

Teleoperation of industrial robot manipulators based on augmented reality

J. Ernesto Solanes* · Adolfo Muñoz ·
Luis Gracia · Ana Martí · Vicent
Girbés-Juan · Josep Tornero

Received: May 19, 2020 / Accepted: date

Abstract This research develops a novel teleoperation for robot manipulators based on augmented reality. The proposed interface is equipped with full capabilities in order to replace the classical teach pendant of the robot for carrying out teleoperation tasks. The proposed interface is based on an augmented reality headset for projecting computer-generated graphics onto the real environment and a gamepad to interact with the computer-generated graphics and provide robot commands. In order to demonstrate the benefits of the proposed method, several usability tests were conducted using a 6R industrial robot manipulator in order to compare the proposed interface and the conventional teach pendant interface for teleoperation tasks. In particular, the results of these usability tests show that the proposed approach is more intuitive, ergonomic and easy-to-use. Furthermore, the comparison results also show that the proposed method clearly improves the velocity of the teleoperation task, regardless of the user's previous experience in robotics and augmented reality technology.

Keywords Augmented reality interface, Industry 4.0, Industrial robot teleoperation, Mixed reality interface, Guiding industrial robots

1 Introduction

1.1 Motivation

Industrial robot manipulators have been utilized in industry for a long time with great efficiency. These robots have allowed to automate most of current

(*Corresponding author. E-mail: esolanes@idf.upv.es)

J. Ernesto Solanes, Adolfo Muñoz, Luis Gracia, Ana Martí and Josep Tornero are with Instituto IDF, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain.

Vicent Girbés-Juan is with Departament d'Enginyeria Electrònica, Universitat de València, Avda de la Universitat s/n, 46100 Burjassot, Spain.

production lines to improve productivity and product quality. For this reason, tasks such as welding, painting or pick & place of products are carried out by industrial robots.

The development of a robot application usually requires performing robot teleoperation at some point. For instance, robots are typically programmed using teaching by showing techniques [5], which requires using robot teleoperation to locate the poses within the workspace that the robot must “learn”. Moreover, other robot applications are essentially a teleoperation task. For example, this is the case when a surgeon commands a robot to perform a surgery.

However, the complexity of teleoperating industrial robots is still nowadays a handicap for potential non-experienced users. The need of previous knowledge in robotics, together with the complexity of most interfaces provided by robot manufacturers, generates dependency of the non-experienced users with the technical service (expert workers) of the robot manufacturer.

Aware of this issue, most robot manufacturers have developed interfaces more user-friendly. Nevertheless, these interfaces are still complex and non-intuitive for non-experienced users.

Hence, this work aims to develop a more intuitive and easier-to-use interface for industrial robot teleoperation in order to improve the performance of both experienced and non-experienced users, which is valuable for most of the current industries.

1.2 Related work

Teleoperation, which consists of remotely controlling a machine using an appropriate interface, has been widely studied in robotics. The main reason is the benefit of tightly coupling the user input with the robot actions [39]. Thus, teleoperation has been applied to a wide variety of robots and applications, from surgical robots [31,29] or robot manipulators [16,3] to aerial robots [23,38] or underwater robots [13,43].

Focusing on industrial robot manipulators, most of the current manufacturers provide a remote interface to teleoperate their robots. For instance, the interface of Fanuc is based on a small touch screen and a complex panel full of buttons to allow the user to teleoperate the robot [15]. In this case, only experienced users can manage this kind of interfaces due to their complexity. Other manufacturers have developed more user-friendly interfaces. For example, Kuka [26] and ABB [1] provide a similar interface where most of the interaction is carried out touching its screen panel, which is easier for non-experienced users. However, previous knowledge about robotics, reference systems interaction and computer science is required to use these interfaces. It is also worth noting that usually each robot manufacturer has its own programming language and software, which encourages the users to acquire robots from the same manufacturer, even if they are not the best option for a given task.

Augmented reality (AR) is used in many applications [32,34] to facilitate natural interaction between human and computer systems, e.g., industrial robot manipulators. This technology is based on the projection of computer-generated graphics, i.e., virtual objects, onto the physical environment [14]. Usually, the interaction with these virtual objects is natural, using motion tracking of users [6,41] or customized devices to suit concrete applications [40, 4] instead of keyboards or button panels.

Many works can be found in the literature using AR technology to facilitate the teleoperation of industrial robots. For instance, [27] proposed the application of AR to control robot manipulators in order to make the interaction with the robot easier. In [44], a teleoperation system for maintenance robots based on AR was proposed. In [28], a robot teleoperation system based on mixed reality combined with leap motion to move the robot end-effector was developed. In [36], a mixed-reality head-mounted display visualization of the intended robot motion over the wearer's real-world view of the robot and its environment was proposed. Thus, the user was able to modify the intended goal pose of the end effector using hand gestures. A similar approach was developed in [19], where a mixed reality head-mounted display enabled the user to create and edit robot motions using waypoints.

As mentioned above, there are many works dealing with the teleoperation of industrial robots using AR technology. However, all these approaches are far from being an alternative to conventional teach pendant (TP) interfaces. In this sense, some of these researches presented usability tests to show the benefits of the proposed AR-based solutions for robot teleoperation but lacked a fair comparison analysis with conventional TP interfaces. To the best of the authors' knowledge, this is the first work proposing an AR-based interface for robot teleoperation that completely replaces conventional TP interfaces, providing a comparison usability analysis with experienced and non-experienced users.

Furthermore, gamepads have been used to interact with AR devices due to their ergonomic (they are specially designed to be used for hours), intuitive and easy-to-use features. For instance, [8] compared a hand gesture interface based on the leap motion sensor, a gamepad interface and a gaze-based interface for locomotion in virtual reality worlds. The results of this study showed that the hand gesture performed better than the gaze-based interface but worse than the gamepad interface. Moreover, [25] compared several seated leaning locomotion techniques to the joystick interface. This study reported that participants preferred the leaning techniques as they are fun, engaging and more realistic, but the joystick interface was still easier to use and control. Furthermore, [45] compared head gesture, hand gesture and gamepad interfaces in virtual environments. The results of this study indicated that participants preferred gamepad devices since they were more familiar and did not take extra efforts for them to learn how to use them.

However, despite all the mentioned advantages, to the best of the authors' knowledge, gamepads have not yet been used to interact with AR devices for robot teleoperation, as proposed in this work.

1.3 Features of the proposed method

This work proposes a novel AR interface to teleoperate industrial robot manipulators which can be used as a substitute of conventional TP interfaces. The proposed interface is based on an AR headset for projecting virtual objects onto the real environment to provide feedback about the task status, and a gamepad to allow the user to interact with the virtual objects and teleoperate the robot in a more ergonomic and intuitive manner. Thus, the user receives visual and sound feedback from the robot sensors (i.e., joint sensors) and external sensors (e.g., camera or Force/Torque sensors) in real-time, which is shown in form of 3D projections in the real environment using the AR headset, yielding an improvement of the user's ergonomics and productivity (the user can be concentrated in the task, not having to check different devices).

In addition, and similar to conventional industrial robot interfaces, the user can command the robot using different modes, i.e., the movement of the robot can be referred to the tool reference system or to the world reference system, as well as to move each axis individually. Furthermore, the user can visualize the tool trajectories carried out during the teleoperation task and save them in order to, for example, use them to program the robot using the well-known teaching by demonstration method.

The proposed approach is validated using a 6 degrees of freedom industrial robot manipulator together with an AR headset and a gamepad. Furthermore, to demonstrate the benefits of the proposed method, several usability tests were conducted to compare the proposed interface and the conventional TP interface for teleoperation tasks. In general, the proposed approach obtained very positive results in these usability tests.

The main contributions and advantages of the proposed method for robot teleoperation are the detailed below.

- **Main contributions** of the proposed approach:
 - The proposed AR-based interface is equipped with *full capabilities* to be used as an alternative to TP interfaces for teleoperation tasks. As mentioned above, there are many works dealing with the teleoperation of industrial robots using AR technology [19, 27, 28, 36, 44], but all these approaches are far from being an alternative to conventional TP interfaces. It is also worth noting that the proposed interface is *generic* and applies to any commercial industrial robot.
 - The proposed method includes not only the AR-based interface (AR headset and gamepad) but also all the *hardware implementation* and software development that is required to make possible the robot teleoperation, see Section 4.3.
 - This work conducts several usability tests not only for the proposed (AR) interface but also for the conventional (TP) interface. Thus, the *comparison results* obtained in this work are more complete and insightful than those presented in previous works [19, 23, 25, 28, 36, 39],

where only the AR-based solutions developed for robot teleoperation were tested.

- Main **benefits** of the proposed approach:
 - The interaction using the AR and gamepad devices is *intuitive, ergonomic* and *easy-to-use* (see the results of the comparison usability tests in Section 5.7), reducing the user’s learning time and technology adaptation.
 - The *velocity* of the teleoperation task *is improved* compared to that obtained with the conventional method regardless of the user’s previous experience in robotics and augmented reality technology (see the results of the comparison usability tests in Section 5.7).
 - *Non-experienced users* can manage industrial robots, decreasing their dependency with the technical service of the robot manufacturer.
 - The *trajectories* generated during the robot teleoperation *can be subsequently used* for other purposes. For example, these trajectories can be used to program industrial robots (teaching by showing method) by means of a post-processing stage, which depends on the programming language of the considered robot.

1.4 Structure of the paper

The paper is organized as follows. Next section presents an overview of the proposed AR interface, which is fully described in Section 3. Moreover, the proposed robot external controller is detailed in Section 4. The benefits of the proposed approach are substantiated by experimental results in Section 5, while Section 6 discusses further details of the study. Finally, some conclusions are given in Section 7.

2 Overview of the proposed application

The application developed in this work, which is schematically shown in Fig. 1, consists of three parts: the *robotic cell* in which the robot operates; the *user interface* in which the AR interface is used; and the *external controller* in which the robot commands and securities are implemented.

In the *robotic cell*, the robot controller receives the joint command¹ \mathbf{q}_c from the external controller, which is obtained from the command given by the user with the gamepad. The robot can be equipped with different sensors. Without loss of generality, this work assumes the use of a camera sensor and a Force/Torque (F/T) sensor, both attached to the robot end-effector. The camera is used by the AR application to show the user the local area pointed by the tool. The F/T sensor has two objectives: on the one hand, it provides the user feedback on the pressure exerted by the tool on the surface; on the

¹ Vector $\mathbf{q} = [q_1 \cdots q_n]^T$ is the robot configuration, where n is the number of joints of the robot.

