Vol 27 (1)

Ingeniería y Competitividad
Vol 27 (1) 2025 doi: 10.25100/iyc.v27i1.14418

ISSN
0123-3033
e- 2027-8284

Original Article

Virtual Reality platform for Cartesian manipulation of digital twin in a UR3 robot

Plataforma en realidad virtual para la manipulación cartesiana del gemelo digital de un robot UR3

Ivan D. Ortiz [1] Oscar A. Vivas z[1]

[1]Universidad del Cauca, Popayán Colombia

How to cite?

Ortiz, I.D., Vivas, O.A. Virtual Reality platform for Cartesian manipulation of digital twin in a UR3 robot. Ingeniería y Competitividad, 2025, 27(1)e-20314418
https://doi.org/10.25100/iyc.v27i1.14418

Recibido: 19-09-24
Evaluado: 28-10-24
Aceptado: 21-11-24
Online: 30-01-25

Correspondence

avivas@unicauca.edu.co

Abstract

Introduction: the development of trajectory programming and control platforms for collaborative robots has significantly advanced with the integration of emerging technologies such as virtual reality.
Objetive: the primary objective of this study is to develop and implement a virtual reality platform that enables the programming and control of trajectories for a UR3 robot. Additionally, the study aims to improve system accessibility and usability by allowing users to define Cartesian trajectories through visual interaction and by integrating a digital twin of the robotic arm.
Methodology: the system was developed using Unity 3D, enabling the creation of an interactive visual interface for users. A digital twin of the UR3 robot was integrated, which synchronizes with the Meta Quest 2 VR headset to provide an immersive experience. Users can define linear trajectories by placing control points, with the ability to easily add or delete points within the virtual environment.
Results: the implementation of the platform allowed users to effectively define and control trajectories within a virtual reality environment. It was observed that users were able to interact with the system intuitively, creating trajectories without prior robotic programming knowledge. Moreover, the use of the digital twin provided an accurate real-time visual representation of the robot's behavior..
Conclusions: this study demonstrates that the integration of virtual reality with collaborative robot trajectory control enhances user accessibility and interaction with the system. This approach not only facilitates learning and trajectory programming but also lays the foundation for future improvements in the graphical interface and control functionalities, enabling system customization according to specific user needs.

Keywords: Virtual Reality, System Trajectory Programming, Industrial Robotics and Digital Twin.

Resumen

Introducción: eEl desarrollo de plataformas de programación y control de trayectorias para robots colaborativos ha avanzado significativamente con la incorporación de tecnologías emergentes, como la realidad virtual.
Objectivo: eEl objetivo principal de este estudio es desarrollar e implementar una plataforma en realidad virtual que permita la programación y el control de trayectorias de un robot UR3. Además, se busca mejorar la accesibilidad y usabilidad del sistema, haciendo posible la definición de trayectorias cartesianas a través de la interacción visual y la integración de un gemelo digital del brazo robótico.
Metodología: el sistema fue desarrollado utilizando el motor de Unity 3D, permitiendo la creación de una interfaz visual interactiva para los usuarios. Se integró un gemelo digital del robot UR3, que se sincroniza con el casco de realidad virtual Meta Quest 2 para ofrecer una experiencia inmersiva. Los usuarios pueden definir trayectorias lineales mediante la colocación de puntos de control, pudiendo agregar o borrar puntos de forma intuitiva en el entorno virtual.
Resultados: la implementación de la plataforma permitió a los usuarios definir y controlar trayectorias de forma eficaz en un entorno de realidad virtual. Se observó que los usuarios pudieron interactuar con el sistema de manera intuitiva, creando trayectorias sin la necesidad de conocimientos previos en programación robótica. Además, el uso del gemelo digital proporcionó una representación visual precisa del comportamiento del robot en tiempo real.
Conclusiones: el estudio demuestra que la integración de la realidad virtual con el control de trayectorias de un robot colaborativo mejora la accesibilidad y la interacción de los usuarios con el sistema. Este enfoque no solo facilita el aprendizaje y la programación de trayectorias, sino que también sienta las bases para futuras mejoras en la interfaz gráfica y en las funcionalidades de control, permitiendo una personalización del sistema según las necesidades específicas de los usuarios.

Palabras clave: Realidad Virtual, Programación de Trayectorias, Robótica Industrial y Gemelo Digital.

