

Jaka_ros 使用说明

版本：V2.0

时间：2022.7.1

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

上海节卡机器人有限公司 版权所有。

上海节卡机器人有限公司对本文档中介绍的产品所包含的相关技术拥有知识产权。

本文档及相关产品按照限制其使用、复制、分发和反编译的许可证进行分发。未经上海节卡机器人有限公司事先书面授权，不得以任何方式、任何形式复制本产品或本文档的任何部分。

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

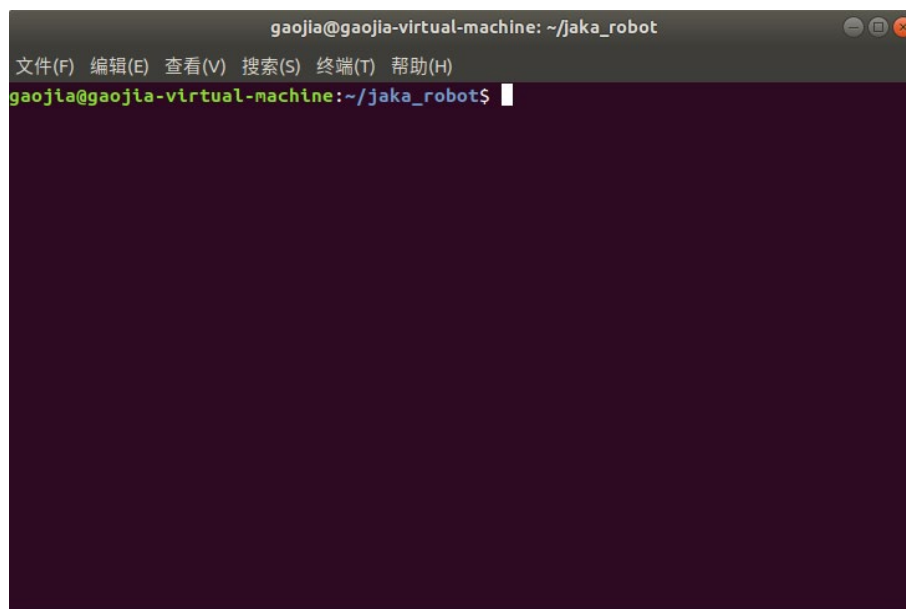
2.1 具体 Ros 驱动接口通讯协议详见《jaka_driver_interface》

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:

```
1 <launch>
2   <param name="ip" value="192.168.126.128" type="str"/>
3   <node pkg="jaka_driver" type="jaka_driver" name="jaka_driver" output="screen" />
4
5 </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
```

```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
```

(4) 启动 robot_start_launch.launch 文件

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

```
/home/gaojia/jaka_robot/src/jaka_driver/launch/robot_start_launch.launch http://localhost...
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ roslaunch jaka_driver robot_start_la
unch.launch
... logging to /home/gaojia/.ros/log/0bcbb2f4-e47c-11ec-acaf-000c29fd0661/roslau
nch-gaojia-virtual-machine-6840.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://gaojia-virtual-machine:43041/

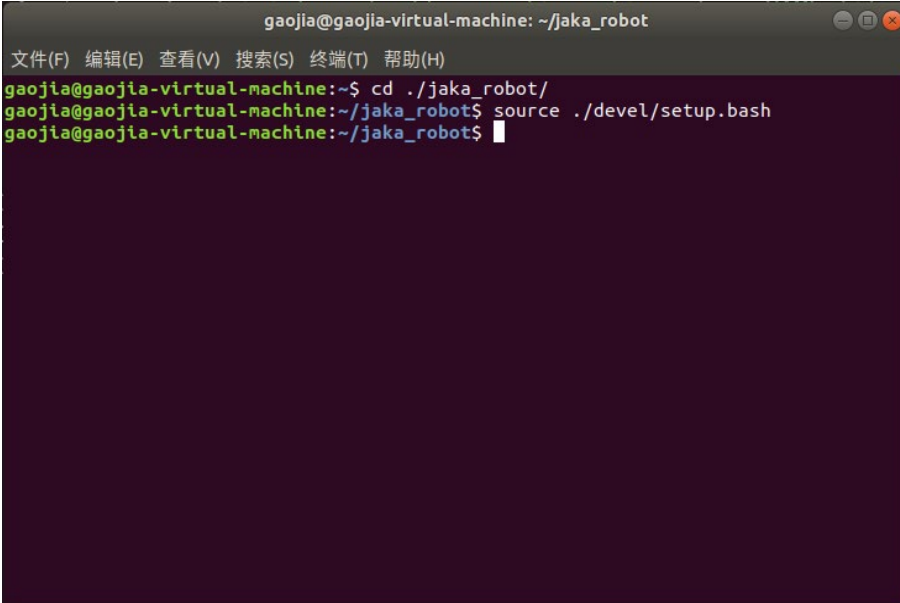
SUMMARY
=====

PARAMETERS
* /ip: 192.168.126.128
* /rostdistro: melodic
* /rosversion: 1.14.11

NODES
/
  jaka_driver (jaka_driver/jaka_driver)

auto-starting new master
```

