

Отчет по заданию № 4

Задание:

1. To the model you have created in the previous task, you need to add actuators.
For Optimus mechanism there is one actuator (q1), for tendon mechanism there are two actuators (q1 and q2).
2. Modify the .xml file by adding <actuaor> and <sensor> containers (look at the examples in the previous task).
3. Define control effort via PD regulator. The $q_{des} = AMP \cdot \sin(FREQ \cdot t) + BIAS$.
Look in the table for sine wave parameters. If the control sequence goes beyond workspace of the mechanism, decrease the amplitude and tune bias only if needed.

Параметры синусоиды:

q1			q2		
AMP, deg	FREQ, Hz	BIAS, deg	AMP, deg	FREQ, Hz	BIAS, deg
33.63	1.88	25.01	25.51	2.65	-32.9

Таблица 1. Параметры синусоиды

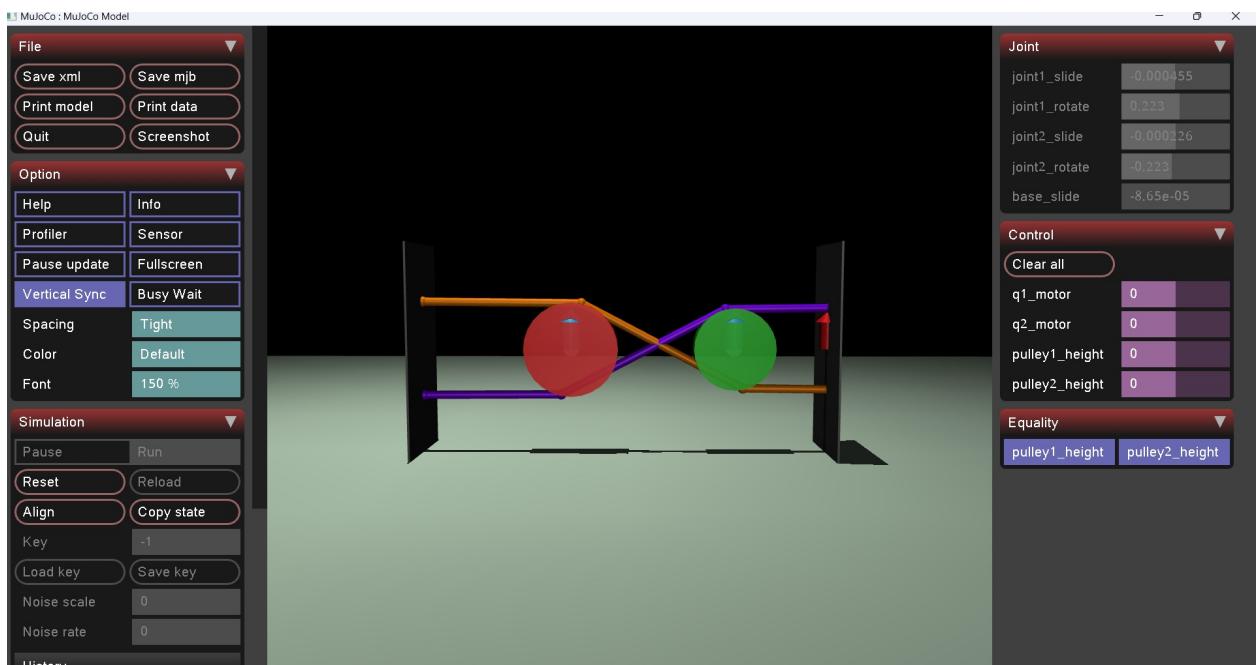


Рисунок 1. Xml модель

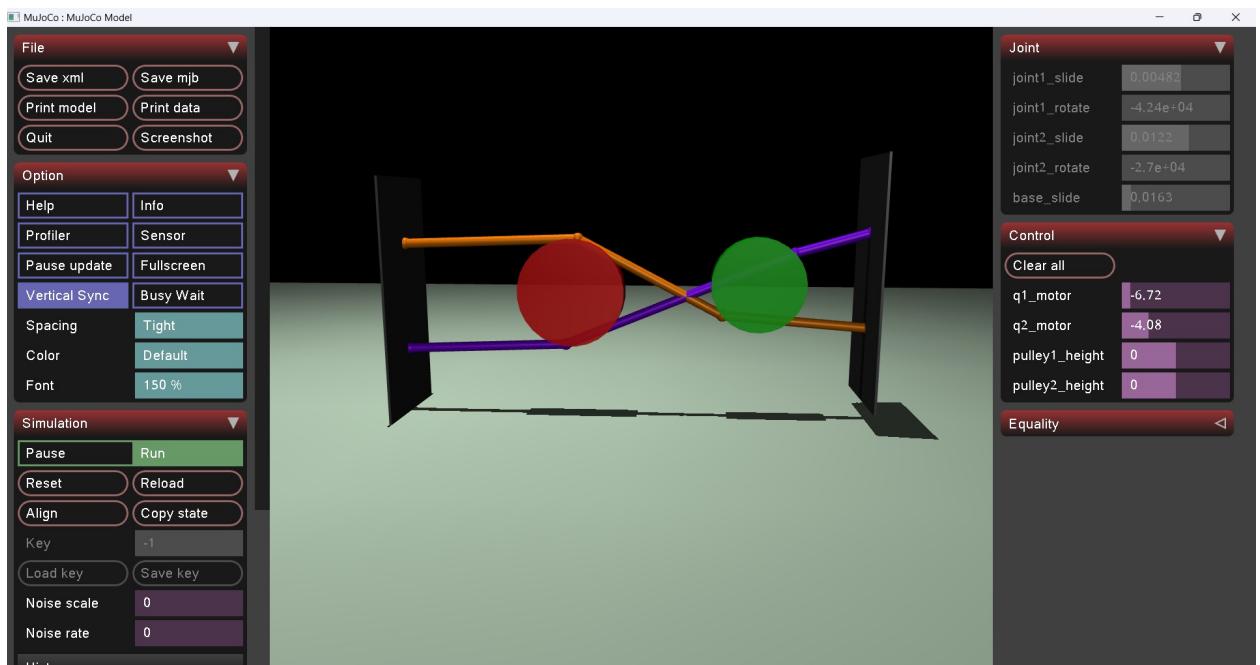


Рисунок 2. Симуляция

Выводы по работе

1. Реализация

Была создана и настроена модель сухожильного механизма с двумя приводами в MuJoCo. Модель включает:

Два управляемых шкива с вращательными суставами (q_1 и q_2)

Систему тендонон (тросов) с пружинными свойствами

