CAMBRIDGE
UNIVERSITY PRESS

REVIEW ARTICLE

A virtual reality-based dual-mode robot teleoperation architecture

Marco Gallipoli[1] (iD), Sara Buonocore[1] (iD), Mario Selvaggio[2] (iD), Giuseppe Andrea Fontanelli[3], Stanislao Grazioso[1] and Giuseppe Di Gironimo[1]

[1]Department of Industrial Engineering, University of Naples Federico II, Naples, Italy
[2]Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy
[3]Herobots s.r.l., Naples, Italy
Corresponding author: Marco Gallipoli; Email: mar.gallipoli@studenti.unina.it

Abstract

This paper proposes a virtual reality-based dual-mode teleoperation architecture to assist human operators in remotely operating robotic manipulation systems in a safe and flexible way. The architecture, implemented via a finite state machine, enables the operator to switch between two operational modes: the *Approach* mode, where the operator indirectly controls the robotic system by specifying its target configuration via the immersive virtual reality (VR) interface, and the *Telemanip* mode, where the operator directly controls the robot end-effector motion via input devices. The two independent control modes have been tested along the task of reaching a glass on a table by a sample population of 18 participants. Two working groups have been considered to distinguish users with previous experience with VR technologies from the novices. The results of the user study presented in this work show the potential of the proposed architecture in terms of usability, both physical and mental workload, and user satisfaction. Finally, a statistical analysis showed no significant differences along these three metrics between the two considered groups demonstrating ease of use of the proposed architecture by both people with and with no previous experience in VR.

1. Introduction

Robotics is meant to improve the quality of life by taking over dangerous, tedious, and dirty jobs that are impossible to perform or unsafe for humans. However, it is still uncommon to come across robotic systems capable of autonomously meeting this demand. For this reason, the interest in remotely operated robotic systems, possibly equipped with advanced features to support human operators in the decision-making process, is steadily increasing. The term teleoperation refers to the operation of a robot from a remote site, through an adequate human-robot interface (HRI[1]) [1]. In this scenario, any high-level decision is made by the human operator, while the robot is just responsible for its execution. When operating the system becomes difficult, a shared control approach can be used where some aspects are controlled directly by the human and others by local sensory feedback loops, whose aim is to lower down the physical and cognitive effort of the user [2, 3]. In this setup, the use of virtual reality (VR) technology can be highly beneficial to enhance the operator experience, providing an immersive interface that is more engaging and stimulating for the user to operate the remote robot.

In this work, we propose a dual-mode VR-based teleoperation architecture, designed with a participatory design and human-centric approach, aiming to propose a system accessible to both VR experts

---

[1]More in general, the acronym HRI identifies the field *human-robot interaction* which addresses the design, understanding, and evaluation of robotic systems that involve humans and robots interacting.

and novices. It mainly consists of an immersive virtual environment, which constitutes the operator interface that represents the digital twin (DT) of the real robot side, endowed with advanced control, planning, and predictive simulation features. Using the virtual interface, the operator can interact with the system through two operational modes, that is, *Approach* and *Telemanip*, whose functionalities are implemented via a finite state machine (FSM). The introduction of two operational modes enables the user to more effectively realize complex and/or dangerous procedures that otherwise cannot be easily carried out. Moreover, the possibility to choose between the two states increases the efficiency of the control: on the one hand, the operator can quickly realize repetitive operations involving large movements using the *Approach State*, by only specifying the target pose for the robot's end-effector, and on the other hand, the *Telemanip State* allows direct control of the robotic system to realize fine movements that are specific of accurate procedures. The proposed architecture has been realized to control a bimanual bartender robotic system through a Virtual Reality Control Room (VRCR) in order to manage its principal tasks, such as preparing a cocktail or recovering from unexpected situations, that is, a glass is dropped outside the reachability area. In order to simplify future customization, the GitHub repository[2] includes the developed experimental setup (VR local side) which can be modified and adapted to control a different robotic system. The selected case study is taken from the BRILLO project (Bartending Robot for Interactive Long-Lasting Operations) [4], as an example of the possible positive impact of the proposed control logic on such simple yet repetitive operations, where a high level of accuracy and safety is needed.

The rest of this paper is structured as follows. Section 2 addresses the state of the art of eXtended Reality (XR)-based teleoperation systems focusing first on the main advantages and objectives of using XR[3] technology for telerobotics tasks (Section 2.1) and, second, on the main current control strategies and relative interfaces (Section 2.2). Section 3 describes the main components of the teleoperation's architecture, while Section 4 describes the two developed operational modes. Finally, Section 5 describes the experimental setup realized for the BRILLO project and the user study conducted on the developed VRCR. The results of the tests and the future improvements conclude the present work.

1.1. Contributions

As explained in Section 2, the use of VR technology, especially for remote control of robotic systems, aims to enhance the operator experience by providing an immersive interface, which is more engaging and stimulating for the user. Within this context, this paper contributes to the state of the art as follows:

- It proposes a VR-based (more in general XR-compatible) dual-mode teleoperation architecture that allows operators to (i) interactively plan, visualize in VR, and semi-autonomously execute large motions with the remote robotic system and (ii) achieve fine motion regulation via direct, scaled, velocity-based teleoperation of the robot end-effector. In the *Approach* mode, safety is further reinforced by the ability to preview the complete movement of the robotic system, enabling the user to instantly abort the command if any part of the motion is deemed dangerous. In *Telemanip* mode, the user can directly guide the robotic system along a user-specified motion trajectory using the controllers. In this operational mode, the user receives additional information from the scene, aiming to enhance oSA. A state machine is used to manage the transitions among different control modes as explained in Section 4.

- It presents the application of participatory design and human-centric approach to design and develop the proposed dual-mode VR-based teleoperation architecture, aiming to propose a system accessible to both VR experts and novices. With this regard, we discuss the conducted

---

[2]https://github.com/MarcoGallipoli/BRILLO_Experimental-Setup.git
[3]eXtended Reality: a term commonly used to include all technologies related to VR, AR, MR.

experimental campaign with both VR experts and novices to assess the usability, accessibility, and satisfaction related to the use of the proposed system. With this work, we aim to contribute to the current lack of human factors-related studies on XR teleoperation systems.

This contribution represents a major milestone in the effort to connect the user with the robot's space through a VR-based interface in an intuitive, natural, and effective manner.

2. Related works

Advances in emergent technologies such as XR are not static. The increasing popularity of XR technologies in recent years has motivated practitioners and researchers to develop new software artifacts to explore the capabilities of new hardware devices. The current literature offers many works that demonstrate the added value of XR to increase operator Situational Awareness (oSA) in such contexts, mainly covering supervision tasks and robot's path planning/programming. The oSA in a collaborative environment is fundamental, as the operator needs to be properly informed about the robotic system status, ongoing tasks, and planning of future tasks. Coherently with Industry 4.0, it envisioned that the next manufacturing systems paradigm will be an adaptive cognitive manufacturing system, coined as ACMS. The innovative paradigm represents more predictive, adaptive, human-centric manufacturing systems, in which the augmented human abilities will play a central role in enhanced decision-making [5]. The cited complex human operators' decision-making process will be supported by a real-time data flux, taken from the robots' DT. DT of a system or component is the digital replica of the latter that mirrors and/or twins the physical component throughout its active life cycle [5, 6]. In light of this, DT is designed to make it possible to support a healthy relationship between human workers and smart automation, aiming to create a safer, more ergonomic, satisfying environment for workers [1]. Firstly, it appeared in the early 2000s as a standalone simulation model, with no contact with its real counterpart, employed as an offline decision support tool during the design and planning of a manufacturing system [7]. Since those years, DT has greatly expanded its potential. Currently, DT is considered as an integrated multi-physics, multi-scale simulation system that uses the most appropriate model, data history, and sensor updates to mirror the operation of its real counterpart throughout its life from design to implementation and actual operation [5].

In this section, we are going to recap the related works about the main aspects of human-robot collaboration, in light of the innovative ACMS paradigm: human-robot interface and control modality.

2.1. XR-based human robot interfaces

Among all the applications of XR technology, one of the most interesting in the last decades is surely the design, development, and validation of XR-based HRI. The main purpose of an HRI is to achieve effectiveness and safety, intuitiveness, and usability to enable operators to interact, cooperate, and collaborate with robotic systems. The integration of VR and augmented reality/mixed reality (AR/MR) into architectural frameworks promises to revolutionize human-robot interaction for, respectively, remote and on-site operations, offering essential tools to enhance user experience [8–10]. The following list outlines the main advantages and objectives of an XR HRI:

1. **Facilitate programming.** The growing affordability of industrial collaborative robots may lead to an increase in user-tailored robotic systems. However, the demands for customization present challenges, requiring specific programming skills for each robot. Bambusek et al. [11] propose an XR interface for reprogramming robotic systems, indicating a promising approach to simplify interaction and enhance adaptability. In general, research has shown that programming robotic manipulators using XR interfaces offers advantages in program creation [12] compared to conventional methods such as tablet or kinesthetic programming. Additionally, it reduces errors by leveraging virtual simulation in a virtual environment during debugging [13].

2. **Support real-time visualization.** A teleoperation system eliminates the need for the user to be physically present within the robotic environment, facilitating remote control and operation. Remote connection serves as both a necessity for prompt intervention and an opportunity to reduce system recovery time in the event of failures. In terms of visualization, the traditional 2D video user interface suffers from considerable limits regarding the operator's awareness: the visibility is limited to a single fixed viewpoint, and the mapping between operator and robot motions is usually inaccurate. Thanks to an immersive first-person 3D experience, a better understanding of the risks and a more informed decision can be reached by the operator. With this regard, in ref. [14], A. Naceri et. al suggest immersive visualization of a virtual environment that accurately reproduces the robot's perspective. Moreover, ref. [15] enhances the VR interface by integrating RGB-D sensors for scenario reconstruction. Effective implementation of VR technology is expected to enhance understanding and control of the remote environment through improved telepresence [16]. It has been observed that a better user experience can be achieved by enabling the robot to track the speaker while discerning the intention of the remote user. In this regard, ref. [17] proposes a human-robot collaborative control framework based on human intention recognition and sound localization.

