electronics

MDPI

Article

Using Mixed Reality for Control and Monitoring of Robot Model Based on Robot Operating System 2

Dominik Janecký, Erik Kučera *, Oto Haffner, Erika Výchlopeňová and Danica Rosinová

Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, 841 04 Bratislava, Slovakia; dominik.janecky@stuba.sk (D.J.); oto.haffner@stuba.sk (O.H.); danica.rosinova@stuba.sk (D.R.)

* Correspondence: erik.kucera@stuba.sk

Abstract: This article presents the design and implementation of an innovative human–machine interface (HMI) in mixed reality for a robot model operating within Robot Operating System 2 (ROS 2). The interface is specifically developed for compatibility with Microsoft HoloLens 2 hardware and leverages the Unity game engine alongside the Mixed Reality Toolkit (MRTK) to create an immersive mixed reality application. The project uses the Turtlebot 3 Burger model robot, simulated within the Gazebo virtual environment, as a representative mechatronic system for demonstration purposes. Communication between the mixed reality application and ROS 2 is facilitated through a publish–subscribe mechanism, utilizing ROS TCP Connector for message serialization between nodes. This interface not only enhances the user experience by allowing for the real-time monitoring and control of the robotic system but also aligns with the principles of Industry 5.0, emphasizing human-centric and inclusive technological advancements. The practical outcomes of this research include a fully functional mixed reality application that integrates seamlessly with ROS 2, showcasing the potential of mixed reality technologies in advancing the field of industrial automation and human–machine interaction.

Keywords: mixed reality; human–machine interface; human–robot interaction; Robot Operating System; mechatronic system; extended reality; gazebo

check for updates

Citation: Janecký, D.; Kučera, E.; Haffner, O.; Výchlopeňová, E.; Rosinová, D. Using Mixed Reality for Control and Monitoring of Robot Model Based on Robot Operating System 2. *Electronics* **2024**, *13*, 3554. https://doi.org/10.3390/electronics13173554

Academic Editors: Antonio José Calderón Godoy, Andrea Bonci, Isaías González Pérez and Francisco Javier Folgado Gaspar

Received: 6 July 2024
Revised: 26 August 2024
Accepted: 4 September 2024
Published: 6 September 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Extended reality is a modern technology used in Industry 4.0 to optimize production structures and work operations through ergonomic evaluation [1]. In digital factories, new methods of process control, monitoring, diagnostics, and visualization are currently being explored [2]. These types of forms are brought about by extended reality.

Virtual, augmented, and mixed reality are all included under extended (computed-generated) reality. There is currently inconsistency among experts regarding how to distinguish between mixed and augmented reality. Multiple definitions exist [3].

The Foundry, a company that creates software for 3D modeling and texturing [4], provided the first definition. It is common in industrial practice to apply this definition.

Virtual reality (VR) is a technology that allows users to interact in an environment that mimics a physical presence in real or imagined worlds [4]. Virtual reality devices include the Oculus Rift, Meta Quest, HTC Vive Pro, and others [5].

A view of a physical, real-world environment that is enhanced (or supplemented) by computer-generated sensory input, such as sound, video, graphics, or GPS data, is known as augmented reality (AR) [4]. Content that is not anchored to or a part of the real world is superimposed upon it to create augmented reality. It is not possible for CG content and real-world content to react to one another [6].

The fusion of the real and virtual worlds to create new settings and visualizations in which real-time interactions occur between digital and physical items is known as mixed

reality (MR). A synthetic content overlay that is anchored to and interacts with the real environment is called mixed reality (MR). The ability of synthetic and real-world contents to respond to one another instantly is the primary feature of machine learning (ML) [4]. Apple ARKit, Android ARCore, and Microsoft HoloLens (Windows Mixed Reality platform) are examples of mixed reality technologies [6].

In scientific teams, as opposed to industrial activity, a different definition is often used. The spectrum between the real and virtual environments—the reality–virtuality continuum—was first described in 1994 by authors Milgram and Kishiho [7] (Figure 1). This continuum defines a mix of the real and virtual worlds. Anything that is between the actual world and fully virtual reality is what they refer to as mixed reality. They make a distinction between augmented reality and augmented virtuality, with the former being nearly synonymous with augmented reality as defined by The Foundry. According to The Foundry's definition, augmented virtuality is equivalent to mixed reality.

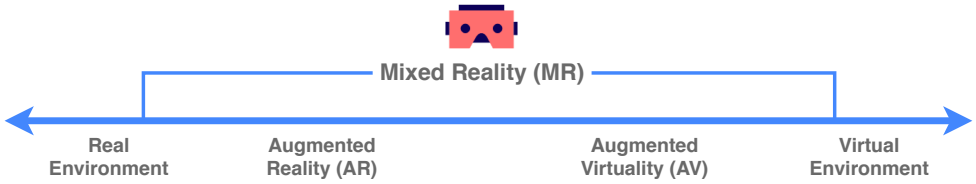


Figure 1. Reality–virtuality continuum.

1.1. Motivation

In the current era of digital advancement and automation, where innovation is not just a nice bonus but an essential part, the robotics and mechatronics field is creating a need for the development of new technologies and approaches that can increase the efficiency, reliability, and performance of robotic systems or the safety of users in hazardous environments. It is at such a time that Robot Operating System 2 (ROS 2) represents a new era in the control of mechatronic systems [8].

As the successor to the original ROS, ROS 2 brings a wide range of enhancements and new features that increase its flexibility, safety, and performance [9]. However, despite this potential, there is still a lack of sophisticated software solutions optimized for this new middleware version. This lack highlights the need for further developments in the integration of ROS 2 into modern technologies.

In this context, mixed reality (MR) emerges as an opportunity to revolutionize the way we interact with the environment around us. MR allows for the enhancement of the real world by using digital elements to display information or even control mechatronic systems remotely, without the use of a tablet or other device. The Microsoft HoloLens 2 headset appears to be one of the most promising devices [10]. With its ability to provide a high level of precision, it offers an ideal platform for the successful use of MR in complex industrial and research applications for robot control.

Coupling ROS 2 with MR technology can have a significant impact in areas where humans are in hazardous environments, where safety and rapid response are priorities. Such environments include fires or radioactive spills, where the remote control of robots and mechatronic systems via MR headsets can be safer and more effective than traditional methods [11]. As an example, consider rescue operations, where the use of an MR headset would enable the remote control of robotic drones, minimizing the risk to one's own health while increasing the ability to respond quickly and effectively to crisis situations. In industrial environments, such a solution would allow technicians to monitor and control robotic equipment from a safe distance, minimizing the risk of injuries and increasing the overall efficiency of equipment maintenance and operation.

The objective of this study is to establish a link between a virtual model of a robotic system using ROS 2 with an MR application. The result is an MR application that will allow for the monitoring and control of the selected virtual model of the robotic system.

The integration of MR with ROS 2 introduces significant advancements in ergonomics and safety and aligns seamlessly with the human-centric approach emphasized in Industry 5.0 [12,13]. By employing MR headsets such as the Microsoft HoloLens 2 (Microsoft, Dublin, Ireland) (Figure 2), workers in hazardous environments can navigate and control robotic systems without the need for handheld controllers or tablets, thereby keeping their hands free for other critical tasks. This ergonomic enhancement not only improves the efficiency and ease of operation but also reduces the physical strain associated with traditional control methods [14].

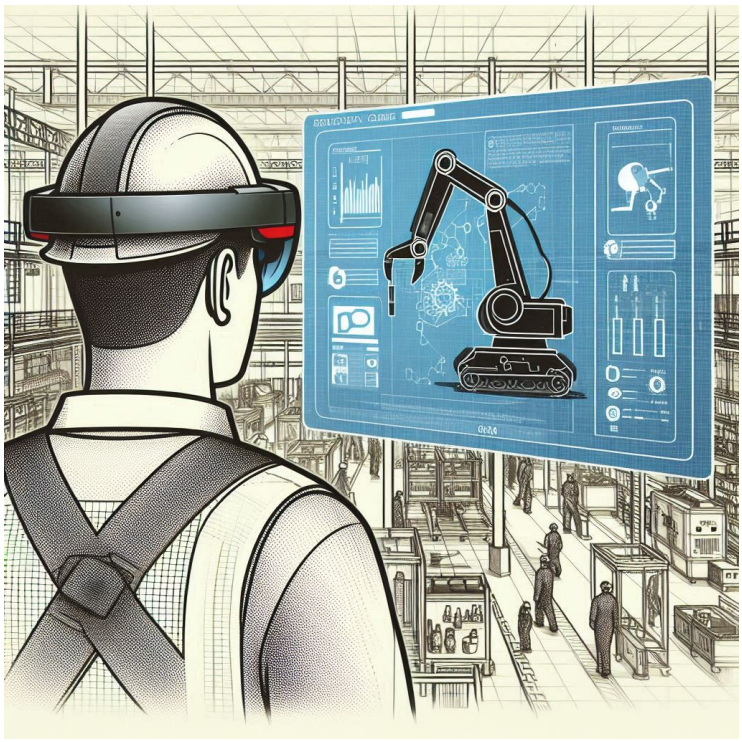


Figure 2. Illustration of mixed reality in industrial environment.

In environments where quick and precise actions are required, such as during emergency response operations or within industrial settings, the ability to operate hands-free is invaluable. Workers can swiftly interact with digital interfaces overlaid in their real-world view, using simple gestures or voice commands. This approach minimizes distractions and allows for more natural and intuitive control of robotic systems, leading to faster response times and higher operational accuracy. The reduction in the need to switch between tools and interfaces further streamlines workflows and enhances productivity. In high-risk environments, such as areas with hazardous materials or unstable structures, maintaining a safe distance from danger zones is crucial.

