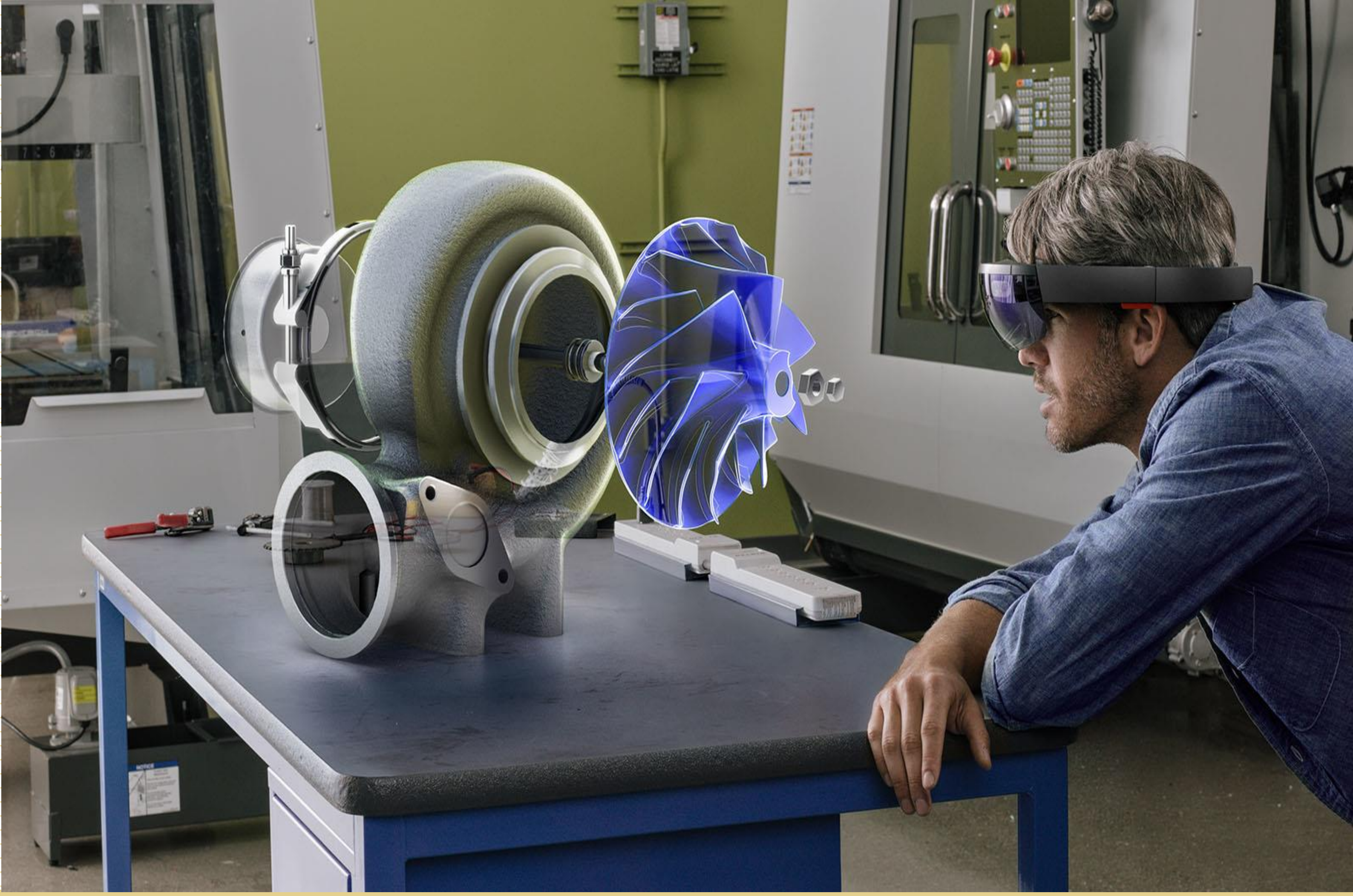

Diogo Emanuel Ferreira Serrario

Application of Mixed Reality devices for Robot Manipulator programming: aspects related with the system implementation

Master thesis in Mechanical Engineering in the specialty of Production and Project

July 2018

· U C ·

UNIVERSIDADE DE COIMBRA

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

DEPARTAMENTO DE
ENGENHARIA MECÂNICA

Application of Mixed Reality devices for Robot Manipulator programming: aspects related with the system implementation

Submitted in Partial Fulfilment of the Requirements for the Degree of Master in Mechanical Engineering in the speciality of Production and Project

Utilização de dispositivos de Realidade Mista para a programação de Robôs Manipuladores: aspetos relacionados com a implementação do sistema

Author

Diogo Emanuel Ferreira Serrario

Advisors

Professor Joaquim Norberto Cardoso Pires da Silva
Engenheiro Doutor Theo Doll
CEO na empresa Söhner Kunststofftechnik GmbH

Jury

President	**Professor Doutor Altino de Jesus Roque Loureiro** **Professor Associado com Agregação da Universidade de Coimbra**
Vowel	**Professor Doutor António Fernando Macedo Ribeiro** **Professor Associado com Agregação da Universidade do Minho** **Professor Doutor Carlos Xavier Pais Viegas** **Professor Auxiliar da Universidade de Coimbra**
Advisor	**Professor Doutor Joaquim Norberto Cardoso Pires da Silva** **Professor Associado com Agregação da Universidade de Coimbra**

Söhner Kunststofftechnik GmbH

KORIS robotics GmbH

Coimbra, July, 2018

“If you wish success in life, make perseverance your bosom friend, experience your wiser counselor, caution your elder brother and hope your guardian genius.”

Joseph Addison, in Eliza Cook’s Journal Volume 11, 1854

ACKNOWLEDGEMENTS

During my journey at University of Coimbra studying mechanical engineering, I learned vast concepts and different subjects that gave me the tools to introduce me on the marketplace, to develop my capacities, and to explore my way of thinking. I am conscious that this is the beginning of my academic journey. Since constant learning is necessary, in order to exercise and improve my working performance. Also important is the capacity of listening other professionals and colleagues.

The dissertation is the last piece of the puzzle to conclude my studies. As of now, I want to express gratitude to all my professors whom transmitted me their knowledge and granted me their valuable time to clarify concepts or questions that might of arose throughout my studies.

I would like to leave some words of gratitude to the people that made this dissertation possible.

To Prof. Norberto Pires, as my advisor, I am grateful for the opportunity to develop such an interesting dissertation's theme and for constantly guiding me on the right path.

To João Neves, my partner in this project. I appreciate all the hard work and loyalty throughout this journey. His future will be bright, indeed.

To Mr. Doctor Theo Doll that gave me the opportunity to do an internship in his company, in Germany, where the theme of this dissertation began and was purposed to be explored even further.

To Diana Lavado and Filipe Ribeiro, my friends of laboratory, that supported me during this semester.

I could not reach the final stage without the people around me in my personal life. They helped me to grow as a human being and, most important, taught me to abide by my duties and responsibilities.

To my friends that I shared the greatest moments with during these years and with who I lived a new variety of experiences. A word as well to my friends outside of the University that always encouraged me.

Lastly, to the people that provided their best aid into me reaching my goals over this journey:

To my mother, my sister and my grandparents, I have to thank the unconditional support, presence during my life and for giving me the opportunity to build a brighter future.

To Alexandra that never lets me down and is always present to support me and guide me with her love.

Abstract

Mixed Reality blends Augmented and Virtual Reality introducing in the real world virtual information to users. The upsurge of this technology in society has had a crucial impact in several areas as education, healthcare, industry, amongst others. The industry field invested in mixed reality in order to improve the products' quality as well as the operator's safety, which corroborates the perfect fit of this technology to the ideals of the industry 4.0 leading to the improvement of the cooperation between humans and machines.

Robotics' field has a lot to benefit from Mixed Reality, i.e., the robot programming could become easier and intuitive through a simple Human-Machine Interface (HMI) that guides the user through the robots' manipulation. In fact, manufacturing tasks such as additive-manufacturing needed new tools to analyze and optimize trajectories before they are executed.

Throughout the dissertation, the implementation of two different apps designed for the Mixed Reality device, *Microsoft HoloLens*, are explained.

The first app, path visualization, is used to visualize the trajectory made by a robot through an "augmented" line. The commands required to store all the position data from the robot are explained in detail as well as the communication protocol TCP/IP used to establish connection with the Mixed Reality environment, *Microsoft HoloLens*.

The second app, graphical path manipulation, allows users to design the trajectory in the Mixed Reality environment and, posteriorly, execute it with the robot. Parameters such as velocity and precision could be defined by the user, thus all the commands that stored the data and organized the robot's trajectory are part of this explanation.

The development of the created apps jointly with this dissertation intends to show to the industry the potential of the Mixed Reality in robotics.

Keywords Mixed Reality, Augmented Reality, Robotics, Human-machine interface (HMI), Path visualization, Graphical path manipulation.

Resumo

A Realidade Mista interliga a Realidade Aumentada e a Realidade Virtual introduzindo objetos virtuais no ambiente dos utilizadores. A génese desta tecnologia na sociedade teve um impacto crucial em diversas áreas como a educação, a saúde, a indústria, entre outras. O ramo industrial investiu na Realidade Mista para melhorar a qualidade dos produtos, assim como a segurança do trabalhador o que corrobora o enquadramento desta tecnologia na industria 4.0 levando a progressos na cooperação entre os homens e as máquinas.

O ramo da robótica tem muito a beneficiar com a Realidade Mista, ou seja, a programação de robôs poderá tornar-se mais fácil e intuitiva através de uma simples Interface Homem-Máquina (IHM) que guia o utilizador na manipulação de robôs. De facto, tarefas de produção como adição de material precisam de ferramentas para analisar e otimizar as trajetórias antes de as realizar.

Ao longo da dissertação, a implementação de duas *apps* diferentes que foram desenvolvidas para o dispositivo de Realidade Mista, *Microsoft HoloLens*, são explicadas.

A primeira *app*, “Visualização da trajetória”, é usada para visualizar a trajetória feita por um robô através de uma linha “aumentada”. Os comandos precisos para guardar todos os dados da posição serão explicados em detalhe, assim como o protocolo de comunicação TCP/IP usado para estabelecer a conexão com o ambiente de Realidade Mista, *Microsoft HoloLens*.

A segunda *app*, “Manipulação gráfica da trajetória”, permite ao utilizador formar a trajetória no ambiente de Realidade Mista e, posteriormente, executá-la com o robô. Os parâmetros como a velocidade e a precisão podem ser definidos pelo o operador. Assim, todos os comandos que guardam os dados e a organização da trajetória do robô fazem igualmente parte do âmbito desta dissertação.

O desenvolvimento das *apps* criadas conjuntamente com esta dissertação pretendem mostrar à indústria o potencial da Realidade Mista na robótica.

