Tampere University

Amezua Hormaza, Leire

A VIRTUAL REALITY ENVIRONMENT FOR TRAINING OPERATORS FOR ASSEMBLY TASKS INVOLVING HUMAN-COBOT INTERACTIONS

Faculty of Engineering and Natural Sciences
Master Thesis
May 2019

ABSTRACT

Amezua Hormaza, Leire: A virtual reality environment for training operators for assembly tasks involving human-cobot interactions
Master Thesis
Tampere University
Automation Engineering
May 2019

The introduction of collaborative robots in the industry requires new training methods. Users without experience in collaborative tasks with robots present insecurity at the beginning, reducing their productivity until they become familiar with this type of process. To address this problem the training method must take place in an environment where the user feels comfortable working with the cobot to overcome the insecurities. This thesis aims at defining a training method for users to get used to working with collaborative robots. The method covers every kind of robot and task. In addition, this research looks for a training that takes place outside the production line so as not to affect the productivity of the plant.

The document presents the background of training methods in the industry and new trends in this field such as virtual reality and the evolution of interactions with robots. A patent landscape is included to evaluate the current situation in the investigation and development of these fields.

This thesis work proposes an interactive and immersive virtual reality training based on WebGL. It consists on a simulation where the operator interacts in real time with a cobot executing a collaborative task. By using WebGL you can access the simulation directly from the browser and without restrictions in the virtual reality equipment. The scenario presents the assembly of a box in collaboration with the YuMi cobot of ABB. The tools, models and techniques used for the implementation are described. Taking advantage of the properties of virtual reality to facilitate the learning of the task, the simulation offers a user assistance system that is explained in detail.

This method has been tested in a group of student engineers who performed the simulation in order to evaluate the effectiveness of this proposal to help operators in their learning of collaborative tasks. The results show a greater acceptance and confidence of the users to perform the task with the cobot after the simulation while they learnt the entire process of the task.

It is concluded therefore with this thesis that the proposed method is valid for user training in collaborative tasks. It is hoped that this work will serve as a basis for future research in the incorporation of WebGL and virtual reality in the training of industrial processes.

Keywords: Virtual reality, operator training, WebGL, industrial robotics, cobots.

The originality of this thesis has been checked using the Turnitin OriginalityCheck service.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AGV	Automated Guided Vehicle
API	Application Programming Interface
AR	Augmented reality
CPS	Cyber-Physical Systems
DOM	Document Object Model
glTF	GL Transmission Format
GPU	Graphics Processing Unit
HTTPS	Hypertext Transfer Protocol Secure
ICT	Information and Computer Technology
IDL	Interface Description Language
IoT	Internet of Things
IPC	Inter-Process Communication
HMD	Head Mounted Display
HRI	Human-Robot Interaction
MR	Mixed Reality
OpenGL	Open Graphics Library
OSMesa	Off Screen Mesa
OTJ	On-The-Job
PC	Personal Computer
RE	Real Environment
SF	Sensory Feedback
TI	Total immersion
UGV	Unmanned Ground Vehicle
VR	Virtual Reality
VRTS	Virtual Reality Training System
VW	Virtual World

.

1. INTRODUCTION

1.1 Motivation

The fourth industrial revolution (i.e., Industry 4.0) drives the industrial equipment manufacturers to include the advances from the Information and Computer Technology (ICT) field. These technologies and concepts have been enriching the final products that can be used at the factory shop floor [1]. These advances focus on the automation of industrial processes. Two trends are highlighted more and more requested for their implementation in the industry: the set of Mixed Reality, Augmented Reality and Virtual Reality; and the Human-Robot Interactions (HRI).

As both the devices and the applications offered by virtual reality are developed, it appears to be an increasingly effective tool in the automation process since it facilitates activities that until now were carried out manually in the plant, for example, the monitoring of real-time industrial plant tasks. The advancement of hardware technology has reached a point where it is now feasible to produce hardware for Virtual Reality (VR) that are actually highly usable, and most importantly, affordable for many consumers [2]. The high accessibility of the virtual reality and the great variety of devices that have been developed with diverse capacities and economic ranges allow an adequate integration of the same in diverse work environments, either in the line of industrial operation, in training or in tasks of management.

The study and development of the second trend, HRI, arises from the need to automate increasingly complex processes that require the technical capabilities of a robot and the criteria of a human. As these tasks become more precise and detailed, the necessity arises for a greater integration of the qualities of the human and the robot. Collaborative robots (Cobots) emerge from this need. Cobots are intended to physically interact with humans in a shared workspace [3], which requires a direct interaction between the robot and the user with no barriers. New lines of research are being developed with this possibility of introducing automation with cobots.

1.2 Justification

Since the two exposed tendencies are still in process of evolution, the study carried out in this thesis work seeks to make a contribution to this line of research looking for possible applications for the industry in which the qualities of virtual reality can be incorporated to the automation with cobots, exploiting the benefits that each of them brings. For this study to add value to the automation process, the applications proposed in this work should cover and reinforce the weaknesses of the current processes used in this field.

1.3 Problem statement

Although the cobots incorporate a high level security system there is distrust on the part of the users towards working with robots with such closeness. This distrust hinders the correct execution of the tasks by the users until they get confidence with the cobots, which translates into a long period of adaptation to work on collaborative tasks. As a consequence, productivity is reduced since the period required to train the user is greater and, therefore, it requires more time for the user to execute the task with the maximum performance. This problem generates the need to introduce new training methods for operators who are not familiar with collaborative task processes. Training can facilitate this process by reducing initial biases, providing knowledge about system capability, and applying a risk assessment based on the behaviour of the automation [4]. These training methods must collect the necessary requirements to be able to answer the following question: “How to train operators to work with collaborative robots without decreasing productivity during the period of adaptation?”.

1.4 Scope and objectives

As abovementioned, the scope of this thesis work is collaborative tasks. The collaborative tasks imply that user and robot work at the same time, in the same workspace and in the same process, usually in manipulation tasks.

The proposal presented in this document is an off-the-job training, thus allowing the operator to learn the task outside the production time and, therefore, avoiding a reduction in the productivity of the plant.

In addition to learning the task, the user has the possibility to become familiar with the cobot outside the production line, so that the period of adaptation required by those operators without experience with cobots is done outside the production time. As a consequence of this, the productivity of the line is not affected by this adaptation period.

With this system the operator has maximum productivity from the moment he/she enters the production line because he/she is already familiar with the task to be carried out in collaboration with the cobot.

This document seeks to incorporate the qualities that virtual reality contributes with the objective of improving the training process.

1.5 Outline

This documents structures as follows: state of the art, design and implementation, tests and results, and conclusions.

The state of the art presents the background of the aspects required for the realization of this thesis and that have been presented throughout the introduction. It introduces the basic and necessary knowledge about each of these aspects to be able to follow up on the following chapters. This chapter includes a patent landscape that allows visualizing the current research trends in these fields.

The chapter of design and implementation presents the research path followed until the application that is proposed in this work. After the research period reflected in the state of the art, a selection of the alternatives that best meet the requirements of the approach is made. Once the parameters of the proposal are defined, an investigation is made on the existing applications in search of proposals that can cover all or part of the requirements. From this research a new solution to the problem exposed in the introduction is proposed. Finally, the implementation of said proposal is carried out.

The implemented application is tested through a series of tests whose results are exposed in the chapter of tests and results. From these results it is concluded the effectiveness of the application and the possible defects that it may have are detected so that it is possible to establish the improvements to be incorporated.

Finally, in the chapter of conclusions it is determined if the application covers the needs exposed in the problem statement and introduces the possible research paths to which this proposal can contribute in future works.

2. STATE OF THE ART

This chapter is a theoretical background in which the concepts used to this thesis are defined and described. The content of this chapter is focused on manufacturing systems, since this is the environment in which the application of the thesis implementation is planned to be applied.

Although all the concepts to be discussed in this section are related to each other, they will be treated separately in 5 subsections. The first section provides a definition for mixed, augmented and virtual realities, while explaining the main characteristics with the help of industrial examples. The second section shows a comparison between the contributions of human and robots to industrial processes. This section introduces the concept of "Cobot". The third section the most common methods used for training operators on industrial tasks. The fourth section summarizes the concepts reviewed during the previous sections using tables, with the purpose of facilitating the selection of the correct methods and devices that will be used in the implementation of the thesis. The last section illustrates the research trends on the fields treated in this chapter through a patent landscape.

2.1 Mixed Reality

Nowadays, the global competition in industry generates the need for fast adaptation of production to the changing demands of the market. To meet this requirement a great advance is needed in current manufacturing technology. The fourth industrial revolution (i.e. Industry 4.0) is an approach based on integration of the business and manufacturing processes which involves the application of the generic concepts of Cyber-Physical Systems (CPS) and Internet of Things (IoT) to the industrial production systems [5].

Mixed Reality (MR) has a growing role in industry 4.0. A MR environment is a combination of real and virtual worlds presented together within a single display [6], which makes it highly useful for industry especially in those areas where the pre-visualization of the operation plays a critical role in the prevention of accidents. Within the industrial scope, MR has diverse applications.

Its ability to simulate the system operation even in extreme situations makes it suitable for detecting problems in advance and finding out the solution. In terms of optimizing the processes, MR allows the worker to perform an analysis and monitoring in real time. Traditionally, the detection of errors in the product was done in the post-process; however, nowadays technologies like MR allow human-robot collaboration so that it is possible to detect some errors in the product and problems in the machine immediately during the processing. The result of these applications translates into a reduction in process costs due to the rapid detection of errors and, therefore, the rapid intervention, among others.

Spacedesign is an example of a MR application that has already been implemented in industry for aesthetic design of free form curves and surfaces. It is a workbench-like 3-D display for free hand sketching, surfacing and engineering visualization. Thanks to stereo glasses that augment the process the user adds shapes, textures and annotation [7]. Figure 1 illustrates the *Spacedesign* application.

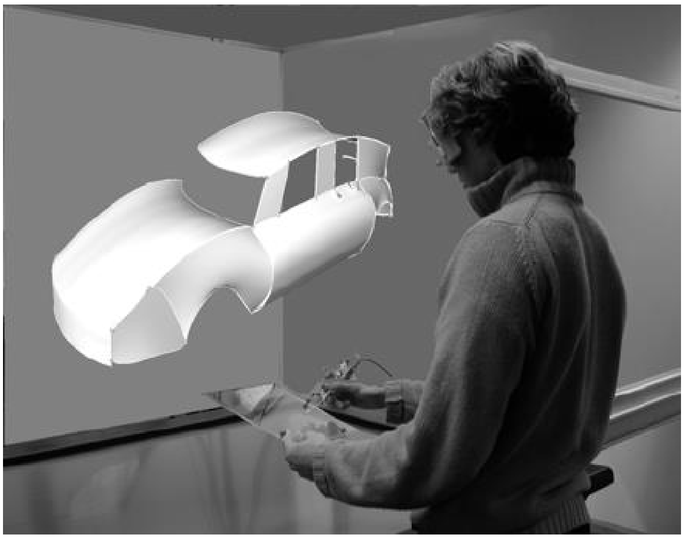


Figure 1. *A car body is realized in Spacedesign using 3D devices and VR/AR visualization [7]*

As already defined above, Mixed Reality is a combination of real and virtual worlds. Figure 2 shows the possible combinations between the real world, the virtual world and human interactions.

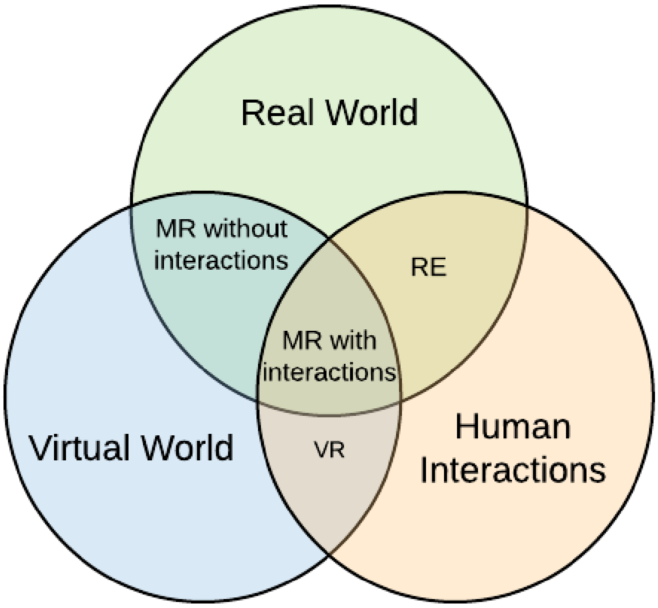


Figure 2. *Venn diagram of Mixed Reality*

The Real Environment (RE) is a totally real world, so the human can interact with it. The Virtual Reality (VR) is a totally 3D computer generated world human can interact with it with the help of hand controllers. Finally, the Mixed Reality (MR) introduces virtual objects in the real world. The human can interact with them or just take the virtual information visually. A special case of MR is Augmented Reality (AR) which contains virtual images overlaid on the view of the user. In the following, the section focuses in AR and VR.

2.1.1 Augmented Reality

An Augmented Reality (AR) system adds computer generated objects on top of the user’s view that enhance the real world. This technology combines the real and virtual objects in a real environment which are aligned with each other. It runs interactively and in real time [8].

The displays that introduce the virtual images on top of the user’s view can be divided into three categories. Firstly, the head-worn display presents augmented information in front of the user’s eyes. The most common way to find this device is in the form of glasses. The second one is the handheld display which needs to have a camera incorporated. The user focuses with this camera on the scenario to be augmented. This scenario is shown with an AR overlay on the screen of the device. Finally, the projective display projects the virtual information directly on the physical objects to be augmented [8].

One example of application that uses projective displays is *smARt.Assembly* which provides a projection based assembly assistance system in industrial processes. This system projects digital guidance information in terms of collecting information and assembly data in a user's physical work area without the use of smart glasses that might limit the field of view of the user [9]. Figure 3 shows a prototype of this application.

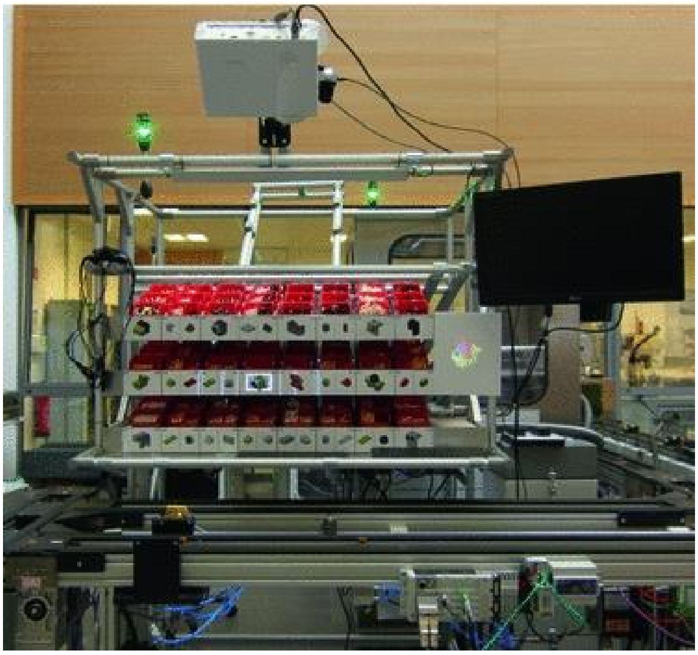


Figure 3. *Prototype of projection-based AR assembly station, with the projector attached above the users' head [9]*

Within the applications that require the use of head-worn or handheld displays, according to [10], there are two types of augmented reality: marker based which uses identifiers or markers, and markerless which uses positional data.

Marker-based Augmented Reality. This kind of AR requires a camera and the use of visual markers. These markers are distinctive pictures (e.g. QR code) that are recognizable by the user. Most are black and white due to the good contrast between these two colours that facilitates the recognition of the drawn pattern. The virtual world is not displayed above the real one unless the user has located the marker through the camera and processed it, as it is depicted in Figure 4.

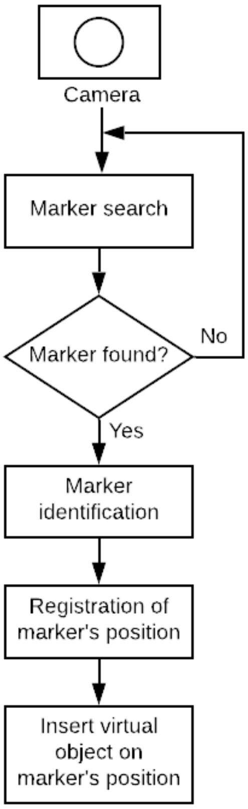


Figure 4. *Flowchart of Marker-based Augmented Reality system*

One application of this type of AR is automotive maintenance. Figure 6.a. illustrates its implementation for the engine maintenance of a real Mercedes-Benz [11]. In this example the markers were attached to an object of the engine. Multiple markers were used to present a view of maintenance and repair instructions in several ways such as textual and video formats, 3D animated models, and a video/audio link to a technician as an example of remote technical assistance.

Markerless Augmented Reality. It uses location data to adapt the virtual content to the user's position. The virtual image is gathered through internet (i.e. using GPS) and displayed on any specific location. The process followed in this type of augmented reality is illustrated in Figure 5.

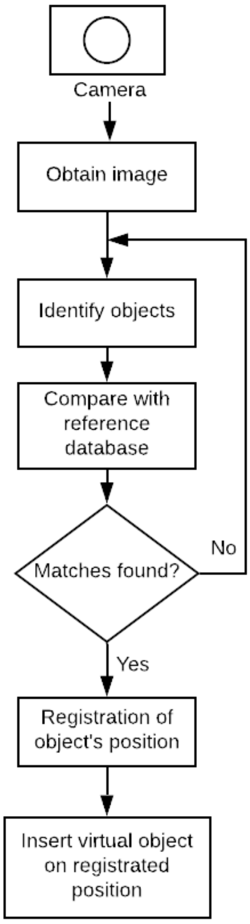


Figure 5. *Flowchart of Markerless Augmented Relality system*

Figure 6.b. shows the application of this technology for assistance of an operator during maintenance operations in an engine of a Renault Clio [12]. Virtual elements are added to guide the operator during the procedure by pointing out virtually different parts of the engine. The addition of these elements implies a robust localization process.

Figure 6. *Maintenance assistance through a) marker based AR [11] and b) markerless AR [12]*

Enhancing the real world does not imply a tangible interaction of the user with the virtual objects, as demonstrated in the previous example (Figure 6), in which the interaction between the user and the virtual world was merely visual. But nevertheless, these interactions provide the user with a more natural and intuitive experience [13]. There are two trends in AR interaction research. The first one uses heterogeneous devices to take advantage of different displays (e.g. projective or handheld displays) so that the user can manipulate and interact with the virtual data. The second integrates the virtual data with the physical world through tangible interfaces in such a way that the user modifies the virtual model by manipulating real objects [8].

2.1.2 Virtual Reality

Virtual Reality (VR) is a simulation of a non-real environment that can be interacted by the user using sensory devices, while this experience can be realistic or fantastic. The study of virtual reality extends to a great variety of fields of diverse disciplines, including the industrial field.

2.1.2.1 Keywords

Below are introduced the key concepts for a better comprehension of VR in the field of interest of this research: virtual world, immersion, total immersion, interactivity and sensory feedback.

Virtual world (VW) is a three-dimensional computer generated environment that can be joined by several users interacting with each other and with the environment.

Immersion is the perception of being present in a non-real world. The grade of immersion experienced by the user determines the nature of the interactions with the virtual environment. There are three levels. In the first level are the non-immersive simulations that the user accesses through a standard screen. The second is the semi-immersive simulation. The user enters in the virtual environment through a surround screen that covers and stimulates both frontal and peripheral vision. In the third level are the fully-immersive simulations which require the use of a head-mounted display and motion detection devices such as joysticks, controllers, data gloves or body suits.

Total Immersion (TI) is the extreme case of fully-immersive simulation. The user loses awareness of being in a virtual environment because the experience feels so real that the user forgets to be in an artificial environment and acts as being in the real world. This grade of immersion can be achieved as long as the environment and its interaction with the user are completely based on the real world.

Interactivity is a critical concept when it comes to engaging the user with the virtual environment. If this environment does not respond to the user’s action in a natural manner the human brain loses the sense of immersion. The most habitual way of interacting with the virtual environment is through hand trackers that recognize the position and orientation of user's hands and use them to make changes in the virtual environment.

Sensory feedback (SF): in order to provide the user with an optimal level of immersion the experience must stimulate vision, hearing and touch senses, which requires sensory feedback. SF is achieved through input-output devices include position trackers, sensing gloves and visual, audio and haptic feedback [14]. Normally, the VR simulations start with the sampling of the user’s head and hand positions which are the initial data for the position trackers to measure the translation and orientation of both. The sensing gloves are used in those simulations that require the measurement of finger positions or haptic feedback. However this is not the most used technology nowadays due to the difficulties when adapting the glove to the hand shape and the higher price. There are several devices for getting visual feedback depending on the level of immersion. Head Mounted Displays (HMD) offer a fully immersive experience as they cover a larger visual field and incorporate audio feedback which provides realism to the simulation.

2.1.2.2 VR System

According to [15], VR systems are characterized by four main technologies: head tracking, hand and gesture tracking, 3D computer graphics and wide-angle stereoscopic displays. All these technologies are depicted in Figure 7.

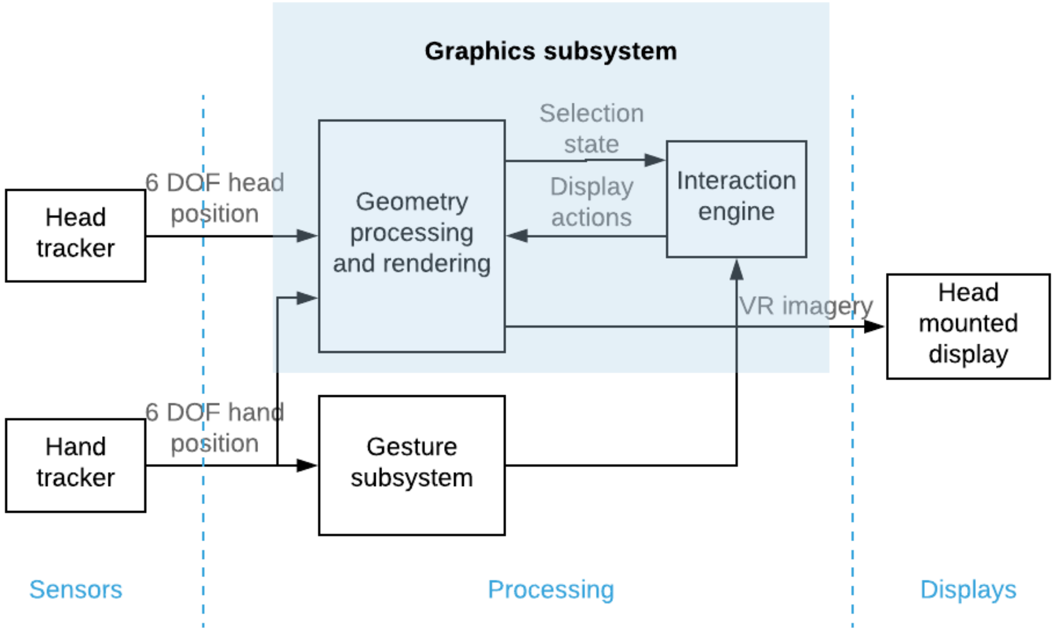


Figure 7. *VR System diagram [16]*

The first technology is the head-tracking which defines the location of the user's head in order to perform the interactions between the user and the virtual environment. Both the user's head and the virtual objects have defined the position and orientation with respect to a global coordinate system. As the virtual environment covers more than the user's field of view, only by moving the orientation of the head can the user access to the entire environment.

