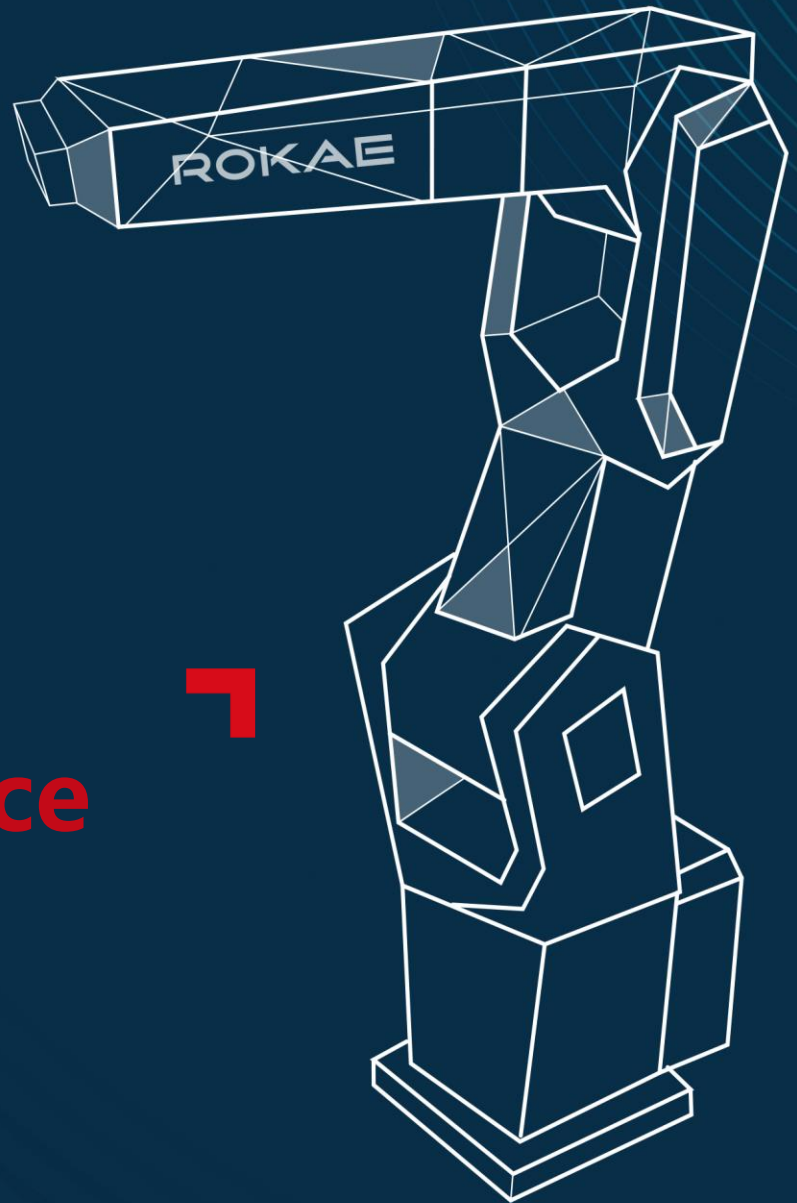


ROKAE 珞石
轻型机器人专家



ROS_Interface

使用手册

让智造更高效

ROS_Interface 使用手册

[类别]

控制系统版本: V2.0

[备注]

