



**Санкт-Петербургский национальный исследовательский
университет информационных технологий, механики и оптики**

Факультет систем управления и робототехники

Лабораторная работа №4

Выполнила студентка группы R4150:

Сафонова А. С.

Преподаватель:

Ракшин Егор Александрович

Санкт-Петербург
2025

Исходные данные

Таблица 1 – Исходные данные

AMP, deg	FREQ, Hz	BIAS, deg
13,16	2,86	17,5

Ход работы

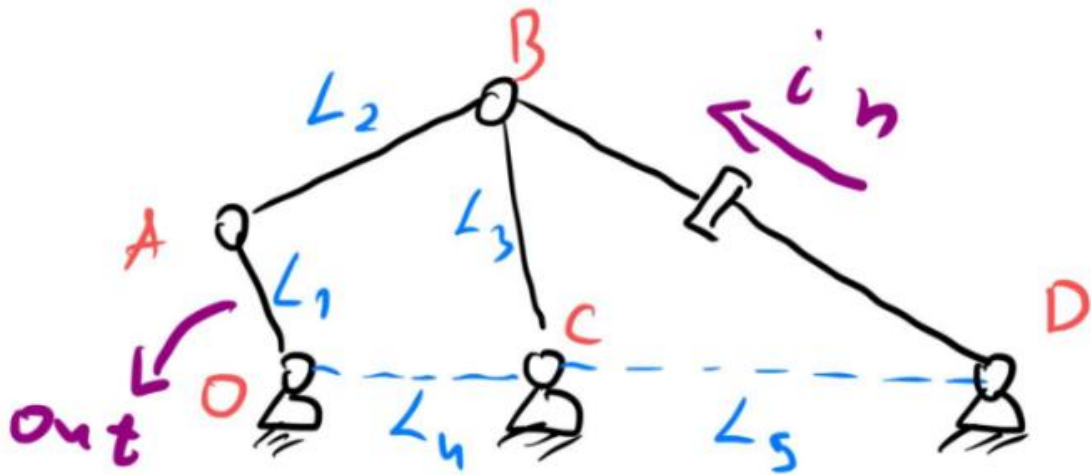


Рисунок 1 – Рассматриваемая система

К описанной модели на рисунке 1 был добавлен привод. Также был реализован ПД-регулятор, который осуществляет необходимое управление.

Файл hml:

```
1 <?xml version='1.0' encoding='UTF-8'><?xml>
2 <mujoco>
3   <option timestep="1e-3"/>
4   <option gravity="0 0 -9.8"/>
5   <asset>
6     <texture type="skybox" builtin="gradient" rgb1="0.08 0.08 0.08" rgb2="0.15 0.15 0.15" width="256" height="256"/>
7     <texture name="grid" type="2d" builtin="checker" rgb1="0.88 0.88 0.88" rgb2="0.72 0.65 0.55" width="300" height="300"/>
8     <material name="grid" texture="grid" texrepeat="10 10" reflectance="0.2"/>
9     <material name="link_mat" reflectance="0.3"/>
10  </asset>
11  <worldbody>
12
13    <light pos="0 0 10"/>
14    <geom type="plane" size="0.5 0.5 0.1" material="grid"/>
15
16    <body name="OAB1" pos="0 0 0" euler="0 0 0">
17
18      <joint name="O" type="hinge" axis="0 -1 0" stiffness="0" springref="0" damping="0"/>
19      <geom name="point O" type="cylinder" pos="0 0 0" size="0.005 0.005" rgba="1 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
20      <geom name="link OA" type="cylinder" pos="0 0 0.039" size="0.005 0.039" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
21
22      <body name="AB1" pos="0 0 0.078" euler="0 0 0">
23        <joint name="A" type="hinge" axis="0 -1 0" stiffness="0" springref="0" damping="0.1"/>
24        <geom name="point B" type="cylinder" pos="0 0 0" size="0.005 0.005" rgba="1 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
25        <geom name="link AB1" type="cylinder" pos="0 0 0.0507"
26          size="0.005 0.0507" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
27        <site name="sc1" size="0.005" pos="0 0 0.1014"/>
28      </body>
29    </body>
30  </worldbody>
31 </mujoco>
```

```

30 <body name="CB2" pos="0.078 0 0" euler="0 0 0">
31   <joint name="c" type="hinge" axis="0 -1 0" stiffness="0" springref="0" damping="0.1"/>
32   <geom name="point C" type="cylinder" pos="0 0 0" size="0.005 0.005" rgba="1 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
33   <geom name="link CB2" type="cylinder" pos="0 0 0.0585" size="0.005 0.0585" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
34   <site name="sC2" size="0.005" pos="0 0 0.117"/>
35 </body>
36 <body name="DFB3" pos="0.39 0 0" euler="0 0 0">
37   <joint name="D" type="hinge" axis="0 -1 0" stiffness="0" springref="0" damping="0"/>
38   <geom name="point B" type="cylinder" pos="0 0 0" size="0.005 0.005" rgba="1 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
39   <geom name="link DB3" type="cylinder" pos="0 0 0.1" size="0.005 0.1" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
40   <body name="FB3" pos="0 0 0.075" euler="0 0 0">
41     <joint name="slider" type="slide" axis="0 0 1" limited="true" range="-0.2 0.2" stiffness="0" springref="0" damping="0"/>
42     <geom name="point B3" type="cylinder" pos="0 0 0" size="0.005 0.005" rgba="1 0.14 0.16 0.5" euler="0 0 0" contype="0"/>
43     <geom name="link FB3" type="cylinder" pos="0 0 0.075" size="0.005 0.15" rgba="0.21 0.32 0.82 0.5" euler="0 0 0" contype="0"/>
44     <site name="sC3" size="0.005" pos="0 0 0.225"/>
45   </body>
46 </body>
47 </worldbody>
48 <actuator>
49   <motor name="motor_0" joint="0" ctrlrange="-10 10" gear="1"/>
50 </actuator>
51 <sensor>
52   <jointpos name="sensor" joint="0"/>
53 </sensor>
54 <equality>
55   <connect name="eq_C1_C2" site1="sC1" site2="sC2" solimp="0.995 0.99 0.001" solref="0.01 1"/>
56   <connect name="eq_C1_C3" site1="sC1" site2="sC3" solimp="0.995 0.99 0.001" solref="0.01 1"/>
57 </equality>
58 </mujoco>

