plot(simulation_times, q1_pos, 'r-', linewidth=2, label='q1 actual')
plt.plot(simulation_times, q2_pos, 'g-', linewidth=2, label='q2 actual')
plt.plot(simulation_times, q_d1_completed, 'r--', linewidth=1, label='q1 desired')
plt.plot(simulation_times, q_d2_completed, 'g--', linewidth=1, label='q2 desired')
plt.title('Joint pos', fontsize=12, fontweight='bold')
plt.xlabel('Time [s]')
plt.ylabel('pos [rad]')

```

```

plt.grid()

plt.legend()

plt.ylim(min(min(q1_pos), min(q2_pos), min(q_d1_completed), min(q_d2_completed)) - 0.01,
          max(max(q1_pos), max(q2_pos), max(q_d1_completed), max(q_d2_completed)) + 0.01)

e1 = q_d1_completed-q1_pos
e2 = q_d2_completed-q2_pos

plt.subplot(2, 2, 3)

plt.plot(simulation_times, e1, 'r-', linewidth=2, label='position err q1')
plt.plot(simulation_times, e2, 'g-', linewidth=2, label='position err q2')
plt.title('Position error', fontsize=12, fontweight='bold')
plt.xlabel('Time [s]')
plt.ylabel('pos err [rad]')

plt.grid()

plt.legend()

```