文档版本: A

本手册中记载的内容如有变更，恕不事先通告。本公司对手册中可能出现的错误均不承担任何责任。

本公司对因使用本手册及其中所述产品而引起的意外或间接伤害均不承担任何责任，敬请谅解。

本公司不可能预见所有的危险和后果，因此本手册不能警告用户所有可能的危险。

禁止擅自复印或转载本手册的部分或全部内容。

如您发现本手册的内容有误或需要改进抑或补充之处，请不吝指正。

本手册的原始语言为中文，所有其他语言版本均翻译自中文版本。

©版权所有 2015-2023 ROKAE 保留所有权利
珞石（北京）智能科技有限公司
中国·北京

目录

1 手册概述	2
1.1 关于本手册	2
1.2 手册对象	2
1.3 如何阅读产品手册	2
1.4 版本记录	2
2 使用说明	3
2.1 兼容性	3
2.1.1 控制器版本和机器人型号	3
2.1.2 硬件设置	3
2.1.3 网络配置	3
2.1.4 操作系统及版本要求	3
2.2 功能包	3
2.2.1 功能包概述	3
2.2.2 环境搭建	4
3 hardware_interface 接口说明	6
4 使用示例	7
4.1 实机控制	7
4.1.1 RVIZ 控制机器人	7
4.1.2 rqt_controller 控制机器人	8
4.1.3 机器人关节状态获取	9
4.2 仿真机控制	9
4.2.1 Moveit-Gazebo 联合仿真	9
4.3 Moveit 实机控制 demo	10
4.3.1 点位规划运动及避障规划	10
5 常见问题	12

1 手册概述

1.1 关于本手册

感谢您购买本公司的机器人系统。

本手册记载了正确使用机器人的以下说明：

- 珞石机器人 ROS_Interface 的使用；

使用该机器人系统前，请仔细阅读本手册与其他相关手册。

阅读之后，请妥善保管，以便随时取阅。

1.2 手册对象

本手册面向：

- 机器人应用开发工程师。

请务必保证以上人员具备基础的机器人操作、C++编程、ROS 操作系统等所需的知识，并已接受本公司的相关培训。

1.3 如何阅读产品手册

本手册包含单独的安全章节，必须在阅读安全章节后，才能进行操作作业。

1.4 版本记录

版本编号	日期	说明
V1.0	2023.02	ROS_Interface 接口，初步适配 xMateER7p 和 xMateCR7 机型。
V1.1	2023.04	ROS Interface 接口，适配 xMateER7p 以及 xMateCR7 机型等。支持 MoveIt 控制实机以及 Gazebo 仿真机。

2 使用说明

ROS_Interface 是珞石机器人提供给客户用于 ROS 二次开发的软件产品，基于 ROKAE_SDK、ros_control 框架和 MoveIt，方便客户对搭载了 xCore 控制系统的机器人进行控制和操作，主要功能包括实时关节运动控制以及机器人相关状态的获取等。该使用说明书主要介绍 ROS 功能包的用途及使用方法。后续用户可根据需求自行集成。

2.1 兼容性

2.1.1 支持的控制器版本和机器人型号

- 控制器版本：xCore v1.7.0.5 以上。
- 机器人型号：xMateCR7 和 xMateER7pro 协作机型。(持续更新中，如有新需求，可联系销售或技术服务)

2.1.2 软、硬件设置

关于机器人本体、控制柜等硬件的设置，请参考对应机型的使用手册。关于 xCore 软件的基本使用，可参考《xCore 机器人控制系统使用手册》。除网络配置外，使用 ROS_Interface 无需其他额外的硬件设置。

2.1.3 网络配置

ROS_Interface 通过以太网 (TCP/IP) 连接机器人。通过有线连接使用户 PC 和机器人连接同一局域网。机器人配置有 2 个网口，一个是外网口，一个是直联网口。直联网口默认静态 IP 地址是 192.168.0.160。具体连接方式：

- 连接方式：机器人与用户 PC 采用网线直连的方式连接。如果用户工控机与机器人不处于同一个网段，需要配置用户 PC 的 IP 使其与机器人静态 IP 地址处于同一个网段，例如 192.168.0.22。如使用虚拟机，网络连接方法需使用**桥接模式**。