The second of the four technologies, hand and gesture tracking is not mandatory for performing a VR simulation, unlike the rest. The use of this technology depends on the level of interaction required by the experience. Hand tracking allows a VR system to determine the position and orientation of the user's hand for enabling interaction with the environment [17]. As the VR environment is the 3D space, it is needed 6 degrees of freedom in order to define the position and orientation of an element. This necessity is covered by the controllers.

The graphics subsystem is connected to head and hand trackers and to the Head Mounted Display It receives the data from the trackers and processes it. Based on this input information, the subsystem generates a series of interactions that are inserted into the HMD. The grade of immersion is favoured by two factors contained in the 3D computer graphics: realistic graphics and real-time interaction with the environment. Real-time interactions are achieved with an appropriate update rate. This rate conditions the quality and complexity of the design of the virtual world, being the most advisable to increase the rate of image updates to greater than 30 times per second.

Wide-angle stereoscopic displays, is responsible for forming images on the visual display (e.g. HMD) in the same way than the human brain does. The vision of a person is the result of combining the information received by both eyes. This is called stereoscopic, or binocular, vision. With each eye independently a range of 150º-160º of vision is obtained. Combining the two eyes provides a single sharper image that widens the field of vision, reaching up to 180º. This model is used to introduce images in the lens of the head-mounted display so that the user gets the maximum possible information from the environment. According to Kalawsky, the minimum features of a VR display system are greater than 110º for horizontal field of view, greater than 60º for vertical field of view, and greater than 30º of stereo overlap [15].

2.1.2.3 Applications

Virtual reality is a proper tool to learn how to perform processes without needing access to the real work space, being able to train workers without damaging the real equipment. Since virtual reality places the user in a totally virtual environment, it is also used to train the user and perform simulations of potentially dangerous tasks. A prototype training system called Look, Stop and Fix trains the worker for recognizing hazard situations and applying preventive actions [18]. It simulates the working areas of mining industry in dangerous situations with the purpose that the worker trains his response to these hazards. The interface of the application is shown in Figure 8.

Figure 8. *Screenshot of available options when trainee has stopped the simulation [18]*

2.2 Human-Robot Interactions. Cobot

Over the years industrial processes have become increasingly complex, which has led to automate a large part of them. At this point Human Robot Interactions (HRI) emerge. HRI is the science of studying people's behaviour towards robots in relationship to the physical, technological and interactive features of the robots, with the objective to develop robots that facilitate the rise of human-robot interactions [19].

As happens every time a modification is introduced, contrary opinions arise in this regard. In the case of automation there were opinions that rejected the use of robots and there were also opinions that defended the complete automation of the process. However, neither of these two proposals is adequate since humans and machines provide different characteristics to the process. On the one hand, the robots optimize the performance of the process and have mechanical characteristics that operators cannot reach, but lack human and logical criteria. On the other hand, humans have experience and logic, but they are unpredictable and imprecise. The appropriate level of automation depends on the characteristics of each process and situation [20]. Table 1shows the strengths and weaknesses provided to the industrial process by humans and robots.

Table 1. *Human vs. Machine Strengths & Weaknesses [20]*

Human component		**Automated control system**	
Strengths	**Weaknesses**	**Strengths**	**Weaknesses**
Judgment	Inconsistent	Consistent	Lacks judgment
Adaptable	Vision, hearing, reach, strength, attention span limited	Predictable	Cannot be programmed for all eventualities
Sentient knowledge	Unpredictable, possibly unreliable	Efficient	Lacks sentient knowledge
Interactive	Subject to emotion, bias, alternative motivations	Uniform, reliable	Constrained by human limitations in design, installation, use
Can use experience	Forgetful, subject to distractions	Fatigue-resistant	Subject to wear and tear
Can learn, adapt	Subject to fatigue	No attention span limits	Adapted responses must be programmed—human programmers

From the abovementioned it can be concluded that automated processes require human intervention to optimize the process. In this type of processes, the operator is responsible for supervision and decision-making tasks, thus reducing injuries due to physically complex or dangerous tasks. But nevertheless, automated control systems are designed by humans, so they are sensitive to human errors and, because of that, the probability of accidents between humans and robots increases, either due to human error or due to machine error. Because of the speed and force with which the robots work, the injuries that can cause to the operator are more serious. For this reason, automation requires a high level security system that protects the user, which is translated into higher costs. It is almost impossible to design a system to respond correctly to every possible situation. Therefore, the engineer responsible for the design process must integrate the operator in a way that it can contribute with criteria and flexibility when the system deviates from the established path, with the purpose of taking advantage of the strengths of Table 1, both the user and the robot. The designer can intervene in five attributes that affect the interactions between humans and robots: level and behaviour of autonomy; nature of information exchange; structure of the team; autonomy, adaptation, learning and training of people and the robot; and shape of the task, which relates to the way the task has been done and how should be done [21].

Failures in a process can occur due to errors in the robots or in the decision making. Part of the errors caused by the latter are related to the operator trust in the automation and whether an operator over-relies or under-relies on the automation as a result. The “reliance” factor represents the probability that an operator will use automation and is influenced by the operator’s self-confidence and level of trust in the automation. Trust is influenced by the actual reliability of the automation and the experience it has working with robots, because the operator gains confidence with the robots as the operator acquires experience with it [22].

Conventional industrial robots have long proven their effectiveness in production chains. They produce massively, occupy a lot of space and often remain in a fixed position. Over the years, automation has been applied to more and more precise and detailed processes which are inaccessible for conventional robots. To fulfil this proposal, the Collaborative Robots (Cobots) were introduced. The cobots are 6-axis articulated robotic arms designed as worker's assistants and not as substitutes.

2.2.1 Cobot

Collaborative robots are intended for direct interaction with a human worker, handling a shared payload [23]. They are compact and occupy little space, so they are easily relocatable. In addition, they are designed so that they are flexible enough work with humans.

In terms of safety, industrial robots, as abovementioned, can sometimes be dangerous. Because of this, they need the installation of security barriers that protect operators. Since they lack sensors that allow them to detect the intrusion of an operator in their workspace they turn into potentially dangerous tools. On the contrary, when a worker intervenes in the operational range of the robot, the robot is stopped immediately to avoid injury. Figure 9 depicts a conventional robot contained in a cell and a cobot working with an operator with no barriers.

Figure 9. *a) industrial ABB robot in a cell [24], b) ABB cobot interacting with an operator [24]*

To achieve safe human-robot collaboration the maximal payload and velocity of the motion are decreased with respect to conventional robots. The load capacity of collaborative robots is around 10 kg and maximal velocity of the motion is limited to 250 mm per second. This light design ensures that de damage suffered after an impact is not as serious as it would be with a standard robot. In order to reinforce the safety for the worker the robot incorporates sensors detecting so that it can prevent collisions by stopping in case of sensing contact with the operator [25]. As a consequence, cobots do not need security barriers to keep operators safe, so both cobots and workers can be integrated into a production chain.

There are several ways to insert cobots in the production lines and each of these ways implies a different kind of HRI, as is illustrated in Figure 10.

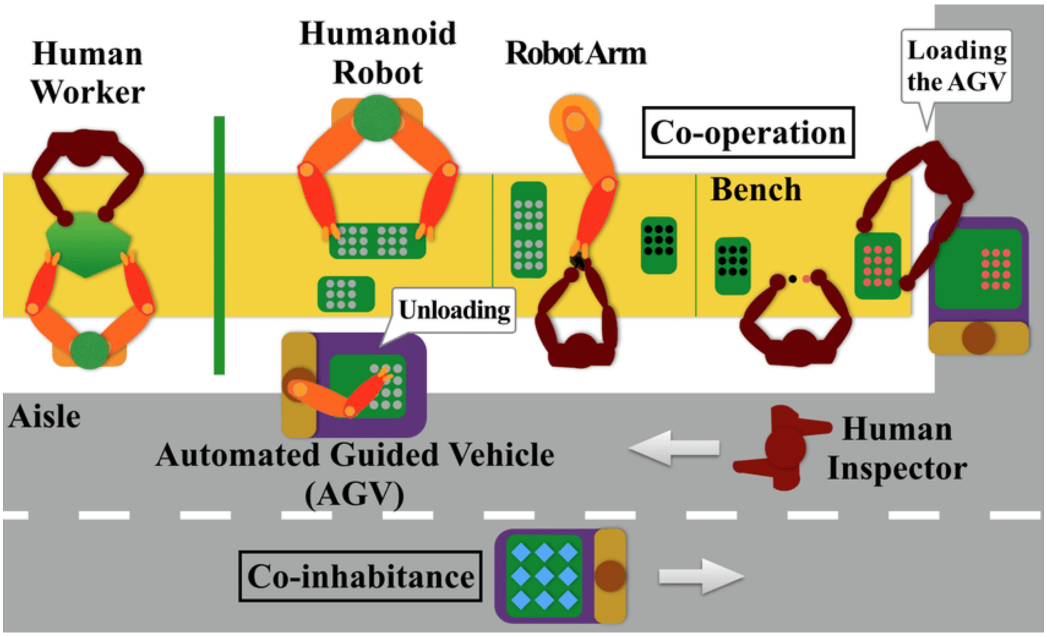


Figure 10. *Human-robot interactions in future production lines [26]*

Following the production line of the figure from left to right, the level of cooperation between the cobot and the operator increases. The first working couple interact through a work table, but the human does not enter the working range of the cobot. The second couple shows the interaction between an automated guided vehicle (AGV) and cobot. The third one is a collaborative interaction between a cobot and a human, which implies that the human and the cobot share the working range of the robot. At the end of the line of work, there is an interaction between an AGV and a human. In addition to these relations, in industrial areas there are also co-inhabitancy interactions, as illustrated at the bottom of the figure, in which humans and AGVs share the area, but they do not collaborate in any task.

As shown in the previous figure there are two types of cobots: humanoid (dual arm) single arm. Single arm cobots are the most extended but dual arm robots are arising in the industrial field. The two-arm design allows for increased application flexibility. Several researches have proven the advantage of using two arms versus one, especially for manipulation tasks [27]. Two arms can work together to handle objects and perform operations on them. By carrying out operations simultaneously, the two arms can also save time compared to using a single arm which would need to perform tasks sequentially. In some applications, they can also save space compared to using two separate single-arm robots. They allow objects to be approached at angles that would not be possible using a single arm by itself. However, they need careful programming to ensure that their arms coordinate their movements and operations perfectly with each other, and there are no agreed standardised methods for setting up and implementing two-arm robot applications. In addition, single arm cobots occupy a space similar to that occupied by a human, which can be critical when designing the layout of the industrial space.

An example of implemented cobot in industry is Walt at the Audi Brussels factory. Figure 11 shows the developed MRK-Systeme robot, along with its interaction with factory workers. The user interacts with the cobot using gestures and the cobot communicates with the user through a robotic head that expresses emotions [28].

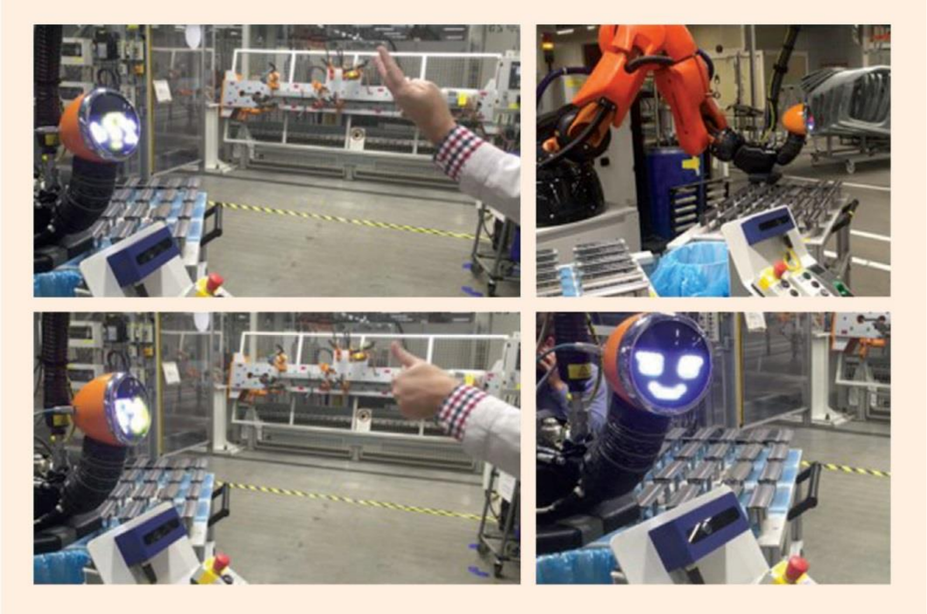


Figure 11. *Views of the cobot Walt at the Audi Brussels factory [28]*

2.3 Human as factory resource

The introduction of automation in the industry has significantly changed the role of operators in the production chain. Due to the constant innovation of industrial processes, it is necessary to qualify workers about new and changing technology trends. This demands introducing novel approaches for knowledge-delivery and skill transfer [29].

The cost that implies for the company the introduction of trainings of the operators is collected in the “Human Capital Theory”. Human Capital Theory was formalized by Becker in 1962 and contemporaneously developed by others. This theory recognises the skills, experience and knowledge that people have and the economic value of these to the companies [30]. It is considered, therefore, the education of the operator as an investment that will provide greater benefits [31].

Training is defined by the Cambridge Dictionary[1] as the process of learning the skills you need to do a particular job or activity. The objective of the training is to provide the employee with the skills that lacks when doing a particular job and to achieve that work with ease. The first step in establishing a training method is to determine the objective of the training and the skills to train. The training methods can be classified into two groups: On-the-job training (OTJ) and Off-the-job training.

OTJ is based on the philosophy of “Learning by doing”, that is, experiential learning. In this method the operator carries out his training period directly in the workplace. The trainee learns the methodology of the task through observation and puts it into practice under the supervision of a superior. This method is suitable for teaching technical skills; however, mistakes made by the trainee lead to errors in the production chain, which is expensive both economically and in time value.

Off-the-job training is performed out from the workstation. The trainee takes away the stress that comes with the real job and focuses on learning experience. Within this type of training are the classroom lectures and exercises and the simulations. As the processes become more complex, the study of virtual reality prototypes for this purpose increases. The VR provides a risk-free environment, where the user can not only learn without fear of getting hurt, but also learn the consequences of the various mistakes that can be made without affecting the real production line. This stress-free situation facilitates the learning of the task and the prevention of accidents [32].

Operators receive training before and during their tasks to ensure safe and efficient operations in industrial plants [33].Traditionally, workers train with hardware prototypes, along with several disadvantages such as the high cost, the limitations in scale and product variants or the limited hardware availability only at a late development stage. In order to bypass these limitations new training methods are being studied based on virtual reality. Virtual training was introduced for the first time more than two decades ago. Over the years, many studies have been conducted on the impact of using virtual reality for training tasks. Many of them have demonstrated a positive impact on the learning of industrial processes (Lin al. 2002; Malmsköld et al. 2007; Adams, Klowden, and Hannaford 2001), Despite these positive results, virtual training has not been introduced as a daily practice in industry yet [29].

---

[1] https://dictionary.cambridge.org/dictionary/english

2.4 Summary

This last section includes the main alternatives that have been shown throughout the state of the art, collected in table form.

Table 2 indicates the suitable situations where to use virtual, augmented or mixed reality, using as comparison parameters if the user is in the workplace or away from it, the interactions between real and virtual objects and the capability of the user to join both of them at the same time, and the level of immersion provided with its corresponding visual devices.

Table 2. *Comparison between MR, AR and VR*

<table>
<tr><th colspan="2"></th><th colspan="3">VR</th><th>AR</th><th>MR</th></tr>
<tr><td colspan="2">OJT (the user is in the workplace)</td><td colspan="3"></td><td>X</td><td>X</td></tr>
<tr><td colspan="2">Off-the-job training</td><td colspan="3">X</td><td>X</td><td>X</td></tr>
<tr><td colspan="2">User is aware about the real world</td><td colspan="3"></td><td>X</td><td>X</td></tr>
<tr><td colspan="2">Real and virtual contents interact with each other</td><td colspan="3"></td><td></td><td>X</td></tr>
<tr><td colspan="2">Immersion</td><td>Low</td><td>Semi</td><td>Fully</td><td>Low</td><td>Semi</td></tr>
<tr><td rowspan="3">Visual Displays</td><td>Head-worn display</td><td colspan="2">Smart-glasses</td><td>HMD</td><td>Smart-glasses</td><td>Smart-glasses</td></tr>
<tr><td>Hand-held display</td><td colspan="3"></td><td>X</td><td>X</td></tr>
<tr><td>Projective display</td><td colspan="3"></td><td>X</td><td></td></tr>
</table>

The shaded cells form groups, that is, smart glasses provide low and semi-immersed experiences while Head Mounted Displays provide a fully-immersive experience.

Table 3 focuses on the Human-Robot Interactions, comparing the differences of these interactions based on using a conventional industrial robot or a cobot. The aspects to be compared are: the payload of the robot, the direct physical contact if there is, and the way they interact depending on how close they are to the human and the robot.

Table 3. *Comparison between robot and cobot*

HRI	Payload	Human-Robot physical contact	Human-Robot Layout depending on the proximity →	
Robot	High	No	Robot isolated in a cell	
Cobot	Low (0-10 kg)	Yes	Shared Worktable (Human is not entering the cobot's operation range)	Shared Workspace (Human enters cobot's operation range)

Table 4 shows the advantages and disadvantages of single and dual arm cobots so it is easier to choose the most suitable option for each situation.

Table 4. *Comparison between single and dual arm cobots*

Single arm	Advantages	Occupy as little space as a person
		Most extended in collaborative field
	Disadvantages	No possibility of multitasking
Dual arm	Advantages	Suitable for manipulation
		Multitasking
		Increased application flexibility
		More similar to a human
	Disadvantages	Occupy more space than single arm cobots
		Difficult coordination of both arms
		no standard methods for setting up and implementing applications

2.5 Patent landscape

One technology that is attracting growing interest from a research and development focus or market entry perspective is Virtual Reality. Although this technology is being developed for its application in a wide variety of disciplines, this analysis focuses on the industrial field since it is the field of application developed in this thesis. The objective of this section is to shed light on recent patent activity in the training of tasks involving robots through VR with the target of providing insight for those working in this area.

It is possible to track patent applications with a visual representation of the data through a patent landscape, which provides graphics and charts to demonstrate patent trends and leading patent assignees. The search database used to obtain the dataset for this thesis is Derwent Innovation[2], by Clarivate Analytics, a patent research application that provides access to globally trusted patent in scientific literature. The search for patents was based on two couples of keywords that define the scope of study of this work: Virtual Reality and industrial environment; human-robot collaboration and training.

Figure 12 illustrates a bar chart for the optimized assignees. The top applicant for filings is Brain Corporation[3], whose target is the development of intelligent systems for everyday machines. Rethink Robotics Inc.[4] is second, with approximately half the number of filings of the top applicant, which is focused in collaborative robots for industrial automation.

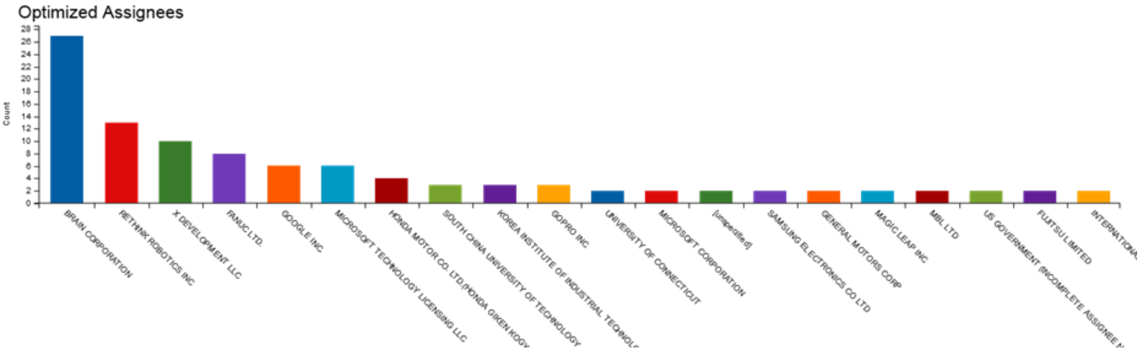


Figure 12. *Bar chart for the optimized assignees*

Figure 13 indicates the total number of patents for the top inventors associated with the top applicants.

---

[2] https://clarivate.com/products/derwent-innovation/
[3] https://www.braincorp.com/
[4] https://www.rethinkrobotics.com/

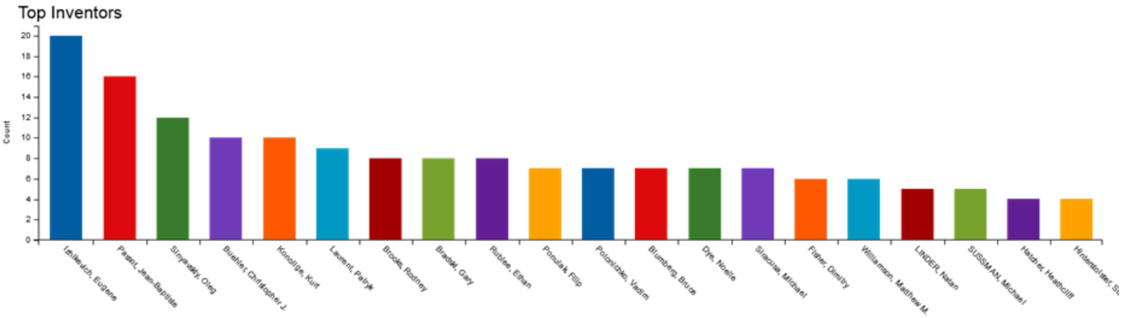


Figure 13. *Number of patents for the top inventors*

Figure 14 illustrates the top countries around the world owning patents based on the selected dataset.

Figure 14. *Top countries and regions*

The clear leader is United States with 91 patent families out of the 161 patents that comprise the dataset. The top two applicants are enterprise from USA with a total of about 40 patent families. Other mayor player with 32 patent families is China.

Figure 15 shows the number of patent families in the data set by priority year since 1995 to 2018. It depicts an upward trend in patent filings in general.

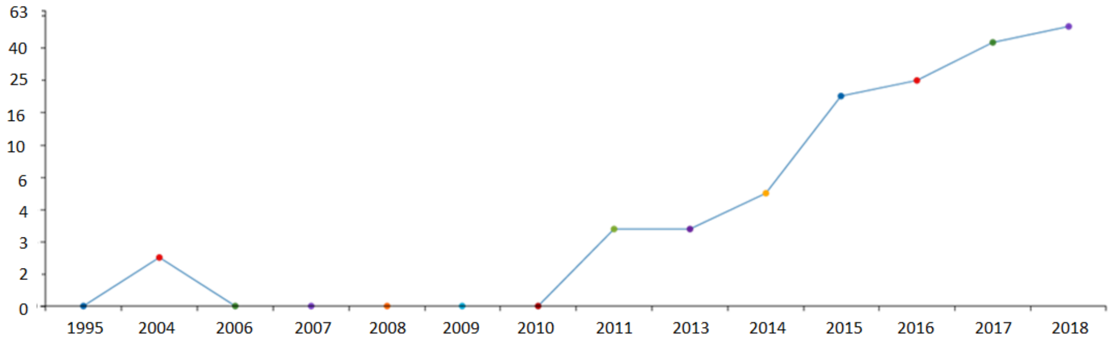


Figure 15. *Patent families by year*

In the early stages of this period, filings were mostly flat until 2010. From 2010 to 2014, fillings began to grow moderately, and from 2014 onwards the filings experienced a very sharp growth.

