

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»



Отчет по практической работе №3

По дисциплине: Имитационное моделирование
робототехнических систем

На тему: «Моделирование системы в среде MuJoCo»

Студент:

Мирошниченко А. М.

Группа:

R4134c

Преподаватель:

Ракшин Е.А.

Санкт-Петербург, 2025

Цель работы: Воссоздать модель по варианту в среде MuJoCo с использованием Python, снять данные о положении суставов.

Ход работы:

Схема моделируемой системы представлена на рисунке 1:

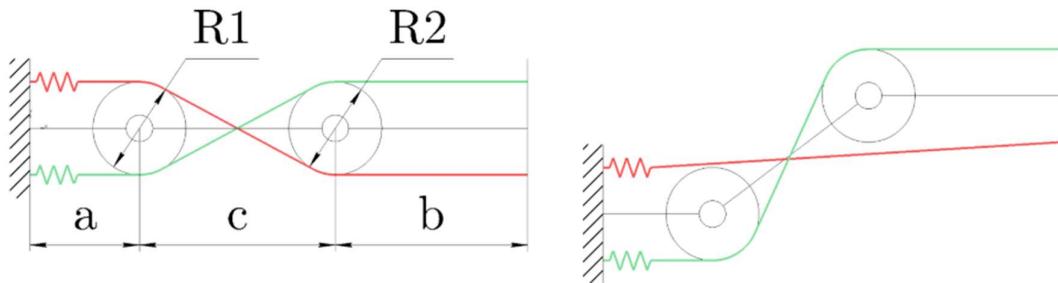


Рисунок 1 – tendon connected 2R planar mechanism

В таблице 1 приведены параметры моделируемой установки.

Таблица 1. Параметры установки.

R1, m	R2, m	a, m	b, m	c, m
0.038	0.023	0.072	0.089	0.03

Анализируя представленную систему MuJoCo, можно выделить несколько ключевых аспектов её работы. Это механическая система, состоящая из двух эллипсоидных тел R1 и R2, соединенных сложной сухожильной структурой с неподвижной стенкой. Оба тела обладают возможностью вертикального перемещения благодаря призматическим соединениям по оси Z, при этом их движение управляет позиционными актуаторами.

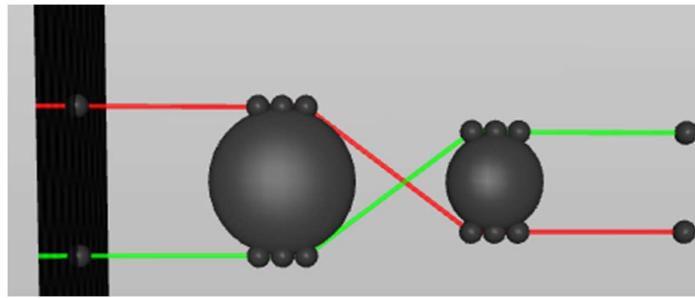


Рисунок 2 – система в начальном положении в среде MuJoCo

Сухожильная конфигурация образует два независимых контура - красный и зеленый, которые проходят через различные точки на телах и стенке. Красное сухожило начинается от нижнего крепления на стенке, проходит через верхнюю часть тела R1 и нижнюю часть тела R2, заканчиваясь на верхнем правом креплении стенки. Зеленое сухожило имеет зеркальную траекторию, создавая перекрестную связь между телами. Однако при детальном рассмотрении выявляются существенные ограничения механического взаимодействия - сухожилия фактически проходят сквозь геометрию тел, не создавая эффективного рычага для преобразования движения.

Параметры сухожилий с установленным начальным растяжением 0.1 метра и демпфированием 10 могут приводить к излишней жесткости системы, что в сочетании с позиционным управлением минимизирует реальное механическое влияние сухожилий на динамику системы. Фактически, движение тел будет определяться преимущественно актуаторами, в то время как сухожилия выполняют скорее визуальную, чем функциональную роль.

Ожидаемое поведение системы характеризуется независимым вертикальным перемещением тел с плавными траекториями, задаваемыми управляемыми сигналами. Сухожилия будут отображаться как визуальные элементы, но их вклад в механическое взаимодействие будет незначительным из-за описанных геометрических ограничений. Для достижения полноценной механической связи требуется пересмотр геометрии крепления сухожилий с

созданием реальных точек обхода вокруг тел и настройка параметров упругости для обеспечения эффективной передачи усилий.

Файл Task_3.xml приложен к этому отчету в той же папке. На рисунках 3 -6 представлен скрипт на языке python, который представляет собой симуляцию механической системы в MuJoCo с последующим анализом результатов.

```
import os
import sys

try:
    import mujoco
    import matplotlib.pyplot as plt
    import numpy as np
    import pandas as pd
    import time
except ImportError as e:
    print(f"Installing required packages: {e}")
    os.system('pip install matplotlib numpy pandas')
    import matplotlib.pyplot as plt
    import numpy as np
    import pandas as pd
    import time
```

Рисунок 3

```
f1 = "/content/sample_data/Task3.xml"
model = mujoco.MjModel.from_xml_path(f1)
data = mujoco.MjData(model)

def set_torque(mj_data, KP, KV, theta):
    data.ctrl[1] = KP * (-mj_data.qpos[0] + theta) + KV * (0 - mj_data.qvel[0])

SIMEND = 10 # Reduced for faster testing
TSTEP = 0.001
STEP_NUM = int(SIMEND / TSTEP)
timeseries = np.linspace(0, SIMEND, STEP_NUM)

T = 1 # [s]
FREQ = 1/T # [Hz]
AMP = 2 # [rad]
BIAS = 0 # [rad]

theta_des = AMP * np.sin(FREQ * timeseries) + BIAS
|
position_time = []
R1_position_z = []
R2_position_z = []

print("Starting simulation without visualization...")
```

