IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 30, 2019, accepted August 28, 2019, date of publication September 4, 2019, date of current version September 18, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2939493

Multimodal Human-Robot Interface for Accessible Remote Robotic Interventions in Hazardous Environments

GIACOMO LUNGHI[1,2], RAUL MARIN[2], MARIO DI CASTRO[1], ALESSANDRO MASI[1], AND PEDRO J. SANZ[2]

[1]CERN, Survey, Measurement and Mechatronics (EN-SMM) Group, 1217 Geneva, Switzerland
[2]Interactive Robotic Systems Laboratory, Jaume I University of Castellón, 12071 Castellón de la Plana, Spain

Corresponding author: Giacomo Lunghi (giacomo.lunghi@cern.ch)

ABSTRACT Human-Robot Interfaces have a key role in the design of secure and efficient robotic systems. Great effort has been put during the past decades on the design of advanced interfaces for domestic and industrial robots. However, robots for intervention in unplanned and hazardous scenarios still need further research, especially when the mission requires the use of multiple robotic systems, to obtain an acceptable level of usability and safety. This paper describes the design and the software engineering process behind the development of a modular and multimodal Human-Robot Interface for intervention with a cooperative team of robots, as well as its validation and commissioning, as it is being used in real operations at CERN's accelerators complex. The proposed Human-Robot Interface allows the control of a heterogeneous set of robots homogeneously, providing the operator, among other features, with live scripting functionalities which can be programmed and adapted in run-time, for example, to increase operator's multi-tasking in a multi-agent scenario. The operator is given the capability to enter in the control loop between the HRI and the robot and customize the control commands according to the operation. To provide such functionalities, well-defined software development approaches have been adopted, for guaranteeing the modularity and the safety of the system during its continuous development. The paper describes the modules offered by the HRI, such as the multimodality, multi-robot control, safety, operators training, and communications architecture, among others. The HRI and the CERN Robotic Framework where it belongs are designed in a modular manner, in order to be able to adapt both, software and hardware architecture in a short time, to the next planned mission. Results present the experience gained with the system, demonstrating a high level of usability, learnability and safety when operated by both, non-experts and qualified robotic operators. The multimodal user interface has demonstrated to be very accurate and secure, providing a unique system to control, in a teleoperated or supervised manner, both single and multiple heterogeneous mobile manipulators. At the moment of writing, the user interface has been successfully used in 100 real interventions in radioactive industrial environments. The presented HRI is a novel research contribution in terms of multimodality, adaptability and modularity for mobile manipulator robotic teams in radioactive environments, especially for its software architecture, as part of the CERN Robotic Framework.

INDEX TERMS Human robot interaction, mobile robots, teleoperators, telerobotics, multiagent systems, software engineering.

I. INTRODUCTION

Mobile manipulators are becoming increasingly common in a wide range of fields, such as domestic robot for human assistance, warehouses management, search and rescue [1], bombs disposal [2] and inspection and maintenance of industrial facilities. In particular, the use of robotic systems in hazardous scenarios provides increased safety to humans, as well as enhanced capabilities to collect useful data from the environment [3].

Remotely controlled robots can be equipped with different hardware according to their purpose: inspection robots are

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Chen.

usually provided with coloured and thermal cameras, and hazardous signal detectors (e.g. radiation monitors, oxygen sensors, temperature sensors) for giving the most complete overview about the environment to the operator; bomb disposal robots, instead, are usually equipped with one or more robotic manipulators for interacting with the hazardous material. Robots, sensors, specialized tools and user interfaces can highly reduce the operation time if properly adapted to the intervention, increase the safety and help the operator to achieve the goal more efficiently.

In the latest years, industries showed a new interest in mobile robots for inspection and maintenance. Some companies present hazardous facilities, and the use of robots would increase not only the safety of the personnel avoiding human access to their plants but also their availability, thanks to reduced reaction time if robots are already available in place. Among these companies, those dedicated to nuclear power production such as Fukushima [4] and ITER [5], but also centres devoted to nuclear research [6], such as CERN, the European Center for Nuclear Research, XFEL [7], the European X-ray free-electron laser, or FERMILAB, present obvious hazardous plants, due to their intrinsic high-radiation level. More in details, CERN has more than 50 km of underground particle accelerators [8], which come not only with high radiation levels, but with the common risks of underground working sites as well, such as oxygen deficiency, limited access, fire risks, and floods.

CERN hosts the biggest particle accelerator complex in the world, containing an enormous variety of industrial and scientific equipment. Performing maintenance on this variety results in a wide list of different tasks, from simple visual inspection to complex interaction with the equipment, such as screwing, sawing, welding, dismantling and assembling, among others. In order to comply with this heterogeneous set of equipment, a heterogeneous team of robots is required, to have the proper solution always available. Nevertheless, this requirement for a variety of robots, sensors and tools, cannot be satisfied with solutions available on the market, since most of the industrial and military robots for interventions [9] can be considered as a black box, each one different from the other, and difficult to extend or adapt to the particular intervention constraints.

In the process of development of mobile robotic solutions for such industrial facilities, two pools can be identified: (1) the pool of robots, sensors and tools, and (2) the pool of operators. In particular, the pool of operators includes all the people responsible for the usage of the robots during an intervention. It is important, therefore, that the pool of robots, sensors and tools inside a team follows the same conventions and shares common features, in order to have, ideally, a pool of operators who can use the entire pool of robots. As previously stated, this is not possible nowadays with current industrial solutions, which come with their Human-Robot Interface (HRI), set of sensors and features.

Furthermore, in the ideal condition of having a pool of industrial robots with the same conventions and HRI, a fully devoted team to control the robots would be required, due to the low usability of the current available platforms. This team would need constant training in a devoted area, resulting in an overall cost increase of the robotics activities. Above all, creating a specialized team for teleoperation would result in a group of extremely skilled people who would have a deep knowledge of the robotic systems, but not of the environment and the equipment that requires the intervention. Taking this into account, it has been considered as a solution to design the whole robots' pool (including procedures, protocols and Human-Robot Interface) in order to allow the equipment's experts, and not only the skilled robot operators, to teleoperate the robots during the intervention.

In this paper, the problem of creating an advanced and unified multimodal Human-Robot Interface (HRI) is addressed. The proposed interface is able to control the entire robots' pool (see figure 1), including all the sensors and the tools in a uniform way, with a particular focus on its usability and learnability. Software development methodology has been used in order to allow the adaptability of the interface to new requirements, coming from new intervention requests or hardware changes. Furthermore, the HRI has been designed in order to provide a complete environment for robotic interventions, including preparation, training, operation and post-data analysis. The HRI should also include advanced features for increasing the intervention safety, the telepresence of the operator and the most complete knowledge about the environment.

A. CERNTAURO PROJECT

The HRI presented in this manuscript belongs to the CERN Robotic Framework, as conceived in the CERNTAURO project [10]. The aim of the project is to create a set of different highly usable robots for inspection, reconnaissance and teleoperation for harsh environments and more specifically for the CERN accelerator complex [6]. The project covers a wide series of technologies, from custom robots development to robot control, network communication, safety, Human-Robot Interaction, Virtual Reality, and Artificial Intelligence, among others.

The CERNTAURO project's goal is to create a complete framework which is modular in order to be adaptable to the specific intervention necessities, and to be upgraded accordingly, as new features need to be integrated.

Further details about the robot's development, in terms of mechanics and control, and their requirements for operation in radioactive environments, which are out of the scope of this paper, can be found in [10].

II. STATE OF THE ART

Building a robotic system flexible, distributed and interoperable requires a systematic approach towards software development, which does not seem to be the norm in the automation application development process [11]. Most of the applied robotic products seem to use device centred paradigms making late changing in the requirements

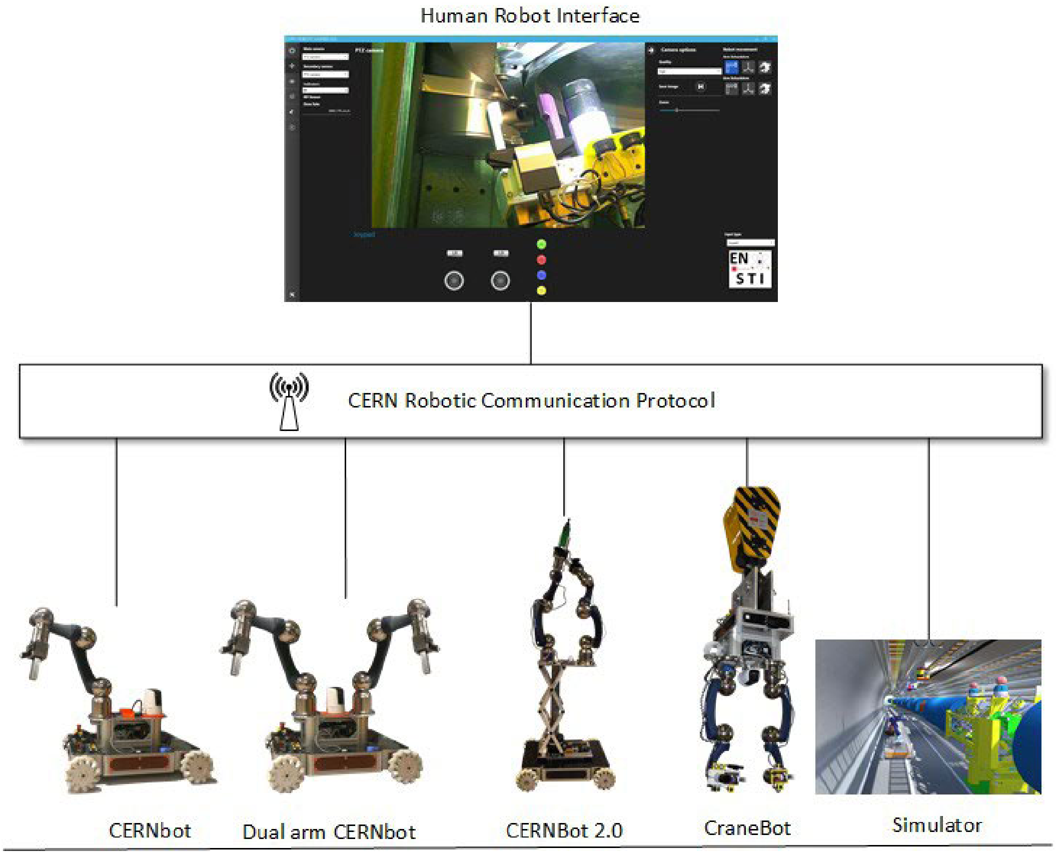


FIGURE 1. Unified multimodal Human-Robot Interface to interact with the heterogeneous team of robots.

inefficient due to the strict bound that has been established with the hardware.

In the latest years, a lot of effort has been put in the design of Human-Robot Interaction mechanisms [12]. However, only a few of them focused on the creation of a complete environment for Human-Robot Interaction, designed in order to broaden the availability of intervention robots to different operator skills [13]. More advanced interaction enables robots to be controlled with fluidity and less tedious operations [14]. Furthermore, standard Human-Computer Interface design techniques, or software engineering development methods, have rarely been applied to Human-Robot Interfaces development for teleoperated robots, as the development of such robots involves people from varying engineering disciplines, many of whom do not have a software engineering background; hence some important software engineering principles are often missed out. In this section, an overview of the state of the art in Human-Robot Interface development is presented. Afterwards, a brief overview of existing robotic solutions currently used in radioactive environments is shown, highlighting the limitations of those systems for CERN's current needs.

A. HUMAN-ROBOT INTERACTION DESIGN OVERVIEW

In the last decades, a lot of effort has been put in the definition of generic rules and guidelines for the development of Human-Robot Interfaces. In the following, some of the most important works are presented.

In [15] three modes of interaction with robots are defined: supervisor, operator, and peer. In particular, in the operator mode, the user is responsible for continuous interaction with the robot by assigning way-points, by tele-operating it or even by reprogramming it on the fly. However, in this work, users interacting in *operator mode* are considered expert users. The bystander user in peer mode, instead, would have no formal training with the robot but requires a coexistence between the robot and the operator in the same environment in order to allow him or her to create a mental model of the robot's behaviour. In the matter of this work, these two roles are not appropriate and cannot be considered as separated, as one of the main requirements is the possibility to provide Human-Robot Interaction modalities to non-expert operators for controlling remote robots, without having, therefore, coexistence in the same environment.

Other studies focused on the development of effective Human-Robot Interaction modalities in service robots applications such as health-care service robots or robots for home environments. In such devices, it has been noticed how the perception of the service quality is highly affected by human emotional experience during the interaction with the robot [16]. The emotions studied for Human-Robot Interaction in service robots, though, are different from those involved in teleoperation scenarios, where the interaction is continuous for several hours and the robot is used rather like a tool than a partner. Furthermore, in service robotics, human interaction is directly addressed to the robot, while in the matter of this work the interaction of the operator with the interface itself can not be neglected, due to the distance between the robot and the operator (in principle several kilometres).

Different frameworks and design principles have been developed for modelling human-robot collaboration in robotic interventions, mainly in the scope of CBRNE (Chemical, Biological, Radiological, Nuclear, Explosives) domain. Among these, the most suitable for the matter of this work appears to be the *Shared Roles Model* [17], which is a compromise between the *Taskable Agent Model* and the *Remote Tool Model* for describing human-robot teaming. In the case of the *Taskable Agent Model*, full autonomy of the robot is the goal of the system, with teleoperation being temporary in nature, if necessary at all. On the opposite end of the human-robot model spectrum is the *Remote Tool Model*, where the robotic system is used exclusively as a tool during a continuous manual teleoperation process. The *Shared Roles Model* is a hybrid approach that assumes robot semi-autonomy with improved human connectivity for communication. However, a HRI that aims to provide a complete environment for robotic interventions to different users should implement all these interaction models, allowing the operator to choose according to the situation.

In [18], four general guidelines for the development of HRI in teleoperation scenarios were defined:

1) *Enhance awareness*: Provide a map of where the robot has been. Provide more spatial information about the robot in the environment to make operators more aware of their robots' immediate surroundings.
2) *Lower cognitive load*: Provide fused sensor information to avoid making the user fuse the data mentally.
3) *Increase efficiency*: Provide user interfaces that support multiple robots in a single window, if possible. In general, minimize the use of multiple windows.
4) *Provide help in choosing robot modality*: Provide the operator assistance in determining the most appropriate level of robotic autonomy at any given time.

However, these guidelines focus more on the type of information to show to the operator, rather than the modality of displaying such information. Moreover, the adaptation of these guidelines to multi-agent intervention is not straightforward. Providing a map of where the robots are can help the operator, but it is not clear how to provide fused sensor information about multi-agent scenarios.

In [19] the results on studies of HRIs for multi-tasked robotic operations are presented. The issues addressed are similar to the ones addressed by this work: the operators are not robotic specialists but task specialists; their primary role is to accomplish the task and not to control the robot; future tasks will be complex and are difficult to predict. With these issues stated, various studies on the operator's multi-tasking have been made. The main outcome is straightforward: multi-tasking reduces the operator's performance independently on his or her skills but depends on the operator's priorities, task difficulties, interference effects and amount of training. Context switching between tasks causes performance degradation according to the number of tasks and their difficulties [20]. However, the operator's limitations on multiple tasks can be mitigated by the use of different modalities [21]. Other studies confirm the benefits of the use of different modalities, highlighting also their limitations, mainly related to operator expectations for the particular feedback [22]. The issues defined in [19] are considered as the same issues that this work is trying to solve.

One of the main open issues when dealing with the development of Human-Robot Interfaces is the evaluation of their performance. In [23] it is proposed to measure efficiency, effectiveness and user satisfaction, as in standard Human-Computer interfaces evaluation. Furthermore, it defines four metrics for the evaluation of HRIs when used by non-expert operators: predictability of the behaviour, capability awareness, interaction awareness and user satisfaction.

In [24] seven principles to evaluate the efficiency of a Human-Robot Interface are proposed. Among these principles, there is also a metric for the evaluation of multi-robot single-operator interface, which can be useful for the matter of this work. In [25], several principles for situation awareness driven design are presented, identifying a series of factors for evaluating the performance of a HRI. Providing to the operator a high level of situation awareness to the operator should be a primary requirement during the development of a HRI.

In conclusion, standard Human-Computer Interaction methods are not universally approved, as well as common Human-Robot Interaction guidelines, but several approaches have been proposed in the last 30 years, both in terms of interaction, displaying of feedback and evaluation. Several studies have been also made in robot teams, in which, however, the control of a single robot is shared by multiple humans. Multi-robot collaboration and control from a single operator do not appear to have received much attention in the research field, and a solution to the problem is proposed in this paper.

B. EXISTING ROBOTIC SYSTEMS FOR RADIOACTIVE ENVIRONMENTS

A lot of custom robotic systems have been developed for inspection and maintenance in radioactive research centres and nuclear reactors.

ITER (International Thermonuclear Experimental Reactor), for example, is developing several robotic solutions for the maintenance of its reactor and facilities. These robotic solutions, though, are not designed to be adapted to an existing environment, but the facility itself is being developed to be *robot-friendly*. Between the robotic systems of ITER [26] can be found a *blanket remote handling system* [27], a *divertor* *remote handling system* [28], a *cask and plug remote handling system* [29], an *in vessel viewing system* [30], a *neutral beam remote handling system* [31] and a *remote handling for hot cell* [5]. This systems have to work in areas with high radioactivity, as the *blanket remote handling system* which it will be operated in a high radiation environment (250 Gy/h max.) and must stably handle the blanket modules, which weigh 4.5t and are more than 1.5m in length, with a high degree of position and posture accuracy [32].

The Joint European Torus (JET) is the flagship for European Fusion Research. This study seeks to address the issue of the need for remote handling in the process of recreating nuclear fusion as a limitless source of clean energy. The JET remote handling system employs a *man in the loop* approach with the robotic Boom and Master-Slave Servo-Manipulator (MSM) system providing the operator with a pair of *remote hands* inside the JET Torus. All remote handling tooling and components are designed to be handled by the *remote hands* like an operator actually working within the JET Torus. Such an approach requires a high degree of operator training, together with systematic methodologies for remote operations task development, tooling and component validation [33], making it not suitable to CERN's environment. Furthermore, interaction with old, not robot-friendly equipment is still an open issue with the proposed robotic systems.