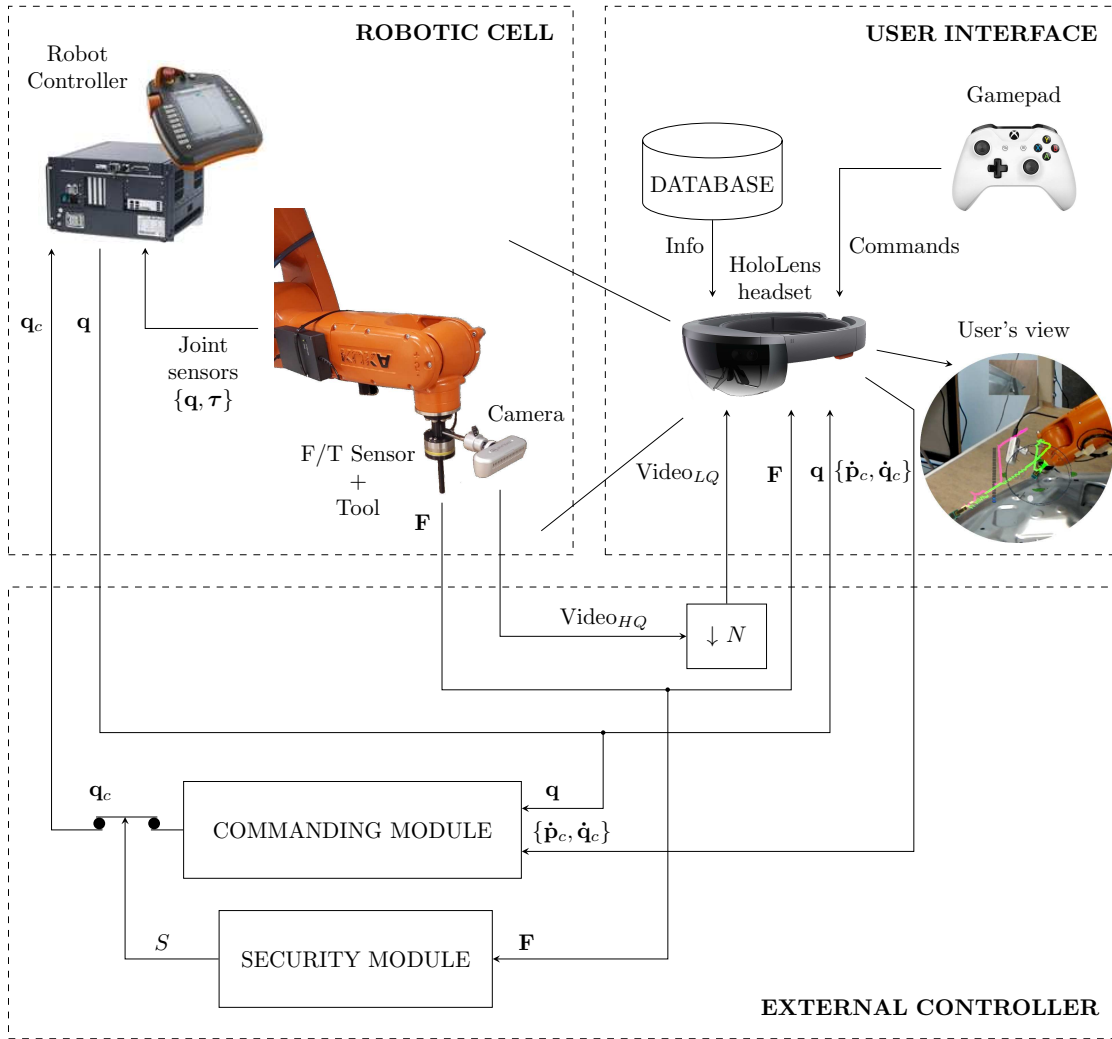


Fig. 1. Industrial robot teleoperation using AR with data from robot, camera, F/T sensor, gamepad and database.

other hand, it is also used as safety indicator in order to stop the robot motion when the measured values are abnormally high.

In the *user interface*, the user is able to visualize several holograms placed at the end-effector of the robot through the AR headset, e.g., the position and orientation of the tool, the direction of the commanded movements, the pressure status, the camera view, the robot velocity (in percentage), the targets on the object, the tool trajectory, among others. The location of these holograms is updated according to the measurements given by the sensors: the robot configuration q , which is obtained by reading the joint values provided

by the joint sensors to the robot controller; the pressure \mathbf{F} obtained from the F/T sensor; and the video streaming provided by the camera. The location of other holograms, such as those showing the targets and floating panels, are read from the AR headset database or referenced to the AR headset reference system. In addition, the user is able to interact with the *user interface* and command the robot through the gamepad device, which communicates via bluetooth with the AR headset. For further details of the user interface functionalities, see Section 3.

In the *external controller*, the commanding module receives the velocity command (either in the Cartesian space² $\dot{\mathbf{p}}_c$ or joint space $\dot{\mathbf{q}}_c$) from the AR application, which corresponds to the gamepad command given by the user. The controller also receives the current value \mathbf{q} of the robot joints. Thus, according to these values, the module computes the joint command \mathbf{q}_c to be sent to the robot controller. For safety, a security module receives the measurement \mathbf{F} of the F/T sensor and interrupts the robot motion if high pressure values are detected. For further details of the external controller modules, see Section 4.

3 Augmented reality-based interface

The proposed AR interface is implemented using Unity 3D Engine. It is composed of three main parts: starting panel, calibration section and teleoperation section. Table 1 shows the main functionalities of each part of the developed interface. In particular, the functionalities of the starting panel allow to check the status of the communications with the gamepad and the external controller and to configure the IP address of the AR device and the gamepad number. Moreover, the functionalities of the calibration section allow to make visible the information of the gamepad layout and to configure the reference system of the virtual world. Finally, the functionalities of the teleoperation section allow to make visible the information panel and to configure the teleoperation options depending on whether the “deadman” button is active or not.

Fig. 2, Fig. 3a and Fig. 3b show the flow diagram of the mentioned three main parts, which are further described below.

The *starting panel* (see Fig. 2) allows the user to change the communication settings, i.e., the IP address according to the network where the external controller can be found, and the gamepad to be used (if multiple gamepads are simultaneously connected). The interface indicates the status of the gamepad and external controller communications, showing a warning message if one or both communications are lost at any time. If the communication with the gamepad is lost while commanding the robot, the AR interface sets to zero the velocity command in order to stop the robot motion. If the communication with the external controller is lost, the safety module activates the security mode and, hence, the robot controller does not receive any command from the commanding module.

² Vector $\mathbf{p} = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$ is the robot pose, where the orientation is given by roll α , pitch β and yaw γ angles.

Table 1. Main functionalities of the developed AR interface.

STARTING PANEL	COMMUNICATIONS	GAMEPAD STATUS EXTERNAL CONTROLLER STATUS
	CONFIGURATION	IP ADDRESS GAMEPAD NUMBER
CALIBRATION SECTION	INFORMATION PANEL	GAMEPAD LAYOUT INFORMATION
	VIRTUAL WORLD REFERENCE SYSTEM	MANUALLY AUTOMATICALLY (LANDMARKS)
TELEOPERATION SECTION	INFORMATION PANEL	GAMEPAD LAYOUT INFORMATION
	DEADMAN BUTTON ACTIVATED	TOOL AXIS POSITION DIRECTION FEED- BACK
		TOOL AXIS ORIENTATION DIRECTION FEEDBACK
		SET MOVEMENT VELOCITY
		SET CARTESIAN MODE (DEFAULT)
	DEADMAN BUTTON DEACTIVATED	SET CAMERA MODE
		TOOL PRESSURE FEEDBACK
		CAMERA VIEW FEEDBACK
SET CARTESIAN MODE (DEFAULT)		
	SET CAMERA MODE	
	SET AXIS MODE	
	SET MOVEMENT VELOCITY	
	ACTIVATE CALIBRATION SECTION	
	SAVE TOOL TRAJECTORIES AND ROBOT COMMANDS	
	HIDE/SHOW TOOL TRAJECTORIES FEEDBACK	
	HIDE/SHOW MARKERS FEEDBACK	

The *calibration section* (see Fig. 3a and Fig. 3b) allows the user to set the virtual world reference system in order to match the real world reference system. This calibration can be carried out automatically using landmarks [20], or manually, using a 3D model of the cell to set the virtual world reference system according to the real robotic cell. The automatic calibration is optional and depends on the possibility of introducing landmarks in the real environment. The manual calibration is required and is easy to be used by users of any education level. Thus, the movement of the virtual world reference system, i.e., the movement of the virtual robotic cell, has been developed to be relative to the AR headset, being more intuitive than using a fixed virtual reference system. In particular, the user can move the virtual cell in the linear X-, Y- and Z-axes relative to its point of view, as well as to reorientate the virtual cell in the angular Z-axis. In addition, the user sets the movement velocity in order to perform quick or fine movements depending on the needs. A floating panel, which can be made visible or not by pressing the menu button of the gamepad, contains the information relative to the gamepad layout.

The *teleoperation section* (see Fig. 3a and Fig. 3b) is the default mode when the communication between the external controller and the user interface is established. When this section starts for the first time, a floating panel appears showing the gamepad layout information as well as the option to go to the calibration section. For security reasons, a “deadman” button is considered

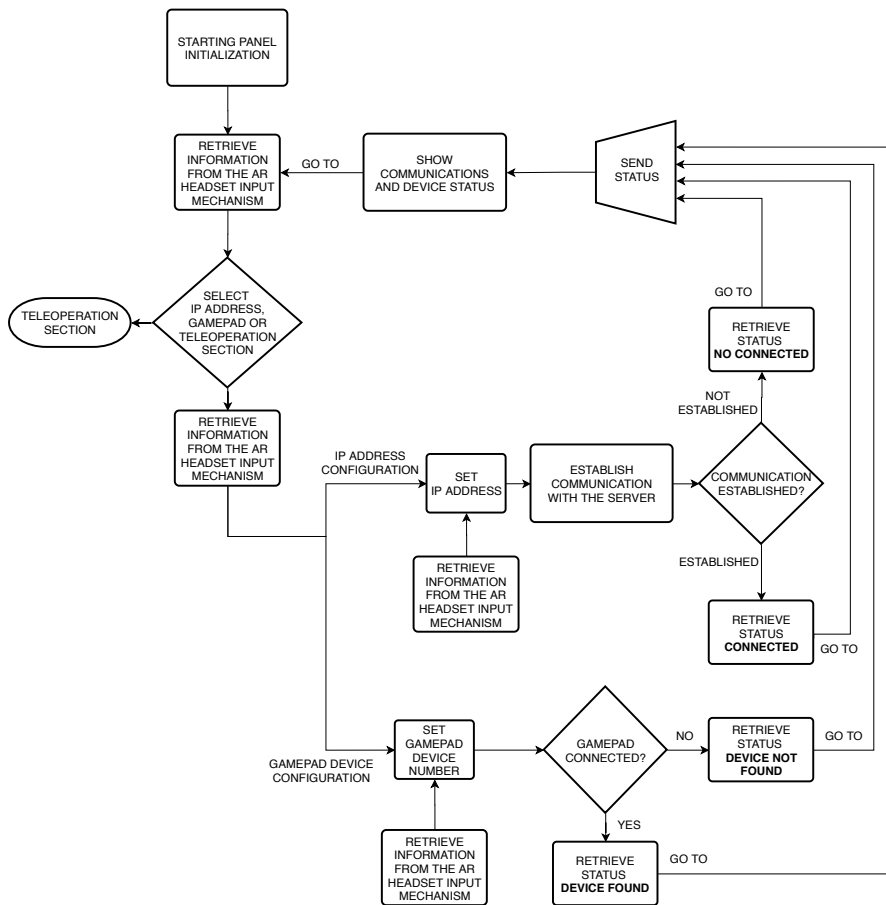


Fig. 2. Flow diagram of the starting panel.

in the gamepad to allow the user to provide command actions to the robot. The user can remove the floating panel by pressing the deadman button or the menu button of the gamepad. In addition, the user can make visible again the floating panel at any time by pressing the menu button as long as the deadman button is deactivated. If the deadman button is deactivated, the user can go to the calibration section to set the the virtual world reference system according to the real robotic cell. In the same way, the user can return to the teleoperation section from the calibration section only when the deadman button is deactivated.