2.1.4 操作系统及版本要求

操作系统	ROS 版本
Ubuntu 20.04	Noetic Ninjemys

2.2 功能包

2.2.1 功能包概述

rokae_ros_Interface 文件夹包含以下功能包：

- rokae_bringup：包含 demo 启动所需要的 launch 文件，以及 controller 的配置文件；
- rokae_hardware_interface：包含 xCore 控制器抽象的硬件接口，用于连接实体机器人与 ros_control；
- rokae_xMateCR7_moveit_config：包含 xMateCR7 机型的 moveit 配置文件；
- rokae_xMateER7p_moveit_config：包含 xMateER7p 机型的 moveit 配置文件；
- rokae_description：包含所需机型的模型配置文件；
- rokae_gazebo：包含启动 Gazebo 所需的配置文件；
- rokae_msgs：包含机器人状态等自定义消息类型；
- rokae_demo：包含演示所需要的 demo 示例；

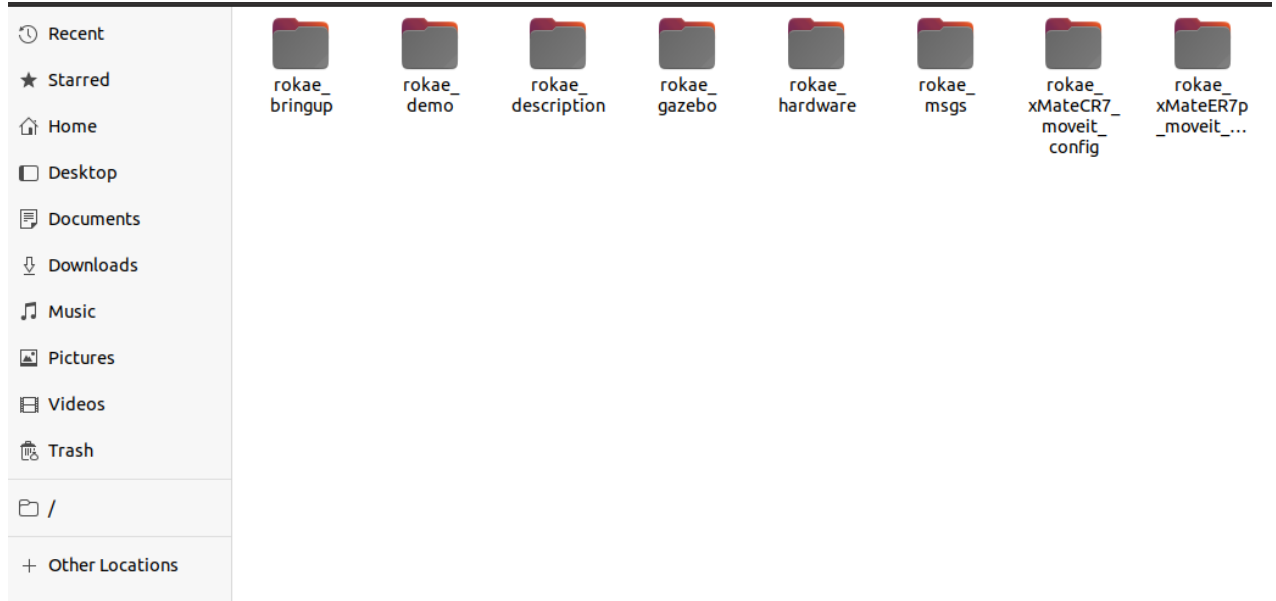


图 2.1 功能包文件夹内容展示

2.2.2 环境搭建

- 安装 [ROS Noetic 版本](#); (内部测试可参考一键安装方法)
- 安装 [Moveit 以及编译工具包](#);
- 安装其他所需要的依赖包 (如已安装可跳过) :

```

> joint_state_publisher_gui
sudo apt-get install ros-noetic-joint-state-publisher-gui

> ros_controllers
sudo apt-get install ros-noetic-ros-control ros-noetic-ros-controllers

> pilz_industrial_motion_planner
sudo apt install ros-noetic-pilz-industrial-motion-planner

> rqt-joint-trajectory-controller
sudo apt-get install ros-noetic-rqt-joint-trajectory-controller

> Industrial_trajectory_filters
sudo apt-get install ros-noetic-industrial-core