3. **Support real-time control.** XR interfaces not only facilitate communication between users and robots but also significantly enhance control capabilities. By overlaying spatial information onto the user's environment, XR tools provide an intuitive interface for commanding and directing robotic actions with precision and clarity. This immersive control mechanism empowers users to manipulate and coordinate robotic tasks seamlessly, leveraging spatial cues to enhance efficiency and accuracy in operation. M. Ostanin et al. show in ref. [18] the adoption of the XR technology to allow the operator to set a goal position for the robot and a series of "via points" in the real environment, through simple gestures. The proposed application's ability to scale the path and utilization of additional cameras/sensors allows to increase the robot positional accuracy, showing that such applications have the potential to be used also for quality estimation after the technological operation.

4. **Improve safety.** XR interfaces have the potential to heighten safety during interactions with robots, particularly in scenarios where real-time movement poses significant risks without supervision. In ref. [19], the operator's control on the robot is not direct, since the user can only interact with the robot by two control elements in the scene that represent the position and orientation of the robot arms' end-effectors. A visual feedback confirms (or not) the feasibility of the required movement for the robot.

5. **Communicate intent.** A well-designed HRI can effectively convey the robot's intentions to the user through spatial information. Works such as refs. [20] and [21] feature control algorithms characterized by the integration of multiple control modalities, further enhancing the interaction between humans and robots. In ref. [20], a differentiation is made between trajectory control, simulating click-and-drag functionality, and positional control, which uses waypoint navigation. On the other hand, in ref. [21], it is possible to command both long-distance and fine movements. In long-distance control, users specify the final position only. In fine control, a continuous input is needed to adjust the robot's movements throughout its trajectory.

6. **Improve productivity.** XR technologies have been demonstrated to be more suitable also for specialized workers, as learning how to use MR interfaces takes less time compared to a classic 72-h training course for industrial robots programming [12]. In ref. [22], authors proposed an on-site application based on MR technology for the visualization of the safety zones as well as the robot's intentions. The added value of this tool is the capability of mapping a robotic arm's environment and consequently facilitating its navigation in a 3D space. The 3D representation of the robot's environment and its visualization in the MR application enable a clearer view of the robot's working environment, in contrast to the existing solution. The conducted experimental campaign has demonstrated that the innovative MR-based solution, compared with the existing

one, has led to a reduction in total assembly time of approximately 24% and an approximately 60% reduction in the number of user errors.

An XR interface improves the user's situational awareness, depth perception, and spatial cognition, as fundamental to effective and efficient teleoperation. The world is passing through a paradigm change toward Society 5.0 and Industry 5.0, and XR technologies are often considered keystone elements of these paradigms. However, such software/hardware solutions are fairly recent, and related human factors have been consistently marginalized so far in telerobotics research. In this paper, we aim to contribute to a deeper understanding of human factors (such as usability, cognitive and physical effort, and satisfaction) related to the use of an XR-based teleoperation architecture, focusing on remote control applications. With this regard, we only discuss the VR-based application of our teleoperation architecture (Sections 4 and 5), but we precise that the presented architecture is easily convertible for on-site operations (AR/MR-based) switching to an AR/MR device connected to the same software platform.

2.2. Telerobotic system control

Telerobotics literally means "robotics at a distance," and it is generally understood to refer to robotics with a human operator in control or human-in-the-loop [1]. Telerobotic systems are generally constituted by two sides: a local operator side, composed of the required systems to send commands to the robot and to receive information about its state, and the remote robot side, which includes the real robot, supporting sensors, and control elements. The physical separation between the two sides can be very small; the robot and the human user can be in the same room as in surgical settings [23] or, alternatively, in two very distant places [24], depending on the application. In most cases, robots are commanded by remote human operators to carry out work in hazardous or uncertain environments such as nuclear plants or outer space. To successfully carry out remote tasks with such systems, it is important to adopt an appropriate control strategy that lets the operator feel physically present at the remote site.

The most effective way to achieve high levels of human involvement (or telepresence) is to implement a bilateral exchange of information between the two sides. This control strategy allows the exchange of data between the local and the remote side, such that forces and torques sensed by the robot can be fed back to the user. Although this technique assures the operator a deep awareness of the interacting robotic system's state, it is very complex to implement, and it could be unstable due to communication delays which, in turn, can influence the fidelity of the information feedback [25]. In recent years, researchers have explored the integration of force feedback in robotic teleoperation systems, aiming to enhance the oSA through haptic feedback. While this approach offers exciting possibilities, it also presents several technical challenges:

- **Complex implementation.** Force feedback provides operators with a deeper awareness of the interacting robotic system's state. However, its implementation is intricate due to factors such as communication delays. Balancing real-time responsiveness and stability remains an ongoing challenge.
- **Safety and transparency.** Ensuring safety during teleoperation is crucial. Operators must accurately perceive forces to prevent collisions or unintended movements. Absolute transparency – where the operator feels directly connected to the robot – is an ideal goal.

A unilateral teleoperation may alternatively be considered, as it is simpler and more stable than the previous one. In this case, the information flow is in one direction, from the local robot interface, guided by the operator, to the remote side [26].

Another aspect that determines the amount of human involvement in the control of a telerobotic system is the level of intelligence or autonomy [27]: on the one side, when no intelligence or autonomy in the system is present, every aspect is directly controlled by the user via the HRI; on the opposite side, the operator can give supervisory high-level commands, which are then refined and executed by

the robot autonomously [1]. When the task's execution is shared, some aspects are controlled directly by the human and others by local sensory feedback loops, whose aim is to lower down the physical and cognitive effort of the user [2, 3]. When the user instead must retain a high level of involvement in the control of the system, haptic or visual cues can be used to provide assistance through appropriate interfaces [28]. For instance, for tasks involving grasping an object, a target-guided control strategy, such as the one proposed in ref. [29], can be adopted: a vision-based algorithm can be used to estimate and predict the next user's target and accordingly provide haptic assistance in the form of virtual fixtures.[4]

As explained in Section 2.1, the recent gains in its capabilities and popularity are making VR interfaces an ideal candidate to generate the realistic and immersive experience needed to teleoperate a robot at a distance while feeling physically present at the remote side. To enable this, users are immersed in a VR control room with multiple sensor displays, feeling like they are inside the robot's head [30]. The movements of their head and hands are retrieved through appropriate sensors and matched to the robot's movements to complete various tasks. In this setting, the user can interact directly with the real robotic system or with a virtual copy of the robot and the environment [31]. In this way, the user is constantly receiving visual feedback from the virtual world overcoming the instability problems caused by possible delays. VR environments can accurately recreate the robot dynamics and the resulting force feedback resulting from the execution of complex tasks, such as bolting and various other dexterous object manipulation activities. The users can additionally interact with controls that appear in the virtual space to, for example, open and close the hand grippers to pick up objects or switch among control modalities. Using this strategy, the human's space is mapped into the virtual space, and the virtual space is then mapped into the robot space to provide a sense of co-location.

3. VR-based teleoperation architecture

As introduced above, a teleoperation system is generally constituted by two distinct sides, communicating with each other: the local side, in a case featuring a VR-based interface, and the remote robot side (Fig. 1). The two sides could be in the same work area or in two distant sites. The data exchange system can either be wired (e.g., via Ethernet) whether they are in the same area or, if required, wireless.

For our purpose, we consider a system in which both the local and remote sides have dedicated workstations; as for the remote one, the workstation interacts with the robot itself through the robot cabinet. Each station can communicate by exchanging messages as shown in Fig. 1. From the local workstation, user tracking data and user input are transmitted to the remote workstation. On the other hand, it receives the tracked markers' pose measured by the remote workstation and the robot's state. The operational mode can be requested by the user, but for safety reasons, it is enabled by the state machine module only in case of no other ongoing activity. During the teleoperation, the remote workstation receives different types of data according to the actual state: the end-effector target pose, in the *Approach State*, or the controller's velocity, in the *Telemanip State*. In case of the target pose, the planner calculates the entire trajectory and sends it to the cabinet. Differently, the controller's tracking, appropriately scaled, is used to compute end-effector velocity, which is then transmitted to the cabinet through the commander module. At this stage, the commands are translated into joint velocity commands, ready to be received by the robotic system. Finally, data related to the robot state are collected from the real environment and sent back to the local workstation.

With reference to Fig. 1, the following two sections describe the main modules/features of the two sides, while Section 4 contains the description of the proposed dual-mode teleoperation architecture implemented in the BRILLO project.