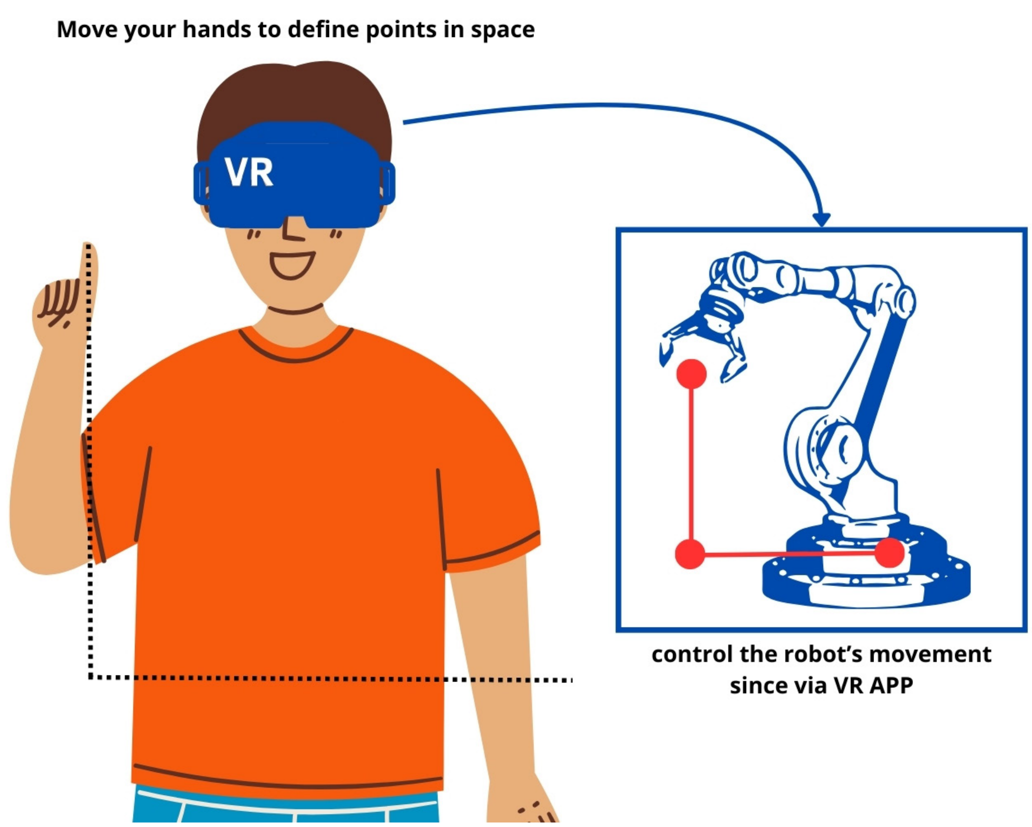
Contribution to the literature

Why was it done?

The study focused on developing and implementing a virtual reality (VR) platform to support the programming and trajectory control of the UR3 collaborative robot, which has six degrees of freedom. This work aligns with the principles of Industry 4.0, emphasizing the integration of cyber-physical systems and emerging technologies like VR to enhance human-robot interaction. The research aims to improve accessibility for new users by providing an intuitive system that enables the visual definition and modification of Cartesian trajectories, setting down a way for future advancements and customizations.

What were the most relevant results?
The most significant outcome was the implementation of a system enabling robot control in both joint and Cartesian modes. This was achieved using a digital model developed in SolidWorks and transferred to Unity 3D via Blender. Integration with virtual reality devices, such as the Meta Quest 2, provided an intuitive interface for creating and modifying trajectories, allowing users to evaluate movements from multiple perspectives. The application supports joint and Cartesian modes within the VR environment, enabling precise adjustment of each robot joint using the Calculated Torque Control (CTC) algorithm with adequate accuracy.

What do these results contribute?
These results contribute to enhancing the accessibility and usability of path programming systems, particularly for users with limited experience in industrial robotics. The platform offers a safe and controlled environment for programming and validating movements before physical implementation, thereby minimizing risks and optimizing processes. Additionally, it



doi: 10.25100/iyc.v27i1.14418

Introduction

Industry 4.0 is one of the most prominent technological trends worldwide, characterized by integrating of cyber-physical systems into logistics and manufacturing processes. This trend arises due to of the development, use, and adoption of emerging technologies, such as the use of robotics and virtual reality systems in industry (1) (2). Its primary goal is to use advanced technologies to assist humans in making decisions and performing tasks that may be exhausting, tedious, or repetitive (3) (4). One way to integrate robots into industry is through collaborative robots, also known as Cobots (5).

Cobots can be easily programmed using trajectory programming systems, which play a critical role in automation by enabling the programming and control of robot motion in production lines using visual tools (6). These visual systems allow operators to define precise trajectories by specifying points in virtual space that the robot can later emulate in physical space, all without having to write a single line code, this allows for better interaction between the operator and the robot/machine (7).

As a result, these systems are considered user-friendly because they do not rely on code to operate and do not require expert programmers to operate (8). In addition, the system must have a good immersion system to fix the trajectories of the robot, allowing a more accurate dimensioning of the workspace within a virtual environment (9). In this context, Virtual Reality (VR) emerges as the ideal solution for robot programming and control (10), as it allows a more natural and precise interaction with the virtual environment (11), facilitating trajectory programming and the visualization of movements in an immersive three-dimensional space (12) (13).

Because of the above, it is appropriate to integrate the use of trajectory execution systems for Cobots with virtual reality systems, as these are immersive and offer easy-to-understand models (13). Unlike traditional designs based on 2D systems (tablet or computer screen), three-dimensional systems (VR goggles) allow better analysis of the robot's movements and design, providing a broader and more detailed perspective through navigation in the virtual world. This makes it possible to fully analyze the designed model and its workspace, and to program Cartesian points to execute trajectories that connect them in three dimensions (14). In this way, a new perspective is offered from the operator's point of view, as this new approach allows errors to be detected and corrected before moving on to the production phase (11).

Related work

In the last decade, there has been a continuous and growing interest in research on trajectory programming systems. Various researchers have presented different approaches and methods for implementing virtual/augmented reality technologies within these systems. For example, in (15), augmented reality (AR) is used to project a robot onto a given surface through a smartphone, and it performs a movement using markers that indicate the start and end points. In the same way, in (16), this technology is used with a tablet to validate different human-robot interactions in an automotive environment, highlighting the discomfort of the operator who has to hold the tablet while using a tool. This issue, although not mentioned in the document, is crucial since the user must hold the phone at all times, making this technology one of the least viable for implementation.

doi: 10.25100/iyc.v27i1.14418

On the other hand, among the technologies that use mixed reality (MR), (17) offers certain advantages over virtual reality (VR), allowing the real space to be accurately mapped within the virtual space. Users can directly perceive and interact with the system's input information, spatial sound, and location by combining virtual objects within the real world with which they can interact (10). However, this could also be considered an advantage of virtual reality, as this technology operates in a controlled environment, preventing external factors such as sunlight or the distance between the glasses and the object from affecting the trajectory programming. In addition, it is always possible to replicate real objects in virtual space with the same dimensions, as shown in (11). Nevertheless, according to (18), when using a mixed reality system, the digital twin seen through the glasses must be positioned over the same robot visible in reality, which can cause uncomfortable overlapping for the operator.

Consequently, virtual reality systems offer unique features. As mentioned in (19) and (20), an advantage of VR systems is their use in teleoperation tasks, allowing robots to be controlled without physical presence, which facilitates industrial safety as mentioned in (21). The operator can control the robot without risk and without stopping production on the assembly line (22). In addition, the consistent support of major technology companies and the decreasing cost of virtual reality glasses have encouraged the use of virtual reality applications. In recent years, research has shown a trend toward this type of technology, as reflected in articles (23), (24), (25) and (26), where virtual reality glasses are used together with digital twins that replicate the movements of a manipulator robot. In (23), a KUKA KR10 robot is used; in (24), a FANUC M-10iA robot; and in (25) and (26), a Mitsubishi Movemaster RV-M1 robot to create their digital equivalents in virtual space, developing applications in an immersive environment that adequately represents their movements, which will be replicated in real space.

