

Movelt configuration

Note: The configuration provided by Yahboom in this course is already complete.
If you are using the mirror system provided by Yahboom, you can skip the Moveit configuration course.

1. Start roscore

Open the terminal and enter the following code

```
roscore
```

2. Start the Movelt configuration tool

Open a new terminal.

Open the terminal and enter the following code

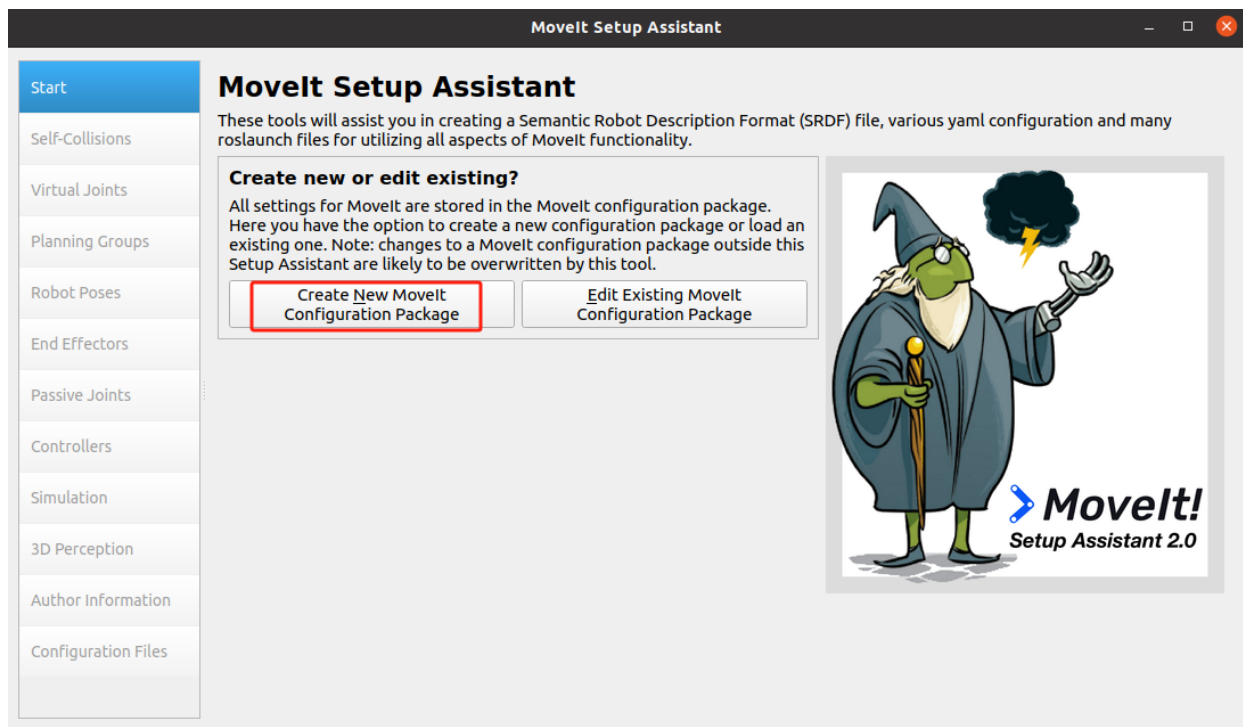
```
roslaunch moveit_setup_assistant moveit_setup_assistant
```

3. Configuration process

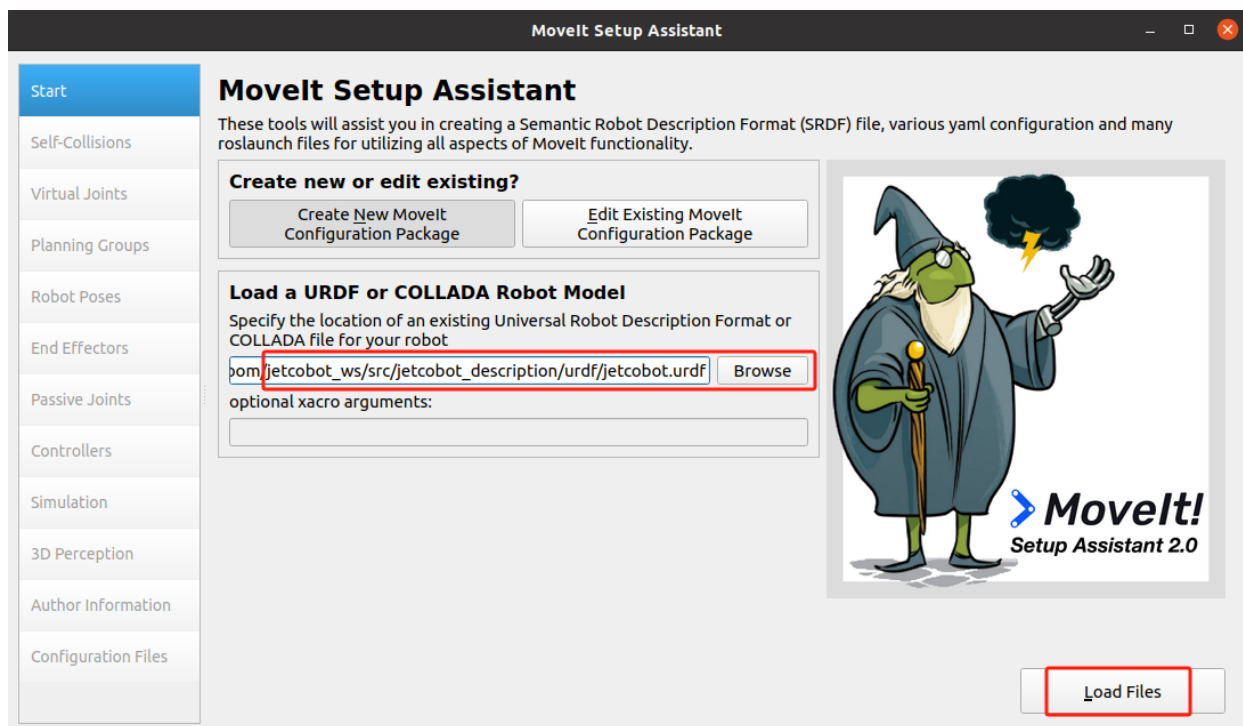
- Loading URDF models

If you get an error message **【Cannot find model】** when loading a model, exit Movelt-->enter the workspace-->update the environment (source devel/ setup .bash)-->restart Movelt configuration.

If you are loading a model to generate configuration for the first time, select the left side. If you are modifying an already generated configuration file, select the right side.



