Digital twin- and extended reality-based telepresence for collaborative robot programming in the 6G perspective

Digital twin- and extended reality-based telepresence for collaborative robot programming in the 6G perspective

Davide Calandra Gabriele Pratticò Alberto Cannavò Claudio Casetti
Fabrizio Lamberti [1]

[1]Politecnico di Torino, Turin, Italy
{firstname.lastname}@polito.it

Abstract

In the context of Industry 4.0, a paradigm shift from traditional industrial manipulators to Collaborative Robots (CRs) is ongoing, with the latter serving ever more closely humans as auxiliary tools in many production processes. In this scenario, continuous technological advancements offer new opportunities for further innovating robotics and other areas of next-generation industry. For example, 6G could play a prominent role due to its human-centric view of the industrial domains. In particular, its expected dependability features will pave the way for new applications exploiting highly effective Digital Twin (DT)- and eXtended Reality (XR)-based telepresence. In this work, a novel application for the above technologies allowing two distant users to collaborate in the programming of a CR is proposed. The approach encompasses demanding data flows (e.g., point cloud-based streaming of collaborating users and robotic environment), with network latency and bandwidth constraints. Results obtained by analyzing this approach from the viewpoint of network requirements in a setup designed to emulate 6G connectivity indicate that the expected performance of forthcoming mobile networks will make it fully feasible in principle.

1 Introduction

In the ongoing industrial revolution, robotics is playing a pivotal role. According to [22], it is possible to distinguish three major research directions in this field. The first direction is industrial robotics, which concerns the domain of re-programmable and multi-purpose robotic manipulators [48]. The second direction considers service robotics, which encompasses mobile autonomous/semi-autonomous robots operating in various fields of human activities. Finally, the third direction is represented by collaborative robotics. This direction represents the new stage in the development of industrial robots, in which Collaborative Robots (CRs), often abbreviated *cobots*, closely interact with humans.

Our paper focuses on the latter direction, as it is regarded as capable of combining the benefits of automating repetitive jobs by using cobots with the flexibility deriving from manual labor performed by a human operator [19] making them work in tandem. In particular, the paper specifically considers a key task in robotics, that is robot programming.

In traditional industrial robot programming, the human contribution is only provided outside the production environment: this approach is usually referred to as offline programming. The detachment between the programming and the production phases may lead to poorly flexible programs. In fact, while executed, these programs cannot be altered. For instance, considering a pick-and-place task, a small delay in the pipeline may lead to an offset in the position of the object to be grabbed by the robot. Due to the fact that it is not possible to modify the program, the task could fail as offline programmed robot's movements occur at a predefined pace. Conversely, the humans directly participate in CR programming, making the resulting programs more flexible and effective [19].

As a matter of example, in the scenario described above, an operator could intervene in the task and specify the new position to be reached by the robot by grabbing it with the hands, activating its movements at the correct time using voice commands, and possibly instructing it on the actions to be executed in case of failures by directly performing them. Considering these possibilities, it is not surprising that several paradigms have been proposed to let operators create and modify robot programs by using hand guidance [52], voice and gestures [27], or through learning by demonstration [61]. In general, it is quite a common practice for a user, when programming a CR, to register the path to follow by

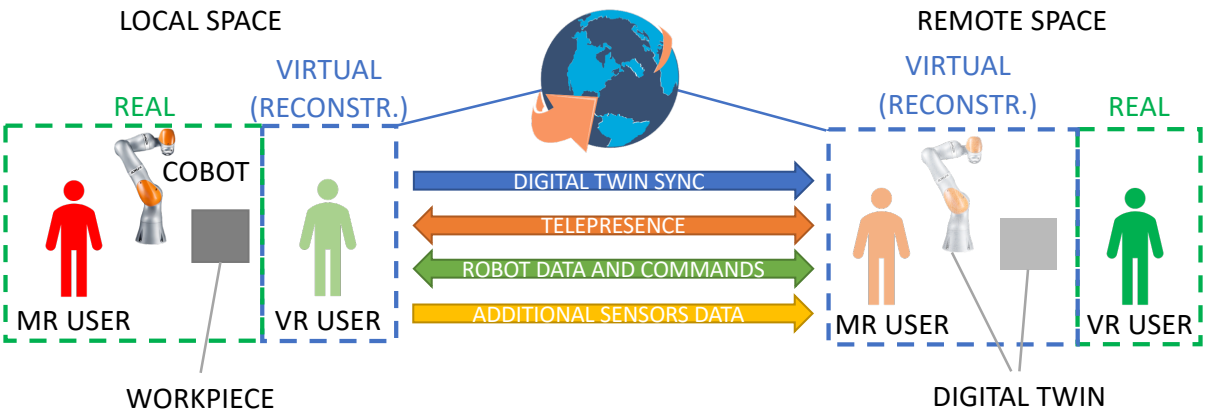


Figure 1: Proposed concept for the novel use of XR and DT technologies for collaborative robotics: A user, co-located with a CR and endowed with MR capabilities, alternates in the programming of the robot with a remote user that operates in immersive VR.

manually guiding the robot (i.e., physically manipulating its parts) instead of moving it with classic axial or Cartesian commands issued by a dedicated control device [8]. However, even considering trivial pick-and-place operations, there could be some tasks in which hand-guiding the CR to the desired position is unfeasible (because that position is not physically reachable by the user), dangerous or simply impractical. In these cases, it is possible to look at the benefits brought by approaches based on *telepresence*, in which a remote user is able to remotely control or program the robot without having to co-locate with it.

Furthermore, it is worth observing that a robot program intended to be used in production is not just made of movements but also of actions the robot should perform (e.g., open/close a gripper tool); in the case of CRs, there could also be some production cycle tasks where it is necessary to alternate CR's duties with actions performed by an operator that share the same working space with the robot. In this respect, cutting-edge technologies like *eXtended Reality* (XR) can offer new opportunities to enhance current programming paradigms. More specifically, XR can provide tools for building digital environments in which intuitive 3D interaction methods and visual feedbacks can be used to allow one or more users to collaborate in CR programming.

The term XR is used to refer to environments combining real and virtual elements, which can be experienced with computing and/or wearable equipment. This sort of "umbrella" term encompasses technologies like Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), and their overlapping fields [59]. Focusing on visualization aspects, VR, AR, and MR belong to what is commonly referred to as the Reality – Virtuality continuum, a spectrum of approaches that can be pursued to show virtual and real objects on display [40]. Based on the definitions given in [9], the goal of VR is to let the users visualize a fully synthetic environment made of digitally reconstructed objects. Instead, with AR, users have the possibility to visualize the "augmented" real environment with digital contents that are superimposed on real objects. Finally, according to [55], MR differs from AR in presenting the digital contents aligned to the real elements.

A system that leverages MR and VR technologies could actually help the users in a CR programming task. For instance, users co-located with the robot could wear MR Head-Mounted Displays (HMDs) to visualize information regarding the working area or the robot's operating conditions (e.g., the status of components, planned trajectories, etc.) while physically interacting with it. At the same time, users equipped with a VR HMD could virtually program the robot without the need to share the same physical space with it.

The real environment, as well as the current robot's conditions and workpieces, could be recreated by making use of two approaches: 3D reconstructions and Digital Twin-based (DT) methods. The first approach relies on virtual models that can be created by using technologies such as graphics software and 3D scanners. Approaches based on DTs, in turn, encompass virtual representations used as digital counterparts of physical objects. For instance, the data gathered by sensors embedded in a robot or placed on the users could be leveraged to update their virtual replicas in real-time [28].

A collaborative approach to CR programming like the one described above, involving users that could be located on-site or remote and combining MR and VR technologies, would be crucial to the success of many next-generation robotics scenarios, as it could optimize the use of resources and foster the development of a sustainable industry. For instance, its application could avoid unnecessary travels of

human operators. Notwithstanding, to the best of the authors' knowledge, such a collaborative approach to CR programming has not been considered in real industrial settings yet. Motivations may be related to the constraints set by current network technologies. In fact, in order to enable collaboration, the devised approach would require reliable connections between the users located remotely and those on-site. However, current networks still present limitations in terms of supported data rates and achievable end-to-end latencies. These limitations could prevent the deployment of remote-control-based XR applications like the described one, which imposes strict and highly deterministic constraints [20].

To cope with the limitations set by current network technologies, also considering the forecasts of initiatives like the European flagship Hexa-X project [17, 58], an increasing number of researchers and practitioners have started analyzing the features of 6G networks to understand their affordances [25, 28, 67]. A particularly interesting aspect is the human-centric vision of 6G [28, 38, 51]. In fact, 5G technology is already enabling new types of services, like high-definition movie streaming, Internet Protocol TV broadcasting, and distributed XR applications. However, new needs are emerging concerning the quality of the physical user experience in such applications [35, 51]. According to this human-centric vision first formalized in [14], humans and their requirements will shape next-generation networking services and applications. In the field of industry, one impact of this vision is the possibility to create applications through 6G, in which the users can benefit from a higher sense of embodiment within the work environment thanks to networks characterized by high-capacity and low-latency connections compared to currently available ones [28].

By building on the above vision and the use cases envisaged for 6G in [25], the present paper proposes and evaluates the feasibility of a novel XR- and DT-based system for CR programming by analyzing its bandwidth and latency requirements using a network setup that emulates the characteristic of the forthcoming 6G networks.

In particular, the paper focuses on a concept scenario supported by XR-based telepresence, in which distant operators act to set up a collaborative production cycle that includes a pick-and-place task (Fig. 1). Specifically, two users, a Local User (LU) operating on-site (co-located with the robot) and another one operating remotely, take turns in programming different parts of the intended cycle. The Remote User (RU) is equipped with a VR headset that enables a complete immersion in a VR environment containing a DT of the robot and the workspace. The RU can explore the environment and interact with its virtual contents through a headset and a pair of hand controllers [60]. Given the fact that the LU and the robot work in close proximity, the task could present possible harmful conditions for the user; moreover, there could be areas that can be accessed by the robot but may be dangerous for the human operator. In these cases, the control can be passed from the RU to the LU, who can safely complete the programming. Depending on the situation, the robot may be programmed using hand guidance (by the LU) or via VR interactions (performed by the RU).

