



睿尔曼机器人 rm_moveit2_config 使用说明书

V1.0



睿尔曼智能科技（北京）有限公司



文件修订记录：

| 版本号 | 时间 | 备注 |
|------|------------|----|
| V1.0 | 2023-11-22 | 拟制 |
| | | |
| | | |



目录

| | |
|---------------------------------|----|
| 1. rm_moveit2_config 说明 | 3 |
| 2. rm_moveit2_config 使用 | 3 |
| 2.1 moveit2 控制虚拟机械臂 | 3 |
| 2.2 moveit2 控制真实机械臂 | 5 |
| 3. rm_moveit2_config 架构说明 | 6 |
| 3.1 功能包文件总览 | 6 |
| 4. rm_moveit2_config 话题说明 | 11 |



1. rm_moveit2_config 说明

rm_moveit2_config 文件夹为实现 moveit2 控制真实机械臂的功能包, 该功能包的主要作用为调用官方的 moveit2 框架, 结合我们机械臂本身的 URDF 生成适配于我们机械臂的 moveit2 的配置和启动文件, 通过该功能包我们可以实现 moveit2 控制虚拟机械臂和控制真实机械臂, 在下文中将通过以下几个方面详细介绍该功能包。

1. 功能包使用。
2. 功能包架构说明。
3. 功能包话题说明。

通过这三部分内容的介绍可以帮助大家:

1. 了解该功能包的使用。
2. 熟悉功能包中的文件构成及作用。
3. 熟悉功能包相关的话题, 方便开发和使用

2. rm_moveit2_config 使用

2.1 moveit2 控制虚拟机械臂

首先配置好环境完成连接后我们可以通过以下命令直接启动节点。

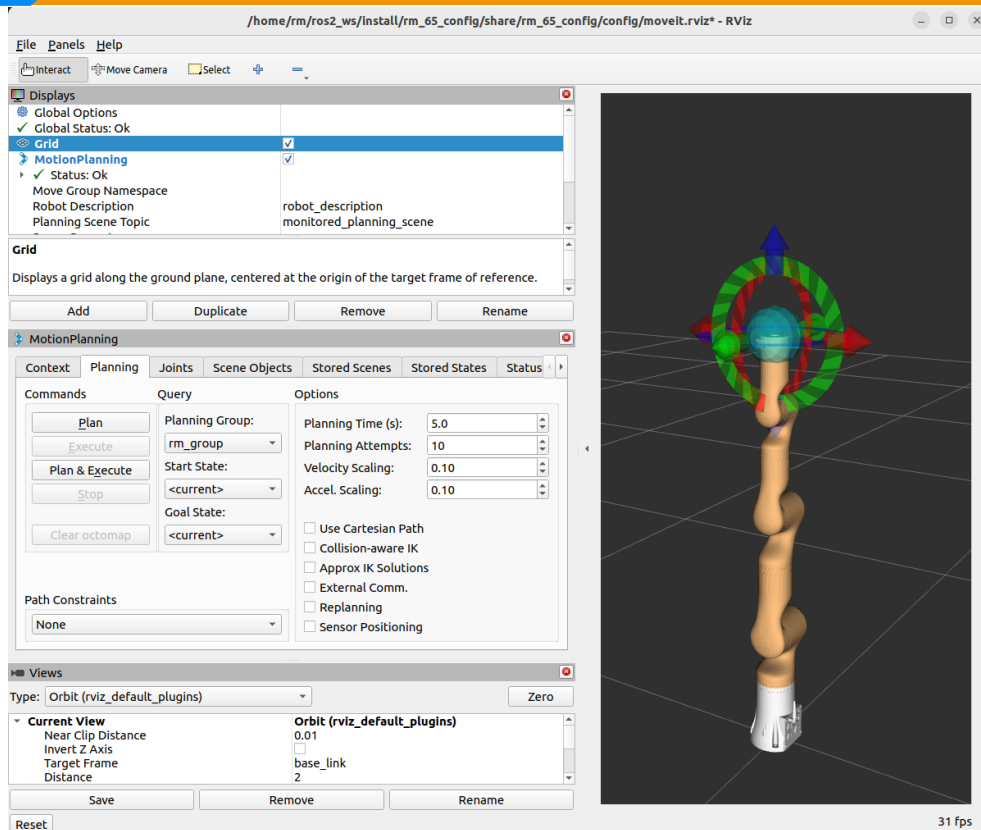
```
rm@rm-desktop:~$ ros2 launch rm_<arm_type>_config demo.launch.py
```