Three modes of teleoperation have been implemented, namely, Cartesian mode, camera mode and axis mode. The Cartesian mode allows the user to move the robot tool with respect to the AR headset reference system, which is more intuitive than using a fixed reference system such as the virtual world reference system. The camera mode allows the user to move the robot tool with

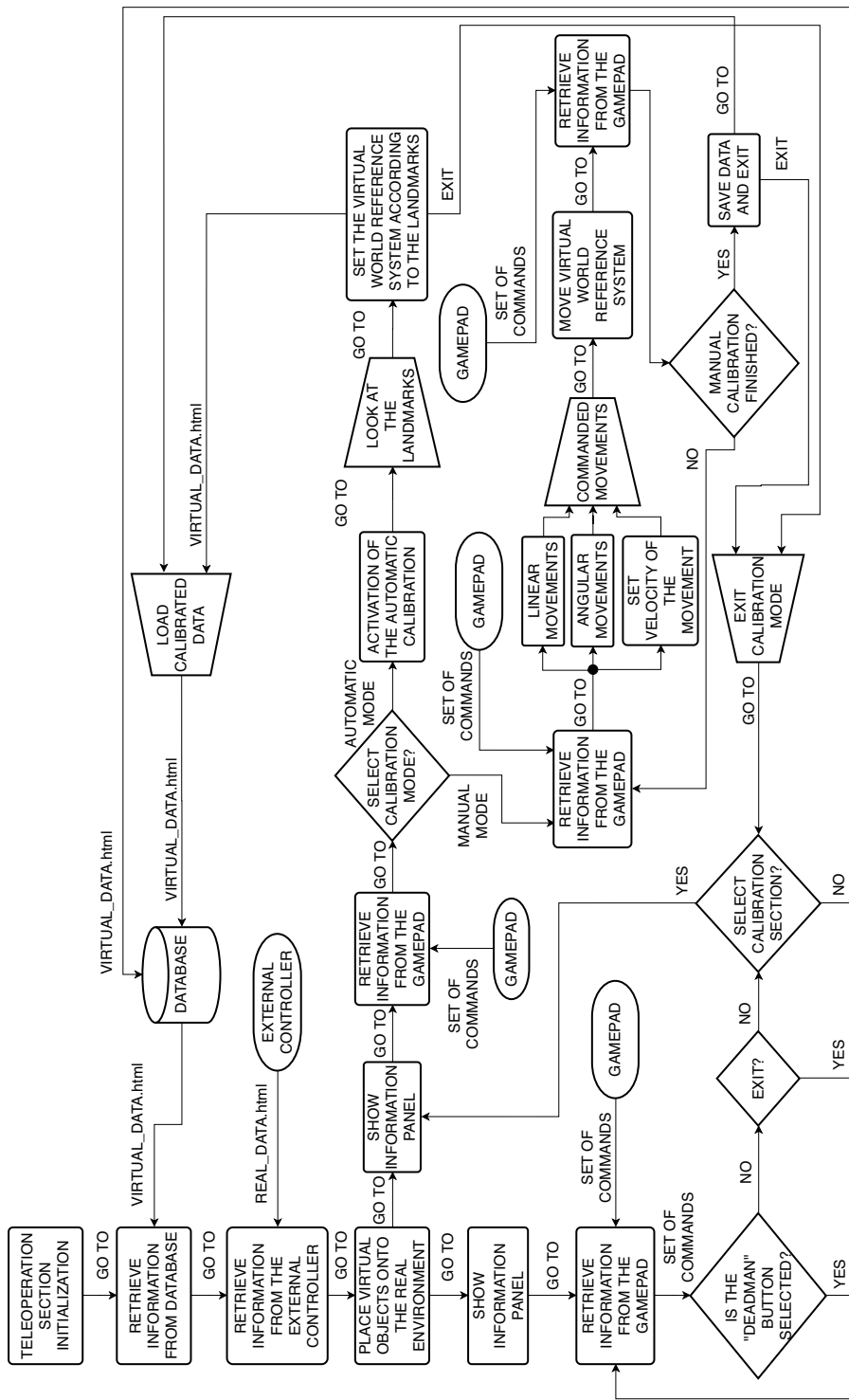


Fig. 3a. Flow diagram of the calibration and teleoperation sections: Top part of the diagram (see the bottom part in Fig. 3b).

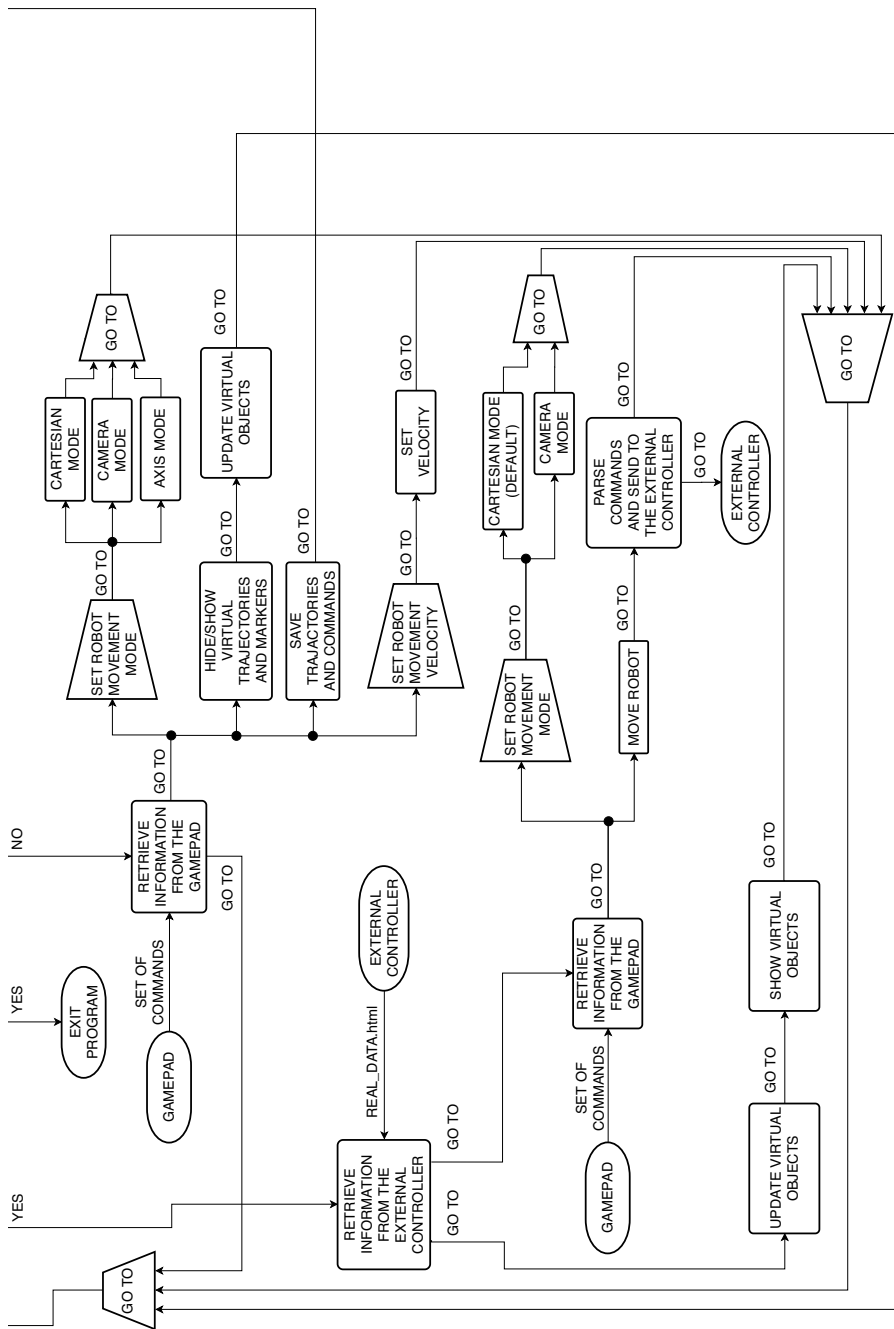


Fig. 3b. Flow diagram of the calibration and teleoperation sections: Bottom part of the diagram (see the top part in Fig. 3a).

respect to the camera reference system placed at the robot end-effector. In both Cartesian and camera modes, the command to the robot is the Cartesian velocity $\dot{\mathbf{p}}_c$, which corresponds to the velocity provided by the user with the gamepad projected onto the virtual world reference system. The axis mode allows the user to move separately each robot joint.

The user can set the movement velocity of the robot in all three modes of teleoperation in order to perform rough or fine robot movements. The movement velocity, which is defined as a percentage of the maximum velocity of the robot (as in most of industrial robot interfaces), can be modified either with the deadman button activated or deactivated.

The Cartesian mode is initially active by default. The user can change between the Cartesian and axis mode as long as the deadman button is deactivated, whilst the Camera mode can be selected with the deadman button activated or deactivated.

If the deadman button is deactivated, the user can make visible or hide hologram information such as object markers or tool trajectories. This option is useful when fine movements are being carried out and the user needs to see the real world with no interference of virtual objects. In addition, the user can save to the database the trajectories and actions carried out during the teleoperation for further analysis or to be used to program the robot (teaching by showing method).

If the deadman button is activated, the user can command the robot in real time and information about the actions carried out are shown in real time in the form of virtual holograms, see Fig 4. In particular, for both Cartesian and camera modes, the direction of the tool movement is represented by two groups of yellow arrows for linear movements, see Fig. 4(a), Fig. 4(b) and Fig. 4(d), whereas a yellow arrowed circle is used for angular movements, see Fig. 4(c) and Fig. 4(e). In addition, the pressure is indicated by a color bar, being dark blue for zero pressure and red for the maximum pressure. Moreover, a beat sound, which increases its frequency as the pressure approaches the maximum value, is sent to the user as information about the pressure status. Other information, such as the value of the movement velocity or the current mode of teleoperation, is displayed for 5 seconds when the user selects the required option. Thus, the user receives the system feedback without being disturbed.