Palavras-chave: Realidade Mista, Realidade Aumentada, Robótica, Interface Homem-Máquina (IHM), Visualização da trajetória, Manipulação gráfica da trajetória

Contents

LIST OF FIGURES ix

ACRONYMS xi

1. INTRODUCTION 1
   - 1.1. Preamble 1
     - 1.1.1. Motivation 1
     - 1.1.2. Goals 3
     - 1.1.3. Structure 3
     - 1.1.4. Brief Note 4
   - 1.2. Related Work 4

2. REAL AND VIRTUAL WORLDS 7
   - 2.1. Concepts 7
   - 2.2. Devices 8
   - 2.3. Potential of Mixed Reality 10
   - 2.4. Society's examples 13

3. SYSTEM IMPLEMENTATION 16
   - 3.1. Software and Hardware 16
     - 3.1.1. Microsoft HoloLens 17
     - 3.1.2. ABB RobotStudio 20
     - 3.1.3. Vuforia 21
     - 3.1.4. Unity 3D 21
   - 3.2. Object Tracking 22
   - 3.3. RobotStudio Concepts 25
     - 3.3.1. Robtarget 25
     - 3.3.2. Matrix "rec_path" 25
     - 3.3.3. Robot Motions 26
     - 3.3.4. Persistent variable 26
     - 3.3.5. Input/Output Signals 27
   - 3.4. Path Visualization – RobotStudio 27
     - 3.4.1. Motion Task 28
     - 3.4.2. TCP/IP server task 31
     - 3.4.3. TCP/IP client task 32
   - 3.5. Graphical Path Manipulation – RobotStudio 34
     - 3.5.1. TCP/IP server task 34
     - 3.5.1. Motion task 37

4. RESULTS DISCUSSION 38
   - 4.1. Object Tracking 38
   - 4.2. Path Visualization 38
   - 4.3. Graphical Path Manipulation 39

5. CONCLUSION 40

BIBLIOGRAPHY 42

LIST OF FIGURES

Figure 1.1. Fields of the Industry 4.0. 1
Figure 1.2. Human-machine interface (HMI) from a robot. 2
Figure 1.3. Digital components overlaid in real objects (Michalos et al., 2016). 5
Figure 1.4. Path planning on the “augmented” workpiece (Li & Chu, 2016). 6
Figure 2.1. Mixed Reality spectrum (Zeller & Brandon, 2018b). 7
Figure 2.2. Hologram visualization with Microsoft HoloLens. 9
Figure 2.3. Virtual Reality device – Oculus Rift. 9
Figure 2.4. Forecast AR and VR market size worldwide – edited from (Statista, 2018). 11
Figure 2.5. Forecast shipments of AR and VR headsets worldwide – edited from (Statista, 2018). 11
Figure 2.6. Forecast user base of AR and VR software market worldwide – edited from (Statista, 2018) 12
Figure 2.7. Pokémon GO – Pokémon example. 13
Figure 2.8. Examples of AR application in pool (Sousa et al., 2016). 14
Figure 2.9. IKEA app – Chair visualization. 14
Figure 2.10. A trainee using HoloLens (left) and experiences a MR training displayed on the HoloLens (right) (Bottino et al., 2017) 15
Figure 3.1. Representation of the real robot work cell. 17
Figure 3.2. Microsoft HoloLens. 18
Figure 3.3. Microsoft HoloLens Menu. 19
Figure 3.4. Air tap gesture (Zeller & Brandon, 2018a). 19
Figure 3.5. Microsoft HoloLens spatial mapping. 20
Figure 3.6. RobotStudio working cell formed by the robot, workpiece and programming paths (yellow arrows). 21
Figure 3.7. Unity 3D representing the CAD tool used in the object tracking. 23
Figure 3.8. “Guide view” displayed by the HoloLens during the object tracking. 24
Figure 3.9. CAD model displayed by the HoloLens after the tool identification. Shows the trajectory’s first point. 24
Figure 3.10. Matrix “rec_path”. 25
Figure 3.11. Declaration of the matrix “rec_path” with the name trajectories. 26

Figure 3.12. Circular movement representing the initial, circular and final points (ABB Robotics, 2015). 26
Figure 3.13. Workpiece with programming path (RobotStudio). 27
Figure 3.14. PROC “execute_option” that contains three options of trajectories (RobotStudio – Motion task). 28
Figure 3.15. Part of the trajectory executed when case 2 is choose in “execute_option” (RobotStudio – Motion task). 29
Figure 3.16. Part of the instruction block of the PROC “main” (RobotStudio – Motion task). 30
Figure 3.17. Trap routine (RobotStudio – Motion task). 31
Figure 3.18. TCP/IP commands (RobotStudio – TCP/IP server task). 31
Figure 3.19. Instruction block when the message received from HoloLens is “start_sending” and “stop_sending” respectively (RobotStudio – TCP/IP server task). 32
Figure 3.20. Instruction block that creates a client and send information to the TCP/IP socket server in HoloLens device (RobotStudio – TCP/IP client task) 33
Figure 3.21. Instruction block that resets the variables (RobotStudio – TCP/IP client task). 34
Figure 3.22. Instruction block that stores the position in function of the index (RobotStudio – TCP/IP server task) 35
Figure 3.23. Instruction block with each type of motion (RobotStudio – TCP/IP server task). 36
Figure 3.24. Instruction block with the parameter velocity (RobotStudio – TCP/IP server task). 36
Figure 3.25. Part of the instruction block with the parameter precision (RobotStudio – TCP/IP server task). 36
Figure 3.26. Instruction block with the message “Data Received” sent to the HoloLens after a successful message being received by the robot controller (RobotStudio – TCP/IP server task). 37
Figure 3.27. Instruction block “free_path” procedure (RobotStudio – Motion task). 37
Figure 4.1. Workpiece designed in the holographic environment (HoloLens view) – left. Workpiece with programming paths (RobotStudio) – right. 39

ACRONYMS

AR – Augmented Reality

BSLD – Basic Support Life Defibrillation

HMD – Head-mounted Display

HMI – Human machine interface

IMU – Inertial measurement unit

ITimer – Interrupt Timer

MR – Mixed Reality

PbD – Programming by Demonstration

QR code – Quick Response code

OT – Object Tracking

SDK – Software Development Kit

VR – Virtual Reality

1. INTRODUCTION

1.1. Preamble

1.1.1. Motivation

Nowadays, technology is constantly growing, innovation is needed in order to maintain not only the users interested but also to compete against other brands. Humanity has crossed four industrial revolutions, at the moment is living its fourth revolution as explained by (Vaidya, Ambad, & Bhosle, 2018). This revolution has affected several fields of society as illustrated in Figure 1.1 with great impacts in Robotics, Additive manufacturing, Augmented Reality amongst others.

Figure 1.1. Fields of the Industry 4.0.

The robotics' field has increased its importance in the current industry and consequently in the economy (Dirican, 2015) defends that the impact of robotics in society will be inevitable and, therefore, solution must be found for its impact.. The industrial processes are faster, safer and easier, by avoiding dangerous and methodic tasks for the operator. At the same time, the final quality has grown, which decreases waisted products, rejected by the quality department. Which, therefore, increases efficiency. At this stage manual labor still keeps its importance in the industrial work's environment since it is necessary users to handle the robots. In order to establish this needed contact between the user and the robot there are Human-machine interface (HMI) as represented in Figure 1.2, which are user-friendly, in other words, simple to use. The goal of this interface is for the worker to control the robot and simultaneously interpret whole the data supplied by the robot. The user's background is crucial to understand the data sent from the robot. Thus, advanced knowledge in programming, mathematics and physics might be necessary in some cases.

Figure 1.2. Human-machine interface (HMI) from a robot.

As mentioned, the industry demands quality. So, it is essential to have advanced machinery and technology, as well as professionals with advanced skills in certain areas. The search for higher quality could be achieved by improving decision-making, which is influenced by the information provided.

That is the reason why Augmented Reality is ground-breaking when used in the industrial sector. AR is the overlay of virtual objects or information in the real world and in real time. This concept is the opposite of Virtual Reality that transforms the surrounding environment in a virtual scenario, different from the reality. Both concepts are completely divergent, but, at the same time, when used differently, they can complement one another. From this it came the emergence of a new concept called Mixed Reality, in which both virtual and real environments are blended with the intent of bringing both AR and VR pivotal technologies into one.

The introduction of Mixed Reality has proven to be of an extremely importance. As implicit before it provides real-time information that will improve decision-making and influence a better overall performance. Consequently, it will impact positively not only the final quality of products, but also the efficiency. Besides, the need for academic professional workers will decrease since users can learn how to use such technology even with a basic academic knowledge.

In the long run, the robotics' area can largely benefit from Mixed reality technology, by improving the manufacturing's quality and cost. That will, undoubtedly, lower the final price of a certain product, which can create more competitiveness and cheaper and accessible goods for the general public (having an effect in low income households and their consumption habits). Overall, it creates space for investments and efforts on a sustainable and fair industrial environment. Important since sustainability is the actual key-word for the future. Thus, this small technology could have pivotal repercussions not only for company's owners but for the whole humanity and their way of living.

1.1.2. Goals

The first goal wanted from this work resides in the development of two apps for a device, in this case, that device is the *Microsoft HoloLens*. The first app intends to visualize the trajectory made by a robot through the "augmented" path. Whereas, the second app pretends to build a new set of trajectories that the robot will follow, while adjusting a range of parameters such as velocity and precision. In other words, the user is able to do a trajectory manipulation. The second goal is to create apps that are accessible to a person with basic knowledge in the robotics' field.

Thus, the first goal is directly related with the robot itself, i.e. and the development of commands that enables the user to send and read information from the robot in the apps described above. The second goal is the explanation of the object tracking, content that will be explained in 3.2.

1.1.3. Structure

This dissertation is divided into chapters that aim to help the clarification and organization of the ideas exposed. The first chapter will explain the factors that motivated the development of this project as well the respective goals. The second chapter contains research about the theme which shows the importance of AR and VR to the society. The third chapter provides an in-depth explanation of the actual progress made into achieving the final goals. Subjects such as the software used or details about how to make the apps functional will be part of this explained content. Followed by the fourth chapter where the apps are presented in its full potential, the functionalities are discussed, and its flaws debated. Lastly, the chapter fifth which consists of the conclusions and the future of the project exposed.

1.1.4. Brief Note

The company Dr. Doll Engineering, located in Germany more precisely in *Schwaigern,* suggested the development, through AR, of Robot's path visualization and the optimization of these paths, during an internship that occurred in the summer of 2017.

The work development happened alongside João Neves. Even though, this project was developed by two people the content described in each of the dissertations is different since a careful division was made. This assignment of different content was achieved by taking into consideration the development made in each section of the project and the person responsible for it.

Lastly, a paper about the whole project development was submitted and accepted by Industrial Robot.

1.2. Related Work

It is possible to state that the Robotics' field is the fastest growing technology, at the moment. Small appliances start to be made in society, from the industry to healthcare. Briefly, a robot is a machine programmable by a computer. Everyone's handling robots in their day-to-day life, even without realizing. Their use starts at home even with a simple vacuum cleaner. The Military has been investing in self-defense robots, some healthcare systems use automated vehicles to deliver medicines. And the industry has benefit from all the Robotic developments by using them in dangerous tasks as welding and routine tasks as palletizing cartoons.

Daily improvements occur in the industrial sector. The challenges faced by the current technology users are the source of development and aspiration for greater machines, which also motivated the use of Mixed Reality for industrial purposes. Several efforts are being made to integrate AR and VR technology because its known that its use would, indeed, be beneficial for workers and owners.

The AR can be used to create a closer connection between the worker and the robot, which creates a more interactive experience. The development of an app for Android by (Michalos, Karagiannis, Makris, Tokçalar, & Chryssolouris, 2016) allowed the user to see virtual information as shown in Figure 1.3, that facilitated the relation between the user and the robot. The virtual information provided by that app has different functionalities. Such as, the display of parts of an assembly. Visual and audio alerts to grant the operator's safety since the increase of connection between the human and robot opens a gap for physical incidents and, lastly, to acquire information about the production status.

