IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 11 July 2022, accepted 2 August 2022, date of publication 16 August 2022, date of current version 24 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3198984

APPLIED RESEARCH

Mixed Reality Human–Robot Interface With Adaptive Communications Congestion Control for the Teleoperation of Mobile Redundant Manipulators in Hazardous Environments

KRZYSZTOF ADAM SZCZUREK[1,2], RAUL MARIN PRADES[2], ELOISE MATHESON[1], JOSE RODRIGUEZ-NOGUEIRA[1], AND MARIO DI CASTRO[1]

[1]European Organization for Nuclear Research (CERN), 1211 Geneva, Switzerland
[2]Department of Computer Science and Engineering, Jaume I University of Castellon, 12071 Castelló de la Plana, Spain

Corresponding author: Krzysztof Adam Szczurek (krzysztof.adam.szczurek@cern.ch)

ABSTRACT Robotic interventions with redundant mobile manipulators pose a challenge for telerobotics in hazardous environments, such as underwater, underground, nuclear facilities, particle accelerators, aerial or space. Communication issues can lead to critical consequences, such as imprecise manipulation resulting in collisions, breakdowns and mission failures. The research presented in this paper was driven by the needs of a real robotic intervention scenario in the Large Hadron Collider (LHC) at the European Organization for Nuclear Research (CERN). The goal of the work was to develop a framework for network optimisation in order to help facilitate Mixed Reality techniques such as 3D collision detection and avoidance, trajectories planning, real-time control, and automatized target approach. The teleoperator was provided with immersive interactions while preserving precise positioning of the robot. These techniques had to be adapted to delays, bandwidth limitation and their volatility in the 4G shared network of the real underground particle accelerator environment. The novel application-layer congestion control with automatic settings was applied for video and point cloud feedback. Twelve automatic setting modes were proposed with algorithms based on the camera frame rate, resolution, point cloud subsampling, network round-trip time and throughput to bandwidth ratio. Each mode was thoroughly characterized to present its specific use-case scenarios and the improvements it brings to the adaptive camera feedback control in teleoperation. Finally, the framework was presented according to which designers can optimize their Human-Robot Interfaces and sensor feedback depending on the network characteristics and task.

INDEX TERMS Collision avoidance, collision detection, communication, human-robot interaction, mixed reality, mobile robots, network bandwidth, point cloud, redundant manipulator, telerobotics.

I. INTRODUCTION

A. HUMAN-ROBOT INTERACTION AND TELEROBOTICS IN HAZARDOUS ENVIRONMENTS

Robotic interventions in hazardous environments are often required due to the presence of dust, fire, pressurized water or radioactivity. The robotic platforms must be reliable, and appropriate user interfaces should be present when an operator is performing remote inspection, maintenance or repair tasks. A consideration of the effort made by the operator and the necessity to avoid their cognitive fatigue must be taken, as this affects the safety and precision. Interventions in this kind of scenario are often performed using out-of-the-box robotic solutions, which cannot be adapted and improved according to the current requirements. An example are Remotely Operated Vehicles (ROVs), which have their

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

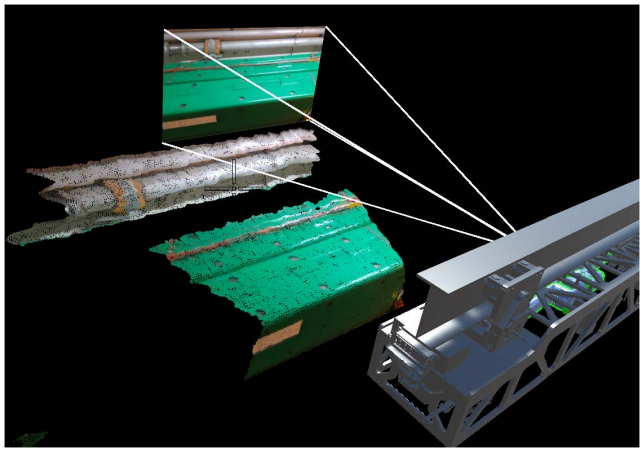

FIGURE 1. 3D perception of the environment, the robot model and video stream, combined in one interface. A fragment of a particle accelerator is captured by a RGBD camera installed on a robotic platform.

FIGURE 2. Virtual reality head-mounted device tested in the CERN teleoperation framework.

own communication system and a very specific user interface, which limits the number of tools, sensors, and devices that can be integrated and controlled.

Most systems rely on 2D-based Graphical User Interfaces, which have been demonstrated as very reliable when used by expert robotic operators, but should be improved when the expertise of the operator controlling the robot must be focused on the complexity of the environment instead [1]. This normally occurs in facilities where the real value of the operator is their knowledge of the machines and procedures that the robot will intervene on, and not their familiarity of the software interface. Interfaces could be enhanced by providing more information in a simple manner, for example, using Virtual, Augmented, and Mixed Reality techniques, as well as a 3D perception of the environment (see Figure 1, where additionally to a video stream, a point cloud of the environment and its relation to the robot can be seen). However, this enhancement requires the transmission of a larger quantity of information from the remote robot to the operator. The communication system behaviour can be affected enormously, especially when using wireless techniques in scenarios with a limited access to low-latency and high-bandwidth network, eg. underground, underwater or search and rescue missions in remote areas. The optimization and improvement of communication system is necessary in order to improve the user interaction experience.

Numerous telerobotic solutions have been used in hazardous intervention scenarios. In nuclear facilities, a critical mission which required a robotic teleoperation in a radioactive environment was the Fukushima Daiichi NPS survey of a radioactive water leakage [2]. During the mission, apart from dealing with radiation and using sufficiently resistant hardware components, the challenges also included the communication link characteristics. The need for portability, transportability and decontamination easiness of emergency response robots have been emphasized in [3] during nuclear facility accidents. The paper listed multiple robots that have been taken into account for such scenarios, however some of them were not suitable for the conditions at Fukushima. Regarding the user interface, they were controlled with a standard 2D GUI with video feedback.

An even more challenging endeavor is that of Mars surface exploration, where the Mars-Earth communication latency and low bandwidth are key obstacles for teleoperation, and for some tasks only supervised or autonomous control are possible. As written in [4], the round-trip communication delay ranged from 8 to 42 minutes for the Mars Sample Return mission. Space exploration also poses radiation and temperature risks which decrease electronics and computation efficiency. Meanwhile, the teleoperation of robots on the Moon from the Earth has to overcome the delay of 3 s. A new control approach for kinesthetic coupling under this extreme condition was studied in [5].

Remote handling in fusion research [6] is motivated by the radiation present near components that have to be installed, repaired or replaced. The robotic arms are mounted in a fixed position or on well defined movable platforms. With such structured systems, the network limitations are easier to overcome by connecting it with a well planned, high-bandwidth wired or wireless designated network.

Particle accelerator complexes present a challenge where strong safety protocols must be kept, access is limited and the installed equipment is aligned with micrometer precision. Any telerobotic operation must not only be able to execute tasks normally done by a person, but also do them safely for the accelerator itself, not to cause any damage to fragile devices. It means that the robotic systems require high-bandwidth wireless communication for the quality of the environmental perception, as well as precise localization of the robots in the tunnels. The techniques can be applied together in order to improve the whole performance of mission, and be combined with on-board artificial intelligence techniques for the situations where the robots might get constrained communication links [7], [8].

In Table A.1 in Appendix A, there is a non-exhaustive state-of-the-art of Human-Robot Interface products in terms of

hazardous environments, 2D and 3D interfaces, operational state, communication type, complexity of manipulator, level of interaction, collision avoidance or detection, adaptive communication for shared and dynamic networks.

B. MIXED REALITY INTERFACES FOR TELEROBOTICS

Mixed Reality can provide added value to telerobotic interfaces and control, for instance by providing more intuitive collision avoidance strategies. The main advantage of collision avoidance is improved safety, however this functionality must be properly managed and integrated to avoid overcomplicating teleoperations. As described in [10], there are multiple methods of presenting real objects in Virtual Reality. They can be depicted with the Chaperone method with boundaries, as a point cloud, or as Virtual Reality objects if they are recognized. The first method displayed only the boundaries of the objects, therefore only partial surface information was available. The second method provided more information but gave less immersion to the operator, because it could significantly block the virtual content display. However, this inconvenience could be mitigated by setting a maximum distance of the point cloud. The third method depended on the precision, availability of model and recognition of real objects and placing them correctly as virtual objects.

The underwater environment is another hazardous environment that presents multiple risks for humans: lack of oxygen if the life support fails or decompression sickness in the case of a rapid ascent. Therefore, underwater tasks such as infrastructure maintenance, recovery of benthic stations or disarming post-war explosives are often addressed with remotely operated submersibles. However, complex tasks require precise manipulation, localization and advanced teleoperation techniques. Some advancements in this field were presented in [9], where underwater robots cooperation and supervision with Virtual Reality system were presented.

The use of Mixed Reality can also help with the execution of complex tasks which require redundant robotic arms. The teleoperation of such arms becomes more complicated without any additional visual 3D or model representation. Moreover, the inverse kinematics management, pose awareness and control or understanding of the arm's movement may be possible only for highly experienced operators. Therefore, as presented in [11], the use of Mixed Reality could greatly improve the efficiency of teleoperation and provide more intuitive and immersive control methods. The inverse kinematics problem may not be solvable with standard techniques, therefore alternative methods could be used specifically for redundant arms, such as the Forward And Backward Reaching Inverse Kinematics (FABRIK) heuristic method introduced in [12]. However, in this method the subsequent movements of the teleoperator arm have the impact on the final pose. This is why the continuous visual and spatial feedback presented to the operator is important.

Mixed Reality Interfaces are still in their early development and research phase, although there have been attempts to standardize them and their design process. As an example, an extensive study of Virtual Reality Human-Robot Interface for underwater robot operation was done in [9], where types of controllers, headsets, simulation engines or frameworks and computer components were tested and characterized. A link between the interface usability and ISO usability standards was established, which lays the foundation for more methodical approach of the Human-Robot Interface design which must meet specific requirements of performance and user satisfaction.

The authors of [13], [14], and [15] proposed a taxonomic framework classifying Mixed Reality Visual Displays. In Figure 3, the currently used CERN Robotic 2D Graphical User Interface (Figure 4) and the Mixed Reality Human-Robot Interface have been classified according to this (black points). Additionally, two next generation interfaces (Mixed Reality using Virtual Reality Head-Mounted Device - Figure 2, and Mixed Reality using holographic Head-Mounted Device) currently being developed are classified (white points). The Virtual Reality and holographic Head Mounted Devices improve the Presence Metaphor. Finally, an ideal Mixed Reality interface using holographic Head-Mounted Devices, with master-slave control and force feedback, as well overlaying the fully modelled environment with a fully registered point cloud, would achieve an interface of unmediated reality, as described in Section 5.3 of [14].

C. COMMUNICATION PROTOCOLS AND POINT CLOUD STREAMING

Mixed Reality Interfaces are based on a 3D representation of the environment, therefore a reliable and fast communication becomes a very important success factor. In this section the topic of communication protocols and point cloud streaming is explained.

The authors of [16] described end-to-end congestion control transport protocol solutions designed for Internet Telerobotics, where time-varying transmission delay and non-guaranteed bandwidth were problems. Since the devices needed to connect over the Internet, they had to manage an IP protocol. It was found that the TCP protocol was better for high-level commands sent with low frequency, where reliability was more important, while the UDP protocol was better for low-level control, where transmission delay played a bigger factor. Therefore, the paper shows the necessity to build a specific UDP-based teleoperation protocol for low latency and adjusted congestion/flow end-to-end control.

Maritime search and rescue communication solution for Unmanned Aerial Vehicles (UAVs) described in [17] was based on Long Term Evolution (LTE) technology for high volume traffic and LoRa for low-rate telemetry. Due to the very high latency and reliability requirements for teleoperated UAVs, the proposed solution was application-aware for mission-critical traffic and users. A prioritizing scheduling strategy was a trade-off between network latency, reliability, and capacity.

An adaptive point cloud streaming compression algorithm was proposed in [18], which adapted to the current

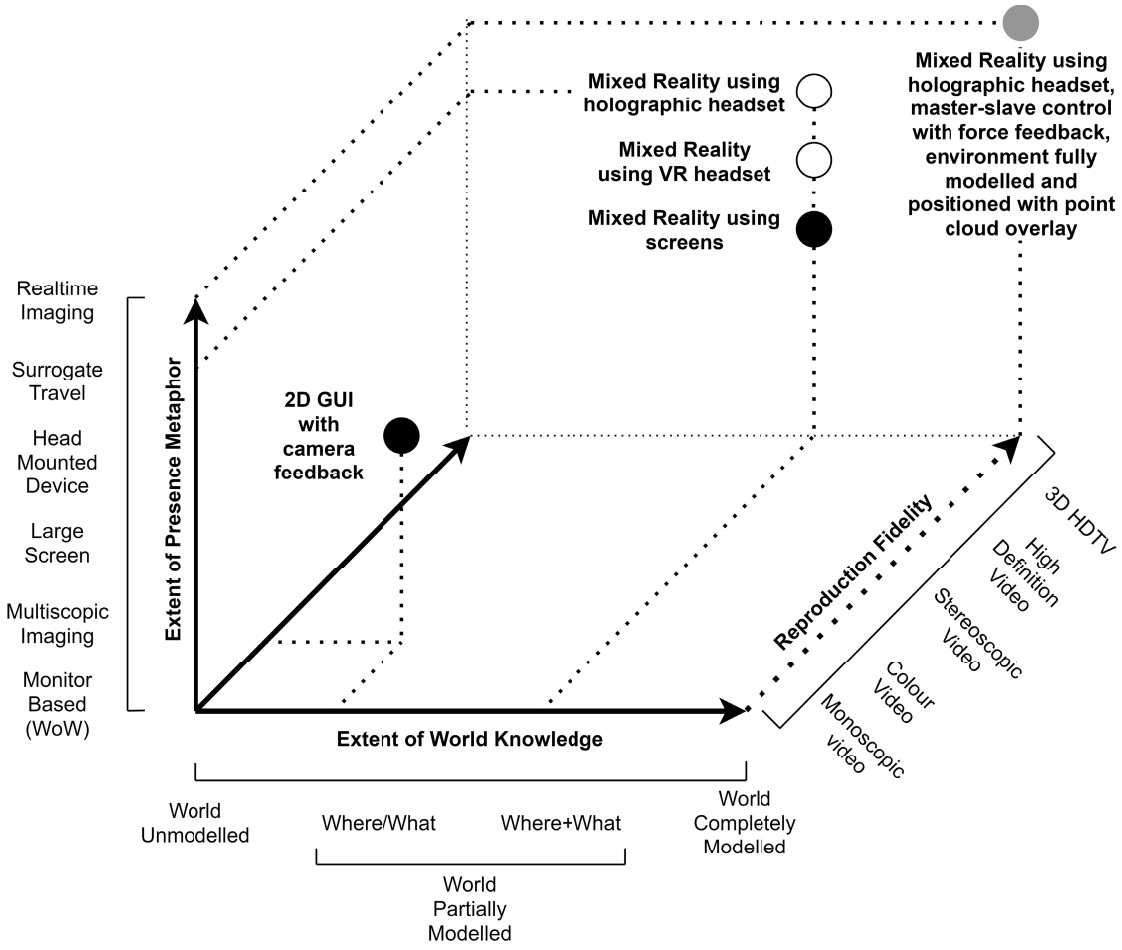

FIGURE 3. Placement of currently used, or being developed, or an ideal CERN mixed reality human-robot interface in the three dimensional taxonomic framework for classifying Mixed Reality displays.

FIGURE 4. CERN Robotic 2D Graphical User Interface used in a real intervention scenario.

available network data rate. The algorithm operated on an ongoing point cloud stream instead of compressing a full point cloud, which avoided additional delay. In the paper, current state-of-the-art point cloud compression solutions were mentioned: Google Draco, MPEG codecs [19], machine learning approaches [20], [21] and the traditional octree [22] or kd-tree based approaches. Point cloud streaming ideas were also described. The first one was the DASH-PC [23], which sent different parts of a point cloud depending on the user's current view and the available network data rate. The second one was adapting voxel length according to the available network data rate [24], [25].

A method using layered structures of point cloud transmission depending on communication channel conditions was described in [26]. The source encoded point cloud was converted into a layered structure where deeper layers hold finer point cloud representations. If the channel conditions were bad, only the upper layers were sent, which gave less detailed representation, while if the conditions were good, all the point cloud without any loss (represented by all layers) was sent.

D. CERN HUMAN-ROBOT INTERFACES EVOLUTION

Robotic interventions require a great level of modularity to achieve solutions in different scientific, hazardous and semi-structured scenarios. This modularity should be provided in both the mechatronics design (i.e. various mechanical configurations, as can be seen in Figure 5) and also

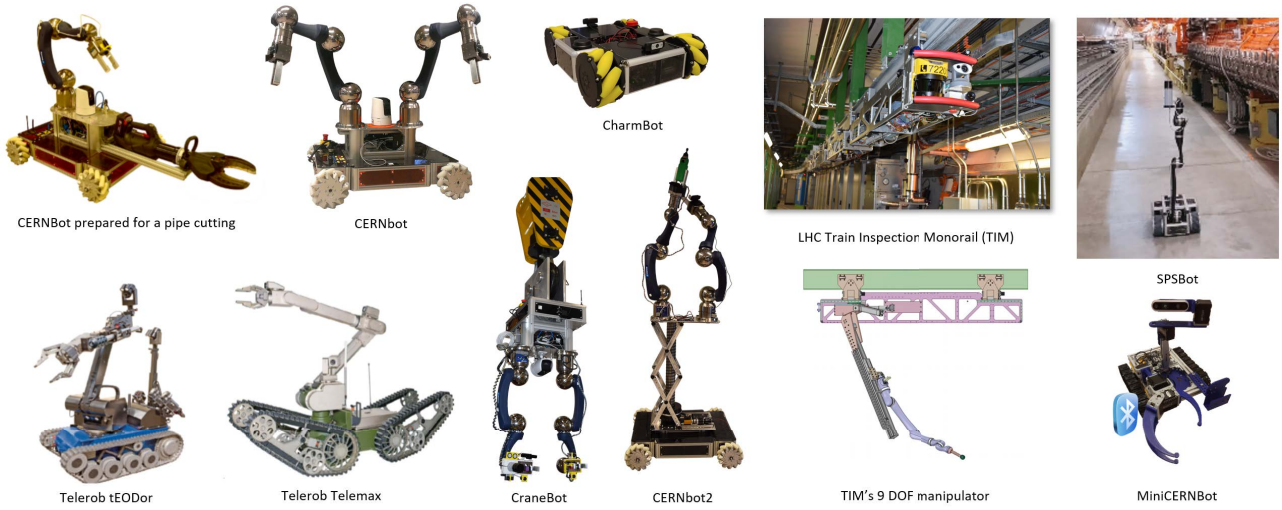

FIGURE 5. Robots used at CERN, consisting of industrial commercial robots (Telerob) and custom-made solutions (CERNBots [27], CharmBot, CraneBot, TIM [28], SPSBot, MiniCERNBot [29]).

the software architecture. The CERN CERNTAURO robotic framework [30] and [31] provides multiple autonomous and supervised teleoperation techniques. It has been tested during the last 8 years in more than 150 real interventions, 500 performed tasks and 500 hours of operation, that required multimodality [32] and flexibility. The framework contains functionalities such as master-slave interaction with time-delay monitoring, multimodal user interface, passivity control when the delay exceeds limits, vision and SLAM-based operation. The system was extended with cooperative multi-robot teleoperation [33].

The cooperative behaviour was tested with the CERN-TAURO project at CERN and with the TRIDENT/TWINBOT projects [34], [35] in an underwater scenario, and proved that one operator can safely handle an intervention with several robots simultaneously. The paper introduced a scripting feature, which allows the operator to supervise vision and artificial intelligence semi-automated behaviours to complement the manual teleoperation. It was especially helpful when multiple robots had to perform a complex task together, such as transporting or assembling a big object. As described in [36], the system was enriched with a depth estimation algorithm to be used with monocular cameras, such as a laparoscopic camera, which have been used in the design of grippers and tools for specific interventions.