The Spallation Neutron Source (SNS), a new Department of Energy facility located at ORNL, is the latest step in the development of accelerator-based neutron research facilities. One thousand beam pulses will be bunched in a ring and then directed to a flowing liquid mercury target that will convert the protons to a pulse of approximately 5 x1015 neutrons. The neutrons will then be slowed to useful energies and guided to 24 instruments where scientists from around the world will have the opportunity to undertake basic research in materials science. A remote handling robotic system, based on Telerob EMSM-2B and a mobile vehicle equipped with a servo-manipulator (Figure 2) has been developed for this facility.

FIGURE 2. The SNS robot control room. It can be noticed the complexity of the control room equipment [34].

Mobile robots for interventions have been developed and studied to manage nuclear disasters as well. First studies were already made during the Chernobyl accident of 1986, which raised the acuteness of the problem of designing mobile robot-based systems capable for a long time to carry out operations in high-level radiation areas [35]. The Fukushima Daiichi nuclear reactor disaster, triggered the tsunami that devastated parts of Japan in March 2011, serves as a reality check on our capacity to effectively use robots for hazardous tasks. Beyond the carefully engineered environments that characterize the nuclear material handling industry, the disappointing performance of several robots [36] has demonstrated just how far we still have to go [37]. The disaster exposed the need for prior planning and continuous training, rehearsal and cooperation between research and development agencies, defence disaster relief agencies, robotic systems manufacturers and engineers at the hazardous facilities where the robots would be used. Continuous updating of robot systems is needed: some robots that could have been deployed relied on obsolete electronic and computer components that could not be replaced.

One of the most complex forms of hazardous operations is the deactivation and decommissioning (D&D) of defunct facilities where nuclear radiation or toxicity hazards preclude human presence. D&D can be thought of as remote demolition for the most part. Some operations are crude such as knocking down building structures and debris removal. Other operations may involve careful disassembly of equipment and devices, size reduction and packaging of handling/storage. These operations are essentially the inverse of remote maintenance and require the dexterous use of tools and handling of objects [37]. Several robotic systems have been developed for these D&D tasks in nuclear applications [38] and for inspection and maintenance [39].

In nuclear fuel reprocessing facilities, remote handling equipment have to deal with severe constraints:

- Maximum integrated dose for 1 year between 104 and 105 Gy.
- Decontamination capability using potentially aggressive products.
- Operation at high temperature (508°C).
- Electromagnetic compatibility.
- Volume and weight compatible with remote operations.
- Compliance with safety and quality standards.
- Diagnosis and maintainability constraints.
- Compatibility with waste management.

For this task, industrial robots can be adapted to the constraints listed here above. That includes the development of radiation tolerant electronic parts, force-feedback control software for master-slave teleoperation and the integration of the overall technologies in order to build a remote handling system [40].

III. DEFINITIONS AND REQUIREMENTS

Considering the statements made so far, in order to create a highly usable Human-Robot Interface for robotic interventions, it is important to place the operator in the centre of the development process. Furthermore, for guaranteeing the modularity and the adaptability of the interface to different robots, sensors and tools, as well as to allow the various modalities of interactions to the operator, the robot itself should not take a focusing role in the development, and the HRI should be as less as correlated to the hardware as possible. For this reason, in this section, only a list of software requirements regarding the interaction between the operator and the HRI is defined and will be considered the only non-changing requirements for future developments and improvements.

During the design of a Human-Robot Interface, it is important to take into account some aspects that are related to the operator and the type of interaction that he or she has with the interface. Such aspects give additional guidelines for the design, both in terms of graphical and behavioural choices and in terms of functionalities. Therefore, in the matter of this work, all the definitions and concepts defined in this section are taken into account afterwards for the implementation of the interface.

A. SOFTWARE DEVELOPMENT PATTERNS

In general, during the requirements definition, operational aspects of the telerobotic operation to be carried out are gathered. Standard software development for Human-Computer Interfaces and automated system follow traditional waterfall model or V-model [41], in which the requirements are defined at the beginning of the development, according to hardware requirements and operational requirements [11]. However, as already stated, such models place in the centre of the development the robotic platform, designed and built for the specific intervention, and the intervention itself. These methods are suitable for interdisciplinary development, as a continuous dialogue between the various specialists is done at the beginning of the process, but do not cope well with requirements changes during the next phases and do not adapt well to different interventions, users and robots.

B. OPERATORS DEFINITION

One of the first steps in the requirements specification for a unified interface is the definition of the users and the context of use. As previously stated, the need of enlarging the operator's pool is of primary interest in CERN Robotics Operations strategy, to take profit from the expertise of skilled technicians that know in detail the scientific and engineering environment.

Normally, three main categories of operators can be defined [42]:

- Expert operators, who have a long experience in telerobotics, followed training courses and already performed several interventions.
- Project involved operators, who did not acquire experience in the teleoperation field, but they were involved in the project's development, acquiring, therefore, a detailed representation of the robotic system.
- Inexperienced operators, who are experts on the equipment that requires the intervention, and never could use the telerobotic system before. This category is the largest and most interesting one since it contains the experts that know the necessity of the equipment where the intervention is to be performed.

The HRI needs to be designed in order to provide clear feedback to an inexperienced operator, but enough detailed information to an expert one, giving the possibility to configure the HRI according to the level and the needs.

C. HUMAN ERROR

A Human-Robot Interface for Telerobotic applications, due to its criticality, should take human error into account as much as possible. Historically, there has been a lack of scientific research on human error, which focused instead on the correct behaviour of a human. Research on this topic started only in the early 80s, for example with studies on human behaviour in nuclear power industry [43] and in air traffic control [44].

Defining human error is not straightforward and several questions about its definition have been addressed [45]. In the context of this work, it is important to be able to separate accidents caused by operator's misbehaviour and errors due to a lack of understanding of the operator with the HRI information. It is well known that an interface should be more transparent to the actual working system, reducing through proper representation the complexity of the data coming from it. The operator must be able to see through the displays (or virtual/augmented reality headsets) and perceive what is going on. In the aviation sector, this is often called situational awareness, while in the robotic field this concept can be associated with telepresence.

While most of the studies on this matter are focused on avoiding human errors when following a written procedure, it is not easy to estimate possible errors when dealing with telerobotic scenarios, which are often unknown and unpredictable.

In the context of remote robotic interventions, human error can be seen as a lack of feedback in the closed-loop control system, which comprises the operator and the robot. If the appropriate feedback is lacking, because of sensory limitations of the robot or misrepresentation of the feedback by the interface, the operator will tend to explore (and possibly lead to an error) the state space of this closed-loop control system in order to find satisfactory feedback [14]. Since the human operator often anticipates or previews the upcoming tasks, the operator could share these thoughts with the interface (e.g. indicating the target of a grasping procedure) which could adapt itself to help the operator in the task by assisted control or by simply providing the most appropriate feedback.

In the more generic topic of Human-Computer Interaction, other well-known forms of errors are lapses and slips. Slips distinguish from lapses by the source of the failure: a slip is a failure during the execution of a procedure, while a lapse is a failure of the operator's working memory.

In semi-autonomous control, a frequent cause of slip error is a capture. Let us suppose that two procedures, A and B, are available and the operator is well trained in the execution of procedure A. If A and B share some tasks, it is possible that while executing procedure B, the operator will drift towards procedure A, leading to an inconsistent one. Captures, and slips more in general, can be reduced if procedures are as much as automatic as possible and require the minimum possible interaction from the operator. This gives a clue on the level of autonomy that should be implemented in the HRI, consistent with the need for teleoperation necessary for carrying out generic interventions.

D. MENTAL WORKLOAD, STRESS AND ATTENTION

Mental workload of operators in human-machine systems has been a matter of research since decades [46]. The final concern of this research is not the mental workload per se, but how mental workload affects the operator's attention and performance, which ultimately affects the overall system performance.

The attention is the cognitive process of selectively concentrating on one aspect of the environment while ignoring other things. During a telerobotic intervention, the operator's attention should be focused constantly on the task. Therefore, the interface must avoid interfering with the operation, for example by displaying pop-ups and messages, which needs to be handled for continuing. The attention of the operator can be monitored by the interface using eye-tracking methods as well, similar to the driver attention detection available in most modern cars.

Mental workload and stress affect the operator's working memory during the operation. According to [47], an actor uses two types of memory, the short-term memory, faster and more resource demanding, and the long-term memory, slower but of higher capacity. While interacting with an interface, an operator makes extensive use of the short-term memory which, therefore, must not be overloaded by the interface: in fact, the short-term memory can store up to 7-9 conceptual elements. Besides, this number drastically reduces in presence of stress. In the short-term memory, the operator must store the status of the environment in which the operation is performed and the status of the robot: since the complexity of the status of the environment is not controllable, the interface should optimize the representation of the status of the robot. For example, there is no need to constantly display the battery level of the robot, which can be shown through a message (or additionally a sound in case of critical levels) only when passing certain thresholds. Another possibility can be letting the operator select the most effective feedback cameras, for giving them more resolution and quality, while being able to see additional cameras (if necessary) in lower detail, for avoiding mental workload. The robot itself can provide vision techniques to identify important situations to be supervised by the user at a particular camera, according to the mission plan.

E. METAPHORS

Metaphors are a central topic in the Human-Computer Interaction field. According to [48], metaphors are central to how humans think. This statement has a significant impact on software engineering and HCI too. Without entering in the details of the metaphors field, it is obvious that an interface for HRI in telerobotic scenarios should have clear and precise metaphors and that the affordance of each component of the interface is clearly perceived by the operator.

However, an additional aspect of this topic that deserves a more accurate analysis is the what-so-called sociocultural embodiment [49]. This term has a wider definition in psychology and sociology, and it defines the relationship that the operator establishes with the interface and with the robot, which is not neutral but tends to be humanized. This means that the information that the interface provides to the operator must be efficient and perceived in a constructive and not invasive way, which could lead instead to some type of mistrust and pejorative attitude with the interface and the robot. Pejorative embodiment could have dramatic consequences in safety-critical interfaces, such as in the case of the air accident at the Madrid Barajas airport of 1983 [50].

Sociocultural embodiment is also a central topic from a different point of view. In the latest years new forms of interaction such as body tracking using RGB-D tracking, gamepads, haptic devices and motion tracking devices have been developed: this enlarged the concept of playing to a broader network of actions, which could lead to misbehaviour while performing safety-critical operations. Furthermore, according to [51], previous forms of interaction technologies produced an experience of simulation, whereas now the emphasis moved towards imitation, modifying the perception that the operator has of the robot, even more humanized if possible.

Using proper and effective metaphors increases exponentially the efficiency of the interface, provided that the previously highlighted drawbacks are taken into account.

F. TOWARDS SYSTEM RELIABILITY

Acquiring data in order to make a statistical analysis of the human errors that could appear while interacting with an interface is not a straightforward process. The credibility of human-error data from simulations is questioned on the basis that realistic stress, boredom, unexpected lack of concentration, and other behaviours are missing during a simulation. The same can be said about data collection during real operation, which could increase the mental workload of the operator, who will feel under evaluation during the process. Historically, an important potential source of human-error data is the training phase of the operators: during the training, the task to be performed is perfectly defined, and human-error detection can be programmed in the interface without interfering with the operator's actions. That being said, some techniques were defined to avoid human errors [14]. Above them, the following ones deserve a highlight in this context:

- The system should be fail-safe or at least fail soft, and able to avoid and recover from human errors.
- The system should provide immediate and clear feedback from the control loop (e.g. visual feedback using cameras should not be delayed).
- The system should restrict, where possible, the acting possibilities of the operator that could be misleading and not correctly perceived (e.g. not every interaction device is suitable for every action).
- the system should warn and alarm the operator when necessary, keeping in mind that too many warnings or alarms overload or distract the operator or condition him/her to ignore them.

This provides a list of software requirements which are always taken into account during the interface development.

IV. PROPOSED ARCHITECTURE

In this section, the proposed Human-Robot Interface is described. The main objective is to build a modular architecture for a highly usable multimodal HRI which allows an inexperienced operator to control multiple agents.

Considering the issues presented in [19] for HRI multitasking, the modularity allows for fast adaptation of the HRI to complex and unpredicted tasks. It will be possible to add to the HRI the interface new tools, robots and sensors with reduced development cost and minimal impact on the operator control. The multimodality allows to better address multitasking when controlling single or multiple agents and its usability allows minimally trained operators to accomplish their tasks.

A. CLOSED-LOOP AND SEMI-AUTONOMOUS CONTROL

In [14], three different control modalities are proposed for Human-Robot Interaction: manual, semi-autonomous and autonomous control. In the context of this paper, all three modalities have been considered. Although teleoperation is mainly a manual interaction modality, the intelligence that is added to the system in order to make the operation safer (such as time-delay passivation, section V-C.2) allow the robot onboard computer to take decisions according to operator's input. Since a fraction of the control is accomplished by control loops closed directly on the onboard computer, according to [14] this is semi-autonomous control in its strict sense. In this category falls as well the mission planning and scripting which will be described later on in this paper (section V-D), as the operator programs by specifying high-level procedures in order to accomplish the mission objective. The HRI allows also full autonomous control, in which the operator only acts as a viewer and interacts with the robot with asynchronous high-level commands (e.g. *move manipulator from point A to point B*).

B. MODEL-VIEW CONTROLLER

In order to guarantee the proper separation between data and its representation, the HRI is designed according to the well-known Model-View Controller (MVC) pattern. MVC is a software architectural pattern, today standard de-facto with its variants, for the development of Graphical User Interfaces, in particular for web applications [52]: the pattern splits the structure of the interface into three separate and interconnected parts in order to provide a separation between data and its representation. It is used often to enhance parallel development and code reuse, but in the context of this work, it appeared particularly effective for enhancing the adaptability of the interface to different robot configurations. The HRI can provide different representations of the same data but can also provide the same representation of different data. For example, the actual position of a robotic manipulator can be shown to the operator in different ways (3D model, sliders indicating the joints position with respect to their limits, 2D sketches, etc.). Collisions of the robotic manipulator with the environment can be notified to the operator through visual messages, haptic devices, vibrations on the controller or sounds. In this case, a unique model takes care of collecting the actual position of the manipulator from the remote controller, and multiple views are available to display it. At the opposite, another example is the visualization of a camera: a camera is always shown to the operator as an image on the screen (unique view); nevertheless, the HRI supports several cameras such as network cameras, USB cameras, thermal cameras, PTZ cameras or RGB-D cameras. Therefore, this design pattern enhances not only its development, maintainability and testability but also its multimodality.

C. MODULARITY

The developed HRI makes of its modularity one of its most important requirements. The modularity appears in every part of the MVC architecture but gives its most important results in the model representation of the robot.

The HRI implements the *RobotPart* interface (figure 4), an abstract class for every component of the robot, being that a mobile platform, a robotic arm, a tool or a sensor. A *RobotPart* object contains all these common information to every component:

- the communication channel with its remote hardware counterpart. Several network protocols are available (e.g. TCP, UDP, UDT [53], HTTP) and the proper protocol is defined in the robot configuration file.
- the list of input and outputs that can be controlled by the operator.
- the parent *RobotPart* object.
- the list of children objects.
- the transformation matrix between the part reference frame and its parent to constantly localize it in the environment.

The HRI adapts its view according to the robot configuration which is defined in the robot configuration file, the mission plan and the available control devices (figure 3). The configuration parser reads all the necessary configuration files and sets up the entire robot structure, including

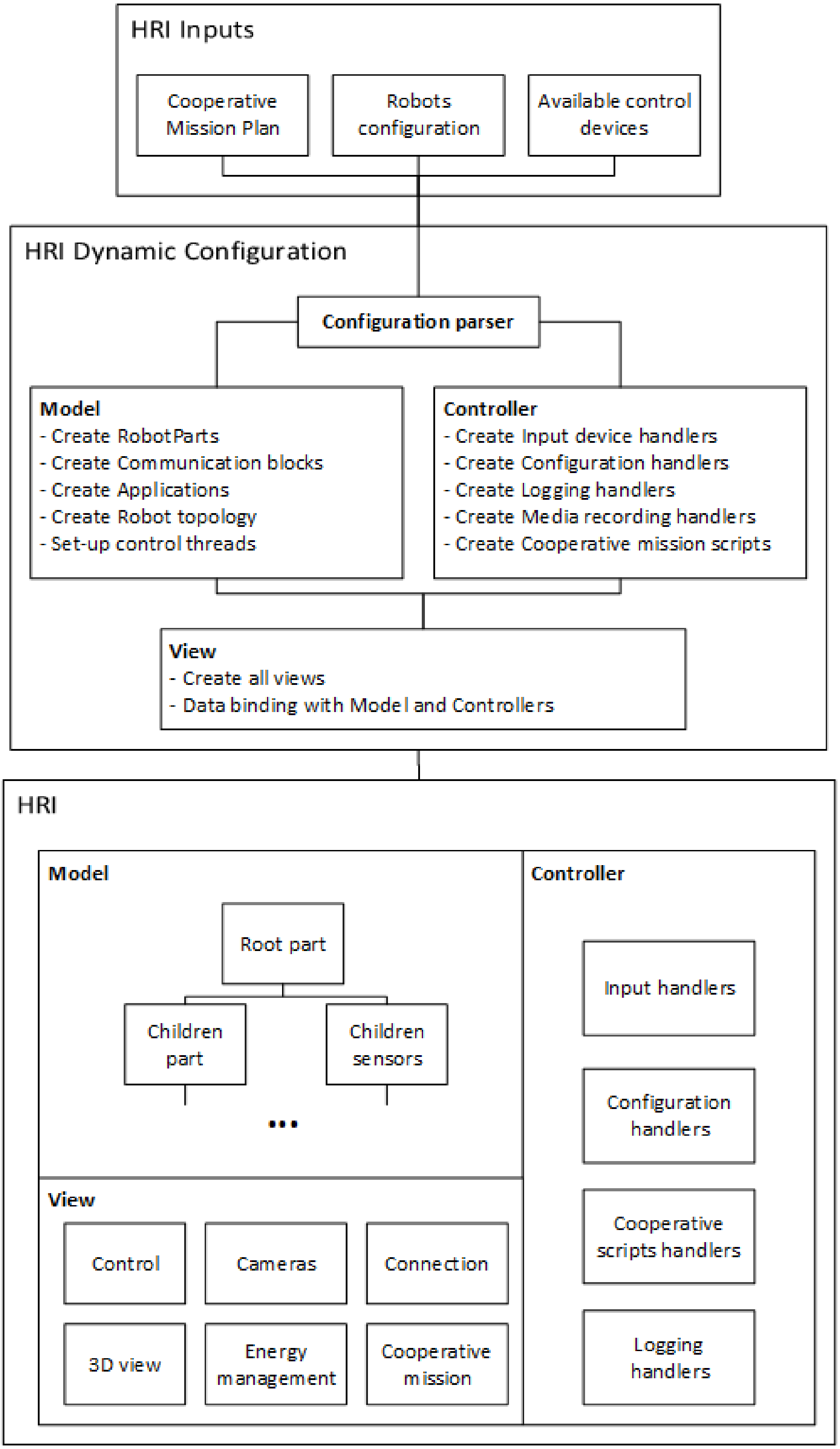


FIGURE 3. **The overall HRI architecture. Starting from the inputs (robot configuration, mission planning and available input devices) the HRI loads the proper module and builds itself in order to provide the correct user experience to the operator.**

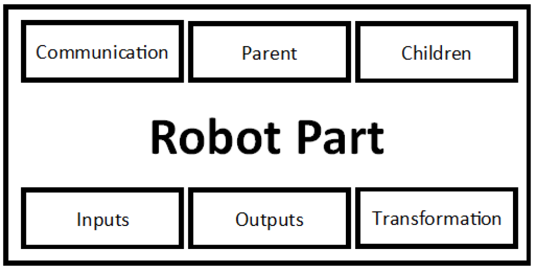


FIGURE 4. The RobotPart abstract class with its default components.

communication channels, 3D representation, already available scripts, etc. The View is created at this stage and the connection between View and Model through the Controller is instantiated. Once the HRI is fully adapted to the robot and mission configuration, the operator can connect to the robot. At this stage, the HRI tests all the network connections to ensure that the configuration file matches the real robot.

