MoveIt inverse kinematics design

1. Usage environment

Motherboard: Jetson Orin Nano/Nx

ROS2: Humble

2. Driving the real machine

Driving the real machine is to convert the joint state information of the robot arm into the control of the real robot arm by subscribing to the `/joint_states` topic of Moveit2.

Note: Since the real robot arm does not have an obstacle avoidance function, some positions may encounter obstacles; so the planned robot arm movements should be as reasonable as possible and avoid positions with obstacles

(It is recommended to use preset positions to demonstrate driving the real machine)

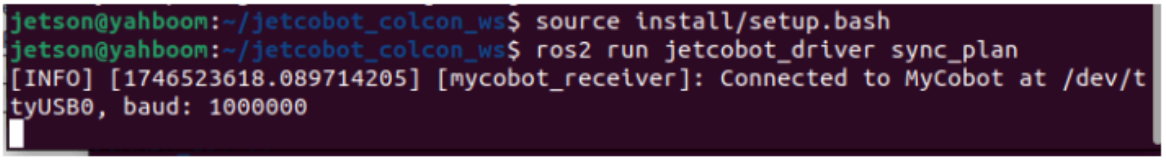
2.1. Start the real machine

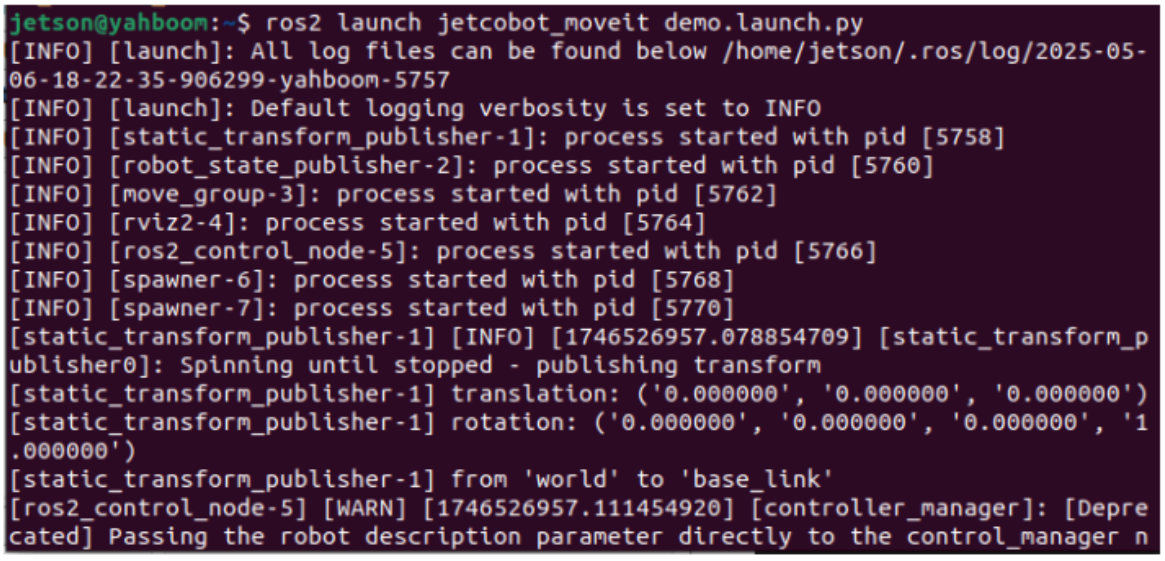
If you do not drive the real machine, simulate the robot arm movements in MoveIt:

```
ros2 run jetcobot_driver sync_plan
```

