Random Movement

Preface

We have built MoveIt environments on both the Jetson_Nano motherboard and the Orin series motherboard. Due to the onboard performance of the Jetson_Nano, running the MoveIt program on the motherboard will be slow and slow to load, and it will take about 3 minutes to complete the loading. Therefore, we recommend that users of the Jetson motherboard run the MoveIt program on the configured virtual machine we provide. The Orin motherboard can run MoveIt smoothly on the motherboard without running it on a virtual machine. Whether running on a virtual machine or on the motherboard, the startup instructions are the same. The following tutorials will take running on the Orin motherboard as an example.

1. Functional Description

After the program is started, the robot arm in Rviz will randomly plan to move to a certain posture in a loop.

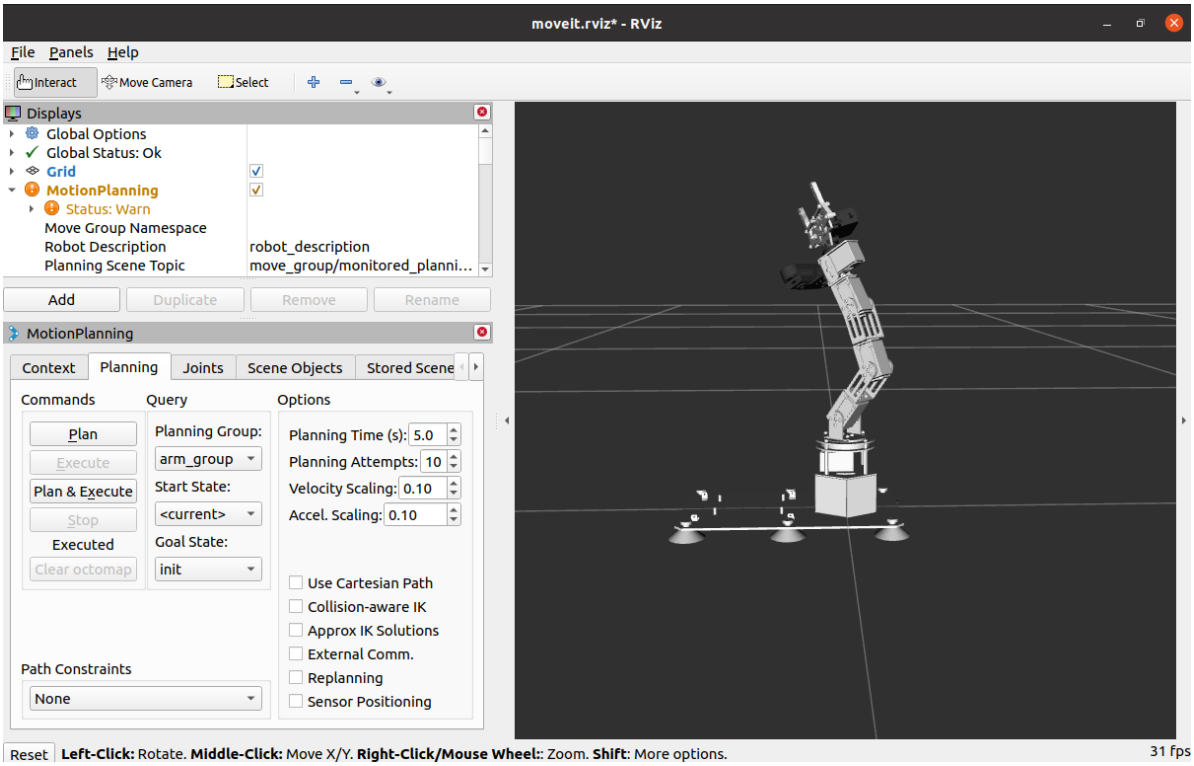
2. Start

Enter the following command in the terminal to start,

```
#Start MoveIt
roslaunch dofbot_pro_config demo.launch
```

After MoveIt is successfully started, enter in the terminal,

```
#Start the random movement program, choose one of the following versions to
start
#Python version
rosrun arm_moveit_demo 01_random_move.py
#C++ version
rosrun arm_moveit_demo 01_random_move
```



After the program runs, the robot arm will cyclically plan the path to run to a random point.

3. Code analysis

Take the analysis of the Python version as an example, the code path:
`/home/jetson/dofbot_pro_ws/src/arm_moveit_demo/scripts/01_random_move.py`

```
#!/usr/bin/env python
# coding: utf-8
import rospy
from time import sleep
#Import the MoveIt library
from moveit_commander.move_group import MoveGroupCommander
if __name__ == '__main__':
    # Initialize node
    rospy.init_node("dofbot_pro_random_move")
    # Initialize arm planning group
    dofbot_pro = MoveGroupCommander("arm_group")
    # Allow replanning after motion planning fails
    dofbot_pro.allow_replanning(True)
    # Set planning time
    dofbot_pro.set_planning_time(5)
    # Number of planning attempts
    dofbot_pro.set_num_planning_attempts(10)
    # Set target position tolerance
    dofbot_pro.set_goal_position_tolerance(0.01)
    # Set target orientation tolerance
    dofbot_pro.set_goal_orientation_tolerance(0.01)
    # Set target tolerance
    dofbot_pro.set_goal_tolerance(0.01)
    # Set maximum speed
    dofbot_pro.set_max_velocity_scaling_factor(1.0)
    # Set the maximum acceleration
    dofbot_pro.set_max_acceleration_scaling_factor(1.0)
    while not rospy.is_shutdown():
```

```
        # Set a random target point
        dofbot_pro.set_random_target()
        # Start movement
        dofbot_pro.go()
        sleep(0.5)
```