---

[4]Virtual fixtures, which are also referred to as active constraints, are haptic-based control algorithms that can aid humans in collaborative manipulation tasks with machines; an exhaustive overview can be found in [42].
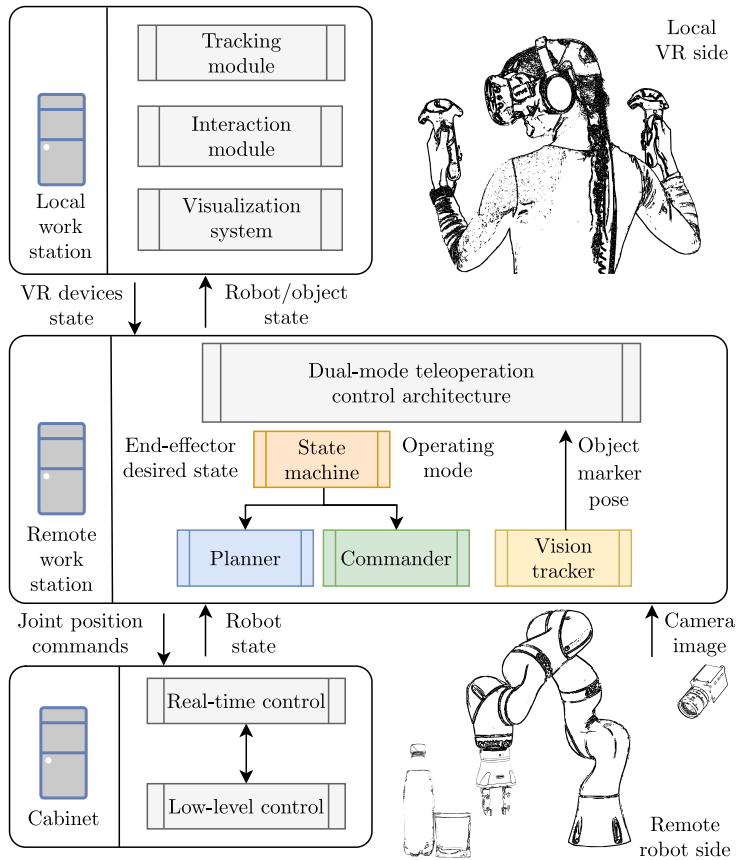
Figure 1. *Scheme of the proposed teleoperation architecture composed of two sides: the local VR side (upper part of the image) features a local workstation with markers' tracking, interaction, and visualization modules for the human operator; the remote robot side (lower part of the image) features a remote workstation that implements the proposed dual-mode teleoperation control architecture that communicates with the robot cabinet responsible of the low-level real-time robot control.*

3.1. Local VR side

The VR side includes systems required to make the operator aware of the real scenario and to enable a safe interaction with the robot. The local workstation, as shown in Fig. 1, is composed of three main components: tracking module, interaction module, and visualization system. Given the operator's potential distance from the real robot, it becomes necessary to digitally reproduce the remote scene. With a 3D visualization of the system, the operator can make more informed decisions. Consequently, immersing the operator in a VR environment provides an accurate representation of the robot's surroundings, enhancing their engagement with the scenario. When immersed in the virtual scene, the operator should be able to know exactly the real objects' poses; these are retrieved by means of a vision-based tracker acting at the remote side. In order to accurately reconstruct the scene in the virtual scenario, the objects are rigidly attached to markers whose pose can be easily measured by the vision tracker module. The relative pose between the marker and the corresponding object is considered as constant during the teleoperation. Once received the markers' pose through the data exchange system, the scenario is meticulously recreated in the VR framework. Additionally, to augment the user's consciousness of the robot side, a 2D video feedback is included in the 3D simulation. It serves as a real-time visual representation of the actual scenario, enabling the user to see an area of the real environment in the virtual one. In case of absence of the 2D video feedback, the operations would rely on the accuracy of the 3D simulation,
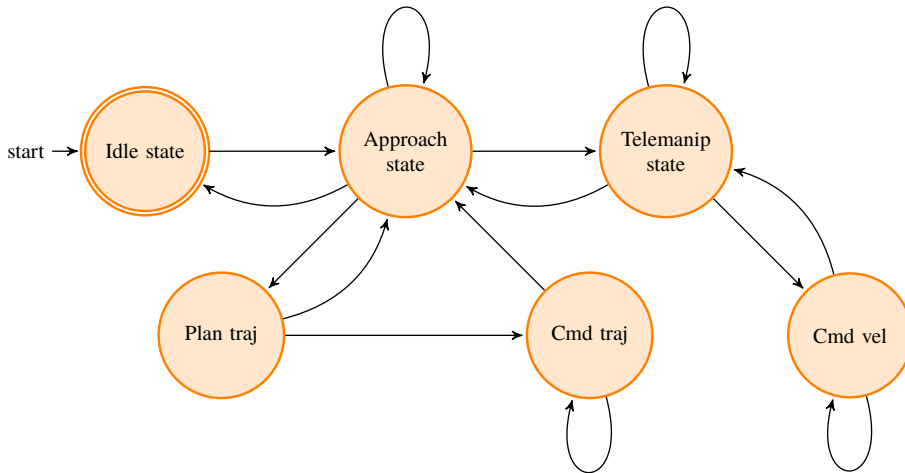
Figure 2. *State machine of the proposed dual-mode teleoperation architecture. Approach and Telemanip are the main states besides the Idle one. In the Approach State, the operator can move around and place the virtual end-effector in the desired pose, plan a trajectory for the robot within the Plan traj state, and subsequently visualize it through the Cmd traj state. In the Telemanip State, instead, the user can realign input devices or command the real robot end-effector velocity in the Cmd vel state. Switching among states is triggered by pressing the input devices buttons.*

which may not be reliable enough. Therefore, the introduction of a 2D video feedback is an additional information which increases the user's awareness and the system's safety. In the virtual scenario, the operator is able to move in order to see the scene from a different point of view. The virtual motion is caused by a real movement of the operator which is tracked by a system of cameras. Moreover, since realizing wide movements in the real area could be dangerous for the user, the virtual movement can be additionally controlled by the VR devices, that is, gloves or controllers.

In the proposed HRI, a one-way interaction is developed; indeed, according to the chosen state, the user can interact only with virtual objects using VR devices. As described in Section 4, in the *Approach State*, it is possible to grab a virtual robot gripper and move it to the target pose; when the user realizes the required commands, the target gripper pose is sent to the remote workstation, and the trajectory is planned and executed to reach the target pose without incurring into possible collisions. On the other hand, in *Telemanip State*, the VR devices are tracked to allow a direct control of the robot.

3.2. Remote robot side

The robot side is composed of two main components: the remote workstation for high-level control and the cabinet for low-level control. The remote workstation implements the dual-mode teleoperation control architecture which is composed of four modules: a state machine, a planner, a commander, and a vision tracker (see Fig. 1).

In order to make the dual-mode teleoperation control architecture usable and maintainable, it is implemented via a state machine, which constitutes its core (see Fig. 2). This is divided into simple construction parts, the states, describing a sequential behavior of a control program [32]. At the starti of the teleoperation, the operator can freely choose the operational mode, while during the operations, to avoid an undesired and dangerous change of state, the operator can just ask to enter a new state. Once the operator has requested to change state, the algorithm checks if there is any ongoing operation that could be dangerous to suddenly interrupt. Therefore, it is possible to actually change state if the robotic system is not controlled by the user, or, in other words, the state machine is not in one of the following states:

- Plan traj: the user has just sent the final target pose to the remote robot side and is waiting for the trajectory's computation.
- Cmd traj: the robotic system is realizing the previously computed trajectory.
- Cmd vel: the robotic system is directly controlled by the user.

This review increases the safety of the system, preventing the user from accidentally changing state.

As better detailed in Section 4, according to the chosen state, the user can directly or indirectly control the robot. When indirect control is enabled through the communication link, the remote workstation can receive the target pose sent by the operator from the local workstation.

Given the desired pose, the planner module tries to identify a possible trajectory for the robot taking into account environmental as well as inherent robotic system constraints (such as joint limits). If a feasible motion plan is found, the result can be seen in the remote workstation in the motion planning framework and, additionally, in the local workstation in the virtual scenario. Once visualized in the VR framework, the trajectory can be approved or disregarded through the interface as explained in Section 4.1. If it is approved, the module commander sends the trajectory to the cabinet, enabling the movement of the real robotic system. On the other hand, if the direct control is enabled, the operator sends to the remote workstation the desired movement which is sent to the cabinet by the commander module.

In the remote workstation, it is necessary to define the robot state and the objects' relative pose to reconstruct the real scenario. In order to characterize the robot condition, proprioceptive sensors measure real robot state data, that is, joint positions and Cartesian pose of the robot end-effector. Moreover, to recreate the robot side in the virtual scenario, a vision tracker module is included. The module is composed of at least one camera that has a double function: it allows the acquisition of a 2D video feedback of the scene which can be used as described in Section 3.1 and allows to track the marker's pose on the robot side.

4. Dual-mode teleoperation control

The proposed teleoperation architecture is based on two main operational modes (*Approach State* and *Telemanip State*). This duality has been introduced to allow a safer and more accurate control of the robot. The architectural framework's structure described here can serve as a template for applications that can take advantage of using a dual-mode teleoperation control method. Considering the general setup described in Sections 4.1 and 4.2, customization of the architecture is feasible, as discussed in Section 5.1. This control logic proves to be advantageous in scenarios involving both stationary or mobile robots, offering the opportunity to select between two distinct control methodologies. By referencing the GitHub repository,[5] it is possible to create a customized project based on the dual-mode teleoperation architecture. This process facilitates the creation of novel experimental setups that align with the architecture we have presented.