During the creation of the appropriate modules according to the robot configuration, the robot topology is built as well. The robot topology defines a tree-like structure in order to maintain a continuous localization of each robot part in the environment. Therefore, for example, a movement of a robotic arm or a mobile platform will trigger a movement of all the children parts, which will be used by the HRI to provide an accurate representation of the robot current configuration (for example in the 3D visualizer), as well as to localize all the data collected by the robot both in space and in time (e.g. radiation measurements). The robot configuration file is implemented using the XML markup language, which naturally embeds a tree-like structure during its definition.

Although the *RobotPart* abstract class represent the most generic interface for each robot component, several specialization of the *RobotPart* class are implemented according to their functionality, in order to maintain a high level of modularity (figure 5). Hence, for example, any camera, being that a standard RGB camera, a PTZ camera, a thermal camera or an RGB-D camera, inherits all from the same common *Camera* abstract class, which provide a common interface for inputs and outputs but does not define the communication protocol which will depend on the specific hardware.

This modularity also allows the control of multiple agents from the same interface instance. The leader robot would be the root of the topology tree and the slave robots would be branches of this tree. The operator can selectively control the single *RobotActuator*, or synchronize and customize the movements of different RobotParts using the live scripting tool (see Script Editor view in Figure 6).

1) APPLICATIONS

In the matter of this work, it is called *application* every higher-level algorithm (e.g. object detection, object tracking, obstacle avoidance, etc.) embedded on the robot. Applications are particular components in the HRI as they usually merge multiple RobotParts objects. Applications are defined in the robot configuration file globally and the particular structure of the application configuration is dependent by the application itself. The HRI implements an *application manager* that is responsible for collecting, activating and deactivating each application.

D. THE VIEWS

The HRI is composed of several views, each one semantically different from the others (see Figure 6).

All the control of the robot is included in the control view: this is to strengthen the separation between the interaction with the HRI and the interaction with the robot. From any other view, it is possible to send commands to the robot. It can also display the main camera, which occupies most of the screen, a secondary camera, displayed smaller in a corner, and live sensor readings. From this view, the operator can select the control mode and the type of input devices he or she wishes to use. The HRI also implements a 3D visualization tool, which displays in real-time the robot configuration and the reconstructed environment, including live environmental measurements if provided. At startup, the HRI is built according to the robot configuration inside a unique window, and the operator can navigate between views using a menu positioned on the right side. Nevertheless, each view can be detached by the main window and open in a secondary window.

E. MULTIMODALITY

According to its definition, a multimodal interface provides different modalities for user interaction (i.e. inputs and outputs). Above that, in this context, it is necessary to make a distinction between two types of interaction: the interaction domain of the user (i.e. the operator) with the interface and the interaction domain of the user with the robot. Although in the second case the user is still interacting with the robot through the interface, according to the principle of telepresence, the user should feel complete transparency between his/her actions and the robot response.

Furthermore, it is particularly important to the operator to understand which modalities are related to the interaction with the robot and which are related to the interaction with the interface, to reduce the rate of slips and lapses. For this reason, an input device is never shared between the two domains of interaction (e.g. the mouse can only be used to interact with the interface, while the keyboard can only be used to interact with the robot). With this distinction, the operator can feel confident that the interaction with the interface will not modify the robot state, with the possibility of compromising the safety of the operation. Furthermore, the operator can always drop the interaction device with the robot, knowing that the robot will not move. Finally, in order to not increase the stress of the operator but still to generate a break between the two different interaction processes, the transition must be smooth but well defined and clear.

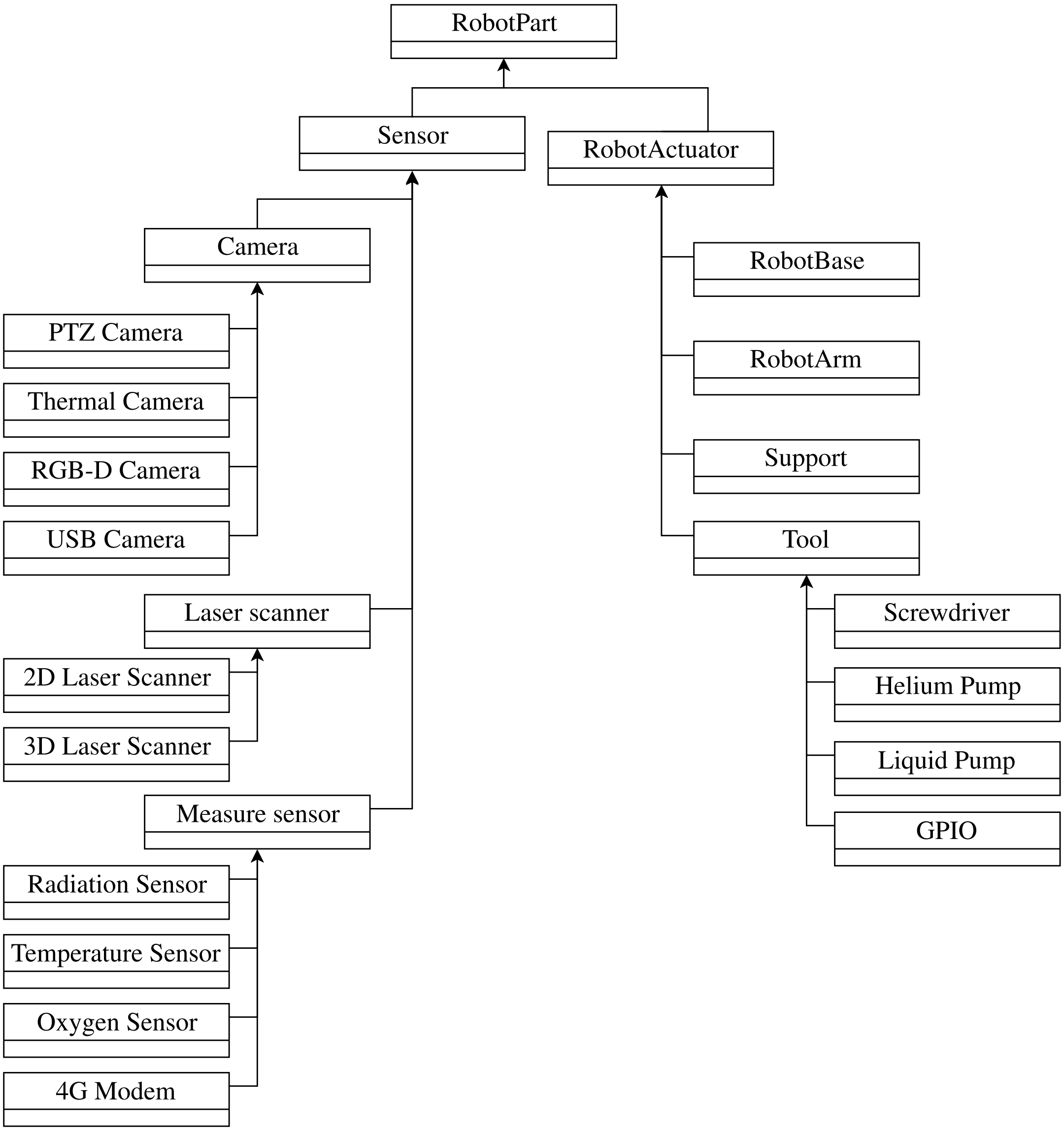


FIGURE 5. UML Diagram of the RobotPart abstract class inheritance.

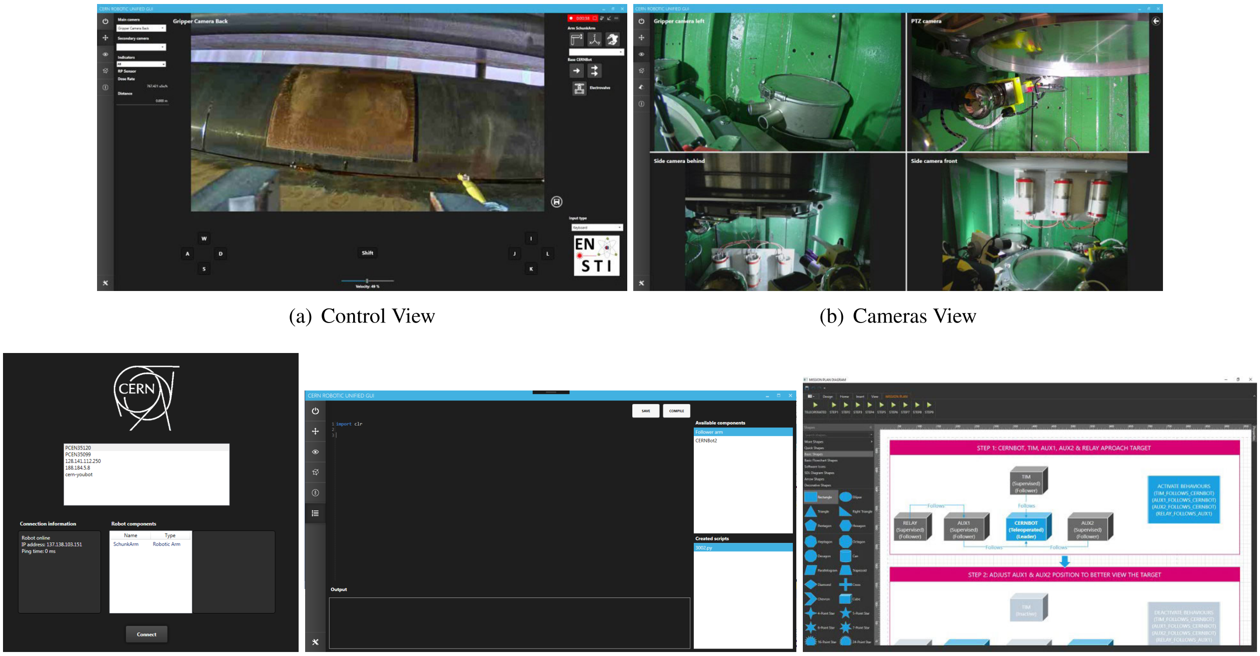
1) INTERACTION WITH THE INTERFACE

The interaction with the interface consists in that domain of actions that will change its behaviour and appearance. As previously stated, the operator should perceive this interaction as a sort of break with the intervention operation. Such a break is useful to reduce the operator's mental workload and stress. However, the transition between the two modalities should be as smooth and simple as possible, in order not to disrupt the short-term memory of the operator.

The interaction with the interface is possible through the mouse, touch screen displays and vocal commands. The three modalities are not mutually exclusive, but experimental results showed that an operator tends to use only one of them during an intervention according to the control device in use. This may be caused by the position of the operator during the intervention with respect to the control station: in fact, in the case of an operator sitting at a desk and using the keyboard or a joypad in front of one or multiple screens to control the robot, both hands are on the input device and for interacting with the interface the dominant hand tends to leave the input device to use the mouse. This is the standard behaviour of a person using a PC and comes naturally to most people.

A different case is the one of using a small-dimensioned haptic device for master-slave control: the operator is sitting at a table with only the dominant hand using the input device. The other hand is free to interact with a touch screen, an action that is perfectly possible for most of the people with the non-dominant hand.

Finally, when using master-slave interaction devices of big dimensions, or body-tracking using RGB-D cameras for control by imitation, the operator stands in front of a big screen at a certain distance and does not have easy access to



(a) Control View (b) Cameras View

(c) Connection View (d) Script editor View (e) Cooperative Intervention plan editor

FIGURE 6. Human robot interface views.

mouse and touch screens. In this case, vocal commands are the most used choice.

2) INTERACTION WITH THE ROBOT

The interaction with the robot is the process in which the operator uses an input device to modify its state. Input devices can be divided into two categories: input devices like keyboards, joysticks, 3D mice and others produce an experience of simulation, in which there is a mapping of actions and feedback between the input device and the robot. The HRI preserves a logical consistency of this mapping between these devices. Obviously, devices that provide an analogue control such as joypads or 6DoF mice allow a more precise operation than the others. In the context of this work, these input devices will be referred as *one-way input devices*.

Haptic devices and body tracking used for master-slave control belong instead to the category of input devices that produce an experience of imitation. In this case, actions and feedback do not need any mapping and they are more suitable for increasing the usability and transparency of the system, enhancing the telepresence. Nevertheless, these kind of devices are not common and an inexperienced operator could find intimidating the interaction with them. In the context of this work, these input devices will be referred as *two-ways input devices*.

3) INTERACTION WITH ROBOTS TEAMS

One of the issues that this work has tried to solve has been to guarantee a uniformity of interaction when controlling a single robot and a team of robots. As previously explained, the modularity of the system allows controlling several robots from the same interface instance. However, additional considerations were made to make the control easier, such as the use of the Cooperative Intervention Plan Editor and the Script editor View (see Figure 6). More details about this feature will be given in section V-D.

F. CONTINUOUS INTEGRATION

Software updates are not currently widely used in automated industrial systems, mainly due to the fear of compromising a working system, although not perfectly functional. However, the HRI presented in this paper must deal properly with software updates, in order to adapt constantly to continuous requirements changes coming from new tasks as well as to solve bugs. Therefore a *continuous release* [54] approach through an automatic one-click update system is implemented in the HRI. Details about the updates are displayed in the related modal and rollback to previous versions is always possible. Nevertheless, updates are not automatic (such as in modern web applications and browsers) but must always be approved by the operator. The continuous integration life-cycle of the HRI is depicted in Figure 7.

Updates are triggered by the operators (through bug reports), by hardware updates (new sensors, tools and robots) or by tasks requests. After a software update has been implemented, various tests must pass: unit tests and integration tests are always performed, while additional tests such as cohesion tests and coupling tests to guarantee that the appropriate level of modularity of the HRI is maintained are made only when dealing with new functionalities (new hardware

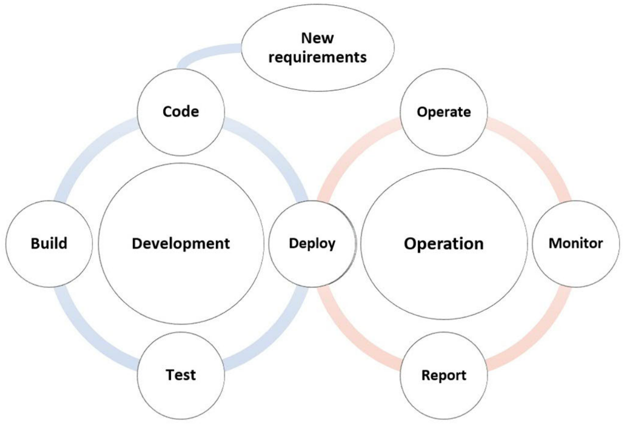


FIGURE 7. Continuous integration lifecycle for the HRI.

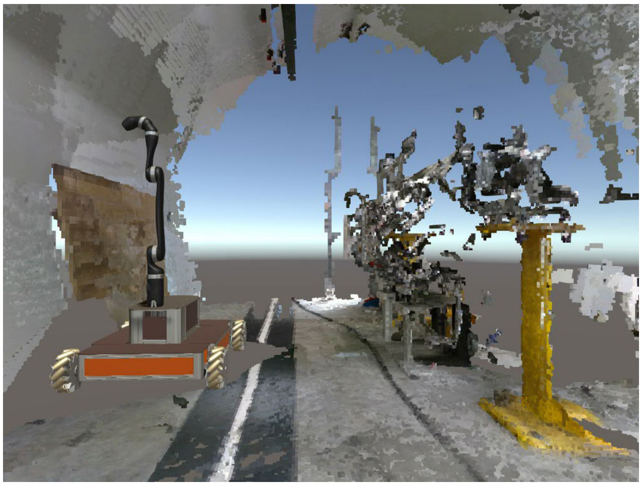


FIGURE 8. Real-time 3D view of a collimator in the LHC tunnel at CERN.

or new interventions). While unit tests and integration tests are automated, using modern testing frameworks available, coupling and cohesion tests are more complex and require an interaction with the hardware. Once all the tests passed, a new release is deployed and feedback from the operators are expected.

V. SEMI-AUTONOMOUS CONTROL FEATURES

The modularity of the interface has been explained previously, which ensures its adaptability to future robotic missions. Besides, the multi-modality of the interface has been explained, which increases its usability to minimally trained operators. In this section, the semi-autonomous control features are presented, which increase the usability of the interface, as well as allow the control of multiple agents minimally reducing the operation performance due to the increased multi-tasking required to the operator.

