Delft University of Technology

RO Internship report

Development of fundamental packages for a 5-DOF manipulator in ROS2

Authors:	Yulei Qiu
Student Number:	5233178
Supervisors:	Mr. Jinfa Chen (Lenovo Research Shanghai)
	Dr.-Ing. Jens Kober (Delft University of Technology)
Internship peroid:	July 2021 - October 2021

February 9, 2023

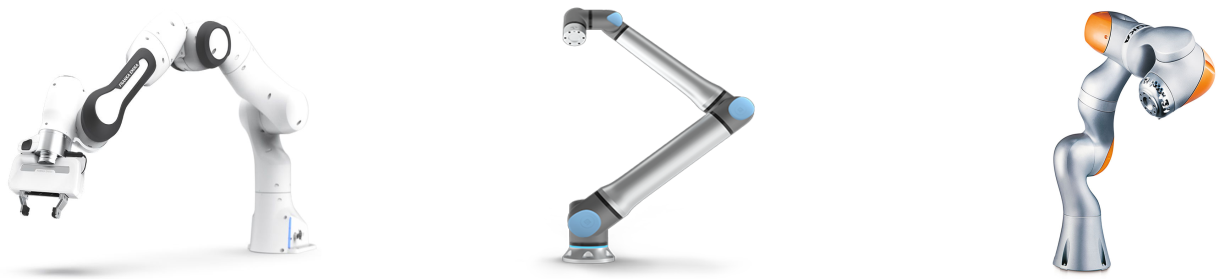
TU Delft Delft University of Technology

Contents

Chapter 1

Introduction

In robotics, a manipulator is a device used to manipulate materials without direct physical contact by the operator. Common commercial industrial robots are serial-link manipulators [1]. Figure 1.1 shows some common commercial manipulators. Traditional manipulators, especially early industrial robotic arms were able to automate repetitive tasks, and they were big, bulky, and highly specialized. Several decades later, robotic arms has become smaller, and application field of manipulator has changed from heavy industry and massive production to novel scenarios where robot tasks require dynamic operation and movement, such as flexible production lines, human assistance in service and medical areas as well as elderly care and rescue [2]. These tasks require mobility as well as manipulation capabilities which former robotic systems would not provide. That's why mobile robotics is gaining popularity in the recent years.



(a) Franka Emika Panda (b) Universal Robots (c) KUKA LBR iiwa

Figure 1.1: Common commerical industrial robots

A mobile manipulator system usually contains a robotic manipulator arm mounted on a mobile platform. Figure 1.2 shows some industrial mobile manipulators. Such systems unite the advantages of mobile platforms and robotic manipulator arms. Autonomous mobile manipulators have been used in many different areas, e.g. space exploration and military operations. The goal of autonomous mobile manipulation is the execution of complex manipulation tasks in unstructured and dynamic environment. Industrial inspection is one of those scenarios where autonomous mobile robots play an important role. Many industrial inspection tasks place in areas that are not only difficult to reach, but may also present safety and health hazards to human inspectors. Autonomous mobile robots are therefore used to take measurements, perform visual inspections, or to conduct surveillance in these tasks.

The application scenarios of autonomous inspection robots in the electrical industry include electrical substations, converter stations, distribution stations, etc. The inspection of electrical substations will gradually become intelligent and autonomous, and hence the overall market scale of intelligent inspection robots is promising in the future. The robot should be equipped with detection and sensing devices like infrared thermal camera and visible light camera to complete the inspection tasks in high-voltage substations. Besides, it should be able to work continuously to discover damage, heat, oil leakage and other internal and external issues, and to provide accurate diagnosis and analysis of the hidden danger of the substation facilities.

During my internship, I participated the development of a 5-DOF manipulator. It is part of a project to develop a quadruped robot inspection system used in an electrical substation. The quadruped robot is equipped with several sensors and a manipulator is mounted on its back. The choice of the manipulator is important since it should be as light as possible and have sufficient accuracy. Our department is trying to develop a manipulator compatible with ROS that meets our needs on both weight and precision. Therefore, the purpose of this internship is to develop the prototype of a light-weighted manipulator with enough accuracy. I was responsible for the software level of the manipulator, developing the corresponding ROS packages such as launching the manipulator and loading the controllers. All the packages are available here.

(a) Lenove Daystar

(b) RB-KAIROS mobile manipulator

(c) Omron mobile manipulator

Figure 1.2: Autonomous mobile manipulators

Chapter 2

Activities

I participated the development of a prototype of a robotic manipulator from the software level. My partner responsible for hardware designed the structure of the manipulator and chose the motors. The structure of the manipulator is shown in Figure 2.1. There are five joints for the manipulator. Each joint is equipped with a motor except joint 2 with two motors. We choose planetary reducer motors from MINTASCA [3], which are light-weighted and have enough accuracy, and is compatible with the bus of the existing quadruped robot. In addition, this type of motor has open API interface and corresponding manuals. Besides, the motor is compatible with C++ and ROS, which makes it easy to debug. For the development, we choose ROS2 because it supports real-time programming better, which is important in the tasks of a manipulator. The assembled manipulator is shown in Figure 2.2a.