4.1. Approach State

The *Approach State* allows the user to control the virtual robot by commanding a target pose. The operator, in the immersive control room, receives information about the robot side through the visualization of the DT and the streaming of 2D video from remote cameras. Therefore, the scene can be visualized in 3D simulation, and additionally, as a safety measure, the user can see the actual scene through a virtual screen inside the simulation. In the virtual scenario, the user can see the preview of the required movement in the presence of two DTs of the real robotic system (shown later in Fig. 3):

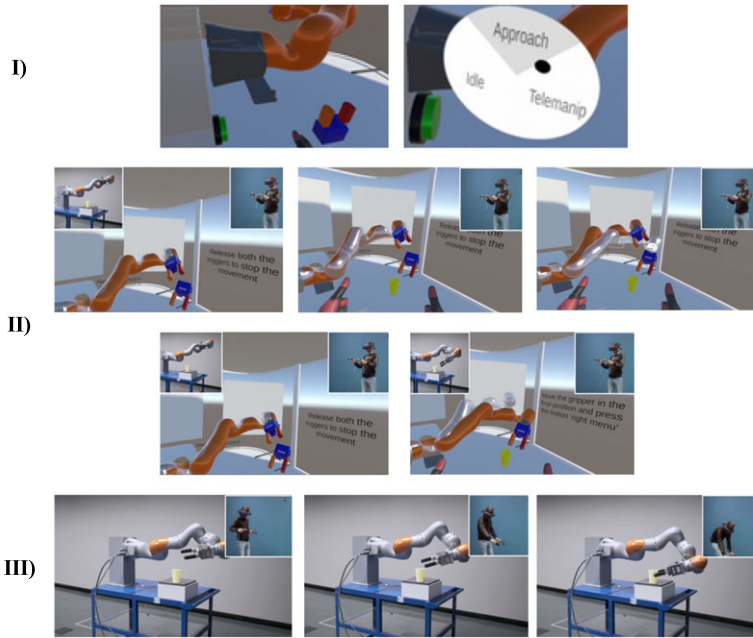[5]https://github.com/MarcoGallipoli/BRILLO_Experimental-Setup.git

Figure 3. *Task execution phases. In phase I, the operator can see the Idle state and open the disk menu to choose the next state. In phase II, the Approach State has been enabled, and it is possible to send the desired pose and to control the Transparent arm and the Opaque arm. In phase III, the user can directly control the Opaque arm to accomplish the task.*

- An opaque twin: the DT of the real robot, reproducing faithfully and directly its movements.
- A transparent twin: an additional virtual replica employed only to show the preview of the commanded movement.

In order to create a reliable virtual scenario, a reference frame on the robot side and its analogous in the virtual environment have been defined. The markers' tracking at the robot side is realized using the ArUco markers.[6] They were chosen for their simplicity, but the system will receive upgrades in the future to incorporate more accurate tracking algorithms. Moreover, to allow a safer control of the robotic system, the obstacles' pose and dimensions are taken into account by the planner. The main obstacles are simplified and represented by cubic shapes with specific dimensions to avoid an unnecessary heavy data flow. At this stage, the planned trajectory, visualized as a preview in the virtual environment, has no influence on the real robot side.

To command a target pose, the operator can interact with the DT of the gripper and grab and release it in the desired position and orientation. When the user confirms it, the chosen pose is sent to the remote workstation as the desired target. As a safety measure, when the command is sent, the target pose cannot be updated unless the trajectory is aborted. In the remote workstation, the real environment has been offline reconstructed in MoveIt[7] motion planning framework, which incorporates the most advanced planners for our scope. Once coded, the environment and the robot can be visualized through the Rviz interface. When the desired robot configuration is received, a planning request is created in the remote workstation and executed by the MoveIt-integrated RRTstar planner. The maximum planning time has been set to 15 s, while the goal tolerance and maximum velocity/acceleration scaling factor have been,

---

[6]An ArUco marker is a square-shaped marker that is synthetic, with a black background and a white inner matrix that depends on the ID of each marker (http://wiki.ros.org/aruco_detect).

[7]https://moveit.ros.org/

respectively, set to 0.04 and 0.2 m. If a feasible motion plan is found, the planner response is a complete yet sparse joint-space robot trajectory. A resampling is thus performed at the robot control cycle time equal to 0.01 s to obtain a smoother one. This can be visualized by the human operator on the VR side and approved or disregarded through the interface before executing it on the real robot. Additionally, the movement of the DT (both transparent and opaque) and, consequently, of the real robotic system can be directly enabled and disabled anytime the user requires, in order to immediately pause (and continue) the ongoing task for any reason.

4.2. Telemanip State

The *Telemanip State* allows the operator to directly control the robot. In this state, the user can interact directly with the robot; therefore, the transparent twin is not in the virtual scenario, and there is not a preview of the movement. As in the *Approach State*, the operator receives information about the actual state and the real environment through a 2D video feedback and a DT of the robotic system. In addition, in this state, the user can see a line that links the end-effector and the virtual target, and the distance between them is constantly updated.

To realize a direct and safe control of the system, the translation is realized by a gradual movement: the operator moves its controllers whose linear velocity is computed and scaled. Therefore, the new velocity is used to move the end-effector. In order to realize an extended movement, the operator can activate and deactivate the movement along the chosen direction. On the other hand, in terms of rotation, the end-effector aligns itself with the controller's orientation. We now proceed to describe how the movements of the human extracted from the controllers are encoded into the corresponding commands. The desired velocity command is extracted at the local side from the controllers' movements and is represented by a twist vector $\mathcal{V}_l$ containing both the linear and the angular velocity components (v, ω). In our case, however, $\mathcal{V}_l$ is not the full controller twist, but the angular part ω is computed from the orientation error as follows:

$$\omega = \frac{1}{2} \left(S(a_i) \, a_l + S(s_i) \, s_l + S(n_i) \, n_l \right)$$

where a_*, s_*, $n_* \in \mathbb{R}^3$ are unit vectors corresponding to the initial (i) and desired (l) rotation matrices. In this way, linear velocities v_l as extracted by the controllers are mapped to linear velocities of the end-effector v_r, while the incremental rotation of the controllers R_l is used to compute angular velocities for the robot end-effector ω_r. The rationale behind this choice stems from the fact that it is much harder for a human to control angular velocities rather than rotations as opposed to the corresponding linear quantities [33].

To clarify all the other computation steps that are carried out within the *Telemanip State* phase, we provide the pseudocode of its implementation: Algorithm 1 shows the initialization and the main loop of the *Telemanip State*. Given the initial end-effector pose $p_r = p_{r,0}$ and $R_r = R_{r,0}$, joint states $q = q_0$ and $\dot{q} = 0$ (measured entering the *Telemanip State*), and the controllers velocity $\mathcal{V}_l$ (computed as explained above), the sequence of looped instructions to retrieve remote robot joint position commands is shown. First, command scaling and rotation are carried out as follows to compute the desired robot end-effector twist $\mathcal{V}_r$:

$$\mathcal{V}_r = sR\mathcal{V}_l,$$

where s is the scaling factor and R is a (6×6) spatial rotation matrix, fixed to match the movements of the controllers to the robot end-effector ones, to render the teleoperation procedure more intuitive.

The upper and lower position limits ($p_{r,u}$ and $p_{r,l}$, respectively) are enforced via the *checkLimits* function by saturating the desired velocity to zero when the next commanded position would exceed them, that is,

$$v_r = 0 \quad \text{if} \quad \left(p_r + v_r dt \geq p_{r,u} \text{ and } v_r > 0 \right) \text{ or } \left(p_r + v_r dt \leq p_{r,l} \text{ and } v_r < 0 \right).$$

Joint velocities are then computed by resorting to a differential inverse kinematics approach using the Jacobian pseudoinversion with a secondary task projected into the null space of the first task's Jacobian.

Algorithm 1 Telemanip State

Data: VR controllers velocity $\mathcal{V}_l$
Result: Robot joint position commands
$p_r = P_{r,0}; R_r = R_{r,0};$ /* End-effector pose initialization */
$q = q_0; \dot{q} = 0;$ /* Joint states initialization */
while *Telemanip State: true* **do**
  $\quad \mathcal{V}_r = \text{s R } \mathcal{V}_l;$ /* Command scaling and rotation */
  $\quad \mathcal{V}_r = \text{checkLimits}(v_r, p_r, p_{r,u}, p_{r,l})$;   /* Check limits */
  $\quad \dot{q} = J(q)^\dagger \mathcal{V}_r + N\dot{q}_N$;   /* Compute joint velocities */
  $\quad q = q + \dot{q} \, dt$;   /* Compute new joint positions */
  $\quad p_r = p_r + v_r \, dt$;   /* Compute new Cartesian pose */
  $\quad R_r = R_r S(\omega_r)$
end

The secondary task has been chosen such that it maintains the robot manipulator as close as possible to its starting configuration. To this end, $\dot{q}_N = (q_0 - q)$ has been set, with N being the matrix projecting vectors into the null space of the Jacobian, that is, $N = (I - J^\dagger J)$. Finally, computed joint velocities are integrated to retrieve the new joint position and the corresponding new Cartesian pose that are used to command the remote robot, where S represents the skew-symmetric matrix operator, p_r is the new robot end-effector position, and R_r its orientation matrix. It is worth to note that, once joint positions are available, the end-effector pose can also be retrieved via forward kinematics computation.

5. Experiments and results

The proposed VR-based dual-mode teleoperation architecture has been developed in the BRILLO scenario (Fig. 4). The project's objective was to design a bimanual robotic system able to handle the typical bartending tasks [4]. Nevertheless, the main purpose of the experimental setup shown in Section 5.1 is to underline the potentials of the architecture described in Section 4. Therefore, the simulation was realized to allow the operator to move the arm in a desired pose using the dual-mode teleoperation.

This section discusses the selected software/hardware architecture, the simulated task, and the experiments conducted for BRILLO case study.