在实际使用时需要将以上的 <arm_type> 更换为实际的机械臂型号, 可选择的机械臂型号有 65、63、eco65、75。

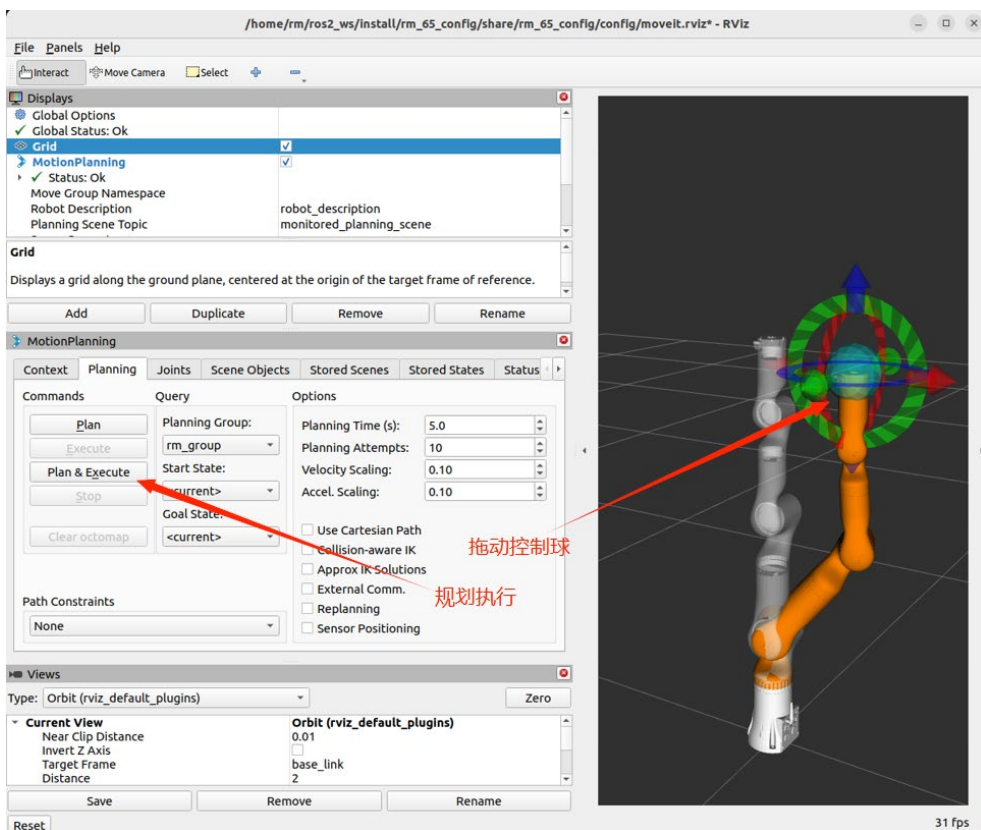
例如 65 机械臂的启动命令:

```
rm@rm-desktop:~$ ros2 launch rm_65_config demo.launch.py
```

