

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
УНИВЕРСИТЕТ ИТМО
ФАКУЛЬТЕТ СИСТЕМ УПРАВЛЕНИЯ И РОБОТОТЕХНИКИ



Лабораторная работа №4
Имитационное моделирование робототехнических систем

Выполнил:
Студент группы R4134с
Воронин В.С.
Преподаватель:
Борисов И. И.

Санкт-Петербург, 2025

1. Оглавление

2. Цель работы	2
3. Описание системы	2
4. Ход выполнения	3
Добавление <actuator> и <sensor>	3
Составление ПД-регулятора	3
5. Моделирование	4
6. Выводы	5

2. Цель работы

1. К модели, созданной в предыдущем задании, необходимо добавить приводы.
2. Добавить <actuator> и <sensor>.
3. Составить ПД-регулятор, который будет управлять системой по $q^{des} = AMP \cdot \sin(FREQ \cdot t) + BIAS$
4. Составить графики и написать отчет с пояснениями

3. Описание системы

В таблице 1 указаны параметры системы, а на рисунке 1 представлено схематическое описание системы.

q1			q2		
AMP, deg	FREQ, Hz	BIAS, deg	AMP, deg	FREQ, Hz	BIAS, deg
25.58	1.03	-42.6	58.7	3.76	-44.3

Таблица 1 Параметры системы

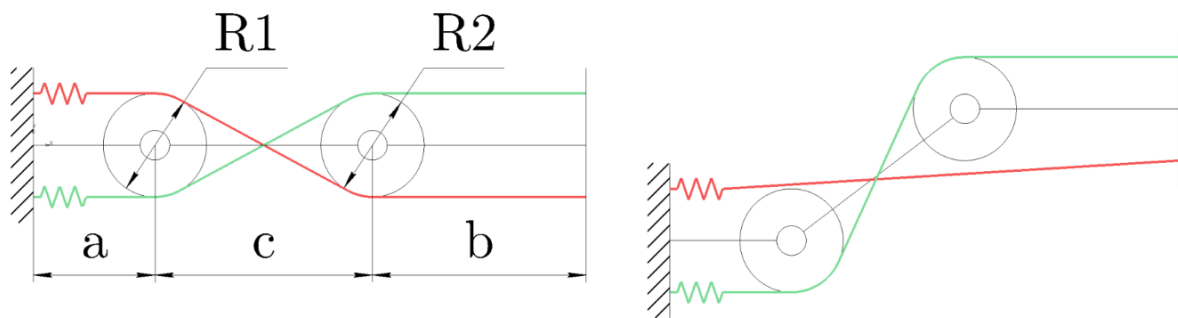


Рисунок 1 Схематическое представление модели

4. Ход выполнения

Добавление <actuator> и <sensor>

К пассивной системе в xml-файла из работы №2 добавим привода:

```
<actuator>
  <general name="motor_R1" joint="joint_R1" ctrllimited="true"
ctrlrange="-100 100"/>
  <general name="motor_R2" joint="joint_R2" ctrllimited="true"
ctrlrange="-100 100"/>
</actuator>

<sensor>
  <jointpos name="sensor_R1_pos" joint="joint_R1"/>
  <jointpos name="sensor_R2_pos" joint="joint_R2"/>

  <jointvel name="sensor_R1_vel" joint="joint_R1"/>
  <jointvel name="sensor_R2_vel" joint="joint_R2"/>
</sensor>
```

Составление ПД-регулятора

Для управления создадим простейший PD-регулятор и путем ручной настройки зададим ему необходимые для оптимальной работы коэффициенты K_p и K_d .

```
def PD_Control_R1(mj_data, KP, KD, desired_pos):
    current_pos = mj_data.qpos[0]
    current_vel = mj_data.qvel[0]
    PD = KP * (desired_pos - current_pos) + KD * (0 - current_vel)
    PD = np.clip(torque, -300, 300)
    mj_data.ctrl[0] = PD

def PD_Control_R2(mj_data, KP, KD, desired_pos):
    current_pos = mj_data.qpos[1]
    current_vel = mj_data.qvel[1]
```

```

PD = KP * (desired_pos - current_pos) + KD * (0 - current_vel)
PD = np.clip(torque, -300, 300)
mj_data.ctrl[1] = PD