Various phenomena affecting the operator were studied in [32]. For example, during the intervention, an operator could get distracted and lose focus on the relevant part of the screen. To track the operator focus, an eye tracking system was employed. An eye-tracking camera checked the position of the pupils and sent the information to a software interlocking movement if the operator was looking elsewhere. Moreover, in the paper it was verified that the real-time feedback from the robot's environment was crucial in remote teleoperation. It concluded that crossing

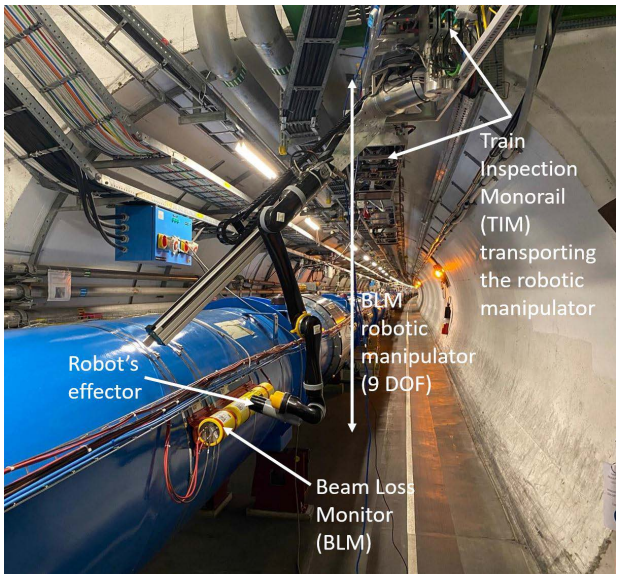

FIGURE 6. Beam loss monitor robotic arm installed on the train inspection monorail in the LHC.

a 300 ms delay threshold should decouple the commands from the robot feedback. As presented in [37], a preliminary study of Mixed-Reality Human-Robot Interface was done and a pilot project was tested in a real intervention scenario inside the LHC accelerator testing the particle beam at CERN. In the scenario, the Beam Loss Monitors robotic measurements were performed [38], where a redundant manipulator (Figure 6) using different trajectories required full perception of the robot's pose. The redundant manipulator was installed on a mobile robotic platform - the Train Inspection Monorail - mounted on a ceiling rail in the LHC (Figure 5).

E. MOTIVATION AND PROBLEM FORMULATION

Virtual, Augmented and Mixed Reality techniques in Human-Robot Interfaces have been used, however the problems of high positioning precision, collision avoidance and network delay/bandwidth constraints for mobile robots with redundant manipulators in hazardous intervention scenarios have not been fully addressed using these techniques.

Currently, there are no accepted standards or frameworks according to which designers can optimize their Human Robot Interfaces depending on the network characteristics. This paper seeks to present a framework that can be adapted to the needs of the interface. The application is given to redundant manipulators in hazardous intervention scenarios where there is a need for high positioning precision and collision avoidance while actively managing network delays/bandwidth constraints.

Much of the state-of-the-art research in Mixed Reality Human-Robot Interface controlling robots is done in controlled laboratory scenarios. However, in reality, a bigger margin of uncontrolled parameters can be expected. In this paper, emphasis is put firstly on the communication network variability, limitations and accessibility and secondly on the consequences of collisions, prioritizing safe operation with maximum possible collision detection and prevention. Both self-collisions and collisions with the environment must be checked. Thirdly, the interface, although providing much more immersion and functionalities in the teleoperation scenario, must still provide very precise positioning of the robotic arm in the industrial and scientific scenario. The positioning precision should be in the range of 1 mm for contact interaction with small elements, such as screws, or in the range of 10 mm for non-contact tasks that avoid collisions and operate, for example, a camera.

The most important aspect of the teleoperation in a hazardous environment is safety. In the Beam Loss Monitors (BLM) measurements project (Figure 6), the manipulation is performed in a very specific environment containing crucial equipment used for the LHC operation, and any collision could cause a major failure of the Collider. The robotic manipulator also carriers a radioactive source which is used to trigger and validate the Beam Loss Monitor sensors. This part is delicate, and contact with the environment should be avoided. The requirements for such missions are as follows:

- Multiple waypoints with complex manipulator configurations that can be recorded and modified;
- Collisions or dangerous close passes check during motions between waypoints. Warning about manipulator's self-collision or a collision with the environment;
- Information on the distance from the end effector to any captured point of the environment displayed during planning or real-time movements;
- The verification of all these conditions from any operator viewpoint in the scene;
- Feedback: internal status of the robot, its 3D pose visualization, and real-time environment view.

Although the LHC is partially modelled, it is not practically possible to position the robot in the modelled environment due to the localization error of the robot and the fact that not all the areas or details have been modelled. This is why the most robust 3D feedback is the real-time point cloud. The necessity to transmit 3D information increases required bandwidth, which is challenging in the 4G shared network with variable delays and bandwidth parameters. The available bandwidth depends on the position in the accelerator and current network use by different devices.

In terms of point cloud transmission, the reliability of the transmission is important (e.g. TCP protocol is more suitable than UDP protocol, as discussed in the state-of-the-art). If a part of the transmission is corrupted, the fragment or even the whole point cloud area will not be received. Some compression mechanisms can be applied and binary format of point cloud can be used to lower the transmitted size, however the biggest impact on throughput is the size of the point cloud (3D area acquired by the camera), the precision (distance between points) and the update frequency (point cloud frame rate). Therefore, application level of the end-to-end congestion control is needed. This was the main motivation of the study presented in this paper regarding point cloud acquisition and automatic setting algorithms that adapt to network conditions. Since the robot shares a public network and the point cloud communication parameters are related to the point cloud acquisition and transmission, where the network cannot control it (the parameters are beyond TCP stream), the interface needs an application-level, end-to-end and multimodal (from manual to supervisory) congestion control of point cloud streaming.

F. PAPER STRUCTURE

The paper is structured as follows: Section II describes the developed Mixed Reality Human-Robot Interface system, Section III describes the experimental setup. The results are presented in Section III-B and discussed in Section IV. Finally, Section V concludes the work.

II. SYSTEM DESCRIPTION

The system uses multimodal teleoperation techniques, from low-level manual control to high-level commands. It includes velocity control of the manipulator in real-time; trajectory specification and motion planning; target-centered task specification (e.g. aligning the end-effector to a normal line from a point cloud perceived object); and collision avoidance or detection. These strategies are applied to the Mixed Reality Human-Robot Interface for a mobile redundant manipulator operated under challenging communication conditions in hazardous intervention scenarios, characterized in Sections I-E and III-A1. As the video and point cloud feedback are crucial, an adaptive congestion control of this feedback was implemented in the interface to provide reliable automatic behaviour during operations.

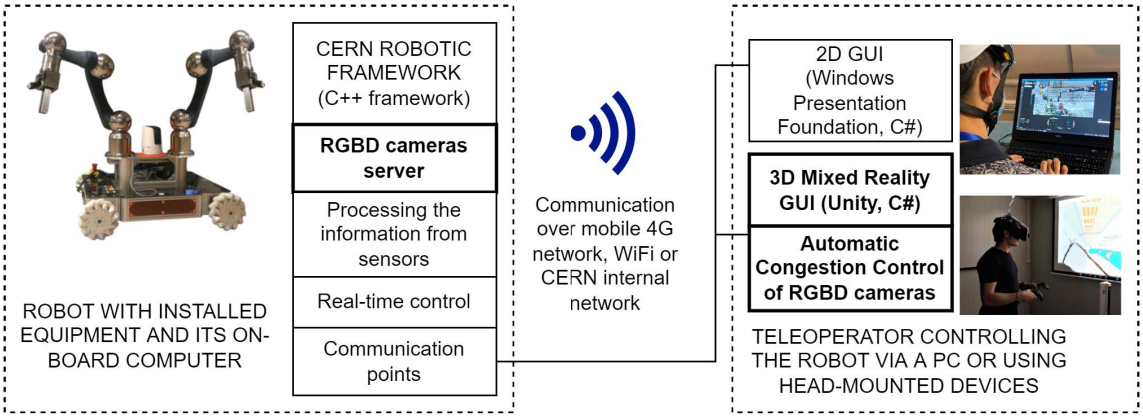

FIGURE 7. Robot-GUI architecture presenting processes that run on the robot and how the connection is established with the user interfaces. The parts which are focused on in this paper are highlighted.

A. OVERALL SYSTEM ARCHITECTURE

The robot is located in a remote area which is usually hazardous for a human. It has an onboard computer that communicates with all physical devices (e.g. motor drivers, cameras, sensors), and runs the CERN robotic framework that processes all the sensory information, controls and internal/external communication. The operator sits in a control centre and connects to the robot over the available network from a Graphical User Interface (GUI). Currently there are two types of interfaces developed for CERN purposes, the 2D GUI and 3D, Mixed Reality GUI. The 2D GUI has been developed using the Windows Presentation Foundation Technology and programmed in C# programming language. The 3D GUI has been developed in Unity, also in C#. Using the same language allowed the core communication and abstraction layers between the two GUIs to be shared. The two GUIs can and usually run in parallel as they provide different functionalities. In this paper, the CERN Mixed Reality Human-Robot Interface is presented and all the experimental data has been produced with its use. A diagram presenting the robot-GUI architecture is shown in Figure 7.

Depending on the operational setup, the robot and GUIs can communicate via a 4G network, WiFi or the cabled network infrastructure. However, the 4G network is mostly used for the reasons described in Section III-A1. Standard robotic intervention use cases and sequences of the functionalities are presented in Appendix B.

B. MIXED-REALITY HUMAN-ROBOT INTERFACE ARCHITECTURE

The global software architecture is shown in Figure 8. It specifies modules responsible for distinct functionalities and how they interact with each other. The main modules of the software are:

1) Arm control (architecture view in Figure 12 and description in Section II-D);
2) Trajectory planning (description in Section II-E);
3) Closest end effector position (description in Section II-H);
4) Target approach with a normal point (description in Section II-G);
5) Collisions (collision integration in the arm control is shown in Figure 12 and description is in Section II-F);
6) Menu (description in Section II-C4);
7) Camera control and automatic settings algorithms (description in Section II-I);
8) Camera hologram (description in Section II-I3);
9) Player movement;
10) Head-Up Display (architecture view in Figure 10 and description in Section II-C1);
11) Network measurements (description in Section II-J).

C. INTERACTION MODES

The user has the possibility to interact with the following interfaces:

- Head-Up Display;
- Holograms in the scene;
- Scene objects interaction;
- Menu interaction.

The executions of commands continues only if a corresponding confirmation key is kept pressed. This ensures a degree of safety and allows fast cancellation of a given command. For example, during a joint control mode, when a key is pressed, the program periodically sends a command to move the joint with a selected speed. As soon as the key is released a stop command is sent. Similarly, an execution of trajectory following is done as long as the key is pressed, and is stopped if key is released.

1) HEAD-UP DISPLAY

The Head-Up Display (HUD) allows the operator to have critical information constantly available during teleoperation. The HUD is shown in Figure 9 and its components in Figure 10. It displays the current connection status to the robot, current control mode of the manipulator and speed

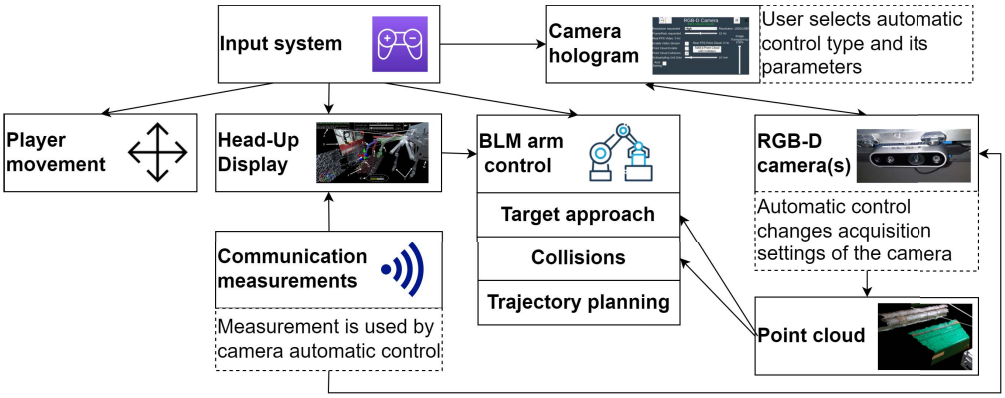

FIGURE 8. CERN mixed reality human-robot interface architecture overview. The components taking part in the camera automatic control has additional description of their role in the end-to-end application level congestion control.

settings (control modes are described in Section II-D), current usage of available bandwidth or round-trip time to estimate the control delay to be expected (network measurements are described in Section II-J), or the Mixed Reality simulation FPS (i.e. frames per second).

2) HOLOGRAMS IN THE SCENE

Some controlled elements of the robot have detailed and numerous settings that the operator must set during the intervention. An example is a camera which has ∼20 parameters to set or display simultaneously. Showing them on the Head-Up Display would take too much space and the information could overload the operator. Using a menu screen would hide the scene and camera video or point cloud feedback. Therefore, a solution of holograms placed in the scene was implemented. The hologram can be seen in Figures 23 and 24. The operator can interact with the hologram from any position in the scene.

3) SCENE OBJECTS INTERACTION

Various objects in the scene can be interactive. The arm target position (Point I in Figure 9) can be moved and the arm follows it. A hologram can be enabled by clicking on the 3D model of the camera. This type of interaction is more intuitive and plays a major role while using a Virtual Reality or Augmented Reality headsets with hand controls.

4) MENU INTERACTION

Certain functionalities require complex structures of information or parametrization. As an example, for the current interface, the menu presents a network measurements screen (Figure 11), which can request bandwidth measurements or observe network behaviour and values during operations.

D. MANIPULATOR CONTROL

The robot arm can be controlled via 2 methods. The motion can be planned and then executed, or the arm can be controlled directly in real-time. Moreover, the control in each of these methods is also multimodal. The arm can use inverse kinematics to approach a desired point by the end-effector or each joint can be controlled separately. The inverse kinematic used in planning is based on the FABRIK algorithm [12] which allows certain joints to be blocked, and during real-time control, the Jocobian-based inverse kinematic is used and computed directly by the on-board computer. For moving between trajectory positions, the position PID with velocity feedforward control is used.

The arm can be controlled in 4 control modes:

1) Real-time direct joint control;
2) Real-time inverse kinematic control;
3) Planning with separate joint control. The speed of angular and linear movement can be adjusted;
4) Planning with FABRIK inverse kinematic [12] control, where the last 6 joints follow the target. One or more of the joints can be blocked to change the behaviour of the inverse kinematics and approach.

E. TRAJECTORY SPECIFICATION

As presented in Figures 9 and 13, the operator can create a trajectory to reach the target. The 3D representation of the environment given by the point cloud feedback facilitates the manipulator's path planning by the operator or any other path planner. Path planner specifications are generalized for this framework. The waypoints can be saved, modified or removed on demand. Afterwards, a waypoint can be selected, a movement preview can be launched to check any potential collisions (more in Section II-F1), and finally the movement to the selected waypoint can be initiated.

F. COLLISIONS

The collision avoidance or detection plays a very important role in the CERN Mixed Reality Human-Robot Interface. The safety during the intervention in the LHC and with the use of the radioactive source is the most important aspect, and any collision or unplanned contact between the source and the environment should be avoided. While self-collisions

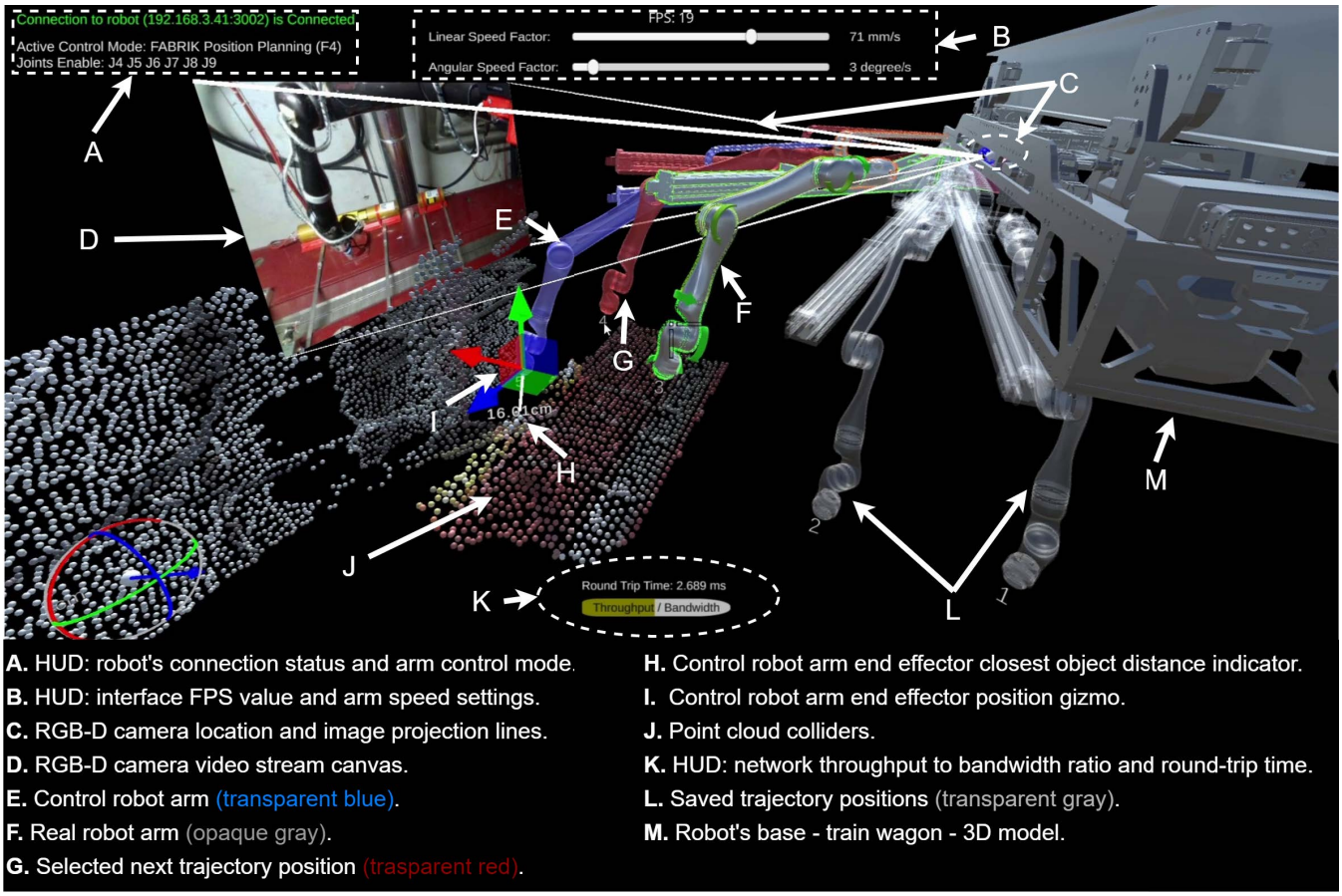

FIGURE 9. CERN mixed reality human-robot interface - operator's view from an LHC intervention scenario.

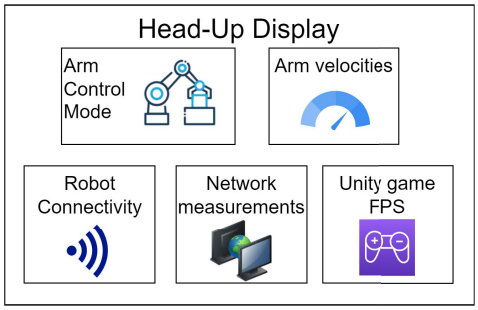

FIGURE 10. Head-Up display components overlaid on the scene content. It is visible at all times during the teleoperation.

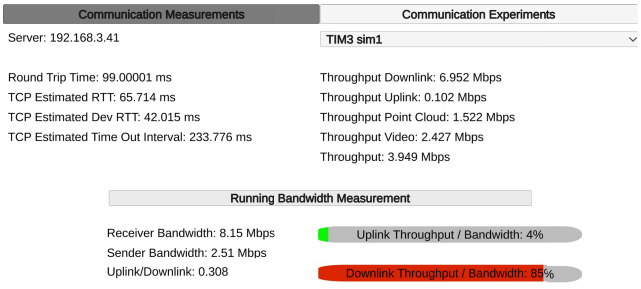

FIGURE 11. Communication measurements screen. It shows all network parameters related to the general communication with the robot and the camera-specific acquisition data transfer information.

can be recognized using the robot model, the collisions with the environment require environmental spatial information. The functionalities described in this section require 3D point cloud feedback, whose parameters are controlled by the adaptive congestion control. Therefore, this automatic control facilitates the reliable collision avoidance or detection. The interface implements the following functionalities:

- Collision avoidance during planning and preview of movements (Section II-F1);
- Real-time collision detection with joint torques (Section II-F2);
- Real-time collision detection of real arm with itself or point cloud (Section II-F3).

They assist the operator by preventing the movement if any recognized collision may happen or limiting the consequence of a collision if it occurs.