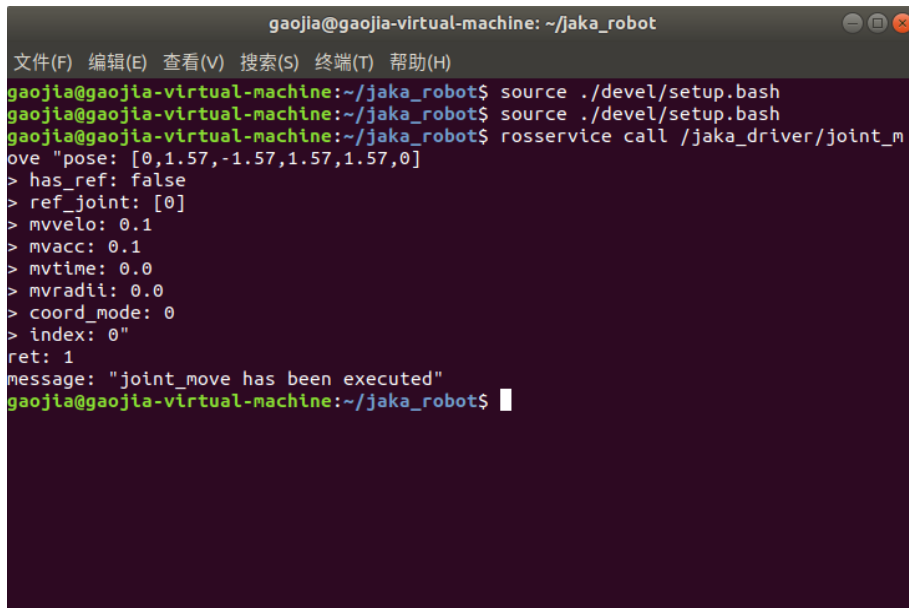
(5) 在工作空间（jaka_robot）右键新打开一个终端，并添加环境变量。



```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~$ cd ./jaka_robot/
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$
```

(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"
```



```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ 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"
ret: 1
message: "joint_move has been executed"
gaojia@gaojia-virtual-machine:~/jaka_robot$
```

(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"
```



```
whm@whm-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whm@whm-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
whm@whm-virtual-machine:~/jaka_robot$ 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"
ret: 1
message: "linear_move has been executed"
whm@whm-virtual-machine:~/jaka_robot$
```

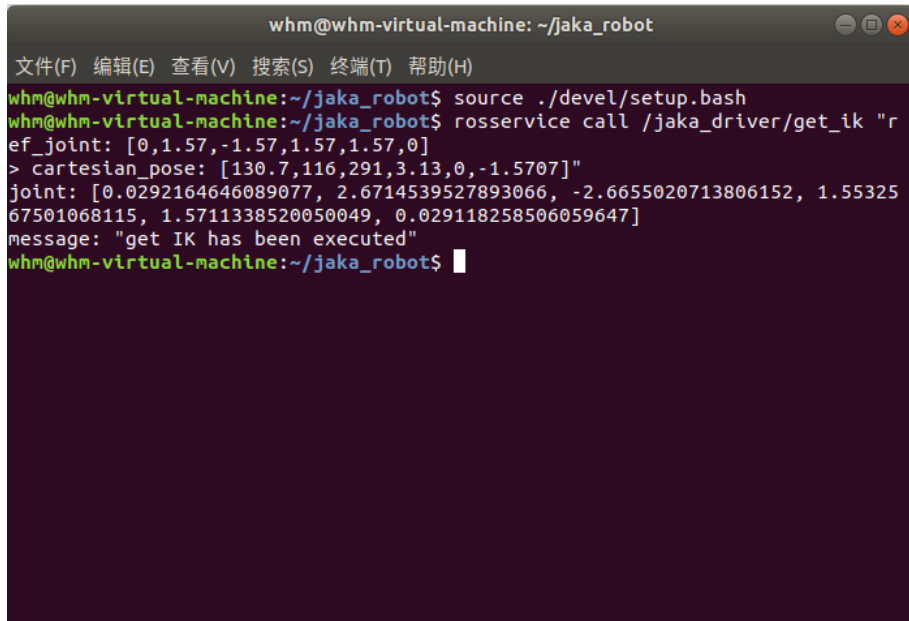
(7) 求解机械臂正解：使用 `rosservice call /jaka_driver/get_fk` 并按要求输入参数，求取正解。