A. TELE-PROPRIOCEPTION

Proprioception (or kinesthesia) is a person sensorial skill of perceiving the position and the movement of the body [55]. The brain integrates information from proprioception and the vestibular system into its overall sense of body position, movement, and acceleration. Tele-proprioception is defined as the operator capacity to associate the position and movement of a robotic platform with respect to the environment from the available points of view [56]. Handling properly the operator perception of the robot increases the operator telepresence in the environment. The proposed HRI implements various strategies for ensuring proper operator telepresence and provide consistent feedback during the entire operation.

1) 3D VISUALIZATION, ENVIRONMENTAL RECONSTRUCTION AND VIRTUAL REALITY

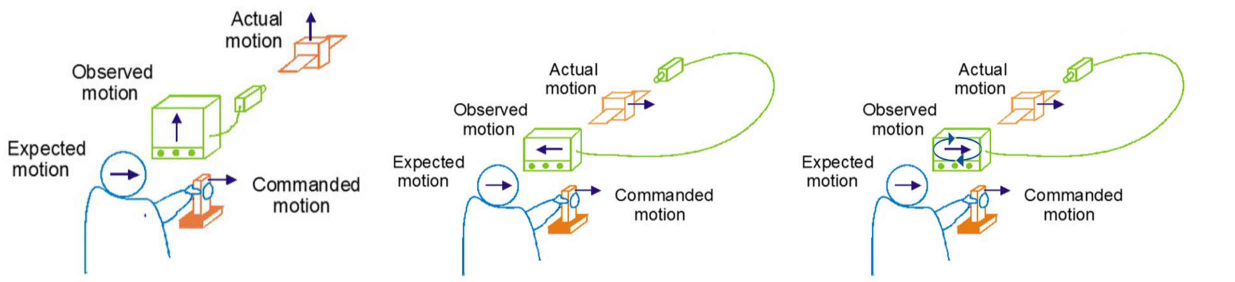
The operator could have difficulties in visualizing complex robot operations, such as multi-agent configurations. Furthermore, the concept of telepresence is difficult to be adapted to multi-agent operations as the operator ‘‘acts’’ as different robots in the field. A solution for providing a more clear overview of the environment surrounding the robot and the robots themselves is the integration in the HRI of a 3D visualization toolkit, with the optional integration with a virtual reality headset. A proper 3D visualizer should, at first, display accurately the various robots, properly positioned with respect to each other. Moreover, it should display a 3D reconstruction of the environment, if appropriate sensors are available (e.g. RGB-D cameras, LIDAR scanners etc.), including environmental information such as radiation, temperature, oxygen level, etc.

Environmental reconstruction is an important feature during inspection and reconnaissance tasks. Inspection tasks using robots are becoming more and more requested, not only for controlling the status of particular equipment but also for preparation for a future robotic or human intervention. In this last particular case, it is crucial not to perform a simple visual survey of the area but to collect as well as much data as possible about the environment in order to optimize human access, reducing personnel exposure to hazards.

In such scenarios, CERN uses a framework for the preparation of human intervention in highly radioactive areas called ALARA (As Low As Reasonably Achievable). CERN ALARA [57], [58] aims to reduce human exposure to radiation at the minimum and to reduce the radiological impact on the environment focusing on three main principles: (1) justification, (2) limitation, and (3) optimization [59].

However, the preparation for the intervention is not straightforward, due to a lack of information about the failed component that requires the intervention and about the environmental characteristics that could be found in place. At CERN, simulation software are used to predict the radiation dose rate in a certain place [60]: however, such simulations could be far from the reality due to the small amount of data about the real radiation in place.

In particular, measuring radiation is not an easy task: the radiation dose rate is extremely dependent on the distance



(a) Geometrical misalignment (b) Observational misalignment (c) Proper compensation of both geometrical and observational misalignment

FIGURE 9. Visual misalignment correction for improved telepresence.

from the radioactive object [61] and a big amount of values has to be collected in multiple places to have a precise radiation dose rate map of the object. This process is not easy when applied to complex objects (e.g. collimators), or to objects which are not easy to reach.

The HRI uses voxels to represent all the environmental information. This allows optimizing the memory requirements when mapping large environments. The HRI receives from each robot different octrees, which are a well known, diffused and efficient representation for voxels [62]: one representing the environment, usually with the highest resolution, and one for each environmental sensor that is integrated in the robot, of which resolution may vary according to the physical quantity that they represent. This method relies on the localization and mapping capabilities of the robots in the environment.

As this process is designed to run in background without any interaction from the operator, octrees are not sent through the network directly as they are, in order to not overload the communication, compromising the control of the robot. Before they are sent to the HRI, a pruning operation is performed [63] and its output is compressed through a loseless fast compression algorithm [64].

Once the octree is received and decompressed, it is added to the 3D visualizer. As already mentioned, the 3D visualization must not compromise the robot teleoperation. However, it is a useful feature that it is required by most of the operators. Therefore, it should not rely on high-end desktop computers for its representation and an optimization method is integrated into the HRI in order to maintain always the visualization and the operation smooth. For each octree, starting from the root node, the bounding box of each octree node is considered. By computing the frustum planes for the current camera view [65], if the considered node is out of view, it is not rendered and all its children nodes are not considered. Furthermore, the distance from the camera and the octree node is computed: if the distance is higher than a threshold, the parent of the considered node is displayed and all its children nodes are discarded. This visualization pruning allows a fast scan of the entire octree and to render only the visible nodes. Moreover, the distance threshold is adjusted dynamically in order to keep the refresh rate of the 3D visualizer around 10 fps, more than acceptable for this kind of visualization.

2) CAMERA ROTATION COMPENSATION

Operator perception about the robot pose in the environment could vary according to the chosen control modality, due to inconsistency between different feedback modalities. One example is the feedback misalignment between visual and haptic feedback when using a master-slave control modality. In this case, the operator is provided with a haptic feedback from the master device which reflects the pose of the remote slave with respect to a specific reference frame (usually the base of the manipulator) and with a visual feedback which is referred to the position and orientation of the camera plane. According to the control modality chosen by the operator, the HRI compensates the orientation of the camera in the camera plane in order to realign the point of view of the operator to one that is consistent with other feedback provided. Figure 9 shows an operator controlling a remote object. In the first case, a wrong mapping between the operator input device and the actual movement of the robot causes what is called geometrical wrong adjustment (figure 9(a)). In the second case, a misalignment of reference frames between the camera plane and the input device frame causes an observational wrong adjustment (figure 9(b)). In the final case, the visual feedback provided to the operator is rotated in order to be consistent with the input device reference frame (figure 9(c)).

3) VISION-AIDED OBJECT GRASPING

As previously mentioned, the HRI supports high-level functionalities to improve the operator's experience and facilitate some tasks. Among these, it is worth to study the use of a vision tracking system for manipulator guidance during object grasping. The system is fully integrated into the control loop, providing the proper feedback according to the modality selected by the operator. The core idea of this functionality is the possibility for the operator to select an arbitrary object to be grasped in the environment and the HRI ''helps'' the operator to accomplish the task. Although this might sound straightforward thanks to modern sensors and algorithms, the need of extending this method to metallic objects installed on metallic supports while allowing depth calculations with a single monocular camera, required detailed investigation for a specific solution.

Most of the objects available in CERN's accelerator environments are metallic featureless objects, making unstable most of the standard computer vision feature tracking algorithms and unreliable RGB-D cameras and laser pointers sensors for depth estimation, due to metallic reflections. The system proposed in [66], uses a monocular camera installed on the manipulator end effector, and multiple Kernelized Correlation Filters (KCF) [67], each one tracking a different part of the object of interest. By considering the movement in space of the robotic arm, known by its kinematics, it is possible to estimate the distance of the end effector from the object (figure 10). With the goal of grasping the object, therefore having the object inside the gripper of the manipulator, the system outputs corrections to the operator's control setpoints and feedbacks to the operator in order to facilitate the approach during the grasping phase.

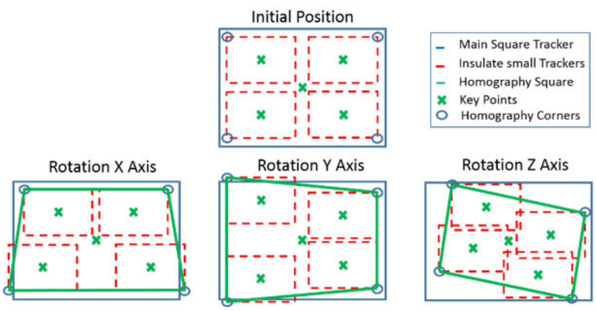


FIGURE 10. Working principle of the depth estimation system for manipulator guidance.

Moreover, the system is connected to an object recognition neural network-based algorithm. The algorithm uses a deep neural network to extract the bounding box of known objects allowing a more precise tracking and depth estimation.

This feature, not only improves the usability of the HRI but the safety of the operation as well, by bounding the movements in the space as the system knows the task goal.

B. WALL RECOGNITION FOR SAFE FAST NAVIGATION

Mobile robot navigation plays an important role in the HRI development as well. Due to hardware limitations or particular configurations of the robot, it is not always possible to have efficient camera views which allow seeing properly the entire robot width during the navigation. However, some of CERN's specific tasks still require fast navigation in accelerator's tunnel, maybe due to short available access time. Robot navigation literature is full of autonomous navigation, obstacle detection and obstacle avoidance algorithm which can be used to assist the operator's driving. In the matter of this work, for example, a specific feature of particle accelerators has been used. CERN's particle accelerators, usually, present a clear wall on one side of the tunnel and a small passage for humans, material and, in this case, robots (around 1.6 metres). Some tasks require navigating in the accelerator's tunnel at a speed higher than 2 m/s, making difficult to remote manually control of the robot, especially in case of network delays. In order to assist the operator, a wall recognition algorithm based on RANSAC [68] using a 2D LIDAR has been implemented. The algorithm computes the relative angle between the robot direction and the recognized wall in order to maintain always the robot platform in the proper direction and at the proper distance from it. The operator can activate and deactivate the assisted driving and will have to take care only of accelerating and decelerating the robot.

Although more complex navigation algorithms can be implemented, this example shows how the operator can activate and deactivate applications available for the various robot parts to get assists in the accomplishment of the task. The modularity of the HRI allows to provide the proper applications according to the equipped sensors and to extend them constantly with state of the art algorithms available in the literature.

C. COMMUNICATION

For safety reasons, CERN underground facilities provide a complete mobile network coverage, to allow any worker to call for emergency from any point, without the need to carry specific equipment such as radio transmitters or others. The mobile network provides 2G coverage for voice communication, 3G coverage for medium throughput data (10/2 Mbps) and 4G coverage for high throughput data (20/20 Mbps). The coverage allows the connection to the standard external provider Access Point Name (APN) for calls and mobile data and to the CERN internal APN, for connection to the CERN intranets (General Purpose Network and Technical Network). In particular, a device connected to the CERN internal APN is assigned a static IP address that is reachable from any other terminal connected to the CERN corresponding intranet (see table 1).

This installation allows communicating with any robot located at any point of the CERN territory by equipping the with 3G/4G modems. This, theoretically, overcomes any distance limitation between the operator and the robot, present for example on commercial robots equipped with a point-to-point radio connection.

The drawback of such a connection is the reliability of the coverage and its performance in terms of throughput and delay. Since the connection with the robot is established over internet protocols, the performance is related to the mobile signal strength in the operation facility, the distance to the robot (in terms of the number of network nodes that are crossed from the control terminal to the robot) and the congestion of the network.

The performance of the communication protocol is fundamental in order to guarantee a smooth operation: several studies demonstrated that teleoperation was significantly affected with a rate of five to six frames per second and became almost impossible to perform when the frame-rate dropped below

TABLE 1. Available networks in the underground accelerator.

Available Network	Bandwidth	Delay	Offered Services
GSM (2G)	less than 50 kbit/s	500 ms	SMS, text and picture messages, MMS, and Voice communication.
UMTS (3G)	10 Mbps down/2 Mbps up	200 ms	Data transmission
LTE (4G)	20 Mbps down/2 Mbps up	40 ms with peaks of 250 ms - 300 ms	Data transmission

three frames per second, in which case would be necessary to apply semi-autonomous control techniques.

The HRI can handle connection losses, providing recovery scenarios procedures, high communication delays and different protocols for ensuring the best solution for the type of data transmitted. For the communication between the robot and the HRI, a custom communication protocol in the application layer has been implemented. The protocol is not related to the underlying internet transport protocol, which can be TCP, UDP, UDT [53], RTSP, or a combination of these for some applications.

1) TIMESTAMP SYNCHRONIZATION

Using internet-based protocols for robotic teleoperation could create problems in terms of communication delays. Hence, the header of every packet of the custom application protocol contains a timestamp in milliseconds indicating when the data contained in the packet was produced. The timestamps of all the connected nodes must be synchronized in order to react properly to variable communication delays.

The synchronization is based on the standard four-timestamp mechanism of the Network Time Protocol (NTP) [69]. This commonly used method measures the transmission round-trip time between connected nodes and uses this to estimate the offset between their respective clocks. The protocol relies on the assumption that, during the synchronization, the communication delay is symmetric.

The protocol works as follows:

1) The HRI sends a packet containing the current HRI timestamp to the robot at time T_s^h.
2) The robot receives the packet at time T_r^r and immediately sends back the current robot timestamp at time T_s^r.
3) The HRI receives the packet containing the robot timestamp at time T_r^h.

In the previous notation, in T_s^h and T_r^h the h indicates the time relative to the HRI reference time, while in T_r^r and T_s^r the r indicates the time relative to the robot reference time. The relationship between T^r and T^h is the following:

$$T^h - T^r = O \tag{1}$$

where O is the offset between the two timestamps taken by the two agents at the same moment and the value estimated by the process.

As previously stated, the method is sensitive to asymmetries in communication delay. Unfortunately, it is not possible to measure communication asymmetries in the network with only two agents. Therefore, the method will estimate $\widetilde{O}$, which will be only an approximation of the real O.

In order to estimate more accurately $\widetilde{O}$, two conditions can be imposed. First, it is safe to assume that the offset between the local timestamp of the robot and the local timestamp of the HRI O, and therefore its estimation $\widetilde{O}$, is constant or slowly changing during the entire intervention time. The unavoidable fluctuations of the clocks between the two agents which will make O vary in time are negligible in the short term. Thus, it is possible to create a moving average of $\widetilde{O}$ in order to filter rapid changes coming from the estimation process, which are more probably generated by communication asymmetries.

Furthermore, during the computation of the communication delay d_P for a specific packet P, the delay can not be lower than 0 (this would mean that one agent received a packet before it was transmitted). However, this situation happens frequently when dealing with highly asymmetric delays. Therefore, the estimated offset $\widetilde{O}$ is, at least, lower estimated by a value d_P, which provides, then, a lower-bound for $\widetilde{O}$.

In section VII-A the performance of the proposed synchronization method are presented by comparing the timestamp offset estimation $\widetilde{O}$ with the standard NTP implementation, in both simulated and real case.

2) MANUAL CONTROL WITH FORCE FEEDBACK AND PASSIVATION

Time-delay passivation is a technique for reducing the energy of the control setpoint according to the delay between the operator and the robot. Several formulas have been proposed during the years, both in the case of constant time delay and time-varying delay [70]. When dealing with an internet-based infrastructure, the delay in the network is influenced by several factors, especially the network load and the 4G signal strength [71].

The time-delay passivation problem has been studied during the years as a stabilization method for force reflection in master-slave control. However, in the context of this work, the operator can choose between different control devices, which not always provide force feedback. Therefore, it is important to take care of the time delay and react according to the control modality chosen by the operator. A more accurate analysis of the stability of the control in presence of time-delay in this context is presented in section V-C.3.

Studies showed that people are generally able to detect latency as low as 10-20 ms [72]. Furthermore, it was demonstrated that when system latency is more than 1 second, operators begin to switch their control strategy to move-and-wait instead of continuous commands to compensate for the delay [73]. CERN 4G network presents an average delay of 40 ms, with peaks of 250-350 ms. Hence, the delay is perceivable by the operator but is not high enough to change the control strategy. Other studies showed that delays above 300 ms would make the operator decouple the commands from the robot's feedback [74]. Furthermore, as already stated, 4G latency is not constant, and it is demonstrated that movement times increase by 64% and error rates by 214% when latency increases from 8.3 ms to 225 ms [75].

The definition of the delay is done by considering the delay of the control set-point from the perspective of the robot control loop. Considering the simplest scenario of an operator using a joypad as a control device and using a standard camera as feedback, the overall delay of the set point provided by the operator is the sum between the transmission delay of the camera image and the transmission delay of the control command. This because the operator sends a specific command to the robot according to the image that is showed by the HRI at that moment, but that it was produced, elaborated and transmitted by the robot sometime in the past.

Therefore, the command delay is defined by the following equation (see formula 2):

$$d_c = t_{rc} - t_i \tag{2}$$

where t_{rc} is the timestamp of the control set-point received from the network by the robot and t_i is the time when the image was produced.

Once the delay is defined, actions can be taken to mitigate its effects. A distinction between one-way input devices and two-ways input devices is done during the network delay mitigation (see section IV-E.2).

In the case of a one-way input device, the proposed solution is a formula that reduces the overall control energy according to the delay. In this control modality, the operator controls the velocity of each actuator, which can be limited according to the communication lag.

This solution could generate confusion on the operator. As the operator has already the possibility to limit the control energy of the commands through a slider widget available on the HRI, an additional limitation caused by network delay could overly slow down the controlled actuator. In this scenario, the operator will be more inclined towards increasing the control energy back on the HRI to obtain the desired movement. If at this moment the delay suddenly reduces, the robot could accelerate unexpectedly. Therefore, the output energy must be a function of both the delay and the control energy. The output energy is computed using the following polynomial function:

TABLE 2. Network Latency Passivation Parameters.

