A NOVEL DESIGN OF VIRTUAL AND MIXED REALITY SCENARIOS FOR AUTOMATION TRAINING

A Thesis

Presented to

the Faculty of the College of Business and Technology

Morehead State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Andrés Salinas-Hernández

Ap+ril 23, 2021

Accepted by the faculty of the College of Business and Technology, Morehead State University, in partial fulfillment of the requirements for the Master of Science degree,

____
Jorge Alberto Ortega-Moody
Director of Thesis

Master's Committee: ______________________________, Chair
Jorge Alberto Ortega-Moody

____
Ahmad Zargari

____
Kouroush Jenab

____
Date

A NOVEL DESIGN OF VIRTUAL AND MIXED REALITY SCENARIOS FOR AUTOMATION TRAINING

Andrés Salinas-Hernández
Morehead State University, 2021

Director of Thesis: ____
Jorge Alberto Ortega-Moody

The manufacturing sector in the United States has greatly benefited from the usage of virtual reality technologies primarily for the last ten years, by allowing a more accessible integration between already developed operations and designs by manufacturers and virtual reality (VR) systems, making the resulting simulations' performances much more fluid and realistic. One of the newest subcategories in VR technology called Mixed Reality (MR), incorporates devices like 3D depth cameras and green screen video captures to stream the user's VR stream around him, as this stream can be seen by third-party observers, which allows for a more compelling experience. This research examines the implementation of both VR and MR platforms of a flexible manufacturing prototype scenario for teaching purposes. The thesis aims to present the pros and cons of using each platform, as the user manipulates its surroundings and interacts with tracking devices that are modeled into objects inside the scenario.

Accepted by: ______________________________, Chair
Jorge Alberto Ortega-Moody

____
Ahmad Zargari

____
Kouroush Jenab

ACKNOWLEDGMENTS

The author would like to express his gratitude to all the professors and students who have reviewed this thesis and gave constructive ideas to improve its quality.

First of all, I would like to express my appreciation and admiration to all members of this thesis committee. To Dr. Ortega-Moody, who guided me not only alongside my professional development but also through my personal development and adaptation to a different lifestyle, and whom I thank for allowing me to expand the horizons for my creativity and knowledge to find ways in which I could enrich every project in which he involved me into. To Dr. Jenab, who kept me pushing towards a consistent pursuit of academic breakthroughs and trusted me as well with confidence in the projects we were involved. And last but not least, to Dr. Zargari for believing in my potential and for all his professional and personal advice during my stay at Morehead State University.

Secondly, I would like to thank my family and friends who stood with me on the good and bad days and gave me their support amidst difficulties faced because of the global pandemic faced since early 2020.

Special thanks are addressed to the Kentucky National Science Foundation (NSF), whose Experimental Program to Stimulate Competitive Research (EPSCoR) provided the appopiate finanlcial and technological supports for the full development of this research.

Finally, to life itself for teaching me the roughest lessons to become a better human being, especially teaching me that even the hardest challenges can be achieved.

TABLE OF CONTENTS

TABLE OF FIGURES

1. CHAPTER ONE: INTRODUCTION

1.1 Background

Automation technology is considered to be more than a discipline, an application of processes and equipment thanks to human control, even though this greatly dismisses human intervention. It is considered that automation behaves like a bond between the mechanization of a process and the computerization that it is required to determine the behavior of it. One important aim of automation technology is that these processes can be efficient and met with a low error rate.

The intrinsic relationship between automation and manufacturing is based on several aspects. Many types of industries today require highly automated products and processes to meet their demands because, as society grows naturally, it increasingly seeks to have better conditions around it and thus contributes to competitiveness between companies. Similarly, as there are more and more professionals training in the different branches related to automation technologies, the industry will seek to take advantage of this by improving its processes, valuing indices such as quality and execution time. Other aspects in which automation may benefit the manufacturing industry are increases in productivity, reduction in labor costs and manual tasks, the improvement of safety measures, etc.

Given the importance of industrial automation for manufacturing companies, it is important that they consider the implementation of training techniques in these technologies for their employees, considering various factors such as prior knowledge in the handling of automatic devices that each company or process uses or intends to implement and the financial resources available for personnel training. At this point, automation training in industrial automation technologies has some key disadvantages: it tends to be an internal resource that very few companies or organizations can afford, and among those that have the economic availability to do so, automation training tends to be a process limited to certain personnel and requires considerably high time demand. Something important to mention is that, due to global crisis situations like the SARS-CoV-2 pandemic, the ability to train these employees in a face-to-face environment is gravely affected.

For these reasons, the development of a virtual scenario-focused simulation of automation technologies training is proposed. The expected result of the development is that researchers will be able to present users with the same concepts required to handle automated industrial equipment used in areas like manufacturing in a safer environment that can offer flexibilities like remote monitoring of the data gathered from the virtual equipment and compatibility with physical devices like programmable logic devices (PLC) to enhance the learning process.

1.2 Purpose

The development of virtual scenarios focused on automation technologies implemented in the area of manufacturing is not something new. Since the development of virtual Reality and augmented reality devices and platforms, several researchers have sought ways in which specific tasks or processes can be adapted with the use of these devices. Many previous works have focused on the optimization of processes in real time or on the variation of different models or scenarios in which they allow the user to visualize the results prior to a real implementation of said process. As technological capabilities become more and more complex in an effort to execute more precise instructions, the people behind the development of virtual scenarios and technologies must deal with more complete and flexible execution models, so that with each new advance, the end-user manages to experience an interaction not so different from the real handling of the simulated processes.

The proposed design of this research contemplates the use of different key features of the virtual experience to allow multiple interaction levels between the user and the surrounding environment, as well as with external elements in the shape of hardware and software that uses the virtual data to establish a connection with ongoing physical processes, similar to the realistic way in which, on an industrial level, data can be shared between different platforms using industrial communication protocols.

1.3 Research Objectives

Objective 1:

To implement both VR and MR technologies of a flexible manufacturing prototype scenario for automation training purposes by integrating different levels of immersion.

Objective 2:

To integrate industrial communication protocols that allow the virtual scenario to establish a connection with physical automation devices for parallel applications.

1.4 Assumptions

The following assumptions are considered as pre-conditions for this research.

The virtual reality scenario corresponds to a CNC manufacturing prototype cell located inside the Morehead State University's School of Engineering & Computer Science Virtual Reality laboratory. The reason for this specific system to be modeled is that the physical elements and devices found on it offer students of the different engineering-related courses to understand basic automation concepts easily, and the overall virtual modeling process for these elements fits under the estimated time of development for the rest of the activities of this project.

The virtual reality scenario's main purpose is to offer the user the opportunity to recreate the same activities of the physical prototype. The behavior of each element inside of the scenario matches their real-life counterpart, thus allowing a proper way in which the operation knowledge gets transferred onto the virtual environment.

1.5 Limitations

The simulated prototype is considered to offer the user the ability to manipulate elements in order to follow an automated sequence of events similar to the physical manufacturing cell. Certain objects inside the scenario can and need to be handled by the user, but there are restrictions as to what the user can and cannot interact. These restrictions are incorporated as warnings inside the environment in order to guide the end-user through the right set of actions.

1.6 Definition of Terms

Virtual Reality:

"The use of computer technology to create an interactive three-dimensional world in which the objects have a sense of spatial presence; virtual environment, and virtual world are synonyms for virtual reality" (Defense Modeling and Simulation Enterprise, 2020).

Augmented Reality:

"A type of virtual reality in which synthetic stimuli are registered with and superimposed on real-world objects; often used to make information otherwise imperceptible to human senses perceptible " (Defense Modeling and Simulation Enterprise, 2020).

Mixed Reality:

"A blend of physical and digital worlds, unlocking the links between human, computer, and environment interaction. This new Reality is based on advancements in computer vision, graphical processing power, display technology, and input systems" (Microsoft, 2020).

PLC:

"(Programmable Logic Controller) An industrial computer used to automate manufacturing, industrial, and other electromechanical processes. PLCs are different from common computers in that they are designed to have multiple inputs and output arrays and adhere to more robust specifications for shock, vibration, temperature, and electrical interference, among other things" (Schneider Electric, 2019).

HMI:

"(Human-Machine Interface) The hardware or software through which an operator interacts with a controller. An HMI can range from a physical control panel with buttons, and indicator lights to an industrial PC with a color graphics display running dedicated HMI software" (U.S. Department of Commerce, 2015).

CPU:

"(Central Processing Unit) Is the portion of a computer that retrieves and executes instructions. The CPU is essentially the brain of a CAD system. It consists of an arithmetic and logic unit (ALU), a control unit, and various registers. The CPU is often simply referred to as the processor. The ALU performs arithmetic operations, logic operations, and related operations, according to the program instructions" (Rosato & Rosato, 2003).

HMD:

"(Head Mounted Display) The current form of hardware delivering virtual reality experiences to users. It is typically in the form of goggles strapped to the head. Integrated with either a mobile phone or display and custom lenses, it is through the headset that the user can view different virtual reality content" (Facebook, 2020).

GPU:

"(Graphics Processing Unit) Is a specialized processor originally designed to accelerate graphics rendering. GPUs can process many pieces of data simultaneously, making them useful for machine learning, video editing, and gaming applications. GPUs may be integrated into the computer's CPU or offered as a discrete hardware unit)" (Intel, 2021).

FOV:

"(Field of view) Is the angle of degrees in the user's visual field within a headset. Having a higher field of view is important because it contributes to the user having a feeling of immersion in a VR experience. The bigger that angle is, the more immersive it feels" (Facebook, 2020).

JSON:

"(JavaScript Object Notation) Is a text syntax that facilitates structured data interchange between all programming languages. JSON is a syntax of braces, brackets, colons, and commas that is useful in many contexts, profiles, and applications" (ECMA International, 2017).

Immersion:

"The sensorimotor contingencies available within a virtual environment, that it, the physical actions required within a specific environment to perceive and interact with a given environment" (Alinier, Young, Farra, & Kardong-Edgren, 2019)

OLE:

"(Object Linking & Embedding) A distributed object system and protocol from Microsoft, also used on the Acorn Archimedes. OLE allows an editor to "farm out" part of a document to another editor and then reimport it" (FOLDOC , 1998).

OPC:

"(OLE for Process Control, later changed to Open Platform Communications) Is an industrial connectivity standard that enables the transfer of automation data between automation hardware and software. The goal of OPC is to make it possible for a software application to access automation data from any control and/or monitoring system, regardless of its vendor" (OPC Training Institute, 2021).

TCP/IP

"(Transmission Control Protocol/Internet Protocol (TCP/IP)) Is the basic communication language or protocol of the Internet. The TCP protocol is responsible for an error free connection between two computers, while the IP protocol is responsible for the data packets sent over the network. The TCP/IP Internet protocol suite developed by the US Department of Defense in the 1970s" (OPC Training Institute, 2021).

Modbus Protocol:

"Is a messaging structure developed by Modicon in 1979. It is used to establish client-server communication between intelligent devices. It is a de facto standard, truly open and the most widely used network protocol in the industrial manufacturing environment. It has been implemented by hundreds of vendors on thousands of different devices to transfer discrete/analog I/O and register data between control devices" (The Modbus Organization, 2021).

Ladder Diagram:

"Ladder diagrams are specialized schematics commonly used to document industrial control logic systems. They are called "ladder" diagrams because they resemble a ladder, with two vertical rails (supply power) and as many "rungs" (horizontal lines) as there are control circuits to represent" (Oliver, 1991).

2. CHAPTER TWO: REVIEW OF LITERATURE

2.1 Virtual Reality

VR is currently considered to be a technology that evolved exponentially thanks to its natural tendency to be a pleasing interactive environment experience. Its adaptability to a wide range of problems and domains by allowing the user to perceive a close-to-reality world in which not only sight but also smell, touch, and hearing are controlled by high-performance computers made it quickly break the erroneous idea that it was meant to be a fantastic, futuristic and short-lived idea born out of science fiction (dated as far back as 1935 on Stanley G. Weinbaum's "Pygmalion's Spectacles").

One of the most important concepts to understand within the virtual reality field is the reality-virtuality continuum, which is represented by a scale that ranges from what is considered physical and visually "real" to what is considered "virtual." The continuum is best represented by a two-dimensional plane of virtuality and reality, with the origin R denoting unmodified Reality and two axes, M and V, representing changes. Figure 1 (Nincarean, Bilal Ali, & Abd halim, 2013) illustrates the continuum.

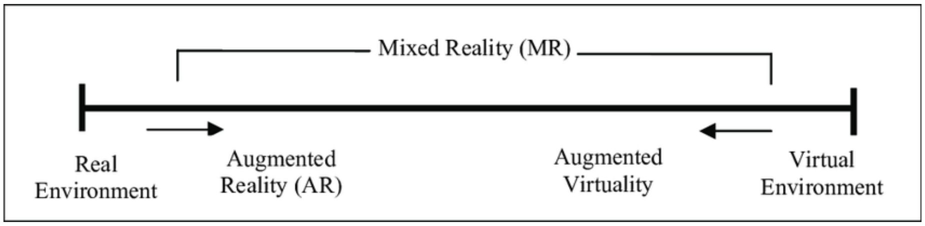


Figure 1. Reality - Virtuality continuum

Four points result within this plane: augmented Reality, augmented virtuality, mediated Reality and mediated virtuality. Furthermore, this model allows the emergence of the concept of "mixed reality", a merging point of both worlds that results in their particular environments and visualizations, as shown in Figure 2 (Flavián, Ibáñez-Sánchez, & Orús, 2018).

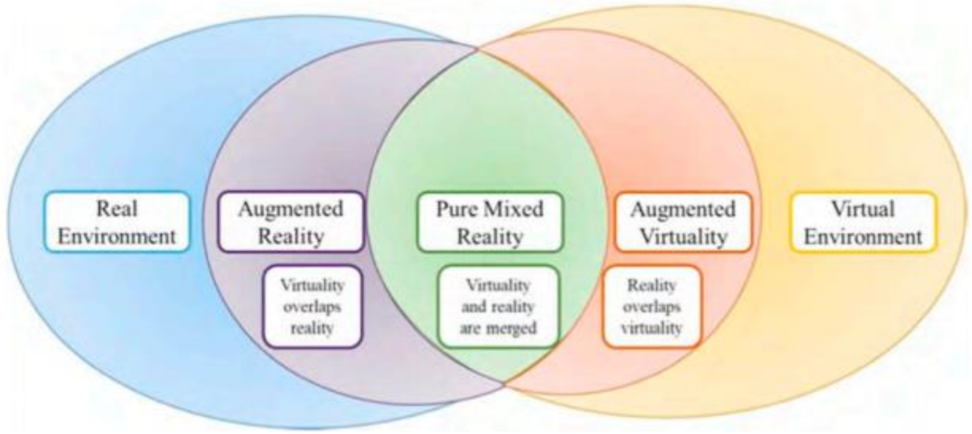


Figure 2. Exemplification of the mixed reality concept

For the purpose of this document, Virtual Reality refers to the experience that allows a person to be immersed in a simulated environment provided by a device called a head-mounted display (HMD). Although something relatively simple to understand, it is important to make this clarification of the VR term, given the existence of different levels of immersion, which in turn can diversify the concept of what VR is. Although there is not a unified scheme, a valid proposed model of VR's immersion levels is presented as follows in Figure 3 (McCarthy, 2020):

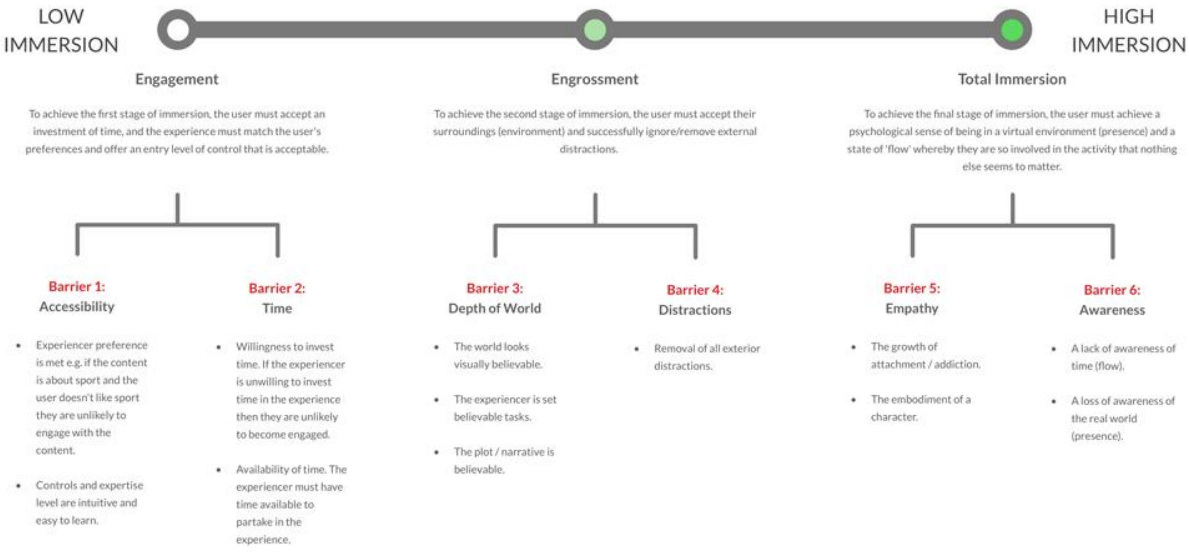


Figure 3. Different levels of immersion

2.1.1 Components of Virtual Reality

When one considers what a "basic" VR setup should look like in the current year, especially given how quickly new accessories are developed and how the concept of VR gets reshaped, many different opinions are encountered. Therefore, experts in the area sometimes struggle to establish normativity on what these components should be. One example of this can be observed in the differences between the following diagrams corresponding to Figure 4 (Burdea & Coiffet, 2003) and Figure 5 (Bamodu & Ye, 2013):

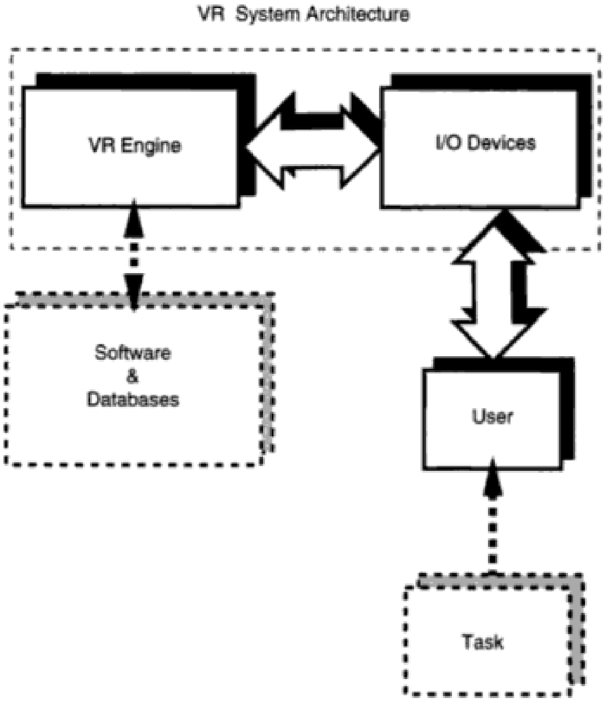


Figure 4. Early architecture for VR systems

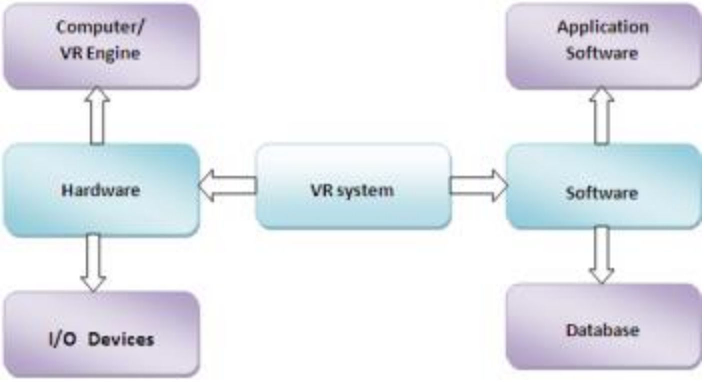


Figure 5. Example of a newer VR system architecture

It is worth noticing, for example, how, currently, the "application software" element is contained within the rest of the software involved in the VR system. How ten years change the perspective of what should be considered "basic" is indeed interesting from the point of view of standardizing concepts.