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

```
whm@whm-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whm@whm-virtual-machine:~/jaka_robot$ clear
whm@whm-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
whm@whm-virtual-machine:~/jaka_robot$ rosservice call /jaka_driver/get_fk "j
oint: [0,1.57,-1.57,1.57,1.57,0]
> "
cartesian_pose: [666.8754272460938, 111.40167236328125, 614.0901489257812, 3
.140796422958374, -0.000796247273683548, -1.5707956552505493]
message: "get FK has been executed"
whm@whm-virtual-machine:~/jaka_robot$
```

(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]"
```

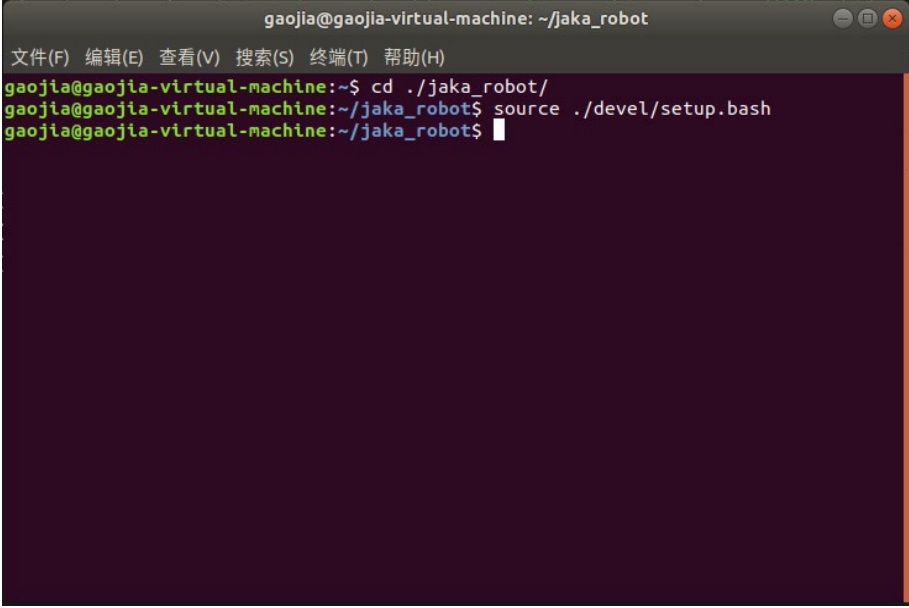
A terminal window titled 'whm@whm-virtual-machine: ~/jaka_robot' showing the execution of a ROS service call. The user runs 'source ./devel/setup.bash' and then 'rosservice call /jaka_driver/get_ik "ref_joint: [0, 1.57, -1.57, 1.57, 1.57, 0]"'. The terminal outputs the cartesian pose: '[130.7, 116, 291, 3.13, 0, -1.5707]' and the joint values: '[0.0292164646089077, 2.6714539527893066, -2.6655020713806152, 1.5532567501068115, 1.5711338520050049, 0.029118258506059647]'. A message 'get IK has been executed' is also displayed.

