USER MANUAL: ROBOT CONTROL UR WITH HOLOLENS 2

Collaboration Scholarship:
Álvaro Javier Orcajo Domingo

Course 2023-2024

INDEX

1. Introduction

In this collaboration scholarship, an application has been developed to innovate in the teaching of robotics through virtual reality, specifically using the Microsoft HoloLens 2 device. This application, created with the Unity development tool along with Microsoft's Mixed Reality Toolkit v3, will teach students about the different types of movements (MoveJ, MoveL, MoveP) available for a Universal Robots UR3e robot.

Additionally, through mixed reality, it will be possible to see the exact path trace made by the robot, thereby creating an immersive environment and unique experiences for the user, with significant information retention and therefore learning. This last statement has been demonstrated through my final degree project carried out during 2022-2023, the documentation of which can be seen in the following:

Final degree project – Álvaro Javier Orcajo Domingo

This project not only innovates in teaching but also enables controlling the robot in mixed reality, an innovative project that can serve as a foundation for other future projects. This also enhances the learning experience for both students and users worldwide in the field of communication via sockets between a UR robot and Mixed Reality or Unity.

Special thanks to Roman Parak´s project, which has been a valuable learning resource for socket connection.

2. Start-up and Configuration

First of all, to connect the robot and ensure it functions correctly, several factors need to be considered. The application has been designed to be as simple as possible, but configuring the device is necessary first.

For safety reasons, before making any robot movements using this application, it is recommended to set the speed to 35% on the Teach Pendant. Also, pay special attention to section 2.3. Robot Model.

Additionally, the gripper (if present) should initially be open and connected to the tool's digital output.

2.1. Type of conexion

There is a connection through an **Ethernet cable**, which requires a port on the computer for the cable. Both HoloLens 2 and other computers may not have this port, in which case a USB to Ethernet adapter should be used.

Communication can be established **wirelessly**, which, in my opinion, is the most convenient method, but the robot would need a connected router.

2.2. IP Configuration

If you want to use the application from the computer in Unity (it is intended for mixed reality but can be used from the computer), you will need to modify the IP (whether wireless or Ethernet) in the network settings on the computer.

On the other hand, if you are connecting to the HoloLens 2 (which is the usual method), you will need to configure the settings within the HoloLens 2 operating system. The following images show the parameters to be changed, with Figure 1 for the computer and Figure 2 for the HoloLens 2:

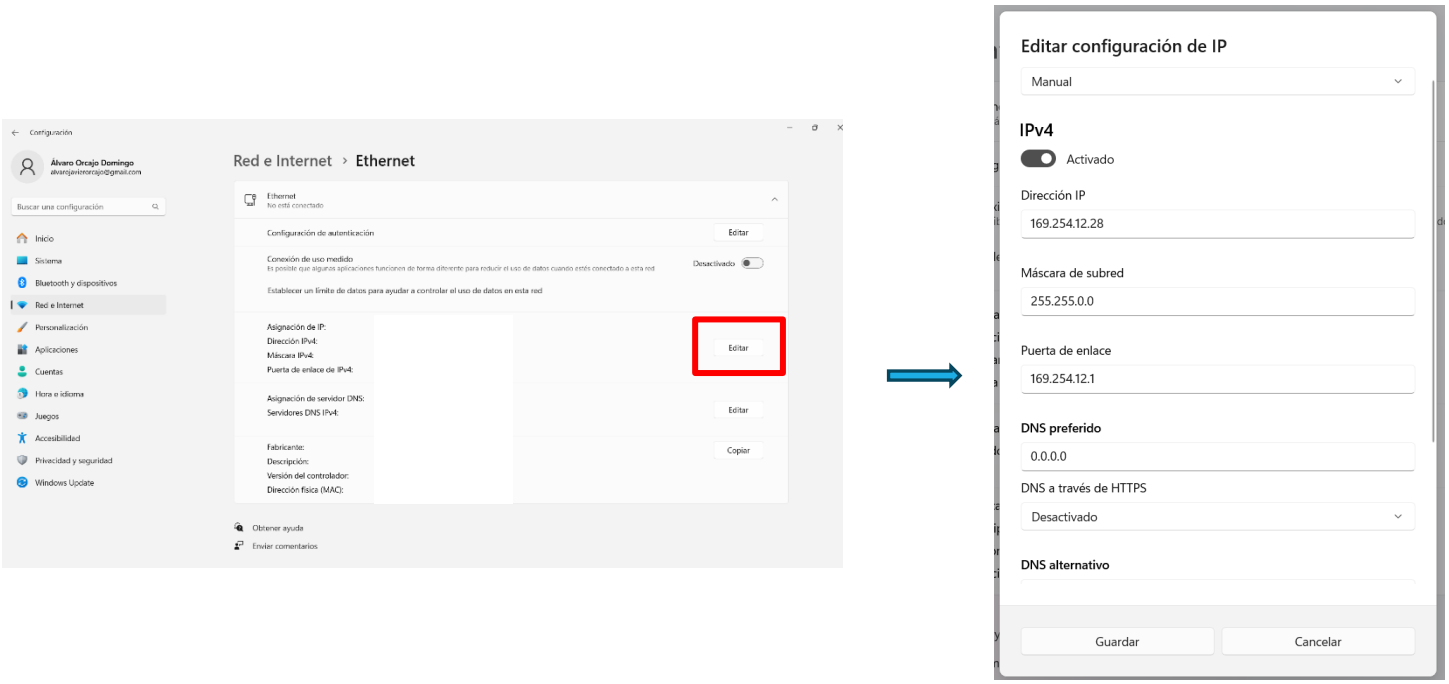

Figura 1. IP Configuration with computer

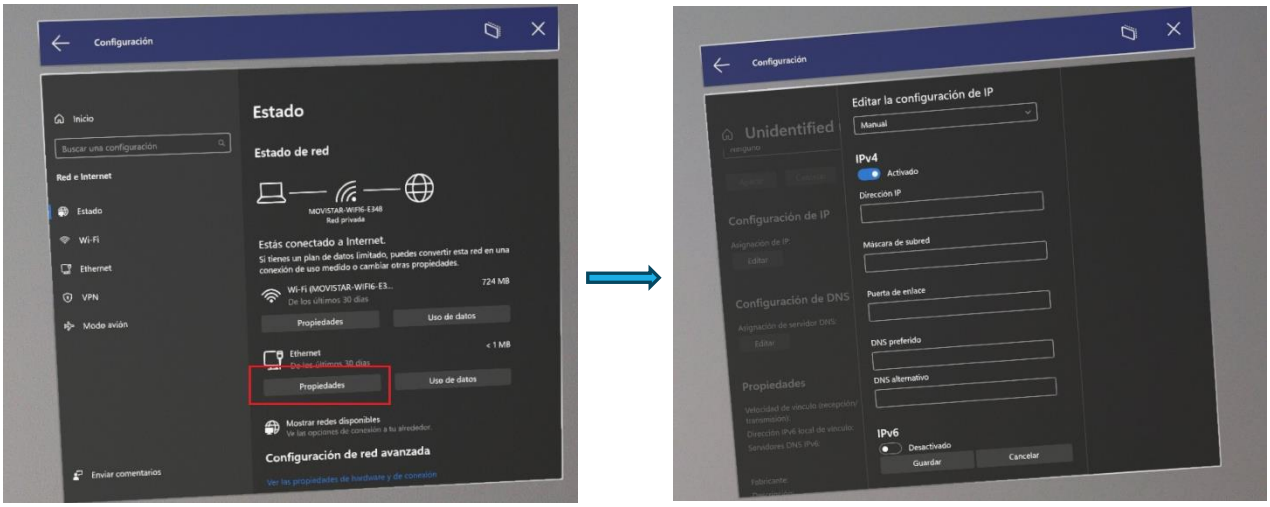

Figura 2. IP Configuration with HoloLens 2

Once the previous steps are completed, you need to specify the IP address within the application, which must match the device's IP address. An interface with a keyboard has been created for entering the IP address, but if the application will always be used for the same IP address, it is recommended to predefine it within Unity on the computer before compiling and debugging on the HoloLens 2, as this will be much more convenient. You can do this in: \Assets\UR_Control_Scripts\UIPanel_Control.cs within the "void Start()" function by setting the variable called “ip_address_txt.text” (on line 95 of the code). Once this is done, simply click the connect button on the connection panel.

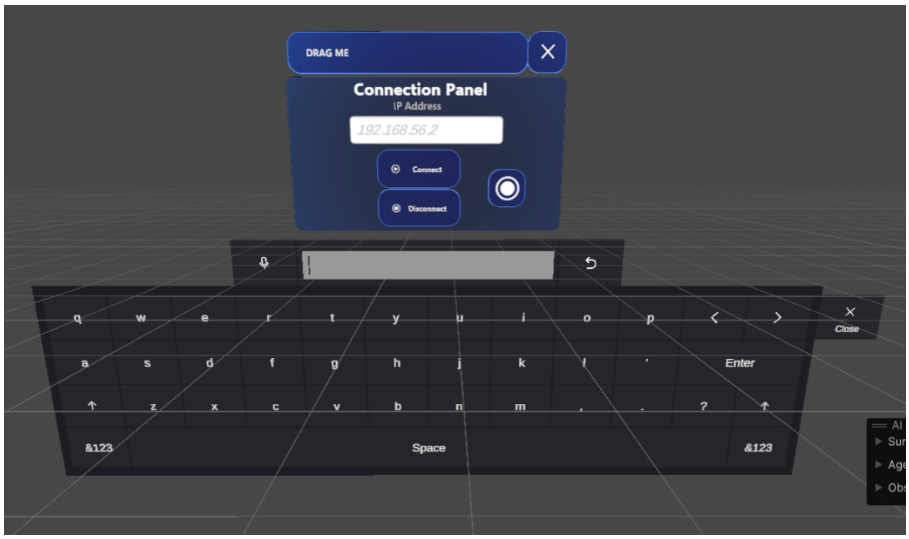

Figura 3. Connection Panel, where the IP address is entered

Additionally, there is another way to connect with the robot and the HoloLens 2: start the application on the computer from Unity and use the Holographic Remoting application on the HoloLens 2. This method eliminates the need to configure the IP on the HoloLens 2, as it can sometimes cause configuration errors on the device. In my experience, when I configured the IP address on the device, for an unknown reason, when switching back to automatic network settings, it did not configure automatically and continued with manual settings, leading to network issues due to an IP address that did not match the Wi-Fi. The issue was only resolved once I connected with Windows Connect.

2.3. Robot Model. Security

Using the model UR3e robot with this application is crucial and essential for both user safety and proper functionality. Using a different model could result in completely unpredictable movements, even if the kinematics were correctly configured, leading to errors similar to those shown in the following image:

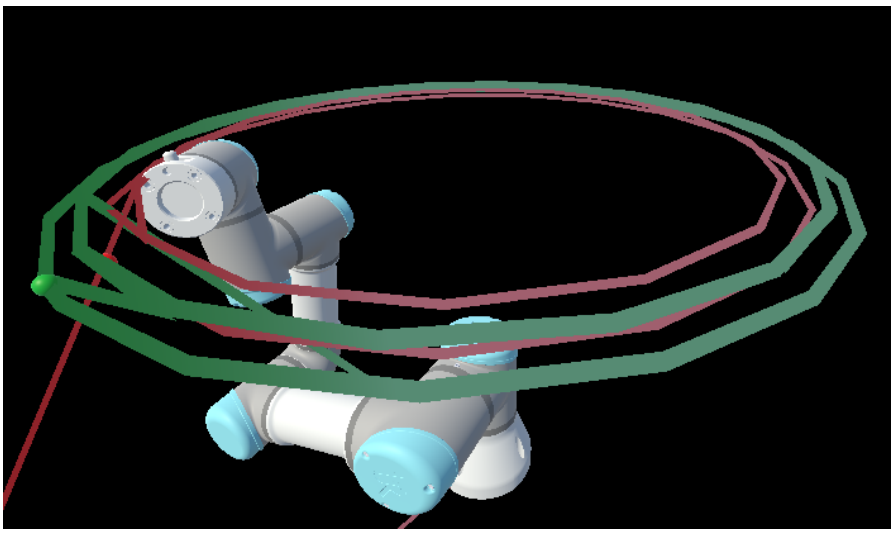

Figura 4. Error with UR3

Additionally, to ensure safety, a sphere representing the robot's reach has been created, and movements will only be sent if they are within this sphere.

2.4. Anchoring

Anchoring is important to have the correct coordinate origin. The anchoring used is manual and is placed at the base of the robot using the hands in mixed reality. The anchor is designed to be as simple and practical as possible. It is shown in the following image:

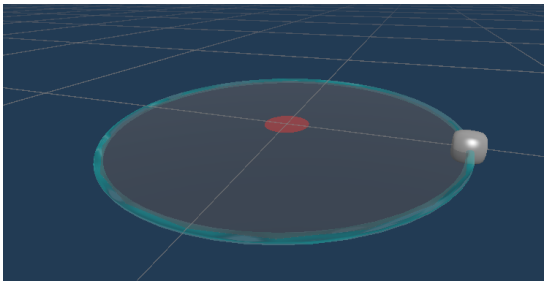

Figura 5. Object for Anchoring the Robot

This object can be manipulated with the hands, both from a distance and up close, as if it were a real object. The object should be placed at the base of the robot so that the base aligns with the red circle on the anchor object, as shown in the following image:

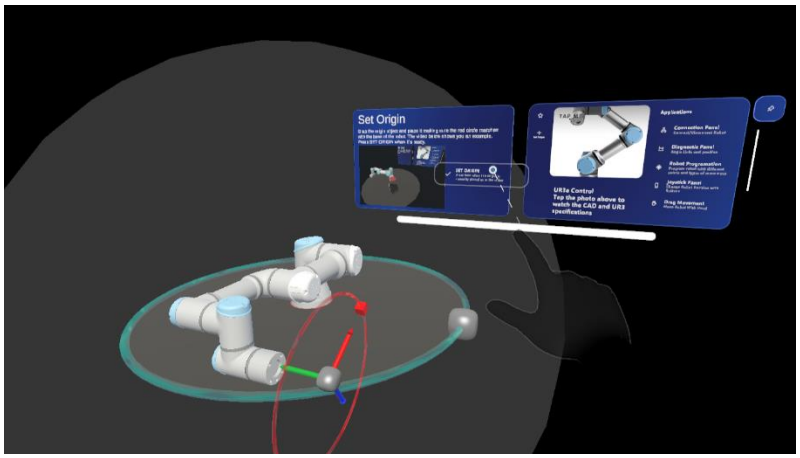

Figura 6. Example where the anchor object is perfectly positioned at the base of the robot

To guide the user, there is an explanatory text along with a video.

This process ensures that the coordinate axis of the application nearly matches the real coordinate axis of the robot. Since this is a manual anchor made with mixed reality, there will be a maximum error of 5 cm if done correctly.

The procedure is to place the anchor object as indicated and then press the "Set Origin" button. If rotation is needed, there are two buttons for this; these buttons will rotate the robot 5 degrees in the

desired direction. If any adjustments to the anchor are necessary, the "Set Origin" button on the main panel will always be available.

2.5. Home

A button labeled "Reset Scene" has been implemented on the main panel, which restarts the application in case of any errors or bugs that were not recorded.