4 External controller

4.1 Commanding module

This module computes the joint command \mathbf{q}_c to be sent in real time to the robot controller depending on the command provided by the user with the gamepad as follows:

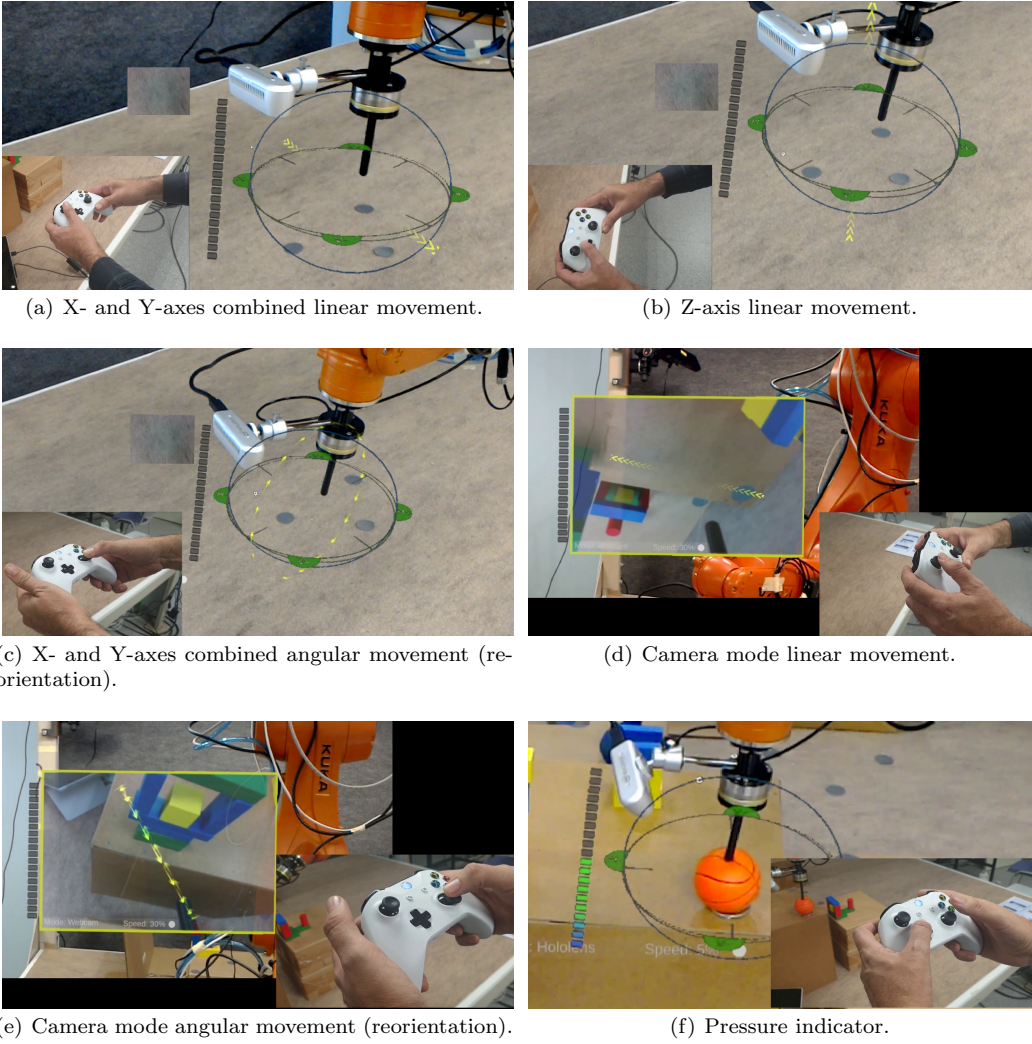


Fig. 4. Teleoperation section of the proposed AR interface: linear movements are indicated by two groups of yellow arrows, whereas angular movements are indicated by a yellow arrowed circle.

1. If the user provides a Cartesian velocity command $\dot{\mathbf{p}}_c$:

$$\mathbf{q}_c = \int_0^T \mathbf{J}^{-1} \dot{\mathbf{p}}_c dt, \quad (1)$$

where \mathbf{J} is the robot Jacobian [37] (the Jacobian pseudoinverse should be used instead of the Jacobian inverse if it is not a square matrix).

2. If the user provides a joint velocity command $\dot{\mathbf{q}}_c$:

$$\mathbf{q}_c = \int_0^T \dot{\mathbf{q}}_c dt. \quad (2)$$

4.2 Security module

For safety reasons, this module interrupts the communication between the *commanding module* and the *robot controller* (see signal S in Fig. 1) if any of the following conditions occurs:

- Interruption of the communication between the external controller and the AR headset.
- The pressure value measured by the F/T sensor exceeds the maximum allowed pressure value.
- Interruption of the communication between the F/T sensor and the external controller.

Moreover, when the communication between the commanding module and the robot controller is interrupted for whatever reason, which is detected by a watchdog timer in the robot controller, the robot motion stops and, in the same manner of other industrial machines, the user has to manually reset the robot system to enable it again.

4.3 Hardware implementation for robot teleoperation

Fig. 5 shows the hardware used in this work to teleoperate the robot, which is composed of: an external controller, a router, an AR headset, a gamepad, a camera, a F/T Sensor and the robot controller provided by the robot manufacturer. The wired and wireless communications among these devices is also shown in the figure with solid and dashed lines, respectively. In particular, all communications are wired except those between the AR headset and the router and gamepad, which are wireless. Note that the router is used as a central hub for the communication between some devices.

The periods of the communications between the devices are also indicated in Fig. 5. In particular, the period is: 10ms for the communication between the external controller and the robot controller; 50ms for the communication between the external controller and the AR headset; 10ms for the communication between the external controller and the F/T sensor; 33ms for the communication between the camera and the external controller; and 50ms for the communication between the gamepad and the AR headset.

Therefore, taking into account these periods, the maximum delay between the movement generated by the user with the gamepad and its reflection in the real robot is equal to 110ms ($50 + 50 + 10$), which is almost imperceptible for the user when teleoperating the robot. Similarly, the maximum delay between

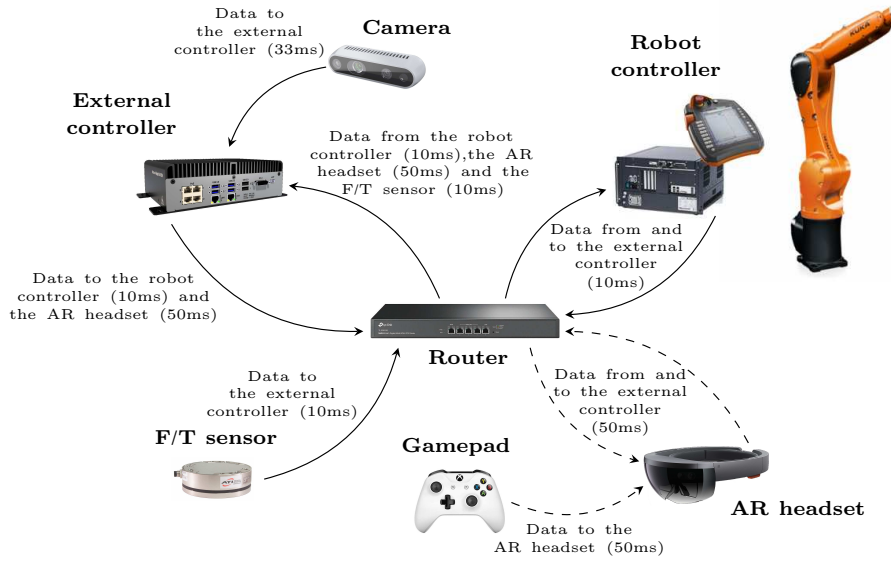


Fig. 5. Hardware implementation used for robot teleoperation. The solid and dashed lines are used to represent wired and wireless communications, respectively.

the actual movement of the robot and its reflection in the user interface is equal to 60ms ($10+50$), which is also imperceptible for the user. Finally, the maximum delay between the measurements of the F/T sensor and camera and their reflection in the user interface is equal to 60ms ($10 + 50$) and 83ms ($33 + 50$), respectively, which are also negligible.

The maximum operational speed and acceleration that permits the teleoperation system is basically given by the robot controller provided by the robot manufacturer. In this sense, the low-level joint control designed by the robot manufacturer is typically fully optimized taking into account a complete robot dynamic model, including frictions, dead zones, etc. Note that, if the user performs fast operations exceeding the maximum operational speed and/or acceleration mentioned above, the system will not get unstable since the robot controller is designed to guarantee always stability, but the accuracy of the robot position would be lost, i.e., the robot controller would not be fast enough to accomplish the user commands. However, these issues about maximum operational commands and stability are also present for the conventional TP interface and, in sum, both interfaces rely on the robot controller, which has already been optimized by the robot manufacturer as mentioned above.

In order to guarantee a safe teleoperation task, if the communication between two devices is interrupted for whatever reason, the security measures indicated in Section 4.2 become active.

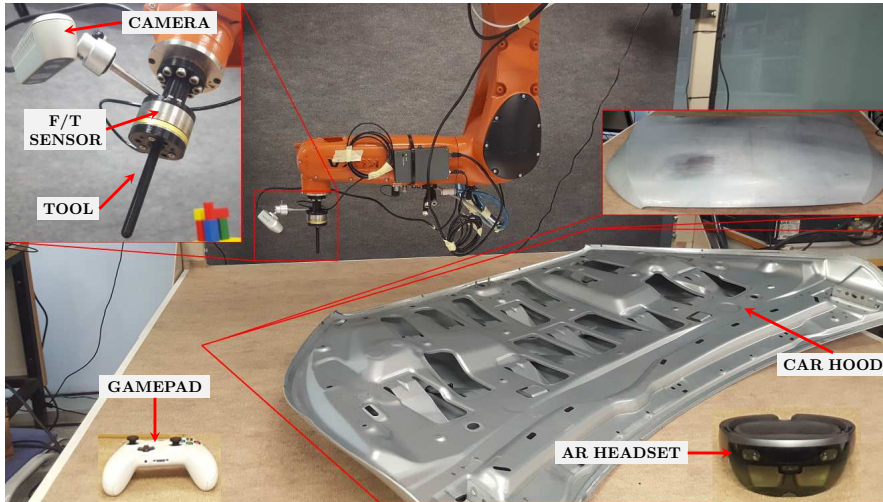


Fig. 6. Setup used for the real experimentation.

5 Experimental Results

5.1 Experimental Setup

Fig. 6 shows the setup considered for the real experimentation in order to evaluate the proposed approach, which consists of: a Kuka KR6 r900 sixx robot; a 45mm F/T steel sensor with DataBox V1.0 electronics (see [17] for further details) located between the last link of the robot and the tool; a self-developed stylus tool attached to the F/T sensor by means of a self-developed adapter; a depth camera Intel Realsense D435; a gamepad device, the Microsoft[®] *Xbox*; an AR headset, the Microsoft[®] *HoloLens glasses*; and a car hood as object to carry out a robot task (e.g., welding, polishing, painting, etc.).

The proposed external controller (see Fig. 1), which is composed of the commanding and security modules (see Section 4), was implemented in an external PC workstation (Intel-R Core-TM i7-5820K CPU 3.30GHz) with the following software: *Ubuntu 16.04* as Operative System; *ROS* (Robot Operating System) *Kinetic* distribution as middle-ware to interface with the Kuka robot; and the *netft_rdt_driver* ROS package provided by the F/T sensor manufacturer (ATI Industrial Automation). An Ethernet switch device was used to connect and communicate (UDP protocol) the robot, the F/T sensor and the workstation. The Intel Realsense camera was connected to a serial port of the PC workstation and transmitted via wifi to the AR headset using the UDP protocol.

The proposed AR interface (see Section 3), which is part of the user interface (see Fig. 1), was developed in an external PC workstation (AMD Ryzen 9 3900x CPU 3.8 GHz, RAM 64 GB and a Nvidia 2080 RTX graphic card) with the following software: *Micorsoft* Windows 10 as Operative System; Visual

Studio 2017 was used as integrated development environment (IDE); Blender 2.7 as 3D modeling suite [22]; and Unity LTS release 2017.4 [24] with back end IL2CPP as 3D game platform using tools such as MixedRealityToolkit-Unity [35]. The Xbox gamepad communicated via bluetooth with the AR headset.

The communication between the external controller and the user interface was made via Ethernet connection and using the ROS Bridge component with UDP protocol.

5.2 Demonstration experiment

A first experiment, namely *demonstration experiment*, was conducted to show experimentally the functionalities and performance of the proposed approach. Without loss of generality, a robot teleoperation task for conditioning (e.g., welding, polishing, etc.) the car hood surface was performed. In particular, two different points of the hood, located on the internal edge, were chosen and marked as targets. Thus, the user receives the feedback of the location of the targets from the AR interface in terms of holograms projected on the car hood. This work assumes that the locations of the car hood and targets have been previously provided by other systems.