```
whm@whm-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
whm@whm-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
whm@whm-virtual-machine:~/jaka_robot$ rosservice call /jaka_driver/get_ik "r
ef_joint: [0,1.57,-1.57,1.57,1.57,0]
> cartesian_pose: [130.7,116,291,3.13,0,-1.5707]"
joint: [0.0292164646089077, 2.6714539527893066, -2.6655020713806152, 1.55325
67501068115, 1.5711338520050049, 0.029118258506059647]
message: "get IK has been executed"
whm@whm-virtual-machine:~/jaka_robot$
```

3 关于 Moveit

3.1 Moveit 和 Gazebo 的联合使用

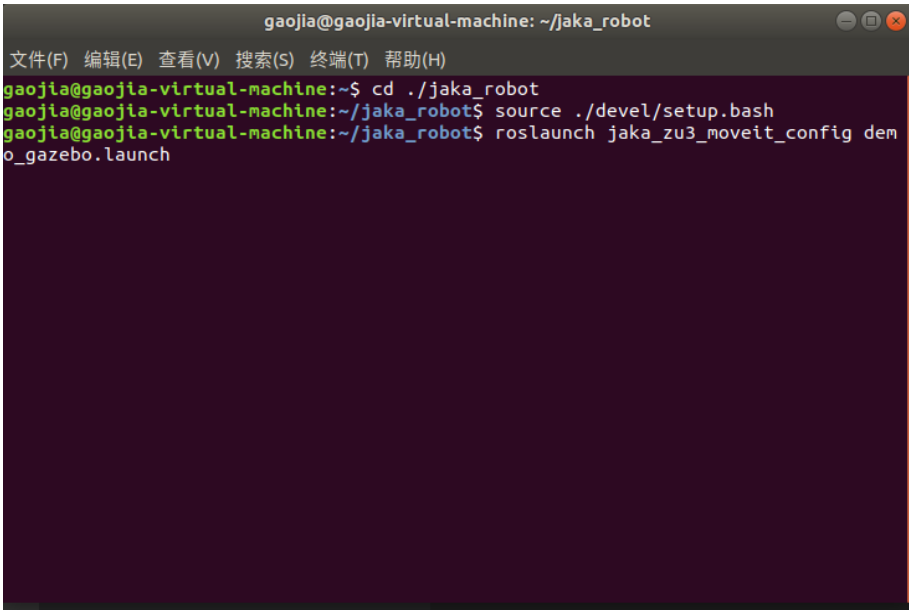
(1) 在工作空间（jaka_robot）右键新打开一个终端，并添加环境变量。