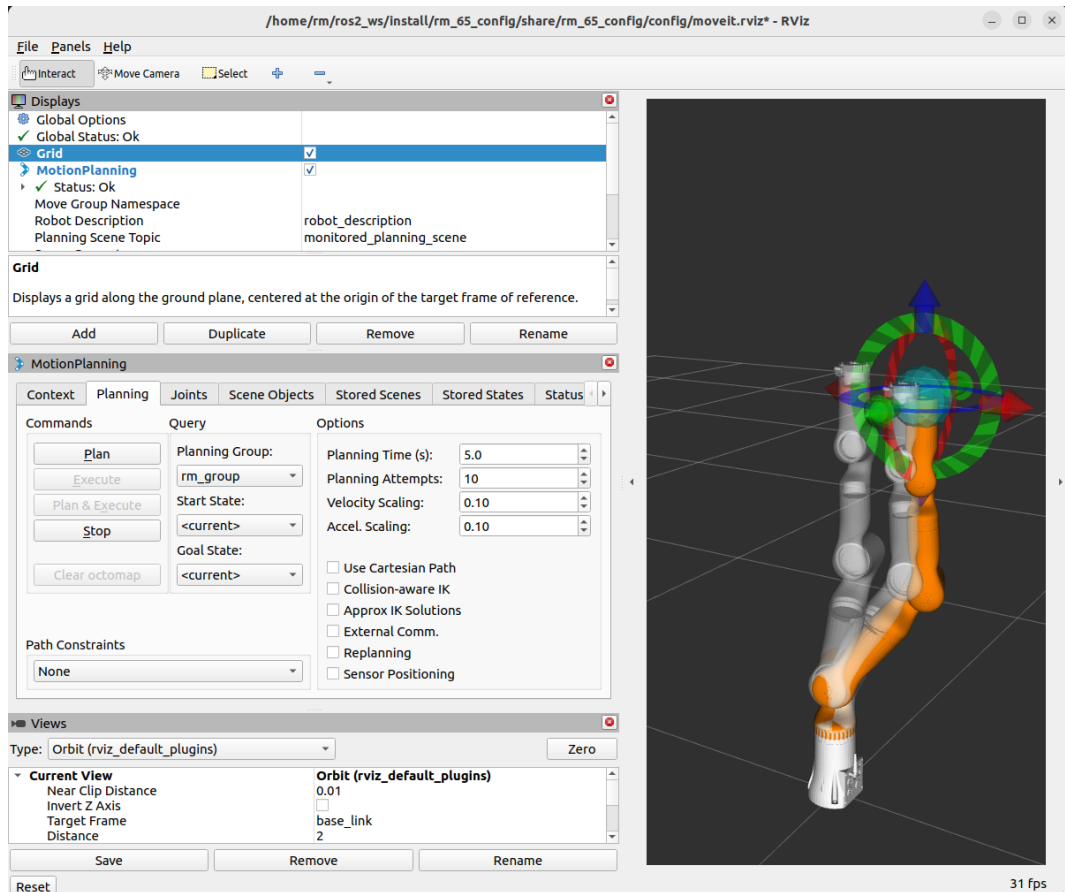
节点启动成功后, 将显示以下画面。



接下来我们可以通过拖动控制球使机械臂到达目标位置，然后点击规划执行。



规划执行。



2.2 moveit2 控制真实机械臂

控制真实机械臂需要的控制指令相对较多一些，如下为详细的控制方式。

首先运行底盘驱动节点。

```
rm@rm-desktop:~$ ros2 launch rm_driver rm_<arm_type>_driver.launch.py
```

接下来需要运行 rm_description 功能包文件。

```
rm@rm-desktop:~$ ros2 launch rm_description rm_<arm_type>_display.launch.py
```

之后需要运行中间功能包 rm_control 的相关节点。

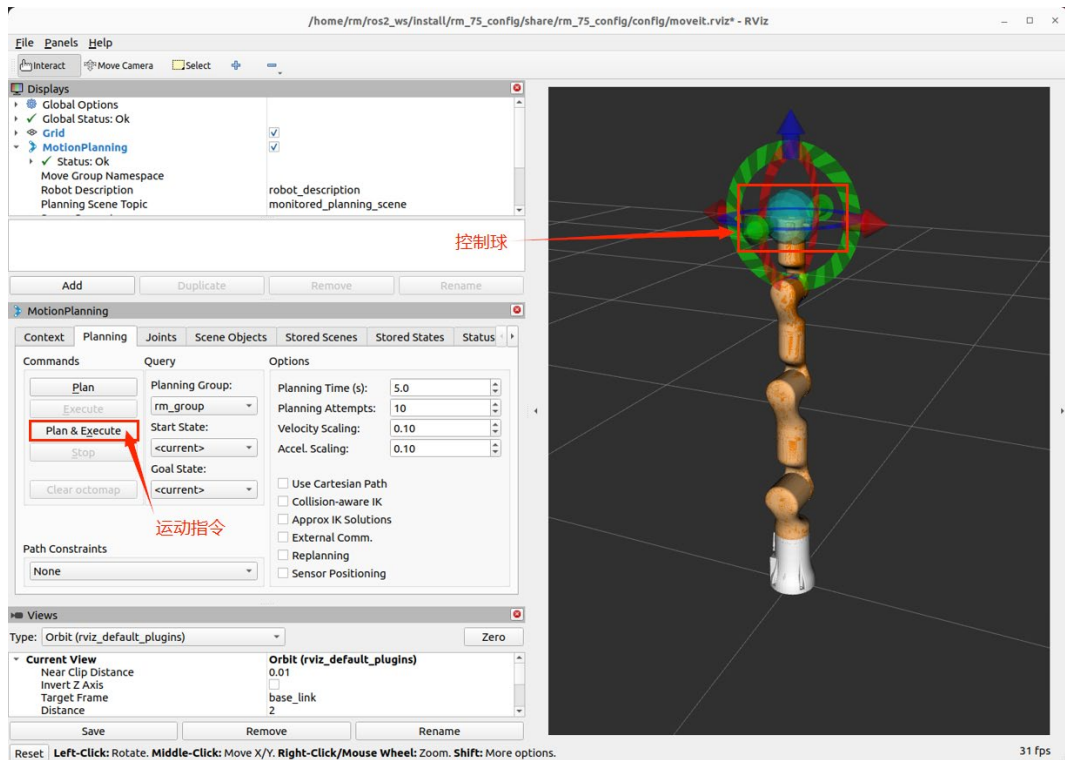
```
rm@rm-desktop:~$ ros2 launch rm_control rm_<arm_type>_control.launch.py
```