This innovative approach is perfectly aligned with the principles of Industry 5.0, which focuses on a human-centric perspective in industrial advancements. Industry 5.0 aims to create a symbiotic relationship between humans and machines, enhancing human capabilities while ensuring their well-being [15]. By integrating MR with ROS 2, technology places the human operator at the center of the control process, empowering them with advanced tools that enhance their safety, efficiency, and comfort. The human-centric design ensures that technology adapts to human needs, rather than forcing humans to adapt to technology, fostering a more inclusive and user-friendly industrial environment. Furthermore, this solution supports the core objectives of Industry 5.0 by promoting sustainability and resilience. The enhanced safety and ergonomic benefits contribute to a healthier workforce, reducing the likelihood of injuries and associated downtime.

1.2. Related Research

Currently, in the field of robotics and automation, the effective use of AR or MR technology in conjunction with ROS 2 is a challenge. The demand for mobile robots has been increasing in recent years due to their flexibility and large variety of application areas compared with static robots. There is also an increasing demand for systems that can offer fast and efficient remote control of ROS 2-based mechatronic/robotic systems. One technology with this potential is the use of AR or MR for interfacing and communicating with ROS 2 middleware. The goal of the various research studies that focus on interfacing AR or MR technologies with ROS 2 is to create a highly interactive and intuitive environment for controlling and monitoring robotic systems. Developing new AR or MR technologies represents a major step in the development of robust robotics applications.

The communication and interaction of the real world with the virtual world is gradually becoming the basis for new implementations of intelligent manufacturing ecosystems. Human–robot interaction is seen as an essential element of the factories of the future. Despite the advancement of various technologies, such as wearable devices and augmented reality, human–robot interaction (HRI) is still extremely challenging. Although progress has been made in the development of various mechanisms to support HRI, there are problems with cost, naturalness, and intuitive interaction and communication among heterogeneous systems. To mitigate these limitations, RoSTAR, a system based on ROS interfacing with augmented reality, has been proposed to allow the user to interact with a robotic arm. This system has the potential to be used in a variety of process tasks, such as robotic bonding, batching, and arc welding, as part of an interoperable, low-cost, portable, and inherently interactive experience [16].

The 2023 research study conducted at University of Oxford was aimed at helping rescuers explore disaster areas by making it easier to control robots by using gestures and allowing them to manage information from the environment being explored. Human–robot interaction was implemented by using a HoloLens device and a robot called ARTUR (A1 Rescue Tasks UPM Robot), which supports ROS and is equipped with a sensor system (Lidar sensor and thermal and RGB cameras). The main contribution of the work was the implementation and evaluation of a mixed reality system based on the Unity-ROS solution that can control the robotic device at a high level. The testing yielded interesting results, such as a 48% improvement in efficiency and a 21.4% improvement in time optimization compared with conventional interfaces. The proposed method was shown to improve the immersion experience of rescuers in controlling a complex robotic system [17].

At the end of 2023, research was underway at Harbin University of Technology that focused on motion control of interactive robotic arms based on mixed reality. The main objective of the work was to implement methods for motion control of selected robotic arms by using an application developed in Unity for mixed reality. The work resulted in the successful motion control of the UR5 and UR5e robots by using an application in Unity interfaced with ROS, with a future plan for deploying the solution on Microsoft HoloLens 2 [18].

The increasing demand for collaborative robot (cobot) projects highlights the need to execute tasks quickly and flexibly while ensuring safe human–robot interactions. Generally, this requires expertise in robotics programming and hardware operation. A proposed solution involves a mixed reality (MR) application integrated with a head-mounted device (HMD) to streamline the programming of complex cobot maneuvers. The solution mentioned in [19] leverages voice and gesture recognition, along with digital panels, enabling users of any robotics skill level to work more efficiently. The Robot Operating System (ROS) platform oversees the cobot and facilitates data transfer between the systems. Additionally, QR (Quick Response) codes are used to establish an accurate frame of reference. The study focuses only on robotic arms and not on the control of mobile robots. Moreover, the exact version of ROS is not specified.

The fourth industrial revolution is characterized by the increasing intelligence and automation of manufacturing systems. However, some tasks remain too complex or costly to fully automate, making it more efficient for machines to collaborate with humans,

not only by sharing the same workspace but also by acting as useful partners. Human–robot interaction systems offer a potential solution by identifying applications where they can be beneficial and addressing the challenges they encounter. The study [20] proposes developing an industrial prototype of a human–machine interaction system using augmented reality (AR), aiming to enable industrial operators with no programming experience to program robots. As can be seen from this description, the study focuses on the programming of robotic arms and, as in the previous case, does not focus on mobile robots, which are the subject of the research described in the submitted article.

The work [21] introduces an augmented reality (AR) application for HoloLens 2, enabling human operators without specialized robotics knowledge to interact effortlessly with collaborative robots. The application provides detailed insights into the robot's state, including joint positions, velocities, and torques. It also visualizes the workspace and allows users to generate and manipulate the end-effector trajectory by directly adjusting waypoints displayed in the AR environment. Additionally, the system verifies and informs the user about the feasibility of the trajectory, considering the workspace limits. As can be seen from the description, the study again deals only with robotic arms.

Currently, drones are already being utilized to spray pesticides over agricultural areas. In mobile robotics, control is mainly managed through a controller or an app on a smartphone or tablet. However, these systems often fall short when specifying spatial target coordinates for robot navigation. The article [22] explores how to address this limitation by using a mobile robot designed for monoculture management. By leveraging a digital twin, Robot Operating System (ROS), and a mixed reality interface via the MS HoloLens, the study demonstrates how spatial target coordinates can be effectively managed. This integration of technologies introduces new possibilities for human–robot interfaces. This study provides interesting insights into the control of mobile robots in mixed reality. However, despite being a new study, it is based on an old version of ROS 1.

A framework for interactively controlling multiple robots with mixed reality interfaces is presented in [23]. The suggested method can work with single or multiple robotic systems, allowing for interaction with unmanned aerial vehicles (UAVs), mobile robots, and industrial manipulators. The robot may be interacted with both interactively and physically thanks to the human–robot interface. A special focus is placed on a visual representation of the robot's intentions during control. ROS is the primary component of the framework that is provided; it may be integrated with any robot or mixed reality interface. The article is very interesting and offers useful practices. Although it is a relatively new study, it is based on an old version of ROS 1. It also contains only limited information regarding user testing of the new HMI form.

The article [24] describes the design and the implementation of a Mixed Reality Server and proposes a communication design concept for the Care-o-Bot robot user interface using the rosbridge stack and the open-source meta-operating system/platform ROS. The article is from 2012 and, despite the title, does not contain mixed reality but deals with the first attempts to display augmented reality on Apple's iPad tablets.

The repository [25] implements glue which connects mixed reality robotics application with robots and infrastructure by using ROS 2. ROS 2 runs directly on the Hololens without going through ROS Bridge—reducing latency and improving network efficiency. However, the repository relies on "ROS2.net", and based on discussions on GitHub, it was found that "ROS2.net" does not work properly on the latest LTS version of Humble Hawksbill's ROS 2 and thus cannot be used effectively in the proposed solution.

1.3. Summary of State of the Art

Although there is a body of research focusing on linking ROS middleware with augmented or mixed reality, we observed a lack of research looking at newer versions of these technologies, specifically ROS 2 and HoloLens 2. It is important to note that the new version of ROS 2 introduces changes in its architecture and usage pattern compared

with the original version, which requires a specific approach when integrating it with augmented and mixed reality technologies. In addition, it is also notable that only a limited amount of research has addressed HoloLens integration, and when they have worked with this hardware, it is often limited to the original HoloLens version 1, not the newer HoloLens version 2. This lack of research leaves the field open for new innovations and experiments. And as it turns out, most studies focus on robotic arms and not mobile robots.

This work has merit, as it addresses this particular research gap. The integration of ROS 2 with mixed reality for HoloLens 2 brings new challenges and opportunities that differ from previous versions. By addressing this combination of technologies, we are opening the door for new ways for robots and humans to interact in the real world. This means combining state-of-the-art technology with the latest innovations, allowing us to achieve results that are not only unique but also fundamental to the future of robotics and augmented or mixed reality.

2. Materials and Methods

2.1. Robot Operating System

A robot's ability to act on user input or autonomously is enabled by software that is loaded into the robot. An easier programming of communication between hardware and software is made possible by the use of ROS. Since ROS carries the words operating system, this can be misleading. It is in fact not an operating system but an open-source framework and set of tools that provides operating system functionality on a heterogeneous computing cluster. Its usability is not limited to robots, but the bulk of the tools provided are aimed at working with peripheral hardware [26].

There are several software platforms, for example, MS Robotics Studio or OpenRTM, but it is unnecessary to compare ROS with them, because each of them is used for different purposes. The goal of ROS is to create a development environment that enables collaboration on a global level in the development of robotic software and that aims to maximize code reuse. ROS is characterized by the following features [27]:

- Distributed process: ROS is implemented as a set of minimally executable units (nodes), with each node operating independently and exchanging data in a systematic manner.
- Package management: Multiple nodes with similar functionality are managed as a package, allowing for easy use, development, sharing, modification. and distribution.
- API: When developing a program, ROS is designed to simply call the API and easily embed it into the code being used.
- Support for different programming languages: Since programming languages such as Python, C++, and C# are popular in the robotics field, ROS provides a client library that supports these languages.
- Public repository: Each package is published on a public repository (e.g., GitHub), and its license is listed.

2.1.1. Differences between ROS 1 and ROS 2

The differences between ROS 1 and ROS 2 should be noted. ROS was first released in 2007 by Open Robotics, with the intention of providing users with open-source software frameworks, libraries, and tools for robot development. ROS consists of nodes that communicate with each other via messages. The practicality of the nodes is based on their usefulness, as the nodes do not have to be on the same system or use the same architecture. Such flexibility makes ROS easily adaptable to user requirements [28].

The first version, ROS 1, was originally designed for academic applications. As hardware continued to evolve and become more affordable over time, the next version of this system was announced at ROSCon in 2014. The first distribution release of ROS 2 was published in 2017. Unlike its predecessor, ROS 2 was designed specifically for commercial projects.