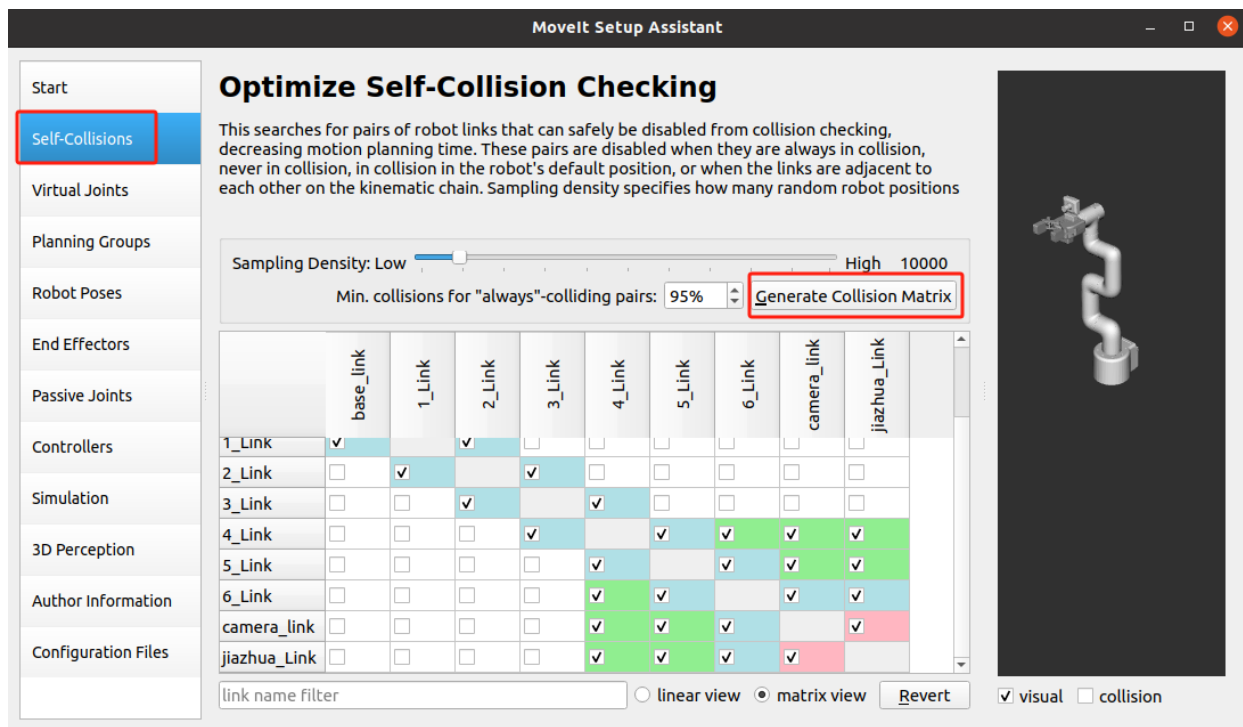
Click the Browse button, find the URDF model file, and click Load in the lower right corner.



- Create an Avoid Collision Matrix (ACM)

Collision detection is a very complex calculation process. For multi-joint manipulators or humanoid robots, the mechanical structure is complex and there are many joints. Collision detection requires a lot of spatial geometry calculations. However, for rigid robots, some joints cannot collide, such as adjacent limbs. The purpose of generating self-collisions here is to tell us which joints will not collide. In the subsequent collision detection algorithm, you can directly skip the detection of these joints to improve the detection efficiency.

After the model is loaded, select [Self-Collisions] and click the [Generate Collision Matrix] button.



- Adding a virtual joint can be understood as a joint that connects the robot and the world.

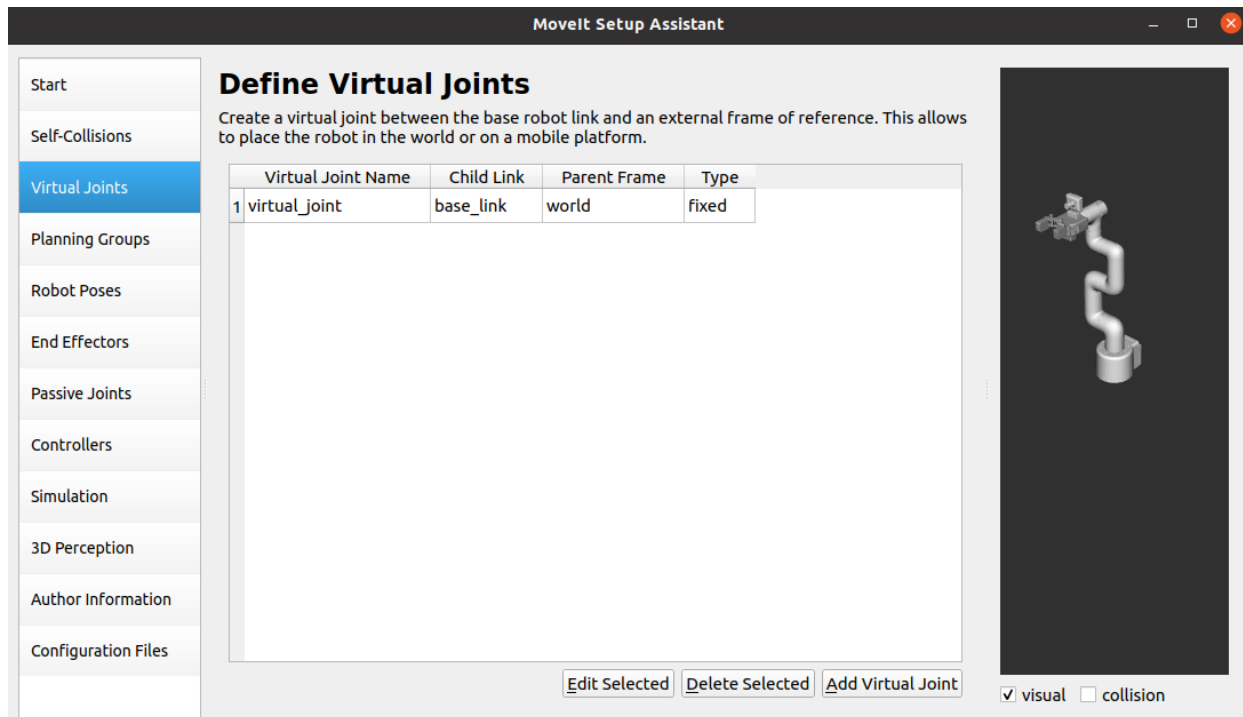
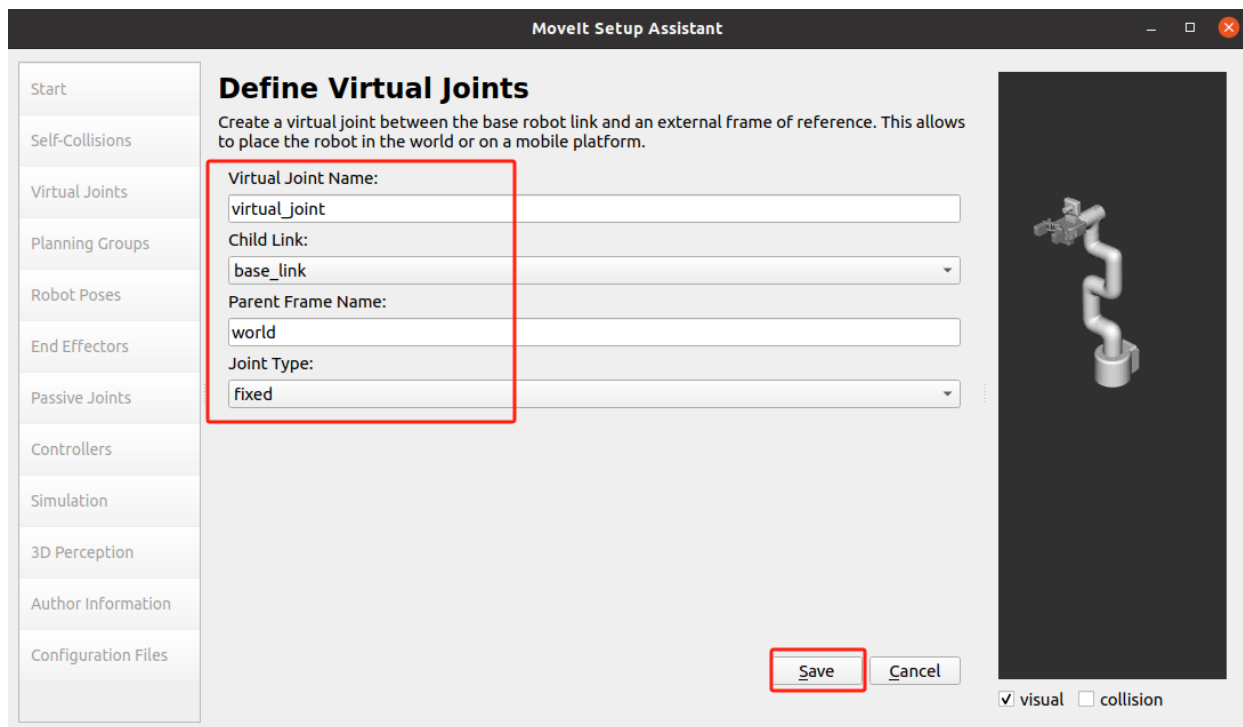