最终需要启动控制真实机械臂的 moveit2 节点。

```
rm@rm-desktop:~$ ros2 launch rm_<arm_type>_config  
real_moveit_demo.launch.py
```

注意以上指令均需要将 <arm_type> 更换为对应的机械臂型号，可选的型号有 65、63、eco65、75。

完成以上操作后将会出现以下界面，我们可以通过拖动控制球的方式控制机械臂运动。



3. rm_moveit2_config 架构说明

3.1 功能包文件总览

当前 rm_driver 功能包的文件构成如下。

| | |
|--|--------------------------|
| └─ rm_63_config | #63 机械臂 moveit2 功能包 |
| ├─ CMakeLists.txt | #63 机械臂 moveit2 功能包编译规则 |
| ├─ config | #63 机械臂 moveit2 功能包参数文件夹 |
| │ └─ initial_positions.yaml | #63 机械臂 moveit2 初始化位姿 |
| │ └─ joint_limits.yaml | #63 机械臂关节限制 |
| │ └─ kinematics.yaml | #63 机械臂运动学参数 |
| │ └─ moveit_controllers.yaml | #63 机械臂 moveit2 控制器 |
| │ └─ moveit.rviz | #63 机械臂 rviz2 显示配置文件 |
| │ └─ pilz_cartesian_limits.yaml | |
| └─ rml_63_description.ros2_control.xacro | #63 机械臂 xacro 描述文件 |



```
| | └─ rml_63_description.srdf          #63 机械臂 moveit2 控制配置文件
| | └─ rml_63_description.urdf.xacro    #63 机械臂 xacro 描述文件
| | └─ ros2_controllers.yaml            #63 机械臂运动控制器
| └─ launch
| | └─ demo.launch.py                  #63 虚拟机械臂 moveit2 启动文件
| | └─ gazebo_moveit_demo.launch.py    #63 仿真机械臂 moveit2 启动文件
| | └─ move_group.launch.py
| | └─ moveit_rviz.launch.py
| | └─ real_moveit_demo.launch.py      #63 真实机械臂 moveit2 启动文件
| | └─ rsp.launch.py
| | └─ setup_assistant.launch.py
| | └─ spawn_controllers.launch.py
| | └─ static_virtual_joint_tfs.launch.py
| | └─ warehouse_db.launch.py
| └─ package.xml
└─ rm_65_config                        #65 机械臂 moveit2 功能包 (文件解释参考 63)
    └─ CMakeLists.txt
    └─ config
        └─ initial_positions.yaml
        └─ joint_limits.yaml
        └─ kinematics.yaml
        └─ moveit_controllers.yaml
```



