

**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 № R4135c
Взглядов Захар Евгеньевич, ИСУ 507015

Преподаватель:
Ассистент ФСУиР
Ракшин Е.А.

Содержание

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

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:

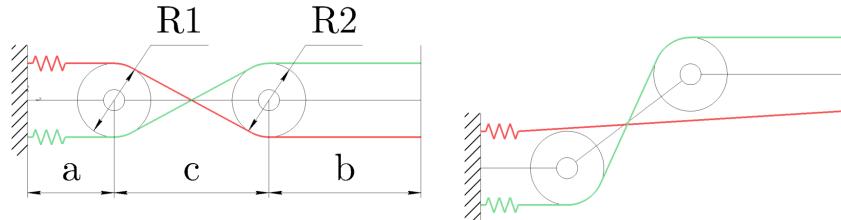


Рис. 1: Схема объекта

Параметры варианта:

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

Параметры синусоидального сигнала, за которым производится сложение:

- Для первого сустава:

- $\text{AMP}_1 = 36.03^\circ$
- $\text{FREQ}_1 = 2.14 \text{ Гц}$
- $\text{BIAS}_1 = -27.5^\circ$

- Для второго сустава:

- $\text{AMP}_2 = 46.29^\circ$
- $\text{FREQ}_2 = 3.03 \text{ Гц}$
- $\text{BIAS}_2 = -41.9^\circ$

