习题1

1. 机械臂系统通常由 机械系统、驱动系统、控制系统和感知系统四部分组成。

2. 机械臂的自由度的Grübler计算公式是什么？



3. 机械臂的自由度通常等于机械臂关节的数目。

4. 姿态可以通过四元数、欧拉角和旋转矩阵三种方式来描述。

5. 一般变换包含平移坐标变换和旋转坐标变换两部分。

6. 机械臂的正运动学是什么？

机械臂的正运动学（forward kinematics，FK）是已知一组关节角向量，求解机械臂末端执行器（工具坐标系）的位置和姿态的静态几何问题。即如何将已知的机械臂关节空间描述转化为笛卡儿空间的描述。

7. 机械臂的逆运动学是什么？

机械臂逆运动学（inverse kinematics，IK）是指已知机械臂末端（工具坐标系）相对于基坐标系的期望位置和姿态，求解一系列满足期望的关节角的问题，相当于将机械臂从笛卡尔空间映射到关节空间。

习题2

1. Ubuntu18.04系统对应使用的ROS版本为Melodic。
2. ROS工作空间默认使用Catkin编译系统。
3. 功能包（package）是ROS代码的组织单元以及Catkin编译的基本单元，必须包含package.xml文件和CMakeLists.txt文件。
4. [单选题]如果要下载一个功能包保存到工作空间中，下列哪个路径是合适的存放位置？（ C ）
5. catkin\_ws B. tutorial\_ws/build C. catkin\_ws/src D. my\_ws/devel
6. 可以使用catkin\_create\_pkg命令来创建一个功能包。
7. 可以使用catkin\_make命令来编译工作空间中的功能包。
8. 节点是ROS中最小的执行单元。
9. 启动 ROS节点管理器的命令是roscore。
10. [多选]关于ROS通信的描述，下列选项正确的有？（ A C）

A. 话题通信是一种异步通信机制

B. 一个话题至少要有一个发布者和一个接收者

C. 一个服务的服务端节点可以对应多个客户端节点

D. 除了最终结果，服务通信也可反馈执行的中间过程状态

1. 简要对比ROS中话题、服务和动作三种通信方式的异同？

|  |  |  |  |
| --- | --- | --- | --- |
|  | 话题（Topic） | 服务（Service） | 动作（Action） |
| 模式 | 发布/订阅 | 请求/应答 | 请求/应答 |
| 结果反馈 | 无 | 有 | 有 |
| 过程反馈 | 无 | 无 | 有 |
| 中断请求 | 无 | 无 | 有 |
| 节点关系 | 多对多 | 一（服务端）对多 | 一（服务端）对多 |
| 应用场景 | 高频的数据传输 | 执行某个任务 | 执行长时间任务 |

1. ROS图资源名称可以分为哪四类？并分别举例。

图资源命名可以分为以下四类：

①基本名称（base）：不能包含波浪号（~）或正斜杠（/），如base\_name、topic\_name。

②私有名称（private）：首字符是波浪号（~），如~private\_param。

③相对名称（relative）：首字符不包含波浪号（~）或正斜杠（/），后续字符中有正斜杠（/）区分前后两个命名空间，如relative/name。

④全局名称（global）：首字符是正斜杠（/），如/global\_name，/globa/name1/name2。

习题3

1. 消息的描述文件以.msg为后缀，通常放在ROS功能包的 msg 子文件夹中。
2. 若想查看geometry\_msgs/Point消息的完整定义，可在终端输入 rosmsg show geometry\_msgs/Point 命令。
3. 使用 rostopic hz /topic 命令可查看话题的发布频率。
4. 使用 rostopic list 命令可列出当前活跃的话题。
5. std\_srvs/SetBool服务类型的详细定义可使用 rossrv show std\_srvs/SetBool 命令查看。
6. 服务可以在终端使用rosservice call命令调用。
7. 使用 rosparam list 命令可列出当前的参数名。
8. 综合实践

（完整代码详见ros\_arm\_exercises仓库里的exercise\_three功能包，下面只对关键步骤进行说明）

1. 在ros工作空间中创建名为exercise\_three的功能包。

$ cd ~/tutorial\_ws/src/

$ catkin\_create\_pkg exercise\_three roscpp rospy std\_msgs

$ cd exercise\_three

$ mkdir msg srv

1. 在exercise\_three功能包中定义名为addend.msg的消息，消息包含三个float类型的成员x、y、z。

在msg文件下建立addend.msg文件，文件内容如下：

float32 x

float32 y

float32 z

参考书3.1.2节的内容修改package.xml和CMakeLists.txt文件，修改完成后编译功能包。

编译通过后，可用 rosmsg show exercise\_three/addend 命令查看消息的定义。

1. 编写话题发布节点pub\_test向话题/addend发布自定义的addend.msg。

源码位于exercise\_three/src/pub\_test.cpp。

可通过以下命令启动：

$ roscore

$ rosrun exercise\_three pub\_test

$ rosrun exercise\_three pub\_test.py

1. 编写话题订阅节点sub\_test订阅话题/addend的消息并计算三个加数的和，打印输出。

源码位于exercise\_three/src/sub\_test.cpp

可通过以下命令启动：

$ rosrun exercise\_three sub\_test

$ rosrun exercise\_three sub\_test.py

1. 在exercise\_three功能包中定义名为CallAdd.srv的服务，服务的请求为exercise\_three/addend.msg类型的成员addend，服务的应答为float类型的成员sum。

在srv文件下建立CallAdd.srv文件，文件内容如下（这里主要练习嵌套消息的使用，请求必须使用exercise\_three/addend.msg类型的成员addend）：