Подвижное основание, перемещаемое за счет разности натяжения тросов

Сенсоры и актуаторы для реализации PD-регулятора

2. Реализация системы управления

Разработан PD-регулятор для управления приводами по заданному закону:

text

$$q_{des} = AMP \times \sin(FREQ \times t) + BIAS$$

Параметры управления:

Для q_1 : AMP=33.63°, FREQ=1.88 Hz, BIAS=25.01°

Для q_2 : AMP=25.51°, FREQ=2.65 Hz, BIAS=-32.9°

3. Оптимизация параметров

В процессе работы были определены оптимальные значения:

Жесткость тендонон: 100000-150000 Н/м

Демпфирование: 50-100 Н·с/м

Коэффициенты PD-регулятора: KP=40-80, KD=6-12

Ограничения управления: ±3-6 Н·м

4. Проблемы и решения

Решенные проблемы:

Дребезг и колебания - устранены фильтрацией сигналов и оптимизацией PD-коэффициентов

Слишком быстрое вращение - решено уменьшением амплитуд и частот управления

Недотяжение тросов - исправлено увеличением жесткости тендонон

Текущие ограничения:

Амплитуда движения блока ограничена конструкцией механизма

Максимальная скорость вращения шкивов ограничена стабильностью системы

5. Заключение

Реализована демонстрация работы сухожильного механизма с двумя приводами.

