MoveIt2 Configuration

Not a required course: The provided system has already been configured.

MoveIt2 is an advanced software framework for robot motion simulation, planning, and control.

1. Environment Setup

Board: Jetson Orin Nano/Nx

ROS2: Humble

1.1. Install MoveIt2

```
sudo apt install ros-humble-moveit*
```

Jetson Orin series boards have been tested and can directly run MoveIt2 related cases. The overall smoothness is acceptable!

1.2. Create Folder

Create MoveIt2 auto-generated configuration files:

```
mkdir ~/dofbot_pro_ws/src/dofbot_pro_moveit
```

2. MoveIt2 Configuration

MoveIt Setup Assistant is a graphical tool used to simplify and accelerate the creation and configuration of MoveIt2 configuration packages.

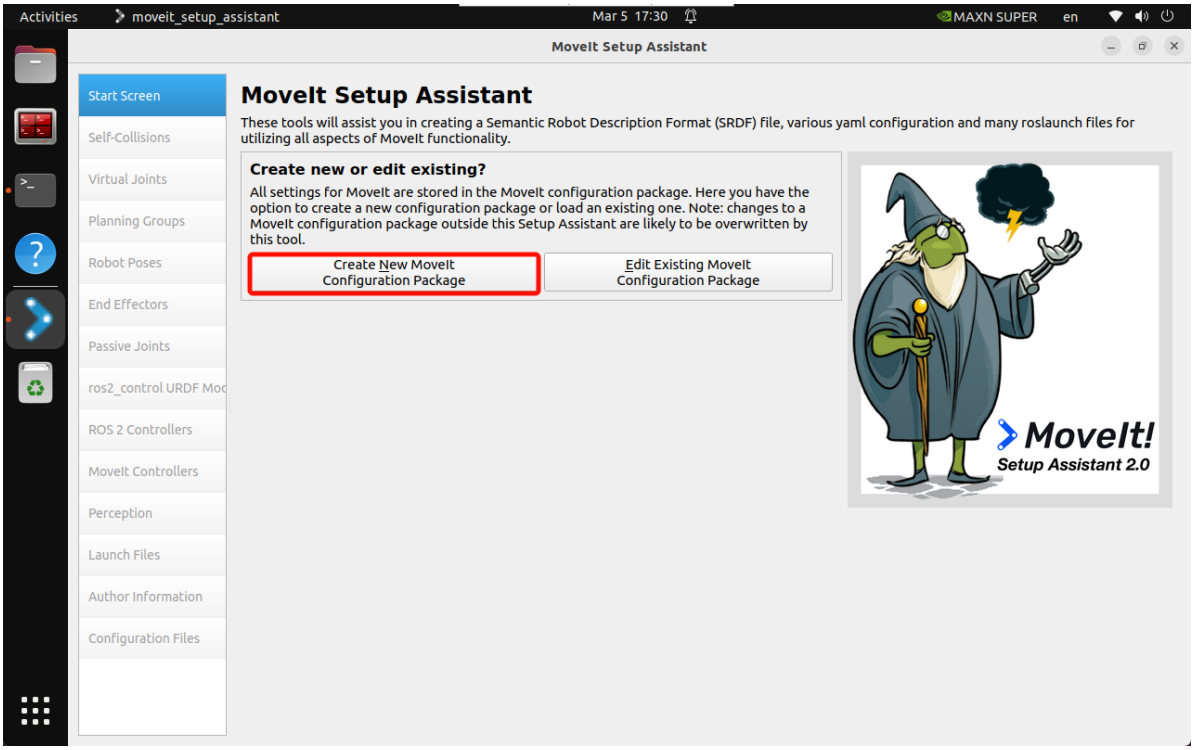
Through this tool, users can easily generate configuration files for robot motion planning and control without manually editing complex configuration files.

2.1. Start Assistant

Start MoveIt Setup Assistant:

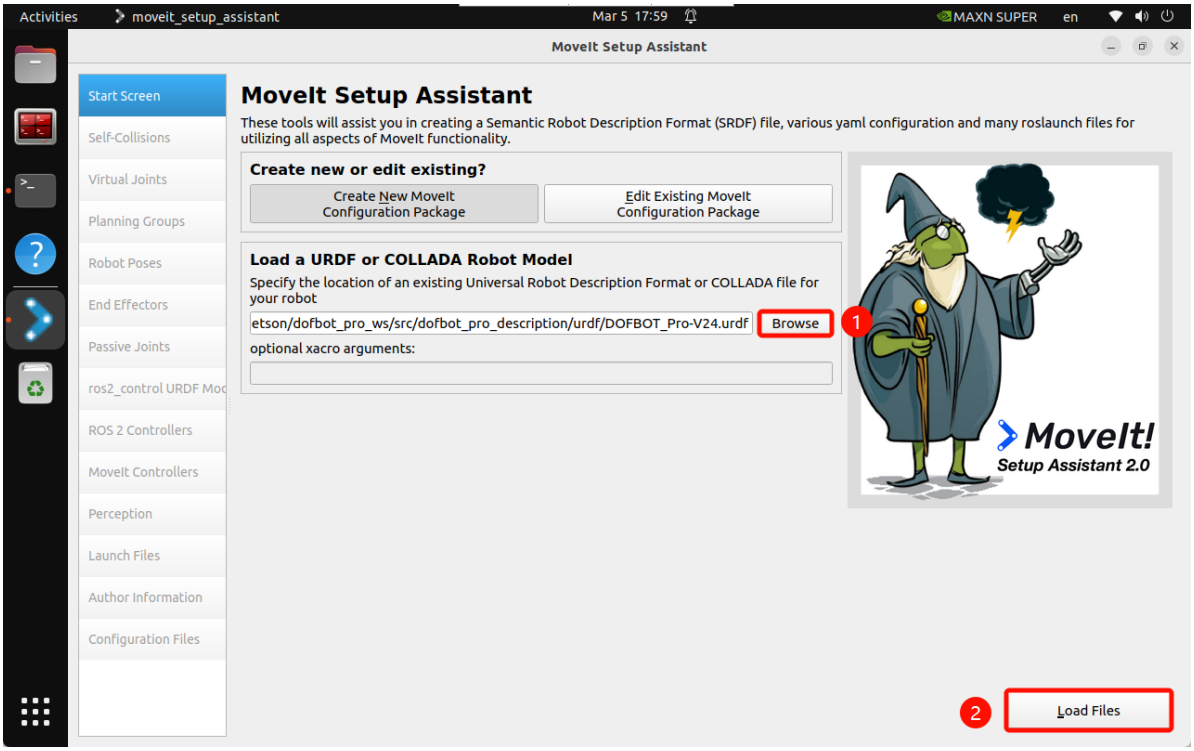
```
ros2 launch moveit_setup_assistant setup_assistant.launch.py
```

For first use, you can choose to create a new MoveIt configuration package. For later simple modifications, you can choose to edit an existing MoveIt package:

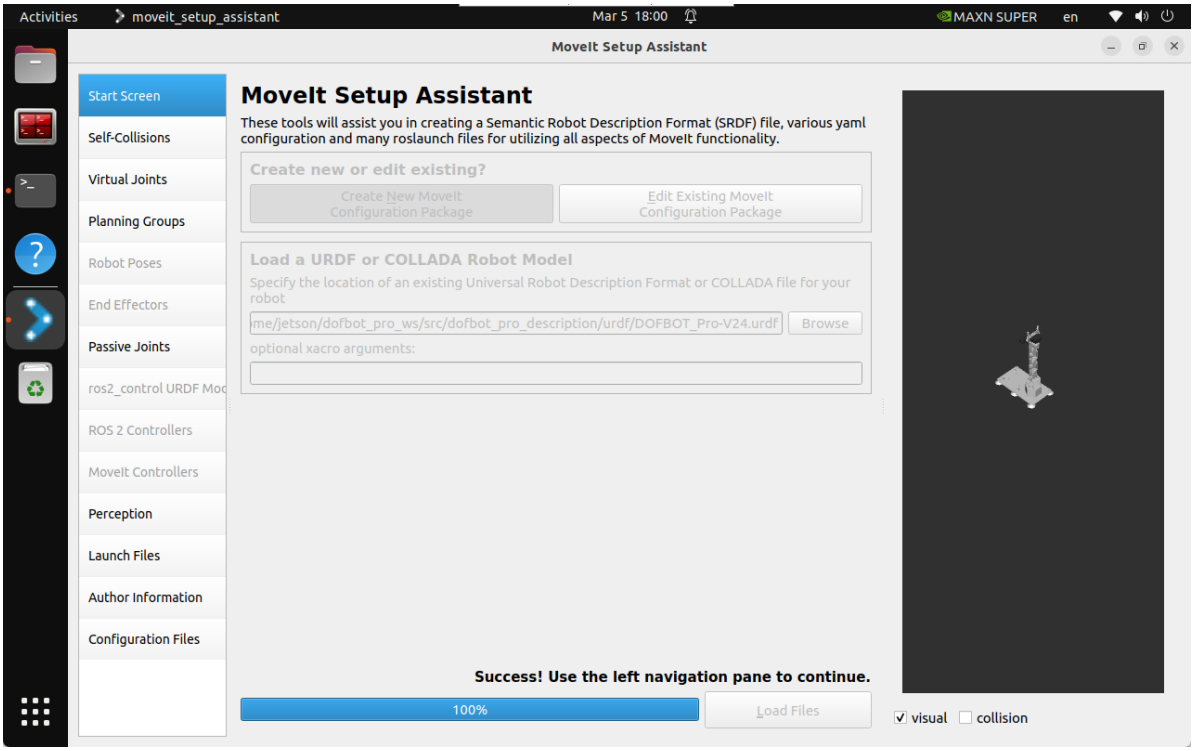


Creating a new MoveIt configuration package requires importing the robot's URDF model: Simply importing the URDF model file to MoveIt Assistant will cause an error. You need to use the compiled package (dofbot_pro_description) URDF file path

```
/home/jetson/dofbot_pro_ws/src/dofbot_pro_description/urdf/DOFBOT_Pro-V24.urdf
```



After selecting the URDF file, click Load Files, and MoveIt Assistant will display the robot model:

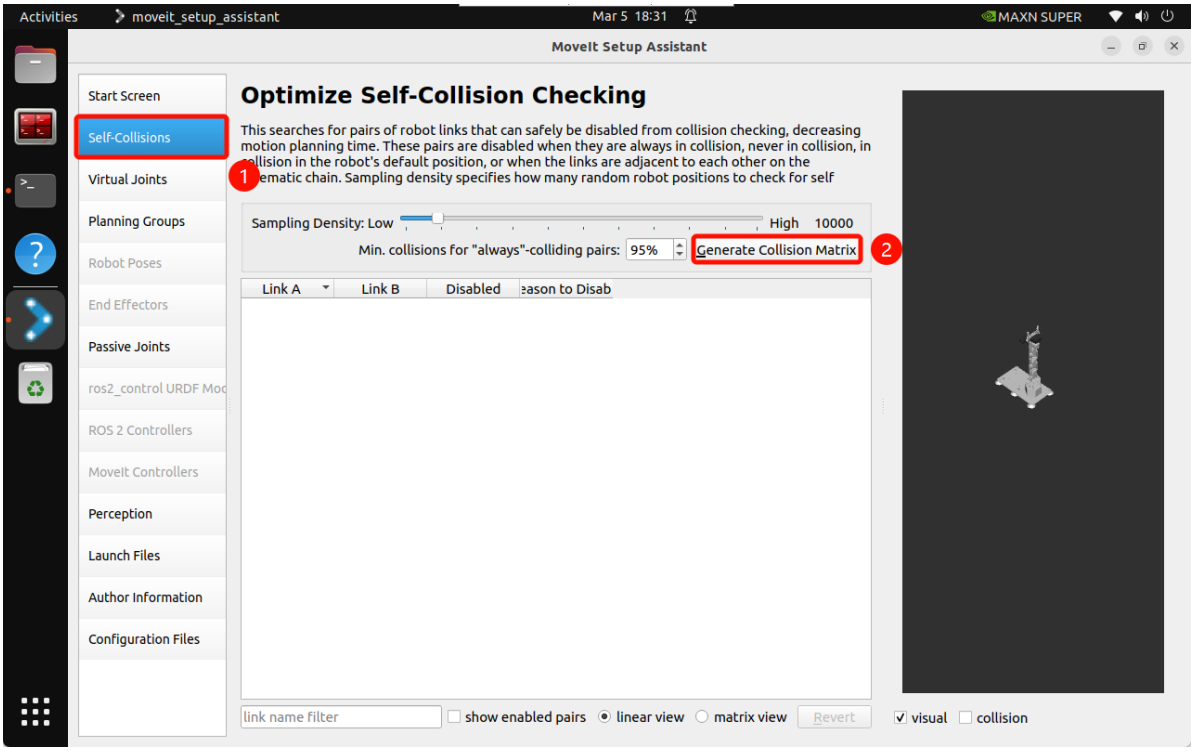


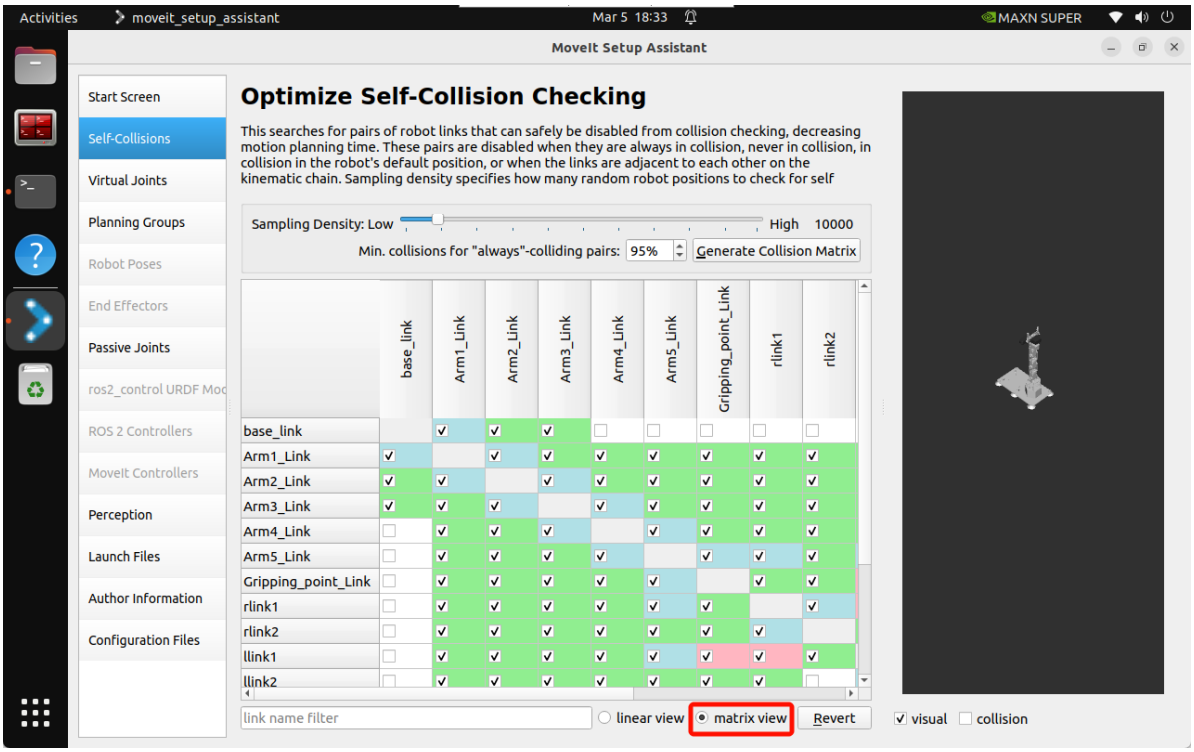
2.2. Collision Detection

The self-collision matrix is a feature used to optimize motion planning. Its main purpose is to generate a matrix that describes whether various links in the robot model may collide with each other.

This matrix can help the motion planner avoid unnecessary collision detection during path planning, thereby improving planning efficiency.

To generate the self-collision matrix, you can click the auto-generate option in MoveIt Setup Assistant:





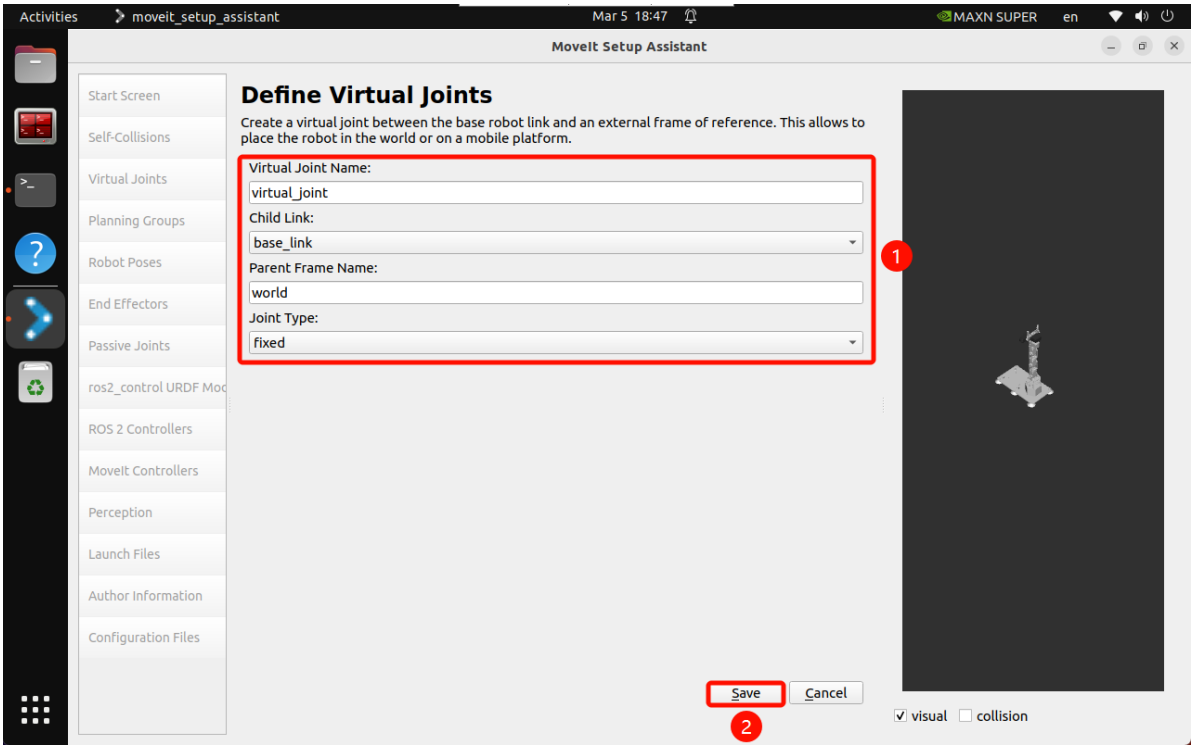
2.3. Virtual Joints

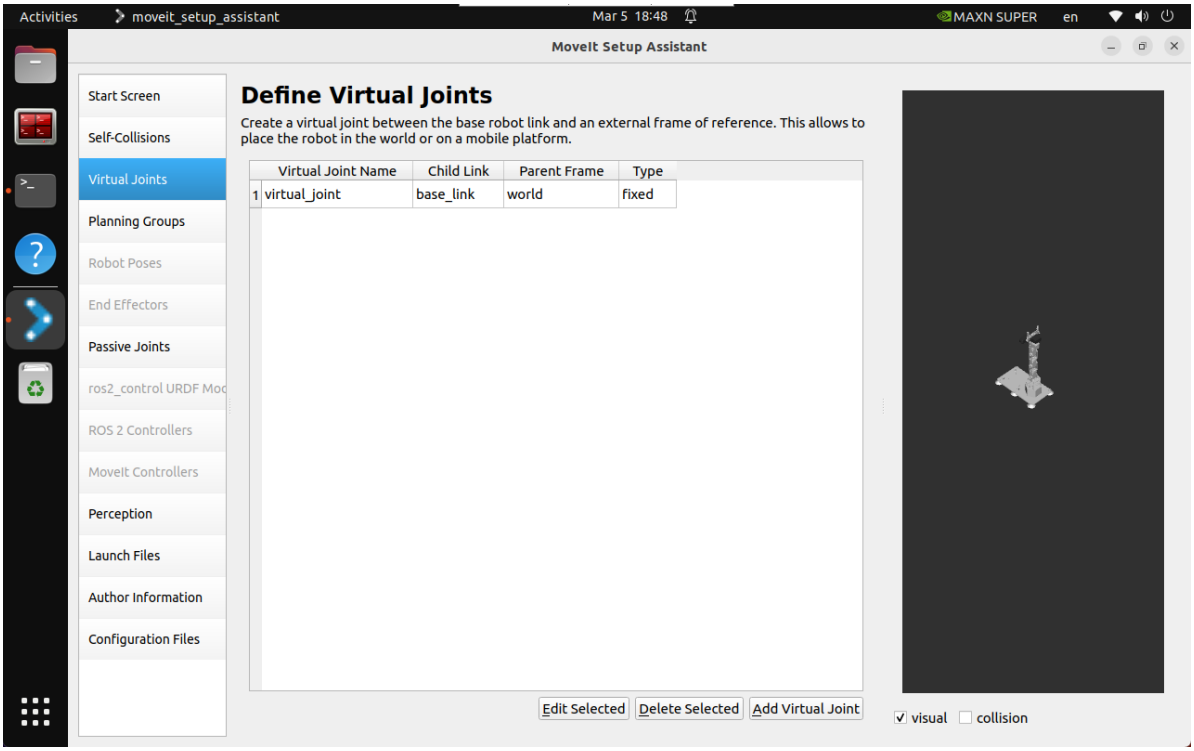
Virtual joints are used to define the relationship between the robot model and the external world.

