



# 睿尔曼机器人 rm\_moveit\_config 使用说明书

V1.2



---

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	2024-1-16	拟制
V1.1	2024-7-4	修订（添加 GEN72 相关配置文件）
V1.2	2024-9-10	修订（添加 ECO63 相关配置文件）



# 目录

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



## 1. rm\_moveit\_config 说明

rm\_moveit\_config 文件夹为实现 moveit 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 moveit 框架，结合我们机械臂本身的 URDF 生成适配于我们机械臂的 moveit 的配置和启动文件，通过该功能包我们可以实现 moveit 控制虚拟机械臂和控制真实机械臂。

- 功能包使用。
- 功能包架构说明。
- 功能包话题说明。

通过这三部分内容的介绍可以帮助大家：

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

代码链接：

[https://github.com/RealManRobot/rm\\_robot/tree/main/rm\\_moveit\\_config](https://github.com/RealManRobot/rm_robot/tree/main/rm_moveit_config)。

## 2. rm\_moveit\_config 使用

### 2.1 moveit 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:~$ roslaunch rm_<arm_type>_moveit_config demo.launch
```

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、eco65、eco63、75、gen72 六维力的型号有 65\_6f、eco65\_6f、75\_6f。

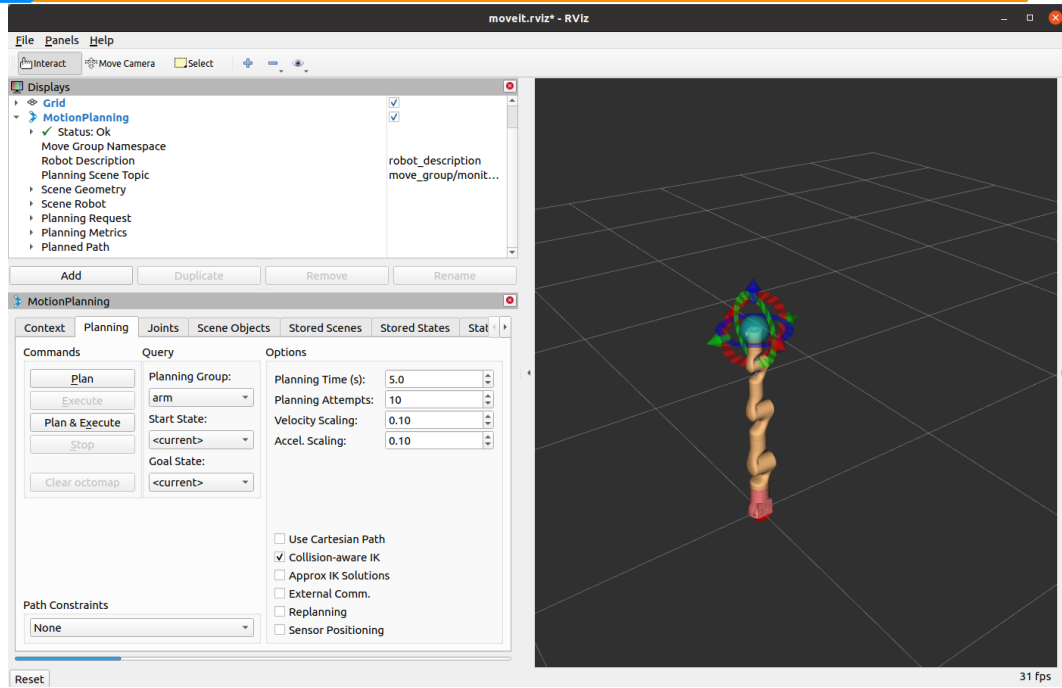
63 机械臂需要使用以下指令启动，63 的六维力型号为 63\_6f。

