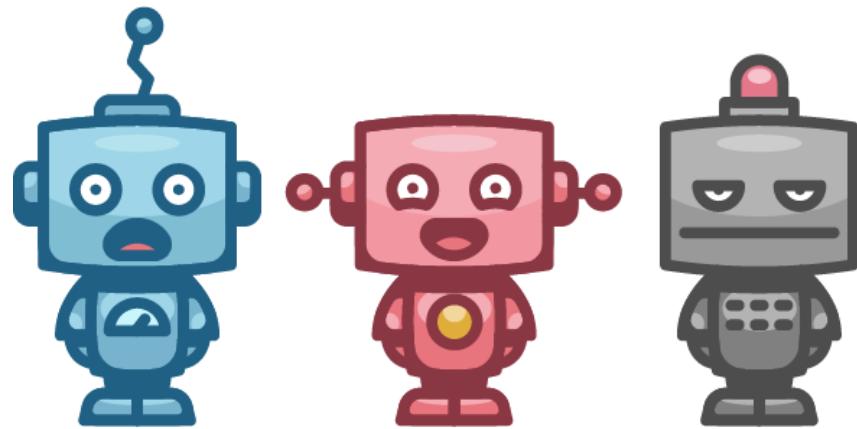


Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования

САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО
ФСУиР

ЛАБОРАТОРНАЯ РАБОТА №4:

Вариант №2



Выполнил: Гусаров С.А.
Группа: Р4133с

ОСНОВНАЯ ЧАСТЬ

Рассмотрим модель системы замкнутой кинематики «Optimus», представленной на рисунке 1 из прошлой лабораторной работы

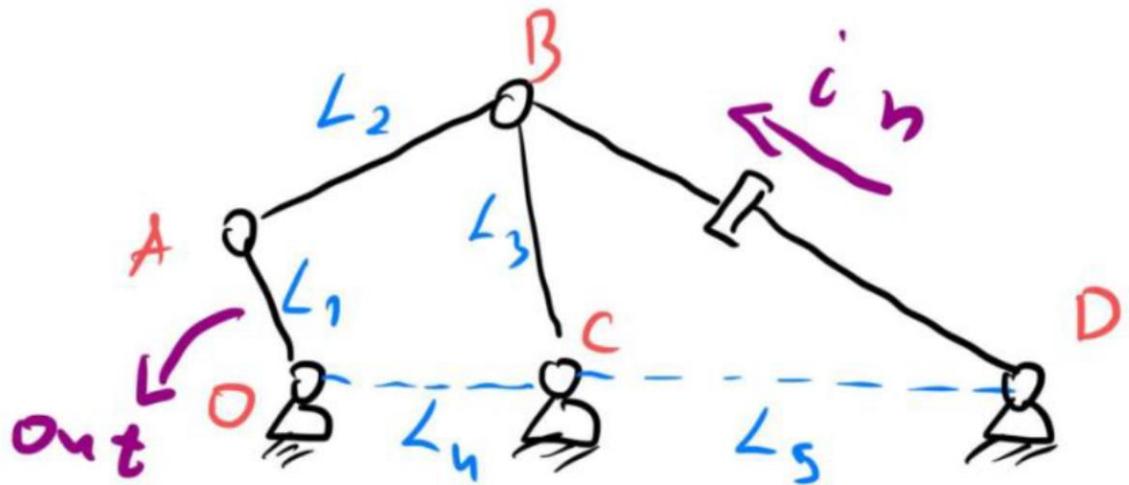


Рисунок 1 – Исследуемая система

Рассмотрим задающий сигнал вида:

$$q_{des}(t) = A \sin(\omega t) + \varphi, \quad (1.1)$$

где $A = 36.61 [^{\circ}]$, $\omega = 1.21 [\text{Гц}]$, $\varphi = -28 [^{\circ}]$