```

- 创建所需工作空间

```

mkdir -p ~/rokae_ros_ws/src
cd ~/rokae_ros_ws/src
catkin_init_workspace

```

将提供的 rokae_ros 中的功能包内容添加到 src 文件夹中。(功能包默认与 7 轴协作机器人匹配, 若使用 6 轴, 需修改 rokae_hardware 功能包中 rokae_hardware.cpp 文件中的轴数。)

语句修改:

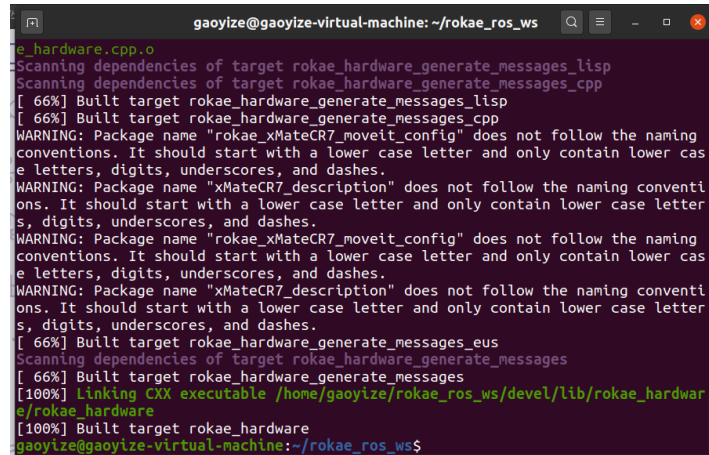
```

原语句: rokae_hardware::RokaeHardwareInterface<7> rokae_hardware_interface
新语句: rokae_hardware::RokaeHardwareInterface<6> rokae_hardware_interface

```

- 编译并将该目录添加到运行目录

```
cd ~/rokae_ros_ws
catkin_make -DCATKIN_WHITELIST_PACKAGES="rokae_msgs"
catkin_make -DCATKIN_WHITELIST_PACKAGES=""
```



```
gaoyize@gaoyize-virtual-machine: ~/rokae_ros_ws
eHardware.cpp.o
Scanning dependencies of target rokaeHardware_generate_messages_lisp
Scanning dependencies of target rokaeHardware_generate_messages_cpp
[ 66%] Built target rokaeHardware_generate_messages_lisp
[ 66%] Built target rokaeHardware_generate_messages_cpp
WARNING: Package name "rokae_xMateCR7_moveit_config" does not follow the naming
conventions. It should start with a lower case letter and only contain lower cas
e letters, digits, underscores, and dashes.
WARNING: Package name "xMateCR7_description" does not follow the naming conventi
ons. It should start with a lower case letter and only contain lower case letter
s, digits, underscores, and dashes.
WARNING: Package name "rokae_xMateCR7_moveit_config" does not follow the naming
conventions. It should start with a lower case letter and only contain lower cas
e letters, digits, underscores, and dashes.
WARNING: Package name "xMateCR7_description" does not follow the naming conventi
ons. It should start with a lower case letter and only contain lower case letter
s, digits, underscores, and dashes.
[ 66%] Built target rokaeHardware_generate_messages_eus
Scanning dependencies of target rokaeHardware_generate_messages
[ 66%] Built target rokaeHardware_generate_messages
[100%] Linking CXX executable /home/gaoyize/rokae_ros_ws/devel/lib/rokaeHardwar
e/rokaeHardware
[100%] Built target rokaeHardware
gaoyize@gaoyize-virtual-machine:~/rokae_ros_ws$
```