The evaluation described in the present paper focused only on aspects regarding network performance; in the future, however, humans-related aspects like usability, perceived realism or intuitiveness, could be additionally investigated.

In summary, the main contributions of the paper are:

- The presentation of a novel approach to CR programming that combines MR, VR and DT technologies to enable distant users' collaboration;
- The assessment of network requirements of the proposed approach with the aim to check its feasibility with current (4G/5G) and future (6G) mobile networks.

2 Related works

In the context of CRs, technological advancements have made machines capable of new tasks and new kinds of interactions with humans. The application of CRs to industrial operations expands every day, encompassing examples from both the industry and research communities. For example, in the automotive sector, the BMW Group has started to make use of CRs for equipping the inside of car doors with sound and moisture insulation since 2013 [5]. CRs are used at Audi to apply adhesive labels on cars' roofs [4]. For what it concerns examples belonging to the academic literature, it is possible to cite the work in [41], in which CRs are leveraged to hold the components to be polished by human operators (in a so-called fixture task). In [30], CRs are used for the handover of components at a pace that changes according to the operators' readiness to receive the objects. In [33], assembly tasks are accomplished with a collaboration between CRs and humans depending on the expected workload and consumed energy. The work in [21]

presents examples of a pick-and-place task in which a CR selects pieces to pick considering their distance, reachability, and the human operator's predicted motion plans. Soldering tasks controlled by humans and practically executed by CRs are reported in [64]. The authors of [18] provide an example of CRs used to inspect the work made by human operators; in particular, the task requires to check whether all the holes of a component have been screwed or not. Finally, in [12], CRs are leveraged to tighten the bolts inserted by a human operator from the other side of a plate.

Despite the widespread use in industrial applications, CRs still face a number of challenges. For example, safety aspects regarding both the robot and the human operators still present many open issues [22]. A confirmation of the pivotal role played by these aspects in the CR domain comes, for instance, from standard ISO/TS 15066:2016, which reports safety requirements for collaborative industrial robot systems and the work environment [32].

In fact, considering that humans operate close to robots in scenarios involving CRs, it is necessary to make them feel like they are working without any safety concerns and with well-defined shared areas [43]. Moreover, in order to foster collaboration, agents should recognize the intention of the counterpart to adapt their own actions [53].

To this aim, AR and MR appear to be fundamental tools to enhance human-robot interaction [15, 26]. For instance, in [29], MR contents are visualized through both a wearable video-see-through device (precisely, Microsoft's HoloLens glasses) or projected onto the real environment in order to provide operators with a feedback on the safety margins they can operate into. The margins visualized through the interactive User Interface (UI) mark the human's and the robot's virtual zone. The human's zone delimits the area where the operator can freely move and the robot cannot enter. The robot's virtual zone is an area that dynamically changes according to the task the robot is performing; the introduction of an object or the entrance of the operator automatically stops the execution of the robot's activities.

The authors of [50] propose a methodology based on a device similar to that used in the previous work to let human operators visualize the future movements of a robotic arm from their own viewpoint overlapped with the real world. The results obtained by the authors show that this approach can increase safety and efficiency, as the human operators no longer focus on aspects regarding safety and can instead concentrate on the tasks to be performed. Besides providing visual feedback, the proposed UI also lets the operators adjust the robot's goal by making use of hand gesture functionalities of the worn device. In [11], AR is exploited through the screen of a mobile device to let a remote expert assist an on-site user on robotics tasks. To this aim, AR contents in the form of spatially anchored and self-explaining cues (e.g., virtual arrows, 3D cursors, and card-based instructions) are attached by the expert to the real environment to guide the operations of the on-site user. Results show that using such an AR-powered remote assistance approach, experts can reduce the time invested in the assistance compared to traditional methods.

Although, in the works seen so far, some operators can be remote, those who collaborate with the robot physically share the same working space. In some cases, however, limitations to mobility due to sustainability constraints [10], hazard conditions [57], etc., could prevent them from being co-located with the robot [63]. For instance, it is possible to mention limitations related to the current pandemic conditions [65], the impact of possible natural disasters [47], or the requirements of home assistance scenarios [39]. In these situations, a DT- and VR-based remotely actionable reconstruction of the robot and its environment could help. In the context of human-robot collaboration, such virtual reconstructions could limit possible risks for the users deriving from the massive forces and high-speed movements of CRs [42]. Even CR programming may take advantage of a digitally separated environment to define robots' trajectories in a more intuitive and natural way. As a matter of example, the digital environment could be leveraged to show 3D controls or to enable advanced functionalities. These functionalities could include the possibility to zoom in and out, navigate the working space, reach unpractical regions (e.g., partially occluded by some workpieces), etc. [46].

Indeed, in recent years, the helpfulness of systems combining networking and XR has started to be demonstrated in various fields, as also highlighted in a recent report by Klynveld Peat Marwick Goerdeler (KPMG) [7]. Although there are still a number of issues mainly regarding network requirements, interesting applications have been presented targeting telesurgery [13], distance training, social interaction [56], collaborative decision-making [34] and object manipulation [45], etc.

To the best of the authors' knowledge, however, solutions that enable the collaboration between an MR-powered operator physically located close to a CR and a remote user that can program or control the robot at a distance have been scarcely investigated. Possible benefits of this configuration could be easily observed by analyzing telepresence/teleoperation scenarios, in which the introduction of MR/VR technologies has proven to be capable of improving remote control capabilities [46]. For instance, in

[16], a framework for robotic teleoperation based on VR is devised to make operators perform complex tasks that require accurate control of a robot's end-effector. The authors of [44] propose an open-source platform for 3D path planning in VR to control Unmanned Aerial Vehicles (UAVs). In [37], a system enabling the teleoperation of a tactile military mobile robot in VR is presented, whereas in [23], VR is used to simulate the behavior of a robotic manipulator in a safe environment for training purposes.

An example of approaches based on the combination of DTs and VR is given in [66], where a human-in-the-loop adaptive system for controlling a robotic arm in pick-and-place tasks is presented. The DT is leveraged to replicate physical components of the real world into a realistic environment experienced through immersive VR. The authors present two control strategies for "procedural" and "declarative" control. The strategy named procedural allows the operator to directly program the robot's behavior through VR interactions, whereas with the alternative strategy, the operator specifies only the final goal, as the planning of the robot's actions is demanded by an AI planner. Another example is reported in [46]: this work proposes an immersive environment representing an industrial facility in which the users can develop and test robot programs using off-the-shelf VR equipment. The interactive environment provides realistic simulations of real working conditions. Although interactive simulations could be characterized by non-negligible costs (e.g., related to software licenses, to the skills required for modeling and programming, and to the hardware needed for running the simulation), they represent an interesting solution for trainers as they can avoid potential damages to the industrial equipment and limit the risks for the trainees. In order to reconstruct the 3D environment, the authors of the considered work propose a process that encompasses several steps. 3D scans of the real facilities are first realized; after the application of suitable filters, the point clouds are imported into a graphics suite to realize the 3D models of the environment and apply animations; finally, the logic of the components is implemented through an interactive graphics engine. In [23], a VR-based environment is proposed to simulate the collaboration among human operators and CRs. Once a particular task has been simulated into the immersive environment under safe conditions, it can be deployed on real robots. The bidirectional connection between the real robot and its digital replica proposed by the authors makes it possible to use the robot as an input device for the simulation environment and, at the same time, lets the operators control the real robot by using 3D interaction devices (like a space mouse). To create the VR-based simulations, a CAD model of the environment and a hierarchical 3D model of each robot's part must be available.

All the solutions combining VR and DTs that have been presented above rely on pre-scanned scenes ([23, 46]) or on artificial environments that do not represent the real working space the robot operates into, thus limiting the effectiveness of such systems [16]. This drawback can be addressed by considering so-called Enhanced Virtual Reality (EVR) systems that can be exploited to reconstruct in real-time the robot's surrounding environment [16, 36, 62]. However, the high amount of data to be transferred to create EVR environments makes it no surprise that they are not yet adopted in practical use cases. Some techniques have been proposed to limit the overall amount of data to be streamed. For instance, the authors of [36] have developed an approach that leverages RGB-D cameras to recognize the objects present in the environment. Once the objects have been detected, the raw data of these objects are discarded from the stream and replaced with 3D models. In [16], the frames containing point clouds generated by two Intel's RealSense D415 cameras used for reconstructing the robot's surrounding environment undergo a three-step process (encompassing compression, frame validation, and decompression) to ensure a data transmission at the maximum frame rate possible.

Based on the capabilities of current mobile networks, the amount of data to be streamed with EVR represents a significant limitation for the concept scenario considered in this paper, given its requirements in terms of bandwidth and latency. According to the forecasts of the Hexa-X project [17, 58], however, the development of next-generation mobile networks will make fully digital representations of entire environments a reality. Moreover, among the use cases considered for the industrial context, the project specifically targets immersive telepresence and advertises the key role that will be played by communication technologies in the shift from robots to cobots [28], thus overlapping the requirements and needs of the scenario considered in this paper. According to studies on the performance of the future 6G networks reported in works such as [25, 28, 67], researchers and developers will be allowed to design applications that merge, as proposed in this paper, multiple domains like remote control and holographic telepresence, leveraging the availability of high-capacity and low-latency connections.

3 Materials and methods

This section describes the system developed to implement the devised alternate programming concept.

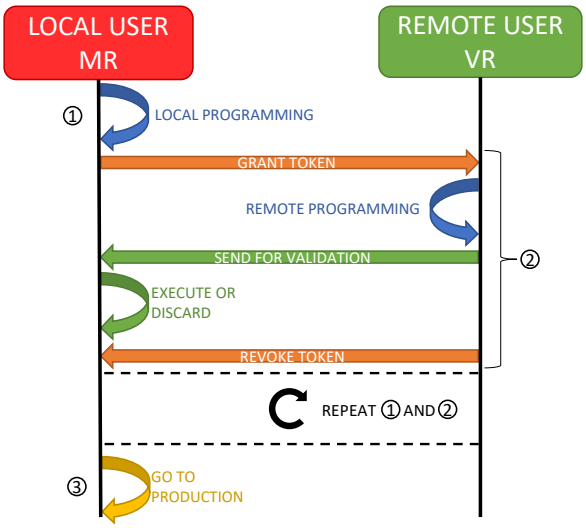


Figure 2: High-level interaction protocol of the alternate programming approach.