Thus, the exploration of VR, AR or MR technologies has allowed the identification of various opportunities and challenges in trajectory programming, with applications in several industrial fields. Augmented reality, although promising, has practical limitations that hinder its viability, especially in the need for portable devices and the lack of comfort they offer when used with other tools. On the other hand, mixed reality offers a more precise integration of the real and virtual environments, but still faces technical challenges such as visual overlap and limited control over different environments. In this way, virtual reality, with its controlled environments and growing support from large corporations, stands out as a solid alternative, especially in the field of teleoperation and industrial safety. The growing trend in the research and application of VR systems, as evidenced by recent studies, suggests a significant potential for the development of innovative and efficient solutions in robot automation and control. In this context, the use of digital twins in immersive environments positions itself as a key tool to optimize human-robot interaction and improve the efficiency of production processes.

Methodology

For the experimental setup used in this project, the activities have been divided into two main components. The first component involves the use of tools to build the digital model of the robot. The second component focuses on the tools for building the trajectory programming system, which is the main focus of this document.

doi: 10.25100/iyc.v27i1.14418

Digital model creation

To create the digital model, it is essential to use a tool that can create three-dimensional models. SolidWorks was selected as the tool of choice for several reasons. First, its ease of use is highlighted by its intuitive interfaces. Second, its extensive 3D object modeling capabilities allow for detailed assembly design. Finally, its robustness in exporting products in various file formats makes it the ideal choice.

Thus, this tool is used to design the robotic arm that will be integrated into the graphics engine, inspired by the design of a UR3 robot (Figure 1) from Universal Robots with six degrees of freedom. The result is shown in Figure 2. In this assembly, each section is divided into folders named after the corresponding axis (Base, Axis1, Axis2, Axis3, Axis4, Axis5, and Axis6) to better identify the parts that belong to each axis of the robot's rotation. This allows them to be exported independently, which is needed later in the graphics engine to generate the rotations of each axis, thus building a more functional model.

Figure 1 Universal Robots UR3 compact Cobot **Fuente:** Universal Robots

doi: 10.25100/iyc.v27i1.14418

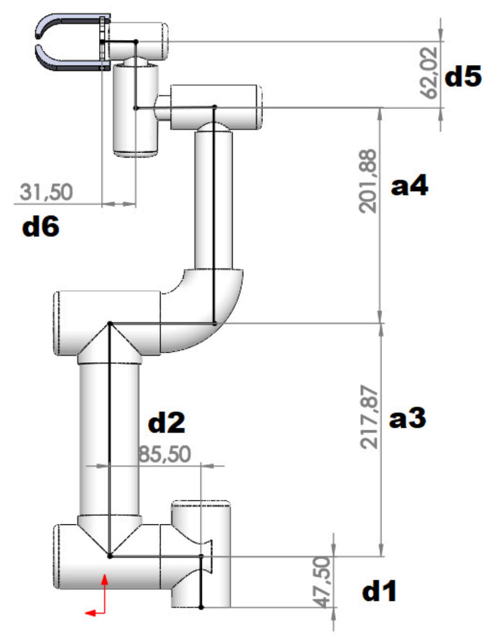


Figure 2 Six Degrees of Freedom robot arm design

Since SolidWorks does not have a direct export tool for Unity, Blender is used as an intermediary tool to read the files in the graphics engine. The objects created in the CAD tool are first exported in PLY (Polygon File Format) and then opened in Blender, which can read this type of file. Within this program, the first step is to define the origin of the piece, which will also be considered its axis of rotation, essential for generating rotational motion. Finally, considering the scales that will be used in Unity, the value in the X, Y and Z boxes is set to 0.001, with these new dimensions the object is small enough to be manipulated and visualized in the VR project. The configurations made for the piece can be seen in Figure 3, which replicates the same configuration for the other six objects created and exported in FBX (FilmBox) format to open and work with them within the graphics engine.

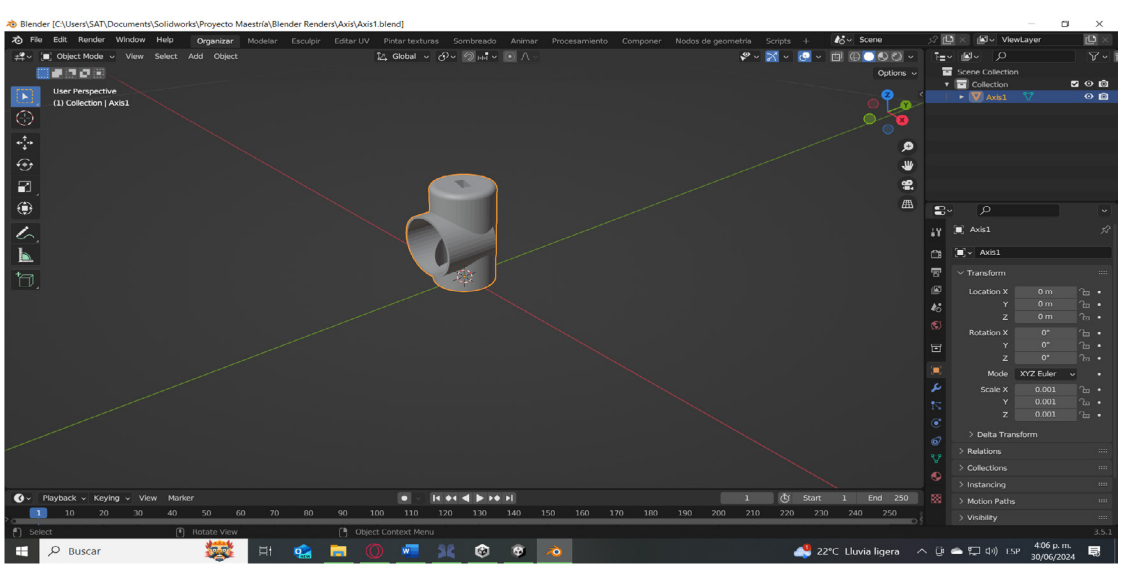


Figure 3 Modifications applied for exporting Game Objects

doi: 10.25100/iyc.v27i1.14418

System trajectory program

To develop the application, the flowchart shown in Figure 4 is created, corresponding to the VR application designed for programming and controlling industrial robots. This diagram illustrates the process from the start of the application to the execution of actions on the robot, highlighting the available functionalities in the articular and cartesian modes of operation.

