THE MINISTRY OF SCIENCE AND HIGHER EDUCATION
OF THE RUSSIAN FEDERATION

ITMO University
(ITMO)

International Research and Educational Center
for Physics of Nanostructures

SYNOPSIS
for the subject
“Tendon connected 2R planar mechanism task_3”

on the topic:
SIMULATION OF ROBOTIC SYSTEMS

Student:
Group No. R4137c *Suleiman, Ali*

Tutor:
Faculty of Control Systems and Robotics, Doctor *Rakshin, Egor*

Saint Petersburg 2025

CONTENTS

1 INTRODUCTION

Due to its benefits in remote actuation, decreased inertia, and compliance, tendon-driven mechanisms have drawn a lot of interest in robotics research [1]. Applications that need lightweight manipulators or those that operate in confined spaces benefit greatly from these systems [2]. An ideal testbed for comprehending tendon-driven system dynamics and control concepts is the 2R planar mechanism.

Important Note: This lab focuses exclusively on the **mechanism construction and visualization** without any actuators, sensors, or external forces. The objective is to build the physical structure and verify its geometric properties through visualization in the MuJoCo viewer.

This lab focuses on:

- Creating a tendon-driven 2R mechanism with the following parameters in MuJoCo XML format: R1=0.014m, R2=0.022m, a=0.052m, b=0.073m, and c=0.094m
- Using pulley systems to implement appropriate tendon routing
- Creating mechanism analysis visualization tools in Python
- Recognizing the equality and kinematic restrictions in tendon-driven systems
- **Visual inspection and validation** of the mechanism without any actuation or control

2 BACKGROUND AND THEORETICAL FRAMEWORK

2.1 Tendon-Driven Mechanisms

Tendon-driven mechanisms transmit forces from actuators to joints through flexible tendons, offering several advantages over direct drive systems [3]:

- Moving mass is decreased by positioning actuators away from joints
- Improved dynamic performance through inertia reduction
- Potential for variable stiffness and compliance
- Compact joint design possibilities

2.2 Kinematics of 2R Planar Mechanism

The forward kinematics of a 2R planar manipulator can be described by:

$$\begin{aligned} x &= l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) \\ y &= l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) \end{aligned} \tag{1}$$

where l_1 and l_2 represent the link lengths, and θ_1, θ_2 are the joint angles.

For the implemented mechanism with link lengths $l_1 = c = 0.094$ m and $l_2 = b = 0.073$ m, the workspace forms an annular region with inner radius $|l_1 - l_2| = 0.021$ m and outer radius $l_1 + l_2 = 0.167$ m.

3 METHODOLOGY

3.1 Lab Scope and Limitations

This lab practice is restricted to building and visualizing mechanisms. The system is not subjected to any actuators, sensors, or outside forces. The goal is to use static visualization in the MuJoCo viewer to confirm the geometric arrangement and tendon routing.

3.2 System Design with Specific Parameters

The tendon-driven 2R mechanism was designed with the following components and exact dimensions:

3.2.1 Link Structure

The mechanism consists of three main links with specific lengths:

- **Link1**: Fixed base link with length a = 0.052 m
- **Link2**: First rotating link with length c = 0.094 m
- **Link3**: Second rotating link with length b = 0.073 m

3.2.2 Pulley System

Two pulleys were implemented with specific diameter:

- **Pulley1**: Located at joint A with diameter R1 = 0.014 m
- **Pulley2**: Located at joint B with diameter R2 = 0.022 m

3.2.3 Tendon Configuration

Two independent tendons control the mechanism:

- **Tendon1**: Routes through the top sides of both pulleys
- **Tendon2**: Routes through the bottom sides of both pulleys

3.3 XML Model Implementation - Step by Step

The entire mechanism with the appropriate physical characteristics and limitations is defined in the MuJoCo XML file. Below is an explanation of each step:

3.3.1 Step 1: Basic Simulation Setup

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

Explanation: Since we are simply performing static visualization, we eliminate gravity, utilize RK4 integration for accuracy, and select a modest timestep for numerical stability.