3.1 Alternate programming concept

In this section, the concept of alternate programming is described in detail, together with its high-level interaction protocol (Fig. 2).

The LU starts to teach the first part of the production cycle by interacting with the CR in a classic fashion (hand guidance, tool activation through buttons on the robot's flange, etc.) and by means of an additional MR UI. Although hand guidance could show some limitations in certain situations, it is decided to consider this interaction method since it represents the most used and intuitive way to operate CRs [52]; in those cases in which hand guidance cannot be used, the proposed programming paradigm allows the LU to pass the control to the RU, who can interact with the robot via alternative methods. The MR UI enables the LU to sequentially record the current CR's status (end-effector position, joints position, reaching time and speed, tool status, etc.) in the created program and, where needed, insert in the program a moment in which the robot should wait for an operator's input before continuing with the program execution. Then the LU, still using the MR UI, can decide (by choice or by necessity) to pass the programming token to the RU, who is spectating this process in the VR environment. The RU acknowledges the acquired control responsibility, and this change is signaled through LED light indicators on the CR (if present, otherwise via the MR UI). The RU keeps programming the robot using the same feature set provided to the LU. In VR, the working area can be observed thanks to a 3D reconstruction of the real environment, whereas the robot can be operated by means of its DT. In particular, by exploiting additional sensors (i.e., depth cameras), it is possible to implement a form of telepresence between the two users, allowing the RU to visualize the point cloud of the LU. At the same time, the depth sensors allow obtaining a reconstruction of the workspace and workpieces that cannot provide a DT by themselves. To reduce the risk of harming the LU or manipulating the robot in an unsafe manner, the RU is not entitled to directly control the robot. In other words, the real CR is not moving, and the RU operates on a duplicate of the CR's DT. Operating in VR allows the RU to overcome the physical boundaries of the working space and reach uncomfortable regions more easily by changing the visualization scale of the reconstructed environment. When done, the RU passes the control back to the LU. The LU double-checks the validity of the part of the program, later referred to also as program "clip", made by the RU (e.g., the use of safe and feasible trajectories, the avoidance of possible collisions, etc.) by visualizing an MR preview of the CR executing it. After this check, the LU can decide to accept or discard that clip. If accepted, it will be eventually executed by the real CR. The alternate programming process continues until the program is finalized. It should be noted that, during the execution of the program, the CR's motion is not under the control of the LU, exposing the latter to possible safety hazards. By design, CRs can be configured to stop their motion during unexpected collisions with the operator. However, supplemental safety features could be implemented if additional data are available (e.g., the position of the LU with respect to the CR) to avoid the risk of collisions.

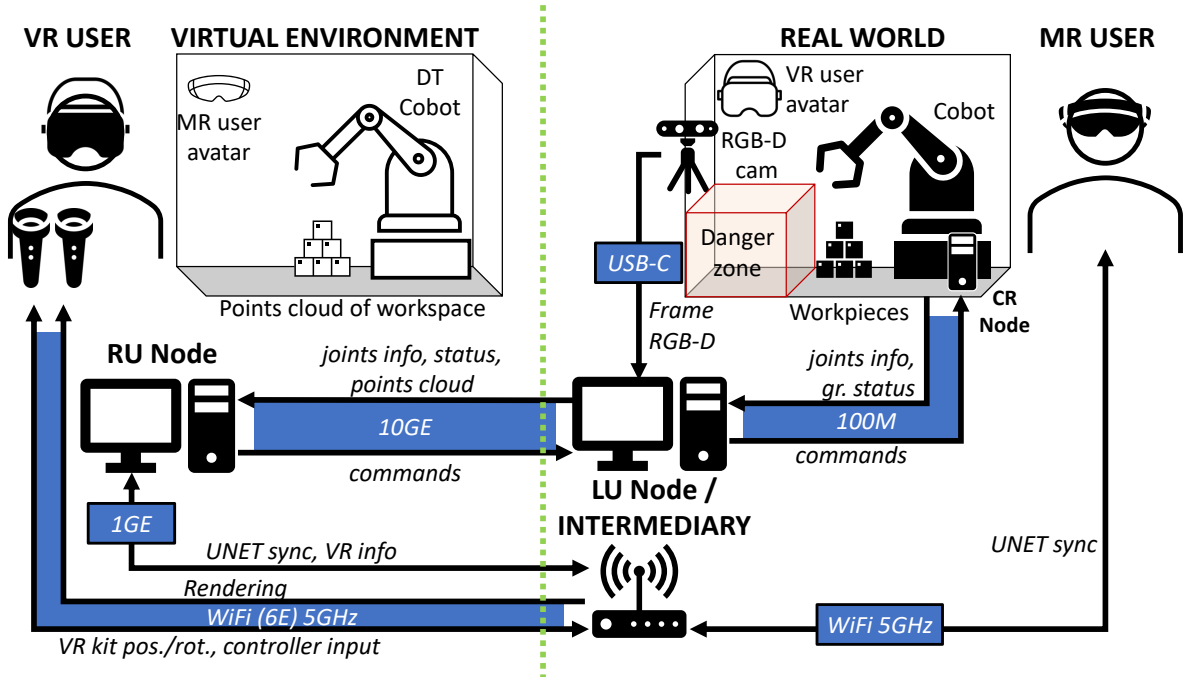


Figure 3: Architecture of the proposed concept implementation considered in the experimental evaluation.

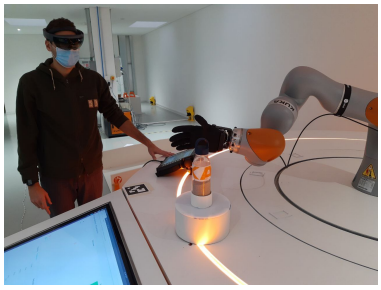
3.2 Architecture and network configuration

This paper aims to evaluate the feasibility of the proposed alternate programming concept from the perspective of network capabilities, with a particular attention to next-generation mobile networks. As 6G connectivity is not available yet, a prototype implementation was set up in a laboratory environment. The realized architecture is depicted in Fig. 3. The setup exploits currently available Wi-Fi 6E (802.11ax) technology to enable 5GHz wireless networking and combines it with 100Mbps to 10Gbps Ethernet connections. Wired networking was used to guarantee the availability of the required bandwidth, whereas mobile networks were not considered as most of the used devices do not support cellular connectivity.

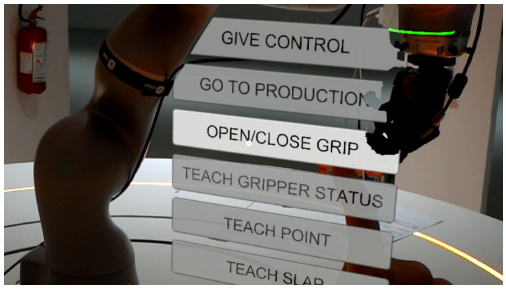
The KUKA LBR Iiwa 14 R820[1] CR platform was used by including it in a workspace volume of 4m × 4m × 3m. The CR is equipped with a qB SoftHand Industry anthropomorphic robotic hand[2] mounted on its flange (Media flange electric). The workspace is captured by an RGB-D camera connected via USB-C to an edge node, which is also directly connected to the CR using the Ethernet X66 connector on the KUKA robot controller, hereafter referred to as the CR node. Two PCs were used to emulate the network edge nodes responsible to provide each user with the required services, later named LU node and RU node. They are both equipped with an Intel i7-8700 CPU (6 cores @ 3.2GHz), an Nvidia GTX 1080 GPU, and a two-port 10Gbps-enabled Intel X540-AT2 network card. The LU node is also hosting a dedicated server, named Intermediary broker, which was implemented using the ZeroMQ (ZMQ) library. The server is configured as a proxy, tunneling connections from/to the robot to/from the RU node.

To implement the alternate programming paradigm, a custom messaging protocol was defined for exchanging the required information (CR's status, programming commands, etc.) and relying on Google's protobuf (v3.10) for the de/serialization. The RU node is responsible for the virtual environment emulation and rendering. In particular, to emulate scenarios envisaged for 6G-based networks in which a client with limited resources demands intensive work, like graphics computations, to a server (this approach is commonly referred to as the thin client paradigm), the VR rendering outcome is delivered to a VR HMD worn by the RU over Wi-Fi via a router directly connected to the RU node. This node is also in charge of synchronizing the CR's DT and status with the wireless MR HMD that is worn by the LU connected as well the Wi-Fi router. The LU and the RU, together with the respective equipment, are placed in two separate rooms distant 25m apart.

[1]KUKA LBR Iiwa: `https://www.kuka.com/lbr-iiwa`
[2]qB SoftHand Industry: `https://qbrobotics.com/products/qb-softhand-industry/`

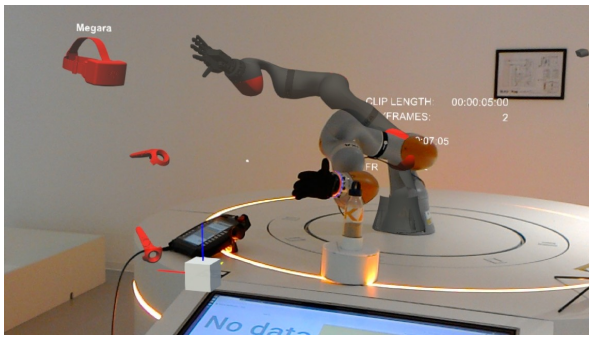


(a) LU wearing an MR HMD (HoloLens) in the real workspace consisting of the CR and a workpiece (a bottle on a stand).