For the purpose of this document, the latter diagram is chosen to enumerate and briefly describe the VR elements employed.

2.1.1.1 Head-Mounted Display (HMD)

One of the main components of any VR system, if not the main component, is the head-mounted display (HMD). This device mostly consists of a helmet that includes a visor that simulates binocular vision, using each eye to transmit slightly different images of the same object with a slight offset between each other. The HMD offers diverse features to help the user enhance the quality of the VR environment, like toggling between different levels of resolution, adjusting the available field-of-view and the tightness of the helmet to the user's head. A deconstructed view of the HMD can be observed on Figure 6 (Yole Développement, 2020):

Figure 6. Part collection for a Vive® VR HMD

With the fast development of VR technology in the last few years, there is already enough diversification of HMDs that it is possible to sort them into two main categories:

2.1.1.1.1 Tethered HMD

Tethered HMDs are those that make use of a special wired connection to the hosting CPU, with some special cases in which each HMD include their own computing unit. Tethered HMDs usually contain a standalone display inside with special electronics, an optical lens, and a tracking system that can register the user's movement in either 3 or 6 DoF, making use of technologies like gyroscopes accelerometers, magnetometers, etc. An example can be observed in Figure 7 (VIVE VR, 2021):

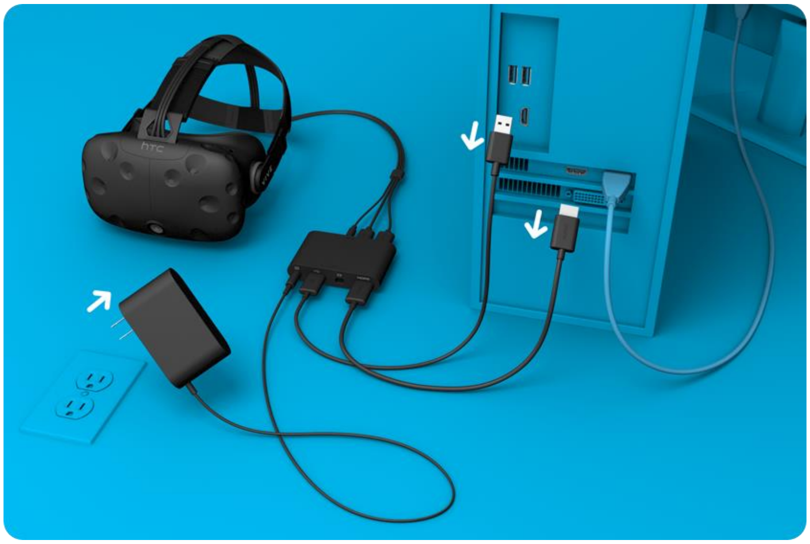


Figure 7. Wiring a tethered VIVE® VR HMD

The main advantage of using tethered HMDs comes when the user/developer intends to pursue quality over simplicity: by having dedicated displays and electronics, these devices offer better high-quality images and environmental tracking that enhance the user's immersion and performance within the VR scenario. The cost, obviously, is that the user's real-life movement capabilities are limited to the total length of the HMD cable. In addition to this, different tethered HMD brands differ in the total number of connections that they need to receive in order to be powered up and start the VR projection.

2.1.1.1.2 Mobile HMD

On the other hand, mobile HMDs are mostly designed as helmets that do not need a physical wired connection to a hosting PC or computing unit. Given that, today, wireless HMDs are a tendency that most developers seek, this type of HMD is not limited to VR but can also make use of AR applications. Therefore, mobile HMDs differ according to the software used on them:

- Smartphone VR: These include mostly the lens and a socket in which another device, like a smartphone, is introduced while already having a pre-built app that acts as the VR environment, such as in Figure 8 (Raffaele, 2017).
- Wireless VR: Some companies have incorporated wireless capabilities onto some of their helmets, allowing users to keep a higher quality transmission while not sacrificing mobility within the workspace.

Figure 8. The Google® Daydream HMD

In the case of HMDs that make use of smartphones, the main disadvantage is that the performance and image quality greatly depends on the type of phone used. Therefore, while the mobile HMD's price is considerably lower than tethered options, the budget increases if the user desires higher quality according to the selected smartphone. Furthermore, wireless VR HMDs can sometimes be affected by the performance of the transmitter/receiver devices that are adapted onto it.

2.1.1.2 Input Devices

The second-largest branch of equipment used in VR environments includes all other devices used to provide data and control signals into the scenario's information processing system. Within the past three to four years, there has been a rapid emphasis on more active experiences and environments for the end-user to interact with, which in turn demands faster and more responsive/intuitive equipment. This is important, given that these traits determine the speed at which one can navigate through all the scenario's pre-established actions.

VR Input devices can be classified according to the corresponding type of input provision for the HMD:

2.1.1.2.1 Controllers

Like most video game consoles, virtual reality setups were initially conceived as entertainment for people, and thus, most of the early versions of these were designed with similar controllers as the then-available market-established consoles and some arcade machines. This was mostly because the movements of the user were more limited. (Some early scenarios could only allow forwards/backward and left/right displacement.)

As technology progressed, translation of user/character movement from real-life models to a virtual environment happened in a more natural way, thus forcing developers to look for ways in which controllers could replicate these commands. The response came in the way of 3DoF/6DoF controllers, the latter being capable of adding more interactive options with the elements of the VR environment; these include properties like integrated trackers and buttons like grippers and triggers. Examples of these can be seen in Figure 9 (Raffaele, 2017):

Figure 9. Examples of a 3 DoF and a 6 DoF controller

2.1.1.2.2 Tracking Device

A tracker is a device that, like its name implicitly says, detects and follows a given coordinate position of an object. Translated into a VR environment, this device identifies and transmits the position of HMDs and the user's location within the scenario and sends it towards a receiver device plugged into the hosting PC.

Tracking devices widely depend on the type of coordinate system that is being used inside the scenario, so it is highly important for the developer to properly calibrate these devices on all of its possible positions and orientations before attaching it to the user or desired object.

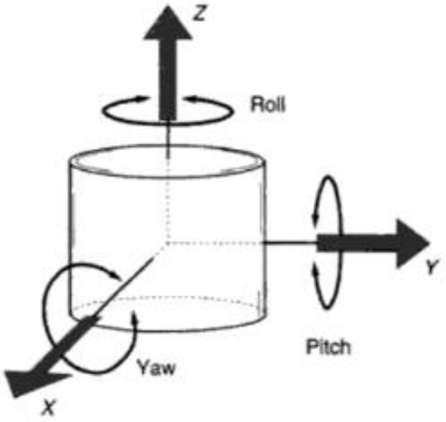


Figure 10. Coordinate systems for tracking devices (Burdea & Coiffet, 2003)

Parameters like accuracy, drift, and latency are considered necessary in the development stage. The more accurate a tracker or the lower latency it has can have a significant influence on the chosen device for a specific scenario. Such correlation between parameters can be seen in Figure 11:

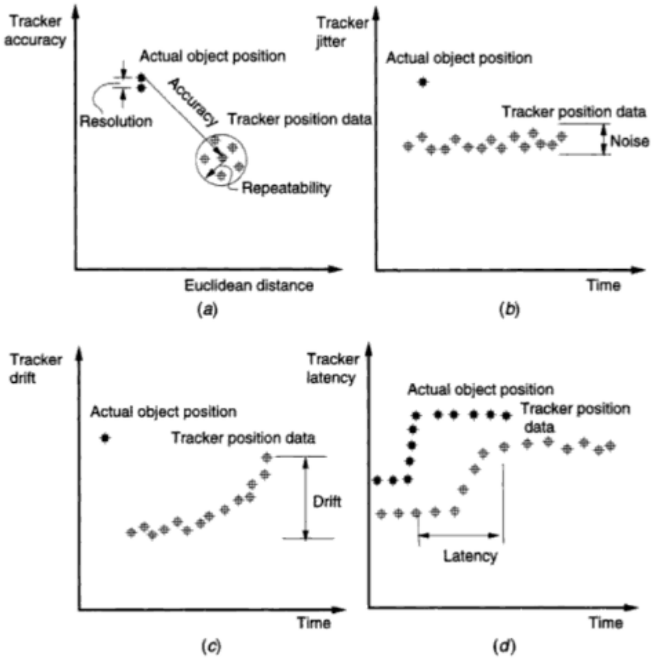


Figure 11. Correlation graph between tracking devices' parameters (Burdea & Coiffet, 2003)

Several types of trackers are available currently in the VR market. Most of these trackers include wireless capabilities and low latency periods. These also can vary according to the body part on which the trackers are mounted:

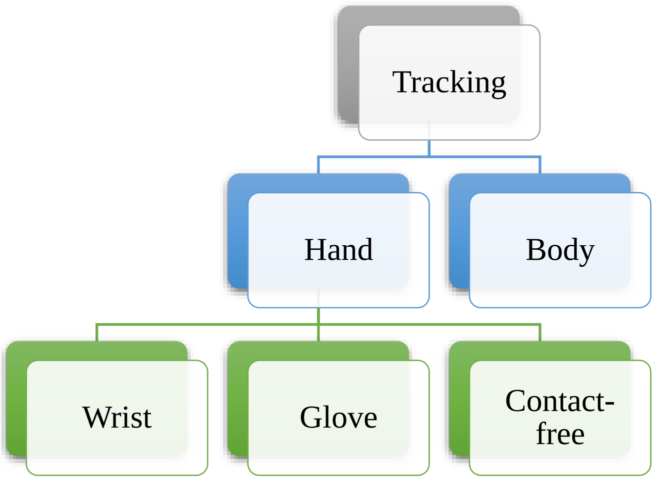


Figure 12. Types of VR trackers according to body location

Figure 13. Examples of hand VR tracking devices (Raffaele, 2017)

2.2 Augmented Reality

When looking at the most common implementations of Virtual Reality in the present day, it is most likely that one could stumble upon certain applications that do not benefit from the fact that the user is immersed into an environment completely different from the real-life objects that surrounds him. Indeed, a lot of information tends to be overlooked in most processes, so while VR environments shine in emulating complicated and critical actions in a safer and more controlled way, they lack the ability to enhance the perception of its typical environment (Lu, Xu, & Wang, 2020).

Augmented Reality is now considered a significant area of research in the field of VR. This technology takes advantage of the wealth of information and data that are capable of being obtained from the environment that surrounds the user, allowing a more practical and graphical display, rather than being presented in a numerical way or in the form of files or databases. Strictly, AR is "an enhanced version of reality created by the use of technology to overlay digital information on an image of something being viewed through a device" (Novak-Marcincin, Barna, Janak, & Novakova-Marcincinova, 2013). The main objective of image interposition is to create a system in which the user can improve the perception of the system being manipulated using as much information or data as possible.

AR systems, unlike VR systems, use HMDs that incorporates a translucent screen, in which the overlay has an effect. Projectors located on both sides of the device generate the virtual environment in the form of holograms, using depth sensors to place the computer-generated images according to what is observed into the user's field of vision. A main difference between these devices compared to VR HMDs is that the former allows the user to identify hand gestures to interact with the environment, thus avoiding the use of special controls. Although providing a much lower-end experience, smartphones represent the most common distribution channel for AR, given how easy these holographic designs can be used alongside current camera technology. In a typical AR system, the camera in a smartphone is used to scan a specially-designed marker, such as a QR barcode or image, which then calculates the camera's position in relation to the environment and projects visual AR content through the screen.

The software used for the creation and development of AR applications is actually the same one used by VR environments since the main difference lies in the plug-ins or scripting that are intended to be used according to the HMD. The main difference is that these environments require a camera to be able to interpret the environment that surrounds the user, so it is also important that the software includes the ability to recognize and incorporate the video stream from said device.

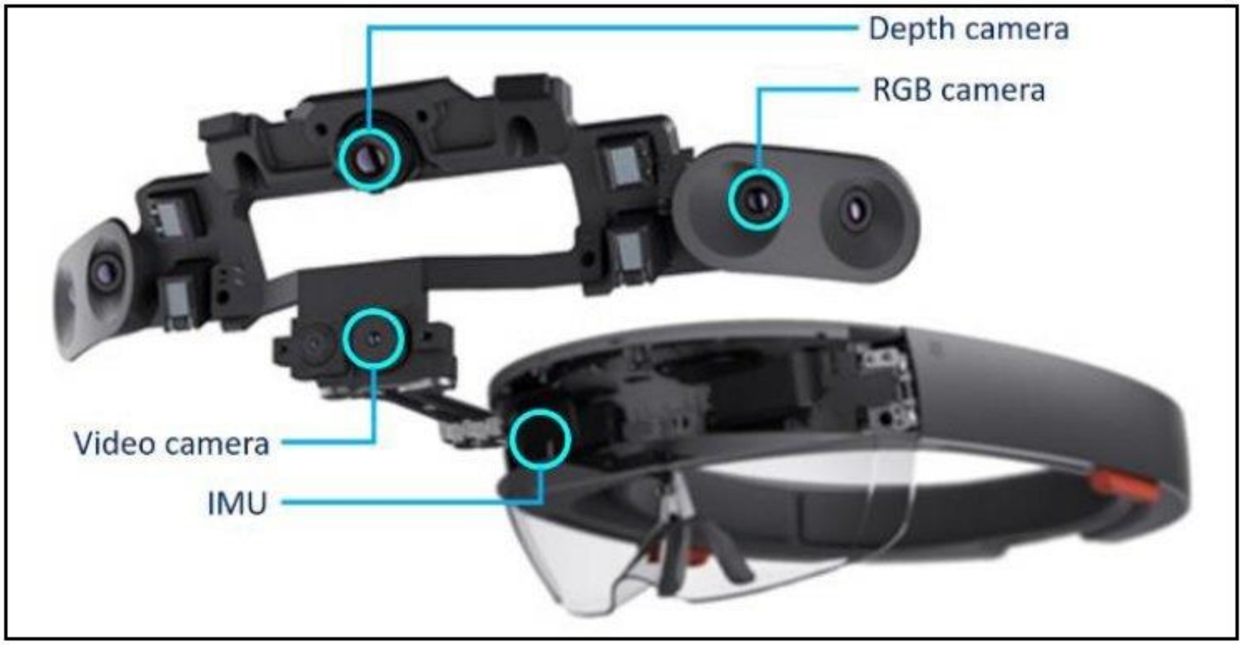


Figure 14. Built-in sensors of the Microsoft® Hololens HMD (Khoshelham, Tran, & Acharya, 2019)

2.3 Mixed Reality

Although the concept of Augmented Reality is wide enough to cover a large number of technologies that enhance the perception of the user's surroundings, it is important to backtrack to the Milgram et al. explanation of the wide spectrum covered by Reality and virtuality. Everything that is located within both extremes of the spectrum can be described as mixed Reality (MR). Even up to the current day, many developers and researchers still strongly debate about the main differences between the discrepancies of AR and MR concepts.

For the purposes of this project, mixed Reality (MR) refers to the intertwining of the virtual world and the physical world at a high level. The main focus on MR environments varies according to the multiple experiences that tend to differ from user to user, including the level of immersion, interaction, number of users and environments, etc. After careful considerations, it was established that the "level of virtuality" dimension had a heavy weight on the decision to consider "mixed reality" as the right framework.

2.3.1 Components of Mixed Reality

Similar to AR, most mixed reality applications make use of HMDs, which contain a screen that composes information without obfuscating the environment in front of the user; this is achieved through the use of a holographic projector mounted on the sides of each eye. In other cases, however, the composition requires that additional elements like stereo cameras and green screens combine to produce a more immersive experience, not only for the user but for other external observers.

2.3.1.1 Stereo Camera

Stereo cameras (or stereoscopic cameras) are image-capturing devices that have two or more lenses, with separate image sensors located on each lens. Each of the lenses displays different images, which then are sent to a processor that applies a perspective rendering method to match the scene projection point-of-view to the user's position. The resulting effect allows the observer to notice changes in depth and placement of the captured objects, exemplified by Figure 15 (Chotrov, Uzunova, Yordanov, & Maleshkov, 2018).

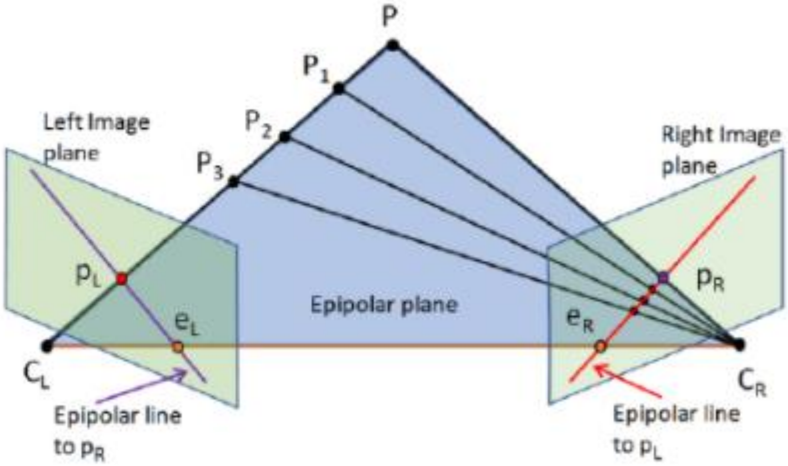


Figure 15. Epipolar geometry of stereo vision

A stereo camera reproduces the way human binocular vision works. They incorporate the camera sensors usually within a 6 to 12 cm separation, thus allowing the capture of 3D video and, more importantly, the estimation of both depth and motion.