Virtual Joint Name We name it virtual_joint.

Child Link refers to the part of the robot that we want to connect the 'world' to. We choose the base_base_link.

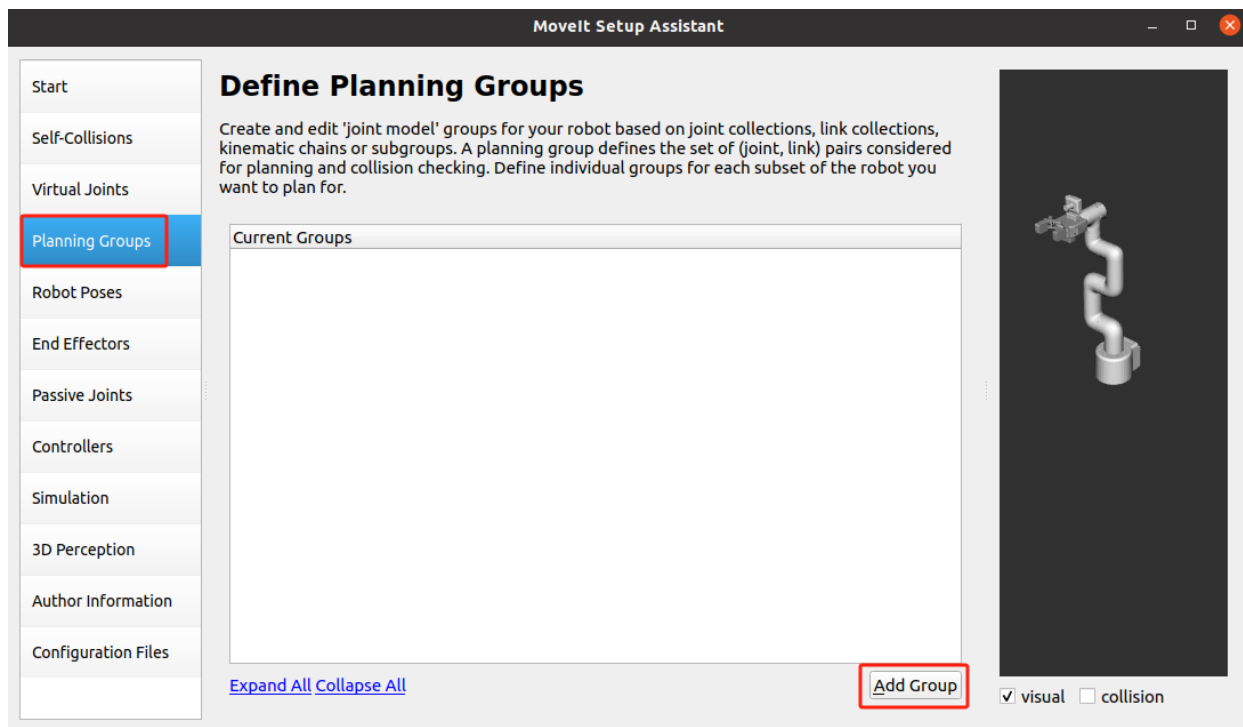
Parent Frame Name is the name of the world coordinates, which is generally called world in ROS.

Joint Type Select Fixed. This means that the robot is fixed relative to the world. The other two types, Planar, refer to the planar mobile base (xy plane + angle), used for mobile robots such as PR2; and Floating, refer to the floating base (xyz position + orientation), such as humanoid robots.



- Create a motion planning group

Planning Group is one of the core functions of MoveIt. Click [Add Group] to add a planning group



- Group Name: Create a group name and set it to [arm_group].
- Kinematic Solver: Here we select [KDL]

Kinematic solver, which is responsible for solving forward kinematics and inverse kinematics (IK). Generally, we choose KDL, The Kinematics and Dynamics Library. This is a kinematics and dynamics library that can solve the forward and inverse kinematics problems of single-chain mechanical structures with more than 6 degrees of freedom.

Of course, you can also use other IK Solvers, such as SRV or IK_FAST, or even develop a new Solver yourself and insert it.

- Kin. Search Resolution: Sampling density of joint space
- Kin. Search Timeout: Solving time. If the device performance is insufficient or there is no solution within the specified time in the actual application process, the time can be increased; for example, set to [0.1], [0.01].

Click [Add Kin.Chain] to add a joint chain.

Create New Planning Group

Kinematics

Group Name:

Kinematic Solver:

Kin. Search Resolution:

Kin. Search Timeout (sec):

Goal Joint Tolerance (m|rad):

Goal Position Tolerance (m):

Goal Orientation Tolerance (rad):

Kin. parameters file: ...

OMPL Planning

Group Default Planner:

Next, Add Components To Group:

Recommended:

Advanced Options:

Click [Expand All], select Base Link as [base_link], select Tip Link as [6_Link]; click [save] to save.

Define Planning Groups

Create and edit 'joint model' groups for your robot based on joint collections, link collections, kinematic chains or subgroups. A planning group defines the set of (joint, link) pairs considered for planning and collision checking. Define individual groups for each subset of the robot you want to plan for.
Note: when adding a link to the group, its parent joint is added too and vice versa.

