

UNIVERSITY OF VERONA

MASTER THESIS

Title . . .

Author:
Michele PENZO
VR439232

Supervisor:
Dr. Diego DALL'ALBA

ICE Lab
Department of Computer Science

December 26, 2020

UNIVERSITY OF VERONA

Abstract

Department of Computer Science

Master of Science

Title ...

by Michele PENZO
VR439232

abstract ...

Acknowledgements

acknowledgements ...

Contents

Abstract	iii
Acknowledgements	v
List of Figures	ix
List of Tables	xi
List of Abbreviations	xiii
1 Introduction	1
1.1 Motivations	1
1.2 Goals	1
1.3 Thesis Overview	1
2 Theory overview	3
2.1 About the robot	3
2.2 Kinematic chains	3
2.2.1 Types of Kinematic Chains	5
2.3 Kinematic	5
2.3.1 Forward Kinematic	5
2.3.2 Inverse Kinematic	5
2.3.3 Denavit–Hartenberg Convention	5
2.4 Theory about robot control	6
2.4.1 Postion control	6
2.4.2 Cartesian impedance control	6
2.4.3 Joint impedance control	6
3 Literature Review	7
3.1 Robot learning from Demonstration	7
3.1.1 Kinesthetic Teaching	7
3.1.2 Teleoperation	7
3.2 Assembling and modulable tasks	7
3.2.1 Assembling tasks	7
3.2.2 Modulable tasks	7
4 The project	9
4.1 Setup overview	9
4.1.1 KUKA Robot Controller	9
4.1.2 Sunrise.Workbench	9
4.2 Project implementation	10
4.2.1 Theach by demonstration	10
4.2.2 Teleoperation	10

5 Experiments and Result	11
5.1 Experiments	11
5.2 Results	11
6 Conclusion	13
6.1 Conclusions	13
6.2 Other possible works	13
Bibliography	15

List of Figures

2.1	The Kuka LBR IIWA	3
2.2	An example of kinematic chain.	4
2.3	A revolute joint, the same in <i>KUKA LBR IIWA</i>	4
2.4	An easy way to understand FK and IK	5
4.1	Robot control via ROS and Sunrise OS	9

List of Tables

List of Abbreviations

LBR	Leicht Bau Roboter
IIWA	Intelligent Industrial Work Assistant
TCP	Tool Center Point
EE	End Effector
DOF	Degree Of Freedom
ROS	Robot Operating System
HMI	Human Machine Interface

Chapter 1

Introduction

1.1 Motivations

1.2 Goals

1.3 Thesis Overview

Chapter 2

Theory overview

This chapter consists in describing the basics notion about the robot, to understand better the entire work made on *KUKA LBR IIWA*. Moreover a description about the kinematic chains, the types of kinematic and about the most used notation is provided. Also an overview of different robot controls is provided.

2.1 About the robot

The *KUKA LBR IIWA* is a robot with collaborative features. It has a 14 kg payload and it has an action range from 800mm to 820mm. It is one of the latest robotic innovation, with the ability to work with humans. It has built in sensors and soft edges that make it safe for human collaboration and the ability to detect movement and touch all over. Moreover it has a media flange, with an internal wiring that is helpful to attach a lot of tools. When a tool is attached it's considered in the kinematic chain. In our case the tool selected is a *Gimatic MPLM3240* that is a electric parallel gripper with 2 self-centering two jaw. It has a total gripping force of 210N.



FIGURE 2.1: The Kuka LBR IIWA

2.2 Kinematic chains

Kinematics is the science of motion that allows to represent positions and velocities of specified points of a complex structure composed by multiple sections, without mention force and torques that cause the motion. To talk about kinematic the concept of *kinematic chains* must be introduced.

A kinematic chain represents the mechanical structure of a robot manipulator. It is a set of ideal rigid links connected by ideal rigid joints and it is considered as a geometric entity, therefore masses and frictions of the kinematic chain are not considered. At the base of the chain there's the fixed frame, usually a table or another rigid support. At the

end there's the TCP that's a point, generally located in the middle of the end-effector, that a manipulator moves to a specific position or along a specified path. In figure 2.2 an example of a simple kinematic chain.

