UNIVERSITY OF VERONA

MASTER THESIS

Towards an optimal method for teaching industrial assembly tasks using collaborative robots: teleoperation vs kinesthetic

Author:
Michele PENZO
VR439232

Supervisor:
Dr. Diego DALL'ALBA

ICE Lab
Department of Computer Science

March 18, 2021

UNIVERSITY OF VERONA

Department of Computer Science

Abstract

Towards an optimal method for teaching industrial assembly tasks using collaborative robots: teleoperation vs kinesthetic

Michele PENZO
VR439232

abstract …

sommario …

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
ROS	**R**obot **O**perating **S**ystem
HMI	**H**uman **M**achine **I**nterface

Chapter 1

Introduction

In this chapter a complete description of the thesis is provided: started from the reasons for which the work was done and concluding with the thesis goals and overview.

The entire work is available on Github at: github.com/michelepenzo/master-thesis.

1.1 Motivations

Within the Industry 4.0, robots are increasingly exploited in production plants. With the ambition to introduce robots into assembly lines the request to be able to quickly reconfigure the workspace requires faster modalities for robot reprogramming. The ease of robot programming is becoming more significant then ever and this process involves a higher degree of automation. For example, welding and painting, are already highly automated in the automotive industry. Instead, demanding assembly tasks which are mainly pick and place or peg into hole tasks are mainly performed manually toady. However, many of these tasks are repetitive and requires high forces and they can be constantly changed. To facilitate reprogramming, the new paradigm which is used more frequently is *Pbd* (Programming By Demonstration) and is often used with assistive and collaborative robots that are installed in industrial environments. This paradigm includes a lot of different approaches described in section 2.1.

1.2 Goals

From *PbD* paradigm described in the previous section, a comparison to find the optimal method for teaching assembly tasks was sought. The main goal of this thesis is find the optimal way for reprogramming assembly tasks. Some questions and hypotheses can be done.

The two proposed approaches, described in 3.2.1 and 3.2.2 are said to be intuitive, but how much when they are used for assembly tasks?

1.3 Thesis Overview

The remainder of this thesis is organized as follows: in chapter 2 the state of the art about robot learning and robot in assembly are introduced, moreover the studies conducted on assembly tasks are presented. In chapter 3 an overview about the robot, the configuration used and how the two modalities for teaching assembly tasks is provided. In chapter 4 an outline of the user study is done. In this chapter the tests that have been made before the experiment, the research on how the performance can be measured and which materials are used for the experiment is done. Subsequently, in chapter 5 how the experiment is conducted, who participated and other details are explained. Instead, in chapter 6 the results are explained in detail. Finally, the thesis is concluded in chapter 7 with conclusion and future works .

Chapter 2

State of the art

This chapter describes the papers and articles 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 have been done in these areas.

2.1 Robot learning

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* 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 [3], where they use kinesthetic and teleoperation using a haptic device and they compare the two ways of interaction. They find that kinesthetic is faster in terms of giving a single demonstration and the demonstrations are more successful.

Along the lines of the previous one, [5] proposes various approaches for gaining knowledge from human demonstrations to perform assembly tasks in a industrial robotic cell. In this work kinesthetic 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 [8] 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 [7], 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 industry

In the age of Industry 4.0, the manufacturing industries increasingly demand more flexible and agile productions systems. The need to convert factories in *smart factories* introduced the necessity to have modular platforms that can be changed over the time. Manipulator robot as KUKA LBR IIWA that can be re-programmed are very useful. They offers a higher level of hardware flexibility, but in order to benefit from this flexibility the demand for new approaches to operating and programming new tasks is inevitable. In my case, after learning from demonstration, I focused on how *collaborative robot* can be used within smart factories and how they can be re-programmed to perform new tasks.

As described in [15], collaborative robots have been increasingly adopted in industries to facilitate human-robot collaboration. In this paper, an overview of collaborative industrial scenarios and programming requirements for cobots to implement effective collaboration are given. The human operator and the cobot share the same workspace to perform manufacturing processes on work pieces. Different definitions of collaborative scenarios and safety measures are given. Always from this paper, a paragraph about learning from demonstration as kinesthetic and teleoperation is described.