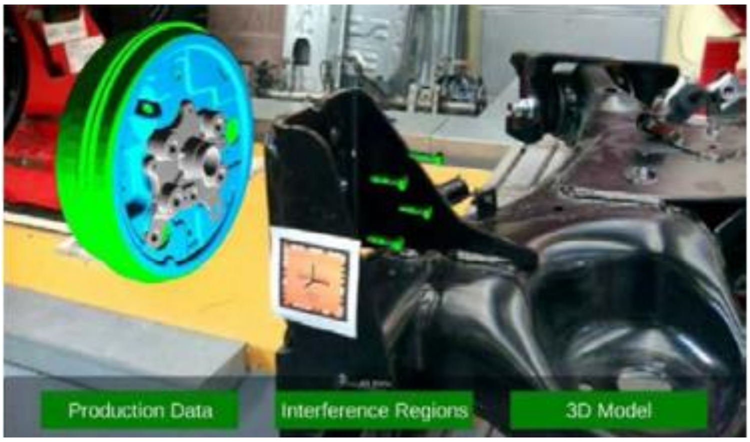

Figure 1.3. Digital components overlaid in real objects (Michalos et al., 2016).

Simulation can be used to test the performance of mobile robots. Mixed reality could be used as a simulation process. With this in mind (Chen, MacDonald, & Wunsche, 2009) designed a system that introduced different types of virtual objects in the working space. While the robot was doing their tasks in the real world it was possible to test if it was avoiding the real and virtual objects.

Programming by demonstration (PbD) is, briefly, when the user shows how to do a certain task and the robot copies it. The combination of PbD and AR could have a large effect, in that specific case both technologies were used to do path planning by (Li & Chu, 2016). The goal was to draw a path in a virtual object and the robot would do that same trajectory. The path could be visualized by the operator as shown in Figure 1.4, which allowed the correction of eventual errors and avoid difficulties in programming paths.

Figure 1.4. Path planning on the “augmented” workpiece (Li & Chu, 2016).

A recent app created by (WyzLink, 2018) for the *HoloLens* intends to manipulate the trajectory. The user can draw a random trajectory implementing several holographic points, that are immediately executed by a virtual robot. This practice will be very useful to observe the trajectory made by a real robot before its execution. This app has similarities with this project since a graphical manipulation path was also created.

In recent years the increasing connection between robotics and Mixed Reality is undeniable, that can be seen through the number of apps being created blending these two. In the near future the apps will be integrated in robots and devices which will be prepared to work with MR.

2. REAL AND VIRTUAL WORLDS

2.1. Concepts

Mixed reality (MR) mergers the real with the virtual world. The combination of both allows the creation of a real time interaction between virtual and real objects. Its use can occur in a real environment surrounded by virtual objects. Or in an environment that is completely virtual only limited by the real spatial world. This concept was introduced by (Milgram & Kishino, 1994) in which MR was defined through a spectrum. The modern spectrum, that can be seen in Figure 2.1, is formed by two opposite sides, on the left side, Augmented Reality (AR), and on the right side, Virtual Reality (VR).

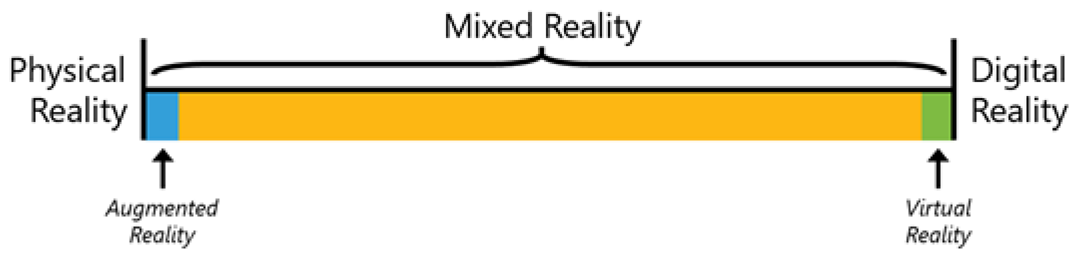

Figure 2.1. Mixed Reality spectrum (Zeller & Brandon, 2018b).

Augmented Reality can be described as the introduction of virtual objects in the real world that can be seen in real time and in three dimensions. Although there is virtual information overlaid in the real world, the integration of different sensations makes the AR a natural experience as it is lived in the real world. AR is quite new in the market, even though, the concept initially emerged in the 20th century. Lately, brands have been adapting their devices to integrate this reality.

Virtual Reality is an environment of simulation that could present similarities with the real world. Although, the environment is completely virtual it presents three dimensions, image, sound amongst other sensations. These sensations lead to a reaction by the human senses. When VR is experienced an interaction is created between the user and the items available in the virtual world. The current market has a higher use of VR than the

AR, namely in game simulation and in certain professional training sessions for areas such as medicine or education. The use of VR technology is currently made by using smartphones and glasses.

Therefore, Mixed Reality benefits since it uses AR and VR's characteristics, by crossing real with virtual and creating only one reality which offers several simultaneous experiences now available in one device. Through the analyze of the spectrum present in Figure 2.1, AR and VR individually are also considered MR. Even though, MR as an experience includes both Augmented and Virtual Realities.

2.2. Devices

Nowadays, technological devices such as tablets and smartphones are part of human's daily life. Technology brands are constantly innovating by presenting new features. Despite all the advantages of MR mentioned above, the current devices are not fully prepared to integrate it. Brands' investments have been made into creating gadgets that support either VR or AR, but not both. In addition to smartphones and tablets, another gadget emerged to the market, the head-mounted display (HMD).

HMD is a display device which is placed on the head and used like a helmet. If it is formed by one optic it is monocular, and if by two optics it is a binocular HMD. The goal of this device is to show content like images or videos and, depending on the type of HMD device, could actually serve to see-through. It is possible to name two types of HMD: holographic devices and immersive devices.

Holographic devices are a see-through display, as the name suggests, it is possible to see the real environment when using them. These devices can show holograms, as seen in Figure 2.2, i.e., 3D images which can play sounds. The combination between the sight of the real environment with the overlay of virtual images indicates that they are more suitable with AR.

Figure 2.2. Hologram visualization with *Microsoft HoloLens*.

On the other hand, immersive devices are opaque displays, that means that they do not access the real environment. In this case the virtual content used replaces the real world and creates a sensation of "attendance". The device is able to convey senses resembling the reality which stimulates and transports the human to believe he is in an actual world. This type of device is used with VR and is extremely prominent in the gaming field. A VR device example is represented in Figure 2.3.

Figure 2.3. Virtual Reality device – Oculus Rift.

Even though, both realities are used individually a clear trend has started to emerge of mixing both AR with VR. This trend blends holographic and immersive characteristics into one device. Thus, the future technology will be more complete and will be able to bring the user to such a credible level of virtual reality that to distinguish what is real and what is not will be challenging.

.

2.3. Potential of Mixed Reality

Even though, Virtual reality and Augmented reality started to be explored from the 20th century onwards, these concepts only gain more relevance in recent years. The first experiments resulted into virtuous innovations like the HMD, already presented in 2.2 invented by (Sutherland, 1968). The developments maintained the right direction and some years later a virtual fixture system developed by (Rosenberg, 1992) was introduced to the world, Rosenberg was a pioneer on the concepts of VR and AR. This innovation was well received by society, which incorporated the concepts introduced into several areas such as entertainment, healthcare, military training and so on.

Since its creation Mixed reality has had a tremendous growing potential that could move billions of dollars. Some years ago, brands such as Google and Microsoft started to invest in VR and AR through the development of HMD. The boom occurred in 2016, when the market started to feel the effect of AR with the launch of the *Microsoft HoloLens* and the game Pokémon Go. Whilst VR had its boom with the launch of the Oculus Rif. The market size of these both technologies expanded gradually, now a 27 billion of U.S. dollars are predicted as market size to 2018, second Figure 2.4. The investment will keep growing in such a way that the value can reach 209.2 billion of U.S. dollars in 2022, approximately ten times more in just four years.

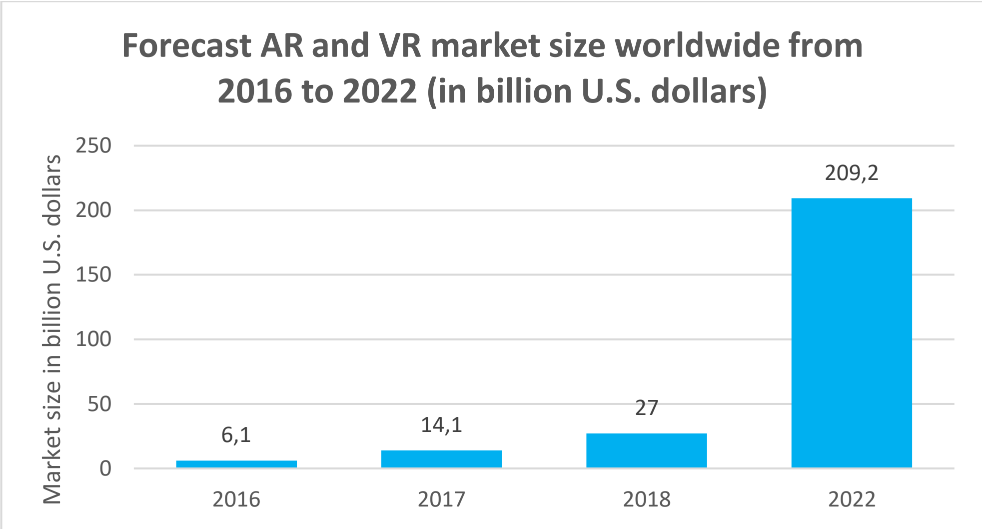

Figure 2.4. Forecast AR and VR market size worldwide – edited from (Statista, 2018).

As mentioned above the boom occurred in 2016 and it was especially with HMD products, as seen in Figure 2.5, with 10 million of units sold. The jump from 2015 to 2016 is visible since in the debut year of HMD products a rapid progression on sales occurred. The sales maintain approximately a constant rate with the prediction of a slight increase during 2018. The highest growth will be in the next years reaching a peak of 59,2 and 68,9 million of units sold in 2021 and 2022, respectively.

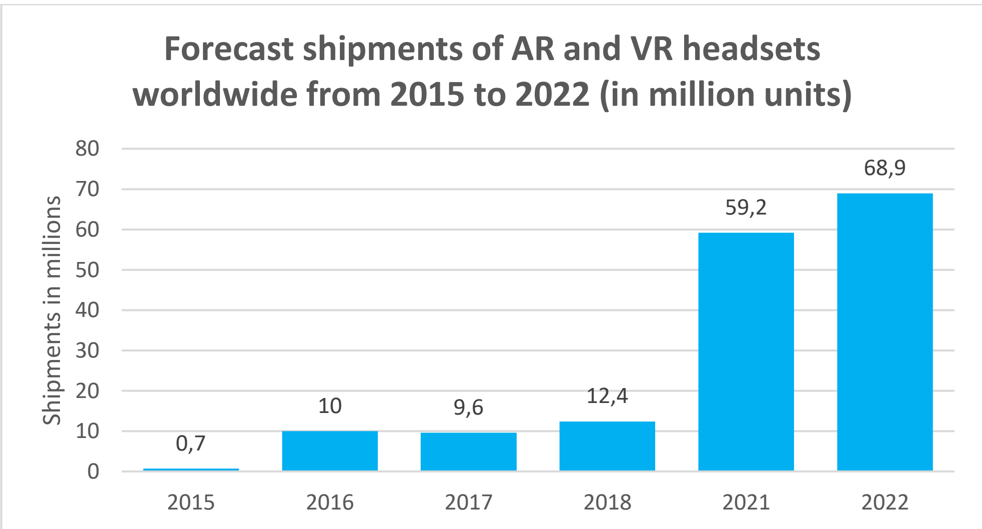

Figure 2.5. Forecast shipments of AR and VR headsets worldwide – edited from (Statista, 2018)

Society has several areas that can and will benefit from mixed reality. The entertainment field, especially Videos Games, are going to be impacted by Mixed Reality, which is natural, when considering that this field also explores the unreal and virtual world. However, with this new technology it will be possible to enhance the user's experience and transform the current senses used in video games into a full sensorial experience in the future. It is predicted that in 2025 there will be 216 million of available software in the market.

