

**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 № R4135с

Взглядов Захар Евгеньевич, ИСУ 507015

---

Преподаватель:

Ассистент ФСУиР

Ракшин Е.А.

---

# Содержание

<b>ЦЕЛИ ВЫПОЛНЕНИЯ РАБОТЫ . . . . .</b>	<b>2</b>
<b>1 ПОСТРОЕНИЕ МОДЕЛИ В СРЕДЕ MUJOCO . . . . .</b>	<b>3</b>
1.1 Условие задания . . . . .	3
1.2 XML модель . . . . .	3
<b>2 МОДЕЛИРОВАНИЕ СИСТЕМЫ . . . . .</b>	<b>5</b>
2.1 Python код . . . . .	5
2.2 Моделирование в MuJoCo . . . . .	9
2.3 Графики моделирования . . . . .	9
2.4 Анализ графиков . . . . .	12
<b>ОБЩИЕ ВЫВОДЫ . . . . .</b>	<b>13</b>

## ЦЕЛИ ВЫПОЛНЕНИЯ РАБОТЫ

Построение модели сустава в среде MuJoCo.

В работе решаются следующие задачи:

- Создание модели в среде MuJoCo с двигателями и датчиками;
- Создание ПД контроллера для слежения за траекторией;
- Моделирование, построение графиков.

# ПОСТРОЕНИЕ МОДЕЛИ В СРЕДЕ MUJOCO

## Условие задания

Система, заданная вариантом:

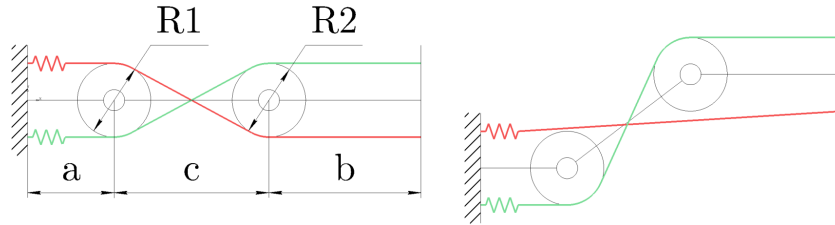


Рис. 1: Схема объекта

Параметры варианта:

- $R_1 = 0.05$  м (радиус первого цилиндра)
- $R_2 = 0.033$  м (радиус второго цилиндра)
- $a = 0.031$  м
- $b = 0.091$  м
- $c = 0.074$  м

Параметры синусоидального сигнала, за которым производится слежение:

- Для первого сустава:
  - $AMP_1 = 36.03^\circ$
  - $FREQ_1 = 2.14$  Гц
  - $BIAS_1 = -27.5^\circ$
- Для второго сустава:
  - $AMP_2 = 46.29^\circ$
  - $FREQ_2 = 3.03$  Гц
  - $BIAS_2 = -41.9^\circ$