```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~$ cd ./jaka_robot/
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$
```

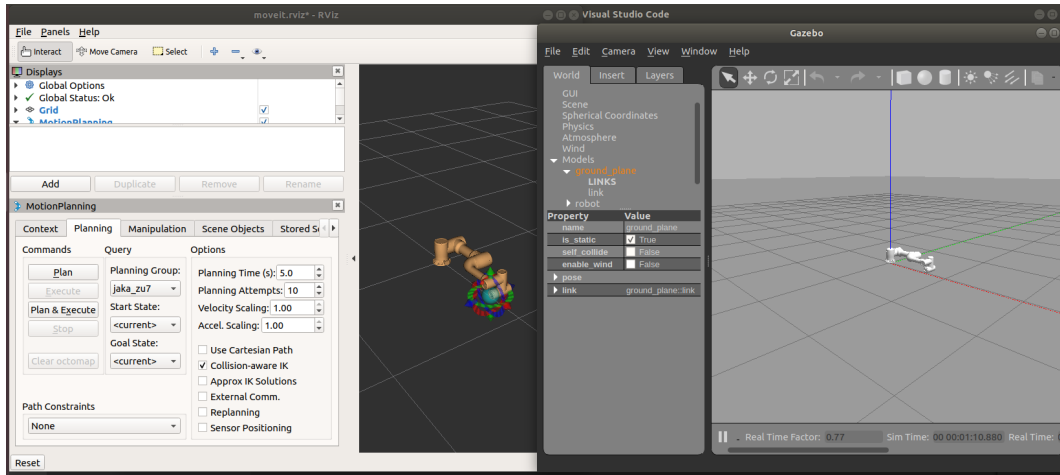
(2) 启动 demo_gazebo.launch 文件

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

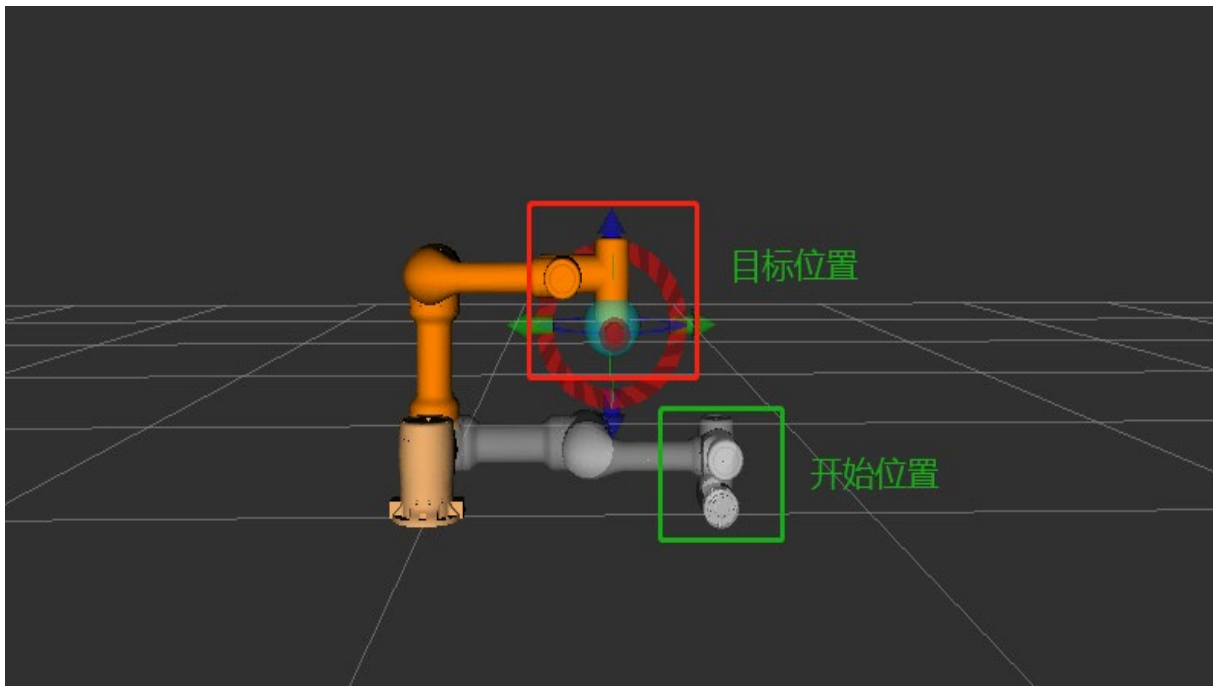


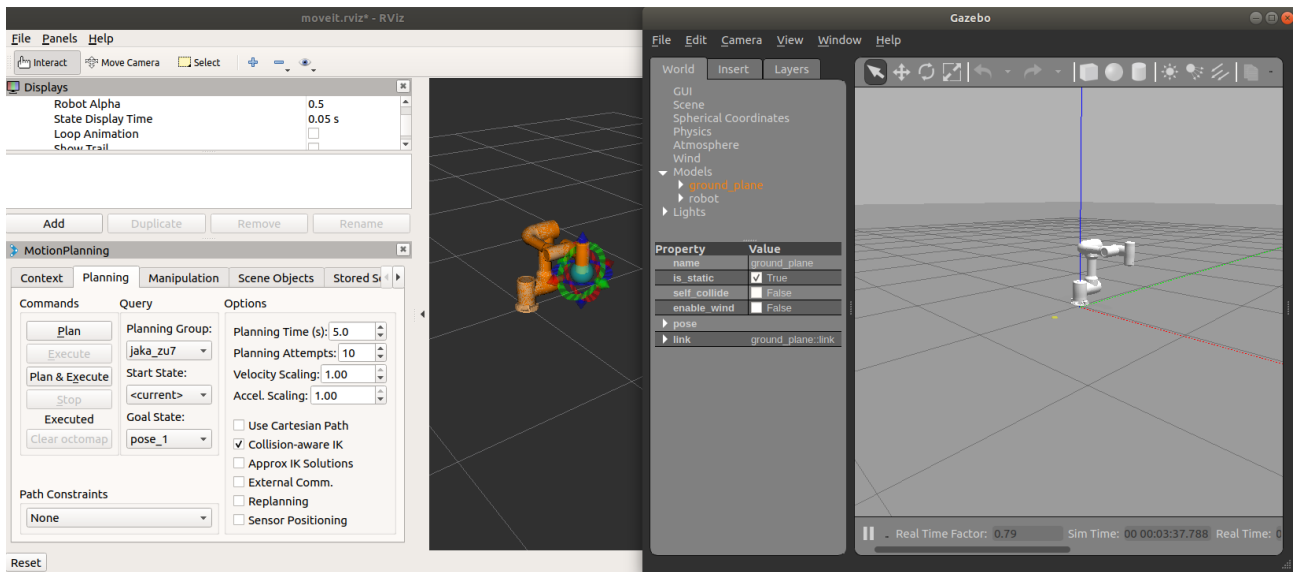
```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~$ cd ./jaka_robot
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ roslaunch jaka_zu3_moveit_config dem
o_gazebo.launch
```

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:

```
src > jaka_planner > launch > moveit_server.launch
1 <launch>
2   <param name="ip" value="192.168.126.128" type="str"/>
3   <node pkg="jaka_planner" type="moveit_server" name="moveit_server" output="screen" />
4 </launch>
```

(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)。

```

moveit_server.launch x moveit_server.cpp x
a_planner > src > moveit_server.cpp > main(int, char * [])
robot.servo_move_enable(false);
ros::Duration(0.5).sleep();
//robot 设置滤波参数
robot.servo_move_use_joint_LPF(0.5);
//robot 上电
robot.power_on();
//robot 上使能
robot.enable_robot();