```

5. Моделирование

Опишем подаваемый сигнал, слегка уменьшив коэффициенты для более точного схождения:

```

# Конвертация в радианы
DEG_TO_RAD = np.pi / 180
RAD_TO_METER = 0.1 # коэффициент преобразования радиан в метры

# Преобразование в метры для линейных суставов
AMP1 = 25.58 * DEG_TO_RAD * RAD_TO_METER * 0.8
FREQ1 = 1.03*0.7
BIAS1 = -42.6 * DEG_TO_RAD * RAD_TO_METER
AMP2 = 58.7 * DEG_TO_RAD * RAD_TO_METER * 0.7
FREQ2 = 3.76*0.7
BIAS2 = -44.3 * DEG_TO_RAD * RAD_TO_METER

```

Эмпирическим путем подберем коэффициенты для ПД-регулятора:

```

KP_R1 = 10000
KV_R1 = 100
KP_R2 = 5000
KV_R2 = 10

```

И получаем следующие графики системы:

Смоделируем систему и получим следующие графики:

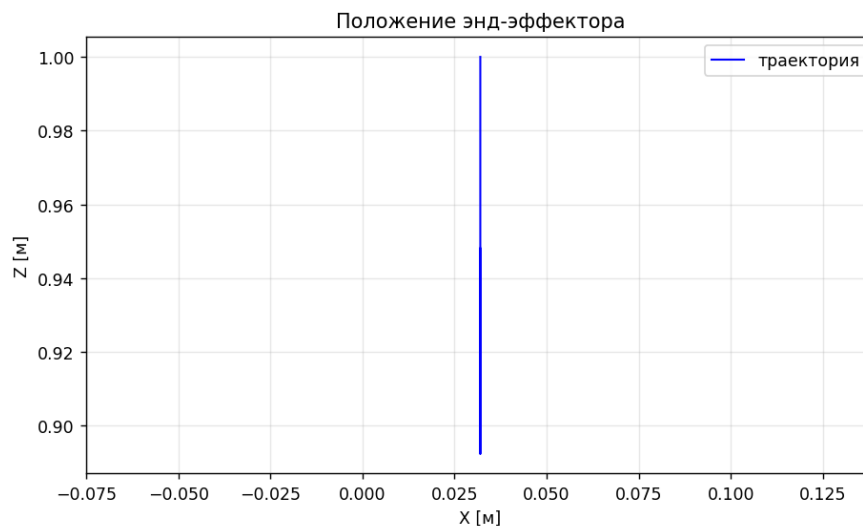


Рисунок 2 График траектории движения end-effector'a

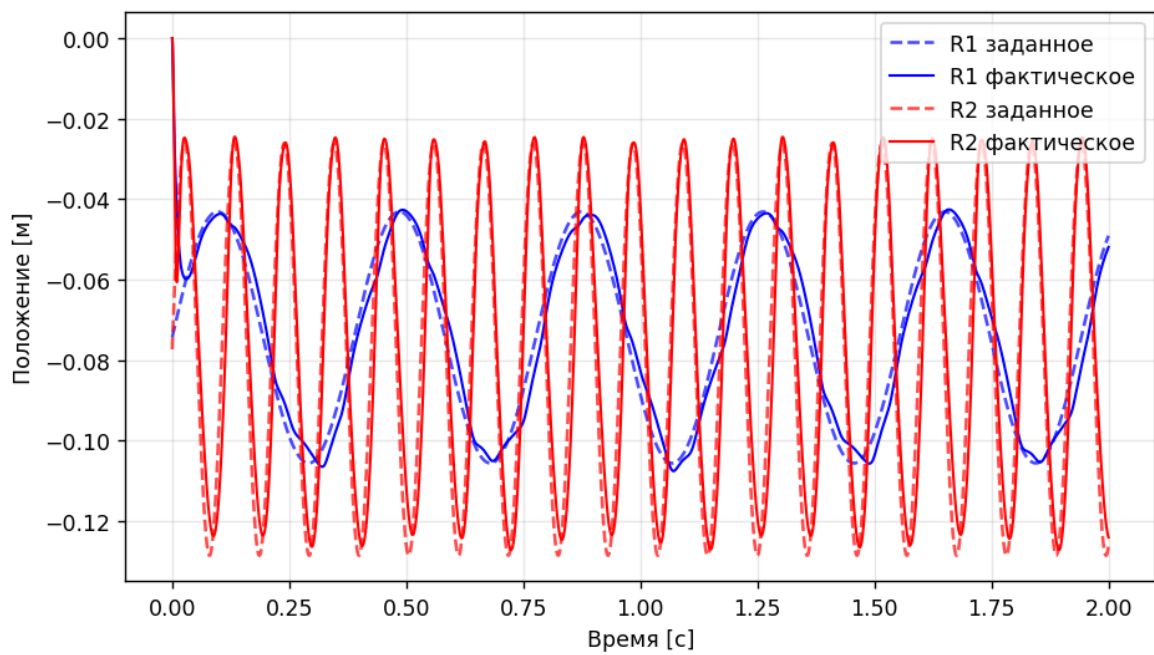


Рисунок 3 Графики зависимости фактического и заданного движения сочленений

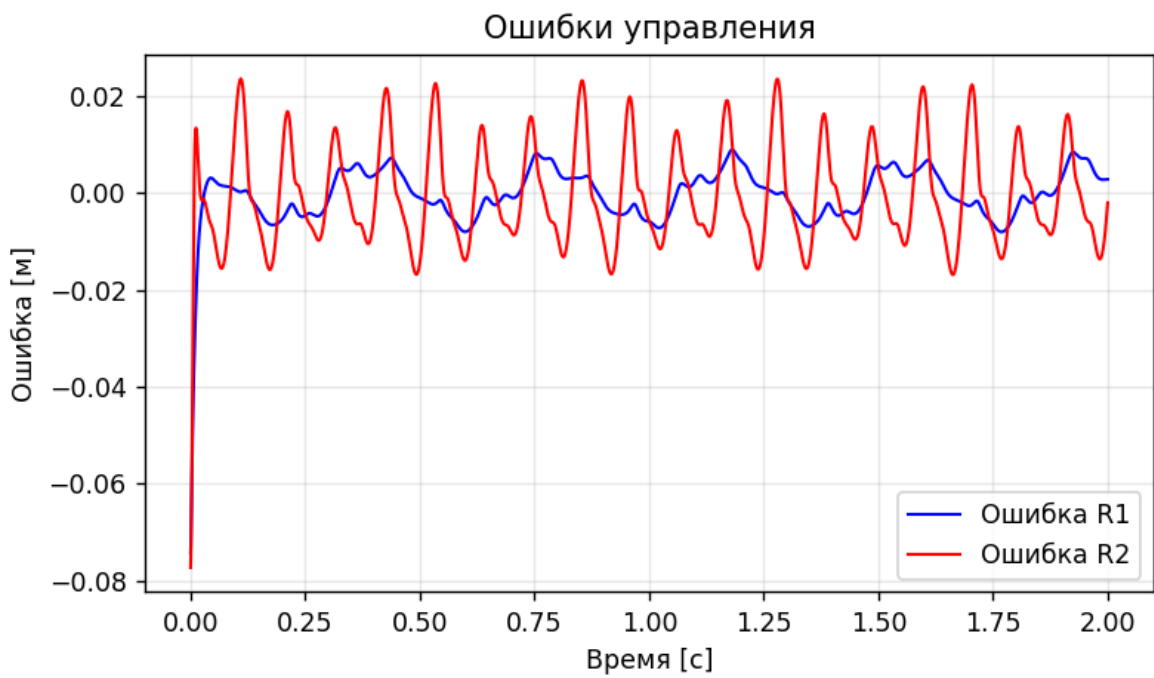


Рисунок 5 График ошибки управления

6. Выводы

В ходе лабораторной работы было проведено моделирование системы RR механизма открытой кинематики с добавлением приводов. Также была

смоделирована система с синусоидальным подающим сигналом, который система успешно повторила, соответственно были написаны функции для управления актуаторами и получения положения энд-эффектора.