UNIVERSITY OF VERONA

MASTER THESIS

Title ...

Author:
Michele PENZO
VR439232

Supervisor:
Dr. Diego DALL'ALBA

ICE Lab
Department of Computer Science

December 29, 2020

UNIVERSITY OF VERONA

Abstract

Department of Computer Science

Master of Science

Title . . .

by Michele PENZO
VR439232

. . .

Acknowledgements

…

Contents

List of Figures

List of Tables

List of Abbreviations

LBR	**L**eicht **B**au **R**oboter
IIWA	**I**ntellingent **I**ndustrial **W**ork **A**ssistant
TCP	**T**ool **C**enter **P**oint
EE	**E**nd **E**ffector
DOF	**D**egree **O**f **F**reedom
ROS	**R**obot **O**perating **S**ystem
HMI	**H**uman **M**achine **I**nterface

Chapter 1

Introduction

1.1 Motivations

1.2 Goals

1.3 Thesis Overview

Chapter 2

State of the art

2.1 Robot learning from Demonstration

2.1.1 Kinesthetic Teaching

2.1.2 Teleoperation

2.2 Assembling and modulable tasks

2.3 User study

Chapter 3

Materials and methods

3.1 Materials

3.2 Methods

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.

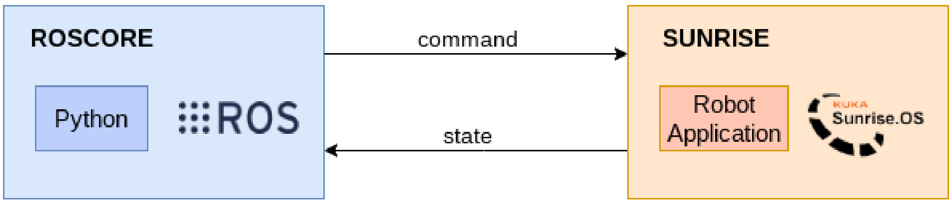

FIGURE 4.1: Robot control via ROS and Sunrise OS

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 providede 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 can also can 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

5.1 Experiments

Chapter 6

Results

oggettivo, tabelle, grafici, boxplot, spiego a chi sta leggendo cosa ne è venuto fuori –> devo aiutare chi sta leggendo i risultati a capire cosa fare

6.1 Results

Chapter 7

Conclusion

7.1 Result discussions

7.2 Conclusions

7.3 Other possible works

References

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.