Edit 'arm_group' Kinematic Chain

Robot Links

- ▼ dummy
 - ▼ base_link
 - ▼ 1_Link
 - ▼ 2_Link
 - ▼ 3_Link
 - ▼ 4_Link
 - ▼ 5_Link
 - ▼ 6_Link
 - camera_link
 - jiazhua_Link

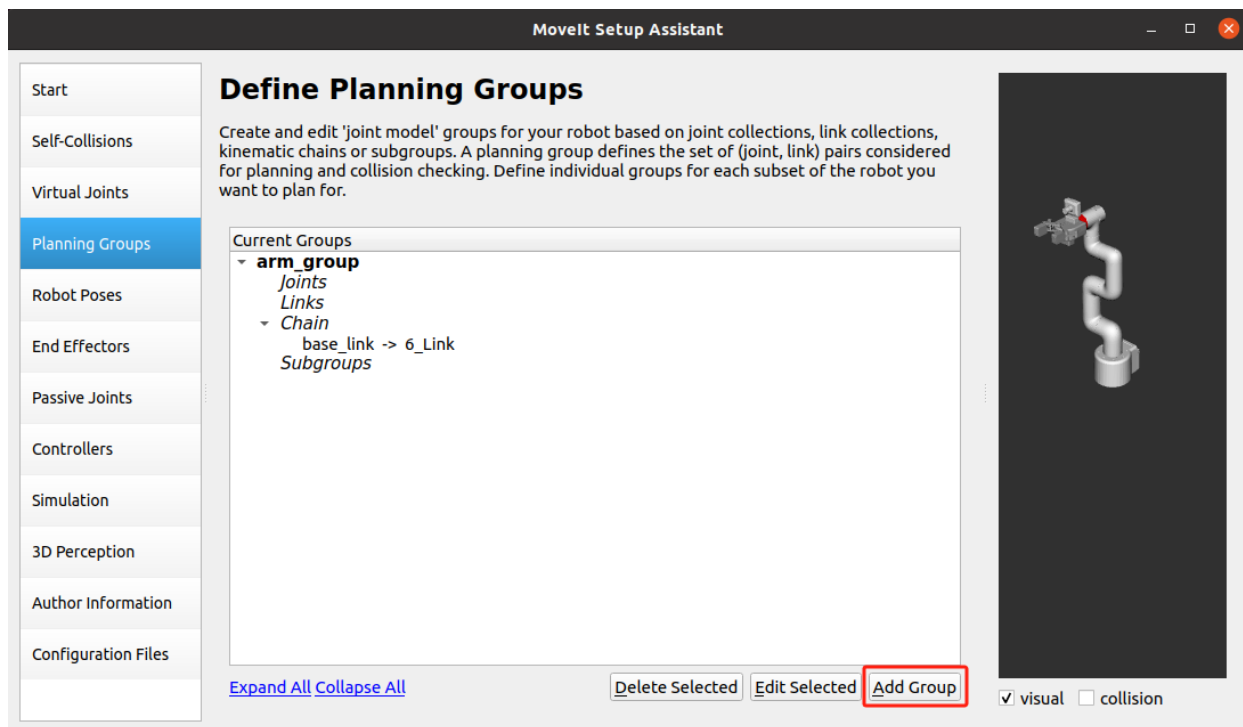
Base Link

Tip Link

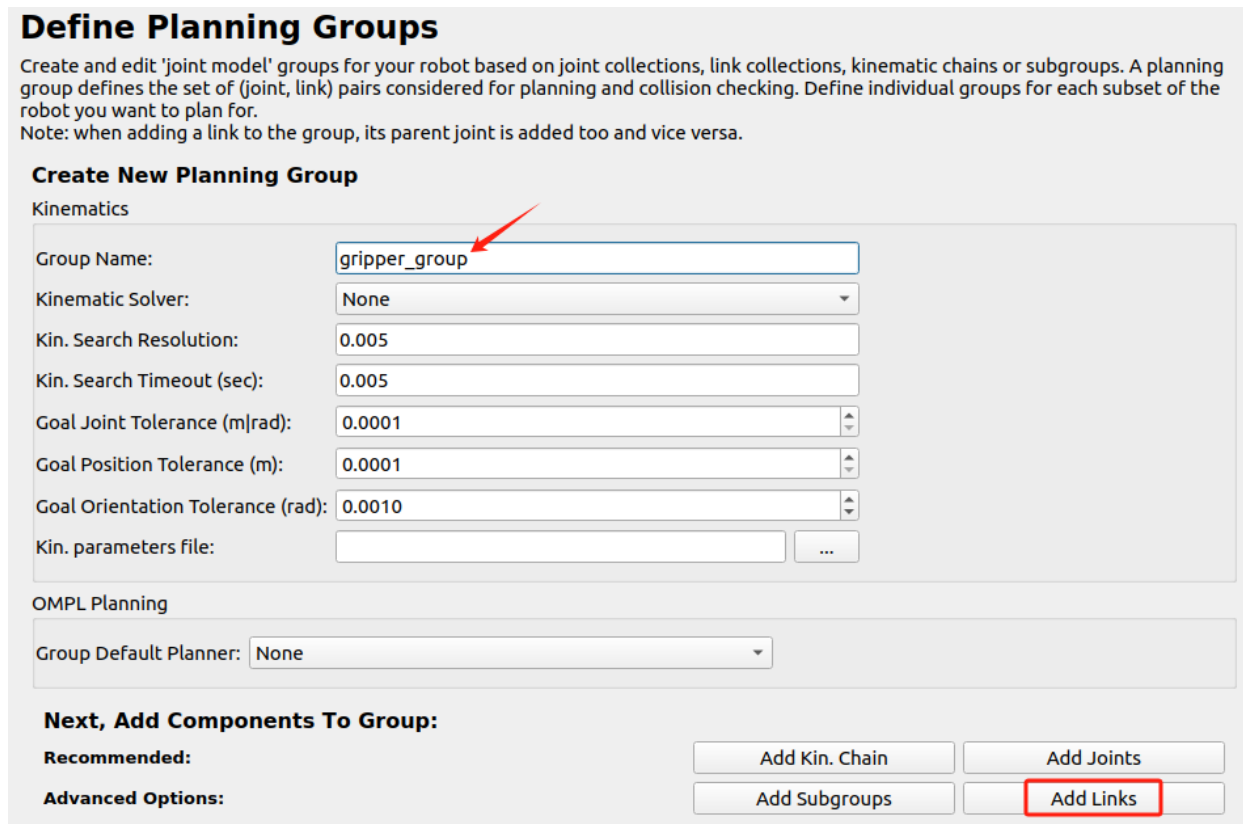
[Expand All](#) [Collapse All](#)

- Add a gripper planning group

Click [Add Group] to add a gripper planning group.



Set the creation group name to [gripper_group], and there is no need to set the kinematic solver; click [Add Links] to add the gripper link.



Select the connecting rod of the gripper part, click [>], automatically add the right side, and click [Save] to save.

Define Planning Groups

Create and edit 'joint model' groups for your robot based on joint collections, link collections, kinematic chains or subgroups. A planning group defines the set of (joint, link) pairs considered for planning and collision checking. Define individual groups for each subset of the robot you want to plan for.

Note: when adding a link to the group, its parent joint is added too and vice versa.