Virtual joints are typically used to describe the connection between the robot base and a fixed reference frame (such as the world coordinate system).

Fix the robot base link `base_link` and `world`:

- Virtual Joint Name: virtual_joint
- Child Link: base_link
- Parent Frame Name: world
- Joint Type: fixed

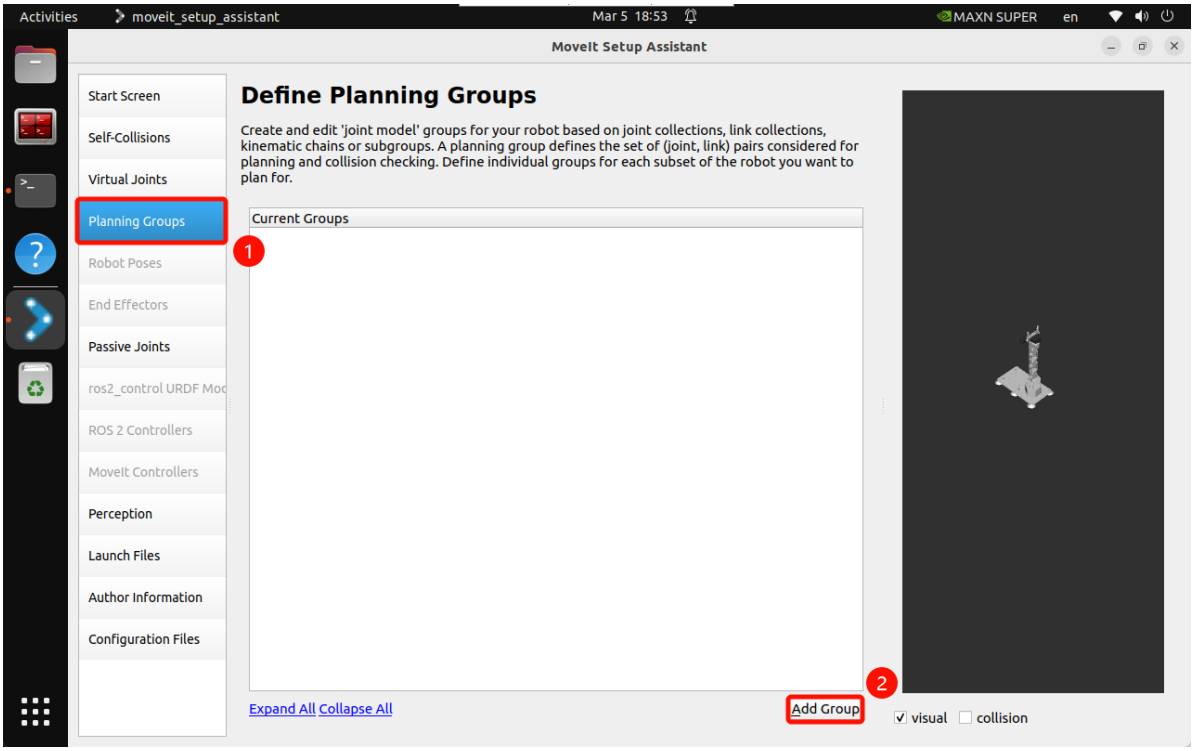




2.4. Motion Planning Groups

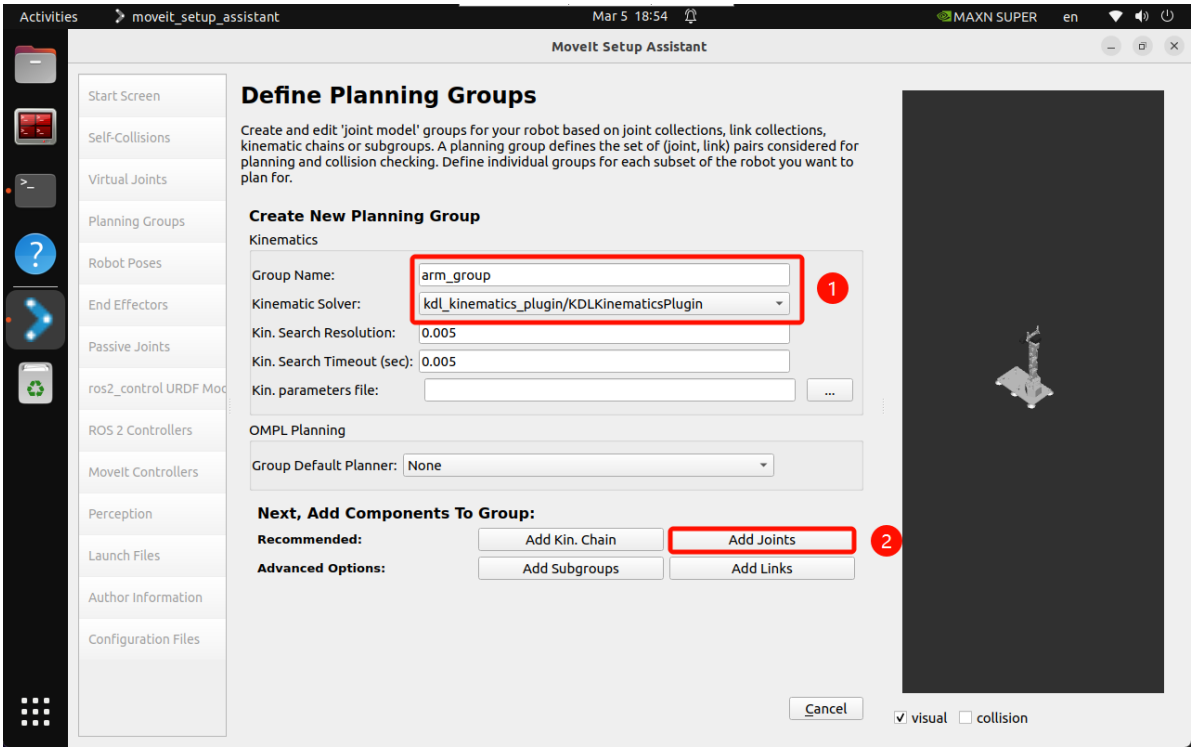
Planning groups are key steps in configuring robot motion planning.

Planning groups define which joints and links in the robot can move together and how their motion is planned.



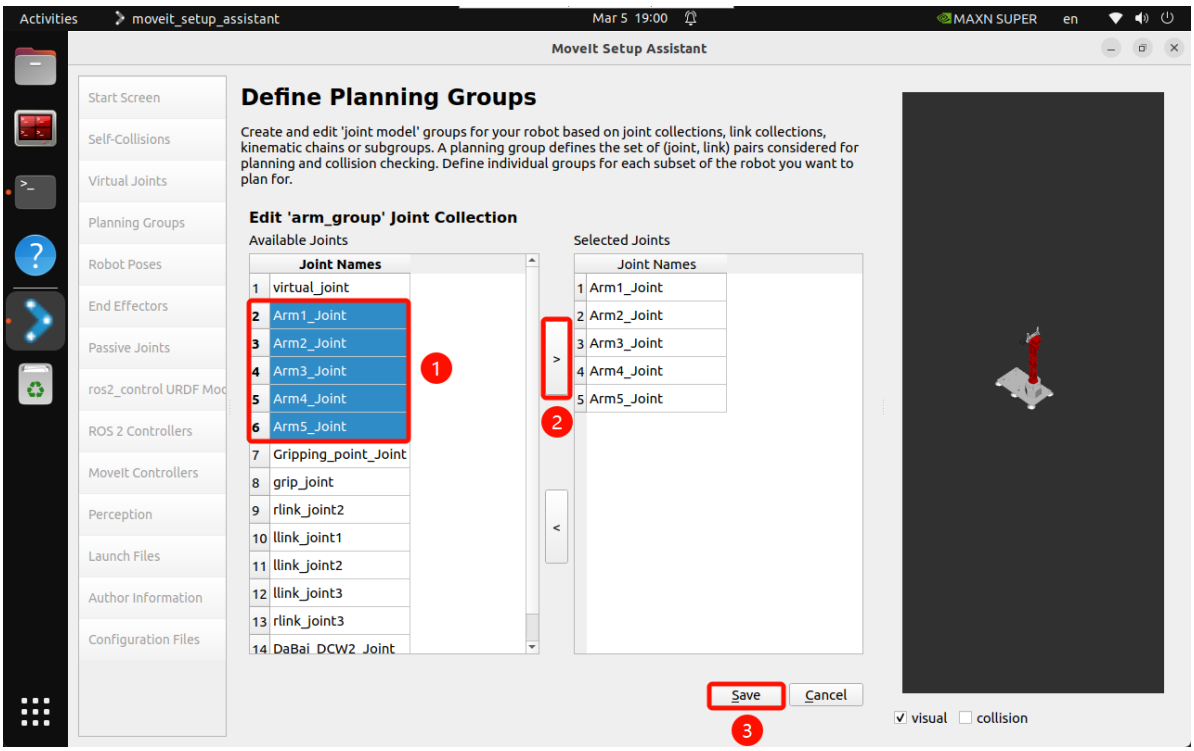
Plan robotic arm group: arm_group

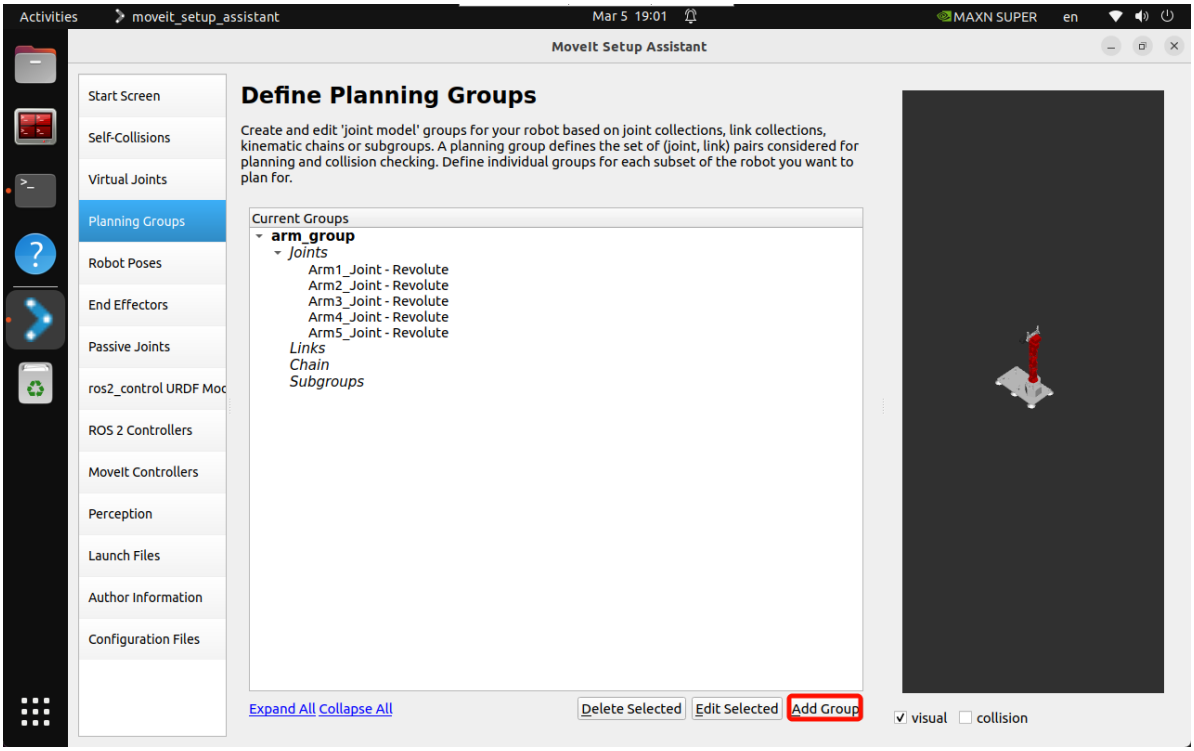
- Group Name：arm_group
- Kinematic Solver：kdl_kinematics_plugin/KDLKinematicsPlugin
- Kin. Search Resolution：0.005
- Kin. Search Timeout：0.005



- Add Joints：Select the robotic arm joints

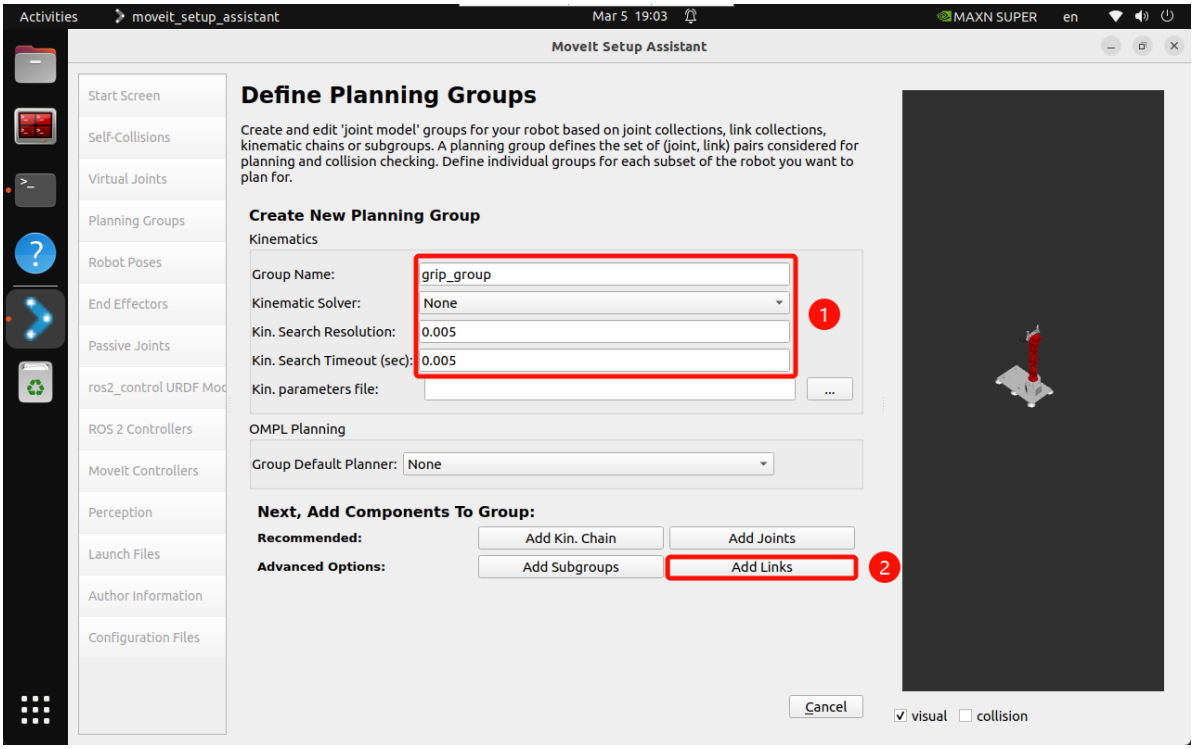
Joints：Arm1_Joint、Arm2_Joint、Arm3_Joint、Arm4_Joint、Arm5_Joint