Добавим к механизму привод и реализуем движение с помощью ПД-регулятора. Приведем код управляющей программы на Python, измененный файл модели в xml-формате, а также визуализацию и графики движения механизма

```

1  <?xml version='1.0' encoding='UTF-8'?>
2  <mujoco>
3      <option timestep="1e-4"/>
4      <option gravity="0 0 -9.8"/>
5
6      <asset>
7          <material name="m_OA" rgba="1 0 0 1"/>
8          <material name="m_AB" rgba="0 1 0 1"/>
9          <material name="m_CB" rgba="0 0 1 1"/>
10         <material name="m_DB" rgba="1 1 0 1"/>
11         <texture type="skybox" builtin="gradient" rgb1="0.8 0.9 1" rgb2="0.2 0.3 0.6" width="265" height="256"/>
12         <texture name="grid" type="2d" builtin="checker" rgb1="0.1 0.1 0.1" rgb2="0.3 0.3 0.3" width="300" height="300"/>
13         <material name="grid" texture="grid" texrepeat="10 10" reflectance="0.1"/>
14     </asset>
15
16     <worldbody>
17         <light pos="0 0 10" diffuse="0.8 0.8 0.8" specular="0.5 0.5 0.5"/>
18         <light pos="2 2 5" diffuse="0.5 0.5 0.5" specular="0.3 0.3 0.3"/>
19         <geom type="plane" size="0.5 0.5 0.1" material="grid"/>
20         <site name="fixed_O" pos="0 0 0.375" size="0.008"/>
21         <site name="fixed_C" pos="0.075 0 0.375" size="0.008"/>
22         <site name="fixed_D" pos="0.1125 0 0.375" size="0.008"/>
23
24         <body name="OAB" pos="0 0 0.375" euler="0 0 0">
25             <joint name="O" type="hinge" axis="0 -1 0" damping="0.1"/>
26             <geom name="point_O" type="sphere" pos="0 0 0" size="0.008"/>
27             <geom name="link_OA" type="cylinder" material="m_OA"
28                 pos="0.0375 0 0" size="0.004 0.0375" euler="0 90 0"/>
29             <site name="sA" size="0.004" pos="0.075 0 0"/>
30             <body name="AB" pos="0.075 0 0" euler="0 0 0">
31                 <body name="AB" pos="0.075 0 0" euler="0 0 0">
32                     <joint name="A" type="ball" damping="0.1"/>
33                     <geom name="point_A" type="sphere" pos="0 0 0" size="0.008"/>
34                     <geom name="link_AB" type="cylinder" material="m_AB"
35                         pos="0.04875 0 0" size="0.004 0.04875" euler="0 90 0"/>
36                     <site name="sB" size="0.004" pos="0.0975 0 0"/>
37                 </body>
38             </body>
39             <body name="CB" pos="0.075 0 0.375" euler="0 0 0">
40                 <joint name="C" type="hinge" axis="0 -1 0" damping="0.1"/>
41                 <geom name="point_C" type="sphere" pos="0 0 0" size="0.008"/>
42                 <geom name="link_CB" type="cylinder" material="m_CB"
43                     pos="0.05625 0 0" size="0.004 0.05625" euler="0 90 0"/>
44                 <site name="sB_CB" size="0.004" pos="0.1125 0 0"/>
45             </body>
46
47             <body name="DB" pos="0.1125 0 0.375" euler="0 0 0">
48                 <joint name="D" type="hinge" axis="0 -1 0" damping="0.1"/>
49                 <geom name="point_D" type="sphere" pos="0 0 0" size="0.008"/>
50                 <geom name="link_DB" type="cylinder" material="m_DB"
51                     pos="-0.0375 0 0" size="0.004 0.0375" euler="0 90 0"/>
52                 <site name="sB_DB" size="0.004" pos="-0.075 0 0"/>
53             </body>
54         </worldbody>
55
56         <actuator>
57             <motor name="motor_O" joint="O" ctrlrange="-10 10" gear="1"/>
58         </actuator>
59
60         <sensor>
61             <jointpos name="sensor" joint="O"/>
62         </sensor>
63
64         <equality>
65             <connect name="eq_C1_C2" site1="sB" site2="sB_CB" solimp="0.995 0.99 0.001" solref="0.01 1"/>
66             <connect name="eq_C1_C3" site1="sB" site2="sB_DB" solimp="0.995 0.99 0.001" solref="0.01 1"/>
67         </equality>
68     </mujoco>