The depth maps that are created from the stereo cameras are created by capturing a distance value for each of the pixels in the images. This distance is internally calculated from the furthest position of the camera sensors to the actual object in the scene. In a similar way, stereo cameras can create 3D point clouds, which are nothing more than a "collection of 3D points that represent the external surface of the scene, and can contain color information" (Stereo Labs Inc., 2020).

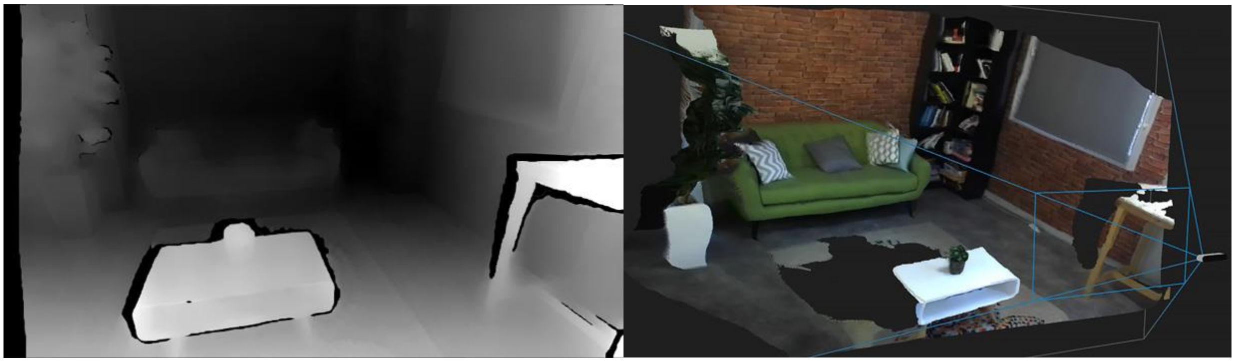


Figure 16. The depth and point-cloud maps

2.3.1.2 Green Screen

Another method for image composition used not only in mixed Reality, but also in many other media that involves movement capturing is chroma key compositing (or chroma keying). This visual-effect technique makes use of a single-color background that, after the image or video is captured, is replaced with another image or video recorded separately. These colors tend to be, in most cases, either blue or green, given the fact that they are the furthest from a person's skin tone, shown as an example in Figure 17 (Stereo Labs Inc., 2020).

Figure 17. Example of a green screen background capture in MR

2.4 Automation Technologies

Automation as a concept is as complicated to define as virtual Reality. Over the coming years, the idea of automated systems has confused many people that have struggled to distinguish up to which point a certain device or operation could be considered "autonomous" or without human aid. With the exponential technological progress accomplished from the 1940s and 1950s onwards, automation was known as a solution for quality improvement and productivity increase among different industries around the world by switching human involvement towards supervisory roles and cognitive tasks.

In the simplest of terms, automation can mean "the execution by a machine agent of a function that was previously carried out by a human" (Parasuraman, 2000). However, it can also refer to the collective range of technologies capable of carrying out these actions. From here, there are a plethora of important concepts that revolve around automation, such as automated control (the systems that manage the logic and instructions for automated devices) and control lops (measurement of values and the comparison with a preset, which in turn is sent to the control system to determine the best course of action).

2.4.1 Sequential Control

In the current industrial field, it is more common to observe a wide arrange of machinery and sectors that rely on automated work with little to no human interaction. This is mostly accomplished by an equally large number of devices in charge of guiding pre-established instructions (most of them inputted by humans) in very specific notation (variables) that constantly modify or maintain numerical or verbal values that these machines recognize.

There are different control schemes that can be implemented in automated systems; some of these are even used in conjunction if needed. Some systems are relatively simple enough to employ abrupt on-off controls when the desired value is met, while other, more complex ones use arithmetical calculations to compensate the continuous error obtained to increase or decrease proportional, integral, and derivate terms in order to maintain stability.

However, for the purposes of this thesis, the focus shifts to a type of control that considers the execution of different tasks depending on various system states. This is commonly referred to as sequential control or system state control, exemplified by Figure 18 (University of Ovideo, 2007). The states refer to the diverse and sometimes specific conditions that can happen in a sequence where one or many elements of a system are involved.

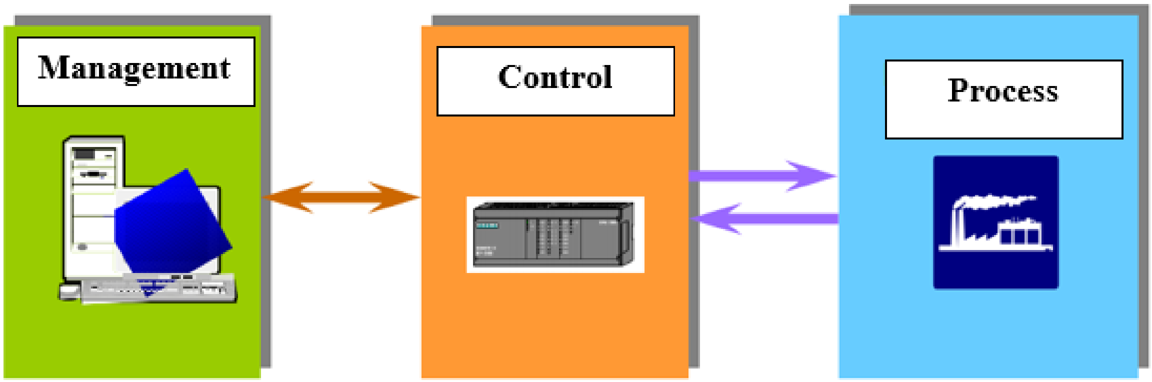


Figure 18. Elements of sequential control systems

2.4.2 Programmable Logic Controller (PLC)

In many industrial applications, the most common way to achieve total control over automated processes is by the aid of programmable logic controllers (PLC). A PLC is "a digitally operated electronic device, which uses a programmable memory for internal storage of instructions to implement specific functions, such as logic, sequencing, recording and control of times, counting and arithmetic operations to control, through digital input/output modules (ON / OFF) or analog (1-5 VDC, 4-20 mA, etc.), various types of machines or processes" (National Electrical Manufacturers Association, 2005).

The field of application of PLCs is exceedingly diverse and includes various types of industries (e.g., automotive, aerospace, construction, etc.), as well as machinery. Unlike general-purpose computers, the PLC is designed for multiple inputs and output signals, wide temperature ranges, immunity to electrical noise, and resistance to vibration and shock. The programs to control the operation of the machine are usually stored in backup batteries or in non-volatile memories.

A PLC is an example of a hard-real-time system where output results must be produced in response to input conditions within a limited time, which will not otherwise produce the desired result.

Figure 19. PLC main and secondary modules (Siemens, 2012)

Among the advantages that PLCs offer is the fact that they make it possible to carry out operations in real time, due to their reduced reaction time. In addition, they are devices that easily adapt to new tasks due to their flexibility when programming them, thus reducing additional costs when preparing projects.

They also allow immediate communication with other types of controllers and computers and even allow network operations. They have a stable construction, as they are designed to withstand adverse conditions such as vibrations, temperature, humidity, and noise. They are easily programmable through quite understandable programming languages, like ladder diagrams, sequential function charts (SFC), function block diagrams (FBD), etc. However, they have certain disadvantages, such as the need for qualified technicians to take care of their proper operation.

2.4.2.1 General Structure of a PLC

In order for the PLC to work, it needs a power supply whose main purpose is to guarantee the internal operating voltages of the controller and its blocks. The most frequently used values are ± 5V, ± 12V, and ± 24V, and there are mainly two power supply modules: those that use an input voltage from the main module and those that use operational power supplies to control the objects.

The main part is the central processing unit (CPU) which contains the processing part of the controller and is based on a microprocessor that allows the use of arithmetical and logical operations to perform different functions, as illustrated in Figure 20 (Farrukh, Halepoto, Chowdhry, Kazi, & Lal, 2017). In addition, the CPU also frequently tests the PLC to find errors in due time.

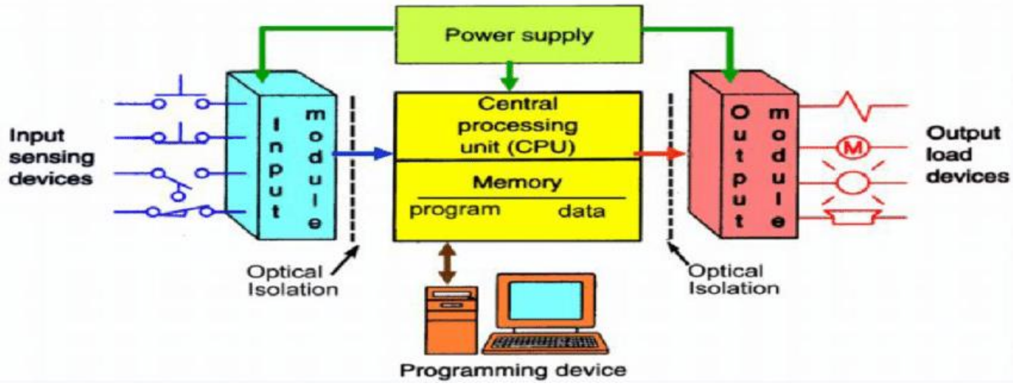


Figure 20. Basic structure of a PLC

The place where data and instructions are stored is a memory that is divided into the permanent memory (PM) and operational memory, also known as random access memory (RAM). The first, PM, is based on ROM, EPROM, EEPROM or Flash; it is where the PLC operating system runs and can be replaced. However, the RAM is where the program in question used is saved and executed, and it is the SRAM type that is usually used.

Finally, the inputs and outputs (I/O) modules are those signal modules (SM) that coordinate the input and output of the signals with those internal to the PLC. These signals can be digital (DI, DO) and analog (AI, AO), and they come from or go to devices such as sensors, switches, actuators, etc. Analog SMs generally use direct current (DC) voltage and a direct current. In this way, optocouplers, transistors, and relays are used in the digital output of the SMs to change the states of the output signal in order to protect these devices from situations such as a short circuit, an overload, or an excessive voltage.

A PLC receives and transfers electrical signals, thus expressing finite physical variables (temperature, pressure, etc.). Therefore, it is necessary to include a signal converter in the SM to receive and change the values to physical variables. There are three types of signals in a PLC: binary, digital, and analog signals.

2.4.3 Industrial Communication Protocols

On many occasions, the concept of "industrial communication protocol" has been heard, but to what it refers to has not been clarified. In order to understand this concept, it is important to understand the role that communication plays in today's highly technological and modern industry. Inside most factories today, many systems are made up of equipment from different manufacturers and operate at different levels of automation. Due to this, and also due to the fact that most factories have their different operating areas considerably distanced from each other, it is often desirable to work in a coordinated way for a satisfactory result of their processes. The concept of industrial communication protocol then becomes "a method for digital data communications between two or more devices in different locations, or on a network" (InduSoft, 2017).

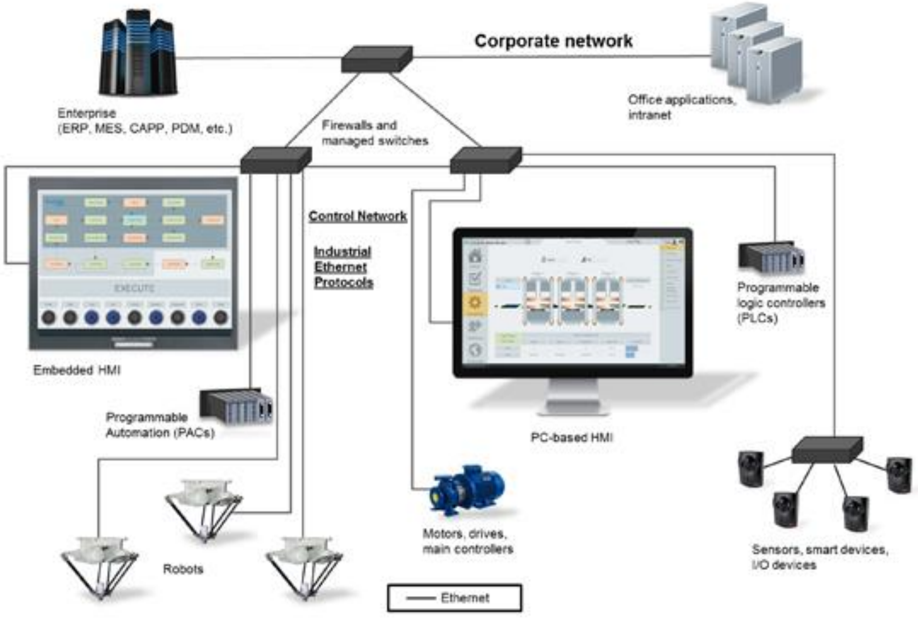


Figure 21. Example of networked devices through Ethernet communications

This type of communication between systems has been used essentially for instrumentation equipment and systems where a low data transfer rate between equipment is necessary, but in a large number of cases today, it can no longer respond to the needs of intercommunication between devices that are demanded. The advantages provided by industrial communication protocols, among others, are:

- Visualization and supervision of production processes
- Quick or instantaneous acquisition of process data

- Improvement of the general performance of the entire process
- Possibility of data exchange between sectors of the process
- Remote programming

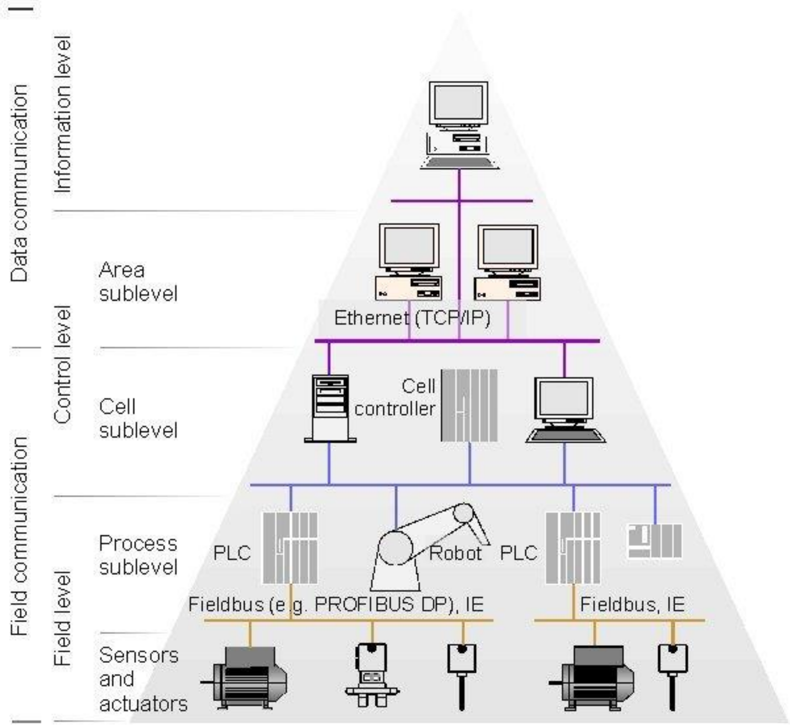


Figure 22. Hierarchy of an industrial automation system (Belai & Drahoš, 2009)

2.4.3.1 Fieldbus

Each of the types of industrial communication protocols must have particular characteristics in order to respond to the needs of intercommunication in real-time. In addition, they must withstand a harsh environment where there is plentiful electromagnetic noise and harsh environmental conditions. In the use of industrial communications, two main areas can be separated: communication at the field level and communication towards SCADA (Supervisory Control and Data Acquisition). In both cases, the data transmission is carried out in real-time or, at least, with a delay that is not significant concerning the process times and can be critical for the field level.

A Fieldbus is, in general lines, "a system of field devices (sensors and actuators) and control devices, which share a bidirectional serial digital bus to transmit information between them, replacing the conventional point-to-point analog transmission." They allow the replacement of the wiring between sensors/actuators and the corresponding control elements. This type of bus, exemplified in Figure 23, must be low cost, with minimum response times, allow serial transmission over a digital data bus with the ability to interconnect controllers with all kinds of simple input-output devices, and allow intelligent slave controllers (Pimentel & Schneider, 2013).

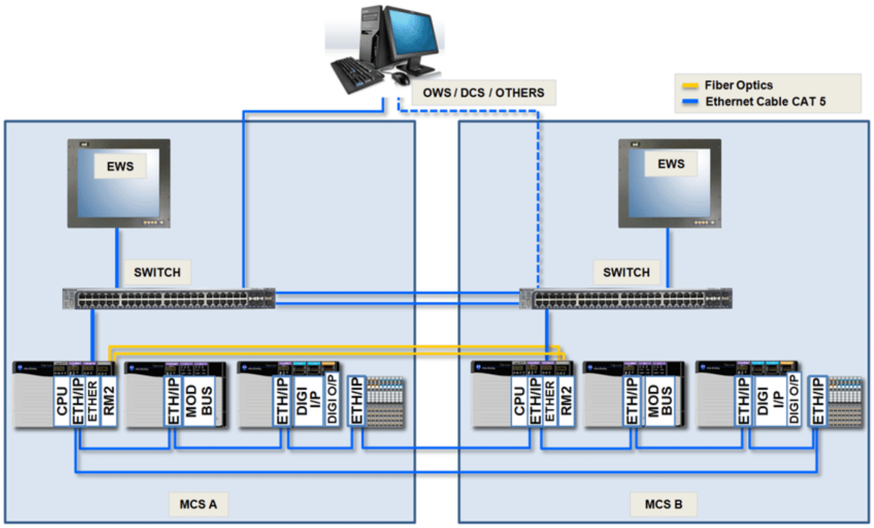


Figure 23. Network topology with Ethernet/IP

When the distance between the instrument and the control system becomes considerable or when many instruments are present in the process, one must consider factors like wiring costs, especially when the need for a large number of reserve drivers is established. For these reasons is that the Fieldbus philosophy is more widely implemented. With this system, it is possible to replace large bundles of conductors with a simple two-wire or fiber optic cable, common to all sensors and actuators, with the consequent economic savings that this entails. The communication of the process variable is entirely digital.

2.4.3.2 OPC Communications

As has been seen, there is a wide variety of industrial communication protocols available for various types of applications and needs. However, one of the most relevant and flexible protocols in the design of automated virtual systems is the OPC protocol.