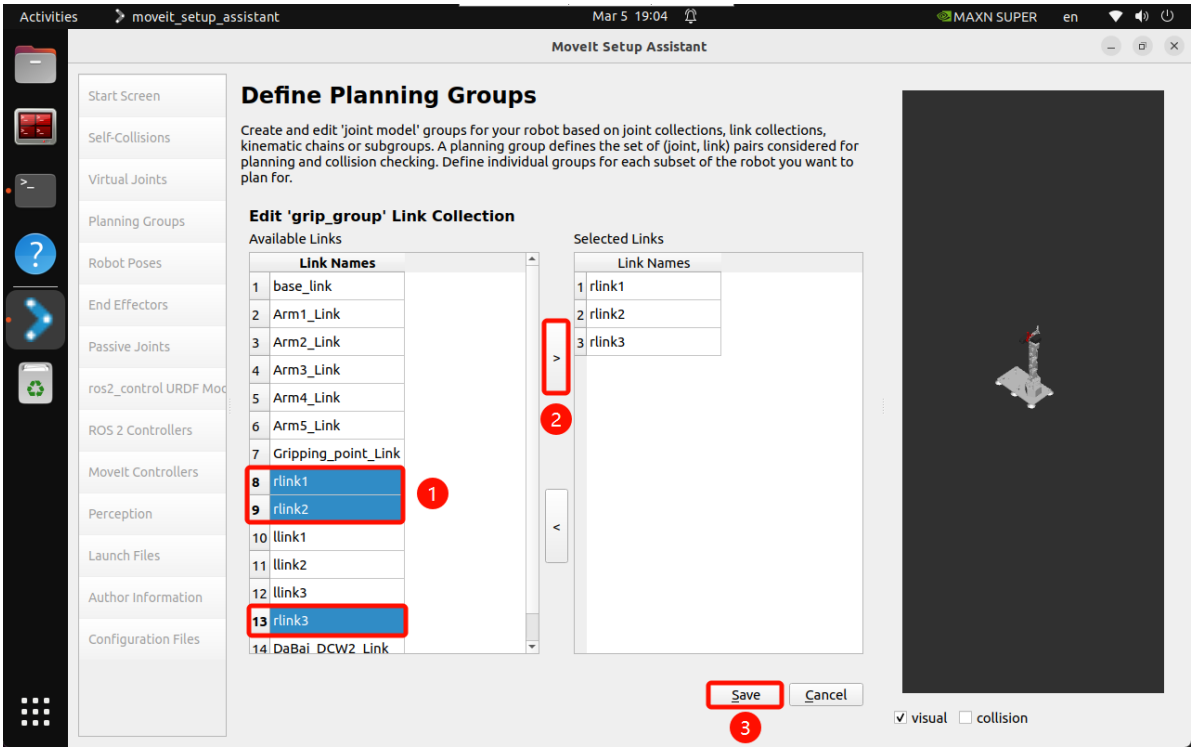
Plan gripper group: grip_group

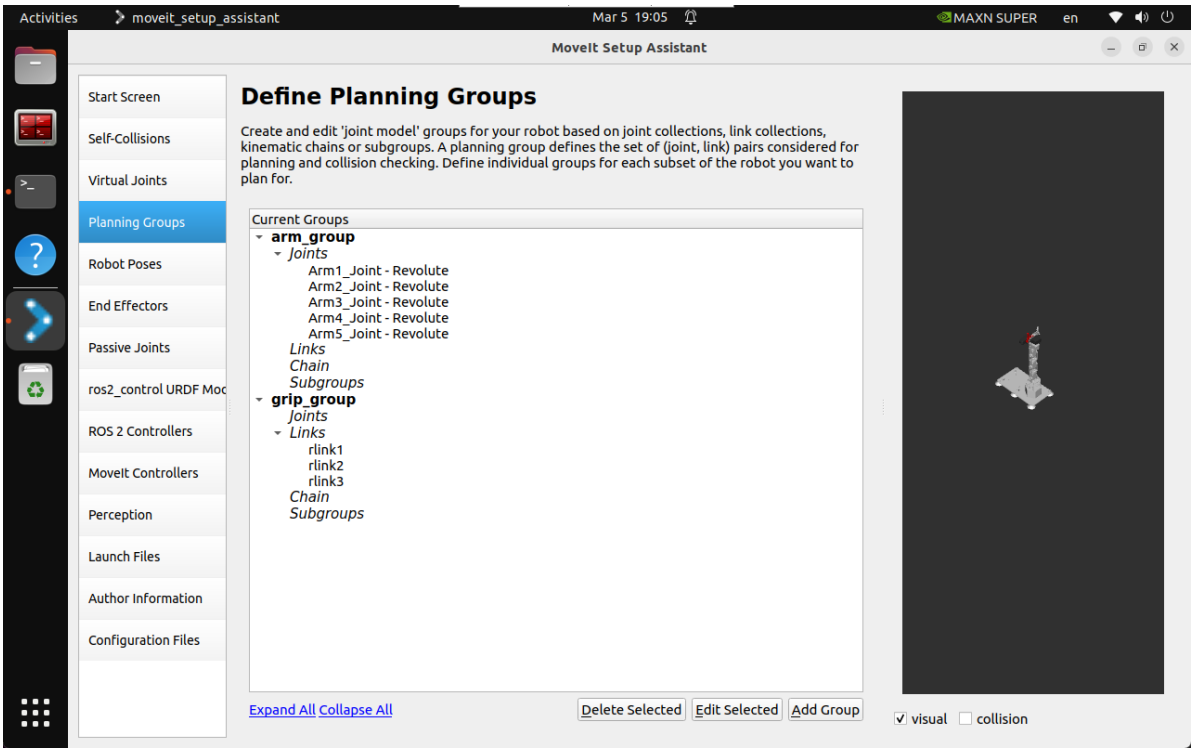
- Group Name：grip_group
- Kinematic Solver：None
- Kin. Search Resolution：0.005
- Kin. Search Timeout：0.005



- Add Links：Select the gripper links

Links：rlink1、rlink2、rlink3

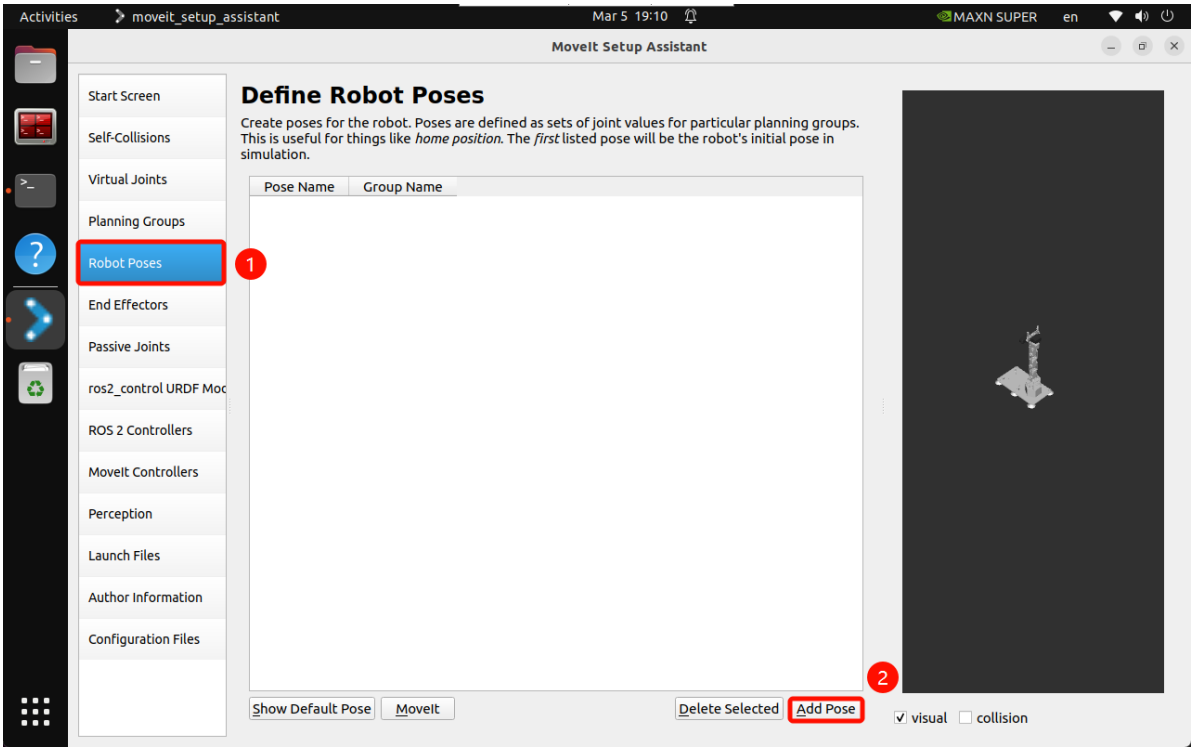




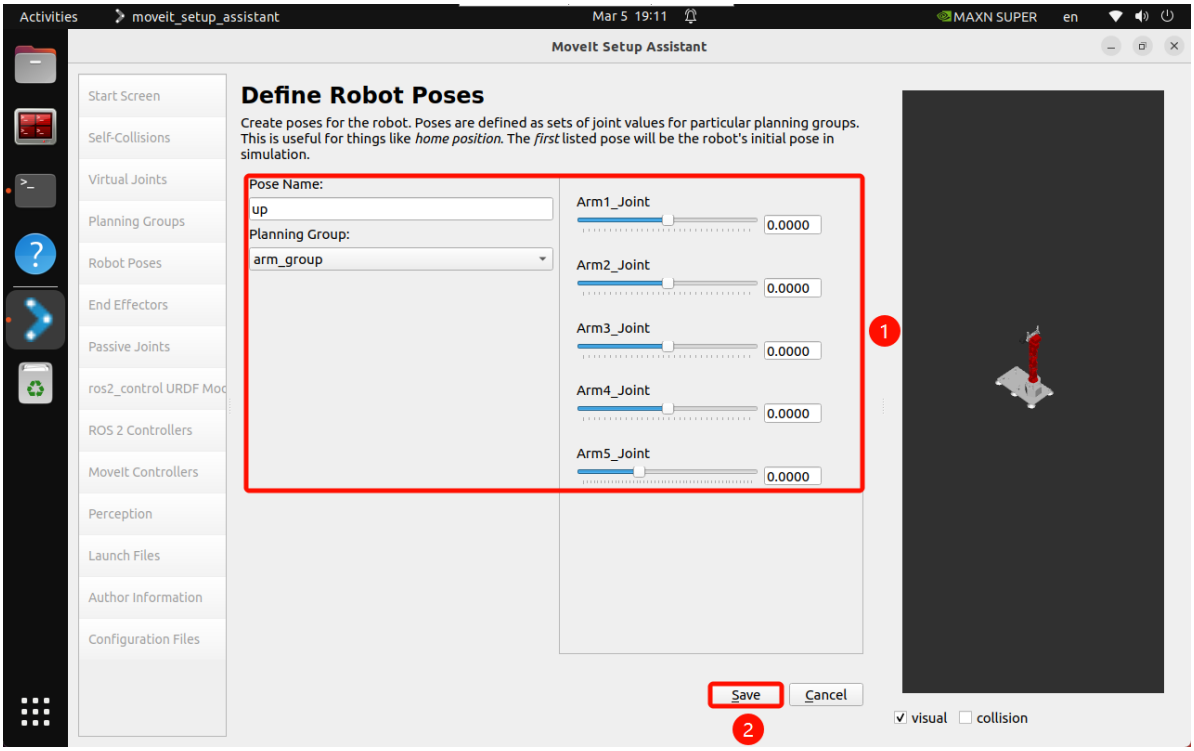
2.5. Robot Poses

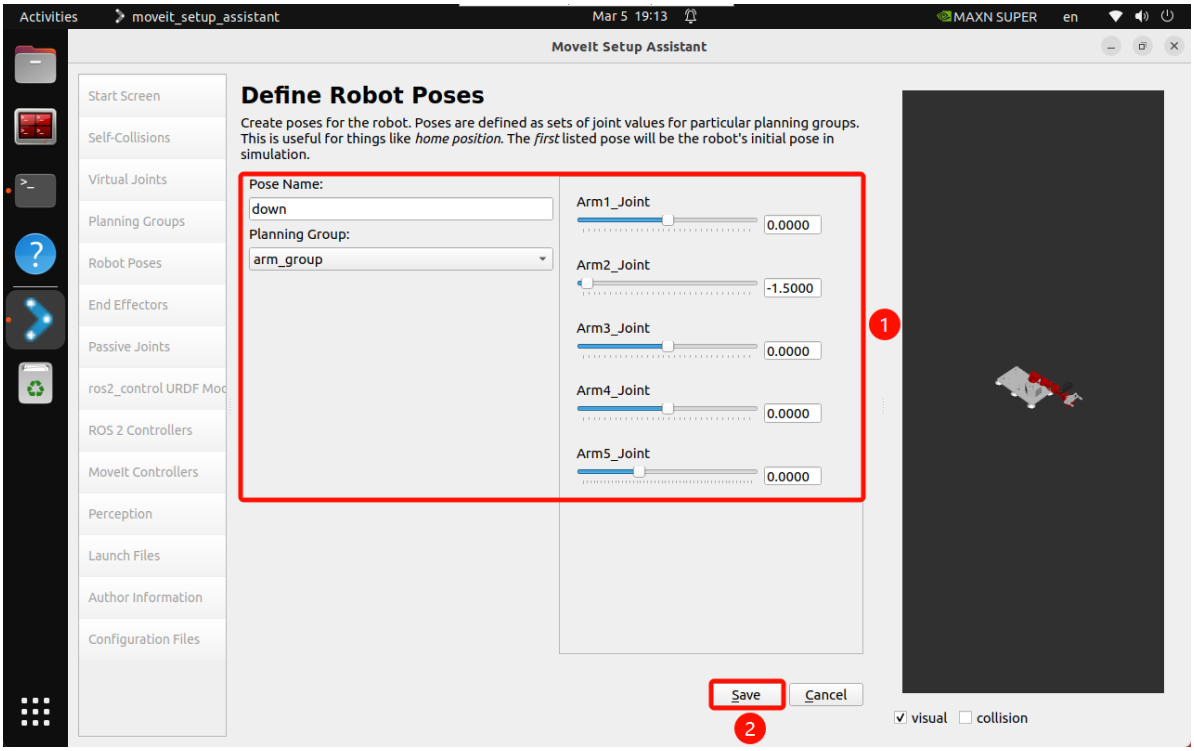
> ### Robotic Arm Positions

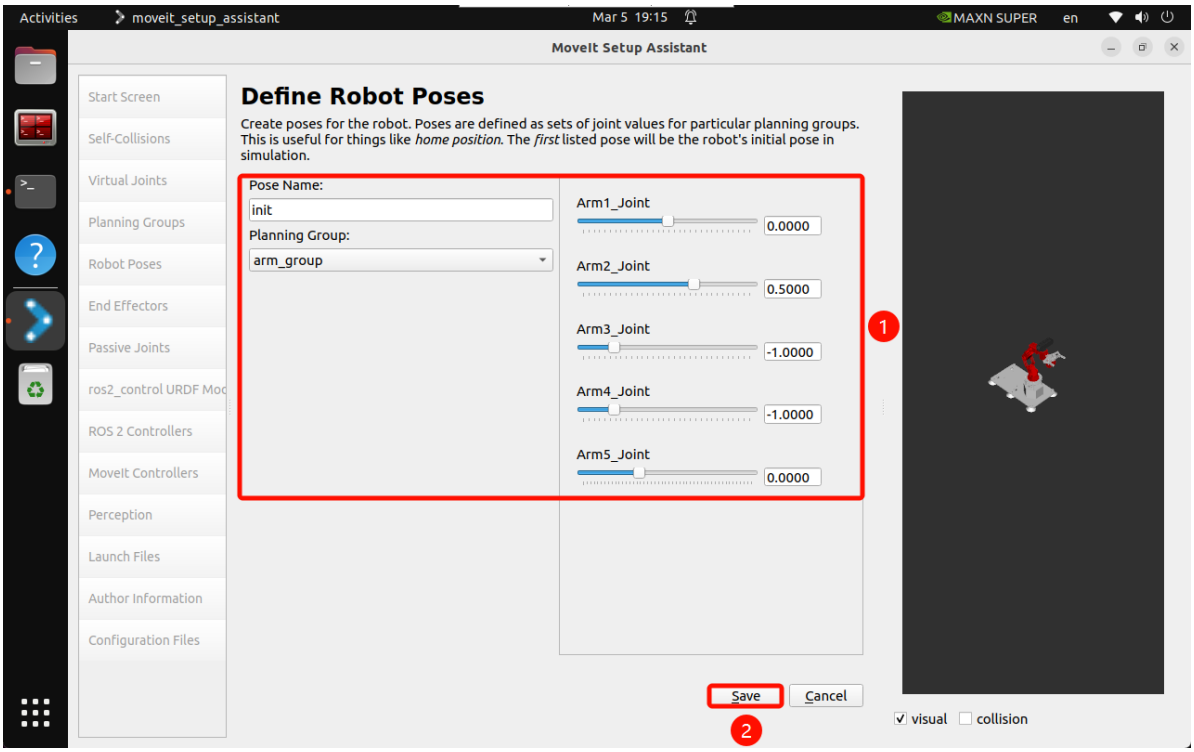
Set three groups of robotic arm predefined poses to the robot configuration:



You can manually move the sliders to set each joint state: up、down、init

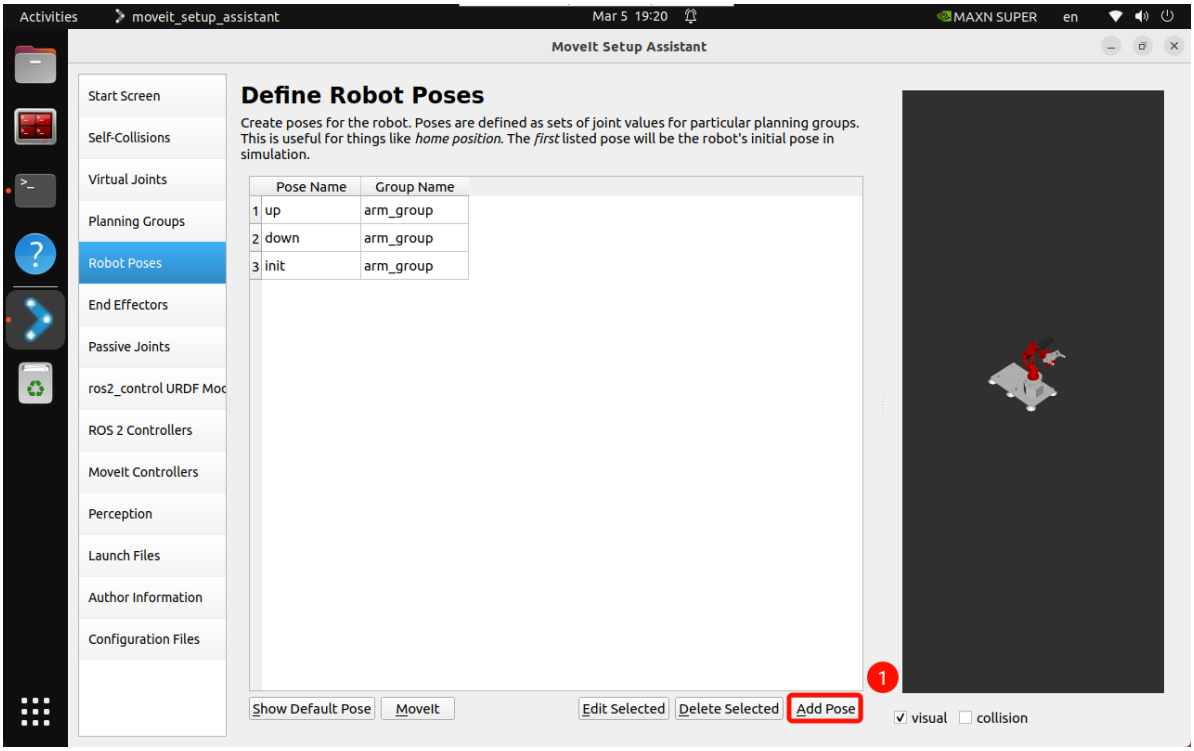




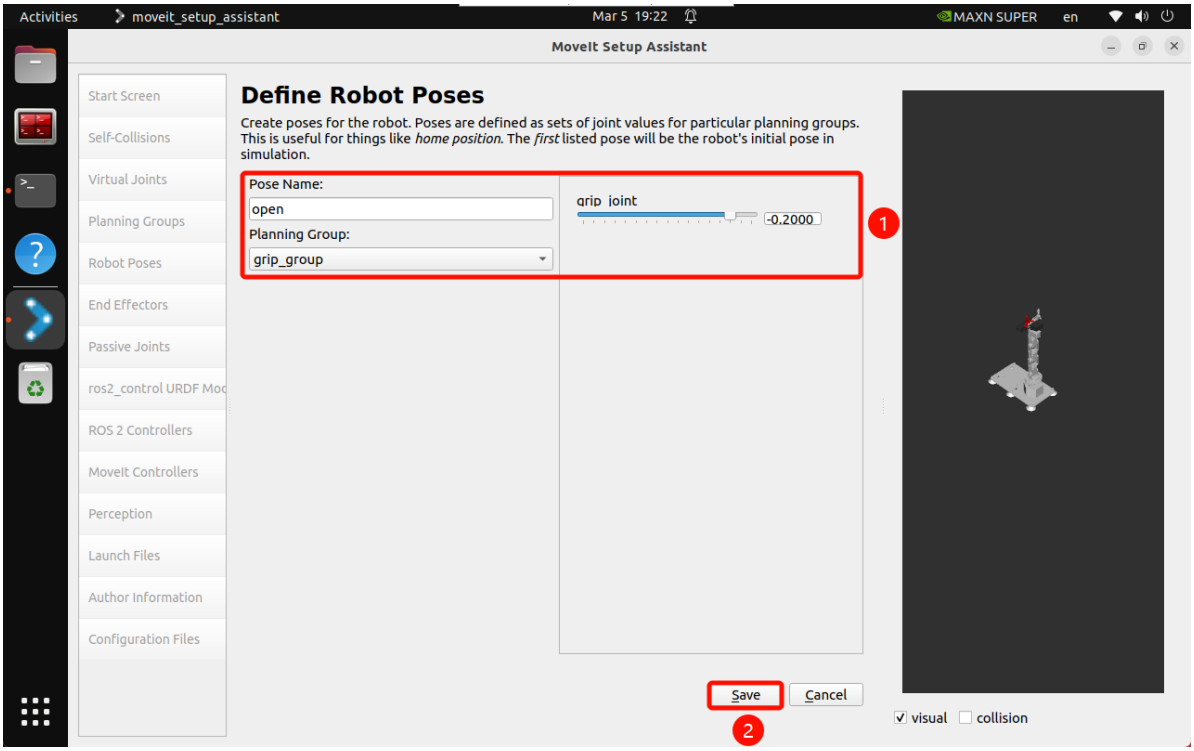


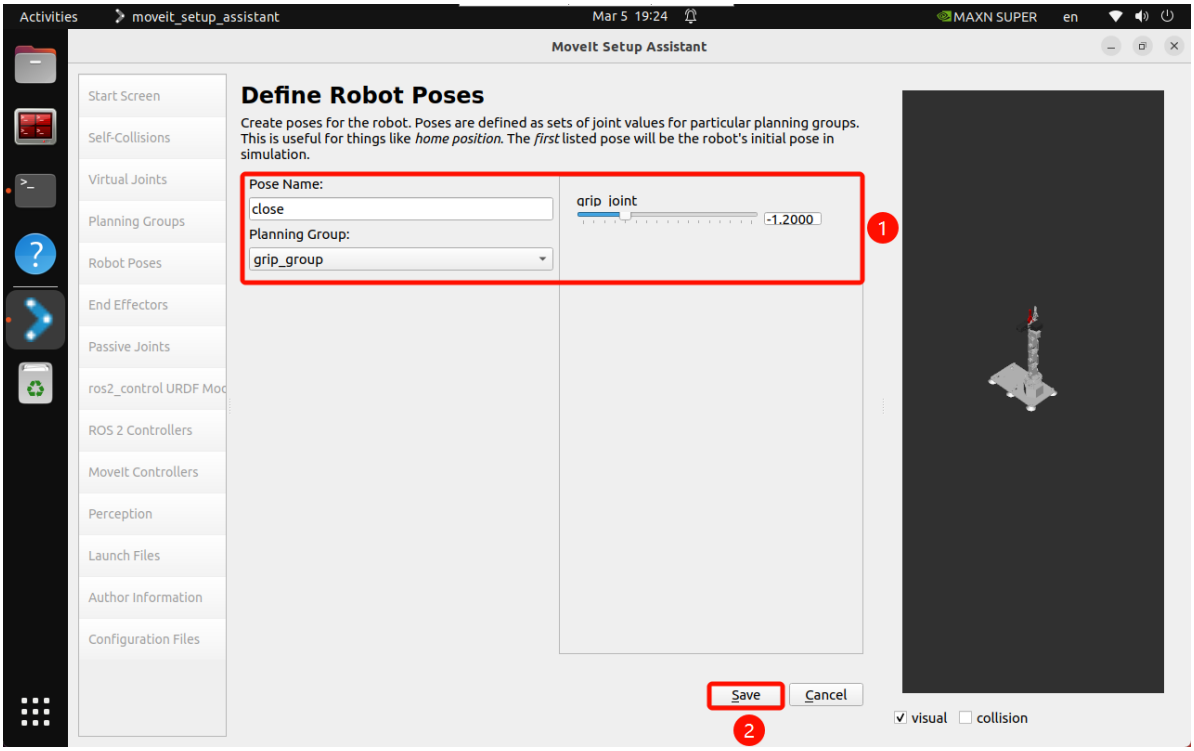
Gripper Positions

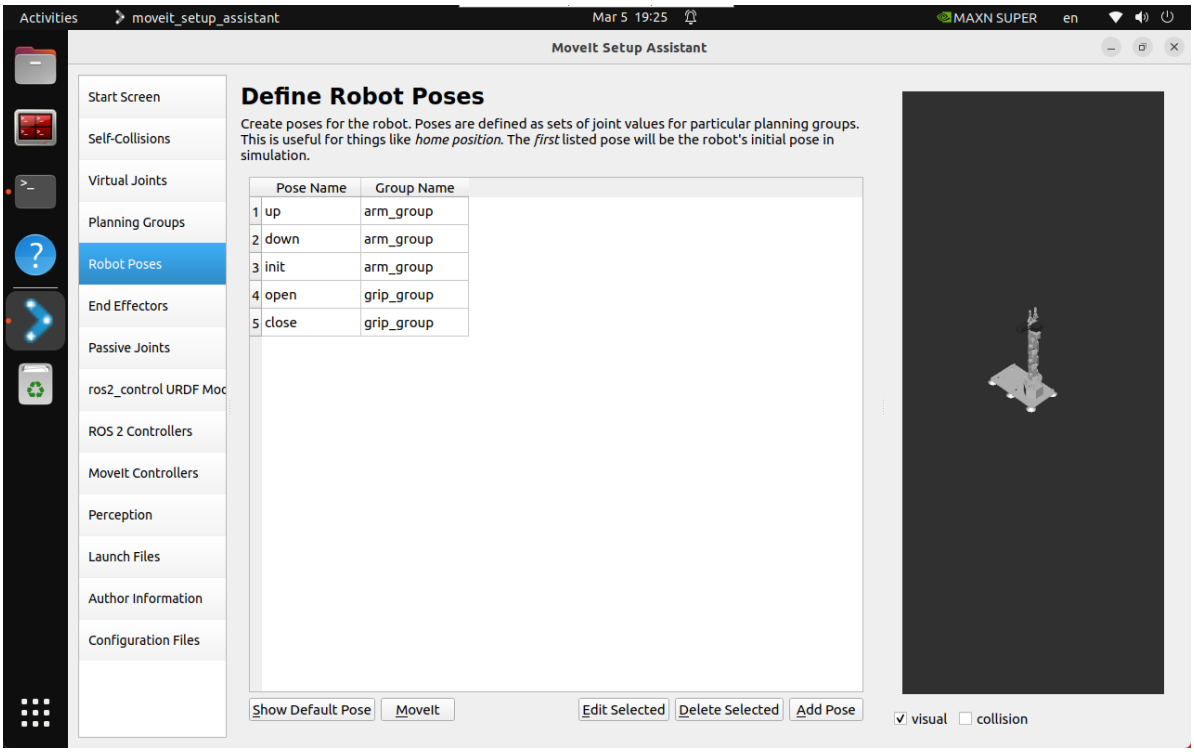
Set two groups of gripper predefined poses to the robot configuration:



You can manually move the sliders to set gripper state: open、close

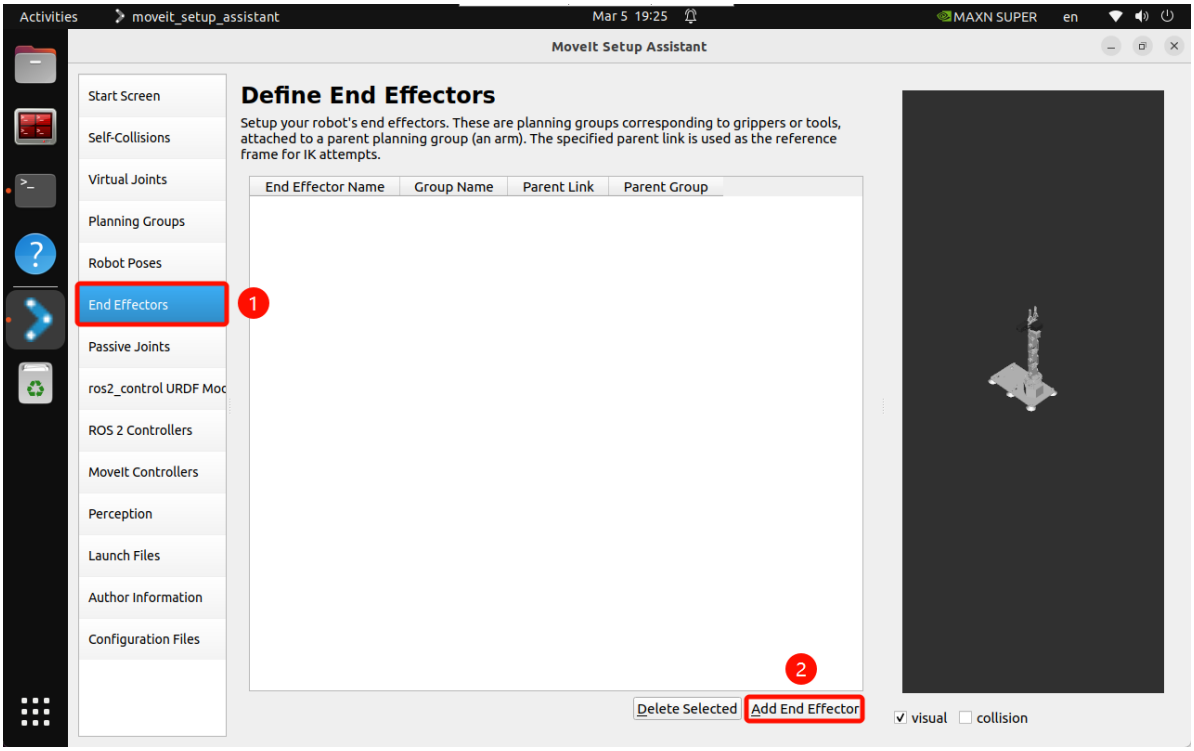






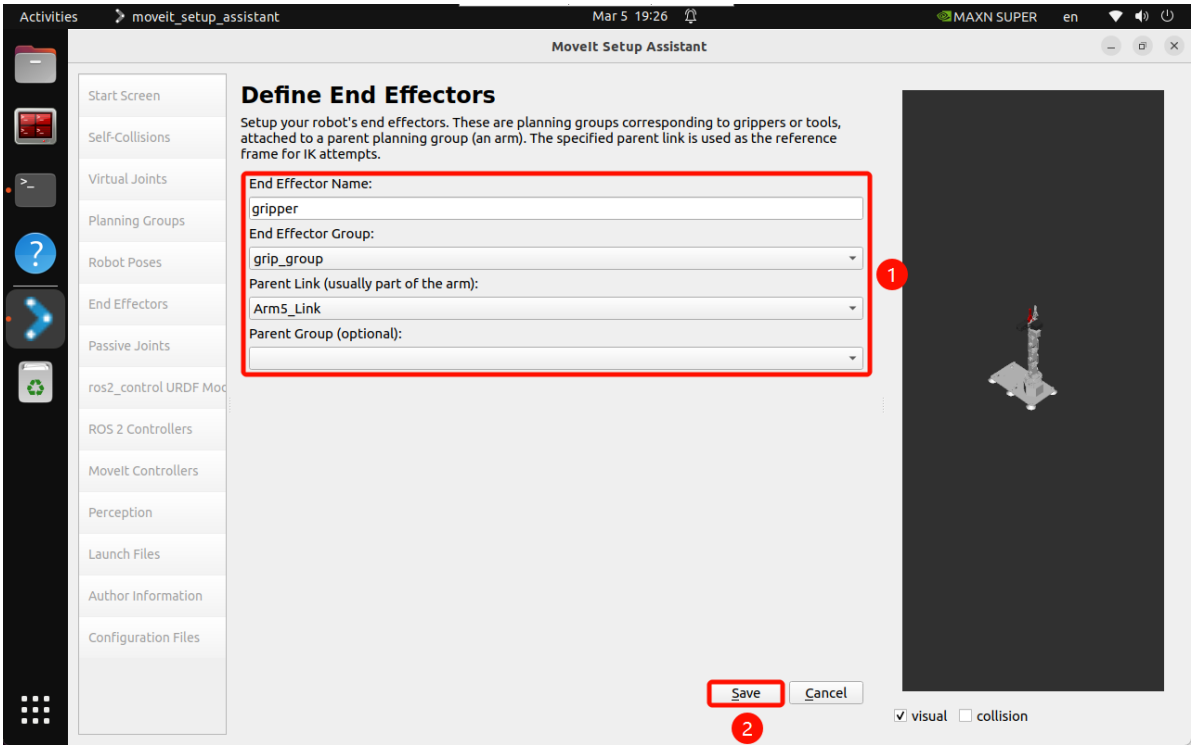
2.6. End Effector

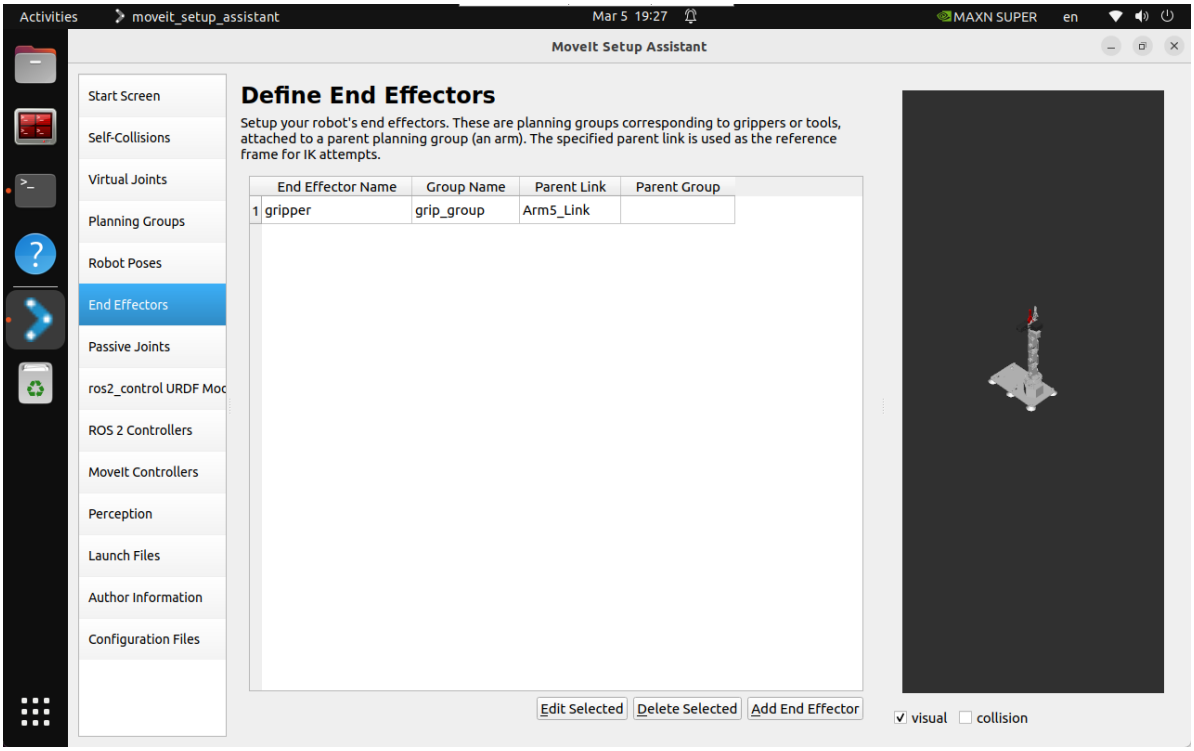
Set the gripper group as the end effector:



End effector: gripper

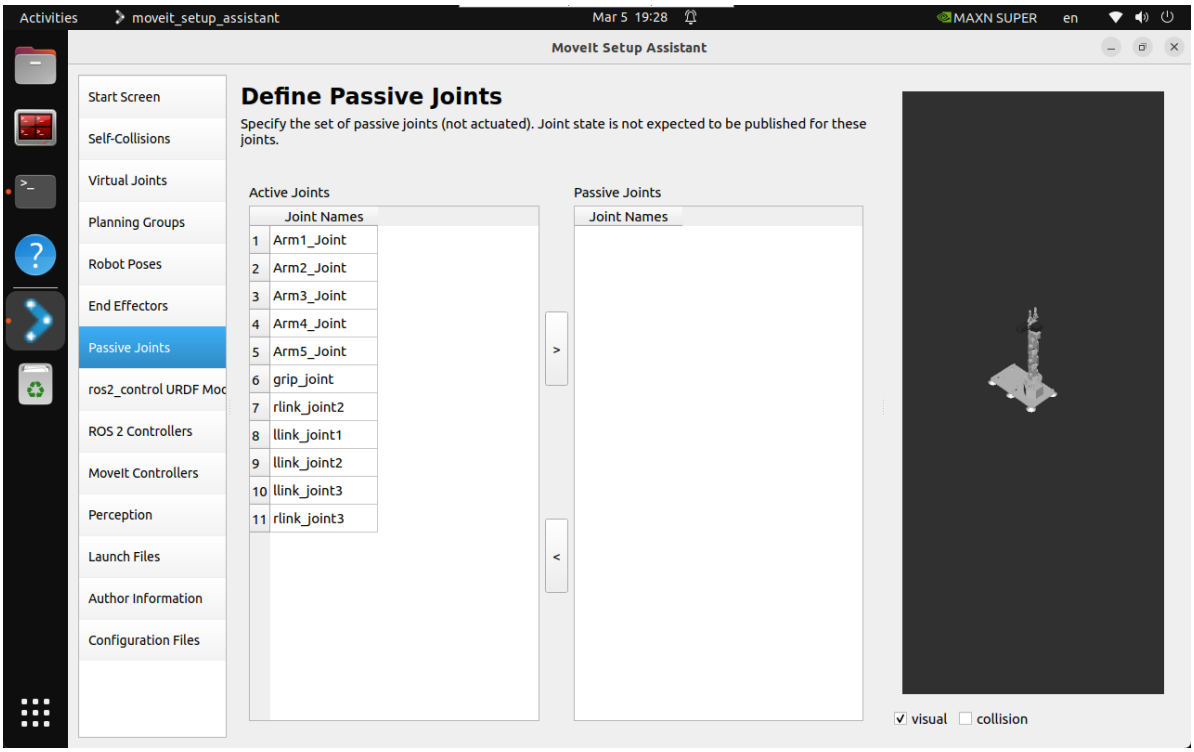
- End Effector Name：gripper
- End Effector Group：gripp_group
- Parent Link：Arm5_Link
- Parent Group：None





