Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение высшего образования
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по практической работе №3

Дисциплина: “Имитационное моделирование робототехнических систем”

Автор: Балакин А.Р.

Вариант: 6

Факультет: СУиР

Преподаватель: Ракшин Е.А.

Санкт-Петербург, 2025

Входные данные

R_1, м	R_2, м	a, м	b, м	c, м
0.02	0.013	0.039	0.083	0.036

Таблица 1 - Значения входных данных

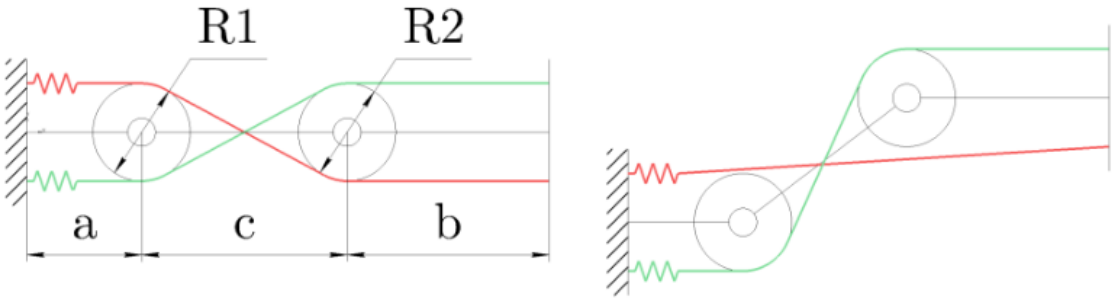

Рисунок 1 - Моделируемый механизм

К сожалению, у меня единственного в варианте отсутствовал параметр R_1. Изначально я хотел взять значение из предыдущего варианта $= 0.024$ м, однако тогда из-за малого расстояния c шкивы бы упирались друг в друга, поэтому я взял указанное в таблице 1 значение.

Задание

a) Написание XML формата

На рисунке 1 можно увидеть два шкива, которые, во-первых, огибаются двумя нитями, а также их центры/оси вращения связаны между собой и с опорами справа и слева. Соответственно базовые параметры “суставов”, необходимых для сохранения подвижности механизмом.

```
<default>
  <joint type="hinge" axis="0 0 1" limited="true" range="-180 180" damping="0.1"/>
  <geom type="capsule" size="0.0025" rgba="0.4 0.4 0.4 1" density="1000"/>
</default>
```

Рисунок 2 - Код default

Далее перейдем к формированию левой опоры.

```
<worldbody>

  <body name="left_support" pos="0 0 0">
    <geom name="left_support_geom" type="box" size="0.001 0.06 0.06" rgba="0.8 0.7 0.6 1"/>
    <site name="left_support_p1" pos="0 0.020 0" size="0.001" rgba="1 0 0 1"/>
    <site name="left_support_mp" pos="0 0 0" size="0.001" rgba="0.4 0.4 0.4 1"/>
    <site name="left_support_p2" pos="0 -0.020 0" size="0.001" rgba="0 0 1 1"/>
  </body>
```

Рисунок 3 - Код для левой опоры

Затем посредством вложенности формируем шкивы, связи между ними и правую опору. Также на опорах и на шкивах формируем точки, через которые будут протянуты нити.

```
  <body name="link1" pos="0 0 0">
    <joint name="joint1" pos="0 0 0"/>
    <geom fromto="0 0 0 0.039 0 0"/>
    <site name="link1_left" pos="0 0 0" size="0.001"/>

    <body name="pulley1" pos="0.039 0 0">
      <geom name="pulley1_geom" type="cylinder" size="0.020 0.002" rgba="0.8 0.7 0.6 0.5"/>
      <site name="pulley1_p1" pos="0 0.020 0" size="0.001"/>
      <site name="pulley1_p2" pos="0 -0.020 0" size="0.001"/>

      <body name="link2" pos="0 0 0">
        <joint name="joint2" pos="0 0 0"/>
        <geom fromto="0 0 0 0.036 0 0"/>

        <body name="pulley2" pos="0.036 0 0">
          <geom name="pulley2_geom" type="cylinder" size="0.013 0.002" rgba="0.8 0.7 0.6 0.5"/>
          <site name="pulley2_p1" pos="0 0.013 0" size="0.001"/>
          <site name="pulley2_p2" pos="0 -0.013 0" size="0.001"/>

          <body name="link3" pos="0 0 0">
            <joint name="joint3" pos="0 0 0"/>
            <geom fromto="0 0 0 0.083 0 0"/>
            <site name="link3_right" pos="0.083 0 0" size="0.001"/>

            <body name="right_wall" pos="0.083 0 0">
              <geom name="right_support_geom" type="box" size="0.001 0.02 0.02" rgba="0.8 0.7 0.6 1"/>
              <site name="right_support_p1" pos="0 -0.013 0" size="0.001" rgba="1 0 0 1"/>
              <site name="right_support_mp" pos="0 0 0" size="0.001" rgba="0.4 0.4 0.4 1"/>
              <site name="right_support_p2" pos="0 0.013 0" size="0.001" rgba="0 0 1 1"/>
            </body>
```