Mixed Reality will be used to assist society in pivotal areas such as education, healthcare and engineering. In terms of the educational system new learning methods will be developed using this technology. The healthcare branch can become more efficient and human errors could rapidly decrease. The engineering field will gain a wider range of software available with an estimation of 3,2 million in 2025 as shown in Figure 2.6.

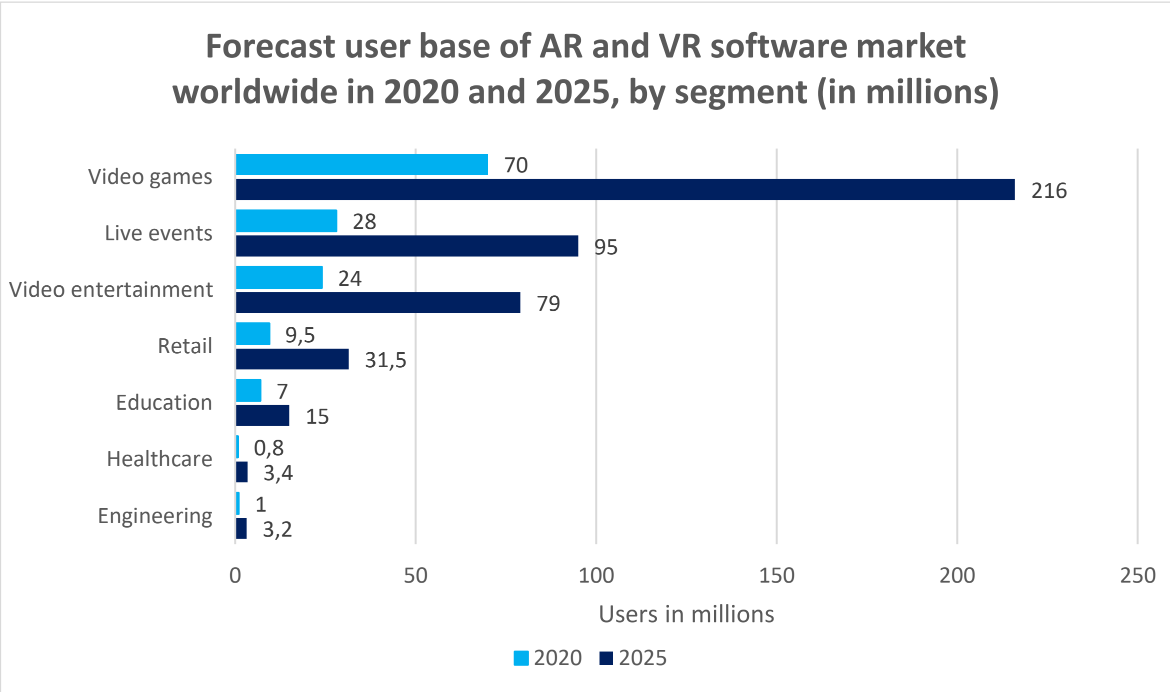

Figure 2.6. Forecast user base of AR and VR software market worldwide – edited from (Statista, 2018) .

2.4. Society's examples

In the sub-chapter 1.2 different ways of using Mixed, Augmented and Virtual reality in robotics were described, but also, as explained above, this technology could potentially be used in different areas of society such as gaming, education, healthcare, engineering, amongst others. So, actual examples of the use of this technology can be seen beyond the robotics' field. The utility of overlaying virtual information is being demonstrated as crucial to supply vital information in real-time.

Gaming, as noted in the chapter above, is the area that invests more in AR and VR. The use of VR technology is used to turn games more realistic. The boom with AR happened in 2016 when the game Pokémon Go became extremely popular, even sending people into a frenetic state. The main goal was to search for Pokémons, as seen in Figure 2.7, through a map based in the real world. The visualization of the Pokémon, virtual object, happens in the real world as if it was actually there.

Figure 2.7. Pokémon GO – Pokémon example.

The entertainment area has used AR as well. The pool player system developed by (Sousa, Alves, & Rodrigues, 2016) uses AR to show the trajectory of the ball that the player will do in accordance to the cue and the white ball position. There is a group of images in Figure 2.8 that show different applications of AR in pool.

Figure 2.8. Examples of AR application in pool (Sousa et al., 2016).

Even well-known companies like IKEA, the Swedish company that sells furniture, has invested in AR. IKEA developed a free app to let the customers see how the furniture fits their home before buying it as illustrated in Figure 2.9. The furniture is augmented through AR conception.

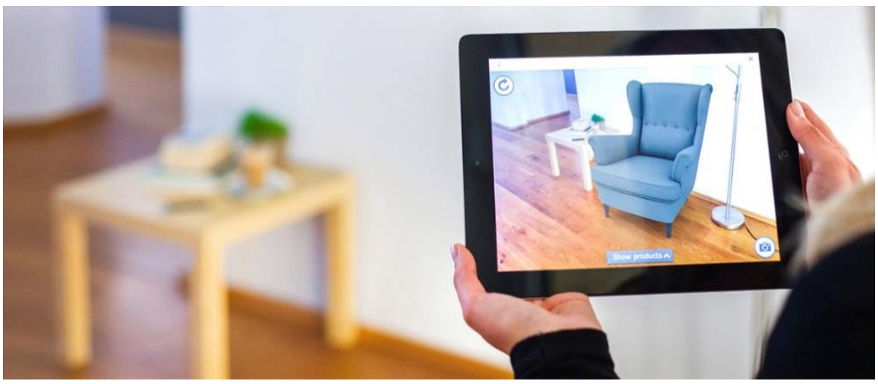

Figure 2.9. IKEA app – Chair visualization.

Healthcare and AR could perform an important role together. This study made by (Bottino et al., 2017) demonstrated the system of Effective Basic Life Support Defibrillation (BLSD) could be highly helpful to train professionals and for self-learning. In this case the *Microsoft HoloLens* were used to apply the BSLD procedure through a virtual environment into a manikin, where it was possible to see the human organs as represented in Figure 2.10. The apprenticeship became highly accessible and cheaper, resulting in more efficient ways of training professionals.

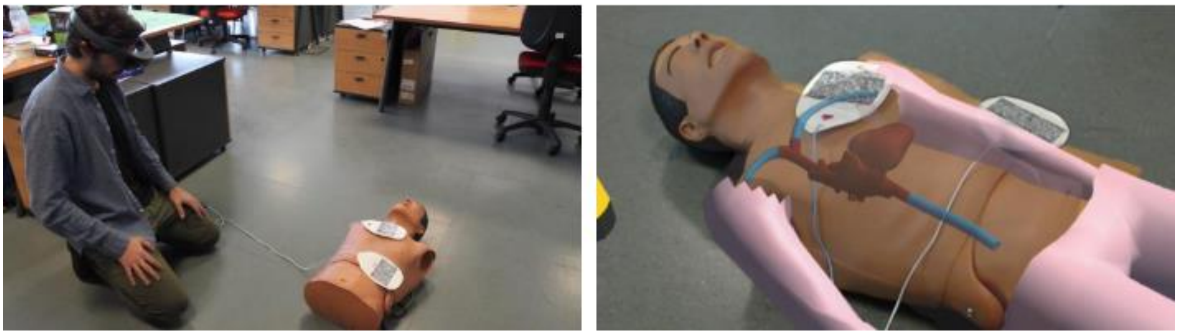

Figure 2.10. A trainee using *HoloLens* (left) and experiences a MR training displayed on the *HoloLens* (right) (Bottino et al., 2017) .

These examples are just a few when compared to the number of existing ones. Nevertheless, they emphasize the value of Mixed Reality in society. If not all areas, almost all could benefit from the mixing of virtual and real worlds.

3. SYSTEM IMPLEMENTATION

3.1. Software and Hardware

During this section the software and the material used will be described, exposing the respective advantages. The choices were made after a period of research, deliberation and a market study in order to verify the best available equipment and software. At the end it was chosen the *Microsoft HoloLens* since it is one of very first devices presenting Mixed Reality. Also needed was a simulation environment to obtain the whole data. ABB *RobotStudio* was chosen for that function since the software presents flexible characteristics by containing a simple programming language. For the MR scene editing, *Unity 3D* was chosen as software recommended by *Microsoft* to work with the *HoloLens*. *Vuforia* package was used to work the object tracking, explained in 3.2. The work cell represented in Figure 3.1 is composed by ABB IRB140 (running the IRC5 robot controller with *Robotware OS* version 5.14 or higher). The robot has a simple tool to follow the workpiece's contours that could be replaced by tools with different purposes like welding, material deposition, amongst others. A support to hold the robot and a table for the workpiece are the other components that are part of the working cell.

At the end two apps were created using the programs mentioned above that could potentially serve as industry's optimization of tasks, and robots' trajectories manipulation.

Figure 3.1. Representation of the real robot work cell.

3.1.1. *Microsoft HoloLens*

The *Microsoft HoloLens*, as seen in Figure 3.2, was the device used during the development of this project. The *HoloLens* was launched in 2016, it is an HMD holographic device with, currently, the higher capacities in the market. Although it is considered a holographic device. It combines both AR and VR which makes it one of the first to present Mixed reality. This factor was pivotal when deciding which device would support this work. The *HoloLens* was conceived to revolutionize this era of technology and impact severely the next generation of smartphones and tablets.

Figure 3.2. *Microsoft HoloLens.*

In general, the hardware is composed by the see-through holographic lenses, one inertial measurement unit (IMU), four "understanding environment" cameras, four microphones, depth camera, one photographic video and video camera and one ambient light sensor.

The *HoloLens* has capacities that go beyond anything else presented before in the technological field. There are several inputs that could be used in order to complete tasks with the device for example gaze, voice, gestures, amongst others. Gaze is a system that indicates where the user is looking at. Normally the user has a cursor that allows him to interact with the surface's surroundings. When the user moves his head, the gaze follows him. The gestures are very important to interact with the holograms and the menus of the *HoloLens*. It is through gestures that it is possible to open, close, select windows and an overall interaction with the holograms. There are pre-defined gestures such as the bloom gesture to open and close the menu (Figure 3.3) and the air tap that simulated a mouse "click" as is explained in Figure 3.4. Alternatively, the voice input can be used to equally close windows just by saying the command *"close"*. Furthermore, it is possible to do questions and manage device controls with the system implement that analyzes your voice input. The user only needs to say the words *"Hey Cortana"* followed by a question or an order, for example, to turn off the device that is activated with the command *"shut down"*.