```

Код на Python:

```

1  import time
2  import mujoco
3  import mujoco.viewer
4  from colorama import Fore, Style
5  import pyfiglet
6  import numpy as np
7  import matplotlib.pyplot as plt
8
9  KP = 0.6
10 KD = 0.01
11
12 def main():
13     model = mujoco.MjModel.from_xml_path("lab.xml")
14     data = mujoco.MjData(model)
15     sensor_name = "sensor"
16     data.qpos[:] = 0.0
17     data.qvel[:] = 0.0
18     log_time = []
19     log_m = []
20     log_q = []
21     log_ctrl = []
22
23     with mujoco.viewer.launch_passive(model, data) as viewer:
24         viewer.cam.lookat[:] = np.array([0.09, 0.0, 0.0])
25         viewer.cam.distance = 0.5
26         viewer.cam.elevation = -30
27         t0 = time.time()
28         step = 0
29         prev_sensed_pos = 0.0
30
31         while viewer.is_running():
32             t = time.time() - t0
33             if t >= 30.0:
34                 break
35             q_des = np.deg2rad(13.16) * np.sin(2.86 * t) + np.deg2rad(17.5)
36             sid = mujoco.mj_name2id(model, mujoco.mjtObj.mjOBJ_SENSOR, sensor_name)
37             sensed_pos = data.sensordata[model.sensor_adr[sid]]
38             if step == 0:
39                 sensed_vel = 0.0
40             else:
41                 dt = model.opt.timestep
42                 sensed_vel = (sensed_pos - prev_sensed_pos) / dt
43             prev_sensed_pos = sensed_pos
44             u = KP * (q_des - sensed_pos) - KD * sensed_vel
45             data.ctrl[0] = u
46             mujoco.mj_step(model, data)
47             viewer.sync()
48             if step % 10 == 0:
49                 log_time.append(t)
50                 log_m.append(np.rad2deg(q_des))
51                 log_q.append(np.rad2deg(sensed_pos))
52                 log_ctrl.append(u)
53             step += 1
54
55     log_time = np.array(log_time)
56     log_m = np.array(log_m)
57     log_q = np.array(log_q)
58     log_error = log_m - log_q

```

```

60 plt.figure()
61 plt.plot(log_time, log_m, 'g--', label='Желаемое значение')
62 plt.plot(log_time, log_q, 'b', label='Фактическое значение')
63 plt.xlabel('Время')
64 plt.ylabel('Угол')
65 plt.title('График сравнения')
66 plt.legend()
67 plt.grid(True)
68 plt.tight_layout()
69 plt.show()
70
71 plt.figure()
72 plt.plot(log_time, log_error, label='Ошибка')
73 plt.xlabel('Время')
74 plt.ylabel('Ошибка')
75 plt.title('Ошибка управления')
76 plt.legend()
77 plt.grid(True)
78 plt.tight_layout()
79 plt.show()
80
81 if __name__ == "__main__":
82     main()
83

```

На рисунке 2 можно увидеть визуализацию модели в MuJoCo.