The established end-of-life (EOL) date for the last version of ROS 1 Noetic is May 2025, after which Open Robotics will officially stop supporting it and move on to develop ROS 2 [29].

The ROS architecture consists primarily of an application layer, middleware, and an operating system. The basic difference between ROS versions is in the system architecture. The differences in the architectures are illustrated in Figure 3, where the primary differences between the versions are color-coded.

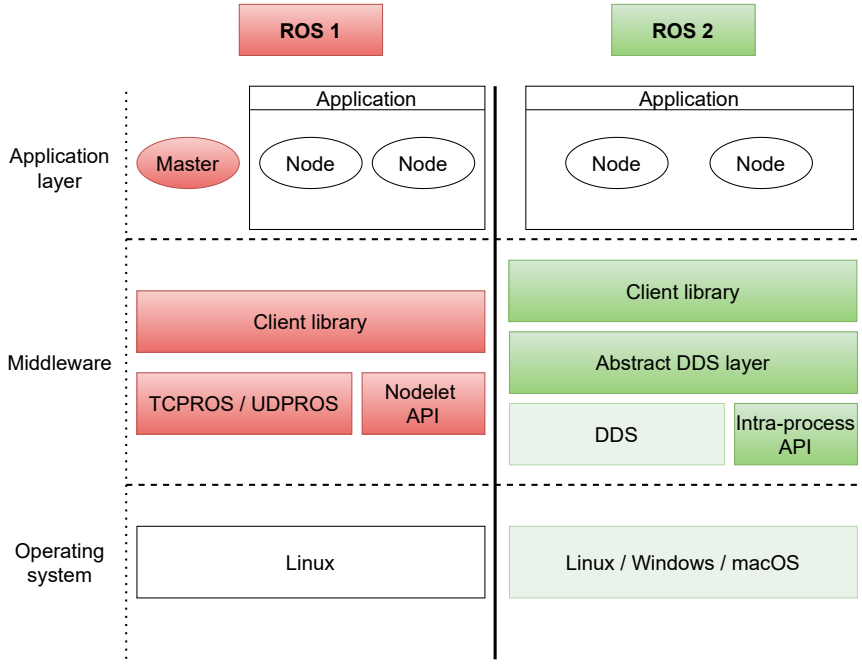


Figure 3. Differences between ROS 1 and ROS 2 architectures.

The ROS 1 implementation involves a TCPROS/UDPROS communication channel that requires a master, unique process in the distributed system. In contrast, ROS 2 is based on the Data Distribution Service (DDS) protocol and includes an abstract DDS layer that makes it unnecessary for the user to be aware of the DDS API. The abstract layer allows for high-level configuration and the optimization of DDS usage. In addition, DDS does not need a master process, which makes communication more fault-tolerant. ROS 1 contains a node called master, which acts as a name server for inter-node communication and message communication. Some communication types, such as topics or services, are not possible without a running master. ROS version 2 does not use a master node due to ROS, and instead, each node in the network can communicate with another node on its own, increasing the fault tolerance of the system [27,30].

In order to make ROS 2 suitable for commercial use and to provide performance and security improvements, other shortcomings had to be addressed. A summary of other differences between the ROS versions is shown in Figure 4 [31].

ROS 1	Category	ROS 2
Custom protocol built on TCP/UDP	**Network transport**	DDS protocol
Central name server (Master node)	**Architecture**	Peer-to-peer network
Linux	**Supported platforms**	Linux, Windows, macOS
Written independently in each language	**Client libraries**	Sharing common basic library
One node per process	**Nodes**	Multiple nodes per process
Minimum experimental support	**Embedded systems**	Commercially supported implementation
Auxiliary protocol	**Access to parameters**	Implemented using service calls
Derived type on assignment	**Types of parameters**	Declared and required type

Figure 4. Differences between ROS 1 and ROS 2.

2.1.2. Nodes and Messages

To develop a robot that uses ROS, it is necessary to understand the basic concepts and components of ROS. In this section, the basic concepts needed to understand the message communication offered by ROS are presented. Among these communication types, we distinguish communication into topics, services, and actions.

Node refers to the smallest unit of the processor running in ROS. It is recommended to create one node for each purpose for ease of reusability. A purpose can be understood as, for example, sensor data conversion, motor drive, navigation, etc. At startup, the node registers information such as name, message type, URI address, and port number. Based on the registered information, the registered node may act as a publisher, subscriber, service server, or service client [27].

Message: Data are sent and received between nodes via a message. A message can be a variable, such as an integer, floating point, string, or Boolean, or a nested structure that contains an array of messages. The data structure of a message is specified in a file in the .msg format [27].

2.1.3. Publish–Subscribe Communication

A topic is a named bus that provides asynchronous message exchange between nodes via topics. Topics use publish–subscribe functions that separate the creation of information from its reception. In general, nodes do not know who they are communicating with. If a node is interested in receiving messages from a topic, it subscribes to that topic. If we are talking about the subscribe function, the node in this case is called a subscriber. Conversely, nodes that create and publish messages to the corresponding topic use the publish function, and these nodes are called publishers. After publishing a message to a topic, each node that is subscribed to that topic accepts the published message. Depending on the purpose, a topic not only allows 1:1 communication between subscriber and publisher but also supports 1:N, N:1, and N:N communication. The communication is illustrated in the diagram (Figure 5).



Figure 5. Communication using topics.

Communication via topics is intended for unidirectional communication, and since nodes remain connected to continuously send and receive messages, they are suitable, for example, for sensor data that require the periodic publishing of messages [27,32].

2.1.4. Communication Using Services

In a distributed system, request-and-response interactions are often required; therefore, the publish–subscribe model with its one-way transmission is not always suitable for use. Communication through services uses a request-and-response model, which is defined by a pair of messages: one message for the request and one message for the response. The ROS node, the service server, offers a service named with a string data type. The node that sends the request message is called the service client. After sending a request, the client waits for a response. The communication is shown in the diagram (Figure 6).

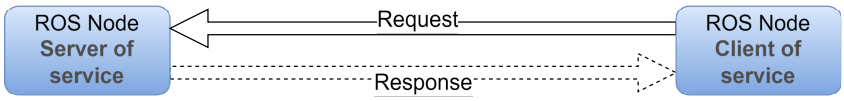


Figure 6. Communication using services.

When communicating through services, it is two-way synchronous communication. Unlike topic communication, this communication is a one-off, and once the request is completed and the response is received, the connection between the two nodes is broken. This is often used to issue a command to the robot to perform a specific action. Since no connection is maintained, using this communication provides a lower load on the network [27,33].

2.1.5. Communication Using Actions

If it takes a long time to meet the desired goal and feedback on progress is required, then action communication is used. This communication uses a model of goals and outcomes that resemble the service model. The difference is that regular feedback is sent to the client. The way the communication is performed is as follows: The ROS node, the action client, sends the action goal to the action server. The action server, which executes the specified action, sends feedback to the action client. The feedback messages are published to the topic. After the action is executed, a final message is sent to the action client, which is the result of the executed action. This type of communication is most often used by robots for navigation tasks. If the requirement for the robot changes, i.e., a task other than the one currently being performed needs to be performed, actions offer the possibility to cancel the first action and initiate a new action. The communication is illustrated in the diagram (Figure 7). Thus, communication via actions is two-way asynchronous communication [27,34].

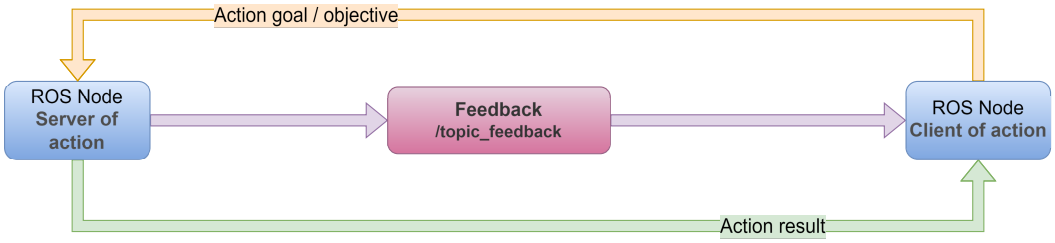


Figure 7. Communication using actions.

2.1.6. Visualization and Simulation Tools

Visualization and simulation of robots are among of the basic tools of every roboticist. A well-designed simulator allows you to quickly test algorithms, design robots, or train and test an artificial intelligence system by using realistic scenarios [35]. There are two development environments for this purpose, RViz and Gazebo.

RViz software is a powerful visualization tool provided by ROS that allows the user to view and interact with the selected robot [36].

Gazebo simulation software is an open-source 3D robot simulator that can simulate real-world physics. By using it, developers can design robots and test algorithms in a digital environment. This software's credibility and reliability have been achieved through Open Robotics' collaboration with the developer community. Each library in the Gazebo software has been designed for a specific purpose, which reduces code clutter, creates consistency among codes, and simplifies the search for solutions [35].

To demonstrate the described solution for controlling and monitoring a robotic device using mixed reality, a virtual model of the robotic device Turtlebot 3 was selected (Figure 8). Turtlebot3 is a popular open-source robot designed for research and education in the field of robotics. It uses the intelligent Dynamixel actuator by ROBOTIS for control. This robot is small, programmable, and affordable, aiming to significantly reduce the size of the platform without sacrificing quality and functionality. The robot's hardware can be modified in different ways depending on the use of optional parts, such as sensors [27].