```
rm@rm-desktop:~$ roslaunch rml_63_moveit_config demo.launch
```

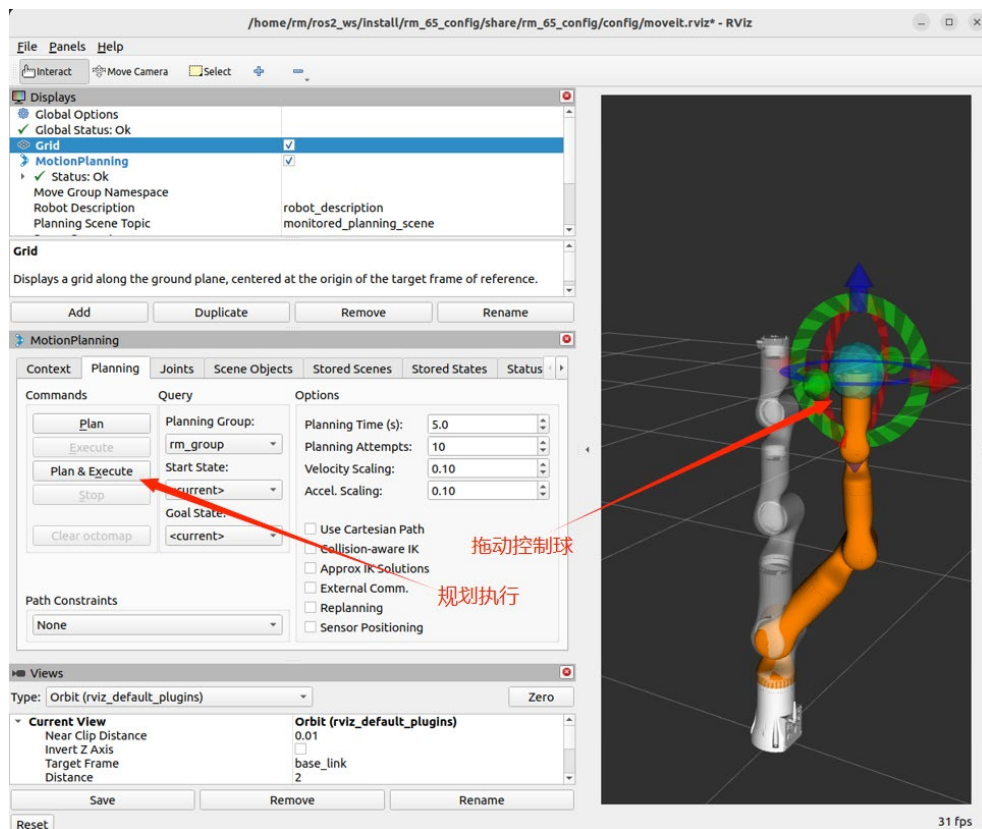
例如 65 机械臂的启动命令：

```
rm@rm-desktop:~$ roslaunch rm_65_moveit_config demo.launch
```

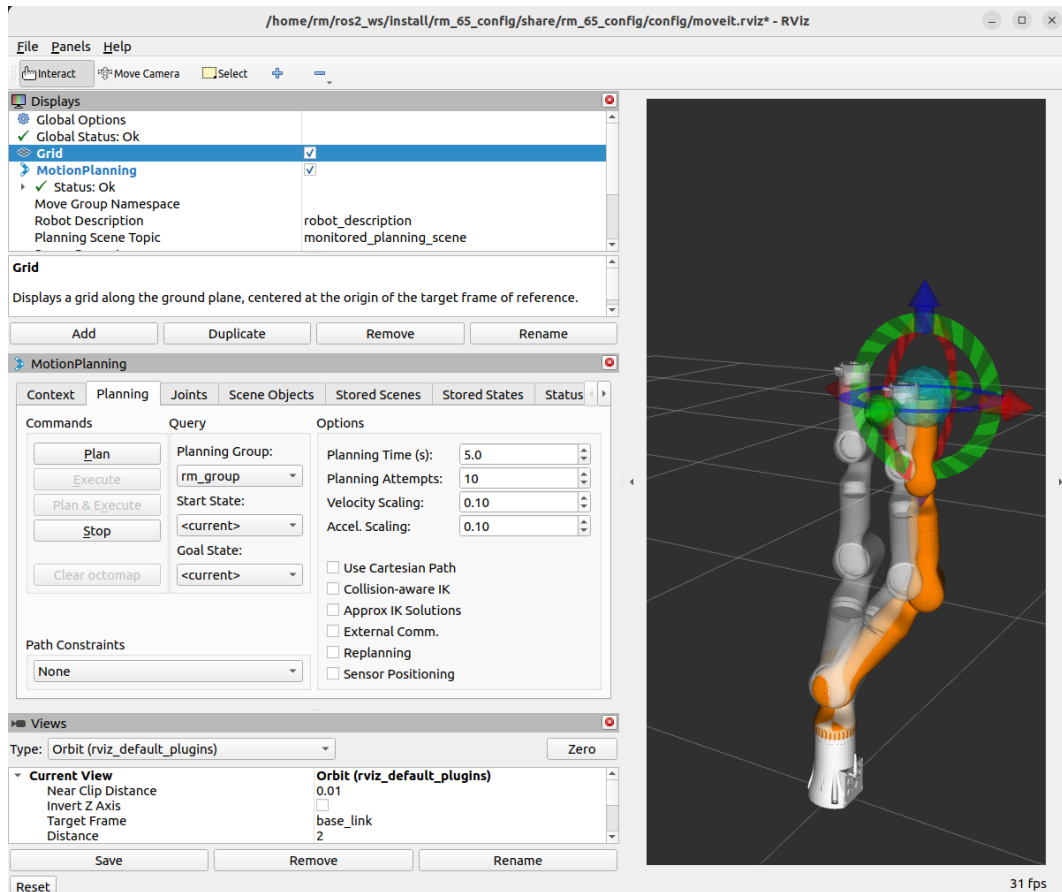
节点启动成功后，将显示以下画面。



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 moveit 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:~$ roslaunch rm_driver rm_<arm_type>_driver.launch
```