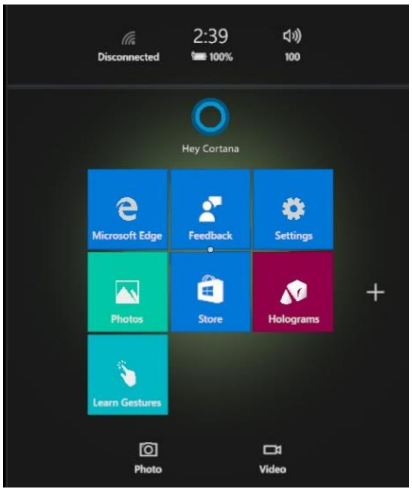

Figure 3.3. *Microsoft HoloLens* Menu.

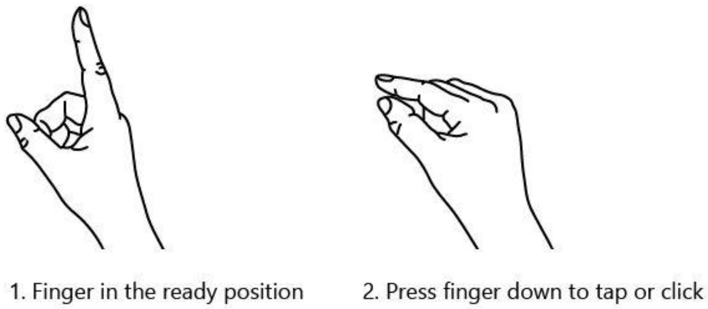

Figure 3.4. Air tap gesture (Zeller & Brandon, 2018a).

The key-factor of the *HoloLens* are the spatial mapping systems. This feature is the main difference between the *Microsoft HoloLens* and other devices in the market. The spatial mapping recognizes the surfaces of the real world, as seen in Figure 3.5, allowing to display virtual content into a real object. For example, it can display a hologram on a table. It is equally important to see when a real object is hidden by a hologram or vice-versa. Spatial mapping turns the interaction between the user and the virtual world something natural and maintains the positions of the holograms where they were assigned. This is only possible because the hologram's positions are locked in relation to the real objects, this functionality is called *anchor*.

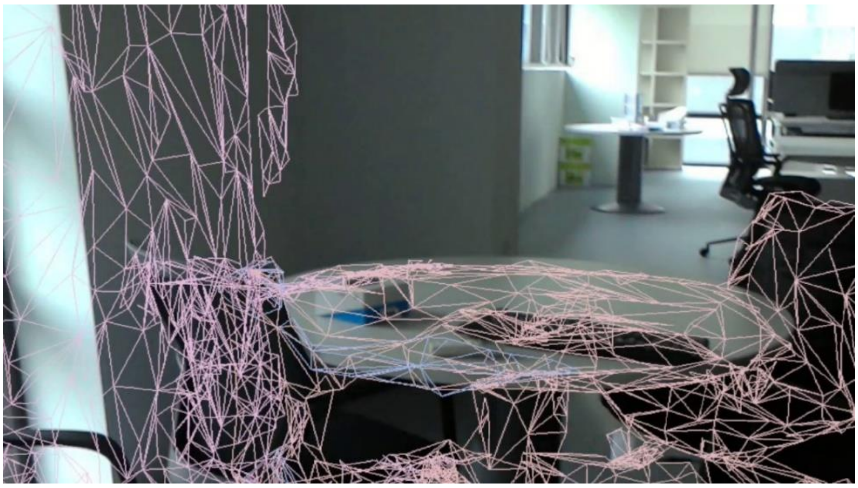

Figure 3.5. *Microsoft HoloLens* spatial mapping.

3.1.2. ABB *RobotStudio*

ABB *RobotStudio* is a simulation software from ABB Robotics, the Swedish well-known robotics brand. This software is powerful and, at the same time, has a programming language simple to use called RAPID. *RobotStudio* contains commands to design a TCP/IP client/server, that is fundamental to have an existing communication between the robot and the *Microsoft HoloLens*. The communication is based in an exchange of messages according to the information requested by the *Microsoft HoloLens*.

The software represents exactly the real work cell, as represented in Figure 3.6, which enables the simulation of the whole task through the software. Thus, time is saved to test the various solutions, being that one of the greatest advantages of the simulation environment.

The program has multi-tasking functions, i.e., it can run different tasks at the same time. Each task requests a certain memory to be executed, when more than one task is executed, the memory is shared according to the needs of each task.

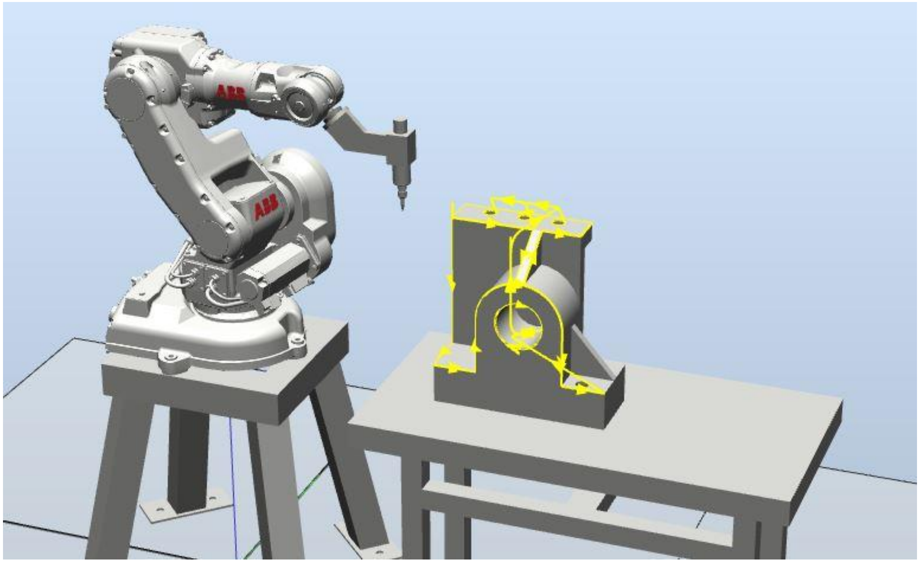

Figure 3.6. *RobotStudio* working cell formed by the robot, workpiece and programming paths (yellow arrows).

3.1.3. *Vuforia*

Vuforia is a Software Development Kit (SDK) focused on AR. The *Vuforia* team created various solutions to improve the implementation of the AR. The methods developed by this team have spread their use into different professional branches of society. For this project it was used one of *Vuforia's* methods, the *model target*, which recognizes 3D objects, this is later explained in 3.2, since it is used for object tracking. Another method is the *image target* that uses an image or a Quick Response code (QR code) to identify virtual information and display it to the user. In an initial phase, this method was used to do object tracking. Nevertheless, due to low accuracy of performance in the pretended task the method became unviable to apply, being later verified that the *model target* was the correct choice. The range of software that could use the tools supplied by *Vuforia* are *Unity 3D*, *iOS*, *Android* and *Universal Windows Platform* (UWP). For this project its use was facilitated since the *Vuforia* package comes integrated in *Unity 3D*.

3.1.4. *Unity 3D*

Unity 3D is a crossed-platform framework. This software is known in the gaming world since it is used to develop games. *Unity 3D* is prepared to create apps to the main smartphones' software and UWP. To develop any app for the *Microsoft HoloLens*, *Unity 3D* is the program recommended

This platform has an environment to build scenes through commands that could be fully programmed with *Microsoft Visual C#* language. *Unity 3D* served to build the object tracking, to draw the robot's paths through AR, to develop the TCP/IP communication from the *HoloLens* and the menus that display in the apps. The object tracking is part of this dissertation's scope. The remainder content is being developed by João Neves in his dissertation.

3.2. Object Tracking

The primary goal of Object Tracking (OT) is to identify the tool in order to match its "augmented" position with the real tool position. This is vital to correctly overlay the robot's path, i.e., to display the robot's path in the respective edges of the workpiece, the real object. In technical terms, there is a match between the robot *spatial coordinates*, received from the robot's controller, with the *HoloLens* equivalent *world coordinates.* Then, a ball (virtual object) is displayed on the tooltip and fixes those coordinates into the system. That ball will be the first point of the trajectory once the path starts to be drawn. Both the robot's controller and the *HoloLens* have the same crucial starting point, this way the device is always calibrated. The process initiates with the tool starting in the same position called the *home position*, this is a standard position defined in the *RobotStudio*. With the definition of this first point the robot's range area becomes perceptible to the operator which then is able to assort the robot's limitations since each mechanical arm has a certain reach area.

The object tracking was executed with a feature available in *Vuforia* (Vuforia, 2018b) recently launched called *model target*. The OT needs the CAD model in to do the correct tracking of the tool. The CAD extension must be converted to a readable extension to fit the *Unity 3D*. The converter is supplied by the *Vuforia* team (Vuforia, 2018a).

The model target is still under development therefore, mistakes might occur during the OT process. Nevertheless, the app is functional and presents a medium quality of performance.

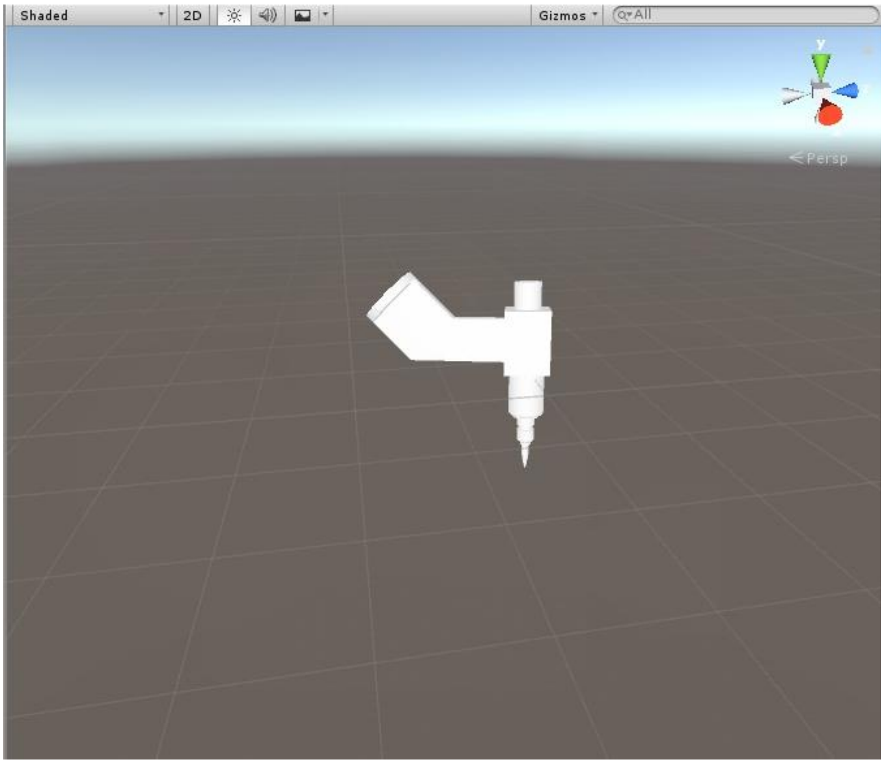

Figure 3.7. *Unity 3D* representing the CAD tool used in the object tracking.

A key point was the design of the CAD model (with *Autodesk Inventor*) with the exact measure of the real tool, that was created with 3D printer. That way, the detection accuracy is higher since the measures between the CAD model and the real tool are perfectly equal.