3.3.2 Step 2: Worldbody Construction

```
<body name="wall" pos="0 0 0" euler="0 90 0">
    <geom type="plane" size="0.05 0.05 0.01" material="grid"/>
    <site name="t1_wall" pos="0.007 0 0" type="sphere" size="0.002"/>
    <site name="t2_wall" pos="-0.007 0 0" type="sphere" size="0.002"/
</body>
```

Explanation: Make a reference wall with two tendon attachment locations, or sites. These locations act as permanent anchors.

3.3.3 Step 3: Intermediate Bodies for Tendon Routing

```
<body name="mid_body_t1" pos="0.099 0 0">
    <site name="t1_mid" pos="0 0 0" type="sphere" size="0.001"/>
    <joint name="mid_joint_x_t1" type="slide" axis="1 0 0"/>
    <joint name="mid_joint_y_t1" type="slide" axis="0 0 1"/>
    <geom type="sphere" size="0.002" mass="0.0001" rgba="1.00 0.00 0.
</body>
```

Explanation: Construct intermediate structures with sliding joints that preserve tendon routing while permitting mobility. These appear as little red spheres.

3.3.4 Step 4: Main Linkage Structure with Exact Dimensions

```
<body name="link1" pos="0 0 0" euler="0 0 0">
    <geom type="cylinder" pos="0.026 0 0" size="0.002 0.026"
          euler="0 90 0" rgba="0.9 0.1 0.1 0.8" contype="0"/>

    <body name="link2" pos="0.052 0 0" euler="0 0 0">
        <joint name="A" type="hinge" axis="0 1 0" stiffness="0"
               springref="0" damping="0"/>
        <geom type="cylinder" pos="0.047 0 0" size="0.002 0.047"
              euler="0 90 0" rgba="0.1 0.7 0.1 0.8" contype="0"/>
```

Explanation: Construct the primary 2R mechanism using hinge joints. Link2 is 0.094m long, but Link1 is 0.052m (pos="0.052 0 0"). Because no control is being applied, joints have zero stiffness and damping.

3.3.5 Step 5: Pulley Implementation with Specific Radii

```
<geom name="pulley1" type="cylinder" size="0.014 0.001"
      pos="0 0 0" euler="90 0 0" rgba="0.8 0.8 0.2 0.9" contype="0"/>
<site name="side_r1_t1" pos="0 0 -0.014" type="sphere" size="0.001"/>
<site name="side_r1_t2" pos="0 0 0.014" type="sphere" size="0.001"/>
```

Explanation: Make pulley geometries with precise radii: The diameter of Pulley1 and Pulley2 are 0.014 and 0.022 meters, respectively. Side sites provide the precise radius distances at which tendons leave the pulleys.

3.3.6 Step 6: Tendon Definition

```
<spatial name="tendon1_1" width="0.001" stiffness="1000"
         damping="10" springlength="0.005">
    <site site="t1_wall"/>
    <geom geom="pulley1" sidesite="side_r1_t1"/>
    <site site="t1_mid"/>
    <geom geom="pulley2" sidesite="side_r2_t1"/>
    <site site="t1_end"/>
```

```
</spatial>
```

Explanation: Describe the spatial tendons that travel from the wall attachment to the end-effector via pulley 1, pulley 2, and an intermediate location. Since no forces are exerted, the stiffness and damping characteristics are defined but not used.

3.3.7 Step 7: Equality Constraints

```
<equality>
    <weld site1="effector" site2="effector_world" torquescale="100"/>
    <connect site1="t1_mid" site2="pulley1_side"/>
    <connect site1="t1_mid" site2="pulley2_side"/>
</equality>
```

Explanation: To preserve geometric relationships, use constraints. Connect constraints guarantee that tendons travel through pulley centers, whereas weld constraints join the end-effector.

