

Movelt 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
```

```
jetson@yahboom:~/jetcobot_colcon_ws$ source install/setup.bash
jetson@yahboom:~/jetcobot_colcon_ws$ ros2 run jetcobot_driver sync_plan
[INFO] [1746523618.089714205] [mycobot_receiver]: Connected to MyCobot at /dev/ttyUSB0, baud: 1000000
```

```
jetson@yahboom:~$ ros2 launch jetcobot_moveit demo.launch.py
[INFO] [launch]: All log files can be found below /home/jetson/.ros/log/2025-05-06-18-22-35-906299-yahboom-5757
[INFO] [launch]: Default logging verbosity is set to INFO
[INFO] [static_transform_publisher-1]: process started with pid [5758]
[INFO] [robot_state_publisher-2]: process started with pid [5760]
[INFO] [move_group-3]: process started with pid [5762]
[INFO] [rviz2-4]: process started with pid [5764]
[INFO] [ros2_control_node-5]: process started with pid [5766]
[INFO] [spawner-6]: process started with pid [5768]
[INFO] [spawner-7]: process started with pid [5770]
[static_transform_publisher-1] [INFO] [1746526957.078854709] [static_transform_publisher0]: Spinning until stopped - publishing transform
[static_transform_publisher-1] translation: ('0.000000', '0.000000', '0.000000')
[static_transform_publisher-1] rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
[static_transform_publisher-1] from 'world' to 'base_link'
[ros2_control_node-5] [WARN] [1746526957.111454920] [controller_manager]: [Deprecated] Passing the robot description parameter directly to the control_manager n
```

3. Inverse kinematics design

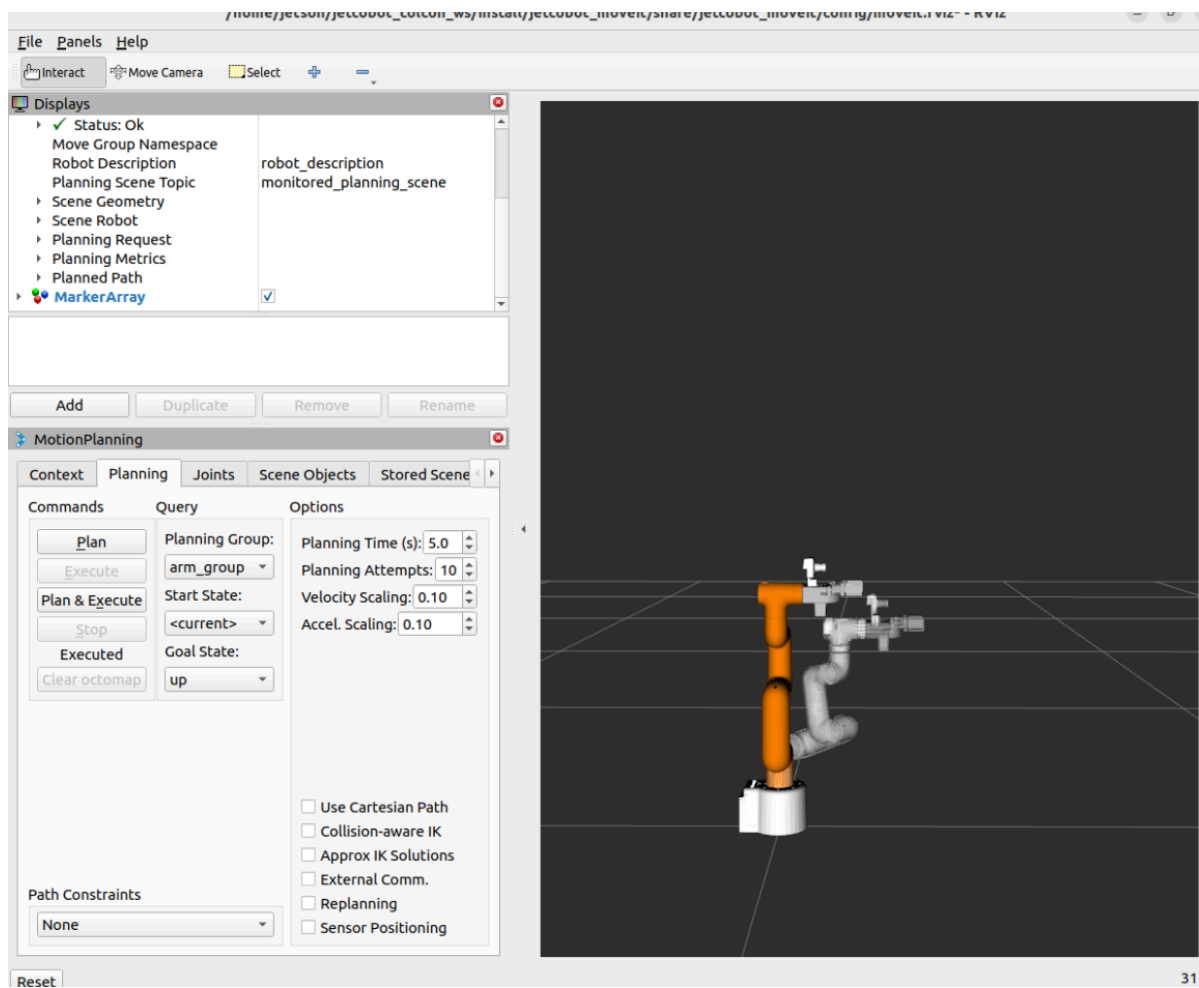
Inverse kinematics refers to the position of the end effector of the given robot arm, MoveIt calculates the angles of each joint of the robot arm, and then plans to the target position by itself.

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
```

```
jetson@yahboom:~/app_jetcobots$ ros2 run jetcobot_moveit set_target_position
[INFO] [1746529014.263781580] [set_target_position]: Initializing RandomMoveIt2Control.
[INFO] [1746529015.838721594] [moveit_rdf_loader.rdf_loader]: Loaded robot model in 1.57427 seconds
[INFO] [1746529015.838859390] [moveit_robot_model.robot_model]: Loading robot model 'jetcobot'...
[WARN] [1746529015.838889343] [moveit_robot_model.robot_model]: Skipping virtual joint 'virtual_joint' because its child frame 'base_link' does not match the URDF frame 'dummy'
[INFO] [1746529015.838908223] [moveit_robot_model.robot_model]: No root/virtual joint specified in SRDF. Assuming fixed joint
