Introduction to programming of industrial manipulators.

Aigerim Gilmanova, Amarsana Galdanova , Amirreza Darvishzadeh and Siba Issa

September 14, 2021

Figure 1: Kuka KR 10 R1100

Abstract

This report is part of a practical assignment in Dynamics of Non Linear Robotics course for 1^{st} year master students at Innopolis University. The objective of this assignment is to introduce us to the programming of industrial manipulators. We have worked on Kuka KR 10 R1100, and operated it in different coordinate frames and after that we moved the robot along trajectory consisting of different chunks. This a descriptive report of our task and the implementation has done in the lab at Innopolis University and documented as videos and they are available on GitHub.

1 Introduction

Industrial manipulators are designed to improve both the performance and the productivity of production and assembly lines. And controlling the functioning of an industrial manipulator is very necessary to get an easy and precise movement.

2 Theory

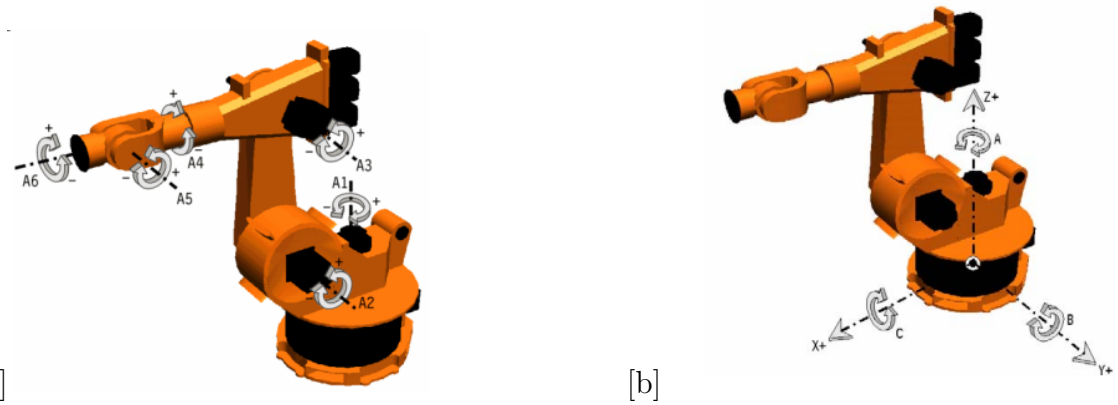
A key requirement in robotics programming is keeping track of the positions and velocities of objects in space. And to do that we need to define a coordinate system.
Coordinates systems are often used to specify the position of a point, but they may also be used to specify the position of more complex figures such as lines, planes, circles or spheres. Now we will talk about the four Coordinate systems of Kuka KR 10 R1100:

1. Axis-Specific motion(Figure 2 (a))
   Each robot axis can be moves individually in a positive or negative direction.

2. WORLD coordinate system(Figure 2(b))
   Fixed, rectangular coordinate system whose origin is located at the base of the robot.

3. Tool coordinate system
   Rectangular coordinate system, whose origin is located in the tool.

4. BASE coordinate system
   Rectangular coordinate system which has its origin on the workpiece that is to be processed.



[a] [b]

Figure 2: Kuka KR 10 R1100 motion types

In addition to the Coordinate systems, it is necessary to talk about KuKa robot motion types, we can divide them into two main categories:

1. Axis-specific motion
   - *PTP (Point To Point)(Figure 2):* The tool is moved along the quickest path to an end point.

2. Path-related motions
   - *LIN (Linear)(Figure 2):* The tool is guided at a defined velocity along a straight line.
   - *CIRC (Circular)(Figure 2):* The tool is guided at a defined velocity along a circular path.

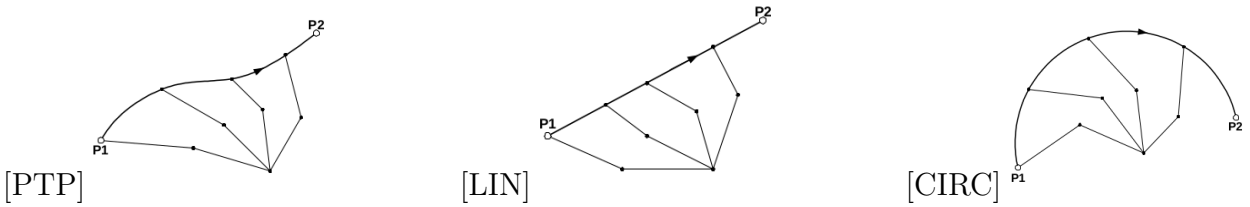


Figure 3: Kuka KR 10 R1100 motion types

3 Procedures

We have used KR 10 R1100 **??** to solve our assignment which consists of two main tasks:

1. Operating the robot in different coordinate frames.

2. Create program that will move robot along trajectory consisting of different chunks

3.1 Operating the robot in different coordinate frames

This task was divided into two main parts; the first one was to move each joint independently in the joint space while the second was to move the end-effector along and around each axis. In order to solve this task we used Kuka Control Panel (KCP) 3.1. First of all, we selected

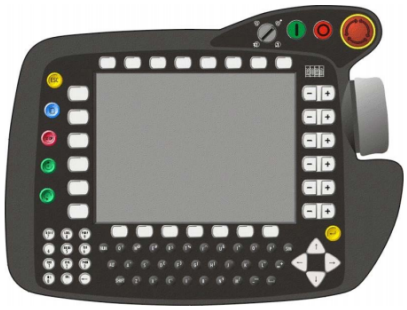


Figure 4: Kuka Control Panel (KCP)

the *Jogging mode*, then we specified the motion to be *Axis-specific motion* then we moved each joint independently by using the keys. After that we moved into the second part of this task by changing the selected mode from *Jogging mode* to *World coordinate system* and moved the end-effector along and around and around X-axis, Y-axis and Z-axis by using the keys as we did in the previous part of this task. We can see below the implemented code.

```
DEF FirstTask()

INI
PIP P1 Vel=30% PDAT4 Tool [1] Base [0]
LIN P2 Vel=2 m/s CPDAT2 Tool[1] Base[0]
LIN P2 Vel=2 m/s CPDAT3 Tool[1] Base[0]
PIP P1 Vel=30% PDAT5 Tool [1] Base [0]
PIP P1 Vel=30% PDAT6 Tool [1] Base [0]
LIN P2 Vel=2 m/s CPDAT4 Tool[1] Base[0]
PTP P4 Vel=30% PDAT7 Tool[1] Base[0]

END
```

3.2 Motion Programming

In this task we created a program that will move the end-effector from point to point and along line and a circle. But it is necessary to mention that we have to define the initial position of the arm at the beginning. And we can in the following lines the impemented code:

```
DEF SecondTask()

INI
PIP P1 Vel-100% PDAT2 Tool [1] Base [0]
LIN P2 Vel=2 m/s CPDAT1 Tool[1] Base[0]
CIRC P3 Vel=2 m/s CPDAT2 Tool[1] Base[0]
PTP P4 Vel=100% PDAT1 Tool[1] Base[0]

END
```

4 Conclusions

Briefly, we have managed to operate Kuka KR 10 R1100 by using the control panel in both joint and world frame. And we also wrote a code to control the motion of the end-effector and make it move from point to point, line and a circle. Our results are documented as videos and they are available on GitHub.

References

www.youtube.com/watch?v=xwzhcihWxio.
www.youtube.com/watch?v=cp87TMZZiME.