Edit 'gripper_group' Link Collection

Available Links

Link Names
1 dummy
2 base_link
3 1_Link
4 2_Link
5 3_Link
6 4_Link
7 5_Link
8 6_Link
9 camera_link
10 jiazhua_Link

Selected Links

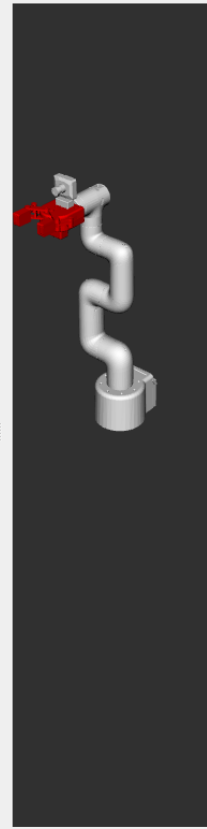
Link Names
1 jiazhua_Link



Save

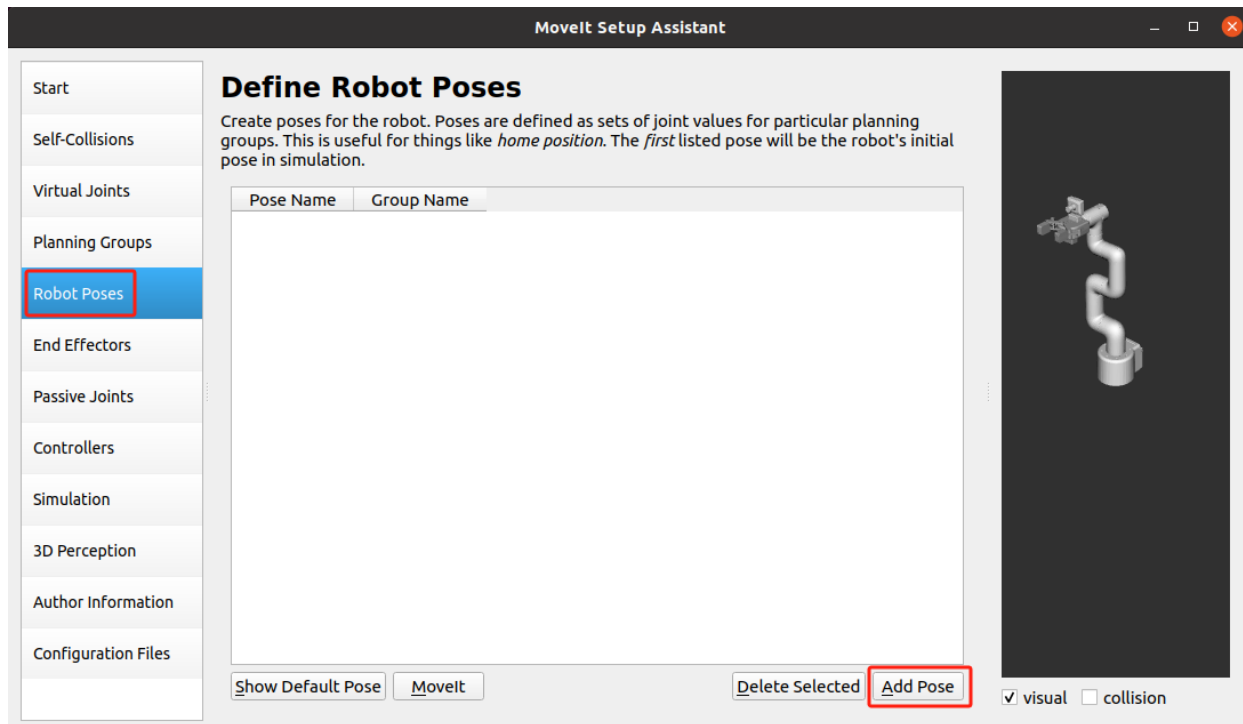
