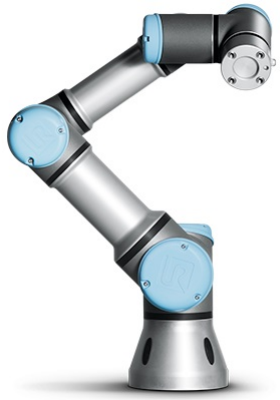
ECE 470
Introduction to Robotics
Lab Manual

Spring 2022



Dan Block
Neeloy Chakraborty
Peixin Chang
Dhruv Mathur
Chuyuan Tao

University of Illinois at Urbana-Champaign

UR3 Python - ROS Interface

Preface

This is a set of laboratory assignments designed to complement the introductory robotics lecture taught in the College of Engineering at the University of Illinois at Urbana-Champaign. Together, the lecture and labs introduce students to robot manipulators and computer vision along with the Robot Operating System (ROS) and serve as the foundation for more advanced courses on robot dynamics, control and computer vision. The course is cross-listed in three departments (Electrical & Computer Engineering, Aerospace Engineering, and Mechanical Science & Engineering) and consequently includes students from a variety of academic backgrounds.

For success in the laboratory, each student should have completed a course in linear algebra and be comfortable with three-dimensional geometry. In addition, it is imperative that all students have completed a freshman-level course in computer programming. *MODERN ROBOTICS MECHANICS, PLANNING, AND CONTROL* (Kevin M. Lynch and Frank C. Park, 2017) is required for the lectures and will be used as a reference for many of the lab assignments. We will hereafter refer to the textbook as *MR* in this lab manual.

These laboratories are simultaneously challenging, stimulating, and enjoyable. It is the author's hope that you, the reader, share a similar experience.

Enjoy the course!

LAB 1

Introduction to the UR3

1.1 LAB 1 Week One

1.1.1 Important

Read the entire lab before starting and especially the "Grading" section as there are points you will receive by completing certain sections or checkpoints by the end of the lab session(s).

1.1.2 Objectives

The purpose of this lab is to familiarize you with the UR3 robot arm and its industrial programming interface called the teach pendant. In this lab, you will:

- Learn how to turn on and activate the UR3, and work with the teach pendant to create a simple program for the UR3
- Use the teach pendant to turn on and off the suction cup gripper and use the gripper in a program
- Demonstrate a sequence of motions that places one block on top of another.

1.1.3 References

- UR3 Owner's Manual:

  `https://www.universal-robots.com/download/manuals-cb-series/user/ur3/315/user-manual-ur3-cb-series-sw315-english-international-en/`

- UR3 Software Manual:

  `https://www.universal-robots.com/download/manuals-cb-series/software/314/software-manual-cb-series-sw314-english-us-en-us/`

- Universal Robots Academy

  `https://academy.universal-robots.com/free-e-learning/e-series-e-learning/e-series-core-track/`

1.1.4 Pre-Lab

Before you come to lab it is very important that you go through the training videos found at Universal Robots website `https://academy.universal-robots.com/free-e-learning/e-series-e-learning/e-series-core-track/`. These training sessions get into some areas that we will not be using in this class (for example you will not be changing safety settings), but go through all of the assignments as they will help you get familiar with the UR3 and its teach pendant. You also may want to reference these sessions when you are in lab.

1.1.5 Task

Using the teach pendant, each team will "program" the UR3 to pick and place blocks. The program may do whatever you want, but all programs must check three predefined locations for two blocks and stack one block on top of another at a fourth predefined position. You will use the gripper's suction feedback to determine if a block is located at one of the three starting block locations. The blocks must be aligned with each other in the stack of two.

1.1.6 Procedure

1. The Pre-Lab asked you to go through the basic UR3 training at Universal Robots website. This training should have shown you how to make simple programs to move the UR3. Initially your TA will demonstrate how to turn on and enable the UR3 as well as how to use the emergency stop button. Then use this lab time to familiarize yourself with the UR3 robot. First play around with simple programs that move the robot between a number of points.

2. To turn on the suction for the suction cup gripper, **Digital output 0** needs to be set high. Set low to turn off the suction. Also **Digital input 0** indicates if the suction cup is gripping something. It will return 1 if it is gripping an object and 0 if not. Modify your above program (or make a new one) to add activating on and off the suction cup gripper.

3. Create a program that defines four spots on the robot's table. Three of these spots are where it is possible a block will be initially located and with a certain orientation. There will only be two blocks. The user will place the blocks in two of the positions. The goal for the robot is to collect the two blocks and stack them on top of each other in the fourth define place on the robot's table. So you will need to use the suction cup gripper's feedback that indicates whether an object is being gripped or

not. Then with some "**If**" instructions complete this task such that the user can put the two blocks in any of the three starting positions. When you are finished, you will demo your program to your TA showing that your program works when two blocks are placed and aligned in the three different configurations, and also does not have a problem if only one block or even no blocks are placed at their starting positions. Tips for creating this program:

- To turn on the suction cup, use the **Set** command and select **Digital Output 0** and turn it on or true. Set it to off or false to turn off the suction.
- **Digital Input 0** indicates if something has been gripped by the suction cup. Go to the **I/O** tab and turn on and off **Digital Output 0** and check which state of Digital Input 0 indicates gripped and upgripped.
- In the Structure tab under Advanced besides "**If ... else**", you may also want to use the Assignment to create a global worker variable that, for example, stores the number of blocks collected. In addition the **SubProg** item creates a subroutine that you may call when performing the same steps. The subroutine's scope allows it to see the variables you create with the **Assignment** item.
- You may want to name your **Waypoints**. This makes your program easier to read. In addition if the robot needs to go to the same point multiple times in your program you can command it to go to the same waypoint name.
- Under the Structure tab you can use the **Copy** and **Paste** buttons to copy a line of code and past it in a different subsection of your code. This cuts down on extra typing. Also note the **Move** up and down buttons along with the **Cut** and **Delete** buttons. **Suppress** is like commenting out a line of code.
- When you add an "**If**" statement and then click on the **Command** tab, tap in the long white box to pull up the keyboard for entering the if condition.

4. Demo this working program to your TA. Your TA may ask you to improve your positioning if the stack does not end up aligned well.

1.1.7 Report

None required. Look at Lab 1 Week Two and Start the longer reading assignment for Lab 2's pre-lab.

1.1.8 Demo

Show your TA the program you created.

1.1.9 Grading

- 2.5 points, attendance.
- 47.5 points, successful demonstration.