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"
11      rgb2="0.6 0.6 0.6" width="300" height="300"/>
12    <material name="grid" texture="grid" texrepeat="10 10" reflectance="
13      0.2"/>
14  </asset>
15
16  <worldbody>
17    <camera name="side view" pos="0.0 -1.5 1.0" euler="90 0 0" fovy="60"/>
18    <light pos="0 0 1"/>
19
20    <body name="wall_left" pos="0 0 0">
21      <geom type="box" size="0.002 0.05 0.05" rgba="0.7 0.7 0.7 1"/>
22    </body>
23
24    <!-- A -->
25    <body name="link_a" pos="0.002 0 0">
26      <geom fromto="0 0 0 0.032 0 0"/>
27      <site name="s1" pos="0 0 0.025"/>
28      <site name="s2" pos="0 0 -0.025"/>
29
30    <!-- R1 -->
31    <body name="r1" pos="0.032 0 0">
32      <!-- <joint name="j_p1"/> -->
33      <geom name="r1" type="cylinder" fromto="0 0.025 0 0 -0.025 0" size
34      ="0.025" rgba="0 0 1 0.5"/>
35      <site name="center_r1" pos="0 0 0" rgba="1 0 0 1"/>
36      <site name="s3" pos="0 0 0.025"/>
37      <site name="s4" pos="0 0 -0.025"/>
38
39    <!-- C -->
40    <body name="link_c" pos="0 0 0">
41      <joint name="j_p1"/>
42      <geom fromto="0 0 0 0.074 0 0"/>
43      <site name="s5" pos="0.04 0 0"/>
44
45    <!-- R2 -->
46    <body name="r2" pos="0.074 0 0">
47      <!-- <joint name="j_p2"/> -->
48      <geom name="r2" type="cylinder" fromto="0 0.0165 0 0 -0.0165 0
49      " size="0.0165" rgba="0 0 1 0.5"/>
50      <site name="center_r2" pos="0 0 0" rgba="1 0 0 1"/>
51      <site name="s6" pos="0 0 0.0165"/>
52      <site name="s7" pos="0 0 -0.0165"/>
53      <site name="center B" pos="0 0 0"/>
54
55    <!-- B -->
56    <body name="link_b" pos="0 0 0">
57      <joint name="j_p2"/>
58      <geom fromto="0 0 0 0.091 0 0"/>
59      <site name="s8" pos="0.091 0 0.0165"/>
60      <site name="s9" pos="0.091 0 -0.0165"/>
61
62      <body name="wall_right" pos="0.091 0 0">
63        <geom type="box" size="0.002 0.02 0.02" rgba="0.7 0.7 0.7
1"/>
64        <site name="end_site" pos="0 0 0" size="0.005" rgba="1 1 0
1"/>
65      </body>
66    </body>
67  </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"
71     springlength="0.2" damping="15">
72     <site site="s1"/>
73     <! --> <site site="s3"/> -->
74     <geom geom="r1" sidesite="s3"/>
75     <site site="s5"/>
76     <! --> <site site="s7"/> -->
77     <geom geom="r2" sidesite="s7"/>
78     <site site="s9"/>
79   </spatial>
80
80   <spatial name="tendon_2" stiffness="700" width="0.001" rgba="0 1 0 1"
81     springlength="0.2" damping="15">
82     <site site="s2"/>
83     <! --> <site site="s4"/> -->
84     <geom geom="r1" sidesite="s4"/>
85     <site site="s5"/>
86     <! --> <site site="s6"/> -->
87     <geom geom="r2" sidesite="s6"/>
88     <site site="s8"/>
89   </spatial>
90</tendon>
91
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 model = mujoco.MjModel.from_xml_path(model_path)
11 data = mujoco.MjData(model)
12
13 timestamp = 0.001
14 simul_time = 15.0
15 num_steps = int(simul_time / timestamp)
16
17 AMP_1 = 36.03
18 FREQ_1 = 2.14
19 BIAS_1 = -27.5
20
21 AMP_2 = 46.29
22 FREQ_2 = 3.03
23 BIAS_2 = -41.9
24
25 COLORS = {
26     'j1_pos': '#1f77b4',
27     'j2_pos': '#2ca02c',
28     'j1_goal': '#d62728',
29     'j2_goal': '#9467bd',
30     'j1_err': '#ff7f0e',
31     'j2_err': '#8c564b',
32     'j1_vel': '#1f77b4',
33     'j2_vel': '#2ca02c',
34     'ctrl1': '#1f77b4',
35     'ctrl2': '#2ca02c'
36 }
37
38
39 def change_stiffness(stiffness):
40     tendon_id_1 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "tendon_1")
41     tendon_id_2 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "tendon_2")
42     model.tendon_stiffness[tendon_id_1] = stiffness
43     model.tendon_stiffness[tendon_id_2] = stiffness
44
45 def change_damping(damping):
46     tendon_id_1 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "tendon_1")
47     tendon_id_2 = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_TENDON, "tendon_2")
48     model.tendon_damping[tendon_id_1] = damping
49     model.tendon_damping[tendon_id_2] = damping
50
51 def controller(pos, goal_pos, vel, kp, kd):
52     return kp * (goal_pos - pos) - kd * vel
53
54 plt.figure(figsize=(12, 8))
55
56 time_steps = []
57 sensor_1_data = []
58 sensor_2_data = []
59 goal_pos_1_data = []
60 goal_pos_2_data = []
61 error_1_data = []
62 error_2_data = []
63 velocity_1_data = []
64 velocity_2_data = []
65 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) +
79                                BIAS_1)
80         goal_pos_2 = np.deg2rad(AMP_2 * sin(FREQ_2 * step * timestamp) +
81                                BIAS_2)
82
83         cur_pos_1 = data.sensor("j1_pos").data[0]
84         cur_pos_2 = data.sensor("j2_pos").data[0]
85         cur_vel_1 = data.sensor("j1_vel").data[0]
86         cur_vel_2 = data.sensor("j2_vel").data[0]
87
88         data.ctrl[0] = controller(cur_pos_1, goal_pos_1, cur_vel_1, 5.0,
89                                   0.01)
90         data.ctrl[1] = controller(cur_pos_2, goal_pos_2, cur_vel_2, 1.0,
91                                   0.001)
92
93         # Collect additional data
94         velocity_1_data.append(cur_vel_1)
95         velocity_2_data.append(cur_vel_2)
96         control_1_data.append(data.ctrl[0])
97         control_2_data.append(data.ctrl[1])
98
99         goal_pos_1_data.append(goal_pos_1)
100        goal_pos_2_data.append(goal_pos_2)
101        sensor_1_data.append(cur_pos_1)
102        sensor_2_data.append(cur_pos_2)
103        error_1_data.append(goal_pos_1 - cur_pos_1)
104        error_2_data.append(goal_pos_2 - cur_pos_2)
105
106        time_steps.append(timestamp * step)
107        time.sleep(timestamp)
108        viewer.sync()
109
110        plt.figure(figsize=(12, 8))
111        plt.plot(time_steps, sensor_1_data, label="Joint 1 Position",
112                  linewidth=2, color=COLORS['j1_pos'])
113        plt.plot(time_steps, sensor_2_data, label="Joint 2 Position",
114                  linewidth=2, color=COLORS['j2_pos'])
115        plt.plot(time_steps, goal_pos_1_data, label="Joint 1 Goal",
116                  linewidth=2, linestyle='--', color=COLORS['j1_goal'])
117        plt.plot(time_steps, goal_pos_2_data, label="Joint 2 Goal",
118                  linewidth=2, linestyle='--', color=COLORS['j2_goal'])
119
120        plt.title('Joint Positions vs Target Trajectories', fontsize=14)
121        plt.xlabel('Time (s)', fontsize=12)
122        plt.ylabel('Angle (rad)', fontsize=12)
123        plt.grid(True, linestyle='--', alpha=0.7)
124        plt.legend(fontsize=10, loc='best')
125        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.