5.1. Experimental setup

The dual-mode teleoperation framework developed for the BRILLO project has been created in the following experimental setup:

- Operator side:

  - Visualization system: Unity 3D as a 3D simulator and a USB camera (Logitech USB C920 HD Pro webcam) as a 2D video feedback.
  - User tracking and interaction module: SteamVR and HTC Vive Pro Set.
- Robot side:

  - Robotic system: BRILLO setup includes two KUKA's Lbr iiwa 14 R820 series,[8] with Schunk EGL 90 PN[9] grippers mounted on the end-effector. Despite this, the two arms have been simulated, while only one physical robot has been employed for the tests.
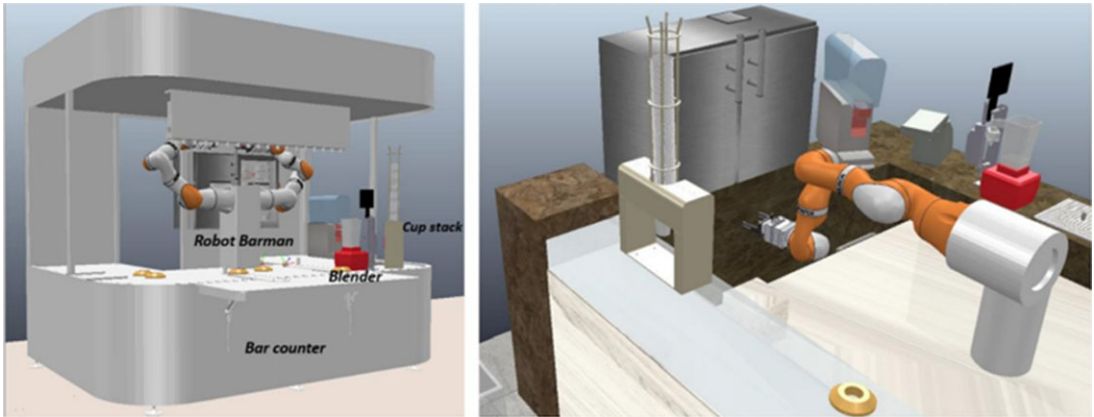

Figure 4. *3D representation of BRILLO scenario. As shown in ref. [4], it has been recreated in CoppeliaSim; the bartender robot consists of two KUKA Lbr 14 R820 and two Schunk EGL 90 PN grippers.*

- Control system: ROS[10] which is used to acquire information by the sensors and to control the FSM.
- Vision tracker: USB Camera, ArUco marker, pose estimation algorithm.
- Planner: MoveIt-integrated RRTstar planner.
- Data exchange system: RosBridge.[11]

The virtual scenario shown in Fig. 3 has been constructed using multiple methods. The BRILLO scenario, shown in Fig. 4, was initially modeled in CoppeliaSim as part of the work deeply described in [4], and it was successively imported into Unity. On the other hand, considering the DT, the meshes and the URDF file were downloaded from the GitHub repository[12] developed by the Autonomous Robotic Manipulation Lab. Therefore, using the Unity URDF importer, it was directly imported into the virtual scene. In the Unity environment, the robotic system's characteristics, such as gravity, inertia, and collision meshes, have been set. Lastly, the glass was designed and modeled during the current project in a CAD modeling software. The pose estimation algorithm has been developed to measure the relative pose between the ArUco markers and the camera. The markers are rigidly attached to the corresponding glass, in order to allow a faithful representation of the object in the scene. The pose estimation algorithm is based on the following reference frames, shown in Fig. 5:

- **Robot Reference Frame (RRF):** it is centered on the robot basis. The whole scene is reconstructed in the virtual environment using RRF as the main frame.
- **Camera Reference Frame (CRF):** it is placed at the camera's focal plane.
- **ArUco Reference Frame (ARF):** it is located at the center of the ArUco marker.
- **Glass Reference Frame (GRF):** it is centered on the glass basis, rigidly attached to the ARF.

In both scenarios, the CRF has a fixed relative pose with respect to the RRF system. A 3D-printed structure is considered as the rigid link between the ARF and the GRF. On the other hand, in the virtual scenario, the two reference frames are rigidly constrained. In the real scene, the ARF relative transform with respect to the CRF is tracked by the camera. The measurement is used in the virtual scenario to reconstruct the scene as reliable as possible. To avoid unnecessary complexity, in the 3D simulation, the camera and the ArUco marker do not appear in the scene.

---

[10]http://wiki.ros.org/

[11]http://wiki.ros.org/rosbridge

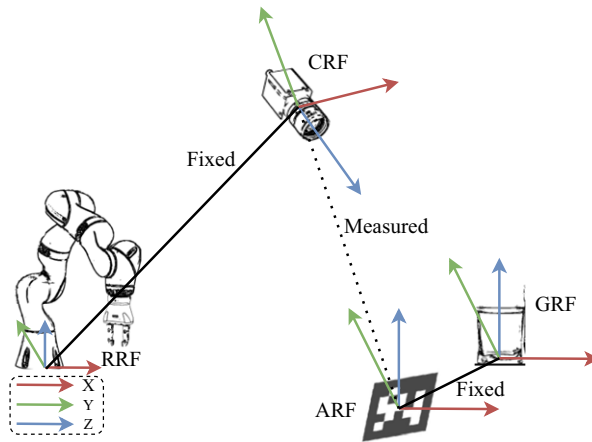[12]https://github.com/UM-ARM-Lab/kuka_iiwa_interface.git

Figure 5. *Reference frames in the vision tracker module. The markers' poses are defined with respect to the RRF. ARF is centered in the center of the ArUco marker, CRF in the focal plane of the camera, and GRF collocated at the center of the glass basis. The camera measures the relative pose of the ARF which is rigidly attached to the GRF.*

According to the architecture detailed in Section 3, Fig. 6 shows the developed communication framework.

- Unity

  – *Controller/velocity*: linear velocity of the tracked controllers.
  – *Controller/pose*: pose of the tracked controllers.
  – *Virtual/gripper/pose*: pose of the virtual gripper.
  – *Obstacle/pose*: pose of the simplified virtual objects in the scene.
  – *Obstacle/size*: size of the simplified virtual objects in the scene.
  – *Controller/left*: buttons input from the left controller (boolean type).
  – *Controller/right*: buttons input from the right controller (boolean type).
  – *Arm/number*: it refers to the number corresponding to the chosen arm to control.
  – *Scene/number/actual*: it refers to the number corresponding to the actual state to control the robotic system.
  – *Scene/number/desired*: it refers to the number corresponding to the user's desired state. To improve safety, the user can ask to change state, and if there is no other operation ongoing, it is possible to move to the next state.

- ROS

  – *Info/banner*: string which describes the actual state to the user.
  – *Joint/simulate/state*: joint state of the transparent robotic arm.
  – *Joint/real/state*: joint state of the opaque robotic arm.
  – *Scene/number/requested*: it refers to the number corresponding to the actual state. Once the user's request has been accepted, this data is updated enabling the new state.
  – *usb_cam/image_raw*: the 2D video feedback is compressed and shown in the local workstation.
  – *ArUco/simple_pose*: the result of the estimation pose algorithm.

Finally, to evaluate the delay between a user input and its realization, it is possible to consider three components:
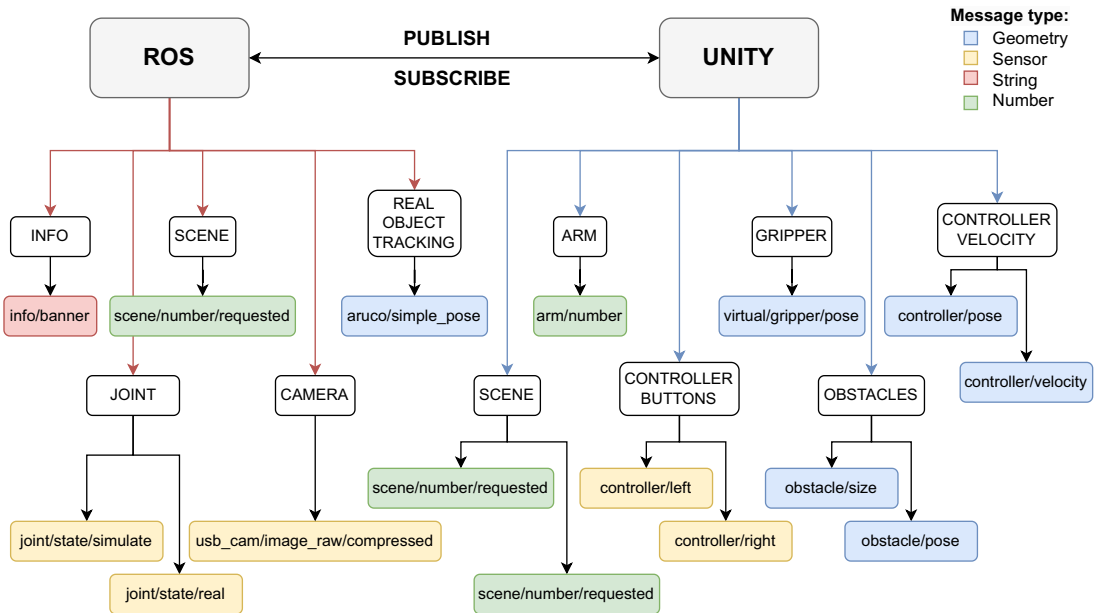
Figure 6. *Communication framework. ROS and Unity publishing and subscribing data into multiple topics. The topics are divided into message types (geometry, sensor, string, number) and organized according to the information they transmit. On the left, the topics written by ROS, and on the right, the ones published by Unity.*

1. Refresh rate: this time pertains to the refreshing of SteamVR inputs. It is a constant value (11 ms) determined by SteamVR.
2. Communication delay SteamVR-Unity: the time frame characterizes the communication delay between SteamVR and Unity, and it can oscillate. In a conservative way, it can be considered as 12 ms.
3. Update topic: this time interval corresponds to the delay to process and read the updated message, with measurements ranging between 10 and 23 ms.

Based on the earlier discussion, we determined a total delay of 46 ms between the local and remote sides. This delay is so minimal that it is imperceptible to the human senses, underlining that it has a negligible impact on the VR experience.

5.2. Task execution

The operator is immersed in the VR control room that reproduces the BRILLO scenario. The procedure for operating the robot using the two available modes is depicted in Fig. 3 and can be described as follows:

1. *Idle State*

    (a) The operator chooses the arm to control by pressing the corresponding virtual button that turns green.
    (b) The user selects the desired control mode by opening the radial menu attached to the VIVE controller.

2. *Approach State*

    (a) The operator grabs and drops the virtual gripper in the final position and orientation. This data is sent to the ROS system by a combination of pressed buttons.

    (b) Once the target pose and the obstacle's poses are received by ROS, the obstacle avoidance trajectory is planned.