Cancel

☒ visual ☐ collision

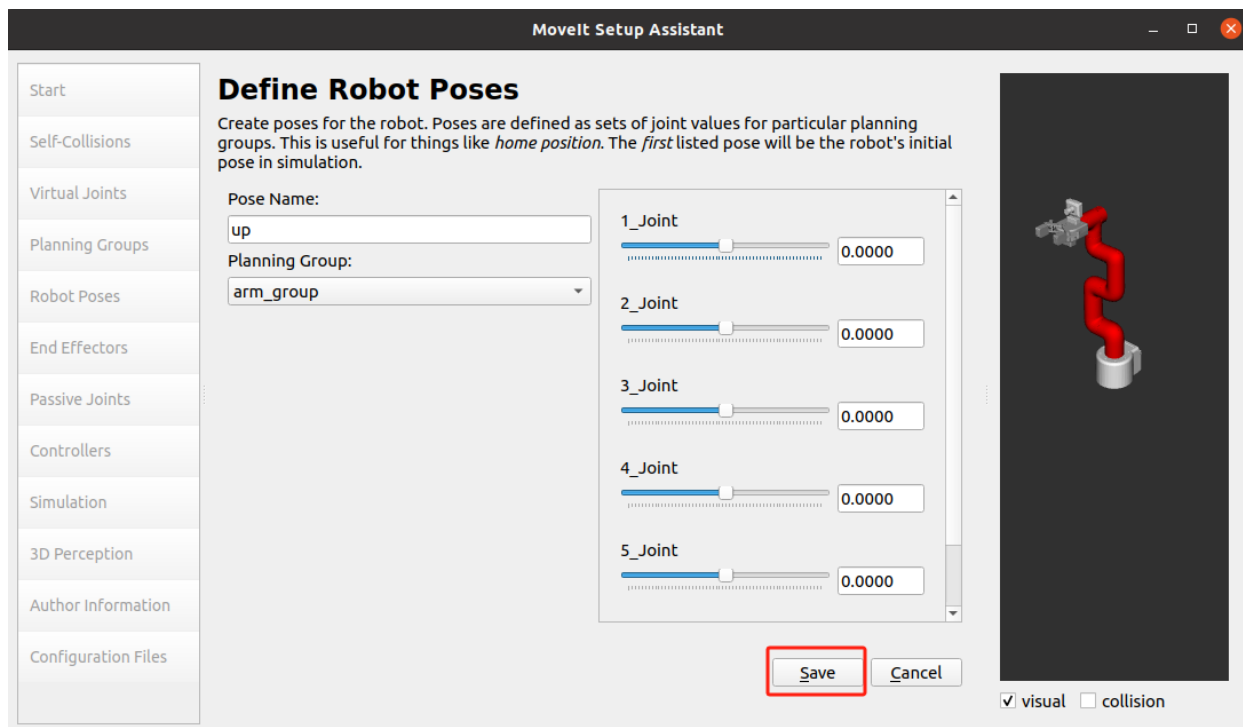


- Create preset poses

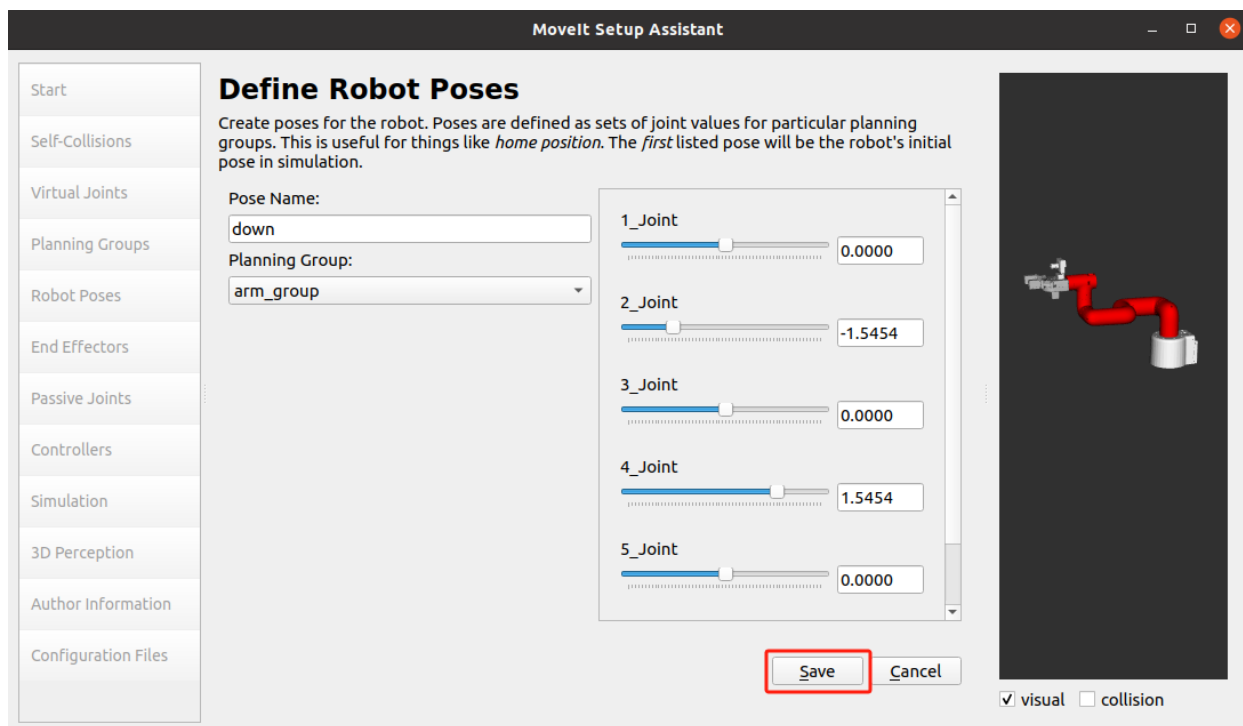
Select [Robot Poses] and click [Add pose] to add a pose.



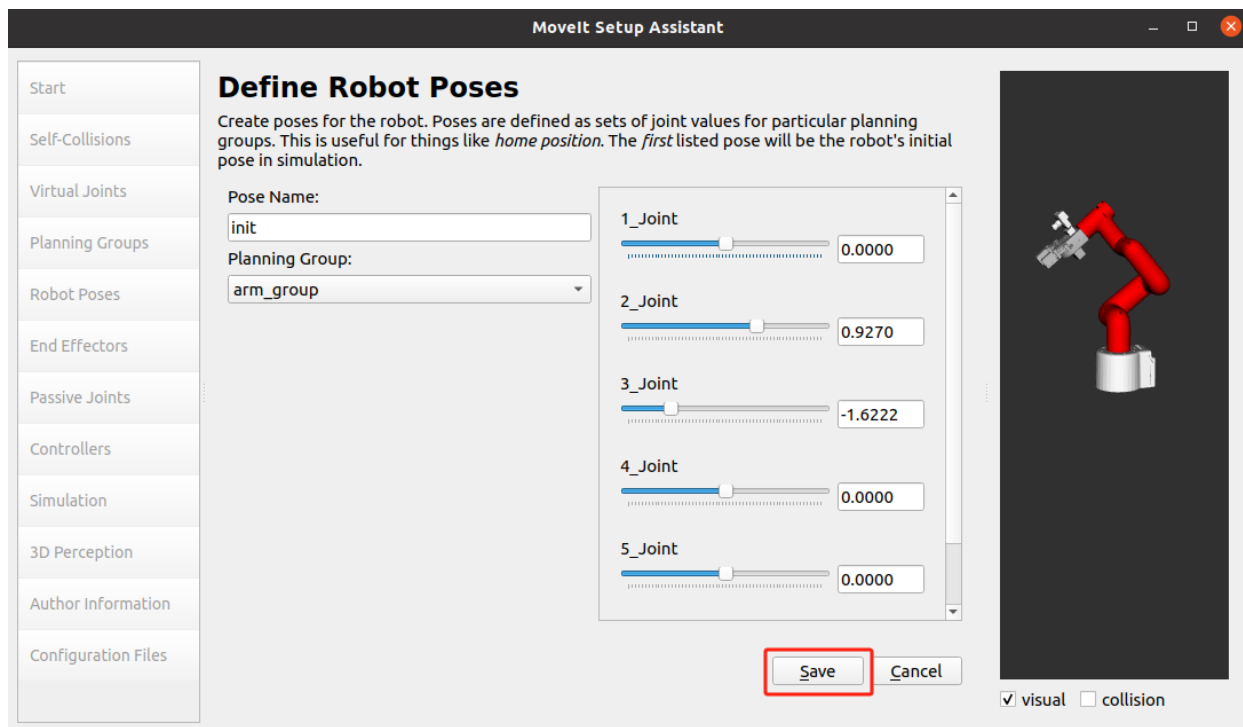
Add the robot arm posture, and set the posture name Pose Name to [up].