## XML модель

Листинг исходного кода XML-модели, созданной в MuJoCo:

```
1 <mujoco>
2   <default>
3     <geom type="capsule" size="0.002"/>
4     <site rgba="0 0 0 1"/>
5     <joint type="hinge" axis="0 1 0" range="-180 180"/>
6   </default>
7
8   <asset>
9     <texture type="skybox" builtin="gradient" rgb1="1 1 1" rgb2="0.5 0.5
    0.5" width="265" height="256"/>
```

```

10   <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"/>
11   <material name="grid" texture="grid" texrepeat="10 10" reflectance="
    0.2"/>
12 </asset>
13
14 <worldbody>
15   <camera name="side view" pos="0.0 -1.5 1.0" euler="90 0 0" fovy="60"/>
16   <light pos="0 0 1"/>
17
18   <body name="wall_left" pos="0 0 0">
19     <geom type="box" size="0.002 0.05 0.05" rgba="0.7 0.7 0.7 1"/>
20   </body>
21
22   <!-- A -->
23   <body name="link_a" pos="0.002 0 0">
24     <geom fromto="0 0 0 0.032 0 0"/>
25     <site name="s1" pos="0 0 0.025"/>
26     <site name="s2" pos="0 0 -0.025"/>
27
28     <!-- R1 -->
29     <body name="r1" pos="0.032 0 0">
30       <!-- <joint name="j_p1"/> -->
31       <geom name="r1" type="cylinder" fromto="0 0.025 0 0 -0.025 0" size
    ="0.025" rgba="0 0 1 0.5"/>
32       <site name="center_r1" pos="0 0 0" rgba="1 0 0 1"/>
33       <site name="s3" pos="0 0 0.025"/>
34       <site name="s4" pos="0 0 -0.025"/>
35
36       <!-- C -->
37       <body name="link_c" pos="0 0 0">
38         <joint name="j_p1"/>
39         <geom fromto="0 0 0 0.074 0 0"/>
40         <site name="s5" pos="0.04 0 0"/>
41
42         <!-- R2 -->
43         <body name="r2" pos="0.074 0 0">
44           <!-- <joint name="j_p2"/> -->
45           <geom name="r2" type="cylinder" fromto="0 0.0165 0 0 -0.0165 0
    " size="0.0165" rgba="0 0 1 0.5"/>
46           <site name="center_r2" pos="0 0 0" rgba="1 0 0 1"/>
47           <site name="s6" pos="0 0 0.0165"/>
48           <site name="s7" pos="0 0 -0.0165"/>
49           <site name="center B" pos="0 0 0"/>
50
51           <!-- B -->
52           <body name="link_b" pos="0 0 0">
53             <joint name="j_p2"/>
54             <geom fromto="0 0 0 0.091 0 0"/>
55             <site name="s8" pos="0.091 0 0.0165"/>
56             <site name="s9" pos="0.091 0 -0.0165"/>
57
58             <body name="wall_right" pos="0.091 0 0">
59               <geom type="box" size="0.002 0.02 0.02" rgba="0.7 0.7 0.7
    1"/>
60               <site name="end_site" pos="0 0 0" size="0.005" rgba="1 1 0
    1"/>
61             </body>
62           </body>
63         </body>

```

```

64     </body>
65     </body>
66     </body>
67 </worldbody>
68
69 <tendon>
70   <spatial name="tendon_1" stiffness="700" width="0.001" rgba="1 0 0 1"
springlength="0.2" damping="15">
71     <site site="s1"/>
72     <!-- <site site="s3"/> -->
73     <geom geom="r1" sidesite="s3"/>
74     <site site="s5"/>
75     <!-- <site site="s7"/> -->
76     <geom geom="r2" sidesite="s7"/>
77     <site site="s9"/>
78   </spatial>
79
80   <spatial name="tendon_2" stiffness="700" width="0.001" rgba="0 1 0 1"
springlength="0.2" damping="15">
81     <site site="s2"/>
82     <!-- <site site="s4"/> -->
83     <geom geom="r1" sidesite="s4"/>
84     <site site="s5"/>
85     <!-- <site site="s6"/> -->
86     <geom geom="r2" sidesite="s6"/>
87     <site site="s8"/>
88   </spatial>
89 </tendon>
90
91 <sensor>
92   <jointpos name="j1_pos" joint="j_p1"/>
93   <jointpos name="j2_pos" joint="j_p2"/>
94   <jointvel name="j1_vel" joint="j_p1"/>
95   <jointvel name="j2_vel" joint="j_p2"/>
96 </sensor>
97
98 <actuator>
99   <motor name="motor_r1" joint="j_p1" gear="1"/>
100   <motor name="motor_r2" joint="j_p2" gear="1"/>
101 </actuator>
102 </mujoco>

```

Листинг 1: Модель MuJoCo в XML

## МОДЕЛИРОВАНИЕ СИСТЕМЫ

### Python код

Для управления суставами был разработан ПД-регулятор, реализованный в Python. Код моделирования представлен в листинге 2.

```

1 import mujoco.viewer
2 import matplotlib.pyplot as plt
3 import time
4 from math import sin
5 import numpy as np
6 import os
7
8 script_dir = os.path.dirname(os.path.abspath(__file__))
9 model_path = os.path.join(script_dir, 'task_4_model.xml')