    (c) The preview of the planned movement is shown to the operator within the virtual environment through the transparent arm, which reproduces the movement in loop. The operator can accept or abort the computed trajectory.

    (d) By a combination of pressed buttons, the user can activate the execution of the opaque robot's movement (DT) and, simultaneously, of the real robot.

3. *Telemanip State*

    (a) By pushing and keeping pushed a combination of buttons, the operator can directly control the robot's end-effector.

    (b) The operator linearly moves the controllers in the desired direction. The robot's end-effector follows it with a scaled linear velocity.

    (c) The robot's end-effector Cartesian orientation is controlled to keep aligned the end-effector and the gripper.

By utilizing both control modes within a single architecture, the operator was able to remotely maneuver the robotic arm and successfully complete a common bartending task, such as approaching a glass on a table.

5.3. Design of user interactions

To systematically review the usability of the system, in this section, we discuss the designed and implemented input system to enable users' task execution specified in Section 5.2. Taking inspiration from participatory design [34], such an interaction system has been designed, implemented, and tested by users in order to gather their opinions about proposed control and interaction modalities and exploit them for future improvements of our frameworks' usability. The following main actions have been implemented and then associated with a specific button of the controller (Fig. 7):

    (I) Trackpad button: navigate **(right)** within the immersive environment and manage menu **(left).**

    (II) Grip button: grasp, move, and release an object **(right or left).**

    (III) System button: consent **(right)**/abort **(left)** next step.

    (IV) Trigger buttons: play **(press both)**/pause **(release one)** the robot's movements.

The aim of the implemented locomotion system was to provide human operators the freedom to explore and investigate the system, especially at the beginning of the remote collaboration, in which they need to understand which is the problem and how to intervene. This action has been enabled through touch on the right VR controller's Trackpad button. By moving up, down, left, and right on the Trackpad (intuitively like arrow keys on PC keyboard), users can move virtually within the BRILLO scenario. On the other side, the left VR controller's Trackpad button is employed to enable access/close the main menu. Users can switch from one state to another by selecting a specific slice of the radial menu and clicking the Trackpad button. The VR controller's Grip button (on both left and right VR controllers) enables users to grab an object (as the digital gripper); while keeping it pressed, users can move the object whatever they want and release it by just releasing the Grip button. The latter has been specifically selected for this action since it is the one that mainly leads users to simulate realistic grabbing gestures by closing the fingers around the VR controller, rather than the other available buttons. Another fundamental
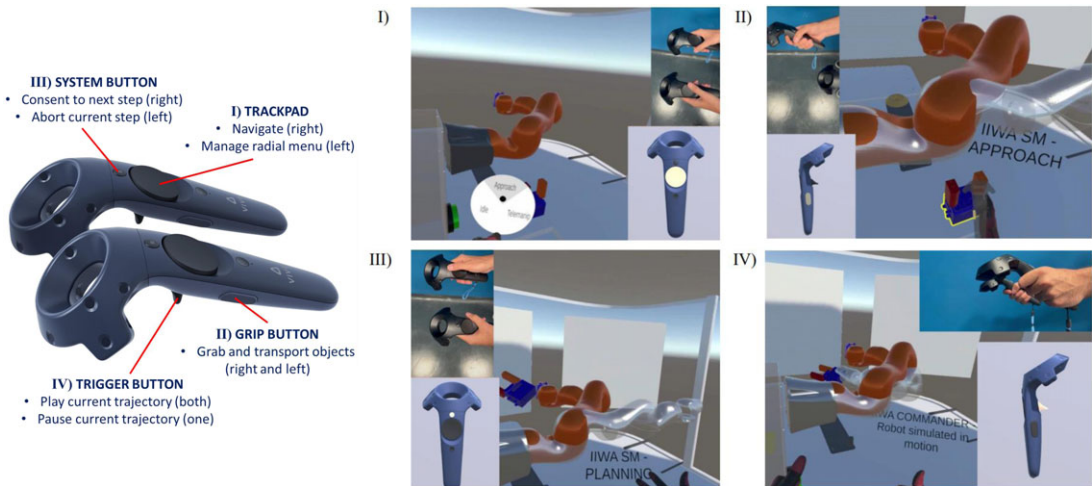
Figure 7. *Controller bindings: (I) Trackpad button, (II) Grip button, (III) System button, and (IV) Trigger button.*

action implemented in the proposed framework based on dual-mode control modality is the possibility to consent/abort the ongoing task. The depicted VR button to enable such actions is the System button, specifically selected as it is not immediately reachable by the user (compared with Trackpad and Trigger) but generally requires a wider hand movement to allow the thumb to reach it. The further movement and therefore a potential greater cognitive and physical effort for the user is actually wanted, as the user should use this button only after appropriate evaluation of the current situation. In the proposed setup, the right System button allows the transition to the next step, while the left one allows to abort the current task (stop any operation). Finally, both Trigger buttons have been selected to manage the ongoing task. In particular, by keeping pressed both Trigger buttons, users give consent to the preview of the planned trajectory (with transparent digital arm) or the execution of planned trajectory/velocity control movement (opaque arm). Whether at least one of the two Trigger buttons is released, the ongoing movement is immediately paused and can be continued only if both the Triggers are simultaneously and continuously pressed. This interaction logic has been designed in order to ensure a proper safety level, enabling users to pause immediately the ongoing task whether necessary. Trigger buttons, rather than the others on VR controllers, have been selected as they physically and cognitively recall the "consent buttons" (also called "dead man button") that are provided on smartpad/control pad with main industrial robotic manipulators.

5.4. Test

The proposed teleoperation system developed with the BRILLO case study has been tested by a sampling of 18 participants, ranging in age from 22 to 28 years old, homogeneous in gender (9 males and 9 females). They were all students belonging to the branch of industrial engineering, currently attending the master's degree. The user study aims to establish the usability of the dual-mode architecture. For the experimental phase, a group of 18 individuals was tasked with controlling the robotic system to reach the glass on the table. Initially, the robotic arm was positioned far from the glass. Each user guided it toward the glass, utilizing both the *Approach State* and the *Telemanip State*, as described in Section 5.2. Any additional actions were intentionally left open for potential implementation in future development. In order to properly analyze the results, an important distinction has been made between participants who had previous experience with VR for personal entertainment (12 people out of 18) and VR novices (6 people out of 18). Given the necessity for a combination of commands within the

Figure 8. *Experimental phases. The BRILLO case study can be divided into three main phases: the training phase, which involved showing a video to the participants to help them understand how to use the HRI; the test execution phase, where the system was tested one by one; and finally, the assessment surveys, during which participants completed their questionnaires.*

system, concerns arose regarding potential challenges for novices; therefore, the examination of the VR and non-VR groups aimed to discern differences in usability between seasoned and inexperienced users. As shown in Fig. 8, all participants, before trying the VR experience for the control of the simulated robotic system, participated in a brief training about the BRILLO project context, the aim of the test, and the operative procedures to be performed. The training was articulated in two phases:

1. Passive phase: the participants saw a video to understand how to control the two arms and the BRILLO's architecture.
2. Active phase: the participants had the opportunity to ask questions about what they had just seen and clarify their doubts.

Subsequently, the participant tested the teleoperation system and filled out the selected questionnaires. The conducted user study on our VR teleoperation framework has covered three main aspects:

- Usability: the capability in human functional terms to be used easily and effectively by the specific range of users, given specified training and user support, to fulfill the specified range of tasks, within the specified range of environmental scenarios [35].
- Workload: the volume of physical and cognitive work necessary for an individual to accomplish a task over time [36].
- Satisfaction: it is generally defined as fulfillment resulting from actual experiences relative to expected experiences [37].

Following the VR test, participants filled out three questionnaires: the System Usability Scale (SUS) [38] to evaluate the usability, the NASA Task Work Index (NASA TLX) [39] to measure the effort required by the user to complete the task, and, finally, the SAT to measure the satisfaction derived from the experience. The SUS consists of 10 questions, each with a score from 1 to 5 (1: strongly disagree,

Table I *SUS score classes.*

Rating:	Score:
Awful	0–51
Poor	51–68
Good	68–80.3
Excellent	80.3–100

Table II. *NASA score classes.*

Workload:	Value:
Low	0–10
Medium	10–30
Somewhat high	30–50
High	50–80
Very high	80–100

5: strongly agree). These questions investigate the user's attitude toward the product. The results are divided into the score classes shown in Table I.

The NASA TLX consists of a double evaluation: the first procedure examines the user's personal importance of the various subscales in task performance in order to be able to assign a weight to each, and the second asks the user to assign a value to the subclasses themselves. The results are divided into the score classes shown in Table II.

Furthermore, a satisfaction questionnaire (SAT) has been subjected to the participants to receive a first feedback about the developed immersive HRI. It is composed of 10 questions:

1. I think it is easy to learn how to use this system.
2. During the experience, I think it is easy to find the information you need.
3. I believe that all the information displayed during the experience is effective in helping the user achieve the goals.
4. The interface of this system is nice.
5. I believe that the system is sufficiently realistic (size of objects, colors, etc.).
6. I enjoyed using the two manual controllers as control mode on the system (to navigate, to control the robot, etc.).
7. Key combinations on controllers are easy to use.
8. I have never felt lost when using the system.
9. Overall, I am satisfied with this system.
10. I would like to use this system again.

For each question, participants could choose one of four responses, ranging from "totally disagree" (with a score of 0) to "totally agree" (with a score of 3), as shown in Table III.

Thus, the final score was a value ranging from a minimum of 0 to a maximum of 30.

