Jaka_ros 使用说明

版本：V2.0

时间：2022.7.1

上海节卡机器人科技有限公司 Shanghai JAKA Robotics Ltd
电话 Tel ：+400 006 2665 | 网站 Web:www.jaka.com
上海：上海市闵行区剑川路 610 号 33-35 幢 | Building 33-35, No.610 Jianchuan Road, Minhang District, Shanghai
常州：江苏省常州市武进国家高新区武宜南路 377 号 10 号楼 | Building 10, No.377 South Wuyi Rd, Changzhou, Jiangsu

文档信息

应用领域			
文件编号		发布日期	
起 草 人		起草日期	2021.10.26
复 审 人		复审日期	
批 准 人		批准日期	

版本记录

版本编号	版本日期	修改者	说明

产权说明

Jaka_ros 使用说明

1 准备工作

1.1 安装 Ubuntu（以 x86 架构 Ubuntu 18.04 为例）

Ubuntu18 安装教程参考网址：

https://blog.csdn.net/baidu_36602427/article/details/86548203

1.2 安装 ROS（与 Ubuntu 18.04 版本对应为 Melodic）

Ubuntu18 安装 ROS 教程参考网址：

https://blog.csdn.net/qq_41450811/article/details/99079041/

（Ubuntu18.04 安装 ROS Melodic 详细过程）

1.3 注意事项：

（1）目前只支持 Ros1 版本。

（2）目前只适用于 X86_64 架构下。

（3）Ros 包是在 Ubuntu18.04 版本下测试的。如果在使用过程中，出现一些报错，建议将工作空间（jaka_robot）中的 build 文件夹和 devel 文件夹删除，重新编译。

2 关于 Jaka_Ros 驱动接口

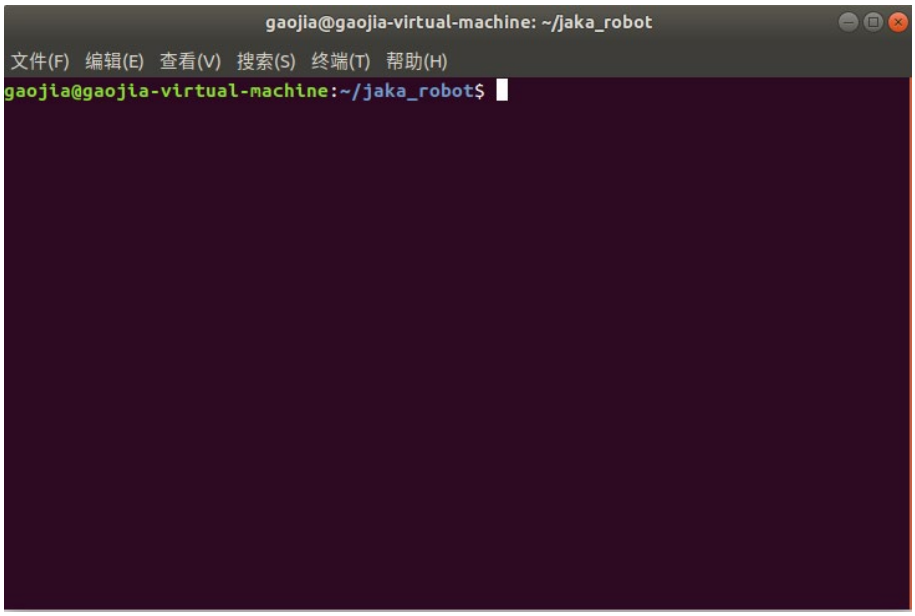
2.1 具体 Ros 驱动接口通讯协议详见《jaka_driver_interface》

2.2 测试机器人各功能服务的参考

（1）修改功能包（jaka_driver）的 launch 文件夹下的 robot_start_launch.launch 文件中的机器人 IP 地址(与实际机器人 IP 地址保持一致)，如图：

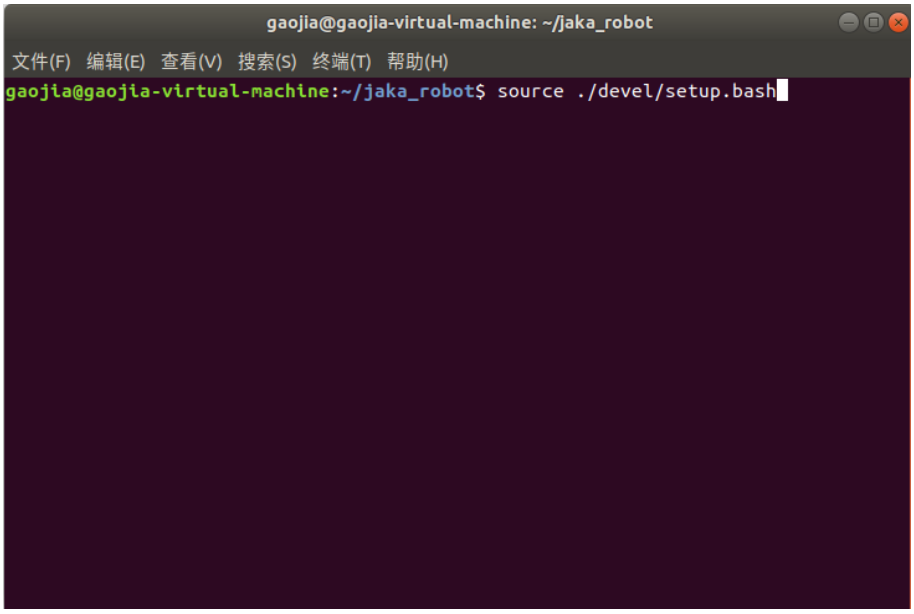
```xml
<launch>
    <param name="ip" value="192.168.126.128" type="str"/>
    <node pkg="jaka_driver" type="jaka_driver" name="jaka_driver" output="screen" />

</launch>
```