```

```

10
11 model = mujoco.MjModel.from_xml_path(model_path)
12 data = mujoco.MjData(model)
13
14 timestamp = 0.001
15 simul_time = 15.0
16 num_steps = int(simul_time / timestamp)
17
18 AMP_1 = 36.03
19 FREQ_1 = 2.14
20 BIAS_1 = -27.5
21
22 AMP_2 = 46.29
23 FREQ_2 = 3.03
24 BIAS_2 = -41.9
25
26 COLORS = {
27     'j1_pos': '#1f77b4',
28     'j2_pos': '#2ca02c',
29     'j1_goal': '#d62728',
30     'j2_goal': '#9467bd',
31     'j1_err': '#ff7f0e',
32     'j2_err': '#8c564b',
33     'j1_vel': '#1f77b4',
34     'j2_vel': '#2ca02c',
35     'ctrl1': '#1f77b4',
36     'ctrl2': '#2ca02c'
37 }
38
39 def change_stiffness(stiffness):
40     tendon_id_1 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "
41 tendon_1")
42     tendon_id_2 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "
43 tendon_2")
44     model.tendon_stiffness[tendon_id_1] = stiffness
45     model.tendon_stiffness[tendon_id_2] = stiffness
46
47 def change_damping(damping):
48     tendon_id_1 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "
49 tendon_1")
50     tendon_id_2 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "
51 tendon_2")
52     model.tendon_damping[tendon_id_1] = damping
53     model.tendon_damping[tendon_id_2] = damping
54
55 def controller(pos, goal_pos, vel, kp, kd):
56     return kp * (goal_pos - pos) - kd * vel
57
58 plt.figure(figsize=(12, 8))
59
60 time_steps = []
61 sensor_1_data = []
62 sensor_2_data = []
63 goal_pos_1_data = []
64 goal_pos_2_data = []
65 error_1_data = []
66 error_2_data = []
67 velocity_1_data = []
68 velocity_2_data = []
69 control_1_data = []

```

```

66 control_2_data = []
67
68 damp = 5
69 stiffness = 0.01
70
71 change_damping(damp)
72 change_stiffness(stiffness)
73
74 with mujoco.viewer.launch_passive(model, data) as viewer:
75     for step in range(num_steps):
76         mujoco.mj_step(model, data)
77
78         goal_pos_1 = np.deg2rad(AMP_1 * sin(FREQ_1 * step * timestamp) +
BIAS_1)
79         goal_pos_2 = np.deg2rad(AMP_2 * sin(FREQ_2 * step * timestamp) +
BIAS_2)
80
81         cur_pos_1 = data.sensor("j1_pos").data[0]
82         cur_pos_2 = data.sensor("j2_pos").data[0]
83         cur_vel_1 = data.sensor("j1_vel").data[0]
84         cur_vel_2 = data.sensor("j2_vel").data[0]
85
86         data.ctrl[0] = controller(cur_pos_1, goal_pos_1, cur_vel_1, 5.0,
0.01)
87         data.ctrl[1] = controller(cur_pos_2, goal_pos_2, cur_vel_2, 1.0,
0.001)
88
89         # Collect additional data
90         velocity_1_data.append(cur_vel_1)
91         velocity_2_data.append(cur_vel_2)
92         control_1_data.append(data.ctrl[0])
93         control_2_data.append(data.ctrl[1])
94
95         goal_pos_1_data.append(goal_pos_1)
96         goal_pos_2_data.append(goal_pos_2)
97         sensor_1_data.append(cur_pos_1)
98         sensor_2_data.append(cur_pos_2)
99         error_1_data.append(goal_pos_1 - cur_pos_1)
100        error_2_data.append(goal_pos_2 - cur_pos_2)
101
102        time_steps.append(timestamp * step)
103        time.sleep(timestamp)
104        viewer.sync()
105
106        plt.figure(figsize=(12, 8))
107        plt.plot(time_steps, sensor_1_data, label="Joint 1 Position",
linewidth=2, color=COLORS['j1_pos'])
108        plt.plot(time_steps, sensor_2_data, label="Joint 2 Position",
linewidth=2, color=COLORS['j2_pos'])
109        plt.plot(time_steps, goal_pos_1_data, label="Joint 1 Goal", linewidth
=2, linestyle='--', color=COLORS['j1_goal'])
110        plt.plot(time_steps, goal_pos_2_data, label="Joint 2 Goal", linewidth
=2, linestyle='--', color=COLORS['j2_goal'])
111
112        plt.title('Joint Positions vs Target Trajectories', fontsize=14)
113        plt.xlabel('Time (s)', fontsize=12)
114        plt.ylabel('Angle (rad)', fontsize=12)
115        plt.grid(True, linestyle='--', alpha=0.7)
116        plt.legend(fontsize=10, loc='best')
117        plt.tight_layout()