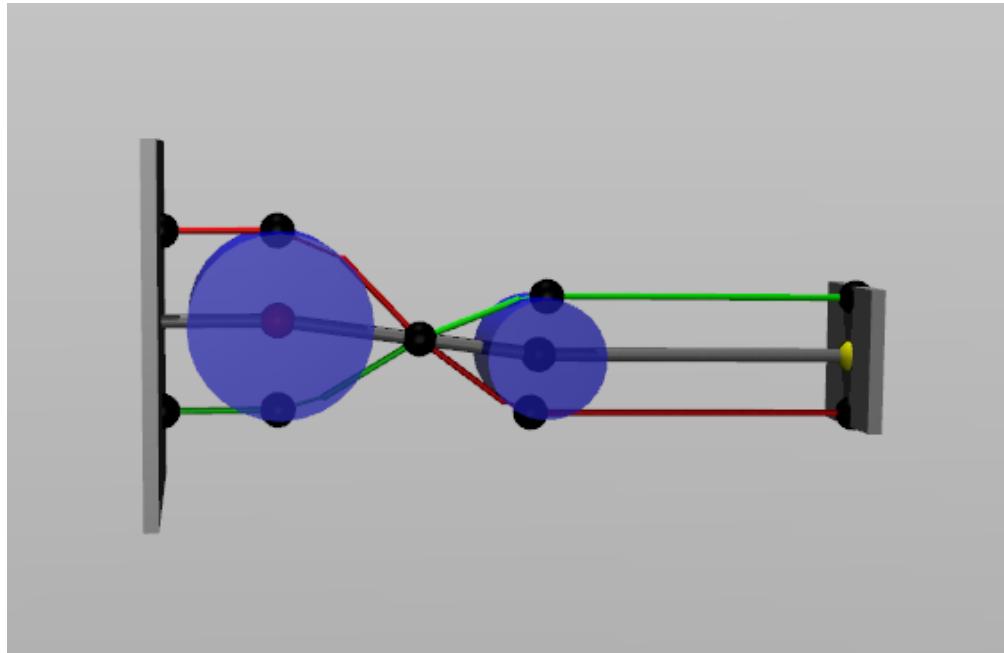


Рис. 2: Модель в действии

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.