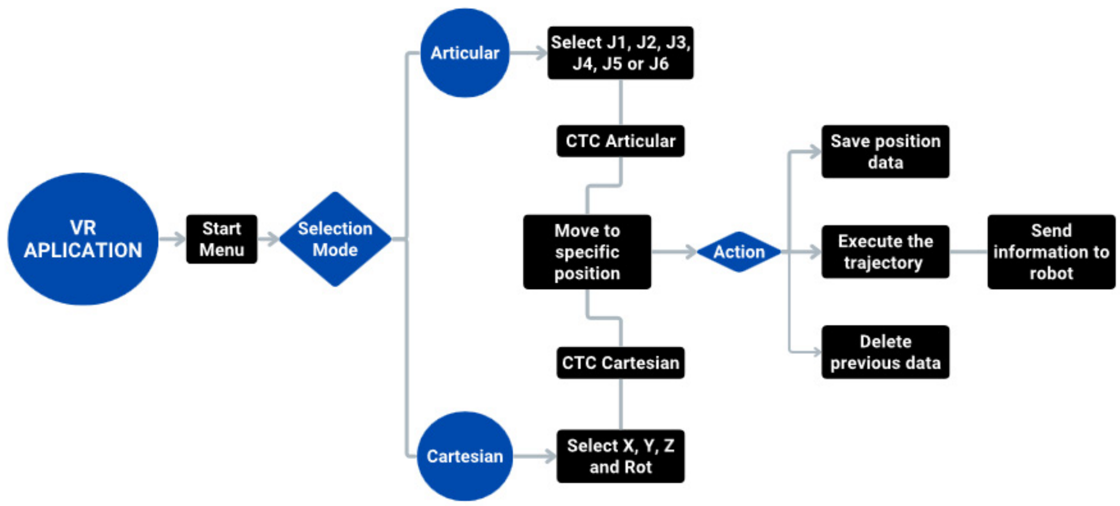


Figure 4. VR application flowchart

When starting the application, the user is guided through a start menu from which the desired operation mode is selected. The VR app provides two main control modes:

Articular Mode: This mode allows the user to select one of the robot's six joints (J1, J2, J3, J4, J5, or J6) for adjustment. Once the joint is selected, the Calculated Torque Control (CTC Joint) algorithm is used to calculate and adjust the Euler angles necessary to achieve the desired orientation. The robot is then moved to the specific position determined by these calculations.

Cartesian Mode: Alternatively, the Cartesian mode allows the user to enter the X, Y and Z coordinates and the desired rotation of the end effector. To convert these coordinates into instructions that the robot can execute, the Calculated Torque Control (CTC Cartesian) algorithm is used to solve the inverse kinematics and calculate the necessary joint angles. The robot is then moved to the specified Cartesian position.

Once the robot has been correctly positioned, the system offers three possible actions:

- Save the position data, allowing it to be stored for future executions.
- Delete previous data if adjustments are required for the robot's programming.

Send the calculated data directly to the robot, including the transfer of information to the physical device, if present, and the execution of the programmed trajectory within the VR application.

This workflow reflects the application's ability to provide precise and flexible control of industrial

doi: 10.25100/iyc.v27i1.14418

robots, integrating advanced trajectory control methods in both articular and cartesian space. To achieve this level of precision and flexibility, Unity 3D is used to create the virtual environment. Unity 3D is a graphics engine known for its ease of use through its editor and intuitive tools. In addition, it offers robust virtual reality (VR) application development capabilities through the Open XR library, which is essential for developing this system. Its compatibility with most virtual reality devices and support from a large community of developers, who provide extensive information and documentation make Unity an ideal tool for solving specific problems and optimizing complex projects in this field.

To integrate the system with the 3D design created in Figure 2, a three-dimensional environment is first created using Unity's editor capabilities to build the virtual world using the ProBuilder tool, developing an immersive environment that includes a workspace based on a laboratory, as shown in Figure 5. This lab contains a table on which the robotic arm will be placed, and using the Open XR tool, a pair of controllers will be integrated for the use of tools and Unity systems that interact with them to apply changes and movements to the robot.