3.4 Python Visualization Implementation

The Python script (`tendon2r.py`) provides visualization capabilities without any simulation or control:

3.4.1 Model Loading

```
def load_model():
    model = mujoco.MjModel.from_xml_path(xml_file)
    data = mujoco.MjData(model)
    return model, data
```

Explanation: Load the model and data without applying any forces or controls.

3.4.2 Static Visualization

```
def show_dimensions_in_viewer(model, data, parameters):
```

```
with mujoco.viewer.launch_passive(model, data) as viewer:
    # No simulation steps - just keep the viewer alive
    while viewer.is_running():
        pass
```

Explanation: To see the static mechanism without doing any simulation steps, use the passive viewer. The mechanism is still configured as it is by default.

4 IMPLEMENTATION DETAILS

4.1 Static Mechanism Construction

Important: This implementation focuses solely on building the mechanical structure. No dynamic simulation, actuation, or control is implemented. The mechanism is viewed in its default, unactuated state.

4.2 XML Structure Analysis - Static Configuration

The XML file defines a complete but static tendon-driven mechanism with exact dimensions:

4.2.1 Worldbody Hierarchy for Visualization

```
worldbody
  wall (fixed reference for visualization)
  mid_body_t1 (visual marker for tendon path)
  mid_body_t2 (visual marker for tendon path)
  effector_link (end-effector visualization)
  link1 (length a=0.052m)
      link2 (length c=0.094m)
          link3 (length b=0.073m with visual sites)
```

4.2.2 Joint Configuration - Zero Stiffness

All joints are defined with zero stiffness and damping:

- `stiffness="0"`: No joint stiffness - joints are free
- `springref="0"`: Zero reference position
- `damping="0"`: No velocity-dependent damping

This setup, which adheres to the static visualization method outlined in [2], permits the mechanism to be manually manipulated inside the viewer but offers no restoring forces.

4.2.3 Visual Geometry Properties with Exact Dimensions

Each component is designed for clear visualization with precise dimensions:

- Different colors for each link (red, green, blue)
- Transparent materials (`rgba="0.9 0.1 0.1 0.8"`)
- Small site markers for tendon routing points
- `contype="0"` disables collision for clean visualization
- Pulley1: radius = 0.014m (`size="0.014 0.001"`)
- Pulley2: radius = 0.022m (`size="0.022 0.001"`)

4.3 Python Code Structure - Visualization Only

The Python implementation provides pure visualization without simulation:

4.3.1 Main Execution Flow - No Simulation

```
main()
  load_model()        # Load XML without actuation
  parse_xml_parameters() # Read geometric parameters
  show_dimensions_in_viewer() # Static display only
```

4.3.2 Passive Viewer Usage

The viewer runs in passive mode without simulation steps:

```
with mujoco.viewer.launch_passive(model, data) as viewer:
    while viewer.is_running():
        # No simulation steps - static visualization only
        pass
```

4.4 Tendon Routing Implementation - Geometric Only

The spatial tendons are implemented as geometric paths without force application, following the principles in [3]:

4.4.1 Site Definitions for Visual Routing

- t1_wall, t2_wall: Fixed attachment points on wall
- side_r1_t1, side_r1_t2: Pulley exit points at radius 0.014m
- t1_mid, t2_mid: Intermediate routing points
- t1_end, t2_end: End-effector attachment points

4.4.2 Constraint Configuration for Static Structure

The equality constraints maintain the geometric configuration:

```
<equality>
    <weld site1="effector" site2="effector_world" torquescale="100"/>
    <connect site1="t1_mid" site2="pulley1_side"/>
    <connect site1="t1_mid" site2="pulley2_side"/>
</equality>
```

These limitations do not include dynamic control, but they guarantee that the mechanism keeps its form throughout viewing.