Parameter	Value used experimentally
p_{00}	-0.01152
p_{10}	1.127
p_{01}	0.0002581
p_{11}	-0.002793
p_{20}	-0.1152
p_{02}	-5.591e-07
p_{21}	-7.57e-05
p_{12}	1.658e-06
p_{03}	7.318e-11

$$\begin{aligned} E_o(d_c, E_c) &= p_{00} + p_{10}E_c + p_{01}d_c + p_{11}E_cd_c + p_{20}E_c^2 \\ &\quad + p_{02}d_c^2 + p_{21}E_c^2\, d_c + p_{12}E_cd_c^2 + p_{03}d_c^3, \end{aligned} \tag{3}$$

where $E_c \in [0; 1]$. As an example, the control energy when moving a robotic arm controlling the speed of each joint is computed as follows:

$$E_c = \frac{\sum_{i=1}^{N} \frac{v_i}{v_{imax}}}{N}, \tag{4}$$

where N is the number of joints of the robotic arm, v_i is the velocity set-point for the i-th joint and v_{imax} is its maximum velocity. The use of velocity set-points instead of positions allows this calculation without taking into account the non-linearities coming from rotational displacement.

The values of the parameters must be computed according to the type of delay that the communication network presents. Considering the studies mentioned before, at the characteristic of the CERN 4G latency, we can assume that $d_c \in [0; 500\,ms]$. If the delay is higher than 500 ms, the output energy is nulled. Moreover, if the delay is lower than 20 ms, the control energy is sent to the actuators unaltered. With these considerations, the values of the parameters presented in table 2 are computed through experimental validation with the current network configuration.

3) STABILITY OF THE CONTROL LOOP

As previously mentioned, the control of a remote robot in presence of variable communication delays could create instabilities in the behaviour of the robot. However, the stability analysis of the control loop should be done separately according to the type of interaction device the operator is using. For a one-way interaction device, such as keyboard, joypads, 3D mice, the control loop in use is the one shown in figure 12. As the interaction device is not able to provide

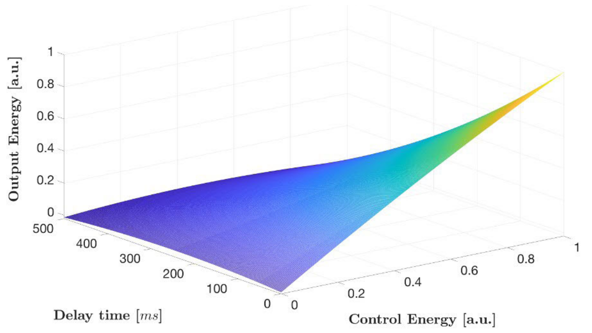
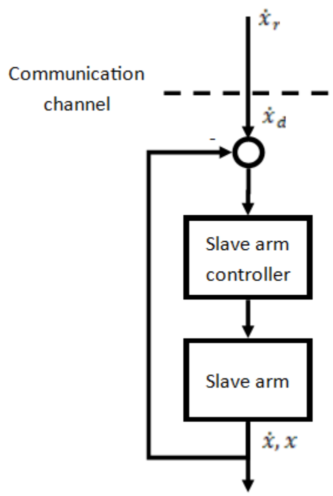


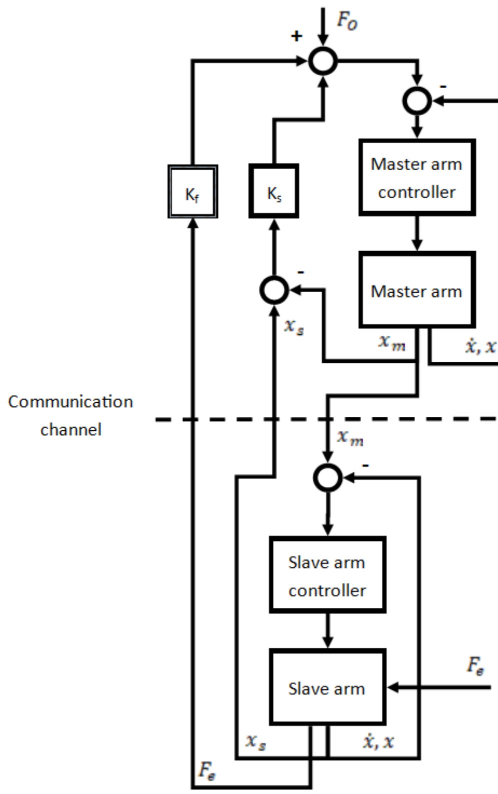
FIGURE 11. Plot of the time-delay passivation function using the parameters defined in Table 2. On the x axis, the control energy is represented in the range [0; 1]. The y axis shows the time delay in seconds in the range [0; 500]. Finally, the z axis represents the output energy in the range [0; 1].

direct feedback (such as force feedback) to the operator, the closed-loop control is available only on the robot side. Therefore, the time-delay passivation method explained so far affects the control set-point before this enters in the closed-loop control of the robot. In the matter of this paper, it is not interesting analyzing the stability of the closed-loop control on the robot side, but only the stability problem caused by the time-delay passivation method. Therefore it is safe to consider that, if the closed-loop control on the robot side is stable, the entire control loop including the operator is stable as well.

A different analysis is necessary for two-way interaction devices, such as a master-slave interaction system, with which, the operator, continuously receives feedback from the robot. In this case a bilateral control loop is involved (figure 12(b)). The specific architecture chosen is known as the Environment-Force-Compensated (EFC), whose stability and transparency has been discussed in [76]. The force feedback coming from the slave arm is a combination of direct forces measured by the manipulator and indirect forces caused by position and velocity error between the master and the slave. An added value to the system is the possibility for the operator to tune K_f in order to obtain higher or lower sensitivity to external contacts and to tune K_s in order to adapt to delays to communication variations through HRI widgets.



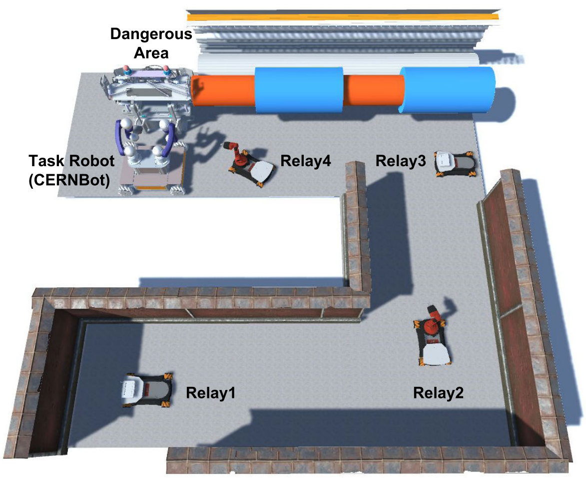
(a) The control system using a one-way interaction devices to control a robotic arm. $\dot{x_r}$ is the velocity set-point from the operator and $\dot{x_d}$ is the velocity set-point after the passivation



(b) The control system using the master-slave interaction to control a robotic arm. F_O is the force applied to the master arm by the operator, F_e is the environmental force applied to the slave arm, x_m is the position of the master arm, x_s is the position of the slave arm, K_f is the gain factor for the environmental force feedback and K_s is the gain factor for the position-error feedback.

FIGURE 12. The two control systems used for controlling the remote manipulators.

(a) Relay robot configuration



(b) Relay robot positioning to reduce communication occlusions

FIGURE 13. **Relay robots for temporary Wi-Fi LANs.**

4) RELAY ROBOTS FOR TEMPORARY WI-FI LANS

The CERN internal mobile coverage is available with good performance in almost every experimental area (i.e. tunnels and caverns). However, in some areas (e.g. radioactive material stocking hangars), the mobile coverage is not available. For this reason, each robot is also equipped with a Wi-Fi access point to allow point-to-point communication (Figure 13(a)). Differently than the 4G, the range of the Wi-Fi limits the connection with the robot. To overcome this limitation, relay robots can extend the Wi-Fi range: these robots will be additional nodes in the created LAN and they can be autonomous or manually positioned in the environment. In both cases, the HRI offers the possibility to control them and all their functionalities as any other robot using the multi-agent control methods provided by the HRI described before.

D. SCRIPTING FOR MULTI-TASKING

Operator multi-tasking is particularly relevant while controlling multiple agents. Certain studies state that, during critical operations and stressful tasks, at least two operators are necessary for each robot [77]. However, other studies demonstrated how an operator can control multiple agents, provided that they exhibit an appropriate level of autonomy [78], [79]. In the proposed HRI, it is possible to provide the required level of autonomy adapted to the task by programming behavioural scripts which can be executed at any moment through the interface.

The scripts are executed in the *model* level of the MVC pattern. Therefore, it is possible to program complex behaviours which can be activated in runtime by the operator, like for example synchronizing the movement of two actuators, executing complex trajectories or implementing visual tracking and servoing techniques. This is done by creating virtual *RobotPart* object which implements the proper methods for that object.

The scripts are usually programmed in advance by a robotic expert, during the planning of the intervention. The activation of the scripts is available trough a mission planning view, in which the operator can select the stage of the intervention and the proper scripts will be automatically activated (figure 6(e)).

E. OPERATORS' TRAINING

One of the key factors for allowing safe and efficient teleoperation is to provide the operators with a training environment. However, it is not feasible for a company whose main business is not robotics, to provide and maintain a training area as well as robots dedicated to training. Therefore, the proposed HRI provides a complete integration with different robotic simulators.

The HRI uses Unity [80], a popular game engine, for 3D visualization and simulation. Unity provides accurate physics and huge environments rendering as well as high-resolution textures. By adding to Unity a communication wrapper, following the same application protocol used by real robots, it is possible to control all the robotic pool in real CERN's scenarios [81] (Figure 14(a)), as well as have access to all the HRI functionalities such as multi-agent control and live scripting.

A communication wrapper with the robotic simulator Gazebo [82], a more popular solution in the robotic community, has been developed as well (figure 14(b)). Gazebo implements a more precise low-level actuator control and allows faster integration of new robotic systems. However, the quality of the 3D environment due to lower quality textures makes the use of this simulator less immersing than Unity.

Both robotic simulators include the entire pool of robots and sensors, training scenarios and real-case scenarios. The entire system can run on a normal PC and is compatible with all the control devices. Simulated cameras help the operator to get confident with their field of view.

The compatibility with the robotic simulator is useful not only to train with no expenses inexperienced operators but also to test various robot configurations, in order to obtain the best one for the intervention that has to be performed. This intervention preparation has been integrated with the

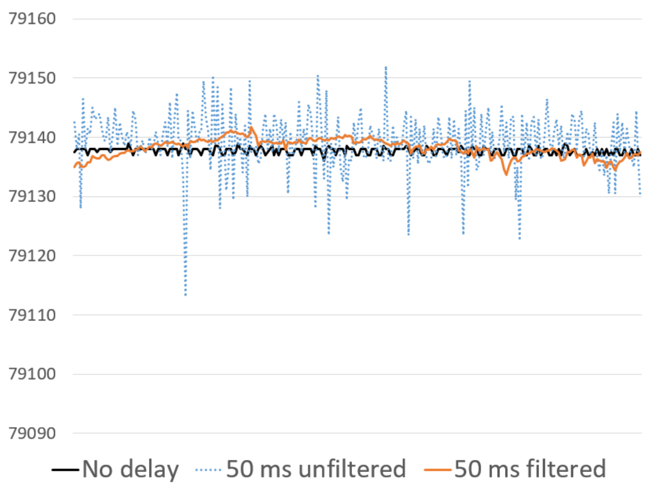


FIGURE 14. **The timestamp offset calculation for 50 ms simulated delay. The black line represents the ground truth, the blue dashed line represents the four-timestamps mechanism of NTP and the orange line the estimated offset.**

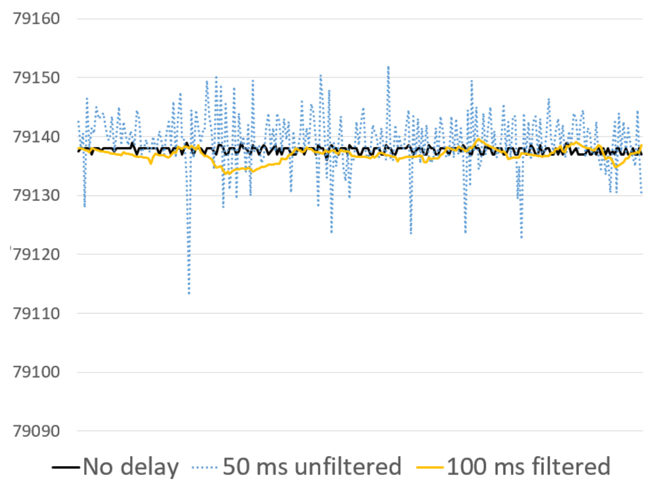


FIGURE 15. **The timestamp offset calculation for 100 ms simulated delay. The black line represents the ground truth, the blue dashed line represents the four-timestamps mechanism of NTP and the yellow line the estimated offset.**

VERO (Virtual Environment for intelligent Robotic Operations) framework: this module is practical for acquainting with the environment in order to have better planning for the interventions. Moreover, with VERO, the operator can train the operations before the interventions. In this way, using this system will save time, avoid unnecessary steps or overexpose to radiation, and foresee possible dangers [10]. In VERO it is possible to integrate environmental information collected during previous operations as well (section V-A.1).

VI. SAFETY

When developing a HRI for robotic interventions in disastrous scenarios there are different levels of safety to take into account. The most important is human safety; afterwards, the robotic teleoperation should be taken out in order to guarantee the safety of the equipment and the robot, in this order. In this section, various levels of safety, which were taken as requirements during the development of this HRI, are explained.

A. HUMAN SAFETY

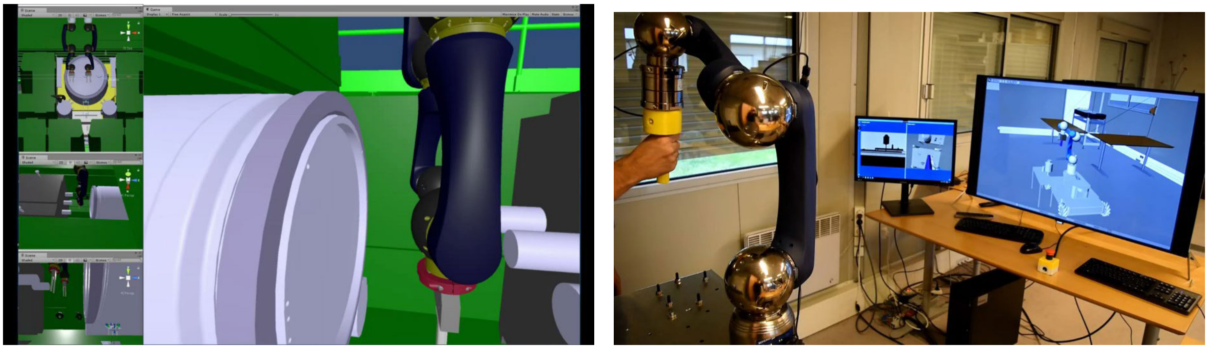
Human safety during a robotic intervention is, in principle, granted by the fact that no human presence is needed in the hazardous area. Nevertheless, recovery strategies must be studied in advance in order to prevent human intervention to recover a broken robot. In this case, multi-agent collaboration is fundamental in order to put both the intervention area and the robot in a safe condition. Recovery scenarios must be created together with facility experts and equipment experts in order to provide the safest solution. A possible future functionality of the intervention-planning tool could be to study possible failures and recovery scenarios. During a recovery strategy, it is not possible to use automated scripts due to the unpredictable state of the system, but different recovery scenarios, according to the type failure, can be created and all the operators would be shown on the planning tool the procedure of the strategy.

B. CONTROL SAFETY

In order to avoid possible damages to the equipment and to the robot, it is necessary to prevent a series of risks coming from the reliability of the communication network and to prevent possible software bugs to have a dramatic effect. Therefore, on both the robot side and the HRI side multiple watchdogs are constantly checking that the entire system is working properly, that the connection between robot and HRI is alive and that the communication messages are consistent. In this matter, for example, the time-delay passivation technique described in Section V-C.2 plays an important role in order to compensate on possible unbounded communication delays. In the HRI, all functionalities related to the actuators must be constantly active. In details, each actuator on the robot (e.g. the robotic platform itself, a manipulator, a tool) has two background threads on the HRI, one responsible of receiving messages about the status of the actuator, and one responsible of sending commands to the actuator. The receiving thread must be constantly running, since the first connection to the robot. If this last has a problem, all the robot functionality are stopped.

C. SAFETY FOR AVOIDING HUMAN ERRORS

Long operations can increase the risk of human error. The operator, usually, is tired and stressed, and the level of attention tends to drop dramatically if the operation lasts longer than expected. A safety feature implemented in the HRI uses a commercial eye-tracker for monitoring the attention status of the operator. The HRI constantly monitors the eye position of the operator with respect to the screens and inhibits the operator's commands once the operator looks away from the screen for a certain amount of time. This allows for safer operation, especially during fast driving.



(a) Example of Virtual Reality simulation using Unity, during a bi-manual intervention.
(b) An operator using the master-slave interaction system to operate a simulated robot in the Gazebo simulator.