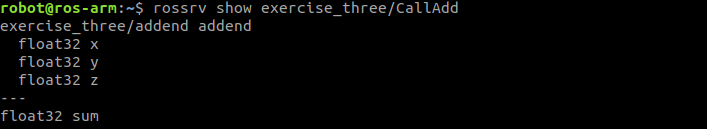
exercise\_three/addend addend

---

float32 sum

按照书3.4.2节的内容修改package.xml和CMakeLists.txt文件，修改完成后编译功能包。

编译通过后，可用 rossrv show exercise\_three/CallAdd 命令查看服务的定义。



1. 编写服务/call\_add的服务端节点，能够将请求的数据相加后返回加数的和。

源码位于exercise\_three/src/add\_server.cpp。

可通过以下命令启动：

$ roscore

$ rosrun exercise\_three add\_server

$ rosrun exercise\_three add\_server.py

1. 编写服务/call\_add的客户端节点，能够发送请求并获取服务的反馈打印输出

源码位于exercise\_three/src/add\_client.cpp。

可通过以下命令启动：

$ rosrun exercise\_three add\_client

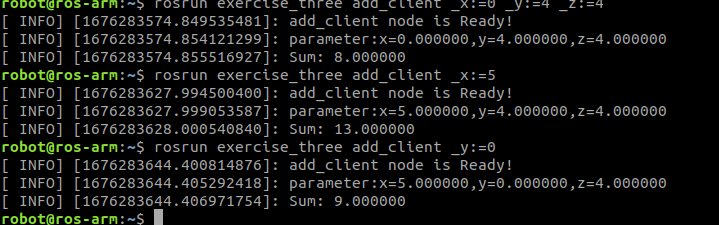
$ rosrun exercise\_three add\_client.py

1. 在习题7服务客户端节点的基础上添加获取参数代码，能够通过参数为请求数据中的加数赋值。

可通过rosrun命令启动节点时传递参数的值：

$ rosrun exercise\_three add\_client \_x:=0 \_y:=4 \_z:=4

$ rosrun exercise\_three add\_client.py \_y:=4 \_z:=4



习题4

1. Action使用.action文件定义，包含goal、result和feedback三部分，中间用“---”符号分隔。

2.[多选] Rviz可以图形化显示哪些类型的数据？（ABCD）

A.激光LaserScan B.点云 PointCloud

C.机器人模型RobotModel D.轨迹 Path

3.启动rqt\_reconfigure动态参数配置窗口，可以在终端输入rosrun rqt\_reconfigure rqt\_reconfigure命令。

4.ROS中通过/tf和/tf\_static两个话题维护整个TF树，话题的消息类型为tf2\_msgs/TFMessage。

5.综合实践

（完整代码详见ros\_arm\_exercises仓库里的exercise\_four功能包，下面只对关键步骤进行说明）

1. 在ros工作空间中创建名为exercise\_four的功能包。

$ cd ~/tutorial\_ws/src/

$ catkin\_create\_pkg exercise\_four roscpp rospy std\_msgs tf tf2

$ cd exercise\_four

$ mkdir launch rviz src

1. 在exercise\_four功能包中编写TF发布节点tf\_broadcaster，以20Hz的频率向外发布base\_link与tool\_link之间的TF，平移为（0.25，0，0.75），初始旋转欧拉角RPY为（0，0，0），其中偏航角Yaw会随着时间变化（可自行设计）。

源码位于exercise\_four/src/tf\_broadcaster.cpp

$ rosrun exercise\_four tf\_broadcaster

1. 在exercise\_four功能包中编写TF监听节点tf\_listener，监听base\_link与tool\_link间最新的TF，并打印输出到屏幕显示。

源码位于exercise\_four/src/tf\_listener.cpp

$ rosrun exercise\_four tf\_listener

1. 在exercise\_four功能包中编写tf\_demo.launch文件，文件内包含tf\_broadcaster节点、tf\_listener节点和rviz的启动，rviz中可显示TF。

源码位于exercise\_four/launch/tf\_demo.launch

$ roslaunch exercise\_four tf\_demo.launch

习题5

1. URDF是一种基于\_XML规范、用于描述机器人结构模型的格式。
2. urdf的<link>包含\_<inertial>\_\_、\_<visual>\_和\_<collision> 三个元素。
3. <joint>标签的type有哪几种类型？

URDF模型中的joint类型

|  |  |
| --- | --- |
| 关节类型 | 描述 |
| continuous | 旋转关节，可以围绕单轴无限旋转，如小车车轮 |
| revolute | 旋转关节，类似于continuous，但是有旋转的角度极限，机械臂的关节多为此类型 |
| prismatic | 滑动关节，沿某一轴线移动的关节，带有位置极限 |
| planar | 平面关节，允许在平面正交方向上平移或者旋转 |
| floating | 浮动关节，允许进行平移、旋转运动，如小车万向轮 |
| fixed | 固定关节，不允许运动的特殊关节 |

1. 可以使用 check\_urdf 命令对urdf文件进行语法检查。
2. RViz中可通过Add按钮添加\_RobotModel\_插件用来显示机器人模型。
3. xacro模型文件的常用语法有哪些？xacro模型和urdf模型有什么关系？

答：（1）常量定义（2）数学计算（3）宏定义和宏调用（4）条件判断（5）文件包含

xacro里的模型仍是urdf模型，只是在编写和管理上有了很大改进。xacro是urdf的进阶版本。

1. /joint\_states话题的消息类型为sensor\_msgs/JointState。
2. 综合实践
3. 创建my\_car\_description功能包。