5.5. Results and discussion

SUS score. The SUS questionnaire results are given in Fig. 9, showing the percentage of scores that falls into each class. The first diagram displays the overall results of the SUS, with no differentiation made between VR novices and experienced users. The majority of responses falls within the "Good"

<div align="center">

Table III. *SAT answer score.*

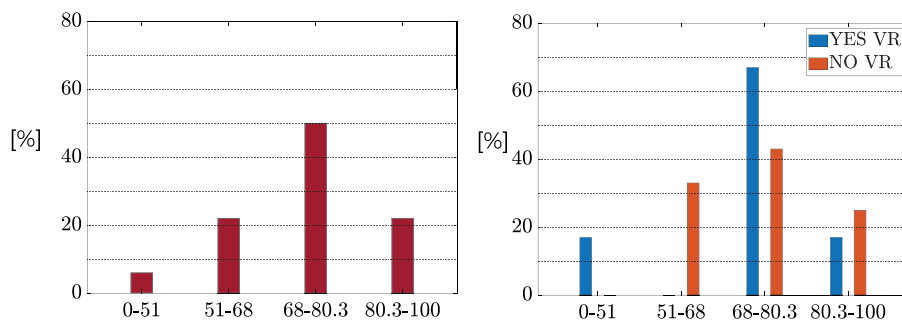| Possible answers:	Score:
Totally disagree	0
I don't mind	1
Partially agree	2
Totally agree	3

</div>

Figure 9. *SUS score. Two diagrams are used to illustrate the results, depicting the distribution of scores in both general (on the left) and with a breakdown into "YES VR" and "NO VR" categories (on the right). The x-axis indicates the score classes, while the y-axis shows the corresponding percentages.*
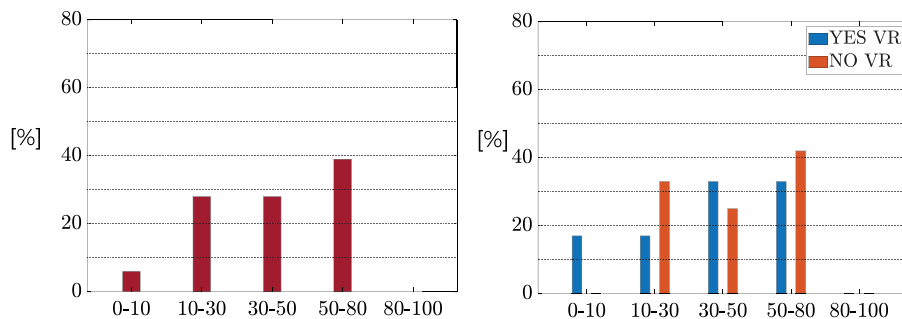
Figure 10. *NASA TLX score. The results are shown in two diagrams, which demonstrate the distribution of scores in both general (on the left) and with a specific categorization into "YES VR" and "NO VR" (on the right). The x-axis represents the score classes, while the y-axis displays the corresponding percentages.*

category, while only a small percentage fall within the "Awful" classification. The average score on the SUS questionnaire was 72/100, which falls in the SUS "Good" class. However, there were no significant differences in the averages of the "NO VR" and "YES VR" populations, and no substantial differences were observed between the two trends. The whole participants consider the VR teleoperation system usable.

NASA TLX score. About the users' workload, the NASA TLX results are given in Fig. 10, showing the percentage of scores that falls into each class. No respondents rated the workload as "Very high," with the majority falling into the "High" category. However, there were no significant differences between the two other top classes, "Somewhat High" and "Medium," and the highest one. The average score on the NASA TLX questionnaire was 39/100, which falls in the "Somewhat high" class. The averages for the two populations "NO VR" and "YES VR" did not differ significantly. While for the "YES VR" group, a regular trend is observed among the various classes, it can be seen that among VR novices, answers
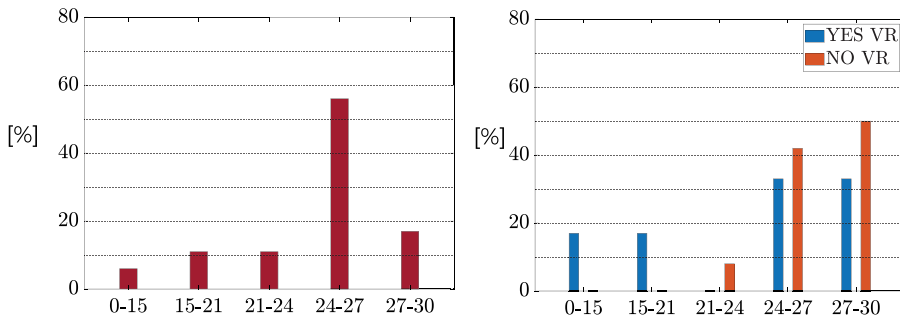
Figure 11. *SAT score. The results are presented through two diagrams, showcasing the distribution of scores in general (on the left), as well as with a distinct categorization into "YES VR" and "NO VR" (on the right). The x-axis denotes the score classes, while the y-axis exhibits the corresponding percentages.*
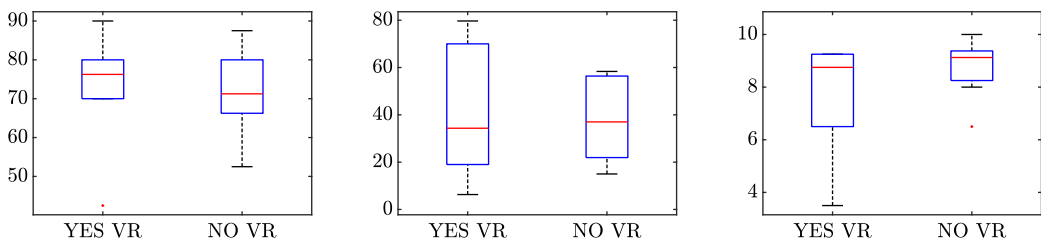

Figure 12. *Box plots providing a visual representation of the statistic study carried out to evaluate the significance of previous experience in VR (YES vs. NO) on the three metrics evaluated in this work: SUS score (left), NASA TLX (center), SAT score (right).*

tend to fall more into the "Medium" and "High" classes. Therefore, the system is considered to require a certain effort to interact with.

SAT score. The SAT results are given in Fig. 11, showing the percentage of scores that falls into each class.

The average score on the SAT questionnaire was 25/30. Regarding the distinct results, differences are noted for the averages of the two populations: 22.3 for the "YES VR" group and 26.25 for the "NO VR" group. Therefore, the group that had previously experienced VR was generally less satisfied with the experienced one.

To assess the significance of having previous experience in VR (YES vs. NO) on the three metrics evaluated in this work, we carried out a statistical study. Box plots displaying the median, quartiles, and any outliers are shown in Fig. 12 for the SUS score (left), NASA TLX (center), and SAT score (right). The analysis of ariance (ANOVA) returned the following p-values, respectively: $p_{SUS} = 0.920$, $p_{TLX} = 0.865$, $p_{SAT} = 0.141$, that are all well above $p = 0.05$ which is typically considered the upper bound to indicate statistical significance. In this case, we are not able to reject the null hypothesis (no difference in the means between the two groups). This result is encouraging since, in other words, it demonstrates the accessibility of our system also to VR novices. In fact, our system has been perceived by users as useful and satisfactory and with low physical/cognitive effort required to conduct VR-based teleoperation tasks.

6. Conclusions and future works

In this work, we proposed a dual-mode architecture to remotely control a robotic system in multiple scenarios with the use of VR technology. The architecture is based on an FSM that allows the operator to easily and rapidly switch between the states (and the respective control modes). In particular, in the *Approach State,* the operator can specify the end-effector target pose, preview the planned trajectory

for the robot, and confirm it (or not), while in the *Telemanip State*, a direct control on the robot's end-effector, with a scaled velocity interface, is provided. The proposed dual-mode VR-based teleoperation architecture has been designed with a human-centric approach, aiming to propose a system accessible to both VR experts and novices. For this reason, we have conducted an experimental campaign to evaluate human factors related to the use of our system. Specifically, we have applied our within a Unity-ROS teleoperation system for the BRILLO project [4]. The experiments have been conducted in MARTE Virtual Reality Laboratory of the University of Naples Federico II, with a sampling of 18 participants. The users tried both the control modes to remotely control one of the robotic arms to reach the required target (a glass fallen on the bartender). A user study in terms of usability, physical and mental workload, and satisfaction level has been conducted on the BRILLO teleoperation architecture, obtaining positive results. The participants consider the system usable and highly satisfying, even if it requires a considerable effort. The statistical study confirmed that our system is perceived as effective and usable for both VR experts/experienced users and VR novices, with no significant variance in the outcomes. This is a strongly encouraging result, demonstrating such a system's accessibility also for non-experienced XR users, with the view of a large-scale use.

These results show the potential of the proposed novel architecture. In line with the participatory design principle applied for the proposed architecture, we are already conducting deeper studies to improve the usability of the system about user interactions' logic; further to this, we aim to focus also on optimal data management and visualization within immersive HRI. With this regard, taking inspiration from refs. [40, 41] for interfaces' usability assessment, we plan to conduct further experiments to gather valuable insights from participants also on graphical aspects, as we did in this experimental campaign for users' interaction modalities. For instance, the first results of conducted experiments on control and interaction modalities have allowed us ideating the possibility for users to choose between a right-handed or a left-handed set of actions in order to make it more usable. Moreover, a possible improvement could be the introduction of the obstacles tracking also in the *Telemanip State* in order to ensure higher safety. Finally, a markerless objects tracking system could be introduced to make it usable in real-world scenarios. The design, implementation, and test of the optimized immersive HRI for teleoperation with discussed innovative features will be addressed in future works, which are currently in progress.

References

[1] G. Niemeyer, C. Preusche, S. Stramigioli, and D. Lee. *Telerobotics* (Springer, Berlin, Heidelberg, 2016) pp. 1085–1108.

[2] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff and P. R. Giordano, "Shared control : Balancing autonomy and human assistance with a group of Quadrotor UAVs," *IEEE Robot. Autom. Mag.* **19**(3), 57–68 (2012).