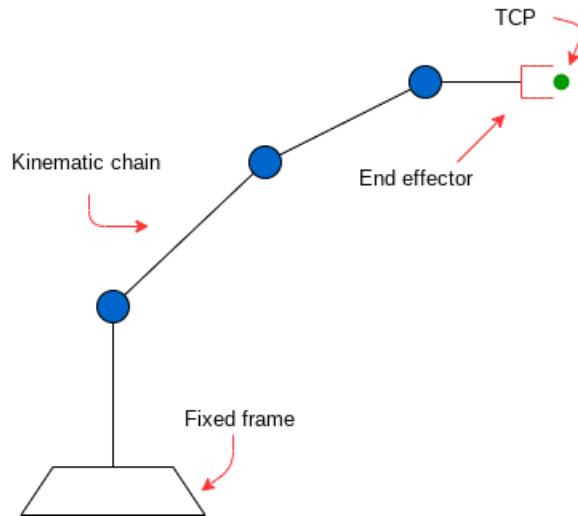


FIGURE 2.2: An example of kinematic chain.

While the end-effector is a rigid body that moves in three dimensional space, the TCP has a position and an orientation based on the fixed frame at the base of the robot. When the TCP is defined as combination of position and orientation, it's said to be a *pose*.

Joints and links are components of the kinematic chain that allows re motion between each other. In robotics there are different types of joints:

- *Revolute* and *prismatic* joints, with a single DOF. The first one allows rotation about a single axis, while the second one provides a traslation along a specific axis. In figure 2.3 a revolute joint.
- *Hooke's* and *sphere* joints. A coupling connecting rigid rods whose axes are inclined to each other.

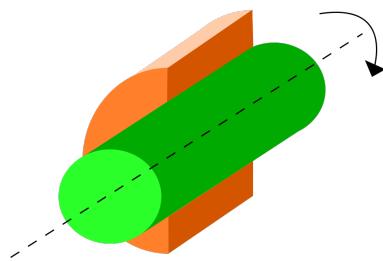


FIGURE 2.3: A revolute joint, the same in *KUKA LBR IIWA*

In our robot there are only revolute joints, in particular the IIWA has seven which correspond to the robot's DOFs. If more DOFs than task variables are available, the manipulator is referred to as a redundant robot, therefore our robot is a redundant robot.

2.2.1 Types of Kinematic Chains

We talk about two different types of kinematic chains:

- *Open kinematic chains*: it's a chain in which every joint connects only two links. Therefore there's only one sequence of links connecting the end of the chain. We can refer to a human arm, so the reachable task space is big but not accurated.
- *Close kinematic chains*: is's a chain where there are more than one link between two joints. It has a cycle structure but it can provide an high accuracy but in a small workspace, so the space reachable by the end-effector is limited.

2.3 Kinematic

From Siciliano et al., 2010, there are two different types of kinematic that describe the behavior of the robot. In the next sections a little introduction about them is provided. Moreover a description on the commonly used notation for selecting references frame is provided in section 2.3.3.

2.3.1 Forward Kinematic

Forward kinematics describes the pose of the EE respect to the world frame, depending on the robot's geometry and based on the angles (revolute) or offset (prismatic) of the robot's joints. The EE pose can be defined by specifying the manipulator's geometry and joint displacements necessary to achieve the pose. When the pose is defined in this way, the TCP is described in the joint space: every joint has a value of offset or angles. A joint motion in joint space produces a motion of the EE in the task space.

2.3.2 Inverse Kinematic

When we talk about inverse kinematics we mean the opposite process. Given a TCP pose respect to the world frame, the necessary joint values to reach that pose must be found. If the TCP pose is out of the reachable joint state the inverse kinematic cannot be found.

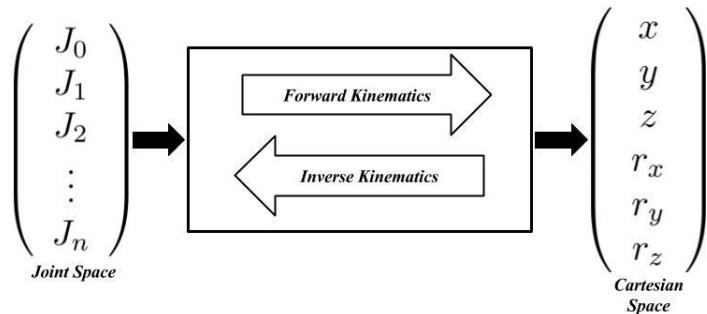


FIGURE 2.4: An easy way to understand FK and IK

2.3.3 Denavit–Hartenberg Convention

As mentioned in the introduction, the *DH* conventions is used for selecting the reference frames. This method allows to attach a reference frame to each link of the robot and it uses certain parameters to describe the position and orientation of each reference frame.

In particular, in this method the relative position and orientation of two consecutive links are defined. Usually the Frame 0 is the frame attached to the robot base and doesn't move.