Added [down] gesture

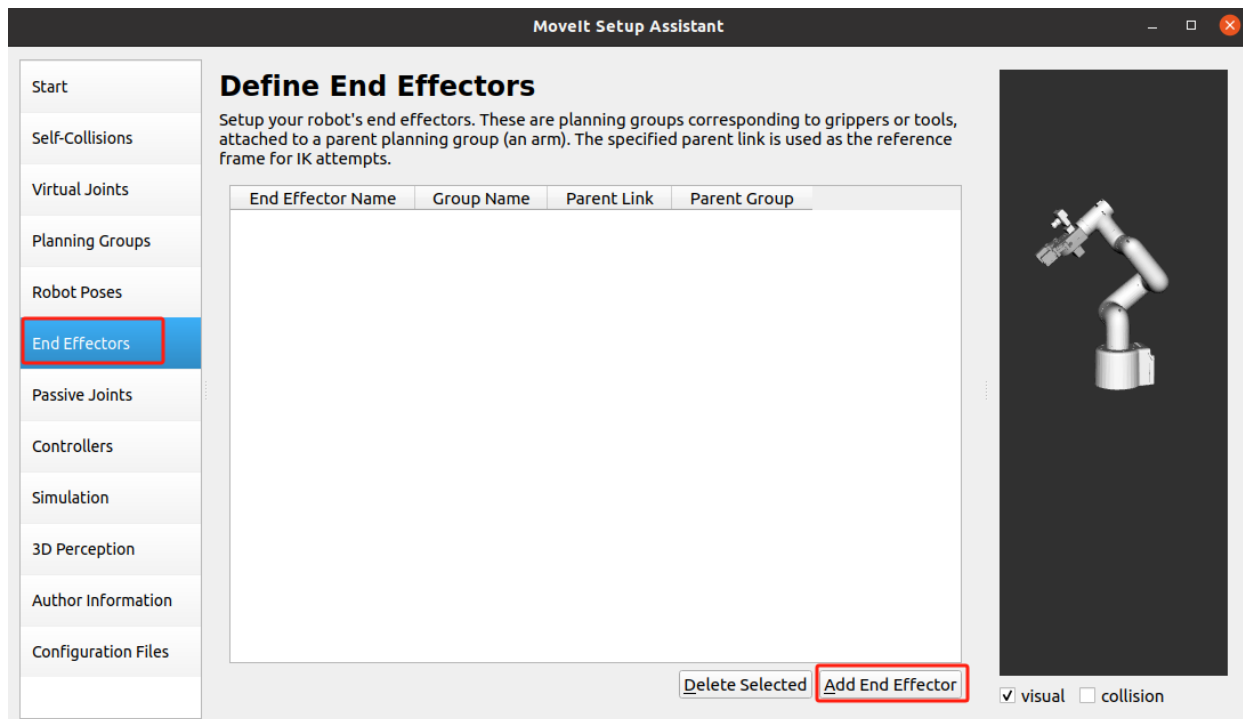


Added [init] gesture



- Set the end effector joint

Select [End Effectors] and set it as shown below



Start

Self-Collisions

Virtual Joints

Planning Groups

Robot Poses

End Effectors

Passive Joints

Controllers

Simulation

3D Perception

Author Information

Configuration Files

Define End Effectors

Setup your robot's end effectors. These are planning groups corresponding to grippers or tools, attached to a parent planning group (an arm). The specified parent link is used as the reference frame for IK attempts.

End Effector Name:

gripper

End Effector Group:

gripper_group

Parent Link (usually part of the arm):

6_Link

Parent Group (optional):

arm_group

Save

Cancel

☒ visual

☐ collision

Start

Self-Collisions

Virtual Joints

Planning Groups

Robot Poses

End Effectors

Passive Joints

Controllers

Simulation

3D Perception

Author Information

Configuration Files

Define End Effectors

Setup your robot's end effectors. These are planning groups corresponding to grippers or tools, attached to a parent planning group (an arm). The specified parent link is used as the reference frame for IK attempts.

	End Effector Name	Group Name	Parent Link	Parent Group
1	gripper	gripper_group	6_Link	arm_group

Edit Selected

Delete Selected

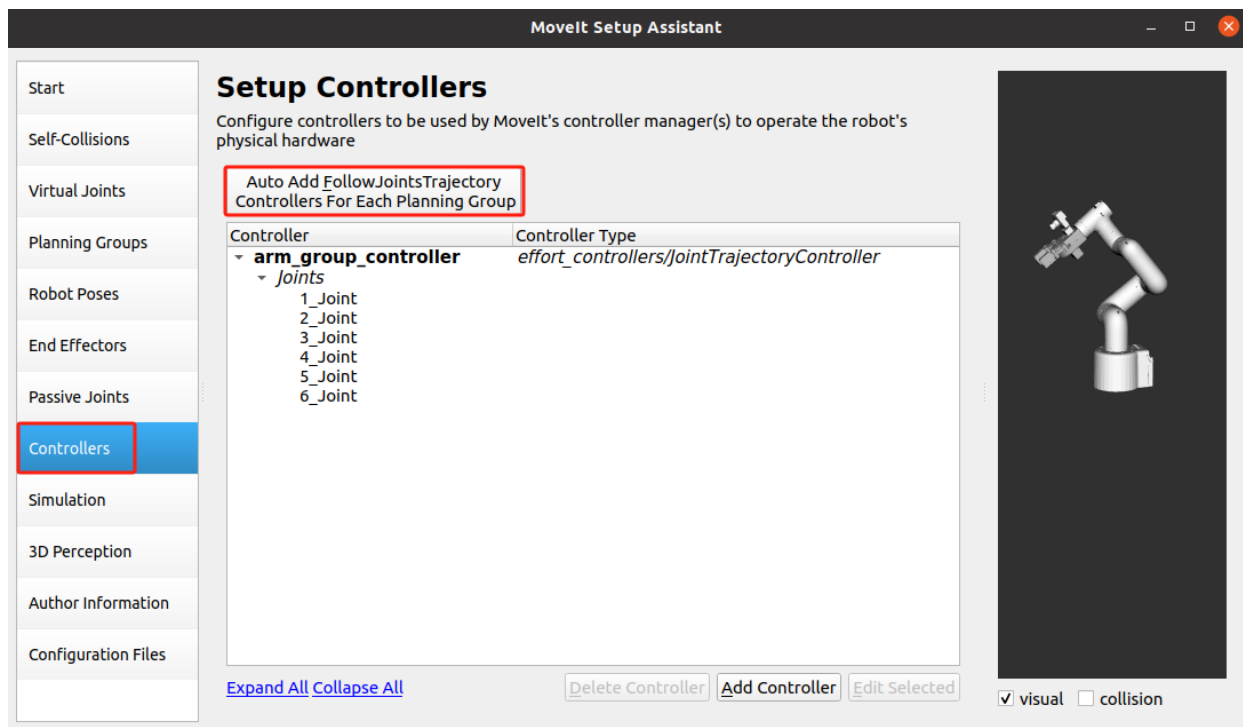
Add End Effector

☒ visual

☐ collision

- Create a ROS controller

Select [ROS Control] and click to automatically add follow joint tracks



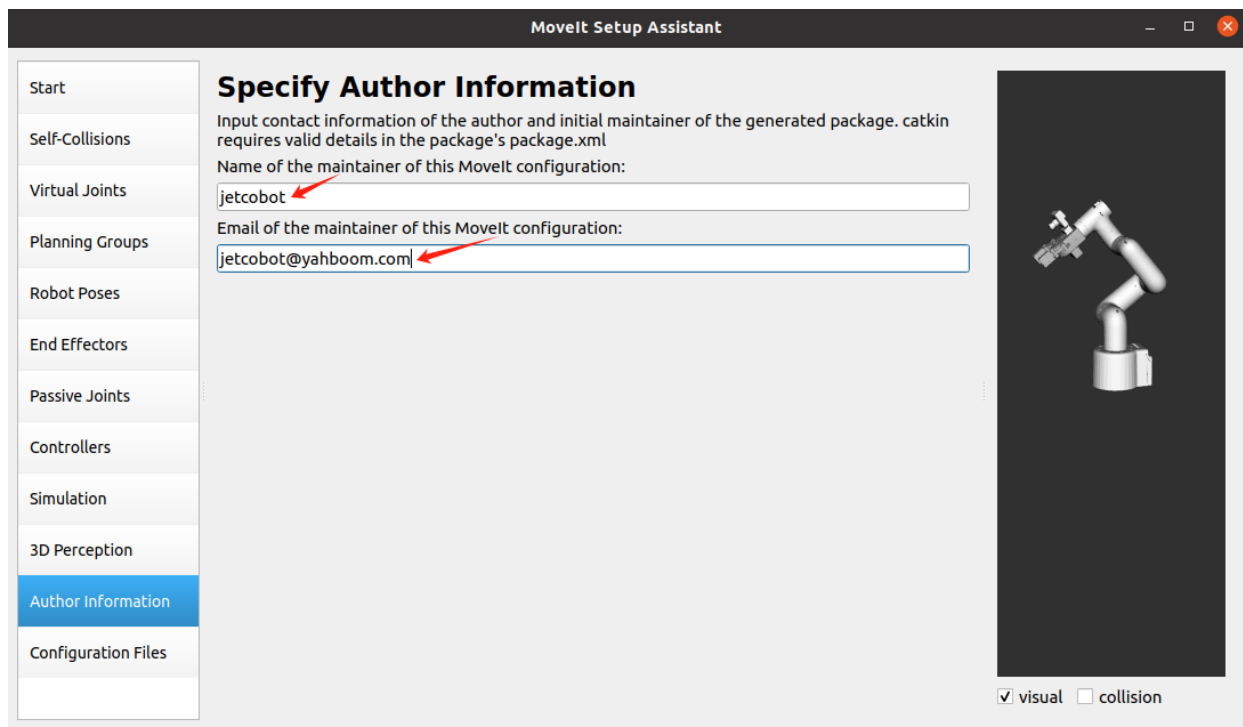
- Add simulation

Select [Simulation], click [Generate URDF]