[3] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Trans. Robot.* **38**(1), 569–583 (2022).

[4] S. Buonocore, S. Grazioso and G. Di Gironimo, "Virtual Teleoperation Setup for a Bimanual Bartending Robot," **In:** *Extended Reality* (L. T. De Paolis, P. Arpaia and M. Sacco, eds.) (Springer Nature Switzerland, Cham, 2022) pp. 306–325.

[5] H. ElMaraghy, L. Monostori, G. Schuh and W. ElMaraghy, "Evolution and future of manufacturing systems," *CIRP Ann.* **70**(2), 635–658 (2021).

[6] A. Mazumder, M. Sahed, Z. Tasneem, P. Das, F. Badal, M. Ali, M. Ahamed, S. Abhi, S. Sarker, S. Das, M. Hasan, M. Islam and M. Islam, "Towards next generation digital twin in robotics: Trends, scopes, challenges, and future," *Heliyon* **9**(2), e13359 (2023).

[7] M. Grieves and J. Vickers. *Digital Twin: Mitigating Unpredictable, Undesirable Emergent Behavior in Complex Systems* (Springer International Publishing, Cham, 2017) pp. 85–113.

[8] F. De Pace, F. Manuri, A. Sanna and C. Fornaro, "A systematic review of augmented reality interfaces for collaborative industrial robots," *Comput. Ind. Eng.* **149**, 106806 (2020).

[9] S. Green, M. Billinghurst, X. Chen and J. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. J. Adv. Robot. Syst.* **5**(1), 1–18 (2008).

[10] R. Suzuki, A. Karim, T. Xia, H. Hedayati and N. Marquardt. "Augmented Reality and Robotics: A Survey and Taxonomy for AR-Enhanced Human-robot Interaction and Robotic Interfaces," **In:** *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, CHI '22* (Association for Computing Machinery, New York, NY, USA, 2022).

[11] D. Bambušek, Z. Materna, M. Kapinus, V. Beran and P. Smrž. "Combining Interactive Spatial Augmented Reality with Head-Mounted Display for End-user Collaborative Robot Programming," **In:** *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*(2019) pp. 1-8.

[12] M. Ostanin, S. Zaitsev, A. Sabirova and A. Klimchik, "Interactive industrial robot programming based on mixed reality and full hand tracking," *IFAC-PapersOnLine* **55**(10), 2791–2796 (2022). 10th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2022.

[13] E. Rosen, D. Whitney, E. Phillips, G. Chien, J. Tompkin, G. Konidaris and S. Tellex. "Communicating Robot Arm Motion Intent Through Mixed Reality Head-Mounted Displays," **In:** *Robotics Research. Springer Proceedings in Advanced Robotics* (N. Amato, G. Hager, S. Thomas, M. Torres-Torriti, eds.), (Springer, Cham, 2020) vol. 10, pp. 301–316.

[14] A. Naceri, D. Mazzanti, J. Bimbo, D. Prattichizzo, D. G. Caldwell, L. S. Mattos and N. Deshpande. "Towards a Virtual Reality Interface for Remote Robotic Teleoperation," **In:** *19th International Conference on Advanced Robotics (ICAR)* (Brazil, Universidade Federal de Minas Gerais – UFMG, 2019) pp. 284–289.

[15] F. D. Pace, G. Gorjup, H. Bai, A. Sanna, M. Liarokapis and M. Billinghurst. "Leveraging Enhanced Virtual Reality Methods and Environments for Efficient, Intuitive, and Immersive Teleoperation of Robots," **In:** *IEEE International Conference on Robotics and Automation (ICRA)* (2021) pp. 12967–12973.

[16] T. Kot and P. Novak, "Application of virtual reality in teleoperation of the military mobile robotic system taros," *Int. J. Adv. Robot. Syst.* **15**(1), 172988141775154 (2018).

[17] J. Du, H. Do and W. Sheng, "Human-robot collaborative control in a virtual-reality-based telepresence system," *Int. J. Soc. Robot.* **13**(6), 1–12 (2021).

[18] M. Ostanin, S. Mikhel, A. Evlampiev, V. Skvortsova and A. Klimchik. "Human-Robot Interaction for Robotic Manipulator Programming in Mixed Reality," **In:** *IEEE International Conference on Robotics and Automation (ICRA)*(2020) pp. 2805–2811.

[19] J. I. Lipton, A. J. Fay and D. Rus, "Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing," *IEEE Roboti. Automat. Lett*. **3**(1), 179–186 (2018).

[20] R. Hetrick, N. Amerson, B. Kim, E. Rosen, E. J. d. Visser and E. Phillips. "Comparing Virtual Reality Interfaces for the Teleoperation of Robots," **In:** *Systems and Information Engineering Design Symposium (SIEDS)*(2020) pp. 1–7.

[21] D. Sun, A. Kiselev, Q. Liao, T. Stoyanov and A. Loutfi, "A new mixed-reality-based teleoperation system for telepresence and maneuverability enhancement," *IEEE Trans. Hum.-Mach. Syst.* **50**(1), 55–67 (2020).

[22] D. Mourtzis, J. Angelopoulos and N. Panopoulos, "Closed-loop robotic arm manipulation based on mixed reality," *Appl. Sci*. **12**(6), 2972 (2022).

[23] A. Munawar and G. Fischer, "A surgical robot teleoperation framework for providing haptic feedback incorporating virtual environment-based guidance," *Front. Robot. AI* **3**, 47 (2016).

[24] A. Y. Mersha, X. Hou, R. Mahony, S. Stramigioli, P. Corke and R. Carloni. "Intercontinental Haptic Teleoperation of a Flying Vehicle: A Step Towards Real-time Applications," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems*(2013) pp. 4951–4957.

[25] R. Daniel and P. McAree, "Fundamental limits of performance for force reflecting teleoperation," *Int. J. Robot. Res*. **17**(8), 811–830 (1998).

[26] M. Dede and S. Tosunoglu, "Fault-tolerant teleoperation systems design," *Indus. Robot Int. J*. **33**(5), 365–372 (2006).

[27] M. Selvaggio, M. Cognetti, S. Nikolaidis, S. Ivaldi and B. Siciliano, "Autonomy in physical human-robot interaction: A brief survey," *IEEE Robot. Automat. Lett*. **6**(4), 7989–7996 (2021).

[28] C.-P. Kuan and K.-y. Young, "VR-based teleoperation for robot compliance control," *J. Intell. Robot. Syst*. **30**(4), 377–398 (2001).

[29] M. Laghi, L. Raiano, F. Amadio, F. Rollo, A. Zunino and A. Ajoudani, "A target-guided telemanipulation architecture for assisted grasping," *IEEE Robot. Automat. Lett*. **7**(4), 1–8 (2022).

[30] B. Omarali, B. Denoun, K. Althoefer, L. Jamone, M. Valle and I. Farkhatdinov. "Virtual Reality Based Telerobotics Framework with Depth Cameras," **In:** *29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*(2020) pp. 1217-1222.

[31] P. Ponomareva, D. Trinitatova, A. Fedoseev, I. Kalinov and D. Tsetserukou. "Grasplook: A VR-based telemanipulation system with R-CNN-driven augmentation of virtual environment," **In:** *20th International Conference on Advanced Robotics (ICAR)*(2021) pp. 166–171.

[32] R. Balogh and D. Obdržálek. "Using Finite State Machines in Introductory Robotics," **In:** *Robotics in Education. RiE 2018. Advances in Intelligent Systems and Computing* (W. Lepuschitz, M. Merdan, G. Koppensteiner, R. Balogh, D. Obdržálek, eds.), vol. 829 (Springer, Cham, 2019) pp. 85–91.

[33] M. Young, C. Miller, Y. Bi, W. Chen and B. D. Argall. "Formalized Task Characterization for Human-robot Autonomy Allocation," **In:** *International Conference on Robotics and Automation (ICRA)* (2019) pp. 6044–6050.

[34] J. Nielsen. *Usability Engineering* (Morgan Kaufmann Publishers Inc, San Francisco, CA, USA,1994).

[35] B. Shackel, "Usability – Context, framework, definition, design and evaluation," *Interact. Comput.* **21**(5-6), 339–346 (2009).

[36] L. Longo, "Subjective Usability, Mental Workload Assessments and Their Impact On Objective Human Performance," **In:** *Human-Computer Interaction - INTERACT* (R. Bernhaupt, G. Dalvi, A. Joshi, D. K. Balkrishan, J. O'Neill and M. Winckler, eds.) (Springer International Publishing, Cham, 2017) pp. 202–223.

[37] D. Lee, "Delivering satisfaction and service quality: A customer-based approach for libraries (review)," *Port. Libr. Acad.* **1**(4), 545–546 (2001).

[38] J. Brooke, "Sus: A retrospective," *J. Usab. Stud.* **8**, 29–40 (2013).

[39] S. G. Hart, "Nasa-Task Load Index (NASA-TLX); 20 years later," *Proc. Hum. Factors Ergonom. So. Ann. Meet.* **50**(9), 904–908 (2006).

[40] G. D. Gironimo, G. Matrone, A. Tarallo, M. G. Trotta and A. Lanzotti, "A virtual reality approach for usability assessment: Case study on a wheelchair-mounted robot manipulator," *Eng. Comput.* **29**(3), 359–373 (2013).

[41] S. Patalano, A. Lanzotti, D. M. D. Giudice, F. Vitolo and S. Gerbino, "On the usability assessment of the graphical user interface related to a digital pattern software tool," *Int. J. Interact. Des. Manufact.* **11**(3), 457–469 (2017).

[42] S. A. Bowyer, B. L. Davies and F. R. Y. Baena, "Active constraints/virtual fixtures: A survey," *IEEE Trans. Robot.* **30**(1), 138–157 (2014).