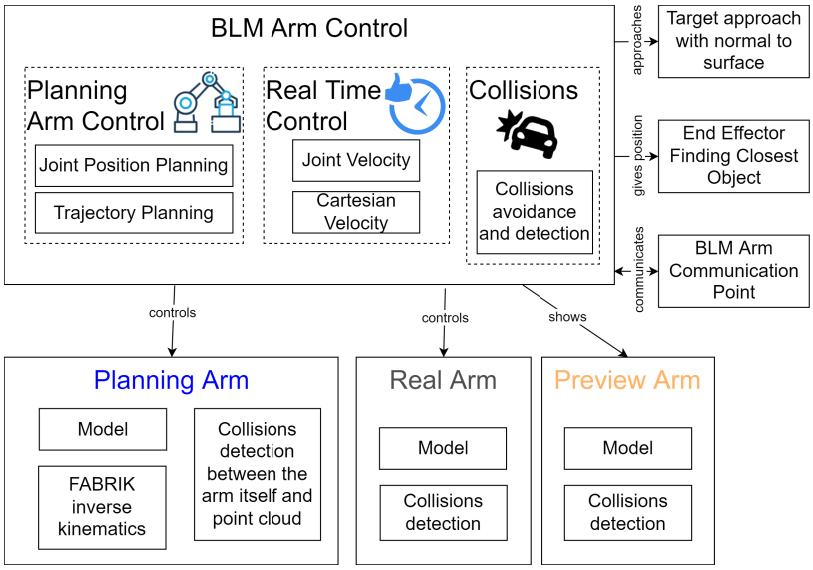

FIGURE 12. BLM arm control architecture. It shows how the main functionalities, responsible for manipulator movement planning, real-time control, preview, collision avoidance and detection, or target approach, are structured.

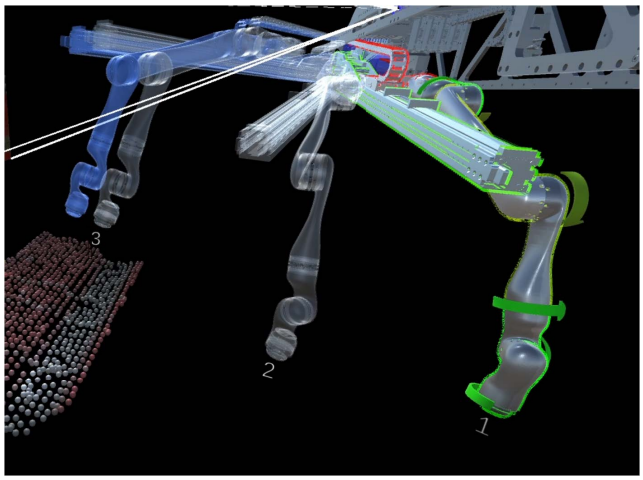

FIGURE 13. Trajectory planning. There are 3 types of arms visible: opaque real arm, transparent waypoints with numbers and blue planning arm.

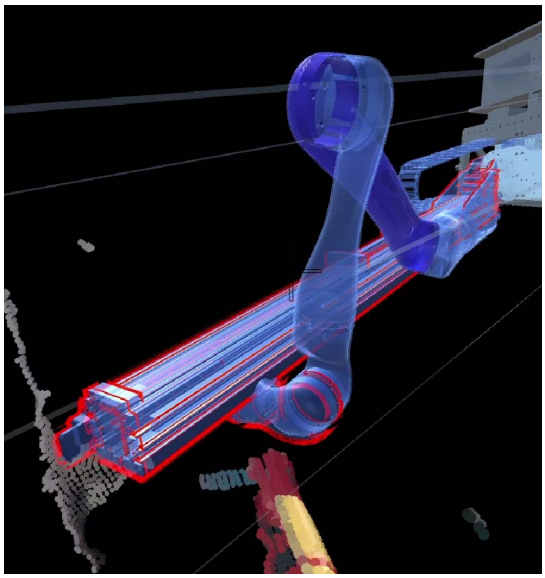

FIGURE 14. Planning arm self-collision detection.

1) COLLISION AVOIDANCE

If a collision is detected by the system during planning control mode, the operator should modify the waypoint or current arm position to avoid the collision. An example of the self-collision of the planning arm is shown in Figure 14, where the end-effector collided with another part of the arm. An example of the preview arm collision is shown in Figure 15, where the arm will self-collide if the movement is executed. The last example of a collision of the planning arm with point cloud is shown in Figure 16, where the end-effector collided with the Beam-Loss Monitor device's point cloud.

2) SENSORY COLLISION DETECTION

The sensory collision detection is represented in the interface by directional arrows and colors indicating the value of a torque or a force in each joint. The movement of the arm is stopped when the torque value exceeds a specified limit. An example of a situation when the arm collided and two joints started to have higher torques is shown in Figure 17.

3) VIRTUAL COLLISION DETECTION

Apart from the sensory collision detection using real joint torque feedback, a virtual real-time collision detection has been implemented. It detects when the real arm model collides with itself or with point cloud. It can happen during an execution of a movement to a waypoint or during real-time direct arm control. When a collision is detected, the

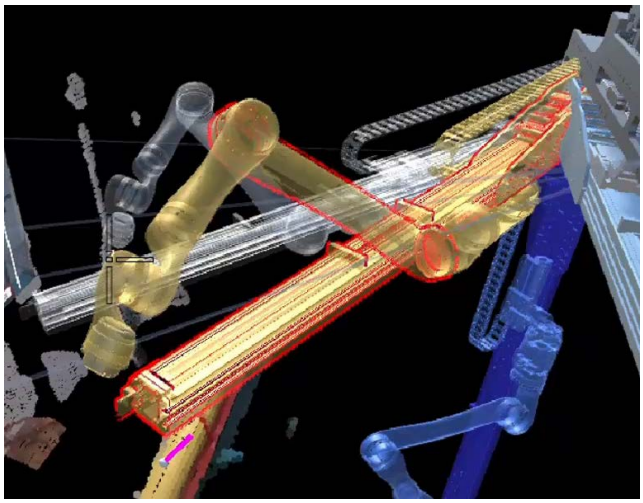

FIGURE 15. Preview arm self-collision detection.

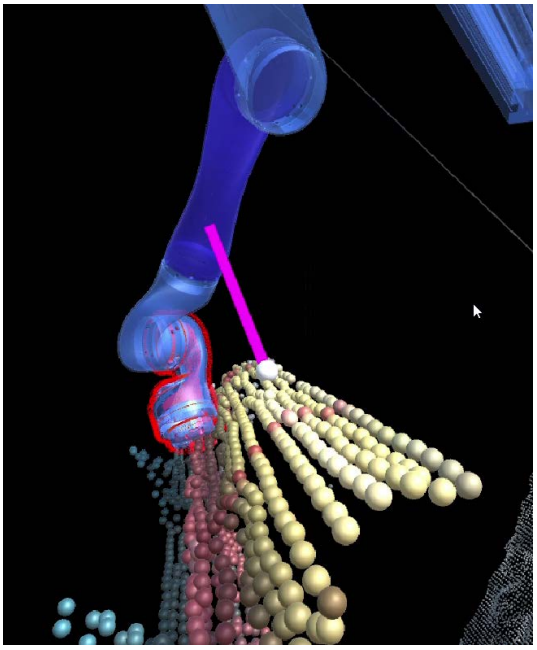

FIGURE 16. Planning arm point cloud collision detection.

movement of the arm is stopped. The situations when the real arm collided can be seen in Figures 17 and 18.

G. TARGET APPROACH WITH A POINT CLOUD NORMAL

The point cloud acquisition allows the interface to visualize the environment in 3D. Additionally, on most point cloud points, a surface normal can be calculated. Using this, a functionality of a simplified target approach has been implemented. The operator can select any point cloud point and a normal will be added in the environment. The direction of the surface normal is based on the surrounding points. However, the direction and distance can be adjusted by using a gizmo (Figure 19). Furthermore, in the planning mode with the inverse kinematics, the arm can automatically move to this point on demand of the operator. This makes any target approach much faster and it requires only 4 quick operations:

1) Select a point cloud point for the normal vector;
2) Adjust the direction/distance of a point on the normal;
3) Command the end-effector to move to this point;
4) Launch movement after the preview arm collision check.

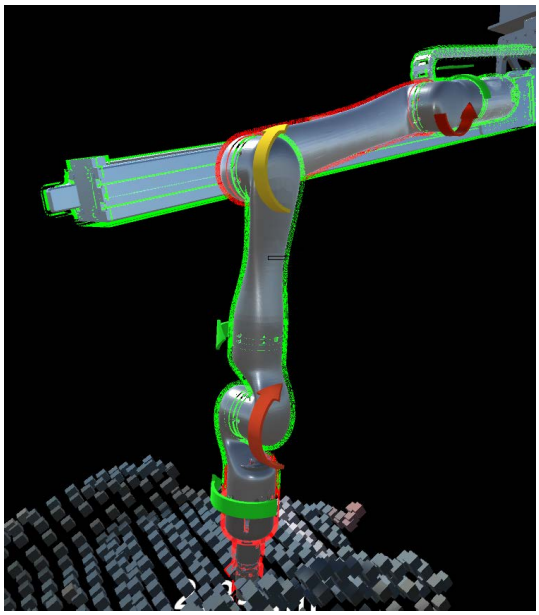

FIGURE 17. Real arm torques during a collision, with point cloud collision visualization.

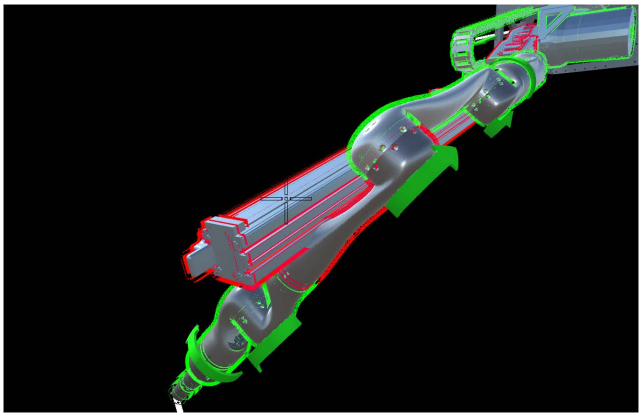

FIGURE 18. Real arm self-collision detection.

H. CLOSEST OBJECT DISTANCE INDICATOR

For further assistance with environment awareness and early prevention of collisions, or for precise end-effector manipulation, a functionality of showing the closest object distance to the end-effector has been implemented. In Figure 20, indicators for real and waypoint arms are presented. The distance is calculated between the closest point cloud point and the end-effector tip. Analogously, the closest object distance is indicated for the planning and preview arms.

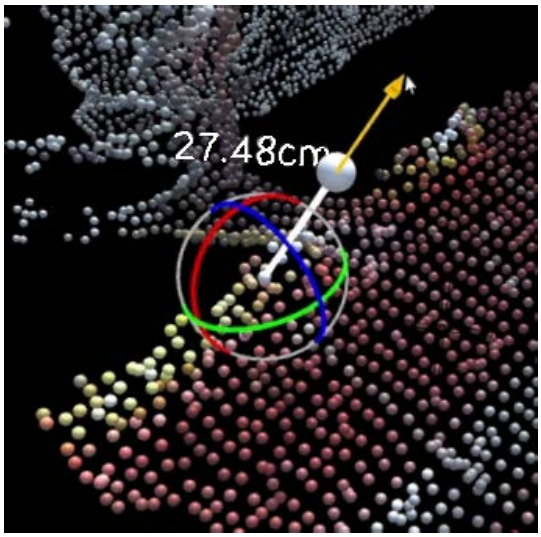

FIGURE 19. Target approach point normal to point cloud surface with gizmo interaction.

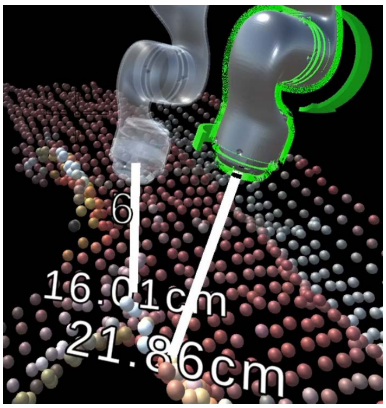

FIGURE 20. Closest object indicators for real and waypoint arms.

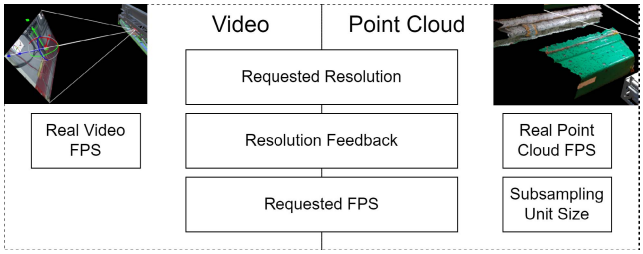

FIGURE 21. Point cloud and video settings. The settings in the middle are shared between video and point cloud acquisition.

I. VIDEO AND POINT CLOUD FEEDBACK

During teleoperations, operators require real-time feedback from the robot. The feedback signals can be a single piece of information such as speed, force or position, or it can have a complex structure, such as a camera feedback. In the presented Mixed Reality Interface, in addition to the 2D video stream, the 3D feedback has a form of point cloud, which resembles reality as much as it is technically possible. The video feedback provides a clearer feedback due to its 2D form, however it does not provide precise 3D spatial information. When approaching an object, only the point cloud can provide direct distance information. The video and point cloud are acquired in real-time according to the parameters specified by the teleoperator. In the presented application an RGB-D camera was used as the point cloud source, but it is universal and can also be acquired from a different sensor type, for example, a LiDAR. The initial processing of the point cloud is done on the onboard computer, including filtering, subsampling, compression and serialization as needed by the task. After processing, it is sent to the interface.

1) RGB-D CAMERA SETTINGS AND ACQUISITION STATUS

Basic camera parameters are shown in Figure 21. Three of them (requested resolution, resolution feedback and requested FPS) are shared between the point cloud and video streams, while real video FPS and video stream enable are specific to video, and real point cloud FPS, point cloud enable and subsampling unit size are specific for point cloud. The subsampling unit size is the distance between point clouds points. This parameter can decrease the size of the transmitted point cloud by decreasing the point cloud density. The real video and point cloud FPS are estimated as the difference between the current and the previous received video or point cloud frame counter divided by a constant interval. The Equation 1 shows this estimation, where C_{current} is the current video or point cloud frame counter, C_{previous} is the previous video or point cloud frame counter, FPS_{real} is the real video or point cloud FPS and Interval is the real FPS update interval that in this project it is set to 3 seconds but can be changed.

$$\text{FPS}_{\text{real}} = \frac{C_{\text{current}} - C_{\text{previous}}}{\text{Interval}} \tag{1}$$

Depending on the model of the RGB-D camera, the available resolutions and corresponding allowed FPS settings may vary. That is why the choice of resolution is simplified to 3 options: LOW, MEDIUM, HIGH which are automatically selected from the available resolutions. The requested FPS and resolution are parameters which can be changed as inputs.

2) UNITY RGB-D CAMERA ARCHITECTURE

The Unity RGB-D camera structure is shown in Figure 22. The component controls the following functionalities:

- Camera control, communication with the camera process in the robot, setting the requested parameters, enabling video or point cloud streams;
- Managing the operator's interaction with the camera hologram;
- Display of a video canvas with projection lines and a point cloud area in the scene;
- Automatic camera settings controlled by camera internal state or network measurements (i.e. the focus of the paper).

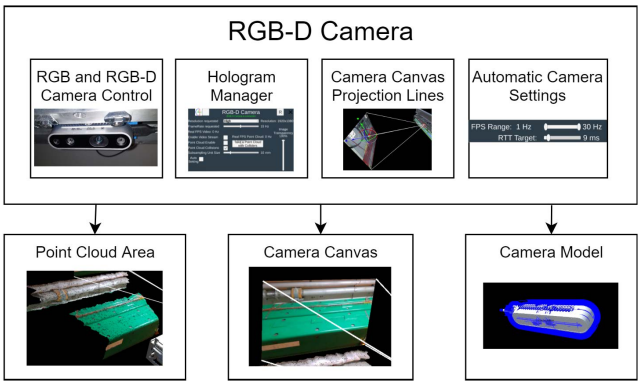

FIGURE 22. RGB-D camera architecture in unity.

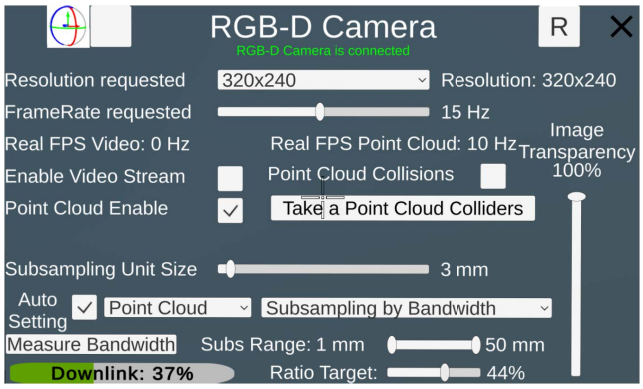

FIGURE 23. Camera hologram with automatic settings enabled. All the parameters of camera acquisition and automatic settings can be controlled here.

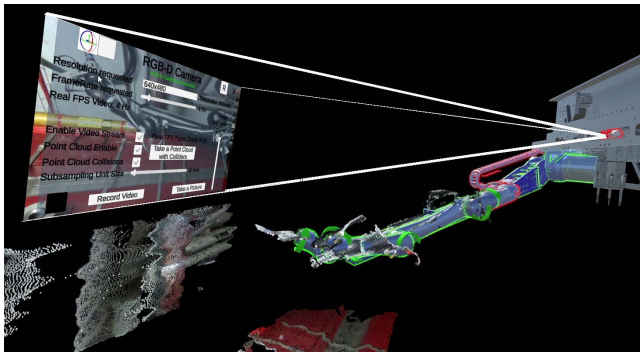

FIGURE 24. Transparent camera hologram view overlaid on the video stream in the 3D environment.

3) CAMERA HOLOGRAM

The teleoperator can modify the parameters of the camera acquisition (Section II-I1), video and point cloud streams, or automatic settings via a hologram. The hologram is a transparent 3D object displayed in the interface (Figures 23 and 24).

4) AUTOMATIC SETTINGS CONTROL

Automatic settings control (accessible to the teleoperator in the hologram shown in Figure 23) adjusts the camera settings depending on a chosen parameter. The operator can decide to use only 2D feedback, or both video and point cloud, or only point cloud. The automatically controlled settings and monitored parameters are camera-related (FPS, point cloud subsampling or resolution) but the monitored parameters can also be network-related (throughput, bandwidth, round-trip time). The synthesis of the automatic setting modes is in Table 1. The algorithms and hologram visual controls are grouped in Appendix C. A selected automatic setting algorithm is executed once per specified amount of time (e.g. 5 seconds). There are 4 distinct types of 12 automatic setting modes:

- **Settings controlled by throughput to bandwidth ratio:** During an intervention a robot can be located in areas where the network coverage is poor with highly limited bandwidth. Furthermore, if it happens that the available bandwidth is used in full capacity, it causes problems of even temporarily lost connection or stopping any video or point cloud stream. Moreover, the available bandwidth can vary over time and even nearby locations. This is why the automating setting of FPS, resolution or subsampling were introduced to allow the system to adapt to current conditions and facilitate the teleoperation. The setting adapts to the current throughput to bandwidth ratio, and the operator has the possibility to launch the available bandwidth measurement at any time. When the ratio is above the setpoint set by the operator, the FPS or resolution decreases or the subsampling increases. Accordingly, they change if the ratio is below the setpoint to give better quality of control. The operator can set the limits of FPS or subsampling within the range that the automatic control algorithm can adjust them. The algorithms using throughput to bandwidth ratio for point cloud FPS and subsampling control are described in Algorithms 1, 2, and for video FPS and resolution control in Algorithms 1, 3 in Appendix C. The holographic control is shown in Figures C.1, C.2, C.3 and C.4.
- **Settings controlled by round-trip time:** During a precise teleoperation it is important to have fast-enough and stable feedback from the robot to avoid collisions and to allow delicate manipulation, particularly if haptic feedback should be implemented. The frequency of camera feedback can be controlled with a setting, however the round-trip time depends on the used network and load. This is why the automatic setting mode monitoring round-trip time allows automatic control of the FPS, resolution and point cloud subsampling. The algorithms using round-trip time for point cloud FPS and subsampling control are described in Algorithms 4, 5, and for video FPS and resolution control in Algorithms 4, 6 in Appendix C. The holographic control is shown in Figures C.5, C.6, C.7 and C.8.
- **FPS controlled by point cloud subsampling or resolution:** Subsampling and resolution settings impact

the size of a single frame and allow the control of feedback precision required by a teleoperator. However, network bandwidth constraints might not allow high enough frame rates. That is why the requested FPS setting is automatically lowered in order to avoid buffering frames and a collapse of the stream. On the other hand, if the requested frame rate is achieved, the automatic setting algorithm tries to slowly increase the setting to increase the frame rate, which improves control quality. The automatic control is described in Algorithm 7 in Appendix C. The holographic control is shown in Figures C.9 and C.10.