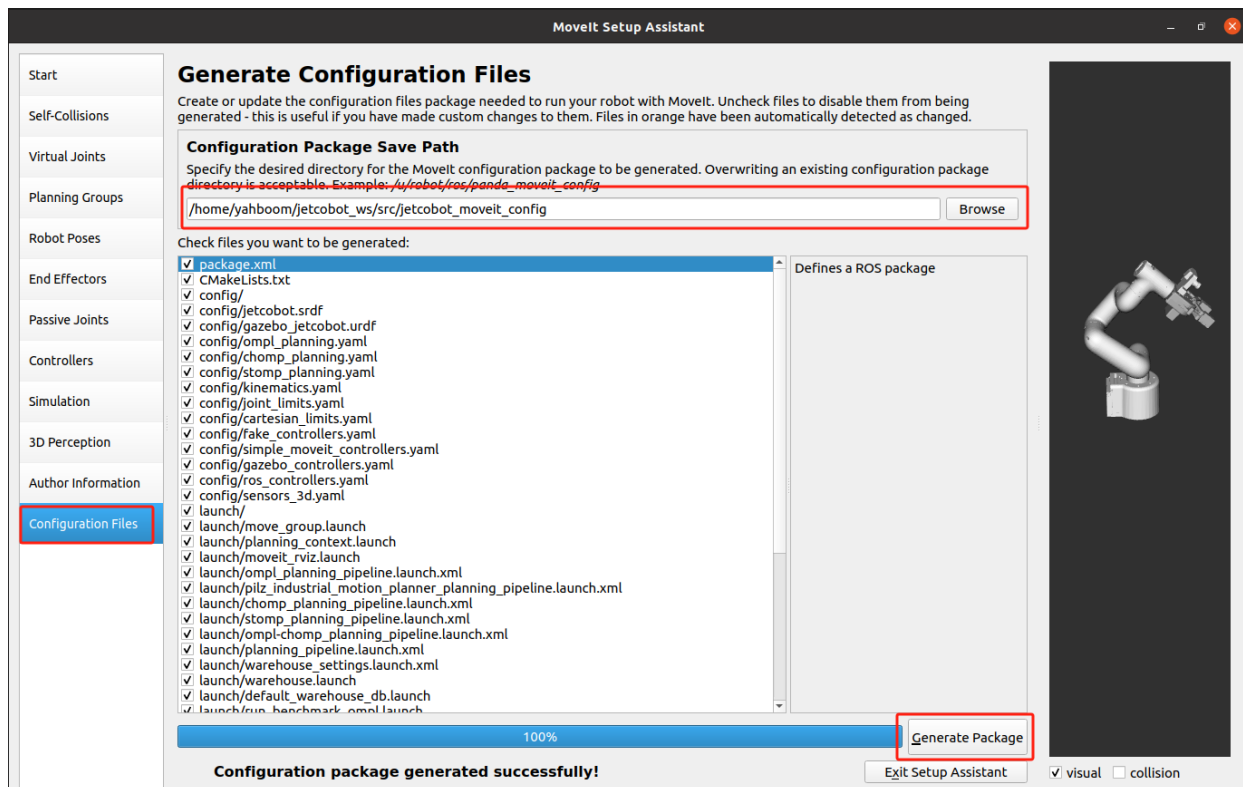
- Add author information. If you don't add it, it won't be generated.

Select [Author Information] and add the following content.



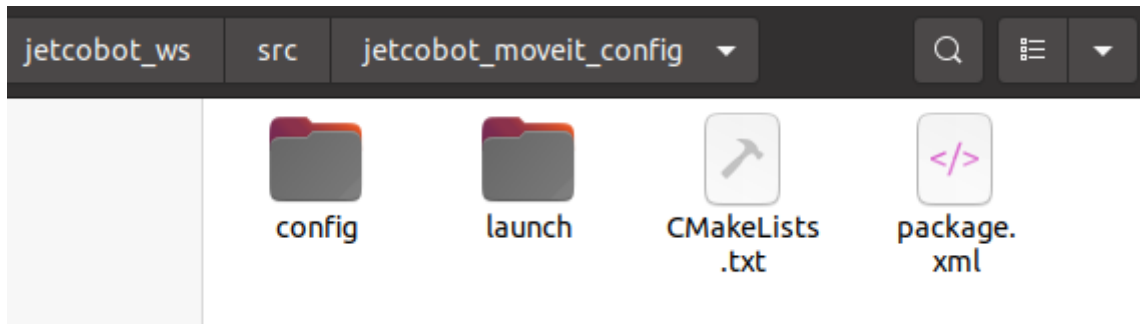
- Generate configuration files

Select [Configuration Files], click [Browse], select the folder you want to place it in (the folder must be empty), click the [Generate Package] button to generate the configuration file. When finished, click [Exit Setup Assistant]



4. Detailed explanation of MoveIt configuration package

Open the newly created [jetcobot_moveit_config] folder and we will find two folders: config and launch.



Config folder

- fake_controllers.yaml: This is the virtual controller configuration file, which allows us to run MoveIt without a physical robot or even without any simulator (such as gazebo) turned on.
- joint_limits.yaml: This records the position, velocity and acceleration limits of each joint of the robot, which will be used in future planning.
- kinematics.yaml: Things set by the motion planning group, used to initialize the kinematics solution library
- jetcobot.srdf: This is an important MoveIt configuration file.
- ompl_planning.yaml: This is where various parameters of various OMPL algorithms are configured.

launch folder

- demo.launch: demo is the summary point of the run. When we open it, we can see that it includes other launch files.
- move_group.launch: As the name implies, the function of move group is to make a planning group move. The default is to use the ompl motion planning library. The others are to set some basic parameters, which can be skipped for now.
- planning_context.launch: Here we can see that the urdf and srdf files used, as well as the kinematic solution library, are defined. It is not recommended to change these manually, but if you need to use different urdf, srdf, you can change them here.
- setup_assistant.launch: If you need to change some configurations, you can run it directly.

5. Configuration validation

Enter the workspace where the configuration file is located and execute the following command

```
cd ~/jetcobot_ws/           # Enter the workspace
catkin_make                 # Compile
source devel/setup.bash     # Update system environment
roslaunch jetcobot_moveit_config demo.launch
# Start moveIT
```

As shown below.