//创建"/joint_states"的话题
ros::Publisher joint_states_pub = nh.advertise<sensor_msgs::JointState>("/joint_states", 10);

//创建action服务端对象
//Server moveit_server(nh, "/jaka_zu3_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &
//Server moveit_server(nh, "/jaka_zu5_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &
//Server moveit_server(nh, "/jaka_zu7_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &
//Server moveit_server(nh, "/jaka_zu12_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &
//Server moveit_server(nh, "/jaka_zu18_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &
Server moveit_server(nh, "/jaka_minicobo_controller/follow_joint_trajectory", boost::bind(&goalCb, _1, &

// 服务器开始运行
moveit_server.start();
cout << "moveit_start" << endl;

while(ros::ok())
{
    //向RVIZ上报robot 关节信息
    joint_states_callback(joint_states_pub);
    rate.sleep();
    ros::spinOnce();
}

//ros::spin();

```

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

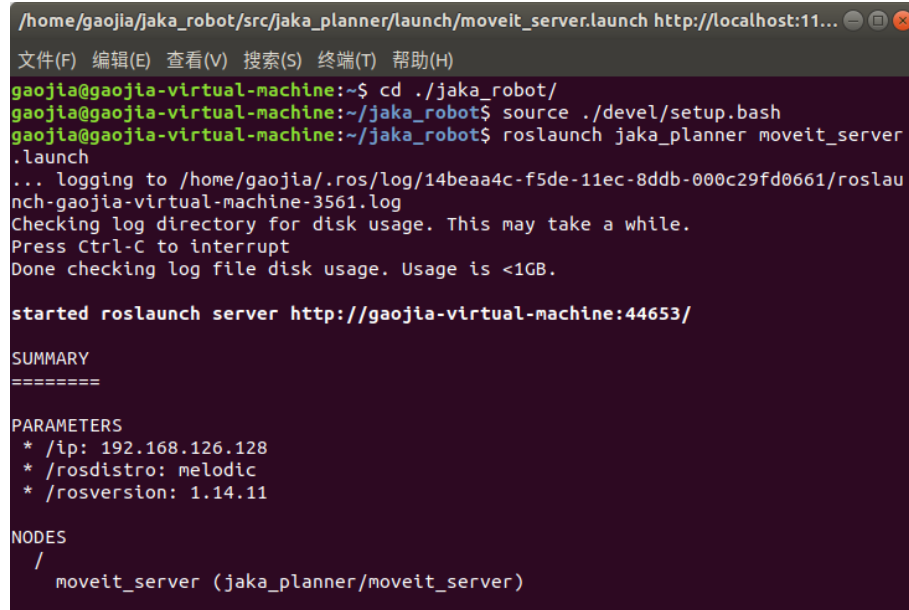
```

gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~$ cd ./jaka_robot/
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$

```

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

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



```
/home/gaojia/jaka_robot/src/jaka_planner/launch/moveit_server.launch http://localhost:11...
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~$ cd ./jaka_robot/
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ roslaunch jaka_planner moveit_server
.launch
... logging to /home/gaojia/.ros/log/14beaa4c-f5de-11ec-8ddb-000c29fd0661/roslau
nch-gaojia-virtual-machine-3561.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://gaojia-virtual-machine:44653/