The main goal for robots in industry is to combine the advantages of robots, which enjoy high levels of accuracy, speed and repeatability, with the flexibility of human workers. In [13], all these aspects are treated. The use of collaborative robots as KUKA LBR IIWA in industrial processes allows that they can be managed through intuitive systems. One of main challenge is safety, it's fundamental prerequisite in the design of products. Some standards are defined and treated very well this paper.

2.3 User study in assembly tasks

As described in 1.2, the main goal of this thesis is to compare different ways of robot learning and make a study over different typologies of users. Much research in the area of robot learning has focused on pick and place tasks while demanding assembly tasks received less attention so far.

Mine user study, and the work made in [9] focuses in assembly tasks. This paper evaluate the discrepancies between kinesthetic and manual assembly in the context of industrial assembly tasks. In particular they conducted this user study with 78 participants with different qualities. During the experiment they asked to complete four tasks (two peg-in-hole and two DIN rail of different difficulty) multiple times to evaluate the learning obtained during the various repetitions. They proved that when the same task is repeated multiple times the learning increase every time, and on the contrary the duration from the first to last trial decrease definitely. These observation confirm the ease of the learning attributed to kinesthetic.

The work described in [12], presents a human-robot interface based on task level programming and kinesthetic. Is was assessed by nine 9 people of varying robotics experience. The main purpose was to obtain feedback from various users and to assess how well they comprehended and operated the system. The test consisted of two separate tasks: a simple pick and place and a more advanced peg in hole task. The users performed the task individually. The presented system shown that kinesthetic ia an intuitive method for robot programming for non-robotics experts.

The work perfectly described in [14] doesn't treat assembly tasks, but propose a new interaction scheme combining kinesthetic and learning within an integrated system architecture. They evaluate this approach in a user study with 49 industrial workers. The tasks consists in a warm-up phase, movement from left working area to right working area and wire-loop movement. The results shown that the interaction concepts implemented on a KUKA LBR IIWA are easy to handle for novice users and yield significantly improved performance for the teach-in of trajectories in task space.

As described in [11], new technologies in the areas of robotic arms have the potential to accelerate the use of robotics for assembly. Additionally, robotic hands are emerging as a next generation EE technology with advanced force control and manipulation capabilities. On this page a set of performance metrics, test methods and associated artifacts are being developed to progress the application of these technologies. These studies have identified based on manual human performance, tool usage, mechanical resistance but another important aspect of performance measurement is providing confidence in the measured results doing multiple repetitions of a task.s

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 [10]) 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.

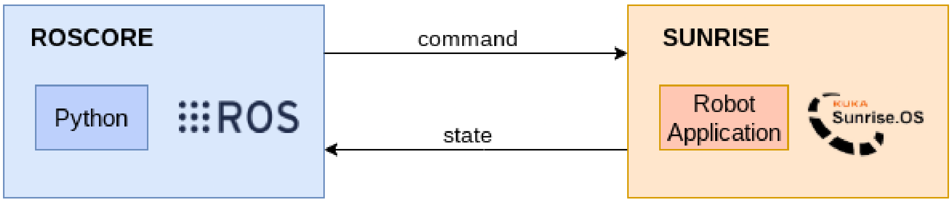


FIGURE 3.1: Robot control via ROS using *iiwa_stack* and Sunrise OS

For further information about *iiwa_stack* see [1] and the related work [4]. Instead for information about Sunrise OS and Workbench see [6].

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.

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.

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:

- for each joint in the robot configuration the values of degrees have been restricted by 2 degrees,
- moreover a protected workspace was added. In this case the robot never goes inside the constraint area.

3.2 Project implementation

As described in section 1.2, the main goal of the thesis is to evaluate, with an user study, different modalities to easily change tasks in industry.