FIGURE 16. **Simulated environments in Unity and Gazebo for procedure validation and operators training.**

TABLE 3. **Network offset estimation using different network delays. $\tilde{\bar{O}}$ indicates the average estimated offset and $\sigma^2_{\tilde{O}}$ the standard deviation of the estimated offset.**

Known delay	NTP		Filtered	
	$\tilde{\bar{O}}$	$\sigma^2_{\tilde{O}}$	$\tilde{\bar{O}}$	$\sigma^2_{\tilde{O}}$
0 ms (Direct LAN)	70.89 ms	0.16 ms	-	-
50 ms	70.47 ms	3.23 ms	70.29 ms	0.74 ms
100 ms	70.89 ms	3.61 ms	70.51 ms	0.46 ms
150 ms	69.41 ms	5.82 ms	69.81 ms	1.25 ms
200 ms	70.93 ms	3.29 ms	70.83 ms	0.99 ms
250 ms	71.47 ms	3.74 ms	70.64 ms	0.69 ms
300 ms	69.52 ms	6.13 ms	69.57 ms	1.11 ms
350 ms	69.75 ms	4.84 ms	70.21 ms	1.14 ms

TABLE 4. **Timestamp offset estimation in a real scenario.**

	$\tilde{O}$ **average**	$\tilde{O}$ **standard deviation**
Non-filtered	2006.73 ms	2.62 ms
Filtered	2005.33 ms	0.65 ms

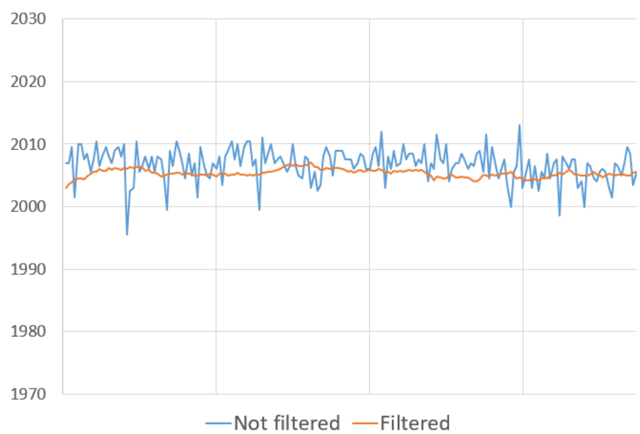


FIGURE 17. **Comparison of timestamp offset estimation in a real scenario.**

VII. RESULTS

A. COMMUNICATION

In this section the estimation of the timestamp offset between the HRI and the connected agent, presented in section V-C.1, is analysed. Table 3 shows the network offset calculation with different network delays and comparing them with a vanilla four-timestamps mechanism implementation of the NTP protocol. Network delays were simulated using *Clumsy*, an open-source software for simulating network lags on Microsoft Windows computers, and *tc*, a Linux standard application for traffic control. These programs were set to simulate a network delay with a 10% variation on their average value. The asymmetry in the network delay was guaranteed by using the traffic control programs independently on both machines, only for outbound packets. For calculating the ground truth offset, standard four-timestamps mechanism of NTP has been used without any traffic control program running. In figure 14 and in figure 16, it can be noticed how the offset estimation using the proposed filtering (both the moving average and the positive delay condition), provide better results than the standard four-timestamps mechanism of NTP.

Table 4 and Figure 17 show a comparison between unfiltered NTP and the proposed filtering for the estimation of the timestamp offset between HRI and robot. Filtered values result in an estimation uncertainty lower than 1 ms, which is more than appropriate for this application. Obviously, it is not possible to compare both estimations with the real value, which is known only in the simulated scenario presented before.

B. EXAMPLE OF REAL INTERVENTIONS

The HRI described in this paper has been used in more than 100 real interventions in CERN's accelerator facilities, for a total of more than 150 hours of operation. In this section, two cooperative interventions are explained in detail, in order to highlight how the HRI plays a central role in the entire intervention pipeline.

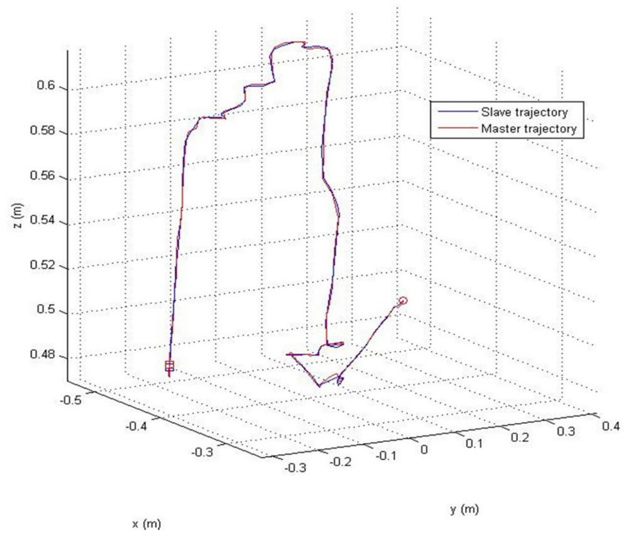


FIGURE 18. **The position error between master and slave of a trajectory performed during the intervention with variable communication delay.**

The requirements of these interventions made not possible the use of the industrial robots already owned by CERN, not only due to their lower usability but also to additional limitations, which will be explained in the detail in the next sections.

1) DUAL ARM MANIPULATION FOR BROKEN WINDOW REPAIR

It was requested to the robotic team to replace a glass window, damaged during physics operation (Figure 19(a)). The glass window was attached to the device through eight socket head screws. The operation did not need to put a new window in place. The intervention presented some challenges:

- The broken window was installed next to another fragile window of the same size, occluding partially the accessibility to the screws.
- The window was positioned at 1.80 m from the ground (Figure 19(b)); therefore, it was necessary to avoid the broken glass to fall and shatter in several pieces.

For the intervention the dual-arm configuration of the CERNbot robot [83] was necessary (Figure 19(c)). The owned industrial robots were not suitable for the intervention as they did not allow dual-arm manipulation and the use of two separate robots was not possible due to space constraints. Furthermore, the height of the broken window was at the limits of reachability of one of the two robots, making the operation safety-critical. In this dual-arm configuration of the CERNbot robot, one robotic arm was equipped with an electrical screwdriver (Figure 19(e)) and the other robotic arm was holding a plastic box (Figure 19(f)), just below the window in order to catch the falling screws and the window. The two arms were also raised in order to reach the proper height, reducing, however, the stability of the platform and requiring more precision during the telemanipulation.

In order to precisely position the electrical screwdriver inside the screw heads, avoiding the risk of damaging them, the master-slave interaction device with force feedback was used (Figure 19(d) and 18). The operator could also activate the electrical screwdriver through two buttons available in the yellow handle. One of the drawbacks of a single master-multiple slaves system is that when switching between the slaves, the master arm needs to move in order to match the slave position.

Nevertheless, as it can be seen in Figure 19(f), thanks to the high usability of the system, the operator can move the master arm with one hand without effort and a multiple masters-multiple slaves system can be put in place in future. Above all, the intervention was carried out in less than one hour without any problem.

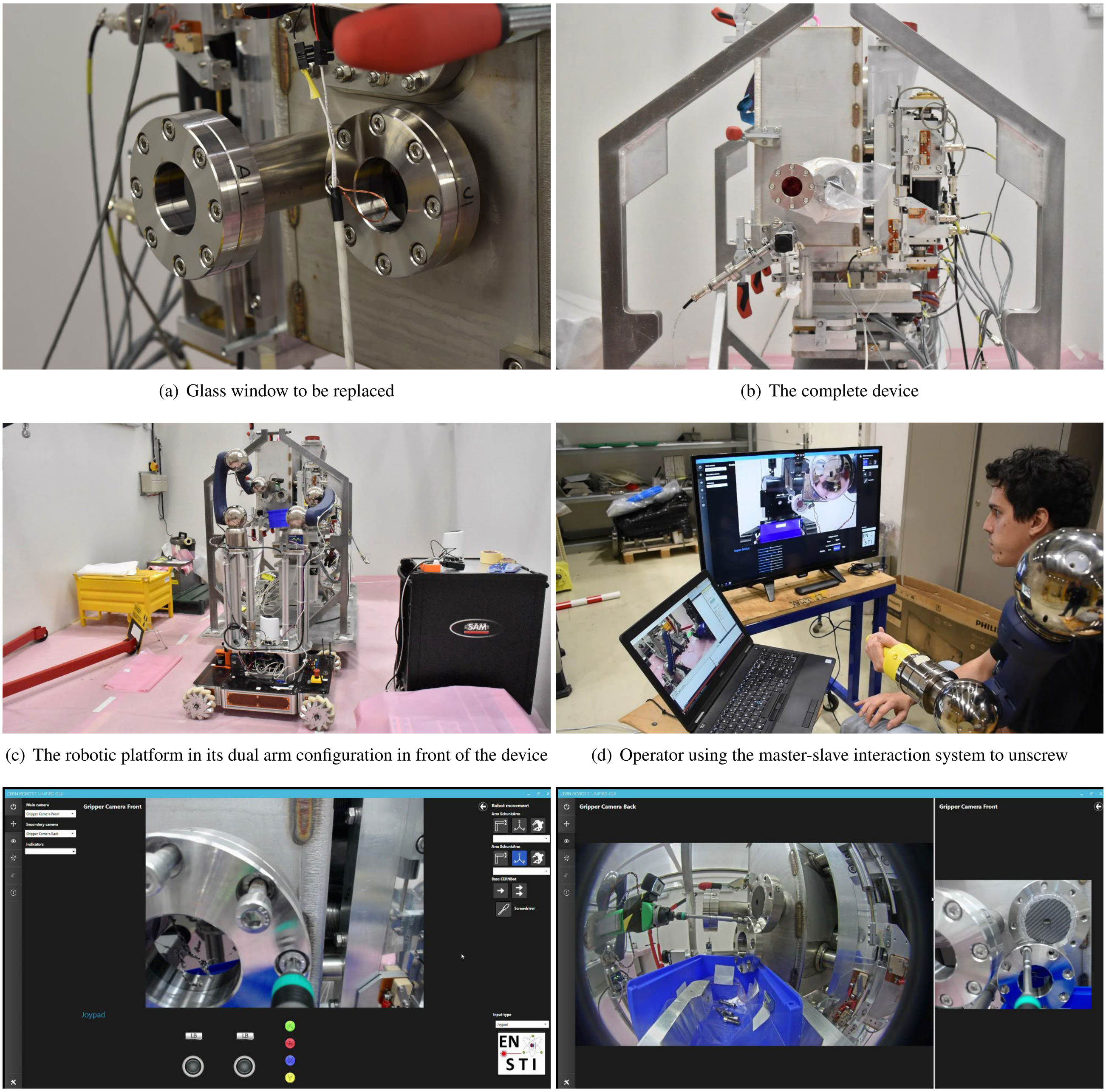
2) DUAL ROBOT INTERVENTION FOR WATER LEAK REPAIR

The robotic team was requested to repair a water leak in-situ that appeared in the cooling system of one accelerator component. The nature of the leak was allowing neither manual repair, due to environmental hazards, neither a remote cut and replace procedure, due to space constraints. It has been chosen, therefore, to create a metallic box around the leak and to use epoxy resin to stop it (Figure 20(d)). In order to build a metallic box easy to assemble around the water pipe, two sides of the box needed to be empty and sealed with expansive foam only once the metallic box was placed around the pipe. Moreover, in order to reduce the water leak rate which could have compromised the hardening of the epoxy resin, a water-reacting glue has been sprayed on the pipe using a syringe. The access to the leak was particularly complicated. For this intervention, a custom configuration of the CERNbot robot with one robotic arm has been built and a secondary robot has been used to give an external view to the operator (Figure 20(a)).

Above the space constraints already mentioned (Figure 20(b)), other intervention specifications did not allow the use of the two industrial robots: as they both rely on the radio signal, the position of the leak was too far from the first safe possible station for the operator. However, as already mentioned, the 4G signal is available all along the CERN's underground accelerators, putting no communication constraints for the custom developed robots. The other constraint was the impossibility of integrating the tools to be used (the resin pump, the foam sprayer gun and others) in the robot control to activate them remotely.

The accelerator component above the water leak is metallic, generating big reflections and occlusions for the 4G signal (Figure 20(c)). During the intervention, big variations in communication delay (above 500 ms) where measured, but the time-delay passivation technique explained earlier intervened properly allowing the completion of the task safely.

Furthermore, due to the complexity of the task, the accelerator tunnel has been covered by both robots several times in order to reach the first safe place for humans to subsequently adapt the robot to the intervention task. Due to the



(a) Glass window to be replaced

(b) The complete device

(c) The robotic platform in its dual arm configuration in front of the device

(d) Operator using the master-slave interaction system to unscrew

(e) An image from the interface while unscrewing

(f) An image from the interface from the second arm

FIGURE 19. **Different pictures from the broken window repair intervention.**

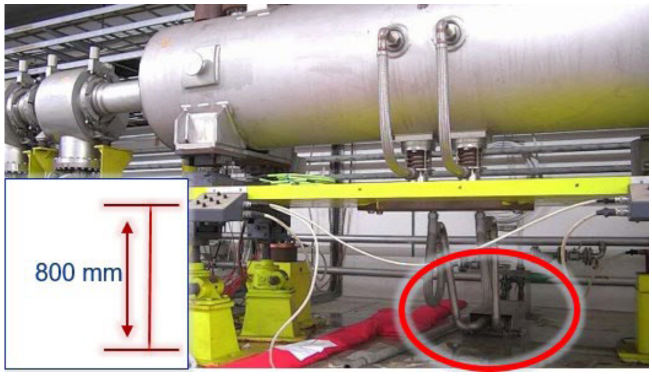
great distance to cover, the stress accumulated during the manipulation in the cluttered environment, and the number of times that the navigation was performed, the wall detection algorithm (Section V-B) appeared to be very useful to reduce the mental workload of the operator during this action and to increase the speed and the safety of the navigation.

C. FORCE FEEDBACK WITH TIME DELAY TESTS

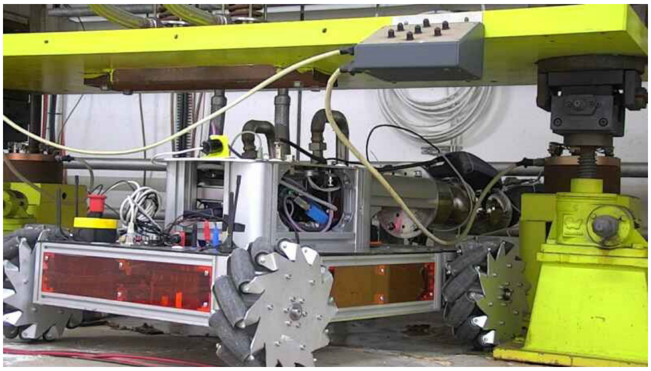
The performance of the master-slave system force feedback has been tested in the communication delay ranges taken into account in this work. The same trajectory has been tested multiple times simulating different communication delays between master and slave. During the trajectory, the slave arm was driven to contact with an external object, to measure the effects on the master arm of the force-feedback with increasing time delay. In figure 21 the behaviour of the control loop is presented. In order to simplify the visualization, the trajectory of a single joint has been taken into account.

It is visible how the contribution of the position feedback in the velocity set-point sent to the master arm joint is higher

(a) The two robots in the accelerator tunnel navigating to the water leak location seen from the safe human station



(b) The location of the water leak below the accelerator component



(c) The difficult accessibility of the CERNbot robot seen from the support robot

(d) The pouring of the resin in the built metallic box for the water leak fix

FIGURE 20. **Different pictures from the water leak in-situ repair intervention.**

and higher when increasing the time delay. This results in a more stiff master arm and a reduction of the overall master arm velocities. The position feedback reduces as well the overshoot which would be otherwise generated by the contact with an external object.

D. USABILITY AND LEARNABILITY TESTS

Various tests were performed in order to prove the usability and learnability of the presented HRI. As previously mentioned, above all the communication and safety features, the main goal of this work is to create a usable Human-Robot Interface which allows inexperienced operators to carry out telemanipulation tasks.

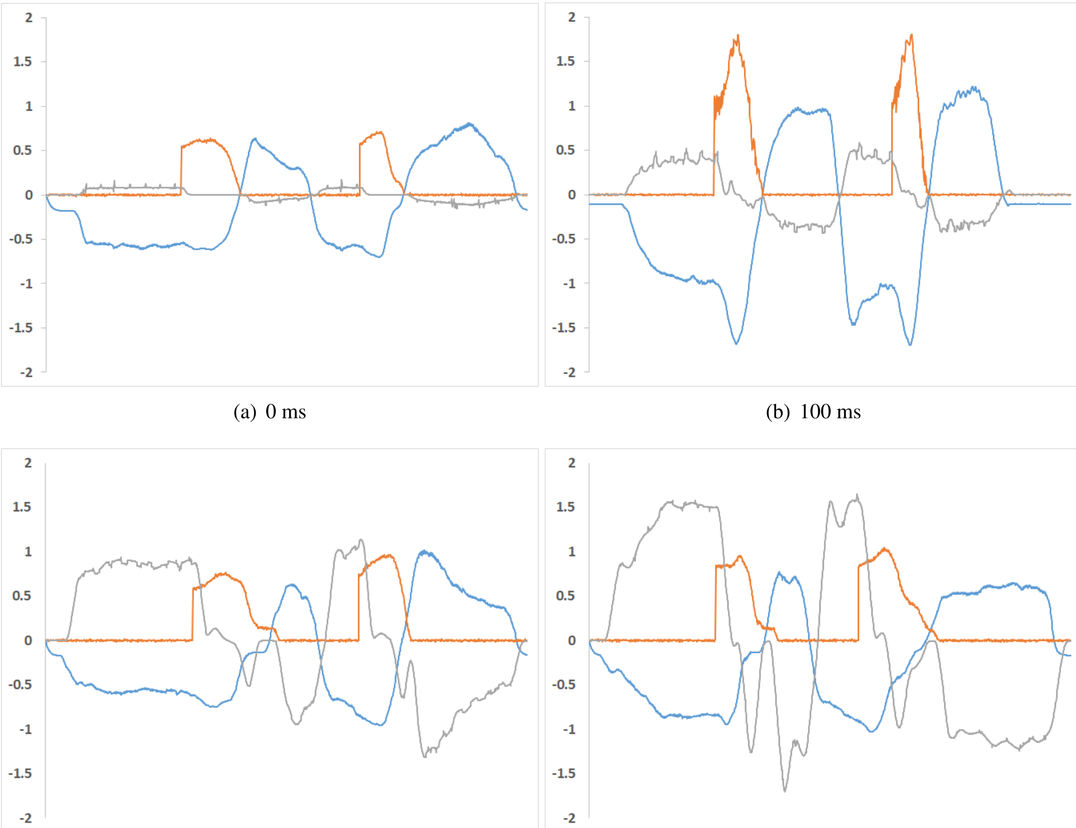
1) TELEMANIPULATION TASK

For this test, a set of inexperienced operators was selected to perform a unique task several times. The task had to be accomplished using either Telerob Telemax (figure 23(b)), one of the commercial service robots owned by the team, with its closed-box HRI, or CERNbot (figure 23(c)), the CERN in-house made robotic platform, with the proposed HRI. Each operator was asked to perform the task with only one of the two robots. The inexperienced operator was supported by an expert operator to explain the basic functionality of both robots before the first attempt and to provide minimal support during the entire test.

The operators were asked to accomplish the task several times, in order to compare the behaviour of the learning process between the two systems.