2.2. Start MoveIt2

```
ros2 launch jetcobot_moveit demo.launch.py
```





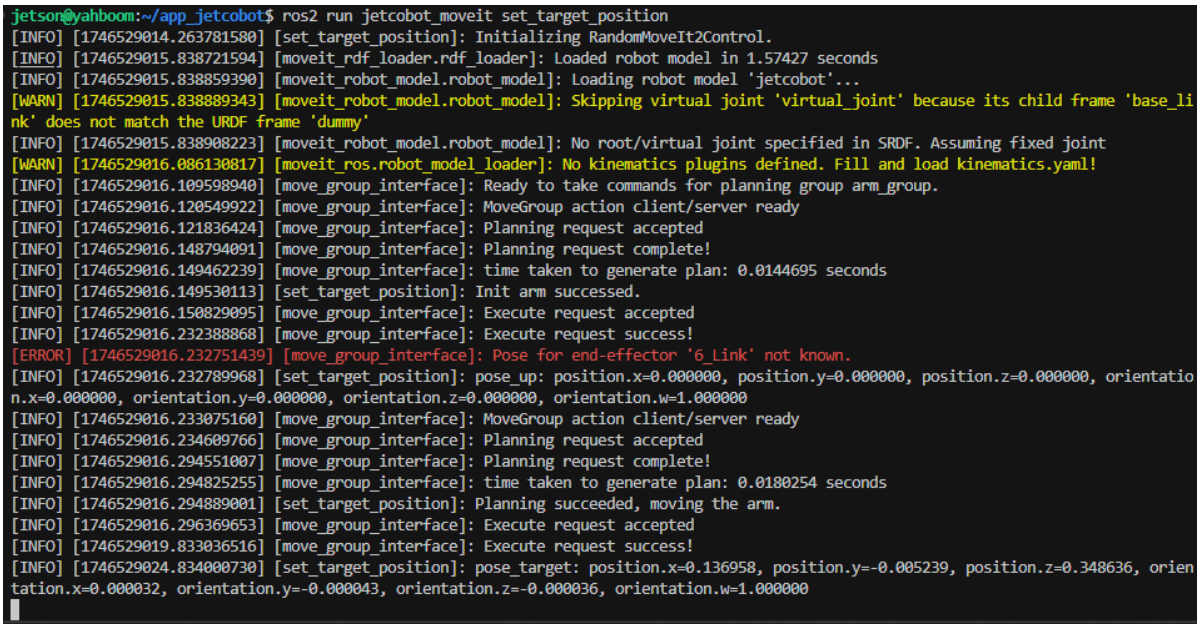
3. Inverse kinematics design

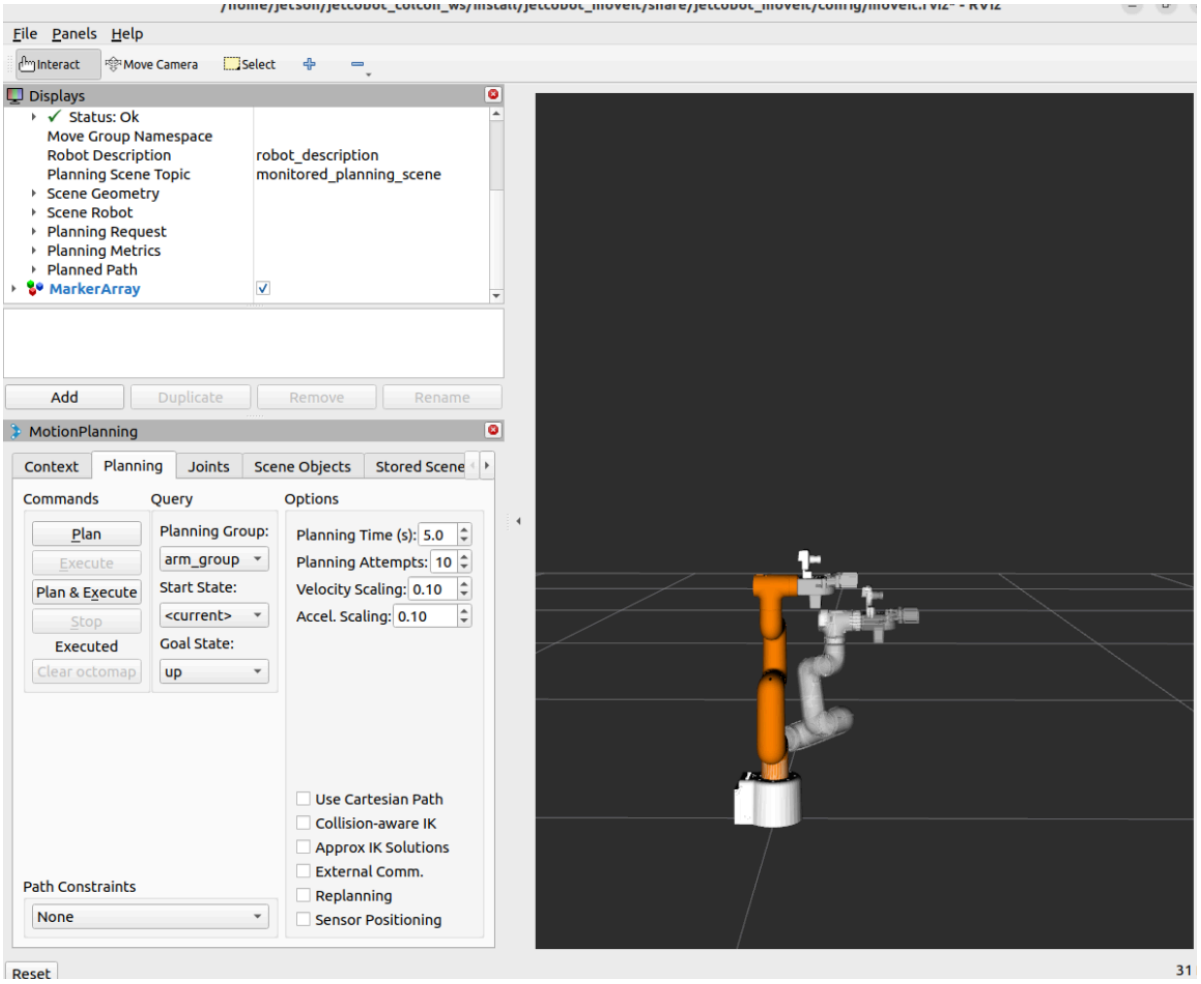
Inverse kinematics refers to the position of the end effector of the given robot arm, MoveIt calculates the angles of each joint of the robot arm, and then plans to the target position by itself.

Start command

The robot needs to be successfully loaded in MoveIt and `You can start planning now!` appears to run the following command: The robot will plan to the position by itself

```
ros2 run jetcobot_moveit set_target_position
```