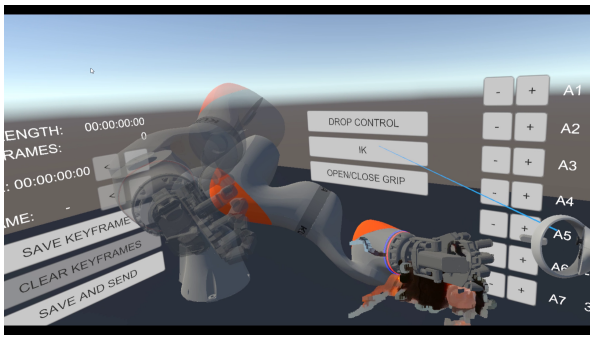


(b) LU's viewpoint while interacting with the MR UI overlapped to the real world.

(c) RU equipped with a wireless VR HMD (Quest 2) and its hand controllers.



(d) LU's viewpoint while spectating the virtual replica of the RU that interacts with the virtual CR (superimposed on the real one).



(e) RU's viewpoint while observing the DT of the (real) CR, the virtual CR, and the point cloud of the workpiece and interacting with the VR UI to create a new program clip.

Figure 4: Some moments of the use of the XR platform during an alternate programming session.

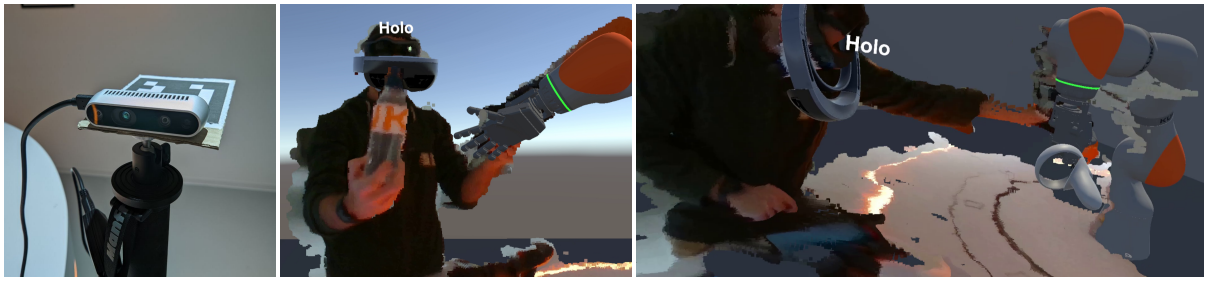
3.3 XR platform

For the development of the XR platform (Fig. 4), Unity 2018.4 was employed. Being conceived as a multi-player application, the Unity legacy high-level network API (UNET HLAPI) was used to implement the network code, and a client-server approach was pursued. According to the UNET specifications[3], a *UNET server* can be either hosted on an *UNET client* (thus becoming a *UNET host*) or on a dedicated machine. For the experiments, it was decided to host it on the VR client running on the RU node. This choice allowed to avoid the presence of a further desktop workstation (and network node), and to simplify the development of the XR platform server (by merging it with the VR client) at the cost of a slightly higher, yet negligible, computational load on the RU node. It should be noted that this simplification could not be adopted in the case of multiple VR users, which would require a non-authoritative version of the VR client.

To allow the communication between the XR platform clients and the other components of the system, NetMQ v4.0.1.6 (native C# port of ZMQ) was used to open various channels between the UNET server and the intermediary application, being the UNET architecture server-authoritative.

Using the UNET Network Transform component, it was possible to sync the position and orientation of the various clients up to 60Hz, and hence to visualize a virtual replica of the respective remote user. Interpolation for transform movements was also employed to achieve a smoother transition between network updates. The synchronization of statuses was obtained via UNET SyncVars, and UNET Hooks were adopted to manage events related to SyncVar changes on all the clients (host included). Finally, a voice communication channel between the various clients was established using the Dissonance VOIP asset for Unity[4], exploiting two additional UNET channels (one reliable and one unreliable).

[3] `UNET:https://docs.unity3d.com/2018.4/Documentation/Manual/UNetOverview.html`
[4] Dissonance Voice Chat: `https://assetstore.unity.com/packages/tools/audio/dissonance-voice-chat-70078`



(a) (b) (c)

Figure 5: 3D reconstruction: (a) Intel D435i camera equipped with an ArUco marker; (b, c) examples of point clouds (telepresent LU interacting with the real CR, workspace and workpiece) visualized in the VR environment as seen from the RU's viewpoint; apart from the point cloud, the LU's head position and orientation are always displayed by means of a 3D model of the MR HMD.

3.3.1 MR user interface

The MR client, designed as a HoloLens (1st Gen) application, takes advantage of the Mixed Reality Toolkit (MRTK) for Unity for the basic device functionalities (i.e., simultaneous localization and mapping, gesture recognition, and ArUco marker detection [24, 49]). The MR client user, once connected to the XR platform server (the VR remote client), is requested to look for an ArUco marker (6×6, 10cm) placed on the robot's base to allow the alignment between the real robot and the virtual elements, whereas a second marker is used to localize the RGB-D camera. Since the UNET Network Transform component only provides the possibility to synchronize the global transforms over the network, the component has to be modified to support local transform synchronization too. In this way, it is possible to easily align the Mixed Reality Playspace to the detected marker and then sync the local coordinates of the MR device with respect to it.

After the alignment, the MR user can visualize an MR UI displaying information coming from the robot. For instance, the values regarding the rotation of each joint can be visualized. At the same time, the MR user can interact with the CR with traditional methods (like hand guidance, control device, and interactive flange) or with the UI. For instance, by leveraging the UI, the MR user can open or close the gripper, teach a point, or pass/take control of the programming flow.

As described in Section 3.1, during the alternate programming, the LU can spectate the actions of the RU, observe a virtual preview of the robot's trajectory and actions programmed by the RU, and decide whether to accept or discard them before possibly sending them to the robot for execution. The MR client running on the HoloLens will be hereafter referred to as $\mathrm{AR}_{\mathrm{HMD}}$ node.

3.3.2 VR environment

The VR client was designed as a device-agnostic application, supporting any OpenVR-compatible kit for immersive VR (typically, a combination of headset and controllers). The SteamVR Unity asset was employed to provide the basic functions (CameraRig, controller input, and teleporting), on top of which the rest of the application was developed. The VR user who once started the hosting of the multi-player session automatically tries to connect to the Intermediary application. Once the client is connected, it starts receiving streams from the other online components of the systems (the CR and the RGB-D camera). At this stage, the VR user cannot do anything except request control tokens and wait for MR users to connect. As soon as the MR client connects to the UNET session, both the VR and MR users are able to see each other. In particular, the MR user sees the visual representation of the VR HMD and the controllers of the counterpart. The VR user sees the visual representation (3D model) of the MR HMD (i.e., the HoloLens), along with one or two spherical indicators when and where the user's hands are detected by the device (in the future, though, this representation could be made more realistic).

As soon as the MR client grants the control token to the VR user, the latter gains access to a dedicated UI that can be used to program a virtual replica of the robot and superimposed on the DT of the real robot. At the same time, the virtual replica is also displayed as an hologram to the MR user. The VR UI is depicted in Fig. 4(e). Taking inspiration from the world of computer animation, the user interacting with the VR UI can create a program by inserting and adjusting keyframes on a timeline. Keyframes are used to associate information about the robot (e.g., the status of the hand, the position

of the end-effector, etc.) with a specific time instant. Considering the fact that the control token can be exchanged between the two users, a program could be composed by parts (clips) created by each one of them. As soon as a clip is ready, the VR user can play it synchronously on all the clients to let the MR user evaluate its feasibility or spot critical conditions. If the counterpart validates the clip, it will be sent to the real robot for execution. During the execution of the programmed motion, the synced position of the LU is continuously checked to detect whether it gets too close to the physical CR. If this happens, a dedicated message is sent to the CR, triggering a safety stop in advance with respect to the possible collision. Alternatively, the MR user could decide to simply discard the proposed motion. In both cases, the control will then be passed to the MR user.

For the considered use case, a Meta Quest 2 was selected as a VR kit. To emulate a remote rendering scenario, the device was connected to the RU node running the VR client through the official Oculus Air Link[5] functionality. In particular, Air Link allows the use of the standalone device as a tethered headset, but without the need for a wired USB-C connection. Following the best practices suggested by the vendor, the Quest 2 was connected to a Wi-Fi 6E 5GHz router (ASUS RT-AX55) and in turn connected via Ethernet (1GE) to the RU node. The VR rendering client running on the Quest 2 will be hereafter referred to as the $\mathrm{VR_{HMD}}$ node.

Apart from the DT of the robot (obtained through a CAD model) and the virtual replica of the MR user, the VR user is also provided with a real-time point cloud reconstructed from an RGB-D camera stream coming from the RU node via the Intermediary application. The device selected for the considered use case was an Intel's RealSense D435i[6], characterized by a 1920×1080 RGB sensor and a 1280×720 Depth sensor. The depth accuracy of the considered camera is less than one percent of the distance from the object [31], with a minimum depth distance of $\sim$28 cm at the maximum resolution. Considering that the camera was positioned on a tripod $\sim$2 m away from the workpiece (within the ideal range of 30 cm to 3 m), the expected depth accuracy in close proximity to the workspace was $<2\%$, theoretically guaranteeing an appropriate level of realism of the 3D reconstruction for the considered task. Outside the range of the camera, the virtual replica of the MR user consisted solely in the 3D model of the head-worn device, which may result in a dramatic loss of perceived realism. However, for the considered task, this issue was not particularly relevant, as the actions to be performed generally fell within the supported range.

The camera is pointed toward the robot's workspace, and its position and orientation are made available as soon as the ArUco marker associated with it is detected by the MR user (Fig. 5(a)). Once it receives the camera parameters of the RGB and D (depth) sensors, the VR client on the RU node can process the RGB-D stream, generating the point cloud and displaying it locally (Fig. 5(b) and Fig. 5(c)). This setup allows the VR user to visualize, in the working space, those elements for which it is not possible to reconstruct a DT since not equipped with sensors. As a matter of example, by leveraging this point cloud-based approach, the VR user shown in Fig. 5(b) can visualize the workpiece (a bottle) handled by the other user, who is physically located in the working space.