Figure 16 shows the patent map that represents the data by means of a graphic representation that takes characteristics of the cartography.

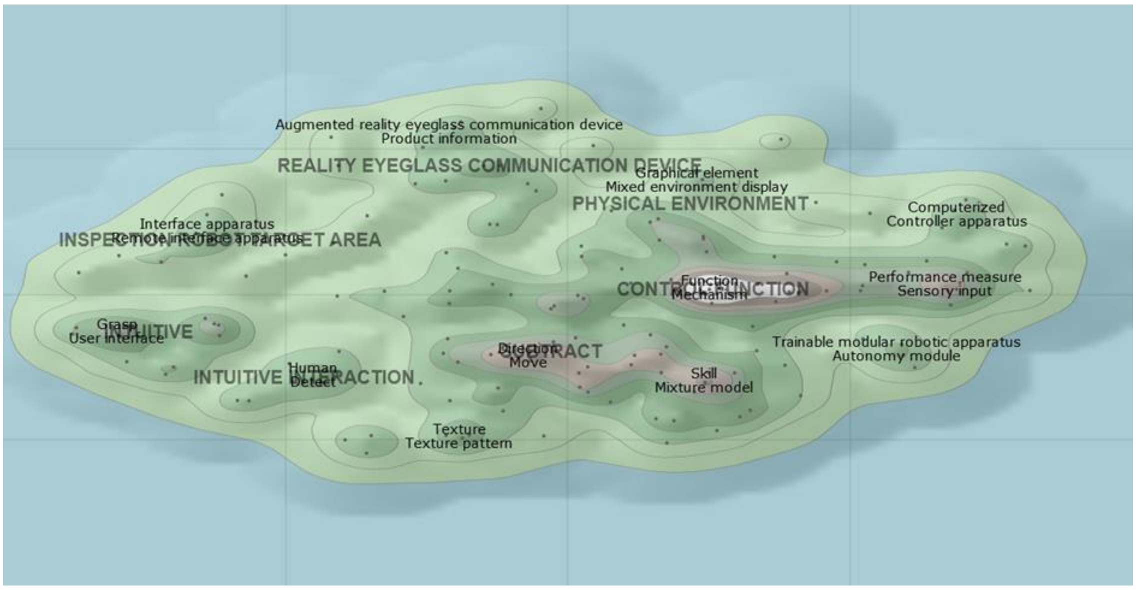


Figure 16. *Patent landscape*

The map shows that the patents are concentrated in the following areas of study: reality eyeglass communication device, physical environment, inspection robot target area, control function and intuitive interaction. The highest concentration of patents in this dataset relates to patents comprising keywords such as “control function”, “skill, mixture model”, “direction, move” and “performance measure, sensory input”.

3. DESIGN AND IMPLEMENTATION

This chapter describes the procedure that has been carried out for the selection of the training method implemented in this thesis. Therefore, it is explained the reasons why it has been determined that the selected proposal is suitable for the case of study.

The steps followed during the thesis work are presented, from the search of the suitable proposal for a training method to work with cobots, to the selection of the tools, devices and software, until the implementation of such proposal.

3.1 Training approach

The aim of this chapter is to design a training method for industrial tasks that involve interactions between humans and cobots. The introduction of robots in the industry has benefits for the productive process as illustrated in Table 1 of the state of the art. But it also has benefits for the operators who work with them since robots provide physical assistance to humans for reducing fatigue and stress [34]. But nevertheless, although the cobots have a high level safety system, it takes time to make the user feel comfortable working with them because of their fear of working with robots that may fail or damage them.

The level of confidence in the robots alters the behavior of humans in the accomplishment of the task. This problem is especially relevant in dealing with cobots since the operator shares the workspace with the robot. From this problem of trust between humans and robots arises the need to stablish a training method in which users feel comfortable and with which they can develop advanced skills in human-robot collaborative activities and become familiar with the collaborative interactions.

Since the cause of this problem is working with the cobot, this approach consists of an off-the-job training that it is carried out outside the workplace, as defined in the state of the art. Within this type of training are class-lectures and simulations. Many studies have been carried out that show that simulations are more effective when performing trainings with operators working with robots, as is the case of the study carried out by Salman Nazira, Alberto Gallaceb, Monica Bordegonic, Simone Colomboa and Davide Mancaa collected in Article [35]. The study focuses on evaluating the cognitive performance, skills and reactions of industrial operators in abnormal situations based on two training methods: Power Point presentation and 3D virtual environment. The experimental results showed that operators trained in a 3D virtual environment obtained better results when tested in emergency conditions.

Several studies and proposals have been made to increase the trust of the operators in the machines through training with simulation. One example of these proposals is the MITPAS Simulation presented in the article [4]. MITPAS is a simulation environment designed for measuring and evaluating the performance of mixed human-robot teams in military and non-military situations, with the purpose of providing meaningful measures of both human and unmanned system performance to prove the proposed configurations and identify new training requirements. Collaborative human-robot behaviour is studied in a simulated location with typical tactical features of roads, forests and a small village, depicted in Figure 17, where the operator controls an unmanned ground vehicle (UGV) for ensuring a safe route. The operator uses a keyboard, trackball mouse, and joystick to control the system.

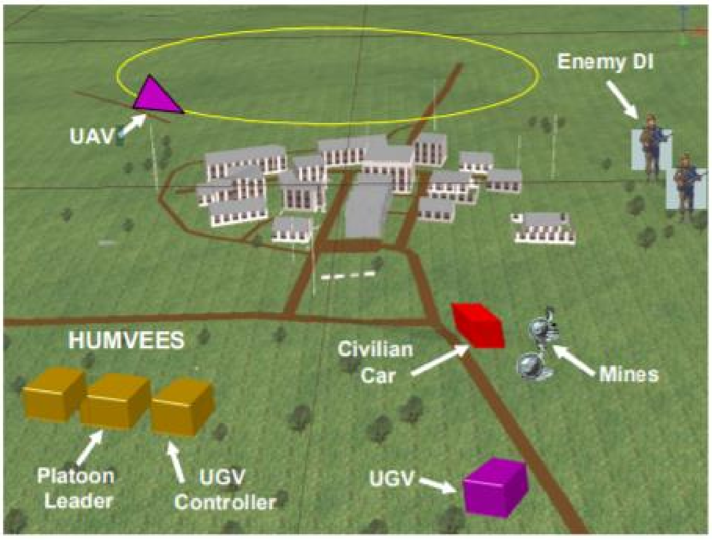


Figure 17. *MITPAS Experimental Environment [4]*

The objective of doing off-the-job training is for the user to learn to work with cobots without the need to have physical contact with them. Since the training will lack the physical robot, it is necessary to have a virtual model that behaves like the real one and with which the user can interact. In the example just described, MITPAS, a simulation is used in which the interaction with the machine is given by means of controls and a simulation on a screen. However, this type of interaction is not applicable to the training of collaborative tasks because it does not allow the user to interact with the virtual model in the same workspace. Therefore, a simulation is proposed in MR, AR and VR which allows the user to interact with a virtual model of the cobot working both in the same workspace.

Both MR and AR have awareness of the real world. This is suitable for trainings in which the user only needs a guide or support to perform the task. The objective of this approach is not only for the user to learn the procedure to be carried out, but also to make them feel comfortable and with sufficient confidence to work with the real robot at the end of the training process. For this purpose, it is best to use VR because it simulates the situation in the most realistic way, so that the user can immerse itself in the real process but with the certainty that the mistakes made during the learning process will not affect him or the chain of production. The benefits of higher immersion have been demonstrated through several experiments like [36], which found positive effects of this kind of immersion, particularly stereoscopy, on spatial understanding and interaction task performance.

Due to the chosen immersion, the most suitable equipment is formed by a Head Mounted Display, as the smart glasses cannot provide enough immersion.

According to [37], a basic VR System is composed by 5 main elements: the Task, the User, I/O Devices, VR Engine and Software & Databases. These components and the communication channels between them are represented in Figure 18, adapted from [37].

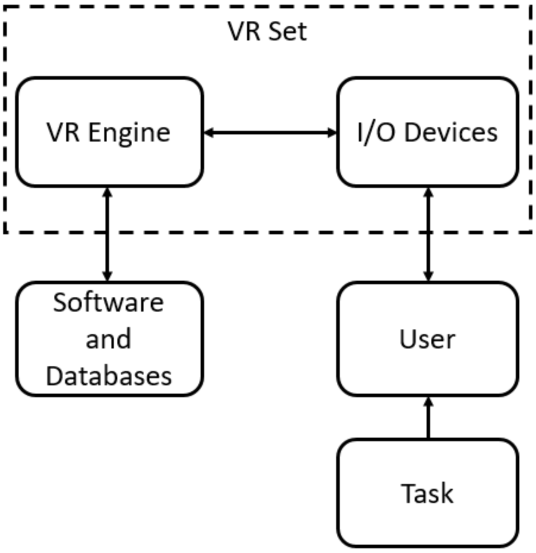


Figure 18. *Basic VR System*

Figure 19 depicts a schema of the communication flow between the first three components (Task-User-I/O Devices). While the user is engaged in the task, he/she provides inputs and receives information to and from the Output Devices, such as the HMD.

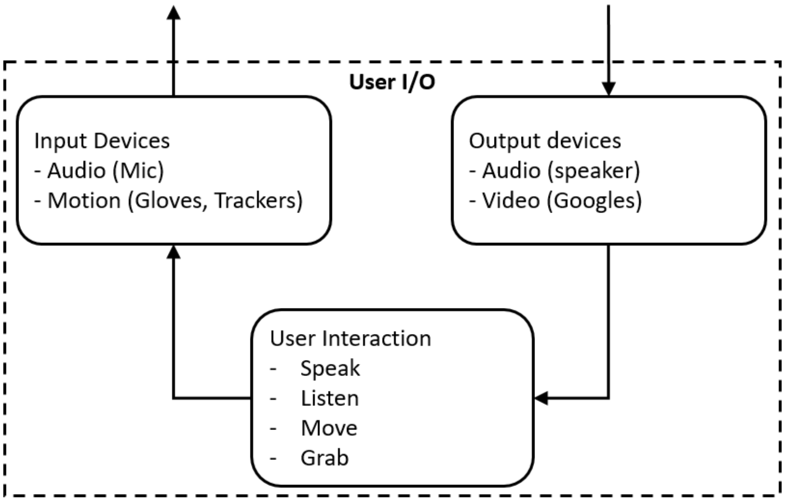


Figure 19. *User I/O communications*

This I/O flow is directly connected to the VR Engine which can be either a multiprocessor or a graphics accelerator. This component processes the input information and sends it to the Software & Databases block. After this process the VR Engine provides some information in the outputs as a response to the inputs received [1].

3.2 Training requirements

In case the cobot software with which to work had the capacity to offer simulations in VR, the problem would be solved. To overcome this limitation this thesis work looks for a solution as generic as possible which allows to work with all types of robot or cobot, and to accept every kind of virtual reality devices. The proposal must comply with a series of requirements:

- Compatible with immersive HMD
- Contain a virtual model of the cobot moving as the real one
- User interacting with the virtual environment
- Capable of simulating collaborative tasks
- Generic

When conducting a background research, meeting these requirements is a proposal of an immersive Virtual Reality Training System (VRTS) for gaining experience in “be-Ware of the Robot” for human–robot collaboration in manufacturing tasks [38]. The virtual environment inspired in an industrial layout, the virtual models of robot and user, and the corresponding interactions are designed in Unity. Figure 20 illustrates the virtual world of the simulation. The user is immersed in the simulation through a HMD and the hand tracking is performed through Kinect sensor. Thanks to these devices the virtual model of the user responds to the gestures of both head and hands, as shown in Figure 21.

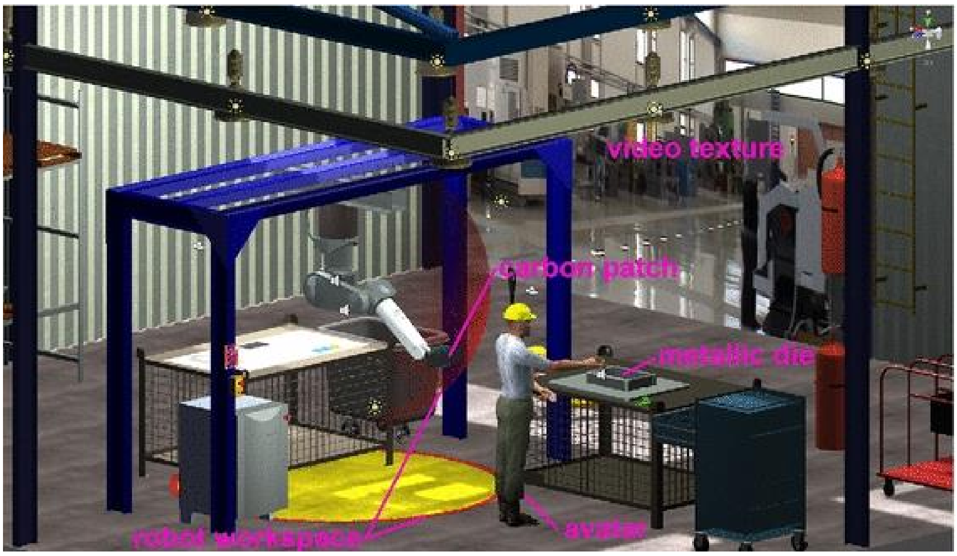


Figure 20. *The shop-floor environment and the main VE components as seen from a third person shooter inside Unity3d [38]*

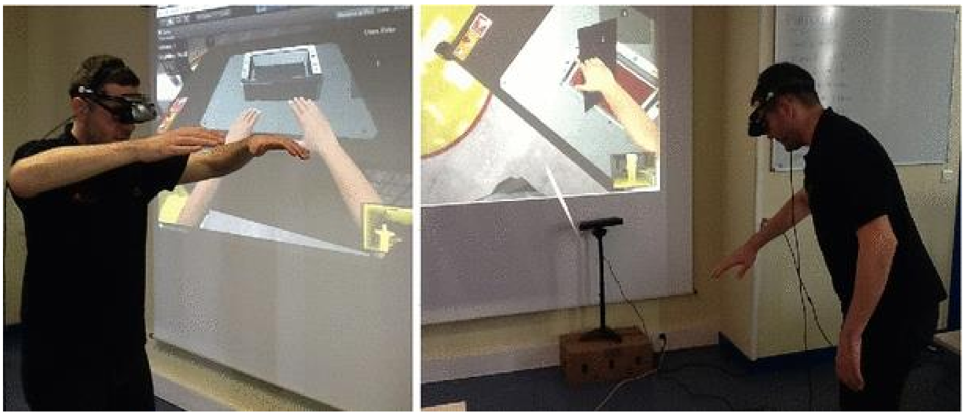


Figure 21. *Hands tracking example-avatar's hands following the user's arm movements (left), and avatar's hand moving a patch and placing it onto the red-coloured surface of the die (right) [38]*

BeWare of the Robot was tested by a group of 30 final year mechanical engineering students that answered a questionnaire for evaluating the experience, mainly the aspects of immersion and involvement. The system had good results in "presence by involvement" and in navigation issues, but also a number of limitations were found in the use of head and hands tracking devices.

As for the Kinect sensor, almost half of the users "lost control" of their hands. That is in part due to the fact that Kinect sensor has difficulties distinguishing the left from the right hand. In addition, when users turn more than 90º to the Kinect sensor axis the virtual model gets stuck for a while, although the user may continue moving. In terms of tracking quality newer I/O devices were recommended, such as the Kinect for Xbox

One sensor for tracking, the Leap Motion device for precise hand/finger tracking as well as immersion devices offering wider field of view and extensive head tracking (e.g. the Oculus Rift).

3.3 Proposal

After analysing this method of training it has been found that, despite being a generic solution, as requested in the requirements, for the training of collaborative tasks with robots, it presents a series of limitations among which are the problems with the tracking of user movements and the need for high levels of programming in C# to carry out the design of both the scenario and the interactions between user and virtual environment.

In order to solve these limitations, this thesis work proposes a new solution: the use of WebVR. This alternative can be applied to any task, using any type of robot and the user can interact with both hands.

WebVR is a JavaScript Application Programming Interface (API) that provides support for virtual reality devices in a web browser, so it is possible to experience VR directly in the browser. It makes easier for everyone to access virtual reality experiences, no matter what VR device is available. There are only two requirements for implementing this alternative: a VR headset (and controllers if needed) and a compatible web browser. Different headsets need different browsers.

WebVR experiences are built on WebGL (Web Graphics Library), developed and maintained by Khronos Group [5]. WebGL is a JavaScript API for rendering interactive 2D and 3D graphics within any compatible web browser [39]. Nowadays, most web browsers do not support some of the plug-ins. New JavaScript standards (HTML5, WebGL, etc.) already provide rich interactive feature, and the current trend is moving to JavaScript. Based on WebGL, there are many 3D graphic engines available in the market such as Unity[6], Unreal[7], Three.js[8] and Babylon.js[9] [40].

Figure 22 depicts the WebGL calls, adapted from [41] to make it generic.

---

[5] https://www.khronos.org/
[6] https://unity.com/
[7] https://www.unrealengine.com/
[8] https://threejs.org/
[9] https://www.babylonjs.com/

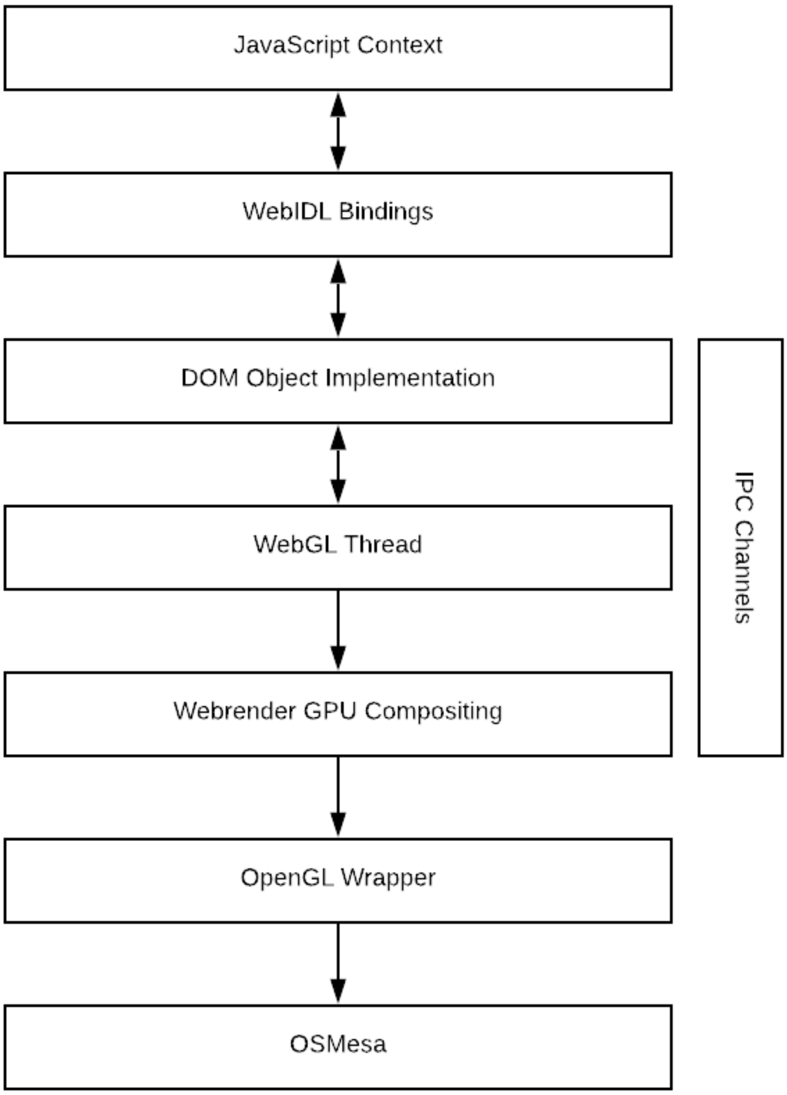


Figure 22. *WebGL calls*

Off-Screen Mesa (OSMesa) interface is used to render in the memory assigned by the user without any dependencies of the windows system or the operating system. Open Graphics Library (OpenGL) is a cross-language, cross-platform Application Programming Interface (API) for rendering 2D and 3D vector graphics. The API is used to interact with a Graphics Processing Unit (GPU) in order to achieve hardware-accelerated rendering. The Document Object Model (DOM) is a cross-platform and language-independent API that treats an XML document as a tree structure wherein each node is an object that represents a part of the document. The DOM represents a document with a logical tree. DOM methods allow access to the tree and one can change the structure, style or content of a document. Layers of Webrender GPU Compositing, WebGL Thread and DOM Object Implementation are contained in Inter-Process Communication (IPC) which allows the processes to manage shared data. Web Interface

Description Language (IDL) format describes APIs that are intended to be implemented in web browsers. On top of the WebVR structure is the JavaScript Context which relates to objects.

The WebVR API is designed for operating with any VR device by following the next steps:

1) Search for available VR devices.
2) Check if the available device is compatible with using browser.
3) The application informs the user about available devices in case they are compatible.
4) User enters the VR mode.
5) Graphical frames are displayed on the VR device.

The WebVR project is accessed by the user throughout internet in the PC. The corresponding server generates a communication flow based on request-response between the user accessing the browser and the virtual environment. If there is virtual reality equipment connected to the PC the user can enter in the virtual world of the WebVR project by wearing the VR-Headset and interact with it through the controllers.

WebVR was first conceived in 2014 by Vladimir Vukićević from Mozilla[10]. Although it is relatively new and has been used in many gaming applications, still in the development stage, it has potential for applications such as immersive analytics [42] or that presented in this thesis work.

WebVR projects comply with all the above requirements and solve the problems presented by the proposal performed with Unity. There are different WebVR frameworks, some of which have more simple and intuitive programming languages than the one required by Unity. In addition, with Unity a lot of plugins and functionalities are needed that with WebVR are not. WebVR allows for easy sharing of an experience on all headsets, while Unity requires downloading the application, with the correct version for runtime.

Once the proposal has been defined it is necessary to select the tools and task to be used in the implementation.

---

[10] https://www.mozilla.org/

3.4 WebVR Project

The web is becoming a delivery platform for virtual reality and 3D experiences rendered with WebGL. Nowadays there are several available frameworks online such as XSeen[11], X3DOM[12], Three.js, React VR, BabylonJS or A-Frame that are presented next.

XSeen is a declarative language for the XR Web. It uses attributes similar to HTML that are fully integrated into the DOM and can be manipulated with JavaScript.

X3DOM is an open-source framework for 3D graphics on the Web. Its objective is to comply with the current HTML5 specification for declarative 3D content and allows including X3D elements as part of any HTML5 DOM tree.

Three.js is a JavaScript library/API for creating and displaying animated 3D computer generated models in a web browser using WebGL.