Рисунок 4 - Код для построения шкивов

После этого задаем нити и актуаторы.

```
<tendon>
    <spatial name="tendon1" limited="true" range="0.1 0.5" width="0.002" rgba="1 0 0 1" stiffness="1000" damping="0.1">
      <site site="left_support_p1"/>
      <site site="pulley1_p1"/>
      <site site="pulley2_p2"/>
      <site site="right_support_p1"/>
    </spatial>
```

Рисунок 5 - Пример задания первой нити

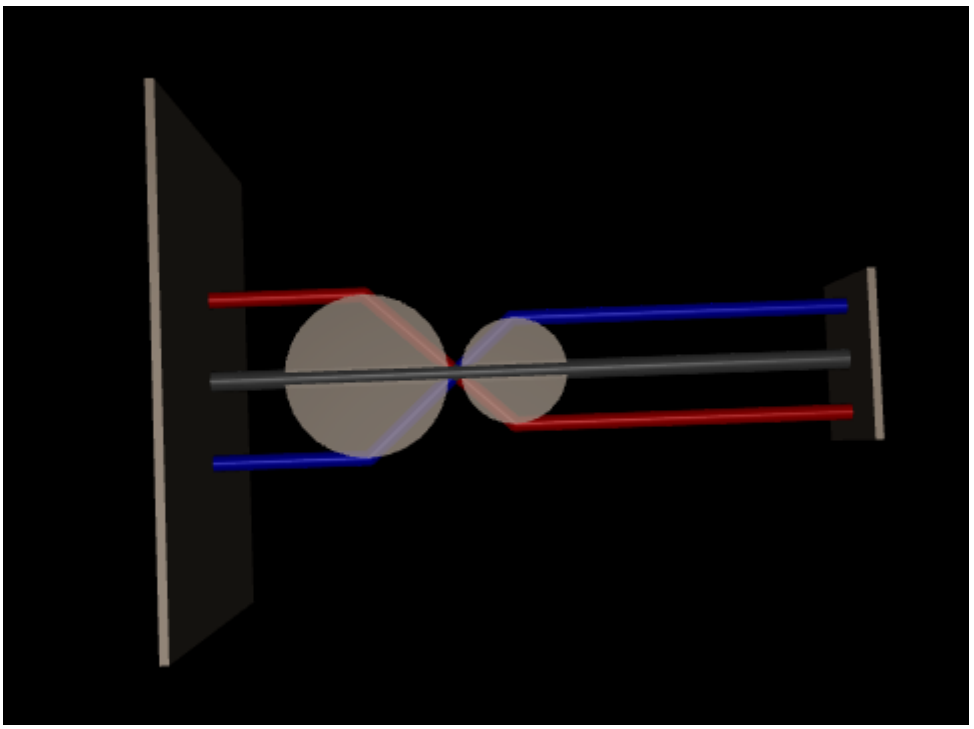

Рисунок 6 - Получившаяся модель

b) Скрипт симуляции

Для запуска симуляции и хранения данных был написан следующий скрипт, подающий на актуаторы синусоидальное управление:

```
model = mujoco.MjModel.from_xml_string(XML_Model)
data = mujoco.MjData(model)
data.qpos[0] = np.deg2rad(0)
data.qpos[1] = np.deg2rad(0)
data.qvel[0] = 0.0
data.qvel[1] = 0.0
time_data = []
qpos_data = []
ctrl_data = []
steps_per_frame = 1

with viewer.launch_passive(model, data) as v:
  while v.is_running():
    for _ in range(steps_per_frame):
      data.ctrl[0] = np.sin(data.time/100)
      data.ctrl[1] = np.cos(data.time/100)
      mujoco.mj_step(model, data)
      time_data.append(data.time)
      qpos_data.append(data.qpos.copy())
      ctrl_data.append(data.ctrl.copy())
    v.sync()

time_data = np.array(time_data)
qpos_data = np.array(qpos_data)
ctrl_data = np.array(ctrl_data)
```