The video of this demonstration experiment can be played at <https://media.upv.es/player/?id=6842c230-99b4-11ea-a399-57da8aaa21ee>, where it can be appreciated how the user employs the AR interface and the gamepad device to guide the robot to both target points in order to carry out the surface conditioning task.

Fig. 7 presents a few frames of the recording: Fig. 7(a), Fig. 7(b) and Fig. 7(c) (interval 12s–1m21s in the recording) show how the user guides the robot to the first target point and how the tool trajectory is being drawn (in magenta) in real time in the 3D workspace (these holograms can be hidden when needed for a better view of the real environment); Fig. 7(d) (time instant 1m32s) shows how the user activates the camera mode to better see the position of the tool with respect to the target on the car hood in order to perform a smooth approach of the robot tool to the car hood surface; Fig. 7(e) and Fig. 7(f) (interval 1m37s–2m22s) show the tool approaching to the car hood surface and the pressure warning indicating that the maximum allowed pressure value has been reached by the user (a beat sound is also provided to warn the user, see video); and Fig. 7(g) and Fig. 7(h) (interval 2m53s–3m21s) show how the user guides the robot to the second target point, generating a second trajectory (in green).

5.3 Setup for the comparison experiment

A second experiment, namely *comparison experiment*, was conducted in order to assess and compare the usability of the proposed AR interface with respect to conventional TP interfaces.

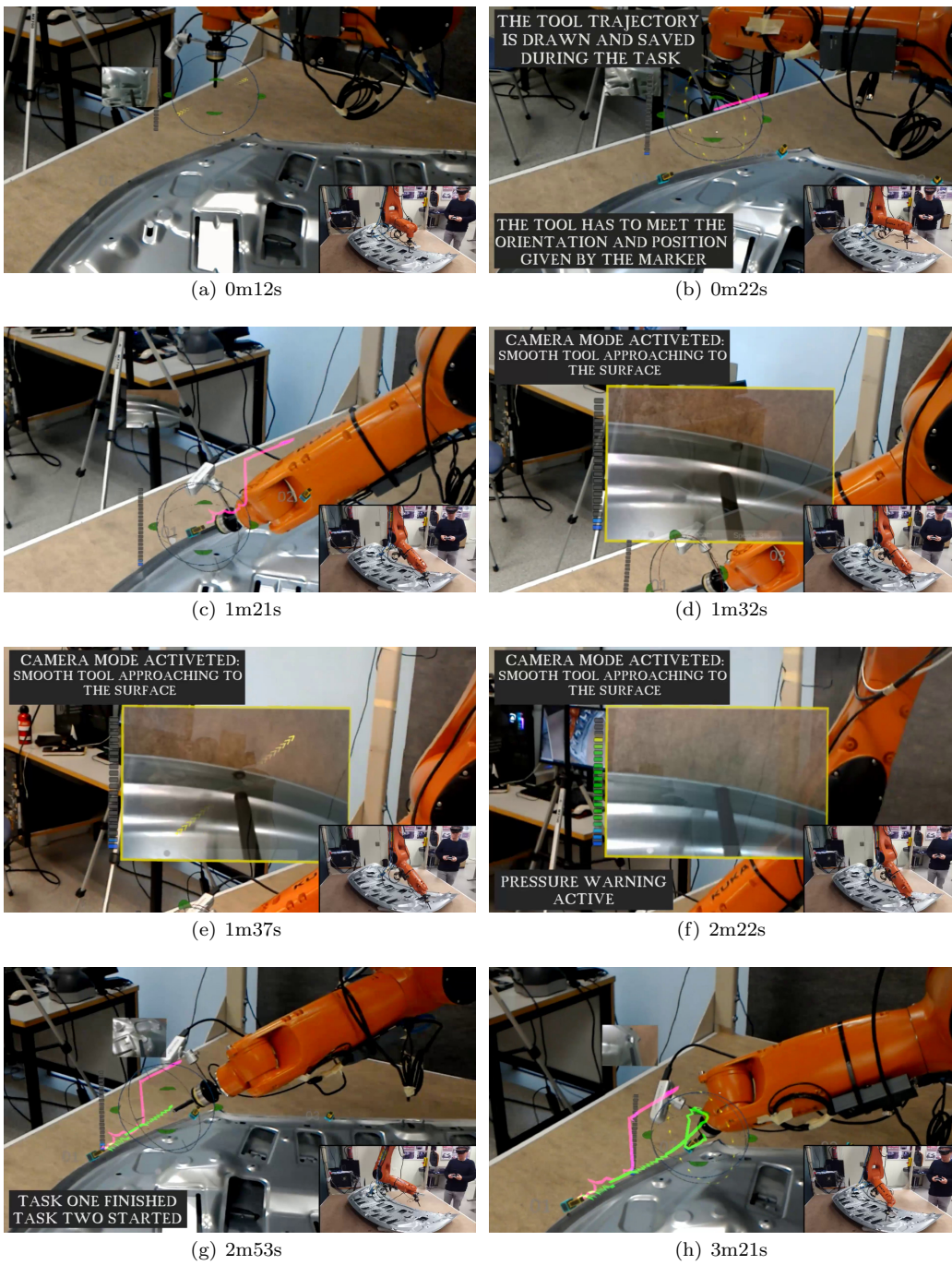


Fig. 7. Frames of the video of the demonstration experiment. The time instant is indicated for each frame.

Fig. 8 shows the setup used for the comparison experiment. In addition to the setup in Fig. 6, it was used: the robot TP; an external display showing the camera view and pressure indicator; five circuits with different complexity made up with wooden blocks; and an external barrier used to prevent the user from having a direct view of some circuits.

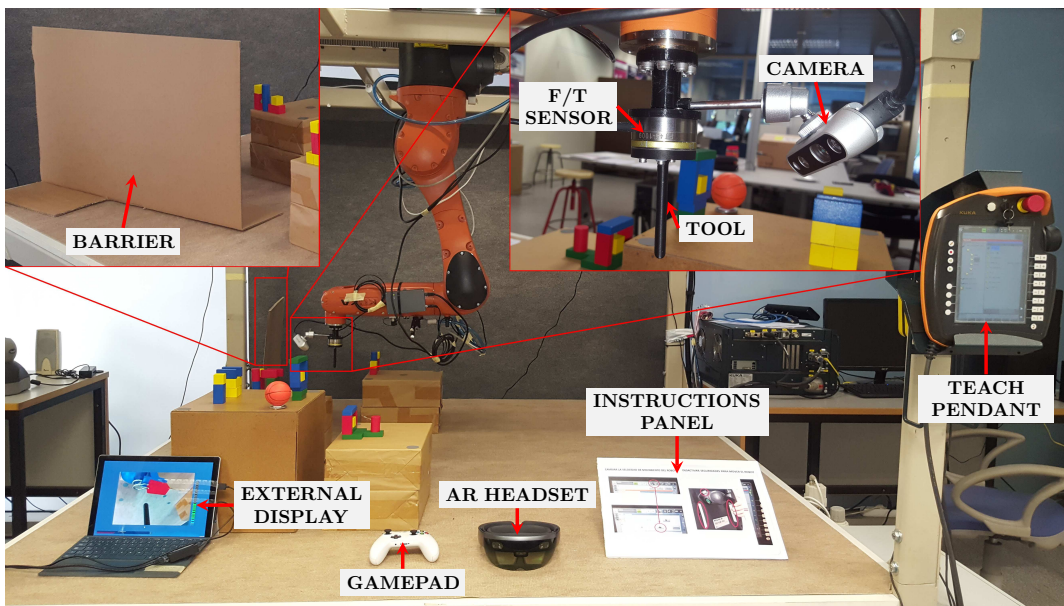
5.4 Usability tests

Similarly to other works [2,18,33], several usability methods typically used to validate software and hardware alongside with participant interviews were used. In particular, two standard questionnaires were used in the comparison experiment: the NASA Task Load index (NASA-TLX) [21] and the System Usability Scale (SUS) [7]. On the one hand, the NASA-TLX questionnaire is considered since it is commonly employed to evaluate digital and physical experiences in working environments. On the other hand, the SUS questionnaire is considered to assess the usability of the proposed approach since it is concise and widely used.

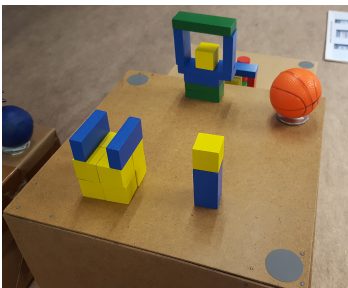
5.5 Conditions of the comparison experiment

The conditions of the comparison experiment were the following:

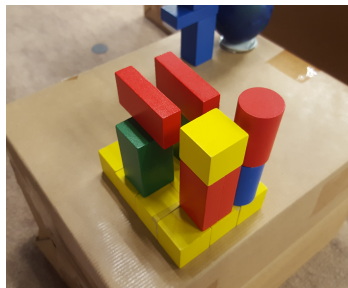
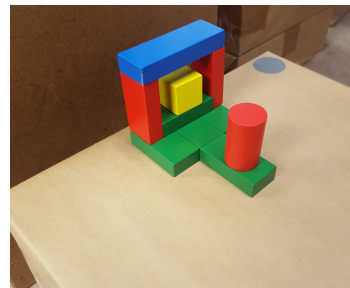
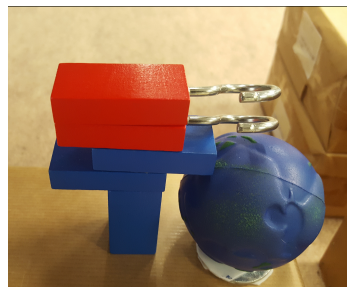
- The **goal** of the test was teleoperating the robot to accomplish the objectives of the four circuits depicted in Figs. 8(c) to 8(f), which were:
 - Circuit 1 (see Fig. 8(c) and [video](#) [9]): throw the yellow cube by passing the tool through the two red prismatic blocks.
 - Circuit 2 (see Fig. 8(d) and [video](#) [10]): throw the yellow cube with the tool without throwing the other blocks.
 - Circuit 3 (see Fig. 8(e) and [video](#) [11]): throw the two yellow cubes with the tool using the camera mode without having a direct view of the circuit.
 - Circuit 4 (see Fig. 8(f) and [video](#) [12]): press the ball with the tool passing through the forks and without exceeding the maximum allowed pressure value.
- **Conditions** of the test:
 - The starting point of the robot for each circuit was the same for all tests and participants.
 - All the participants performed the test twice, once with the conventional TP interface and once with the proposed AR interface.
 - Half of the participants started the test with one interface, whereas the other half started the test with the other interface.
- **Recording** of the test results:
 - The time that takes the participant to complete each circuit was individually recorded.



(a) Overall set-up.



(b) Training.

(c) Circuit 1 (see [video \[9\]](#)).(d) Circuit 2 (see [video \[10\]](#)).(e) Circuit 3 (see [video \[11\]](#)).(f) Circuit 4 (see [video \[12\]](#)).**Fig. 8.** Setup used for the comparison experiment.