（2）在工作空间（jaka_robot）右键打开一个终端。



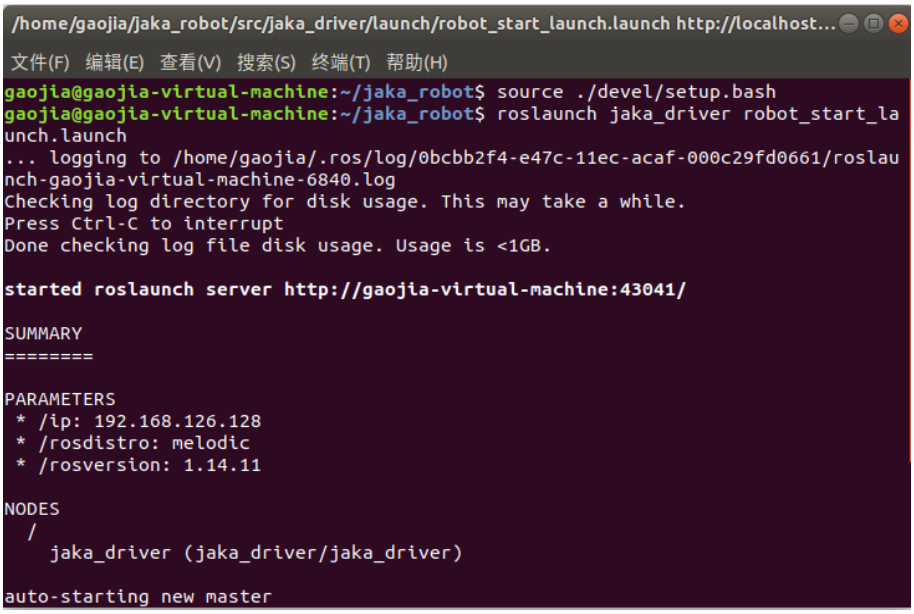
（3）从 ’setup.bash’ 文件添加环境变量。

```
$ source devel/setup.bash
```

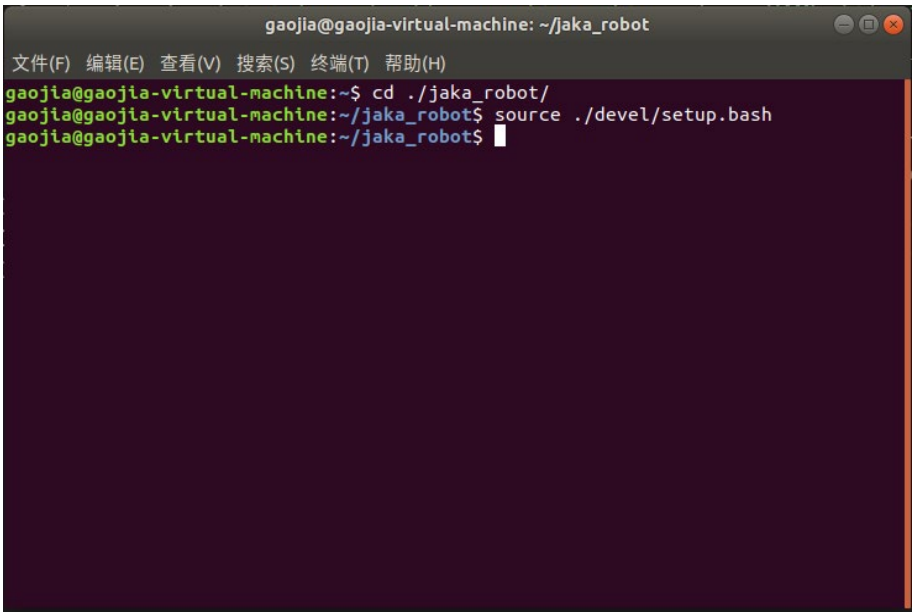


（4）启动 robot_start_launch.launch 文件

```
$ roslaunch jaka_driver robot_start_launch.launch
```

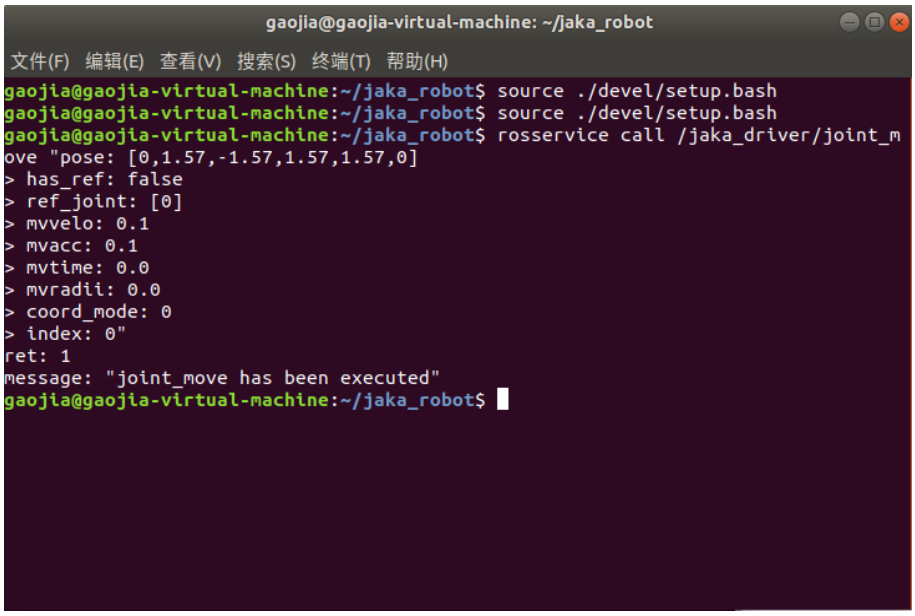


（5）在工作空间（jaka_robot）右键新打开一个终端，并添加环境变量。



（6）**关节运动：**使用 rosservice call /jaka_driver/joint_move 并按要求输入参数，控制机器人运动

```
$ rosservice call /jaka_driver/joint_move "pose: [0,1.57,-1.57,1.57,1.57,0]
  has_ref: false
  ref_joint: [0]
  mvvelo: 0.1
  mvacc: 0.1
  mvtime: 0.0
  mvradii: 0.0
  coord_mode: 0
  index: 0"
```