- **Resolution or point cloud subsampling controlled by FPS:** In certain operation scenarios the teleoperator needs to have high-enough feedback frequency which is task and environment specific. The frequency defines the delay added to network delay that the teleoperator feels while controlling a robot. However, in this case the maximum size of a single feedback frame (point cloud or video) can be limited by network bandwidth. This is why in this automatic setting mode the resolution or point cloud subsampling are automatically adjusted to achieve the requested FPS. Similarly to the previous point, overloading the network can lead to a total collapse of stream due to buffering. On the other hand, if the network allows, the algorithm tries slowly to increase the precision (by increasing resolution or lowering subsampling). It is important to note that this mode does not use any direct network monitoring, it is based only on the camera behaviour. The automatic control is described in Algorithms 8 and 9 in Appendix C. The holographic control is shown in Figures C.11 and C.12.

J. NETWORK MEASUREMENTS

During an intervention, the operator can check the current network parameters and measured bandwidth, as shown in the menu screen in Figure 11 in Section II-C4 which provides a full overview of the network performance situation.

The camera acquisition throughput is proportional to the FPS and resolution, and inversely proportional to the subsampling (applicable for point cloud only), as presented in the Equation 2. The formulas calculating the round-trip time and deviation were inspired by the TCP procedures [39].

$$\text{Throughput} \sim \text{FPS} \cdot \frac{1}{\text{subsampling}} \cdot \text{resolution} \tag{2}$$

Another parameter is the round-trip time, which is calculated according to the Equation 3, and standard values are shown in the Equation 4.

$$\text{RTT} = (1-\alpha) \cdot \text{PreviousRTT} + \alpha \cdot \text{CurrentRTT} \tag{3}$$

where $\alpha = 0.125$, then

$$\text{RTT} = 0.875 \cdot \text{PrevRTT} + 0.125 \cdot \text{CurrRTT} \tag{4}$$

The round-trip time oscillations are also important to assess the network behaviour; the Equations 5 and 6 present how it is calculated.

$$\text{DevRTT} = (1-\beta) \cdot \text{PrevDevRTT} + \beta \cdot |\text{CurrRTT} - \text{PrevRTT}| \tag{5}$$

where $\beta = 0.25$, then

$$\text{DevRTT} = 0.75 \cdot \text{PrevDevRTT} + 0.25 \cdot |\text{CurrRTT} - \text{PrevRTT}| \tag{6}$$

The timeout interval is an indicative parameter in the communication performance, it is presented and calculated according to the Equation 7.

$$\text{TimeoutInterval} = \text{PrevRTT} + 4 \cdot \text{PrevDevRTT} \tag{7}$$

III. EXPERIMENTS

A. EXPERIMENT SET-UP

1) NETWORK CHARACTERIZATION

During the CERN robotic BLM validation measurements the robot is connected to the operator via the 4G/LTE network present in the LHC tunnel - currently the only suitable communications link with large coverage of the LHC. A large number of equipment would be necessary to provide full Wi-Fi coverage and it is not feasible to adequately protect standard Wi-Fi active electronic devices from ionising radiation, as explained in [40]. The 4G network uses a coaxial antenna cable, which since it is a passive device, can be exposed to radiation. The authors registered download maximum speed of 70 Mbps and upload maximum speed 14 or 30 Mbps depending on location. The antenna is shared between other networks, such as the LoRA network used for other purposes. Therefore, the CERN network in the LHC is characterized by volatility, variable bandwidth and latency and lack of information of the current use of the network which can all have an impact on robotic operations [32], [33].

During the experiments, 4 connection types were tested (Table 3) to provide a wider set of behaviours. Network parameters measured for all connection types are shown in Table 4. The Ethernet cable connection was expected to be a reference value as a maximum bandwidth and minimum round-trip-time that could be achieved. The connection of Ethernet over CERN General Purpose Network (GPN) is characterized by relatively very good bandwidth

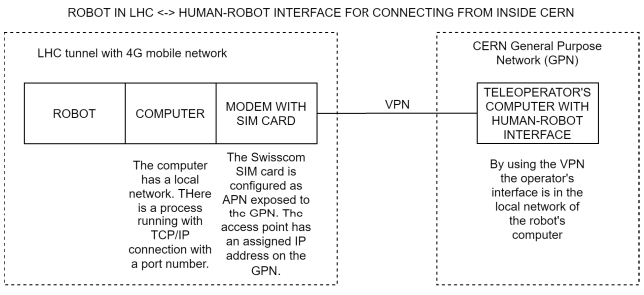

FIGURE 25. Connection between the camera acquisition system installed in a robot and the interface using automatic setting. A connection type using VPN 4G modem in the LHC tunnel is shown here.

TABLE 1. Point cloud and video automatic setting modes. All automatic modes are listed, the automatically controlled setting and settings controlled by operator are specified for each algorithm.

	Automatic setting mode	Automatically controlled settings	Settings controlled by operator
Point cloud	FPS by throughput to bandwidth ratio	FPS	Throughput to bandwidth ratio target, subsampling, FPS limits
	FPS by round-trip time		Round-trip time limit, subsampling, FPS limits
	FPS by subsampling		Subsampling, FPS limits
	Subsampling by throughput to bandwidth ratio	Subsampling	Throughput to bandwidth ratio target, FPS, subsampling limits
	Subsampling by round-trip time		Round-trip time limit, FPS, subsampling limits
	Subsampling by FPS		FPS, subsampling limits
Video	FPS by throughput to bandwidth ratio	FPS	Throughput to bandwidth ratio target, resolution, FPS limits
	FPS by Round-trip time		Round-trip time limit, resolution, FPS limits
	FPS by resolution		Resolution, FPS limits
	Resolution by throughput to bandwidth ratio	Resolution	Throughput to bandwidth ratio target, FPS
	Resolution by round-trip time		Round-trip time limit, FPS
	Resolution by FPS		FPS

TABLE 2. Automatic settings behaviours and dependencies of parameters. The arrows represent changes, i.e.: in the round-trip time dependent algorithms, when the round-trip time increases, an automatic setting algorithm will decrease the requested FPS or resolution, or increase subsampling to maintain the requested round-trip time target.

Parameter	Parameter change	Automatic settings controlled		
		Requested FPS	Subsampling	Resolution
$\frac{\text{Throughput}}{\text{Bandwidth}}$	↗	↘	↗	↘
	↘	↗	↘	↗
Round-trip time	↗	↘	↗	↘
	↘	↗	↘	↗
Real FPS	↗	↗	↘	↗
	↘	↘	↗	↘

and round-trip time values, however the bandwidth starts to vary over time-dependent infrastructure load. The Wi-Fi over CERN GPN significantly lowers the available bandwidth (74 Mbps) and the round-trip time is much longer (20-30 ms) and has a jitter of 5 ms. The VPN with 4G modem connection type had the worst results, with significant round-trip time (43-132 ms), depending highly on the bandwidth usage and had a jitter of 26 ms), and a low bandwidth value (12 Mbps) although the deviation of the bandwidth was low (< 1 Mbps). The VPN with 4G modem connection type is largely used for real intervention scenarios at CERN.

2) PARAMETERS OF VIDEO AND POINT CLOUD

During the experiments, all relevant settings and measurements were logged at the moment of acquisition of the point cloud or video frame to understand and recreate the state of the user interface and the network. The parameters presented in the graphs in Section III-B are:

- requested resolution, FPS and subsampling unit size of point cloud;
- real FPS for video or point cloud - calculated as number of received frames per last second;
- downlink throughput / bandwidth ratio;
- measured round-trip time.

3) EXPERIMENTS OVERVIEW

The network and camera video or point cloud streaming parameters and behaviour were tested. The experiments were undertaken separately for both the point cloud and video streams. In each experiment, there was one setting (i.e. FPS, resolution, subsampling, throughput to bandwidth ratio or round-trip time limit) that was the experiment setpoint, changing in time according to the experiment sequence. Other settings were either constant, not used or automatically controlled by the algorithms. In total, there were 16 types of experiments. During the first half of the experiment the demand for throughput was increasing, and in the second part decreasing. There were 4 experiments in which the automatic setting algorithms were not enabled to test the behaviour and encountered problems, and 12 experiments where there was one automatic setting algorithm enabled to test if and how the problems were solved or what was the improvement in the behaviour. The overview of experiments, showing which parameter was the setpoint and which were automatically controlled or constant, is shown in Table 5.

Additionally, the times of point cloud processing on the robot/server and the operator/client sides were measured to provide the range of the processing delay in addition to the network delay.

4) HARDWARE

As shown in Figure 25, the connections were established between 2 machines: the operator interface and the robot computer. All the variations of interfaces were tested. The

TABLE 3. Connection types between a robot and an operator's computer.

Network connection type	Robot side's connection	Interconnection	Operator's computer
Ethernet cable directly	Ethernet cable connector	Sirgle cable	Ethernet cable connector
Ethernet over CERN GPN	Ethernet cable connection to CERN General Purpose Network	CERN General Purpose Network cabled indrastructure	Ethernet cable connection to CERN General Purpose Network
VPN 4G modem in the LHC tunnel	4G modem	Network operator 4G connection in the LHC tunnel, VPN connection to CERN General Purpose Network cabled infrastructure	Ethernet cable connection to CERN General Purpose Network
Wi-Fi over CERN GPN	Wi-Fi connection to CERN General Purpose Network Wi-Fi infrastruture	CERN General Purpose Network cabled infrastructure	Wi-Fi connection to CERN General Purpose Network Wi-Fi infrastruture

TABLE 4. Bandwidth and its deviation, round-trip time and its jitter measurements for all connection types.

Network connection type	Downlink bandwidth [Mbps]	Round-trip time [ms]		Jitter [ms]	Bandwidth standard deviation [Mbps]
		Bandwidth usage = 0%	Bandwidth usage = 100%		
VPN with 4G modem	11.99	43.4	131.5	26.00	0.61
Wi-Fi over CERN GPN	73.95	20.33	30.3	5.64	17.88
Ethernet over CERN GPN	885.8	0.04	1.61	0.19	38.1
Ethernet cable directly	941.8	0.24	1.47	0.28	0.42

FIGURE 26. Teltonika RUTX12 used for experiment with the VPN 4G modem over CERN GPN connection type experiments.

experiments were done with a hardware configuration shown in Table 6. The client was a Windows machine running the Unity project, while the robot onboard computer (Intel NUC) ran Linux Ubuntu. In the experiments with the VPN connection over 4G network, a mobile modem was used (Figures 26 and 27) and the onboard computer was the server. In the experiments over Wi-Fi or cabled connection over CERN GPN, a DELL computer was the server. The Intel RealSense D415 and D435 camera were used for the experiments (Figures 28, 29 and 30).

B. RESULTS STRUCTURE

The results are organized in the order presented in Table 5 and divided in two groups; the first where the automatic control of settings was not used and the second where the automatic control was used. Although the experiments were done with all the network connection types (Table 3), in each experiment section there are results and graphs with chosen connection type(s) that best present the situation, problem or solution.

C. EXPERIMENTS WITH NO AUTOMATIC SETTINGS

The first group of experiments with no automatic control of settings is presented in this section. The behaviours and problems shown here are often encountered during real robotic interventions. The consequences for teleoperation caused by these behaviours are synthesized in Table 8 in Section III-G.

1) POINT CLOUD EXPERIMENT: CHANGING FPS

The FPS setpoint was changing over time, while the point cloud subsampling and resolution settings were constant. In Figures 31 and 32 the requested FPS could not be achieved due to the available bandwidth limit. After reaching the limit, the real FPS started to oscillate around its maximum, and the round-trip time increased 2 times instantaneously.

2) POINT CLOUD EXPERIMENT: CHANGING SUBSAMPLING

The subsampling setpoint was changing over time and the point cloud requested FPS and resolution settings were constant. In Figures 33 and 34 the acquisition FPS decreased significantly after reaching the bandwidth limit. The used connection type was Ethernet over CERN GPN. An analogous situation is shown for the Wi-Fi connection type (Figures 35 and 36), which has a lower bandwidth, and which

TABLE 5. Overview of experiments.

Setting control mode	Point cloud or video	N°	Experiment name	Settings			Network settings	
				FPS	Resolution	Subsampling	Throughput to bandwidth ratio	Round-trip time
Manual control of settings	Point cloud	1	Changing FPS	**Experiment setpoint**	Const	Const	Not used	
		2	Changing subsampling	Const		**Experiment setpoint**		
	Video	3	Changing FPS	**Experiment setpoint**		Not used		
		4	Changing resolution	Const	**Experiment setpoint**			
Automatic settings	Point cloud	5	FPS by throughput to bandwidth ratio	Controlled by automatic setting algorithm	Const	Const	**Experiment setpoint**	Not used
		6	FPS by round-trip time				Not used	**Experiment setpoint**
		7	FPS by subsampling			**Experiment setpoint**		Not used
		8	Subsampling by throughput to bandwidth ratio	Const		Controlled by automatic setting algorithm	**Experiment setpoint**	
		9	Subsampling by round-trip time				Not used	**Experiment setpoint**
		10	Subsampling by FPS					Not used
	Video	11	FPS by throughput to bandwidth ratio	Controlled by automatic setting algorithm	Const	Not used	**Experiment setpoint**	
		12	FPS by round-trip time				Not used	**Experiment setpoint**
		13	FPS by resolution		**Experiment setpoint**			Not used
		14	Resolution by throughput to bandwidth ratio	Const	Controlled by automatic setting algorithm		**Experiment setpoint**	
		15	Resolution by round-trip time				Not used	**Experiment setpoint**
		16	Resolution by FPS	**Experiment setpoint**				Not used

caused the problem to start earlier and for a lower requested FPS setting.

3) VIDEO EXPERIMENT: CHANGING FPS

When using VPN 4G modem connection in the LHC tunnel, the available bandwidth was limited and affected the video acquisition. After consuming all the available bandwidth, the requested FPS could not be achieved. The increase of the round-trip time, which doubled instantly by exceeding 200 ms, and its oscillation was very significant (100%), when the throughput reached the bandwidth. The observed behaviour is shown in Figures 37 and 38.

4) VIDEO EXPERIMENT: CHANGING RESOLUTION

The video resolution setpoint was changing over time, while the requested FPS was constant. The VPN 4G modem in the LHC tunnel connection type was used. As presented in Figures 39 and 40, for HIGH resolution, the requested FPS was not achieved due to bandwidth limit. The round-trip time increased by 4 times when the resolution was switched from MED to HIGH, and the throughput and round-trip time became unstable and started significant oscillations. The temporary real FPS falls were caused by switching resolution.

D. EXPERIMENTS WITH AUTOMATIC MODES

In this section, the experiments using automatic setting modes are presented. A synthesis of scenarios and solved problems or improvements due to the automatic settings is presented in Table 8 in III-G Section.

1) POINT CLOUD FPS CONTROLLED BY THROUGHPUT TO BANDWIDTH RATIO

The point cloud throughput to bandwidth ratio setpoint was changing over time, while the requested FPS was controlled

TABLE 6. The Human-Robot Interface's and robot's machines specification.

	Human-Robot Interface machine (client)	Robot's machine (server)	
Device model	MSI GL75 Leopard 10SFR, rev 1.0	DELL Latitude E6230, 089JM2, rev. A00	Intel NUC11TNKv70QC
CPU	Intel(R) Core(TM) i7-10750H	Intel(R) Core(TM) i7-3520M	Intel(R) Core™ i7-1185G7
GPU	NVIDIA GeForce RTX 2070	Integrated, Intel	Integrated, Intel
Memory	16 GB, DDR4, 1333 MHz	4 GB, DDR3, 1600 MHz	64GB, DDR4, 3200 MHz
System	Windows 10 Pro, version 20H2	Linux Ubuntu 20.04.2 LTS	Linux Ubuntu 20.04.2 LTS
Benchmark	Graphics 7435, CPU 6622 (3DMark TimeSpy v1.2), 523 with 1 thread, 3747 with 12 threads (CPU-Z v.17.01.64)	35855 with 1 thread, 64978 with 4 threads (CPU-X v.3.2.4, slow prime numbers)	68000 with 1 thread, 127358 with 8 threads (CPU-X v.3.2.4, slow prime numbers)

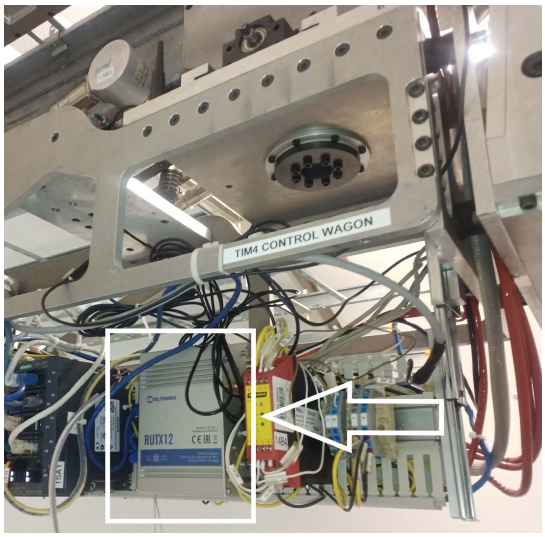

FIGURE 27. Teltonika RUTX12 installed in the Train Inspection Monorail wagon. The position of the router device in the robot is highlighted.

FIGURE 28. Intel RealSense D415 used for experiments, mounted on a tripod. The camera consists of a pair of depth sensors, an infrared projector, and an RGB sensor. The D415 model has a rolling shutter.

by the automatic setting algorithm to follow the setpoint, and the resolution and subsampling were constant. The Wi-Fi over CERN GPN connection type is presented. In Figure 41, the requested FPS adjustments are changing the throughput correctly, and the real FPS value is following it precisely. The requested FPS adapts automatically to the size of the point cloud and the specified ratio and subsampling, while maintaining a stable acquisition stream.

FIGURE 29. Intel RealSense D435, also used in the experiments, installed in the train inspection monorail wagon. The D435 model, compared to D415, has a wider field of view (FOV) and a global shutter.

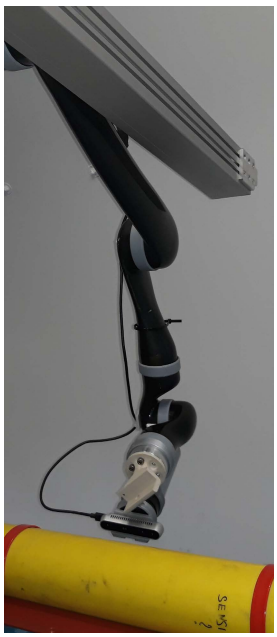

FIGURE 30. The D415 camera mounted on the effector of the BLM manipulator that approached a beam loss monitor.

2) POINT CLOUD FPS CONTROLLED BY ROUND-TRIP TIME

The point cloud round-trip time setpoint was changing over time, the requested FPS was controlled by the automatic setting algorithm to follow the setpoint, and the resolution and subsampling were constant. The Wi-Fi over CERN GPN connection type is presented. In Figures 42 and 43, the requested round-trip time was automatically followed by adjusting the requested FPS setting. Despite major oscillations of the round-trip time characterizing this connection type, the setpoint was followed correctly.

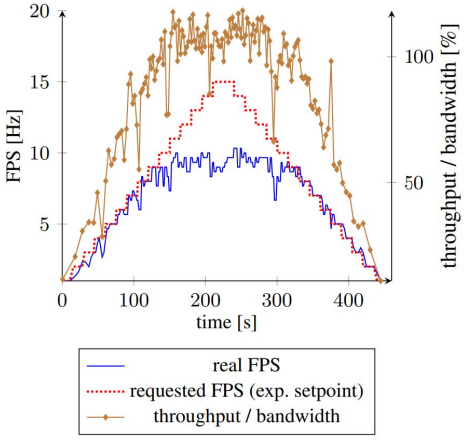

FIGURE 31. Point cloud experiment without automatic settings (exp. n° 1 in Table 5). Requested FPS as setpoint. Throughput to bandwidth ratio, real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 10 mm.

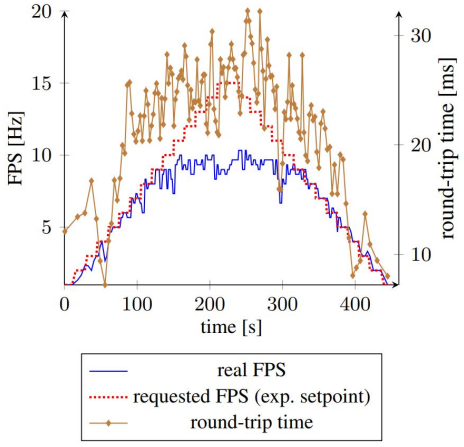

FIGURE 32. Point cloud experiment without automatic settings (exp. n° 1 in Table 5). Requested FPS as setpoint. Round-trip time, real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 10 mm.