The OPC (OLE for Process Control, later Open Platform Communications) is a communication standard in the field of industrial process control and supervision, based on Microsoft's Object Linking & Embedding technology, which offers a standard interface for communication, allowing individual software components to interact and share data. OPC communication is carried out through a client-server architecture. It is an open and flexible solution to the classic problem of proprietary drivers. Virtually all of the major manufacturers of process, instrumentation, and control systems have included OPCs in their products.

2.4.3.2.1 OPC Architecture

OPC uses a client-server approach to communication. The OPC server is in charge of encapsulating the information and making it available through its interface, while the OPC client connects to the OPC server and accesses the available information. In classic OPC, the interfaces are based on Microsoft's COM and DCOM technology, while in the latest version of the standard of OPC UA, two protocols are used: a high-performance binary TCP protocol (OPC-TCP) and a second based in web services (HTTP). This network configuration can be seen in Figure 24 (Mahmoud, Sabih, & Elshafei, 2015).

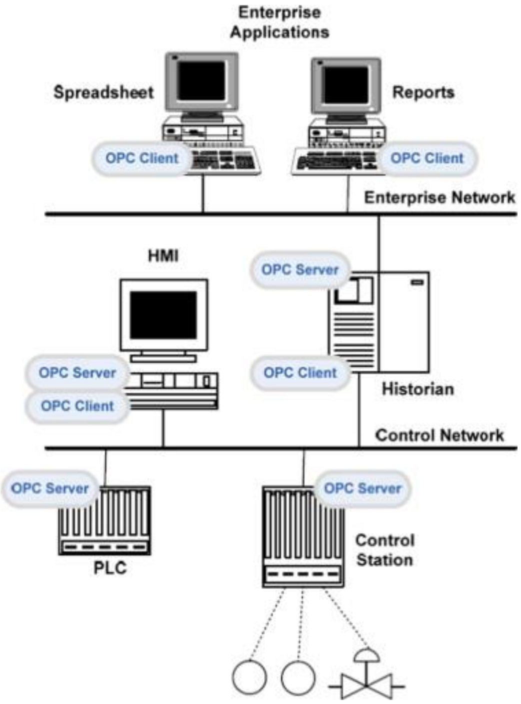


Figure 24. Topology of an OPC protocol network

Classic OPC provides the standard specifications for Data Access (DA), Historical Data Access (HDA), and Alarms and Events (A&E). These OPC specifications are widely accepted in the automation industry. Classic OPC, which is based on Microsoft's old COM/DCOM1 technology, has led to the development of new specifications known as OPC UA (Unified Architecture) (Gutiérrez-Guerrero & Holgado-Terriza, 2019).

The main objective of OPC UA is to maintain the functionality of the classic OPC and to move from Microsoft's COM/DCOM technology to state-of-the-art services technology. Using web service technology, OPC UA becomes platform-independent and can therefore be applied in situations where classic OPC is no longer used. OPC UA can be seamlessly integrated into manufacturing enterprise systems (MES) and enterprise resource planning (ERP) systems and works not only on Unix/Linux systems with Java but also on drivers and smart devices that have specific operating systems capable of real-time operation. Of course, compatibility with previous OPC specifications is a requirement for OPC UA.

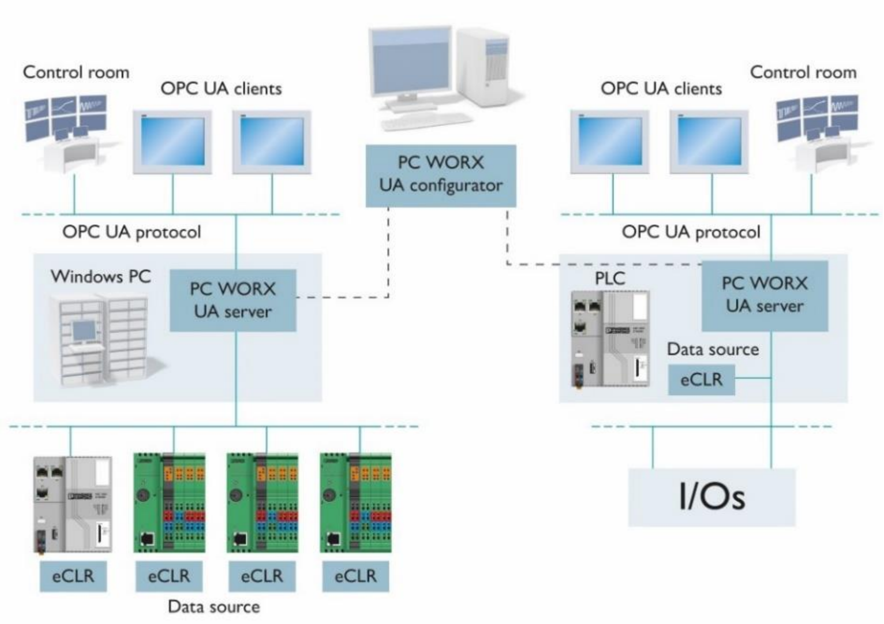


Figure 25. Topology of an OPC UA network

3. CHAPTER THREE: METHODOLOGY

3.1 Research Design

The main objective of this project consisted of the design of a virtual environment in which a flexible manufacturing prototype for automation training can be displayed and interacted with by incorporating a mixed reality layout. One of the first steps in the realization was to identify the main problem, which was found in a prototype training manufacturing cell located within the Virtual Reality Laboratory at Morehead State University (Figure 26).

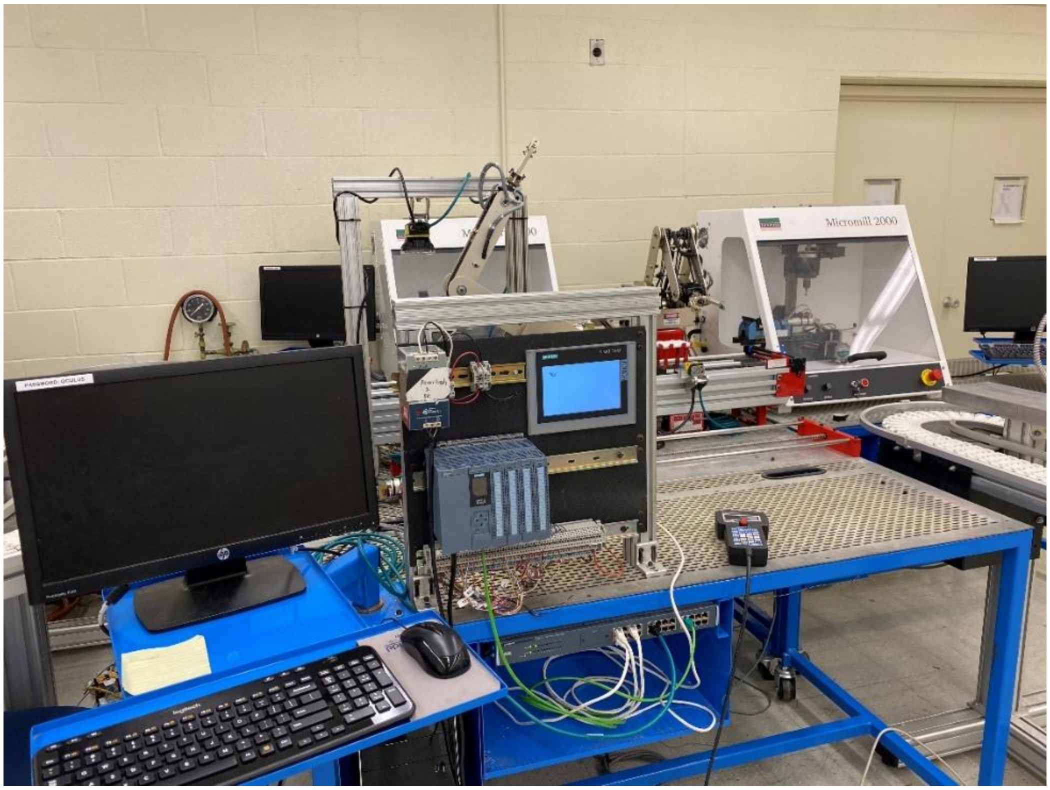


Figure 26. Manufacturing training prototype located in the VR Laboratory

The type of research design followed was one of the descriptive (case-study) types. The main for this is that the object of study required a degree of manipulation, not only in its physical components but also in the virtual model obtained. In addition to this, one of the main results was translated into the appreciation of the differences between the design based on Virtual Reality and Mixed Reality. However, the study did involve a degree of experimentation, given the fact that many controllable variables affected the outcome of important dependent variables.

3.2 Setup Development Environment

In order to better understand the basis of the project, a brief description of the composing elements of the prototype must be included.

3.2.1. CNC Machines

Computer Numerical Control (CNC) machines are computer-controlled devices enabled to treat materials such as wood, foam, plastic, medium-density fiberboard (MDF), metal, among many others. The emergence of CNC machines has optimized the technical process of industrial and artistic creation. These devices also automate machine sequences and techniques that allow the creation of parts that manually could not be done.

The sequence in which a CNC machine typically executes its commands in order to manufacture the desired material and produce a component is described as following:

- A part program is written using specific codes that were standardized for machine control, called G and M codes, which tell the machine where to move and to describe the sequence of operations that the machine must perform to manufacture the component.
- The part program is then loaded onto the machine's computer, called a controller. In this stage, the CNC software allows the program to be edited or graphically simulated in order to get a complete preview of the finished product.
- The controller then processes the part program and sends signals to the machine in order to direct the machine through the required sequence of operations.

The training manufacturing prototype includes two different CNC machines produced by Denford.

3.2.1.1 Lathe

The CNC lathe is a two-axes CNC training machine tool that is designed for turning synthetic material such as wax, plastics, acrylics, and non-hardened metals such as aluminum. In each of these cases, the appropriate tooling, spindle speeds, and feed rates should be used as recommended by the material supplier (Denford, 2003).

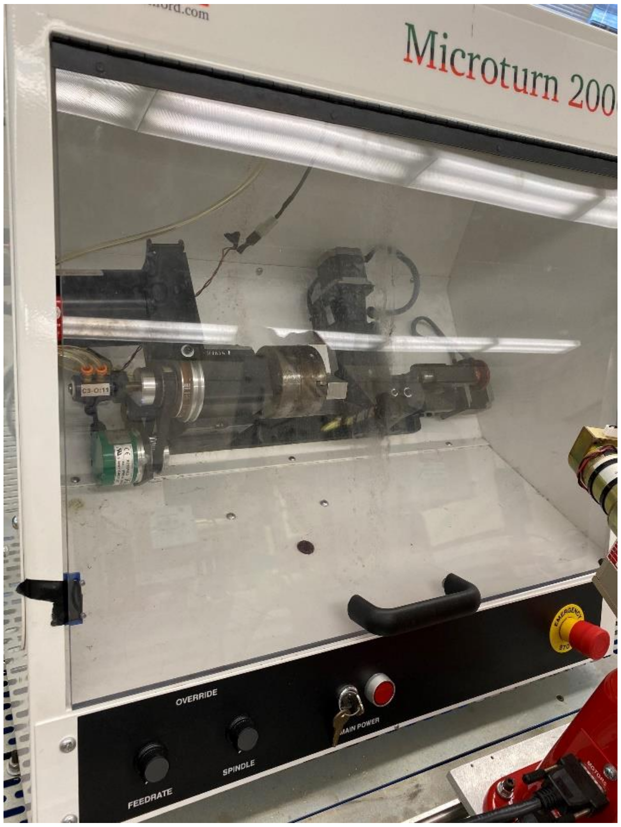


Figure 27. Detail of the Microturn lathe tool

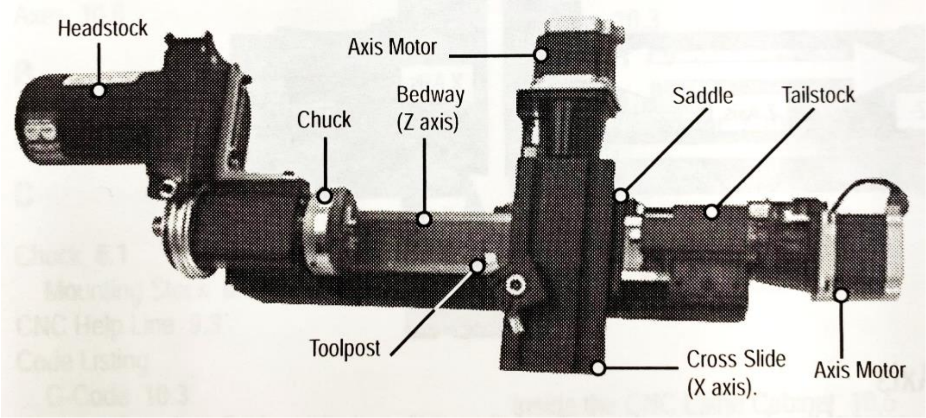


Figure 28. Machine parts of the lathe

3.2.1.2 Mill

The CNC mill is a full three-axis CNC training machine tool designed for milling synthetic material such as wax, plastics, acrylics, and non-hardened metals such as aluminum. In each case, the appropriate tooling, spindle speeds and feed rates, should be used as recommended by the material supplier (Denford, 2003).

Figure 29. Detail of the training prototype's mill

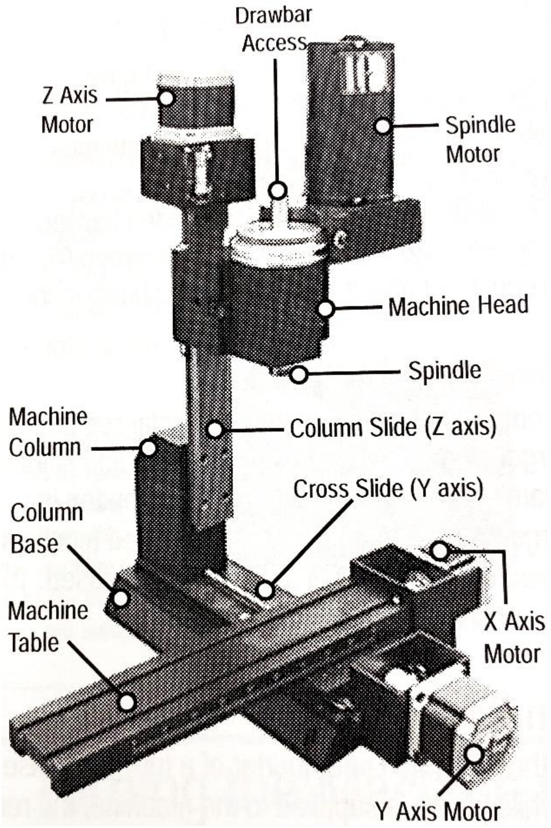


Figure 30. Machine Parts of the mill

3.2.2 Robotic Arm

Although the definition for "robot" is as complex and can vary among different authors, most of them would agree that it can be considered as an electromechanical device that can emulate motions similar to a human being. On that note, a robotic arm can be defined as an articulated electric servo system designed to replicate the articulations of a human arm.

Robotic arms are fabricated with industrial-grade components and are composed of articulations, called joints, that can move the device in various translational and rotational points (axes). Typically, robotic arms include a final actuator in the shape of a gripper or a tool, which is used to interact with other external objects, depending of the purpose of use of the robotic arm.

The robotic arms located within the training prototype are the Amatrol Pegasus II robotic arms, and they move around five different joints or axes, as well as a translational axis by being mounted on top of a linear conveyor. The robotic arms also include a teach pendant, which works as a programmable handheld device used to teach and store movement points for the robot, as well as its own controller with independent inputs and outputs.

Figure 31. One of the two Pegasus robotic arms

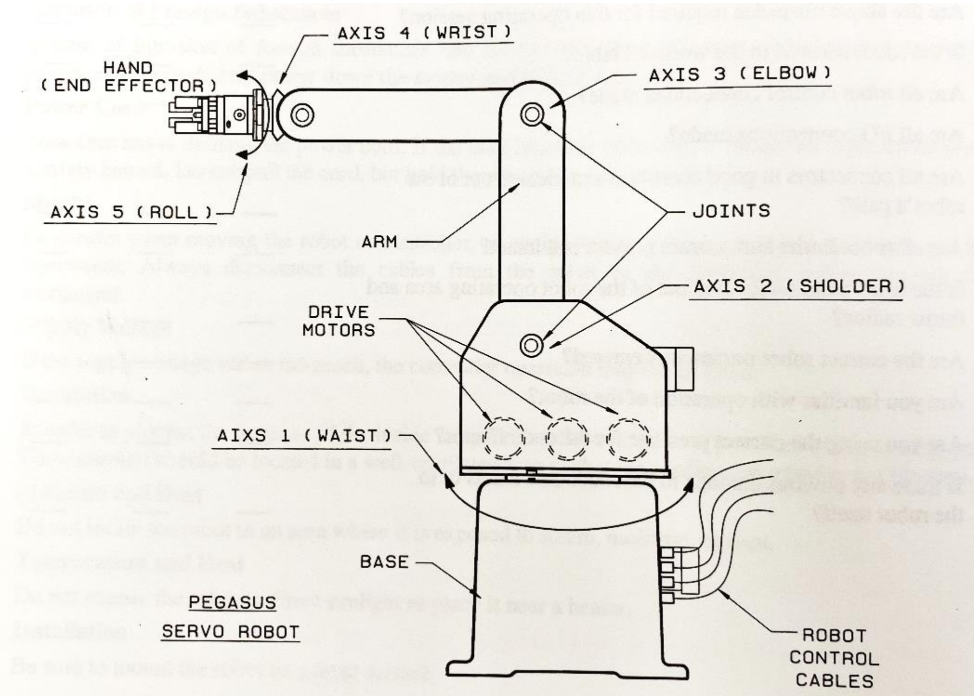


Figure 32. Locations of the different robotic arm axes

3.2.3 Actuator

An actuator is a device that converts energy into motion or is used to apply force. The device takes energy from a certain source (which can be energy created by air, liquid, or electricity) and converts it into the desired movement. The two types of basic movement desired are linear and rotary, but oscillatory movement is also common.

Linear actuators work by converting energy into linear movements, which are used for pushing or pulling. Rotary actuators, on the other hand, convert energy into oscillatory movements and are generally used in different valves, such as butterfly or ball valves.

In the training prototype, there are different types of actuators. Each of them performs a very specific task within the system.

3.2.3.1 Conveyor Belt

The system has two conveyor belts, which move along a system of drums driven by a motor. Materials placed on top of the belt move from point to point while the belt rotates around the drum in the opposite direction to maintain uninterrupted movement.

Conveyor belt systems are generally designed according to the type of movement desired and can have cyclical, finite, linear paths, etc. The elements that are located on top of the different conveyor belts correspond to supports for different types of pieces. In the case of the manufacturing training prototype, the belts are designed for parts in the form of blocks and cylinders.