The tool's identification is the first function activated when turning on the *Microsoft HoloLens*. The identification can be divided into two steps. Firstly, it is displayed the guide view of the tool as shown in Figure 3.8. The user needs to move his head to match the guide view with the real object. After that the "augmented" tool will appear on the screen (Figure 3.9), which means that the identification was well succeeded, with the ball correctly attached to the tooltip. Secondly, the "augmented" tool is turned off keeping only the virtual ball (initial point) so that the user's vision area can be cleared.

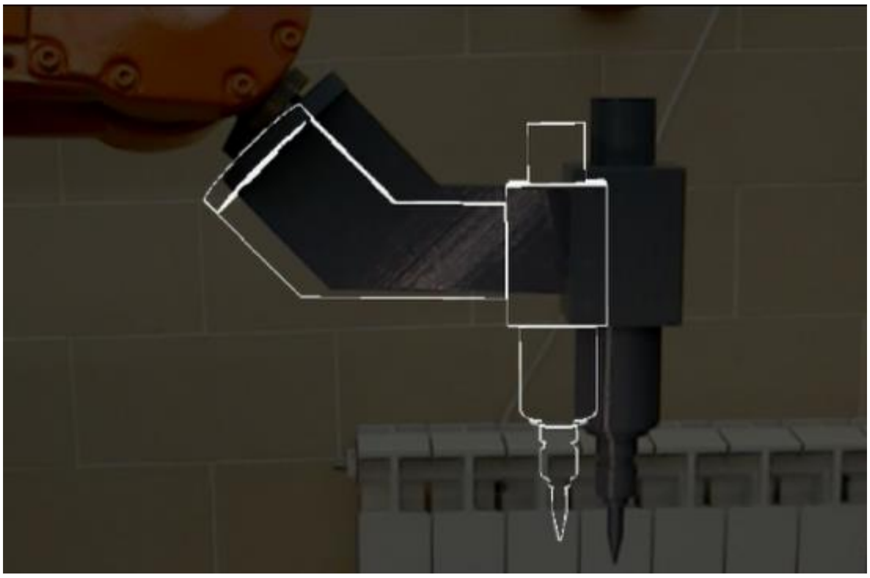

Figure 3.8. *"Guide view"* displayed by the HoloLens during the object tracking.

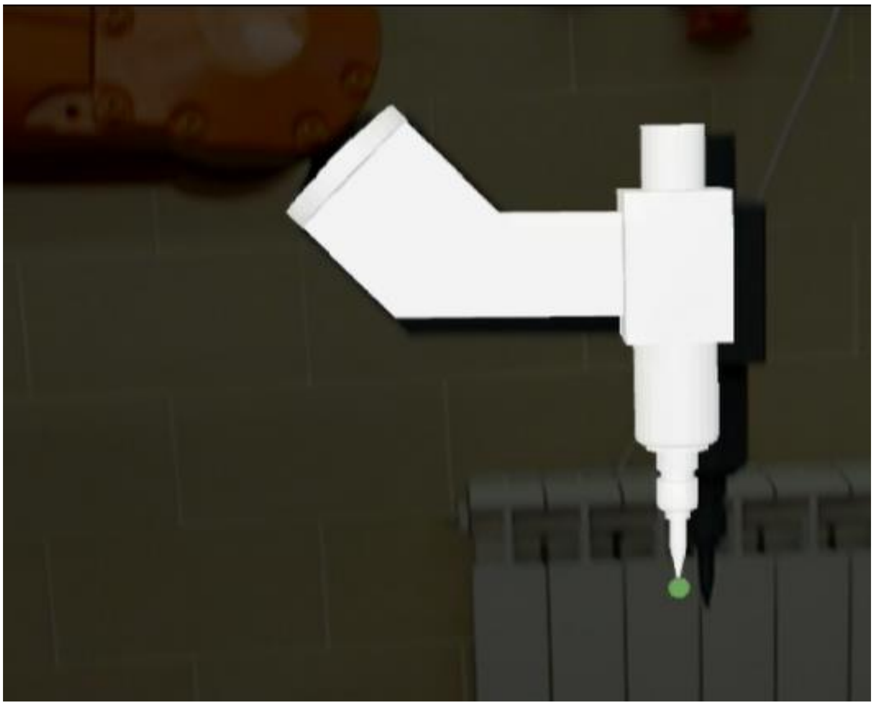

Figure 3.9. CAD model displayed by the *HoloLens* after the tool identification. Shows the trajectory's first point.

During the detection process the environment conditions were vital to get a successful identification and maximize the exercise. The brightness is one of the fundamental aspects. Over the testing phase it was difficult to correctly detect the tool due to the poor lighting conditions. Thus, a source light was used to focus on the tool.

3.3. *RobotStudio* Concepts

3.3.1. *Robtarget*

A r*obtarget* defines the robot's position and its axis. Its first three elements represent the coordinates x, y and z in millimeters. Followed by four orientations (q_1, q_2, q_3, q_4) and four axis configurations (cf_1, cf_4, cf_6, cf_x). The last configuration is always zero for this working robot (ABB IRB140) because it is a robot of six axis. The last elements that constitute the *robtarget* are the position of the six external joints (eax_a, eax_b, eax_c, eax_d, eax_e, eax_f). Detailed information about the targets can be read in (ABB Robotics, 2015).

3.3.2. Matrix *"rec_path"*

The matrix *"rec_path"* as seen in Figure 3.10, stores the data from each point that composed a certain trajectory. This trajectory can be programed in the robot's controller or can be created by the user in the holographic environment. The data can be acceded whenever it is needed unless the order to delete it is given. In total the matrix has six elements. Each element has the data type, followed by its name. The first element stores the *robtarget*. The second stores the number of the motion, that might be either linear or circular. The third and the fourth elements store the velocity and the precision respectively. While the fifth and the sixth store the tool and the *workobject*'s name.

```
RECORD rec_path
    robtarget position;
    num motion_type;
    speeddata velocity;
    zonedata precision;
    string tool_name;
    string wobj_name;
ENDRECORD
```

Figure 3.10. Matrix *"rec_path"*.

The matrix was declared with the name *"trajectories"* being a persistent variable to be recognize in other tasks as shown in Figure 3.11 . This variable has the capacity to store 300 points, but this number can be higher according to the function performed by the robot.

```
PERS rec_path trajectories{300};
```

Figure 3.11. Declaration of the matrix *"rec_path"* with the name trajectories.

3.3.3. Robot Motions

There are three types of motions available in *RobotStudio*. The linear motion represented by *MoveL*, the circular motion represented by *MoveC* and the joint motion represented by *MoveJ*. The definition of each motion requires information regarding the *robtarget*, velocity, precision and the *workobject,* coordinate system that the robot is related to. Circular motions need two *robtargets* to be defined corresponding to the circle point (*CirPoint*), which is a trajectory's middle point and final point (*Topoint*) as illustrated in Figure 3.12.

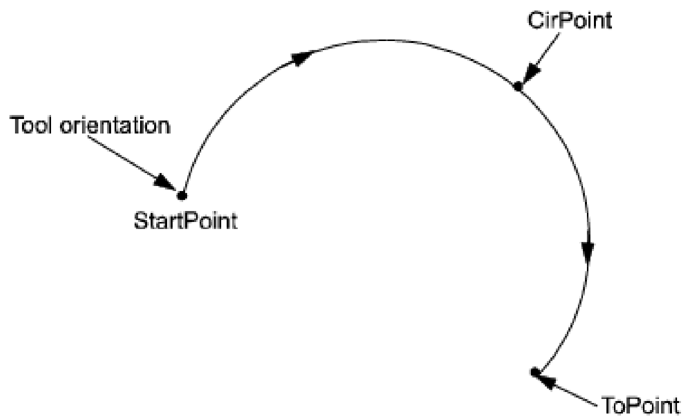

Figure 3.12. Circular movement representing the initial, circular and final points (ABB Robotics, 2015).

3.3.4. Persistent variable

A persistent variable uses the keyword PERS to be declared in RAPID programming language. This type of variables have an important characteristic that sets them apart from the others since the persistent variables remember the last value assigned to them even if the program was stopped or started from the beginning which justifies the importance of the use of this type of variables in a multi-tasking environment. During the explanation of each task whenever a variable has the same name means that is a persistent variable, which is the case of the variable "*trajectories*" mentioned in Figure 3.11.

3.3.5. Input/Output Signals

Signals are very helpful for the communication between the robot and the sorrounding environment. Signals can be used to alert the operator or other equipments. Input signals are set outside RAPID while output signals are set inside RAPID. Whenever an input signal is defined, usually by other equipment, it means the robot has a task to perform. On the other hand, if the output signal is set, usually it is an equipment that needs to work instead of the robot. These digital outputs could be assigned to the values 0 or 1.

3.4. Path Visualization – *RobotStudio*

Path Visualization intends to show the trajectory made by a real robot while it is working. This trajectory is represented through the *Microsoft HoloLens* with an "augmented" path displayed onto the workpiece. The path is designed in *RobotStudio* as illustrated in Figure 3.13. The protocol to obtain and send the data to the *Microsoft HoloLens* is explained thereafter. The visualization path is divided into three tasks that will be explained below.

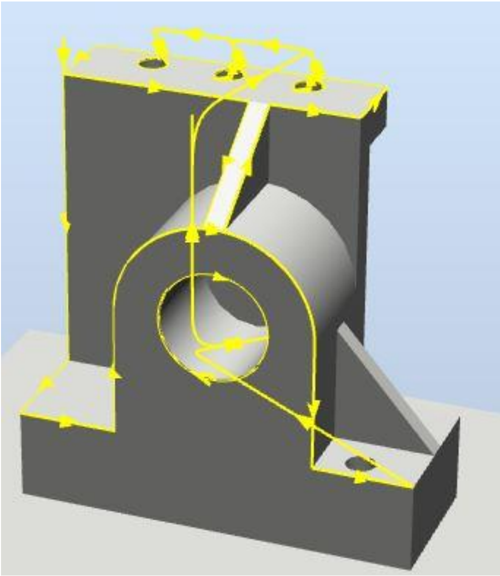

Figure 3.13. Workpiece with programming path (*RobotStudio*).

3.4.1. Motion Task

The motion task runs in normal mode being the only one that can move the robot. This task is essential for the robot movement and it is the only one that executes the commands to start and stop the robot. Those commands are given through a holographic menu in the *Microsoft HoloLens*.

The motions stored, in this task, are created in the robot simulation environment through a functionality called *AutoPath*, which allows to generate *robtargets* according to the workpiece's shape. Afterwards in order to design the path, the targets are synchronized to RAPID, that is the programming language of RobotStudio.

This task contains different procedures that are declared with the keyword PROC. Whenever the program starts, the procedure with the name *main* is automatically executed followed by other procedures called by programming order. The PROC "*execute_option*", as seen in Figure 3.14 has a menu that contains three trajectories options. Those trajectories are implemented after the variable "*comando*" is assigned to a value different than 0. If the variable *"comando"* is one, two or three the trajectories *holes*, *full* and *go_home* are respectively executed by the robot. In the first case, the trajectory executed contains all circular motions present in the workpiece, whereas the second case corresponds to the trajectory showed in Figure 3.13 and the third case to a standard robot position which is applied before starting any trajectory. A default wait time is set to 1.5 seconds when the variable "*comando*" is assigned to a different value from 1, 2 or 3.

```
PROC execute_option()
    TEST comando
        CASE 1: holes;
        CASE 2: full;
        CASE 3: go_home;
        DEFAULT: WaitTime 1.5;
    ENDTEST
```

Figure 3.14. PROC "*execute_option"* that contains three options of trajectories (*RobotStudio* – Motion task).