之后需要运行中间功能包 rm\_control 的相关节点。

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

最终需要启动控制真实机械臂的 moveit 节点。

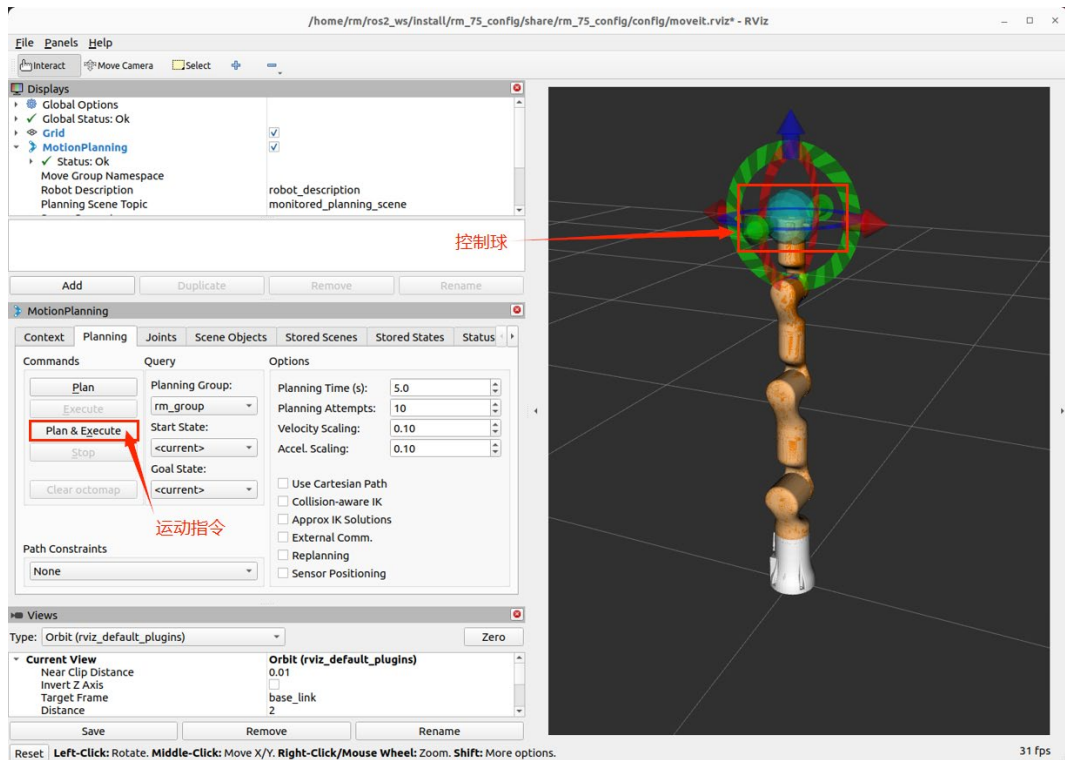
```
rm@rm-desktop:~$ roslaunch rm_<arm_type>_moveit_config demo_realrobot.launch
```