图 2.2 编译成功

3 hardware_interface 接口说明

ROS hardware_interface 接口是属于 ros_control 的抽象接口部分，主要的功能为与机器人连接、注册控制器句柄以及机器人关节命令与关节状态的下发与获取等。RokaeHardwareInterface 类主要包括以下接口部分：

init()
bool RokaeHardwareInterface::init(ros::NodeHandle& root_nh) 初始化接口：用于初始化机器人状态、参数状态以及句柄状态。
参数 [in] root_nh: ROS 句柄
返回 True: 初始化成功; False: 初始化失败。
initParameters()
bool RokaeHardwareInterface::initParameters(ros::NodeHandle& root_nh) 初始化参数：从 param server 中获取所需要的参数。
参数 [in] root_nh: ROS 句柄
返回 True: 初始化成功; False: 初始化失败。
initRobot ()
bool RokaeHardwareInterface::initRobot() 初始化机器人：连接机器人，获取当前机器人状态，机器人上电，开启实时控制模式。
返回 True: 初始化成功; False: 初始化失败。
initROSInterface ()
bool RokaeHardwareInterface::initROSInterface() 初始化控制器句柄：注册控制器句柄。
返回 True: 初始化成功; False: 初始化失败。
read()
void RokaeHardwareInterface::read() 获取当前机器人状态并保存至控制器变量。
write()
void RokaeHardwareInterface::write() 将控制器更新的指令下发至机器人。

4 使用示例

目前的 ROS_Interface 接口主要基于 ros_control 实现 xMateCR7 和 xMateER7pro 协作机型的关节运动控制和机器人运动状态的获取。通过 RVIZ 对机器人进行可视化，通过 Moveit 对路径进行规划。

4.1 实机控制

本次示例版本默认加载使用 [joint trajectory controller](#)。

4.1.1 RVIZ 控制机器人

- 启动 hardware_interface 节点

打开终端，设置工作路径

```
cd ~/rokae_ros_ws/src
source devel/setup.bash
```

运行相关节点 (local_ip 为本机 ip, 必须添加, 如果 ip 错误或缺失将会提示报错), 界面显示如图 4.1 所示。

```
roslaunch rokae_bringup xMateCR7_control.launch robot_ip:=192.168.0.160 local_ip:=[]
```

- 控制机器人实机

运行节点后, RVIZ 界面中可以显示当前机器人模型 (当前位姿) 并加载 Moveit 和 joint_trajectory_controller 控制器。Moveit 中加载了 oml, chomp 以及 pilz-industrial-motion-planner 规划器, 通过选择需要的规划算法, 拖动 RVIZ 中的机器人, 点击 motion_planning 插件中的 “plan”, 轨迹规划确认合理后, 点击 “excution”, 实体机器人会运动至选定位置 (轨迹无误也可点击 “Plan & Execute” 按钮)。规划器参数设置见图 4.2。

