

Pick-and-Place Manipulation Using a Robotic Arm in ROS 2

Team members: Xiangzhang (ISU:508513)

Chen Yuheng (ISU: 500419)

1.Project goal and problem statement

Goal:

Develop a simple robotic arm control system that performs a *pick-and-place* task in a simulated environment using ROS 2 (Humble/Foxy).

- (1) The arm should be able to:
- (2) Move from a home position to a “pick” position,
- (3) Grasp a small object (cube or cylinder),
- (4) Transfer it to a defined “place” position,
- (5) Release it safely.

Problem Statement:

Design and implement software capable of controlling a manipulator to perform a sequence of coordinated motions (trajectory planning, gripper control, position feedback) within simulated constraints. The project focuses on:

- (1) Understanding ROS 2 node communication and message types,
- (2) Applying MoveIt 2 motion planning principles,
- (3) Creating a reproducible simulation for testing robot control logic.

2.Methods and Algorithms

- (1) Robot Model & Environment: Load URDF/Xacro model of a simple 6-DOF manipulator

(ex: UR5, Panda, or a custom arm)

ROS 2 Tools / Libraries: Gazebo

(2) Motion Planning: Use MoveIt 2 to plan trajectory from start→pick→place→home

ROS 2 Tools / Libraries: MoveIt 2 (Planning Group + MotionPlanner API)

(3) Gripper Control: Open/close by publishing to gripper joint topic

ROS 2 Tools / Libraries: ROS 2 publisher node

(4) Task Sequencing: Implement a simple FSM (Finite State Machine) or Python script controlling sequence

ROS 2 Tools / Libraries: “rclpy” ,“python-launch”

(5) Visualization: Use RViz 2 markers or Gazebo camera to show task execution

ROS 2 Tools / Libraries: RViz 2

3. Team Members and Task Distribution

Member1(Xiangzhang): System Modeling

Main tasks: Set up environment, URDF integration, MoveIt 2 configuration package, testing trajectory execution

Member2(Chen Yuheng): Implement the pick-and-place logic node in Python, design the finite state sequence, prepare presentation and video demo