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、eco65、eco63、75、65\_6f、eco65\_6f、75\_6f、gen72。

注意在使用 63 时应使用如下启动指令，若使用六维力设备时，应将 63 更换为 63\_6f。

```
rm@rm-desktop:~$ roslaunch rml_63_moveit_config demo_realrobot.launch
```

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\_moveit\_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_65_moveit_config	#65 机械臂 moveit 功能包
CMakeLists.txt	#65 机械臂 moveit 功能包编译规则
config	#65 机械臂 moveit2 功能包参数文件
cartesian_limits.yaml	
chomp_planning.yaml	
controllers_gazebo.yaml	#gazebo 仿真控制器
controllers.yaml	#真实机械臂控制器
fake_controllers.yaml	
joint_limits.yaml	#65 机械臂关节限制
kinematics.yaml	#65 机械臂运动学参数
ompl_planning.yaml	
rm_65.srdf	#65 机械臂 moveit 控制配置文件
ros_controllers.yaml	#65 机械臂运动控制器
sensors_3d.yaml	
launch	
chomp_planning_pipeline.launch.xml	
default_warehouse_db.launch	
demo_gazebo.launch	#65 仿真机械臂 moveit2 启动文件
demo.launch	#65 虚拟机械臂 moveit2 启动文件
demo_realrobot.launch	#65 真实机械臂 moveit2 启动文件
fake_moveit_controller_manager.launch.xml	
gazebo.launch	
joystick_control.launch	



```
— move_group.launch
— moveit_planning_execution.launch
— moveit.rviz
— moveit_rviz.launch
— ompl_planning_pipeline.launch.xml
— pilz_industrial_motion_planner_planning_pipeline.launch.xml
— planning_context.launch
— planning_pipeline.launch.xml
— rm_65_moveit_controller_manager.launch(复件).xml
— rm_65_moveit_controller_manager.launch.xml
— rm_65_moveit_sensor_manager.launch.xml
— ros_controllers.launch
— run_benchmark_ompl.launch
— sensor_manager.launch.xml
— setup_assistant.launch
— trajectory_execution.launch(复件).xml
— trajectory_execution.launch.xml
— warehouse.launch
— warehouse_settings.launch.xml
package.xml
rm_75_moveit_config          #75 机械臂 moveit 功能包 ( 文件解释参考 65 )
— CMakeLists.txt
— config
— cartesian_limits.yaml
— chomp_planning.yaml
— controllers_gazebo.yaml
— controllers.yaml
— fake_controllers.yaml
— joint_limits.yaml
— kinematics.yaml
— ompl_planning.yaml
— rm_75.srdf
— ros_controllers.yaml
— sensors_3d.yaml
— launch
— chomp_planning_pipeline.launch.xml
— default_warehouse_db.launch
— demo_gazebo.launch
— demo.launch
— demo_realrobot.launch
— fake_moveit_controller_manager.launch.xml
— gazebo.launch
— joystick_control.launch
— move_group.launch
— moveit_planning_execution.launch
— moveit.rviz
— moveit_rviz.launch
— ompl_planning_pipeline.launch.xml
— pilz_industrial_motion_planner_planning_pipeline.launch.xml
— planning_context.launch
— planning_pipeline.launch.xml
— rm_75_moveit_controller_manager.launch(复件).xml
— rm_75_moveit_controller_manager.launch.xml
```