3.4 Cobot application

The robot application was written in Java (v.1.7) for the Kuka Sunrise OS (v.1.16.2) using the proprietary SDK modules. In particular, the SmartServoMotion (v1.16) and the HandGuidance (v1.0) were used respectively to control the movement of the CR (when a piece of the program, including robot motion, is executed) and dis/enable the hand guidance control. For the networking layer, the ZMQ native Java implementation (JeroMQ v.0.4.3) was used. The CR node exchanges messages with the LU node using the already mentioned custom messaging protocol and digests it using a Finite State Machine (FSM) that implements the alternate programming behavior. An anthropomorphic hand is attached to the CR's flange and set up in software using the KUKA I/O Configurator provided by the KUKA WorkVisual (v5.0.5); thus, it can be accessed, and its status changes are controlled by the FSM.

3.5 RGB-D camera client

The RGB-D camera client was implemented using the librealsense2 (v2.49.0) SDK and the relative C# wrappers. In particular, the application connects to the Intermediary application on the RU node, accesses the device connected via USB-C, retrieves the relative parameters (e.g., intrinsic and extrinsic parameters of the sensors, framerate, and depth scale), polls the sensors for frames, and then streams

[5] https://support.oculus.com/airlink/
[6] https://www.intelrealsense.com/depth-camera-d435i/

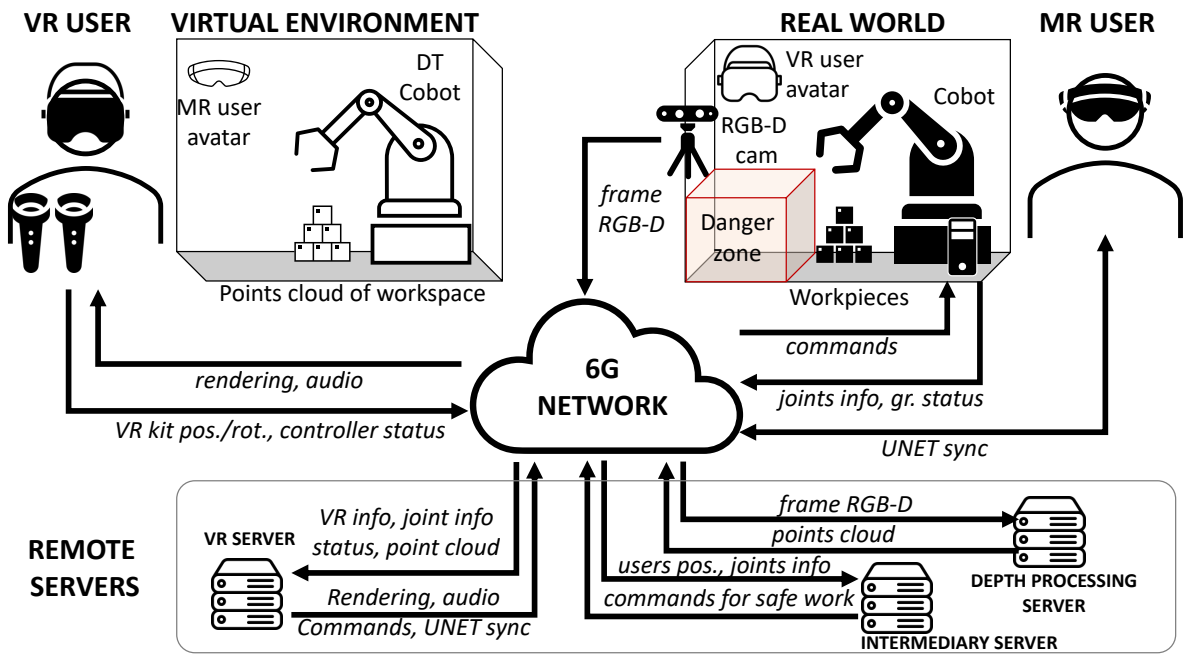


Figure 6: Conceptual architecture of the system leveraging the 6G network.

the data to the Intermediary application. Parameters are sent at the beginning of the stream, and a newly connected client asks for them every time. The size of an RGB-D frame and one second of stream generated by the RealSense D435i are reported in Table 1.

The table shows that one second of raw RGB-D data approximately requires a throughput of 2Gbps. Although the theoretical uplink limit of 5G networks being deployed is 10Gbps [2], real-world performance will unlikely reach that value [6]. For this purpose, the advent of 6G may play a fundamental role due to the 1Tbps theoretical bandwidth limit. Hence, to emulate the availability of a 6G network, a wired 10 Gbps network was employed to connect the RGB-D camera client, the Intermediary application, and the VR client. In particular, the Intermediary application and the RGB-D camera client were executed on the LU node, which was connected to the LU node running the VR client with a 30m CAT-6a Ethernet cable. Upon reception of an RGB-D message, the XR platform takes advantage of an external C++ dll to generate the point cloud and the relative UV map, implementing the generation algorithm openly available in the librealsense2 SDK. To speed up the process, the XR platform uses a multi-threaded approach, subdividing the frame into eight sub-frames processed in parallel.

3.6 Deploying the concept on a 6G network

The architecture described in Fig. 3 could be adjusted to consider the characteristics of a 6G network as shown in Fig. 6. Differently than in the architecture presented in Fig. 3, the various components are directly connected to the mobile network through dedicated adapters. Moreover, the network capabilities make it possible to deploy services, such as the remote rendering and safety checks, onto remote machines.

4 Results

In this section, the results obtained from the execution of a use case task based on the proposed concept in the described laboratory setup are reported. The aim was to estimate the network requirements of the considered use case in order to assess its feasibility with current mobile network technologies (or discover that next-generation networks are instead required). Thus, the following research hypotheses are formulated:

- *H1*: 4G/5G networks are not able to provide the bandwidth required by the considered use case;
- *H2*: The latency of 4G/5G networks is not acceptable for the considered use case.

Table 1: Size of the color and depth channels, for one frame and for one second, of the Intel D435i data stream (1920 × 1080 RGB, 1280 × 720 Depth, 30 fps).

D435i	**R**	**G**	**B**	**D**	**Total**
1 channel	8 bits	8 bits	8 bits	16 bits	40 bits
1 frame	15.82 Mb	15.82 Mb	15.82 Mb	14.06 Mb	61.52 Mb
1 second	474.60 Mb	474.60 Mb	474.60 Mb	421.87 Mb	1.80 Gb

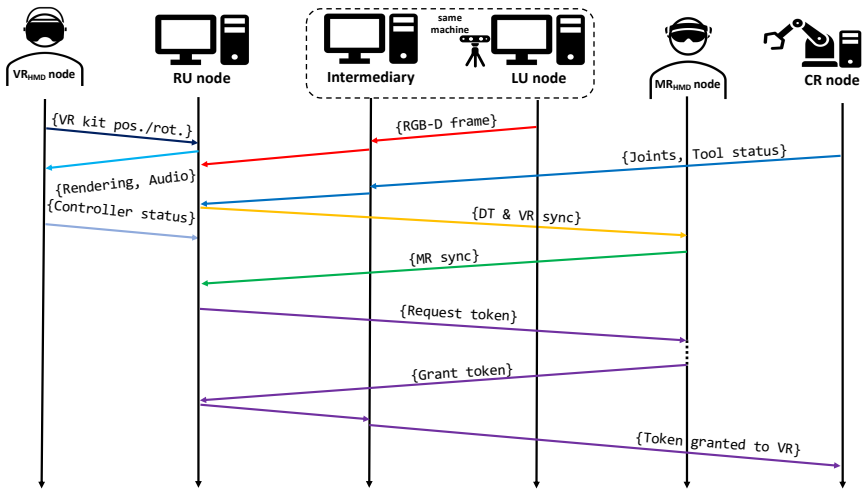


Figure 7: Example of messages exchanged among the various hops.

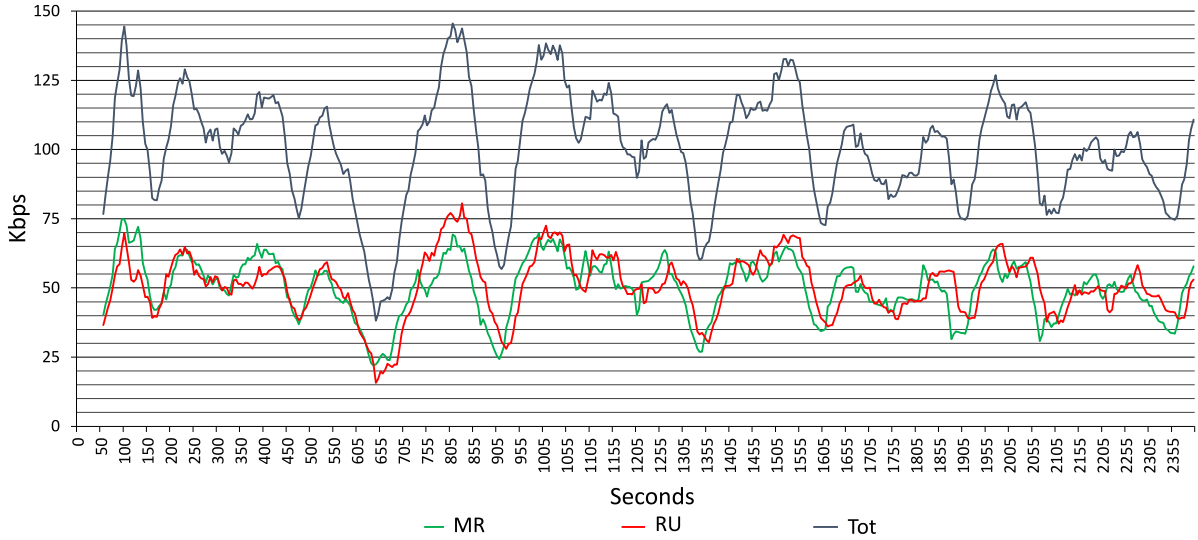
The task was performed by two users skilled in CR programming and the use of the devised system. The evaluation considered only aspects regarding networking, as this is the main goal of this paper. In the future, however, human factors could be included in the evaluation to widen its scope.