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

1. create the mode relative to teach by demonstration,
2. create a way to teleoperate 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 2.1, teach by demonstration or also called *kinestethic teaching* is a way to move the robot in gravity compensation mode. Using the *iiwa_stack*, the gravity compensation mode was implemented using joint impendance 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 impendance, 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 waypoints, 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.

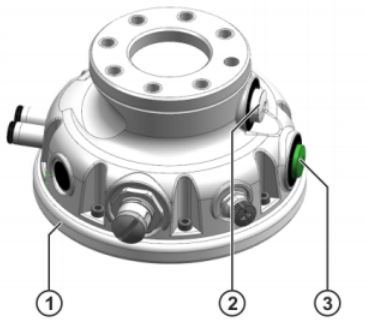


FIGURE 3.2: Media flange: ① led strip, ② enabling switch, ③ application button

3.2.2 Teach with remote control

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 the 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 [8]. 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 control mode that 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

Methods and materials

This chapter describes in the first section how robot was configured, the preliminary analyzes on the robot and how the experiment is conducted. In the second section is described how the experiment was prepared and the materials used for tests.

4.1 About the robot

The KUKA LBR IIWA is a robot with collaborative features. It has a 14 kg payload and an action range from 800mm to 820mm. I's 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 4.1: The KUKA LBR IIWA

4.2 Methods

The first thing that was done was the calibration of the gripper using the application provided by *Kuka* and usable from the *SmartPad*. This application consists of move the sixth and seventh joint with multiple movements and provide a way to calculate the weight, center of mass and inertia matrix of the tool attached on media flange. This operations was made in different configurations (in the most used for this work) for several times and all times gave reasonable values. Finally, the configured tool was loaded as `ros_param` and recognized from the stack.

Using *ROS* and the open source stack available on [1], many `topics` can be used to calculate joint position and velocity, torque on joint, cartesian pose and force on referenced frame. Before the user study some data are collected to understand how the final experiment can be evaluated.

The topic called `CartesianWrench`, will show the external force measured by the robot according to the used reference frame. In our case, the force is measured on the media flange. To understand if we could could consider the force applied on the EE, some experiments were conducted to understand the reliability and responsiveness of the values returned from this topic. This test were conducted using objects of different weight calibrated on a scale with precision to the gram and attached to both of the two jaws on the gripper. Every weighing was performed in three different configuration of the robot and for two times. In table 4.1 the results obtained from the experiment.

	400 gr	**700 gr**	**1000 gr**	**1300 gr**	**1600 gr**
Home pose	4,777	7,566	10,129	13,302	15,926
Home pose	4,702	7,531	10,145	13,332	16,083
On pallet	4,475	7,267	9,541	13,696	16,253
On pallet	4,477	7,245	12,139	13,641	16,208
On buffer	2,349	6,581	9,626	12,254	15,385
On buffer	2,895	6,615	9,718	12,330	16,244
Mean	3,946	7,134	10,216	13,093	16,017
Std deviation	1,046	0,435	0,975	0,640	0,333

TABLE 4.1: Results from experiment to evaluate force on EE

From the test we can understand that the values obtained are reasonable and in line with expectations, even if in the case of robots with elbow in high position (on pallet) the values are underestimated. This test was done in position and impedance control and the results are similar.

Another test that was made using the force applied on EE analyzes the maximum force that can be applied by the robot on a linear surface before that the collision avoidance feature is activated. This test also helped us to understand the reliability of the read values. It was made using position control in figure 4.2 and impedance control in 4.3.