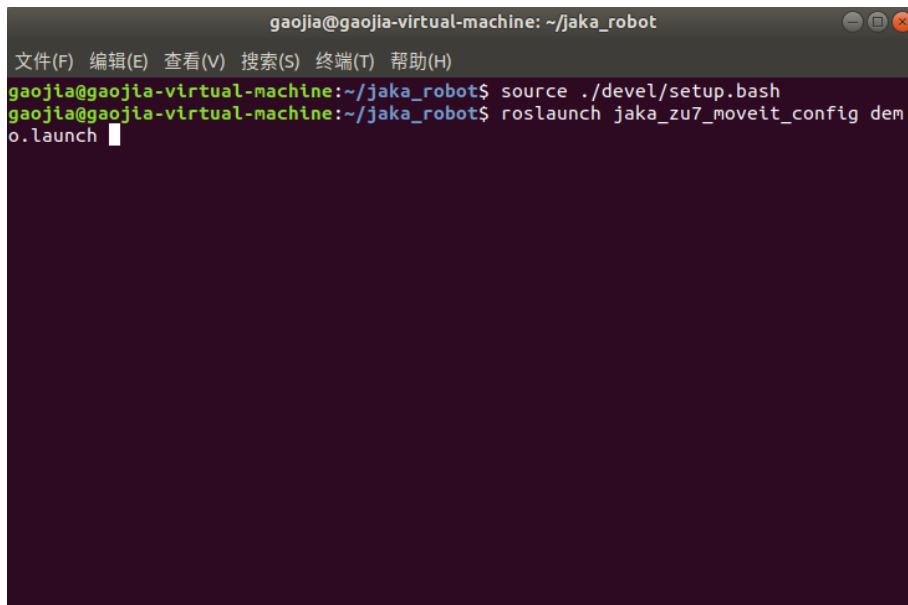
SUMMARY
=====

PARAMETERS
* /ip: 192.168.126.128
* /rostdistro: melodic
* /rosversion: 1.14.11

NODES
/
  moveit_server (jaka_planner/moveit_server)
```

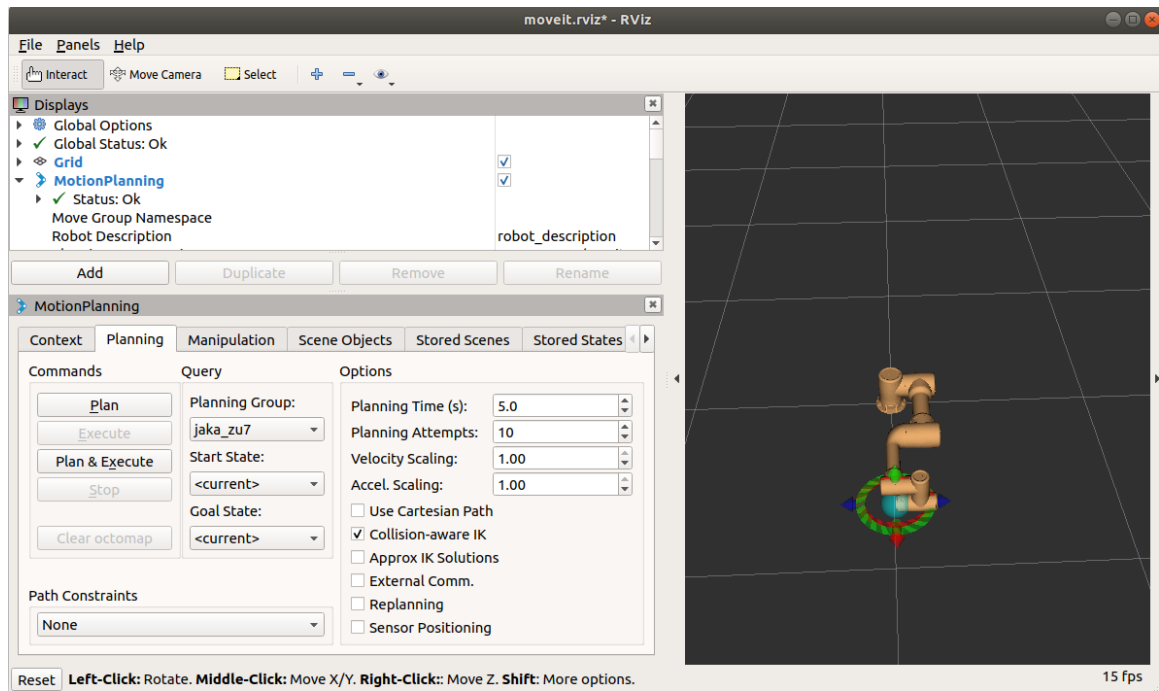
(5) 启动 demo.launch 文件

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

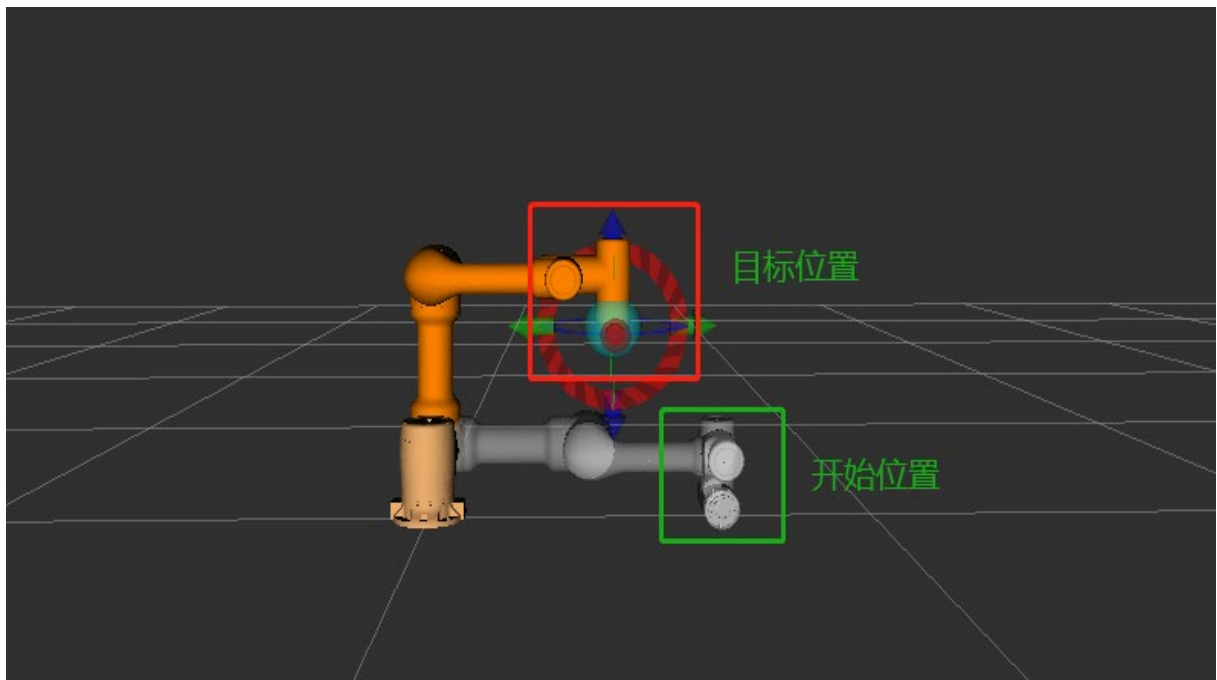