（7）**直线运动：**使用 rosservice call /jaka_driver/linear_move 并按要求输入参数，控制机器人运动

$ rosservice call /jaka_driver/linear_move "pose: [417,115,373,-2.2214,2.2214,0]

has_ref: false

ref_joint: [0]

mvvelo: 100

mvacc: 100

mvtime: 0.0

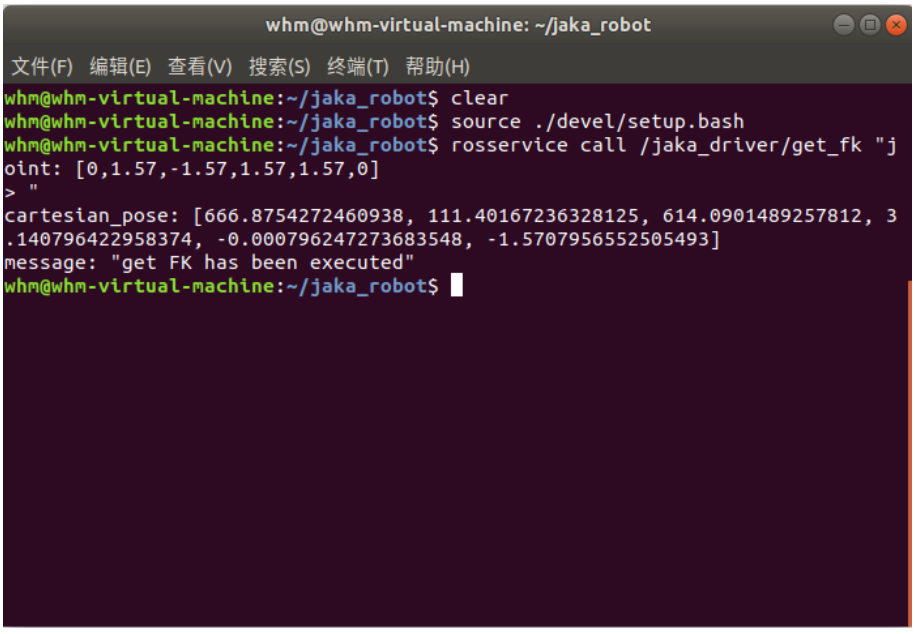
mvradii: 0.0

coord_mode: 0

index: 0"

（7）**求解机械臂正解：**使用 rosservice call /jaka_driver/get_fk 并按要求输入参数，求取正解。

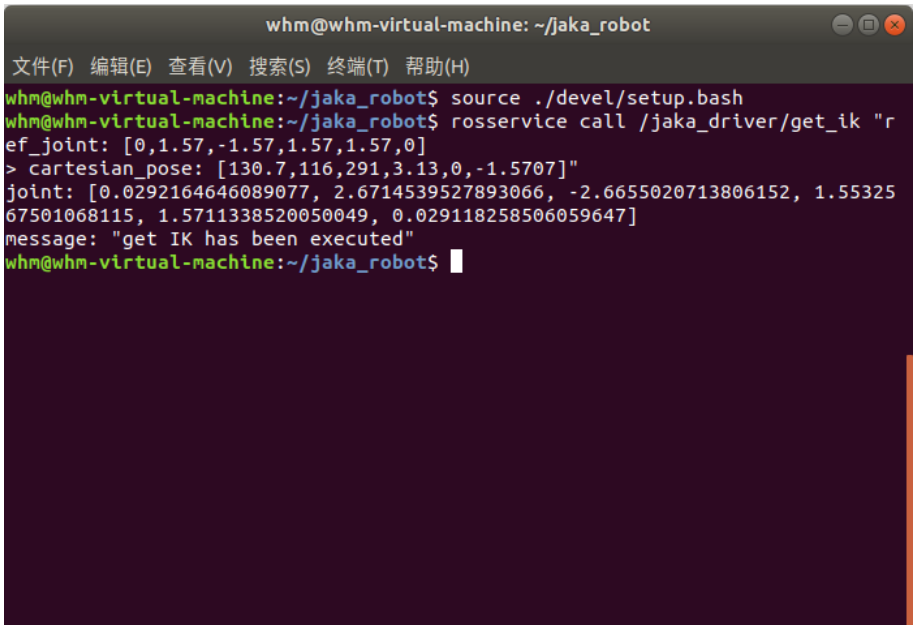
$ rosservice call /jaka_driver/get_fk "joint: [0,1.57,-1.57,1.57,1.57,0]"



（8）**求解机械臂逆解：**使用 rosservice call /jaka_driver/get_ik 并按要求输入参数，求取逆解。

$ rosservice call /jaka_driver/get_ik "ref_joint: [0,1.57,-1.57,1.57,1.57,0]