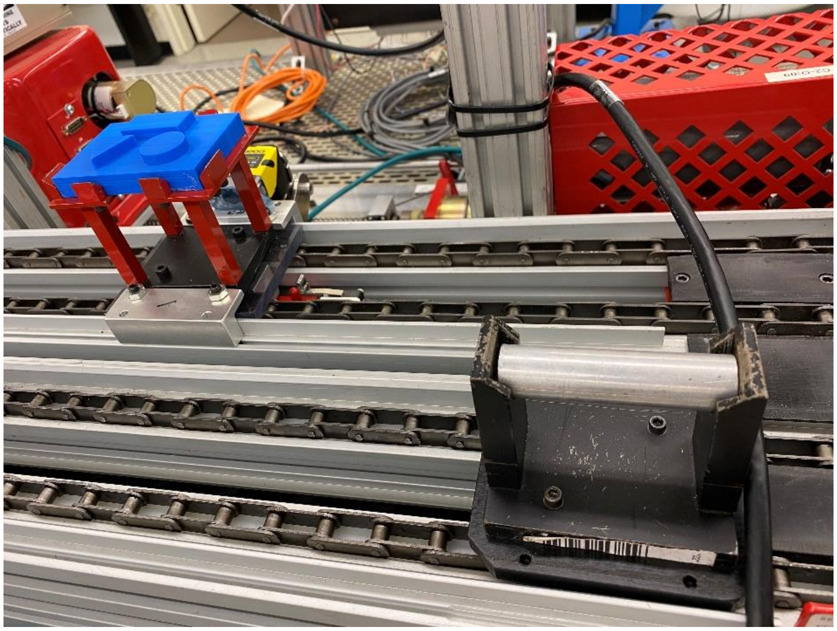


Figure 33. Two types of holders

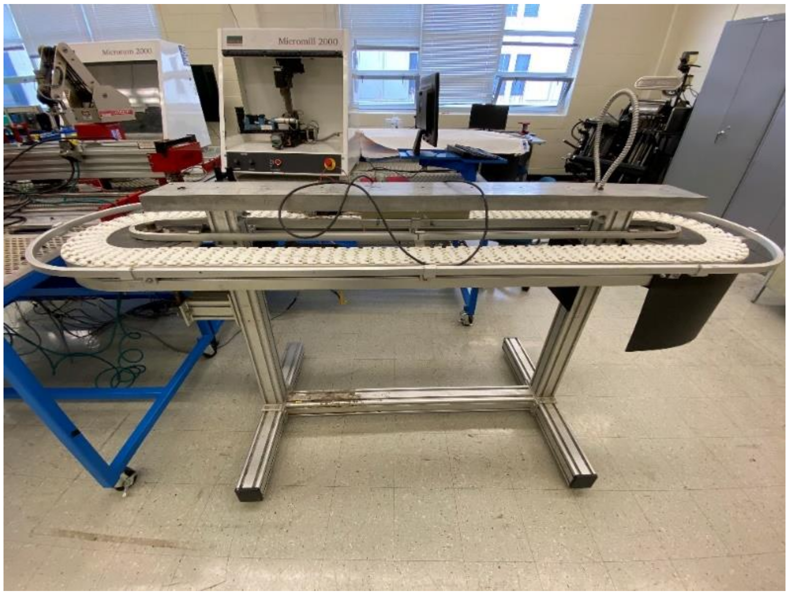


Figure 34. One of the training prototype's conveyors

3.2.3.2 Slider

A slider, as the name implies, is generally a linear motion actuator that is responsible for rapidly moving elements from one point to another through a guide, generally in conveyor belt systems. Most of these actuators are energized from a pneumatic (pressurized air) or mechanical impulse.

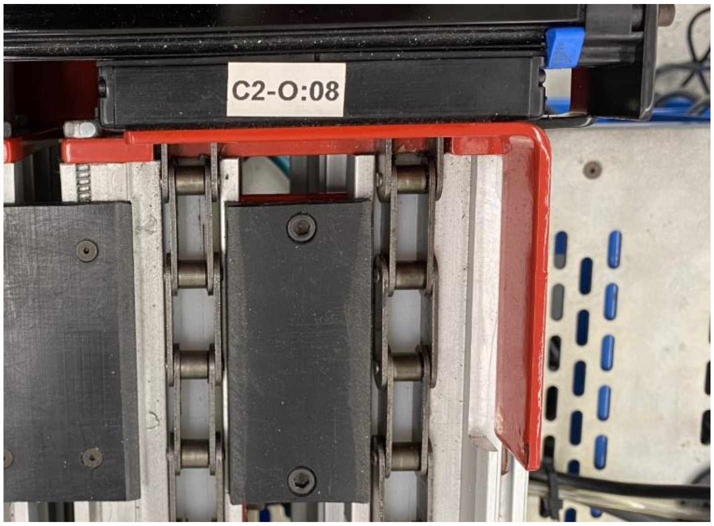


Figure 35. Slider actuator

3.2.3.3 Linear Actuator

A linear actuator is a device that, as mentioned previously, converts a certain type of energy into linear movements. In the particular case of the training prototype, there are two types of pneumatic actuators that perform linear movements on the part supports. One of them moves to raise or lower the support in a fixed position so that the robotic arms can manipulate the parts, and the other actuator performs the function of blocking the advance of the supports along the conveyor belt.

Figure 36. Linear actuator for the main conveyor

3.2.4 Sensors

Sensors, also known as transducers, are one of the fundamental components of modern data acquisition systems. A sensor is a device that detects a change in the environment and responds to an electrical stimulus from a control system. A sensor converts a physical phenomenon into a measurable analog voltage (or sometimes a digital value), which in turn can be sent to a human-readable display or transmitted for further reading or processing.

Depending on the type of sensor, its electrical output can be a voltage, current, resistance, or another electrical attribute that varies over time. Some sensors are available with digital outputs, thus generating a series of scaled or unscaled data bytes.

3.2.4.1 Snap-action Switch

A snap-action switch is a switch that only has a single input and can be connected to and switched between two outputs. This means that it has one input terminal and two output terminals. Snap-action switches can serve a variety of functions in a circuit. It can serve as an on-off switch, depending on how the circuit is connected, or it can be used to connect circuits to any two different paths that a circuit may need to function.

The way of operation of these devices is that, when activated, they can give information to the controller (in this case, the robot controller) to indicate that support is in position to be moved by the sliders.

Figure 37. Snap-action sensor

3.2.5 Machine Vision Inspection System

The industrial machine vision inspection system is one of the technologies that make the difference in certain essential tasks in industrial production. These artificial vision systems applied to robotics facilitate the solution in industrial phases as decisive as quality controls or the detection of defective products. Artificial vision is an industrial technology applicable to different sectors and production phases. It is one of the most effective and innovative automated and intelligent methods for acquiring, processing, and analyzing images in production processes.

3.2.5.1 Camera

The smart camera is one of the most popular technological advances in relation to vision sensors and stand out for their computing power—capable of providing a solution to any need for industrial vision—image resolution and easy installation. This facilitates that their applications are vastly varied, regardless of activity or phase of the production chain. Its most innovative aspect lies in its processing capacity, which provides it with storage and availability to connect with other automated systems, due to the use of input and output mechanisms.

One of the main advantages of the machine vision hardware used in the prototype is that it includes an inspection software that allows statistical analysis of measurements and faults in the patterns of the manufactured parts in such a way that it offers a pass/fail result of the product.

Figure 38. Cognex® machine vision camera sensor

3.2.5.2 Barcode Reader.

Another of the hardware devices incorporated in the vision system is the barcode readers. This electronic device consists of a scanner capable of reading the bar codes by means of a laser to later send the data, through a Wi-Fi antenna or cable, to a terminal or computer.

The prototype has three barcode readers, which access different types of information that are embedded within a barcode located on each of the supports: this, in turn, specifies the type of material to be placed and the shape of the part to be manufactured.

3.2.6 Piece Holder

A piece holder is a common element found in many production lines. This is an object whose only function is to provide support and transportation for either raw materials going into a specific process and for finished products obtained from said process. The manufacturing training prototype includes two types of piece holders: one destined to support blocks going into the mill and the other dedicated to transport the cylindrical pieces towards the lathe.

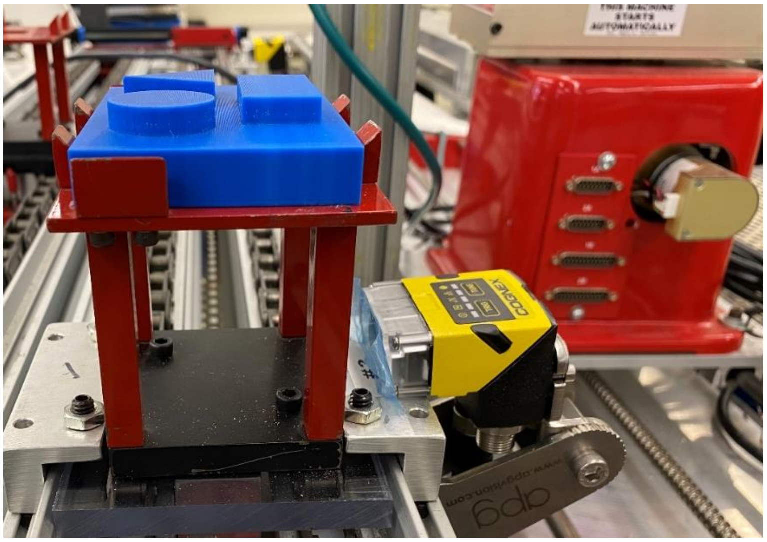


Figure 39. Cognex® machine vision inspection system barcode reader, along with a piece holder

3.3 Methodology

The main function of the manufacturing prototype is as follows: one of the two robotic arms located in the stations grabs different pieces of raw material in the form of acrylic/wooden blocks and aluminum cylinders and moves them into different holders inside of one of the conveyors. These holders are transported to different stops, in which they get a certain barcode scan by sensors located in fixed positions. According to the type of part, the barcode readers send a specific set of data to the PLC, which, in response, tells the robot controller to move and place the piece into the desired CNC machine (the blocks to a mill and the cylinders to a lathe). Once there, and also according to the given barcode, the machines will execute one of two different sets of M codes to generate a specific model with different geometric references.

When finished, the CNC machines will communicate with the PLC and the robot controller to move the other robotic arm towards the machine, pick up the finished part and deposit it back into the holder. Then, the holder will continue its trajectory inside the conveyor until it reaches the vision system. At that point, the camera takes a capture of the finished part and analyzes a certain set of parameters. According to a given set of specifications, the software indicates if the part is either accepted or rejected. This status determines the correct final conveyor in which it is placed.

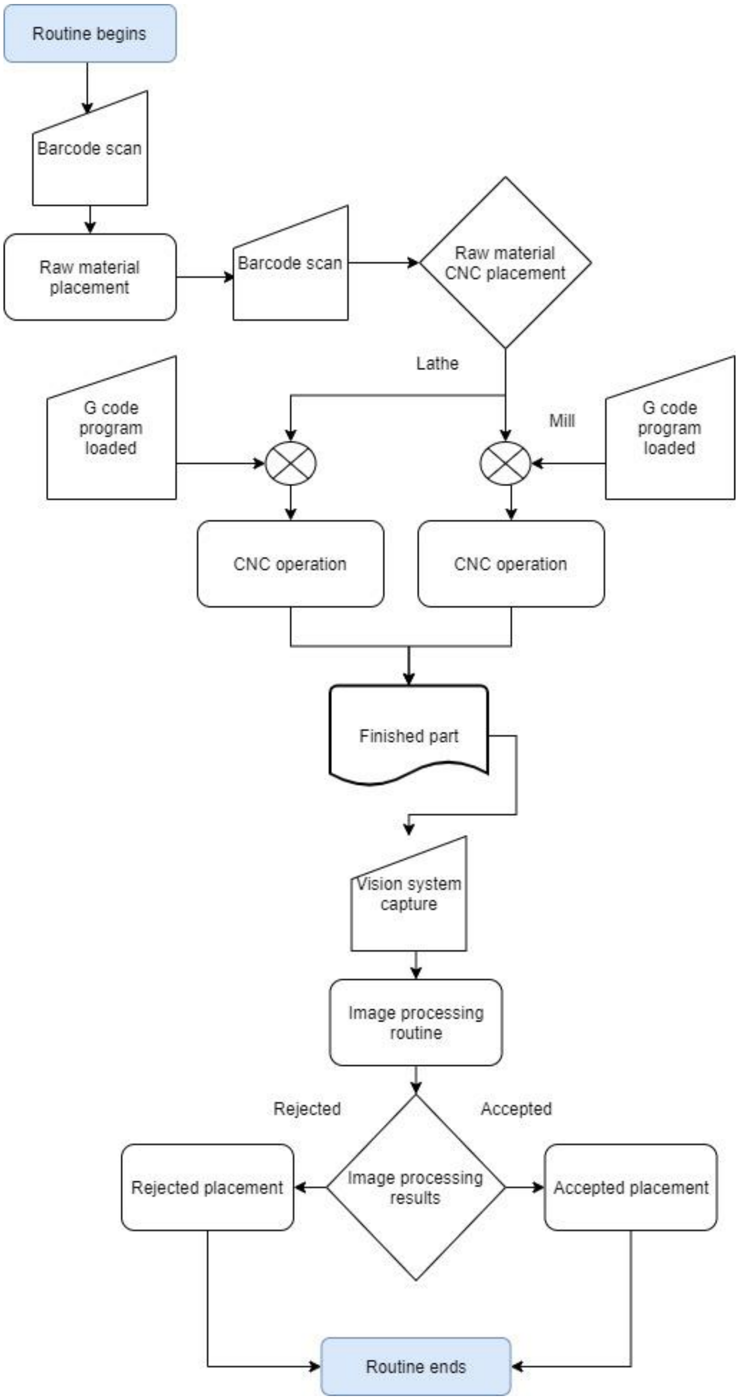


Figure 40. Flowchart representing the manufacturing training prototype's routine

3.3.1 3D Design Stage

The proposed environment is composed of different 3D models created by computer-aided design (CAD) software. The selected models were imported into the Solidworks® CAD software, chosen because of their flexibility when handling different types of 3D model files and the wide range of operations used to achieve the desired shapes. Once the CAD model is finished, shown as an example in Figure 41, it is then exported to the 3DS Max modeling software, represented in Figure 42.

One of the main reasons for this is because the scenario has to adapt to the minimum specifications that the computational platform (in this case, laptops belonging to the testing crew) can afford. Too heavy a render can cause a significant delay in the number of actions that the laptop processor has to run. Basic geometrical shapes are the exceptions since the game engine can create them with a relatively simple mesh, and they are used to represent objects that are either core to the functions of the prototype or part of the interactive elements in it.

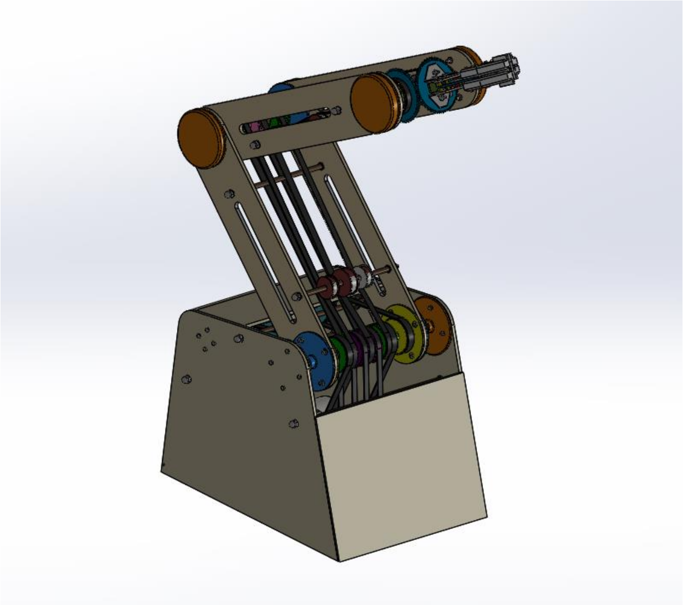


Figure 41. 3D model of the Amatrol robotic arm

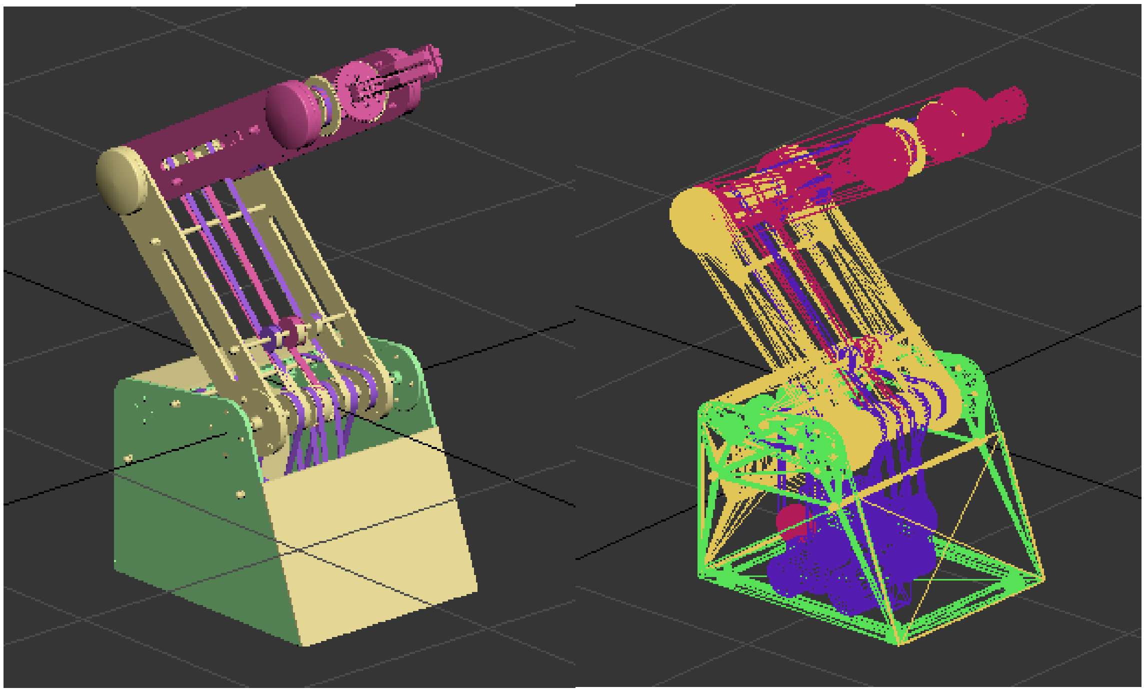


Figure 42. Face and mesh models od the 3D robotic arm

3.3.2 Game Engine Stage

These models are exported into a game engine in which they are arranged according to the needs of the virtual environment. The game engine software allows the environment to simulate real-life conditions within the virtual object's physical properties. Actions like grabbing, releasing, twisting, and pressing must be as realistic as possible for the user. Therefore, the more detailed the actions are programmed, the easier it is for the user to adapt to the real objects when given a chance.