BabylonJS is an open source 3D engine based on WebGL and Javascript used to create 3D experiences with HTML5, WebG and WebVR.

A-Frame is an open-source web framework for building virtual reality experiences. It is a three.js framework that brings the entity-component-system pattern to the DOM.

The option chosen for this project is A-Frame since it meets all the requirements to carry out this work and is compatible with a wide range of VR devices.

A-Frame is a web framework for building virtual reality (VR) experiences. A-Frame is based on top of HTML, which is an easy and intuitive programming language. It provides a declarative and extensible structure to three.js. A-Frame was developed by Mozilla to develop VR content and now is maintained by the co-creators of A-Frame within Supermedium. A-Frame supports most VR headsets such as Vive, Rift, Windows Mixed Reality, Daydream, GearVR, Cardboard, Oculus Go. A-Frame defines fully immersive interactive VR experiences that go beyond basic 360° content, making full use of positional tracking and controllers. If not having a VR Headset A-Frame still works on standard desktop and smartphones. It makes simple the use of VR headset and controls as there is no need of installations [43]. Its code, HTML, is easy to read, understand, and copy-and-paste. Being based on top of HTML, A-Frame is accessible to everyone: from web developers to VR enthusiasts. Updates of 3D object are made in memory with little overhead. A-Frame runs components such as geometries, materials,

---

[11] https://xseen.org/

[12] https://www.x3dom.org/

lights, animations, models, text, and controls for most headsets. A-Frame has been used by companies such as Google, Samsung, Ford, Chevrolet, among others.

3.5 Head Mounted Displays (HMD)

Generally all the HMDs can be classified into two groups: high-end HMDs, such as the Oculus Rift or HTC Vive, and mobile-based HMDs that include the Samsung Gear VR and Google Card-board.

High-end HMDs provide a comfortable user experience with a separate display screen, a complex device structure and advanced technology. Sometimes they include strong computation ability with powerful sensor system [40]. Figure 23 illustrates some examples of HDMs.

Figure 23. *Various kinds of HMDs [40]*

The training to be implemented needs to engage the user in the simulated task will be used in this implementation. In this implementation High-end HMDs are used since they have higher quality than mobile-based. In addition, the training needs hand tracking. By selecting a HMD with compatible hand-controllers the problems of hand-tracking presented in the study performed with kinetic sensor are solved. Sensor gloves would also solve those problems but the controllers are more suitable for this training as such grade of precision is not needed and the controllers are a more economic option.

In this thesis work the Oculus Rift[13] equipment (Figure 24) is used since they meet the requirements that have just been imposed and are both the HMD and its controllers are compatible with A-Frame.

---

[13] https://www.oculus.com/rift/

Figure 24. *Oculus Rift equipment [44]*

This equipment is composed of three elements: a wired headset, touch-controllers and two sensors for translating the user’s movements into VR.

Figure 25 shows the connection of Rift equipment. The HMD and the controllers are detected by the sensors which are connected to the PC passing through an external GPU that accelerates 3D rendering process, making it possible to run 3D application smoothly inside web browsers.

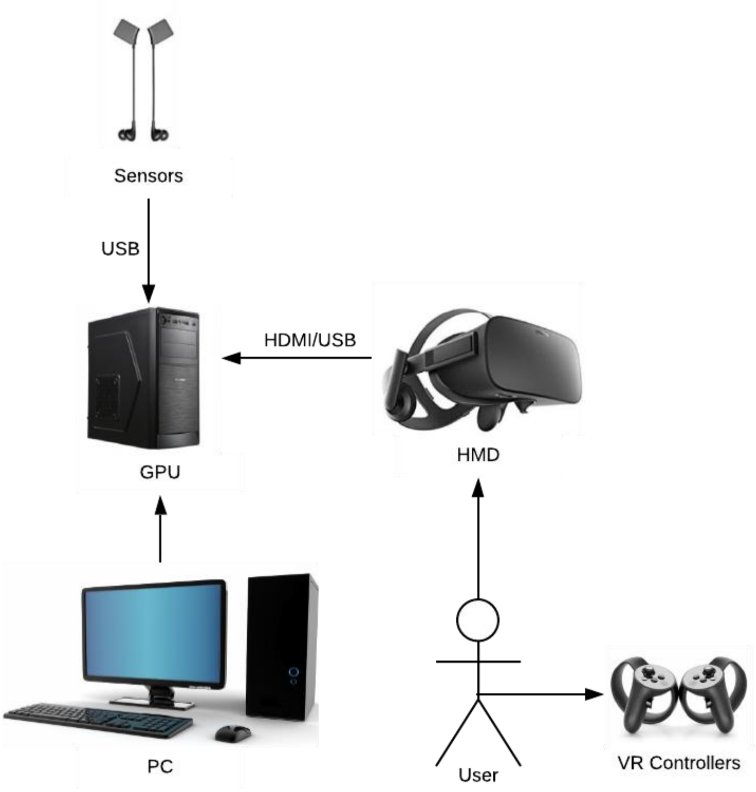


Figure 25. Connection of the Rift equipment

3.6 Cobot

Based on the state of the art, the use of a two-armed robot is chosen for this proposal since it resembles a human being more. Although a dual arm robot requires more attention from the user as both arms of the robot move at the same time, it is used in this implementation because its human-like appearance might favour the user's initial trust in the cobot.

The selected cobot for this application is YuMi, the ABB[14] commercial cobot. YuMi (IRB 14000), illustrated in Figure 26, is a dual arm collaborative robot from ABB.

---

[14] https://new.abb.com/

Figure 26. *The YuMi robot from ABB used in the research work [24]*

It was introduced in 2015 as the world's first genuinely collaborative robot. Its objective is to enable users and robots safely and productively work side-by-side, without any fence. As cobot, it includes safety in its functionality. YuMi meets the demands of greater flexibility in automated manufacturing. It is designed with two arms to perform human-like movements in assembly tasks with small components that require precise actions in a small space [45].

The technical data (Table 5) [46] of the cobot are within the established range to ensure the safety of the operator:

Table 5. *Technical data IRB 14000 YuMi*

Reach	0.559 m
Payload	0.5 kg
Max TCP Velocity	1.5 m/s
Max TCP Acceleration	11 m/s*s
Acceleration time 0-1m/s	0.12s
Position repeatability	0.02 mm

This cobot is already implemented in some production lines; for example, DEONET[15], a European market-leading manufacturer of promotional products. As human and cobot work together without barriers, the worries of operators for getting injured arise. That is

---

[15] https://www.deonet.com/

the reason why the learning curve is quite steep during at the beginning, until the user gets comfortable with the cobot and how it interfaces with him. This fact was verified in DEONET by Joanie Slegers, assembly department manager: “Having the YuMi on our plant floor took some getting used to for our workers, but once they saw that the robot was safe and helped boost our productivity, they really welcomed the new member of our team” [47].

As mentioned above, when the software is included in virtual reality simulations, it could be used to train users in this particular case in which an ABB robot is used. However, after analysing this option, there are greater limitations than the type of robot.

ABB's RobotStudio software offers simulations of the paths programmed for the robot in virtual reality [48]. This software allows the usage of the VR set for teaching a virtual model offline which facilitates the simulation of robots’ functionality [1]. RobotStudio provides the performance of training, programming, and optimization tasks without affecting the production line. The main advantages of this software are risk reduction, quicker start-up and increased productivity.

RobotStudio is built on the ABB Virtual Controller, which uses exactly the same software as the physical one that runs the production robots. This provides highly realistic simulations with real robot programs. The programing of the robot’s sequences and paths is done in RAPID code. RobotStudio has libraries that collect all the robots from ABB, including the YuMi, so the simulation in VR of YuMi assembling the box can easily be done by inserting the RAPID code of the process and entering the virtual environment. Figure 27 and Figure 28 show the interface of the software and its VR simulation respectively.

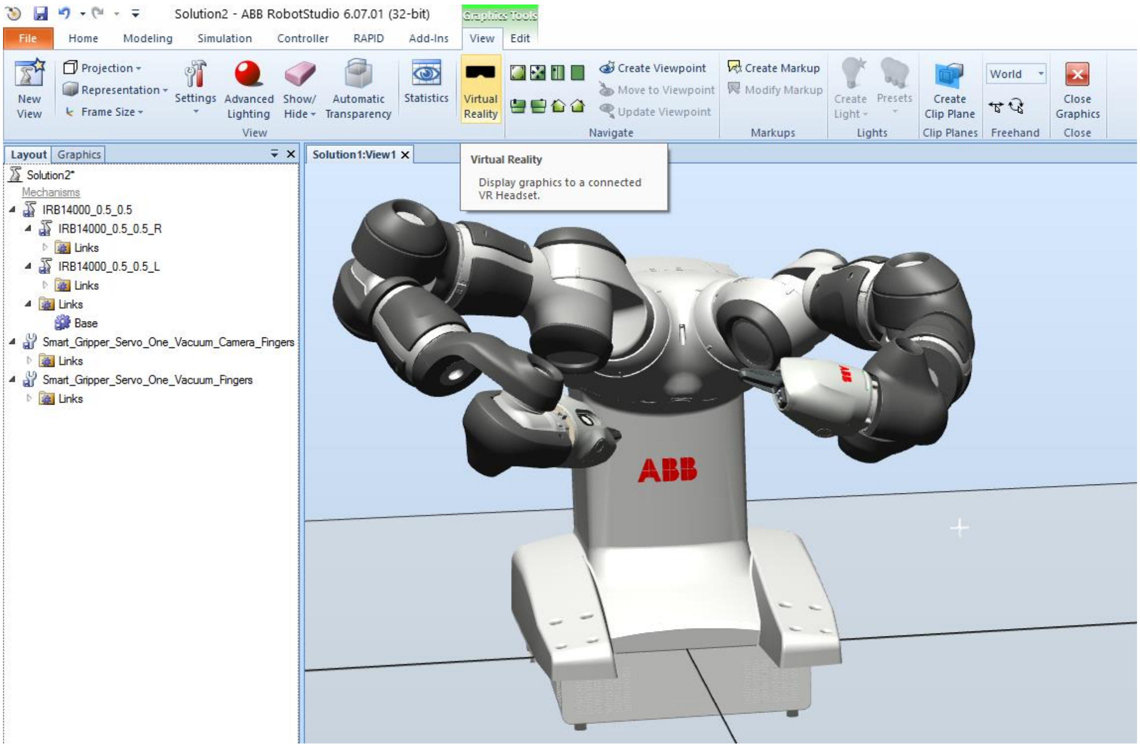


Figure 27. *RobotStudio interface with YuMi model*

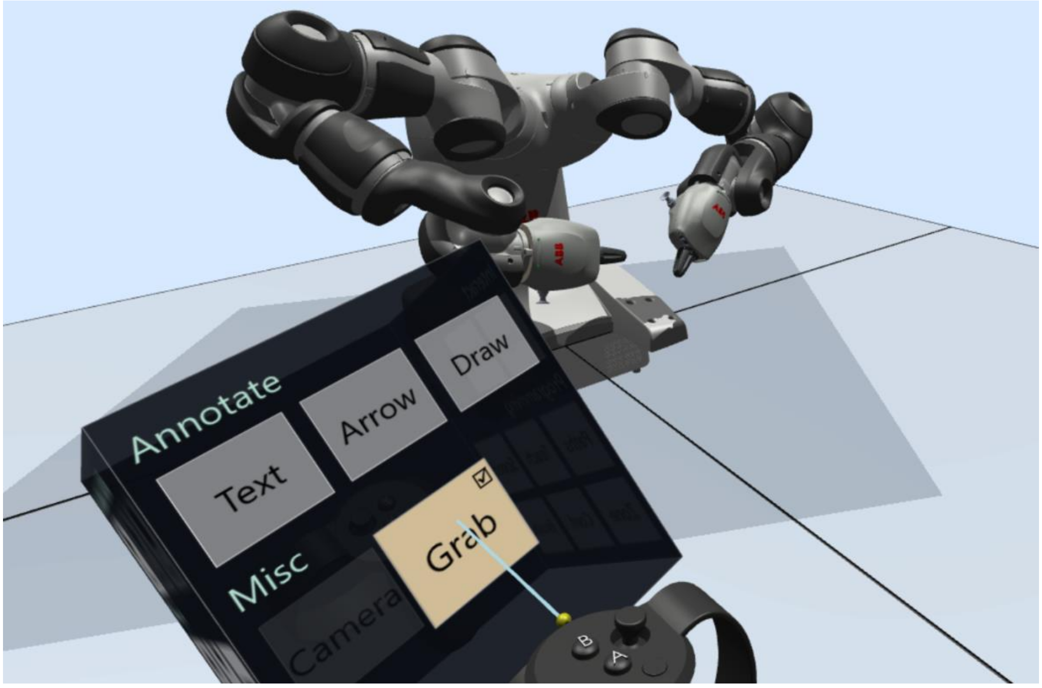


Figure 28. *Virtual environment of RobotStudio with YuMi model*

As shown in Figure 28, the VR environment consists on the display of the workspace built in the offline mode and the interactions are performed through the virtual controllers that correspond to the ones connected to the PC (Rift Touch Controllers in this case). The left hand contains a menu and the right hand is used for selecting the different options of the menu and interacting with the virtual YuMi. Thanks to the different menu options the user can manipulate the links of the robot ('grab' option shown in

Figure 28), generate arrows, draws and text labels, control the sequences of the simulation or adjust the robot's position, among others.

Table 6 compares the advantages and disadvantages of the exposed alternatives for this specific case which are RobotStudio and WebVR project so it is possible to visualize the advantages of the proposal of this thesis even when having VR mode in the software.

Table 6. *Comparison between RobotStudio and WebVR for virtual simulation*

RobotStudio	Advantages	High quality graphics
		Accurate and realistic movements
		Same simulation programing as for the real robot
	Disadvantages	Limited to ABB robots
		Left hand is occupied by a menu, so it is limited to the simulation of tasks in which the user only requires interactions with one hand
WebVR	Advantages	Applicable to any kind of robot
		No need for any software installation
		User can interact with both hands, which seems more natural to the user
	Disadvantages	Low quality of the graphics

If focusing on the quality and accuracy of the simulation, the best choice would be RobotStudio. However, the goal of this study is to design a training method applicable to all possible industrial tasks with cobots. This is not achievable for RobotStudio; therefore, the balance between the advantages and disadvantages of WebVR exceeds that of RobotStudio.

So, the most suitable alternative to be implemented is WebVR, which, although has graphics quality limitations, provides a more natural interaction for the user as it can use both hands [49].

3.7 Collaborative Task

Due to the fact that the scope of this thesis work is collaborative tasks, it is necessary to choose a collaborative task to be carried out with the selected cobot. Since the two-arm cobots are suitable for the manipulation of objects, as defined in the state of the art, an assembly task is sought. With bimanual manipulation, a robot can simultaneously control two objects, one in each hand, in order to better control the objects with respect to one another, and interact with the user [50]. A collaborative task requires both YuMi and the operator working together for its performance at the same time. In this section two alternatives are presented: a crane and a box.

3.7.1 Crane

The first alternative is a crane. Figure 29 shows the CAD model of the assembled crane and Figure 30 illustrates its exploded view for showing all parts the crane is formed by and the order they are assembled.

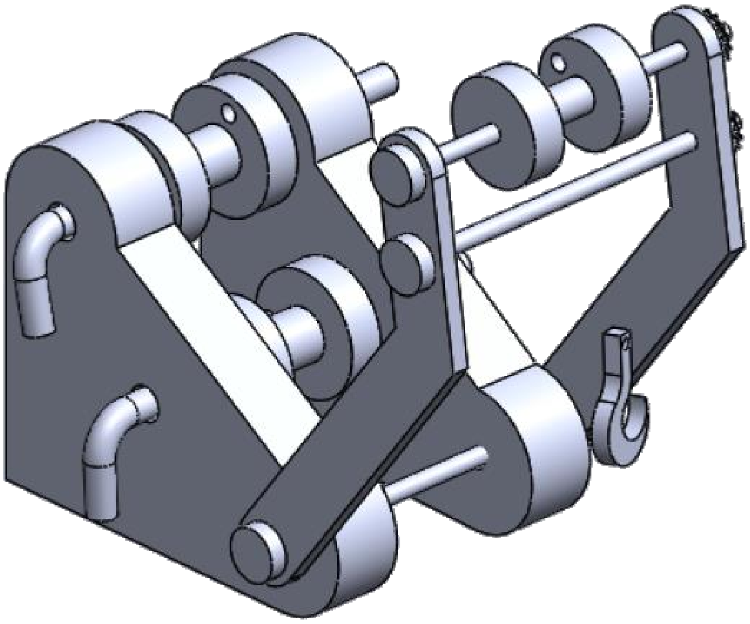


Figure 29. *CAD model of the crane*

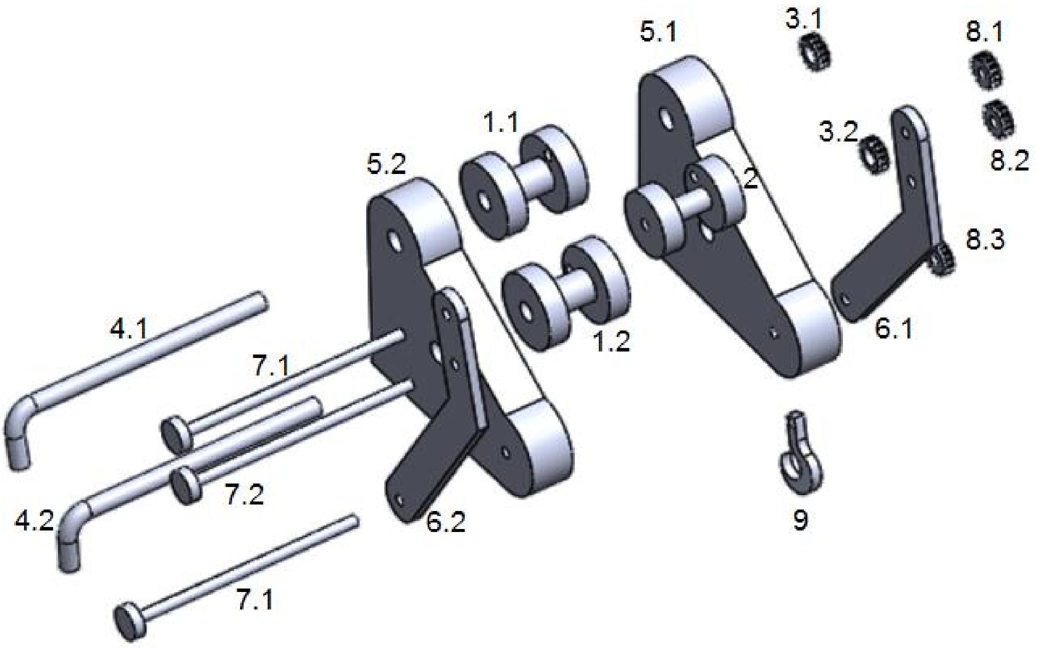
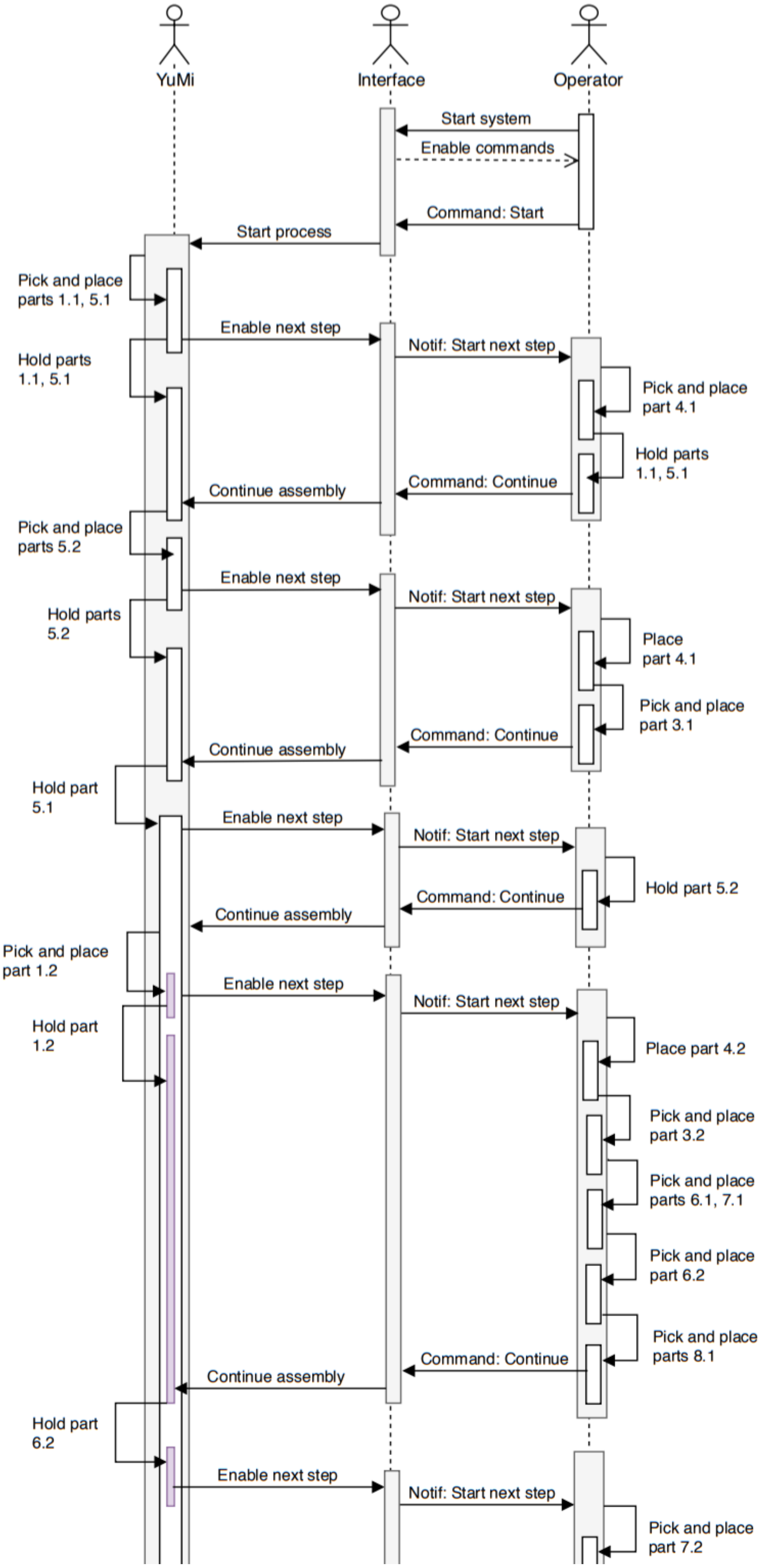


Figure 30. *Exploded view of the crane*

Figure 31 depicts the 3D printed prototype of this crane. To make it work it is necessary to make two rope circuits. By rotating the handles two movements are performed: raise the position of the mobile part of the crane and raise and lower the hook.

Figure 31. *3D printed prototype of the crane*

The following sequence diagram, Figure 32, describes the steps needed to assembly the box distinguishing the tasks performed by the YuMi, the user and the interface which coordinates the movements of both YuMi and user. This implementation was not performed due to time constrains.