The task was repeated 10 times (the number was chosen arbitrarily) in order to account for possible variability in network conditions. Duration of a trial was about 15 minutes. In each trial, the users exchanged the token at least two times in order to test the system under realistic operational conditions. The task regards the programming of a fictional pick-and-place operation. To emulate a region in the workspace volume in which the LU cannot perform a hand-guided teaching, the workpiece is placed on a 3D-printed support representing, e.g., a danger zone in the range of the CR. The users are requested to collaboratively produce the following program:

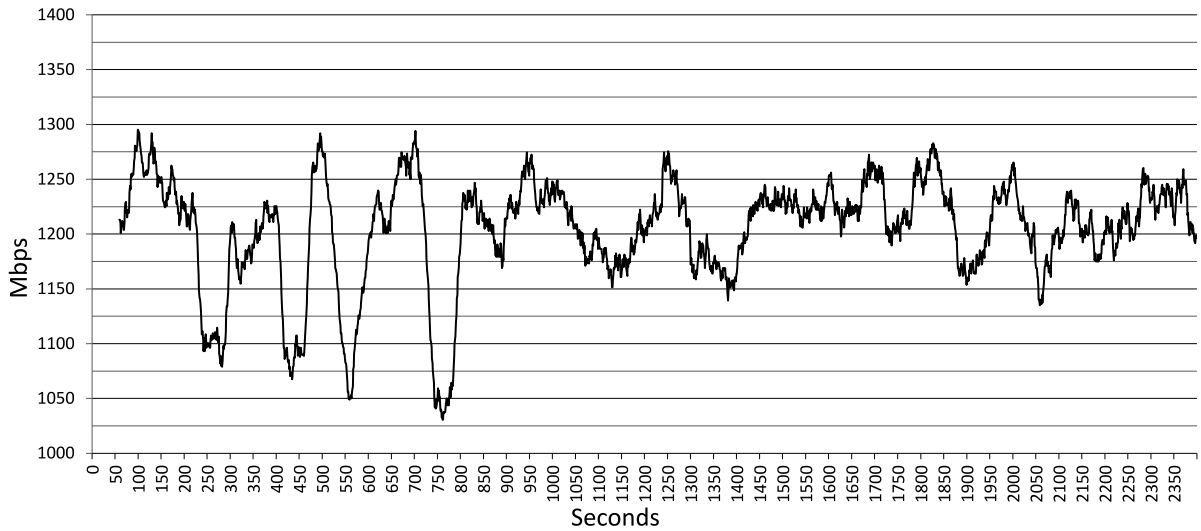
1. The robot starts with the tool and the anthropomorphic hand in a safe position;
2. The hand must open;
3. The hand has to be moved to pick the workpiece;
4. The hand must close (i.e., perform the grab action);
5. The workpiece must be moved outside the danger zone to a safe and cozy region reachable by the LU;
6. The workpiece has to be released (mid-air) when the CR receives an input by the LU (a light slap on the flange);
7. The LU provides the input again and the production cycle loops from step 1 (unless it is finished).

The collaboration between the two users is supported by the 3D reconstructed environment displayed in VR, which is obtained by combining the information about the CR's status and the RGB-D camera flow. More specifically, the RGB-D camera was positioned so that it always frames the workpiece during the execution of the whole program. The RU can observe in real-time the robot's surrounding and the current status of the workpiece without the need to rely on pre-scanned scenes or tracking all the workspace elements. Messages exchanged during the execution of the task are exemplified in Fig. 7.

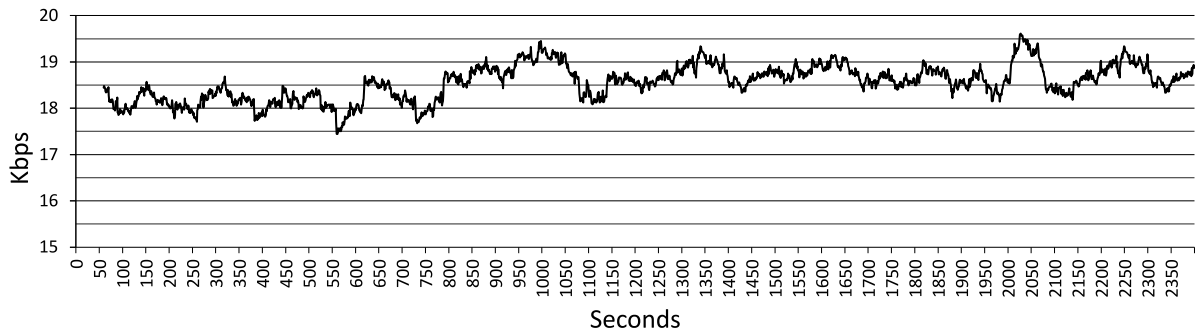
For the VR rendering, an average bandwidth between the VR_{HMD} and the RU nodes of 72Mbps was measured, with a *motion-to-photon* latency (delay between a movement of the HMD and the visualization



(a) Outbound node bandwidth ($\mathrm{MR_{HMD}}$–RU) for the UNET network layer ($\mu_{MR} = 50.03Kbps$, $\sigma_{MR} = 24.14Kbps$, $\mu_{RU} = 50.47Kbps$, $\sigma_{RU} = 23.05Kbps$, $\mu_{Tot} = 100.51Kbps$, $\sigma_{Tot} = 37.28Kbps$).



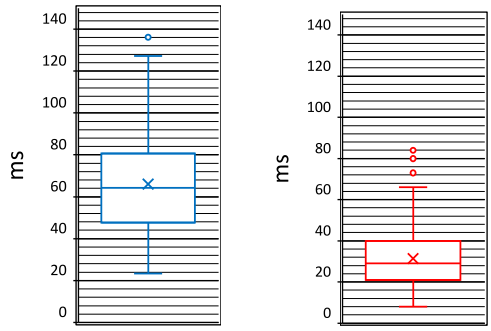
(b) Message bandwidth for the ZMQ network layer (LU–RU), considering only the depth data streaming ($\mu = 1.18Gbps$, $\sigma = 0.26Gbps$).



(c) Message bandwidth for the entire ZMQ network layer traffic (CR–RU), excluding the depth data streaming ($\mu = 18.55Kbps$, $\sigma = 3.20Kbps$)

Figure 8: Networking results regarding bandwidth (plots for one trial created using a moving average window of 60s).

update) of $L_{VR_{HMD}-RU} = 57$ms (min 55ms and max 69ms). This outcome is in line with the Air Link configured parameters and the expected reference performance provided by Meta.



(a) ZMQ Latency (CR–RU).

(b) UNET Latency (MR $_{\text{HMD}}$–RU).

Figure 9: Networking results regarding latency (all trials).

The overall bandwidth occupations during one trial are depicted in Fig. 8, whereas other latencies are reported in Fig. 9. Considering the UNET layer, the end-to-end latency between the MR visualization and the VR UNET synchronization on the RU node was on average equal to $L_{MR_{HMD}-RU} = 31.4$ms (min 23.5ms and max 84ms), with limited bandwidth requirements ($BW_{MR_{HMD}-RU} = 100.8$Kbps, min 1.9Kbps and max 228.8Kbps). It is worth considering that, in order to obtain the CR's DT visualization latency as observed by the LU in MR, the above delay should be added up to that required for a joints status message to reach the RU node. In the experiments, this latency was measured, on average, as equal to $L_{CR-RU} = 66.7$ms (min 23.3ms and max 148.3ms). A comparable amount of time was also required for a given command triggered through the MR UI to reach the CR node.

5 Discussion

In this section, previously reported results of the experimental activity are discussed on a per-requirement basis considering the formulated hypotheses.

5.1 Bandwidth

The plots in Fig. 8 represent networking results in terms of bandwidth. It can be observed that bandwidth usage varies largely among the different connections. More specifically, the following three cases can be distinguished:

- Regarding the LU-RU connection, implemented through the UNET library within the XR platform, results showed very low but fluctuating total bandwidth values (∼100Kbps). This result is mainly related to the presence of the VOIP channel and to the client-server synchronization of the XR devices.
- For the CR-RU connection, the bandwidth usage resulted in even lower values. This is due to the fact that this connection is simply used for frequent but small messages communicating the robot's status (joints and tool) to the XR platform and, sporadically, for other messages such as commands or program clips sent to the CR.
- Considering the connection between the depth camera client (running on the LU node) and the RU, the bandwidth usage peaked at around 2Gbps. This value is related to the RGB-D stream required to generate the point cloud inside the VR client running on the RU node.

The obtained results suggest that the current generation of mobile networks would not be capable of supporting the required bit rates, which reach peaks of 2Gbps in the worst case. This observation confirms H1. In fact, looking at performance data available for 4G networks, it can be noted that they can provide up to 150Mbps [1]. Instead, considering 5G networks, they rarely reach 1Gbps [6] (far below the theoretical limit of 10Gbps [2]). In this scenario, 6G appears to be a game-changer for the feasibility of the analyzed use case, as its expected bandwidth limit of the order of 1Tbps [3] would support the 3D reconstruction of the workspace (for telepresence and/or DT realization). Moreover, such

bandwidth availability would theoretically enable the combined use of more than one RGB-D camera without requiring additional processing time for compression and decompression.

5.2 Latency

Also, in the case of latency, the obtained results are characterized by high inhomogeneity.

- The latency between the $\mathrm{MR_{HMD}}$ and the RU nodes (related to the UNET network layer) was about 30ms, mainly attributable to the use of a Wi-Fi connection, the only network interface available on the HoloLens. It should be noted that to obtain the total latency between the $\mathrm{MR_{HMD}}$ and $\mathrm{VR_{HMD}}$ nodes, the $L_{VR_{HMD}-RU}$ delay of $\sim 60ms$ related to the remote rendering has to be considered too.
- As for the latency between the CR and RU nodes, results showed non-intuitive and numerically worse values. Despite the use of a wired Ethernet connection, the measured delay ranged between 23ms and 150ms. Further investigations showed that these performance issues were not related to the network configuration but to some limitations of the CR's Java Sunrise system.
- Due to the above performance issues, the overall latency between the $\mathrm{MR_{HMD}}$ and CR nodes (obtained by summing up the previous two delays) stood at rather high average values ($\sim$100ms). However, despite these values, the overall network performance allowed the proper functioning of the latency-dependent safety stop feature mentioned earlier (considering the relatively low CR movement speeds involved in the considered use case).