3. Interface Panels and Their Uses

Before describing the functionality panels, an interface has been created to manage the entire application, allowing navigation through all existing features. This main panel is shown below:

Figura 7. Main Panel

For the development of the application, four panels have been created, with a fifth panel planned for future development. The panels are described below, including existing issues with the Mixed Reality Toolkit that I have encountered and possible future directions for each panel.

3.1. Connection Panel

This panel allows the input of the robot's IP address. A keyboard has been implemented for this purpose, which is currently under development by Microsoft. This means the keyboard works but may not function perfectly. The panel is shown in Figure 3.

Clicking on the text box will open the keyboard. The connect and disconnect buttons start or end communication with the UR robot via sockets. Another icon indicates the connection status: green if connected and red if not.

3.2. Diagnostic Panel

This panel displays the robot's joint and Cartesian positions in real time. It includes a top panel for easy movement. The panel is shown in Figure 8.

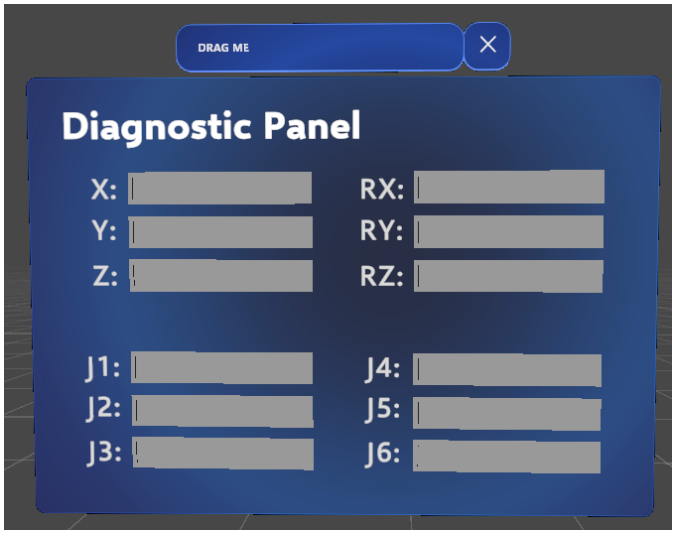

Figura 8. Diagnostic Panel

3.3. Joystick Panel

This panel allows the robot to be moved using URScript's speedl commands. It includes buttons to adjust both the speed and acceleration of the robot. Some coordinate changes require significant joint movement (necessitating low speeds), while others need less joint movement (requiring higher speeds). The movement buttons will move the robot along the chosen axis, while the rotation buttons will rotate the robot around the selected axis. The panel is shown in Figure 9.

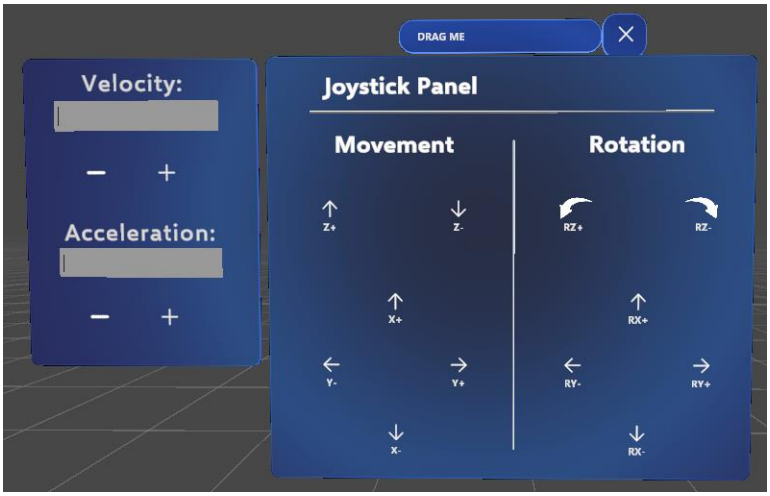

Figura 9. Joystick Panel

3.4. Robot Programming Panel

This panel allows programming of the robot by sending multiple points that can be moved by hand, either close or from a distance. There is a text panel that alerts users to possible errors, such as points being out of range. In such cases, the point cannot be sent to the robot for safety reasons. Once the desired points are placed, they will execute sequentially when the "send" button is pressed.