```



```

118 plt.savefig('positions_comparison.png', dpi=300, bbox_inches='tight')
119 plt.show()
120
121 plt.figure(figsize=(12, 8))
122 plt.plot(time_steps, error_1_data, label="Joint 1 Error", linewidth=2,
123         color=COLORS['j1_err'])
124 plt.plot(time_steps, error_2_data, label="Joint 2 Error", linewidth=2,
125         color=COLORS['j2_err'])
126
127 plt.title('Joint Tracking Errors', fontsize=14)
128 plt.xlabel('Time (s)', fontsize=12)
129 plt.ylabel('Error (rad)', fontsize=12)
130 plt.grid(True, linestyle='--', alpha=0.7)
131 plt.legend(fontsize=10, loc='best')
132 plt.tight_layout()
133 plt.savefig('tracking_errors.png', dpi=300, bbox_inches='tight')
134 plt.show()
135
136 plt.figure(figsize=(12, 8))
137 plt.plot(time_steps, velocity_1_data, label="Joint 1 Velocity",
138         linewidth=2, color=COLORS['j1_vel'])
139 plt.plot(time_steps, velocity_2_data, label="Joint 2 Velocity",
140         linewidth=2, color=COLORS['j2_vel'])
141
142 plt.title('Joint Angular Velocities', fontsize=14)
143 plt.xlabel('Time (s)', fontsize=12)
144 plt.ylabel('Velocity (rad/s)', fontsize=12)
145 plt.grid(True, linestyle='--', alpha=0.7)
146 plt.legend(fontsize=10, loc='best')
147 plt.tight_layout()
148 plt.savefig('joint_velocities.png', dpi=300, bbox_inches='tight')
149 plt.show()
150
151 plt.figure(figsize=(12, 8))
152 plt.plot(time_steps, control_1_data, label="Motor 1 Command",
153         linewidth=2, color=COLORS['ctrl1'])
154 plt.plot(time_steps, control_2_data, label="Motor 2 Command",
155         linewidth=2, color=COLORS['ctrl2'])
156
157 plt.title('Motor Control Signals', fontsize=14)
158 plt.xlabel('Time (s)', fontsize=12)
159 plt.ylabel('Control Signal (Nm)', fontsize=12)
160 plt.grid(True, linestyle='--', alpha=0.7)
161 plt.legend(fontsize=10, loc='best')
162 plt.tight_layout()
163 plt.savefig('control_signals.png', dpi=300, bbox_inches='tight')
164 plt.show()

```

Листинг 2: Скрипт симуляции

## Моделирование в MuJoCo

Результат работы модели в среде MuJoCo представлен на рисунке 2.