One of the best game engines available for virtual scenario design is Unity 3D®, free software that is also mainly used for game development. Since its creation in 2002, the engine has undergone a series of upgrades that make it possible to work with third-party apps and hardware, including VR and AR devices.

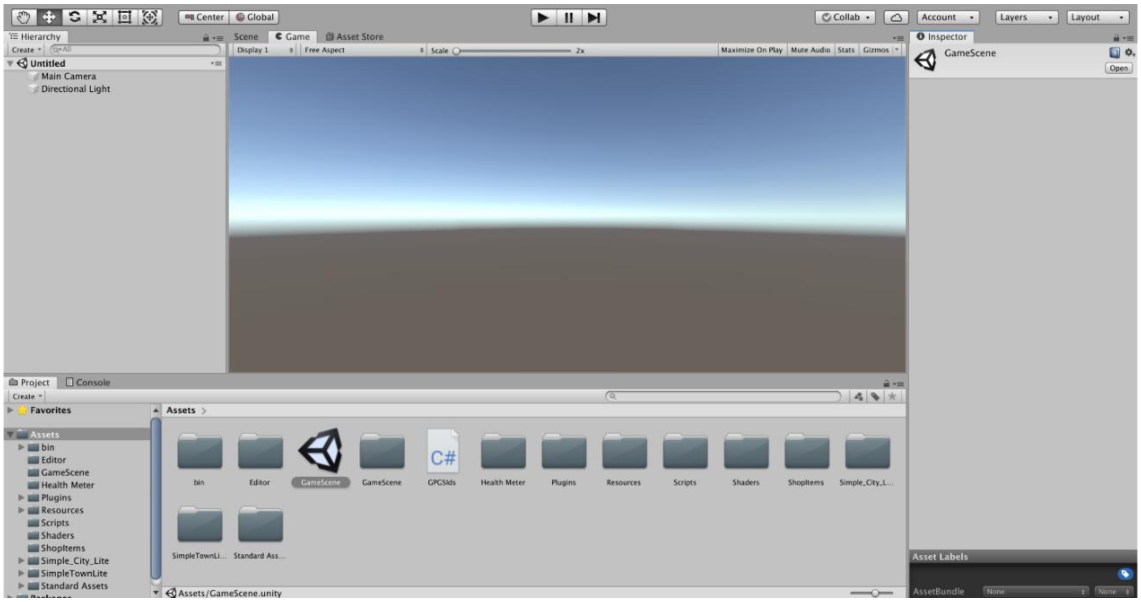
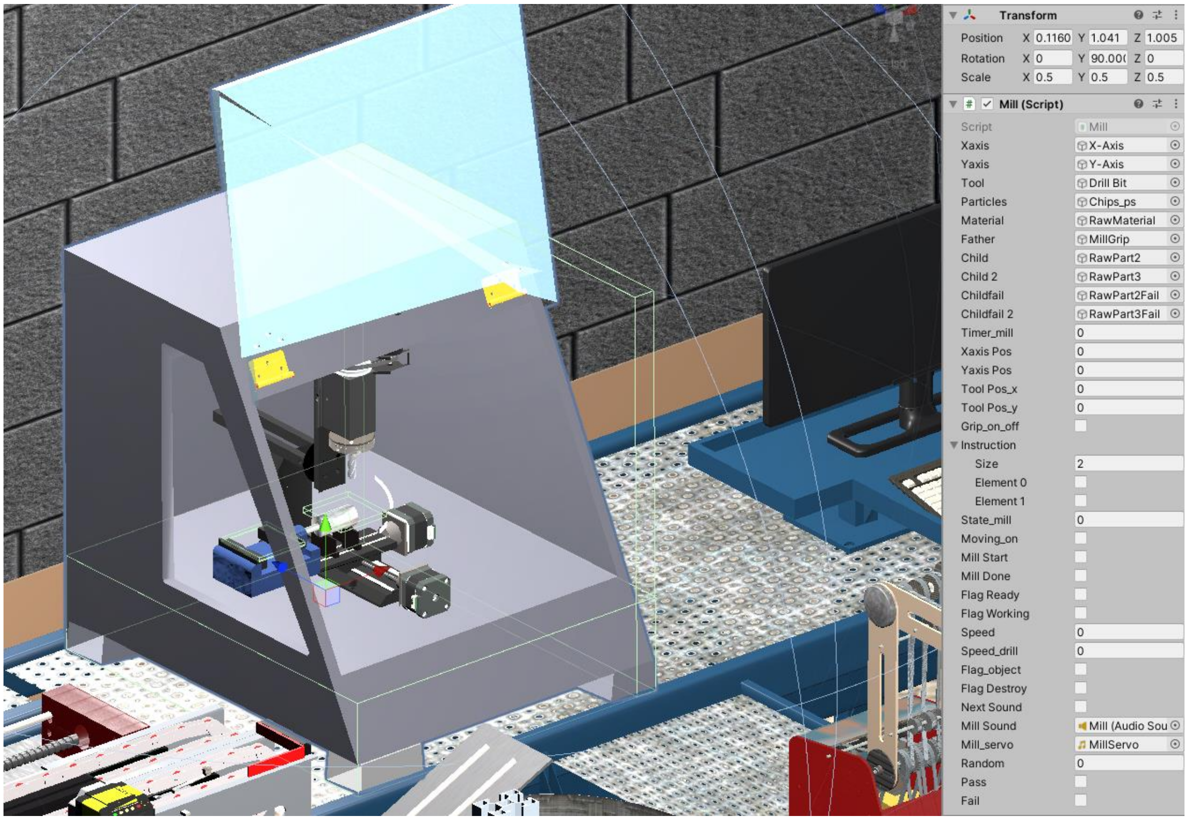


Figure 43. View of a scene in Unity 3D®

The game engine consists of different working environments, called "scenes" (Figure 43), in which the virtual objects get placed and arranged according to the desired specifications of what should be simulated. The placed objects, called "GameObjects," can be modified afterward in a wide number of properties like position, orientation, size, weight, material, and its animations.

The GameObjects are subsequently animated through C# scripting. Since the purpose of the scenario is to emulate the same physical actions of the real-life training prototype, the logic chosen for the scripts is sequential, which matches the same logic implemented in the robotic arms, the CNC machines, and the conveyor movement are coordinated by the PLC. C# offers a clear, powerful, and robust availability to create the necessary codes, but another advantage is that most VR/AR/MR packages that belong to the hardware used can already be found in this programming language.



Transform
Position X 0.1160 Y 1.041 Z 1.005
Rotation X 0 Y 90.00(Z 0
Scale X 0.5 Y 0.5 Z 0.5

Mill (Script)

Script	Mill
Xaxis	X-Axis
Yaxis	Y-Axis
Tool	Drill Bit
Particles	Chips_ps
Material	RawMaterial
Father	MillGrip
Child	RawPart2
Child 2	RawPart3
Childfail	RawPart2Fail
Childfail 2	RawPart3Fail
Timer_mill	0
Xaxis Pos	0
Yaxis Pos	0
Tool Pos_x	0
Tool Pos_y	0
Grip_on_off	☐
Instruction	
Size	2
Element 0	☐
Element 1	☐
State_mill	0
Moving_on	☐
Mill Start	☐
Mill Done	☐
Flag Ready	☐
Flag Working	☐
Speed	0
Speed_drill	0
Flag_object	☐
Flag Destroy	☐
Next Sound	☐
Mill Sound	Mill (Audio Sou
Mill_servo	MillServo
Random	0
Pass	☐
Fail	☐

Figure 44. Properties for the mill GameObject

3.3.2.1 Colliding Properties

One of the most important properties of the virtual objects is the way in which each of them interacts with the rest of the environment and the user itself. For that, a "Rigidbody" property is used, which adds a gravity and mass component to the object so that it can behave like a real object. However, it also enables an element called "Collider," which is an invisible geometrical limit that closely approximates the size of the GameObject. When colliders interact, their surfaces need to simulate the properties of the material they are supposed to represent.

Another useful mode for colliders is their function as triggers. The scripting system can detect when collisions occur and initiate actions using reserved C# functions. However, it can also use the physics engine simply to detect when one collider enters the space of another without creating a collision. A collider configured as a "Trigger" does not behave like a solid object and will simply allow other colliders to pass through.

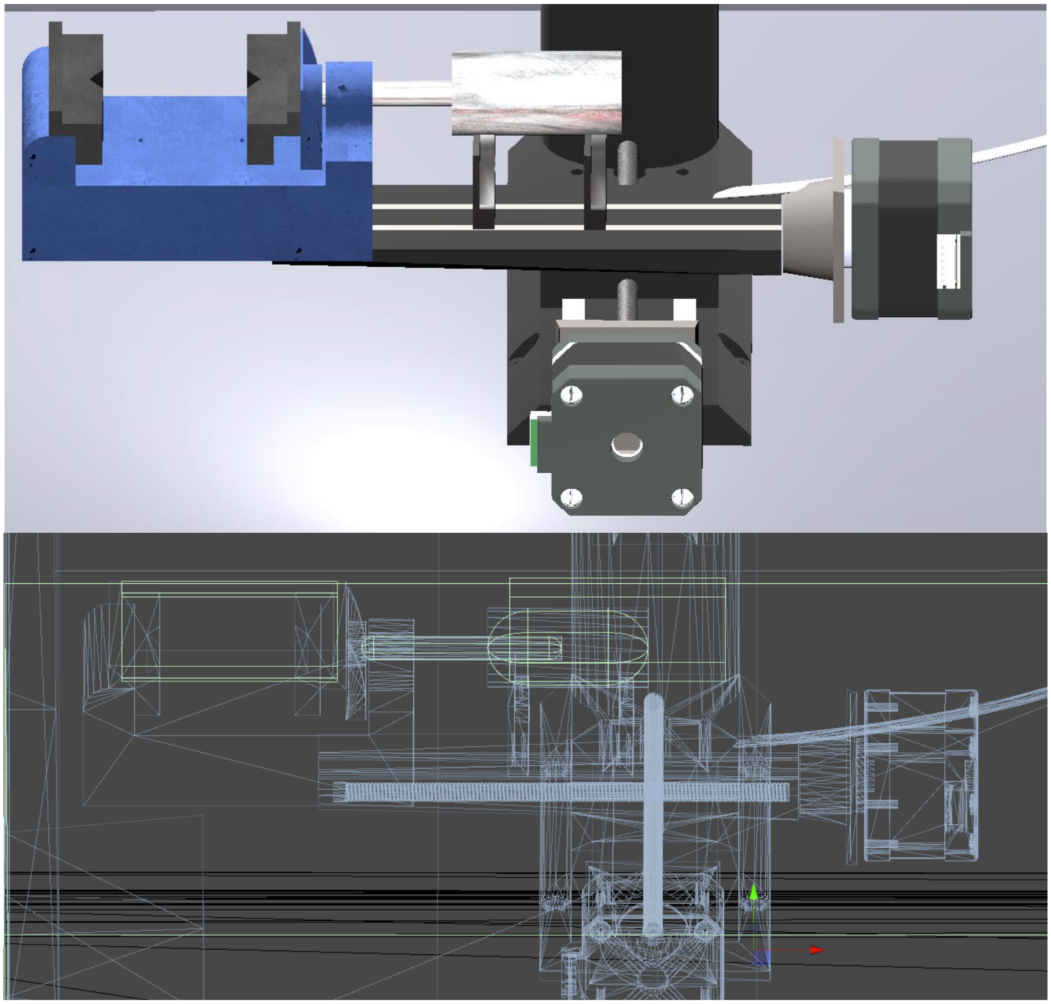


Figure 45. Corresponding collider elements of the mill tool

3.3.3 Virtual User Integration

Once the main functions for each virtual object were programmed and tested, the next step was to work with the placement of the virtual user, commonly referred to as an "avatar," and its input control commands. The way in which this can be achieved is by means of plug-ins or added extra configurations that enable Unity 3D® to recognize the hardware inputs coming from the VR setup.

The chosen HMDs for the project, both the Oculus Rift® and the VIVE Pro®, have official packages that include virtual objects, scripts, materials, plug-ins, etc., that can be easily configured inside of the Unity scenes. One of the main elements is the "PlayerController" object (Figure 46), which includes the reference for both the placement of the HMD stream and the user's input controllers capsuled into one single object.

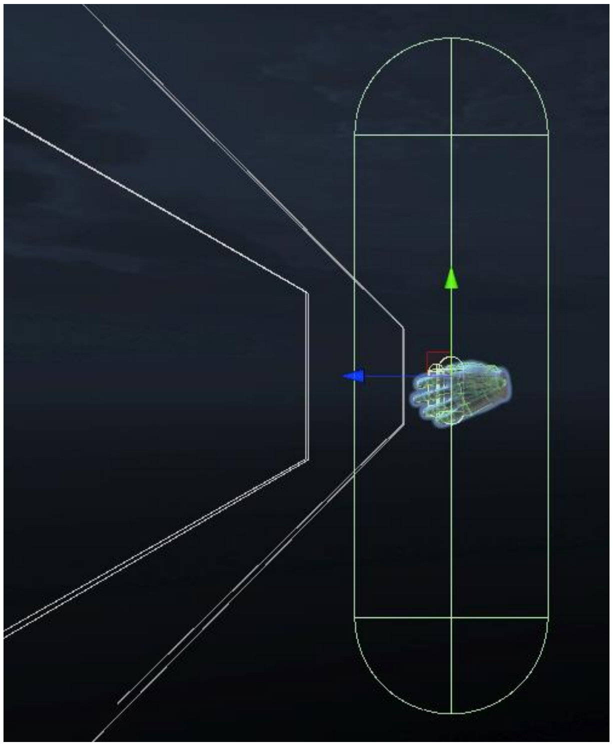


Figure 46. Detailed view of the PlayerController object

3.3.4 Calibration Stage

For the MR layout, the HMD is used alongside motion tracking devices and a 3D stereo camera. All of these devices' software development kits (SDK) were imported into the game engine, which allows the use of their resources to be mixed with the rest of the scene. The green-screen background, which is used to incorporate the area that projects the stream of the virtual environment window, must be set up in an optimal area that can be used to set up the screen, like the corner of a room, which provides accurate depth and dimensional references to the tracking devices.

The green screen setup consisted on painting a limited area within the VR Laboratory of said color, and placing green-colored mats on the floor. This way, the projection of the virtual environment could match the physical limits of the testing area.

Figure 47. Green screen background used for the research

Once all elements were placed, the last step consisted of calibrating the depth camera so that its stream could match a virtual camera placed on the virtual scenario, which in turn could show different perspectives of the user within the composed image. This way, if the user switched positions by walking behind an object in the VR environment, it could be shown the same way on the depth camera stream inside the Unity scene. To do so, the depth camera SDK included a scene in which different perspective points needed to be placed using the controllers, aligning the depth map with the avatar's position.

Finally, the last calibration that needed to be done corresponded to the virtual gloves placed on each of the user's hands. Each glove included a tracker that sent its signal to the hosting PC, and within the SDK, the actions of each hand gesture (grabbing, pointing, etc.) were set up.

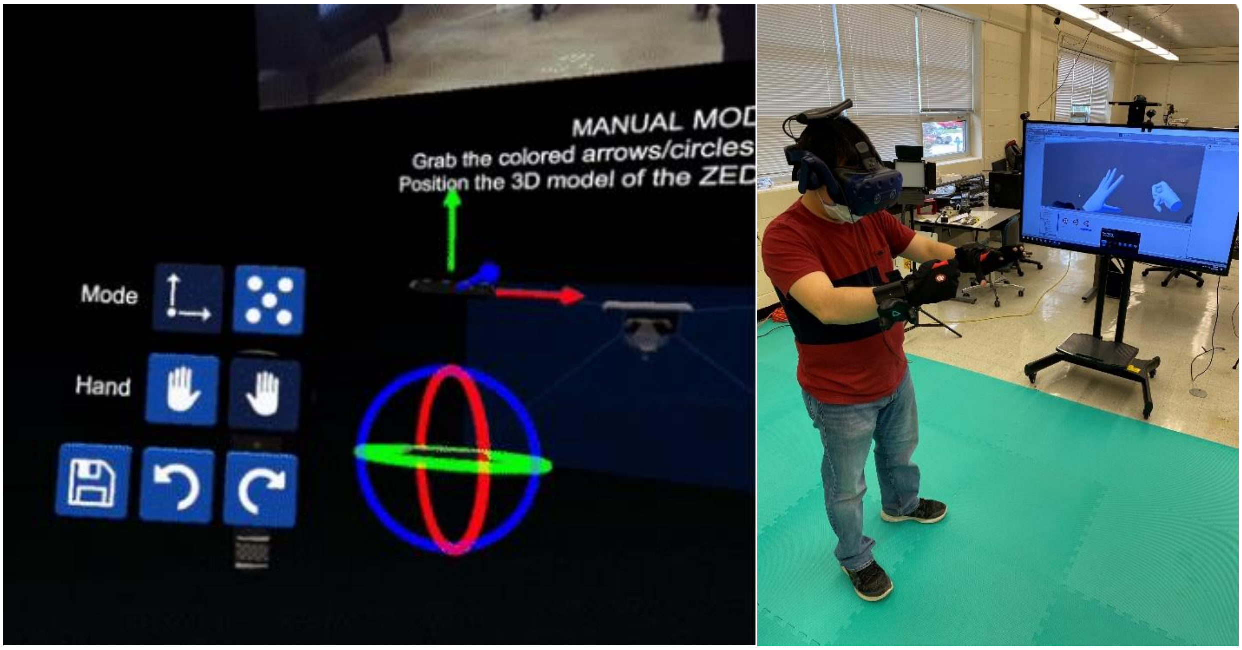


Figure 48. The calibration process for both the depth camera and the virtual gloves

3.3.5 PLC Integration

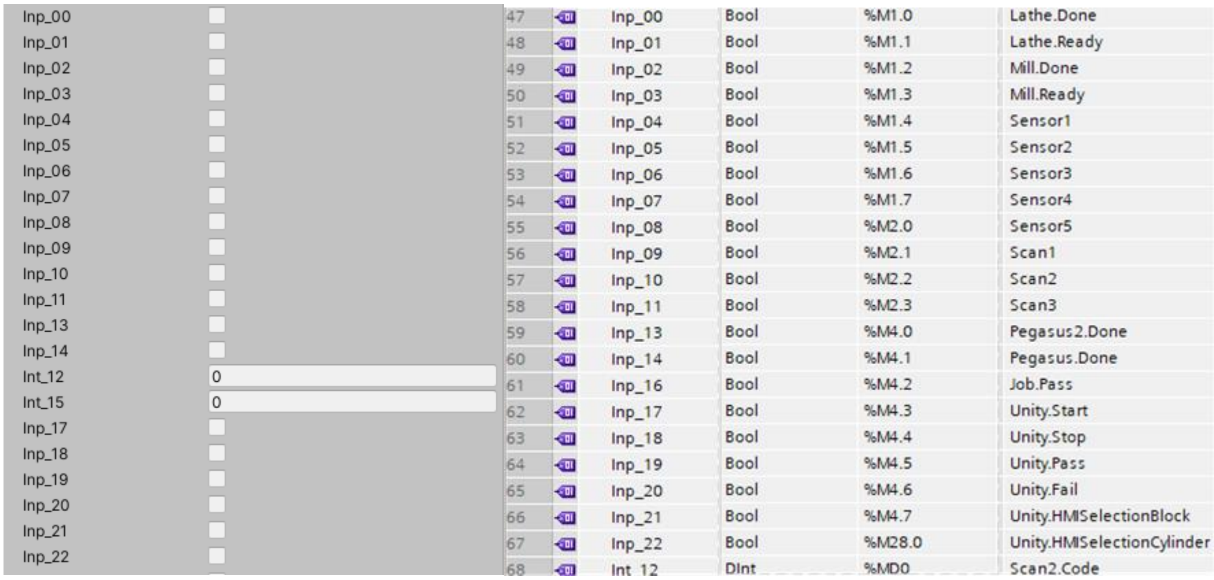
Up to this point of development, the virtual scenario functions as an isolated system. By understanding the real-life training prototype, the user is able to manipulate the internal objects to run a given sequence of instructions in order to see a finished product, which are the manufactured pieces. However, it still remains a system that runs similar to a black box, that is, the true sequence remains something that only the VR designer knows how to operate, and thus, the automation process merely becomes an animation for the user.