$ catkin\_create\_pkg my\_car\_description std\_msgs roscpp rospy urdf

1. 在功能包内自行设计并创建一个四轮车底盘的urdf模型。

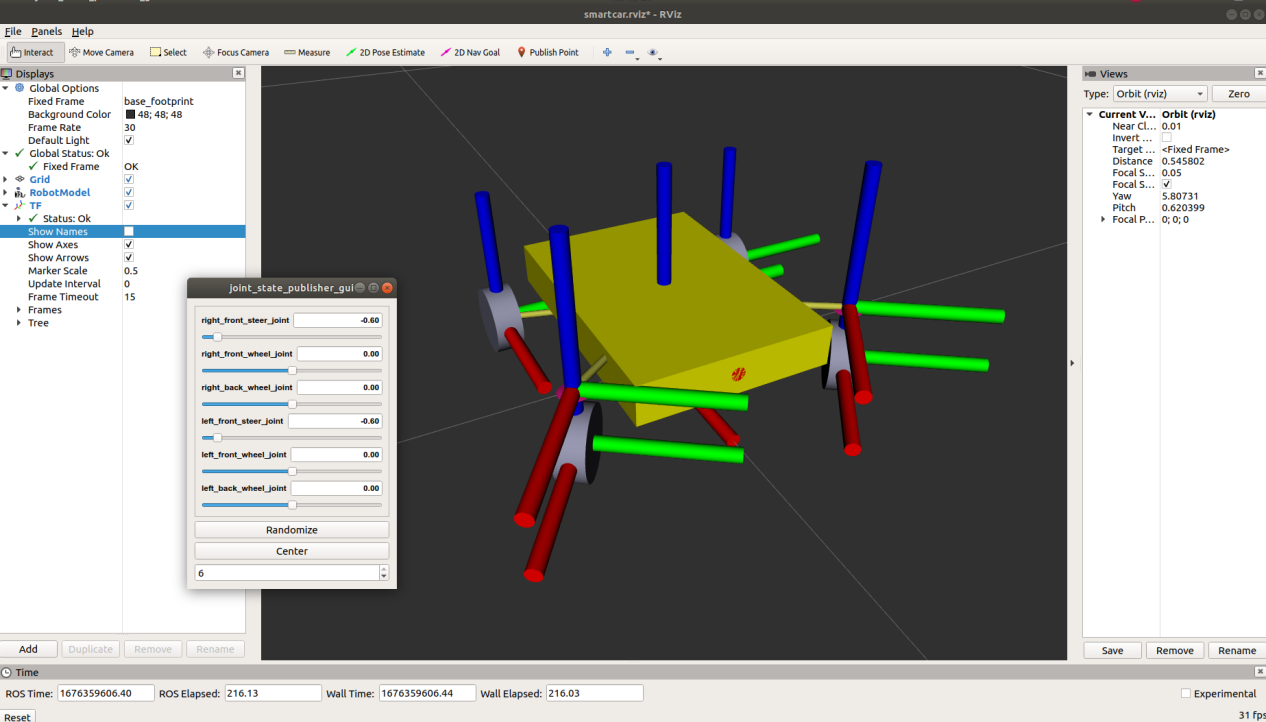
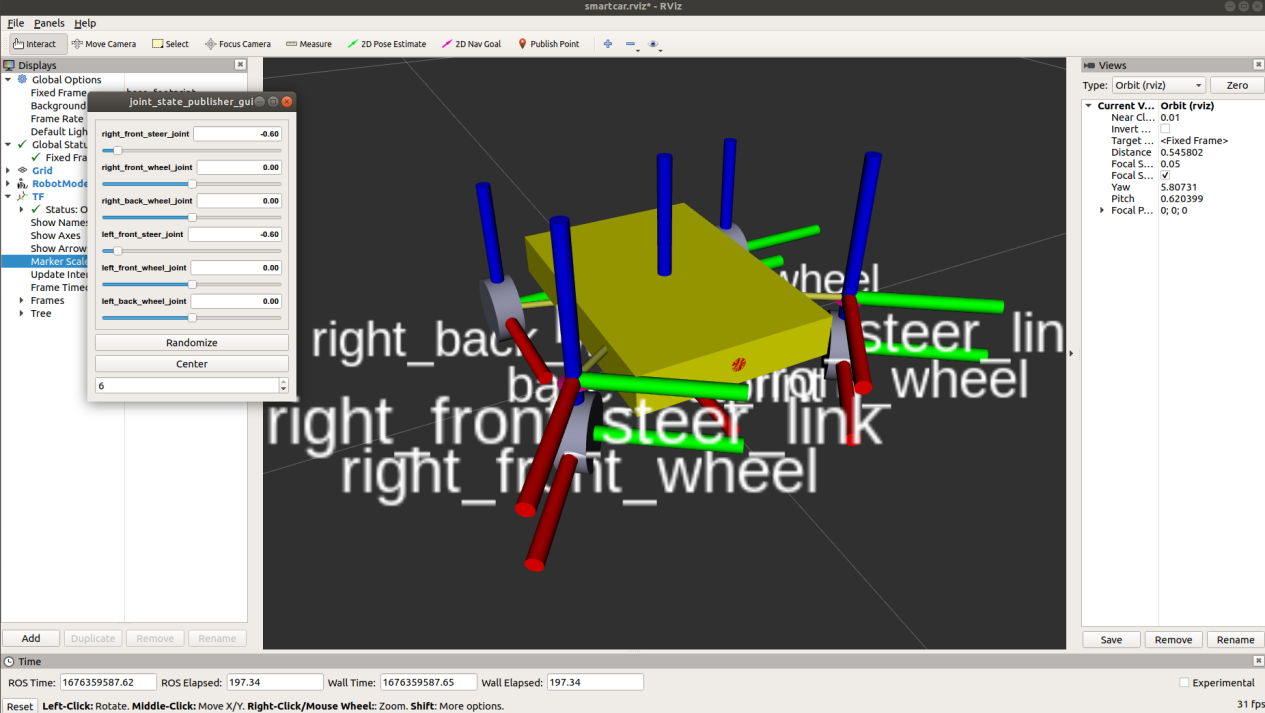
urdf文件一般放置于功能包的urdf或robot文件夹下，我们先在my\_car\_description功能包里创建urdf文件夹，再在文件夹内创建my\_car.xacro文件。文件内容详见功能包。