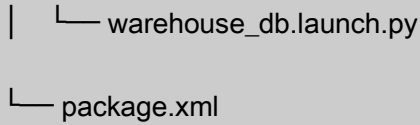

```
| | └─ moveit.rviz
| | └─ pilz_cartesian_limits.yaml
| | └─ rm_65_description.ros2_control.xacro
| | └─ rm_65_description.srdf
| | └─ rm_65_description.urdf.xacro
| | └─ ros2_controllers.yaml
| └─ launch
| | └─ demo.launch.py
| | └─ gazebo_moveit_demo.launch.py
| | └─ move_group.launch.py
| | └─ moveit_rviz.launch.py
| | └─ real_moveit_demo.launch.py
| | └─ rsp.launch.py
| | └─ setup_assistant.launch.py
| | └─ spawn_controllers.launch.py
| | └─ static_virtual_joint_tfs.launch.py
| | └─ warehouse_db.launch.py
| └─ package.xml
└─ rm_75_config #75 机械臂 moveit2 功能包 ( 文件解释参考 63 )
    └─ CMakeLists.txt
    └─ config
    | └─ initial_positions.yaml
```



```
| | └─ joint_limits.yaml
| | └─ kinematics.yaml
| | └─ moveit_controllers.yaml
| | └─ moveit.rviz
| | └─ pilz_cartesian_limits.yaml
| | └─ rm_75_description.ros2_control.xacro
| | └─ rm_75_description.srdf
| | └─ rm_75_description.urdf.xacro
| | └─ ros2_controllers.yaml
| └─ launch
| | └─ demo.launch.py
| | └─ gazebo_moveit_demo.launch.py
| | └─ move_group.launch.py
| | └─ moveit_rviz.launch.py
| | └─ real_moveit_demo.launch.py
| | └─ rsp.launch.py
| | └─ setup_assistant.launch.py
| | └─ spawn_controllers.launch.py
| | └─ static_virtual_joint_tfs.launch.py
| | └─ warehouse_db.launch.py
| └─ package.xml
└─ rm_eco65_config #eco65 机械臂 moveit2 功能包 ( 文件解释参考 63 )
```



```
├─ CMakeLists.txt
├─ config
│   ├── initial_positions.yaml
│   ├── joint_limits.yaml
│   ├── kinematics.yaml
│   ├── moveit_controllers.yaml
│   ├── moveit.rviz
│   ├── pilz_cartesian_limits.yaml
│   ├── rm_eco65_description.ros2_control.xacro
│   ├── rm_eco65_description.srdf
│   ├── rm_eco65_description.urdf.xacro
│   └─ ros2_controllers.yaml
├─ launch
│   ├── demo.launch.py
│   ├── gazebo_moveit_demo.launch.py
│   ├── move_group.launch.py
│   ├── moveit_rviz.launch.py
│   ├── real_moveit_demo.launch.py
│   ├── rsp.launch.py
│   ├── setup_assistant.launch.py
│   ├── spawn_controllers.launch.py
│   └─ static_virtual_joint_tfs.launch.py
```



关于 moveit2 的话题说明，为使其话题结构更加清晰明白在这里以节点话题的数据流图的方式进行查看和讲解。

```
rm@rm-desktop:~$ ros2 run rqt_graph rqt_graph
```

```

graph TD
    IMD([Interactive_marker_display_34195381403392]) --> MG((/move_group))
    TL1([Transform_listener_imp1_7010030650]) --> MG
    TL2([Transform_listener_imp1_5602a6a7a50]) --> MG
    TL3([Transform_listener_imp1_55a8020463d0]) --> MG
    JRE[trajectory_execution_event] --> MG
    MG --> MA[move_action]
    MG --> ET[execute_trajectory]
    MG --> DLP[display_planned_path]
    MG --> MSCM[moveit_simple_controller_manager]
    MG --> JRD[jrm_driver]
    MG --> JS[joint_states]
    JRD --> JDC[jrm_drivermoveit_caffd_cmd]
    JDC --> JRD
    JS --> RSP[robot_state_publisher]
    RSP --> MGPR[move_group_private_54565348264240]
    MGPR --> PS[planning_scene]
    MGPR --> PSW[planning_scene_world]
    MG --> JRC[jrm_group_controller]
    JRC --> JRCFT[jrm_group_controller_follow_joint_trajectory]
    JRCFT --> JRCFTAF[jrm_group_controller_follow_joint_trajectory_actionfeedback]
    JRCFTAF --> JRCFTAS[jrm_group_controller_follow_joint_trajectory_actionstatus]
    JRCFTAS --> JRCFT
    JRCFT --> JRC
    JRC --> MG
  