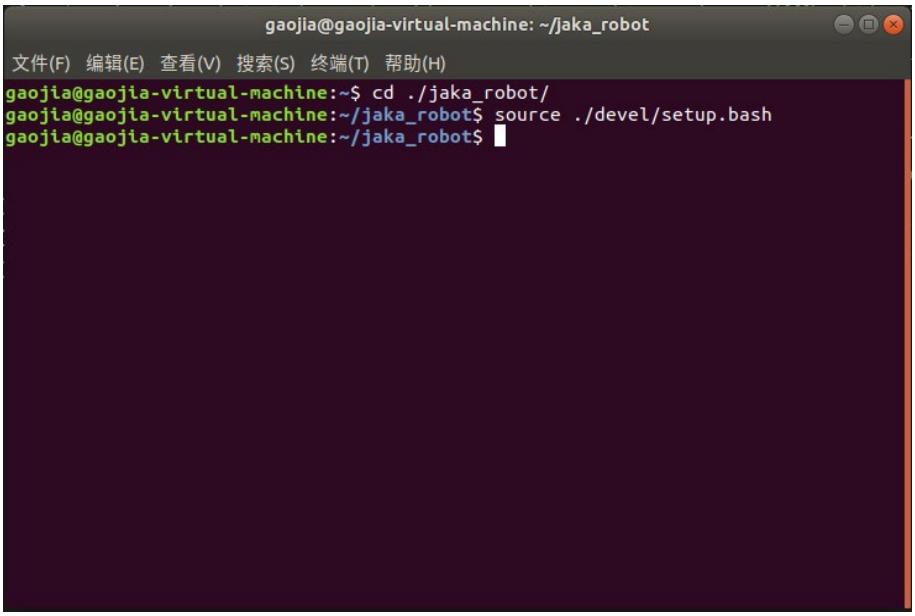
cartesian_pose: [130.7,116,291,3.13,0,-1.5707]"