YuMi
Interface
Operator
Start system
Enable commands
Command: Start
Start process
Pick and place parts 1.1, 5.1
Enable next step
Notif: Start next step
Hold parts 1.1, 5.1
Pick and place part 4.1
Hold parts 1.1, 5.1
Command: Continue
Continue assembly
Pick and place parts 5.2
Enable next step
Notif: Start next step
Hold parts 5.2
Place part 4.1
Pick and place part 3.1
Command: Continue
Continue assembly
Hold part 5.1
Enable next step
Notif: Start next step
Hold part 5.2
Command: Continue
Continue assembly
Pick and place part 1.2
Enable next step
Notif: Start next step
Hold part 1.2
Place part 4.2
Pick and place part 3.2
Pick and place parts 6.1, 7.1
Pick and place part 6.2
Pick and place parts 8.1
Command: Continue
Continue assembly
Hold part 6.2
Enable next step
Notif: Start next step
Pick and place part 7.2

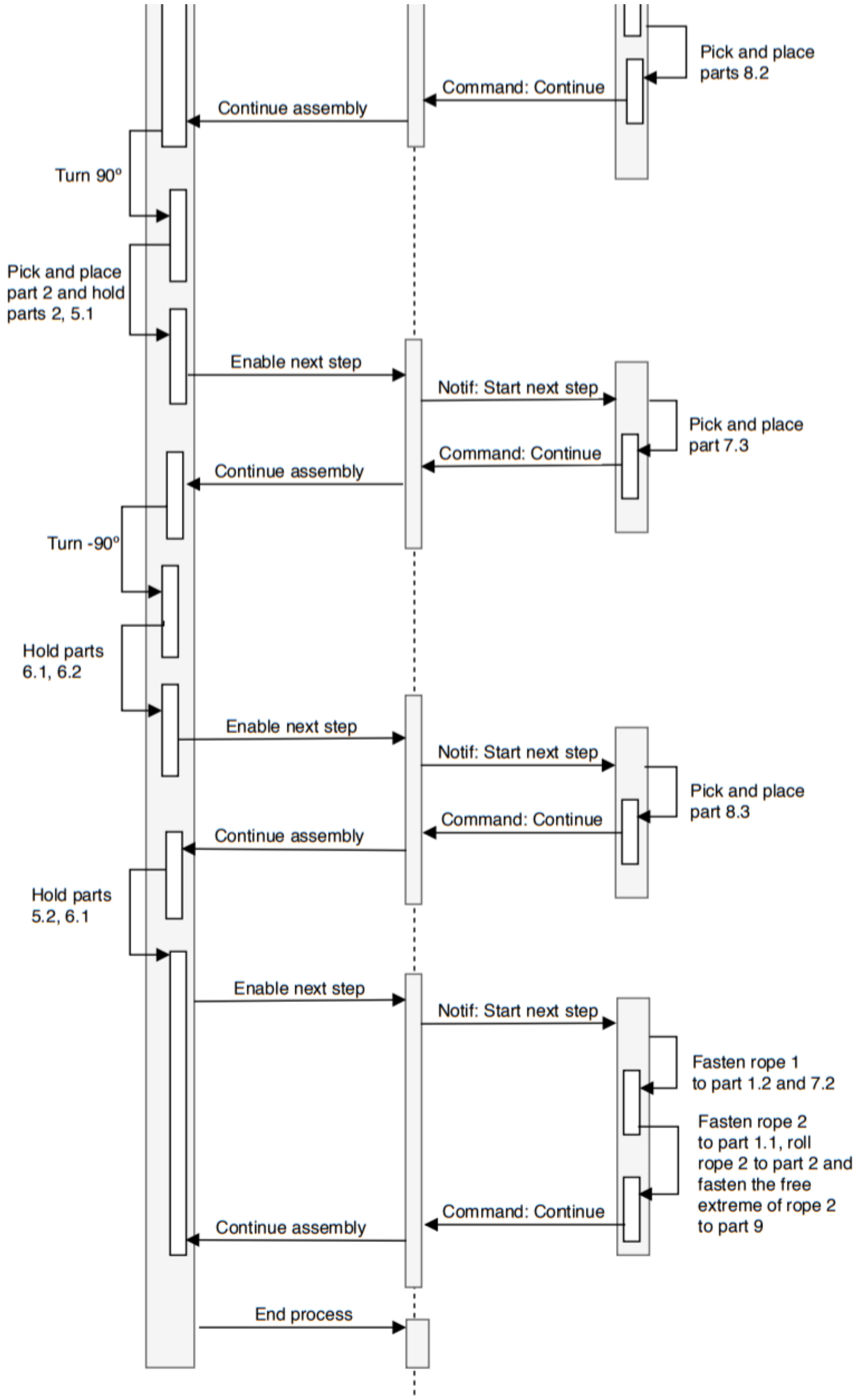


Figure 32. *Sequence diagram of the crane assembly process*

3.7.1 Box

The box [51] is made of wood and consists of 6 sides, linked with a total of 12 screws if the top side is not locked and 16 screws if it is locked. Figure 33 shows the CAD model of the assembled box and Figure 34 illustrates its exploded view for showing all parts the box is formed by and the order they are assembled.

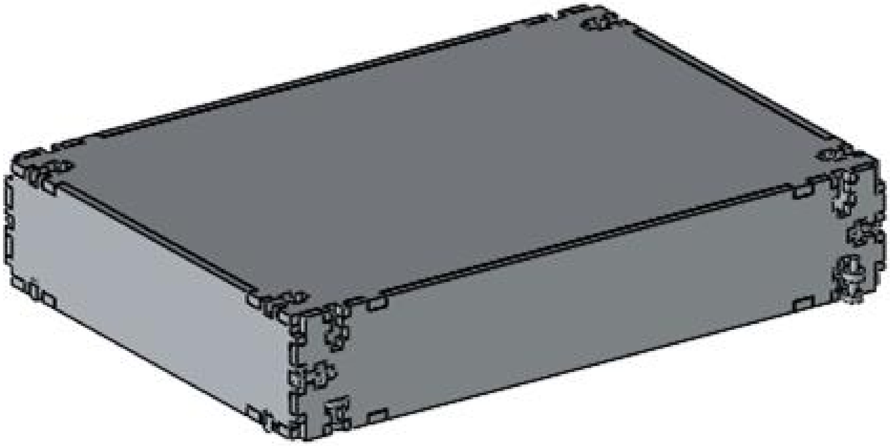


Figure 33. *CAD model of the box*

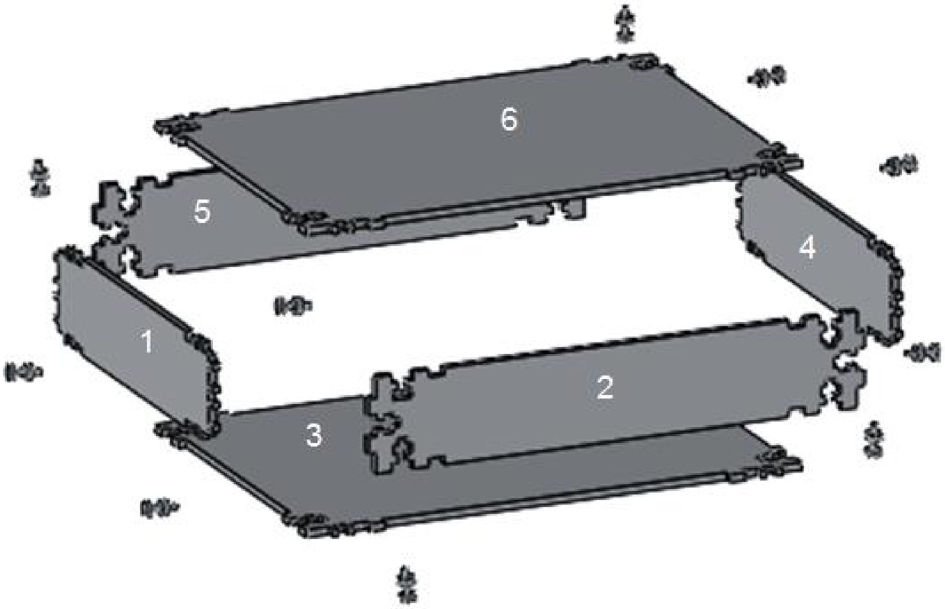


Figure 34. *Exploded view of the box*

Figure 35 depicts the wooden prototype of this box.

Figure 35. Wooden prototype of the box

YuMi’s work is to pick up the different parts of the box and assembly them in a proper position for the user’s activity. The human operator must link the parts that the robot is holding with bolts and screws and the interface is responsible of coordinating the tasks of user and cobot. The task consists of an iterative process that is repeated until the box is completed. This process is explained through the sequence diagram of Figure 36.

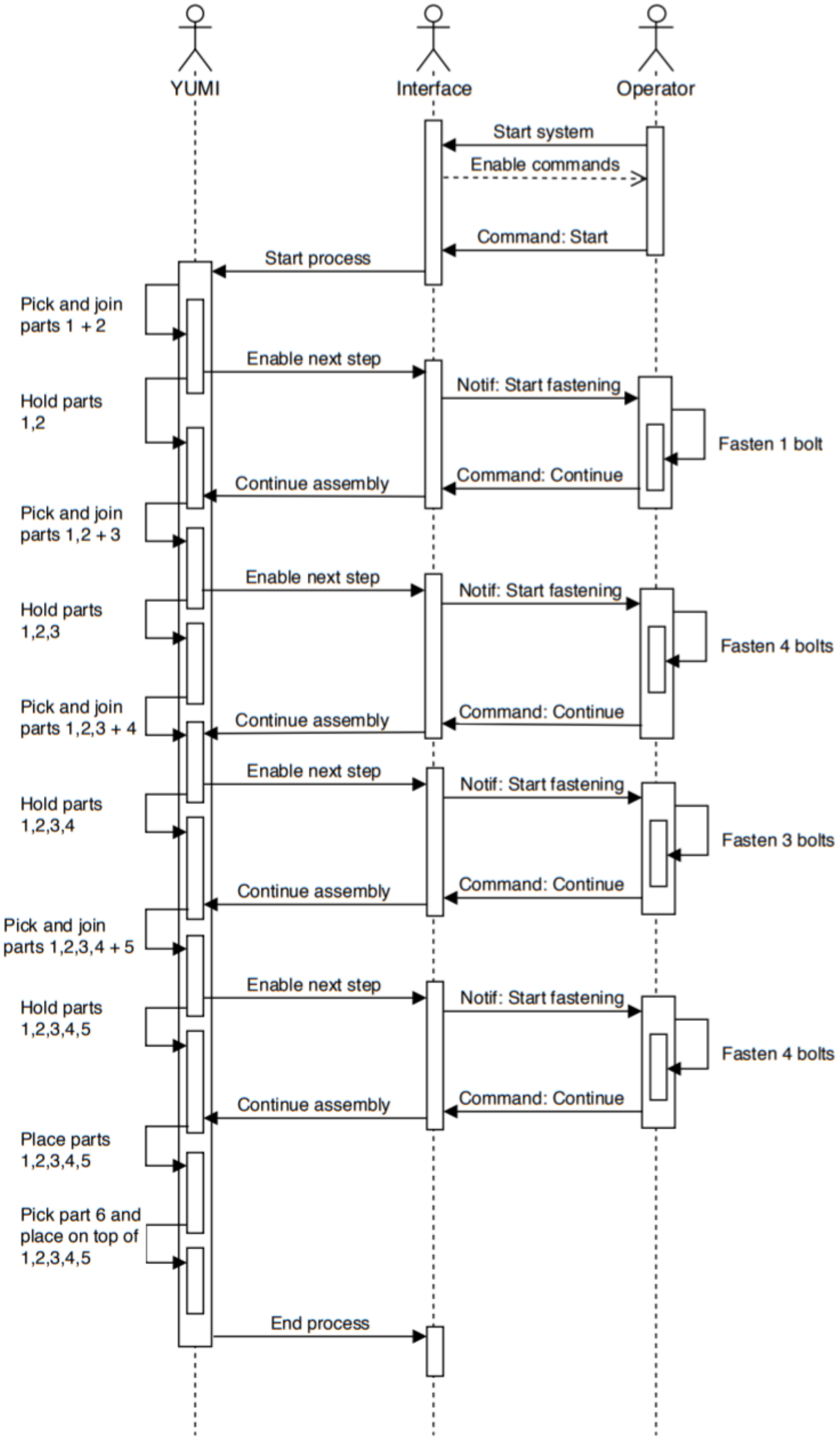


Figure 36. *Sequence diagram of assembly process of the box*

3.7.1 Task Selection

Although both alternatives are valid for the case study, the design of the crane presents more disadvantages in comparison with that of the box. The crane requires precise tolerances so that the pieces do not show resistance when they are joined or have an excess of slack that annuls their functionality. These tolerances are obtained through the machining of the pieces, which means a higher cost and time compared to the production of the pieces of the wooden box.

Since the assembly process of both is valid without providing advantages one option over the other, and since the difference in material does not make any difference to the study of training, the task to be implemented is the box. Figure 37 illustrates one of the steps of the collaborative process.

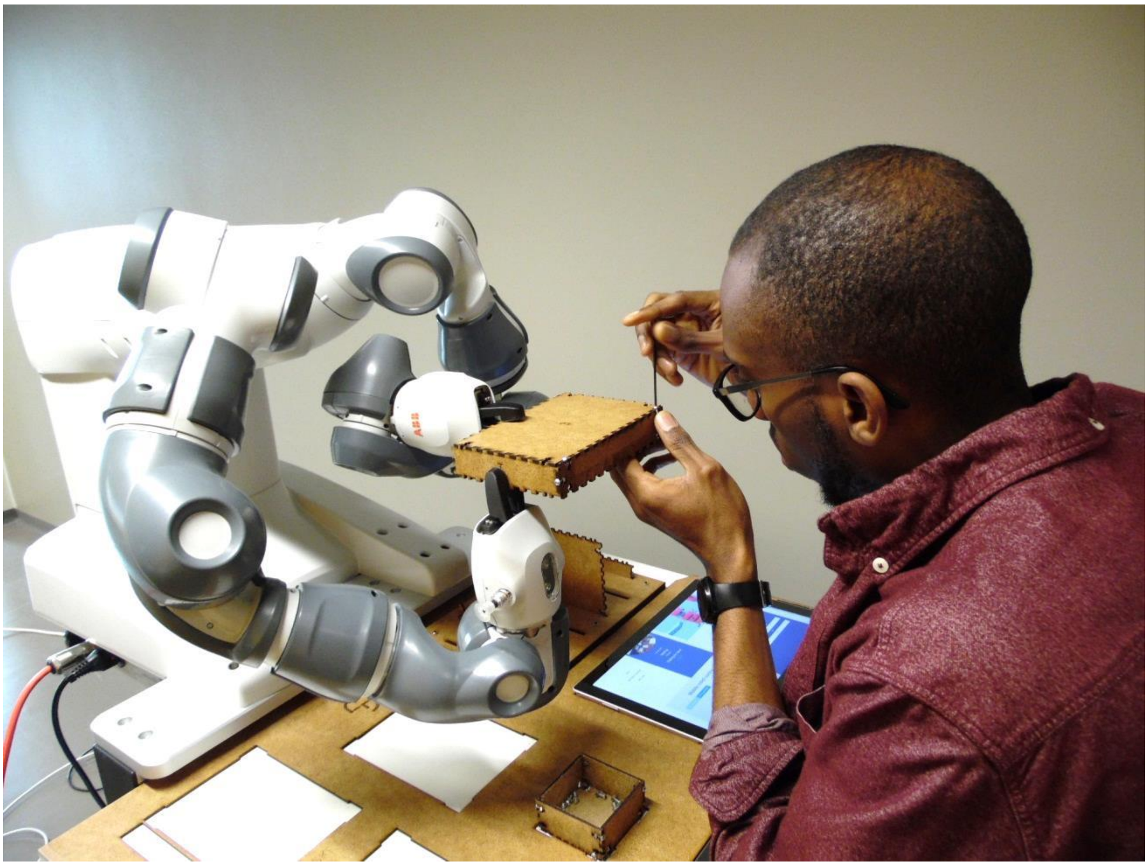


Figure 37. Operator working in the box assembly with YuMi

3.8 Flowchart

Once the proposal has been selected, the virtual reality team, the cobot and the task, it is proceeded to its implementation. The main steps to follow are shown in the form of a flowchart in Figure 38.

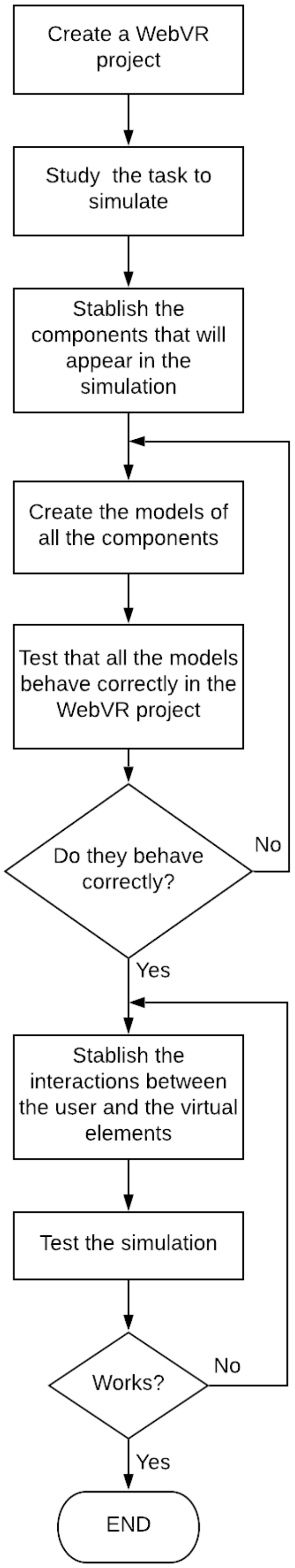


Figure 38. *Flowchart of a WebVR project including virtual model simulation*

3.9 VR Environment

The A-Frame project is opened from Glitch[16], which is a web that allows creating a site from the browser. Glitch provides a URL instantly and is continuously deployed within HTTPS (Hypertext Transfer Protocol Secure). Glitch generates the server needed for the communication between the user accessing the browser and the virtual environment. The interface of the A-Frame's project of this implementation when wearing the VR Headset is shown in the Figure 39. The images correspond to the ones seen by each of the eyes when wearing a Head-Mounted Display (stereoscopic vision).

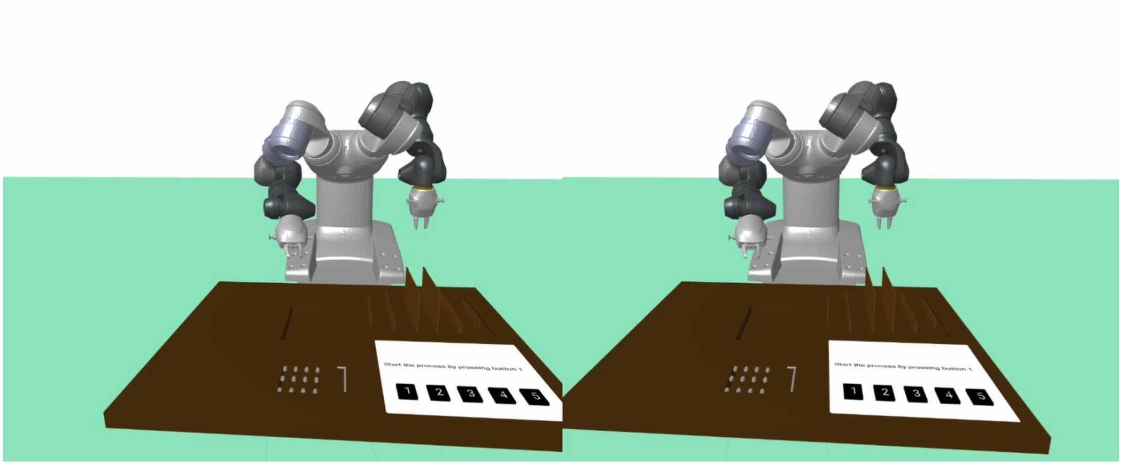


Figure 39. *Stereoscopic vision wearing Oculus Rift headset*

Summing up, for opening the VR application the user has to follow four steps, illustrated in Figure 40.

1) Access a computer with a VR equipment installed
2) Open the A-Frame project in Glitch
3) Open the virtual environment through "Show-Live" mode
4) Wear on the Headset and join the virtual environment

---

[16] https://glitch.com/

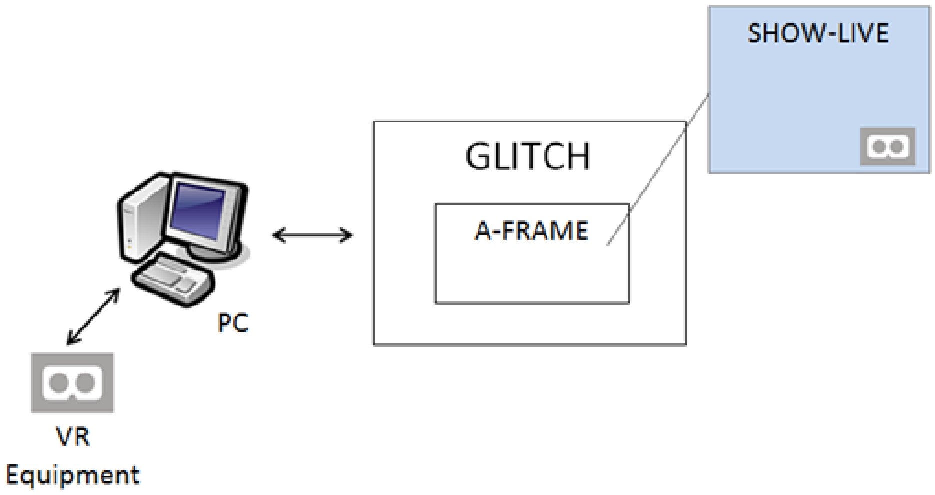


Figure 40. *Communications diagram of the implementation*

The A-Frame Project opened in Glitch is structured in:

6) ‘assets’ which is a space where to upload the files that will be used in the code
7) ‘license’ which contains the copyright and permissions of A-Frame software
8) ‘readme’ which is a space for adding a description of the developed code
9) ‘index’ which is HTML document where the code of the VR experience is developed.

This html document created in the ‘index’ has a DOM hierarchy as depicted in Figure 41, which is formed by the head and the body. The head is composed of the title and the scripts. There are two types of scripts: the ones which contain the address of the libraries imported, mainly the ones of [52]; and the ones which define a function. The body has the elements needed for building the scene: assets that load the uploaded files that will be used in the simulation, code lines to define the background of the scene (floor and sky) and primitives such as box, cylinder or circle, or a combination of them for creating more complex elements.