The operators were asked to pick a LEMO push-pull self-latching connector from a plastic box and insert it in its compatible plug (figure 23(a)). The performance of the task was measured by execution time. Moreover, in order to provide a baseline comparison time, the task was executed with both robots by expert operators; their accomplishment time can be considered as a physical lower bound in the execution of the task.

Figure 22(a) and figure 23(c)) shows the averaged execution time between the set of operators with respect to the number of attempts to perform the task using Telemax and CERNbot. It is evident how the operators' learning curve gets closer and closer to the average execution time of expert operators. However, the first attempt using CERNbot and the proposed HRI is almost 10 minutes faster than its counterpart using Telemax. This shows how the first impact with the proposed HRI is much more effective than the commercial one. Furthermore, after 10 attempts, an inexperienced operator using CERNbot almost matched expert operators' execution



(a) 0 ms

(b) 100 ms

(c) 250 ms

(d) 400 ms

FIGURE 21. **Master arm set-point components for a single joint. In blue the joint torque sensed on the master side, in orange the joint torque sensed on the slave side, in grey the position error feedback.**

time, showing how the system's learnability is higher using the proposed HRI.

These tests allowed the comparison between a commercial solution and the custom made robot. However, the custom made robot, as previously mentioned, allows helping the operator using already known information about the task to perform. In the case of this test, for example, the known vertical position of the connector allowed to create a script to help the operator to maintain the proper orientation of the manipulator. Figure 22(c) shows the result of this test. The first 11 attempts are the same as before. Later on, the operator was given the possibility to activate the orientation control script. It is evident how the execution time keeps decreasing, even below averaged expert operators' time.

Figure 22(d) shows the comparison of the execution time for inexperienced operators using Telemax and CERNbot.

In order to provide a numerical reference for future tests, the time-based efficiency of the task [84] has been calculated. The results are reported in table 5. Both in the case of expert operators and minimally trained operators, the time-based efficiency is higher when using the proposed interface. Furthermore, the time-based efficiency is higher for expert operators as well when using the proposed HRI. Besides, the expert relative efficiency has been calculated 6, which indicates which is the level of performance of an inexperienced operator with respect to an expert operator. In this case, minimally trained operators are much closer to expert operators when using the proposed interface.

TABLE 5. **Time-based efficiency of the manipulation tasks.**

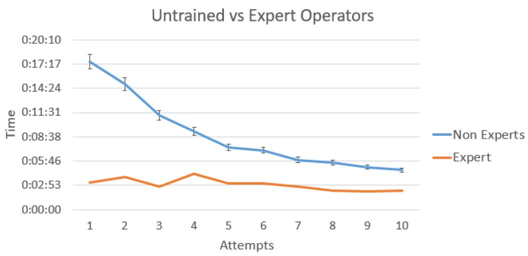
Robot	Expert operator	Inexperienced operator
Telerob Telemax	0.06 goals/min	0.3 goals/min
CERNBot	0.18 goals/min	0.48 goals/min

TABLE 6. **Expert-relative efficiency of the manipulation tasks.**

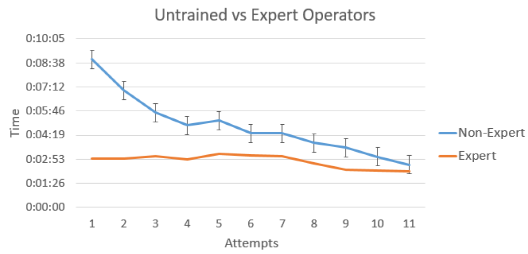
Robot	Expert relative efficiency
Telerob Telemax	39.4 %
CERNBot	51.5 %

2) NAVIGATION TASK WITH TIME-DELAY

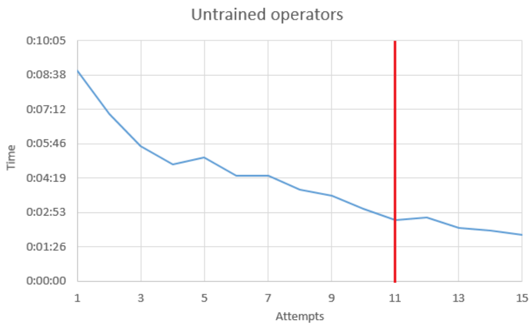
The time-delay passivation method using *one-way input devices*, presented in section V-C.2, has been tested in a



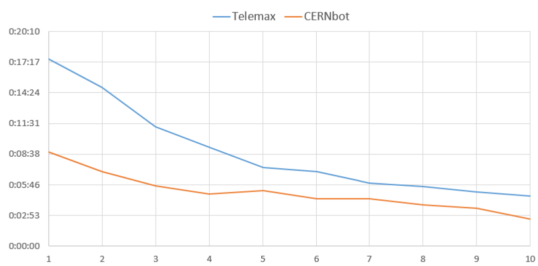
(a) Execution time over number of attempts using Telerob Telemax



(b) Execution time over number of attempts using CERN's CERNbot



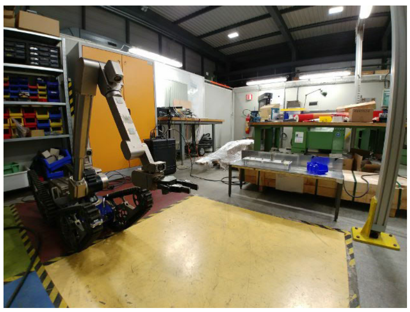
(c) Execution time over number of attempts using CERN's CERNbot for inexperienced operators. The red line indicates the activation of intervention scripts



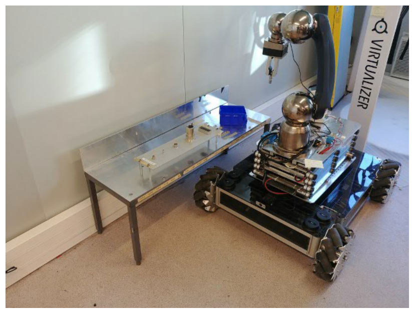
(d) Comparison of the execution time over number of attempts between inexperienced operators using Telerob Telemax and CERN's CERNbots

FIGURE 22. Usability tests performance charts.

(a) The experimental setup table



(b) Telerob Telemax used during the experiment



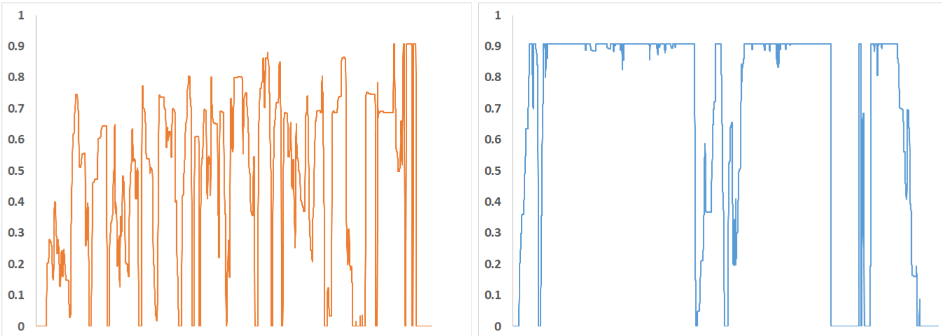
(c) CERN's CERNbot in the configuration used for the experiment

FIGURE 23. Different pictures from the usability tests.

navigation task. The operator was asked to navigate in a cluttered environment using the CERNbot platform, using one of the available interaction modalities. The test was carried out by simulating a highly variable communication delay between 200 and 400 ms and both with and without the proposed passivation, to analyse the control strategy chosen by the operator. The choice of this delay range is to exploit the threshold of 300 ms proposed in [74], in which an operator would decouple the commands from the robot's feedback. The velocity setpoint that the operator sends to the robot is shown in figure 24(a) in the case of no time-delay passivation applied, and in figure 24(b) with the proposed passivation. Only the linear velocity sent to the robot is presented in the charts. In the second case, the operator control is smoother than in the first case, in which a strategy of control-and-wait appears.

3) MULTI-AGENT MANIPULATION FOR TRANSPORT

In order to validate the use of the scripting for multi-agent control (section V-D), a transport task has been designed. The operators were requested to use two mobile platforms, each equipped with a single robotic arm (figure 25). The operators were asked to drive the two mobile platform in the proximity of the object to transport, grasp it and transport it back. The task has been performed multiple times for each operator, both with and without the scripting behaviours.



(a) Velocity setpoint from the operator without time delay passivation (b) Velocity setpoint from the operator with time delay passivation

FIGURE 24. Velocity setpoint comparison with high time delay.

TABLE 7. Time-based efficiency of the transport task.

	Time-based efficiency
With scripting	0.37 goals/min
Without scripting	0.12 goals/min

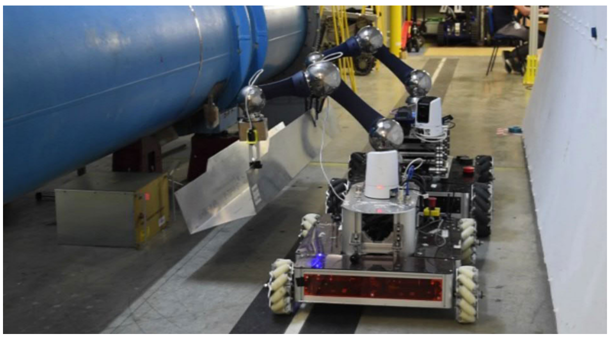


FIGURE 25. The two robots used for the multi-agent manipulation test while transporting the object.

Two scripts were programmed in advance for this task: one script was responsible for synchronizing the movement of the two mobile platforms, and another script was responsible for synchronizing the movement of the two manipulators. In the mission planner the following stages were added (figure 6(e)):

- synchronized approach the object with the two platforms
- rough synchronized approach using the two manipulators
- precise positioning of the manipulators independently
- synchronized grasping and lifting of the object
- synchronized recovery of the object using the two platforms

The results of the test are presented in table 7. The time-based efficiency of the task using scripting behaviours is more than three times higher than the efficiency without scripting. However this measure does not take into account the increased safety by using the scripting behaviours: once the two agents are mechanically engaged by the grasped object, it resulted particularly challenging for the operators to transport the object without dropping it. The low efficiency of the task without scripting is also caused by failures in the task execution. Differently, the synchronization of the movements through scripting allowed a safer transport of the object with a full success rate of the task for all the operators.

VIII. CONCLUSION AND FUTURE WORK

In this paper, a Human-Robot Interface for remote robotic intervention in hazardous environments has been presented. The proposed interface was designed modular, to ensure its adaptability to new robots and tasks, and multimodal, to provide high usability and efficiency even in multi-agent scenarios.

The HRI allows controlling a heterogeneous set of robots, by adapting itself to the robot configuration. The modularity allows to control as well multiple agents from a unique interface and to profit of advanced feature to increase multi-tasking and the overall control performance.

Furthermore, a big effort has been put in the development of different control modalities, to allow the operator to choose the most suitable control interface according to the needs. The operator can choose between one-way input devices, which do not provide any direct feedback and two-ways input devices which provide direct feedback during the control.

In order to increase the operator multi-tasking, control scripts can be programmed and executed at any moment during the operation. These control scripts are in addition to the already existing supervised control features and allow to program functionalities which are adapted to the task. The scripts are executed in the *model* context of the interface, and therefore have access to all the data coming from the robot as well as to all the control functionalities.

The HRI has been validated through several tests with both expert and inexperienced operators, and through more than 100 real interventions in CERN's facilities, demonstrating better usaability and adaptability to the tasks with respect to other owned commercial solutions. Furthermore, in this paper, we provided efficiency values for the current implementation of the interface, which can be used as a reference both for future and improvements and other works.

The work presented in this paper, as part of the CERN-TAURO project, is in continuous evolution according to new projects and requests. During this evolution, the HRI will continue to get new features and improvements with the goal of broadening the use of robots in the entire organization. Furthermore, in order to validate the concepts and techniques used in this work, additional tests outside CERN environment and the adaptation to different robots in other scenarios are foreseen.

REFERENCES

[1] A. Davids, "Urban search and rescue robots: From tragedy to technology," *IEEE Intell. Syst.*, vol. 17, no. 2, pp. 81–83, Mar. 2002. doi: 10.1109/5254.999224.

[2] A. K. B. Motaleb, M. B. Hoque, and M. A. Hoque, "Bomb disposal robot," in *Proc. Int. Conf. Innov. Sci., Eng. Technol. (ICISET)*, Oct. 2016, pp. 1–5.

[3] R. R. Murphy, "Rescue robotics for homeland security," *Commun. ACM*, vol. 47, no. 3, pp. 66–68, Mar. 2004. [Online]. Available: http://doi.acm.org/10.1145/971617.971648

[4] T. Yoshida, K. Nagatani, S. Tadokoro, T. Nishimura, and E. Koyanagi, "Improvements to the rescue robot quince toward future indoor surveillance missions in the Fukushima Daiichi nuclear power plant," in *Field and Service Robotics*. Berlin, Germany: Springer, 2014, pp. 19–32. doi: 10.1007/978-3-642-40686-7_2.

[5] J.-P. Friconneau, V. Beaudoin, A. Dammann, C. Dremel, J. Martins, and C. Pitcher, "ITER hot cell—Remote handling system maintenance overview," *Fusion Eng. Des.*, vol. 124, pp. 673–676, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0920379617300157

[6] D. J. Miller, *Handbook of Accelerator Physics and Engineering*, A. W. Chao, K.-H. Mes, M. Tigner, and F. Zimmermann, Eds., 2nd ed. 2014. doi: 10.1080/00107514.2014.885574.

[7] K. Flottmann, "The European XFEL project," in *Proc. JAERI*, 2005, p. 4. [Online]. Available: http://www.iaea.org/inis/collection/NCLCollectionStore/_Public/37/005/37005766.pdf

[8] C. Lefevre. (Dec. 2008). *The CERN Accelerator Complex*. [Online]. Available: http://cds.cern.ch/record/1260465

[9] P. J. Springer, *Military Robots Drones: A Reference Handbook*. ABC-CLIO, 2013. [Online]. Available: https://www.abc-clio.com/ABC-CLIOCorporate/product.aspx?pc=A3319C

[10] M. Di Castro, M. Ferre, and A. Masi, "CERNTAURO: A modular architecture for robotic inspection and telemanipulation in harsh and semi-structured environments," *IEEE Access*, vol. 6, pp. 37506–37522, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8391705/

[11] A. Dubey, "Evaluating software engineering methods in the context of automation applications," in *Proc. 9th IEEE Int. Conf. Ind. Inform. (INDIN)*, Jul. 2011, pp. 585–590.

[12] M. A. Goodrich and A. C. Schultz, "Human–robot interaction: A survey," *Found. Trends Hum.-Comput. Interact.*, vol. 1, no. 27, pp. 203–275, 2007. [Online]. Available: https://scholarsarchive.byu.edu/facpub

[13] D. Perzanowski, A. C. Schultz, W. Adams, E. Marsh, and M. Bugajska, "Building a multimodal human-robot interface," *IEEE Intell. Syst.*, vol. 16, no. 1, pp. 16–21, Jan. 2001. [Online]. Available: http://ieeexplore.ieee.org/document/1183338/

[14] T. B. Sheridan, *Telerobotics, Automation, and Human Supervisory Control*. Cambridge, MA, USA: MIT Press, 1992. [Online]. Available: https://dl.acm.org/citation.cfm?id=134291

[15] J. Scholtz, "Creating synergistic cyberforces in Alan C. Schultz and Lynne E. Parker (eds.), multi-robot systems: From swarms to intelligent automata," in *Proc. NRL Workshop Multi-Robot Syst.*, 2002, pp. 84–177.

[16] T. Zhang, B. Zhu, L. Lee, and D. Kaber, "Service robot anthropomorphism and interface design for emotion in human-robot interaction," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2008, pp. 674–679.

[17] R. R. Murphy and J. L. Burke, "From remote tool to shared roles," *IEEE Robot. Autom. Mag.*, vol. 15, no. 4, pp. 39–49, Dec. 2008.

[18] J. L. Drury, D. Hestand, H. A. Yanco, and J. Scholtz, "Design guidelines for improved human-robot interaction," in *Proc. CHI Extended Abstr. Hum. Factors Comput. Syst.*, 2004, p. 1540.

[19] C. A. Avizzano, "Human-robot interactions in future military operations," *Ind. Robot, Int. J.*, vol. 38, no. 5, 2011.

[20] A. T. Welford, "The 'psychological refractory period' and the timing of high-speed performance—A review and a theory," *Brit. J. Psychol. Gen. Sect.*, vol. 43, no. 1, pp. 2–19, 1952.

[21] C. D. Wickens, "Multiple resources and mental workload," *Hum. Factors*, vol. 50, no. 3, pp. 449–455, 2008.

[22] L. A. Jones and N. B. Sarter, "Tactile displays: Guidance for their design and application," *Hum. Factors*, vol. 50, no. 1, pp. 90–111, 2008.

[23] J. Scholtz and S. Bahrami, "Human-robot interaction: Development of an evaluation methodology for the bystander role of interaction," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 4, Oct. 2003, pp. 3212–3217.

[24] M. A. Goodrich and D. R. Olsen, "Seven principles of efficient human robot interaction," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Conf. Theme-Syst. Secur. Assurance (SMC)*, vol. 4, Oct. 2003, pp. 3942–3948.

[25] J. M. Riley, L. D. Strater, S. L. Chappell, E. S. Connors, and M. R. Endsley, "Situation awareness in human-robot interaction: Challenges and user interface requirements," in *Human-Robot Interactions in Future Military Operations*. 2010, pp. 171–192.

[26] C. Damiani, J. Palmer, N. Takeda, C. Annino, S. Balagué, P. Bates, S. Bernal, J. Cornellá, G. Dubus, S. Esqué, "Overview of the ITER remote maintenance design and of the development activities in Europe," *Fusion Eng. Des.*, vol. 136, pp. 1117–1124, Nov. 2018.

[27] Y. Noguchi, M. Saito, T. Maruyama, and N. Takeda, "Design progress of ITER blanket remote handling system towards manufacturing," *Fusion Eng. Des.*, vol. 136, pp. 722–728, Nov. 2018.

[28] S. Esqué, C. van Hille, R. Ranz, C. Damiani, J. Palmer, and D. Hamilton, "Progress in the design, R&D and procurement preparation of the ITER divertor remote handling system," *Fusion Eng. Des.*, vol. 89, nos. 9–10, pp. 2373–2377, 2014.