For example, when the second case is chosen, the PROC "*path_10*" executes the trajectory described in Figure 3.15 and all motions are executed with the order delineated by the robot.

```
PROC Path_10()
    SetDO sensor,1;
    MoveL Target_190,v200,z10,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_10,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_20,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_30,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_40,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_50,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_60,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_70,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_80,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_90,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_100,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_110,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_120,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_130,v100,z1,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveC Target_140,Target_150,v100,z1,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_160,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_170,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
    SetDO sensor,0;
    MoveL Target_230,v300,z10,tool_diogo_jnp\WObj:=peca_de_trabalho;
    MoveL Target_240,v100,fine,tool_diogo_jnp\WObj:=peca_de_trabalho;
```

Figure 3.15. Part of the trajectory executed when case 2 is choose in "*execute_option*" (*RobotStudio – Motion* task).

Beyond the motions, other programming lines stand-out such as the "*SetDo*" commands change the value of the variable "*sensor*" digital output, to 0 or 1. It is used to inform when the motions need to be shown in the graphical *HoloLens* environment because there are auxiliary motions that are not part of the workpiece trajectory. When the "*sensor*" value is 1 the motions are shown on *HoloLens*. Otherwise, they are not shown.

As mentioned above, the "*main*" procedure is the first to be executed. According to Figure 3.16 when the persistent variable "*first_verification*" has the value 1 and the variable "*Start_collecting*" takes the value TRUE the digital output "*do_initiate*" is set to 1 and "*do_finish*" is set to 0, while "*Star_collecting*" value is FALSE, both digital outputs will have opposite values from the ones previously mentioned. Consequently, when the digital output "*do_initiate*" is set to 1 an Interrupt Timer (ITimer) is activated while the digital output "*do_finish*" is set to 1 the ITimer is deactivated.

```
WaitUntil first_verification = 1;

IF Start_collecting = TRUE THEN
     SetDO do_initiate,1;
     SetDO do_finish,0;
ENDIF
IF Start_collecting = FALSE THEN
     SetDO do_initiate,0;
     SetDO do_finish,1;
ENDIF

CONNECT timein WITH rec_points;

IF DOutput (do_initiate) = 1 THEN
    ITimer clock_time, timein;
ENDIF

IF DOutput (do_finish) = 1 THEN
    IDelete timein;
ENDIF
```

Figure 3.16. Part of the instruction block of the PROC *"main"* (*RobotStudio* – Motion task).

From the moment that the Itimer is activated, a TRAP routine by the name *"rec_points" is called* as represented in Figure 3.17. This routine stores the robot's coordinates in the variable *"p_1"*. In order to store points with different coordinates, the actual position is compared with the last position. On the other hand, if the vertexes are different each coordinate x, y and z are stored in the variable *"trajectories"* as explained in Figure 3.11. Otherwise, the value is not stored, and the verification will be cyclically repeated after a period stipulated by the user. The variable *"clock_time"* has the value chosen by the user, being 0.1 seconds less than the time supported by *RobotStudio*.

The persistent variable *"cnt"* is a counter that represents the index of the matrix *"rec_path"*.

```
TRAP rec_points

    p1 := CRobt();
    Actual_position := p1.trans;

   IF Actual_position <> Last_position THEN

        Last_position := p1.trans;
        Send_points := TRUE;
        cnt := cnt + 1;

   ENDIF

ENDTRAP
```

Figure 3.17. Trap routine (*RobotStudio* – Motion task).

3.4.2. TCP/IP server task

This task runs in semi-static mode (background), meaning it is hidden from the user. A TCP/IP socket server is implemented with three different functions as seen in Figure 3.18. To create the socket, it is used the function *"SocketCreate"* with the name "*server_socket"*, followed by the "Socket*Bind*" of the "*server_socket" for a specific* IP and Port. The IP address must be from the robot's controller which is running the function "Socket*Listen"* that is responsible for constantly checking for incoming connections. After that the "*server_socket"* starts to act like a server being ready to answer the client's requests, which in this case is the *HoloLens* (MR device).

```
SocketCreate server_socket;
SocketBind server_socket, "172.16.28.172", 2004;
SocketListen server_socket;
```

Figure 3.18. TCP/IP commands (*RobotStudio* – TCP/IP server task).

The communication is done through messages that whenever are sent by the *HoloLens* are stored in variables which search for a match within the available commands and executes according to the respective instruction block. However, if the variable does not match any available command, the system stops and alerts the user.

There is a process to store the variables designed with conditional constructors (statements with “if”) in which the message is separated into a new variable whenever a blank space is found, therefore the message can be split into several variables. For example, if the message received is “read_point” it is required to only store one variable since there are no blank spaces. With this specific message, all the information regarding the point will be read such as position, velocity and precision. On the other hand, if the message is “read_point velocity”, two variables are necessary to store this message and just the velocity of the point will be read, which represents the second stored variable.

There are different types of received messages that can be related to the functioning of the robot such as start or stop and activate or deactivate the motors or requests to the robot make a trajectory.

The message that enables to store the points’ position has the name *“start_sending”* while *“stop_sending”* disables the acquisition of points’ position as shown in Figure 3.19.

```
IF command = "start_sending" THEN

    Start_Collecting := TRUE;
    first_verification:= 1;
    ok_str:=StrToVal(parameter1,clock_time);

ENDIF

IF command = "stop_sending" THEN

    Start_Collecting := FALSE;
    close := 1;

ENDIF
```

Figure 3.19. Instruction block when the message received from *HoloLens* is “start_sending” and “stop_sending” respectively (*RobotStudio* – TCP/IP server task).

3.4.3. TCP/IP client task

The TCP/IP client task works in semi-static mode with resemblance of the previous task. This task works like a client since it is used to send a message with information to the *HoloLens'* TCP/IP server. A crucial detail is that the events are asynchronous, which means the information is sent to *HoloLens* once it becomes available.

When the command *"start_sending"*, shown on Figure 3.19, is received in the robot's controller from the *HoloLens,* the variable *"start_collecting"* becomes TRUE and it creates a socket which connects to the *HoloLens* (it is ready to receive the messages). After the first point being stored, as shown in Figure 3.17 the variable *"send_points"* becomes TRUE, sending a message to the HoloLens with the coordinates of the last available position and the value of the *"sensor"*, that indicates the existence of auxiliary motions. All this process can be seen in Figure 3.20. and the *HoloLens* device will design the trajectory according to the information received.

```
IF Start_collecting = TRUE THEN

        IF init_serv = 0 THEN

            SocketCreate server_unity;
            SocketConnect server_unity,"172.16.21.57",30000;
            init_serv := init_serv + 1;

        ENDIF

        IF send_points = TRUE THEN

             SocketSend server_unity \Str:= ValToStr(Last_position.x) +
             " " + ValToStr(Last_position.y) +
             " " + ValToStr(Last_position.z) +
             " " + ValToStr(DOUTPUT(sensor));
             send_points := FALSE;

         ENDIF
    ENDIF
```

Figure 3.20. Instruction block that creates a client and send information to the TCP/IP socket server in *HoloLens* device (*RobotStudio* – TCP/IP client task)

Whenever the robot controller receives the command *"stop_sending",* represented in Figure 3.19, the variable *"close"* becomes 1, thus the socket closes and the connection is no longer available. The variables are reset to be ready to be used in the next acquisition of points as seen in Figure 3.21. The variable "*Last_position"* is set to coordinates (0,0,0) in order to have a point comparable to the actual position in the TRAP routine (Figure 3.17).

```
IF close = 1 THEN

        SocketClose server_unity;
        init_serv:=0;
        send_points := FALSE;
        Start_collecting := FALSE;
        first_verification:= 0;
        Last_position := [0,0,0];
        cnt := 0;
        close := 0;

ENDIF
```

Figure 3.21. Instruction block that resets the variables (*RobotStudio* – TCP/IP client task).

3.5. Graphical Path Manipulation – RobotStudio

The graphical path manipulation intends users to be able to create free trajectories through "augmented" straight and curve lines in the graphical environment in order to be executed by the robot. The user has the possibility to change specific parameters for each point such as velocity and precision. After the creation of the desired trajectory, a message is sent to the robot to reproduce it, which is later analyzed and the retrieved data is stored. Hence, the robot executes the trajectory according to the user's specifications.

3.5.1. TCP/IP server task

With resemblance to path visualization, the TCP/IP server task receives data from the client, the *HoloLens* device, through a message that contains information regarding the motion created by the user in the graphical environment. As previously explained, the data is retrieved from a single message through its separation by blank spaces making a total of seven different variables. Before splitting the message, it is fully stored in a single variable in order to verify that all elements are present. Although, if such conclusion is not verified the user receives an error message. The first element presented in the described message is the ID of the point which is represented by the index of the variable *"trajectories"*.

The second, third and fourth elements are respectively the x, y and z coordinates. The first vertex (id = 1) is obtained by reading the current position which is considered the initial position of the robot since the first vertex edited by the user is, by default, the tooltip's vertex. The position of the following vertexes is defined through an offset. This means that every new position is calculated through the previous position considering the displacements in x, y and z. Finally, the index of the last vertex must be defined in order to be possible to know the last vertex that the robot controller needs to read. This step is implemented in the last "If statement" of Figure 3.22.

```
IF id = 1 THEN
    trajectories{id}.position := CRobT();
ENDIF

IF id >= 2 THEN
trajectories{id}.position := Offs (trajectories{id-1}.position, new_pointx, new_pointy, new_pointz);
ENDIF

IF last_id < id THEN
    last_id := id;
ENDIF
```

Figure 3.22. Instruction block that stores the position in function of the index (*RobotStudio* – TCP/IP server task)

The fifth element indicates whether the motion is linear or circular through a letter, as represented in Figure 3.23. Whenever this variable corresponds to "L", the motion is linear, thus assigning the value 1 to the element *"motion.type",* whereas if the "*str_motion"* equals either "C" or "CL" the variable "*motion_type*" will have the values 2 or 0 respectively and both motions are circular, requiring both points to define the arc in which "C" represents the middle point and "CL" the final point of the circular movement. Therefore, whenever the fifth element of the message corresponds to "C" the same element of the next message will be equal to "CL". However, if the received fifth element has the value "CL" the correspondent value of "*motion_type*" will not be used since the next movement could be either linear or circular but it is still important to avoid communication errors.

```
IF str_motion = "L" THEN
    trajectories{id}.motion_type := 1;
ENDIF

IF str_motion = "C" THEN
    trajectories{id}.motion_type := 2;
ENDIF

 IF str_motion = "CL" THEN
    trajectories{id}.motion_type := 0;
ENDIF
```

Figure 3.23. Instruction block with each type of motion (*RobotStudio* – TCP/IP server task).

The sixth element is velocity which is composed by four elements: velocity of the tool's center point, orientation, linear external axes and rotational external axes. Amongst those four components the velocity of the tool's center point is the only one that is established by the user and corresponds to the velocity that the robot's tool moves. The remaining components are established by default as shown in Figure 3.24.

```
trajectories{id}.velocity := [velocity,500,5000,1000];
```

Figure 3.24. Instruction block with the parameter velocity (*RobotStudio* – TCP/IP server task).

