Assignment 1 – Robot Programming

Introduction

The task of programming an industrial robot involves new concepts and issues when compared with conventional software programming.

In robot programming a major concern is the movement on physical equipment, therefore special attention has to be paid to the movement of the robot.

Offline robot programming offers the possibility to visualize in a 3D environment the behavior of an industrial robot for a given code. Therefore, it is possible to optimize routines, ensure its robustness and verify that the routines are safe for operators and other pieces of equipment. Then the code can be deployed on the physical industrial robot. For this assignment Robot Studio will be used as the offline programming environment.

Industrial robots are programmed with high level programming languages, which are specific to each manufacturer. Nevertheless, it is common to find similar commands across them. For this ABB robots will be programmed by using RAPID language.

For this assignment use the RAPID manuals provided in Moodle, the presentation file contains the most common buttons you need to use in Robot Studio and template files are given to start to work on each of the tasks

Objectives

- After this assignment you should understand the main terms and concepts involved in the task of programming an industrial robot.
- Get practice with high level industrial robot programming language, learn the main commands and instructions.
- Interact with a real industrial robot and its components. Learn the basic steps to deploy code on it.

Environment

1. Robot Studio to emulate and practice offline robot programming.

2. RAPID language for programming ABB industrial robots

3. Robotic cell containing: ABB robot IRB 140

Tasks

The assignment consists on three tasks which are described next.

The task two needs to be demoed in the laboratory's ABB IRB140 robots.

Please consult and check the RAPID manuals provided in Moodle.

Task 1 – Welding routine

For this task you will program a 6 DoF ABB robot. Choose any model that accomplish this task. A recommended model is the IRB 1520 robotic arc welder. The physical structure of this manipulator gives it positioning flexibility which is required for welding tasks.

a) Generate a program that follows the external and the internal part of the given piece as illustrated in Figure 1.
b) Make your robot wait for 3 seconds. Then assume another workpiece is placed on top of the workpiece that the robot already processed. Repeat the welding path on the new workpiece.

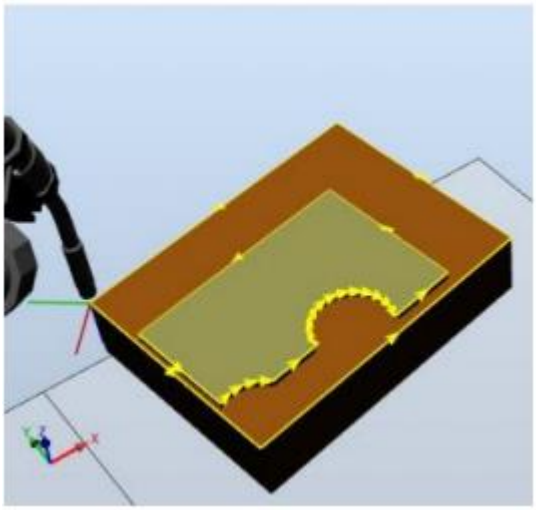

Figure 1. Welding path

Considerations

Be sure that the torch comes from top when it starts the welding at the paths.

Be sure the torch leaves vertically when it finishes welding each of the paths

Use the .rspag projecte provided in Moodle for this task

Task 2 – Pick and Place

Your task is to program one robot, the ABB IRB 140. First create and test your code offline (Robot Studio) and later on the real equipment. For deploying your code in the real equipment, a supervisor will help you to do it. Reservation system is available in Moodle

The template you are given contains the workstation and its components that you will face in reality. This workstation is composed of conveyors and two IRB 140 which are placed upside down, see Figure 2

Program a simple Pick & Place routine for the robot 1. Your robot should pick the red workpiece shown in the Figure 3. It should move it through couple of arbitrary via points and then place it in the green basket located under the robot.