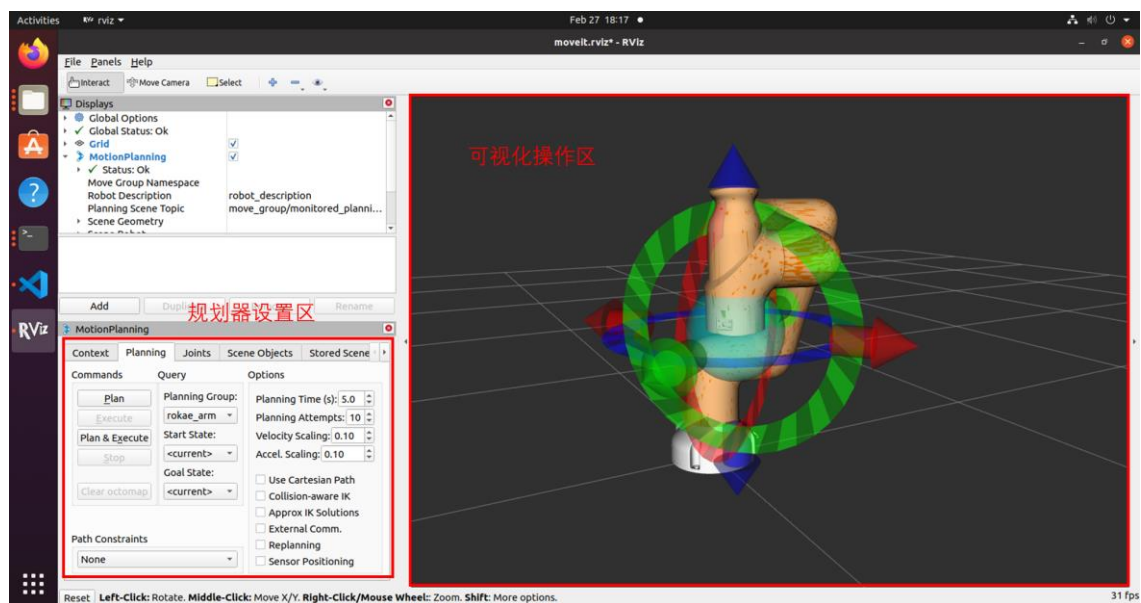


图 4.1 RVIZ 应用界面 (xMateCR7)

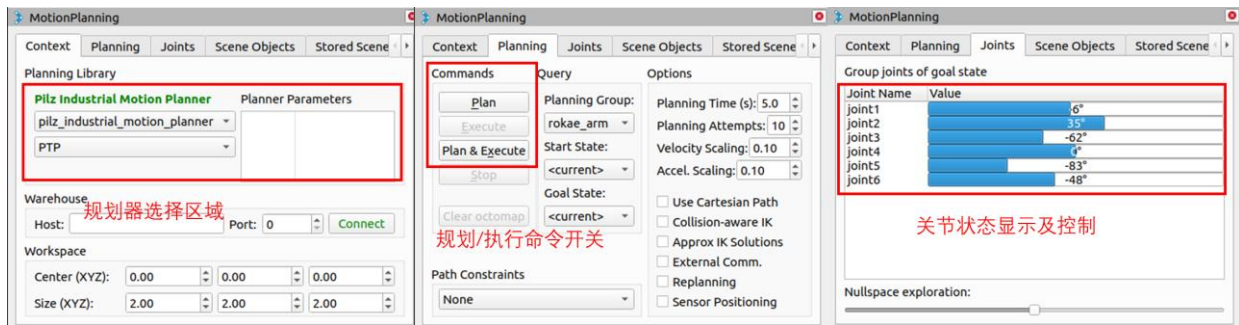


图 4.2 规划器设置参数界面

4.1.2 rqt_controller 控制机器人

- 启动 hardware_interface 节点

打开终端，设置工作路径

```
cd ~/rokae_ros_ws/src
source devel/setup.bash
```

运行相关节点 (local_ip 为本机 ip，必须添加，如果 ip 错误或缺失将会提示报错)

```
roslaunch rokae_bringup xMateCR7_control.launch robot_ip:=192.168.0.160 local_ip:=[]
```

- 启动 rqt_joint_trajectory_controller

另外打开新终端，运行如下指令

```
roslaunch rqt_joint_trajectory_controller rqt_joint_trajectory_controller
```

- 控制机器人

将控制器命名空间选定为 “/controller_manager”，控制器选择为 “position_joint_trajectory_controller”
 点击运行按钮，**设置好相关的速度参数**，通过可视化关节角度，即可控制机器人移动。