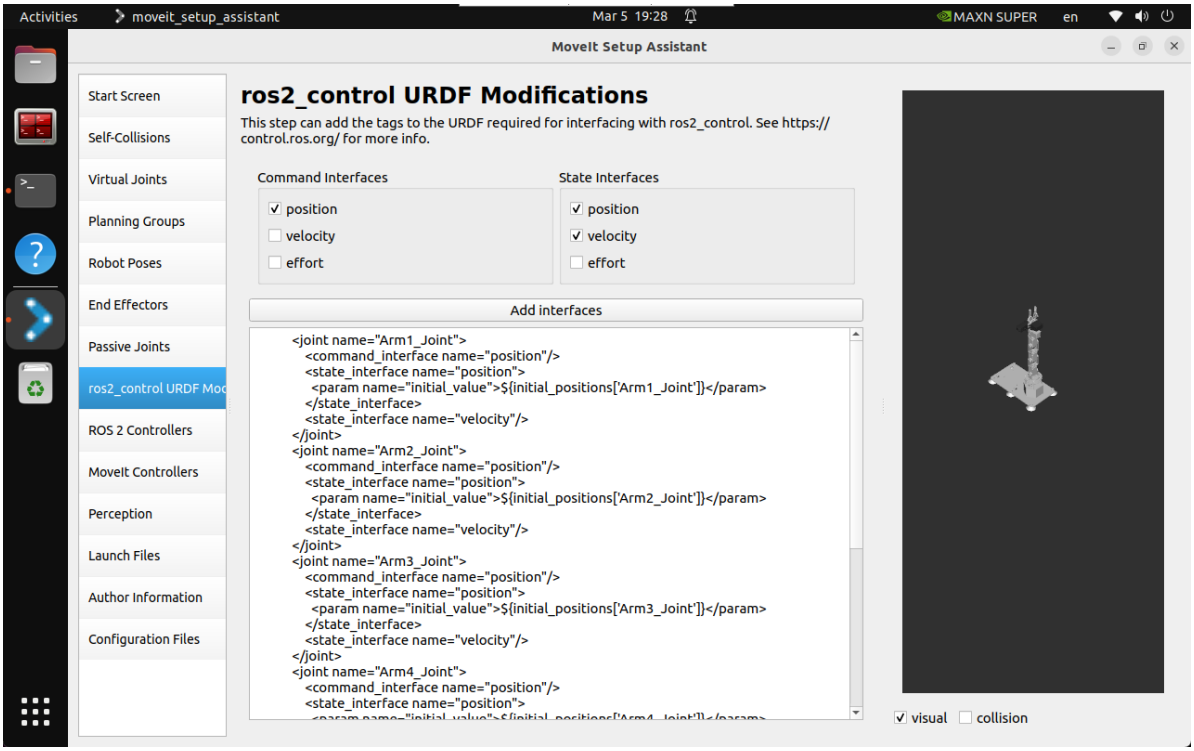
2.7. Passive Joints

Passive joints are non-driven joints that cannot be directly controlled: The robotic arm has no passive joints, so skip this step.



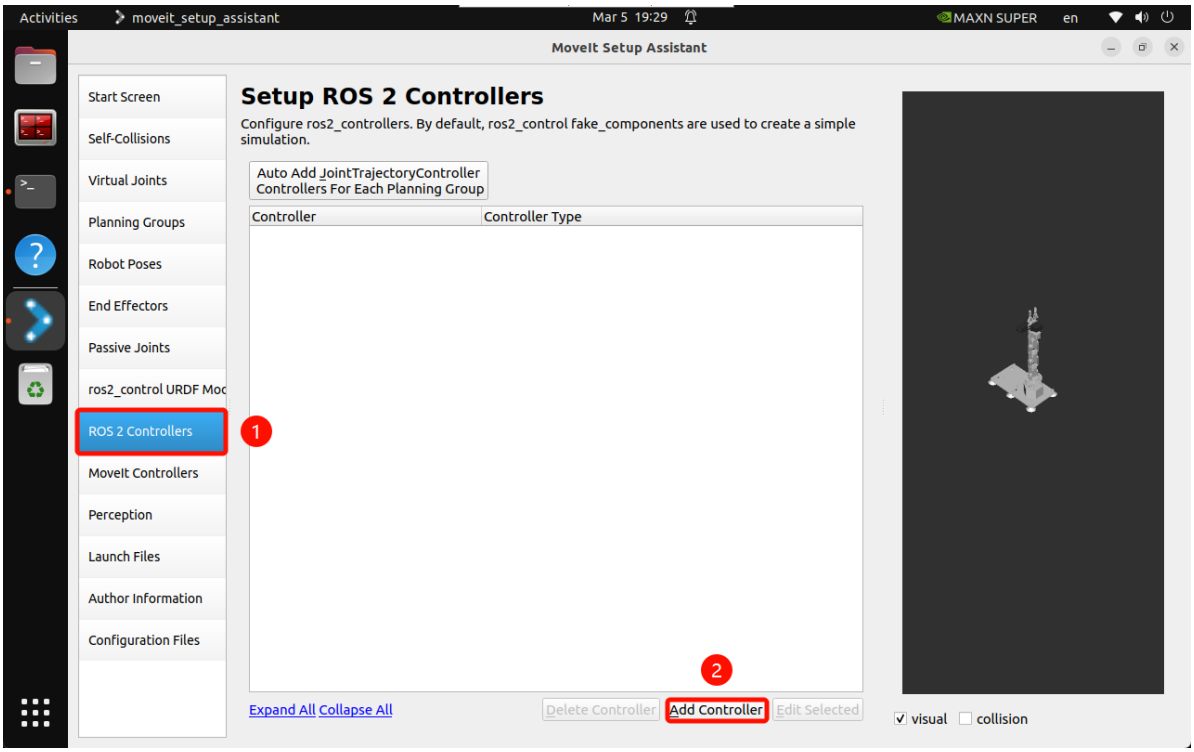
2.8. URDF Files

MoveIt Setup Assistant will automatically set command interfaces and state interfaces for each joint: Use the default generated information.



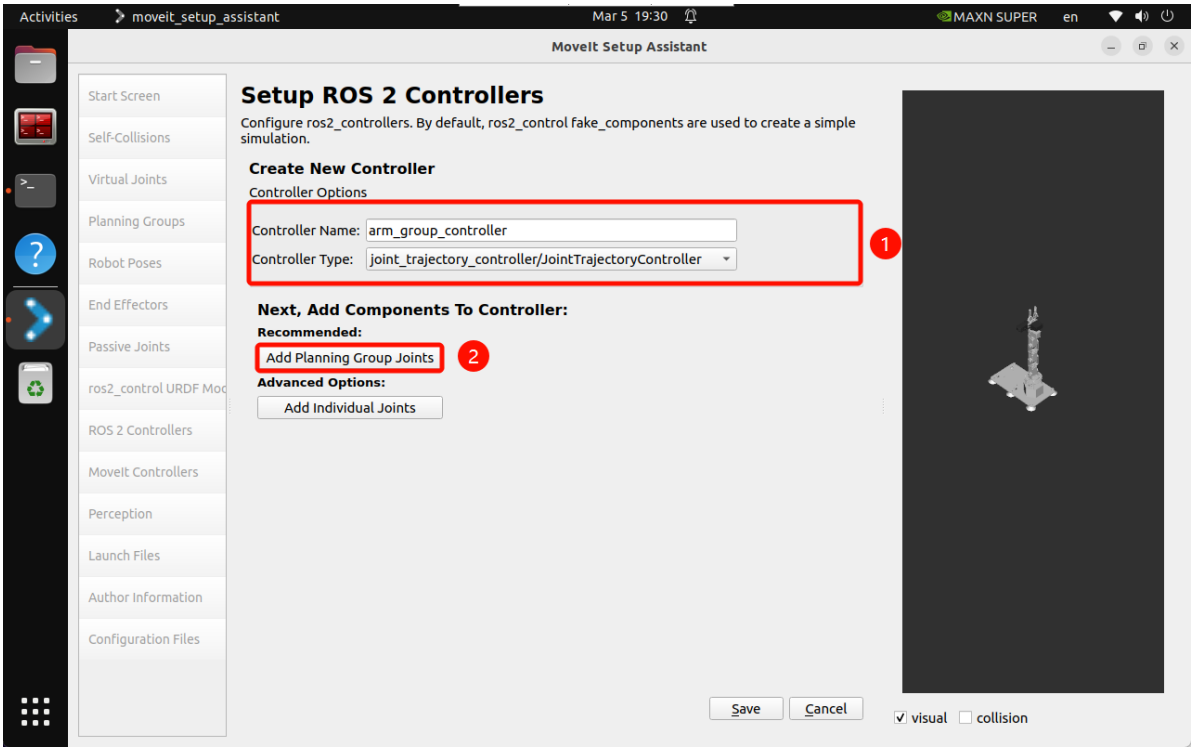
2.9. ROS2 Controllers

ROS2 Controllers is a framework for robot real-time control, which can be used to automatically generate simulation controllers to drive robot joints.

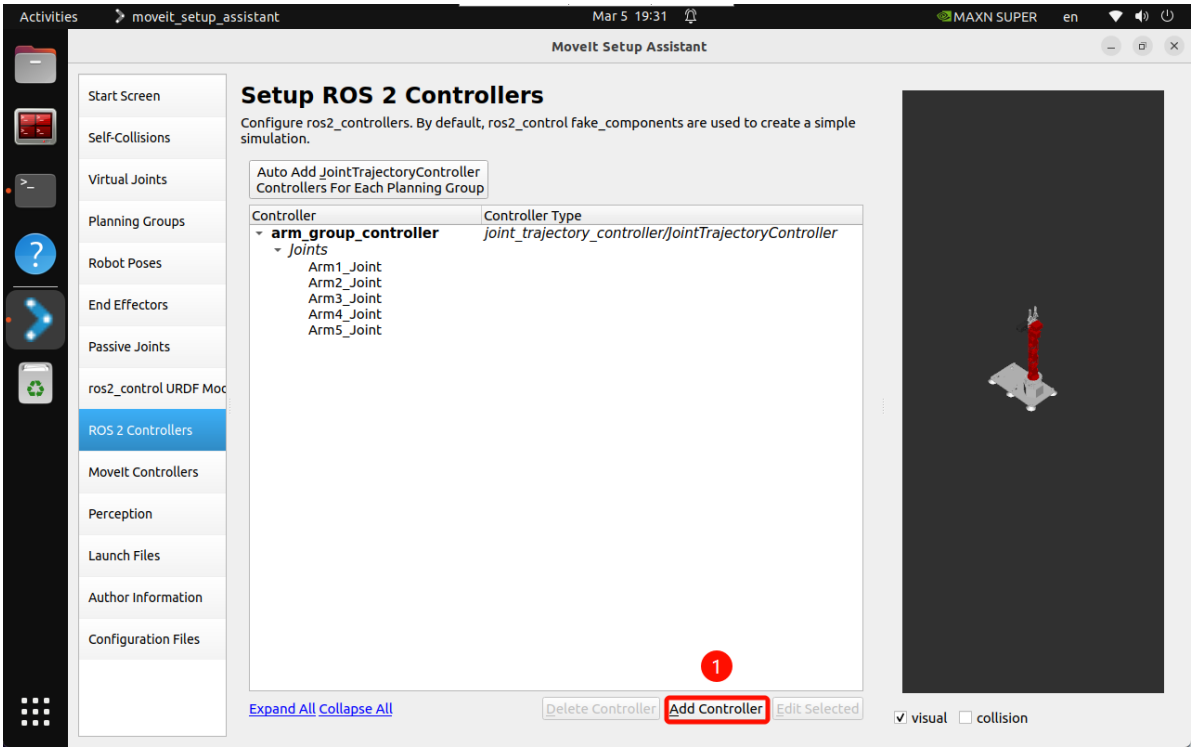


Add robotic arm controller:

- Controller Name：arm_group_controller
- Controller Type：joint_trajectory_controller/JointTrajectoryController
- Add Planning Group Joints：arm_group

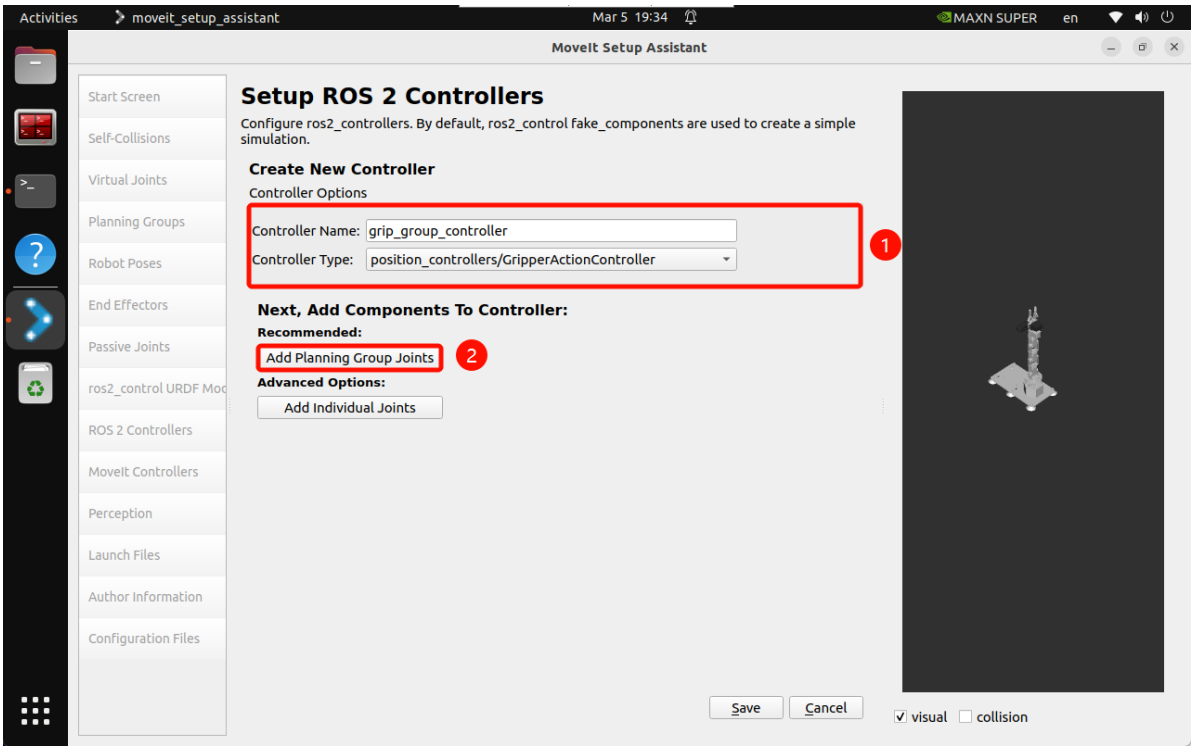

Activities
moveit_setup_assistant
Mar 5 19:30
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Setup ROS 2 Controllers
Configure ros2_controllers. By default, ros2_control fake_components are used to create a simple simulation.
Create New Controller
Controller Options
Controller Name: arm_group_controller
Controller Type: joint_trajectory_controller/JointTrajectoryController
1
Next, Add Components To Controller:
Recommended:
Add Planning Group Joints
2
Advanced Options:
Add Individual Joints
Save
Cancel
visual
collision

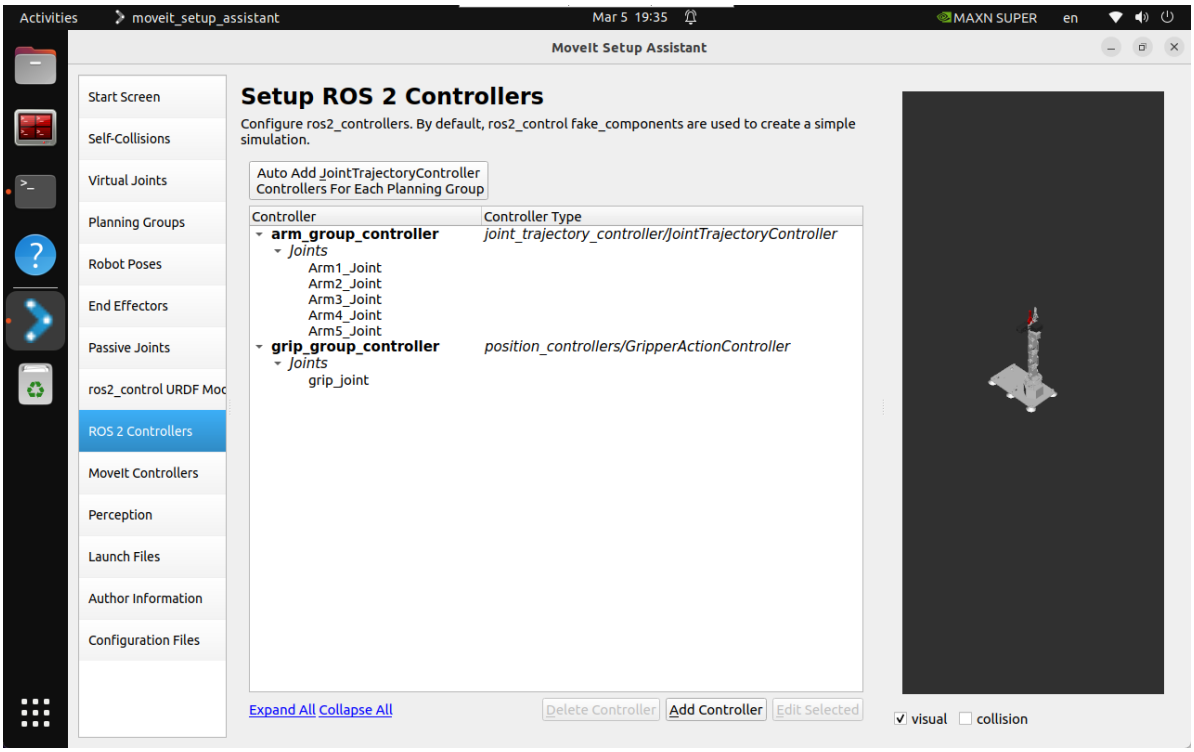

Activities
moveit_setup_assistant
Mar 5 19:31
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Setup ROS 2 Controllers
Configure ros2_controllers. By default, ros2_control fake_components are used to create a simple simulation.
Edit 'arm_group_controller' Joints groups collection
Available Groups
Group Names
1 arm_group
2 grip_group
1
>
2
<
Selected Groups
Group Names
1 arm_group
Save
Cancel
3
visual
collision



