

UNIVERSITY OF VERONA

MASTER THESIS

---

**Kinesthetic teaching vs. teleoperation:  
a user study about industrial tasks**

---

*Author:*  
Michele PENZO  
VR439232

*Supervisor:*  
Dr. Diego DALL'ALBA

ICE Lab  
Department of Computer Science

March 18, 2021



UNIVERSITY OF VERONA

# *Abstract*

Department of Computer Science

Master of Science

**Kinesthetic teaching vs. teleoperation:  
a user study about industrial tasks**

by Michele PENZO  
VR439232

abstract ...

sommario ...



# *Acknowledgements*

...



# Contents

<b>Abstract</b>	<b>iii</b>
<b>List of Figures</b>	<b>ix</b>
<b>List of Tables</b>	<b>xi</b>
<b>List of Abbreviations</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivations . . . . .	1
1.2 Goals . . . . .	1
1.3 Thesis Overview . . . . .	1
<b>2 State of the art</b>	<b>3</b>
2.1 Robot learning from demonstration . . . . .	3
2.2 Robot in assembly tasks . . . . .	4
2.3 User study . . . . .	4
<b>3 The project</b>	<b>5</b>
3.1 Setup overview . . . . .	5
3.1.1 KUKA Robot Controller . . . . .	5
3.1.2 Sunrise.Workbench . . . . .	6
3.1.3 Safety configuration . . . . .	6
3.2 Project implementation . . . . .	6
3.2.1 Teach by demonstration . . . . .	6
3.2.2 Teach with remote control . . . . .	7
<b>4 Materials and methods</b>	<b>9</b>
4.1 Materials . . . . .	9
4.2 Assembly methods . . . . .	9
<b>5 Experiments</b>	<b>11</b>
5.1 Participants . . . . .	11
5.2 Assembly tasks . . . . .	11
<b>6 Results</b>	<b>13</b>
6.1 Results . . . . .	13
<b>7 Conclusion</b>	<b>15</b>
7.1 Result discussions . . . . .	15
7.2 Conclusions . . . . .	15
7.3 Future works . . . . .	15
<b>References</b>	<b>17</b>



# List of Figures

3.1	Robot control via ROS using <i>iiwa_stack</i> and Sunrise OS . . . . .	5
3.2	Media flange: ① led strip, ② enabling switch, ③ application button . . .	7
3.3	The scheme of ps4 pad: the model used for teleoperation . . . . .	8



# List of Tables



# List of Abbreviations

<b>LBR</b>	<b>Leicht Bau Roboter</b>
<b>IIWA</b>	<b>Intelligent Industrial Work Assistant</b>
<b>TCP</b>	<b>Tool Center Point</b>
<b>EE</b>	<b>End Effector</b>
<b>ROS</b>	<b>Robot Operating System</b>
<b>HMI</b>	<b>Human Machine Interface</b>



## Chapter 1

# Introduction

The entire work is available on Github at: [github.com/michelepenzo/master-thesis](https://github.com/michelepenzo/master-thesis).

### 1.1 Motivations

### 1.2 Goals

### 1.3 Thesis Overview



## Chapter 2

# State of the art

This chapter describes all the papers and articles that are present in literature and that have been treated in this thesis. Starting from robot learning via demonstration and teleoperation to conclude with the state of the art regarding assembly task in industry and presenting the studies that are that have been done in these areas.

## 2.1 Robot learning from demonstration

The main goal of robot learning is to create a way for program robot in simple way that are suitable to be used by everyday people. Two interactions methods are compared: *kinesthetic teaching* and *teleoperation*. In the former, the user physically guides the robot and in the latter the use controls the robot with a pad.

A similar work as mine is [2], where they use kinesthetic teaching and teleoperation using a haptic device and they compare the two ways of interaction. They find that kinesthetic teaching is faster in terms of giving a single demonstration and the demonstrations are more successful.

Along the lines of the previous one, [4] proposes various approaches for gaining knowledge from human demonstrations to perform assembly tasks in a industrial robotic cell. In this work kinesthetic teaching and teleoperation using wireless joystick are compared for create point to point movements. Unlike the previous work an experiment is done: three tasks with different aspects have been done by some users.

In mine work a teleoperation mode has been developed. In [7] they compare a number of teleoperations mode, exploring both the number of dimensions of the control input as well as the most intuitive control spaces. This work propose four methodologies to find a way to move the robot in teleoperation using a wireless pad. The modalities are based on mapping joints as full joint and reduced joint or based on task space as using full task space or reducing task space. In their case, since their use case was concrete spraying, the best way to implement teleoperation was to reduce task space, but from the experiment the best way in other cases was to implement full task space as described in 3.2.2.