- A circuit was recorded as “failure” when the participant throws away any block or part of the circuit that is not the target or breaks the tool.
- **Methodology** to conduct the test:
 - The participant fills in a first form with relevant data: gender, age, sex, education level, AR knowledge level and robotics knowledge level.
 - The participant watches a first training video showing the functionalities of the first interface that will be used to perform the test.
 - The participant practices what has seen in the first training video with the training circuit shown in Fig. 8(b).
 - The participant carries out the test with the first interface.
 - The participant fills in the NASA-TLX and SUS questionnaires related to the experience of performing the test with the first interface.
 - The participant watches a second training video showing the functionalities of the second interface that will be used to perform the test.
 - The participant practices what has seen in the second training video with the training circuit shown in Fig. 8(b).
 - The participant carries out the test with the second interface.
 - The participant fills in the NASA-TLX and SUS questionnaires related to the experience of performing the test with the second interface.
 - The participant makes comments about their global perception and answers some additional general questions.
- **Additional considerations** of the test:
 - It took around 45 minutes for each participant to complete the whole test applying the methodology mentioned above and two persons were required to assist him or her during the whole process.
 - The training lasted between 10 to 15 minutes.
 - Two modes of teleoperation were used by the participants to carry out the tests: Cartesian mode and camera mode, see Section 3.

5.6 Sample selection

Most of the works proposing a new interface for robot applications show its performance for just one user. However, there are a few researches that conduct some kind of usability test to prove its performance with several participants, as discussed below.

In [39] 18 participants were considered to test a robotic application based on AR and drones. A virtual reality application with drones is presented in [23] and 20 participants were considered to prove its performance. A robot arm application with AR is developed in [28], [36] and [19] and 34, 32 and 20 participants, respectively, were considered to test it. Finally, in [25] 16 participants were considered to test a virtual reality application.

Note that the mentioned works considered a similar number of participants, ranging from 16 to 34 participants and with an average value of 23.3. Therefore, twenty-three (23) participants are considered in this work for the

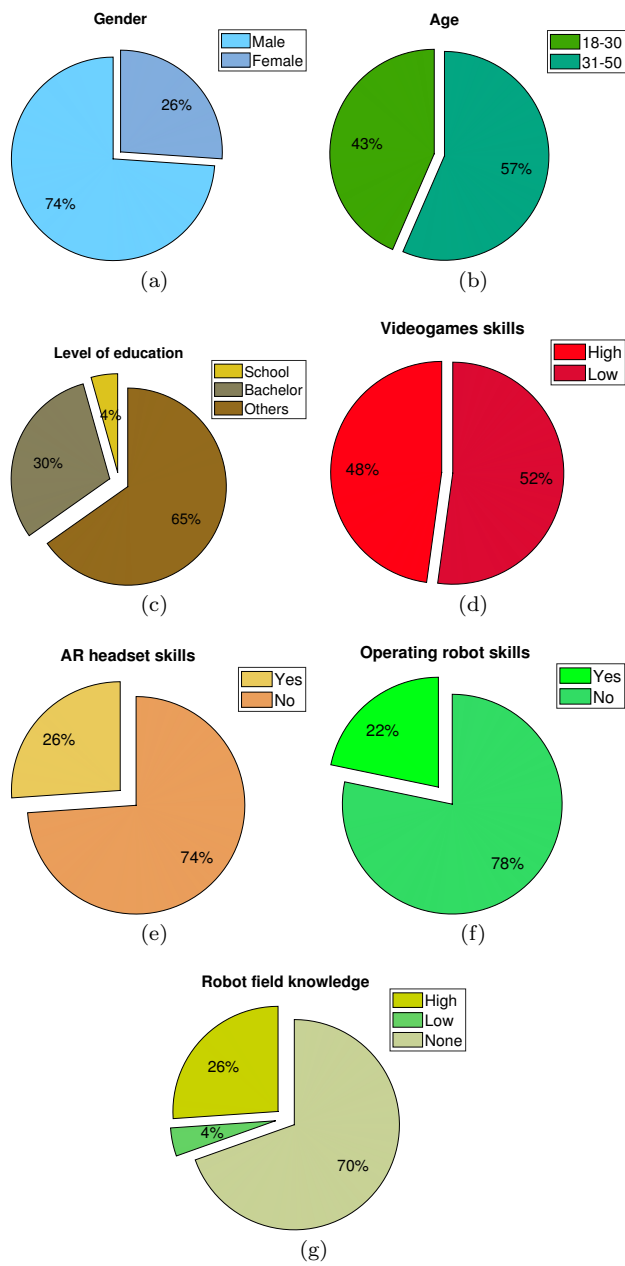


Fig. 9. Basic data of the participants involved in the comparison experiment.

comparison experiment. It is worth noting that, in contrast to the mentioned works, not only the proposed (AR) interface is tested in the experiment but also the conventional (TP) interface is tested. Thus, from a comparative point of view, the results detailed below are more complete than those presented in the mentioned works.

It is important to remark that considering a specific sample size gives rise to a certain *margin of error* [42]. In particular, for a sample size of 23 participants and considering a confidence level of 95% and unlimited population size, the margin of error is of only 20.02%, which means that there is a 95% chance that the real value is within $\pm 20.02\%$ of the value obtained with the selected sample, which is fairly reasonable.

Furthermore, in order to have a representative sample, the 23 participants selected for the comparison experiment had different backgrounds. The basic data related to these participants is indicated in Fig. 9. In particular, it can be seen that 52% of the participants had low experience playing video games and most of them had no previous experience with the Xbox gamepad. In addition, 74% of the participants had no previous experience using an AR headset. Furthermore, 70% of the participants had no previous knowledge of robotics and 78% of them had never operated an industrial robot.

5.7 Results of the comparison experiment

Fig. 10 and Table 2 comparatively show the amount of time that took the participants to carry out each of the four circuits with the TP and AR interfaces. In particular, Fig. 10 depicts the corresponding time histograms, whereas Table 2 indicates the mean value \bar{t}_i and standard typical deviation σ_i obtained for each case.

It can be seen in Table 2 that the mean value of the time needed to accomplish each circuit is significantly improved (around thirty percent) using the proposed AR interface: 35% for circuit 1; 53% for circuit 2; 21% for circuit 3; and 31% for circuit 4. Moreover, not only the mean value improved for all circuits but also the standard deviation improved for all of them except for circuit 3. This is due to three participants that although they improved their time to accomplish circuit 3 using the AR interface, see Fig. 10(e) and Fig. 10(f), they did not improve as much as the other 20 participants, giving rise to a value of standard deviation slightly larger than that obtained with the TP interface.

Table 3 shows the total number of failures committed by the participants for each circuit. Note that, in general, less failures were committed using the proposed AR interface than using the conventional TP interface.

In order to further analyze the individual performance of the participants in the comparison experiment, Fig. 11 shows the improvement achieved by *each participant* measured as $\frac{t_{TP} - t_{AR}}{t_{TP}}$, where t_{TP} and t_{AR} are the times obtained by the participant using the TP and AR interfaces, respectively. In the graphs, blue bars and red bars are used to represent positive and negative

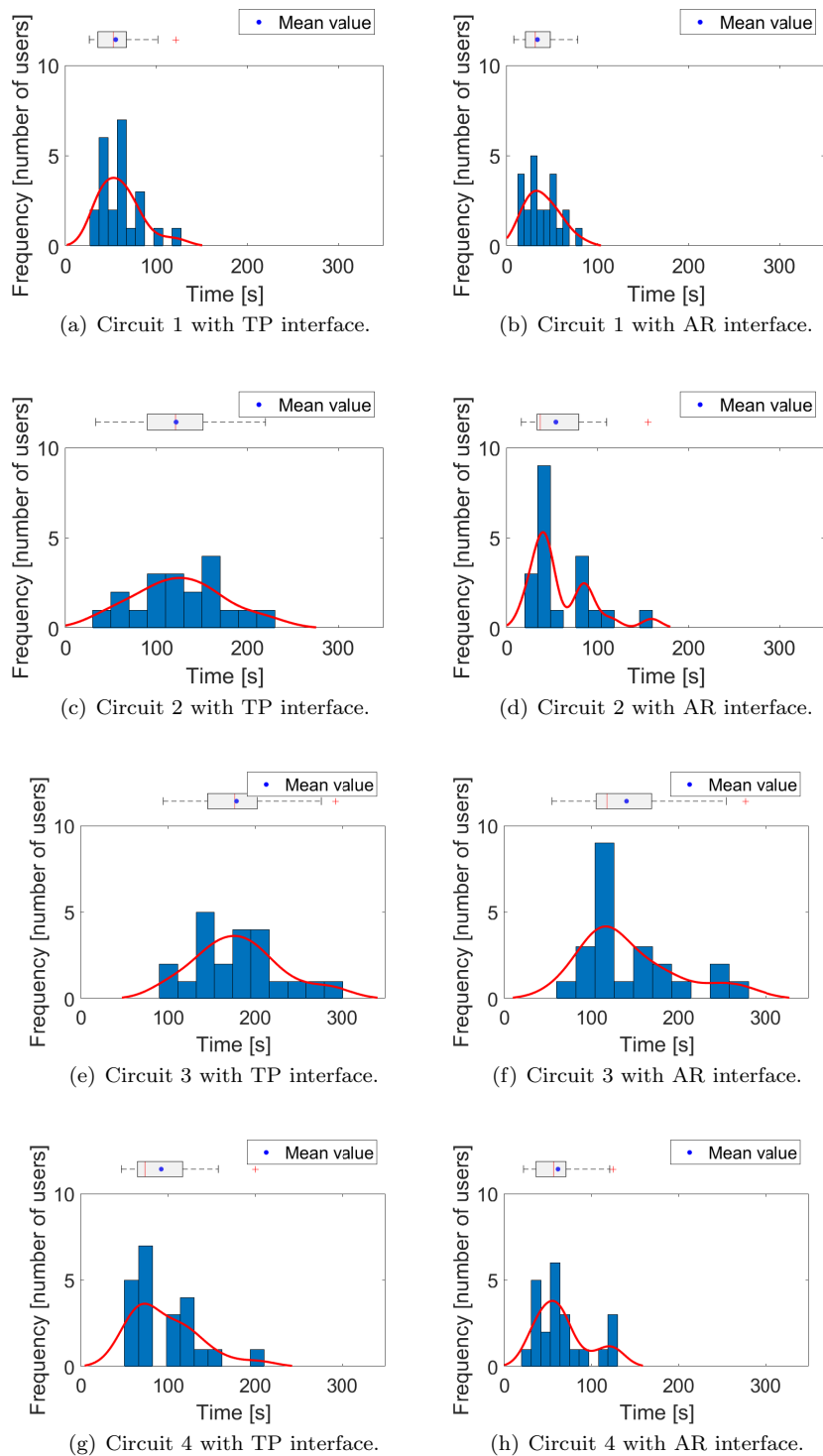


Fig. 10. Comparison of the time histograms obtained for each circuit with the TP and AR interfaces.

Table 2. Comparison of the amount of time that took the participants to carry out each circuit with the TP and AR interfaces. Nomenclature: \bar{t}_i mean value for circuit i ; σ_i standard typical deviation for circuit i .

	CIRCUIT			
	1 { \bar{t}_1, σ_1 }	2 { \bar{t}_2, σ_2 }	3 { \bar{t}_3, σ_3 }	4 { \bar{t}_4, σ_4 }
TP interface	{60.5s, 23.3s}	{125, 7s, 48.9s}	{181.5s, 49.8s}	{96.8s, 38.6s}
AR interface	{39.4s, 18.3s}	{59.3s, 34.9s}	{143.9s, 56.6s}	{66.8s, 29.9s}
Improvement of \bar{t}_i using AR $\left(\frac{\bar{t}_{i,TP} - \bar{t}_{i,AR}}{\bar{t}_{i,TP}}\right)$	35%	53%	21%	31%

Table 3. Total number of failures.