In figure 4.2 we can observe that using position control the curve of the force applied is exponential and it stops about a $14N$ that is the maximum force that the KUKA LBR IIWA can impart. Instead, in figure 4.3 we can immediately spot the difference between impedance and position control. We can see that the maximum force that can be applied before that collision avoidance is activated is no longer $14N$ but $7N$. This is due to impedance control that is an approach to dynamic control relating force and position. A *spring constant* defines the force output for a compression of the spring, instead a *damping constant* defines the force output for a velocity input. Using impedance control we are controlling the force of resistance to external motions that are imposed by the environment. In fact, in figure 4.3 we can see the behavior of the spring: it reacts to the force applied on the surface by adapting (the first part of the curve), but when the force is too much and

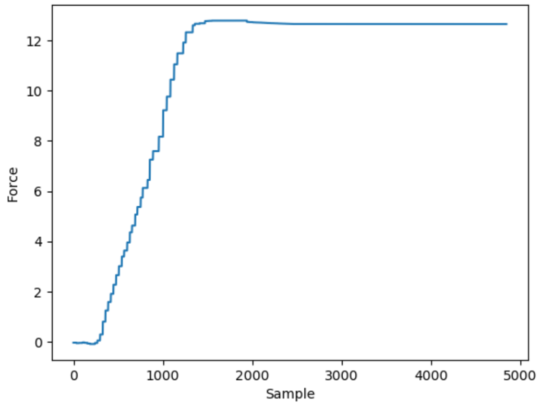


FIGURE 4.2: Force calculated on EE in position control

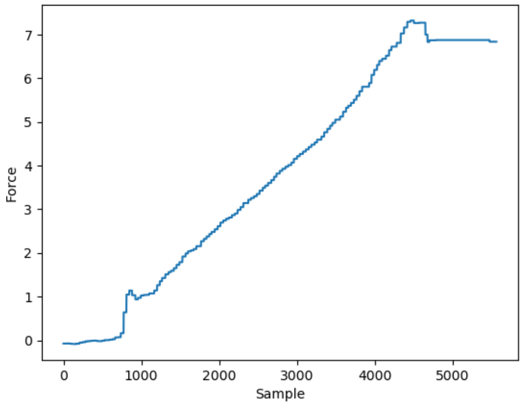


FIGURE 4.3: Force calculated on EE in impedance control

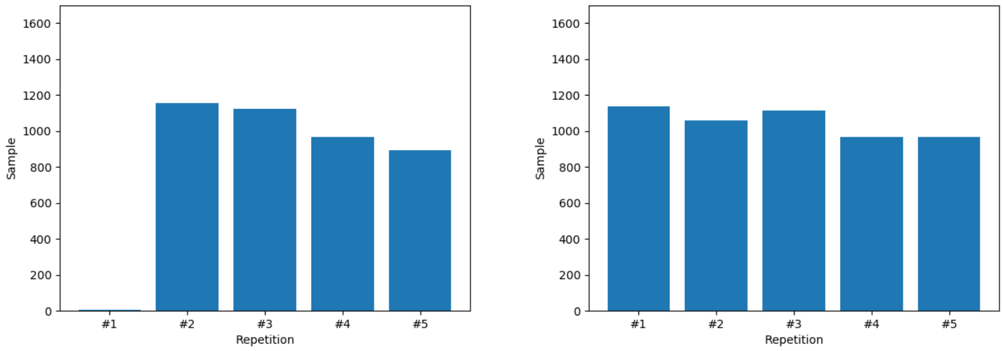
the damping constant too low to perform an adaptive movement it stops and the robot applies the maximum possible force. Since there is the damping constant, the maximum force will therefore be $7N$.

As preliminary action a test of the final experiment discussed in section 5 has been done for test how the experiment can be done by the participants. Especially, during this test all the possible data were collected subscribing to `topic` as:

- `CartesianPose`: position and orientation of EE,
- `CartesianWrench`: force applied on EE,
- `JointPositionVelocity`: position and velocity of all joints,
- `JointExternalTorque` and `JointTorque`: torque applied on joints.

This phase of the test was made by an inexperienced user who had to do the same task for N times. The value of N that has been chosen is 5: a relatively high value that will be reduced in the final experiment. The task consisted to move four Lego blocks from a pick position to a place position. This task was repeated for both the modalities: kinesthetic and teleoperation. The data collected before the final experiment provides a solid basis to be able to perform a preliminary analysis to understand how to perform the final experiment by collecting only the data necessary for the final evaluation.