In [6], a framework for robot learning by multiple human demonstrations is introduced. Through the demonstrations, the robot learns the sequence of actions for an assembly task without the need of pre-programming. Additionally, the robot learns every path as needed for object manipulation. Moreover the proposed framework copes with changes in the position and orientation of the objects to be manipulated and also provides obstacle avoidance.

## 2.2 Robot in assembly tasks

In the early 2000's, robot used to perform assembly tasks were still too few, especially the robot used to perform assembly tasks. In the age of Industry 4.0, the need to convert factories in *smart* factories introduced the necessity to have modular platforms that can be changed over the time. In this case, manipulator robot as KUKA LBR IIWA that can be re-programmed are very useful. In my case, after learning from demonstration, I focused on how *collaborative robot* can be used within smart factories.

As described in [9] ...

## 2.3 User study

## Chapter 3

# 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.

### 3.1 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 compatible with ROS (presented in [8]) 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 send new ones as command to it, using Python script or ROS functionalities already implemented as services, topics, actions.

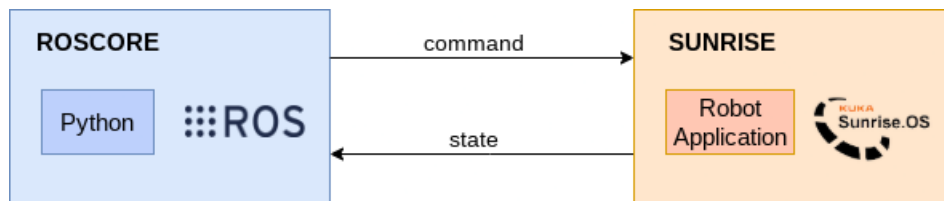


FIGURE 3.1: Robot control via ROS using *iiwa\_stack* and Sunrise OS

For further information about *iiwa\_stack* see [1] and the related work [3]. Instead for information about Sunrise OS and Workbench see [5].

#### 3.1.1 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.

### 3.1.2 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 following strategies: position control, velocity control, joint and cartesian impedance control. 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.

### 3.1.3 Safety configuration

A correct thing to do before start the using of the robot is to check and modify the safety configuration loaded by default. In my case:

- I had to set a new value for every joint in robot configuration. I restrict the value about 2 degrees.
- moreover I added a protected workspace. In this case the robot never goes inside the constraint area.

## 3.2 Project implementation

As described in section 1.2, the goal of the thesis was to ...

Starting from the goal, the work was divide into two phases:

1. create the mode relative to teach by demonstration,
2. create a way to tele-operate the robot in a simple way.

As described in the next sections, in every phase a way to save waypoints and an action on the gripper was implemented. After that the action was captured by the script and was saved in an `.csv` file. With a dedicated program and that file, all the actions saved into the file can be replicated by the robot. Then the description of the two developed modalities.

### 3.2.1 Teach by demonstration

As described in section ??, teach by demonstration or also called *kinesthetic teaching* is a way to move the robot in gravity compensation mode. Using the *iiwa\_stack*, the gravity compensation mode was implemented using joint impedance control mode where for every joint in robot configuration a stiffness and damping value is setted. Stiffness value must be grater than 0 and it is expressed in  $Nm/rad$ , instead damping value must be between 0 and 1. After changing the control mode to joint impedance, the robot seems falls. A force contrary to the gravity must be carried out to keep it up.

On media flange, the green button (③ in figure 3.2) was dedicated for saving commands. When the button is pressed, according to the duration of the pressure an action is recorded:

- one click for save a waypoint;
- 2 seconds pressure for save action on the gripper;

- 5 seconds pressure to exit from teaching program.

When an action is taken, the led strip ( ① in figure 3.2 ) change color: green for way-points, blue for gripper's actions. For safety, as the robot must be held up with the hands, no closing action are performed on two jays. In the next figure the Media Flange mounted on KUKA LBR IIWA.