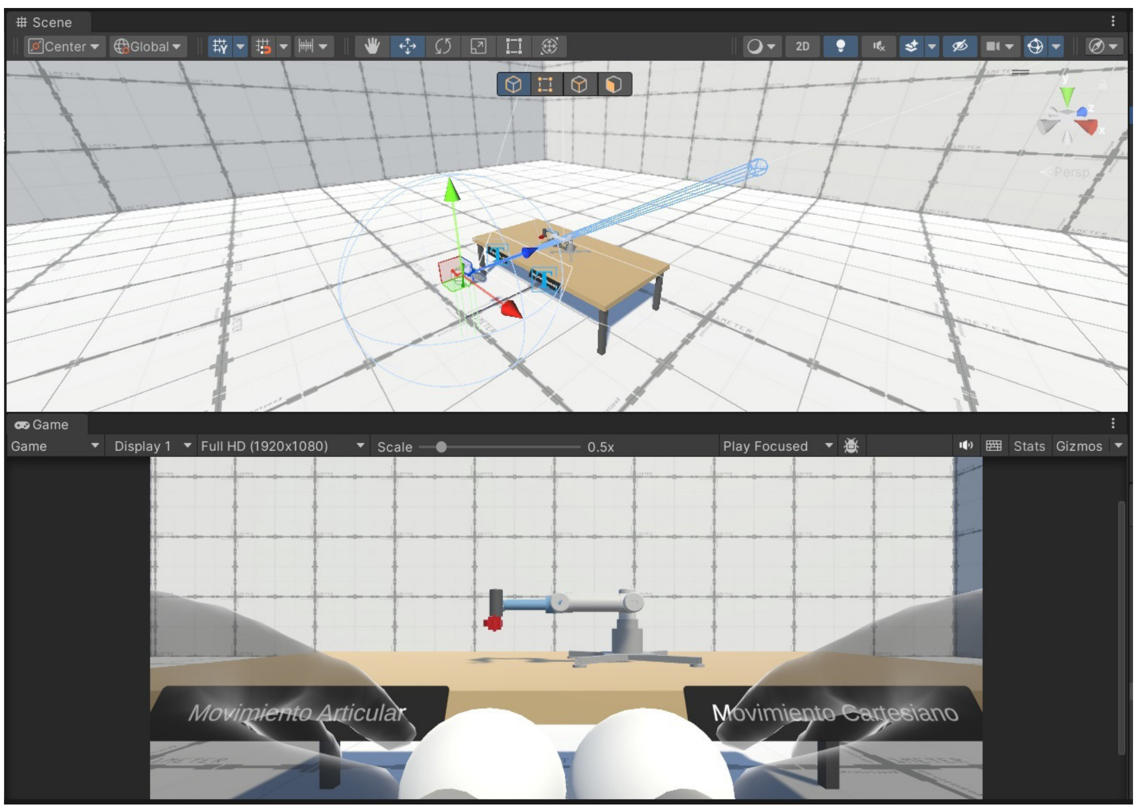


Figure 5 Creating the virtual environment

To control the system, the Meta Quest 2 virtual reality headset is used, a device known for providing a high-quality virtual reality experience that is highly portable and easy to use. It includes two controllers and a headset with a Qualcomm Snap Dragon XR2 processor, 6 GB of RAM, 128 GB of storage, a resolution of 1832 x 1920 per eye, a refresh rate of 60, 72, and 90 Hz, and supports Wi-Fi and Bluetooth connections (27).

The system shown in Figure 6 aims to allow the user to define the cartesian trajectories of the robot using the controllers provided by the Meta Quest 2 system.

doi: 10.25100/iyc.v27i1.14418

Figure 6 Virtual reality headset and controls: Meta Quest 2

Results and discussion

To ensure that the system is capable of executing the requested trajectories, the robot was programmed to perform a series of predetermined movements to monitor the robot's performance and verify that the created trajectory system executes the requested trajectories, both in articular and cartesian coordinates.

Articular movement

For execution trajectories at articular space, a CTC (Computed Torque Control) was implemented, as shown in Figure 7. This control, developed in Simulink, regulates each of the robot's six joints during trajectory execution in the three-dimensional VR environment. The articular movement function is used to adjust each joint according to the specific requirements of a programmed trajectory.

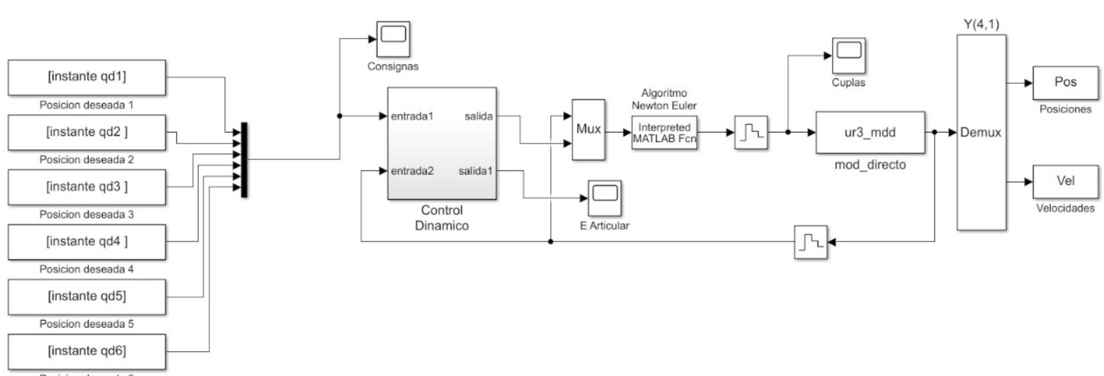


Figure 7 Articular CTC control

doi: 10.25100/iyc.v27i1.14418

The precision of the control system and the accuracy of the desired angles were evaluated by comparing the data obtained during the experiment with the original angles intended for each joint to confirm the level of precision present for each joint. This comparison is illustrated in Figure 8 and allows us to determine the error in articular space, which, according to the graph, shows a greater degree of error in the fourth joint, with an error of $15x10^{-4}$ radians (less than two milliradians).