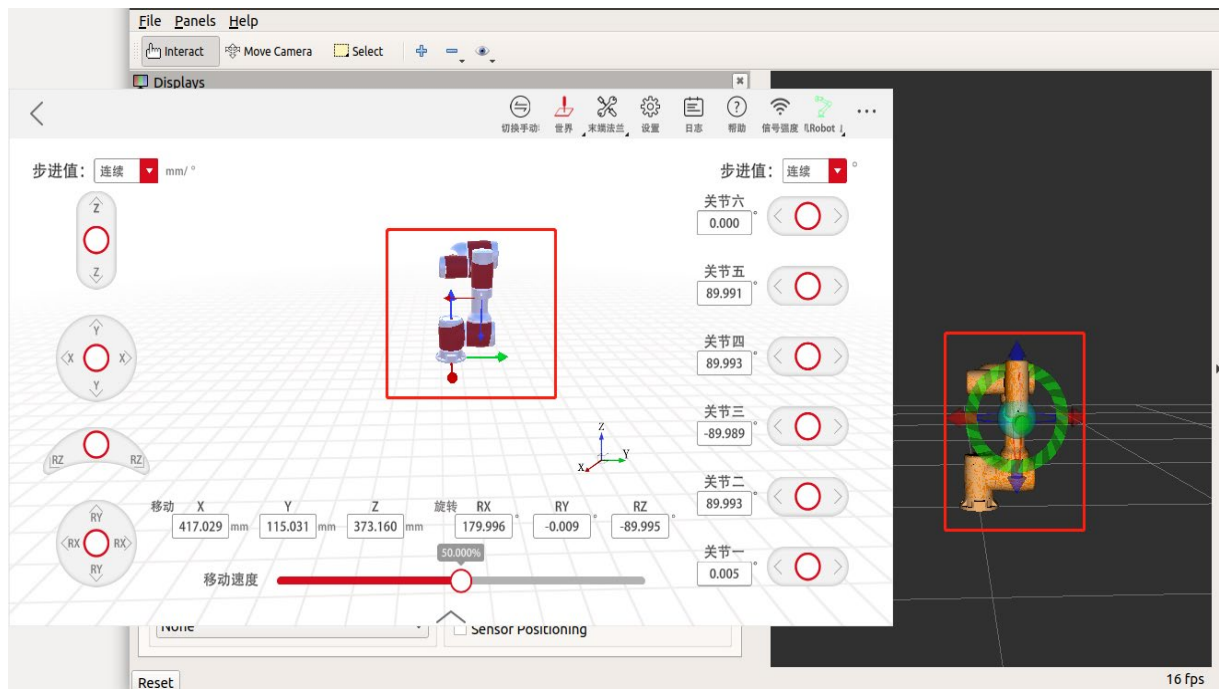
```
gaojia@gaojia-virtual-machine: ~/jaka_robot
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
gaojia@gaojia-virtual-machine:~/jaka_robot$ source ./devel/setup.bash
gaojia@gaojia-virtual-machine:~/jaka_robot$ roslaunch jaka_zu7_moveit_config dem
o.launch
```

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