```

```

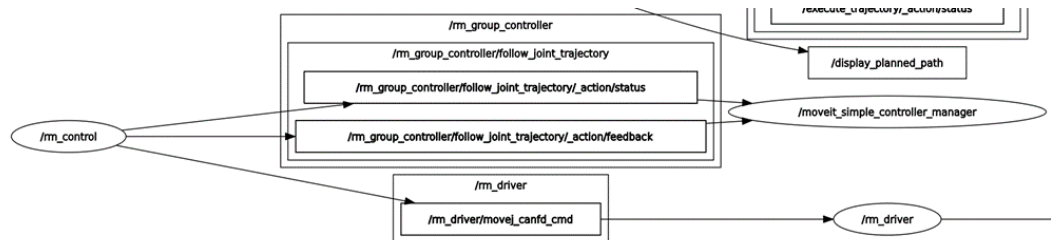
graph LR
    Input(( )) --> rm_driver1([/rm_driver])
    rm_driver1 --> joint_states[/joint_states/]
    joint_states --> robot_state_publisher([/robot_state_publisher/])
    joint_states --> move_group_private([/move_group_private_94569348264240/])
    attached_collision_object[/attached_collision_object/] --> move_group_private
    planning_scene[/planning_scene/] --> move_group_private
    planning_scene_world[/planning_scene_world/] --> move_group_private
    rm_control([/rm_control/]) --> status[/rm_group_controller/follow_joint_trajectory/_action/status/]
    rm_control --> feedback[/rm_group_controller/follow_joint_trajectory/_action/feedback/]
    status --> move_group_private
    feedback --> move_group_private
    move_group_private --> rm_driver2[/rm_driver/]
    rm_driver2 --> move_group_cmd[/rm_driver/move_group_cmd/]
    move_group_cmd --> rm_driver3([/rm_driver/])
    move_group_private --> display_planned_path[/display_planned_path/]
    move_group_private --> simple_controller_manager([/moveit_simple_controller_manager/])
    simple_controller_manager --> rm_driver3
  
```

生活美好，臂不可少
<http://www.realman-robotics.com>

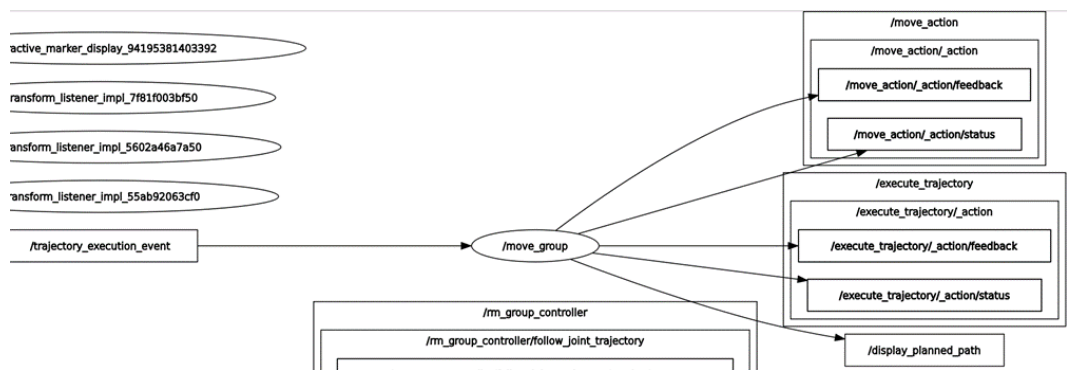


发布关节间的 TF 变换; /move_group_private 是 moveit2 的相关节点, moveit2 在规划时也需要实时获取当前机械臂的关节状态信息, 所以也订阅了该话题。

由图可知 rm_driver 还订阅了 rm_control 的 /rm_driver/movej_canfd_cmd 话题, 该话题是机械臂透传功能的话题, 通过该话题 rm_control 将规划的关节点发布给 rm_driver 节点控制机械臂进行运动。



rm_control 为 rm_driver 与 moveit2 之间通信的桥梁, 其通过 /rm_group_controller/follow_joint_trajectory 动作与 /moveit_simple_controller_manager 进行通信, 获取规划点, 并进行插值运算, 将插值之后的数据通过透传的方式给到 rm_driver。



Moveit2 本身涉及的节点有 move_group、move_group_private、moveit_simple_controller_manager, 它们的主要作用为实现机械臂的运动规划, 并将规划信息等数据显示在 rviz 中, 另一方面还需要将规划数据传递到 rm_control 端, 进行进一步细分。