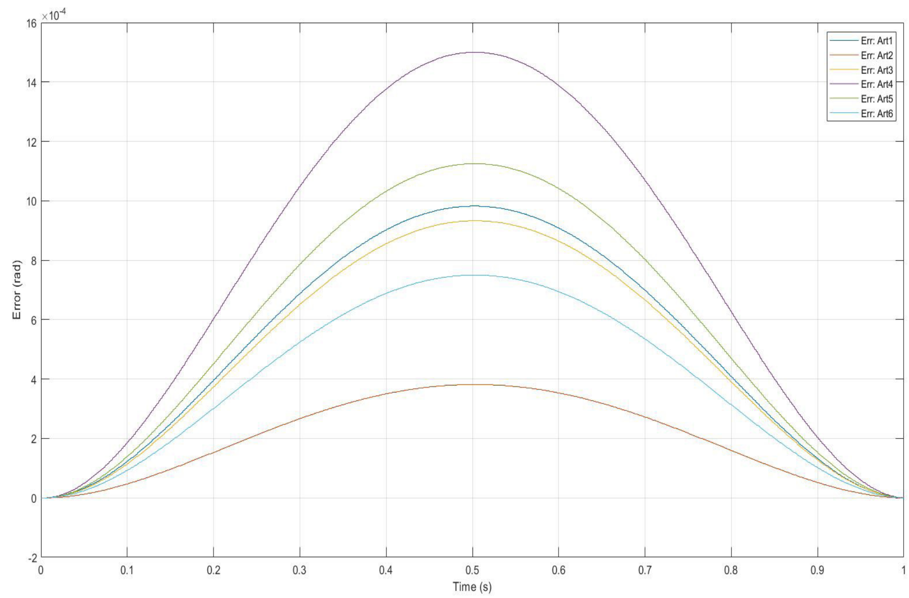


Figure 8 Articular Error

The obtained data was integrated into the articular movement system, allowing precise programming of each point in the trajectory. Figure 9 shows how the slider is manipulated to adjust the rotation angle of the robot's first wrist before saving the trajectory point. This process ensures that the digital twin accurately follows the desired angle, allowing a preview of movement to be performed in virtual three-dimensional space. This is achieved through the use of the virtual reality headset and controllers, facilitating a comprehensive evaluation of the movement from all possible perspectives.

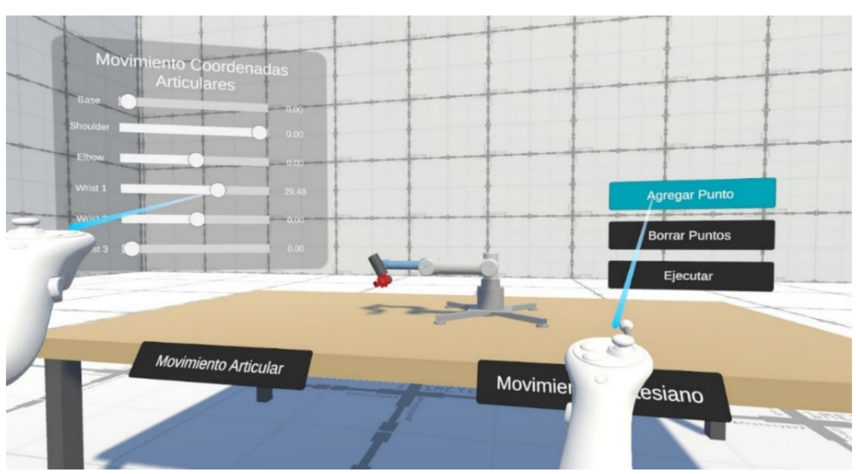


Figure 9 Generating points from articular coordinates

doi: 10.25100/iyc.v27i1.14418

Cartesian movement

In this case, the implementation of trajectories is managed using cartesian CTC control, which offers significant advantages in terms of precision and adaptability to different load conditions, by regulating the movements of each robot joint. The control implemented via Simulink is shown in Figure 10, which allows the precise inclusion of values previously obtained from the CAD model of the robotic arm, designed in SolidWorks.

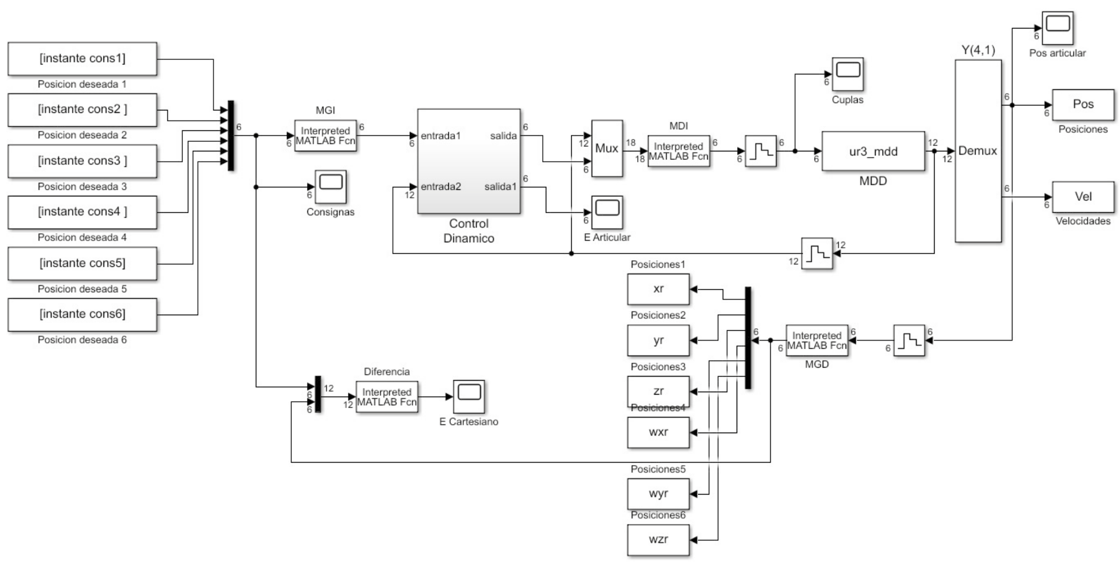


Figure 10 Cartesian CTC Control

Once the robot data is entered into Simulink, the panel for controlling the robot using Cartesian coordinates is displayed, as shown in Figure 11. Using the controllers in the virtual world, it is possible to manipulate the coordinates of the end effector, which determines the point to which the robot must move. This allows the desired points to be added along the entire trajectory to begin the test.