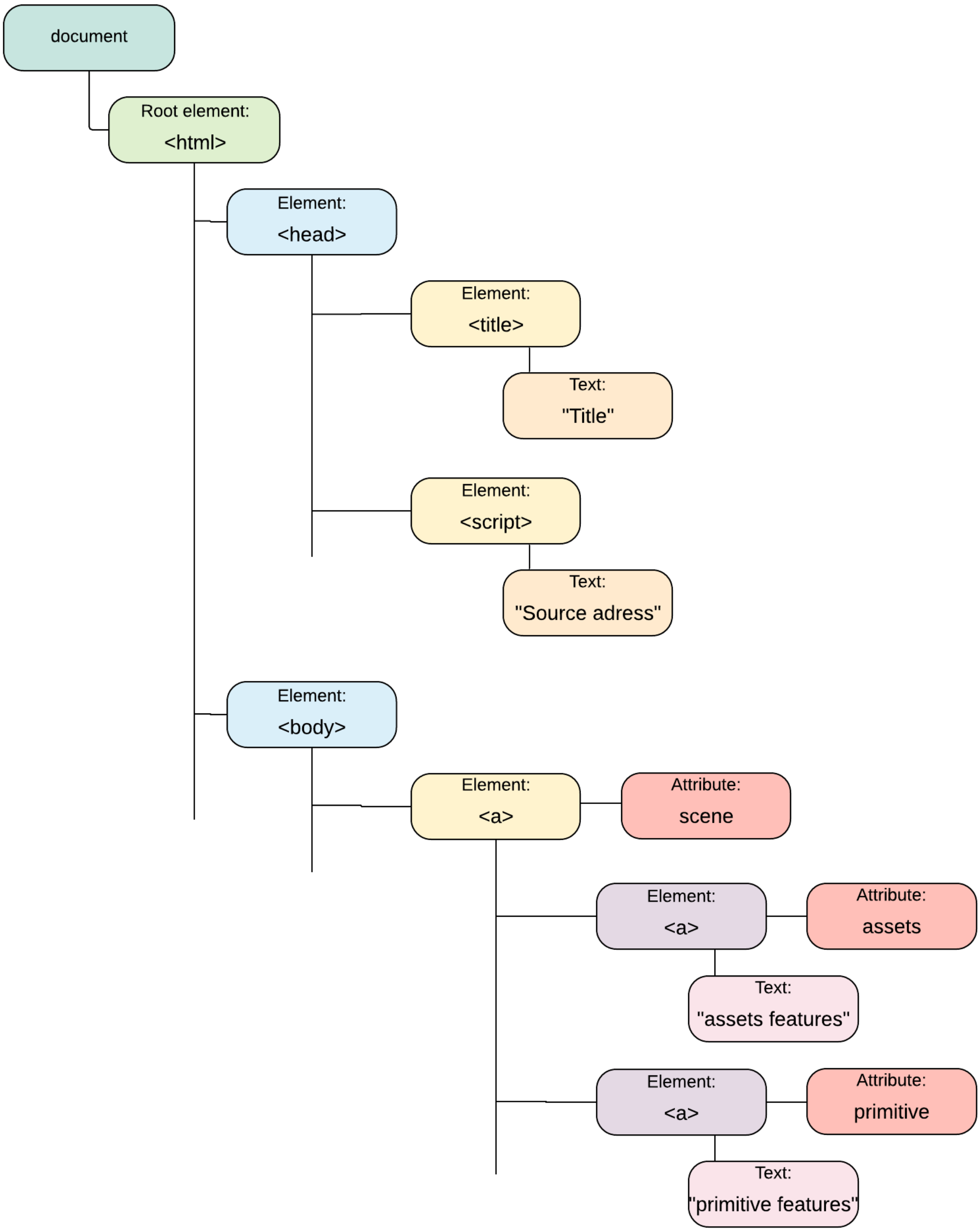


Figure 41. *DOM hierarchy of an html document*

A general overview of the simulation to be performed allows identifying the main components involved in the simulation. In the study case of this implementation there are three main components: virtual model, interface and user, as illustrated in the class diagram of Figure 42.

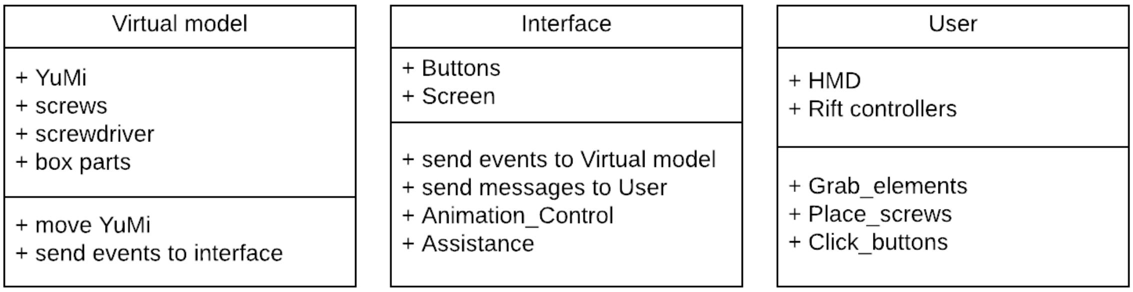


Figure 42. *Class diagram of the implemented simulation*

The virtual model contains all the elements needed for building the box. It moves the YuMi model which is responsible of picking and holding the parts of the box during the assembly process. The interface orchestrates the virtual model and the user by sending events to the virtual model and sending messages to the user through the screen. In addition, it determines the action to be performed by YuMi model based on the clicked button by the user and provides visual assistance to the user for following the process correctly. Finally, the user grabs and places the screws and informs the interface to start the next step by clicking the buttons of the screen.

3.10 Task analysis

The scenario, shown in Figure 43, includes a table which contains all the parts for assembling the box, i.e., box sides, bolts and screws and the interface for communication between the user and the virtual cobot.

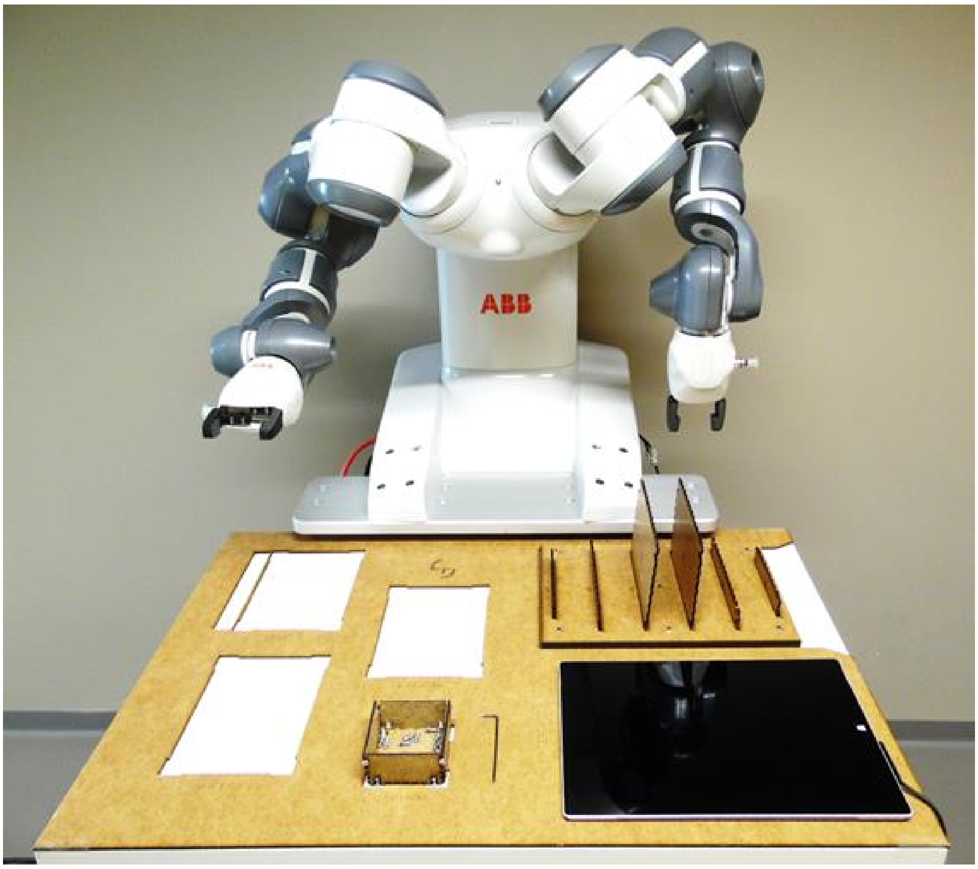


Figure 43. *Scenario of the implementation*

As the training aims to get the user used to working with the cobot, the simulation must be as similar as possible to the reality. When carrying out the study of the task, a series of aspects necessary for its recreation in virtual reality must be taken into account: the sequence of events, the movements of the cobot and its velocity, the elements that are involved in the simulation apart from YuMi and the interactions between YuMi and the operator.

During this process YuMi executes an action composed of a series of movements with which the cobot takes the pieces from the box and places them in a suitable way for the user to place the screws. The action is activated by the user by pushing a button. When YuMi finishes the action the user activates the assistance which provides the user the instructions to place the screws.

3.11 Virtual components

The main element of the simulation is the cobot. Apart from this, at the time of establishing the components that take part in the simulation, a classification has to be made: the ones managed by the cobot and those managed by the user. This step is relevant since it affects the way they are introduced in the virtual environment.

Table 7. *Classification of the components involved in the simulation*

Manipulated by the cobot	**Manipulated by the user**
Parts of the box	Tool to screw
Screws	Screws
Work-table	Buttons

To ensure that the simulation is immersive and the user learns the movements of the cobot, these must be as realistic as possible and, therefore, the virtual model must behave in the same way as the real cobot does. The method used to replicate YuMi's movements is to generate a virtual model that contains all the actions performed during the task. This model must contain all the elements that the cobot uses during its actions, collected in the previous table (Table 7).

Since the elements that the user handles are of simple geometry, their models are created directly in A-Frame. This facilitates the addition of interactions between the user and these elements.

3.11.1 Cobot’s manipulation

The models must be entered in the A-Frame project in an appropriate format to reproduce the actions. This format must be accepted by A-Frame. The most convenient format is glTF (GL Transmission Format) that is used for 3D scenes and models using the JSON standard, which is capable of supporting animations. This glTF file is uploaded to the project in the ‘assets’ space.

The design of the YuMi model has been acquired from the 3D CAD drawings available on the manufacturer’s website [46]. These drawings are provided as a group of STEP files, each file corresponding to each of the links of the robot. On the other hand, it is needed a software for animating the virtual model with accuracy so it seems as realistic as possible. The software selected for adding the actions to the 3D model is Blender[17], as it fulfils the requirements of the proposal: a YuMi model moving as realistic as the original one. Blender is a 3D computer graphics software toolset that allows creating visual effects and interactive 3D applications, among other features. It is free and open source software written in Python and is available for Windows, Mac, and Linux. Since the format of the model that has been downloaded from ABB is not combinable with

---

[17] https://www.blender.org/

Blender, it is necessary to use an intermediary that allows conversion of the format of the model to one that can be imported into Blender. The intermediary used in this implementation is FreeCAD. The STEP files of the robot are imported into this program and the models of the robot's links are connected one to each other in order to assembly the YuMi model. Once the 3D model is assembled it is exported in COLLADA format (.dae) which is compatible with Blender.

The other elements are inserted in Blender in a simpler way. They are designed in a CAD application like SolidWorks or SolidEdge and exported as STL format, which is compatible with Blender. The program used in this implementation is SolidEdge.

The result of the animation process in Blender, explained in next subsection, is a single file containing all the imported elements performing the actions involved in the assembly box. Blender does not have the ability to export glTF files, thus it is needed another intermediary to get the desired format. There are some available glTF converters online that make possible to export the file with the animations as a .glb extension. The converters developed by blackthread.io[18], which is the one used in this thesis work, or modelconverter.com[19] are some of the available options online. The file imported to the converter requires a compatible format for the converter and for blender exporter, and must support animations. For the described use case, the chosen format is FBX.

The programs used to introduce the model in A-Frame in glTF format are shown in Figure 44, as well as the formats used to import and export files in each of these programs.

---

[18] https://blackthread.io/gltf-converter/

[19] https://modelconverter.com/

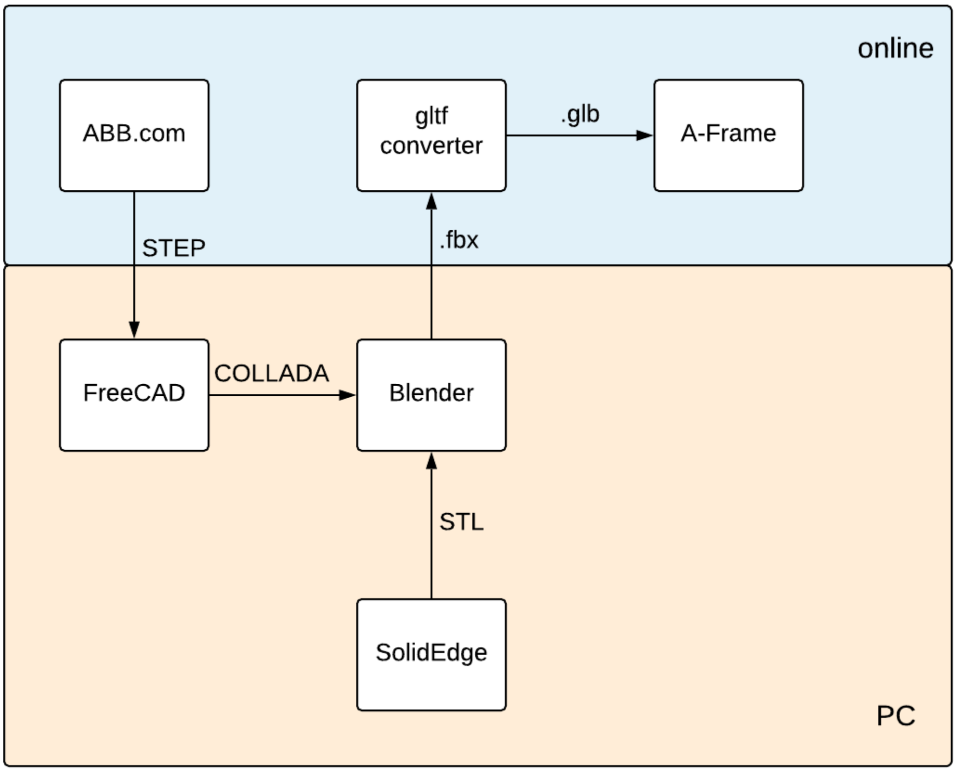


Figure 44. *Format transformations for inserting virtual models in A-Frame*

Animation in Blender is based on a set of key frames, which is a marker of time which stores the value of a property. In this implementation Blender has been configured so that 1 second equals to 30 frames in order to have high accuracy when creating the movements. The inserted key frames indicate the position of the links through location (X, Y, Z) and rotation (W, X, Y, Z). The links have a rotational movement relative to the link to which they are attached. Figure 45 depicts the 7 axes of rotation of one of the two arms of the cobot. The fingers of the end-effector move linearly for closing and opening.

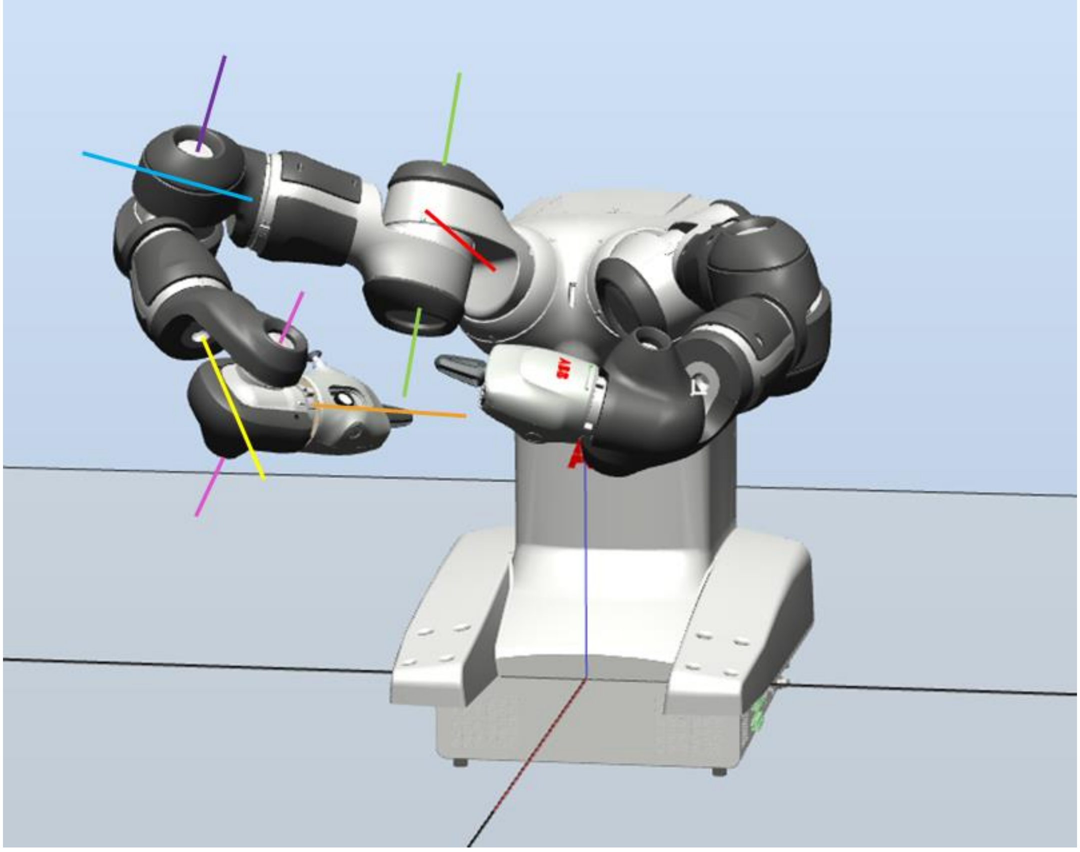


Figure 45. *Rotation axis of YuMi's arm*

In the previous section "action" has been defined as the set of movements that the YuMi performs in each of the iterations. In the real world the YuMi maintains the final position of the action until the user has placed all the screws. However, in the virtual environment, when an action ends it returns to the initial position of said action. In order to avoid this, when finishing an action YuMi model jumps to an "idle" action which is continuously repeated in which the YuMi maintains the final position of the performed action until the operator end his/her task. Therefore, there have to be created two actions for each iteration: the first corresponds to the set of movements performed by the real robot and the second is an "idle" action.

Figure 46 illustrates the scenario created in Blender and that is the basis on which the actions are going to be created. The colour of the table and box's elements is changed by setting a RGB combination to the 'material' property. The YuMi's material is imported with the YuMi model in the COLLADA file.

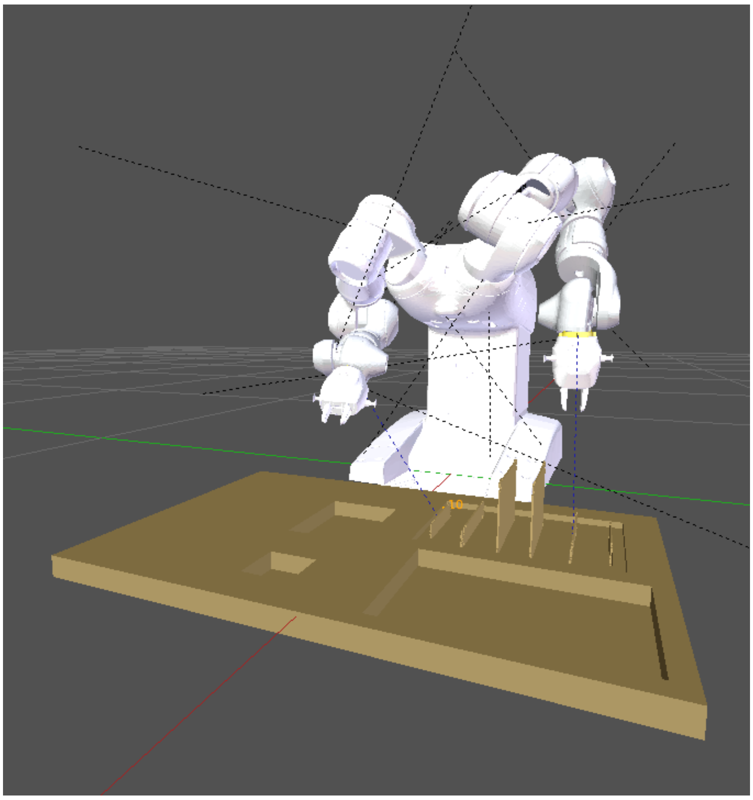


Figure 46. Scenario of the implementation in Blender

To facilitate the way in which this sequence of movements is visualized in virtual reality, it is important that the actions are applied to a single element. In this way, by calling just one action all the elements (YuMi's links and the parts of the box) will move at the same time and in a coordinated way. This single element that will cover all the elements is obtained by creating an "armature" of the robot. The "armature" is like a skeleton formed by a set of bones attached to each of the links, so when moving the bone the corresponding link moves in the same way. As shown in Figure 47 the bone has a local coordinate system defined. When defining the movements of the YuMi model, the key frames of the bones that are attached are saved instead of the positions of the links themselves.

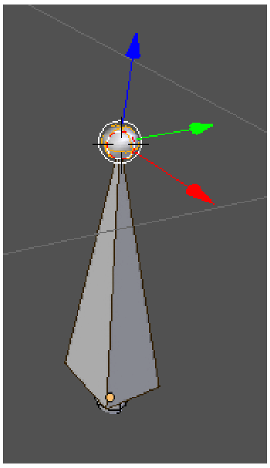
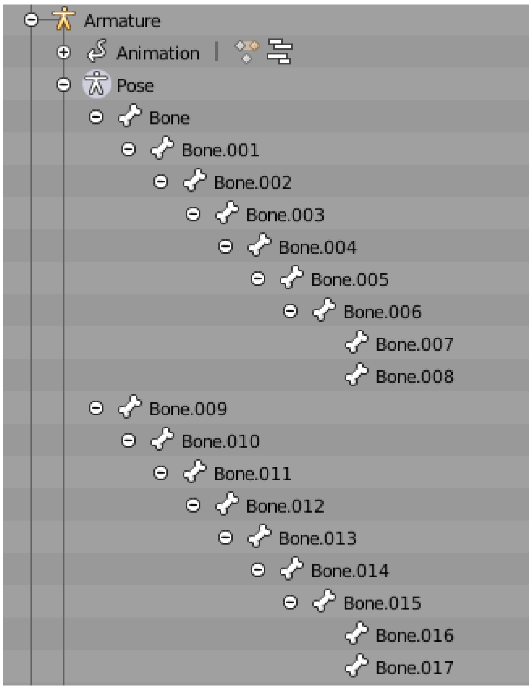


Figure 47. *Example of bone that is attached to the virtual model*

The origin of the bone is placed on its corresponding link, at the point of contact with the following link and with an orientation such that the axis ‘y’ of the bone coincides with the axis of rotation of the link. So the bone rotates in the same way as the link does and, since both link and bone are connected, the link will also move.

To create the existing connection between the links, a parent-child hierarchy must be generated. In this way, by moving one of the links, all the parent links related to it as children will follow him in the movement. That is, it will be possible to move both the end individually and in conjunction with the rest of the arm. The hierarchy of both arms is illustrated in Figure 48. For example, when moving the bone “Bone.005” bones 6, 7 and 8 will follow this movement and bones from 0 to 4 will stay quiet. When moving all of these bones at the same time, it will keep the physical attachment that the real links of the arm have between them, while having a relative movement.



- Armature
  - Animation
  - Pose
    - Bone
      - Bone.001
        - Bone.002
          - Bone.003
            - Bone.004
              - Bone.005
                - Bone.006
                  - Bone.007
                  - Bone.008
    - Bone.009
      - Bone.010
        - Bone.011
          - Bone.012
            - Bone.013
              - Bone.014
                - Bone.015
                  - Bone.016
                  - Bone.017

Figure 48. *Hierarchy of the armature in Blender*

Therefore, the action will be applied to the armature and the inserted key frames will correspond to the position of each of the bones. For creating the paths of the actions, when performing the study of the task, some reference positions and the time that passes from one position to another must be taken. Having in mind the relation between time and frames (1 second = 30 frames), the key frames are inserted in the frame-line of Blender. The model of YuMi will move from one key frame to the next in the defined time. In case the model does not move correctly or accuracy enough, more key frames must be inserted. It is of great importance that the model moves loyally to the real cobot to obtain a good immersion of the user in the virtual training and that said training allows him to know the way in which the cobot will move when it is incorporated into the real production line. In the same way it is important to have in mind the different velocities of the links when deciding which of the positions will be saved as key frames, as the pick and place actions have slower motion.