3 关于 Moveit

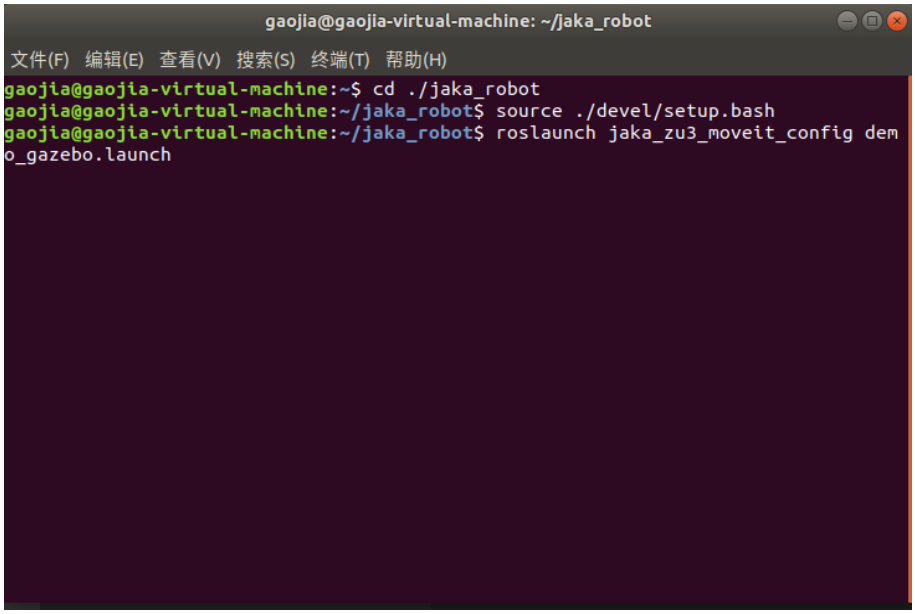
3.1 Moveit 和 Gazebo 的联合使用

（1）在工作空间（jaka_robot）右键新打开一个终端，并添加环境变量。

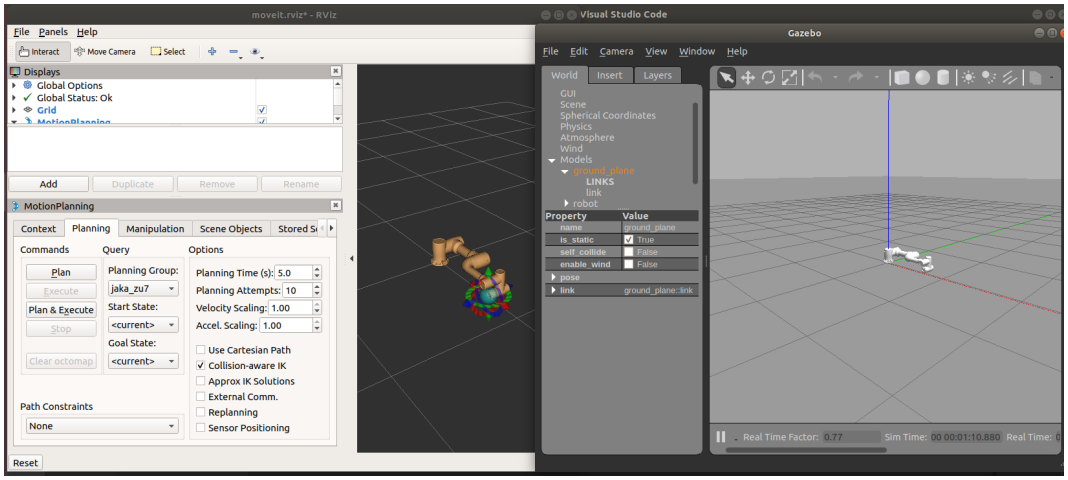


（2）启动 demo_gazebo.launch 文件

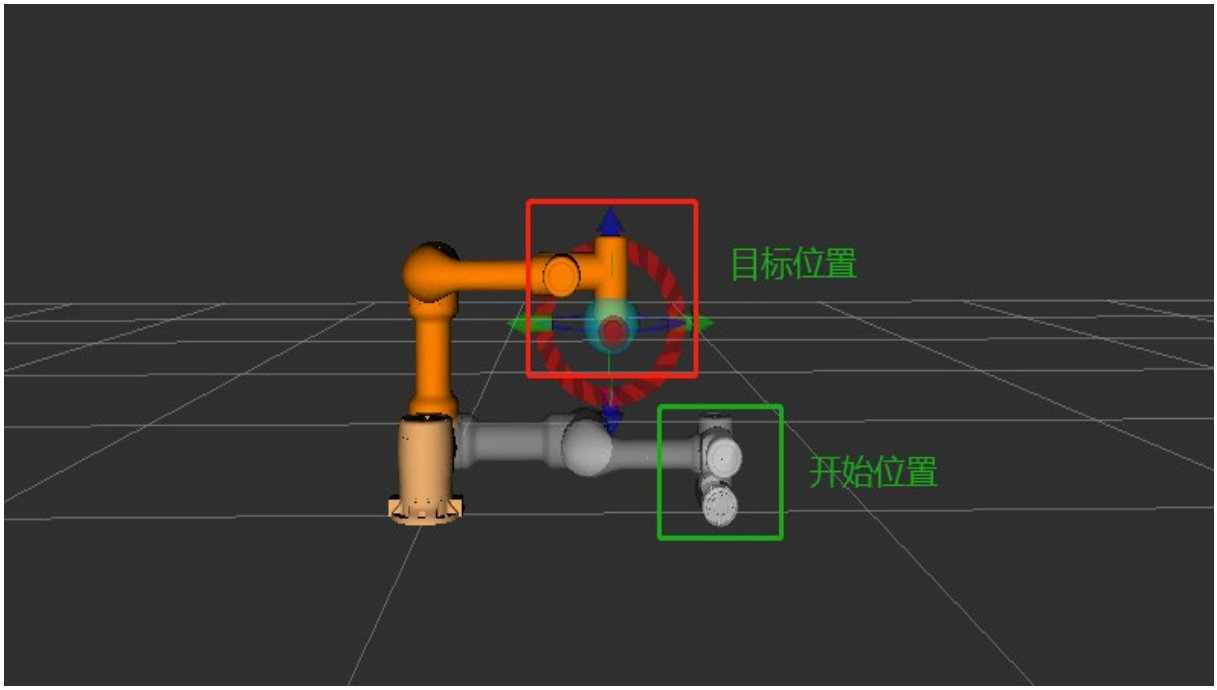
```
$ roslaunch jaka_zu7_moveit_config demo_gazebo.launch
```

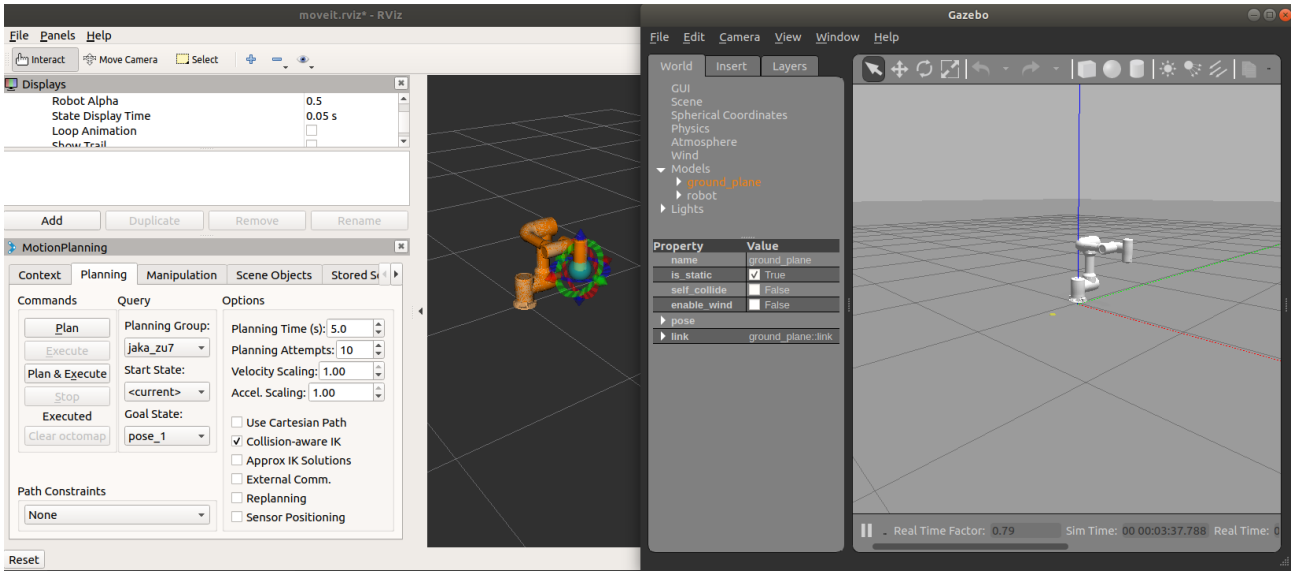


生成如下界面



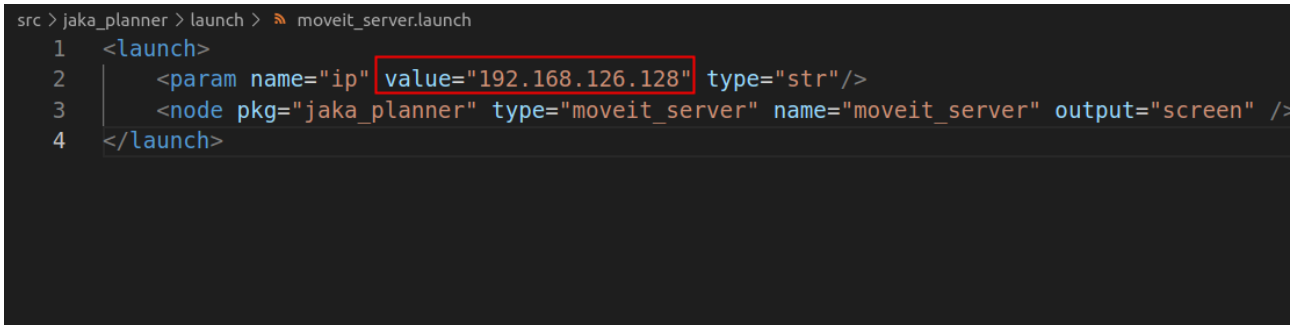
（3）从 RVIZ 界面的“Goal state”选择一个目标位置，点击“Plan & Execute”，RVIZ 界面会显示机器人的轨迹，并驱动 Gazebo 中仿真机器人到设置的目标位置