The Pick & Place routine has the next requirements:

1. Create your pick and place targets
2. Define pre-pick/pre-place targets on top of the pick/place targets
3. Move with free joint movement between via points and pre-pick, pre-place targets, speed of V500
4. Move with linear movements between pre-pick/pre-place and the actual pick and place targets with speed of V50
5. After opening/closing the gripper wait for a fraction of a second before you execute the next robot movement

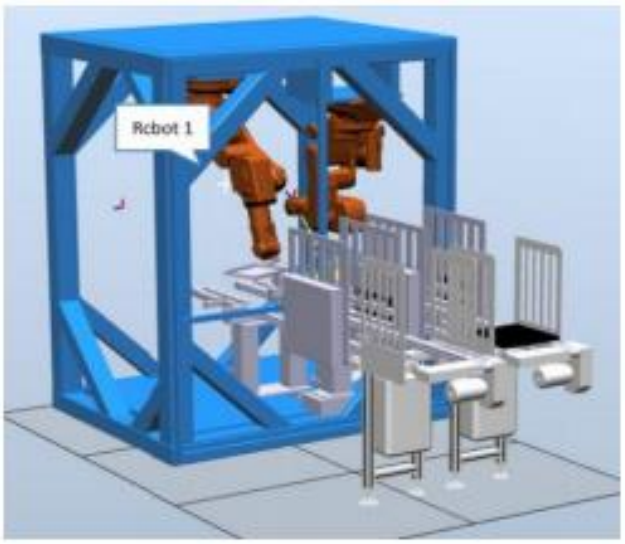

Figure 2. Laboratory workstation

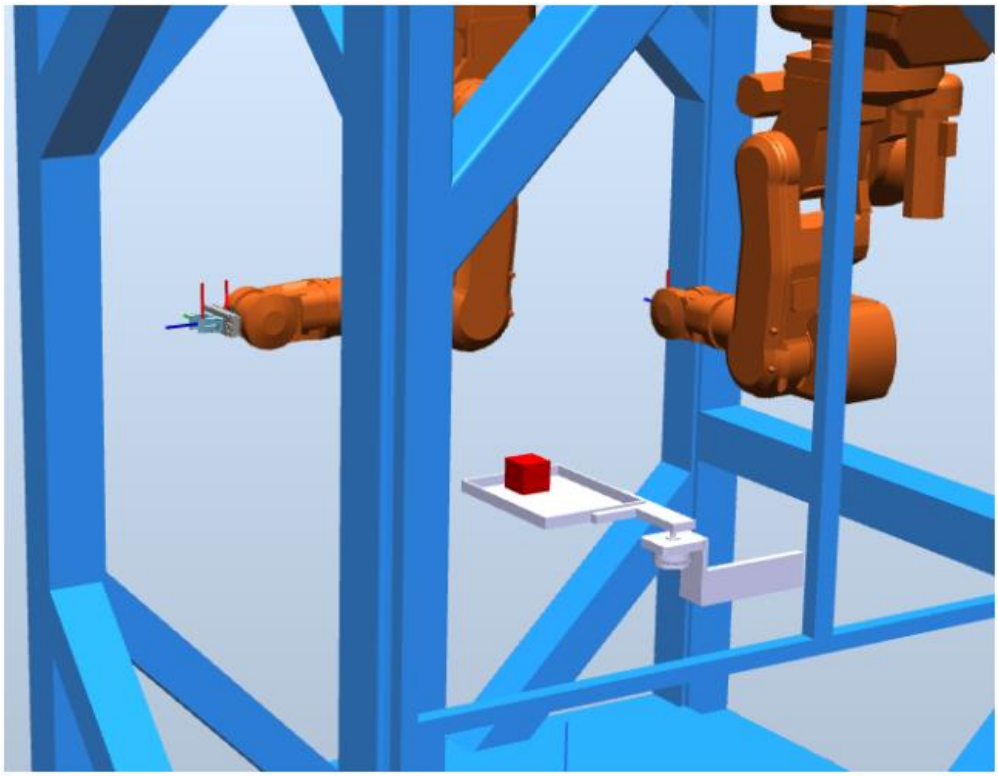

Figure 3. Piece to Pick and Place

Considerations:

Be sure your code does not move the robot close to other components, in order to avoid collisions. This is because the code will actually be deployed on real equipment.

Use the .rspag projecte provided in Moodle for this task

Task 3 – Palletizing robot

Objectives:

- Hands on Robot Studio
- Digital signals (I/O)
- Understand interaction of the Robot controller with sensors/actuators (peripherals)
- Implement palletizing routines with Rapid programming language
- Usage of the measurement tool

Use the **Introduction to Rapid programming and Rapid Technical reference** (provided in Moodle) if needed.