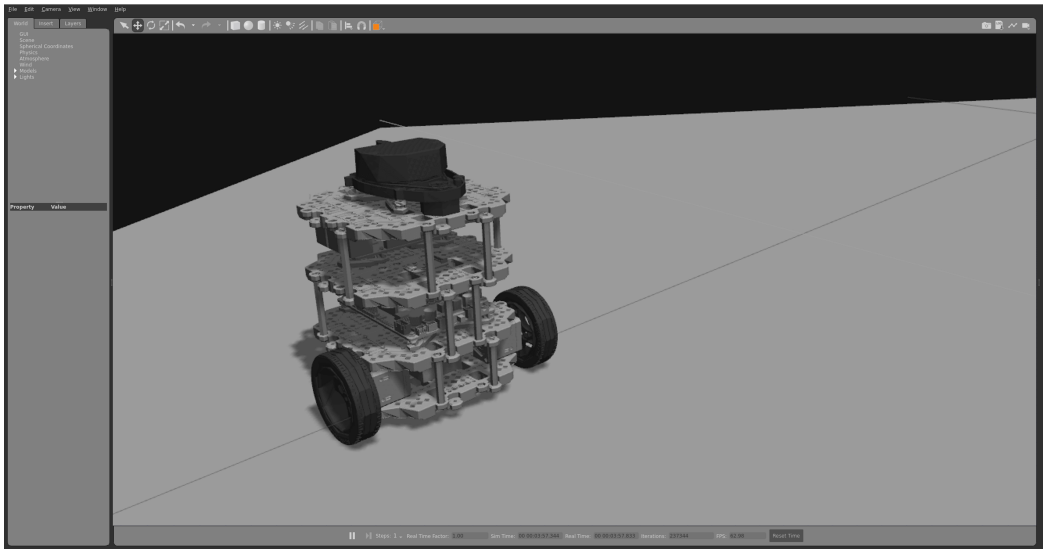


Figure 8. Virtual model of Turtlebot 3 in Gazebo environment.

2.2. Mixed Reality Development

A device designed to be worn on the head to provide a simulated visual environment for augmented or mixed reality is called a headset. Thus, for augmented or mixed reality, it provides a view of the real world to which it adds digitally generated objects for an augmented view. Examples of such devices include the Microsoft HoloLens 2. Headsets contain lenses made from an LCD display or other display mechanism, an embedded microprocessor, and memory [37].

Microsoft HoloLens 2 (Figure 2) is Microsoft's second-generation augmented/mixed reality headset. It is a high-tech device that combines the real world with digital objects by using holograms that allow you to interact with digital content in your environment. This headset is an improvement on its predecessor with several new features and enhancements, including the following:

- A larger field of view that allows the user to see and interact with digital content without having to turn around all the time;
- Better ergonomics, which means that it is lighter and better suited to head sizes for a more comfortable fit;
- An improved controller, or Microsoft HoloLens 2 Clicker, which allows for more precise control when interacting with holographic objects;
- Better performance, enabled by improved hardware that offers faster and smoother display and interaction with holograms [37].

For viewing and interacting with digital objects, HoloLens 2 uses a variety of features, including the following [38]:

- **Hand tracking**: HoloLens 2 adapts to the hands for more precise interaction, e.g., touching, grasping, and moving, with objects and holograms.
- **Voice support**: By using the built-in microphone, voice commands can be entered in multiple languages, allowing the user to quickly control the device without using their hands.
- **Eye tracking**: With this feature, the device adapts holograms to the user's eyes in real time.
- **Spatial mapping**: To better anchor digital objects to real objects and surfaces, the spatial mapping of the physical environment is used, as shown below in Figure 9.

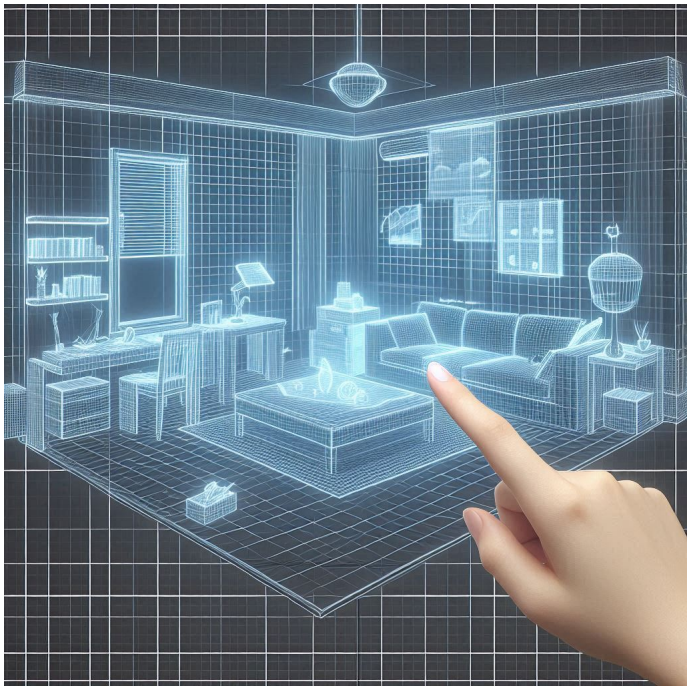


Figure 9. Illustration of room mapping using Microsoft HoloLens 2.

For easier development of mixed reality applications, Microsoft has created a toolkit called Mixed Reality Toolkit (MRTK), which provides game engines such as Unity or Unreal Engine with various configurations and modules for developing AR/MR applications and user interface for HoloLens 2 [38].

2.3. Summary

Based on the results of the analysis, Unity was chosen as the 3D engine because of its easier learning curve and Microsoft's support within the MRTK for MR application development. The MR application supports ROS 2 software and interacts with it through messaging. All types of communication are represented: topics, services, and actions. The Turtlebot 3 robot, which supports ROS 2 and is simulated by using Gazebo software, was selected as the mechatronic system suitable for demonstration purposes. The hardware used for design and testing was HoloLens 2.

3. Results

Based on the analysis and technological options, key factors were identified that were taken into account when designing the solution. The proposed solution includes system structure, component and technology selection, and implementation method. The main goal was to create a robust, flexible, and friendly solution that will enable the efficient control and monitoring of the robot and will serve as a basis for the implementation of

future developments. The application for HoloLens 2 will offer an intuitive user interface and gesture control.

3.1. Requirement Specification

The application system for interfacing ROS 2 with mixed reality should meet the following requirements:

- The application must be compatible with the HoloLens 2 headset.
- The 3D game engine Unity will be used to implement the application.
- The application will be implemented for mixed reality.
- Turtlebot 3, Burger model, will be used as the mechatronic system suitable for demonstration purposes.
- The virtual environment Gazebo will be used to simulate the robot.
- The ROS 2 meta-operating system will be used as middleware.
- The application will allow for the monitoring and control of the selected mechatronic system model.
- The interface of the application will be intuitive to use.
- Three types of communication will be used to illustrate the functionality of the interfacing:
  - Communication publish–subscribe (topics);
  - Communication via services;
  - Communication via actions.
- Communication between nodes is provided by a TCP/IP network connection.

The system should consist of two parts that will communicate with each other: backend and frontend (Figure 10).

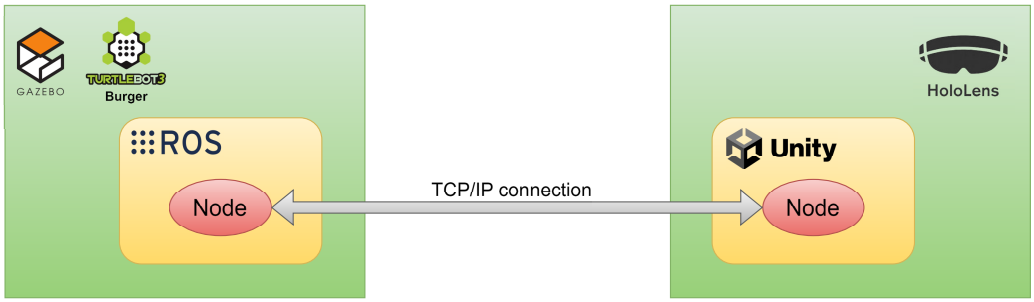


Figure 10. ROS 2 integration design with HoloLens 2 application.

The frontend part consists of an application created in Unity that was designed for use exclusively on the HoloLens 2 headset. The application displays a user interface through which the user can connect to the selected ROS by using its IP address, enter commands for the robotic device, and retrieve and monitor specified device values.

The backend part consists of the selected mechatronic system, Turtlebot 3, which can establish communication with the frontend part through ROS 2. The simulation of the robot was performed in the Gazebo virtual environment. The communication between ROS and Unity nodes is provided by a TCP/IP connection. The backend part was developed on the Linux operating system. Windows Subsystem for Linux (WSL) was used to create the Linux environment on the Windows operating system.

3.2. Graphical User Interface Designing

The first step in developing an augmented reality application is to design a user interface (UI) that should be intuitive and easy for the user to use. For the UI design, a freely available toolkit called MRTK by Microsoft was used; it contains objects, libraries, and interface elements to facilitate the development of an augmented reality application. Selected objects from MRTK were modified as needed.

The interface was divided into five sections, and the design of the sections is shown in Figure 11:

1. The first section allows for connection to the ROS device and displays the connection status;
2. The second section contains buttons (Forward, Reverse, Right, Left, and Stop) and sliders (speed input) to control the robot;
3. The third section allows for the monitoring and visualization of the data obtained from the robot;
4. The fourth section covers sending requests to the robot;
5. The fifth section allows for the customization of the interface appearance to the user's preferences, and there is color change and opacity for the UI.