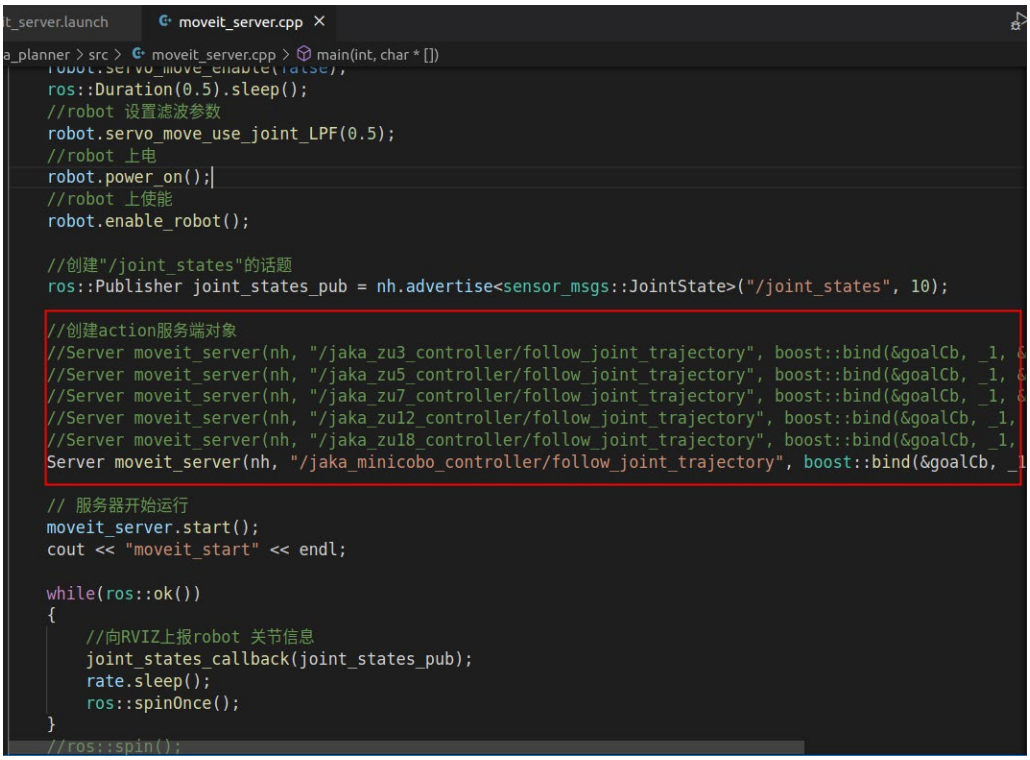


3.2 Moveit 和真实机器人的联合使用

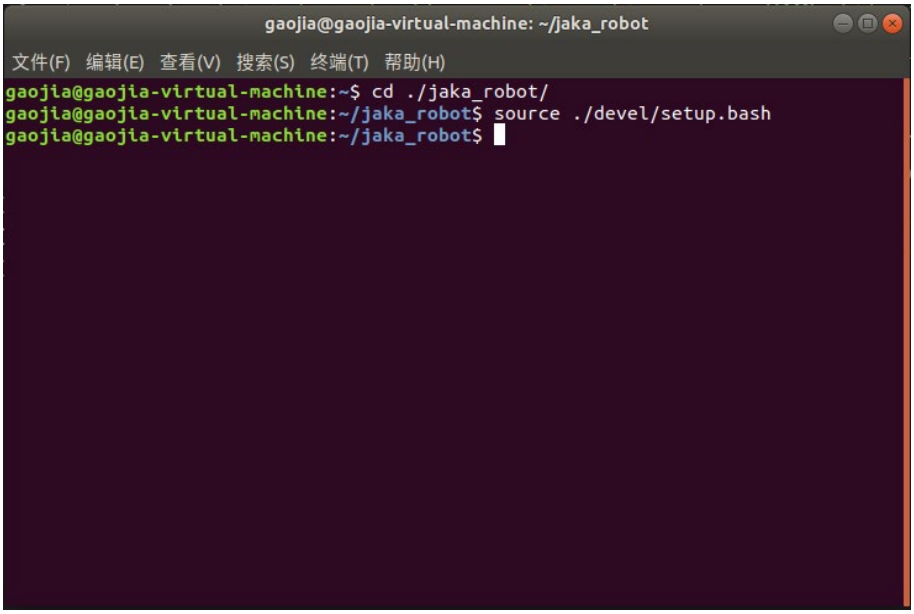
（1）修改功能包（jaka_planner）的 launch 文件夹下的 moveit_server.launch 文件中的机器人 IP 地址(与实际机器人 IP 地址保持一致)，如图：



（2）注意功能包（jaka_planner）的 src 文件夹下的 moveit_server.cpp 文件中的创建 Moveit 服务端对象与客户端一致。例如：使用 jaka_zu7_moveit_config 时，要对应 Server moveit_server(nh,"/jaka_zu7_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &moveit_server), false)。