A future direction for this panel would be to develop a way to send rotations, as currently only initial rotations are maintained. As this tool is intended for teaching robotics students, it is recommended to position the robot initially so that the type of movement is clearly traced while maintaining the rotation. Sending rotations has been deemed unnecessary for the educational purpose of teaching different types of movements. The panel is shown in Figure 10.

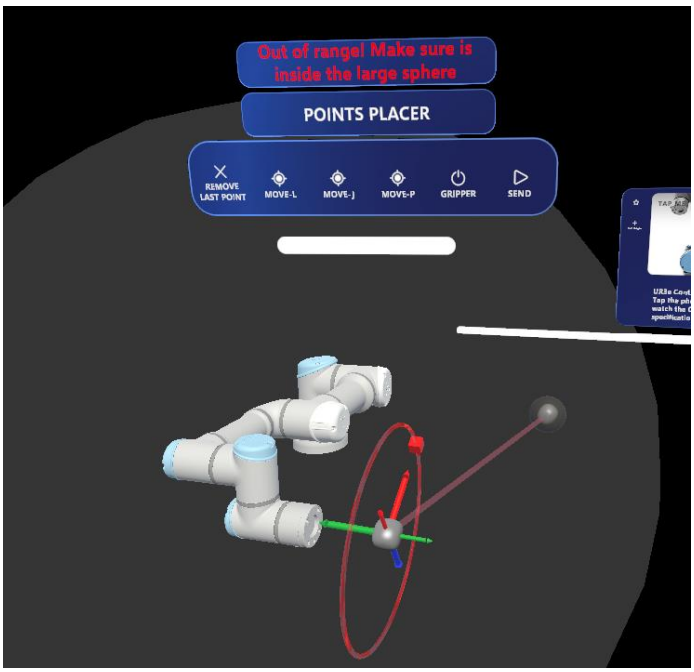

Figura 10. Robot Programming Panel

It is worth noting that when a point is created, a line simulates the robot's movement. Three types of movement can be chosen: MoveL, MoveJ, and MoveP. Additionally, there is a gripper action that moves the gripper or tool installed on the robot.

It is important to mention that apart from sending points, the robot creates a green trail that marks its path at all times. This allows the CAD of the robot to be anchored far from the actual robot, enabling programming on the Teach Pendant while seeing the virtual object's path for total safety and enhanced learning by observing movement type differences. This is shown in the results video in section 5.

3.5. Hand Movement Panel

This panel is proposed for future development, as it was beyond the scope of this project. It would allow the user to grab the end of the robot or any joint and move it by dragging it with the hand, always maintaining a speed that ensures user safety.

4. Unity Editor

To modify the existing project, follow these steps:

1. Download the project from the GitHub repository: UR_HoloLens2_Unity
2. After downloading, it is recommended to decompress it in C:/ to avoid file path length issues.
3. Open "Unity Hub," click the "Add" button, and select the folder "UR_Hololens2" located at \UR_Hololens2_Unity\UnityProjects
4. Once the project is opened, navigate to the "Scenes" folder in the "Project" window in Unity and open the "PruebaUR" scene.
5. You might be prompted to import the TMP package (a text package). After importing, restart Unity and reopen the project to apply the changes.
6. Once reopened, the project is ready for editing. If there is an error when entering play mode, ensure that "Holographic Remoting" is active. This can be checked by opening the window in "Mixed Reality/Remoting/Holographic Remoting for Play Mode."

5. Results

The results of this project can be viewed graphically in the following video:

6. Contact

If you have any questions or issues, please contact me via email. I will be happy to assist:

Álvaro Javier Orcajo Domingo

alvarojavierorcajodomingo@gmail.com