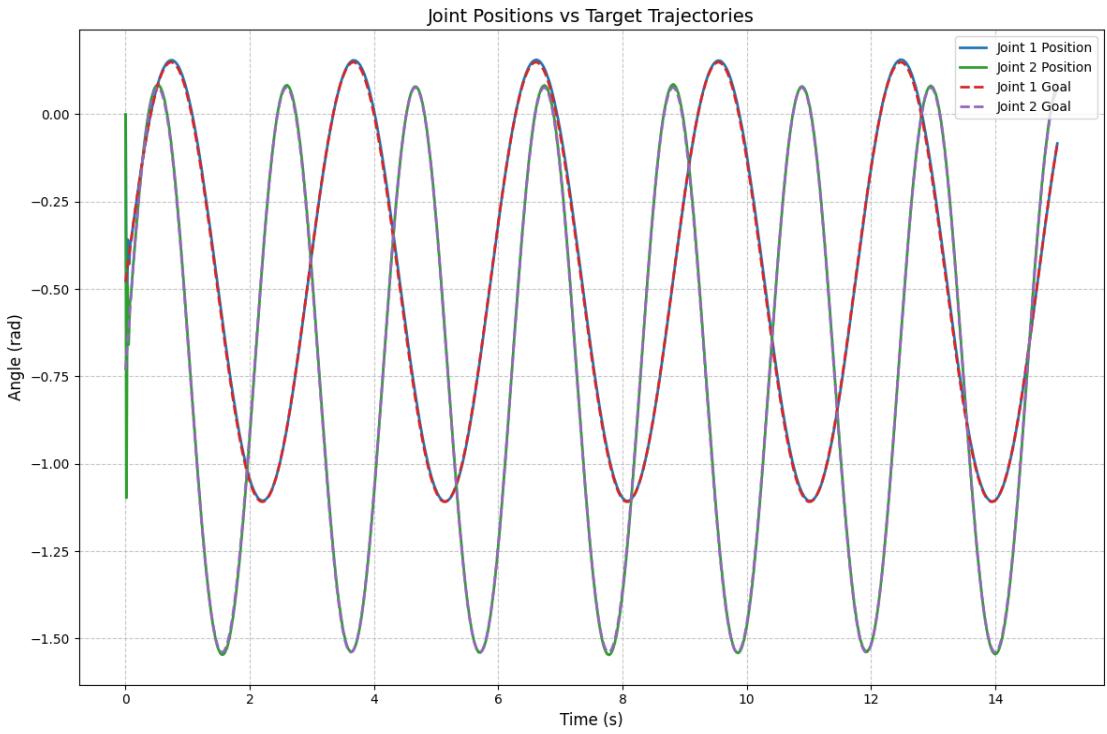


Рис. 3: График угловых положений суставов

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.