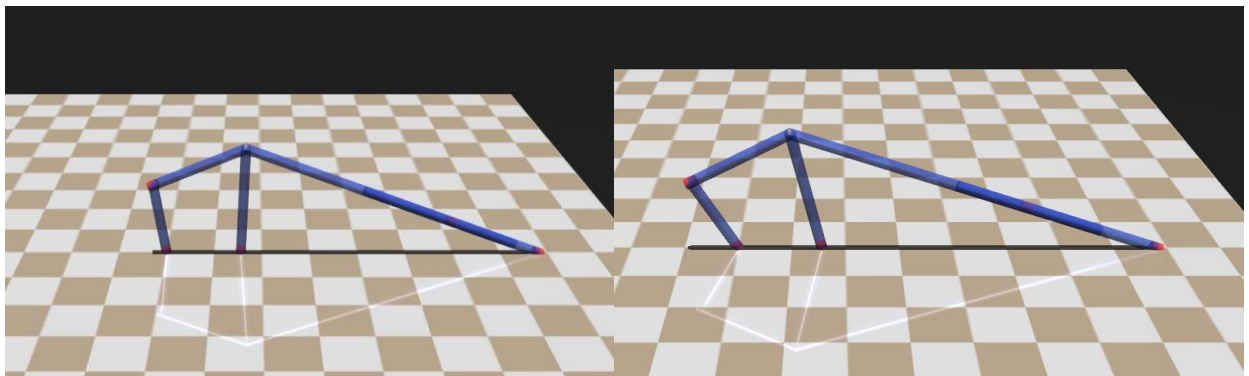


Рисунок 2 – Визуализация модели

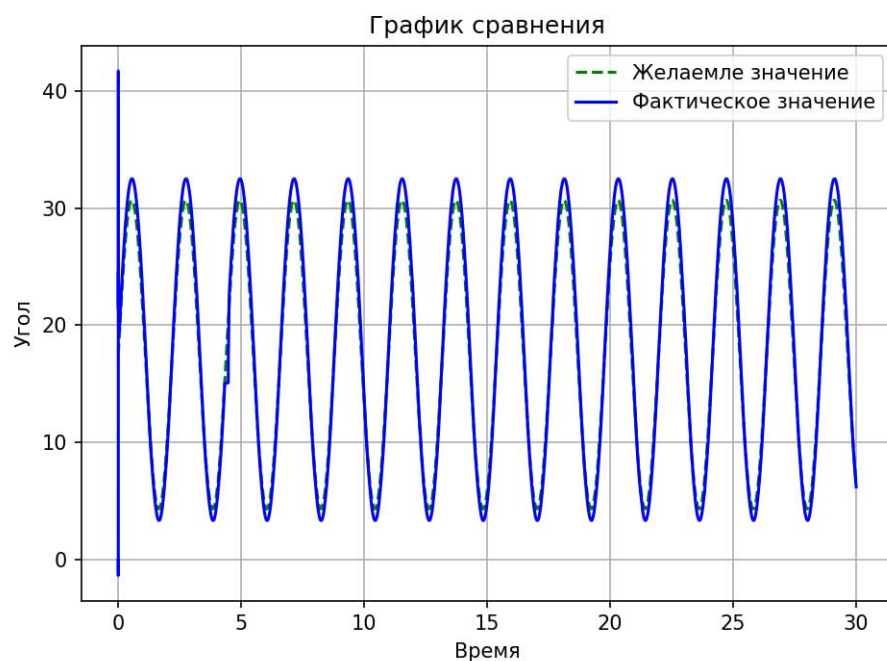


Рисунок 3 – График сравнения

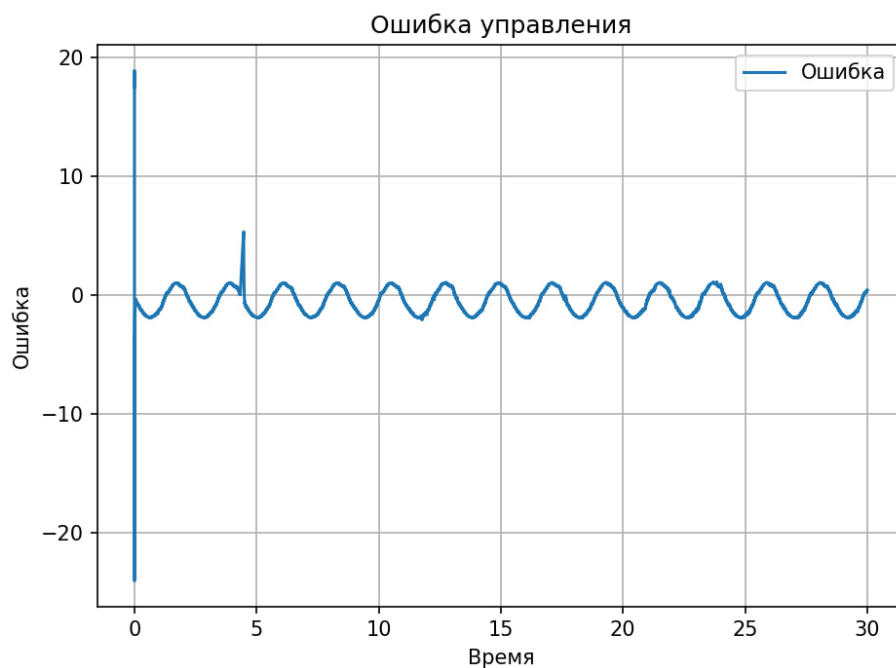


Рисунок 4 – График ошибки управления

Выводы

В лабораторной работе было проведено моделирование коленчатого механизма Optimus в MuJoCo, был реализован ПД регулятор и подобраны оптимальные коэффициенты $K_P = 0,6$ и $K_D = 0,01$. На рисунках 3 и 4 можно увидеть работу механизма.