As abovementioned, all the elements must be incorporated in the armature to get all the sequence of movements in one single action; so the parts of the box must be attached to bones which belong to the armature but with no parent-child hierarchy. The pick and place actions of these box's parts is done by adding "child of" constrains. They act as temporary parent-child hierarchy linked to the bone of the robot's end effector. Figure 49.a. shows the pieces of the box with no parent constrain while Figure 49.b. shows one of the pieces linked to the hand of the robot (the temporary parent constrain has already started).

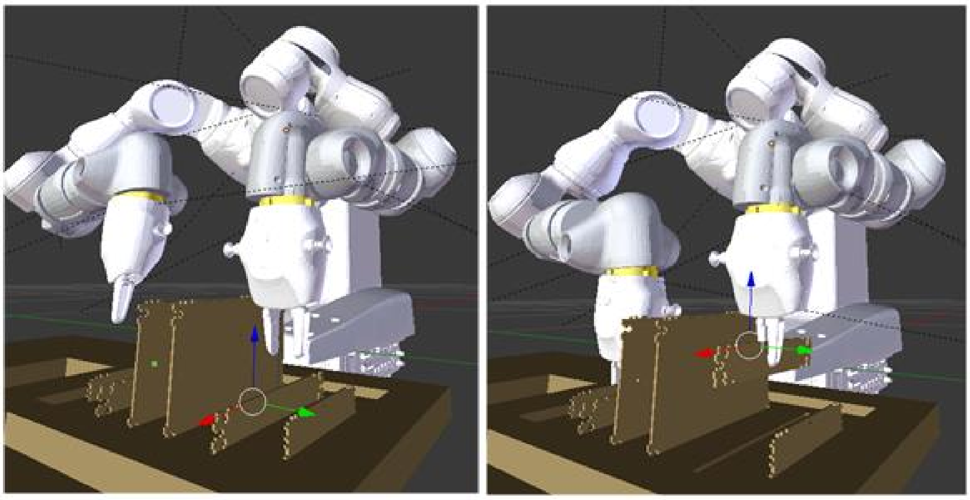


Figure 49. Pieces of the box a) without and b) with parent constrain

3.11.2 User’s manipulation

The manipulated elements must be defined differently from the YuMi model in the html code to provide interaction with the user's Rift controllers. The most appropriate for this type of interaction is that the elements are created directly in A-Frame. The shape of these elements is made of primitives such as box, circle, cylinder or plane, or a combination of them. The user can grab these elements whenever they have the attribute “grabbable” and click them in case they have the attribute “clickable”. The interactions with these elements are explained in next section.

3.12 Interactions

The user's interactions with the virtual environment are done through the Rift controllers. Although all the buttons of the controls can be used in the A-Frame projects, for this application only the buttons indicated in Figure 50 are used: trigger of both controls, A and B.

Figure 50. Used buttons of Rift controllers during the simulation

The function that each of these buttons activates is explained in the table below. These functions are responsible for performing the user's interactions with the virtual environment and also the interactions of the virtual environment with the user, made through the exchange of events through the interface.

Among these functions is collected the assistance process available to the user to facilitate the learning process of the task. This assistance can be divided into three visual aids. The first is to give the user a series of instructions written on the screen. These instructions indicate to the user what the step that is currently running is and what the next step will be for him/her to do. The second is the signalling of the holes in which the screws must be placed so that the user gets used to placing the screws always in the same order and so, in the long run, it is faster performing the task. This help is activated in each of the iterations of the process once YuMi has finished its task. It consists of a yellow dot on the hole in which the screw must be placed, which facilitates the user to detect the next hole to be filled. Once the screw has been placed its corresponding point disappears and the next appears. Finally, this assistance facilitates the user the task of taking the screws since being very small elements the controllers can have problems detecting the collision with them. For this the screw to be used at each moment is illuminated and raised with respect to the plane in which the other screws are located so that it is easily identified by the user.

Table 8 shows the main functions that are activated by the use of Rift controllers by the user.

Table 8. *Functions activated with Rift controllers in the simulation*

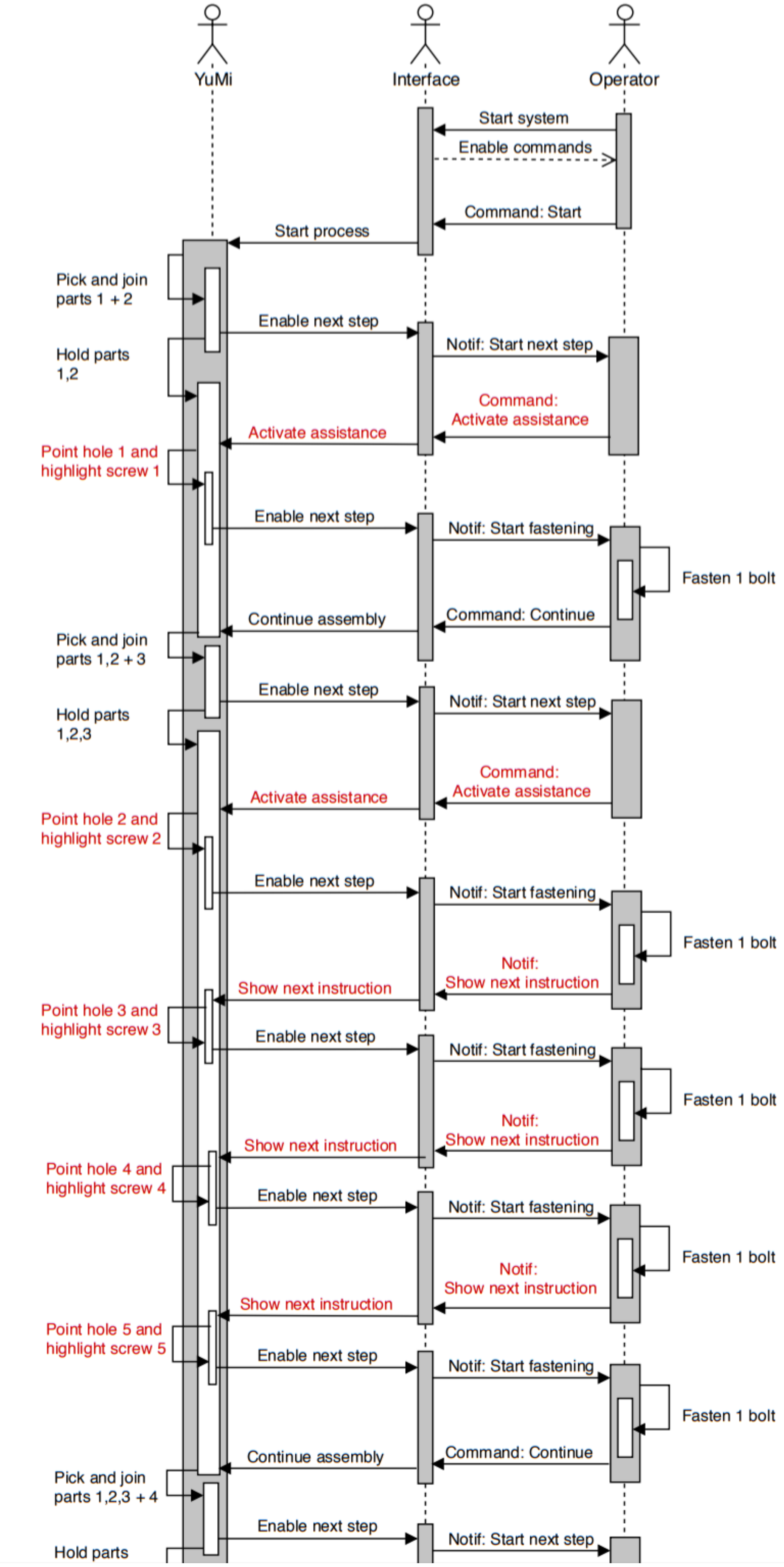
Function	Button	Description
Grab	trigger	User can grab “grabbable” elements by getting close to it and pressing the trigger button. The element keeps grabbed while the trigger button is pressed. For dropping the element the user must release the button.

Click	trigger	User can select “clickable” elements by pointing the virtual buttons with the laser control and pressing the trigger button of the Rift controller. When selecting one of the buttons, the code jumps to ‘animation_control’ function where the interface starts a specific based on the selected button. When YuMi finishes the action it starts an ‘idle’ action that holds the box’s parts on the correct position for the user to place the screws. It keeps this position until the user informs to start the next iteration. At the same time the instructions of the screen change to explain the user the next step.
Assistance	B	By pressing button B the code enters the ‘Assistance’ function which activates the assistance of the corresponding step: visual help for the user to know in which hole he/she must place the screw. When activating the assistance the instructions on the screen change to show the next step.
Place_screws	A	By pressing button A the code enters function ‘Place_screws’ which adjusts the screw in the hole. When all the screws of the corresponding iteration have been placed no more helping dots appear and the instructions on the screen inform the user that there are no more screws left to place by indicating to push the button corresponding to the next step.

3.13 Final VR training

The sequence diagram of the final VR training is shown in Figure 51, highlighting the differences between the real process and the virtual one with red font of text. The process starts with the robot in the initial position and the instructions of the screen inform the user to start the simulation by pressing the start button. When the button is pressed the instruction informs the user to activate the assistance when YuMi stops. YuMi picks two of the box pieces and holds them until it receives the next command. The user activates the assistance by pressing button A, which explains the user how to place the screw, points the hole to place the screw and facilitates the user the screw to be used.

After the user has placed the screw the screen shows the next instructions to be followed. In the first step only one screw is placed; in the second, four; in the third, three, and in the fourth, four. When the user has placed a screw, if the step requires placing more screws the assistance points the next hole and the next screw. If all the screws of the step have been placed the instructions inform the user to continue by clicking the next button. After all the screws have been placed the user activates the last step, in which he/she does not participate. YuMi leaves the box on the table and the places the last piece on top of the box. YuMi returns to its initial position and when it stops moving the simulation is finished.


YuMi
Interface
Operator
Start system
Enable commands
Command: Start
Start process
Pick and join parts 1 + 2
Enable next step
Notif: Start next step
Hold parts 1,2
Command: Activate assistance
Activate assistance
Point hole 1 and highlight screw 1
Enable next step
Notif: Start fastening
Fasten 1 bolt
Continue assembly
Command: Continue
Pick and join parts 1,2 + 3
Enable next step
Notif: Start next step
Hold parts 1,2,3
Command: Activate assistance
Activate assistance
Point hole 2 and highlight screw 2
Enable next step
Notif: Start fastening
Fasten 1 bolt
Notif: Show next instruction
Show next instruction
Point hole 3 and highlight screw 3
Enable next step
Notif: Start fastening
Fasten 1 bolt
Notif: Show next instruction
Show next instruction
Point hole 4 and highlight screw 4
Enable next step
Notif: Start fastening
Fasten 1 bolt
Notif: Show next instruction
Show next instruction
Point hole 5 and highlight screw 5
Enable next step
Notif: Start fastening
Fasten 1 bolt
Continue assembly
Command: Continue
Pick and join parts 1,2,3 + 4
Enable next step
Notif: Start next step
Hold parts

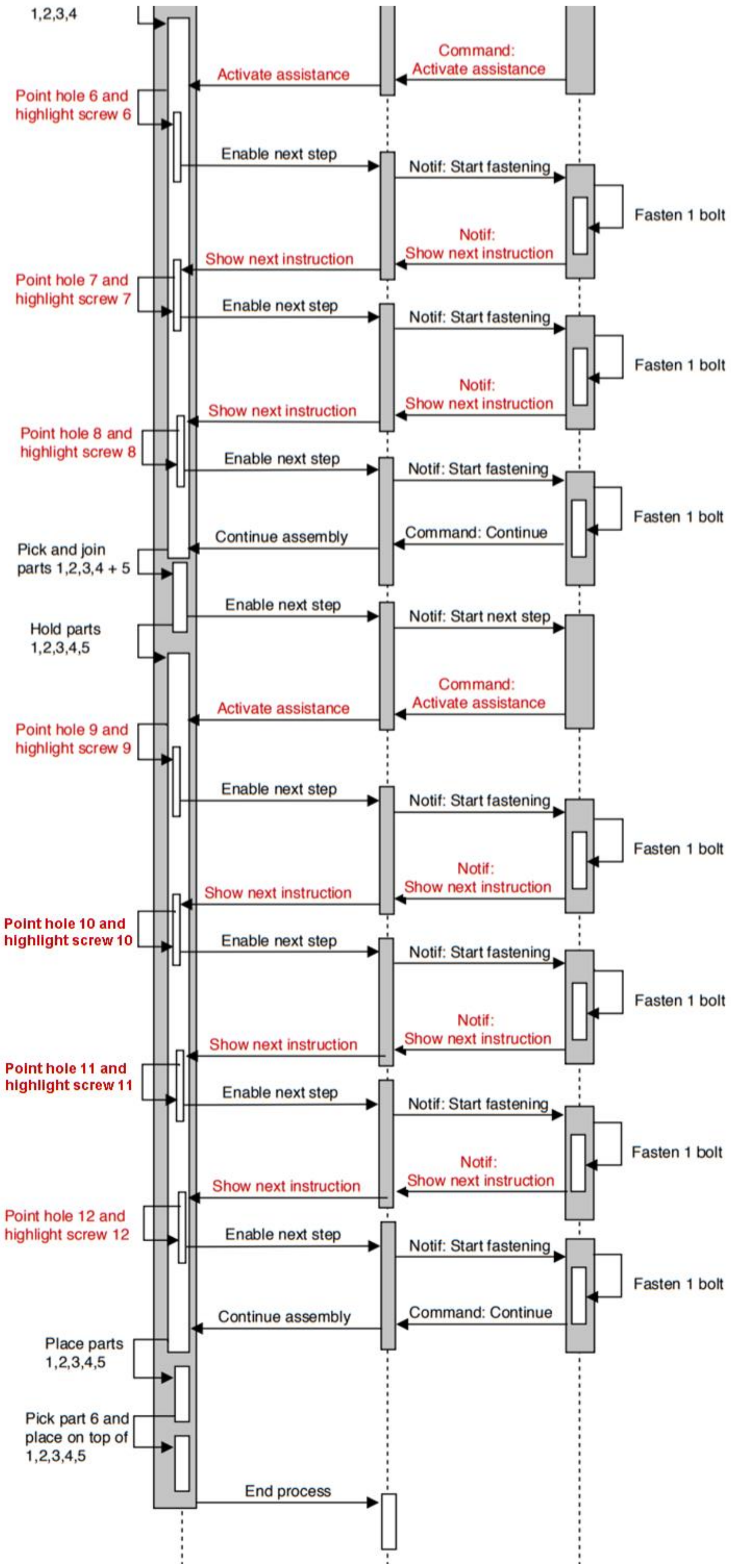
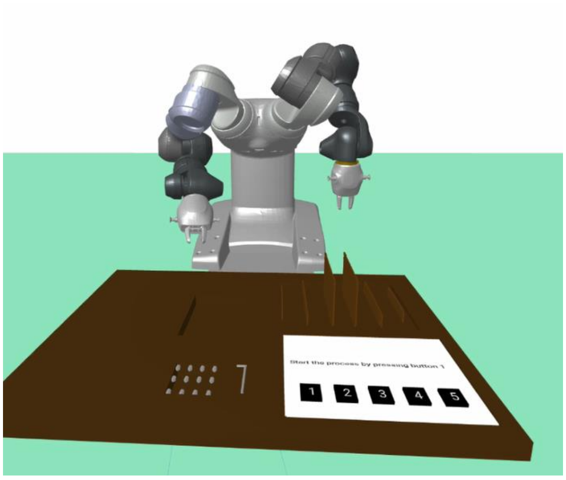
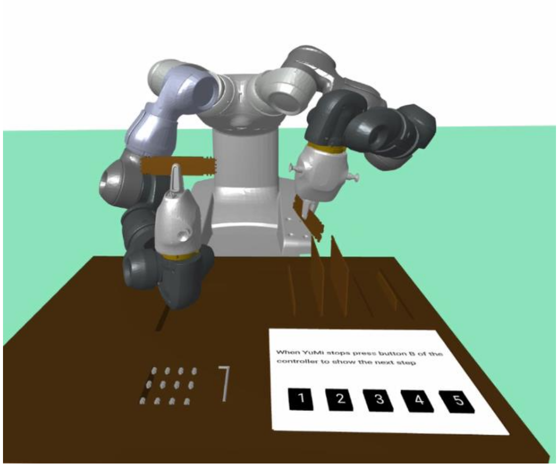
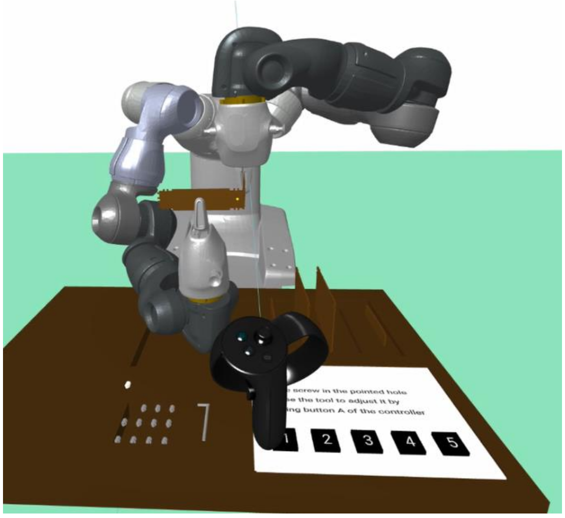
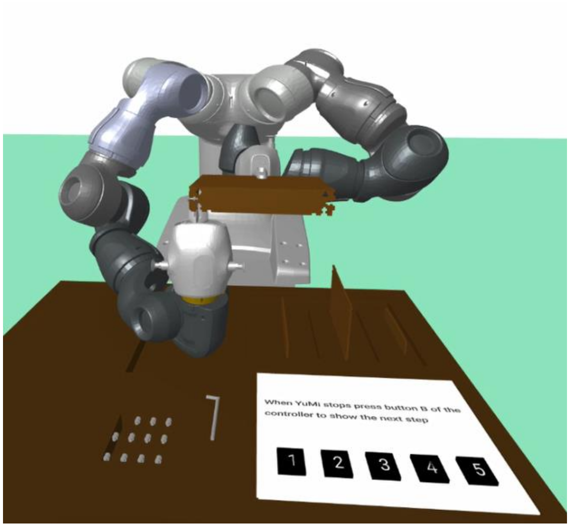
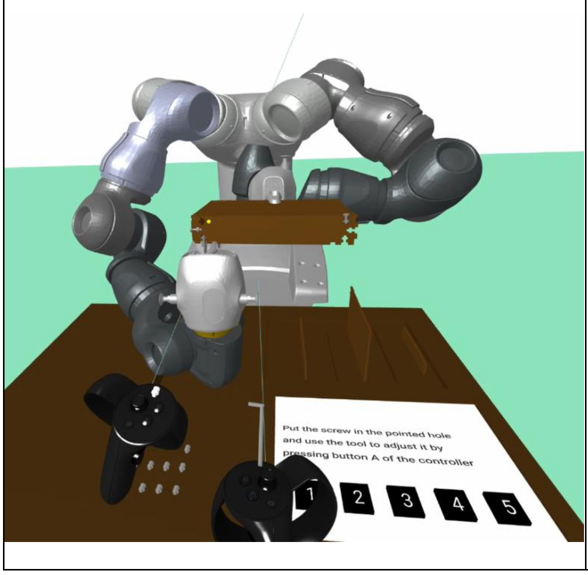
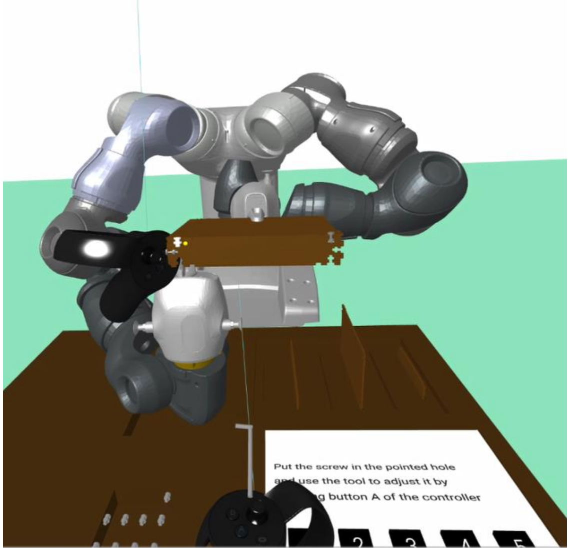
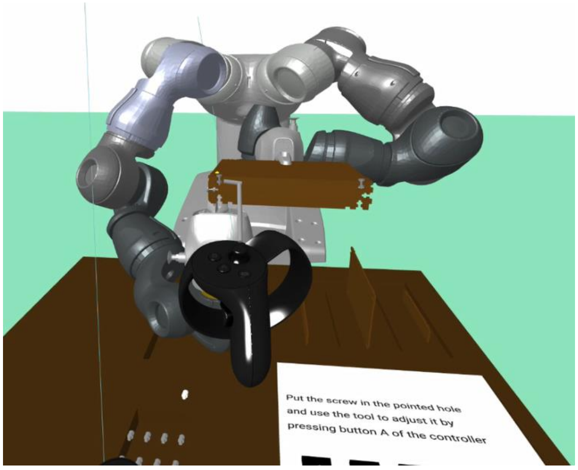


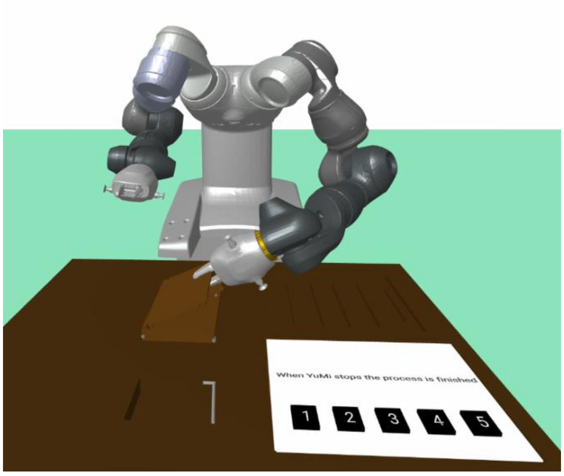
Figure 51. *Sequence diagram of the implemented VR training*

The resulting simulation is illustrated through the images of Table 9, where it can be seen some of the process of the sequence diagram.

Table 9. *Simulation steps visually displayed*





Following the order of the cells from left to right and up to down, the first image shows the default scene that appears when the user starts the animation. The virtual model is waiting for the user to push the start button, as it is indicated in the instructions of the screen. Second, YuMi picks the first two parts of the box to hold them in the proper position for the user to work. Third, the user pushes B button to activate the assistance, so the instructions change, the next screw to place is highlighted and the hole where to place the screw is pointed with a yellow dot. This process is repeated in each iteration as shown in fourth image and fifth images where the transition of the scene when activating the assistance is shown. Then the user grabs a screw and places it in the pointed hole (sixth image). In image seven the user adjusts the screw by pressing button A, and the yellow point changes position to the next hole and the next screw to grab is highlighted. Finally, when all the screws have been located YuMi places the box on the table and closes it by placing onto the box the top part.