3) POINT CLOUD FPS CONTROLLED BY SUBSAMPLING

The point cloud subsampling setpoint was changing over time, the requested FPS was controlled by the automatic setting algorithm to follow the setpoint, and the resolution was constant. The Wi-Fi over CERN GPN connection type is presented. In Figures 44 and 45, by changing the subsampling, the requested FPS automatically adjusted to maintain the used bandwidth at high level. This mode does not need any network measurement and is based only on the internal acquisition parameters. The bandwidth limit was probed by increasing the requested FPS until the acquisition failed to achieve the setting, then decreasing it accordingly. The round-trip time was maintained at a similar level with natural oscillations for this connection type.

4) POINT CLOUD SUBSAMPLING CONTROLLED BY THROUGHPUT TO BANDWIDTH RATIO

In this experiment, the point cloud throughput to bandwidth ratio setpoint was changing over time, the subsampling was

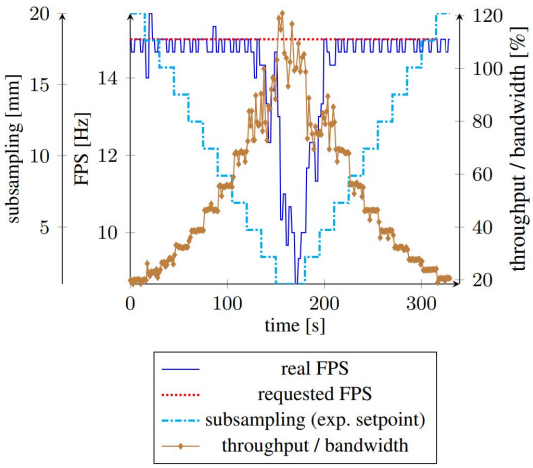

FIGURE 33. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: Ethernet over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

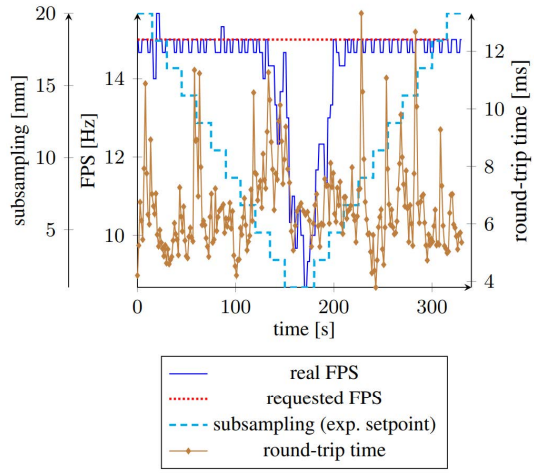

FIGURE 34. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: Ethernet over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

controlled by the automatic setting algorithm to follow the setpoint, and the resolution and requested FPS were constant. The Wi-Fi over CERN GPN connection type is presented. In Figures 46 and 47, the subsampling setting was adapting to the throughput to bandwidth ratio target, while maintaining the real FPS at the value set by the requested FPS setting, even for the highest setpoint values. The round-trip time approximately doubled when the throughput to bandwidth ratios had minimum and maximum values. There was an occasional spike, which was observed for this connection type.

5) POINT CLOUD SUBSAMPLING CONTROLLED BY ROUND-TRIP TIME

The point cloud round-trip time setpoint was changing over time, the subsampling was controlled by the automatic setting algorithm to follow the setpoint, and the resolution and requested FPS were constant. The Wi-Fi over CERN GPN

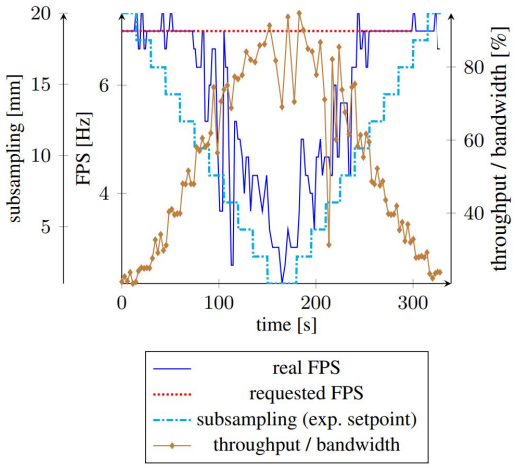

FIGURE 35. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 7 Hz.

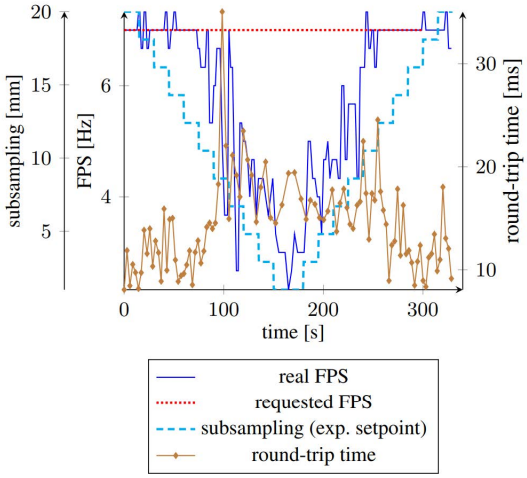

FIGURE 36. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 7 Hz.

connection type is presented. In Figures 48 and 49, the round-trip time target was followed, although the achieved round-trip time had significant oscillations, which characterize the used network. Because of these oscillations, the subsampling had to change often by up to 30%. Occasionally, when the throughput to bandwidth ratio was close to 100%, the real FPS value decreased by a few frames per second below the requested FPS.

6) POINT CLOUD SUBSAMPLING CONTROLLED BY FPS

The point cloud requested FPS setpoint was changing over time, the subsampling was controlled by the automatic setting algorithm to follow the setpoint, and the resolution was constant. The Wi-Fi over CERN GPN connection type is presented. In the Figures 50 and 51, the real FPS followed the requested FPS target by automatically adjusting the subsampling setting. The bandwidth use was indirectly maximized whenever possible to have a minimum subsampling

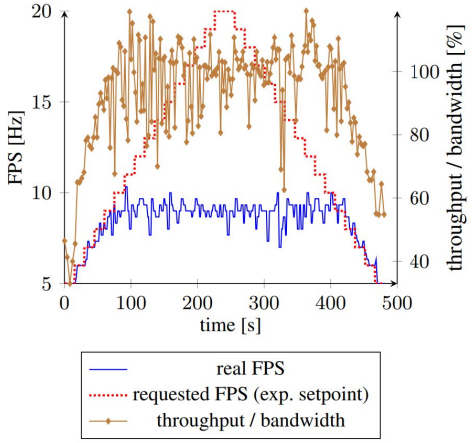

FIGURE 37. Video experiment without automatic settings (exp. n° 3 in Table 5). Requested FPS as setpoint. Throughput to bandwidth ratio, real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

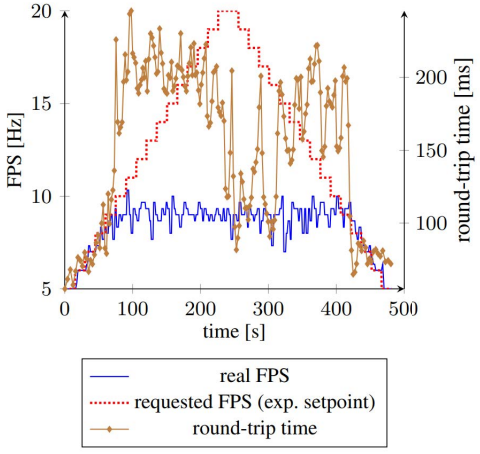

FIGURE 38. Video experiment without automatic settings (exp. n° 3 in Table 5). Requested FPS as setpoint. Round-trip time, real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

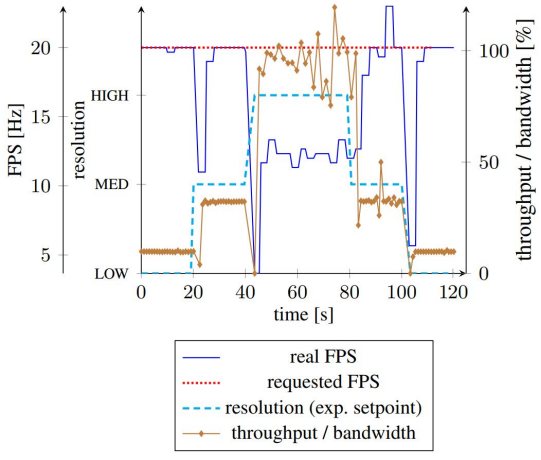

FIGURE 39. Video experiment without automatic settings (exp. n° 4 in Table 5). Resolution as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 20 Hz.

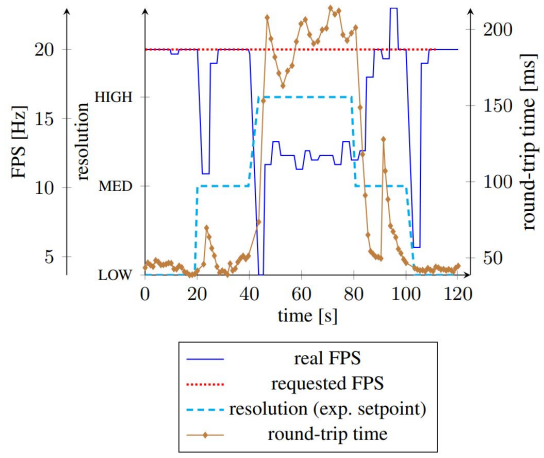

FIGURE 40. Video experiment without automatic settings (exp. n° 4 in Table 5). Resolution as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 20 Hz.

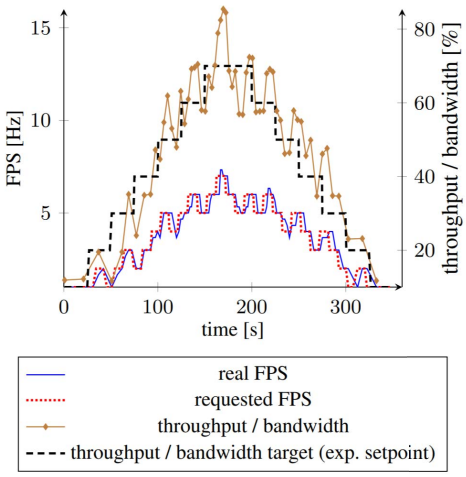

FIGURE 41. Point cloud experiment with automatic settings (exp. n° 5 in Table 5). FPS was controlled by throughput to bandwidth ratio target as the setpoint. The throughput to bandwidth ratio, requested and real FPS behaviour are presented with the onnection type of Wi-Fi over CERN GPN and with constant parameters: resolution = MEDIUM, subsampling = 10 mm.

for a given requested FPS setting. The round-trip time was proportional to the throughput to bandwidth ratio with an occasional spike.

7) VIDEO FPS CONTROLLED BY THROUGHPUT TO BANDWIDTH RATIO

The video throughput to bandwidth ratio setpoint was changing over time, the requested FPS was controlled by the automatic setting algorithm to follow the setpoint, and the resolution was constant. The VPN over CERN GPN connection type is presented. In Figures 52 and 53, the requested FPS setting automatically adapted to the throughput to bandwidth ratio target changes. The round-trip time increased when more throughput was used, and had significant oscillations characteristic to this connection type.

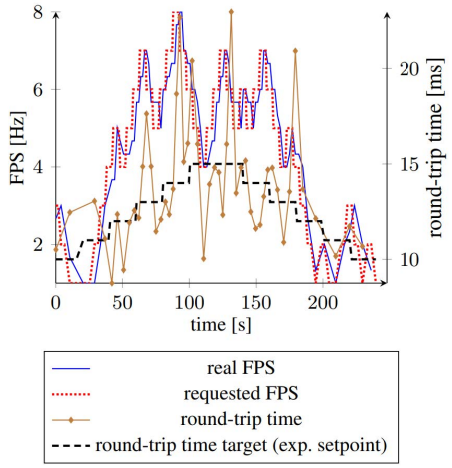

FIGURE 42. Point cloud experiment with automatic settings (exp. n° 6 in Table 5). FPS controlled by round-trip time target as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 15 mm.

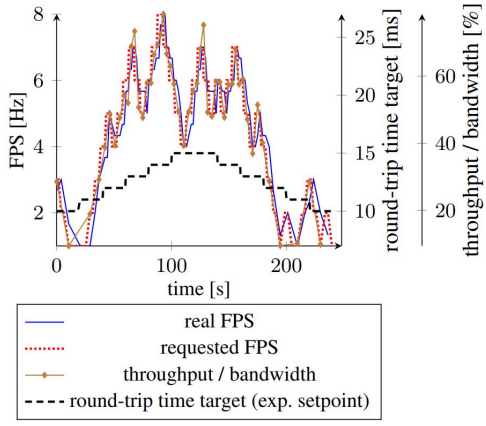

FIGURE 43. Point cloud experiment with automatic settings (exp. n° 6 in Table 5). FPS controlled by round-trip time target as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 15 mm.

8) VIDEO FPS CONTROLLED BY ROUND-TRIP TIME

The video round-trip time setpoint was changing over time, the requested FPS was controlled by the automatic setting algorithm to follow the setpoint, and the resolution was constant. The VPN 4G modem in the LHC tunnel connection type is presented. In Figures 54 and 55, the round-trip time correctly followed the requested target, though it was characterized with significant oscillations (up to 80% temporary increase), which are characteristic to this connection type. The throughput to bandwidth ratio was proportional to the real FPS setting that was automatically controlled.

9) VIDEO FPS CONTROLLED BY RESOLUTION

The video resolution was changing over time, and the FPS was controlled by the automatic setting algorithm to follow the setpoint. The VPN 4G modem in the LHC tunnel connection type is presented. In Figures 56 and 57, the requested FPS adapted to the active resolution by probing the highest

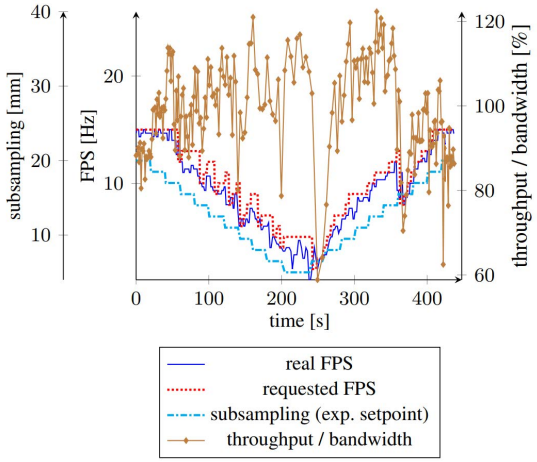

FIGURE 44. Point cloud experiment with automatic settings (exp. n° 7 in Table 5). FPS controlled by subsampling as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM.

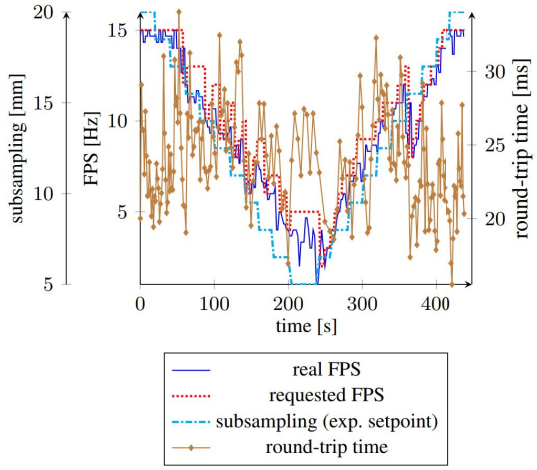

FIGURE 45. Point cloud experiment with automatic settings (exp. n° 7 in Table 5). FPS controlled by subsampling as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM.

possible value allowed by the bandwidth. With the HIGH resolution, the bandwidth limit was reached and the FPS had to be limited to around 3 times from the value that was used for the MED resolution. Temporary drops of the real FPS value were caused by resolution changes in the camera acquisition process.

10) VIDEO RESOLUTION CONTROLLED BY THROUGHPUT TO BANDWIDTH RATIO

The video throughput to bandwidth ratio was changing over time, and the resolution was controlled by the automatic setting algorithm to follow the setpoint. The VPN 4G modem in the LHC tunnel connection type is presented. In Figures 58 and 59, the resolution setting correctly followed the throughput to bandwidth ratio setpoint. Temporary drops of the real FPS value come from the resolution change in the camera acquisition process.

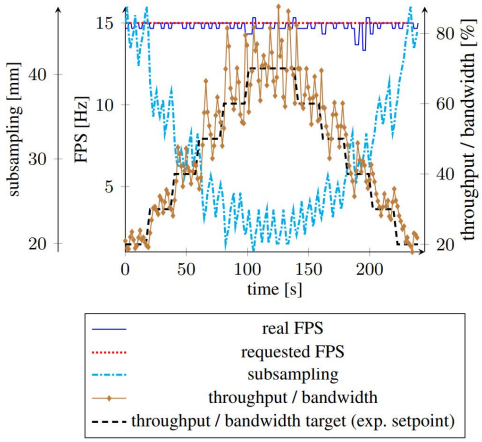

FIGURE 46. Point cloud experiment with automatic settings (exp. n° 8 in Table 5). Throughput to bandwidth ratio target as setpoint. Throughput to bandwidth ratio, subsampling, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

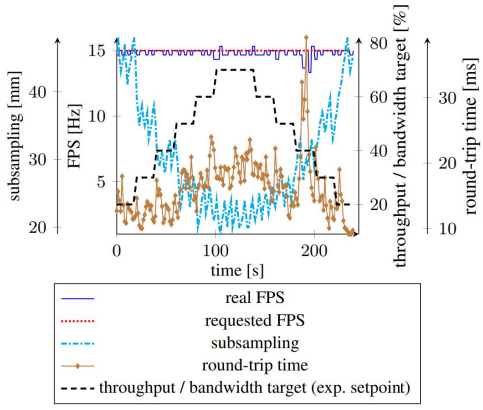

FIGURE 47. Point cloud experiment with automatic settings (exp. n° 8 in Table 5). Throughput to bandwidth ratio target as setpoint. Round-trip time, subsampling, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

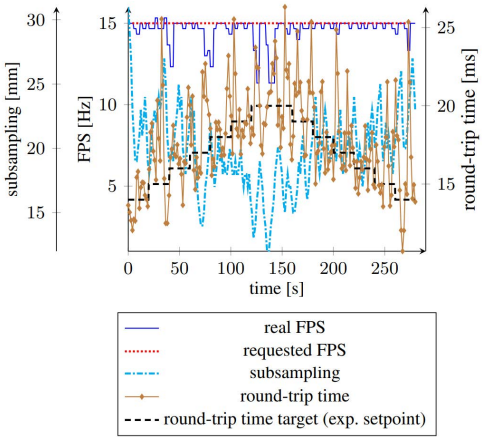

FIGURE 48. Point cloud experiment with automatic settings (exp. n° 9 in Table 5). Subsampling unit size controlled by round-trip time target as setpoint. Round-trip time, subsampling, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

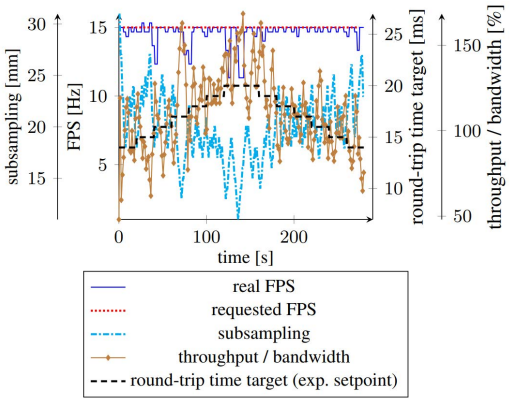

FIGURE 49. Point cloud experiment with automatic settings (exp. n° 9 in Table 5). Subsampling unit size controlled by round-trip time target as setpoint. Throughput to bandwidth ratio, subsampling, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

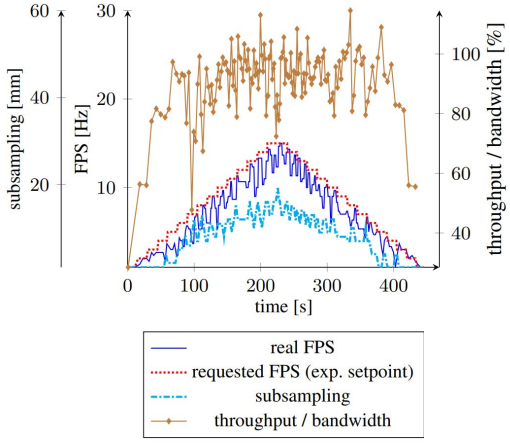

FIGURE 50. Point cloud experiment with automatic settings (exp. n° 10 in Table 5). Subsampling unit size controlled by FPS as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM.