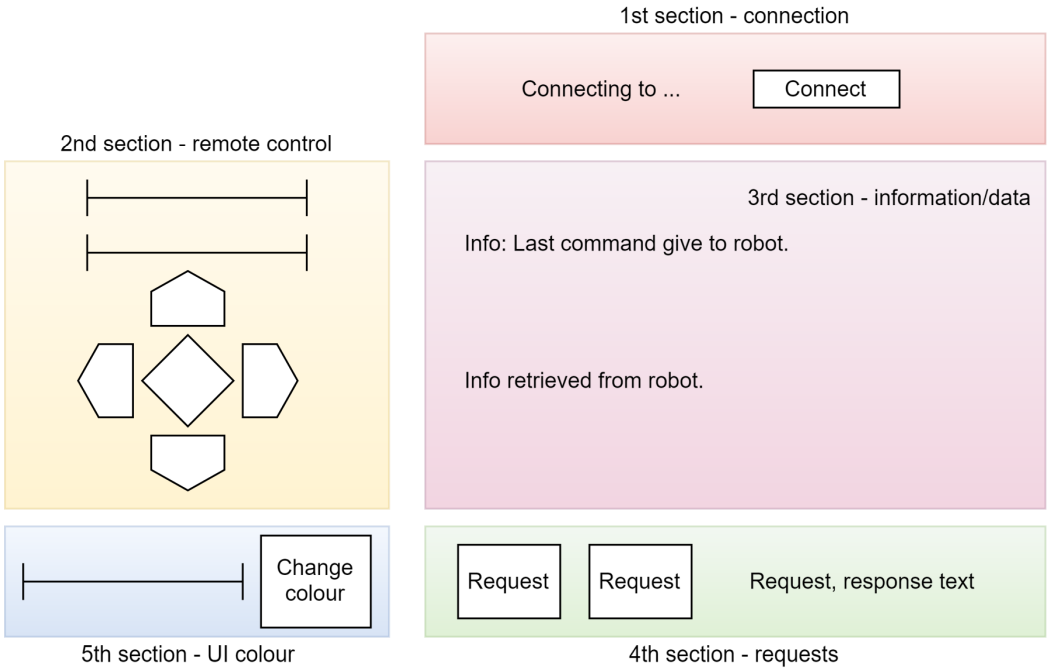


Figure 11. Design of graphical user interface.

3.3. Design of Communication among Nodes

The application node created for HoloLens 2 communicates with the ROS 2 node via messages. Since each type of communication has its advantages and disadvantages for certain applications, it is necessary to design the proper use of each communication type (Figure 12).

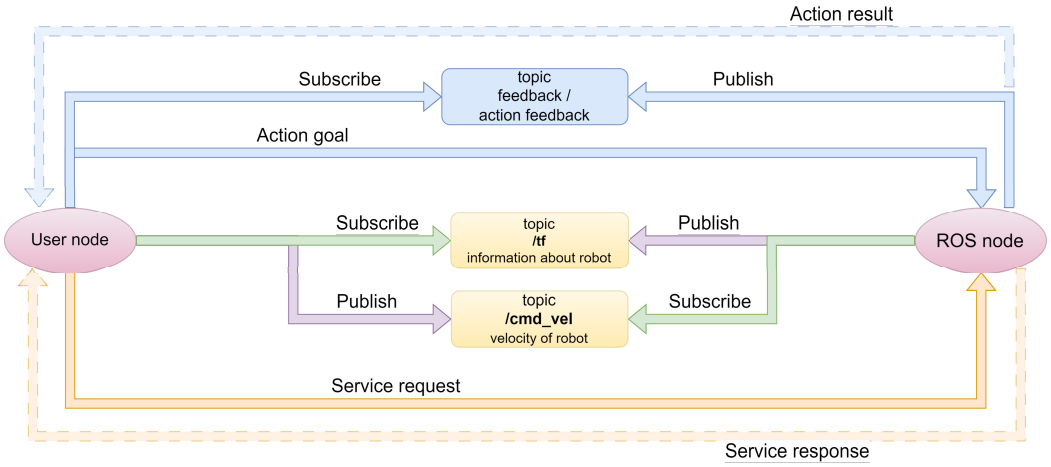


Figure 12. Design of communication among nodes.

The publish–subscribe communication is designed for one-way communication and is therefore suitable for continuously retrieving data from the robot. This communication is used in the third section on the UI that was described above. Also, because of the continuity and speed of the link, publish–subscribe is also used to send velocities for the robot's motion. The /tf and /cmd_vel topoids are used for this purpose.

The **topic /tf** (transform) provides information about the transformations of the robot, which includes information about the position, orientation, and interrelationships among different parts of the robot. The application node subscribes to this topic in order to obtain continuous information about the robot.

The **topic /cmd_vel** (velocity command) is a basic topic for robot motion control and is used to publish motion commands for the robot, where velocity commands are expressed as linear velocity expressed in m/s and angular velocity expressed in rad/s. The application node publishes user-specified velocities to this topic.

To illustrate the communication through services, the possibility of sending a request to a ready service from ROS and to a service created by us is created. Sending requests and receiving replies is implemented in the fourth section on the UI. ROS has **service /turtlebot3_laserscan/get_parameters**, which is used to obtain parameters from the laser. The application node sends a request to obtain the parameters.

Since ROS does not have predefined actions, to demonstrate communication via actions, it is necessary to create a custom action that the user can send a target to and receive feedback and results back from. Sending goals and receiving feedback and responses are implemented in the fourth section on the UI.

3.4. Implementation

After creating the system design and communication design and setting up the Unity project and the working environment for ROS, it was time for the implementation of the solution itself. The versions used are summarized in Table 1. The hardware parameters of the used computer and HoloLens device are stated in Tables 2 and 3.

Table 1. Software versions.

Software	**Version**
Operating system	Windows 10
Unity Editor	2022.3.12f
MS Mixed Reality Feature Toolkit	1.0.2209.0
MS Visual Studio	Community 2022
Mixed Reality OpenXR Plugin	1.7.2
Windows Subsystem for Linux	v2
Ubuntu	22.04.3 LTS
ROS2	Humble
Gazebo	11.10.2
ROS TCP Connector	0.7.0

Table 2. Hardware used for development.

Hardware	**Version**
Processor	Intel Core i5 (7th generation)
Operating memory (RAM)	16 GB
Graphics card	NVIDIA GeForce GTX 1050 (laptop version)

Table 3. Parameters of HoloLens 2 [39].

Parameter	Specification
Display	See-through holographic lenses (waveguides)
Resolution	2 k 3:2 light engines
Holographic density	>2.5 k radiants (light points per radian)
Eye-based rendering	Yes
Sensors	Depth sensor, 5-channel microphone array, accelerometer, gyroscope, magnetometer, 4 visible light cameras for head tracking, and 2 IR cameras for eye tracking
Head tracking	4 visible light cameras
Eye tracking	2 IR cameras
Processor	Qualcomm Snapdragon 850
Memory	4 GB LPDDR4x system DRAM
Storage	64 GB UFS 2.1
Connectivity	Wi-Fi: 802.11ac 2 × 2, Bluetooth 5.0, and USB Type-C
Operating system	Windows Holographic, version 2004
Weight	566 g
Battery life	2–3 h of active use

Currently, there are two options for linking ROS 2 with the Unity application:

- Package ROS2 For Unity by RobotecAI [40];
- Package ROS TCP Connector [41] by Unity Technologies.

Due to the incompatibility of the ROS2 For Unity and Unity Editor versions, we decided to use ROS TCP Connector, which contains four main features:

- Component ROSConnection, which provides communication between ROS and Unity;
- Tool to generate messages as classes in C# for ROS message types;
- Default configurations for visualizing ROS information;
- A set of extensions that convert geometry between Unity and other coordinate frameworks [41].

The control or sensing of hand gestures is provided by MRTK and OpenXR plugin. The most difficult gesture to perform is the air tap (the execution of the gesture is shown in Figure 13). This gesture is used in the application to grab an object or to zoom in and out of an object. To ensure proper user interaction with the UI objects, pre-made objects from the installed MRTK packages were used. The objects were modified as needed. The interaction includes grabbing the object, zooming in and out, or moving it to another location. A script for the UI objects was also used to activate the user's follow state. This script is activated by tapping the button with the paperclip symbol. Each UI panel is separate, which means that while one panel can follow the user, another panel can be statically in place.

The user interface designed in Figure 11 served as a basis for its implementation. Five parts for the UI were implemented.

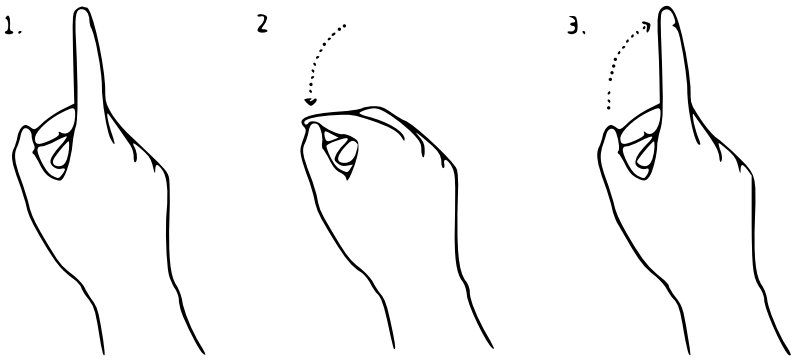


Figure 13. Air tap gesture demo for HoloLens 2.

Section 1 is used to connect an application node to an ROS 2 node. Node connections are made by using the ROSConnector script, which must be connected to at least one object in the scene. If the nodes are not connected, two input fields are displayed (Figure 14). The user taps on the appropriate field and is presented with a keyboard to enter the IP address and port into the field. The "Connect to ROS" button launches the script to connect to ROS. If the Unity application node is successfully connected to the ROS 2 node, the subscribe–publish methods for topic communication are automatically triggered. The script treats the port input field to make the specified string a number. If the *Connect()* method fails to connect to the node, the user is notified by a dump and can try to enter the IP address or port again. If the method succeeds in connecting to the node, the connection panel disappears, and the ROS connection information panel is displayed (Figure 15).

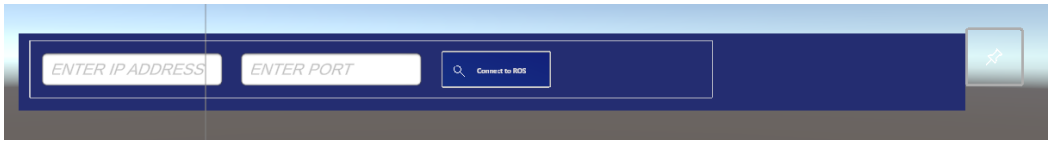


Figure 14. Section 1 on UI—input fields.