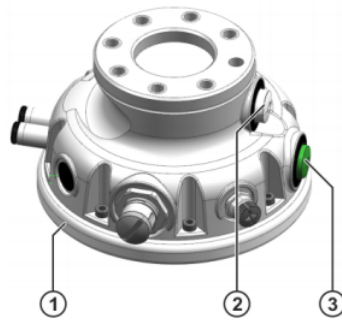


FIGURE 3.2: Media flange: ① led strip, ② enabling switch, ③ application button

### 3.2.2 Teach with remote control

As described in section ??, remote control or *teleoperation* indicates operation of a system or robot at a distance. In our case, teleoperation is intended as control the KUKA LBR IIWA with a **PlayStation 4 pad** and it was developed in two ways:

- *full task space*: in this modality the robot will move respect to the EE. A movement along  $x$ ,  $y$  and  $z$  is performed using linear and angular velocity. With linear velocity you will move the pose of the robot in task space, instead with angular you will move the orientation of the EE.
- *reduced joint space*: in this other modality not all joints are considered. Some joint have been blocked, you can move only eh 1, 2 and 6 joint. It's also possible change the orientation of the EE using angular velocity.

Both of them modalities are performed using L1 or R1 as deadman buttons with right or left analog. This two modalities of teleoperation were take from [7]. In both cases, the buttons used are the same:

- $\times$ : close or open gripper, therefore save action and the actual pose;
- $\triangle$ : save actual pose;
- $\square$ : change to control mode from position to cartesian impedance, and vice versa.

Using the pad, it's possible to use the internal vibration. Therefore, when an external force grater than a preset value is detected the **pad** start vibrating. Based on the controller that the user is using, impedance or position control, the force for activation is different.

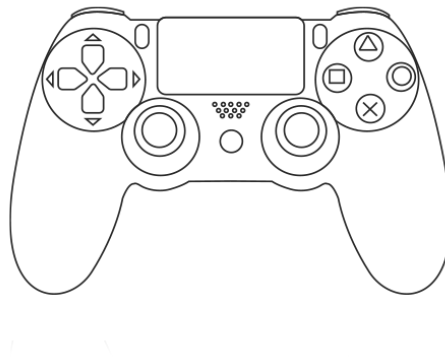


FIGURE 3.3: The scheme of ps4 pad: the model used for teleoperation

## **Chapter 4**

# **Materials and methods**

### **4.1 Materials**

### **4.2 Assembly methods**



## Chapter 5

# Experiments

### 5.1 Participants

### 5.2 Assembly tasks



## **Chapter 6**

# **Results**

### **6.1 Results**



## **Chapter 7**

# **Conclusion**

### **7.1 Result discussions**

### **7.2 Conclusions**

### **7.3 Future works**



# References

- [1] URL: [https://github.com/IFL-CAMP/iiwa\\_stack](https://github.com/IFL-CAMP/iiwa_stack).
- [2] Baris Akgun and Kaushik Subramanian. “Robot Learning from Demonstration: Kines-  
thetic Teaching vs. Teleoperation”. In: (2011). URL: [https://www.cc.gatech.edu/  
~ksubrama/files/HRIFinalBK.pdf](https://www.cc.gatech.edu/~ksubrama/files/HRIFinalBK.pdf).
- [3] Hennersperger Christoph et al. “Towards MRI-based autonomous robotic US acqui-  
sitions: a first feasibility study”. In: *IEEE transactions on medical imaging* 36.2 (2017),  
pp. 538–548.
- [4] Timotej Gaspar, Miha Denisa, and A. Ude. “Knowledge Acquisition Through Hu-  
man Demonstration for Industrial Robotic Assembly”. In: (2019). URL: [http://www.  
reconcell.eu/files/publications/raad\\_19.pdf](http://www.reconcell.eu/files/publications/raad_19.pdf).
- [5] Sebastian Schleisner Hjorth. “Investigation and Implementation of Workspace re-  
strictions for the KUKA LBR iiwa”. 2019.
- [6] Maria Kyrarini et al. “Robot Learning of Industrial Assembly Task via Human Demon-  
strations”. In: *Autonomous Robots* 43 (Jan. 2019). URL: [https://link.springer.com/  
article/10.1007/s10514-018-9725-6](https://link.springer.com/article/10.1007/s10514-018-9725-6).
- [7] Christopher E. Mower et al. “Comparing Alternate Modes of Teleoperation for Con-  
strained Tasks”. In: *2019 IEEE 15th International Conference on Automation Science  
and Engineering (CASE)* (2019). URL: [https://ieeexplore.ieee.org/document/  
8843265](https://ieeexplore.ieee.org/document/8843265).
- [8] Morgan Quigley et al. “ROS: an open-source Robot Operating System”. In: *ICRA  
Workshop on Open Source Software* 3 (Jan. 2009).
- [9] Shirine El Zaatari et al. “Cobot programming for collaborative industrial tasks: An  
overview”. In: *Robotics and Autonomous Systems* 116 (2019), pp. 162–180. URL: [https:  
//doi.org/10.1016/j.robot.2019.03.003](https://doi.org/10.1016/j.robot.2019.03.003).