The use of mobile networks in place of Wi-Fi may already lead to the lowering of most of the latency values, in particular between the LU and RU nodes. Indeed, leveraging current 4G networks could yield even worse overall performance due to latency values around 50ms [1]. This consideration partially confirms H2. Nevertheless, 5G networks guarantee latency values around 10ms [1], and 6G is expected to bring these values down to 1ms [3]. Hence, with 5G and, in the future, with 6G, it will be possible to reduce also the delay between the $\mathrm{MR_{HMD}}$ and CR nodes, leaving as an open issue only the latency between the RU and CR nodes and permitting to set higher velocities for the CR movements with the safety stop feature properly functioning.

6 Conclusions

This paper describes a new approach for CR programming that allows two distant users to collaborate by leveraging XR technologies, a paradigm that would be key in the development of a next-generation sustainable industry. In particular, a user equipped with an MR headset and working close to the robot, and a remote user immersed in a VR environment representing the workspace in which the other user is operating, can alternate in the programming task. CR sensors and RGB-D camera data are leveraged to create DTs and 3D reconstructions that allow the remote user to visualize in real-time the robot's surrounding environment together with the actual conditions of the workpieces and safely operate onto them.

The devised approach imposes strict requirements in terms of the amount of data to be transferred and end-to-end latency, which may not be satisfied by current mobile networks. These requirements are mainly associated with the use of XR technologies in terms of 3D reconstructions and DTs synchronization. To cope with this limitation, the reported study looks at the benefits expected from the shift to 6G technology. In order to quantify the network requirements of this novel approach, a prototype implementation obtained by emulating 6G using currently available network technologies is described.

Results in terms of bandwidth and latency observed in the experimental setup confirmed the partial impossibility of realizing the proposed approach with current wireless technologies. Specifically, replacing a 6G connection with Wi-Fi incurs limitations due to the lack of traffic scheduling, advanced MIMO and beamforming capabilities, as well as wider channel bandwidths. Even considering the performance of 5G networks, the analyzed scenario appears to be not fully implementable, especially considering the bandwidth requirements. These limitations, once identified, however point to the theoretical feasibility of forthcoming 6G networks.

It is also worth observing that, although the used setup did not strive to be energy-efficient, the availability of 6G with its advanced solutions like AI-powered air interfaces [54] is expected to lead to a 50% reduction in transmit power over existing mobile networks for the same bandwidth and data rate. Together with other technological advances, such as enhanced semiconductor manufacturing techniques, the carbon footprint of the envisaged system will be reasonably low.

Experiments suggested that a lightweight version of the proposed scenario (in terms of network requirements) that may be supported by 5G could be obtained by applying compression algorithms to RGB-D camera frames and reducing the speed of the CR's movements (to make the latency required to support safety checks negligible). Although these solutions could reduce the requirements in terms of bandwidth and latency, respectively, they would introduce new issues, like new delays associated with RGB-D frames compression/decompression, and the slowdown of the production cycle.

Future works will focus on extending the evaluation concerning networking aspects by analyzing different setups. For instance, alternative tethering tools could be considered for extending wireless connectivity. Moreover, the availability of 6G networks could enable the shift towards the thin client paradigm, where all the computations (e.g., regarding emulation or rendering aspects) are demanded to different network nodes that produce the results to be visualized by the MR/VR users); in order to support this paradigm, further analyses would be needed to estimate the additional network requirements.

Lastly, it is worth remarking that the present study did not consider human factors. Future efforts will be devoted to investigating user experience with the system from many perspectives, including usability, intuitiveness, ease of use, physical effort (e.g., due to the prolonged use of VR/MR), perceived quality and sense of immersion in the virtual environment, etc. In particular, in order to improve the realism of the experience, floating 3D models representing the headset and hands of the remote user could be replaced with a full-body avatar. Moreover, realism could also be enhanced by acting on the 3D reconstruction process. As a matter of example, by combining the depth stream of a different, yet compatible, RGB-D camera like the Intel SR300, it may be possible to increase the depth accuracy at a short range (from 2% of the distance from the object of the D415i to 1 mm of the SR300 [31]), hence improving the 3D reconstruction of the workpiece without impairing that regarding the telepresence of the MR user. Besides introducing interesting XR-related challenges in terms of user experience, these changes may also bring new requirements. For instance, transmitting more complex geometries could have a huge impact on the required bandwidth, whereas latency could be influenced by the need to synchronize 3D objects' movements.

Acknowledgements

This work has been partly funded by the European Commission through the H2020 project Hexa-X (Grant Agreement no. 101015956). The authors want to acknowledge the help given by KUKA Roboter Italia Spa which made available the equipment used in the experimental phase.

References

[1] 5G vs. 4G: How will the newest network improve on the last? `https://www.digitaltrends.com/mobile/5g-vs-4g/`, accessed 28 September 2022. 2022.

[2] 5G Performance - European 5G Observatory. `https://5gobservatory.eu/info-deployments/5g-performance/`, accessed 28 September 2022. 2021.

[3] 6G: Going Beyond 100 Gbps to 1 Tbps. `https://www.keysight.com/it/en/assets/7121-1152/white-papers/6G-Going-Beyond-100-Gbps-to-1-Tbps`, accessed 28 September 2022. 2021.

[4] Human-robot cooperation at Audi. `https://www.springerprofessional.de/en/manufacturing/production---production-technology/human-robot-cooperation-at-audi/14221870`, accessed 28 September 2022. 2017.

[5] Innovative human-robot cooperation in BMW group production. `https://www.press.bmwgroup.com/global/article/detail/T0209722EN/innovative-human-robot-cooperation-in-bmw-group-production`, accessed 28 September 2022. 2013.

[6] UK mobile performance in review 2H 2020. `https://rootmetrics.com/en-GB/content/uk-mobile-performance-in-review-2H-2020`, accessed 28 September 2022. 2021.

[7] Future of Extended Reality. `https://assets.kpmg/content/dam/kpmg/au/pdf/2022/future-of-XR-white-paper.pdf`, accessed 28 September 2022. 2022.

[8] Gopika Ajaykumar, Maia Stiber, and Chien-Ming Huang. Designing user-centric programming aids for kinesthetic teaching of collaborative robots. *Robotics and Autonomous Systems*, 145:103845, 2021.

[9] Ronald T Azuma. A survey of augmented reality. *Presence: Teleoperators & virtual environments*, 6(4):355–385, 1997.

[10] Jean Botev and Francisco J Rodríguez Lera. Immersive robotic telepresence for remote educational scenarios. *Sustainability*, 13(9):4717, 2021.

[11] Davide Calandra, Alberto Cannavò, and Fabrizio Lamberti. Improving AR-powered remote assistance: A new approach aimed to foster operator's autonomy and optimize the use of skilled resources. *The International Journal of Advanced Manufacturing Technology*, 114(9):3147–3164, 2021.

[12] Andrea Cherubini, Robin Passama, Philippe Fraisse, and André Crosnier. A unified multimodal control framework for human–robot interaction. *Robotics and Autonomous Systems*, 70:106–115, 2015.

[13] Alexander Clemm, Maria Torres Vega, Hemanth Kumar Ravuri, Tim Wauters, and Filip De Turck. Toward truly immersive holographic-type communication: Challenges and solutions. *IEEE Communications Magazine*, 58(1):93–99, 2020.

[14] Shuping Dang, Osama Amin, Basem Shihada, and Mohamed-Slim Alouini. What should 6G be? *Nature Electronics*, 3(1):20–29, 2020.

[15] Francesco De Pace, Federico Manuri, Andrea Sanna, and Claudio Fornaro. A systematic review of augmented reality interfaces for collaborative industrial robots. *Computers & Industrial Engineering*, 149:106806, 2020.

[16] Francesco De Pace, Gal Gorjup, Huidong Bai, Andrea Sanna, Minas Liarokapis, and Mark Billinghurst. Leveraging enhanced virtual reality methods and environments for efficient, intuitive, and immersive teleoperation of robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 12967–12973. IEEE, 2021.

[17] G D'Aria et al. Expanded 6G vision, use cases and societal values – Including aspects of sustainability, security and spectrum. `https://hexa-x.eu/d1-2-expanded-6g-vision-use-cases-and-societal-values-including-aspects-of-sustainability-security-and-spectrum/`, accessed 28 September 2022. 2021.

[18] Ilias El Makrini, Kelly Merckaert, Dirk Lefeber, and Bram Vanderborght. Design of a collaborative architecture for human-robot assembly tasks. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1624–1629. IEEE, 2017.

[19] Shirine El Zaatari, Mohamed Marei, Weidong Li, and Zahid Usman. Cobot programming for collaborative industrial tasks: An overview. *Robotics and Autonomous Systems*, 116:162–180, 2019.

[20] Gerhard Paul Fettweis and Holger Boche. 6G: The personal tactile internet – And open questions for information theory. *IEEE BITS the Information Theory Magazine*, 1(1):71–82, 2021.

[21] Volker Gabler, Tim Stahl, Gerold Huber, Ozgur Oguz, and Dirk Wollherr. A game-theoretic approach for adaptive action selection in close proximity human-robot-collaboration. In *Proceedings of the IEEE Interantional Conference on Robotics and Automation*, pages 2897–2903. IEEE, 2017.

[22] Rinat Galin and Roman Meshcheryakov. Automation and robotics in the context of Industry 4.0: The shift to collaborative robots. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*, pages 1–5. IOP Publishing, 2019.

[23] Luigi Gammieri, Marco Schumann, Luigi Pelliccia, Giuseppe Di Gironimo, and Philipp Klimant. Coupling of a redundant manipulator with a virtual reality environment to enhance human-robot cooperation. *Procedia CIRP*, 62:618–623, 2017.

[24] Sergio Garrido-Jurado, Rafael Muñoz Salinas, Francisco Madrid-Cuevas, and Rafael Medina-Carnicer. Generation of fiducial marker dictionaries using mixed integer linear programming. *Pattern Recognition*, 51:481–491, 2016.

[25] Marco Giordani, Michele Polese, Marco Mezzavilla, Sundeep Rangan, and Michele Zorzi. Toward 6G networks: Use cases and technologies. *IEEE Communications Magazine*, 58(3):55–61, 2020.