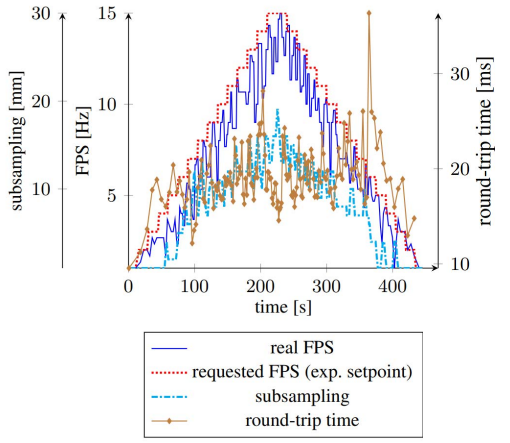

FIGURE 51. Point cloud experiment with automatic settings (exp. n° 10 in Table 5). Subsampling unit size controlled by FPS as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM.

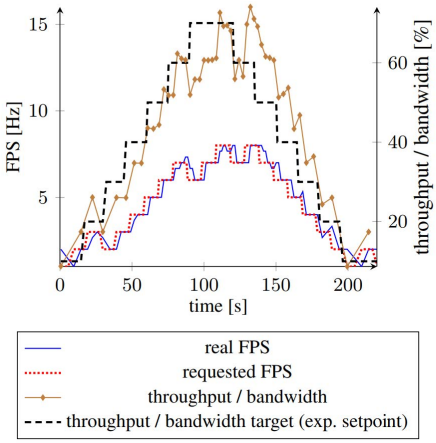

FIGURE 52. Video experiment with automatic settings (exp. n° 11 in Table 5). FPS controlled by throughput to bandwidth ratio as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

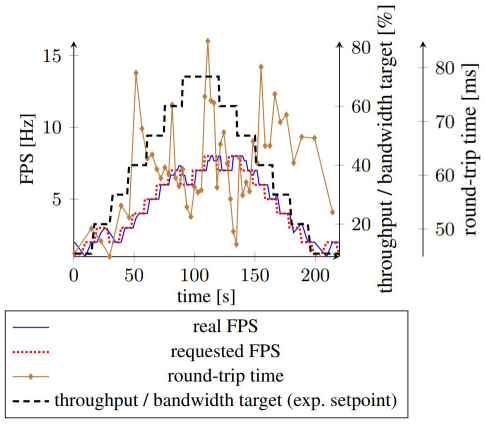

FIGURE 53. Video experiment with automatic settings (exp. n° 11 in Table 5). FPS controlled by throughput to bandwidth ratio as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

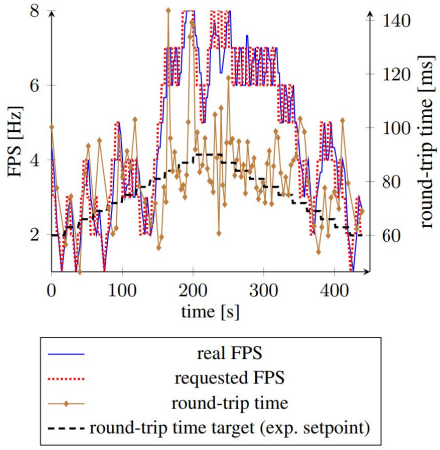

FIGURE 54. Video experiment with automatic settings (exp. n° 12 in Table 5). FPS controlled by round-trip time target as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

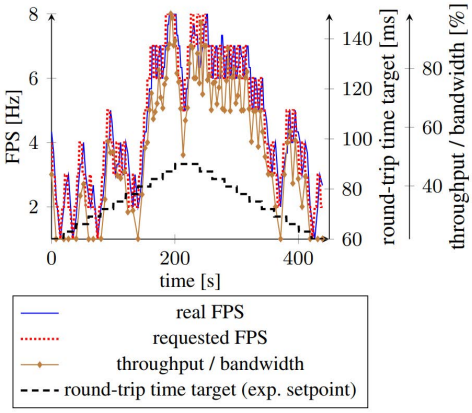

FIGURE 55. Video experiment with automatic settings (exp. n° 12 in Table 5). FPS controlled by round-trip time target as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

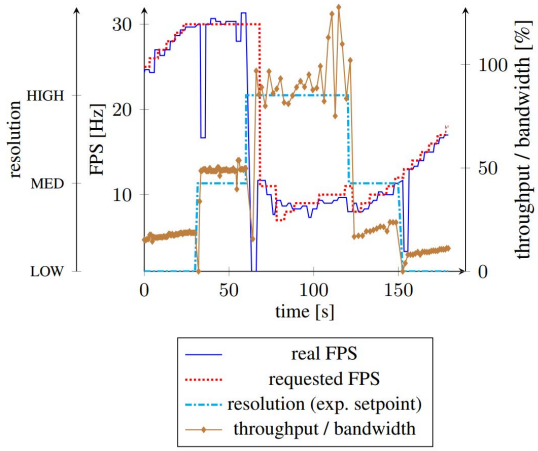

FIGURE 56. Video experiment with automatic settings (exp. n° 13 in Table 5). FPS controlled by resolution as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel.

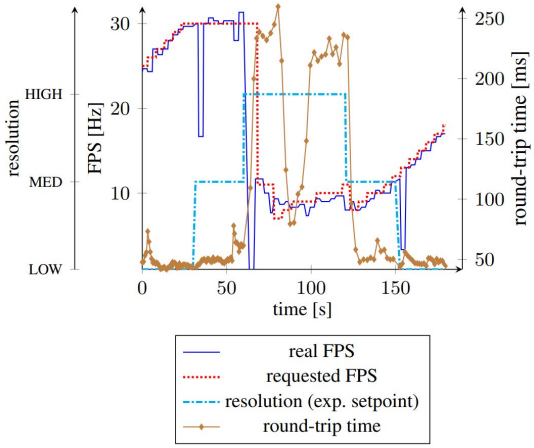

FIGURE 57. Video experiment with automatic settings (exp. n° 13 in Table 5). FPS controlled by resolution as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel.

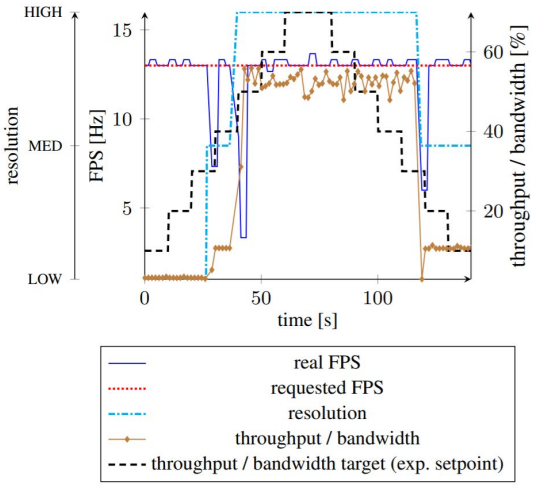

FIGURE 58. Video experiment with automatic settings (exp. n° 14 in Table 5). Resolution controlled by throughput to bandwidth ratio target as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 13 Hz.

11) VIDEO RESOLUTION CONTROLLED BY ROUND-TRIP TIME

The video round-trip time was changing over time, and the resolution was controlled by the automatic setting algorithm to follow the setpoint. The VPN 4G modem in the LHC tunnel connection type is presented. In Figures 60 and 61, the round-trip time setpoint was changing the resolution setting to achieve the target value. The round-trip time increased especially when the resolution changed from MED to HIGH.

12) VIDEO RESOLUTION CONTROLLED BY FPS

The video requested FPS was changing over time, and the resolution was controlled by the automatic setting algorithm to follow the setpoint. The VPN 4G modem in the LHC tunnel connection type is presented. In Figures 62 and 63, the resolution was changing in the function of the requested FPS. If the requested FPS could not be achieved, the resolution dropped, and correspondingly, if the FPS was achieved for a given resolution, an attempt to increase the resolution was made.

E. POINT CLOUD PROCESSING TIME MEASUREMENTS

The measured times of point cloud processing are shown in Table 7. The processing time increases with the size of the transmitted point cloud, and can even reach ∼245 ms for a 1 mm resolution point cloud. The total time does not include the network delay, which is communication type dependent, and the camera internal latency (ranging, for example, from 25 ms to 127 ms for the D435 camera and 848 × 100, according to [41]), which generally depends on the camera capture mode, frame rate and resolution. The execution time of automatic setting algorithms was less than 1 μs.

TABLE 7. Times of point cloud processing on the robot/server side (capturing, subsampling, serializing) and the operator/client side (deserializing, rendering, automatic settings). The times are presented in function of subsampling unit size for a fixed number of the initially acquired point cloud points.

Number of point cloud points before subsampling	Subsampling unit size [mm]	Number of point cloud points after subsampling	Capturing [ms]	Subsampling [ms]	Serializing [ms]	Deserializing [ms]	Rendering [ms]	Automatic settings [ms]	Total time [ms]
230400	1mm	230400	1.2	9.9	52.9	136.5	44.4	0.0	244.8
	5mm	87938		16.8	16.0	52.4	19.8		106.1
	10mm	39117		15.8	6.7	21.2	7.9		52.8
	20mm	10411		16.4	2.8	6.2	2.4		29.1

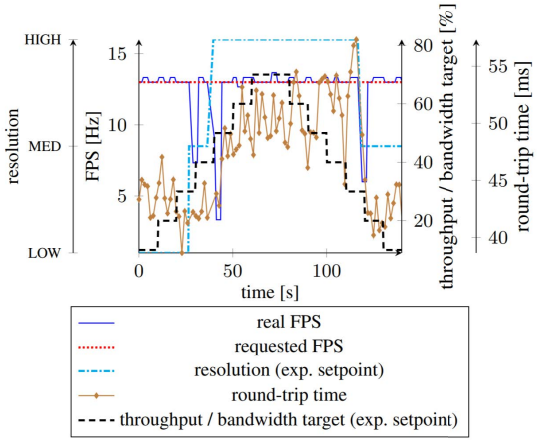

FIGURE 59. Video experiment with automatic settings (exp. n° 14 in Table 5). Resolution controlled by throughput to bandwidth ratio target as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 13 Hz.

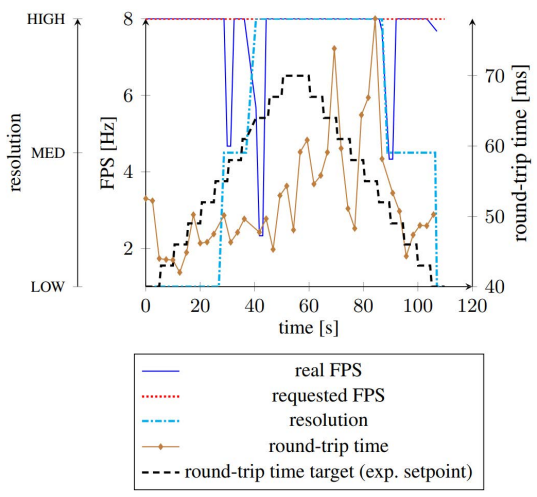

FIGURE 60. Video experiment with automatic settings (exp. n° 15 in Table 5). Resolution controlled by round-trip time target as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 8 Hz.

F. RELATION OF ROUND-TRIP TIME AND THROUGHPUT TO BANDWIDTH RATIO

The main difference between the bandwidth and round-trip time measurements is that the bandwidth measurement interrupts the teleoperation, taking some time and using the full bandwidth which sometimes might not be acceptable. The round-trip time measurement is done on-the-fly and it is not

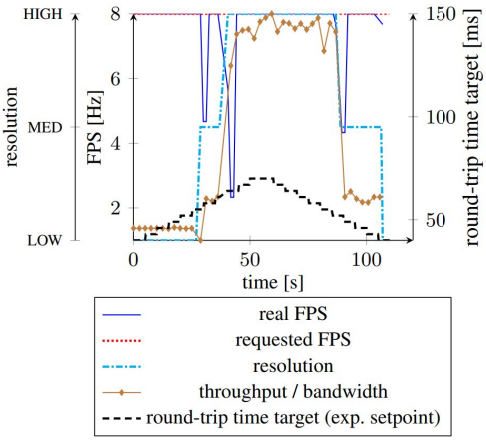

FIGURE 61. Video experiment with automatic settings (exp. n° 15 in Table 5). Resolution controlled by round-trip time target as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: FPS = 8 Hz.

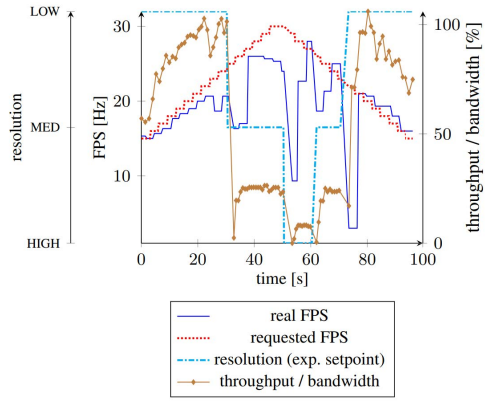

FIGURE 62. Video experiment with automatic settings (exp. n° 16 in Table 5). Resolution controlled by FPS as setpoint. Throughput to bandwidth ratio, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel.

intrusive. However, when the camera settings are constant, the throughput-to-bandwidth values are more stable over time than the round-trip time values, which are characterized by a jitter. The jitter magnitude depends on the connection type, as presented in Table 4. As described in Section II-J, the presented round-trip time was calculated with the Equation 3, which is a paramaterizable filter. Therefore, it is possible to change the weights, which would increase/decrease the response time though the oscillations would be minimized/reinforced accordingly. In the presented graphs, the standard weights were used (Equation 4).

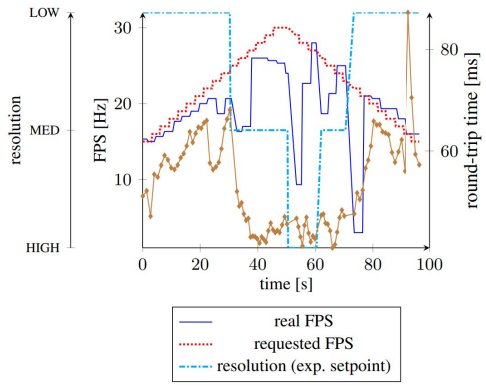

FIGURE 63. Video experiment with automatic settings (exp. n° 16 in Table 5). Resolution controlled by FPS as setpoint. Round-trip time, requested and real FPS behaviour presented. Connection type: VPN 4G modem in the LHC tunnel.

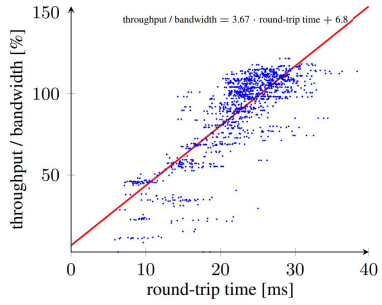

FIGURE 64. Point cloud experiment without automatic settings (exp. n° 1 in Table 5). Requested FPS as setpoint. Throughput to bandwidth ratio vs. round-trip time behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 10 mm.

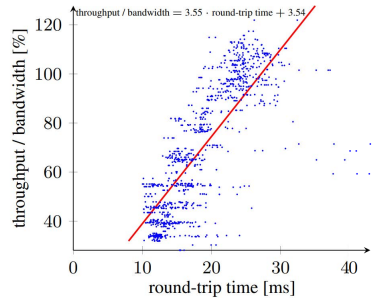

FIGURE 65. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Throughput to bandwidth ratio vs. round-trip time behaviour presented. Connection type: Wi-Fi over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 7 Hz.

During the experiments, the round-trip time was found to be correlated to the throughput to bandwidth ratio. The round-trip time generally increases when the throughput increases. Moreover, for certain situations, the values are proportional with a certain precision, indicating that the bandwidth measurement could be approximated by the round-trip time measurement. In Figures 64 and 65, the proportional correlations are sufficient. The used connection type was Wi-Fi and the point cloud acquisition was tested. The scattering depends on the variability of the network bandwidth and round-trip time jitter. However, as shown in Figures 66 and 67, the round-trip time changes with respect to the throughput to

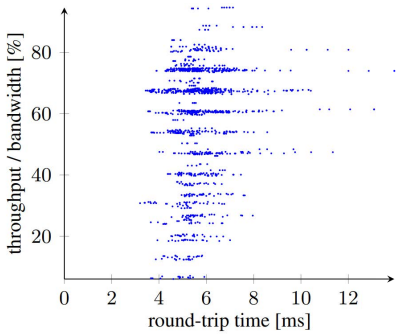

FIGURE 66. Point cloud experiment without automatic settings (exp. n° 1 in Table 5). Requested FPS as setpoint. Throughput to bandwidth ratio vs. round-trip time behaviour presented. Connection type: Ethernet over CERN GPN. Constant parameters: resolution = MEDIUM, subsampling = 3 mm.

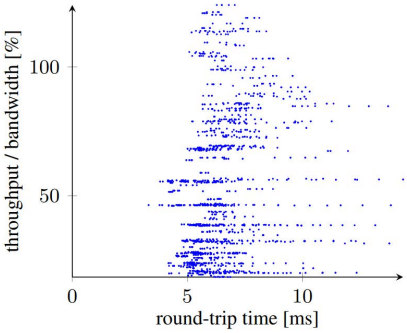

FIGURE 67. Point cloud experiment without automatic settings (exp. n° 2 in Table 5). Subsampling as setpoint. Throughput to bandwidth ratio vs. round-trip time behaviour presented. Connection type: Ethernet over CERN GPN. Constant parameters: resolution = MEDIUM, requested FPS = 15 Hz.

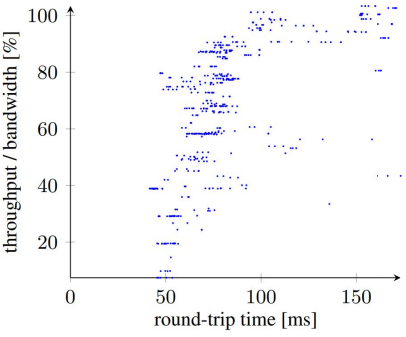

FIGURE 68. Video experiment without automatic settings (exp. n° 3 in Table 5). Requested FPS as setpoint. Throughput to bandwidth ratio vs. round-trip time behaviour presented. Connection type: VPN 4G modem in the LHC tunnel. Constant parameters: resolution = HIGH.

bandwidth ratio might be relatively big and with a significant jitter. The connection type used here was the Ethernet over CERN GPN and the point cloud was acquired. For the VPN 4G modem in the LHC tunnel connection type, the correlation is visible (Figure 68), however when almost full bandwidth is used, the round-trip time started oscillating with large amplitudes. Therefore, in the last 3 examples, the correlation cannot be used practically in real-time for the approximation of throughput-to-bandwidth ratio by the round-trip time.

TABLE 8. The summary of automatic settings modes: their desired intervention scenario uses, problems that are encountered when the automatic settings are not used and the improvement or solutions to problems when the automatic settings are used.

Scenarios of use	Problems for teleoperation	Adaptive congestion control type	Improvement thanks to the the adaptive congestion control	Limitation
Poor network coverage with highly limited bandwidth and its variation over time or in nearby locations. E.g. traversing a tunnel with scarcely distributed access points.	Because of variation, the network might get overloaded unexpectedly which leads to temporarily losing connection or stopping video or point cloud streams. Full bandwidth use increases delay multiple times which also impacts command safety.	**Settings controlled by throughput to bandwidth ratio**	The system adapts to environment change, especially the different number of point cloud points, while keeping the network load at the same value. The streaming is not collapsed because of overload, and the round-trip time is more stable.	If the bandwidth decreases significantly in different location, it has to be measured again. However, if the robot is stationary it does not require frequent measurements.
Precise teleoperation with fast-enough feedback from the robot to avoid collisions and to allow delicate manipulation. A strict round-trip time limitation is required. E.g. bilateral, force feedback control; operating in real-time close to abstacles or dynamic environment.	The round-trip time depends on the used network and varying load, often depending on external and uncontrolled factors. Varying round-trip time undermines the predictability and confuses the operator.	**Settings controlled by round-trip time**	This mode is most useful when the round-trip time parameter is crucial. It adapts fast to network parameters change and does not require bandwidth measurement.	Fast reaction of the algorithm keeping the round-trip timat at a level can change subsampling, resolution or FPS by a significant factor.
Constant rate of feedback is required, which translates into responsiveness of the robot. E.g. supervising the robot when it is performing an autonomous task.	Overloading the network can lead to a total collapse of stream due to buffering. Responsivess of the robot is highly deteriorated.	**Resolution or point cloud subsampling controlled by real FPS**	The resolution or point cloud subsampling are automatically adjusted to achieve requested FPS, while maintaining maximum available resolution or point cloud size. The collapse of stream is avoided. This mode does not use any network measurement.	The settings are controlled so that the bandwidth is used in full, therefore if there are other robots in the vicinity, it may create problems.
The teleoperator requires specific precision of point cloud or resolution of video with compromised but maximum available framerate. E.g. avoiding collisions with small obstacles such as cables, which are only visible with high resolution video or 1-3mm point cloud spacing resolution.	Network bandwidth constraints might not allow high enough framerates, thus a buffering of frames and a collapse of the stream may follow.	**Requested FPS controlled by point cloud subsampling or video resolution**	The buffering and stream collapse is avoided, while maintaining the required feedback precision. This mode does not use any network measurement and adapts fast to network changes.	As above, the drawback when the bandwidth is used in full.