Add gripper controller:

- Controller Name：grip_group_controller
- Controller Type：position_controllers/GripperActionController
- Add Planning Group Joints：grip_group

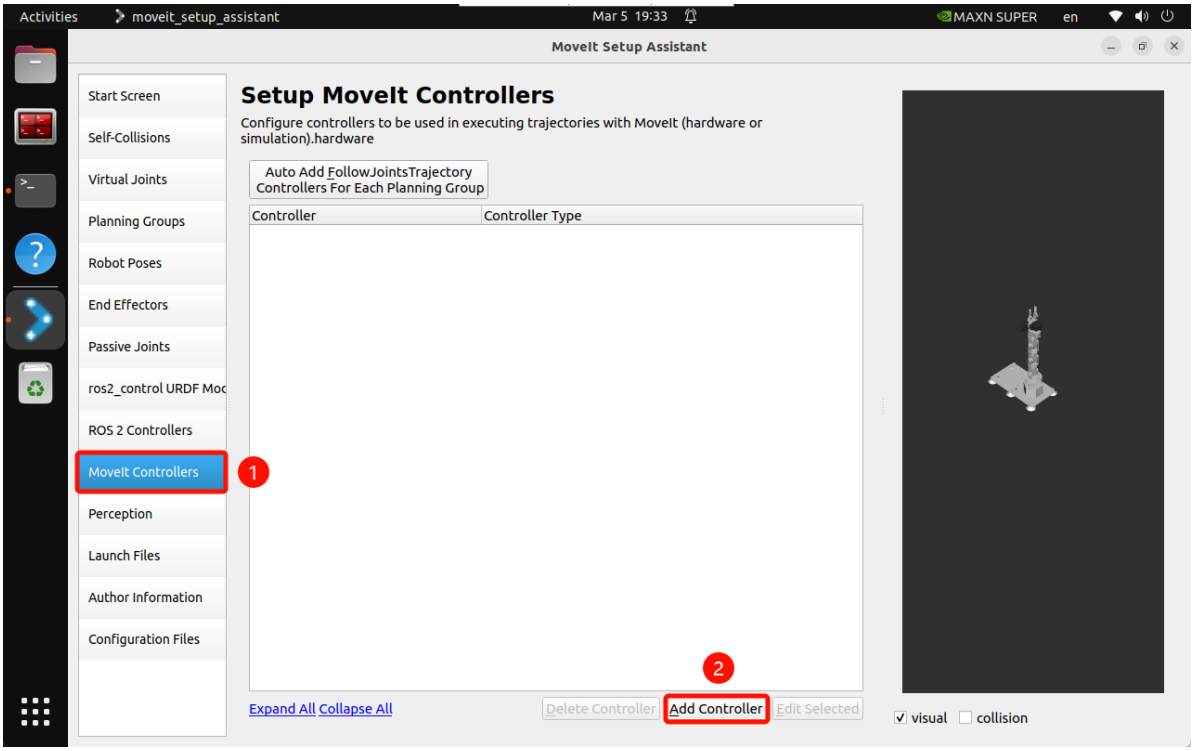




2.10. MoveIt Controllers

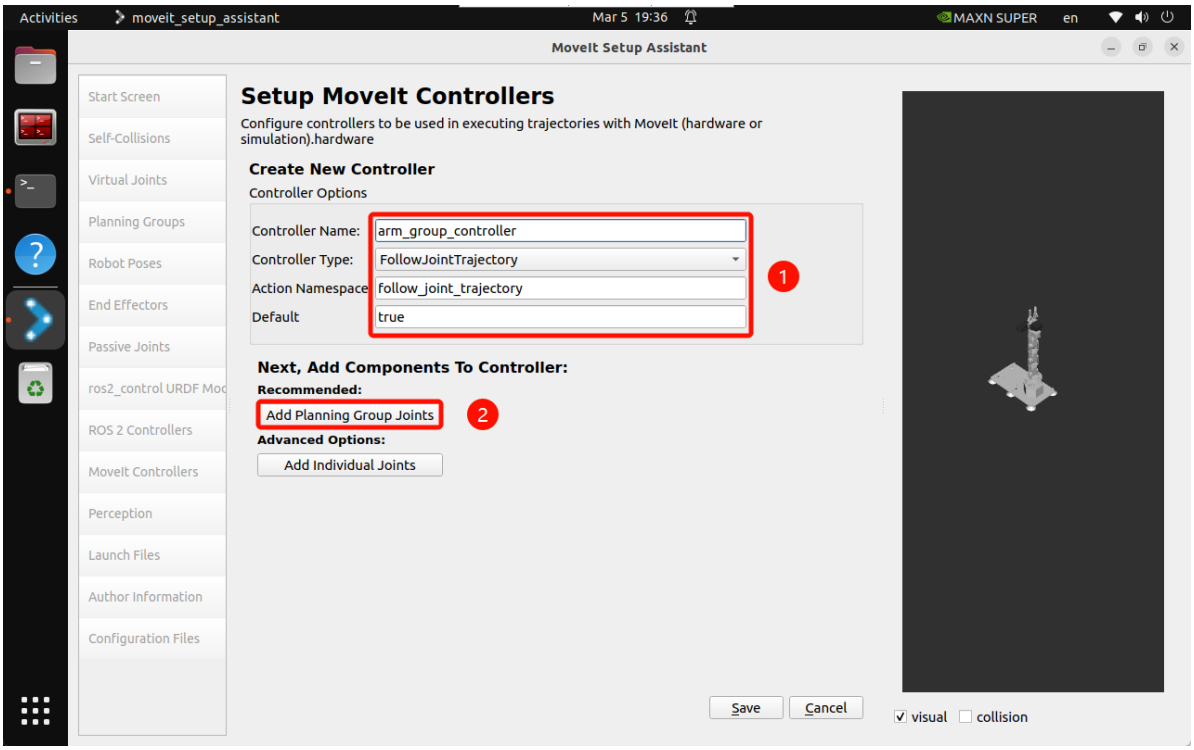
MoveIt requires trajectory controllers with the FollowJointTrajectoryAction interface to execute planned trajectories. This interface sends the generated trajectories to the robot's ROS2 controllers.

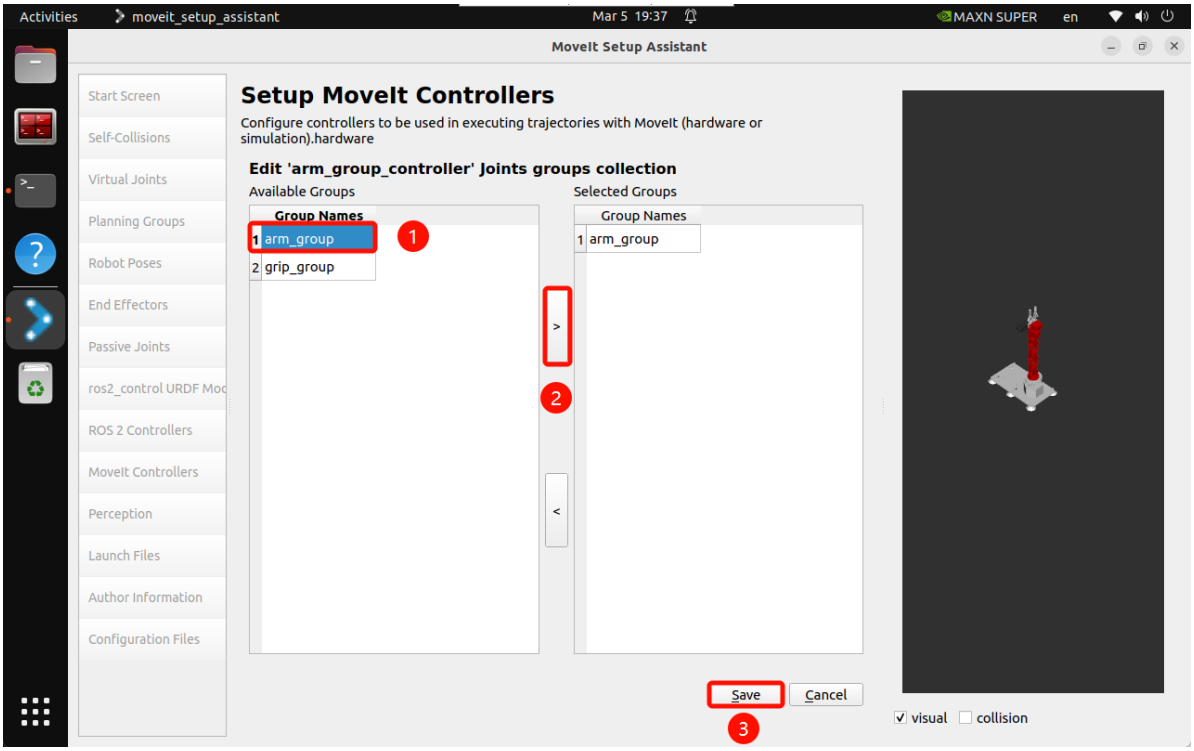
The added MoveIt Controllers need to ensure that the controller names match the names configured in ROS2 Controllers.

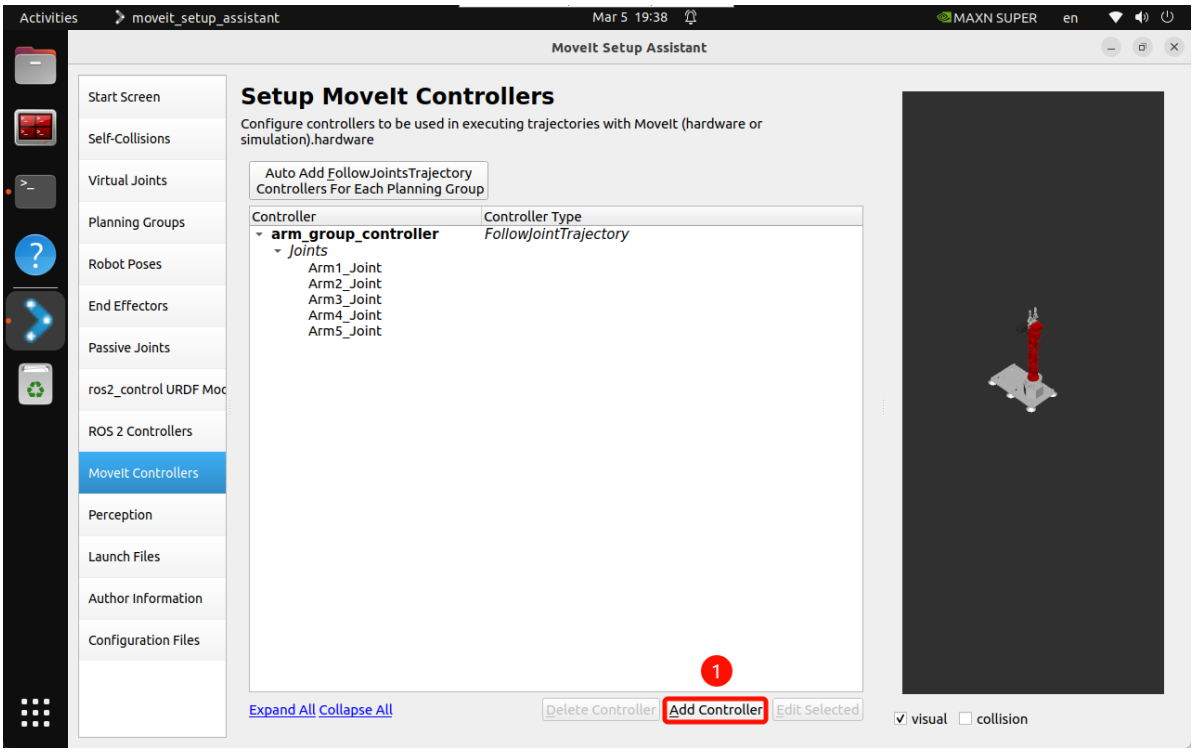


Add robotic arm controller:

- Controller Name：arm_group_controller
- Controller Type：FollowJointTrajectory
- Add Planning Group Joints：arm_group

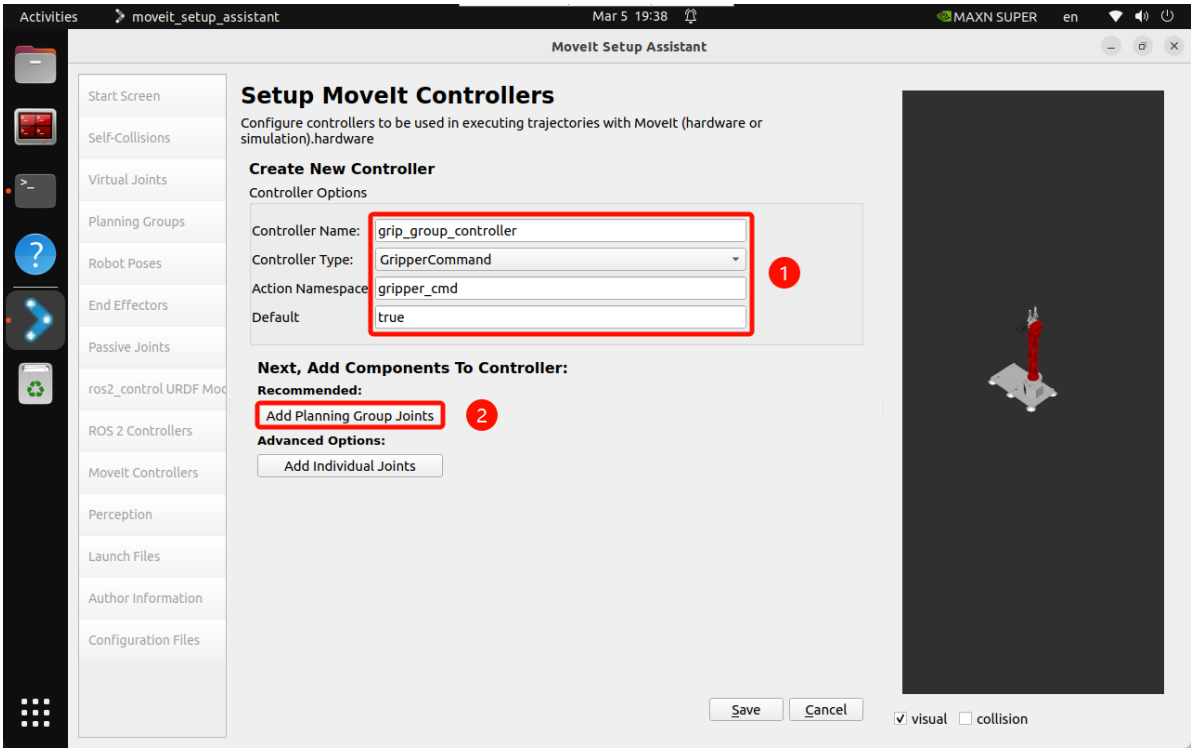


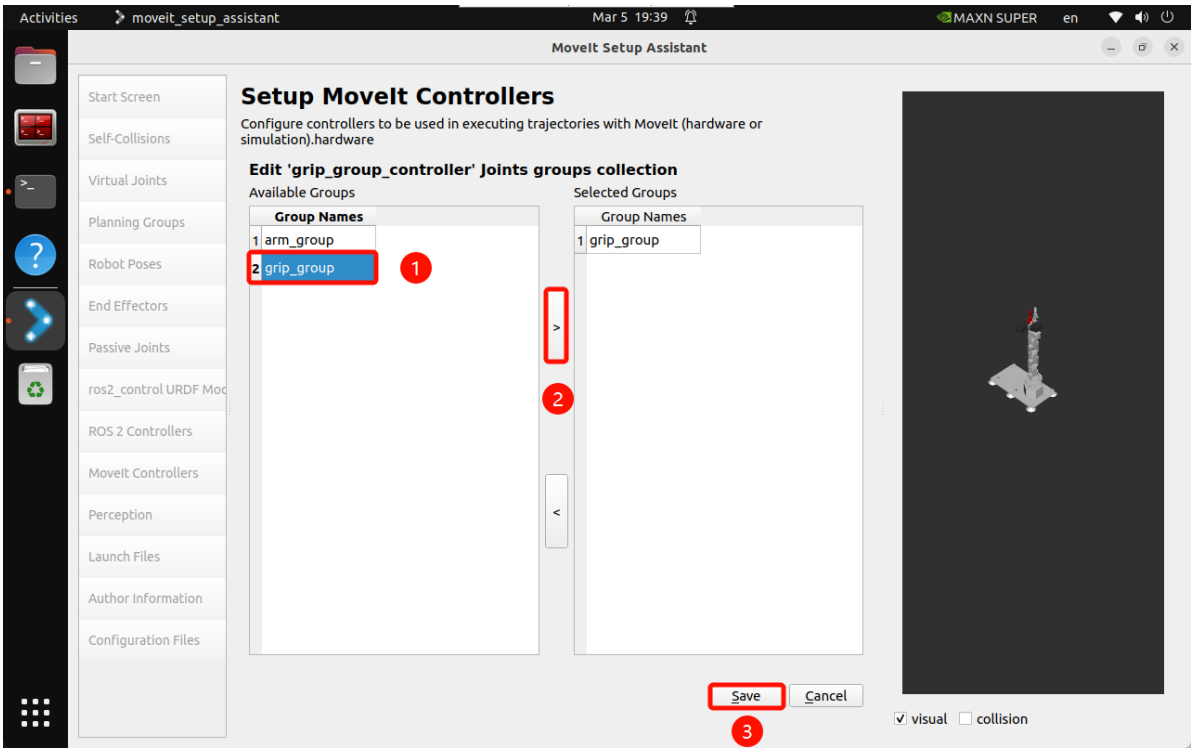


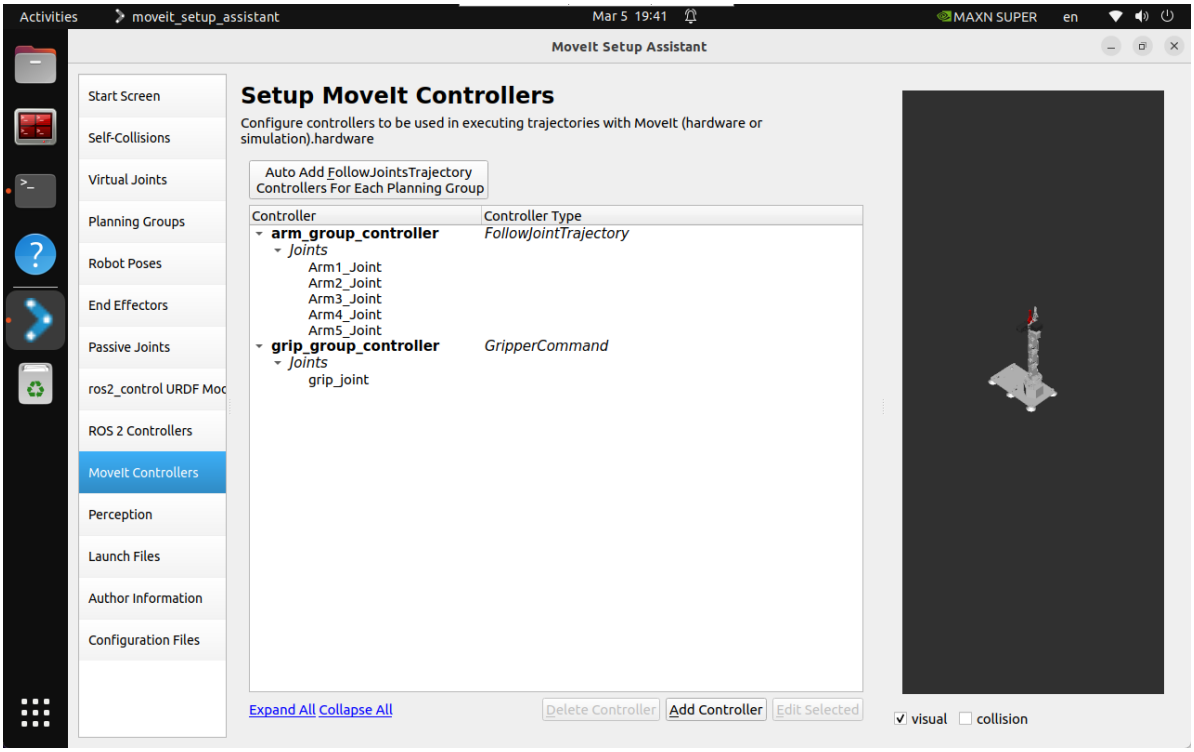


Add gripper controller:

- Controller Name：grip_group_controller
- Controller Type：Gripper Command
- Add Planning Group Joints：grip_group

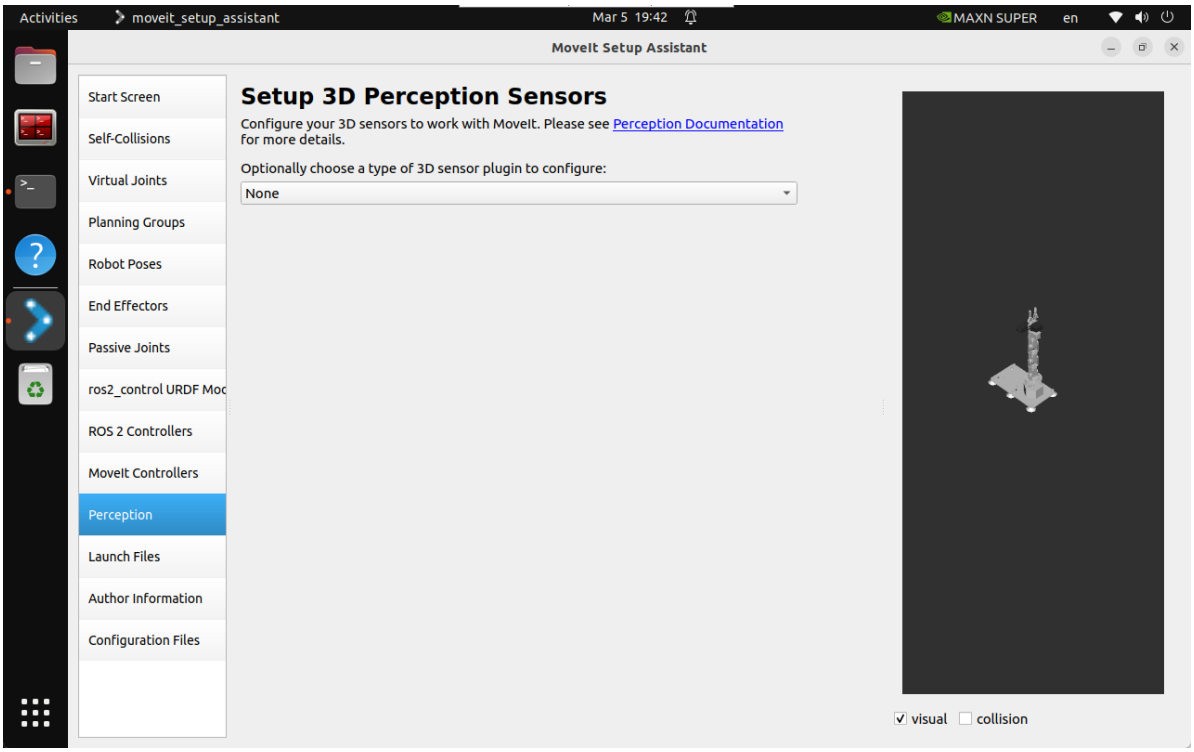

Activities
moveit_setup_assistant
Mar 5 19:38
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Setup MoveIt Controllers
Configure controllers to be used in executing trajectories with MoveIt (hardware or simulation).hardware
Create New Controller
Controller Options
Controller Name: grip_group_controller
Controller Type: GripperCommand
Action Namespace: gripper_cmd
Default: true
1
Next, Add Components To Controller:
Recommended:
Add Planning Group Joints
2
Advanced Options:
Add Individual Joints
Save
Cancel
visual
collision


Activities
moveit_setup_assistant
Mar 5 19:39
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Mod
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Setup MoveIt Controllers
Configure controllers to be used in executing trajectories with MoveIt (hardware or simulation).hardware
Edit 'grip_group_controller' Joints groups collection
Available Groups
Group Names
1 arm_group
2 grip_group
1
>
2
<
Selected Groups
Group Names
1 grip_group
Save
Cancel
3
visual
collision



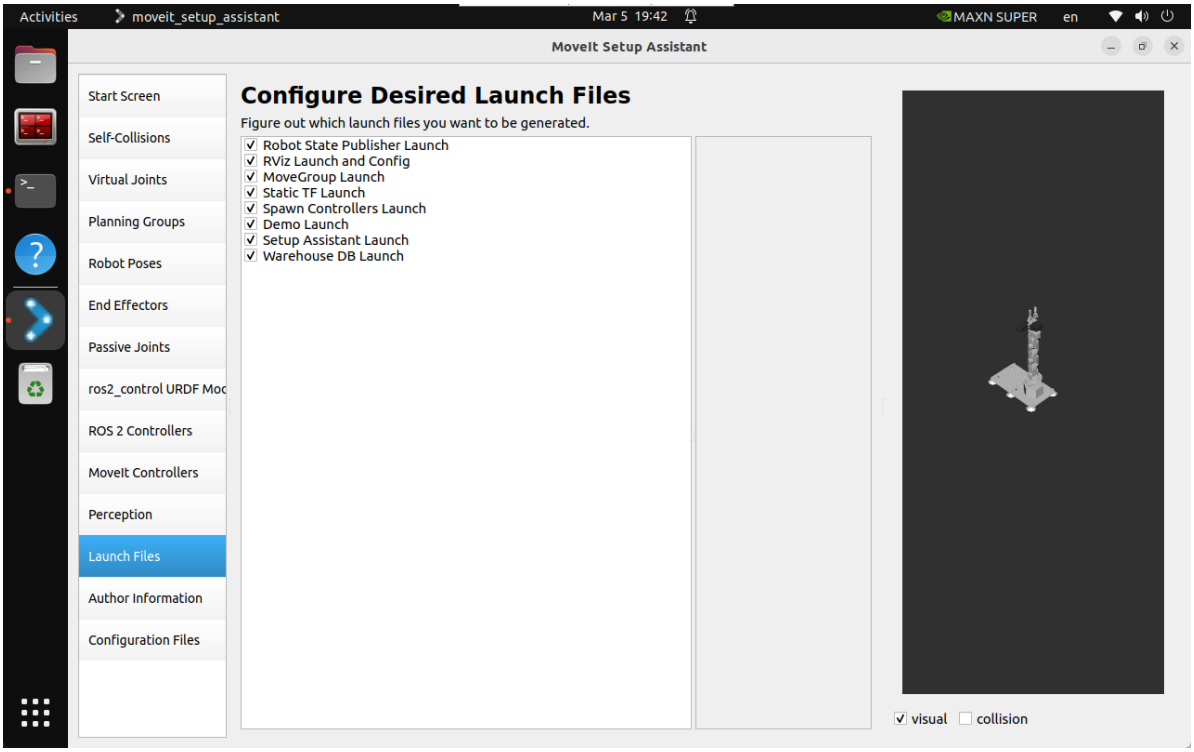
2.11. Sensors

Configure the settings for 3D sensors used by the robot: No configuration needed.



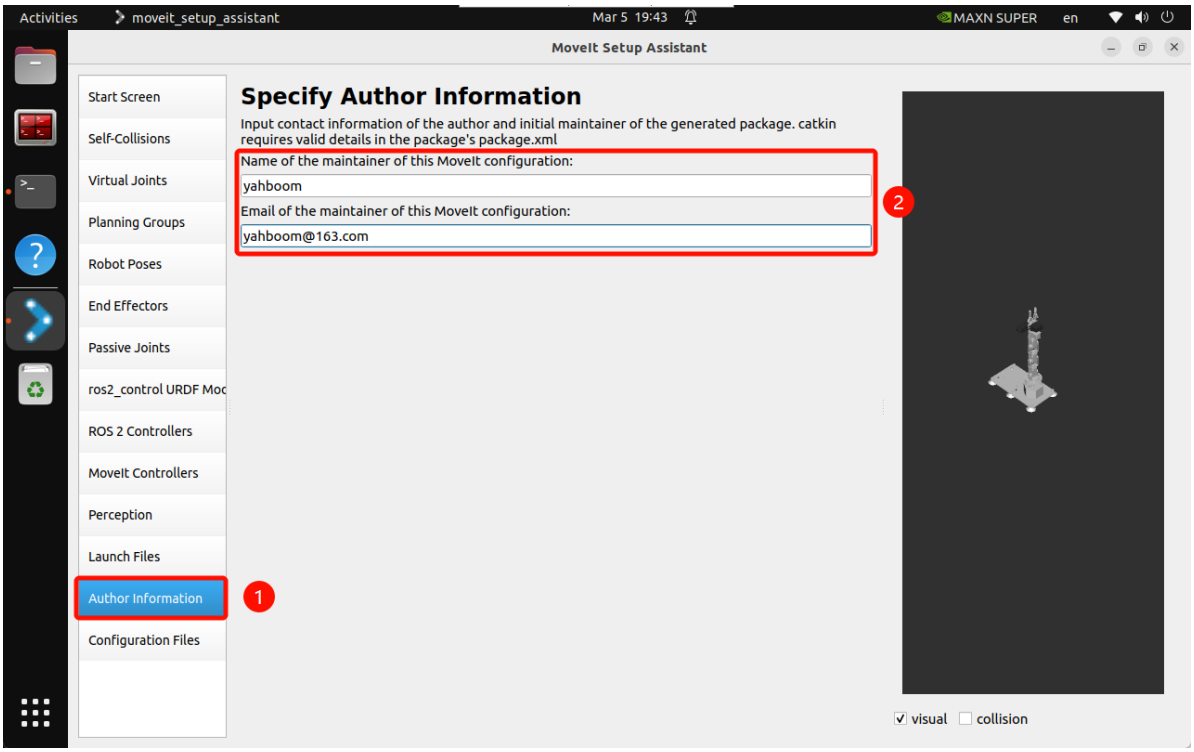
2.12. Launch Files

Configure auto-generated launch files: Use default options.



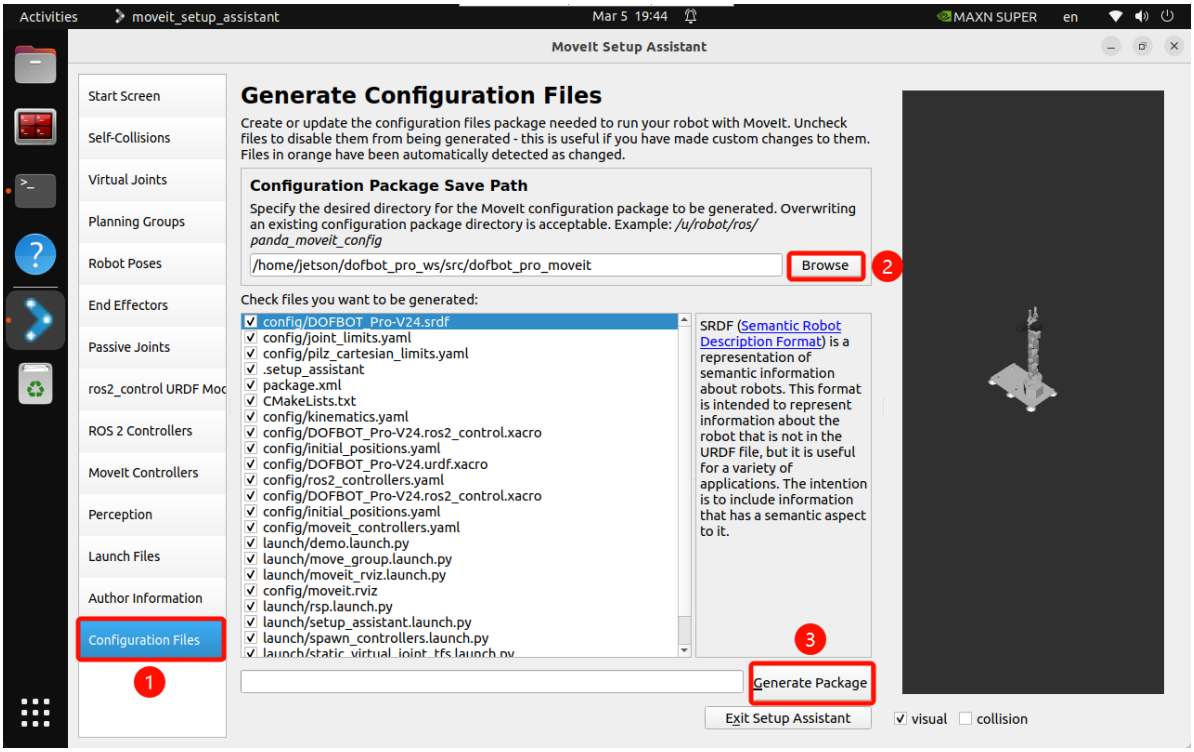
2.13. Author Information

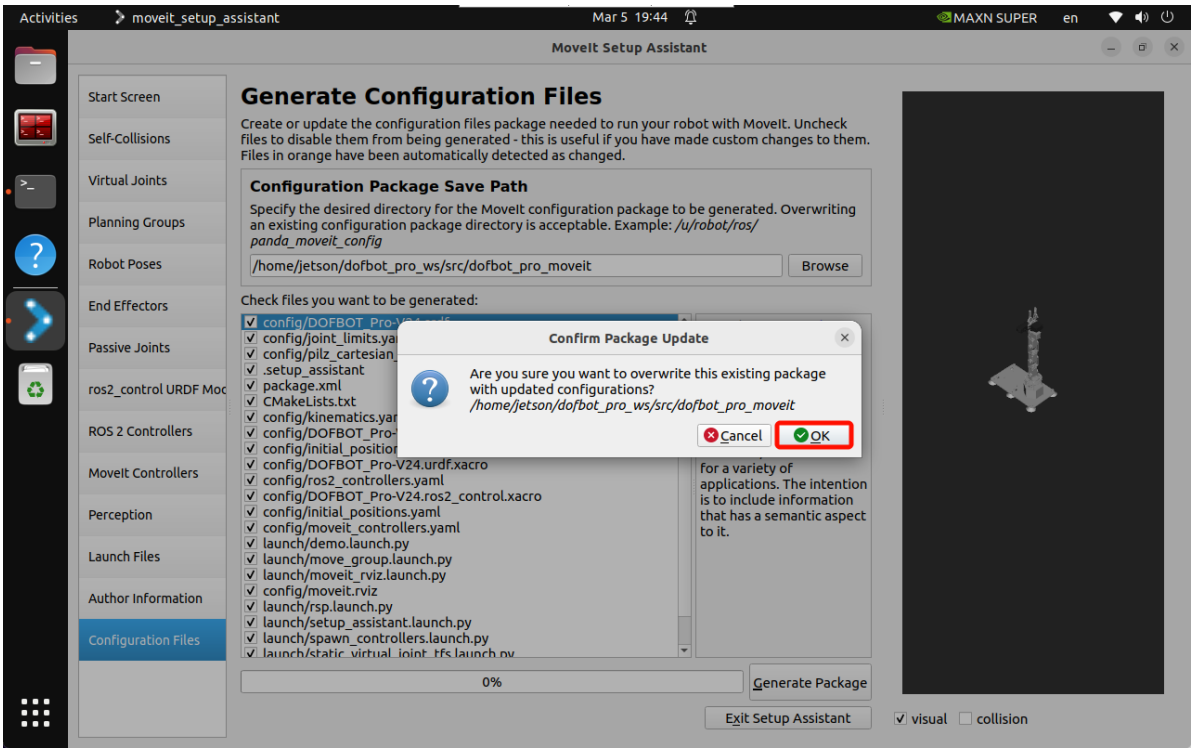
Add author information for package generation:

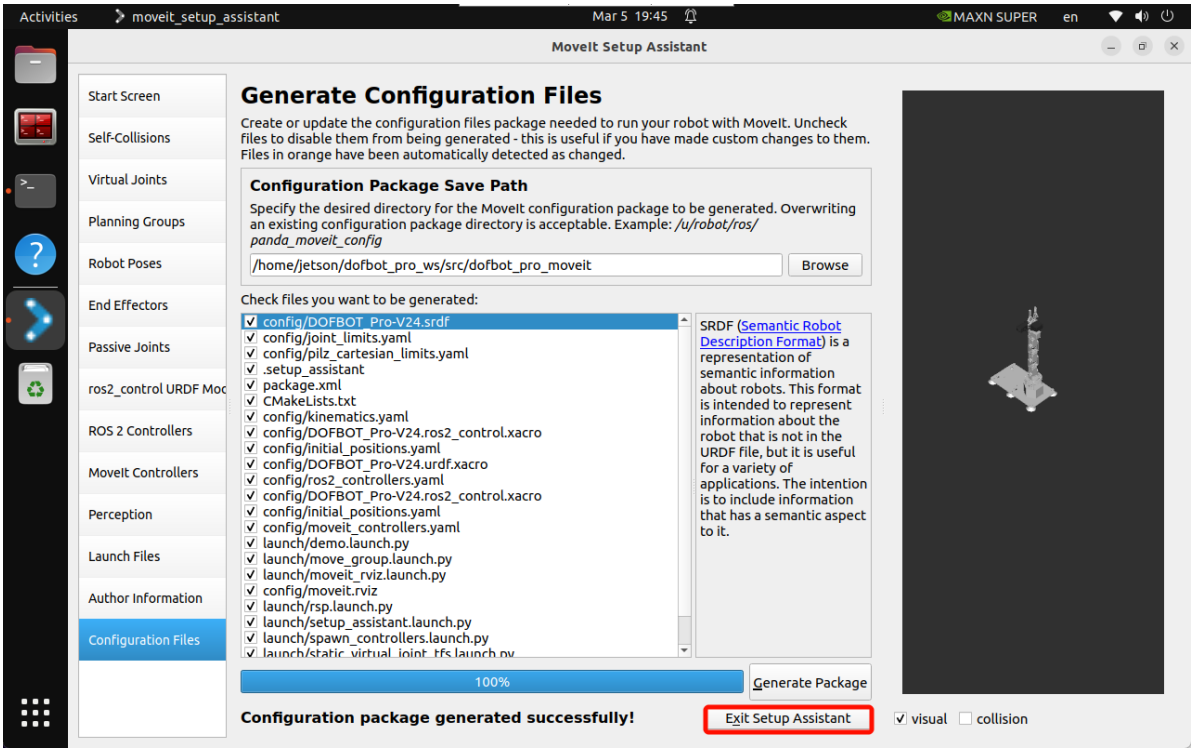


2.14. Generate Configuration

Generate configuration files to the specified folder:

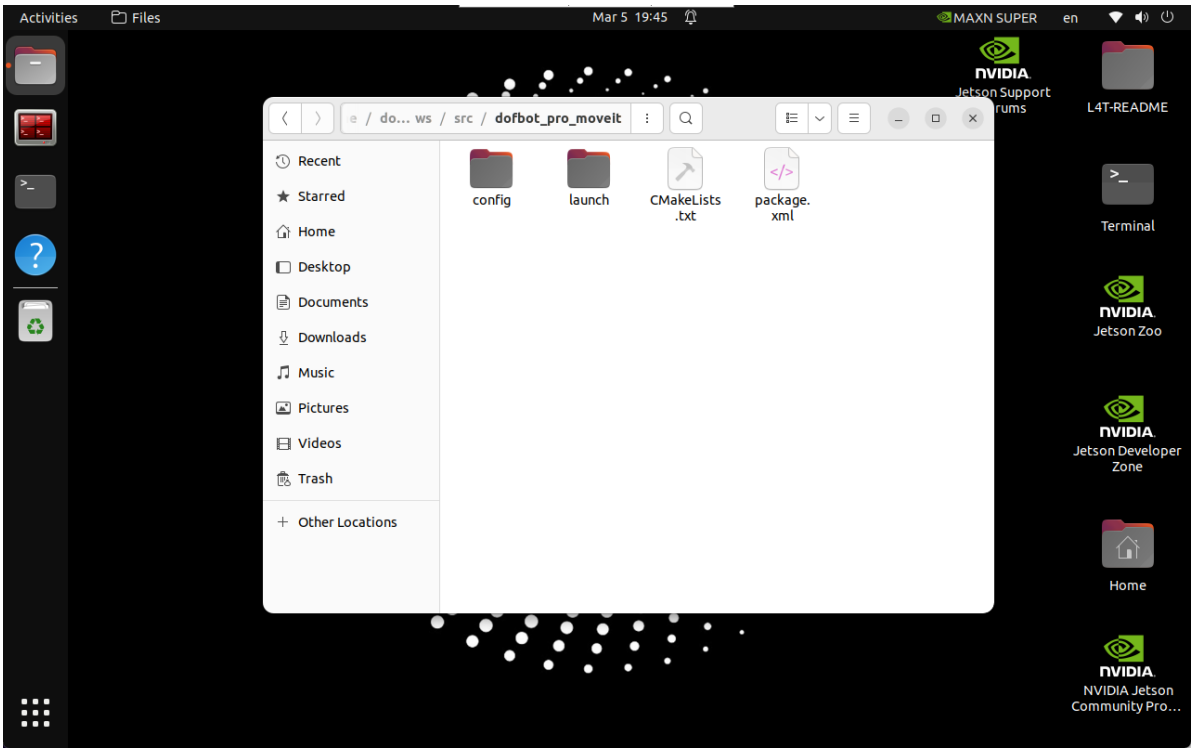

Activities
moveit_setup_assistant
Mar 5 19:44
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Moc
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
1
Generate Configuration Files
Create or update the configuration files package needed to run your robot with MoveIt. Uncheck files to disable them from being generated - this is useful if you have made custom changes to them. Files in orange have been automatically detected as changed.
Configuration Package Save Path
Specify the desired directory for the MoveIt configuration package to be generated. Overwriting an existing configuration package directory is acceptable. Example: /u/robot/ros/panda_moveit_config
/home/jetson/dofbot_pro_ws/src/dofbot_pro_moveit
Browse
2
Check files you want to be generated:
config/DOFBOT_Pro-V24.srdf
config/joint_limits.yaml
config/pilz_cartesian_limits.yaml
.setup_assistant
package.xml
CMakeLists.txt
config/kinematics.yaml
config/DOFBOT_Pro-V24.ros2_control.xacro
config/initial_positions.yaml
config/DOFBOT_Pro-V24.urdf.xacro
config/ros2_controllers.yaml
config/DOFBOT_Pro-V24.ros2_control.xacro
config/initial_positions.yaml
config/moveit_controllers.yaml
launch/demo.launch.py
launch/move_group.launch.py
launch/moveit_rviz.launch.py
config/moveit.rviz
launch/rsp.launch.py
launch/setup_assistant.launch.py
launch/spawn_controllers.launch.py
SRDF (Semantic Robot Description Format) is a representation of semantic information about robots. This format is intended to represent information about the robot that is not in the URDF file, but it is useful for a variety of applications. The intention is to include information that has a semantic aspect to it.
3
Generate Package
Exit Setup Assistant
visual
collision


Activities
moveit_setup_assistant
Mar 5 19:44
MAXN SUPER
en
MoveIt Setup Assistant
Start Screen
Self-Collisions
Virtual Joints
Planning Groups
Robot Poses
End Effectors
Passive Joints
ros2_control URDF Moc
ROS 2 Controllers
MoveIt Controllers
Perception
Launch Files
Author Information
Configuration Files
Generate Configuration Files
Create or update the configuration files package needed to run your robot with MoveIt. Uncheck files to disable them from being generated - this is useful if you have made custom changes to them. Files in orange have been automatically detected as changed.
Configuration Package Save Path
Specify the desired directory for the MoveIt configuration package to be generated. Overwriting an existing configuration package directory is acceptable. Example: /u/robot/ros/panda_moveit_config
/home/jetson/dofbot_pro_ws/src/dofbot_pro_moveit
Browse
Check files you want to be generated:
Confirm Package Update
Are you sure you want to overwrite this existing package with updated configurations?
/home/jetson/dofbot_pro_ws/src/dofbot_pro_moveit
Cancel
OK
config/DOFBOT_Pro-V24.urdf.xacro
config/ros2_controllers.yaml
config/DOFBOT_Pro-V24.ros2_control.xacro
config/initial_positions.yaml
config/moveit_controllers.yaml
launch/demo.launch.py
launch/move_group.launch.py
launch/moveit_rviz.launch.py
config/moveit.rviz
launch/rsp.launch.py
launch/setup_assistant.launch.py
launch/spawn_controllers.launch.py
for a variety of applications. The intention is to include information that has a semantic aspect to it.
0%
Generate Package
Exit Setup Assistant
visual
collision



3. Configuration Files

Enter the `dofbot_pro_moveit` folder:



4. Configuration Verification

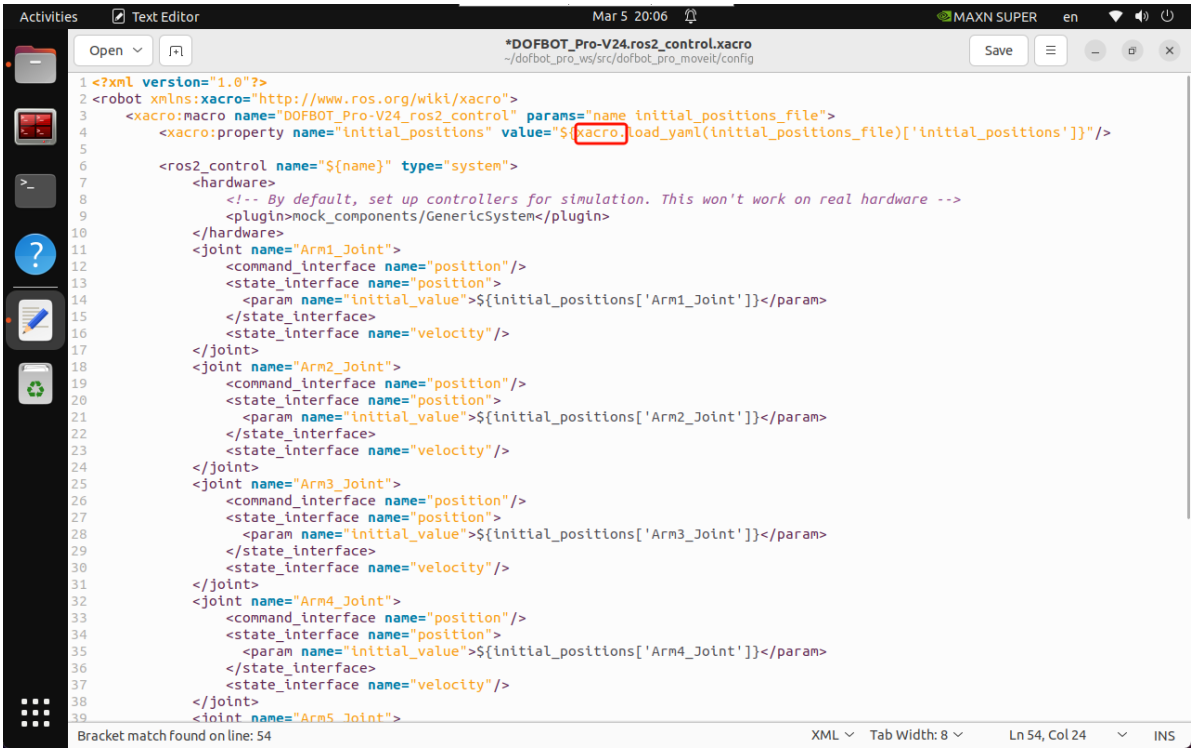
The configuration files generated by MoveIt Setup Assistant need simple modifications to remove startup warnings and model loading issues.

4.1. Modify Configuration Files

DOFBOT_Pro-V24.ros2_control.xacro

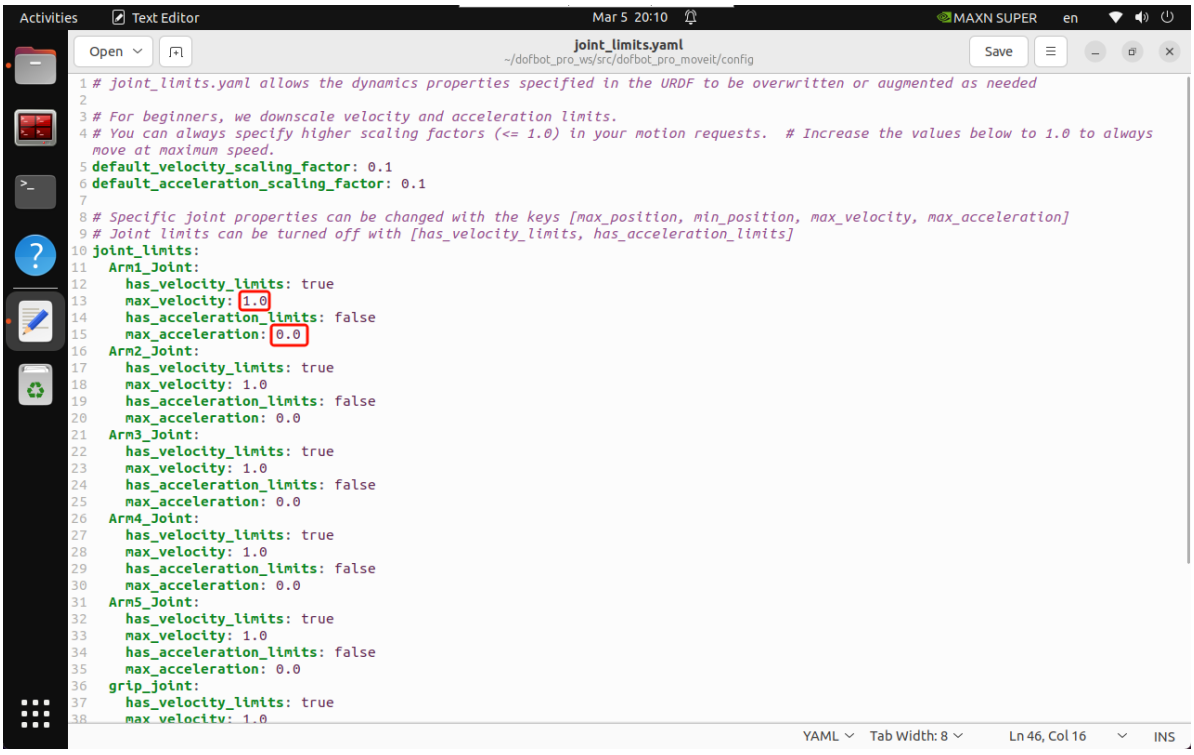
Modify the `DOFBOT_Pro-V24.ros2_control.xacro` file: Modify the yaml loading function

```
load_yaml(initial_positions_file)['initial_positions']
Change to
xacro.load_yaml(initial_positions_file)['initial_positions']
```



joint_limits.yaml

Modify the `joint_limits.yaml` file: Change all joint maximum velocities and accelerations to decimal numbers



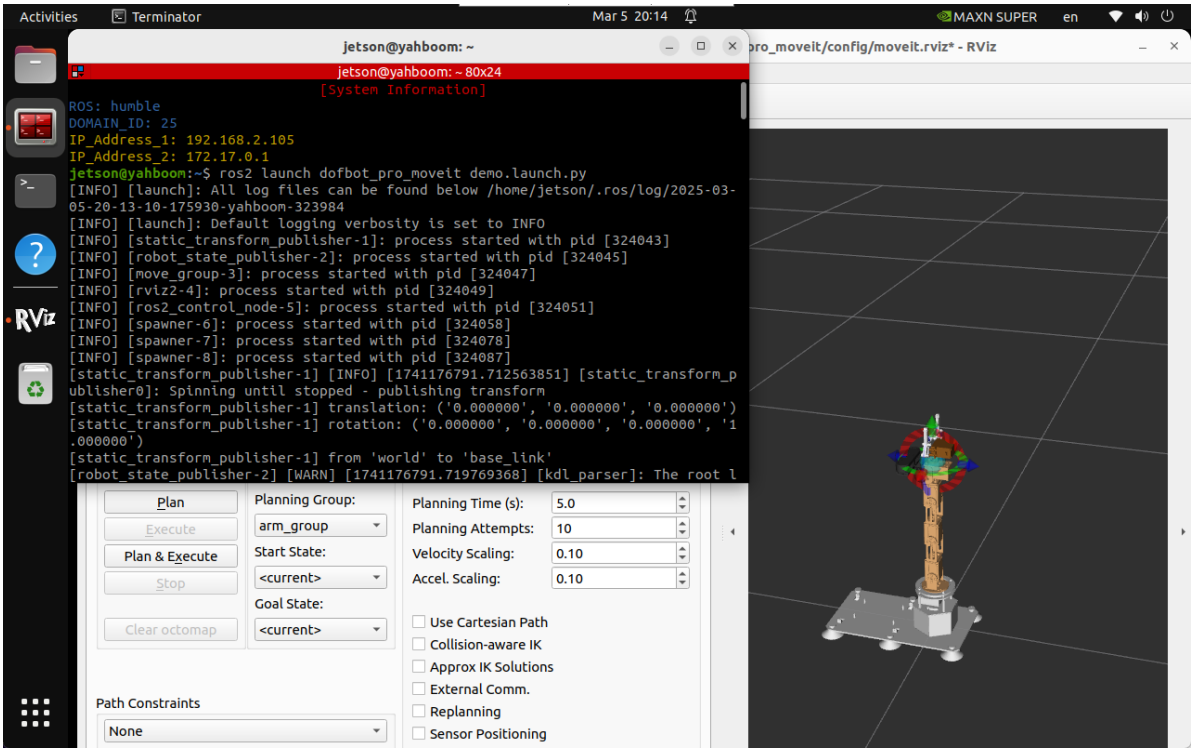
4.2. Compile Package

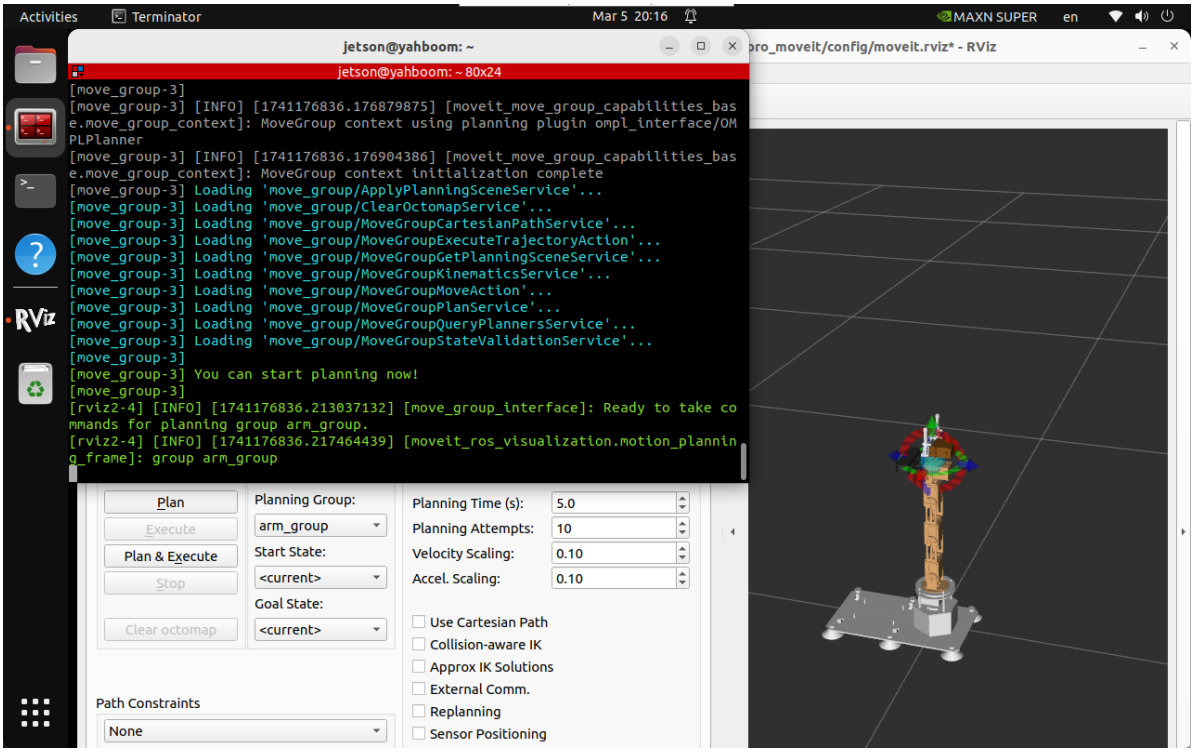
```
cd ~/dofbot_pro_ws
colcon build --packages-select dofbot_pro_moveit
source install/setup.bash
```

4.3. Launch MoveIt

```
ros2 launch dofbot_pro_moveit demo.launch.py
```

The simulation startup is relatively slow. Wait until the terminal shows `You can start planning now!` or the robotic arm shows trajectory balls (trajectory balls are new spheres added to the robotic arm), indicating loading is complete.





References

MoveIt2 Humble：https://moveit.picknik.ai/humble/index.html

MoveIt2：https://moveit.picknik.ai/main/index.html