```

Рисунок 1 – Код xml-файла модели

Полученный механизм покажем на рисунке 2

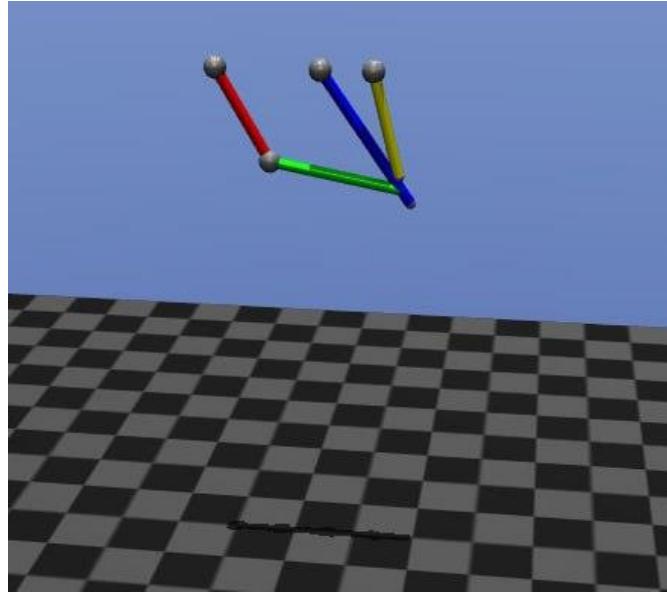


Рисунок 2 – Модель механизма в среде MoJoCo

Ниже представим код программы для запуска визуализации данной модели с реализацией ПД-регулятора

```
1 import time
2 import mujoco
3 import mujoco.viewer
4 import numpy as np
5 import matplotlib.pyplot as plt
6
7 xml_file = "model.xml"
8
9 omega = 1.31
10 a = np.deg2rad(36.61)
11 phi = np.deg2rad(-28)
12
13 kp = 25
14 kd = 1.5
15
16 def main():
17     model = mujoco.MjModel.from_xml_path(xml_file)
18     data = mujoco.MjData(model)
19     sensor_name = "sensor"
20
21     data.qpos[:] = 0.0
22     data.qvel[:] = 0.0
23
24     data_t = []
25     data_qref = []
26     data_q = []
27     data_e = []
28
29     with mujoco.viewer.launch_passive(model, data) as viewer:
30         |
```

```

31     t0 = time.time()
32     step = 0
33     prev_sensed_pos = 0.0
34
35     while viewer.is_running():
36         t = time.time() - t0
37         if t >= 30.0:
38             break
39
40         q_des = a * np.sin(omega * t) + phi
41         sid = mujoco.mj_name2id(model, mujoco.mjOBJ_SENSOR, sensor_name)
42         sensed_pos = data.sensordata[model.sensor_addr[sid]]
43
44         if step == 0:
45             sensed_vel = 0.0
46         else:
47             dt = model.opt.timestep
48             sensed_vel = (sensed_pos - prev_sensed_pos) / dt
49
50         prev_sensed_pos = sensed_pos
51
52         u = kp * (q_des - sensed_pos) - kd * sensed_vel
53         data.ctrl[0] = u
54
55         mujoco.mj_step(model, data)
56         viewer.sync()
57
58         if step % 10 == 0:
59             data_t.append(t)
60             data_qref.append(np.rad2deg(q_des))
61             data_q.append(np.rad2deg(sensed_pos))
62             data_e.append(u)
63
64             step += 1
65
66
67     data_t = np.array(data_t)
68     data_qref = np.array(data_qref)
69     data_q = np.array(data_q)
70     data_e = data_qref - data_q
71
72     plt.figure()
73     plt.plot(data_t, data_qref, 'r--', label='Reference')
74     plt.plot(data_t, data_q, 'b', label='Current')
75     plt.xlabel('time, [s]')
76     plt.ylabel('q(t)')
77     plt.legend()
78     plt.grid(True)
79     plt.tight_layout()
80     plt.show()
81
82     plt.figure()
83     plt.plot(data_t, data_e)
84     plt.xlabel('time, [s]')
85     plt.ylabel('e(t)')
86     plt.grid(True)
87     plt.tight_layout()
88     plt.show()
89
90 if __name__ == "__main__":
91     main()
92

```

Рисунок 3 – Код управляющей программы на Python

На графиках рисунков 4 – 6 покажем переходную характеристику движения предложенного механизма, а также ошибку регулирования при различных значениях коэффициентов регулятора.

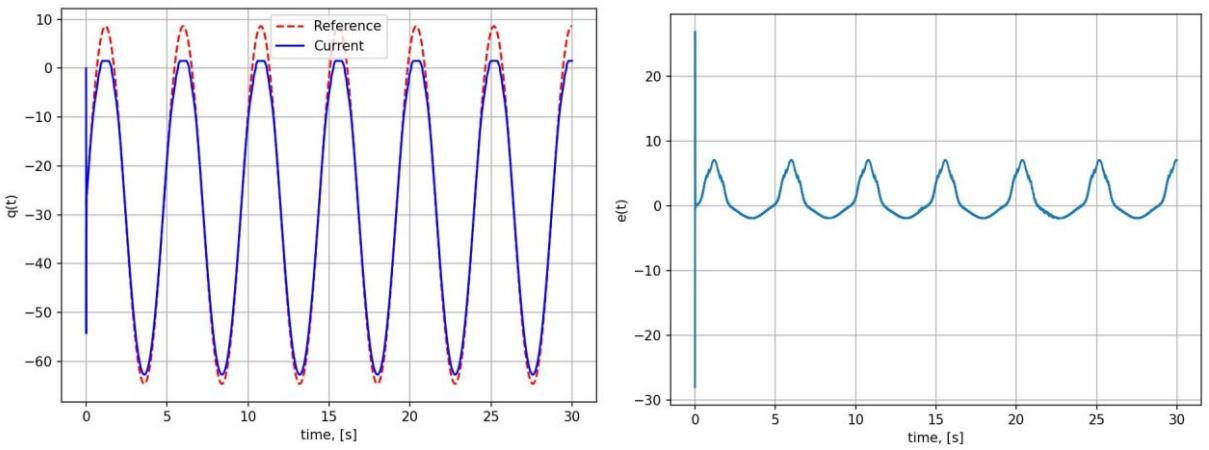


Рисунок 4 – Переходная характеристика и ошибка регулирования
при $k_p = 15, k_d = 0.5$