4. TESTS AND RESULTS

This chapter aims to verify the validity of the implemented application through a test. When designing the test some limitations were found.

The group of people with whom to carry out the tests must represent the type of users that would use this training method. The profile sought is therefore that of workers who normally work in assembly tasks of industrial plants and who, without having experience working with cobots, could have to work on collaborative tasks in the future.

The tests carried out with this type of user could provide information not only on the effectiveness of the application, but also on its capacity to reduce the adaptation time required by the operator in comparison with the training methods that are currently used and its influence in the productivity of the production line.

These experiments require two test groups working in an industrial plant in collaborative tasks. Each of the groups would carry out different trainings. After finishing the training, the time necessary to learn the task and perform it in a comfortable and natural way along with the cobot in each of the operators of both groups would be measured. The differences found in the productivity of the plant after comparing a work group that performs the training within the plant and another group that performs the training outside the job proposed in this thesis would also be measured.

However, having no access to this type of group of tests, the tests have been carried out with a group of users that collect the necessary characteristics to prove that the proposed training allows the user to gain confidence with the cobot and learn the steps and interactions of the assembly process.

This testing group is formed by 10 degree and master engineering students who have never worked with cobots but who know the characteristics of the processes of industrial plants. There were male and female users and their age range was between 21 and 26 years. The participants performed the simulation where they assembled the abovementioned box. Each participant was tested individually, and after the experiment, they answered a questionnaire. This questionnaire and the results of each participant are detailed in Appendix A.

Table 10 shows the information that is intended to be obtained from the answers of the users to some of the questions to which they responded evaluating the range questions from 1 to 5.

Table 10. *Analysis of the questionnaire*

	Question	Goal	Possible answers		Acceptable range
1	How comfortable do you feel relating to working with YuMi? (Before the simulation)	Quantifying the level of trust and the biases they have relating to working with cobots	1-2	not comfortable and not confidence	1-3
			3	not afraid of the cobot but not comfortable	
			4-5	comfortable and interested	
2	How comfortable did you feel during the simulation?	Evaluating how the participant feels during the execution of the simulation	1-2	overwhelmed by the feeling of being somewhere else	4-5
			3	comfortable in the virtual world but confused by the task to be executed	
			4-5	comfortable in the environment and with the task	
3	How realistic did you find the simulation?	Detecting the weaknesses of the training so that it is possible to improve	1-2	nothing in the task reminded of a real process	3-5
			3	the graphics were not realistic and the interactions were difficult to perform	
			4	the graphics were not realistic but the interactions seemed like reality	
			5	interactions and graphics were realistic and felt like really grabbing objects	
4	How easy was to follow the process?	Detecting weaknesses in the as-	1-2	confusing instructions	4-5

		sistance process so that it is possible to improve	3	Could follow the process but doubting about the steps	
			4-5	It was intuitive and the instructions were clear	
5	Would you be able to reproduce the process?	Evaluating if the participant gets the necessary skills for reproducing the process	1-2	Cannot repeat the process	4-5
			3	Can repeat the process but having troubles to screw	
			4-5	Can reproduce the whole process	
6	How comfortable do you feel relating to working with YuMi? (After the simulation)	Quantifying the level of trust they have relating to working with cobots after the simulation	1-2	not comfortable and not confidence	4-5
			3	not afraid of the cobot but not comfortable	
			4-5	comfortable and interested	
7	Did you learn how the YuMi interacts during the process?	Evaluating if the participant becomes familiar with the movements of the cobot and the interactions with it	1-2	did not realize how the cobot was moving	4-5
			3	learnt the movements but not features like the velocity	
			4-5	learnt the movements and interaction with the cobot as well as its main features like the velocity	

Figure 52 illustrates the average results of the participants for each of the questions presented in the table above.

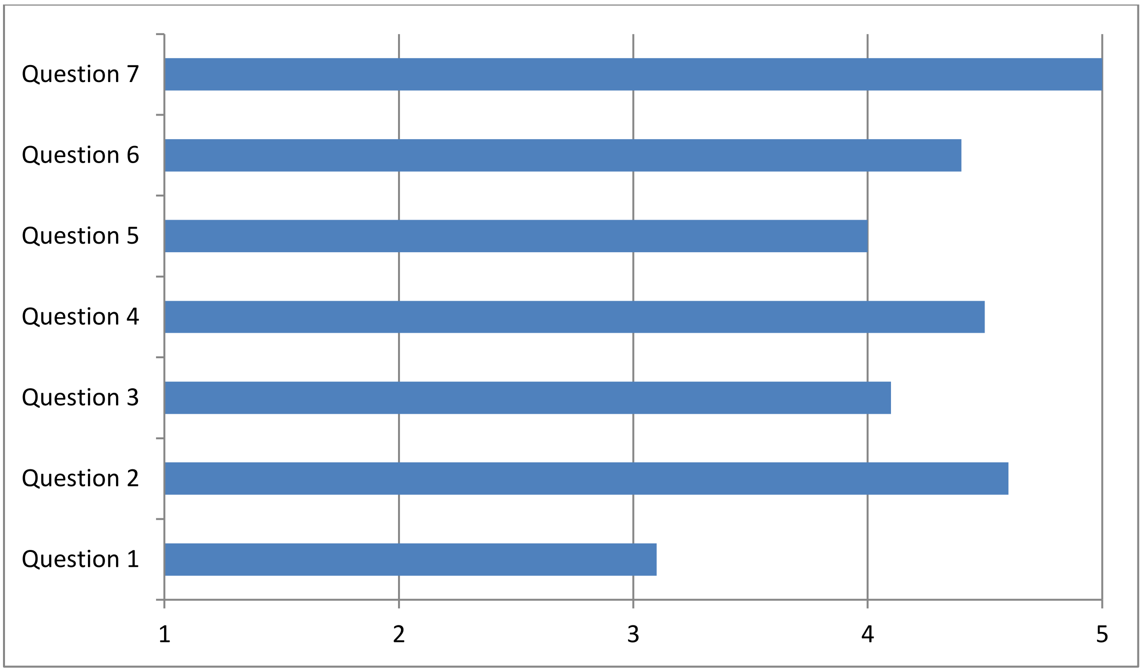


Figure 52. *Bar diagram of the results of the test*

Next, the results obtained from the test are analysed comparing them with the values of the acceptable range.

The answers of the users to Question 1 vary from 1 to 5, with an average of 3.1. The acceptable range for this case of study is between 1 and 3, since in order to obtain valid results it is necessary that the users can experience an improvement in the trust towards the cobots. 50% of the participants responded within the acceptable range. Those whose response exceeded 3 claimed not to be prejudiced or distrustful of the cobot despite having no experience working with cobots.

Since the simulation seeks to make the operator feel comfortable during his/her training period to favour the acceptance of collaborative tasks the range of suitable answers to Question 2 is between 4 and 5. The average value is 4.6 so it is within the range. Those few participants evaluating with lower marks said they felt more overwhelmed by the fact of working with virtual reality for the first time than by the content of the simulation, causing them to lose their focus on the task.

Question 3 relates to the realism of the simulation. The average result, 4.3, is within the acceptable range. Most of the criticisms focused on the difficulties of grabbing the screws since none of the users was used to working with the virtual reality equipment and, being a small element requires more precision for its manipulation. They also found that the graphics were not as realistic as they expected but that both the movements of the cobot and its speed and the interactions with it were realistically executed. Those evaluating with the highest mark thought that, even though the cobot was visibly a virtual element its structure as well as its movements were the same as those of the real model and that, therefore, it was realistic enough for the realization of the task.

Question 4 was answered with an average of 4.5. Participants found the process easy to keep in mind that they had never done this process and that until the moment of performing the simulation they did not know any of their steps or elements. The element most valued in this aspect was visual assistance. The help to take the screws helped them with the problems of grip already mentioned, the yellow points that indicated the place in which they had to put the screws in each moment and the instructions of the screen told them to follow the process from the beginning to the end. However, several users had problems not remembering which button corresponded to each of the functions and sometimes pressed the button several times followed by accident, which affected the performance of the simulation. Once this error is detected, the simulation could be improved by adding messages displayed to the user when pressing the buttons to ensure that it is the button that the user wants to press before executing the action linked to said button.

Question 5 had an acceptable range of 4 to 5 and the average result is 4. Many of the users were able to reproduce the entire process in the real environment; however, some of them, in spite of showing to know the process and the steps to carry out had problems with the process of screwing.

Question 6 is the same as Question 1 but done once the simulation has been completed. This is done with the purpose of detecting an increase in the confidence of the users when it comes to getting to work with real cobot. Fortunately, all users said they felt more confident after knowing the movements of the cobot and the steps of the process, except those who answered with the highest grade and therefore have not experienced differences between the two questions. The average answer is 4.4 versus 3.1 of Question 1.

Question 7 is the one that has obtained the best results. All participants responded with the maximum note that after the realization of the simulation they learned not only the steps of the process to be carried out but also the interactions that take place between the operator and the cobot, the movements and the speed with which the cobot and the proximity to the cobot with which they have to work in a collaborative task. Many of the users indicated that the fact that the virtual environment focuses only on the task to be carried out helped to maintain the concentration in this task.

5. CONCLUSIONS

Users who work in an industrial scenario do not necessarily have a previous formation related to robots. This lack of experience working with robots can generate unease especially when facing collaborative tasks with them. This unease lengthens the user's training period since it requires more time to accommodate collaborative tasks. This longer period of time translates into lower productivity because during this time the operator does not work at maximum performance. This problem was introduced in the first chapter of this document. This thesis work has been trying to solve this problem by searching a training method for operators that get to answer the question presented in the introduction: “How to train operators to work with collaborative robots without decreasing productivity during the period of adaptation?”

In order to answer the question this thesis work includes a research of the background in the state of the art that allows knowing the methods performed up to date and the trends in development process. Based on the existing proposals for operator training methods, a new solution is sought that allows bypass the problems and limitations detected in said proposals.

The solution presented in this thesis work incorporates virtual reality technology. The training consists of an immersive virtual reality simulation rendered in WebVR that allows the user to perform the training and adaptation period outside the production line and without affecting productivity.

Its implementation consists on the assembly of a box in collaboration with YuMi cobot from ABB. The effectiveness was tested with a group of participants who lacked experience with cobots. The results show that the operators feel more confident when working on a collaborative task with cobot after doing the training. In addition, this training method allows them to follow and learn the process and become familiar with the movements and interactions with the cobot.

It is concluded therefore that this proposal is valid for solving this problem since it answers the exposed research question.

A more detailed evaluation has to be done in future work, by using a large number of participants with more suitable characteristics, as explained in Chapter 4, and performing different tasks in the virtual training. A second phase of training should be carried out in which the virtual environment of the simulation does not only focus on the task to be performed, but also introduces the user in an industrial environment so that the operator becomes familiar with the movements and noises which will be surrounded when entering the production line.

One way to improve the training of users is to incorporate the readings of their emotions, so that the training can be adapted to the needs of each user based on their mental state in each moment. There are some headsets like the one created by Emotiv[20] which allow to measure the signals emitted by the brain in order to evaluate the mental state of the user by analysing the lectures of emotions like stress, engagement and focus, among other. It can be achieved through the use of helmets that incorporate virtual reality and reading of emotions [53].

---

[20] https://www.emotiv.com/

REFERENCES

[1] L. Amezua Hormaza, W. M. Mohammed, B. Ramis Ferrer, R. Bejarano, and J. L. Martinez Lastra, ‘On-line Training and Monitoring of Robot Tasks through Virtual Reality’, Jul. 2019.

[2] H. Hedelin, *Design and evaluation of a user interface for a WebVR TV platform developed with A-Frame*. 2017.

[3] A. M. Djuric, R. J. Urbanic, and J. L. Rickli, ‘A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems’, *SAE Int. J. Mater. Manuf.*, vol. 9, no. 2, pp. 457–464, 2016.

[4] A. Freedy, E. DeVisser, G. Weltman, and N. Coeyman, ‘Measurement of trust in human-robot collaboration’, in *2007 International Symposium on Collaborative Technologies and Systems*, Orlando, FL, 2007, pp. 106–114.

[5] A. Rojko, ‘Industry 4.0 Concept: Background and Overview’, *Int. J. Interact. Mob. Technol. IJIM*, vol. 11, no. 5, pp. 77–90, Jul. 2017.

[6] P. Milgram and F. Kishino, ‘A Taxonomy of Mixed Reality Visual Displays’, *IEICE Trans Inf. Syst.*, vol. E77-D, no. 12, pp. 1321–1329, Dec. 1994.

[7] M. Fiorentino, R. de Amicis, G. Monno, and A. Stork, ‘Spacedesign: A Mixed Reality Workspace for Aesthetic Industrial Design’, in *ISMAR*, 2002.

[8] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, ‘Recent advances in augmented reality’, *IEEE Comput. Graph. Appl.*, vol. 21, no. 6, pp. 34–47, Nov. 2001.

[9] O. Sand, S. Büttner, V. Paelke, and C. Röcker, ‘smARt.Assembly – Projection-Based Augmented Reality for Supporting Assembly Workers’, in *Virtual, Augmented and Mixed Reality*, vol. 9740, S. Lackey and R. Shumaker, Eds. Cham: Springer International Publishing, 2016, pp. 643–652.

[10] A. Katiyar, K. Kalra, and C. Garg, ‘Marker Based Augmented Reality’, *Adv. Comput. Sci. Inf. Technol.*, vol. 2, no. 5, p. 5, 2015.

[11] H. Regenbrecht, G. Baratoff, and W. Wilke, ‘Augmented Reality Projects in Automotive and Aerospace Industry’, 2005.

[12] V. Gay-Bellile, S. Bourgeois, M. Tamaazousti, S. N. Collette, and S. Knodel, ‘A mobile markerless Augmented Reality system for the automotive field’, 2012.

[13] D. W. Seo and J. Y. Lee, ‘Direct hand touchable interactions in augmented reality environments for natural and intuitive user experiences’, *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3784–3793, Jul. 2013.

[14] G. Burdea, P. Richard, and P. Coiffet, ‘Multimodal Virtual Reality: Input-Output Devices, System Integration, and Human Factors’, *Int. J. Hum.-Comput. Interact.*, vol. 8, p. 5, Jan. 1996.

[15] R. A. Earnshaw, *Virtual Reality Systems*. Academic Press, 2014.

[16] H. E. Neely, R. S. Belvin, J. R. Fox, and M. J. Daily, ‘Multimodal interaction techniques for situational awareness and command of robotic combat entities’, in *2004 IEEE Aerospace Conference Proceedings (IEEE Cat. No.04TH8720)*, 2004, vol. 5, pp. 3297-3305 Vol.5.

[17] R. Belleman, B. Stolk, and R. De Vries, ‘Immersive Virtual Reality on commodity hardware’, May 2001.

[18] E. van Wyk and R. de Villiers, ‘Virtual reality training applications for the mining industry’, in *Proceedings of the 6th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa - AFRIGRAPH ’09*, Pretoria, South Africa, 2009, p. 53.

[19] J. Lindblom and B. Alenljung, ‘Socially Embodied Human-Robot Interaction: Addressing Human Emotions with Theories of Embodied Cognition’, *Handb. Res. Synth. Hum. Emot. Intell. Syst. Robot.*, pp. 169–190, 2015.

[20] J. M. Haight, ‘Automated Control Systems Do They Reduce Human Error And Incidents’, 2007.

[21] M. Goodrich and A. Schultz, ‘Human-Robot Interaction: A Survey’, *Found. Trends Hum.-Comput. Interact.*, vol. 1, pp. 203–275, Jan. 2007.

[22] *Automation and human performance: Theory and applications*. Hillsdale, NJ, US: Lawrence Erlbaum Associates, Inc, 1996.

[23] M. Peshkin and J. E. Colgate, ‘Cobots’, *Ind. Robot Int. J. Robot. Res. Appl.*, vol. 26, no. 5, pp. 335–341, Jul. 1999.

[24] ‘ABB Group - Leading digital technologies for industry’. [Online]. Available: https://new.abb.com. [Accessed: 20-May-2019].

[25] A. Vysocky and P. Novak, ‘HUMAN – ROBOT COLLABORATION IN INDUSTRY’, *MM Sci. J.*, vol. 2016, no. 02, pp. 903–906, Jun. 2016.

[26] C. Liu and M. Tomizuka, *Robot Safe Interaction System for Intelligent Industrial Co-Robots*. 2018.

[27] ‘Exhibit No. 3: A humanlike approach - SMErobotics’. [Online]. Available: http://www.smerobotics.org/AUTOMATICA/Exhibit_No_3.html. [Accessed: 27-May-2019].

[28] I. E. Makrini *et al.*, ‘Working with Walt: How a Cobot Was Developed and Inserted on an Auto Assembly Line’, *IEEE Robot. Autom. Mag.*, vol. 25, no. 2, pp. 51–58, Jun. 2018.

[29] D. Gorecky, M. Khamis, and K. Sebastian, ‘Introduction and establishment of virtual training in the factory of the future’, *Int. J. Comput. Integr. Manuf.*, pp. 1–9, Jul. 2015.

[30] H. Barton and R. Delbridge, ‘Development in the learning factory: training human capital’, *J. Eur. Ind. Train.*, vol. 25, no. 9, pp. 465–472, Dec. 2001.

[31] E. R. Eide and M. H. Showalter, ‘Human Capital’, in *International Encyclopedia of Education (Third Edition)*, P. Peterson, E. Baker, and B. McGaw, Eds. Oxford: Elsevier, 2010, pp. 282–287.

[32] J. Goulding, W. Nadim, P. Petridis, and M. Alshawi, ‘Construction Industry Offsite Production: A Virtual Reality Interactive Training Environment Prototype’, *Adv. Eng. Inform.*, vol. 26, pp. 103–116, Jan. 2012.

[33] S. Nazir, L. J. Sorensen, K. I. Øvergård, and D. Manca, ‘Impact of training methods on Distributed Situation Awareness of industrial operators’, *Saf. Sci.*, vol. 73, pp. 136–145, Mar. 2015.

[34] A. De Santis, B. Siciliano, A. De Luca, and A. Bicchi, ‘An atlas of physical human–robot interaction’, *Mech. Mach. Theory*, vol. 43, no. 3, pp. 253–270, Mar. 2008.

[35] S. Nazir, A. Gallace, M. Bordegoni, S. Colombo, and D. Manca, ‘Performance comparison of different training methods for industrial operators’, 2012.

[36] D. A. Bowman and R. P. McMahan, ‘Virtual Reality: How Much Immersion Is Enough?’, *Computer*, vol. 40, no. 7, pp. 36–43, Jul. 2007.

[37] M. Akay, ‘Virtual Reality Technology, by G. Burdea and P. Coiffet’, *Proc.- IEEE*, vol. 83, no. 9, p. 1294, 1995.

[38] E. Matsas and G.-C. Vosniakos, ‘Design of a virtual reality training system for human–robot collaboration in manufacturing tasks’, *Int. J. Interact. Des. Manuf. IJIDeM*, vol. 11, no. 2, pp. 139–153, May 2017.

[39] T. Parisi, *WebGL: Up and Running*. O’Reilly Media, Inc., 2012.

[40] Q. Ye, W. Hu, H. Zhou, Z. Lei, and S. Guan, ‘VR Interactive Feature of HTML5-based WebVR Control Laboratory by Using Head-mounted Display’, *Int. J. Online Eng. IJOE*, vol. 14, p. 20, Mar. 2018.

[41] ‘WebVR Coming to Servo: Part 1’, *Mozilla Mixed Reality Blog*, 24-Oct-2016. [Online]. Available: https://blog.mozvr.com/webvr-coming-to-servo-part-1/. [Accessed: 20-May-2019].

[42] N. Polys *et al.*, ‘Immersive analytics: Crossing the gulfs with high-performance visualization’, in *2016 Workshop on Immersive Analytics (IA)*, 2016, pp. 13–18.

[43] ‘A-Frame – Make WebVR’, *A-Frame*. [Online]. Available: https://aframe.io. [Accessed: 20-May-2019].

[44] ‘Oculus Rift: VR Headset for VR Ready PCs | Oculus’. [Online]. Available: https://www.oculus.com/rift/#oui-csl-rift-games=robo-recall. [Accessed: 20-May-2019].

[45] ‘ABB’s Collaborative Robot -YuMi - Industrial Robots From ABB Robotics’. [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi. [Accessed: 20-May-2019].

[46] 'IRB 14000 YuMi Data - ABB's Collaborative Robot -YuMi (Industrial Robots From ABB Robotics)'. [Online]. Available: https://new.abb.com/products/robotics/industrial-robots/irb-14000-yumi/irb-14000-yumi-data. [Accessed: 20-May-2019].

[47] 'DEONET, Netherlands, | ABB'. [Online]. Available: https://new.abb.com/products/robotics/case-studies/deonet-netherlands. [Accessed: 20-May-2019].

[48] 'RobotStudio - ABB Robotics'. [Online]. Available: https://new.abb.com/products/robotics/robotstudio. [Accessed: 20-May-2019].

[49] F. Caputo, M. Emporio, and A. Giachetti, 'Single-Handed vs. Two Handed Manipulation in Virtual Reality: A Novel Metaphor and Experimental Comparisons', 2017.

[50] A. Edsinger and C. C. Kemp, 'Two Arms Are Better Than One: A Behavior Based Control System for Assistive Bimanual Manipulation', in *Recent Progress in Robotics: Viable Robotic Service to Human*, vol. 370, S. Lee, I. H. Suh, and M. S. Kim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 345–355.

[51] R. Bejarano, B. Ramis Ferrer, W. M. Mohammed, and J. L. Martinez Lastra, 'Implementing a Human-Robot Collaborative Assembly Workstation', Jul. 2019.

[52] 'Don McCurdy', *Don McCurdy*. [Online]. Available: https://www.donmccurdy.com/. [Accessed: 20-May-2019].

[53] admin, 'Developers', *Neurable INC*, 22-Jun-2017. [Online]. Available: http://neurable.com/developers. [Accessed: 20-May-2019].

APPENDIX A: TEST QUESTIONNAIRE

Before the training:

1- How comfortable do you feel relating to working with YuMi? (1-5)

After the training:

2- What is the simulation about?
3- How many screws did you use?
4- How comfortable did you feel during the simulation? (1-5)
5- How realistic did you find the simulation? (1-5)
6- How easy was to follow the process? (1-5)
7- Would you be able to reproduce the process?
8- How comfortable do you feel relating to working with YuMi? (1-5)
9- Did you learn how the YuMi interacts during the process? (1-5)

Table 11. *Results of the questionnaire*

		User									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Question	**1**	4	1	1	4	5	2	2	3	5	4
	2	*	*	*	*	*	*	*	*	*	*
	3	16	16	12	12	12	12	16	12	16	12
	4	5	5	4	4	5	3	5	5	5	5
	5	4	4	4	3	5	3	5	4	5	4
	6	5	5	4	5	4	3	4	5	5	5
	7	Yes	**	**	**	Yes	Yes	**	Yes	**	**
	8	5	4	3	5	5	3	4	5	5	5
	9	5	5	5	5	5	5	5	5	5	5

* All the users answered, using different words, that the task consisted on the assembly of a box in collaboration with a cobot.

** Users knew the steps to follow but had some difficulties screwing.