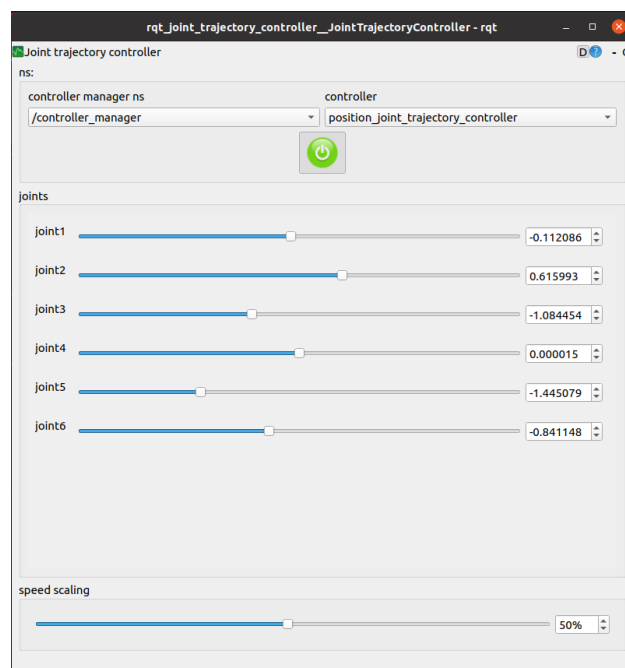


图 4.2 rqt_controller 控制界面

4.1.3 机器人关节状态获取

- 启动 hardware_interface 节点

打开终端，设置工作路径

```
cd ~/rokae_ros_ws/src
source devel/setup.bash
```

运行相关节点 (local_ip 为本机 ip，必须添加，如果 ip 错误或缺失将会提示报错)

```
roslaunch rokae_bringup xMateCR7_control.launch robot_ip:=192.168.0.160 local_ip:=[]
```

- 获取机器人状态

另外打开新终端，运行如下指令，即可获取机器人的当前关节位置。

```
rostopic echo /joint_states
```

The image shows two terminal windows side-by-side. The left window shows the output of 'rostopic echo /joint_states' before movement, with joint positions ranging from approximately -0.11 to 3.1158. The right window shows the output after movement, with joint positions ranging from approximately -0.65 to 3.1158. Both windows show the same header information (seq, stamp, frame_id) and joint names (joint1 to joint6).