There are two version of the *DH* convention: we use the standard one. The standard version, once the reference frame have been established provides four parameters that describe the position and the orientation of a single frame:

- a_i : the link length is the distance from Z_i to Z_{i+1} measured along X_i
- α_i : the twist angle is the angle from Z_i to Z_{i+1} measured about X_i
- d_i : the offset lenght is the distance from X_{i-1} to X_i measured along Z_i
- θ_i : the joint angle is the angle between X_{i-1} to X_i measured about Z_i

The first two parameters depend only on the connection of link i with the consecutive one. One of the other two remaining parameters varies depending on the type of joint, in particular:

- if the joint i is a *prismatic joint*, the joint variable is d_i
- if the joint i is a *revolute joint*, the joint variable is θ_i

After that the parameters have been found for every link, it's possible to express the coordinate transformation between link $i - 1$ and link i using translation and rotation matrices.

2.4 Theory about robot control

2.4.1 Postion control

2.4.2 Cartesian impedance control

2.4.3 Joint impedance control

Chapter 3

Literature Review

3.1 Robot learning from Demonstration

3.1.1 Kinesthetic Teaching

3.1.2 Teleoperation

3.2 Assembling and modulable tasks

3.2.1 Assembling tasks

3.2.2 Modulable tasks

Chapter 4

The project

This chapter describes the general setup, its components and a small overview on the tools used for develop the project. Finally the project is explained.

4.1 Setup overview

The *KUKA LBR IIWA* redundant manipulator is programmed using the KUKA's Sunrise Workbench platform and its Java API's. The usage of an open source stack (Christoph et al., 2017) compatible with ROS allows the usage of the robot in a simple way.

A Sunrise project, containing one or more Robotic Application can be synchronized to the robot cabinet and executed from the SmartPad.

The *iiwa stack* provide a Robotic Application that can be used with the robot. It establishes a connection to machines connected via Ethernet to the robot cabinet via ROS. The machine, with ROS installed, will be able to send and receive ROS messages to and from the Robotic Application. The messages used in this stack are taken from the messages available in a standard ROS distribution, but there are other custom ones inside the *iiwa_msgs* folder.

With the stack is simple to manipulate the messages received from the robot and set new ones as command to it, using Python code or all the ROS functionalities already implemented as services, topics, actions.

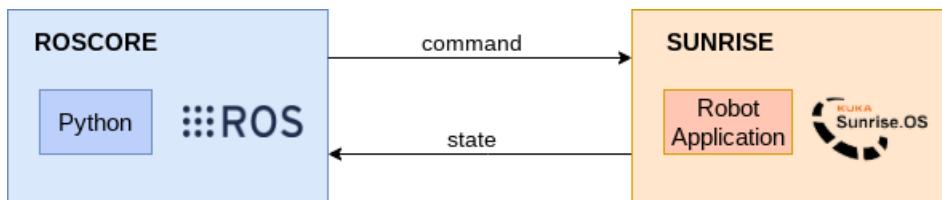


FIGURE 4.1: Robot control via ROS and Sunrise OS

4.1.1 KUKA Robot Controller

The *KUKA LBR IIWA* is controlled via the KUKA Robot Controller, also known as the KUKA Sunrise Cabinet. The KRC is responsible for the transmission control inputs as well as the reading the data of the integrated sensors. In our case, for controlling the Kuka we use the Java application provided by the stack that runs into the SmartPad.

4.1.2 Sunrise.Workbench

The Sunrise.Workbench is a tool used to program robot applications in Java, which are loaded into and are executed on the KRC. It offers the possibility to control the robot

with the strategies described in section [2.4](#). It can also execute the commonly motion patterns as: spline, point-to-point, linear and circular motions.

Since we have a gripper mounted on media flange, there's a task always active in background that provides a method that can be called via `ros_service` to open and close the two jaw. We also have another background task for the `rgb` led present on media flange.

4.2 Project implementation

4.2.1 Theach by demonstration

4.2.2 Teleoperation

Chapter 5

Experiments and Result

5.1 Experiments

5.2 Results

Chapter 6

Conclusion

6.1 Conclusions

6.2 Other possible works

Bibliography

- Christoph, Hennersperger et al. (2017). "Towards MRI-based autonomous robotic US acquisitions: a first feasibility study". In: *IEEE transactions on medical imaging* 36.2, pp. 538–548.
- Siciliano, Bruno et al. (2010). *Robotics: Modelling, Planning and Control*. Springer Publishing Company, Incorporated. ISBN: 1849966346.