To fully make the environment achieve its purpose, it requires communication with the external world. The actions that take place within the virtual scenario would also have to serve a mirrored behavior in the physical prototype. The proposed method in which this was accomplished made use of the OPC industrial communications protocol, which allowed the inputs and outputs of the system to match the inputs and outputs coming from an external PLC module that included the actual prototype code.

3.3.5.1 Signal Identification

The first step of this integration consisted on identifying the I/O signals that came from both the virtual scenario and the external PLC. For the scenario, a GameObject that encompassed all actuators and sensors, as well as the I/O signals that are used on the robotic arms, the CNC machines, and the machine vision system, was created. The script for this object was then linked to another code instance in which a serialization method was used to format the incoming and upcoming signals. This was done using the JavaScript Object Notation (JSON) serialization method, which is a format that encodes the data into a string instruction that gets sent to a pre-established HTTP website that, in turn, deserializes the message and sends the data to user-defined tags.

Conversely, the PLC has to be programmed to run the automation sequence of the virtual environment. The chosen method was to implement a ladder diagram to define a state machine that runs the specific sequences for the CNC final product models, the movement of the robotic arms and their defined positions, and the acceptance/rejection status achieved by the machine vision analysis. The inputs and outputs generated within the PLC code were named in a similar structure as the ones found inside the VR scenario, in an effort to ease the linking process (Figure 49).



Inp_00	☐
Inp_01	☐
Inp_02	☐
Inp_03	☐
Inp_04	☐
Inp_05	☐
Inp_06	☐
Inp_07	☐
Inp_08	☐
Inp_09	☐
Inp_10	☐
Inp_11	☐
Inp_13	☐
Inp_14	☐
Int_12	0
Int_15	0
Inp_17	☐
Inp_18	☐
Inp_19	☐
Inp_20	☐
Inp_21	☐
Inp_22	☐

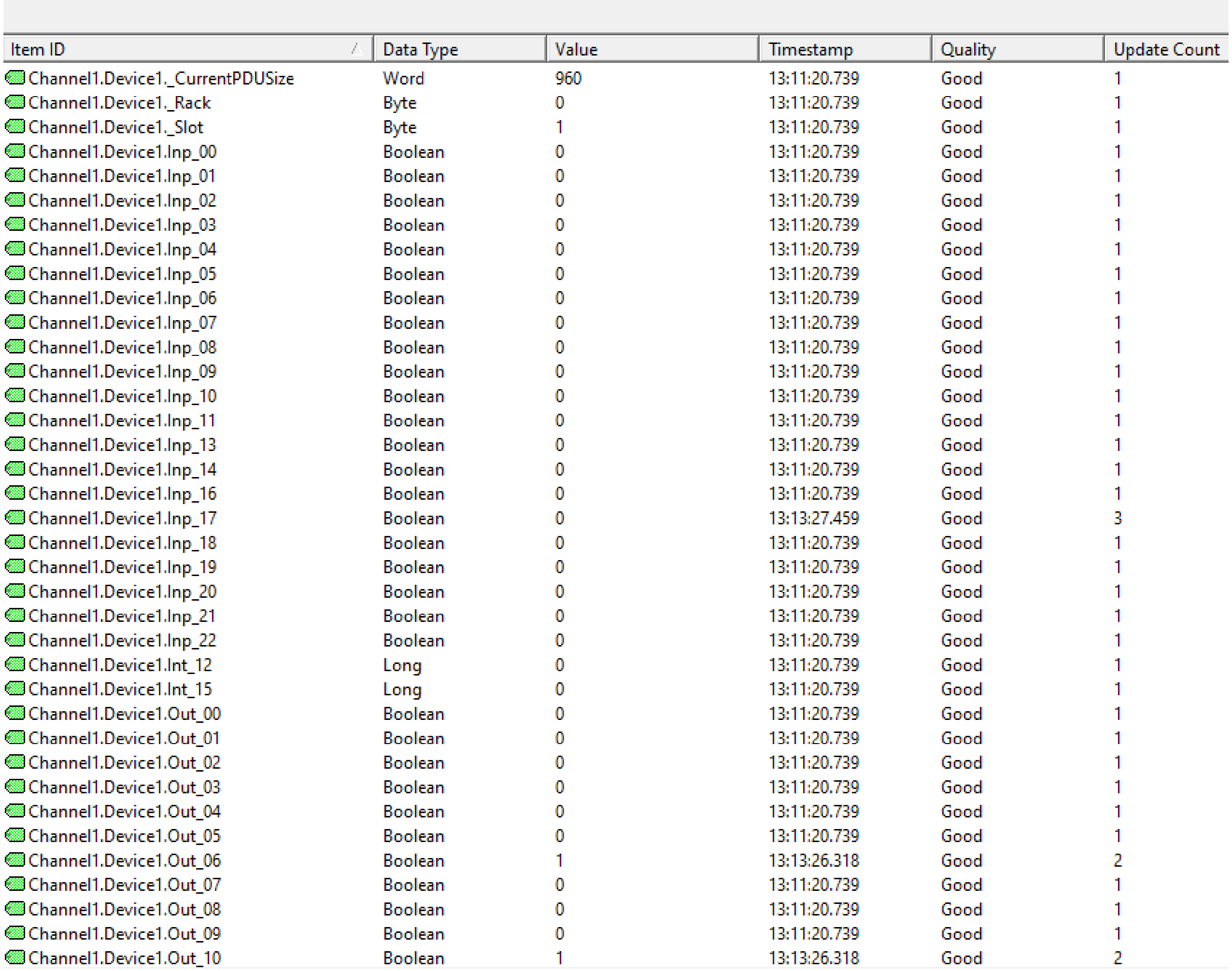
47	Inp_00	Bool	%M1.0	Lathe.Done
48	Inp_01	Bool	%M1.1	Lathe.Ready
49	Inp_02	Bool	%M1.2	Mill.Done
50	Inp_03	Bool	%M1.3	Mill.Ready
51	Inp_04	Bool	%M1.4	Sensor1
52	Inp_05	Bool	%M1.5	Sensor2
53	Inp_06	Bool	%M1.6	Sensor3
54	Inp_07	Bool	%M1.7	Sensor4
55	Inp_08	Bool	%M2.0	Sensor5
56	Inp_09	Bool	%M2.1	Scan1
57	Inp_10	Bool	%M2.2	Scan2
58	Inp_11	Bool	%M2.3	Scan3
59	Inp_13	Bool	%M4.0	Pegasus2.Done
60	Inp_14	Bool	%M4.1	Pegasus.Done
61	Inp_16	Bool	%M4.2	Job.Pass
62	Inp_17	Bool	%M4.3	Unity.Start
63	Inp_18	Bool	%M4.4	Unity.Stop
64	Inp_19	Bool	%M4.5	Unity.Pass
65	Inp_20	Bool	%M4.6	Unity.Fail
66	Inp_21	Bool	%M4.7	Unity.HMISelectionBlock
67	Inp_22	Bool	%M28.0	Unity.HMISelectionCylinder
68	Int_12	DInt	%MD0	Scan2.Code

Figure 49. Detail of inputs used for both the PLC and the virtual environment

3.3.5.2 Communication Using OPC Protocol

Once both ends of the I/O signals were defined, the final step consisted in finding a platform in which the serialization process could occur, considering that scripting it entirely within the Unity 3D® editor could have proven to be a problematic task. Fortunately, thorough research resulted in the findings of open-source software that could support a wide arrange of industrial protocols with the ability to interact within themselves according to OPC interoperability standards. The chosen software was the Kepware® KEPServerEX platform (Figure 50).

One crucial tool found inside the software was the Internet-of-Things (IoT) Gateway plug-in, which allows the use of web servers that integrate real-time industrial data streams into device clouds. This way, the transition from the VR environment and the PLC automation process could occur seamlessly, mostly due to the high-end capabilities of both the clients and server's hardware, as well as the VR Laboratory local network. The IoT plug-in matches the incoming data from the server to the tags created using the PLC's TCP/IP Ethernet driver (which consists of a channel and device with a local IP as an ID to identify and differentiate it from the VR scenario IP). It then links to the local IP of the physical PLC, which in turn sends its output signals in the inverted process back to the VR environment.



Item ID	Data Type	Value	Timestamp	Quality	Update Count
Channel1.Device1._CurrentPDUSize	Word	960	13:11:20.739	Good	1
Channel1.Device1._Rack	Byte	0	13:11:20.739	Good	1
Channel1.Device1._Slot	Byte	1	13:11:20.739	Good	1
Channel1.Device1.Inp_00	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_01	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_02	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_03	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_04	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_05	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_06	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_07	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_08	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_09	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_10	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_11	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_13	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_14	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_16	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_17	Boolean	0	13:13:27.459	Good	3
Channel1.Device1.Inp_18	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_19	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_20	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_21	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Inp_22	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Int_12	Long	0	13:11:20.739	Good	1
Channel1.Device1.Int_15	Long	0	13:11:20.739	Good	1
Channel1.Device1.Out_00	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_01	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_02	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_03	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_04	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_05	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_06	Boolean	1	13:13:26.318	Good	2
Channel1.Device1.Out_07	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_08	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_09	Boolean	0	13:11:20.739	Good	1
Channel1.Device1.Out_10	Boolean	1	13:13:26.318	Good	2

Figure 50. Client window showing the PLC routine tags

4. CHAPTER FOUR: FINDINGS

4.1 Performance

When conducting the first tests within the virtual stage, several positive and negative details were found. The response time of the simulation was quite fast and without long latency times, mostly due to the fact that the hardware utilized allowed the support of the video transmission to the HMD. However, one of the devices affected was the depth camera because the support of its SDKs compatible with the Unity software did not yet have the latest updates.

One of the negative points of the current virtual environment configuration is that the HMD works best only when it remains directly connected to the PC. This is because tests were carried out with a wireless adapter for the HMD, allowing the user to move freely without having to worry about tripping. However, the connection of the wireless adapter with the HMD was very sensitive, and therefore the researcher decided to physically connect the device instead.

The manipulation tests of the virtual objects within the scenario were very favorable since the user was able to execute the command, control, and rotation of objects without any kind of difficulty (Figure 51). This point made it possible to reinforce the objective that each person capable of manipulating virtual objects with ease would be able to make better use of reason and intuition when working with the physical training prototype.

When reviewing the use of the green screen projection, a great advantage noted was that the final integration resulted in the creation of a high-detailed MR environment suited for automation training and teaching. Real-life tracking resulted convincing enough for the users/observers, given the fact that they could see both actions being played out in the environment without the user being distracted while operating the virtual prototype.

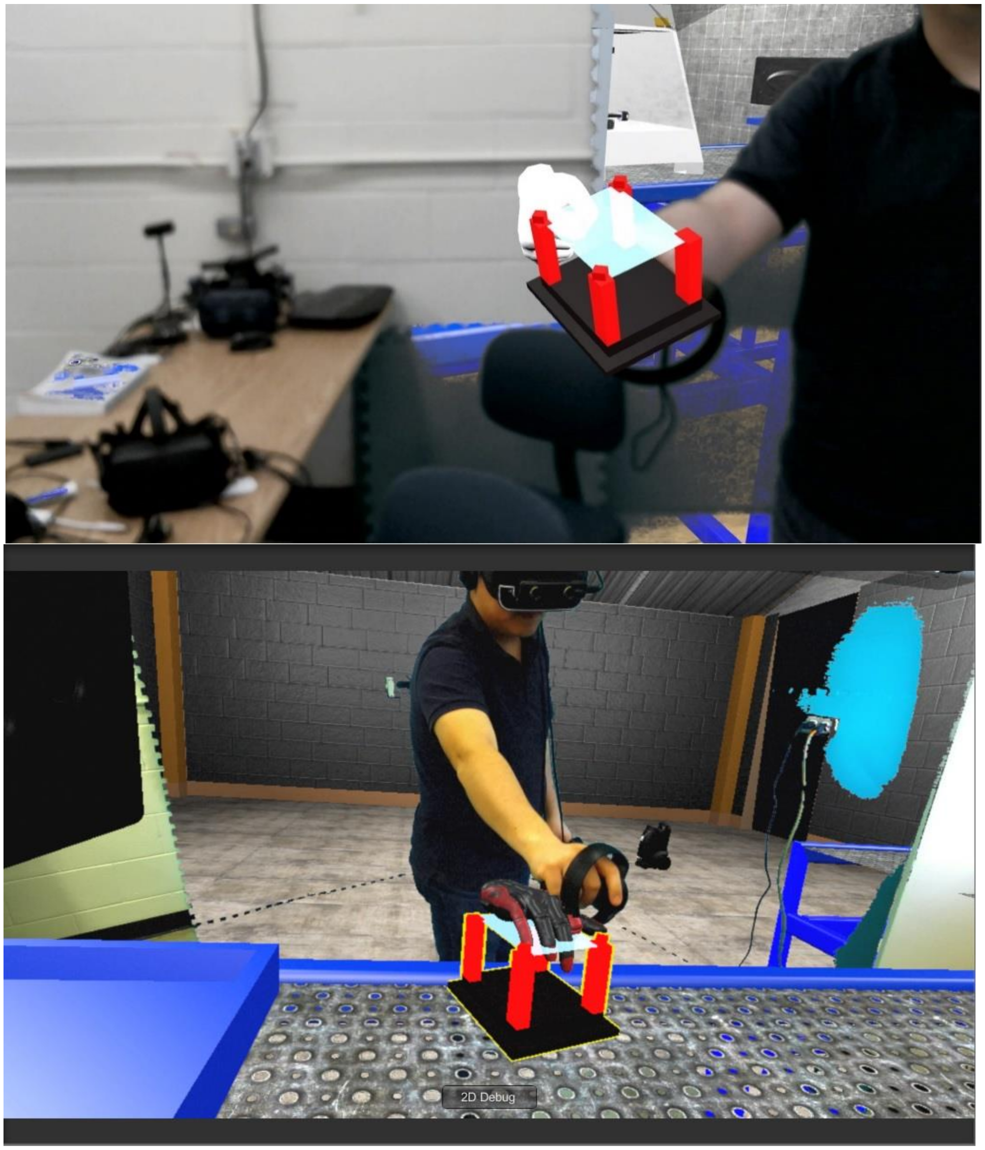


Figure 51. Test subject manipulating a virtual object using the mixed reality configuration

4.2 Machine Vision Inspection Analysis

One of the most important elements that needed to be tested within the VR scenario was the machine vision inspection system. While the real-life training prototype included this system by default and could send the camera captures to the inspection software, it was also intended to test the possibility of capturing screenshots generated within the scenario and run the same analysis on it so that a true virtual experience could be obtained by the user.

A screenshot capture routine was coded using the camera GameObjects placed on the Unity environment. Once the finished objects were manufactured and moved over to the inspection area, the camera GameObject placed on the 3D model of the Cognex® camera captured a still frame of its point of view and stored it on an in-scene folder. This, in turn, was placed as the default folder that the machine vision software uses to load the images needed for its analysis.

The images were run through a pass/fail routine, searching for a specific feature (distance, geometric shape, edge detection, etc.) in each of the captures. If it is found, then it counts as a pass, and the produced virtual object is considered accepted. If not, the piece is considered rejected. Results were positive, and the generated patterns were able to match the features of the manufactured virtual products.