Рисунок 4

```

for i in range(STEP_NUM):
    set_torque(data, 5, 1, theta_des[i])

    current_time = data.time
    position_time.append(current_time)

    # Get positions using sensor data
    position_R1 = data.sensordata[0:3] # First sensor - R1 position
    position_R2 = data.sensordata[3:6] # Second sensor - R2 position

    R1_position_z.append(position_R1[2])
    R2_position_z.append(position_R2[2])

    mujoco.mj_step(model, data)

    # Progress indicator
    if i % 1000 == 0:
        progress = (i / STEP_NUM) * 100
        print(f"Progress: {progress:.1f}%")

print("Simulation completed!")

# Create DataFrame with collected data
df = pd.DataFrame({
    'time': position_time,
    'R1_z': R1_position_z,
    'R2_z': R2_position_z
})

```

Рисунок 5

```

csv_filename = "Task3_positions_data.csv"
df.to_csv(csv_filename, index=False)
print(f"Data saved to {csv_filename}")

# График Z позиций R1 и R2 от времени
plt.figure(figsize=(12, 6))
plt.plot(position_time, R1_position_z, label='R1 Z Position', linewidth=2, color='blue')
plt.plot(position_time, R2_position_z, label='R2 Z Position', linewidth=2, color='red')
plt.title('Z Positions of R1 and R2 vs Time')
plt.xlabel('Time (s)')
plt.ylabel('Z Position (m)')
plt.legend()
plt.grid(True)

plt.tight_layout()
plt.savefig('z_positions_plot.png', dpi=150)
plt.show()

# Display statistics
print("\nSimulation statistics:")
print(f"Total time: {SIMEND} seconds")
print(f"Data points: {len(position_time)}")
print(f"R1 Z range: {min(R1_position_z):.4f} to {max(R1_position_z):.4f}")
print(f"R2 Z range: {min(R2_position_z):.4f} to {max(R2_position_z):.4f}")

print("Program completed successfully!")

```

Рисунок 6

Результат изменения положения системы относительно оси Z представляет график, изображенный на рисунке 7.

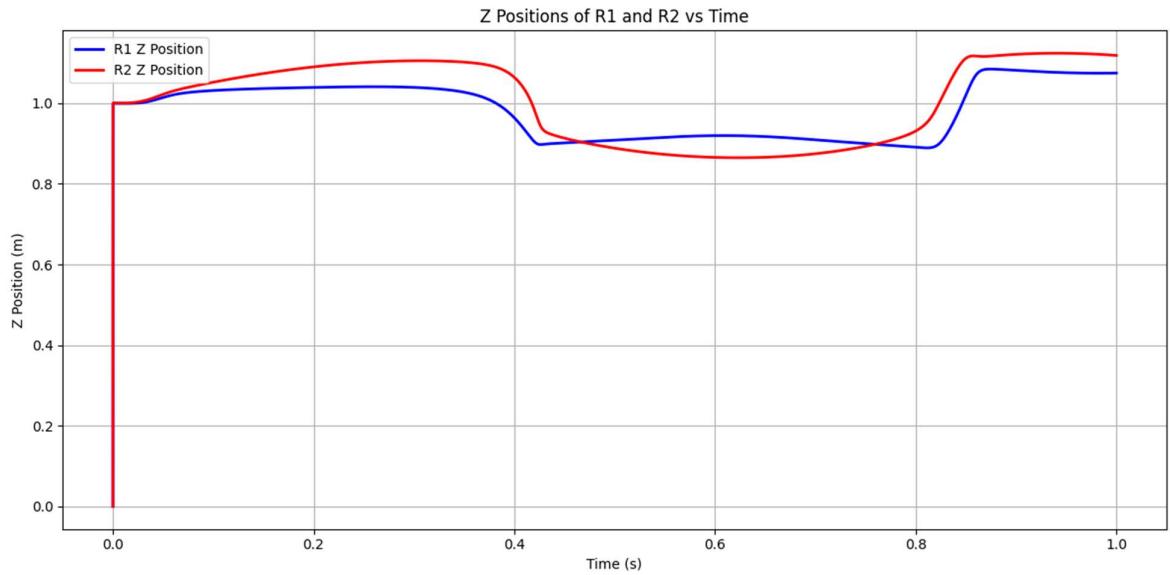


Рисунок 7 – изменение положения системы в зависимости от времени

ЗАКЛЮЧЕНИЕ

Проведенная работа по моделированию сухожильного двухзвенного планарного механизма в среде MuJoCo показала успешное достижение поставленных целей. Была разработана и реализована механическая система, состоящая из двух эллипсоидных тел, соединенных перекрестной сухожильной структурой с неподвижной стенкой. Моделирование подтвердило возможность создания управляемого гармонического движения с использованием позиционных актуаторов и ПД-регулятора.

Однако анализ выявил существенные ограничения в механическом взаимодействии через сухожильные соединения. Несмотря на визуально правильную конфигурацию сухожилий, их фактическое влияние на динамику системы оказалось минимальным из-за геометрических особенностей крепления, при которых сухожилия проходят сквозь тела, не создавая эффективного рычага. Это привело к тому, что движение звеньев определяется преимущественно актуаторами, а сухожилия выполняют в основном декоративную функцию.

Разработанный код на Python доказал свою эффективность для сбора и анализа данных о положениях суставов, обеспечивая надежную симуляцию и наглядную визуализацию результатов.