Finally, the seventh element is precision, which represents how close the tool should be to the intended point before moving to the next position. As shown in Figure 3.25 there are a few examples of precisions defined and automatically recognized by the system. For example, the first case means the precision chosen by the user is 0 millimeters.

```
TEST precision

CASE "0": trajectories{id}.precision := z0;
CASE "1": trajectories{id}.precision := z1;
CASE "5": trajectories{id}.precision := z5;
```

Figure 3.25. Part of the instruction block with the parameter precision (*RobotStudio* – TCP/IP server task).

Every time a message is received a confirmation is sent the *HoloLens* device as seen in Figure 3.26.

```
SocketSend client_socket1 \Str:= "Data Received";
```

Figure 3.26. Instruction block with the message "Data Received" sent to the *HoloLens* after a successful message being received by the robot controller (*RobotStudio* – TCP/IP server task).

3.5.1. Motion task

The motion task runs in normal mode resembling the path visualization. The data retrieved from the task described in the previous subchapter is used in this task for the robot to be able to execute the trajectories created by the user in the holographic environment. The procedure "*free_path*", present in Figure 3.27, has a loop that runs until the last vertex. As previously described the "*motion_type*" could have the value either 1 or 2 according to the motion type which can respectively be linear or circular, thus both the velocity and precision are established by the user.

```
PROC free_path()

    FOR i FROM 1 TO  last_id DO

        If  trajectories{i}.motion_type = 1 THEN

            MoveL trajectories{i}.position,
                  trajectories{i}.velocity,
                  trajectories{i}.precision,
                  tool_diogo_jnp \WObj:= tool_center;

        ENDIF

        IF trajectories{i}.motion_type = 2 THEN

            MoveC trajectories{i}.position ,trajectories{i+1}.position,
                  trajectories{i}.velocity,
                  trajectories{i}.precision,
                  tool_diogo_jnp \WObj:= tool_center;

        ENDIF

    ENDFOR

ENDPROC
```

Figure 3.27. Instruction block *"free_path"* procedure (*RobotStudio* – Motion task).

4. RESULTS DISCUSSION

The work developed was tested through the robot cell presented in Figure 3.1, in a laboratory environment. The results are positive since both apps are functional and can be implemented in the industry. After a detailed analysis some problems were identified and overcoming those shortcomings is part of the future work.

4.1. Object Tracking

With regards to the object tracking, the accuracy obtained it is not enough, which can be due to the placement of the initial point not being exactly in the tooltip and consequently, all the trajectory is affected.

There are several conditions that affect the accuracy. Firstly, the object tracking is very sensitive to the illumination conditions. The tool identification was executed from different angles and perspectives concluding that the side used to do the object tracking must have enough incident light in order to make a correct identification. It is to be noted that the light cannot be excessive to not overshadow the tool. Other factors responsible for a correct identification are the tool's shape and angle as well as higher amount of its details which leads to a faster object tracking.

Since the *model target* is a mechanism created by *Vuforia,* it is necessary to wait for improvements or try to find new mechanisms to do the object tracking. For now, the identification is suitable to do tasks without considerable precision.

4.2. Path Visualization

The path visualization seems to have a great accuracy as illustrated in Figure 4.1. From the comparison between the trajectory programmed in RobotStudio and the trajectory displayed in the holographic environment, it is perceptible that all details of the workpiece are successfully achieved with great accuracy, even the circles which are the most difficult part. The main handicap, in this app, is the cycle time that the *RobotStudio* allows to acquire points. Some applications can require a lower cycle time to reproduce trajectories with higher precision. To reproduce the holographic workpiece in Figure 4.1, the lowest cycle time allowed was used: 0.1 seconds. This means that at each 0.1 seconds a point is stored. With a time thus brief, a non-overloaded network is required. Otherwise, the trajectory will present defects.

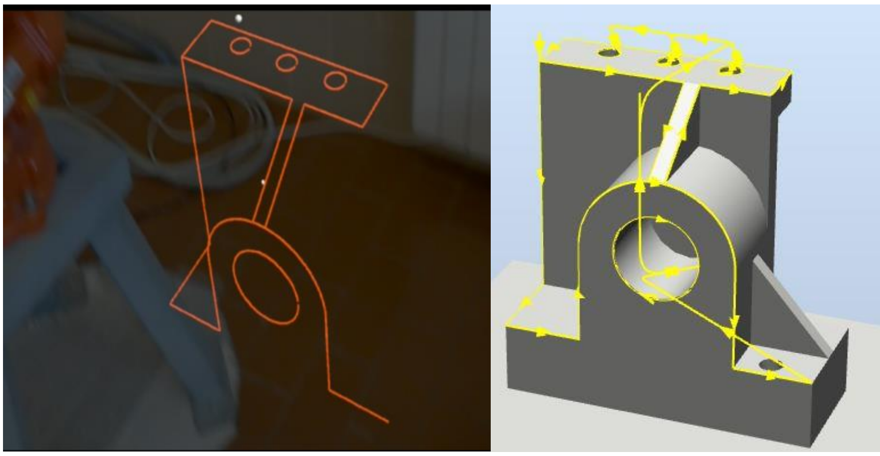

Figure 4.1. Workpiece designed in the holographic environment (*HoloLens* view) – left. Workpiece with programming paths (*RobotStudio*) – right.

4.3. Graphical Path Manipulation

The graphical path manipulation executes the path designed by the user with high accuracy and with resemblance of the path visualization, a non-overloaded network is crucial. During the tests, it became perceptible that unreachable points could be created by the user. It is intended to create a mechanism to avoid these points, alerting the user that the point is unreachable. This mechanism will be an improvement that can avoid errors not mentioned while the trajectory is being executed.

5. CONCLUSION

In this dissertation, the concepts Mixed Reality, Augmented Reality and Virtual Reality were introduced. These concepts are emerging between several areas of society from education to the industry. In the robotics' field, the correct application of these technologies can improve the product's quality and the user's safety. Robots' programming will be no longer a problem since the devices will be prepared to show how to perform certain tasks. Mixed Reality fits perfectly in the industry 4.0 opening a new range of apps that can be implemented in the industry. With Mixed Reality supplying more information to operators, the cooperation between machines and humans will be much better.

A system implementation of two apps that could be introduced in the industry were also presented. The first app, Path Visualization, shows the "augmented" trajectory in a *HoloLens* device, made by a robot. The required communication with the robot's controller was implemented through the TCP/IP protocol and the positions of the robot were stored in a matrix with a certain frequency.

The second app, Graphical Path Manipulation, allows users to create a trajectory that will be reproduced by the robot. The data is received by the robot's controller and organized into several variables that are stored in a matrix. This data is used to form the motions, which the robot executes, with the respective parameters defined by the user such as velocity and precision.

Common to both apps, the object tracking is the initial phase. Through the identification of the tool, the initial vertex is positioned in the tooltip and the trajectory is designed from it. Without this function, the trajectory could be incorrectly overlaid in the workpiece.

The main challenge was the network overload that interfered in the performance accuracy in both apps. In order to successfully achieve the object tracking it is required good illumination conditions as well as the right-angle detection. Finally, a mechanism of unreachable vertexes needs to be designed to avoid the creation of trajectories that are not feasible by the user.

The *HoloLens* device is a powerful tool that has several advantages such as spatial mapping which can be introduced in the industry. On the other hand, the device is expensive which opens a window of opportunities to adapt these apps to other devices like tablets and smartphones.

Finally, the proposed goals were successfully accomplished which can be corroborated through the video (Serrario, Neves, & Pires, 2018) that presents both apps performing in the laboratory robot cell. The next proposed step would be to apply the apps to the industrial robotics.

BIBLIOGRAPHY

ABB Robotics. (2015). Technical reference manual, RAPID Instructions, Functions and Data types, 1546.

Bottino, A., Ingrassia, P. L., Lamberti, F., Salvetti, F., Strada, F., & Vitillo, A. (2017). Holo-BLSD: an Augmented Reality self-directed learning and evaluation system for effective Basic Life Support Defibrillation training. *18th International Meeting on Simulation in Healthcare (IMSH 2018)*, (4), 7–8.

Chen, I. Y.-H., MacDonald, B., & Wunsche, B. (2009). Mixed reality simulation for mobile robots. *ICRA 2009: IEEE International Conference on Robotics and Automation*, (June), 232–237. https://doi.org/10.1109/ROBOT.2009.5152325

Dirican, C. (2015). The Impacts of Robotics, Artificial Intelligence On Business and Economics. *Procedia - Social and Behavioral Sciences*, *195*, 564–573. https://doi.org/10.1016/j.sbspro.2015.06.134

Li, P., & Chu, C. (2016). Augmented Reality based Robot Path Planning for Programming by Demonstration Augmented Reality based Robot Path Planning for Programming by Demonstration. *Design Engineering Workshop*, (December), 1–5.

Michalos, G., Karagiannis, P., Makris, S., Tokçalar, Ö., & Chryssolouris, G. (2016). Augmented Reality (AR) Applications for Supporting Human-robot Interactive Cooperation. *Procedia CIRP*, *41*, 370–375. https://doi.org/10.1016/j.procir.2015.12.005

Milgram, P., & Kishino, F. (1994). A Taxonomy of Mixed Reality Visual-Displays. *Ieice Transactions on Information and Systems*, *E77d*(12), 1321–1329.

Rosenberg, L. (1992). The use of virtual fixtures as perceptual overlays to enhance operator performance in remote environments. *Dtic.Mil*, (July 1992). Retrieved from http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA292450&Location=U2&doc=GetTRDoc.pdf

Serrario, D., Neves, J., & Pires, N. (2018). *Application of Mixed Reality in Robot Manipulator Programming*. Retrieved from https://www.youtube.com/watch?v=EWVPy2d39II

Sousa, L., Alves, R., & Rodrigues, J. M. F. (2016). Augmented reality system to assist inexperienced pool players. *Computational Visual Media*, *2*(2), 183–193. https://doi.org/10.1007/s41095-016-0047-3

Statista. (2018). Forecast - Augmented and Virtual Reality. Retrieved June 3, 2018, from https://www.statista.com

Sutherland, I. E. (1968). A head-mounted three dimensional display. *Proceedings of the December 9-11, 1968, Fall Joint Computer Conference, Part I on - AFIPS '68 (Fall, Part I)*, 757. https://doi.org/10.1145/1476589.1476686

Vaidya, S., Ambad, P., & Bhosle, S. (2018). Industry 4.0 - A Glimpse. *Procedia Manufacturing*, *20*, 233–238. https://doi.org/10.1016/j.promfg.2018.02.034

Vuforia. (2018a). Model Targer Generator User Guide. Retrieved February 19, 2018, from https://library.vuforia.com/articles/Solution/model-target-generator-user-guide.html

Vuforia. (2018b). Model Targets User Guide. Retrieved February 19, 2018, from https://library.vuforia.com/articles/Solution/model-targets-user-guide.html

WyzLink. (2018). *Mixed Reality Robotic control*. Retrieved from https://www.youtube.com/watch?v=QtFvLXbJrwA

Zeller, M., & Brandon, B. (2018a). Gestures - Microsoft HoloLens. Retrieved June 4, 2018, from https://docs.microsoft.com/pt-pt/windows/mixed-reality/gestures

Zeller, M., & Brandon, B. (2018b). What is mixed reality? Retrieved June 2, 2018, from https://docs.microsoft.com/pt-pt/windows/mixed-reality/mixed-reality