[26] Scott A Green, Mark Billinghurst, XiaoQi Chen, and J Geoffrey Chase. Human-robot collaboration: A literature review and augmented reality approach in design. *International Journal of Advanced Robotic Systems*, 5(1):1, 2008.

[27] Patrik Gustavsson, Anna Syberfeldt, Rodney Brewster, and Lihui Wang. Human-robot collaboration demonstrator combining speech recognition and haptic control. In *Proceeding of the 50th CIRP Conference on Manufacturing Systems*, pages 396–401. Elsevier, 2017.

[28] Bin Han and Hans D. Schotten. Multi-sensory HMI to enable digital twins with human-in-loop: A 6G vision of future industry. `https://doi.org/10.48550/arXiv.2111.10438`, accessed 28 September 2022. 2021.

[29] Antti Hietanen, Roel Pieters, Minna Lanz, Jyrki Latokartano, and Joni-Kristian Kämäräinen. AR-based interaction for human-robot collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing*, 63:101891, 2020.

[30] Chien-Ming Huang, Maya Cakmak, and Bilge Mutlu. Adaptive coordination strategies for human-robot handovers. In *Proceeding of the Robotics: Science and Systems Conference*, volume 11, pages 1–10. Robotics: Science and Systems Foundation, 2015.

[31] Intel®. Depth Resolution of Intel® RealSense™ Depth Camera D435 and Intel® RealSense™ Camera SR300. `https://www.intel.com/content/www/us/en/support/articles/000026260/emerging-technologies/intel-realsense-technology.html`, accessed 28 September 2022. 2021.

[32] ISO/TS 15066:2016. ISO/TS 15066:2016 Robots and robotic devices – Collaborative robots. Standard, International Organization for Standardization, 2016.

[33] Lars Johannsmeier and Sami Haddadin. A hierarchical human-robot interaction-planning framework for task allocation in collaborative industrial assembly processes. *IEEE Robotics and Automation Letters*, 2(1):41–48, 2016.

[34] Theo Kanter, Uno Fors, and Rahim Rahmani. Immersive networking - A framework for virtual environments with augmented reality in human decision-making. *International Journal of Multimedia and Ubiquitous Engineering*, 11(6):43–60, 2016.

[35] Latif U Khan, Ibrar Yaqoob, Muhammad Imran, Zhu Han, and Choong Seon Hong. 6G wireless systems: A vision, architectural elements, and future directions. *IEEE Access*, 8:147029–147044, 2020.

[36] Sebastian Kohn, Andreas Blank, David Puljiz, Lothar Zenkel, Oswald Bieber, Bjorn Hein, and Jorg Franke. Towards a real-time environment reconstruction for VR-based teleoperation through model segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–9. IEEE, 2018.

[37] Tomáš Kot and Petr Novák. Application of virtual reality in teleoperation of the military mobile robotic system TAROS. *International Journal of Advanced Robotic Systems*, 15(1): 1729881417751545, 2018.

[38] Kevin Luo, Shuping Dang, Basem Shihada, and Mohamed-Slim Alouini. Prospect theory for human-centric communications. *Frontiers in Communications and Networks*, 2:8, 2021.

[39] Honghao Lv, Geng Yang, Huiying Zhou, Xiaoyan Huang, Huayong Yang, and Zhibo Pang. Teleoperation of collaborative robot for remote dementia care in home environments. *IEEE Journal of Translational Engineering in Health and Medicine*, 8:1–10, 2020.

[40] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information and Systems*, 77(12):1321–1329, 1994.

[41] Stefanos Nikolaidis, Przemyslaw Lasota, Ramya Ramakrishnan, and Julie Shah. Improved human-robot team performance through cross-training, an approach inspired by human team training practices. *The International Journal of Robotics Research*, 34(14):1711–1730, 2015.

[42] John O Oyekan, Windo Hutabarat, Ashutosh Tiwari, Raphael Grech, Min H Aung, Maria P Mariani, Laura López-Dávalos, Timothé Ricaud, Sumit Singh, and Charlène Dupuis. The effectiveness of virtual environments in developing collaborative strategies between industrial robots and humans. *Robotics and Computer-Integrated Manufacturing*, 55:41–54, 2019.

[43] Marie-Pierre Pacaux-Lemoine and Damien Trentesaux. Ethical risks of human-machine symbiosis in Industry 4.0: Insights from the human-machine cooperation approach. *IFAC-PapersOnLine*, 52 (19):19–24, 2019.

[44] Jesse Rawlins Paterson, Jiwoong Han, Tom Cheng, Paxtan Huish Laker, David Livingston McPherson, Joseph Menke, and Allen Y Yang. Improving usability, efficiency, and safety of UAV path planning through a virtual reality interface. In *Proceeding of the Symposium on Spatial User Interaction*, pages 1–2. ACM, 2019.

[45] Vasco Pereira, Teresa Matos, Rui Rodrigues, Rui Nóbrega, and João Jacob. Extended reality framework for remote collaborative interactions in virtual environments. In *Proceedings of the International Conference on Graphics and Interaction (ICGI)*, pages 17–24. IEEE, 2019.

[46] Luis Pérez, Eduardo Diez, Rubén Usamentiaga, and Daniel F García. Industrial robot control and operator training using virtual reality interfaces. *Computers in Industry*, 109:114–120, 2019.

[47] FJ Perez-Grau, R Ragel, F Caballero, A Viguria, and A Ollero. Semi-autonomous teleoperation of uavs in search and rescue scenarios. In *Proceedings of the 2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1066–1074. IEEE, 2017.

[48] F. Gabriele Pratticò and Fabrizio Lamberti. Towards the adoption of virtual reality training systems for the self-tuition of industrial robot operators: A case study at KUKA. *Computers in Industry*, 129:103446, 2021.

[49] Francisco J. Romero-Ramirez, Rafael Muñoz Salinas, and Rafael Medina-Carnicer. Speeded up detection of squared fiducial markers. *Image and Vision Computing*, 76:38–47, 2018.

[50] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays. *The International Journal of Robotics Research*, 38(12-13):1513–1526, 2019.

[51] Walid Saad, Mehdi Bennis, and Mingzhe Chen. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, 34(3):134–142, 2019.

[52] Mohammad Safeea and Pedro Neto. KUKA Sunrise Toolbox: Interfacing collaborative robots with MATLAB. *IEEE Robotics Automation Magazine*, 26(1):91–96, 2019.

[53] Friederike Schneemann and Frederik Diederichs. Action prediction with the jordan model of human intention: A contribution to cooperative control. *Cognition, Technology & Work*, 21(4):711–721, 2019.

[54] A. Skalidi et al. AI-driven communication & computation co-design: Gap analysis and blueprint. `https://hexa-x.eu/wp-content/uploads/2021/09/Hexa-X-D4.1_v1.0.pdf`, accessed 28 September 2022. 2021.

[55] Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is mixed reality? In *Proceedings of the CHI Conference on Human Factors in Computing systems*, pages 1–15. ACM, 2019.

[56] Maria Torres Vega, Christos Liaskos, Sergi Abadal, Evangelos Papapetrou, Akshay Jain, Belkacem Mouhouche, Gökhan Kalem, Salih Ergüt, Marian Mach, Tomas Sabol, Albert Cabellos-Aparicio, Christoph Grimm, Filip De Turck, and Jeroen Famaey. Immersive interconnected virtual and augmented reality: a 5G and IoT perspective. *Journal of Network and Systems Management*, 28(4): 796–826, 2020.

[57] James Trevelyan, William R Hamel, and Sung-Chul Kang. Robotics in hazardous applications. In *Springer handbook of robotics*, pages 1521–1548. Springer, 2016.

[58] Mikko A Uusitalo, Mårten Ericson, Björn Richerzhagen, Elif Ustundag Soykan, Patrik Rugeland, Gerhard Fettweis, Dario Sabella, Gustav Wikström, Mauro Boldi, Marie-Helene Hamon, et al. Hexa-X the European 6G flagship project. In *Proceeding of the Joint European Conference on Networks and Communications & 6G Summit*, pages 580–585. IEEE, 2021.

[59] Minna Vasarainen, Sami Paavola, Liubov Vetoshkina, et al. A systematic literature review on extended reality: Virtual, augmented and mixed reality in working life. *Internationa Journal of Virtual Reality*, 21(2):1–28, 2021.

[60] Sara Ventura, Eleonora Brivio, Giuseppe Riva, and Rosa M Baños. Immersive versus non-immersive experience: Exploring the feasibility of memory assessment through 360 technology. *Frontiers in Psychology*, 10:2509, 2019.

[61] Yuqi Wang, Yudie Hu, Shirine El Zaatari, Weidong Li, and Yong Zhou. Optimised learning from demonstrations for collaborative robots. *Robotics and Computer-Integrated Manufacturing*, 71: 102169, 2021.

[62] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In *Robotics Research*, pages 335–350. Springer, 2020.

[63] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality. In *Robotics Research*, pages 335–350. Springer, 2020.

[64] Mahisorn Wongphati, Hirotaka Osawa, and Michita Imai. Gestures for manually controlling a helping hand robot. *International Journal of Social Robotics*, 7(5):731–742, 2015.

[65] Geng Yang, Honghao Lv, Zhiyu Zhang, Liu Yang, Jia Deng, Siqi You, Juan Du, and Huayong Yang. Keep healthcare workers safe: application of teleoperated robot in isolation ward for covid-19 prevention and control. *Chinese Journal of Mechanical Engineering*, 33(1):1–4, 2020.

[66] Enes Yigitbas, Kadiray Karakaya, Ivan Jovanovikj, and Gregor Engels. Enhancing human-in-the-loop adaptive systems through digital twins and VR interfaces. In *Proceedings of the 2021 International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*, pages 30–40. IEEE, 2021.

[67] Volker Ziegler and Seppo Yrjölä. How to make 6G a general purpose technology: Prerequisites and value creation paradigm shift. In *Proceedings of the Joint European Conference on Networks and Communications & 6G Summit*, pages 586–591. IEEE, 2021.