[29] D. Locke, C. G. Gutiérrez, C. Damiani, J.-P. Friconneau, and J.-P. Martins, "Progress in the conceptual design of the iter cask and plug remote handling system," *Fusion Eng. Des.*, vol. 89, nos. 9–10, pp. 2419–2424, 2014.

[30] G. Dubus, A. Puiu, P. Bates, C. Damiani, R. Reichle, and J. Palmer, "Progress in the design and R&D of the ITER in-vessel viewing and metrology system (IVVS)," *Fusion Eng. Des.*, vol. 89, nos. 9–10, pp. 2398–2403, 2014.

[31] R. Shuff, M. van Uffelen, C. Damiani, A. Tesini, C.-H. Choi, and R. Meek, "Progress in the design of the ITER neutral beam cell remote handling system," *Fusion Eng. Des.*, vol. 89, nos. 9–10, pp. 2378–2382, 2014.

[32] M. Saito, K. Anzai, T. Maruyama, Y. Noguchi, K. Ueno, N. Takeda, and S. Kakudate, "Development of radiation hard components for ITER blanket remote handling system," *Fusion Eng. Des.*, vol. 109, pp. 1502–1506, Nov. 2016.

[33] S. Sanders, "Remote operations for fusion using teleoperation," *Ind. Robot, Int. J.*, vol. 33, no. 3, pp. 174–177, 2006.

[34] XXXXX

[35] E. Potemkin, P. Astafurov, A. Osipov, M. Malenkov, V. Mishkinyuk, and P. Sologub, "Remote-controlled robots for repair and recovery in the zones of high radiation levels," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 1992, pp. 80–82.

[36] S. Kawatsuma, M. Fukushima, and T. Okada, "Emergency response by robots to Fukushima-Daiichi accident: Summary and lessons learned," *Ind. Robot, Int. J.*, vol. 39, no. 5, pp. 428–435, 2012.

[37] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Springer, 2016.

[38] D. Sands, "Cost effective robotics in the nuclear industry," *Ind. Robot, Int. J.*, vol. 33, no. 3, pp. 170–173, 2006.

[39] B. L. Luk, K. Liu, A. A. Collie, D. S. Cooke, and S. Chen, "Tele-operated climbing and mobile service robots for remote inspection and maintenance in nuclear industry," *Ind. Robot, Int. J.*, vol. 33, no. 3, pp. 194–204, 2006.

[40] P. Desbats, F. Geffard, G. Piolain, and A. Coudray, "Force-feedback teleoperation of an industrial robot in a nuclear spent fuel reprocessing plant," *Ind. Robot, Int. J.*, vol. 33, no. 3, pp. 178–186, 2006.

[41] S. Balaji and M. S. Murugaiyan, "Waterfall vs. v-model vs. agile: A comparative study on SDLC," *Int. J. Inf. Technol. Bus. Manage.*, vol. 2, no. 1, pp. 26–30, 2012.

[42] G. Lunghi, R. M. Prades, and M. D. Castro, "An advanced, adaptive and multimodal graphical user interface for human-robot teleoperation in radioactive scenarios," in *Proc. 13th Int. Conf. Inform. Control, Autom. Robot.*, 2016, pp. 224–231. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005971402240231

[43] J. Rasmussen, "Human errors. A taxonomy for describing human malfunction in industrial installations," *J. Occupat. Accidents*, vol. 4, nos. 2–4, pp. 311–333, Sep. 1982. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/0376634982900414

[44] J. T. Reason, *Human Error*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[45] J. W. Senders and N. P. Moray, *Human Error: Cause, Prediction, and Reduction*. Mahwah, Nj, USA: Lawrence Erlbaum Associates, 1991.

[46] F. W. Taylor, *Scientific Management*. Evanston, IL, USA: Routledge, 2004. [Online]. Available: https://books.google.ch/books?hl=es&lr=&id=3jXZpwWopf4C&oi=fnd&pg=PP1&dq=Taylor,+F.+W.+(2004).+Scientific+management.+Routledge.&ots=SCTfgkh6FI&sig=NvqY6e4Ag8JIx72wAEHhaDsDqJc#v=onepage&q=Taylor%2C%20F.%20W.%20(2004).%20Scientific%20management.%20Routledg

[47] A. Baddeley, "Working memory and conscious awareness," in *Theories Memory*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 1993, pp. 11–28.

[48] J. Waterworth and K. Hoshi, *Designing With Blends*. Cham, Switzerland: Springer, 2016, pp. 47–59. [Online]. Available: http://link.springer.com/10.1007/978-3-319-30334-5_4

[49] W. Farr, S. Price, and C. Jewitt, "An introduction to embodiment and digital technology research: Interdisciplinary themes and perspectives," Tech. Rep., Mar. 2012. [Online]. Available: http://eprints.ncrm.ac.uk/2257/

[50] D. Benyon and M. Imaz, "Metaphors and models: Conceptual foundations of representations in interactive systems development," *Hum.-Comput. Interact.*, vol. 14, no. 1, pp. 159–189, 1999. [Online]. Available: https://pdfs.semanticscholar.org/a214/73157c1da68460b2a653a19777c54fcdf5cf.pdf

[51] S. De Castell, L. Boschman, and J. Jenson, "In and out of control: Learning games differently," *Archives*, vol. 2, no. 3, pp. 1–16, 2009. [Online]. Available: http://journals.sfu.ca/loading/index.php/loading/article/viewArticle/66

[52] A. Leff and J. T. Rayfield, "Web-application development using the model/view/controller design pattern," in *Proc. 5th IEEE Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Sep. 2001, pp. 118–127.

[53] Y. Gu and R. L. Grossman, "UDT: UDP-based data transfer for high-speed wide area networks," *Comput. Netw.*, vol. 51, no. 7, pp. 1777–1799, 2007.

[54] P. M. Duvall, S. Matyas, and A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*. London, U.K.: Pearson, 2007.

[55] J. Jerosch and M. Prymka, "Proprioception and joint stability," *Knee Surg., Sports Traumatol., Arthroscopy*, vol. 4, no. 3, pp. 171–179, 1996.

[56] D. Hagner and J. G. Webster, "Telepresence for touch and proprioception in teleoperator systems," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-18, no. 6, pp. 1020–1023, Nov. 1988.

[57] T. Wijnands EN-RSO. *CERN ALARA Approach*. [Online]. Available: https://indico.cern.ch/event/164550/sessions/19960/attachments/197409/277083/ALARA_presentation.pdf

[58] G. Dumont, F. B. D. S. Pedrosa, P. Carbonez, D. Forkel-Wirth, P. Ninin, E. R. Fuentes, S. Roesler, and J. Vollaire, "Integrated operational dosimetry system at CERN," *Radiat. Protection Dosimetry*, vol. 173, nos. 1–3, pp. 233–239, Apr. 2017. [Online]. Available: https://academic.oup.com/rpd/article-lookup/doi/10.1093/rpd/ncw327

[59] D. Forkel-Wirth, S. Roesler, M. Silari, M. Streit-Bianchi, C. Theis, H. Vincke, and H. Vincke, "Radiation protection at CERN," Mar. 2013, *arXiv:1303.6519*. [Online]. Available: http://arxiv.org/abs/1303.6519

[60] G. Battistoni, F. Cerutti, A. Fassò, A. Ferrari, S. Muraro, J. Ranft, S. Roesler, and P. R. Sala, "The FLUKA code: Description and benchmarking," in *Proc. AIP Conf.*, Batavia, IL, USA, vol. 896, 2007, pp. 31–49. [Online]. Available: http://aip.scitation.org/doi/abs/10.1063/1.2720455

[61] J. C. Leader, "Atmospheric propagation of partially coherent radiation," *J. Opt. Soc. Amer.*, vol. 68, no. 2, pp. 175–185, 1978. [Online]. Available: https://www.osapublishing.org/abstract.cfm?URI=josa-68-2-175

[62] D. Meagher, "Geometric modeling using octree encoding," *Comput. Graph. image Process.*, vol. 19, no. 2, pp. 129–147, Jun. 1982.

[63] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auton. Robot.*, vol. 34, no. 3, pp. 189–206, 2013.

[64] Yann Collet. (2013). *Lz4: Extremely Fast Compression Algorithm*. [Online]. Available: https://code.google.com/

[65] U. Assarsson and T. Moller, "Optimized view frustum culling algorithms for bounding boxes," *J. Graph. Tools*, vol. 5, no. 1, pp. 9–22, 2000.

[66] M. Di Castro, C. V. Almagro, G. Lunghi, R. Marin, M. Ferre, and A. Masi, "Tracking-based depth estimation of metallic pieces for robotic guidance," in *Proc. IEEE Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 5503–5508.

[67] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, Mar. 2015.

[68] M. A. Fischler and R. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[69] D. L. Mills, "Internet time synchronization: The network time protocol," *IEEE Trans. Commun.*, vol. 39, no. 10, pp. 1482–1493, Oct. 1991.

[70] R. Lozano, N. Chopra, and M. Spong, "Passivation of force reflecting bilateral teleoperators with time varying delay," in *Proc. Mechatronics Forum*, 2002, pp. 24–26. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.6.7801

[71] N. Chopra, M. Spong, S. Hirche, and M. Buss, "Bilateral teleoperation over the Internet: The time varying delay," in *Proc. Amer. Control Conf.*, vol. 1, Jan. 2003, pp. 155–160. [Online]. Available: http://ieeexplore.ieee.org/document/1238930/

[72] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill, "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity," in *Proc. 1st Symp. Appl. Perception Graph. Vis. (APGV)*, 2004, p. 39. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1012551.1012559

[73] J. Corde Lane, C. Carignan, B. Sullivan, D. Akin, T. Hunt, and R. Cohen, "Effects of time delay on telerobotic control of neutral buoyancy vehicles," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 3, May 2002, pp. 2874–2879. [Online]. Available: http://ieeexplore.ieee.org/document/1013668/

[74] R. Held, A. Efstathiou, and M. Greene, "Adaptation to displaced and delayed visual feedback from the hand," *J. Exp. Psychol.*, vol. 72, no. 6, pp. 887–891, 1966. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0023868

[75] I. S. MacKenzie and C. Ware, "Lag as a determinant of human performance in interactive systems," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. (CHI)*, 1993, pp. 488–493. [Online]. Available: http://portal.acm.org/citation.cfm?doid=169059.169431

[76] K. Hashtrudi-Zaad and S. E. Salcudean, "Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation," *IEEE Trans. Robot. Autom.*, vol. 18, no. 1, pp. 108–114, Feb. 2002.

[77] R. R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 34, no. 2, pp. 138–153, May 2004.

[78] C. A. Miller, H. B. Funk, M. Dorneich, and S. D. Whitlow, "A playbook interface for mixed initiative control of multiple unmanned vehicle teams," in *Proc. 21st Digit. Avionics Syst. Conf.*, vol. 2, Oct. 2002, p. 7E4.

[79] P. Mitchell, M. Cummings, and T. Sheridan, "Management of multiple dynamic human supervisory control tasks," in *Proc. 10th Int. Command Control Res. Technol. Symp.*, May 2005, vol. 2, no. 1, pp. 1–3.

[80] (2008). *Unity Game Engine*. Accessed: Oct. 9, 2008. [Online]. Available: http://unity3d.com

[81] T. Hakulinen, P. O. Friman, E. van Uytvinck, D. Widegren, C. Delamare, T. S. Pettersson, and G. Fournier, "Integration of CATIA/smarteam into CERN's corporate engineering data management system," Tech. Rep. CERN-TS-2008-012, 2008.

[82] N. P. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IROS*, vol. 4, Sep. 2004, pp. 2149–2154.

[83] M. Di Castro, A. Masi, L. R. Buonocore, M. Ferre, R. Losito, S. Gilardoni, and G. Lunghi, "JACoW: A dual arms robotic platform control for navigation, inspection and telemanipulation," Tech. Rep., 2018.

[84] *Ergonomics of Human-System Interaction: Part 210: Human-Centred Design for Interactive Systems*, document ISO 9241-210:2010, Cham, Switzerland, 2009.

GIACOMO LUNGHI was born in Genova, Italy, in December 1990. He received the M.D. degree in software engineering from the University of Genova, in 2015. His master's thesis was on robot autonomous navigation developed for CERN accelerator's environment, and is currently developing his Ph.D. thesis at CERN, Geneva, Switzerland, in the matter of robotic teleoperation for accelerator's facilities in a joint collaboration with the Jaume I University of Castellon, Spain. He joined the CERN EN-SMM-MRO Section, in 2014, and has been part of CERN robotic development since then. In the matter of his work, he is involved in the entire robotic development pipeline, from hardware design, low-level control, robot intelligence, and robot communication to human–robot interaction. In 2018, he won the Best ''Application-Oriented'' Paper Award during the IROS 2018 Workshop in New Horizons for Underwater Intervention Missions: from Current Technologies to Future Applications Workshop.

RAUL MARIN received the B.Sc. degree in computer science engineering and the Ph.D. degree in engineering from the Jaume I University of Castellon, Spain, in 1996 and 2002, respectively. The subject of his Ph.D. was on the development of a supervisory controlled telerobotic system via Web, by using object and speech recognition, 3D virtual environments, grasping determination, and augmented reality. In 1996, he was with the Nottingham University Science Park, U.K., studying multimedia and simulation techniques for human-computer interfaces. In 1997, he joined Lucent Technologies (Bell Labs Innovations Research and Development), where he was a Researcher, a Software Developer, and a Software Architect with the Switching and Access Division. In 1999, he began to teach and research at the Jaume I University, as a Professor. Since 2009, he has been serving as an Associate Professor at the Department of Computer Science, Jaume-I University, Spain, where he lectures computer networking and robotics. He has teaching experience in Computer Science Engineering Degree, the Intelligent Systems Master's Degree, and the EU EMARO Advanced Robotics Master's Degree, among others. He was appointed as a Visiting Scientist with Blaise Pascal University, in 2002, University Polithecnique of Madrid, in 2007, University Federal of Brasilia, in 2016, and European Organization for Nuclear Research (CERN), in 2015, 2018, and 2019, studying new techniques for telemanipulation in hazardous environments. He has authored or coauthored more than 150 research publications on these subjects. He has participated in many research projects, such as FP6 Group of unmanned assistant robots deployed in aggregative navigation supported by scent detection (GUARDIANS), and FP7 Marine robots and dexterous manipulation for enabling autonomous underwater multipurpose intervention missions (TRIDENT) Project. His research interests include robotics, rescue, and underwater, including subjects, such as localization, networks of sensors and actuators, object recognition, telerobotics, and education.

MARIO DI CASTRO received the M.Sc. degree in electronic engineering from the University of Naples Federico II Italy. From 2007 to 2011, he was with DESY, involved in charge of advanced mechatronics solutions for synchrotron beamlines and industrial controls. Since 2011, he has been with CERN, and since February 2018, he leads the Mechatronics, Robotics, and Operation Section in the Survey, Mechatronics and Measurements Group. The section is responsible for the design, installation, operation, and maintenance of control systems on different platforms (PLC, PXI, VME) for all the equipment under the group's responsibility, mainly movable devices characterized by few um positioning accuracy (e.g. scrapers, collimators, shielding, and target) in hard radioactive environment. Important section activities are the robotic support in hazardous environments for the whole CERN accelerators. His research interests include automatic controls, mechatronics, and motion control in harsh en vironment and robotics.

ALESSANDRO MASI was born in Napoli, Italy, in July 1976. He received the M.D. degree in electronic engineering and the Ph.D. degree in computer science from the University of Naples Federico II, in 2001 and 2005, respectively. His Ph.D. research was carried out at the Group of Magnetic Measurements and Tests, CERN, on high-accuracy measurement systems for superconducting magnets of the new particle accelerator, the large hadron collider. In April 2005, he was a CERN Staff Member with the Group Sources, Targets, and Interactions Group STI. In January 2008, he takes the leadership of the Equipments Controls and Electronic (ECE) Section. Since February 2018, he has been the Head of the Group at CERN of Survey, Mechatronics and Measurements. The SMM Group develops and maintain a centralized competence in survey, mechatronic systems, test, and measurement. The group is in charge of maintaining a competence in the development of radiation tolerant electronics, and provides support CERN wide for radiation tests and radiation monitoring for evaluating the dose to electronics installed in radiation areas. The groups develop robotic platforms adapted to intervention in the accelerator environment, and deploy those solutions in collaboration with all groups in the accelerator and technology sector.

PEDRO J. SANZ received the B.Sc. degree in physics from the University of Valencia (UV), the M.Sc. degree in engineering (CAD/CAM) from the Technical University of Valencia (UPV), and the Ph.D. degree in computer engineering from Universitat Jaume I (UJI). He was appointed as a Visiting Scientist at different universities, such as TUM, Germany, in 2000 and 2016, Blaise Pascal, France, in 2002, Bologna, Italy, in 2008. He is currently a Full Professor with the Computer Science and Engineering Department, Universitat Jaume I (UJI), Spain, and the Head of the Interaction and Robotic Systems Lab (IRS-Lab). His main research interests include multisensory-based grasping and dexterous manipulation, telerobotics and human-robot interaction (HRI), all of them applied to real life scenarios, including assistive and underwater robotics. Since 1990, he has been active in research and development within several projects on advanced robotics (e.g. Coordinator of European FP7-TRIDENT Project, from 2010 to 2013) and continues (e.g. Coordinator of Spanish TWINBOT Project, from 2018 to 2020). He is the author or coauthor of a broad range of research publications and is an Active Member of the different scientific societies, such as the IEEE (RAS, SMC, and OES), EUCog, and euRobotics. He was a member of the Advisory Committee of the IEEE Systems Council (2008–2012) and the Humanoids Competition Chair in 2014 IEEE-RAS International Conference on Humanoid Robots, Madrid, in 2014. He was the Chair of several tutorials and workshops within outstanding international conferences on robotics (IROS, IFAC, and ICMA). He was a recipient of the Best Thesis in Electronics and Computer Science Domains, National Research Prize, from the Artigas Spanish Foundation (UPM, Madrid, Spain), in 1996. He has served as an Associate Editor for some outstanding journals, such as the IEEE RAM and the IEEE SMC Part C. He has been the Coordinator of the Spanish Robotics Network (CEA-IFAC), from 2012 to 2016.