Therefore, my main work is to develop the necessary ROS2 packages for a manipulator. After discussion with my supervisor, Jinfa Chen and reference to the repository of mainstream manipulators, we finally developed the following packages:

- Description: This package contains the Unified Robot Description Format (URDF) and 3D model files for the manipulator. URDF is an XML format for representing a robot model, such as links, joints, transmissions and sensors. The 3D model files are stored in STL format, describing the shape of the robot. The URDF files are written in 'xacro' format so that users have the ability to customize which part could be loaded to the parameter server.

- Hardware Interface: This package contains the hardware interface of the manipulator. Hardware Interfaces are used by ROS control in conjunction with several ROS controllers such as position, velocity and joint state controller to send and receive commands to hardware.

- Bringup: This package contains launch files and runtime configurations for the manipulator. The launch files integrate controllers and/or Moveit 2.

- Moveit Configuration: This package contains the necessary configuration files to get the manipulator worked within MoveIt 2. The user is allowed to choose whether to have MoveIt work with physical robot hardware or a fake hardware.

2.1 Description

The Description package can launch an RViz window for the visualization of the manipulator. Figure 2.2b shows the visualization environment. The GUI allows users to drag the sliders the change the joint states of the manipulator.

2.2 Hardware Interface

This package integrate the ROS hardware interface and motor's SDK. It contains the driver of the actuator. The motion of the manipulator is realized by the rotation of each joint, i.e. the rotation of each actuator. This package works with the description so that the manipulator move given a trajectory. A demo video is available here.

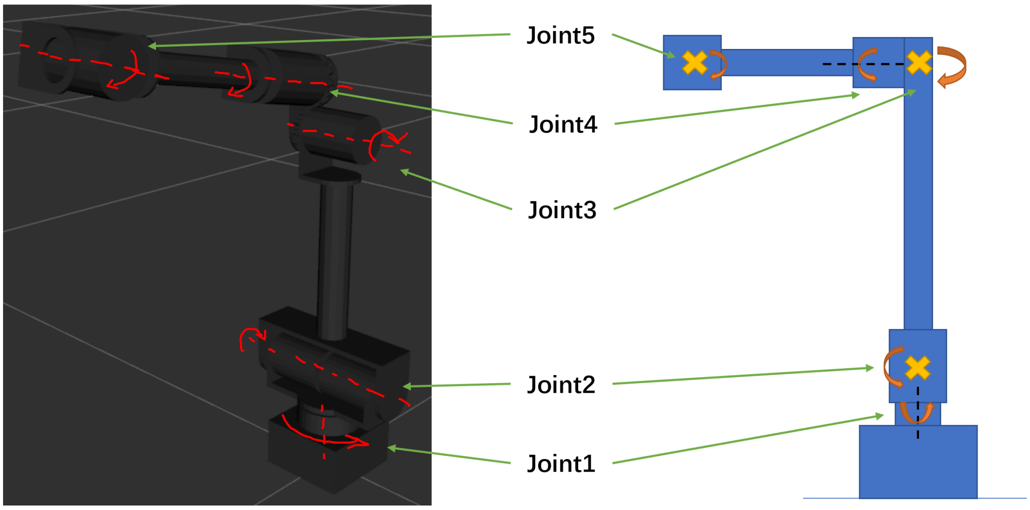
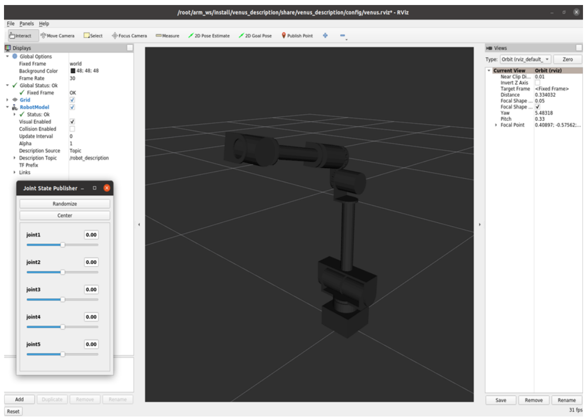


Figure 2.1: The structure of the manipulator

(a) The manipulator in real world



(b) The visualization of the manipulator in RViz with slider to control the joint states

Figure 2.2: The developed manipulator in real world and RViz.

2.3 MoveIt Configuration

This package creates a motion plans in MoveIt using RViZ and the MoveIt Display plugin. The 3D model of the manipulator is also import in the environment. The MoveIt Display plugin allows the user to setup virtual environments (planning scenes), create start and goal states for the robot interactively, test various motion planners, and visualize the output. Figure 2.3 shows the manipulator in the environment as well as the MoveIt plugins. The user can drag the operation ball on the end-effector and the planning task will be executed. A demo video is available here.