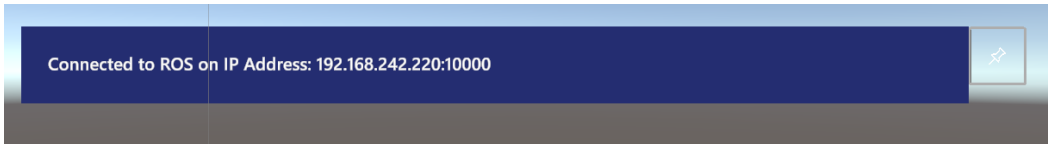


Figure 15. Section 1 on UI—connection information.

Section 2 is used to control the connected robot in the simulation. The panel contains two sliders: one to determine the linear velocity and the other to determine the angular velocity. Both velocities have a default setting of 0, with a maximum value of 1. The panel also contains buttons, the functionality of which is described below each one. The buttons can be used to move the robot forward, backward, right, and left. The last button, the middle one, stops the robot. The panel is shown in Figure 16.

Section 3 (Figure 17) contains a text listing for the last command given to the robot and other text listings for information about the robot. The information that is retrieved from the robot is as follows:

- Position;
- Orientation;
- Velocity—linear and angular.

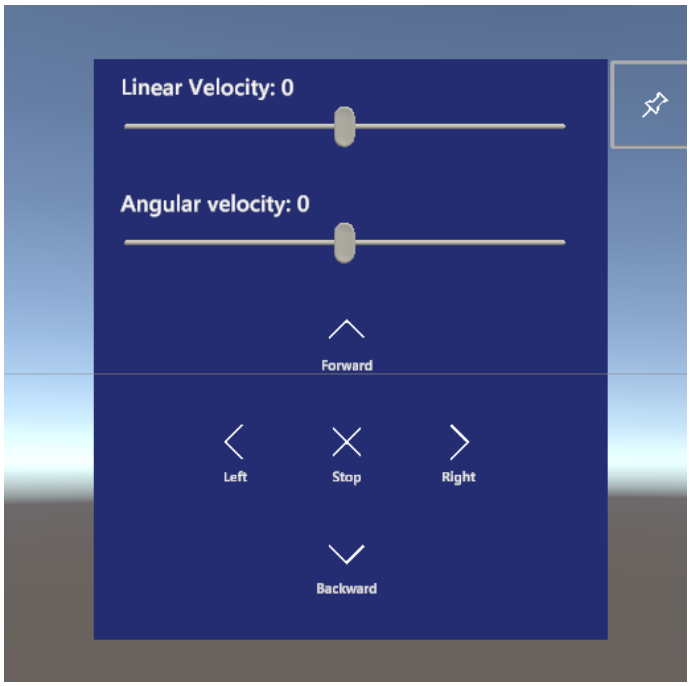


Figure 16. Section 2 on UI—buttons and sliders to move the robot.



Figure 17. Section 3 on UI—text printouts of information about the robot.

Section 4 (Figure 18) contains buttons for sending requests to services and sending the target for an action. The first and second buttons send the request for the service, and the third button sends the target for the action. By using the "Increase radius" and "Decrease radius" buttons, the user specifies the radius value to be sent along with the action target (goal). There is also a text box on the panel that displays the service response or feedback and the action response.

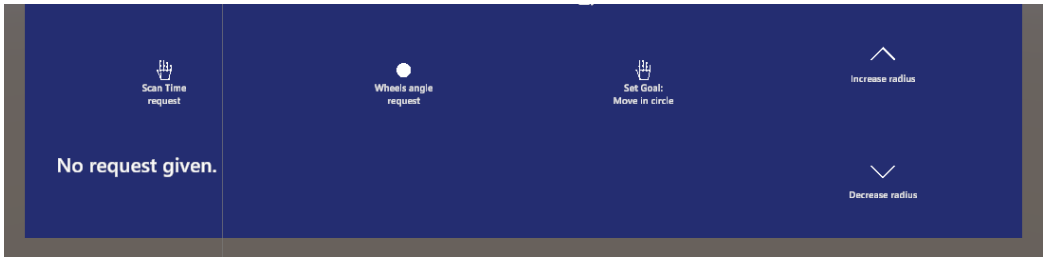


Figure 18. Section 4 on UI—sending requests and targets/goals.

Section 5 (Figure 19) contains a button that allows the user to change the color of the UI according to their own preference. The colors are predefined and pressing the button iterates between them. By using the slider, the user can adjust the opacity of the UI.

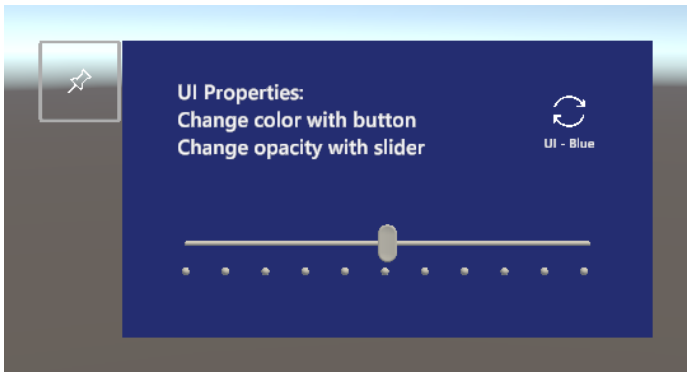


Figure 19. Section 5 on UI—UI design (color).

The ROS and Unity connection is available by using ROS TCP Connector. The server's TCP endpoint is run as an ROS node, which facilitates the transfer of messages from Unity to ROS and vice versa from ROS to Unity (shown in Figure 20). ROS expects serialized messages, so the ROS TCP Connector repository includes a message generation module "MessageGeneration" that allows messages to be generated as C# classes and offers serialization and deserialization functions for .msg files. The ROSConnection module provides the necessary Unity scripts for the publish–subscribe methods, as well as functions for calling services.

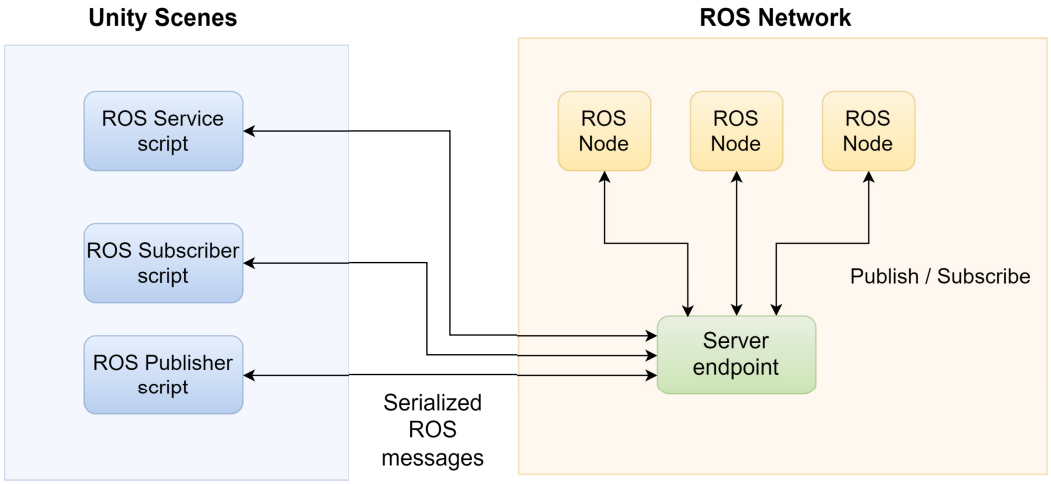


Figure 20. Sending messages by using ROS TCP Connector.

4. Discussion

The output is a functional mixed reality application called "ROS2ControllerApp" that is compatible with the HoloLens 2 headset. The application allows for the monitoring and control of the selected mechatronic/robotic system and data visualization. The 3D game engine Unity was used in the implementation of the application. Interaction with objects and gesture capture are implemented by using Microsoft's MRTK module.

The application communicates with ROS version 2 through messages. The communication types used are publish–subscribe communication and communication via services and actions. The working environment for ROS 2 is implemented on the Linux operating system, which is implemented on the Windows operating system through Windows Subsystem for Linux (WSL).

TurtleBot 3, Burger model, a robot that supports ROS 2, was selected as the virtual mechatronic system suitable for demonstration purposes. The robot was simulated by using the Gazebo simulation software.

The communication between the Unity application and ROS 2 is provided by ROS TCP Connector, which mediates messages between nodes. Custom packages were created for ROS 2, which can also be used to communicate through services and actions.

The following are links to videos where you can see one aspect of the application system being tested:

- UI panels—https://youtu.be/oFmZZGJhHJY (accessed on 6 July 2024).
- UI design—https://youtu.be/Gh0B_lNP3FQ (accessed on 6 July 2024).
- Robot control—https://youtu.be/BqPlZtHbmN8 (accessed on 6 July 2024).
- Robot information—https://youtu.be/rjoIXPL71nk (accessed on 6 July 2024).
- Services—https://youtu.be/Hz0qAguXTic (accessed on 6 July 2024).
- Action—https://youtu.be/p5YZoMU49Ao (accessed on 6 July 2024).

The solution was tested by four people in the workplace (one female and three males), all four of whom had previous experience with augmented/mixed reality on smartphone and tablet. A previously published solution for the monitoring and control of mechatronic systems using augmented reality displayed on a smartphone/tablet screen was developed at the researchers' workplace [3]. The mixed reality experience in the headset was evaluated by these individuals as superior to the smartphone-/tablet-based approach. The reasons for this preference included enhanced immersion, as the headset allowed for a more integrated and engaging experience. The hands-free operation provided by the headset significantly improved work efficiency, as users could interact with the system without the need to hold a device. Additionally, the direct overlay of information within the user's field of view improved situational awareness and allowed for more intuitive control and monitoring of the robotic systems. The improved ergonomics and comfort of using the headset, especially in scenarios requiring prolonged usage, also contributed to the overall positive assessment. However, the current weight of the device may pose a challenge for extended use.