1. 在功能包内创建launch启动文件，能够在rviz中查看四轮车模型。

在my\_car\_description包里新建launch文件夹用来存放所有的launch文件。在文件夹内新建display.launch文件。

可通过以下命令启动：

$ roslaunch my\_car\_description display.launch



习题6

1. MoveIt!的核心节点是 move\_group 。
2. 逆运动学求解算法以ROS插件的形式集成到MoveIt!中，可使用KDL、TRAC-IK、IKFAST等求解器或编写自己的求解器插件。
3. SRDF文件的常用标签有哪些？

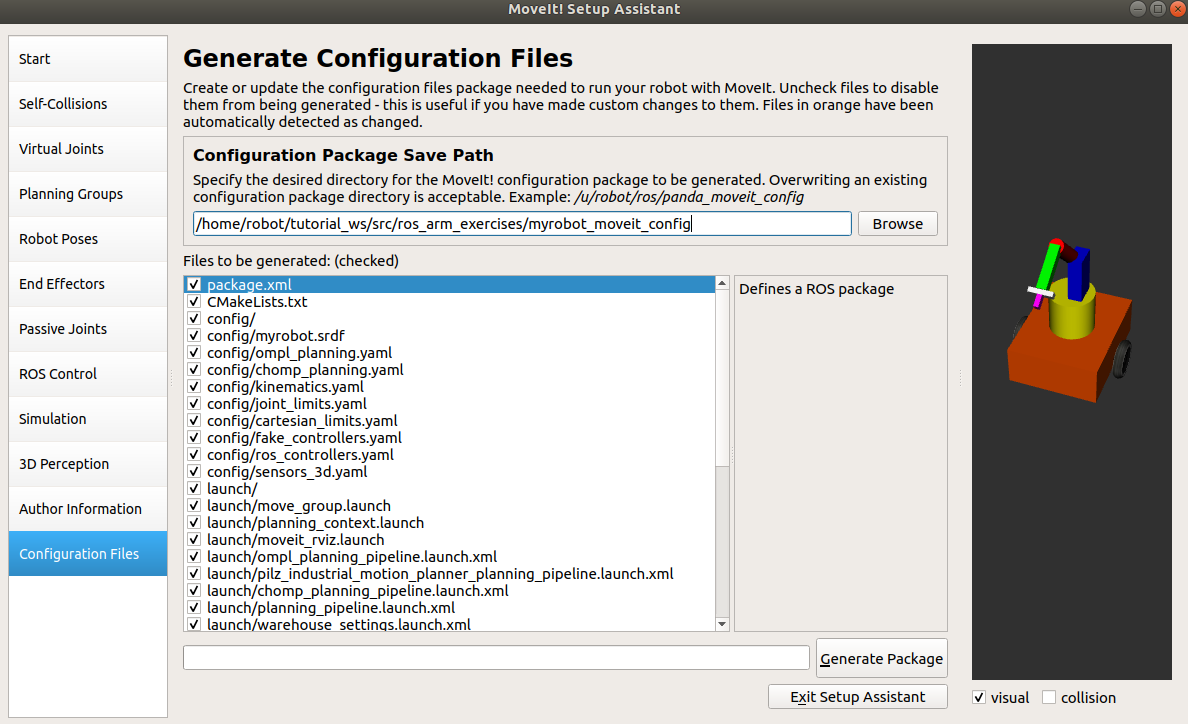
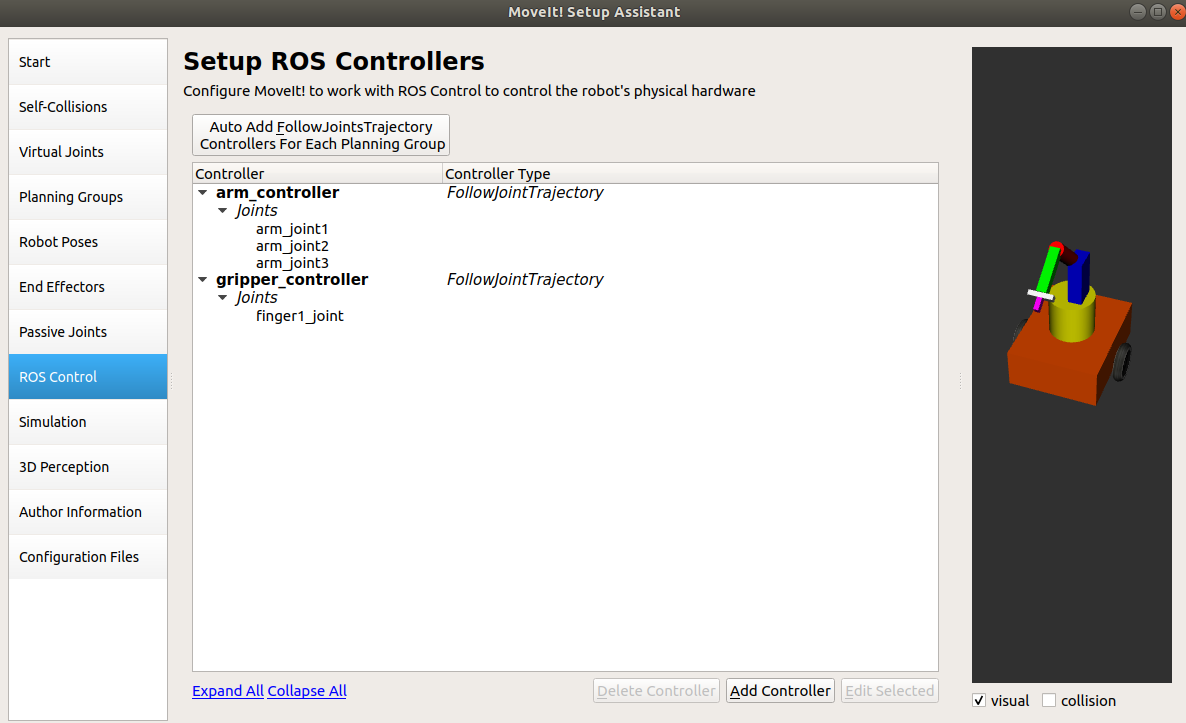
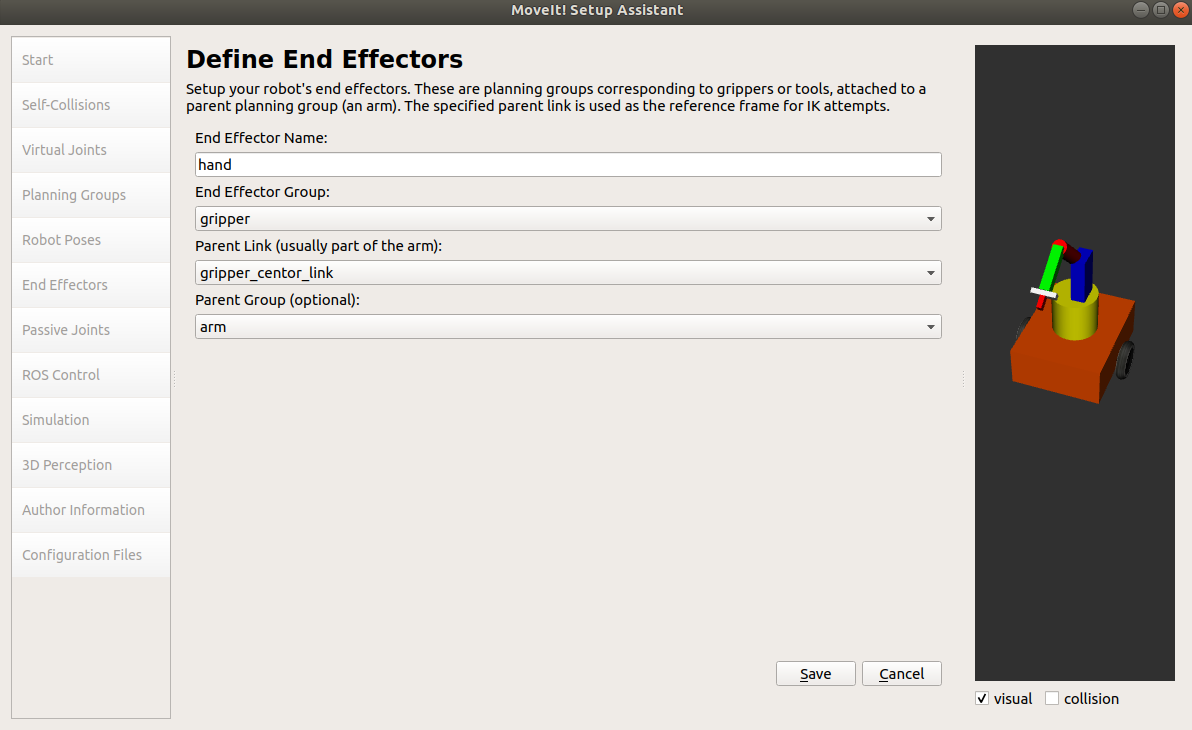
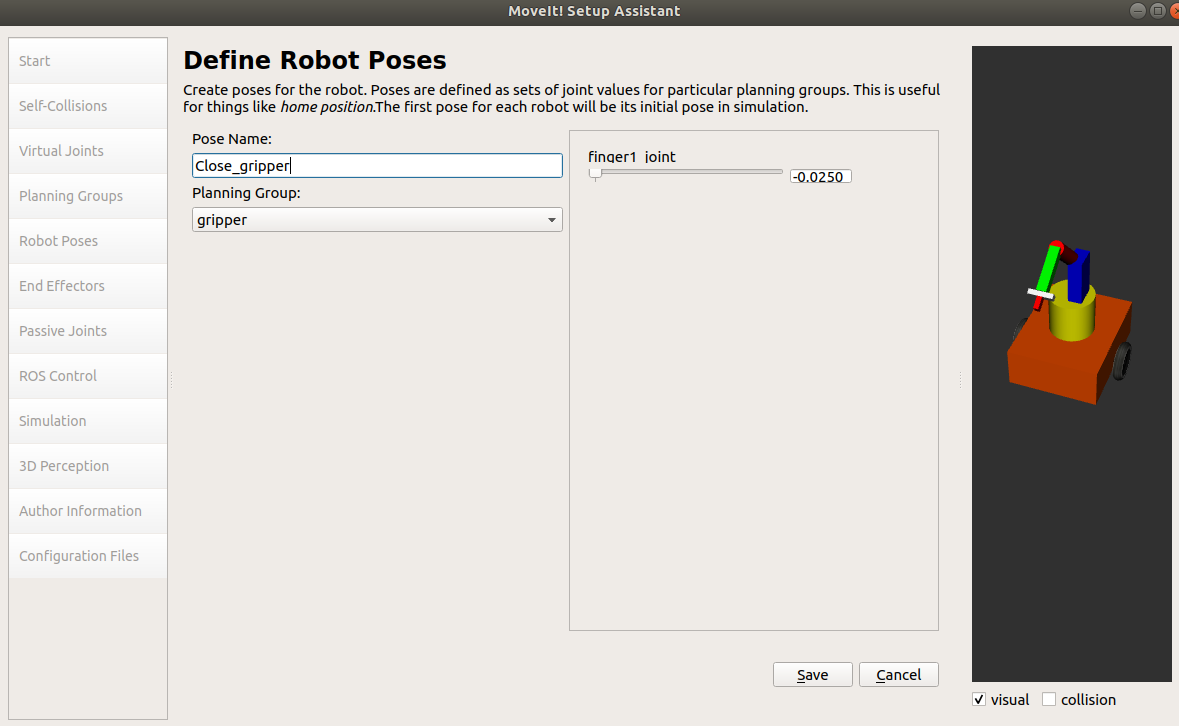