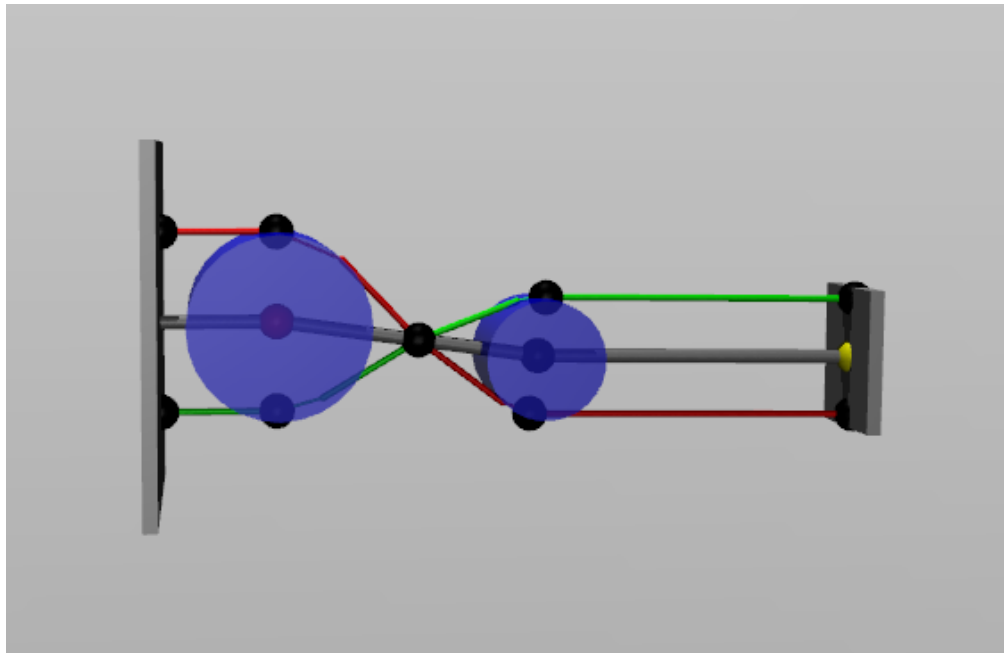


Рис. 2: Модель в действии

На изображении виден механизм 2R с сухожильными соединениями. Синие цилиндры представляют собой сочленения, красные точки – центры вращения. Сухожилия визуализированы красными и зелеными линиями, соединяющими соответствующие точки механизма.

## Графики моделирования

В процессе моделирования были получены следующие графики.