Рисунок 7 - Скрипт симуляции

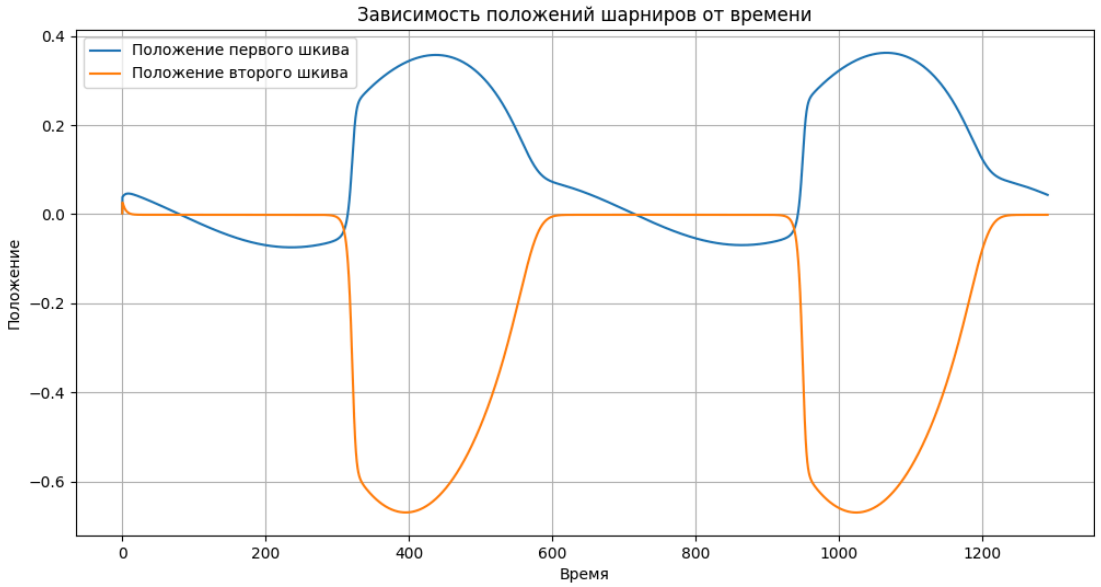

Рисунок 8 - Графики положений шкивов

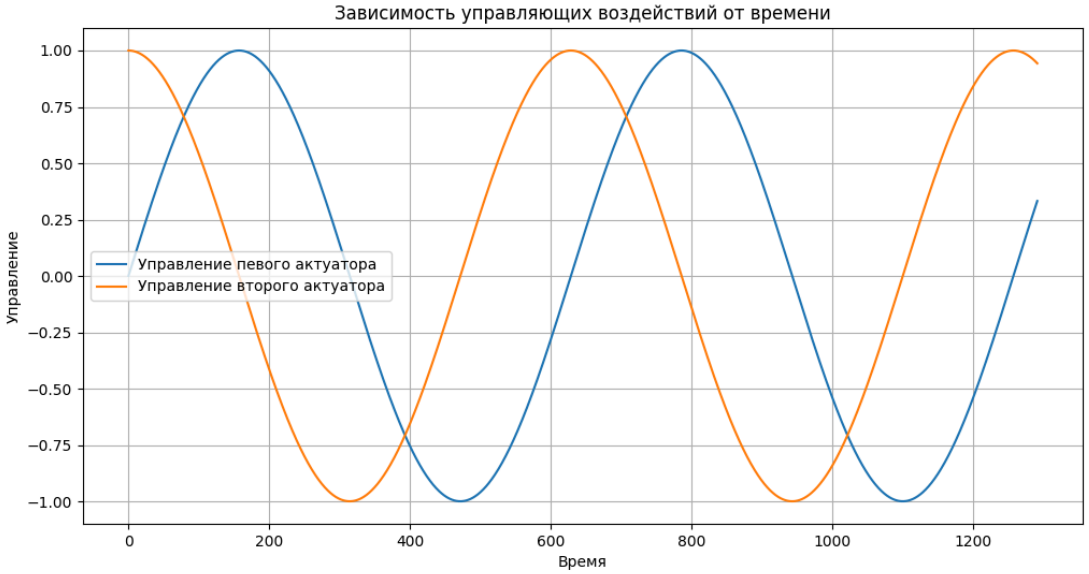

Рисунок 9 - Графики управляющих воздействий

Выводы

В ходе выполнения данного практического задания был описан заданный механизм в формате XML в соответствии с требуемыми параметрами, а также написан скрипт, осуществляющий запуск симуляции и сохранением данных для дальнейшего построения графиков.