These observations and conclusions are also important for the use of the automatic settings modes. From experience, the operator might choose the round-trip time based modes in the networks which have more stable parameters and the presented correlations are approximately linear, in order to have more stable behaviour of the camera acquisitions. While using a network which has variable round-trip times, the preferred modes can be based on the throughput measurement or even only on the internal acquisition settings and the real FPS feedback.

G. RESULTS SUMMARY

Table 8 summarizes in which scenarios the automatic settings modes should be used, what problems they solve, and what the improvements are due to the application of the adaptive congestion control.

H. VIDEO DEMONSTRATIONS

Various videos can further extend the understanding of the interface used in these experiments. At [42], there is a video showing how the teleoperation has been performed with the CERN 2D GUI for an intervention, as a reference and as a comparison to the solution of the Mixed Reality Human-Robot Interface presented in this paper. Secondly, at [43], there is a video showing the use of the interface in the same real scenarios executed by the same robot in the LHC. Lastly, at [44], there is a video showing the use of all the automatic settings modes by a teleoperator.

IV. DISCUSSION AND FUTURE WORK

The automatic settings provide an easier way of camera control for a non-expert operator who might not know exactly which parameter should be changed in case of, for example, network latency variation. Each control mode gives a different way of control, mitigating control effort. However, there is still room for simplification based on the control scenario that is used. For example, the scenario can be focused on precise manual manipulation, master-slave control, short or long distance between the robot and a target or operating in a very network-volatile environment. Thus, the setting would be rather a robotic scenario than a certain camera parameter that is controlled. The automatic settings selection can also depend on the interaction mode. In manual control, more FPS is needed; in a supervisory mode, a compromise between point cloud resolution and FPS is sufficient for collision avoidance and movement observation; in an autonomous mode the highest precision might be required.

The processing time of the point cloud (Section III-E) can change ∼8 times (from ∼245 ms to ∼29 ms) for the subsampling unit size increase from 1 mm to 20 mm. This delay has to be added to already affected network round-trip time by saturated bandwidth (Section III-D9, the round trip time can change from 50 ms to 250 ms when the throughput increases to the available maximum). In such a situation, the action of the automatic settings algorithm, based on network parameters, to increase the subsampling will have a positive impact on lowering the processing time. However, an automatic control based also on processing times measured on client and server sides may be added to the framework for delay-critical applications.

For further immersion increase using the Mixed Reality Human-Robot Interface for teleoperation in hostile environments, multiple advancements are proposed:

1) Apply the proposed framework to other operational scenarios, including those with very long and varying delays, such as the space telerobotics scenarios.
2) Study how the presence of multiple robots and/or other devices influences the available bandwidth while using the 4G network at CERN, particularly at different times of the day and what can cause a bottleneck.
3) In the higher level of telepresence with master-slave (bilateral) control and force feedback, the round-trip time is crucial. Therefore, a study using Augmented Reality with point cloud feedback should be done to verify if the proposition presented in this paper of automatic setting of subsampling and resolution based on round-trip time will be enough in the hazardous scenario.
4) Point cloud area streaming can be customized according to the current operator's viewpoint and look direction. Limiting which part of the cloud is sent according to the operators focus should be explored. Optionally, the point density can be increased in the observed area.
5) Explore point cloud stream density based on the distance of the operator's view to the point cloud area. As the rendering resolution is limited, saving on bandwidth can be achieved by lowering the density when the operator's position is far from point cloud.
6) The proposed collision detection and avoidance techniques greatly increase the operator's confidence and perception abilities to prevent collisions. However, for precise contact manipulations, a dedicated end effector force and torque sensor should be integrated.

V. CONCLUSION

The communication/network system is a key aspect of hazardous teleoperation scenarios where an operator requires a high amount of environmental awareness. An increased redundancy and complexity of the mobile manipulator can also increase the risk of collisions. These collisions can be avoided by the implemented Mixed Reality techniques, either by a spacial collision model and point cloud based avoidance, or detection of torque changes. However, these techniques require a significant amount of point cloud data sent from the robot to the control interface, which can be much larger than that of video streams. This can greatly impact the round-trip time and used bandwidth, and is why the automatic setting framework has been proposed and implemented in the CERN Mixed Reality interface.

The point cloud parameters need to be adjusted continuously to control the congestion of the network and flow. The adaptive congestion control in the application level has been designed to optimize the available bandwidth or to fulfil the specialized needs of the teleoperator. In the presented application context, it is possible to adjust the video and point cloud settings. These methods can be combined and used on top of the different compression methods which have been widely explored and are continuously improved. The automatic setting modes also enable the user to switch to the interaction mode that is best suited to the task.

The interface can improve the operator's environmental awareness and achieve better collision avoidance by providing immersion in the scene and automatic mechanisms to prevent or stop the movement when a collision is detected. Two types of collisions were taken into account: arm self-collision or collision with the robot's environment, both executed in real-time. The first type was based on the knowledge of the exact robot's model. The second type was based on the information taken from 3D point cloud acquisition or joints torque feedback.

Full network information was made available to the operator to facilitate the assessment of the current network conditions. The higher network bandwidth requirements due to the camera acquisition were met with algorithms of real-time video and 3D point cloud settings adaptation to the network. This allowed the optimization of the communication load and fulfilled the teleoperation goal with regards to camera feedback; whether a higher precision was needed, or a maximum delay to encounter, or maximum bandwidth use to manage. Twelve automatic setting modes were proposed with algorithms based on FPS, resolution, point cloud subsampling, round-trip time or throughput to bandwidth ratio. Each mode was thoroughly characterized and tested to present its specific use-case, advantages and how it lowers the operator's workload.

A framework was presented according to which designers can optimize their Human-Robot Interfaces and environmental video or point cloud feedback depending on the network characteristics and intervention scenarios. Based on the experience learnt during the development and experiments of Mixed Reality techniques for the described application and constraints, multiple further advancements or research were described. The solutions presented in this paper can also be applied in many other fields where network parameters vary and any Mixed Reality control using 3D environmental feedback is needed. These fields can be automotive environment sensing or remote driving, teleoperation in space or underwater, or terrestrial teleoperation over large distances.

APPENDIX A HUMAN-ROBOT INTERFACES - STATE-OF-THE-ART

TABLE A.1. Non-exhaustive state-of-the-art of Human-Robot Interfaces in terms of hazardous environments, 2D and 3D interfaces, if it is operational, communication type, complexity of manipulator, level of interaction, collision avoidance or detection, adaptive communication for shared and dynamic networks, and references.

Interface product	Hazardous environment	2D interfaces	3D user interface	Operational	Communication	Complexity of manipulator	Level of interaction	Collisions avoidance or detection	Adaptive communication for shared or dynamic network	References
Telerob tEODor, Telemax	Military applications, difficult terrain, used at CERN in radioactive zones	Video stream	X	Yes	Long-range radiofrequency	Telemax: 7 degrees of freedom	Manual teleoperation	X	X	[1], [2]
Fukushima robots: Monorobo, RESQ, SMERT, MARS	Nulear plant radioactively contaminated areas	Video stream	X	Yes	Cabled	Up to 6 degrees of freedom	Manual teleoperation	X	X	[3]
Rescue mobile robot Quince redesigned for Fukushima missions	Nulear plant radioactively contaminated areas	Video stream	3D laser scan view	Yes	Cabled, inter-robot wireless 2.4 GHz	2 degrees of freedom	Manual teleoperation	X	X	[4]
Mascot 4	Fusion energy tokamak JET	Video stream	Precisely modelled and simulated environment	Yes	Cabled	7 degrees of freedom of each manipulator	Manual operation with force feedback	Force feedback	X	[5]
TWINBOT	Underwater	Video stream in 2D interface	3D modelled robot and environment with 2D real-time perception	Yes, in lab scenario	Radiofrequency and sonar	4 degrees of freedom	Teleoperation, high-level supervision and autonomous	X	X	[6], [7], [8]. [9]
Perserverance rover	Mars	Video stream	Spatial mapping of the environment	Yes	Communication with throught Mars orbiters, UHF and X-band	5 degrees of freedom manipulator	Supervised and autonomous control	Manipulator-robot collisions avoidance in simulation before execution	The Mars Relay Operations Service for communication control	[10], [11]
CERN 2D GUI	Underground particle accelerator	Video stream	X	Yes	2G, 3G, 4G network, Wi-Fi	Up to 9 degrees of freedom	Manual teleoperation	Torqe information display	X	[12], [13]
CERN 3D GUI	Underground particle accelerator	Video stream shown in 3D interface	Fully modelled robot controlled in real tim in 3D space with 3D point cloud feedback	Yes	2G, 3G, 4G network, Wi-Fi	9 degrees of freedom	Real-time teleoperation, supervised position and trajectory control, target-based task specification	Collisions with robot, point cloud and self-collision avoidance in planning and real-time, additionally real-time collisions detection with torques	Automatic Congestion Control of camera feedback	Pilot project [13]

APPENDIX B
ROBOTIC INTERVENTION USE CASES

A standard teleoperation sequence using a Mixed Reality Interface during the Beam Loss Monitor measurement intervention scenario in the LHC consists of the following phases:

1) Before setting the camera(s) control parameters and moving the manipulator, the maximum bandwidth parameter must be tested to enable automatic control mode by throughput to bandwidth ratio (more in Section II-J). The bandwidth measurements can be repeated at any time on demand.
2) Setting up the camera(s) control parameters (Section II-I1) to achieve the desired point cloud or video resolution, frame rate, or automatic behaviour (Section II-I4). The position or transparency of the video feedback can be changed if it is obstructing the view.
3) Planning the manipulator's trajectory from the current real position to the desired position. The operator moves the planning manipulator using a convenient planning control mode (explained in Section II-D) and saves one or more waypoints. During planning, the operator will be notified if a self- or point cloud collision happens.
4) Before executing the movement to a selected waypoint, a collision check and preview are done to check if any self- or point cloud collisions occur (Section II-F).
5) The operator decide to execute the movement, acts on a control button and observes the environment with real-time point cloud and video feedback. The movement can be quickly interrupted by releasing the button.
6) Collision check and execution of movement are repeated for each waypoint. At any time the waypoint can be modified or removed.
7) The manipulator can be also controlled directly without planning (real-time direct control, Section II-D). The direct control allows adjusting a single joint or use the robot's inverse kinematics.
8) The final approach to the BLM device can also be accelerated by using the Target Approach feature (Section II-G) that allows any point cloud point to be selected, which creates an object which is normal to the surface and which can be adjusted using a gizmo (rotation or distance). Then, in the planning control mode the end effector can be moved exactly to this point with a single click.
9) At any time, there is an indicator showing a distance from the end effector of real, planning, preview or waypoint manipulators to any object recorded with point cloud in the scene. This helps the operator to predict any potential collisions or plan the movement precisely.

The use case diagram is shown in Figure B.1, which links graphically the phases described above.

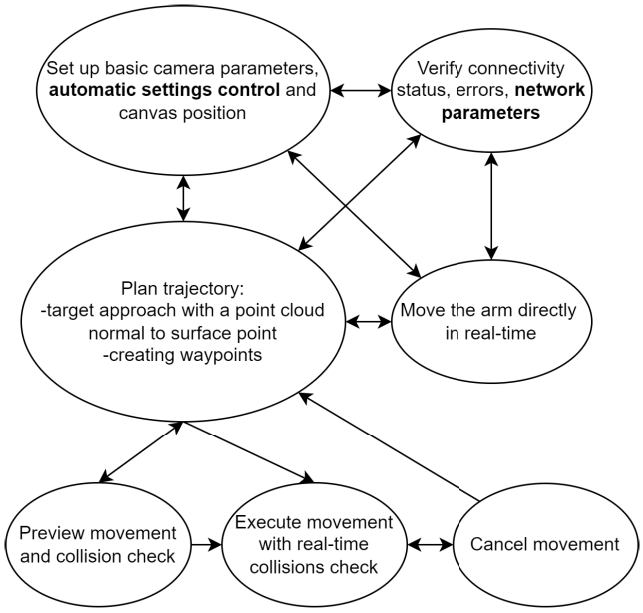

FIGURE B.1. Robotic intervention use cases with the Mixed Reality Interface. The parts taking part in the automatic end-to-end congestion control protocol for the video and point cloud feedback are highlighted.

APPENDIX C
AUTOMATIC CAMERA SETTING ALGORITHMS AND HOLOGRAM CONTROL VIEWS

FIGURE C.1. Hologram automatic setting for point cloud FPS controlled by throughput to bandwidth ratio mode.

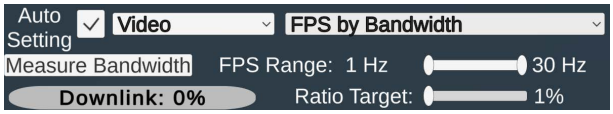

FIGURE C.2. Hologram automatic setting for video FPS controlled by throughput to bandwidth ratio mode.

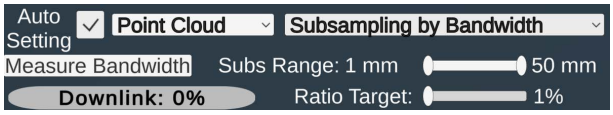

FIGURE C.3. Hologram automatic setting for point cloud subsampling unit size controlled by throughput to bandwidth ratio mode.

FIGURE C.4. Hologram automatic setting for video resolution controlled by throughput to bandwidth ratio mode.

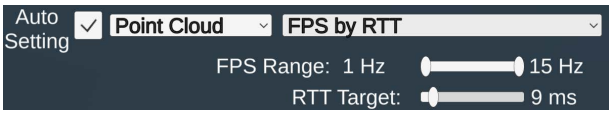

FIGURE C.5. Hologram automatic setting for point cloud FPS controlled by round-trip time mode.

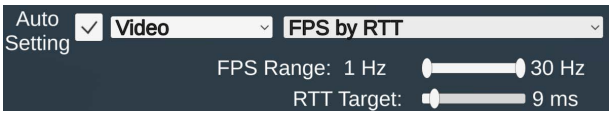

FIGURE C.6. Hologram automatic setting for video FPS controlled by round-trip time mode.

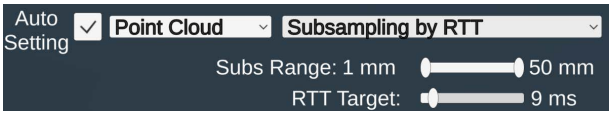

FIGURE C.7. Hologram automatic setting for point cloud subsampling unit size controlled by round-trip time mode.

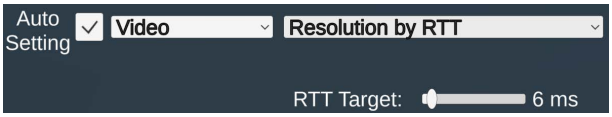

FIGURE C.8. Hologram automatic setting for video resolution controlled by round-trip time mode.

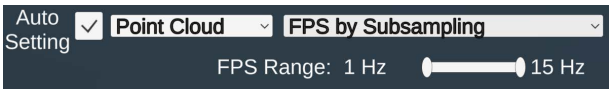

FIGURE C.9. Hologram automatic setting for point cloud FPS controlled by subsampling unit size mode.

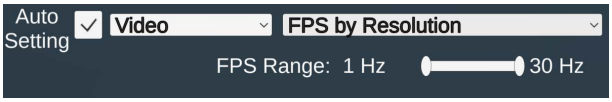

FIGURE C.10. Hologram automatic setting for video FPS controlled by resolution mode.

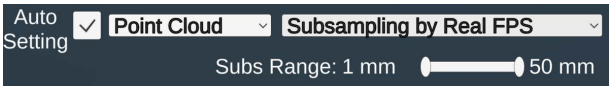

FIGURE C.11. Hologram automatic setting for point cloud subsampling unit size controlled by FPS mode.

FIGURE C.12. Hologram automatic setting for resolution controlled by FPS mode.

Algorithm 1 Algorithm of Automatic Point Cloud or Video FPS Setting Controlled by Throughput to Bandwidth Ratio

Inputs: Throughput: downlink throughput [Mbps],
Bandwidth: downlink bandwidth [Mbps],
RatioTarget: downlink throughput to downlink bandwidth ratio target,
ReqFPS: requested FPS [Hz],
MinFPS: minimum FPS limit [Hz],
MaxFPS: maximum FPS limit [Hz].
Output: NewFPS: new requested FPS [Hz].
Temp: CalcRatio: calculated throughput to bandwidth ratio.
Constants: DECR = 1 Hz: FPS decrement value,
INCR = 1 Hz: FPS increment value.

```
CalcRatio = Throughput / Bandwidth
if CalcRatio > RatioTarget then
    if ReqFPS + INCR ≤ MaxFPS then
        if ReqFPS + INCR ≤ MinFPS then
            NewFPS = MinFPS
        else
            NewFPS = ReqFPS + INCR
        end if
    else
        NewFPS = MaxFPS
    end if
else if ReqFPS − DECR ≥ MinFPS then
    if ReqFPS − DECR ≥ MaxFPS then
        NewFPS = MaxFPS
    else
        NewFPS = ReqFPS − DECR
    end if
else
    NewFPS = MinFPS
end if
```

Algorithm 2 Algorithm of Automatic Settings Subsampling Unit Size Controlled by Throughput to Bandwidth Ratio

Inputs: Throughput: downlink throughput [Mbps],
Bandwidth: downlink bandwidth [Mbps],
RatioTarget: downlink throughput to downlink bandwidth ratio target.
CurrSubs: current subsampling unit size [mm],
MinSubs: minimum subsampling limit [mm],
MaxSubs: maximum subsampling limit [mm].
Output: NewSubs: new subsampling unit size [mm].
Temp: CalcRatio: calculated throughput to bandwidth ratio.
Constants: DECR = 1 mm: subsampling decrement value.
INCR = 1 mm: subsampling increment value.

```
CalcRatio = Throughput / Bandwidth
if CalcRatio > RatioTarget then
    if CurrSubs + INCR ≤ MaxSubs then
        if CurrSubs + INCR ≤ MinSubs then
            NewSubs = MinSubs
        else
            NewSubs = CurrSubs + INCR
        end if
    else
        NewSubs = MaxSubs
    end if
else if CurrSubs − DECR ≥ MinSubs then
    if CurrSubs − DECR ≥ MaxSubs then
        NewSubs = MaxSubs
    else
        NewSubs = CurrSubs − DECR
    end if
else
    NewSubs = MinSubs
end if
```

Algorithm 3 Algorithm of Automatic Settings Video Resolution Controlled by Throughput to Bandwidth Ratio

Inputs: Throughput: downlink throughput [Mbps],
Bandwidth: downlink bandwidth [Mbps],
RatioTarget: downlink throughput to downlink bandwidth ratio target.
CurrResolution: current video resolution,
Output: NewResolution: new video resolution.
Constants: HIGH: high video resolution.
MEDIUM: medium video resolution.
LOW: low video resolution.
DIFF = 20%: difference between target and current throughput to bandwidth ratio.

```
CalcRatio = Throughput / Bandwidth
if CalcRatio − DIFF > RatioTarget then
    if CurrResolution == HIGH then
        NewResolution = MEDIUM
    else if CurrResolution == MEDIUM then
        NewResolution = LOW
    else if CurrResolution ≠ LOW then
        NewResolution = LOW
    end if
else if CalcRatio + DIFF < RatioTarget then
    if CurrResolution == LOW then
        NewResolution = MEDIUM
    else if CurrResolution == MEDIUM then
        NewResolution = HIGH
    else if CurrResolution ≠ HIGH then
        NewResolution = HIGH
    end if
end if
```

Algorithm 4 Algorithm of Automatic Point Cloud or Video FPS Setting Controlled by Round-Trip Time

Inputs: CurrRTT: current round-trip time [ms],
RTTTarget: round-trip time target [ms].
ReqFPS: requested FPS [Hz],
MinFPS: minimum FPS limit [Hz],
MaxFPS: maximum FPS limit [Hz].
Output: NewFPS: new requested FPS [Hz].
Constants: DECR = 1 Hz: FPS decrement value.
INCR = 1 Hz: FPS increment value.

```
if CurrRTT > RTTTarget then
    if ReqFPS + INCR ≤ MaxFPS then
        if ReqFPS + INCR ≤ MinFPS then
            NewFPS = MinFPS
        else
            NewFPS = ReqFPS + INCR
        end if
    else
        NewFPS = MaxFPS
    end if
else if ReqFPS − DECR ≥ MinFPS then
    if ReqFPS − DECR ≥ MaxFPS then
        NewFPS = MaxFPS
    else
        NewFPS = ReqFPS − DECR
    end if
else
    NewFPS = MinFPS
end if
```

Algorithm 5 Algorithm of Automatic Settings Subsampling Unit Size Controlled by Round-Trip Time