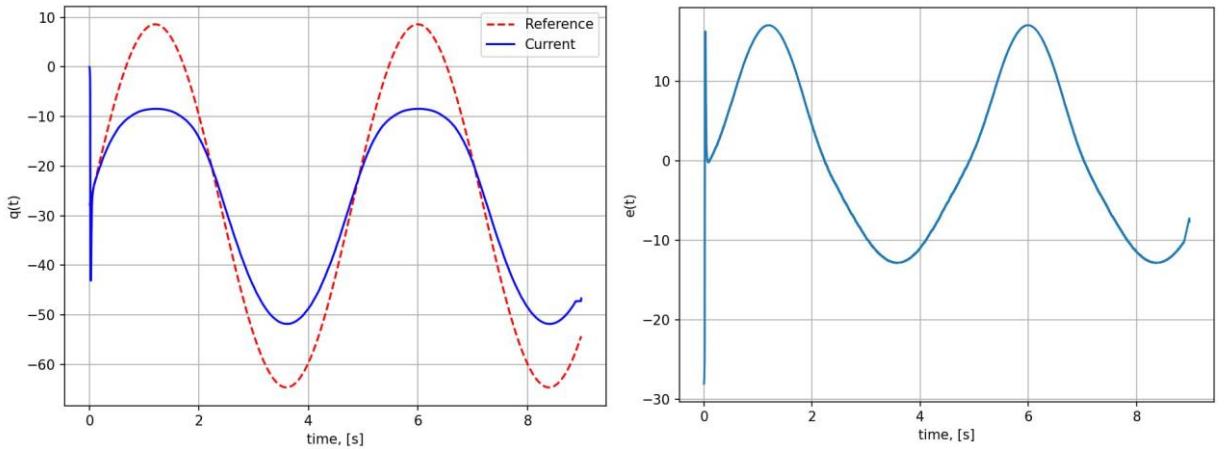


Рисунок 5 – Переходная характеристика и ошибка регулирования
при $k_p = 10, k_d = 0.07$

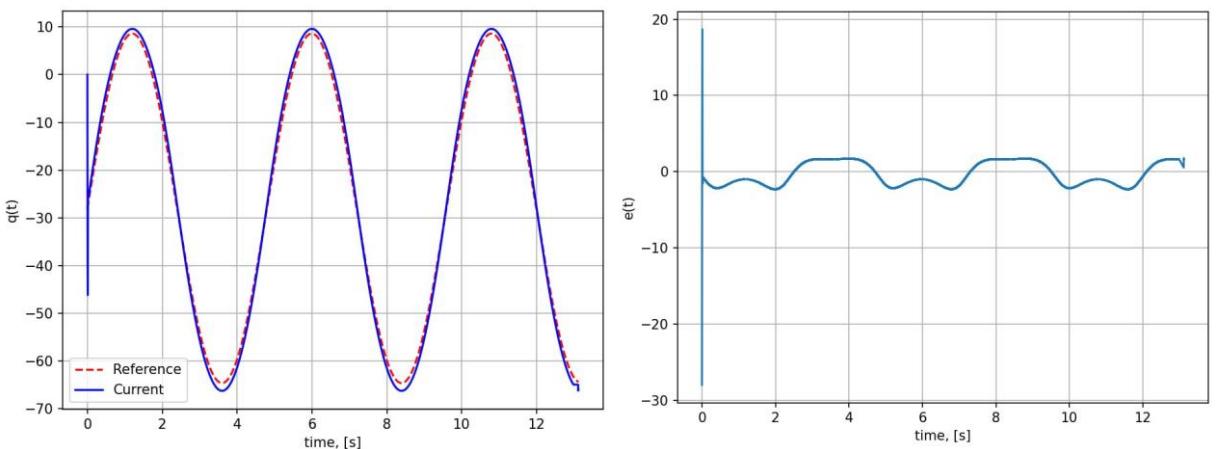


Рисунок 6 – Переходная характеристика и ошибка регулирования
при $k_p = 10, k_d = 0.07$

ЗАКЛЮЧЕНИЕ

В ходе данной лабораторной работы мы составили и описали модель системы замкнутой кинематики «Optimus». А также провели моделирование в среде MoJoCo и привели программу, xml-файл и вид механизма в данной среде.

Мы реализовали движение механизма за эталонным сигналом при помощи классического ПД-регулятора, а также привели графики переходной характеристики и ошибки слежения для различных наборов пропорционального и дифференциального коэффициентов.

Лабораторную работу считаю выполненной полностью.