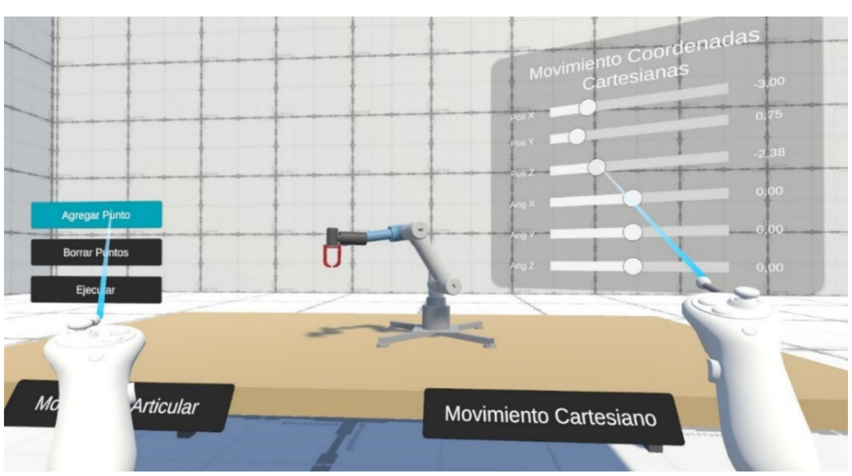


Figure 11 Generating points from cartesian coordinates

doi: 10.25100/iyc.v27i1.14418

This data is programmed into the three-dimensional environment shown in Figure 11, and when the Execute button is pressed, the digital twin will follow the trajectories in the three-dimensional environment using the previously programmed X, Y, Z coordinates and orientations, ensuring at all times that the robot is following the planned path. After the programmed motion is executed, the data captured where the last joint was positioned is compared with the desired positions, and by calculating this difference, the resulting error is obtained for each dimension.

To evaluate the precision of the control system and the accuracy of the executed trajectories, the data obtained during the experiment were compared with the original desired positions. This comparison, shown in Figure 12, shows the position error for each of the signals on the left, and the orientation error on the right. Starting with Figure 12 (a), it is observed that the maximum Cartesian position error is on the order of $1x10^{-3}$ metros (one millimeter) and the Cartesian orientation error, seen in Figure 12 (b), is on the order of $5x10^{-5}$ radians (less than 50 microradians). This level of accuracy indicates that the implemented CTC control is relatively low for use in industrial applications.

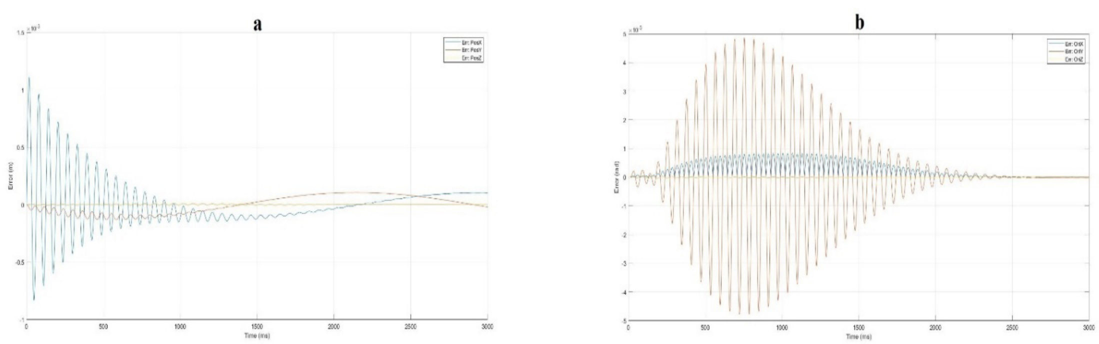


Figure 12 Cartesian error position (a) and orientation (b)

Therefore, we can consider that the use of VR devices in trajectory programming systems would facilitate the intuitive creation of Cartesian trajectories, allowing users to easily add and modify desired points. This approach could improve the accessibility and usability of trajectory programming systems for new users and provide a robust framework for future improvements and customizations.

Conclusions

This paper presents an application developed in Unity 3D for handling and controlling the trajectories of a digital twin. SolidWorks was used for modeling, creating a six-degrees-of-freedom model based on Universal Robots' UR3. Subsequently, parts were exported to Unity via Blender, allowing for effective integration into the virtual environment. In addition, a control system was used in both articular and cartesian modes using the CTC algorithm to achieve sufficiently precise movements.

doi: 10.25100/iyc.v27i1.14418

The application developed in Unity implemented the use of a virtual reality system through the Meta Quest 2 headset and controllers, which allowed interaction with the digital twin through buttons and sliders. In the articular mode, each joint was controlled individually, while in the cartesian mode, points were defined in both position and orientation for X, Y, and Z at final joint.

In addition, the development of this system demonstrated a flexible approach to trajectory programming, highlighting the usefulness of virtual reality in creating and modifying trajectories, as it allows the evaluation of the point's position in space from all possible angles, allowing the pre-planning of trajectory execution. Therefore, future work proposes to build and test the system on a physical prototype with a six-degree-of-freedom robot to evaluate its performance with Cartesian movements and to compare the trajectories programmed by the user with those executed by the real robot.

CRediT authorship contribution statement

Conceptualization - Ideas: Oscar A. Vivas. Data Curation: Ivan D. Ortiz. Formal analysis: Ivan D. Ortiz. Acquisition of funding: Oscar A. Vivas. Investigation: Ivan D. Ortiz, Oscar A. Vivas. Methodology: Ivan D. Ortiz, Oscar A. Vivas. Project Management: Oscar A. Vivas. Resources: Oscar A. Vivas. Software: Ivan D. Ortiz, Oscar A. Vivas. Supervision: Oscar A. Vivas. Validation: Ivan D. Ortiz. Visualization - Preparation: Ivan D. Ortiz, Oscar A. Vivas. Writing - original draft - Preparation: Ivan D. Ortiz, Oscar A. Vivas. Writing - revision and editing - Preparation: Ivan D. Ortiz, Oscar A. Vivas.

Financing, Conflict of Interest, Etics Implications: doest not declare.

References

1. Ministerio de las tecnologías de la información y la telecomunicación - Aspectos básicos de la industria 4.0. [Online].; 2019 [cited 2023 03 23]. Available from: https://colombiatic.mintic.gov.co/679/w3-article-124767.html.