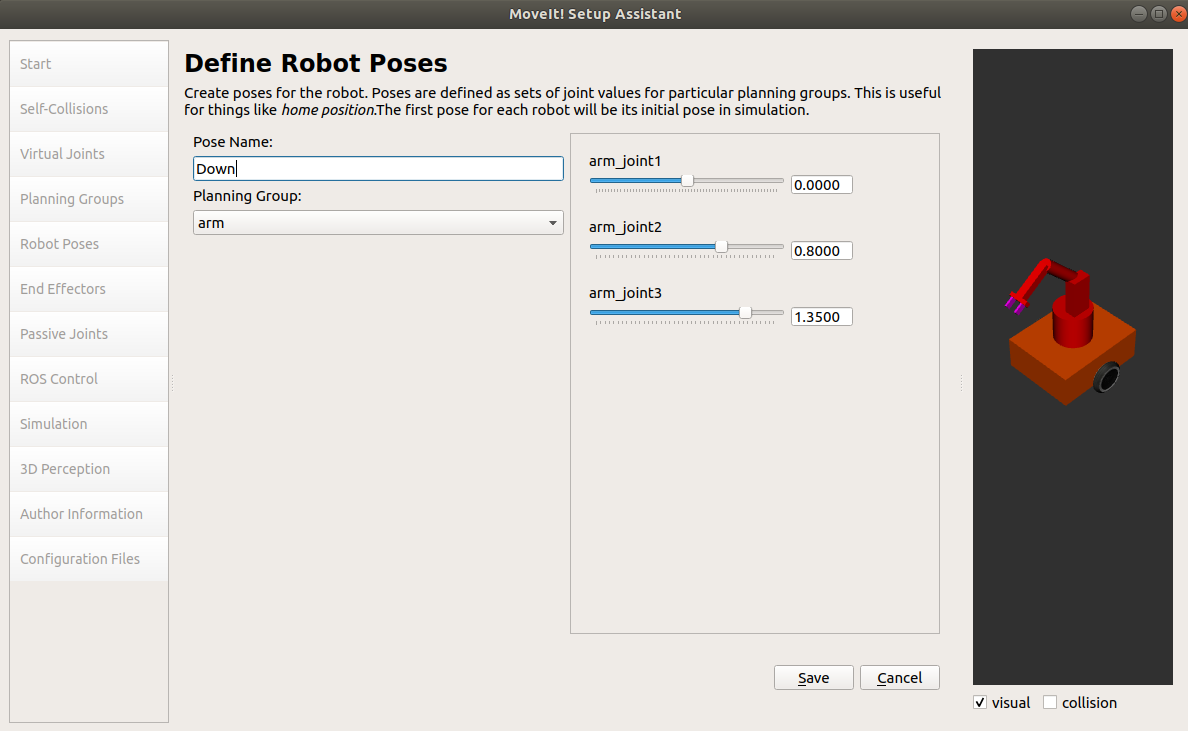
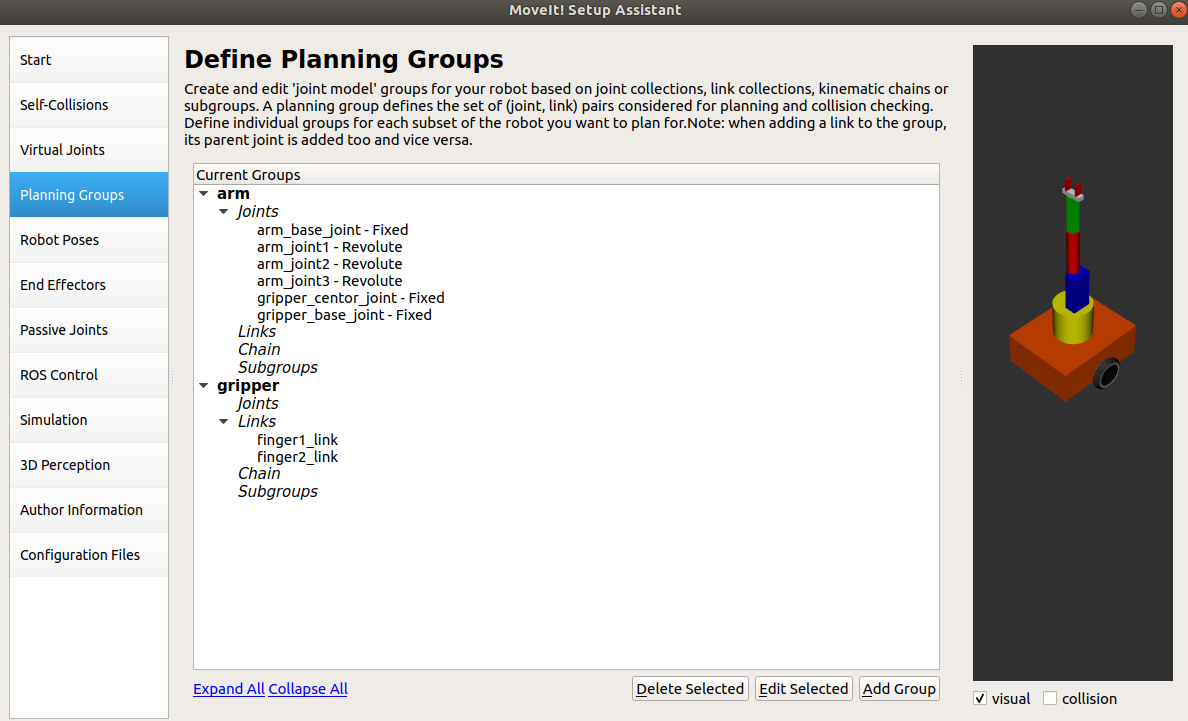
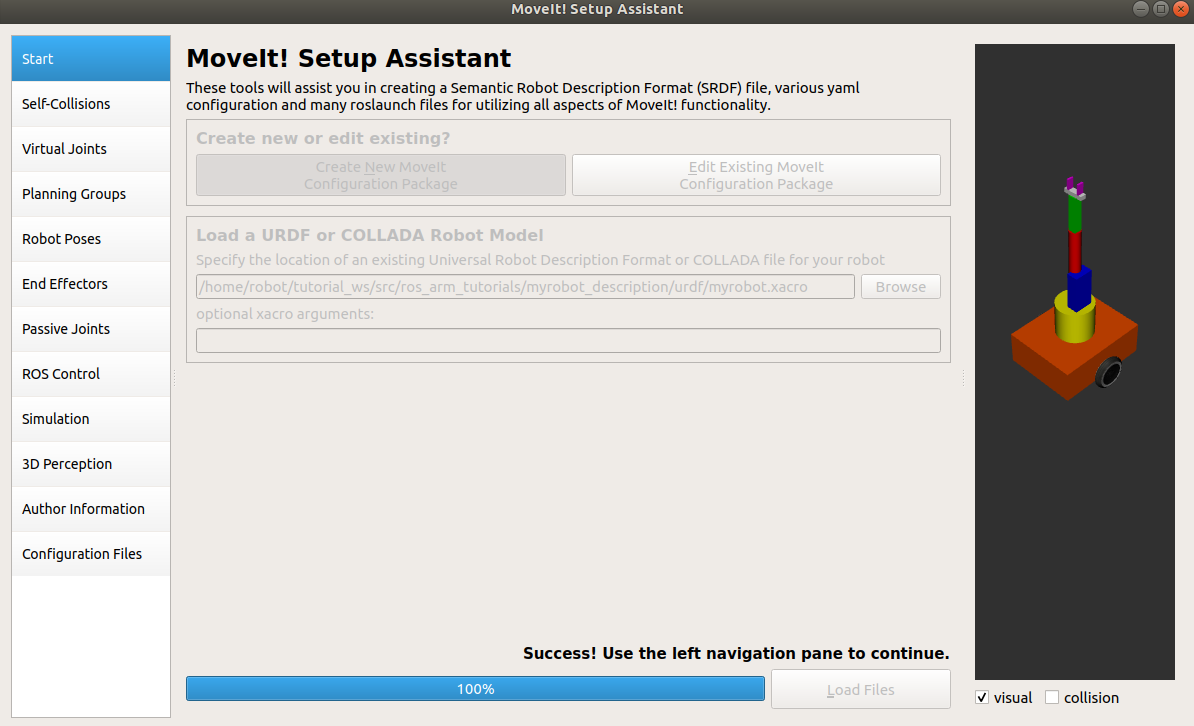
答：根标签为<robot>， <robot>根标签下包含<group>、<group\_state>、<end\_effector>、<virtual\_joint>、<disable\_collisions>标签。