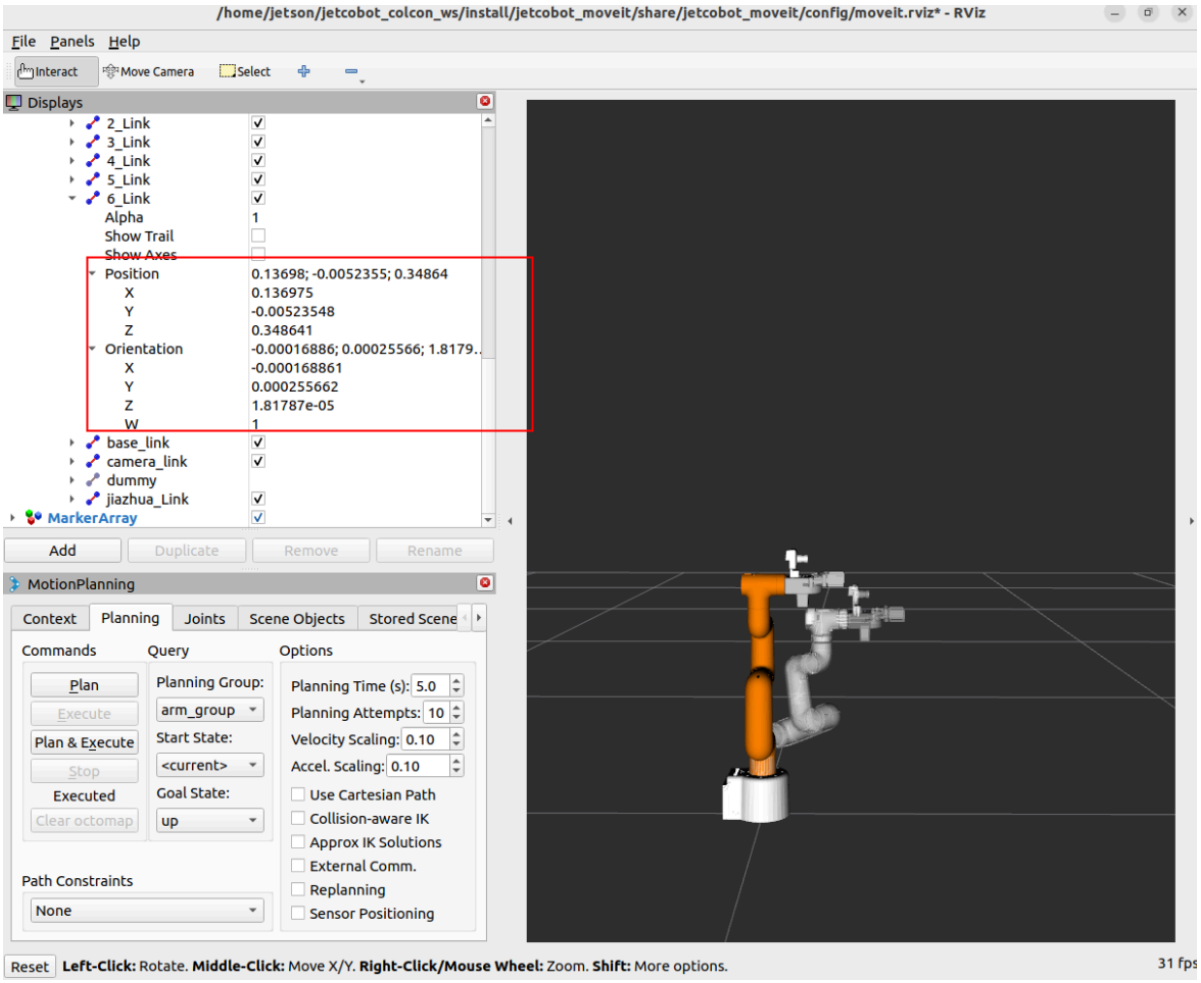


Notes

Since the position of the end effector of a given robot is difficult to determine, it is recommended that users follow the following method to find a reachable pose.

Check the pose of `6_Link` and then replace the target pose in the program: the data changes only after the robot needs to be planned and executed to the specified pose.

`MotionPlanning` → `Planned Path` → `Links`: Positions, Orientation


/home/jetson/jetcobot_colcon_ws/install/jetcobot_moveit/share/jetcobot_moveit/config/moveit.rviz* - RViz
File Panels Help
Interact Move Camera Select
Displays
2_Link
3_Link
4_Link
5_Link
6_Link
Alpha 1
Show Trail
Show Axes
Position 0.13698; -0.0052355; 0.34864
X 0.136975
Y -0.00523548
Z 0.348641
Orientation -0.00016886; 0.00025566; 1.8179..
X -0.000168861
Y 0.000255662
Z 1.81787e-05
W 1
base_link
camera_link
dummy
jiazhua_Link
MarkerArray
Add Duplicate Remove Rename
MotionPlanning
Context Planning Joints Scene Objects Stored Scene
Commands
Plan
Execute
Plan & Execute
Stop
Executed
Clear octomap
Query
Planning Group:
arm_group
Start State:
<current>
Goal State:
up
Options
Planning Time (s): 5.0
Planning Attempts: 10
Velocity Scaling: 0.10
Accel. Scaling: 0.10
Use Cartesian Path
Collision-aware IK
Approx IK Solutions
External Comm.
Replanning
Sensor Positioning
Path Constraints
None
Reset
Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.
31 fps