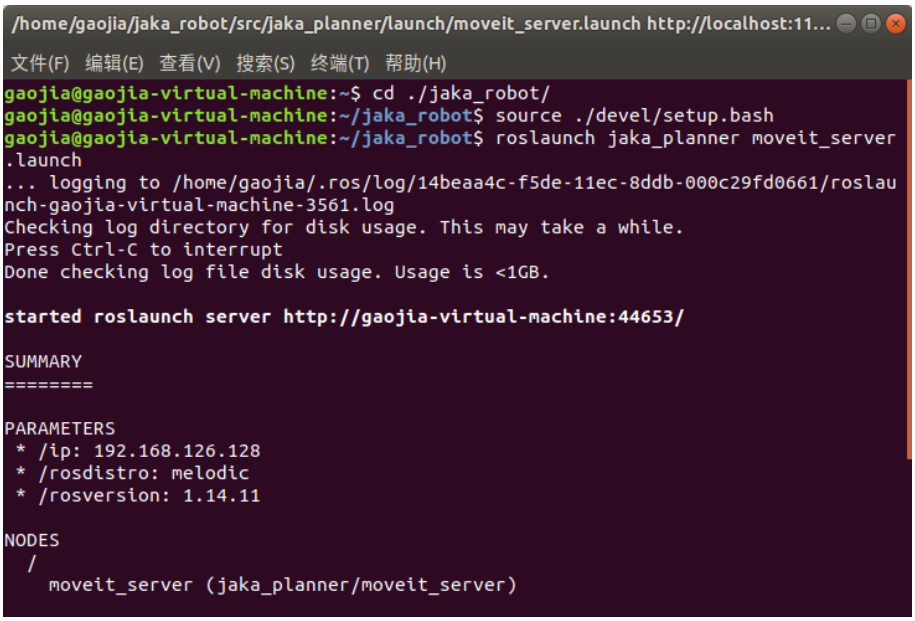


（3）在工作空间（jaka_robot）右键新打开一个终端，并添加环境变量。



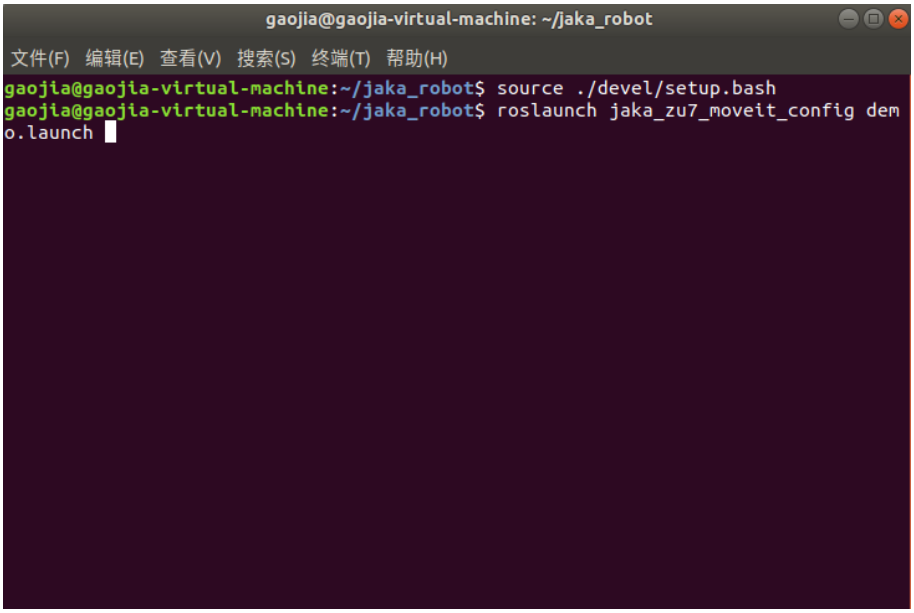
（4）启动 moveit_server.launch 文件(Moveit 的服务器端)

```
$ roslaunch jaka_planner moveit_server.launch
```