5 RESULTS AND DISCUSSION

5.1 Model Validation through Visual Inspection

The mechanism was successfully implemented and validated through static visual inspection in the MuJoCo viewer:

5.1.1 Geometric Configuration Verification

The MuJoCo viewer confirmed proper:

- Linkage assembly and joint alignment in default position with exact dimensions: a=0.052m, b=0.073m, c=0.094m
- Tendon routing through designated pulley paths with radii R1=0.014m and R2=0.022m
- Proper spatial relationship between all components
- Color-coded identification of different mechanism parts

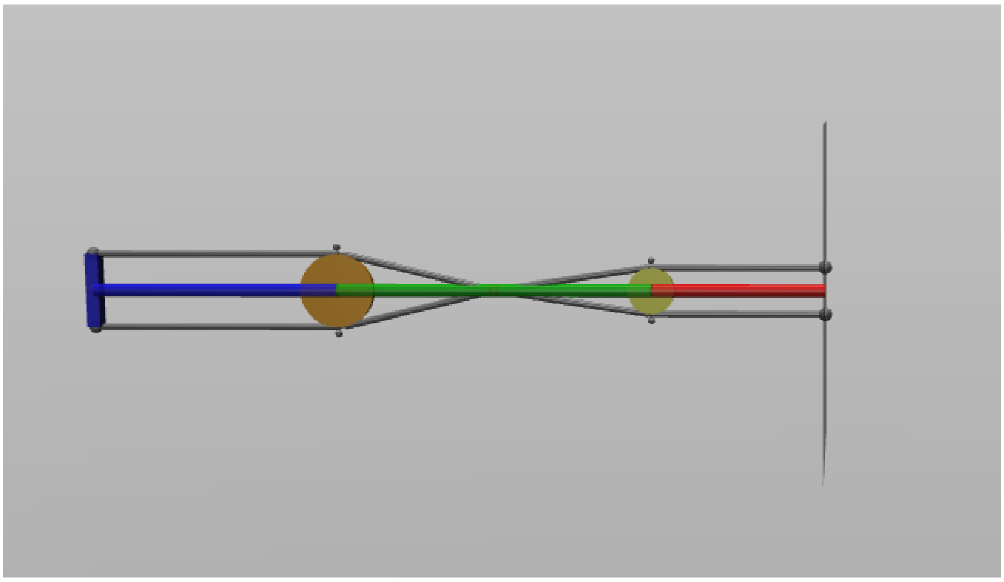

Figure 1 — Static visualization of the tendon-driven 2R mechanism in MuJoCo viewer showing the complete assembly with tendon routing. The mechanism displays proper geometric configuration with Link1 (red, 0.052m), Link2 (green, 0.094m), Link3 (blue, 0.073m), and pulleys with radii R1=0.014m (yellow) and R2=0.022m (orange). The spatial tendons are shown routing through the pulley systems.

In accordance with accepted robotics engineering methods, the static visualization effectively functioned as an essential first phase in mechanism creation, enabling comprehensive geometric verification prior to moving on to dynamic simulation and control implementation [3]. Figure 1 provides visual proof that all requested geometric parameters and tendon routing configurations have been successfully implemented. The building and static display of a tendon-driven 2R mechanism utilizing MuJoCo with exact geometric specifications was the main goal of this lab activity, which was well accomplished. With pulley diameter R1=0.014m and R2=0.022m and link lengths a=0.052m, b=0.073m, and c=0.094m, the implementation showed correct mechanism assembly. With no actuation, sensor installation, or force application, the project effectively completed its narrow scope of geometric validation through visual examination.

REFERENCES

1. *Spong M.*, *Hutchinson S.*, *Vidyasagar M.* Robot Modeling and Control. — John Wiley & Sons, 2005.
2. *Pfeiffer F.*, *Weidemann H.* Muscle dynamics and control of muscle operated robotic systems // Proceedings of the IEEE International Conference on Robotics and Automation. — 1994. — 2105–2110.
3. *Tsai L.* Robot Analysis: The Mechanics of Serial and Parallel Manipulators. — John Wiley & Sons, 1999.