Your task is to program the IRB6620 industrial robot for a palletizing task. This robot can lift big payloads, up to 60 kg objects.

In automation, palletizing refers to stacking products onto a pallet, see Figure 4

Figure 4. Palletizer Robot

For this task, you are given the system components in a Robot Studio Pack and Go file, which can be located in Moodle with the name of **palletizing.rspag** In the RAPID section of that file there is a piece of code with instructions that you can use for testing purposes.

The system consists in a conveyor belt, a palletizing robot, boxes and a pallet, see these components in Figure 5

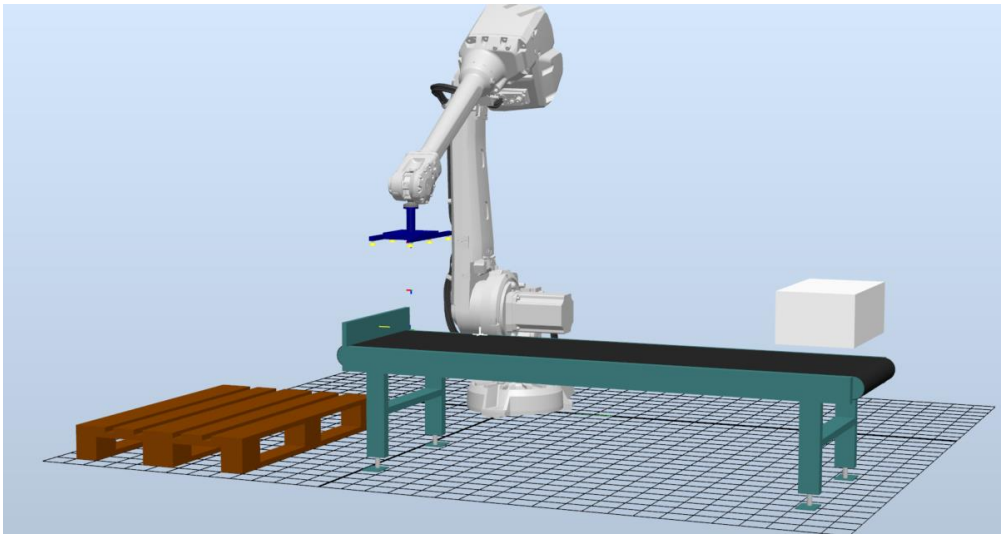

Figure 5 Palletizing system

The boxes are transported to the palletizing robot through the conveyor belt. When the box is ready to be picked from the conveyor, the digital input **diBoxInPos** in the robot controller goes to high (true). The box should be picked from the end of the conveyor and then placed in the pallet, see Figure 6

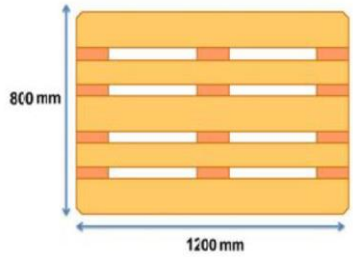

Figure 6 Euro pallet dimensions

Notice that the arrangement of the boxes in the pallet changes, depending on the level. This gives more stability to the load placed in the pallet. Figure 7 shows the dimensions of the boxes and how many levels (layers) of boxes should be placed. This figures also show the arrangement of the boxes in the pallet depending on the level.

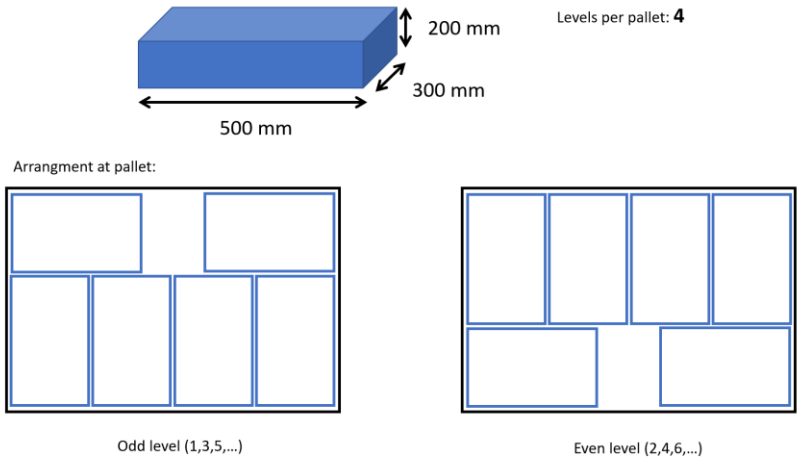

Figure 7 Box dimensions and arrangement at pallet

In order to grasp the boxes, a vacuum gripper is used. Place the gripper on top of the boxes and then activate the digital output **doVacuum** to grasp the box. Turn off the same digital output to release the box.