[WARN] [1746529016.086130817] [moveit_ros.robot_model_loader]: No kinematics plugins defined. Fill and load kinematics.yaml!
[INFO] [1746529016.109598940] [move_group_interface]: Ready to take commands for planning group arm_group.
[INFO] [1746529016.120549922] [move_group_interface]: MoveGroup action client/server ready
[INFO] [1746529016.121836424] [move_group_interface]: Planning request accepted
[INFO] [1746529016.148794091] [move_group_interface]: Planning request complete!
[INFO] [1746529016.149462239] [move_group_interface]: time taken to generate plan: 0.0144695 seconds
[INFO] [1746529016.149530113] [set_target_position]: Init arm succeeded.
[INFO] [1746529016.150829095] [move_group_interface]: Execute request accepted
[INFO] [1746529016.232388868] [move_group_interface]: Execute request success!
[ERROR] [1746529016.232751439] [move_group_interface]: Pose for end-effector '6_Link' not known.
[INFO] [1746529016.232789968] [set_target_position]: pose up: position.x=0.000000, position.y=0.000000, position.z=0.000000, orientation.x=0.000000, orientation.y=0.000000, orientation.z=0.000000, orientation.w=1.000000
[INFO] [1746529016.233075160] [move_group_interface]: MoveGroup action client/server ready
[INFO] [1746529016.234609766] [move_group_interface]: Planning request accepted
[INFO] [1746529016.294551007] [move_group_interface]: Planning request complete!
[INFO] [1746529016.294825255] [move_group_interface]: time taken to generate plan: 0.0180254 seconds
[INFO] [1746529016.294889001] [set_target_position]: Planning succeeded, moving the arm.
[INFO] [1746529016.296369653] [move_group_interface]: Execute request accepted
[INFO] [1746529019.833036516] [move_group_interface]: Execute request success!
[INFO] [1746529024.834000730] [set_target_position]: pose_target: position.x=0.136958, position.y=-0.005239, position.z=0.348636, orientation.x=0.000032, orientation.y=-0.000043, orientation.z=-0.000036, orientation.w=1.000000
```



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

RViz interface showing a robotic arm model and its configuration.

Displays Panel:

- 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 ✓

MotionPlanning Panel:

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

Path Constraints: None

Use Cartesian Path ☐
Collision-aware IK ☐
Approx IK Solutions ☐
External Comm. ☐
Replanning ☐
Sensor Positioning ☐

Reset Left-Click: Rotate. Middle-Click: Move X/Y. Right-Click/Mouse Wheel: Zoom. Shift: More options.

31 fps