图 4.3 机器人状态监控-移动前后

4.2 仿真机控制

4.2.1 Moveit-Gazebo 联合仿真

- 启动 gazebo 节点

打开终端，设置工作路径

```
cd ~/rokae_ros_ws/src
source devel/setup.bash
```

运行相关节点

```
roslaunch rokae_gazebo xMateCR7_gazebo.launch
```

- 通过 RVIZ 界面拖动位置，点击“plan”，轨迹确认无误后点击“excute”后可以发现 Gazebo 中的仿真机器人会接收指令进行运动。

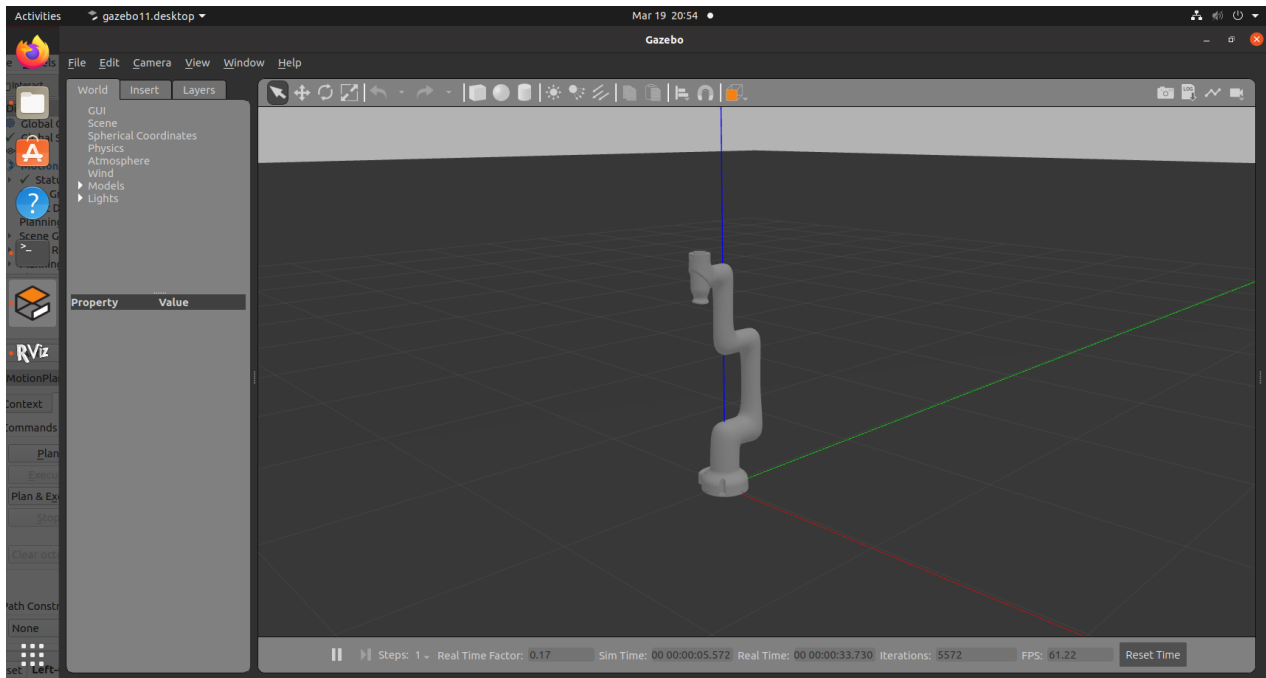


图 4.4 Gazebo 仿真机显示

4.3 Moveit 实机控制 demo

4.3.1 点位规划运动及避障规划

本 demo (包括 C++ 版本以及 python 版本) 以 xMateCR7 机型为例, 实现机器人单目标笛卡尔空间点位规划运动、多目标关节空间点位规划运动以及单目标关节空间避障运动规划运动。

- 启动 hardware_interface 节点

打开终端, 设置工作路径

```
cd ~/rokae_ros_ws/src
source devel/setup.bash
```

运行相关节点 (local_ip 为本机 ip, 必须添加, 如果 ip 错误或缺失将会提示报错)

```
roslaunch rokae_bringup xMateCR7_control.launch robot_ip:=192.168.0.160 local_ip:=[]
```

- 运行 Demo (以下指令二选一即可)

打开新窗口, 并在命令行输入以下指令

(C++ 版本)

```
roslaunch rokae_demo CR7_motion_demo_cpp
```

(python 版本)

```
roslaunch rokae_demo CR7_motion_demo_python.py
```

动作描述:

- 机器人运动至设定 Home 点;
- 障碍物生成;
- 单目标笛卡尔空间规划运动至设定点;
- 单目标关节空间避障规划运动至设定点;

-
- 障碍物去除;
 - 多目标笛卡尔空间连续运动规划;
 - 返回至指定点。

注意:

- 默认为不需要回到初始位置。如果需要运动前回到初始位置，则运行前需根据指定位置（Python 与 C++的代码均在主函数添加障碍物的代码下面）的注释修改代码;
- 程序会规划轴空间运动避开障碍物并运动到指定位置，规划的路径会显示到 Rviz 界面，如果 Moveit 命令窗口报错或者对路径不满意，则直接点击回车重新规划路径;
- 如果该窗口中没有报错，则可以输入 e，执行该路径规划

5 常见问题

问题 1: 机型启动报错;

解决方法: 首先排查是否机型对应的 launch 文件是否正确; 确定无误后排查机器人的轴数与 hardware 中的轴数是否一致, 最后检查网线以及网段是否满足要求。

问题 2: xMateCR7 机型 192.168.0.160 网段无法连接机器人;

解决方法: 外网口直连有可能是 192.168.2.160 网段, 具体参见《xCore 机器人控制系统使用手册 V1.7.0》

问题 3: hardware 节点启动出现实时控制错误等问题;

解决方法: 手动通过 HMI 关闭 RCI, 重新启动节点。

ROKAE 珞石
轻型机器人专家



 **400-010-8700**

北京总部：北京市海淀区农科院西路6号海青大厦A座7层

山东分公司：济宁市邹城市中心店镇机电产业园恒丰路888号

苏州分公司：苏州工业园区星湖街328号创意产业园1-A1F

深圳分公司：深圳市宝安区中粮福安机器人智造产业园10栋1楼