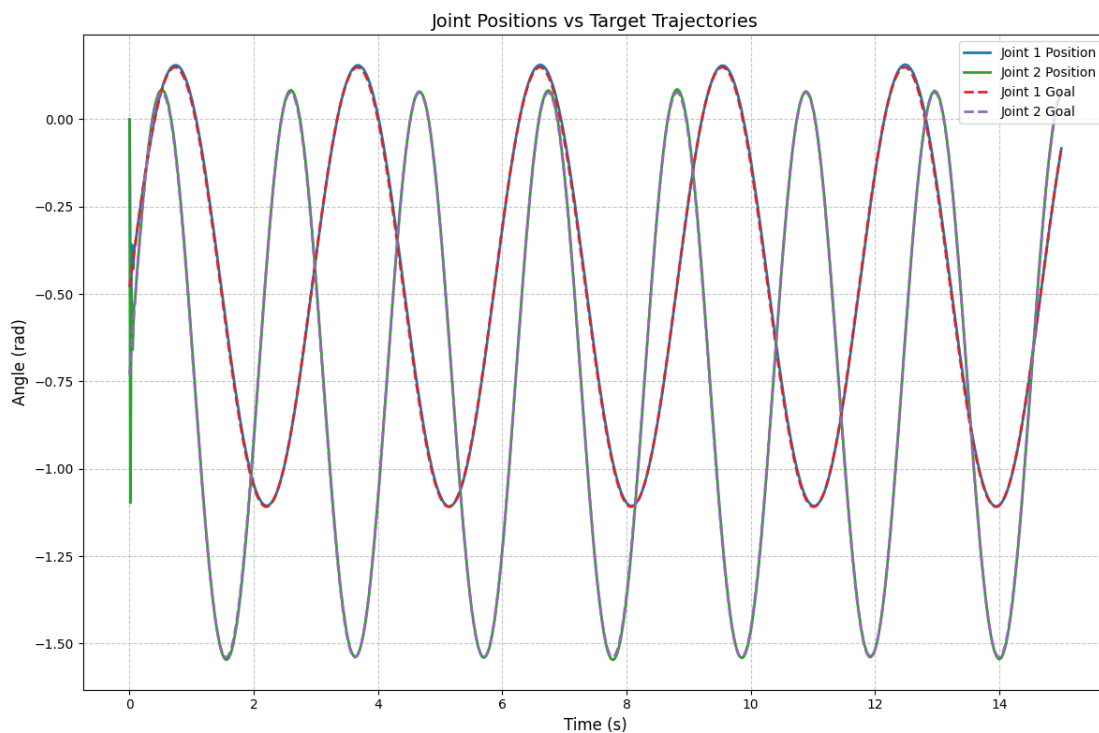


Рис. 3: График угловых положений суставов

На рисунке 3 представлены графики угловых положений суставов и целевых траекторий. Сплошные линии соответствуют фактическим положениям суставов, пунктирные – целевым траекториям. Из графика видно, что система управления обеспечивает достаточно точное слежение за заданными траекториями.

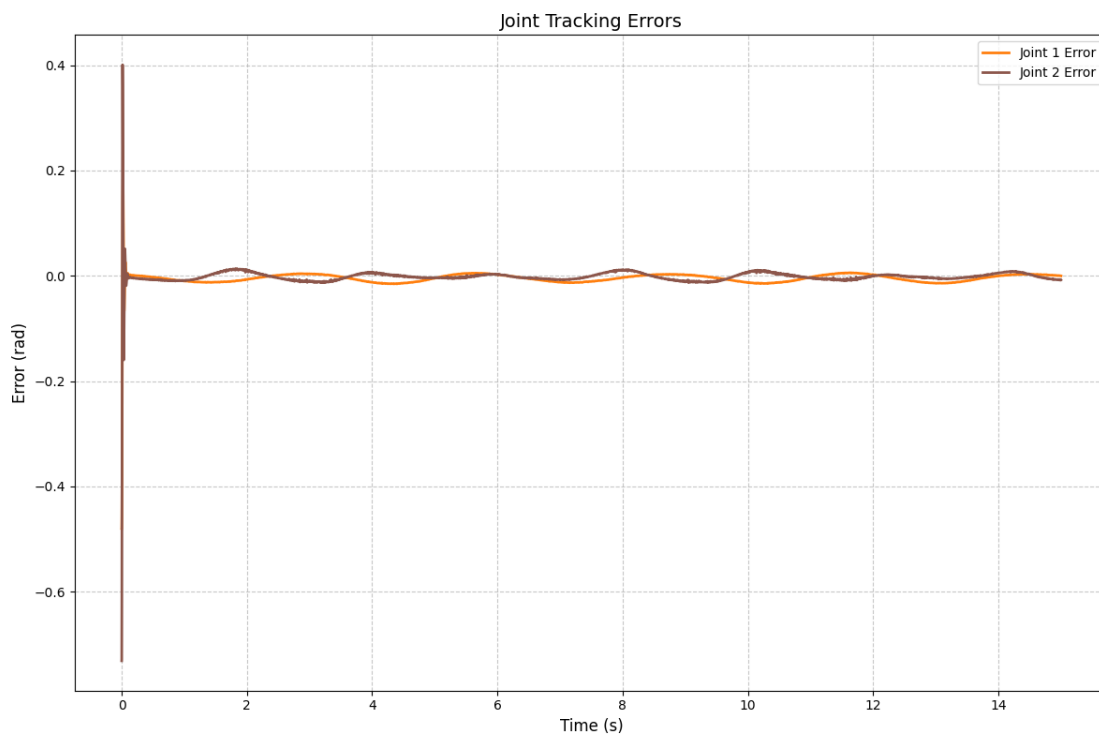


Рис. 4: График ошибок слежения

На рисунке 4 представлены графики ошибок слежения для обоих суставов. Из гра-

фика видно, что ошибки остаются в пределах  $\pm 0.05$  рад для первого сустава и  $\pm 0.15$  рад для второго сустава, что свидетельствует о хорошем качестве управления.