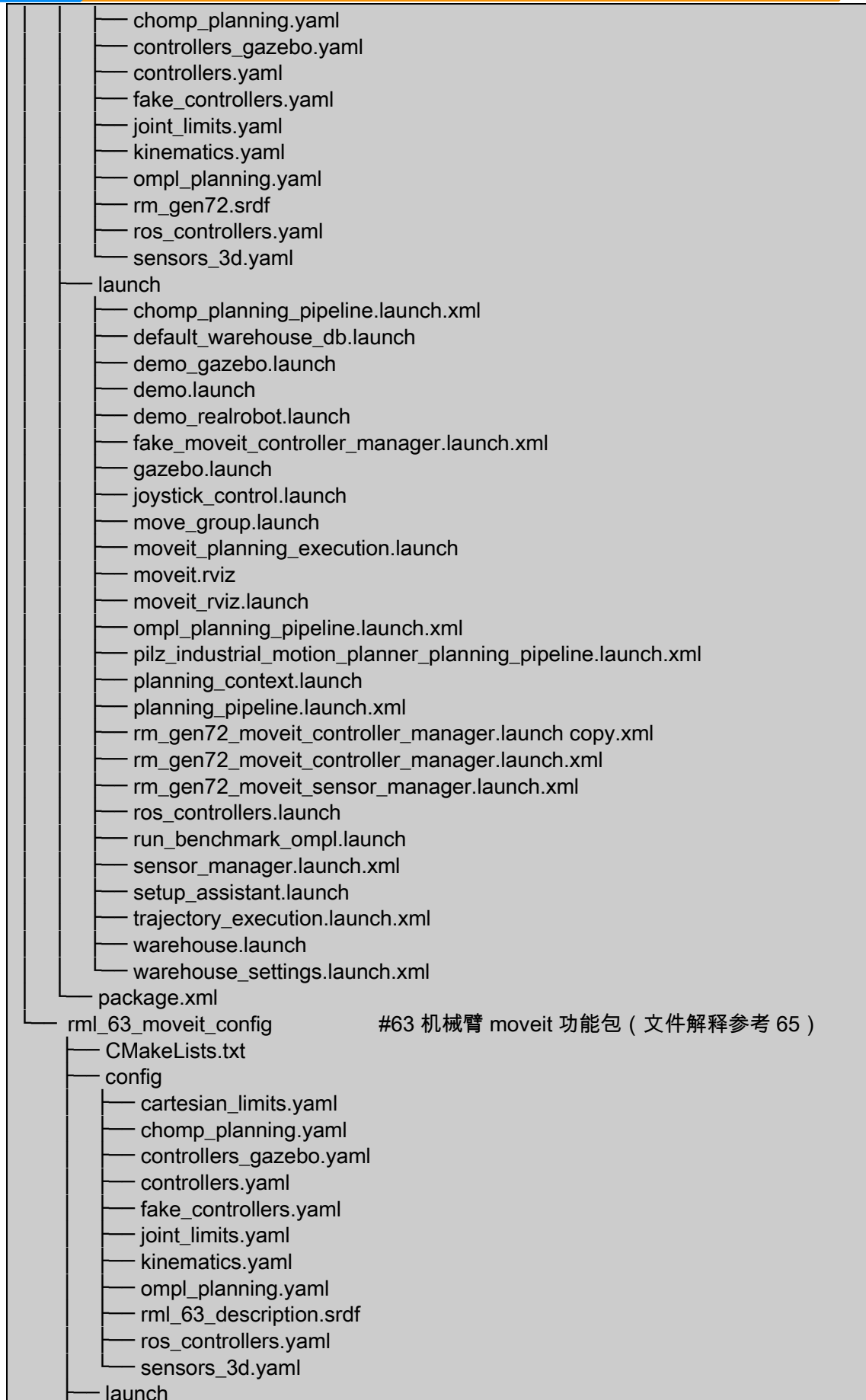


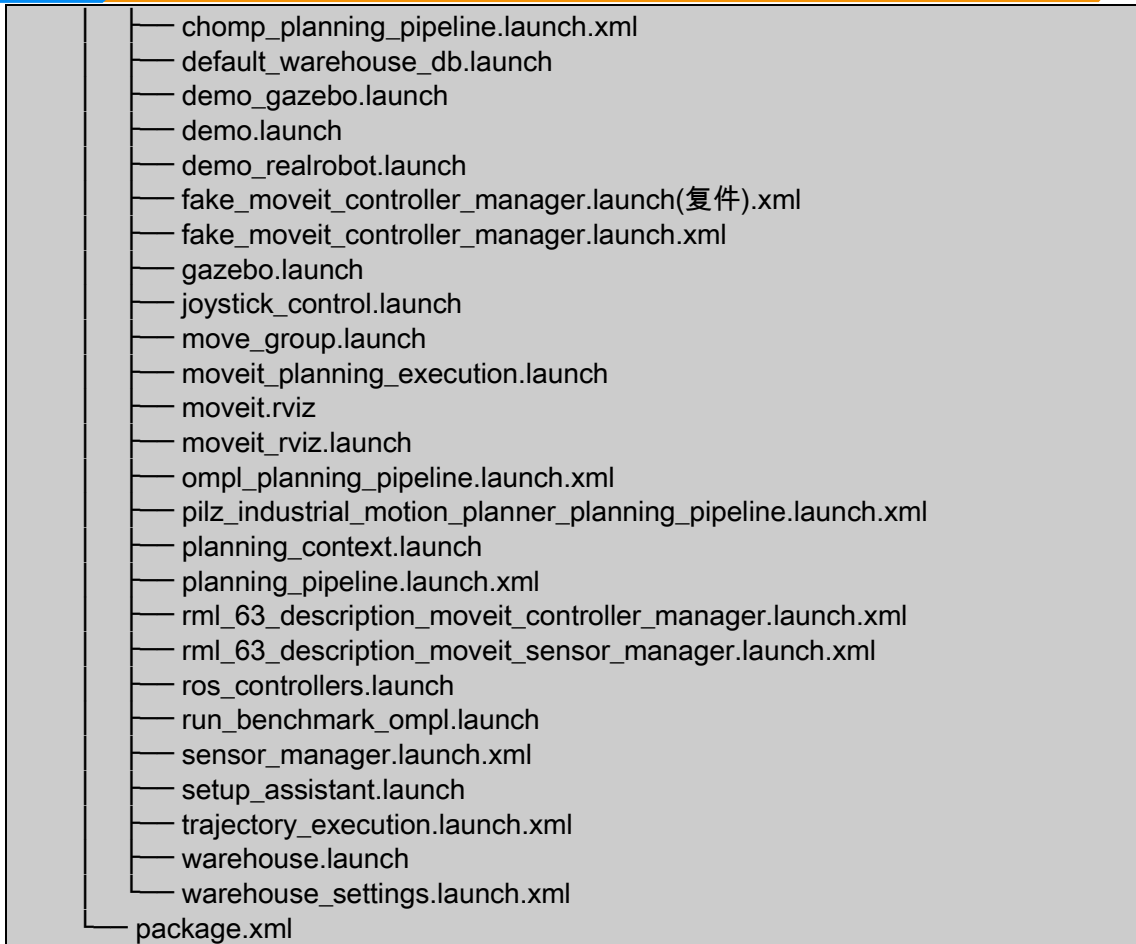


```
— rm_75_moveit_sensor_manager.launch.xml
— ros_controllers.launch
— run_benchmark_ompl.launch
— sensor_manager.launch.xml
— setup_assistant.launch
— trajectory_execution.launch.xml
— warehouse.launch
— warehouse_settings.launch.xml
— package.xml
— rm_eco63_moveit_config      #ECO65 机械臂 moveit 功能包 ( 文件解释参考 63 )
  — CMakeLists.txt
  — config
    — cartesian_limits.yaml
    — chomp_planning.yaml
    — controllers_gazebo.yaml
    — controllers.yaml
    — fake_controllers.yaml
    — gazebo_controllers.yaml
    — joint_limits.yaml
    — kinematics.yaml
    — ompl_planning.yaml
    — rm_eco63.srdf
    — ros_controllers.yaml
    — sensors_3d.yaml
  — launch
    — chomp_planning_pipeline.launch.xml
    — default_warehouse_db.launch
    — demo_gazebo.launch
    — demo.launch
    — demo_realrobot.launch
    — fake_moveit_controller_manager.launch.xml
    — gazebo.launch
    — joystick_control.launch
    — move_group.launch
    — moveit_planning_execution.launch
    — moveit.rviz
    — moveit_rviz.launch
    — ompl_planning_pipeline.launch.xml
    — pilz_industrial_motion_planner_planning_pipeline.launch.xml
    — planning_context.launch
    — planning_pipeline.launch.xml
    — rm_eco63_moveit_controller_manager.launch copy.xml
    — rm_eco63_moveit_controller_manager.launch.xml
    — rm_eco63_moveit_sensor_manager.launch.xml
    — ros_controllers.launch
    — run_benchmark_ompl.launch
    — sensor_manager.launch.xml
    — setup_assistant.launch
    — trajectory_execution.launch.xml
    — warehouse.launch
    — warehouse_settings.launch.xml
  — package.xml