2. Rejikumar G, Raja V, Arunprasad P, Persis J, Sreeraj K. Industry 4.0: key findings and analysis from the literature arena. Benchmarking: An International Journal. 2019. https://www.emerald.com/insight/content/doi/10.1108/bij-09-2018-0281/full/pdf

3. Moran R, Teragni M, Zabala G. Physical Bits: A Live Programming Environment for Educational Robotics. In Lepuschitz W, Merdan M, Koppensteiner G, Balogh R, Obdržálek D, editors. Robotics in Education; 2021; Cham: Springer International Publishing. p. 291-303. https://doi.org/10.1007/978-3-030-67411-3_26

4. Mital A, Pennathur A. Advanced technologies and humans in manufacturing workplaces: an interdependent relationship. International Journal of Industrial Ergonomics. 2004; 33: p. 295-313. https://doi.org/10.1016/j.ergon.2003.10.002

5. ESIC. Robótica Colaborativa: Qué es y Qué papel tienen las personas. [Online]. [cited 2023 03 04]. Available from: https://www.esic.edu/rethink/tecnologia/robotica-colaborativa-papel-tienen-las-personas.

doi: 10.25100/iyc.v27i1.14418

6. Kohrt C, Stamp R, Pipe A, Kiely J, Schiedermeier G. An online robot trajectory planning and programming support system for industrial use. Robotics and Computer-Integrated Manufacturing. 2013. https://doi.org/10.1016/j.rcim.2012.07.010

7. Brito T, Queiroz J, Piardi L, Fernandes L, Lima J, Leitão P. A Machine Learning Approach for Collaborative Robot Smart Manufacturing Inspection for Quality Control Systems. Procedia Manufacturing. 2020. https://doi.org/10.1016/j.promfg.2020.10.003

8. Zhu Z, Hu H. Robot Learning from Demonstration in Robotic Assembly: A Survey. Robotics. 2018. https://doi.org/10.3390/robotics7020017

9. Cohavi O, Levy-Tzedek S. Young and old users prefer immersive virtual reality over a social robot for short-term cognitive training. International Journal of Human-Computer Studies. 2022. https://doi.org/10.1016/j.ijhcs.2022.102775

10. Litvinova Y, Rilke RM, Guenther C. Me, myself, and I: Image concerns and honesty in immersive VR. Computers in Human Behavior. 2023 December; 149: p. 107950. https://doi.org/10.1016/j.chb.2023.107950

11. Holubek R, Ružarovský R, Velíšek K, Janíček M. Novel trend in case study of industrial robot programming and production system design using the virtual reality. IOP Conference Series: Materials Science and Engineering. 2021. https://doi.org/10.1088/1757-899X/1009/1/012023

12. Park BJ, Hunt SJ, Martin C, Nadolski GJ, Wood BJ, Gade TP. Augmented and Mixed Reality: Technologies for Enhancing the Future of IR. Journal of Vascular and Interventional Radiology. 2020; 31: p. 1074-1082. https://doi.org/10.1016/j.jvir.2019.09.020

13. Bolano G, Roennau A, Dillmann R, Groz A. Virtual Reality for Offline Programming of Robotic Applications with Online Teaching Methods. In 17th International Conference on Ubiquitous Robots (UR); 2020. https://doi.org/10.1109/UR49135.2020.9144806

14. Costa G, Petry M, Moreira A. Augmented Reality for Human-Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review. Sensors. 2022. https://doi.org/10.3390/s22072725

15. Fang H, Ong S, Nee A. Robot Path and End-Effector Orientation Planning Using Augmented Reality. Procedia CIRP. 2012. https://doi.org/10.1016/j.procir.2012.07.034

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

17. Ostanin M, Klimchik A. Interactive Robot Programing Using Mixed Reality. IFAC-PapersOnLine. 2018. https://doi.org/10.1016/j.ifacol.2018.11.517

18. Rosen E, Whitney D, Phillips E, Chien G, Tompkin J, Konidaris G, et al. Communicating Robot Arm Motion Intent Through Mixed Reality Head-Mounted Displays. In Robotics Research.: Springer International Publishing; 2019. p. 301-316. https://doi.org/10.1007/978-3-030-28619-4_26

19. Zhang T, Mccarthy Z, Jow O, Lee D, Chen X, Goldberg K, et al. Deep Imitation Learning for Complex Manipulation Tasks from Virtual Reality Teleoperation. In ; 2018. https://doi.org/10.1109/ICRA.2018.8461249

doi: 10.25100/iyc.v27i1.14418

20. Baklouti E, Jallouli M, Amouri L, Amor B. Remote control of mobile robot through 3D virtual reality environment. In International Conference on Individual and Collective Behaviors in Robotics; 2013. https://doi.org/10.1109/ICBR.2013.6729276

21. Kot T, Novák P. Application of virtual reality in teleoperation of the military mobile robotic system TAROS. International Journal of Advanced Robotic Systems. 2018 January; 15. https://doi.org/10.1177/1729881417751545

22. Ružarovský R, Holubek R, Delgado Sobrino DR, Velíšek K. A case study of robotic simulations using virtual commissioning supported by the use of virtual reality. MATEC Web of Conferences. 2019. https://doi.org/10.1051/matecconf/201929902006

23. Murhij Y, Serebrenny V. An application to simulate and control industrial robot in virtual reality environment integrated with IR stereo camera sensor. IFAC-PapersOnLine. 2019. https://doi.org/10.1016/j.ifacol.2019.12.473

24. Garg G, Kuts V, Anbarjafari G. Digital Twin for FANUC Robots: Industrial Robot Programming and Simulation Using Virtual Reality. Sustainability. 2021. https://doi.org/10.3390/su131810336

25. Crespo R, García R, Quiroz S. Virtual Reality Application for Simulation and Off-line Programming of the Mitsubishi Movemaster RV-M1 Robot Integrated with the Oculus Rift to Improve Students Training. Procedia Computer Science. 2015. https://doi.org/10.1016/j.procs.2015.12.226

26. Crespo R, García R, Quiroz S. Virtual Reality Simulator For Robotics Learning. In International Conference on Interactive Collaborative and Blended Learning (ICBL); 2015. https://doi.org/10.1109/ICBL.2015.7387635

27. Meta. Get started with Meta Quest 2. [Online].; 2020. Available from: https://www.meta.com/quest/products/quest-2/tech-specs/.

doi: 10.25100/iyc.v27i1.14418