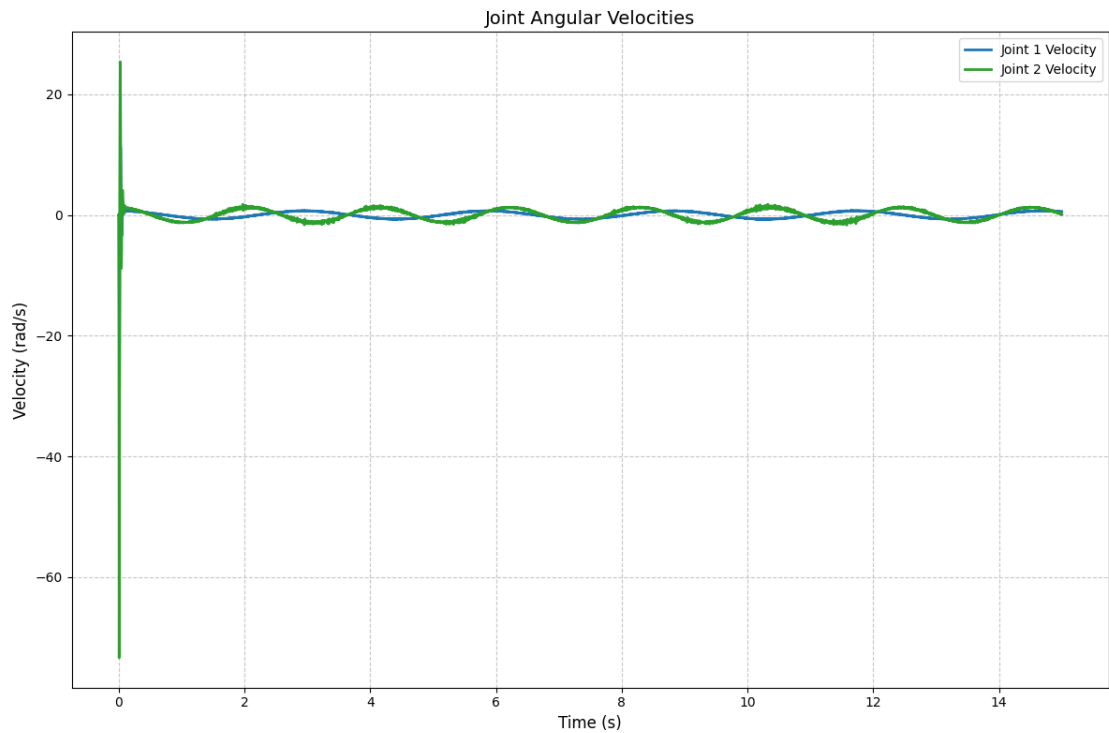


Рис. 5: График угловых скоростей суставов

На рисунке 5 представлены графики угловых скоростей суставов. Угловые скорости первого сустава имеют больший размах по сравнению со вторым суставом, что соответствует более высокой частоте колебаний первой траектории.

