

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Лабораторная работа №1
по дисциплине «Имитационное моделирование робототехнических систем»

Выполнил:
студент гр. R41341с
Борисов М. В.

Преподаватель:
Бжихатлов И. А.

Санкт-Петербург
2021 г.

Цель работы

- Средствами симулятора собрать модель схвата манипулятора
- Использовать сенсор для обнаружения объектов в схвате
- Написать скрипт активации схвата на основе данных датчика

Ход работы

Соберём из простейших элементов примитивный схват манипулятора. Для этого воспользуемся кубами и линейными актуаторами. Соответственно кубы представляют собой конкретные детали схвата, а актуаторы это некоторый виртуальный линейный мотор. При этом актуатору не обязательно находиться в непосредственном контакте с деталью, поскольку у него нет физических свойств.

Добавив на левую часть схвата сенсор получим модель изображённую на рисунке 1.

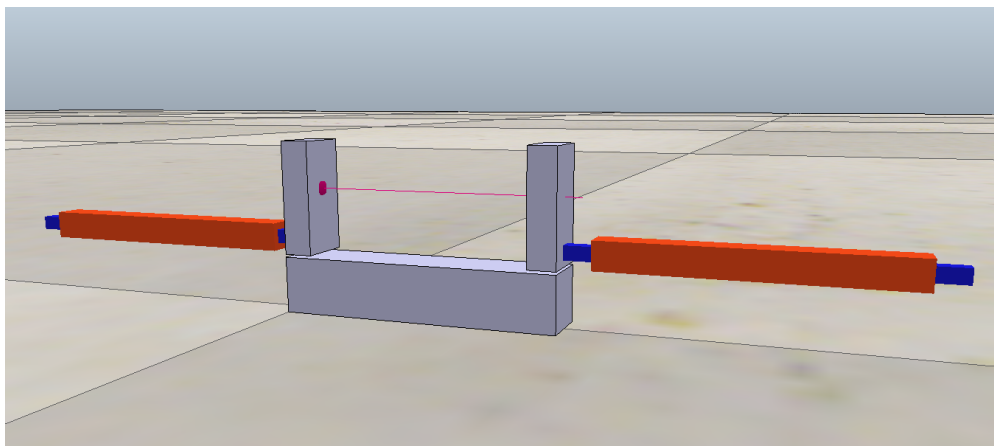


Рис. 1: Модель схвата

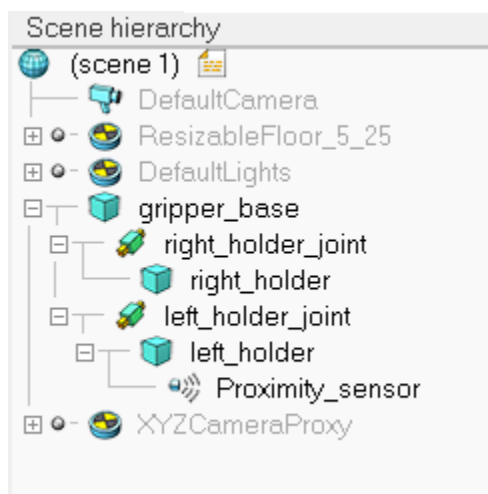


Рис. 2: Дерево модели в процессе работы

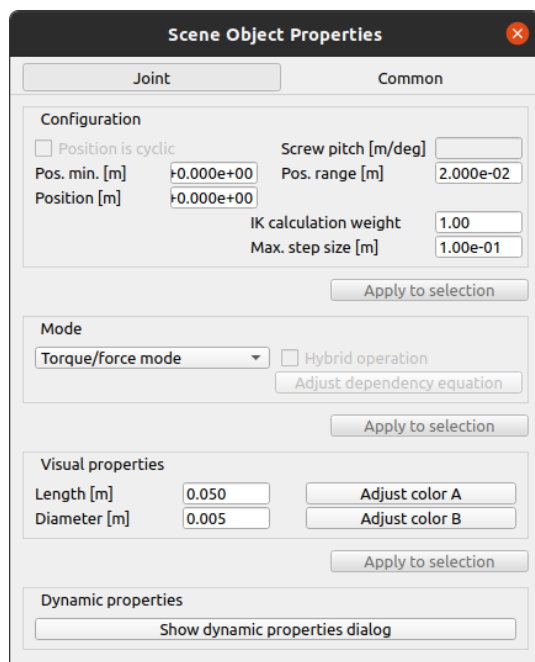


Рис. 3: Параметры линейного актуатора

Добавим в сцену объект для обнаружения датчиком и напишем простой скрипт (Приложение А), который считывает с датчика данные и, как только в его пределах видимости появляется деталь, подаёт команду на схват.