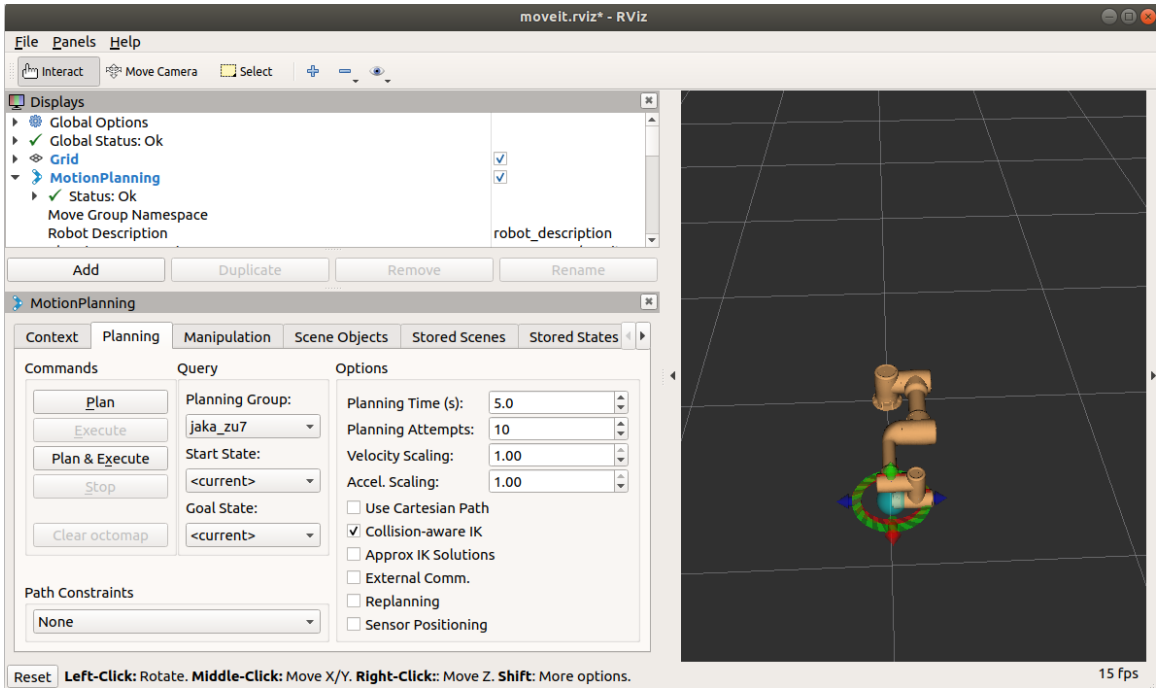


（5）启动 demo.launch 文件

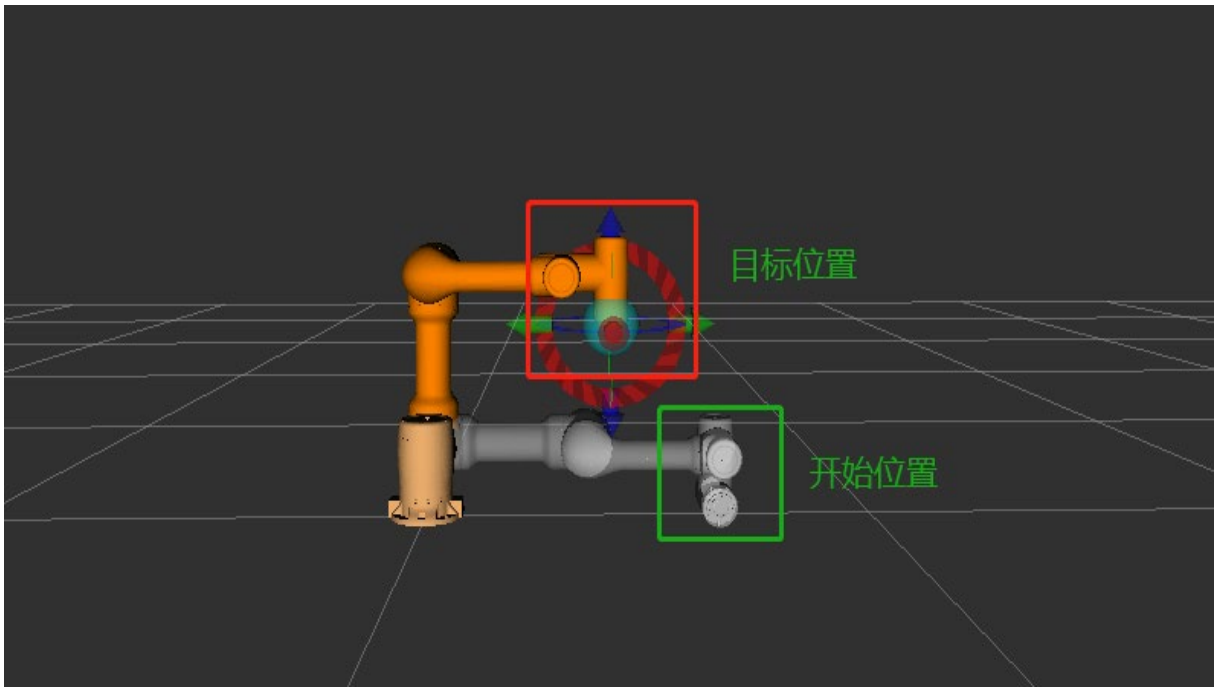
```
$ roslaunch jaka_zu7_moveit_config demo.launch
```

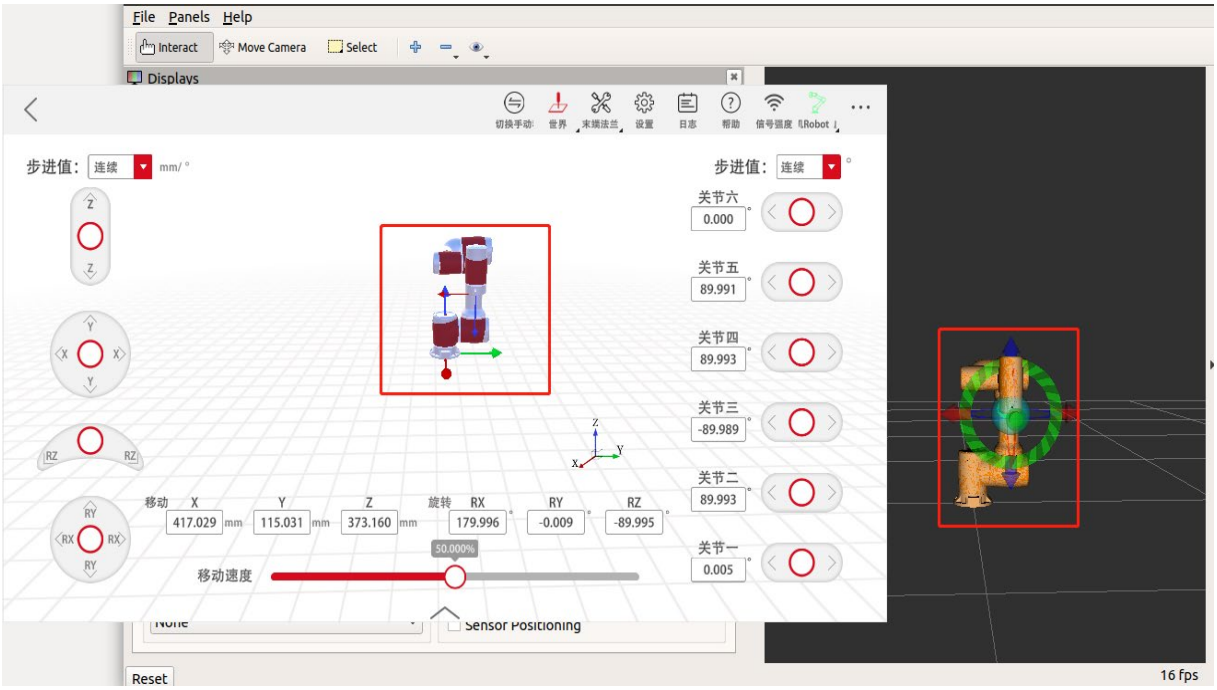


生成如下界面



（6）从 RVIZ 界面的“Goal state”选择一个目标位置，点击“Plan & Execute”，RVIZ 界面会显示机器人的轨迹，并驱动实体机器人到设置的目标位置





注意：如果需要代码端控制，请自行参考 Moveit 有关 C++或者 Python 的 API