Inputs: CurrRTT: current round-trip time [ms],
RTTTarget: round-trip time target [ms].
CurrSubs: current subsampling unit size [mm],
MinSubs: minimum subsampling limit [mm],
MaxSubs: maximum subsampling limit [mm].
Output: NewSubs: new subsampling unit size [mm].
Constants: DECR = 1 mm: subsampling decrement value,
INCR = 1 mm: subsampling increment value.

```
if CurrRTT > RTTTarget then
    if CurrSubs + INCR ≤ MaxSubs then
        if CurrSubs + INCR ≤ MinSubs then
            NewSubs = MinSubs
        else
            NewSubs = CurrSubs + INCR
        end if
    else
        NewSubs = MaxSubs
    end if
else if CurrSubs − DECR ≥ MinSubs then
    if CurrSubs − DECR ≥ MaxSubs then
        NewSubs = MaxSubs
    else
        NewSubs = CurrSubs − DECR
    end if
else
    NewSubs = MinSubs
end if
```

Algorithm 6 Automatic Settings Video Resolution Controlled by Round-Trip Time Algorithm

Inputs: CurrRTT: current round-trip time [ms],
RTTTarget: round-trip time target [ms].
CurrResolution: current video resolution,
Output: NewResolution: new video resolution.
Constants: HIGH: high video resolution.
MEDIUM: medium video resolution.
LOW: low video resolution.
DIFF = 3 ms: difference between target and current round-trip time.

```
if CurrRTT − DIFF > RTTTarget then
    if CurrResolution == HIGH then
        NewResolution = MEDIUM
    else if CurrResolution == MEDIUM then
        NewResolution = LOW
    else if CurrResolution ≠ LOW then
        NewResolution = LOW
    end if
else if CurrRTT + DIFF < RTTTarget then
    if CurrResolution == LOW then
        NewResolution = MEDIUM
    else if CurrResolution == MEDIUM then
        NewResolution = HIGH
    else if CurrResolution ≠ HIGH then
        NewResolution = HIGH
    end if
end if
```

Algorithm 7 Algorithm of Automatic Point Cloud FPS Setting Controlled by Subsampling Unit Size or Video FPS Setting Controlled by Resolution

Inputs: ReqFPS: requested FPS [Hz],
RealFPS: real FPS [Hz],
MinFPS: minimum FPS limit [Hz],
MaxFPS: maximum FPS limit [Hz].
Output: NewFPS: new requested FPS [Hz].
Temp: CalcFPS: calculated new requested FPS [Hz].
Constants: DIFF = 2 Hz: difference between new and current FPS.
INCR = 1 Hz: FPS increment value.

if ReqFPS − RealFPS ≥ DIFF **then**
  if RealFPS ≥ MaxFPS **then**
    NewFPS = MaxFPS
  else if RealFPS ≤ MinFPS **then**
    NewFPS = MinFPS
  else
    NewFPS = RealFPS
  end if
else if ReqFPS − RealFPS ≤ 0 **then**
  if ReqFPS + INCR ≤ MaxFPS **then**
    if ReqFPS + INCR ≤ MinFPS **then**
      NewFPS = MinFPS
    else
      NewFPS = ReqFPS + INCR
    end if
  else
    NewFPS = MaxFPS
  end if
end if

Algorithm 8 Algorithm of Automatic Settings Point Cloud Subsampling Unit Size Controlled by FPS

Inputs: ReqFPS: requested point cloud FPS [Hz],
RealFPS: real point cloud FPS [Hz],
CurrSubs: current subsampling unit size [mm],
MinFPS: minimum FPS limit [Hz],
MaxFPS: maximum FPS limit [Hz].
Output: NewSubs: new subsampling unit size [mm].
Temp: CalcSubs: calculated new subsampling unit size [mm].
Constants: DIFF = 2 Hz: difference between new and current FPS.
DECR = 1 mm: subsampling decrement value.

if ReqFPS − RealFPS > DIFF **then**
  CalcSubs = CurrSubs · (1 + 0.1 · (ReqFPS − RealFPS) / RealFPS)
  if CalcSubs ≤ MaxFPS **then**
    if CalcSubs ≤ MinFPS **then**
      NewSubs = MinFPS
    else
      NewSubs = CalcSubs
    end if
  else
    NewSubs = MaxFPS
  end if
else if ReqFPS − RealFPS ≤ 0 **then**
  CalcSubs = CurrSubs - DECR
  if CalcSubs ≥ MinFPS **then**
    if CalcSubs ≥ MaxFPS **then**
      NewSubs = MaxFPS
    else
      NewSubs = CurrSubs − DECR
    end if
  else
    NewSubs = MinFPS
  end if
end if

Algorithm 9 Automatic Settings Video Resolution Controlled by Video FPS Algorithm

Inputs: ReqFPS: requested point cloud FPS [Hz],
RealFPS: real point cloud FPS [Hz],
CurrResolution: current video resolution,
Output: NewResolution: new video resolution.
Constants: HIGH: high video resolution.
MEDIUM: medium video resolution.
LOW: low video resolution.
DIFF = 4: difference between real and requested FPS.

if ReqFPS − RealFPS > DIFF **then**
  if CurrResolution == HIGH **then**
    NewResolution = MEDIUM
  else if CurrResolution == MEDIUM **then**
    NewResolution = LOW
  else if CurrResolution ≠ LOW **then**
    NewResolution = LOW
  end if
else if ReqFPS − RealFPS ≤ 0 **then**
  if CurrResolution == LOW **then**
    NewResolution = MEDIUM
  else if CurrResolution == MEDIUM **then**
    NewResolution = HIGH
  else if CurrResolution ≠ HIGH **then**
    NewResolution = HIGH
  end if
end if

ACKNOWLEDGMENT

The authors thank to the Interactive Robotics Laboratory (IRSLab), Jaume I University of Castellon, for the exchange of ideas and discussions related to telemanipulation systems.

REFERENCES

[1] J. Carff, M. Johnson, E. M. El-Sheikh, and J. E. Pratt, "Human-robot team navigation in visually complex environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 3043–3050.
[2] T. Sakaue, S. Yoshino, K. Nishizawa, and K. Takeda, "Survey in Fukushima daiichi NPS by combination of human and remotely-controlled robot," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot. (SSRR)*, Oct. 2017, pp. 7–12.
[3] S. Kawatsuma, R. Mimura, and H. Asama, "Unitization for portability of emergency response surveillance robot system: Experiences and lessons learned from the deployment of the JAEA-3 emergency response robot at the Fukushima Daiichi nuclear power plants," *ROBOMECH J.*, vol. 4, no. 1, pp. 1–7, Dec. 2017, doi: 10.1186/s40648-017-0073-7.
[4] M. Bajracharya, M. W. Maimone, and D. Helmick, "Autonomy for Mars rovers: Past, present, and future," *Computer*, vol. 41, no. 12, pp. 44–50, Dec. 2008.
[5] M. Panzirsch, H. Singh, T. Kruger, C. Ott, and A. Albu-Schaffer, "Safe interactions and kinesthetic feedback in high performance earth-to-moon teleoperation," in *Proc. IEEE Aerosp. Conf.*, Mar. 2020, pp. 1–10.
[6] R. Buckingham and A. Loving, "Remote-handling challenges in fusion research and beyond," *Nature Phys.*, vol. 12, no. 5, pp. 391–393, May 2016, doi: 10.1038/nphys3755.
[7] C. P. Sesmero, S. V. Lorente, and M. Di Castro, "Graph SLAM built over point clouds matching for robot localization in tunnels," *Sensors*, vol. 21, no. 16, p. 5340, Aug. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/16/5340
[8] M. Di Castro, G. Lunghi, A. Masi, M. Ferre, and R. M. Prades, "A multi-dimensional RSSI based framework for autonomous relay robots in harsh environments," in *Proc. 3rd IEEE Int. Conf. Robotic Comput. (IRC)*, Feb. 2019, pp. 183–188.
[9] M. de la Cruz, G. Casañ, P. Sanz, and R. Marín, "Preliminary work on a virtual reality interface for the guidance of underwater robots," *Robotics*, vol. 9, no. 4, p. 81, Oct. 2020. [Online]. Available: https://www.mdpi.com/2218-6581/9/4/81

[10] K. Kanamori, N. Sakata, T. Tominaga, Y. Hijikata, K. Harada, and K. Kiyokawa, "Obstacle avoidance method in real space for virtual reality immersion," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Oct. 2018, pp. 80–89.
[11] A. Martín-Barrio, J. J. Roldán-Gómez, I. Rodríguez, J. del Cerro, and A. Barrientos, "Design of a hyper-redundant robot and teleoperation using mixed reality for inspection tasks," *Sensors*, vol. 20, no. 8, p. 2181, Apr. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/8/2181
[12] A. Aristidou and J. Lasenby, "FABRIK: A fast, iterative solver for the Inverse Kinematics problem," *Graph. Models*, vol. 73, no. 5, pp. 243–260, Sep. 2011. 10.1016/j.gmod.2011.05.003
[13] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE Trans. Inf. Syst.*, vol. 77, no. 12, pp. 1321–1329, Dec. 1994.
[14] P. Milgram, H. Takemura, A. Utsumi, and F. Kishino, "Augmented reality: A class of displays on the reality-virtuality continuum," *Proc. SPIE*, vol. 2351, pp. 282–292, Dec. 1994.
[15] P. Milgram and H. W. Colquhoun, "A framework for relating head-mounted displays to mixed reality displays," *Proc. Hum. Factors Ergonom. Soc. Annu. Meeting*, vol. 43, no. 22, pp. 1177–1181, Sep. 1999, doi: 10.1177/154193129904302202.
[16] R. Wirz, R. Marín, J. M. Claver, M. Ferre, R. Aracil, and J. Fernández, "End-to-end congestion control protocols for remote programming of robots, using heterogeneous networks: A comparative analysis," *Robot. Auto. Syst.*, vol. 56, no. 10, pp. 865–874, Oct. 2008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889008000845
[17] P. Gorczak, C. Bektas, F. Kurtz, T. Lubcke, and C. Wietfeld, "Robust cellular communications for unmanned aerial vehicles in maritime search and rescue," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Sep. 2019, pp. 229–234.
[18] D. Laniewski and N. Aschenbruck, "On the potential of rate adaptive point cloud streaming on the point level," in *Proc. IEEE 46th Conf. Local Comput. Netw. (LCN)*, Oct. 2021, pp. 49–56.
[19] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 133–148, Mar. 2019.
[20] T. Huang and Y. Liu, "3D point cloud geometry compression on deep learning," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 890–898.
[21] M. Quach, G. Valenzise, and F. Dufaux, "Learning convolutional transforms for lossy point cloud geometry compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 4320–4324.
[22] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 778–785.
[23] M. Hosseini and C. Timmerer, "Dynamic adaptive point cloud streaming," in *Proc. 23rd Packet Video Workshop*, Jun. 2018, pp. 25–30, doi: 10.1145/3210424.3210429.
[24] C. Moreno, Y. Chen, and M. Li, "A dynamic compression technique for streaming Kinect-based point cloud data," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 550–555.
[25] C. Moreno and M. Li, "A progressive transmission technique for the streaming of point cloud data using the Kinect," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Mar. 2018, pp. 593–598.
[26] A. Akhtar, B. Kathariya, and Z. Li, "Low latency scalable point cloud communication," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2369–2373.
[27] M. Di Castro, L. R. Buonocore, M. Ferre, S. Gilardoni, R. Losito, G. Lunghi, and A. Masi, "A dual arms robotic platform control for navigation, inspection and telemanipulation," in *Proc. 16th Int. Conf. Accel. Large Experim. Phys. Control Syst.*, 2018, pp. 1–5.
[28] M. Di Castro, M. L. B. Tambutti, M. Ferre, R. Losito, G. Lunghi, and A. Masi, "I-TIM: A robotic system for safety, measurements, inspection and maintenance in harsh environments," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Aug. 2018, pp. 1–6.
[29] J. M. Garcés, C. V. Almagro, G. Lunghi, M. Di Castro, L. R. Buonocore, R. M. Prades, and A. Masi, "MiniCERNBot educational platform: Antimatter factory mock-up missions for problem-solving STEM learning," *Sensors*, vol. 21, no. 4, p. 1398, Feb. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1398
[30] 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.
[31] M. Di Castro, M. Ferre Pérez, and A. Masi, "A novel robotic framework for safe inspection and telemanipulation in hazardous and unstructured environments," Ph.D. dissertation, Dept. Automat., Elect. Electron. Eng. Comput. Sci., Universidad Politécnica de Madrid, Madrid, Spain, 2019. [Online]. Available: http://cds.cern.ch/record/2708835
[32] G. Lunghi, R. Marin, M. Di Castro, A. Masi, and P. J. Sanz, "Multimodal human-robot interface for accessible remote robotic interventions in hazardous environments," *IEEE Access*, vol. 7, pp. 127290–127319, 2019.
[33] C. V. Almagro, G. Lunghi, M. Di Castro, D. C. Beltran, R. M. Prades, A. Masi, and P. J. Sanz, "Cooperative and multimodal capabilities enhancement in the CERNTAURO human–robot interface for hazardous and underwater scenarios," *Appl. Sci.*, vol. 10, no. 17, p. 6144, Sep. 2020. [Online]. Available: https://www.mdpi.com/2076-3417/10/17/6144
[34] D. Ribas, P. Ridao, A. Turetta, C. Melchiorri, G. Palli, J. J. Fernández, and P. J. Sanz, "I-AUV mechatronics integration for the TRIDENT FP7 project," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2583–2592, Oct. 2015.
[35] A. Solis, R. Marin, J. Marina, F. J. Moreno, M. Ávila, M. D. L. Cruz, D. Delgado, J. V. Marti, and P. J. Sanz, "An underwater simulation server oriented to cooperative robotic interventions: The educational approach," in *Proc. OCEANS*, San Diego, CA, USA, Sep. 2021, pp. 1–6.
[36] C. V. Almagro, M. Di Castro, G. Lunghi, R. M. Prades, P. J. S. Valero, M. F. Pérez, and A. Masi, "Monocular robust depth estimation vision system for robotic tasks interventions in metallic targets," *Sensors*, vol. 19, no. 14, pp. 1–28, 2019.
[37] K. Szczurek, R. Prades, E. Matheson, H. Perier, L. Buonocore, and M. Di Castro, "From 2D to 3D mixed reality human-robot interface in hazardous robotic interventions with the use of redundant mobile manipulator," in *Proc. 18th Int. Conf. Informat. Control, Autom. Robot.*, 2021, pp. 388–395.
[38] R. Schmidt, R. Assmann, E. Carlier, B. Dehning, R. Denz, B. Goddard, E. B. Holzer, V. Kain, B. Puccio, B. Todd, and J. Uythoven, "Protection of the CERN large hadron collider," *New J. Phys.*, vol. 8, p. 290, 2006.
[39] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th ed. Boston, MA, USA: Pearson, 2016.
[40] S. Agosta, R. Sierra, and F. Chapron, "High-speed mobile communications in hostile environments," *J. Phys. Conf.*, vol. 664, no. 5, p. 52001, 2015, doi: 10.1088/1742-6596/664/5/052001.
[41] T. Sonoda, J. N. Sweetser, T. Khuong, S. Brook, and A. G.-J. Rev. *High-Speed Capture Mode of Intel RealSenseT Depth Camera D435*. Accessed: Jun. 28, 2022. https://dev.intelrealsense.com/docs/high-speed-capture-mode-of-intel-realsense-depth-camera-d435
[42] (2022). CERN Robotics Team BE-CEM-MRO. *Video Demonstration: Beam Loss Monitors Robotic Measurements With 2D GUI*. [Online]. Available: https://videos.cern.ch/record/2295957
[43] K. A. Szczurek. (2022). *Video Demonstration: Summary of Mixed Reality Human-Robot Interface With Adaptive Communications Congestion Control for the Teleoperation of Mobile Redundant Manipulators in Hazardous Environments*. [Online]. Available: https://videos.cern.ch/record/2295954
[44] K. Szczurek. (2022). *Video demonstration: Adaptive Communications Congestion Control for Mixed Reality Human-Robot Interface*. [Online]. Available: https://videos.cern.ch/record/2295956

KRZYSZTOF ADAM SZCZUREK received the M.Sc.Eng. degree in control engineering and robotics from the Wrocław University of Science and Technology, Poland, in 2017. He is currently pursuing the Ph.D. degree in computer science and robotics with the Jaume I University of Castellon, Spain. From 2013 to 2014, he worked with American Axle & Manufacturing on industrial controls for the automotive sector. In 2017, he worked with Nokia on the 5G communication technology. From 2015 to 2016, and since 2017, he has been with the CERN, working on automation, control software and robotics projects. He is passionate about mixed reality human-robot interfaces, and specialized in design and implementation of control systems consisting of complex and state-of-the-art solutions, based on PLC/SCADA and real-time systems (C++/C#/Unity and LabVIEW).

RAUL MARIN PRADES 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 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 worked with 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) and worked as a Researcher, a Software Developer, and a Software Architect at the Switching and Access Division. In 1999, he began to teach and research as a Professor at the Jaume I University of Castellon. Since 2009, he has been working as an Associate Professor at the Department of Computer Science, Jaume I University of Castellon, where he lectures computer networking, and robotics. He has been appointed as a Visiting Scientist at Blaise Pascal University, in 2002, the University Polithecnique of Madrid, in 2007, the University Federal of Brasilia, in 2016, and European Organization for Nuclear Research (CERN) (2015, 2018, 2019, 2020, and 2021–2022), studying new techniques for telemanipulation in hazardous environments. He has teaching experience in computer science engineering degree, the intelligent systems master, and the EU EMARO advanced robotics master, among others. He has participated in research projects, such as FP6 GUARDIANS (group of unmanned assistant robots deployed in aggregative navigation supported by scent detection), FP7 TRIDENT Project (marine robots and dexterous manipulation for enabling autonomous underwater multipurpose intervention missions), and H2020 El-Peacetolero (embedded electronic solutions for polymer innovative scanning tools using light emitting devices for diagnostic routines). His research interests include robotics and rescue and underwater, including subjects such as localization, networks of sensors and actuators, object recognition, telerobotics, and education. He is the author or coauthor of around 150 research publications on these subjects.

ELOISE MATHESON received the B.Sc./B.Eng. degrees in mechatronics (space) engineering from the University of Sydney Australia, in 2010, the M.Sc. degree in advanced robotics from the École Central de Nantes, France, in 2014, and the Ph.D. degree in surgical robotics from Imperial College London, U.K., in 2021. The subject of the Ph.D., was the research, development and clinical evaluation of the human–machine interface of a novel steerable soft catheter for neurosurgery. From 2014 to 2016, she was an Engineer at the Telerobotics and Haptics Laboratory, European Space Agency, The Netherlands largely working on telerobotic activities under the METERON Project. In 2020, she joined the Mechatronics, Robotics and Operations Section, CERN, as a Mechatronics Engineer, working on beam intercepting device mechatronic systems and the development and integration of robotic solutions in the accelerator complex. Her research interests include tele-operation, supervisory control and autonomous operations, haptics, and human–machine interfaces.

JOSE RODRIGUEZ-NOGUEIRA received the B.Sc. degree in electronics, robotics and mechatronics engineering from the University of Malaga, Spain, in 2020. The subject of his Degree Dissertation was about the implementation of visual odometry in a one to ten scale autonomous vehicle in ROS and GAZEBO. In September of 2020, he started the Automatic and Robotic Master with the Politechnic University of Madrid, which he is still finishing. He realized an internship with the Vision & Aerial Robotics Group, Center of Automatics and Robotics, Madrid. There he implemented a unity UAV simulator into a ROS framework called Aerostack. He also worked on the implementation of an inertial and visual odometry algorithm to estimate the UAV trajectory. In July of 2021, he started a trainee internship at CERN developing a robot teleoperation mixed reality GUI.

MARIO DI CASTRO received the M.Sc. degree in electronic engineering from the University of Naples ''Federico II,'' Italy, and the Ph.D. degree in robotics and industrial controls from the Polytechnic University of Madrid, Spain. From 2005 to 2006, he was an Intern and a Technical Student at the CERN in charge of advanced magnetic measurements and studies for LHC superconducting magnets. From 2007 to 2011, he works at EMBL c/o DESY in charge of advanced mechatronics solutions for synchrotron beamlines controls. Since 2011, he has been working at the CERN, where he leads the Mechatronics, Robotics and Operation Section. The section is responsible for the design, installation, operation, and maintenance of advanced control systems based on different control platforms for movable devices characterized by few um positioning accuracy (e.g. scrapers, collimators, goniometers and target) in harsh environment. Important section activities are the design, construction, installation, operation, and maintenance of robotic systems used for remote maintenance in the whole CERN accelerator complex and quality assurance. His research interests include modular robots, tele-robotics, human–robot interfaces, machine learning, enhanced reality, automatic controls, mechatronics, precise motion control in harsh environment, and advanced robotics also for search and rescue scenarios.