	CIRCUIT			
	1	2	3	4
TP interface	0	4	1	1
AR interface	0	3	0	0

improvements, respectively. Note that most of the participants improved their performance using the AR interface, i.e., only 12 out of 92 (23x4) cases had a negative improvement.

An interesting aspect to be analyzed is the dependence of the results discussed above on the participant previous skills and abilities in the fields of robotics and video games. For this purpose, four groups of participants are considered: participants who frequently play video games; participants who never play video games; participants who had managed robots; and participants who had never managed robots. For each of these four groups it was computed the following improvement ratio: $\frac{\bar{t}_{TP} - \bar{t}_{AR}}{\bar{t}_{TP}}$ [%], where \bar{t}_{TP} and \bar{t}_{AR} are the mean values of the achieved times using the TP and AR interfaces, respectively.

Fig. 12 comparatively shows the improvement obtained by these four groups, where it can be seen that no significant differences have been obtained: frequent video game players and not video game players obtained similar improvements, see Fig. 12(a); whereas participants who had managed robots and participants who had never managed robots also obtained similar improvements, see Fig. 12(b). Therefore, an important conclusion that can be made from this results is that the proposed approach is *intuitive and easy-to-use*, since the participants not familiar with robots and the proposed devices and AR technology obtained a similar improvement to those participants with previous knowledge on these subjects.

Fig. 13 presents the scores of the NASA-TLX questionnaire for the TP and AR interfaces. Note that the mental demand in the case of the TP interface was 63.5%, which is a high score for these kind of tasks, whilst in the case of the AR interface was 44.1%, which is a typical score for these kind of tasks. The same occurs in the case of the physical demand, a score of 46.5% was

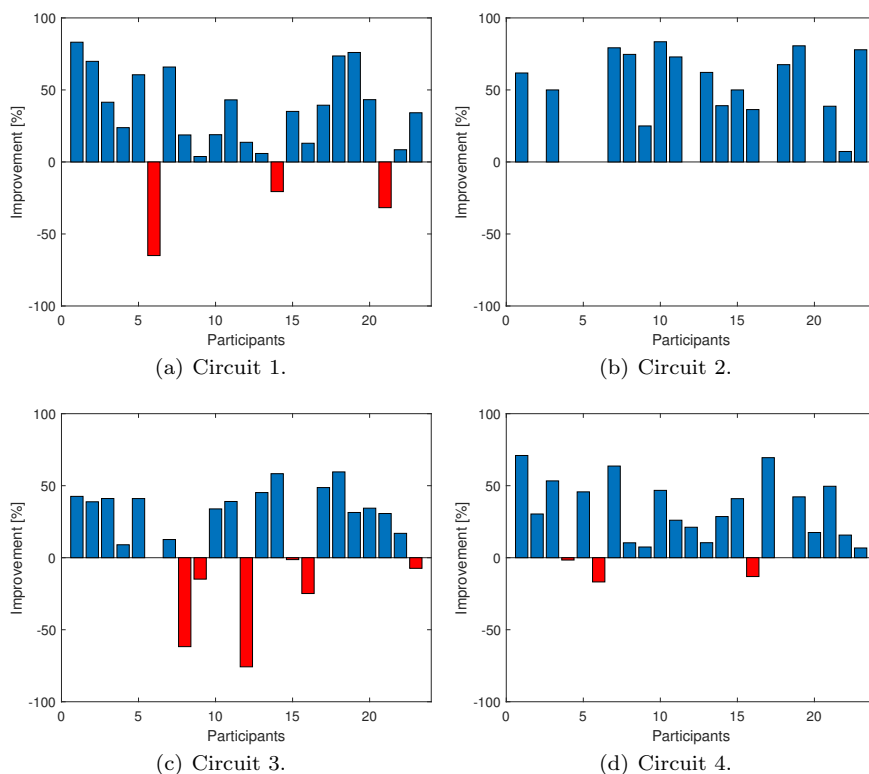


Fig. 11. Improvement obtained by the participants using the AR interface with respect to the TP interface ($\frac{t_{TP} - t_{AR}}{t_{TP}}$): blue bars and red bars represent positive and negative improvements, respectively.

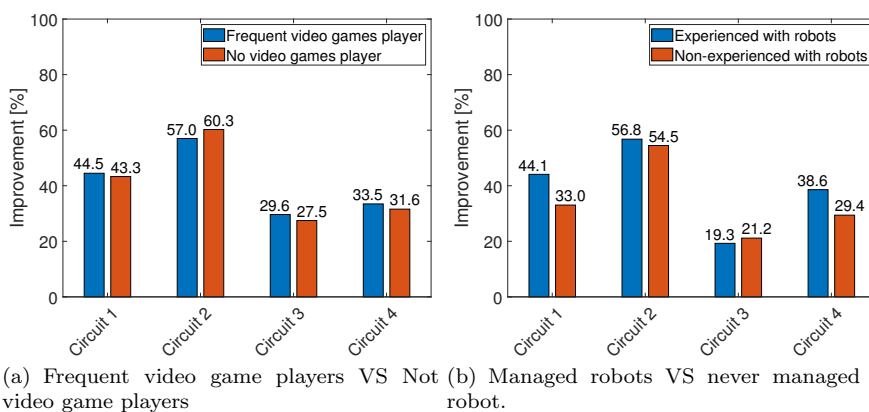


Fig. 12. Comparison of the improvement ($\frac{\bar{t}_{TP} - \bar{t}_{AR}}{\bar{t}_{TP}}$) obtained by the participants according to their previous skills in video games and robotics.

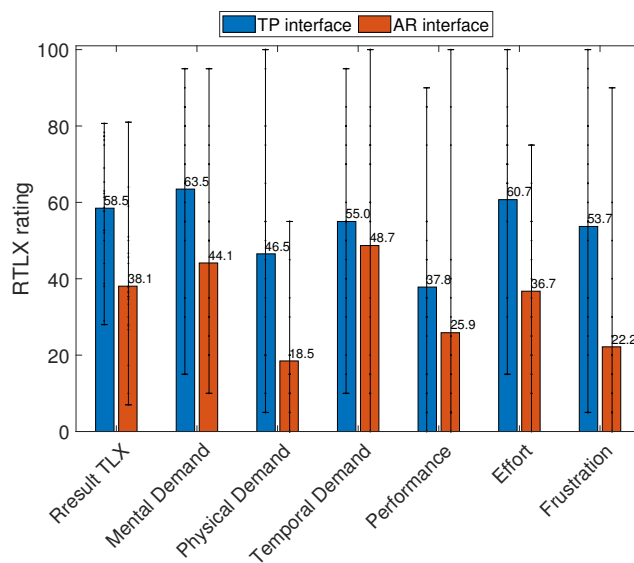
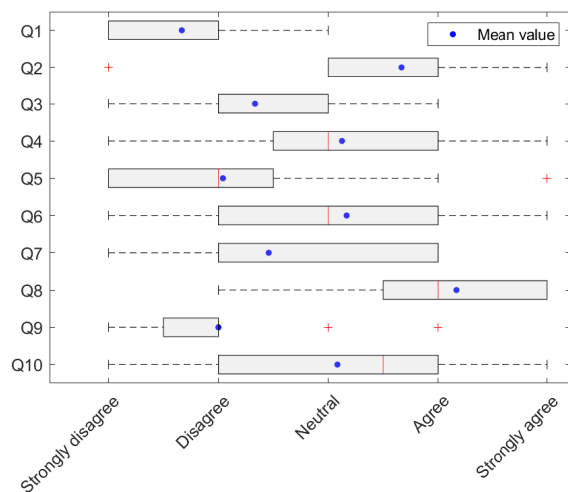


Fig. 13. Results of the NASA-TLX questionnaire.

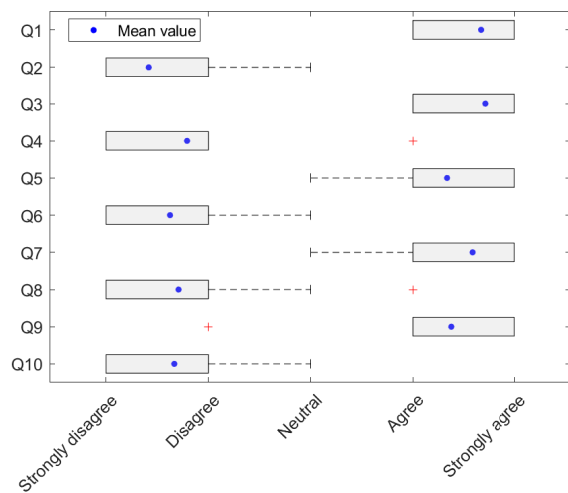
obtained for the TP interface, which is much higher than that obtained for the AR interface, 18.5%. These results are also consistent with the effort and frustration scores, which indicate 60.7% and 53.7%, respectively, for the TP interface, and 36.7% and 22.2%, respectively, for the AR interface.

The results of the SUS questionnaire indicate that the overall perceived usability for the TP interface was 33,2 out of 100 (minimum value: 0; maximum value:62.5; standard deviation: 15.7), which represents a low level of usability, whilst that for the proposed AR interface was 86,3 out of 100 (minimum value: 70; maximum value: 97.5; standard deviation: 9.3), which represents a high level of usability. Moreover, Fig. 14 presents the results obtained for all the questions of the SUS questionnaire, see Table 4. In particular, the comparison in Fig. 14(c) shows that the participants clearly preferred the proposed AR interface over the conventional TP interface. In this sense, most of the participants would prefer to use the proposed AR interface since they found it easier to use. The participants also found the AR interface well integrated and consistent. Furthermore, they felt more confident with the AR interface and believed that it could be utilized by any kind of user regardless of their education level.

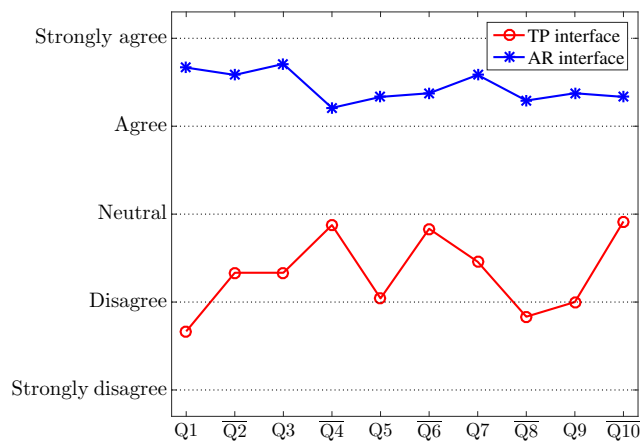
Beside the NASA-TLX and SUS questionnaires, the participants were asked to include remarks about their experience with the TP and AR interfaces, as discussed below.



(a) TP interface.



(b) AR interface.



(c) Comparison of TP and AR interfaces: mean values obtained for the SUS questionnaire. The results of questions {Q2, Q4, Q6, Q8, Q10} have been “negated” in order to express them in a positive way, i.e., for all questions *strongly agree* and *strongly disagree* indicate the best and worst assessment results, respectively.

Fig. 14. Results of the SUS questionnaire.

Table 4. Standard SUS questionnaire [7].