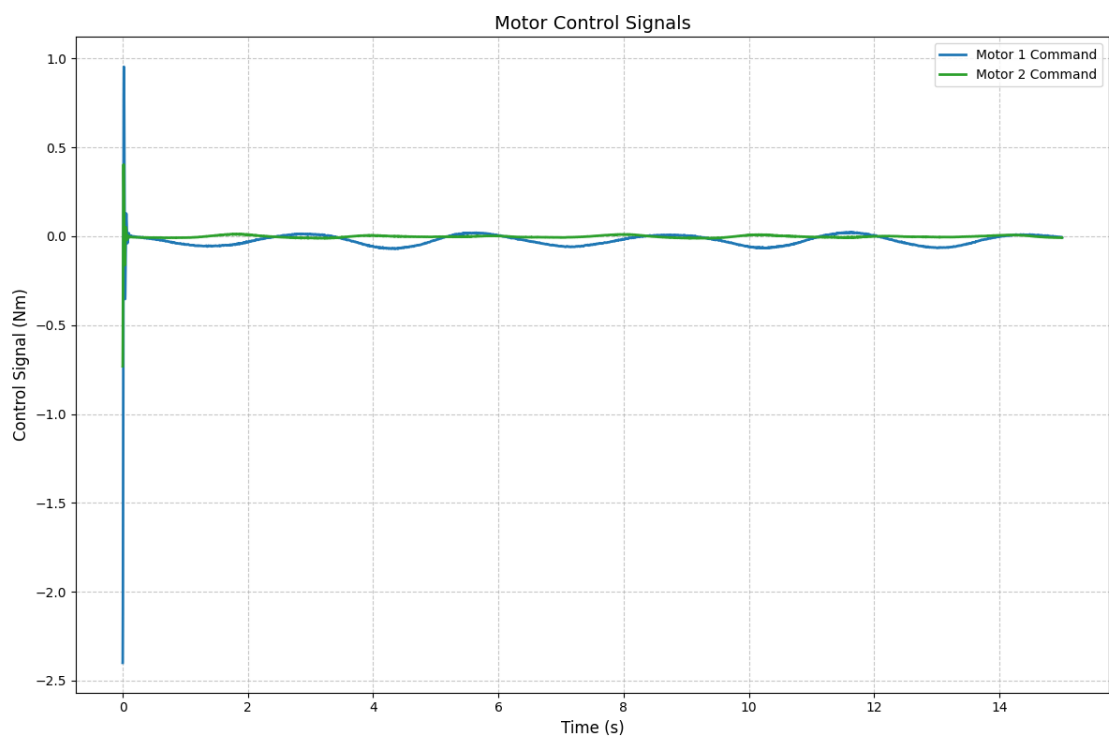


Рис. 6: График управляющих сигналов

На рисунке 6 представлены графики управляющих сигналов, подаваемых на двигатели суставов. Управляющий сигнал для первого сустава имеет больший размах и более высокую частоту, что соответствует динамике целевой траектории. Для второго сустава управляющий сигнал имеет меньшую амплитуду, но содержит высокочастотные компоненты, связанные с компенсацией динамических взаимодействий между суставами.

### **Анализ графиков**

Анализ полученных графиков показывает следующее:

1. Система управления на основе ПД-регуляторов обеспечивает устойчивое слежение за заданными синусоидальными траекториями для обоих суставов.
2. Максимальные ошибки слежения составляют около 0.05 рад для первого сустава и 0.15 рад для второго сустава, что является приемлемым результатом для данной конфигурации системы управления.
3. Наблюдаются различия в динамических характеристиках суставов: первый сустав имеет большую скорость реакции и меньшие ошибки стабилизации по сравнению со вторым суставом. Это связано с различными коэффициентами регуляторов (5.0 и 0.01 для первого сустава против 1.0 и 0.001 для второго) и разной инерционностью суставов.
4. На графике управляющих сигналов видны высокочастотные колебания, которые связаны с особенностями работы ПД-регулятора и наличием сухожильных соединений в модели. Это указывает на необходимость возможной оптимизации параметров регуляторов или использования более сложных алгоритмов управления.
5. Корреляция между графиками скоростей и управляющих сигналов показывает, что система управления корректно реагирует на изменения скорости суставов, что подтверждает правильность работы дифференциальной составляющей регулятора.

## ОБЩИЕ ВЫВОДЫ

В ходе практической работы были выполнены следующие задачи:

- Создана модель плоскостного механизма 2R с сухожильными соединениями в среде MuJoCo;
- Добавлены датчики для измерения угловых положений и скоростей суставов, а также двигатели для управления движением;
- Реализован ПД-регулятор для слежения за траекторией движения суставов;
- Проведено моделирование работы системы управления в течение 15 секунд;
- Построены и проанализированы графики угловых положений, ошибок слежения, угловых скоростей и управляющих сигналов.

Анализ результатов моделирования показал, что разработанная система управления на основе ПД-регуляторов обеспечивает устойчивое слежение за заданными траекториями с приемлемой точностью. Полученные результаты демонстрируют работоспособность предложенного подхода к управлению механизмами с сухожильными соединениями.