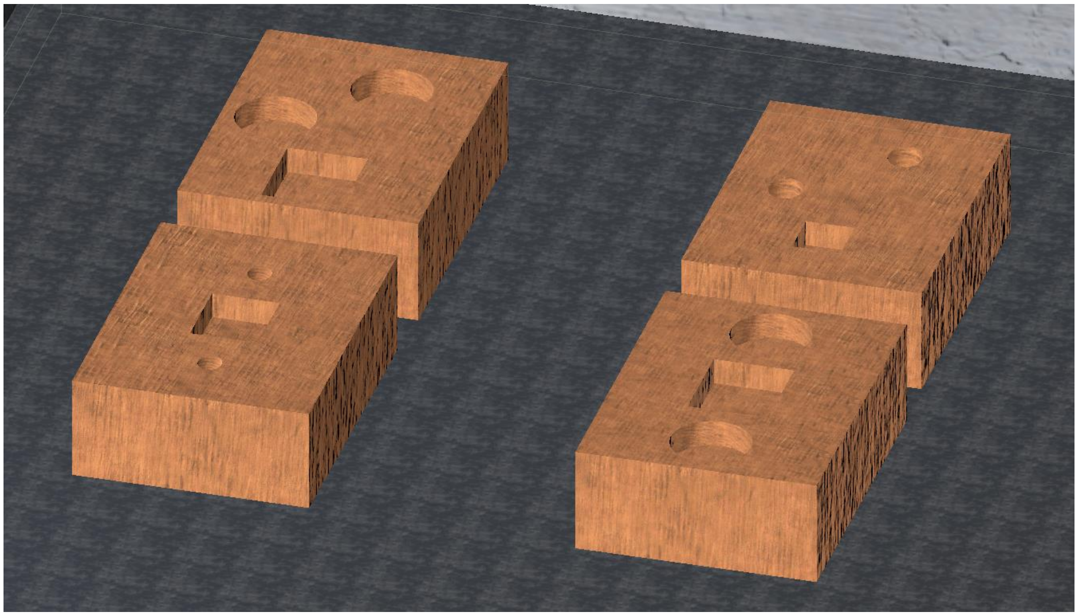


Figure 52. Results for both accepted and rejected mill products. Objects with wider geometrical features are considered to be accepted

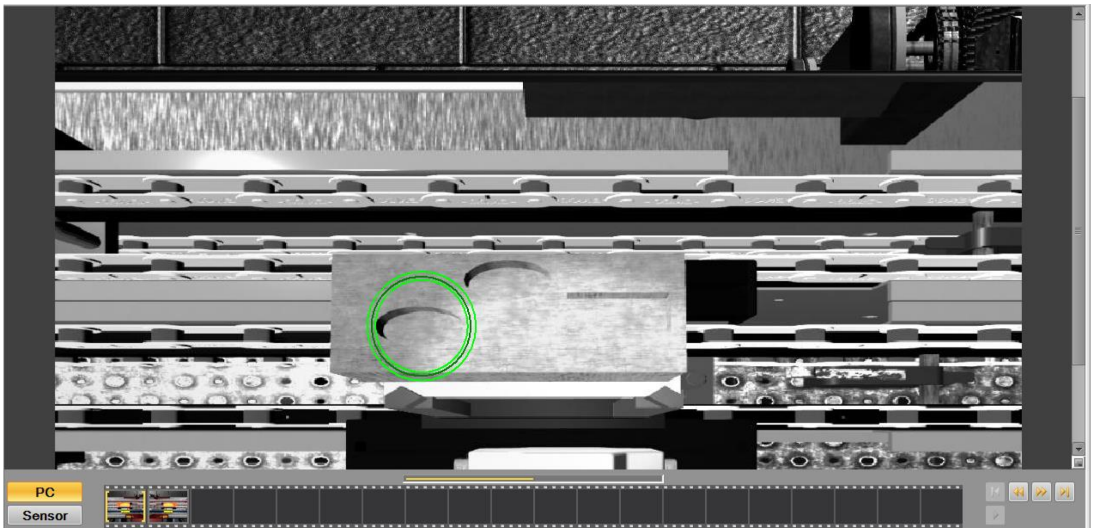


Figure 53. Geometry detection routine executed into the machine vision software

4.3 Graphic User Interface Integration

When performing the different tests of the virtual scenario, it was observed that the learning obtained within the process could be used within an academic testing environment in order to obtain a more generalized feedback about the pros and cons of the current performance of the project. For this, it was important that users (Morehead State University students who were enrolled in subjects related to PLC programming) could have access to a much friendlier interface than the one developed in the laboratory test environment.

The task of the research then shifted to the design of a graphic user interface (GUI) within the Unity 3D® scene that would allow the user not only to visualize the I/O signals of the virtual scene but also to give them the opportunity to manually manipulate each one to visualize their behavior. For this, manual connection and disconnection buttons for the OPC protocol platform were implemented, as well as buttons to switch the virtual visualization through the HMD to a simpler navigation mode using the keyboard and mouse connected to the PC.

The reception of students was generally positive, arguing that movement within the virtual environment proved to be simple enough for them to intuitively understand the controls of the scenario, as well as which signals were needed to manipulate the different virtual objects.

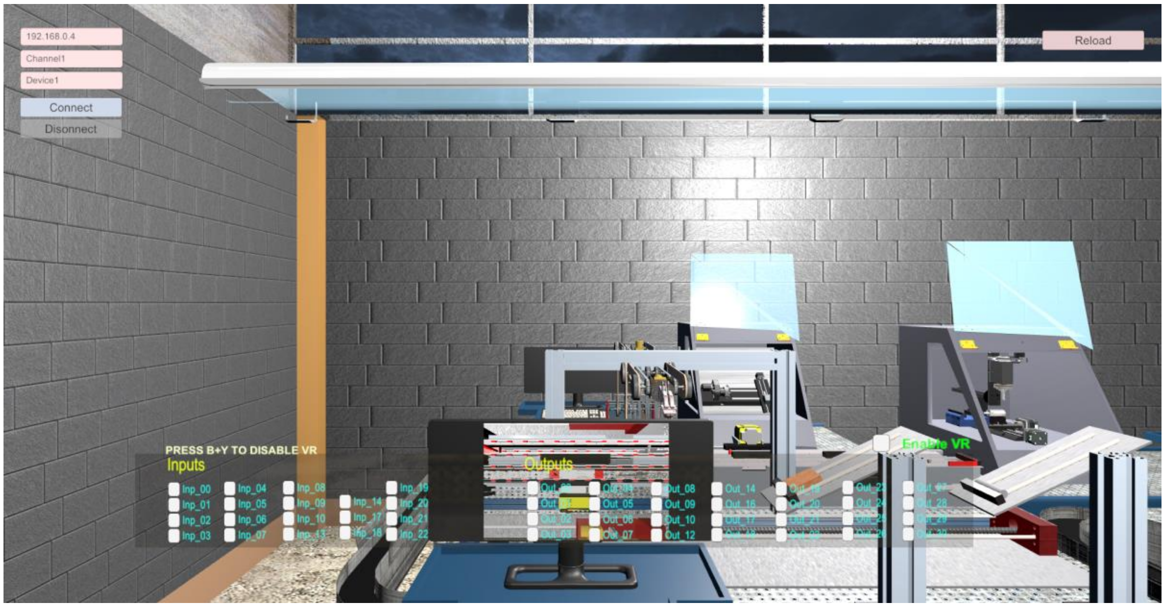


Figure 54. Final design for the First-Person View GUI

5. CHAPTER FIVE: CONCLUSIONS

The purpose of this research was focused on developing a novel way of approaching users to a realistic experience that could offer them the same knowledge in manipulating a training manufacturing prototype as they would get in real life, but as the project kept on adding new features, it allowed the fact that, by enhancing the user's experience of its surroundings on both VR and MR environments, it could increase its intuitiveness and break the hardware unfamiliarity gap that many people have when attempting to use of this type of equipment.

It is worth noting that the MR training experience is not yet focused on people with physical disabilities involved in sight and touch. While it was not the intention of this research to include this population, further research can be done to figure out a new design that could make use of other technology that could bridge this gap. Also, some other technical problems were noted: software development kits' (SDK) parameters needed to be adjusted to fit the specifications of the VR hardware, and the right calibration and patch versions needed to match in order for the environment to run. Otherwise, frame drops could be experienced because of its current graphics processing unit (GPU) hardware.

If needed, the modification of the scenario is possible, but new processes could take time to be modeled/scripted, especially when they need to be built from the very early stages. Since most of the work was mostly being done by one person, it can only be concluded that adding a larger group of developers could greatly decrease the design stage time and focus on more frequent tests.

Finally, it is noted that a great enhancement for this research would be the technological migration of this environment into an AR environment that could benefit from the real-time data obtained by running the physical training prototype alongside its virtual counterpart. By doing this, future users can compare performance and data values between the two systems in order to bring a more synergetic environment, in which multiple operators and observers can take actions to stabilize and keep both systems operational, and find new ways in which these platforms can be used more frequently by both the manufacturing industry and academic programs.

6. REFERENCES

Alinier, G., Young, H. M., Farra, S. L., & Kardong-Edgren, S. S. (2019, January 6). A Call to Unify Definitions of Virtual Reality. *Clinical Simulation in Nursing, 31*, pp. 28-34. Retrieved from https://doi.org/10.1016/j.ecns.2019.02.006

Bamodu, O., & Ye, X. (2013). Virtual Reality and Virtual Reality System Components. *2nd International Conference On Systems Engineering and Modeling. 35*, pp. 765-767. Beijing: Atlantis Press. doi:https://doi.org/10.2991/icsem.2013.192

Belai, I., & Drahoš, P. (2009, 01 01). THE INDUSTRIAL COMMUNICATION SYSTEMS PROFIBUS AND PROFInet. *Applied Natural Sciences, 1*, pp. 329-336. Retrieved from Business Industrial Network: https://www.researchgate.net/publication/229018407_THE_INDUSTRIAL_COMMUNICATION_SYSTEMS_PROFIBUS_AND_PROFInet

Burdea, G. C., & Coiffet, P. (2003). *Virtual Reality Technology* (Vol. 2). Paris, France: Wiley-IEEE Press.

Chotrov, D., Uzunova, Z., Yordanov, Y., & Maleshkov, S. (2018). Mixed-Reality Spatial Configuration with a ZED Mini Stereoscopic Camera. *Technologies and Education for a Smart World. 2*, pp. 21-24. Sofia: FDIBA. Retrieved from http://conference.fdiba.tu-sofia.bg/uploads/2018/Proceedings/INF5.pdf

Defense Modeling and Simulation Enterprise. (2020, September 15). *M&S Glossary - Terms & Definitions*. Retrieved from M&S Glossary: https://www.msco.mil/MSReferences/Glossary/MSGlossary.aspx

Denford. (2003). *CNC Lathe User's Manual* (5th ed.). Medina, Ohio, United States of America: Denford.

Denford. (2003). *CNC Mill User's Manual* (5th ed.). Medina, Ohio, United States of America: Denford.

ECMA International. (2017, December). *ECMA-404 The JSON data interchange syntax*, 2nd. Retrieved from ECMA International: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/

Facebook. (2020). *Glossary*. Retrieved from Facebook 360 Video: https://facebook360.fb.com/glossary/

Farrukh, M., Halepoto, I., Chowdhry, B., Kazi, H., & Lal, B. (2017, February 1). Design and Implementation of PLC based Automatic Liquid Distillation System. *Indian Journal of Science and Technology*, *10*(29), pp. 1-6. doi:10.17485/ijst/2017/v10i29/117368

Flavián, C., Ibáñez-Sánchez, S., & Orús, C. (2018, November 2). The impact of virtual, augmented and mixed reality technologies on the customer experience. (Elsevier, Ed.) *Journal of Business Research, 100*, pp. 547-560. doi:https://doi.org/10.1016/j.jbusres.2018.10.050.

FOLDOC . (1998, December 17). *Object Linking and Embedding*. Retrieved from Free Online Dictionary of Computing: http://foldoc.org/Object+Linking+and+Embedding

Gutiérrez-Guerrero, J. M., & Holgado-Terriza, J. (2019, May 29). Automatic Configuration of OPC UA for Industrial Internet of Things Environments. *Electronics, 8*(6), pp. 1-17. doi:https://doi.org/10.3390/electronics8060600

InduSoft. (2017, February 2). How HMIs can solve the industrial Ethernet communication problem. (L. Langnau , Ed.) Austin, Texas, United States of America. Retrieved from https://www.designworldonline.com/hmis-can-solve-industrial-ethernet-communication-problem/

Intel. (2021, Mach 1). *What Is a GPU?* Retrieved from Intel: https://www.intel.com/content/www/us/en/products/docs/processors/what-is-a-gpu.html

Khoshelham, K., Tran, H., & Acharya, D. (2019, April 6). INDOOR MAPPING EYEWEAR: GEOMETRIC EVALUATION OF SPATIAL MAPPING CAPABILITY OF HOLOLENS. In I. A. Sci. (Ed.), *ISPRS Geospatial week 2019. XLII-2*, pp. pp. 805-810. Enschede: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. doi:10.5194/isprs-archives-XLII-2-W13-805-2019

Lu, Y., Xu, X., & Wang, L. (2020, September 6). Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios. *Journal of Manufacturing Systems*, *56*, pp. 312-325. doi: https://doi.org/10.1016/j.jmsy.2020.06.010

Mahmoud, M. S., Sabih, M., & Elshafei, M. (2015, February). Using OPC technology to support the study of advanced process control. (Elsevier, Ed.) *ISA Transactions*, *55*, pp. 155-167. doi:https://doi.org/10.1016/j.isatra.2014.07.013.

McCarthy, S. (2020). *Terminology and Definitions*. Retrieved from The VR Glossary: http://www.vrglossary.org/

Microsoft. (2020, August 26). *What is Mixed Reality?* Retrieved from Mixed Reality: https://docs.microsoft.com/en-us/windows/mixed-reality/discover/mixed-reality

National Electrical Manufacturers Association. (2005, February 2). *Programmable Controllers (PLC), Part 2: Equipment Requirements and Test*, 3.0. (National Electrical Manufacturers Association) Retrieved from Standards: https://www.nema.org/docs/default-source/standards-document-library/ia2-2.pdf?sfvrsn=b6292e24_2

Nincarean, D., Bilal Ali, M. B., & Abd halim, N. D. (2013, May 1). Mobile Augmented Reality: The Potential for Education. *Procedia - Social and Behavioral Sciences, 103*, pp. 657–664. doi:10.1016/j.sbspro.2013.10.385

Novak-Marcincin, J., Barna, J., Janak, M., & Novakova-Marcincinova, L. (2013, November 26). Augmented Reality Aided Manufacturing. (Elsevier, Ed.) *Procedia Computer Science, 25*, pp. 23-31. doi:https://doi.org/10.1016/j.procs.2013.11.004.

Oliver, K. G. (1991). *Basic Industrial Electricity: A Training and Maintenance Manual* (Vol. 91). New York, New York, United States of America: Industrial Press Inc.

OPC Training Institute. (2021). *OPC Dictionary*. Retrieved from OPC Training Institute: https://www.opcti.com/Dictionary.aspx?type=1&AspxAutoDetectCookieSupport=1

Pimentel, J., & Schneider, G. A. (2013). *Network Topology Protocol Change from ControlNet to Ethernet/IP for a Master Control Station in a Subsea Production System.* Universidade Tecnológica Federal do Paraná, Department of Electronics. Curitiba: Universidade Tecnológica Federal do Paraná. Retrieved from http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/2410/1/CT_CEAUT_IV_2014_13.pdf

Raffaele, R. C. (2017). *Virtual Reality Immersive user interface for first person view games.* University of Beira Interior. Covilhã: University of Beira Interior. Retrieved from https://www.researchgate.net/publication/318572588_Virtual_Reality_Immersive_user_interface_for_first_person_view_games

Rosato, D., & Rosato, D. (2003). Computer-aided Design. In D. Rosato, & D. Rosato, *Plastics Engineered Product Design* (Vol. 1, pp. pp. 344-380). United States of America: Elsevier.

Schneider Electric. (2019). *Glossary*. Retrieved from Schneider Electric: https://product-help.schneider-electric.com/Machine%20Expert/V1.1/en/m251sys/m251sys/glossary/glossary.htm

Siemens. (2012, November 27). *SIMATIC S7-1500 CPUs*. (Siemens) Retrieved from Siemens Products and Services: https://new.siemens.com/global/en/products/automation/systems/industrial/plc/simatic-s7-1500/cpus.html

Stereo Labs Inc. (2020, January). *Capturing VR in Mixed Reality*. Retrieved from Unity - Green Screen VR Capture: https://www.stereolabs.com/docs/unity/green-screen-vr/

The Modbus Organization. (2021). *Modbus FAQ: About the Protocol*. Retrieved from Modbus: https://www.modbus.org/faq.php?Print.x=36&Print.y=9

U.S. Department of Commerce. (2015, May). *Human-Machine Interface (HMI) - Glossary*. Retrieved from National Institute of Standards and Technology: https://csrc.nist.gov/glossary/term/Human_Machine_Interface

University of Ovideo. (2007). *Sequential Control Systems - Subject I.* Retrieved from Systems and Automation Engineering: http://www.isa.uniovi.es/~idiaz/SA/Teoria/SA.Tema1.pdf

VIVE VR. (2021). *Connecting the headset to your computer*. Retrieved from VIVE: https://www.vive.com/ca/support/vive/category_howto/connecting-the-headset-to-your-computer.html

Yole Développement. (2020, May 4). A bright future? Displays and optic innovations transform the AR & VR industries. *Displays and optics for AR & VR 2020*. (Y. Développement, Ed.) Lyon, Rhone-Alpes, France. Retrieved from http://www.yole.fr/iso_upload/News/2020/PR_Displays_Optics_AR_VR_YOLEGROUP_April2020.pdf

7. APPENDIX Case Study: Implementation of Virtual Scenario into an Undergraduate Course at Morehead State University

Once the finished virtual environment was tested by the researcher, it became necessary that the performance of said environment was tested by a larger sample group. A suggestion that resulted from the main advisor of this project was the inclusion of this virtual platform as part of an undergraduate course assigned within the Spring 2021 catalog.

The chosen group belongs to the SE 488 Automation Systems course, in which students majoring in Systems Integration Engineering receive education in the use of automation technologies, such as PLC devices and industrial communication protocols. The basis of this course matched the criteria needed for each participant to offer feedback to the researched about their experience with the virtual scenario.

Figure 55. SE 488 students working on ladder logic diagrams

For this course, students began learning the fundamentals of PLC programming, from the very definition of a PLC, to the different programming languages used to establish the behavior of the device's input and output signals. The students also reviewed and implemented the basic ladder diagram elements needed to design the same main routine obtained by the researcher using a different PLC brand. This step served the purpose of comparing the response that could be obtained from different PLC hardware suppliers, which turned out to be minimal.

Finally, the students were able to manipulate the finished virtual environment as part of their final practices. The students then proceeded to implement their built-in routine into the VR environment and manipulate all its interactable elements, while understanding how the automated system sends the appropriate signals to the PLC coding software. This step resulted fundamental mainly because, as one student said, "given the current global pandemic, it would have been really to observe these types of processes in person, and with this option, it serves its purpose of showing how quick and seamlessly a certain process can be controlled."

Figure 56. Student manipulating interactable objects within the VR manufacturing environment

When the researcher asked the students about their own feedback, most of them agreed that the performance of the VR environment and the first-person navigation resulted manageable and very friendly-user. However, there were certain observations that are considered as important feedback, the most notorious being the initial delay obtained when switching from a first-person perspective to the HMD stream.

Another important observation noted by the students was that they felt certain GameObject movements to be drastically slow when compared to other virtual objects. The slowest object turned out to the 3D models of the Pegasus robotic arms. This can be explained due to the fact that it is the object with the largest number of textures included, and the one object whose mesh included the largest number of polygons.

As a final conclusion of this case study, the implementation of the first version of the virtual environment resulted convincing enough for people who had different approaches to both VR technology and automation technologies. The noted observations can be reworked in the second version of the platform, in the hopes that the following test group experience an even better performance.