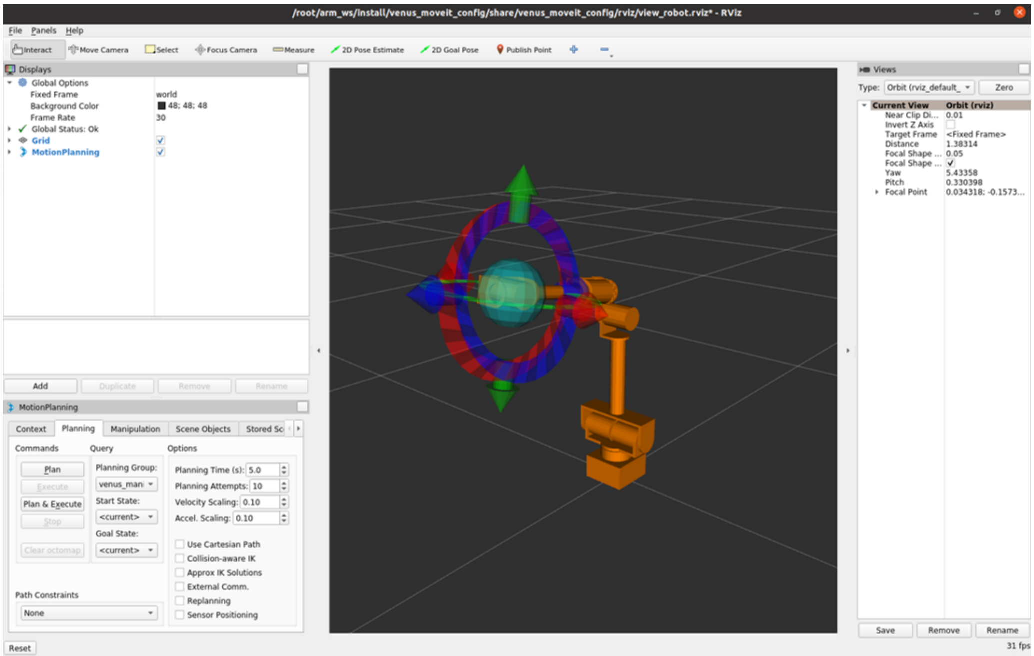


Figure 2.3: The manipulator and MoveIt interface

2.4 Bringup

This package allows the manipulator to run in two ways: controller-loaded and MoveIt-based. In the first way the manipulator can be driven by sending trajectory commands through test nodes. For the second way, the user can plan and move the manipulator in Moveit2.

Chapter 3

Learning Experience

My supervisor in the company set up some learning goals for me during the internship:

1. Version control tool: We use Gitlab during the intern for version control. And I was required to commit everyday to record the progress.
2. Text editor: The goal is to learn at least one terminal-based text editor. Jinfa recommended Vim to me. Therefore, I learned to use Vim to edit the code in the terminal directly.
3. Good coding habit and unified style
   (a) Write code with good comment habit.
   (b) Use a unified style. Since most of my code is related to ROS, I took the Python code guide for ROS as reference.
4. Documentation
   (a) I learned markdown syntax for writing documents in Git.
   (b) The other requiremnt is to Write organized, clear documents that easy to follow. A good document should let others reproduce my work according to your documents.
5. Debug ability: Learn at least one IDE for debugging , such as Pycharm, Sublime Text and RoboWare Studio.
6. Algorithm: Solve at least 2 algorithm problems a week. I used Leetcode for the algorithm problems, and I also discussed the problem I cannot solve in the weekly meeting with Jinfa.
7. Learn ROS /ROS2. This is also a requirement for the development of the manipulator. I already knew how to use ROS, so I learned ROS2 throuhg the offical documents easily.
8. Daily log, weekly update. I wrote a work log before getting off the work, and commit the daily log once a week.

Chapter 4

Summary

During my internship experience in Lenovo Research Shanghai with Mr. Jinfa Chen, I was able to develop my development, communication and cooperation skills. I participated in the development of a manipulator prototype from the software level as part of an industrial inspection solution project. My main contribution is the ROS packages that contains the description of a robot, the integration of control and planning as well as the connection to actual hardware. Although I found some tasks challenging, I believe they were valuable in developing my own skills. All the codes are available online here here.

In conclusion, my experience in Lenovo was crucial in my development as a robotics engineer. I will take the lessons and skills I learned and apply them to my next journey, my MSc thesis.

Bibliography

[1] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control*. CRC Press, dec 2003.

[2] M. Sereinig, W. Werth, and L.-M. Faller, "A review of the challenges in mobile manipulation: systems design and RoboCup challenges," *e & i Elektrotechnik und Informationstechnik*, vol. 137, no. 6, pp. 297–308, sep 2020.

[3] "Mintasca." [Online]. Available: https://mintasca.com/