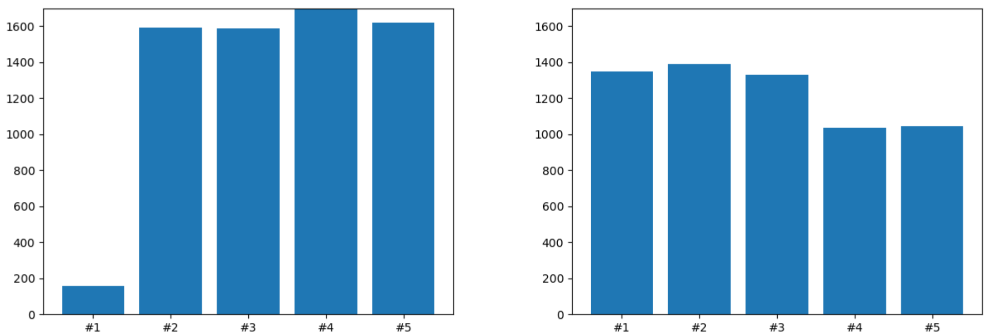
As represented in figure 4.4 we can say that the time necessary to complete the task was similar between the two users, but in the figure 4.4a the first trial wasn't completed due a mistake by the user. In this case wasn't made to repeat the experiment because this was a test and all the possible problems had to be understood. We can also say that with more attempts the time for complete the task is always minor.



(A) User inexperienced (B) User with experience

FIGURE 4.4: Difference between users in kinesthetic

In figure 4.5 we can say that the times for complete the task in teleoperation are high respect to kinesthetic. We can also say that user inexperienced is always slower than user with experience. Even in this case the user inexperienced made a mistake (figure 4.5a) because made a collision with a Lego block but it was included anyway in the results.



(A) User inexperienced (B) User with experience

FIGURE 4.5: Difference between users in teleoperation

From the two failures, it's noticed that the user doesn't know very well the environment and how to use the robot. Before the final experiment an through explanations about kinesthetic and teleoperation is provided to the user.

4.3 Materials

From the data collected by the test experiment described in the previous section we decided to use for our experiment new objects for the final one with the users. At first, some Lego or boxes are used for the tests, but in the final experiment was decided to use some objects that can make the final scenario more similar to reality. *Siemens* describe a robot learning challenge that provides an *Innovation Challenge* which motivated many labs to focus their attention to this area of research. Their question was: “how can we benchmark different learning algorithms and apply them to the challenges of industrial automation?”. Although in this work no learning algorithm was applied, to perform tasks that are applicable in industry this benchmark was download from [2] and printed with 3D printers. After that, some tests were conducted with this printed objects before submitting the final experiment to the sample of users.

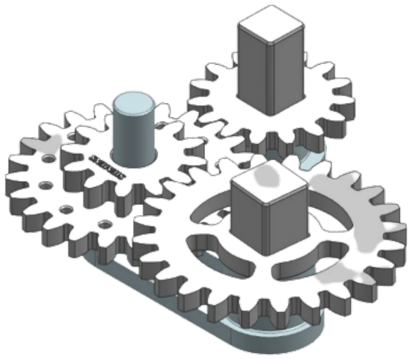


FIGURE 4.6: Siemens challenge: the complete benchmark

Chapter 5

Experiment

In this chapter a complete description of the experiment in detail is provided. Starting from an overview about the participants: who they are and their peculiarity. In section 5.2 the experimental design was divided into phases and described in every step. Finally was described which data were collected.

5.1 Participants

5.2 Experimental design

The experimental design describes the entire flow which is done by every participants of the experiment. It has been divided into 3 stages for convenience: the first is the phase before the experiment, the second the experimental phase and the last one the final phase after the experiment.

Pre-experiment phase

In this phase all the procedures necessary to conduct the experiment correctly. First of all a privacy policy for data collection was signed by the users. After that a teach phase was carried out by all the users in order tu understand how they can be move the robot (kinesthetic and teleoperation) and which buttons they need to press to perform the actions. At this stage users are asked to move a Lego block from a pick position to a place one performing a 90° rotation to teach him how to use rotation feature. This simple pick and place was performed in by the users in both the modalities. If the user complete this phase an unique id is assigned to him. Moreover every user perform the pre-experiment and experiment phase without seeing other users do the same.