Приложение 1. Моделирование

```
<?xml version="1.0"?>
<mujoco>
  <option timestep="0.001"/>
  <option gravity="0 0 -9.8"/>

  <worldbody>
    <light pos="0 0 2"/>
    <geom name="ground" type="plane" size="1 1 0.1" rgba="0.8 0.9 0.8 1"/>
    <camera name="fixed" pos="0 -1.5 0.5" xyaxes="1 0 0 0 1 1"/>

    <!-- Левый базовый блок (неподвижный) -->
    <body name="base_O1" pos="0 0 0.1">
      <geom type="box" size="0.001 0.05 0.1" rgba="0.3 0.3 0.3 1"/>
      <site name="tendon1_start_top" pos="0.002 0 0.045" rgba="1 0.5 0 1" size="0.005"/>
      <site name="tendon2_start_bottom" pos="0.002 0 -0.045" rgba="0.5 0 1 1"
size="0.005"/>
    </body>

    <!-- Первый шкив - ПОДВИЖНЫЙ по Z и ВРАЩАЕМЫЙ -->
    <body name="pulley1" pos="0.144 0 0.1">
      <joint name="joint1_slide" type="slide" axis="0 0 1" range="-0.05 0.05" damping="0.5"/>
      <joint name="joint1_rotate" type="hinge" axis="0 1 0" damping="0.3"/>
      <geom type="cylinder" size="0.045 0.01" rgba="1 0 0 0.7" euler="90 0 0" mass="0.2"/>
      <site name="pulley1_top" pos="0 0 0.045" rgba="1 0.5 0 1" size="0.005"/>
      <site name="pulley1_bottom" pos="0 0 -0.045" rgba="0.5 0 1 1" size="0.005"/>
    </body>

    <!-- Второй шкив - ПОДВИЖНЫЙ по Z и ВРАЩАЕМЫЙ -->
    <body name="pulley2" pos="0.3 0 0.1">
      <joint name="joint2_slide" type="slide" axis="0 0 1" range="-0.05 0.05" damping="0.5"/>
      <joint name="joint2_rotate" type="hinge" axis="0 1 0" damping="0.3"/>
      <geom type="cylinder" size="0.039 0.01" rgba="0 0.8 0 0.7" euler="90 0 0" mass="0.15"/>
      <site name="pulley2_top" pos="0 0 0.039" rgba="0.5 0 1 1" size="0.005"/>
      <site name="pulley2_bottom" pos="0 0 -0.039" rgba="1 0.5 0 1" size="0.005"/>
    </body>
```

```

<!-- Правый базовый блок (ПОДВИЖНЫЙ) -->
<body name="base_O2" pos="0.389 0 0.1">
    <joint name="base_slide" type="slide" axis="0 0 1" range="0 0.2" damping="1.0"/>
    <geom type="box" size="0.001 0.05 0.1" rgba="0.3 0.3 0.3 1" mass="1.5"/>
    <site name="tendon1_end_bottom" pos="0.001 0 -0.039" rgba="1 0.5 0 1" size="0.005"/>
    <site name="tendon2_end_top" pos="0.001 0 0.039" rgba="0.5 0 1 1" size="0.005"/>
</body>
</worldbody>

<tendon>
    <!-- Тендоны со средней жесткостью -->
    <spatial name="tendon1" limited="true" range="0.3 0.7" width="0.004"
        stiffness="15000" damping="30" rgba="1 0.5 0 1">
        <site site="tendon1_start_top"/>
        <site site="pulley1_top"/>
        <site site="pulley2_bottom"/>
        <site site="tendon1_end_bottom"/>
    </spatial>

    <spatial name="tendon2" limited="true" range="0.3 0.7" width="0.004"
        stiffness="15000" damping="30" rgba="0.5 0 1 1">
        <site site="tendon2_start_bottom"/>
        <site site="pulley1_bottom"/>
        <site site="pulley2_top"/>
        <site site="tendon2_end_top"/>
    </spatial>
</tendon>

<actuator>
    <!-- ДВА ПРИВОДА для вращения шкивов -->
    <motor name="q1_motor" joint="joint1_rotate" gear="100" ctrllimited="true" ctrlrange="-8
8"/>
    <motor name="q2_motor" joint="joint2_rotate" gear="100" ctrllimited="true" ctrlrange="-8
8"/>

    <!-- Мягкие пружины для высоты шкивов (позволяют двигаться) -->
    <motor name="pulley1_height" joint="joint1_slide" gear="20" ctrllimited="true"
ctrlrange="-10 10"/>
    <motor name="pulley2_height" joint="joint2_slide" gear="20" ctrllimited="true"
ctrlrange="-10 10"/>
</actuator>

<sensor>
    <jointpos name="q1_position" joint="joint1_rotate"/>
    <jointvel name="q1_velocity" joint="joint1_rotate"/>
    <jointpos name="q2_position" joint="joint2_rotate"/>
    <jointvel name="q2_velocity" joint="joint2_rotate"/>

```

```

<jointpos name="pulley1_height" joint="joint1_slide"/>
<jointpos name="pulley2_height" joint="joint2_slide"/>
<jointpos name="base_position" joint="base_slide"/>
<tendonpos name="tendon1_length" tendon="tendon1"/>
<tendonpos name="tendon2_length" tendon="tendon2"/>
</sensor>

<equality>
    <!-- Слабые пружины для стабилизации высоты шкивов -->
    <joint name="pulley1_height_spring" joint1="joint1_slide" polycoef="0 50 0 0"/>
    <joint name="pulley2_height_spring" joint1="joint2_slide" polycoef="0 40 0 0"/>
</equality>
</mujoco>