When the pallet gets completely loaded, the robot controller should stop working

Considerations:

Manipulate the inputs manually, emulating that these signals come from real sensors.

Consider all the boxes come in the center of the conveyor, and their bottom is placed just on top of the conveyor.

Place the boxes in the pallets from top and **avoid collisions** with other boxes.

Task 3.1:

- Unpack the project in **palletizing.rspag** found in moodle
- Run the **sample** rapid code within the Rapid module
- Tip: Use the I/O system to force digital signals manually as depicted in Figure 8

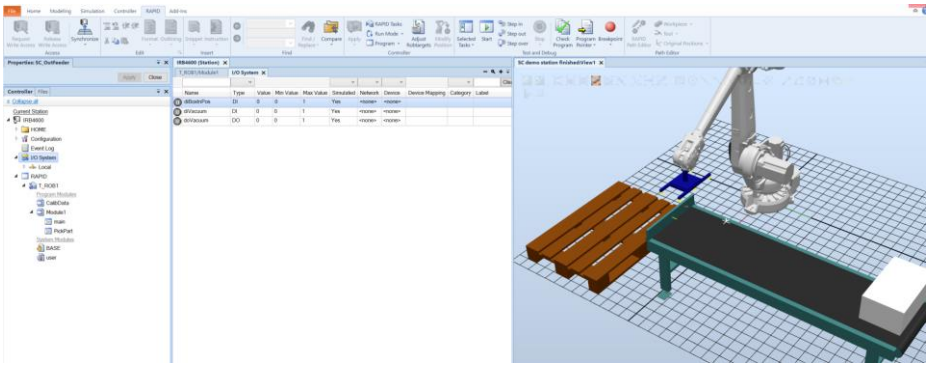

Figure 8. Manipulation of digital signals

Task 3.2:

- Solve the palletizing problem and generate the Rapid code
- **Notice**: There is no need to make the actual simulation of the box moving from one point to another. It is enough that the robot performs the movements to correct positions

- **Tips:**
- Create work objects in the pallets to manipulate more easily your points
- Use the measurement tool in robot studio, to determine distances within the models.
- Use the command Offs to generate points in runtime by using one and then applying on it specific offsets
- Use the creation of different work objects if you think that is an easier approach

Demo and submission deadlines

Work in groups of **three** persons.

In order to complete this assignment, you need to:

- Complete tasks 1, 2 and 3
- Demo task 2 physically
- Through Moodle send the next in a Zip file:
    - PDF report of the three tasks
    - Tasks solutions packaged as .rspag projects
    - Be sure to include your student numbers

There report and .rspag files submission deadline is on **2.12.2022**

Task 2 should be prepared in advance (offline programming) so that in the Hands ON/demo day, your main task is to teach the targets and run the demo.

Days when you can come to try and demo your code are provided below.

	29.11.2022	30.11.2022	1.12.2022	2.12.2022
10:00-10:30	xx	xx	xx	xx
10:30-11:00	xx	xx	xx	xx
11:00:11:30	xx	xx	xx	xx
11:30-12:00	xx	xx	xx	xx
12:00:12:30	xx	xx	xx	xx
12:30-13:00	xx	xx	xx	xx
13:00:13:30	xx	xx	xx	xx
13:30-14:00	xx	xx	xx	xx

In Moodle there is a booking spreadsheet where you can book 30 min slot to come and setup your demo. Book it with just the student number of one of the group members.

In the demo day, all the group members must be present.

If you need to make the demo before these dates, it is possible upon request.

Contact Luis Gonzalez, luis.gonzalezmoctezuma@tuni.fi

*12.12.2022 is a day where some teams will have a chance to demo their task 2. Needs prior approval from instructor.