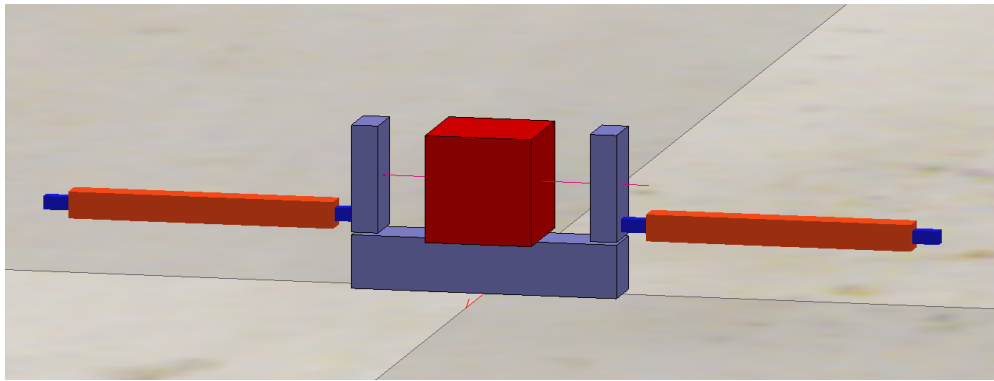


Рис. 4: Схват с объектом для обнаружения

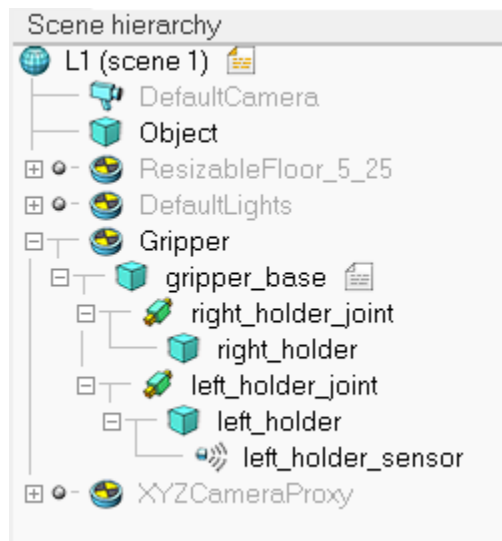


Рис. 5: Дерево модели по окончании работы

Вывод

В работе были изучены примитивы и простейший сенсор предоставляемые симулятором. Также в процессе были освоены базовые принципы языка программирования Lua, на котором пишутся скрипты для описания поведения объектов.

А. Управляющий скрипт

```

1  function sysCall_init ()
2      left_holder = sim.getObjectHandle("left_holder_joint")
3      right_holder = sim.getObjectHandle("right_holder_joint")
4      sensor = sim.getObjectHandle("left_holder_sensor")
5      min_gap = 1e-4 -- minimal gap that should be between sensor and
                       ↳ object
  
```

```

6      dist_thresh_to_grip = 50e-3 / 2 -- distance after which we decide
      ↪ to grab object
7      joint_grip_velocity = 0.005
8      print(min_gap)
9      print(dist_thresh_to_grip)
10     print(joint_grip_velocity)
11 end
12
13 function sysCall_actuation()
14     result, distance = sim.readProximitySensor(sensor)
15     if result > 0 then
16         stop = (distance <= min_gap)
17         grab = (distance <= dist_thresh_to_grip)
18         print(distance)
19         print(grab)
20         print(stop)
21         if grab and not stop then
22             sim.setJointTargetVelocity(left_holder,
                ↪ joint_grip_velocity)
23             sim.setJointTargetVelocity(right_holder,
                ↪ joint_grip_velocity)
24         else
25             sim.setJointTargetVelocity(left_holder, 0)
26             sim.setJointTargetVelocity(right_holder, 0)
27         end
28     end
29 end
30 end
31
32 function sysCall_sensing()
33     -- put your sensing code here
34 end
35
36 function sysCall_cleanup()
37     -- do some clean-up here
38 end
39
40 -- See the user manual or the available code snippets for additional
    ↪ callback functions and details

```