To build on the results obtained from our initial testing, it is important to underscore the necessity of verifying the system's basic functionality and ensuring its proper operation in a real-world setting. The current tests were crucial to confirming that the MR interface, designed for robot control and monitoring, performs as expected when integrated with the ROS 2 framework. As we move forward, future research could focus on a more in-depth evaluation of the ergonomics of the user interface (UI) and the HMI aspects of the system.

This could involve collaboration with experts in user experience (UX) design to refine the interface's usability and comfort, especially in high-stress environments where prolonged usage might be required. Additionally, aspects such as gesture recognition accuracy, and the physical strain imposed by the MR headset could be areas of further study. User experience assessments could also explore the impact of varying interface designs on task performance and user satisfaction, potentially involving psychologists to evaluate the cognitive load and stress factors associated with extended MR usage.

This article is the first contribution to a larger project being developed at the authors' workplace. The project focuses on new HMI methods for workers operating in hazardous environments (space, smokey environments, and radioactive environments). These workers wear spacesuits, exoskeletons, or other thick suits with gloves that prevent them from effectively using smartphones or tablets to control mobile robotic devices (rovers, firefighting robots, etc.). In such situations, gesture control is offered as an effective method. Current methods for gesture-based, non-contact human–machine interaction often use camera systems that sense the human. These methods are based on 2D RGB image recognition or 3D depth data. The disadvantage is that the human operator has to be in the field of view of the camera system and in the correct rotation at all times. RGB cameras are prone to error in poor lighting or with different skin tones. Three-dimensional depth cameras can be prone to error if the operator has specialized work clothing that changes the proportions of the human body [42]. Motion capture-based body sensing technology (MoCap) partially or completely eliminates these flaws. State-of-the-art MoCap suits incorporate small inertial sensors that can continuously record full-body positions with high accuracy without a human operator in a dedicated field of view. The high accuracy of the inertial sensors

allows for the continuous manual control of variables such as acceleration/deceleration [43]. These MoCap suits are foldable and can also be used in rugged specialized work suits, such as pyrotechnic suits, exoskeletons, chemical suits, radiation suits, space suits, and refractory suits. Alternatively, inertial sensors can be incorporated directly into them.

If a worker is going to control a mobile robot by using a MoCap suit, the question arises as to how the worker will monitor the diagnostic variables sent by this robot. Alternatively, where will he/she track whether the gesture that they performed was correctly recognized and interpreted? As mentioned earlier, a tablet or other screen may not be the appropriate route. The proposed concept is thus unique, as it focuses on mixed reality displayed not on a smartphone or tablet display but on a selected headset. Thus, the innovation lies in the fact that the methodology is designed for a mixed reality headset and not a conventional mobile device (smartphone or tablet), which also provides the worker with greater immersion and a better overview of the variables being monitored. While using the headset, the worker's hands remain free, which increases both work efficiency and safety at work. Such a headset can be integrated under the helmet of an astronaut, firefighter or other worker, with diagnostic information being displayed directly on the headset.

The integration of mixed reality and tracking technology will offer workers greater immersion, better overview, and the ability to keep their hands free to create gestures for subsequent device control. Motion tracking technologies will improve user experience and efficiency when the worker interacts with the device by making immersive not only perception but also physical interaction in the form of body movement.

A schematic of the proposed solution, the further implementation of which will be the subject of further research, can be seen in Figure 21. A brief description follows:

1. A worker with a MoCap suit performs the gesture.
2. The motion data are processed by a software kit for the motion capture suit (depending on the specific type of suit).
3. The motion data are evaluated by a computational intelligence application (processing application for gesture recognition), and the gesture is recognized.
4. Based on the recognized gesture, a command is sent to the ROS 2-based mobile robotic system.
5. The mobile robotic system based on ROS 2 sends diagnostic information to the MR headset, which is integrated under the worker's helmet.
6. In the MR application, the worker can monitor diagnostic information or monitor/confirm the correctness of the recognized control gestures.

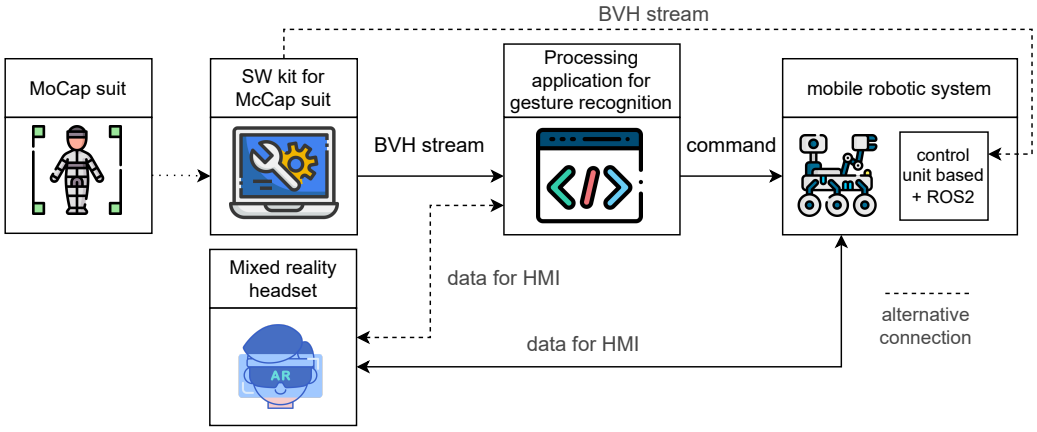


Figure 21. Scheme of proposed solution—upcoming research.

The integration of the two project lines (MoCap control and mixed reality in HoloLens) will create a functional polygon where a novel method of interacting with robotic devices, using body position sensing using MoCap technology, will be coupled with a mixed reality graphical user interface via a headset. This will enable efficient and touch-free

control and the monitoring of mechatronic systems—particularly with a focus on robotic platforms. Operators will wear a MoCap suit that will continuously and with high precision record their movements, enabling manual control of various variables when working with mechatronic devices. They will also use a mixed reality headset, and the project will develop versatile and advanced techniques for creating graphical user interfaces for controlling and driving robotic platforms. This integration will offer workers greater immersion, ergonomics, comfort, better visibility, and hands-free working, resulting in increased efficiency and safety in industrial environments. An example of a specific case study where this integration can be leveraged is working with robotic systems in pyrotechnic suits, where the MoCap suit will provide the operator with freedom and safety when manipulating equipment, and the mixed reality in the headset will allow for the intuitive condition monitoring and diagnostics of mechatronic processes.

In order to tackle this complex task, it was necessary to break it down into sub-tasks. The aspect of motion capture was previously addressed in an earlier article [44], where we delved into the intricacies of capturing and processing motion data to ensure accurate and reliable gesture recognition. This foundational work laid the groundwork for integrating motion capture technology with robotic control systems.

In the current study, we aimed to build upon this foundation by focusing on the sub-task of integrating mixed reality with ROS 2. Our objective was to create a seamless connection between the mixed reality environment and the ROS 2 framework, enabling the effective monitoring and control of robotic systems. This integration not only enhances user immersion and interaction but also ensures precise and real-time communication between the mixed reality application and the robotic platforms, thereby improving overall system efficiency and functionality.

5. Conclusions

In conclusion, the coupling of ROS 2 with MR technology through devices like the HoloLens 2 represents a significant leap forward in the ergonomics and safety of industrial operations. This approach not only keeps workers' hands free for essential tasks but also protects them from environmental hazards, aligning with the human-centric principles of Industry 5.0. As industries continue to evolve, such innovations will play a pivotal role in ensuring that technological advancements serve to enhance human capabilities while safeguarding their health and well-being.

The application benefits of the article can be summarized in the following points:

- **Creating a mixed reality application for Robot Operating System 2.**
  Despite the fact that the older version of ROS, ROS 1, is currently on its way out and will soon be discontinued, there are few applications that link augmented or mixed reality with ROS 2. Therefore, a demonstration application was developed within the framework of the described project. This application demonstrated all types of communication among ROS nodes—topics (publish–subscribe), actions, and also services. Thus, the presented methodology can be replicated for a larger number of scenarios.
- **Human–machine interface in mixed reality implemented through a headset and not by using a mobile device (tablet or smartphone).**
  Augmented and mixed reality in HMI and industrial applications is often handled by mobile devices such as mobile phones and tablets. Their advantage is lower cost, but their use does not leave the operator's hands free. Applications using headsets are still few. Therefore, a method and an application system that use the modern Microsoft HoloLens 2 headset instead of a mobile device have been introduced. In the case of its use, the operator's hands remain free for other tasks. This solution thus provides better ergonomics and safety.
- **Methodology and application system support inclusiveness within the Industry 5.0 concept.**
  Moreover, the results support inclusiveness within the Industry 5.0 concept. By

leveraging mixed reality technology, we can create a more accessible and adaptable working environment that accommodates a diverse workforce. This inclusive approach ensures that technological advancements benefit all users, regardless of their physical capabilities, and promotes a collaborative and supportive industrial ecosystem. This inclusiveness aligns with the core tenets of Industry 5.0, which emphasize the importance of human-centric and sustainable industrial development.

In conclusion, the results are beneficial and can be further modified for development of new forms of HMI and technical practice in Industry 4.0 and 5.0.