— rm_eco65_moveit_config      #ECO65 机械臂 moveit 功能包 ( 文件解释参考 65 )
```



```
— CMakeLists.txt
— config
  — cartesian_limits.yaml
  — chomp_planning.yaml
  — controllers_gazebo.yaml
  — controllers.yaml
  — fake_controllers.yaml
  — gazebo_controllers.yaml
  — gazebo_rm_eco65.urdf
  — joint_limits.yaml
  — kinematics.yaml
  — ompl_planning.yaml
  — rm_eco65.srdf
  — ros_controllers.yaml
  — sensors_3d.yaml
  — simple_moveit_controllers.yaml
  — stomp_planning.yaml
— launch
  — chomp_planning_pipeline.launch.xml
  — default_warehouse_db.launch
  — demo_gazebo.launch
  — demo.launch
  — demo_realrobot.launch
  — fake_moveit_controller_manager.launch.xml
  — gazebo.launch
  — joystick_control.launch
  — move_group.launch
  — moveit_planning_execution.launch
  — moveit.rviz
  — moveit_rviz.launch
  — ompl-chomp_planning_pipeline.launch.xml
  — ompl_planning_pipeline.launch.xml
  — pilz_industrial_motion_planner_planning_pipeline.launch.xml
  — planning_context.launch
  — planning_pipeline.launch.xml
  — rm_eco65_moveit_controller_manager.launch copy.xml
  — rm_eco65_moveit_controller_manager.launch.xml
  — rm_eco65_moveit_sensor_manager.launch.xml
  — ros_controllers.launch
  — ros_control_moveit_controller_manager.launch.xml
  — run_benchmark_ompl.launch
  — sensor_manager.launch.xml
  — setup_assistant.launch
  — simple_moveit_controller_manager.launch.xml
  — stomp_planning_pipeline.launch.xml
  — trajectory_execution.launch.xml
  — warehouse.launch
  — warehouse_settings.launch.xml
— package.xml
— rm_gen72_moveit_config      #GEN72 机械臂 moveit 功能包 ( 文件解释参考 65 )
  — CMakeLists.txt
  — config
    — cartesian_limits.yaml
```