Experiment phase

The experiment consists in three subtasks in ascending order of difficulty

Some clarifications are provided to unify all users:

- every time that a trial is repeated the KUKA LBR IIWAwere positioned over the first object that would be manipulated,
- every trial is considered valid, even if a collision by the user or any other mistake that is made by the user compromises the trial,
- if there is an error that does not depend by the user, the trail was repeated

Post-experiment phase

5.3 Data collection

Chapter 6

Results

6.1 Results analysis

6.2 Result discussion

Chapter 7

Conclusion

7.1 Conclusions

7.2 Future works

References

[1] URL: https://github.com/IFL-CAMP/iiwa_stack.

[2] URL: https://new.siemens.com/us/en/company/fairs-events/robot-learning.html.

[3] Baris Akgun and Kaushik Subramanian. "Robot Learning from Demonstration: Kinesthetic Teaching vs. Teleoperation". In: (2011). URL: https://www.cc.gatech.edu/~ksubrama/files/HRIFinalBK.pdf.

[4] Hennersperger Christoph et al. "Towards MRI-based autonomous robotic US acquisitions: a first feasibility study". In: *IEEE transactions on medical imaging* 36.2 (2017), pp. 538–548. URL: https://ieeexplore.ieee.org/document/7637013.

[5] Timotej Gaspar, Miha Denisa, and A. Ude. "Knowledge Acquisition Through Human Demonstration for Industrial Robotic Assembly". In: (2019). URL: http://www.reconcell.eu/files/publications/raad_19.pdf.

[6] Sebastian Schleisner Hjorth. "Investigation and Implementation of Workspace restrictions for the KUKA LBR iiwa". 2019. URL: https://projekter.aau.dk/projekter/files/306506063/Investigation_and_Implementation_of_Workspace_restrictions_for_the_KUKA_LBR_iiwa.pdf.

[7] Maria Kyrarini et al. "Robot Learning of Industrial Assembly Task via Human Demonstrations". In: *Autonomous Robots* 43 (Jan. 2019). URL: https://link.springer.com/article/10.1007/s10514-018-9725-6.

[8] Christopher E. Mower et al. "Comparing Alternate Modes of Teleoperation for Constrained Tasks". In: *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)* (2019). URL: https://ieeexplore.ieee.org/document/8843265.

[9] rne Muxfeldt, Jan-Henrik Kluth, and Daniel Kubus. "Kinesthetic Teaching in Assembly Operations - A User Study". In: (2014). URL: https://www.researchgate.net/publication/270475204_Kinesthetic_Teaching_in_Assembly_Operations_-_A_User_Study.

[10] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *ICRA Workshop on Open Source Software* 3 (Jan. 2009). URL: http://www.willowgarage.com/sites/default/files/icraoss09-ROS.pdf.

[11] *Robotic Grasping and Manipulation for Assembly*. URL: https://www.nist.gov/el/intelligent-systems-division-73500/robotic-grasping-and-manipulation-assembly.

[12] Casper Schou et al. "Human-robot interface for instructing industrial tasks using kinesthetic teaching". In: *2013 44th International Symposium on Robotics, ISR 2013* (Oct. 2013), pp. 1–6. URL: https://ieeexplore.ieee.org/document/6695599.

[13] Valeria Villani et al. "Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications". In: *Mechatronics* 55 (2018), pp. 248–266. URL: https://doi.org/10.1016/j.mechatronics.2018.02.009.

[14] Sebastian Wrede et al. "A User Study on Kinesthetic Teaching of Redundant Robots in Task and Configuration Space". In: *Journal of Human-Robot Interaction* 2 (Mar. 2013), pp. 56–81. URL: https://dl.acm.org/doi/pdf/10.5898/JHRI.2.1.Wrede.

[15] 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.