1. 综合实践
2. 参考6.2节，使用配置助手对第5章搭建的myrobot.xacro机器人模型进行配置，生成myrobot\_moveit\_config功能包。

首先在终端输入以下命令启动配置助手（Setup Assistant）：

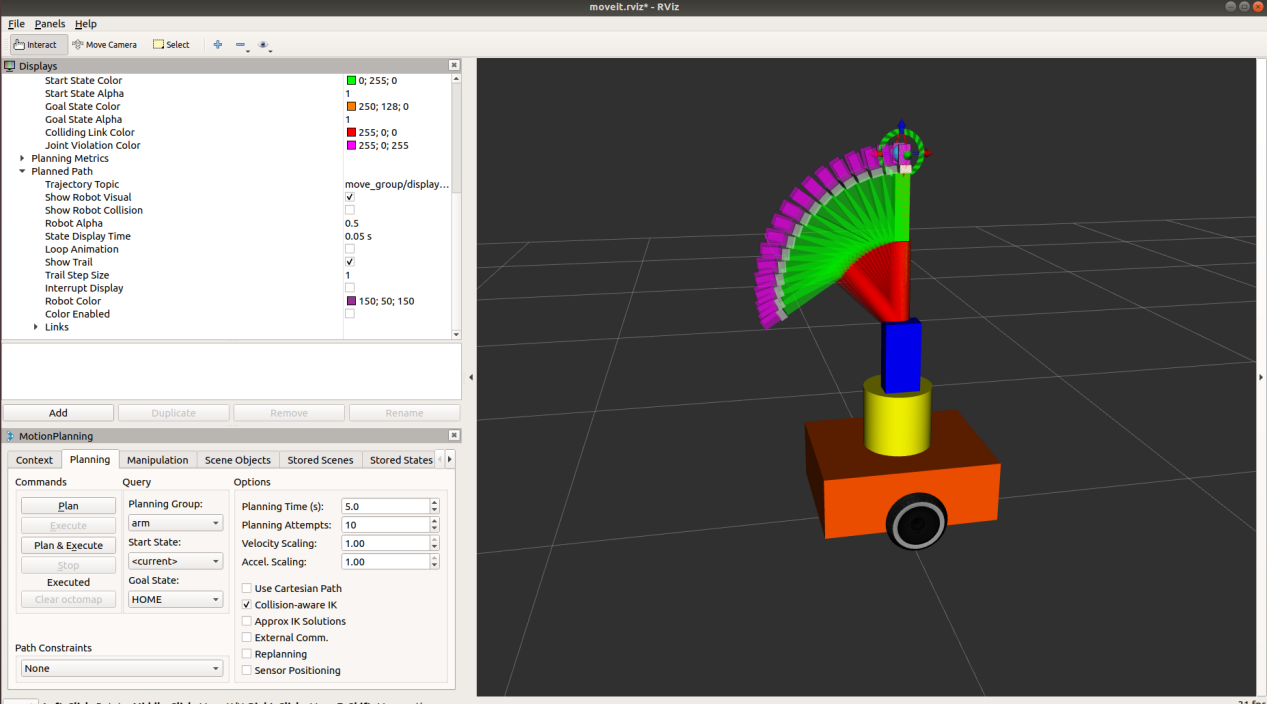
$ roslaunch moveit\_setup\_assistant setup\_assistant.launch

部分设置如下图：



1. 启动myrobot\_moveit\_config里的demo.launch，参考书6.3节使用MotionPlanning插件测试机械臂的运动规划和控制执行。

$ roslaunch myrobot\_moveit\_config demo.launch



1. 参考书6.6节，使用命令行工具与myrobot进行交互。

习题7

1.综合实践

1）创建exercise\_seven功能包；

2）在功能包内编写pose\_plan节点，在节点内实现以下功能：

①让XBot-Arm机械臂规划并运动到target\_pose目标处。target\_pose的位置xyz为（0.3，-0.3, 0.3），姿态用RPY欧拉角表示为（0,0,-pi/4）。

②让机械臂规划并运动到target\_joint\_positions目标处。 target\_joint\_positions 中六个关节的位置为[-0.9, -1.0, 0.2, 0.9, -0.76, 1.5]。