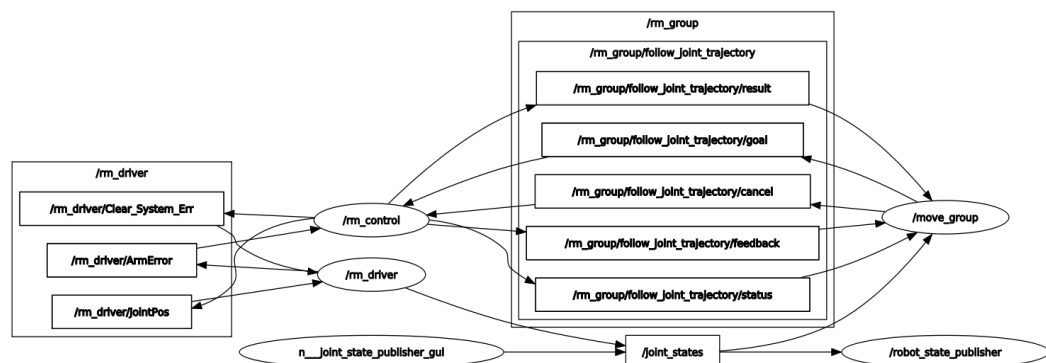
## 4. rm\_moveit\_config 话题说明

关于 moveit 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。

在启动如上控制真实机器人的节点后可以运行如下指令查看当前话题的对接情况。

```
rm@rm-desktop:~$ rosrn rqt_graph rqt_graph
```

运行成功后界面将显示如下画面。



该图反应了当前运行的节点与节点之间的话题通信关系, 首先查看/rm\_driver 节点, 该节点在 moveit 运行时订阅和发布的话题如下。



由图可知，rm\_driver 发布的/joint\_states 话题在持续被/robot\_state\_publisher 节点和/move\_group 节点订阅。/robot\_state\_publisher 接收/joint\_states 是为了持续发布关节间的 TF 变换；/move\_group 是 moveit 的相关节点，moveit 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/jointPos 话题，该话题是机械臂透传功能的话题，通过该话题 rm\_control 将规划的关节点位发布给 rm\_driver 节点控制机械臂进行运动。

rm\_control 为 rm\_driver 与 moveit 之间通信的桥梁，其通过/rm\_group /follow\_joint\_trajectory 动作与/move\_group 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。