Q1	I think that I would like to use this system frequently
Q2	I found the system unnecessarily complex
Q3	I thought the system was easy to use
Q4	I think that I would need the support of a technical person to be able to use this system
Q5	I found the various functions in this system were well integrated
Q6	I thought there was too much inconsistency in this system
Q7	I would imagine that most people would learn to use this system very quickly
Q8	I found the system very cumbersome to use
Q9	I felt very confident using the system
Q10	I needed to learn a lot of things before I could get going with this system

6 Discussion

The results obtained in the comparison experiment above show that proposed AR interface provides a relevant improvement in terms of velocity in carrying out the tasks. In particular, the results of Table 2, Table 3, Fig. 10 and Fig. 11 show a clear benefit of the proposed AR interface over the conventional TP interface, regardless of the user's previous experience in robotics and/or AR technology, see Fig. 12. Also, the results in Fig. 13 and Fig. 14 indicate that the users clearly preferred the proposed AR interface over the conventional TP interface.

Several insightful comments made by the participants after the tests are discussed below.

Some participants found the AR headset not very ergonomic, although more ergonomic than the TP interface. Other aspect commented was the small field of view of the AR headset, which sometimes made it difficult to be comfortable during the task.

It is worth noting that some of the participants in the comparison tests were experienced workers from Kuka company in Spain, i.e., workers whose main task consists of programming Kuka industrial robots using conventional TP interfaces. The feedback given by these experienced workers has also been very positive, highlighting the ease of interaction with the proposed AR interface and the speed at which any kind of user, experienced or non-experienced workers, can interact with the robot. Moreover, they pointed out that current TP interfaces are not ergonomics, especially when they have to carry them for many hours. Although they also indicated that they thought that the AR headset used in the tests was too heavy to carry it for a long period of time.

Regarding these AR headset issues pointed out by experienced and non-experienced users, it is worth mentioning that AR headset manufacturers are

currently developing new devices in order to meet industrial needs and demands. For instance, the new Microsoft AR headset, expected to be available during the year 2020, has new characteristics that make the AR headset more ergonomic [30].

7 Conclusions

This work developed an augmented reality interface to teleoperate industrial robot manipulators by experienced and non-experienced users. To achieve this goal, augmented reality (i.e., a real environment that includes computer-generated graphics) and visual and sound feedback were used to provide the user information about the task in real-time. In addition, the interaction of the user with the augmented environment and the robot was carried out using a gamepad. Furthermore, the main functionalities of conventional teach pendant interfaces were implemented.

The feasibility and benefits of the proposed interface was proven through real experimentation using a 6R industrial robot. In this sense, several usability tests were carried out with experienced and non-experienced users to show the benefits and drawbacks of the proposed interface compared to a conventional teach pendant interface. In this sense, the results showed that the proposed augmented reality interface was more intuitive, ergonomic and easy-to-use than the conventional teach pendant interface. Furthermore, the comparison results also showed that the proposed method clearly improves the velocity of the teleoperation task, regardless of the user's previous experience in robotics and augmented reality technology.

Acknowledgements This work was supported by the Spanish Government under the project DPI2017-87656-C2-1-R.

References

1. ABB: Abb teach pendant (Accessed 04/24/2020)
2. Attig, C., Wessel, D., Franke, T.: Assessing personality differences in human-technology interaction: An overview of key self-report scales to predict successful interaction. In: C. Stephanidis (ed.) *HCI International 2017 – Posters' Extended Abstracts*, pp. 19–29. Springer International Publishing, Cham (2017)
3. Bandala, M., West, C., Monk, S., Montazeri, A., Taylor, C.J.: Vision-based assisted tele-operation of a dual-arm hydraulically actuated robot for pipe cutting and grasping in nuclear environments. *Robotics* **8**(2), 42 (2019)
4. Bermejo, C., Hui, P.: A survey on haptic technologies for mobile augmented reality. *ArXiv abs/1709.00698* (2017)
5. Billard, A., Calinon, S., Dillmann, R., Schaal, S.: *Robot Programming by Demonstration*, pp. 1371–1394. Springer Berlin Heidelberg, Berlin, Heidelberg (2008)
6. Bostanci, E., Kanwal, N., Ehsan, S., Clark, A.F.: User tracking methods for augmented reality. *International Journal of Computer Theory and Engineering* pp. 93–98 (2013)
7. Brooke, J.: "SUS-A quick and dirty usability scale" *Usability evaluation in industry*. CRC Press (1996). ISBN: 9780748404605

8. Cardoso, J.C.S.: Comparison of gesture, gamepad, and gaze-based locomotion for VR worlds. In: Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology, pp. 319–320. Association for Computing Machinery, New York, NY, USA (2016)
9. Circuit 1 demonstration: <https://media.upv.es/player/?id=8be7f8e0-99b4-11ea-a399-57da8aaa21ee> (Accessed 05/14/2020)
10. Circuit 2 demonstration: <https://media.upv.es/player/?id=afda6530-99b4-11ea-a399-57da8aaa21ee> (Accessed 05/14/2020)
11. Circuit 3 demonstration: <https://media.upv.es/player/?id=56c62d20-99b5-11ea-a399-57da8aaa21ee> (Accessed 05/14/2020)
12. Circuit 4 demonstration: <https://media.upv.es/player/?id=9461f740-99b5-11ea-a399-57da8aaa21ee> (Accessed 05/14/2020)
13. Codd-Downey, R., Jenkin, M.: Wireless teleoperation of an underwater robot using li-fi. In: 2018 IEEE International Conference on Information and Automation (ICIA), pp. 859–864 (2018)
14. Craig, A.B.: Chapter 2 - augmented reality concepts. In: A.B. Craig (ed.) Understanding Augmented Reality, pp. 39 – 67. Morgan Kaufmann, Boston (2013)
15. Fanuc: Fanuc ipendant (Accessed 04/24/2020)
16. Ferreira, A., Bastos-filho, T.F., null, Cheein, F.A., Postigo, J.F., Carelli, R.: Teleoperation of an industrial manipulator through a tcp/ip channel using eeg signals. In: 2006 IEEE International Symposium on Industrial Electronics, vol. 4, pp. 3066–3071 (2006)
17. Fondazione Istituto Italiano di Tecnologia: Force/torque sensors (2015). Accessed 04/04/2020
18. Franke, T., Attig, C., Wessel, D.: A personal resource for technology interaction: Development and validation of the affinity for technology interaction (ati) scale. International Journal of Human-Computer Interaction **0**(0), 1–12 (2018)
19. Gadre, S.Y., Rosen, E., Chien, G., Phillips, E., Tellex, S., Konidaris, G.: End-user robot programming using mixed reality. In: 2019 International Conference on Robotics and Automation (ICRA), pp. 2707–2713 (2019)
20. Grahn, I.: The vuforia sdk and unity3d game engine : Evaluating performance on android devices
21. Hart, S.G., Staveland, L.E.: Development of nasa-tlx (task load index): Results of empirical and theoretical research. In: P.A. Hancock, N. Meshkati (eds.) Human Mental Workload, *Advances in Psychology*, vol. 52, pp. 139 – 183. North-Holland (1988)
22. Hess, R.: Blender Foundations: The Essential Guide to Learning Blender 2.6. Focal Press (2010)
23. Isop, W.A., Gebhardt, C., Nageli, T., Fraundorfer, F., Hilliges, O., Schmalstieg, D.: High-level teleoperation system for aerial exploration of indoor environments. *Frontiers in Robotics and AI* **6**, 95 (2019)
24. Jackson, S.: Unity 3D UI Essentials. Packt Publishing (2015)
25. Kitson, A., Hashemian, A.M., Stepanova, E.R., Kruijff, E., Riecke, B.E.: Comparing leaning-based motion cueing interfaces for virtual reality locomotion. In: 2017 IEEE Symposium on 3D User Interfaces (3DUI), pp. 73–82 (2017)
26. Kuka: Kuka smartpad teach pendant (Accessed 04/24/2020)
27. Li, C., Fahmy, A., Sienz, J.: An augmented reality based human-robot interaction interface using kalman filter sensor fusion. *Sensors* **19**(20), 4586 (2019)
28. Liang, C., Liu, C., Liu, X., Cheng, L., Yang, C.: Robot teleoperation system based on mixed reality. In: 2019 IEEE 4th International Conference on Advanced Robotics and Mechatronics (ICARM), pp. 384–389 (2019)
29. Marinho, M.M., Adorno, B.V., Harada, K., Deie, K., Deguet, A., Kazanzides, P., Taylor, R.H., Mitsuishi, M.: A unified framework for the teleoperation of surgical robots in constrained workspaces. 2019 International Conference on Robotics and Automation (ICRA) (2019)
30. Microsoft HoloLens (2nd gen) hardware details: <https://www.microsoft.com/en-us/hololens/hardware> (Accessed 04/24/2020)
31. Munawar, A., Fischer, G.: A surgical robot teleoperation framework for providing haptic feedback incorporating virtual environment-based guidance. *Frontiers in Robotics and AI* **3**, 47 (2016)

32. noz, A.M., Mahiques, X., Solanes, J.E., Martí, A., Gracia, L., Tornero, J.: Mixed reality-based user interface for quality control inspection of car body surfaces. *Journal of Manufacturing Systems* **53**, 75 – 92 (2019)
33. noz, A.M., Mahiques, X., Solanes, J.E., Martí, A., Gracia, L., Tornero, J.: Mixed reality-based user interface for quality control inspection of car body surfaces. *Journal of Manufacturing Systems* **53**, 75 – 92 (2019)
34. noz, A.M., Martí, A., Mahiques, X., Gracia, L., Solanes, J.E., Tornero, J.: Camera 3D positioning mixed reality-based interface to improve worker safety, ergonomics and productivity. *CIRP Journal of Manufacturing Science and Technology* **28**, 24 – 37 (2020)
35. Park, D.Y.: Mrtk: Open-source building blocks for windows mixed reality experiences (2017). <https://github.com/cre8ivepark/>
36. Rosen, E., Whitney, D., Phillips, E., Chien, G., Tompkin, J., Konidaris, G., Tellex, S.: Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *The International Journal of Robotics Research* **38**(12-13), 1513–1526 (2019)
37. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, UK (2009)
38. Vitor, R., Keller, B., D’Angelo, T., Azpurua, H., Bianchi, A.G.C., Delabrida, S.: Collaborative teleoperation evaluation for drones. In: *Proceedings of the 18th Brazilian Symposium on Human Factors in Computing Systems, IHC ‘19*. Association for Computing Machinery, New York, NY, USA (2019)
39. Walker, M.E., Hedayati, H., Szafir, D.: Robot teleoperation with augmented reality virtual surrogates. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 202–210 (2019)
40. WANG, D., GUO, Y., LIU, S., ZHANG, Y., XU, W., XIAO, J.: Haptic display for virtual reality: progress and challenges. *Virtual Reality & Intelligent Hardware* **1**(2), 136 – 162 (2019)
41. Wei, J., Ye, G., Mullen, T., Grundmann, M., Ahmadyan, A., Hou, T.: *Instant motion tracking and its applications to augmented reality* (2019)
42. Wonnacott, T.H., Wonnacott, R.J.: *Introductory statistics for business and economics*. Wiley New York (1990)
43. Xu, P., Zeng, Q., Zhang, G., Zhu, C., Zhu, Z.: Design of control system and human-robot-interaction system of teleoperation underwater robot. In: H. Yu, J. Liu, L. Liu, Z. Ju, Y. Liu, D. Zhou (eds.) *Intelligent Robotics and Applications*, pp. 649–660. Springer International Publishing, Cham (2019)
44. Yew, A.W.W., Ong, S.K., Nee, A.Y.C.: *Immersive augmented reality environment for the teleoperation of maintenance robots* (2017)
45. Zhao, J., Allison, R.S.: Comparing head gesture, hand gesture and gamepad interfaces for answering yes/no questions in virtual environments. *Virtual Reality* pp. 1–9 (2019)