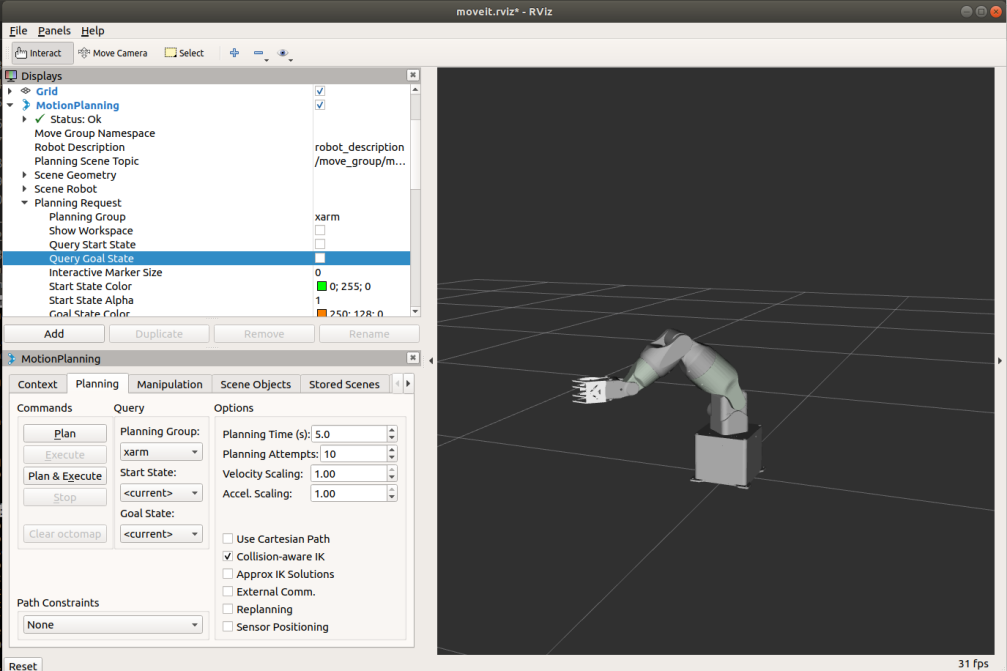
③控制机械臂回到初始位置。

完整代码详见仓库功能包exercise\_seven。

$ roslaunch xarm\_moveit\_config demo.launch

$ rosrun exercise\_seven pose\_plan.py

$ rosrun exercise\_seven pose\_plan



3）在功能包内编写pick\_place节点，在节点内实现以下功能：

①在规划场景中添加桌面table和目标物体box。目标物体为边长6厘米的正方体，位于桌面上方，位置xyz为(0.4,0,0.03)，姿态用RPY欧拉角表示为（0,0,0）。

②让机械臂对目标box进行抓取，抓取时，手爪竖直向下。

③自行设计放置位置，抓取成功后将目标box放置到放置点。

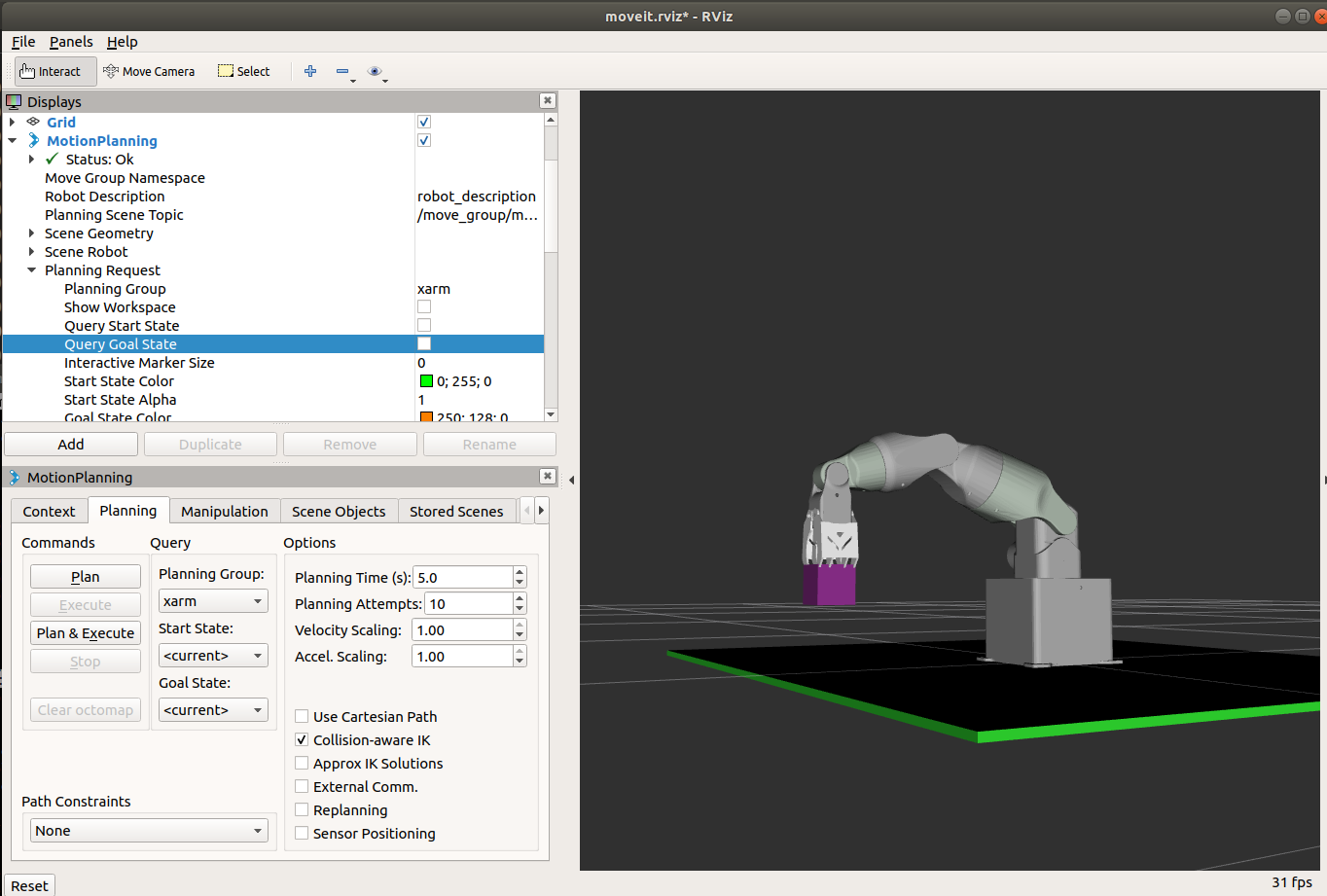
④控制机械臂回到初始位置并删除规划场景中的物体。

完整代码详见仓库功能包exercise\_seven。

$ roslaunch xarm\_moveit\_config demo.launch

$ rosrun exercise\_seven pick\_place.py

$ rosrun exercise\_seven pick\_place



习题8

1. sensor\_msgs/Image消息用来存储未压缩的图像信息，sensor\_msgs/CompressedImage消息用来存储压缩图像信息，sensor\_msgs/PointCloud2消息用来存储点云信息。
2. rqt工具箱中的rqt\_image\_view可以用来查看图像话题的信息。
3. cv\_bridge功能包能够实现OpenCV图像和ROS图像消息格式之间的转换。

习题9

1. 简要概括视觉抓取应用中用到的关键技术。

相机的选型以及内参标定、物体检测、目标识别定位、手眼标定、抓取姿态分析、机械臂运动规划和避障。

2.ar\_track\_alvar功能包里的createMarker节点可以创建不同大小、ID数据的AR标签。

3.ar\_track\_alvar节点将AR标签识别结果发布到了话题/ar\_pose\_marker上，话题的消息类型为ar\_track\_alvar\_msgs/AlvarMarkers。

4.easy\_handeye功能包提供了两种标定方式：眼在手外和眼在手上。