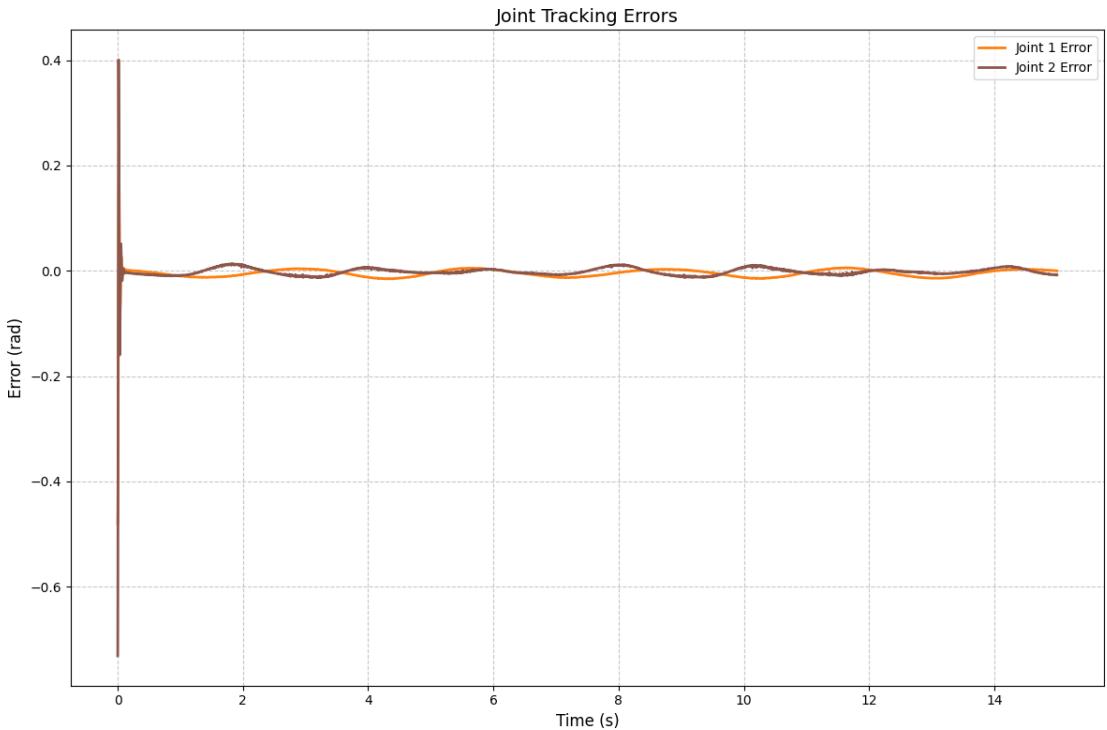


Рис. 4: График ошибок слежения

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

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

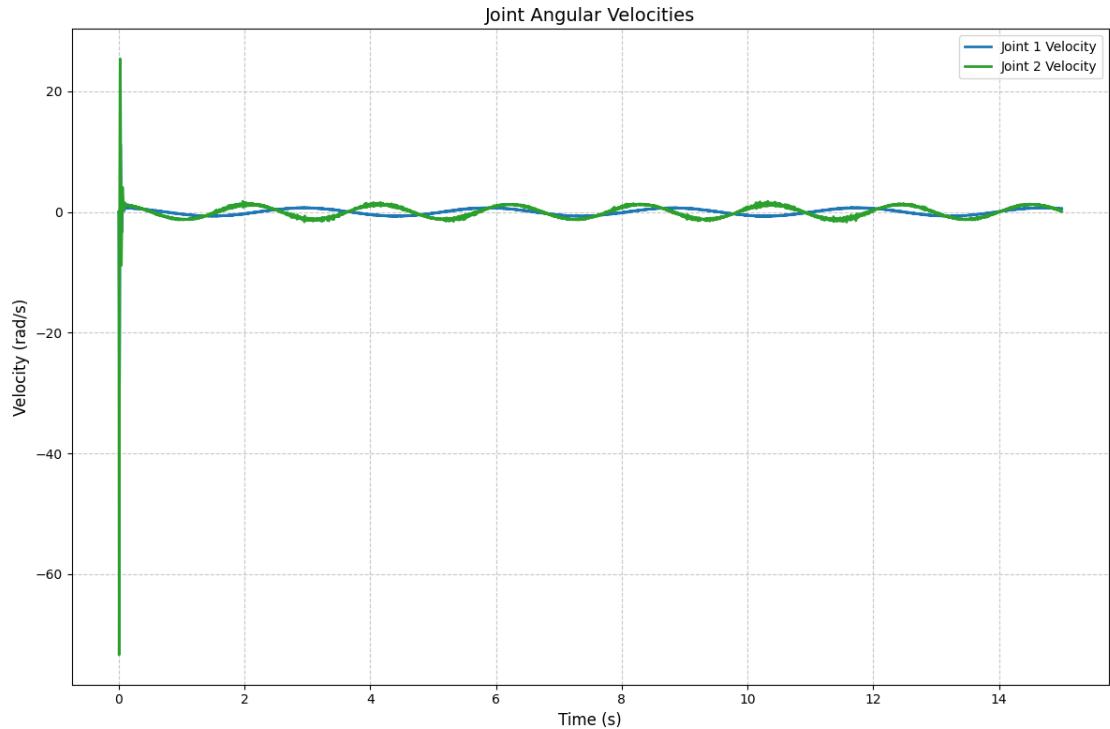


Рис. 5: График угловых скоростей суставов

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.

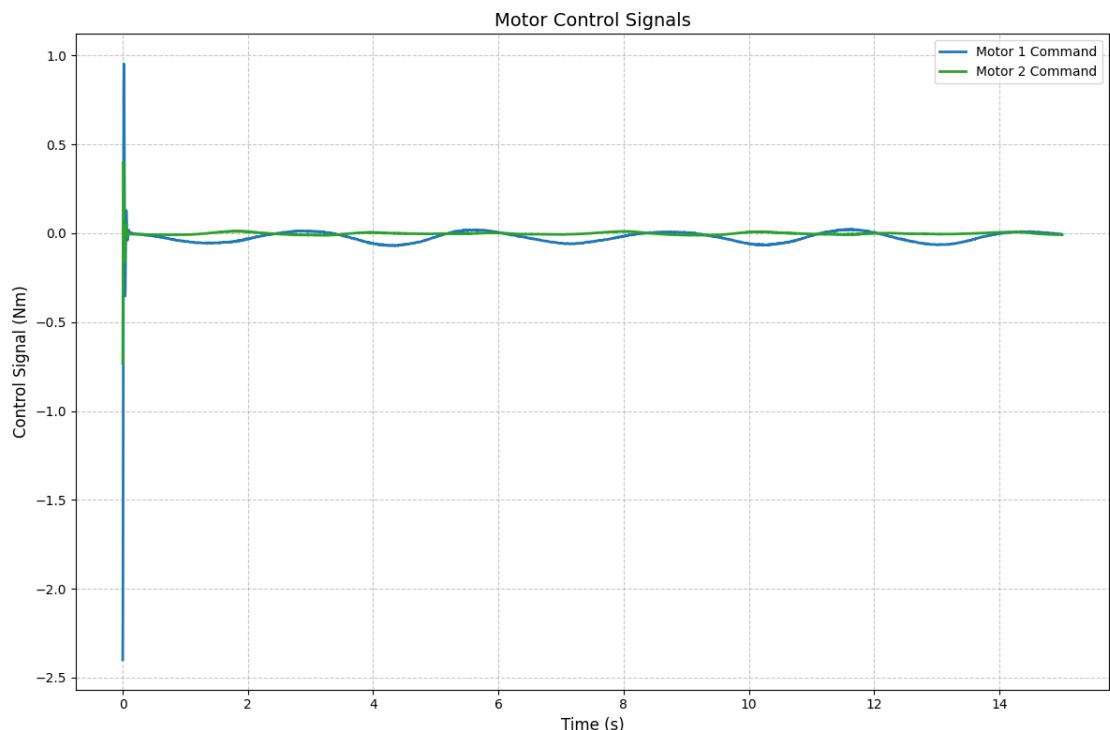


Рис. 6: График управляемых сигналов

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.