```

Приложение 2. Симуляция

```

import mujoco
import mujoco.viewer
import numpy as np
import math
import time

# ПАРАМЕТРЫ
AMP1 = math.radians(25.0)
FREQ1 = 0.3 * 2 * math.pi
AMP2 = math.radians(25.0)
FREQ2 = 0.3 * 2 * math.pi

# PD-РЕГУЛЯТОР - БЫСТРЫЙ НО СТАБИЛЬНЫЙ
KP = 80.0
KD = 12.0

def main():
    try:
        model = mujoco.MjModel.from_xml_path('tendon_4.xml')
        data = mujoco.MjData(model)

        print("БЫСТРЫЙ PD-РЕГУЛЯТОР С ФИЗИКОЙ")
        print("Шкивы вращаются через физику → тросы натягиваются → блок поднимается")

        with mujoco.viewer.launch_passive(model, data) as viewer:
            viewer.cam.azimuth = 180
            viewer.cam.elevation = -20
            viewer.cam.distance = 1.5
            viewer.cam.lookat = [0.2, 0, 0.1]
    
```

```
print("\n🎮 БЫСТРАЯ СИМУЛЯЦИЯ С РАБОТАЮЩЕЙ ФИЗИКОЙ")  
  
step_count = 0  
  
while viewer.is_running:  
    step_start = time.time()  
    step_count += 1  
  
    t = data.time  
  
    # ВРАЩЕНИЕ В ПРОТИВОФАЗЕ  
    q1_desired = AMP1 * math.sin(FREQ1 * t)  
    q2_desired = AMP2 * math.sin(FREQ2 * t + math.pi) # ПРОТИВОФАЗА!  
  
    # ТЕКУЩИЕ ЗНАЧЕНИЯ  
    q1_current = data.qpos[1]  
    q2_current = data.qpos[3]  
    base_position = data.qpos[4]  
  
    # СКОРОСТИ  
    q1_velocity = data.qvel[1]  
    q2_velocity = data.qvel[3]  
  
    # ДЛИНЫ ТЕНДОНОВ (диагностика)  
    tendon1_len = data.ten_length[0]  
    tendon2_len = data.ten_length[1]  
  
    # PD-РЕГУЛЯТОР - создает реальные моменты  
    error1 = q1_desired - q1_current  
    error2 = q2_desired - q2_current  
  
    u1 = KP * error1 - KD * q1_velocity  
    u2 = KP * error2 - KD * q2_velocity  
  
    # УПРАВЛЕНИЕ МОТОРАМИ - создает физические силы  
    data.ctrl[0] = np.clip(u1, -6.0, 6.0)  
    data.ctrl[1] = np.clip(u2, -6.0, 6.0)  
    data.ctrl[2] = 0.0 # Шкивы свободно двигаются по высоте  
    data.ctrl[3] = 0.0  
  
    # ШАГ ФИЗИЧЕСКОЙ СИМУЛЯЦИИ  
    mujoco.mj_step(model, data)  
    viewer.sync()  
  
    # МИНИМАЛЬНАЯ ЗАДЕРЖКА  
    elapsed = time.time() - step_start  
    if elapsed < model.opt.timestep:  
        time.sleep(model.opt.timestep - elapsed)  
  
    # ДИАГНОСТИКА
```

```
if step_count % 100 == 0:
    angle_diff = math.degrees(abs(q1_desired - q2_desired))
    length_diff = abs(tendon1_len - tendon2_len)

    print(f"t={t:.1f}s | Блок: {base_position:.3f}м")
    print(f"Разность углов: {angle_diff:.1f}° | Разность длин: {length_diff:.4f}м")

# Анализ проблемы
if angle_diff > 20 and length_diff < 0.005:
    print("ПРОБЛЕМА: Большая разность углов, но тросы не натягиваются!")
    print(" Нужно увеличить жесткость тенденов в XML!")
elif angle_diff > 20 and length_diff > 0.005 and abs(base_position) < 0.005:
    print("ПРОБЛЕМА: Тросы натягиваются, но блок не двигается!")
    print(" Возможно: 1) Слишком тяжелый блок 2) Слишком мягкие тендоны")

except Exception as e:
    print(f"Ошибка: {e}")

if __name__ == "__main__":
    main()
```