Author Contributions: E.K. and E.V. proposed the idea in this paper and prepared the software application; O.H. and D.J. designed the experiments, E.V. performed the experiments; O.H., D.J. and D.R. analyzed the data; E.K. and E.V. wrote the paper; O.H., D.J. and D.R. edited and reviewed the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Scientific Grant Agency of the Ministry of Education, Research, Development and Youth of the Slovak Republic No. 1/0107/22 and No. 1/0637/23 and by the Slovak Agency for Research and Development—grant No. APVV-23-0123.

Data Availability Statement: The datasets used and analyzed during the current study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cárdenas-Robledo, L.A.; Hernández-Uribe, Ó.; Reta, C.; Cantoral-Ceballos, J.A. Extended reality applications in industry 4.0.–A systematic literature review. *Telemat. Inform.* **2022**, *73*, 101863. [CrossRef]
2. Revolutionize Manufacturing with Augmented Reality Solutions. Available online: https://www.rockwellautomation.com/en-us/products/software/factorytalk/innovationsuite/augmented-reality.html (accessed on 8 May 2023).
3. Stark, E.; Kučera, E.; Haffner, O.; Drahoš, P.; Leskovský, R. Using Augmented Reality and Internet of Things for Control and Monitoring of Mechatronic Devices. *Electronics* **2020**, *9*, 1272. [CrossRef]
4. The Foundry. VR? AR? MR? Sorry, I'm Confused. 2017. Available online: https://www.foundry.com/insights/vr-ar-mr/vr-mr-ar-confused (accessed on 2 June 2020).
5. Gilmore, J.N.; Blair, B.W. Make Room for VR: Constructing Domestic Space and Accessibility in Virtual Reality Headset Tutorials. *Telev. New Media* **2024**, 15274764241232333. [CrossRef]
6. Bucsai, S.; Kučera, E.; Haffner, O.; Drahoš, P. Control and monitoring of IoT devices using mixed reality developed by unity engine. In Proceedings of the 2020 Cybernetics & Informatics (K&I), Velke Karlovice, Czech Republic, 29 January–1 February 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.
7. Milgram, P.; Kishino, F. A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* **1994**, *77*, 1321–1329.
8. Macenski, S.; Foote, T.; Gerkey, B.; Lalancette, C.; Woodall, W. Robot operating system 2: Design, architecture, and uses in the wild. *Sci. Robot.* **2022**, *7*, eabm6074. [CrossRef] [PubMed]
9. DiLuoffo, V.; Michalson, W.R.; Sunar, B. Robot Operating System 2: The need for a holistic security approach to robotic architectures. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881418770011. [CrossRef]
10. Soares, I.; Sousa, R.B.; Petry, M.; Moreira, A.P. Accuracy and repeatability tests on HoloLens 2 and HTC Vive. *Multimodal Technol. Interact.* **2021**, *5*, 47. [CrossRef]
11. Stothard, P.; Squelch, A.; Stone, R.; Van Wyk, E. Towards sustainable mixed reality simulation for the mining industry. *Min. Technol.* **2019**, *128*, 246–254. [CrossRef]
12. Battini, D.; Berti, N.; Finco, S.; Zennaro, I.; Das, A. Towards industry 5.0: A multi-objective job rotation model for an inclusive workforce. *Int. J. Prod. Econ.* **2022**, *250*, 108619. [CrossRef]
13. Verma, A.; Bhattacharya, P.; Madhani, N.; Trivedi, C.; Bhushan, B.; Tanwar, S.; Sharma, G.; Bokoro, P.N.; Sharma, R. Blockchain for Industry 5.0: Vision, Opportunities, Key Enablers, and Future Directions. *IEEE Access* **2022**, *10*, 69160–69199. [CrossRef]
14. Gášová, M.; Gašo, M.; Štefánik, A. Advanced industrial tools of ergonomics based on Industry 4.0 concept. *Procedia Eng.* **2017**, *192*, 219–224. [CrossRef]
15. Huang, S.; Wang, B.; Li, X.; Zheng, P.; Mourtzis, D.; Wang, L. Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *J. Manuf. Syst.* **2022**, *64*, 424–428. [CrossRef]
16. Shamaine, C.X.E.; Qiao, Y.; Henry, J.; McNevin, K.; Murray, N. RoSTAR: ROS-based telerobotic control via augmented reality. In Proceedings of the 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), Tampere, Finland, 21–24 September 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–6.
17. Cruz Ulloa, C.; del Cerro, J.; Barrientos, A. Mixed-reality for quadruped-robotic guidance in SAR tasks. *J. Comput. Des. Eng.* **2023**, *10*, 1479–1489. [CrossRef]

18. Chen, H. Motion Control of Interactive Robotic Arms Based on Mixed Reality Development. *arXiv* **2024**, arXiv:2401.01644.
19. Calderón-Sesmero, R.; Duque-Domingo, J.; Gómez-García-Bermejo, J.; Zalama, E. Development of a Human–Robot Interface for Cobot Trajectory Planning Using Mixed Reality. *Electronics* **2024**, *13*, 571. [CrossRef]
20. Soares, I.; Petry, M.; Moreira, A.P. Programming Robots by Demonstration Using Augmented Reality. *Sensors* **2021**, *21*, 5976. [CrossRef]
21. Carriero, G.; Calzone, N.; Sileo, M.; Pierri, F.; Caccavale, F.; Mozzillo, R. Human-Robot Collaboration: An Augmented Reality Toolkit for Bi-Directional Interaction. *Appl. Sci.* **2023**, *13*, 11295. [CrossRef]
22. Wittenberg, C.; Bauer, B.; Schloer, N. Mixed reality control of a mobile robot via ROS and digital twin. In Proceedings of the Human Factors in Robots, Drones and Unmanned Systems, AHFE International, San Francisco, CA, USA, 20–24 July 2023. [CrossRef]
23. Ostanin, M.; Yagfarov, R.; Devitt, D.; Akhmetzyanov, A.; Klimchik, A. Multi robots interactive control using mixed reality. *Int. J. Prod. Res.* **2021**, *59*, 7126–7138. [CrossRef]
24. Chivarov, N.; Penkov, S.; Angelov, G.; Radev, D.; Shivarov, N.; Vladimirov, V. Mixed Reality Server and Remote Interface Communication for ROS Based Robotic System. *Int. J. Autom. Austria* **2012**, *20*, 144–155.
25. Microsoft. ROS and Microsoft Mixed Reality Toolkit (MRTK) Integration. 2023. Available online: https://github.com/ms-iot/ros_msft_mrtk (accessed on 13 August 2024).
26. Ademovic, A. An Introduction to Robot Operating System: The Ultimate Robot Application Framework. Available online: https://www.toptal.com/robotics/introduction-to-robot-operating-system (accessed on 3 September 2024).
27. Pyo, Y.; Cho, H.; Jung, R.; Lim, T. *ROS Robot Programming: A Handbook*, 1st ed.; ROBOTIS Co., Ltd.: Seoul, Republic of Korea, 2017.
28. Cappellari, T. The ROS 1 vs. ROS 2 Transition. Available online: https://www.swri.org/industry/industrial-robotics-automation/blog/the-ros-1-vs-ros-2-transition (accessed on 27 April 2024).
29. Venkatadri, A. ROS 1 vs. ROS 2 What Are the Biggest Differences? Available online: https://www.model-prime.com/blog/ros-1-vs-ros-2-what-are-the-biggest-differences (accessed on 11 April 2024).
30. Maruyama, Y.; Kato, S.; Azumi, T. Exploring the performance of ROS2. In Proceedings of the EMSOFT '16: 13th International Conference on Embedded Software, Pittsburgh, PA, USA, 1–7 October 2016; pp. 1–10. [CrossRef]
31. Elmofty, O. ROS2 vs. ROS1— Key Differences and Which One Is Better? Available online: https://www.medium.com/@oelmofty/ros2-how-is-it-better-than-ros1-881632e1979a (accessed on 11 April 2024).
32. Topics. Available online: https://wiki.ros.org/Topics (accessed on 27 April 2024).
33. Services. Available online: https://wiki.ros.org/Services (accessed on 27 April 2024).
34. Weon, E. Creating ROS 2 Actions. Available online: https://www.foxglove.dev/blog/creating-ros2-actions (accessed on 27 April 2024).
35. Gazebo. Available online: https://classic.gazebosim.org/#gsc.tab=0&gsc.q=citation&gsc.page=1 (accessed on 27 April 2024).
36. RViz UserGuide. Available online: https://wiki.ros.org/rviz/UserGuide (accessed on 27 April 2024).
37. Rouse, M. Augmented Reality Headset. Available online: https://www.techopedia.com/definition/23143/augmented-reality-headset-ar-headset (accessed on 27 April 2024).
38. HoloLens 2. Available online: https://www.microsoft.com/en-us/hololens/hardware#document-experiences (accessed on 27 April 2024).
39. About HoloLens 2. Available online: https://learn.microsoft.com/en-us/hololens/hololens2-hardware (accessed on 7 August 2024).
40. Ros2 for Unity. Available online: https://github.com/RobotecAI/ros2-for-unity (accessed on 27 April 2024).
41. ROS TCP Connector. Available online: https://github.com/Unity-Technologies/ROS-TCP-Connector (accessed on 27 April 2024).
42. Regazzoni, D.; Vecchi, G.D.; Rizzi, C. RGB cams vs. RGB-D sensors: Low cost motion capture technologies performances and limitations. *J. Manuf. Syst.* **2014**, *33*, 719–728. [CrossRef]
43. Simonetto, M.; Arena, S.; Peron, M. A methodological framework to integrate motion capture system and virtual reality for assembly system 4.0 workplace design. *Saf. Sci.* **2022**, *146*, 105561. [CrossRef]
44. Haffner, O.; Kučera, E.; Beňo, L.; Pribiš, R.; Pajpach, M.; Janecký, D. Gesture Recognition System Based on Motion Capture Suit and LSTM Neural Network. In Proceedings of the 2023 International Conference on Modeling, Simulation & Intelligent Computing (MoSICom), Dubai, United Arab Emirates, 7–9 December 2023; pp. 71–76. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.